

12-2021

Adaptive Navigation of Three-Dimensional Scalar Fields with Multiple UAVs

Robert Ka-Hing Lee

Follow this and additional works at: https://scholarcommons.scu.edu/eng_phd_theses



Part of the [Mechanical Engineering Commons](#)

Recommended Citation

Lee, Robert Ka-Hing, "Adaptive Navigation of Three-Dimensional Scalar Fields with Multiple UAVs" (2021). *Engineering Ph.D. Theses*. 37.

https://scholarcommons.scu.edu/eng_phd_theses/37

This Dissertation is brought to you for free and open access by the Student Scholarship at Scholar Commons. It has been accepted for inclusion in Engineering Ph.D. Theses by an authorized administrator of Scholar Commons. For more information, please contact rscroggin@scu.edu.

Santa Clara University

Department of Mechanical Engineering

Date: December 10, 2021

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY

Robert Ka-Hing Lee

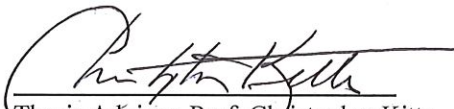
ENTITLED

Adaptive Navigation of Three-Dimensional Scalar Fields with Multiple UAVs

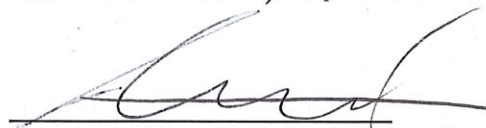
BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE

OF


DOCTOR OF PHILOSOPHY IN MECHANICAL ENGINEERING




Thesis Advisor: Prof. Christopher Kitts




Chairman of Department: Prof. Hohyun Lee



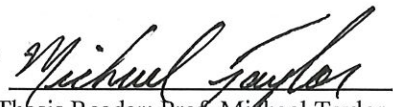
Thesis Reader: Prof. Yi Fang



Thesis Reader: Prof. Robert Marks



Thesis Reader: Dr. Michael Neumann



Thesis Reader: Prof. Michael Taylor

Adaptive Navigation of Three-Dimensional Scalar Fields with Multiple UAVs

By

Robert Ka-Hing Lee

Dissertation

Submitted in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
in Mechanical Engineering
in the School of Engineering at
Santa Clara University, 2021

Santa Clara, California

*Dedicated to my family who has supported me through my educational
journey spanning nearly two decades*

Acknowledgements

I wish to express my sincere appreciation to our Father in Heaven for providing me with the opportunity to attend Santa Clara University and the persistence throughout this educational endeavor.

Thanks to my academic advisor Professor Christopher A. Kitts and Research Associate Dr. Michael A. Neumann for their patience, guidance, and insight which have been crucial along this journey.

In addition, I want to thank my family for supporting me throughout this educational quest that has spanned nearly two decades. Thank you to my wife Annie, parent in-laws (Mr Chee Ling Li. and Mrs. Betty Li), and my children (Derrick, Angelica, and Alison). Without their support, I would not have been able to attain this goal that seemed beyond reach just a few years ago.

This work builds upon previous research and development efforts at Santa Clara University's Robotic Systems Laboratory (SCU RSL). As a requirement for completion of the degree, excluding sections 6.5 and 6.6, the work presented in this dissertation was published as a journal article in Lee et al. [76].

Adaptive Navigation of Three-Dimensional Scalar Fields with Multiple UAVs

Robert Ka-Hing Lee

Department of Mechanical Engineering
Santa Clara University
Santa Clara, California
2021

Abstract

Adaptive Navigation (AN) control strategies allow an agent to autonomously alter its trajectory based on realtime measurements of its environment. Compared to conventional navigation methods, AN techniques can potentially reduce the time and energy needed to explore scalar characteristics of unknown and dynamic regions of interest (e.g., temperature, concentration level). Multiple Uncrewed Aerial Vehicle (UAV) approaches to AN can improve performance by exploiting synchronized spatially-dispersed measurements to generate realtime information regarding the structure of the local scalar field for use in navigation decisions. This dissertation presents initial results of a comprehensive program to develop, verify, and experimentally implement mission-level AN capabilities in three-dimensional (3D) space using Santa Clara University's (SCU) unique multilayer control architecture for groups of vehicles. Using SCU's flexible formation control system, this work builds upon prior 2D AN research and provides new contributions to 3D scalar field AN by a) demonstrating a wide range of 3D AN capabilities using a unified, multilayer control architecture, b) extending multivehicle 2D AN control primitives to navigation in 3D scalar fields, and c) introducing state-based sequencing of these primitive AN functions to execute 3D mission-level capabilities such as isosurface mapping and plume following. Functionality is verified using high-fidelity simulations of multivehicle drone clusters which account for vehicle dynamics, outdoor wind gust disturbances, position sensor inaccuracy, and scalar field sensor noise. This dissertation presents the multilayer architecture for multivehicle formation control, the 3D AN control primitives, the sequencing approaches for specific mission-level capabilities, and simulation results that demonstrate these functions.

Table of Contents

Abstract.....	vi
List of Figures.....	viii
List of Tables.....	ix
Summary of Symbols.....	x
1.0 Introduction	1
1.1 Adaptive Navigation in 2D Scalar Fields.....	2
1.2 Adaptive Navigation in 3D Scalar Fields.....	5
1.3 Contributions to 3D Scalar Field Adaptive Navigation.....	6
2.0 Vehicle Dynamics and Control	8
3.0 Multirobot Control Architecture.....	11
3.1 Cluster-Space Formation Control	11
3.2 Adaptive Navigation Control.....	13
3.3 Mission-Specific State Control.....	14
4.0 Local Extrema Seeking	16
4.1 Four-Robot Tetrahedron Cluster	16
4.2 Gradient Estimation	18
4.3 Local Extrema Seeking Control Primitive	18
4.4 Source Seeking Simulation Results.....	19
4.4.1 Stationary Source Seeking in a Static Field Simulation.....	19
4.4.2 Moving Source Seeking and Tracking in a Time-Varying Field Simulation.....	21
5.0 Isosurface Navigation and Mapping.....	24
5.1 Basic Isosurface Navigation Control Primitive	24
5.2 Basic Isosurface Navigation Simulation	25
5.3 Isosurface Mapping State Machine	27
5.4 Isosurface Mapping Mission Simulation	32
6.0 Downstream Plume Following	40
6.1 Nine-Robot Prismatic Cluster.....	41
6.2 Downstream Plume Following Control Primitive	43
6.3 Plume Location and Descent State Machine	44
6.4 Downstream Plume Following Mission Simulation.....	47
6.5 Plume Location and Descent with Cluster Resizing State Machine.....	53
6.6 Downstream Plume Following with Cluster Resizing Mission Simulation	56
7.0 Summary, Discussion, and Future Work.....	66
REFERENCES	70
APPENDIX A	76
APPENDIX B.....	78
APPENDIX C	82
APPENDIX D	89
APPENDIX E.....	91
APPENDIX F.....	115
APPENDIX G	180

List of Figures

Figure 2-1: Parrot® AR.Drone	8
Figure 2-2: Typical Von Kármán Turbulence Model Simulated Wind Gusts	9
Figure 3-1: Multilayered Control Architecture Block Diagram.....	11
Figure 3-2: Adaptive Navigation Layer	14
Figure 4-1: Four-Robot Virtual Structure Lengths and Internal Angles.....	17
Figure 4-2: Multiple simulation runs showing flight paths of a four-robot cluster seeking a stationary and time-invariant source.	20
Figure 4-3: Time history of average sensed scalar field value from cluster initial position trial III.....	21
Figure 4-4: Angular deviation time history of cluster velocity from estimated gradient starting from initial position trial III.....	21
Figure 4-5: Flight path of a four-robot cluster seeking/tracking a moving and time-varying source.	22
Figure 4-6: Time history of average sensed scalar field values while seeking and tracking a moving and time-varying field.	23
Figure 4-7: Angular deviation time history of cluster velocity from estimated gradient.....	23
Figure 5-1: Basic isosurface AN.....	26
Figure 5-2: Time history of average sensed scalar field values during trial A.....	27
Figure 5-3: Angular deviation time history of cluster velocity from estimated gradient for trial A.....	27
Figure 5-4: Isosurface mapping while constrained to a plane which intersects the surface.....	28
Figure 5-5: State machine for structured isosurface mapping.....	29
Figure 5-6: Isosurface mapping of symmetric plume.....	33
Figure 5-7: Time histories of the cluster Z-coordinate and average measured scalar field level.	34
Figure 5-8: Isosurface mapping of an asymmetric plume	35
Figure 5-9: Time histories of the cluster Y-coordinate and average measured scalar field level.	36
Figure 5-10: Isosurface mapping of overlapping plumes	37
Figure 5-11: Time histories of the cluster X-coordinate and average measured scalar field level.	38
Figure 5-12: Isosurface mapping of overlapping plumes	39
Figure 6-1: Differential drive compensation signals for ridge following in 2D with a five-robot cluster ...	40
Figure 6-2: Plume following outward using a nine-robot cluster in a right-rectangular prism formation ...	41
Figure 6-3: Nine-robot formation lengths & internal angles	42
Figure 6-4: State machine for plume location and descent.....	45
Figure 6-5: Scenario A multiple simulation trials showing flight paths of nine-robot clusters that locate and then follow a scalar plume outward	48
Figure 6-6: Scenario A, trial 3 flight path of a nine-robot cluster.....	49
Figure 6-7: Scenario A, trial 3 time histories of measured scalar field values from cluster’s “back” plane.....	50
Figure 6-8: Scenario A, trial 3 time histories of measured scalar field values from cluster’s “front” plane.....	50
Figure 6-9: Scenario B flight path of a nine-robot cluster	52
Figure 6-10: Scenario B time histories of measured scalar field values from cluster’s “back” plane	53
Figure 6-11: Scenario B time histories of measured scalar field values from cluster’s “front” plane.....	53
Figure 6-12: State machine for plume location and descent while resizing the cluster	54
Figure 6-13: Scenario C flight path of a nine-robot cluster	58
Figure 6-14: Scenario C time histories of measured scalar field values from cluster’s “back” plane	60
Figure 6-15: Scenario D with fixed formation flight path of a nine-robot cluster.....	61
Figure 6-16: Scenario D with fixed formation time histories of measured scalar field values (upper plot) from cluster’s “back” plane and state machine transitions between states 2 and 3 (lower plot).	62
Figure 6-17: Scenario D with resizable formation flight path of a nine-robot cluster.....	63
Figure 6-18: Scenario D with resizable formation time histories of measured scalar field values (upper plot) from cluster’s “back” plane and state machine transitions between states 2 and 3 (lower plot).....	64
Figure 7-1: Drone cluster utilized by Santa Clara University to demonstrate Adaptive Navigation	69

List of Tables

TABLE 1-1: Applicability of Scalar Field Features	1
TABLE 2-1: Von Kármán Wind Turbulence Model Parameters.....	9
TABLE 4-1: Four-Robot Formation Robot-Space Variables	17
TABLE 4-2: Four-Robot Formation Cluster-Space Variables	17
TABLE 4-3: Four-Robot Formation Shape Variables	19
TABLE 5-1: State Machine Control Parameters for Scenario A	33
TABLE 5-2: State Machine Control Parameters for Scenario B	35
TABLE 5-3: State Machine Control Parameters for Scenario C1.....	37
TABLE 5-4: State Machine Control Parameters for Scenario C2.....	39
TABLE 6-1: Nine-Robot Formation Robot-Space Variables.....	42
TABLE 6-2: Nine-Robot Formation Cluster-Space Variables	42
TABLE 6-3: Nine-Robot Formation Shape Variables for Scenario A.....	47
TABLE 6-4: State Machine Control Parameters for Scenario A	47
TABLE 6-5: Nine-Robot Formation Shape Variables for Scenario B	51
TABLE 6-6: State Machine Control Parameters for Scenario B	51
TABLE 6-7: Nine-Robot Formation Shape Variables for Scenario C	57
TABLE 6-8: State Machine Control Parameters for Scenario C	57
TABLE 6-9: Nine-Robot Formation Shape Variables for Scenario D with Fixed Formation	61
TABLE 6-10: State Machine Control Parameters for Scenario D with Fixed Formation	61
TABLE 6-11: Nine-Robot Formation Shape Variables for Scenario D with Resizable Cluster.....	63
TABLE 6-12: State Machine Control Parameters for Scenario D with Resizable Cluster	63

List of Symbols

Symbol	Description	Symbol	Description
n	Number of vehicles in cluster	P	Number of propellers (per vehicle)
m	Degrees of freedom per vehicle	k	Propeller correction coefficient
${}^G x_i \quad {}^G y_i \quad {}^G z_i \quad {}^G \psi_i$	i th vehicle coordinates and heading	ρ	Air density
$D_x \quad D_y \quad D_z$	Vehicle level drag components	A	Propeller disc area (per propeller)
$V_x \quad V_y \quad V_z$	Wind disturbance velocity components	τ	Propeller axial thrust (hovering)
$K_u \quad K_v$	Vehicle gain matrices	\vec{R}	Vehicle specific variables vector
M	Vehicle mass	\vec{C}	Cluster specific variables vector
m_f	Air mass flow rate	J	Jacobian matrix
σ	Sensor noise standard deviation	\vec{D}_{diff}	Differential signals vector
\vec{g}_{grad}	Estimated 3D gradient vector	K_p	Proportional gain matrix
$g_x \quad g_y \quad g_z$	Estimated 3D gradient components	$Q \quad Q'$	Control selector matrices
\vec{S}	Measured scalar field values vector	$\hat{X}_c \quad \hat{Y}_c \quad \hat{Z}_c$	Cluster frame unit vectors
s_i	i th measured scalar field value	R_i	i th vehicle reference frame
S	Cluster translational speed	${}^G \hat{n}$	Navigation reference vector
R	Cluster rotational speed	n_{des}	Mapping plane and ${}^G \hat{n}$ intersection
$d_{extrema}$	Extrema following direction	$n_{cluster}$	Cluster position (mean of vehicles)
d_{orbit}	Isosurface mapping direction	$n_{threshold}$	Cluster to mapping plane distance
K_{surf}	Isosurface mapping gain	Δn	Isosurface mapping planes delta
${}^G X_B \quad {}^G Y_B \quad {}^G Z_B$	Cluster coordinates for 4-robot cluster	\vec{P}_B	Cluster point B position vector
${}^G X_C \quad {}^G Y_C \quad {}^G Z_C$	Cluster coordinates for 9-robot cluster	P_{B_proj}	\vec{P}_B projected onto ${}^G \hat{n}$
${}^G \Phi_c \quad {}^G \theta_c \quad {}^G \psi_c$	Cluster roll/pitch/yaw	s_{des}	Desired scalar value of isosurface
${}^G x_n \quad {}^G y_n \quad {}^G z_n$	Vehicle n coordinates	$s_{cluster}$	Mean of scalar field measurements
${}^G \psi_n$	Vehicle n heading w.r.t global frame	$s_{threshold}$	Scalar field threshold for isosurface
${}^c \psi_n$	Vehicle n heading w.r.t cluster frame	$s_{prom\ threshold}$	Plume following start threshold
$L_{12} \quad L_{13} \quad L_{B4}$	4-robot cluster formation lengths	$s_{term\ threshold}$	Plume following end threshold
$L_{C2} \quad L_{C3} \quad L_{C4} \quad L_{C5} \quad L_{C6} \quad L_{C7} \quad L_{C8} \quad L_{C9}$	9-robot cluster formation lengths	$\delta_{threshold}$	Distance between map start / end
$\alpha \quad \beta \quad \xi$	4-robot cluster shape angles	$\rho_{threshold}$	Angle between map start / end
$\alpha_3 \quad \alpha_4 \quad \alpha_5 \quad \alpha_6 \quad \alpha_7 \quad \alpha_9$	9-robot cluster shape angles (α)	$\gamma_{threshold}$	Angle between \vec{g}_{grad} and ${}^G \hat{n}$
$\beta_3 \quad \beta_4 \quad \beta_5 \quad \beta_6 \quad \beta_7 \quad \beta_9$	9-robot cluster shape angles (β)	$\Delta_{threshold}$	Differentials magnitude threshold
ζ	9-robot cluster shape angle (ζ)	$\Psi_c\ threshold$	Cluster alignment threshold
K_n	Mapping constraint plane gain	$K_{rot\ hold}$	Cluster rotational gain
K_x	Forward differential gain in \hat{X}_c	K_{pitch}	Pitch differential gain
K_y	Lateral differential gain in \hat{Y}_c	K_{yaw}	Yaw differential gain
K_z	Lateral differential gain in \hat{Z}_c	s_{target}	Desired sensed scalar field value

1.0 Introduction

Conventional navigation techniques typically involve user-provided waypoints or a predefined trajectory for an agent to follow over time. In contrast, Adaptive Navigation (AN) methods¹ allow an agent to autonomously alter its trajectory or direction based on realtime measurements of the environment.

The most basic form of adaptive navigation requires an explicit user-specified destination while allowing the control system to alter the vehicle's path, such as re-routing to avoid obstacles or traffic congestion. More advanced AN systems such as enemy evasion while being pursued or autonomously determining the optimal escape route do not require user specified destinations in any form. Increasingly sophisticated forms of AN are being actively researched to enable one or more agents to navigate to or along features of interest within a continuous scalar field without prior knowledge of the region, such as finding a local maximum of temperature, light intensity, or pollution concentration level.

Scalar field features represent real-world phenomena in a wide range of practical applications such as environmental monitoring/characterization, disaster response, and exploration. Table 1-1 summarizes a number of these features and their applicability in the environmental monitoring domain while a detailed discussion of scalar field features is provided in Kitts et al. [1]. In many applications where conventional methods require exhaustive mapping of an entire region, particularly when the field is time-varying, AN can dramatically reduce the time and/or cost of locating and characterizing these features.

TABLE 1-1: Applicability of Scalar Field Features

Feature	Physical Application Feature
Maximum	Hot spot, source of pollution, gas or radio emission
Minimum	Quiet/safe spot, anoxic point
Contour	Defining hazard regions, service levels
Ridge	Path to impact area from a source, path of optimal service when leaving a source
Trench	Path of maximum attenuation from a source, path of minimum exposure to a source
Saddle Point	Low energy path between minima, low exposure path between sources
Front	Atmospheric phenomena affecting weather, marine phenomena fueling bio-productivity

¹ Although the term "adaptive navigation" is used routinely in the literature, the scope of work discussed in this dissertation focuses primarily on guidance laws that allow one or more robots to move to or along features of interest in a scalar field.

1.1 Adaptive Navigation in 2D Scalar Fields

Navigation techniques for two-dimensional (2D) scalar fields have been proposed using a wide variety of control strategies with single and multiple planar vehicles.

Extrema Seeking: The basic task of autonomously locating a local source (i.e., maxima seeking) in a planar scalar field without prior knowledge of its location has attracted an immense amount of interest. Simulations using a single vehicle demonstrated this task by assuming simplified vehicle kinematics or point mass models, with control strategies based on the gradient per Zhang et al. [2], directional derivative per Mayhew et al. [3], sliding mode per Matveev et al. [4][5], stochastic approximations per Azuma et al. [6], or hybrid approaches per Kashyap et al. [7] and Kamthe et al. [8].

Experimentally, Mayhew et al. [9] demonstrated extrema seeking with a single planar vehicle in a small indoor testbed ($\approx 8 \text{ m}^2$) controlled with a line minimization-based algorithm while Shigaki et al. [10] used a bioinspired fuzzy inference algorithm to control a wheeled robot equipped with a gas sensor. Neumann et al. [11] used a single AirRobot AR-100-B UAV held at fixed altitude to locate a gas source outdoors ($\approx 320 \text{ m}^2$) using three different bioinspired spatial dithering algorithms.

To improve performance by taking distributed synchronized measurements of the field while also increasing resiliency under failure scenarios, multiple vehicle systems have been proposed. Using multiple, spatially distributed vehicles overcomes the primary drawback to many single vehicle AN control approaches, which is the need to execute time-consuming spatial dithering maneuvers in order to obtain local field information upon which navigation decisions are made per Neumann et al. [11], Ogren et al. [12], Zhuo [13], and Eu. et al. [14]. A drawback with multivehicle systems is the required increase in system sophistication due to the additional complexity associated with group control. Prior research with simulations, assuming negligible agent dynamics and external disturbances, successfully demonstrated multivehicle extrema seeking using gradient-based control strategies for agents in leader-follower roles per Biyik et al. [15] and Zhu et al. [16], fixed formations per Moore et al. [17] and Renzaglia et al. [18] or bioinspired swarms as in Gazi et al. [19], Turgeman et al. [20], and Li et al. [21]. Atanasov et al. [22] used a loosely controlled formation using a probabilistic gradient-estimation control strategy, while Zhuo [13] dithered vehicle motion with amplitudes proportional to the square of the source location estimation error. Li et al. [23] simulated source seeking with a multivehicle swarm that was controlled using a Quantum-Leading-Following-Based

optimization algorithm while allowing for obstacle avoidance. Yuan et al. [24] successfully demonstrated 2D source seeking using a leader-follower gradient-based control strategy in simulations with detailed dynamic models.

Experimentally, prior research demonstrated source seeking in 2D with multiple planar vehicles in controlled indoor environments with a limited workspace. Li et al. [25] used three EPFL E-puck wheeled robots in a fixed formation controlled with a gradient based strategy to demonstrate light source seeking in a small-scale testbed ($\approx 4 \text{ m}^2$). Wu et al. [26] used two K-Team Khepera III wheeled robots in an enclosed area ($\approx 9 \text{ m}^2$) controlled with a nonexplicit gradient estimation strategy to demonstrate light source seeking, while Bourne et al. [27] implemented a Bayesian-based motion planning algorithm with three iRobot Create wheeled robots to seek a gas plume source in an indoor area ($\approx 36 \text{ m}^2$). Wu et al. [28] controlled Seven TurtleBot3 wheeled robots in an indoor area ($\approx 25 \text{ m}^2$) with a particle swarm control strategy to seek a time-varying and moving source.

Contour Following: The more complex task of tracking/tracing a level set (i.e., contour following) at a pre-specified value in a 2D scalar field without prior knowledge of its location, has also attracted significant research interest and requires a further increase in AN sophistication. Simulations using a single vehicle demonstrated this task by assuming simplified vehicle kinematics with sliding mode per Matveev et al. [29], variational Bayesian-based per Newaz et al. [30], and hybrid level set tracking with obstacle avoidance per Matveev et al. [31] control strategies.

Experimentally, prior research demonstrated planar level set tracing/tracking with single vehicle systems. Srinivasan et al. [32] used a wheeled robot to track illumination level in a small-scale indoor testbed ($\approx 2 \text{ m}^2$) while Li et al. [33] and Tian et al. [34] completed plume tracing of Rhodamine dye by deploying an Autonomous Underwater Vessel (AUV) at a fixed depth in outdoor field applications ($\approx 30\text{k} \text{ m}^2$) controlled with a moth-inspired strategy. A hybrid experiment using an Activ-Media Pioneer 3-DX wheeled robot under sliding-mode control, navigated with respect to a simulated scalar field (computed, not sensed) in a small indoor testbed ($\approx 7 \text{ m}^2$) per Matveev et al. [35]. The work in Shen et al. [36] simulated contour mapping with a single quadcopter to experimentally demonstrate fixed altitude holding with an anti-windup PI-controller.

Extending level set tracking/tracing in 2D to multiple vehicle systems requires further system sophistication to manage group control. Simulations using multivehicle systems controlled with a wide array

of schemes have successfully demonstrated this task assuming negligible agent dynamics and external disturbances. Jin et al. [37] used a single vehicle control approach in parallel for a group of agents, where the robots initially stationed around a time-varying elliptical contour moved independently to maintain their position on the contour; robots then shared their location to collectively generate an estimate of the contour. Zhang et al. [38] performed cooperative filtering using Hessian information to move a robot group along noisy level curves with active formation shaping to minimize gradient estimation error. Kang et al. [39] used a bioinspired control strategy to direct a lead robot along a plume using a single robot movement strategy with two other robots flanking the leader; as the leader moved out of the plume, a flanking robot would theoretically remain in the plume and become the new leader. Other teams have also demonstrated this task with fixed, circular formations of agents using a gradient-based control strategy per Han et al. [40][41]. Note that a concern for any multivehicle system is ensuring safety through the inclusion of collision avoidance laws; Paul et al. [42] contains a wide range of such strategies.

Experiments with multiple vehicles have been performed where the UAVs were held at fixed altitudes in small and controlled indoor testbeds. Cook et al. [43] implemented gradient-based control of three DJI Flamewheel 450 UAVs in fixed circular formations to demonstrate moving light source seeking and three Crazyflie 2.0 UAVs to perform level set tracking/tracing about a simulated scalar field (computed, not sensed) in an enclosed area of approximately 2 m². With seven Crazyflie 2.1 UAVs at fixed altitudes, Datar et al. [44] demonstrated swarming towards simulated scalar field sources (computed, not sensed) in region of approximately 12 m².

Prior 2D AN Work at Santa Clara University: Prior work by this research group has presented AN control strategies for both locating local extreme points and following contours, and has demonstrated these in simulation, lab experiments, and field demonstrations. Furthermore, this prior work has proposed new AN objectives to include moving along ridges/trenches, locating saddle points, and navigating along fronts; all of these control capabilities have been verified in both simulation and experiment in Kitts et al. [1], and McDonald et al. [45][46].

1.2 Adaptive Navigation in 3D Scalar Fields

Research in 3D AN techniques is significantly less developed than 2D techniques.

Extrema Seeking: Simulated single vehicles with simplified kinematic and point mass models have been used to demonstrate extrema seeking by Cochran et al. [47][48] with a gradient following strategy, while Matveev et al. [49] used hybrid-sliding mode control, and Lin et al. [50] utilized forward and angular velocity regulation as the robot approached the source. Experimentally, in a small indoor testbed ($\approx 3\text{m} \times 3\text{m} \times 2\text{m}$), Eu. et al. [14] demonstrated source seeking of a dynamic gas plume with a single UAV performing fuzzy-logic control-based spatial dithering.

Multirobot simulations have demonstrated source seeking in 3D while assuming negligible agent dynamics and external disturbances. Ogren et al. [12] proposed a formation that adapted to its environment to optimize gradient climbing while Al-Abri et al. [51] implemented a bioinspired swarm controlled with a Speeding Up and Slowing Down strategy which synchronized the direction of motion of the robots by varying the speed of each vehicle based on the measured scalar field value. Briñón-Arranz et al. [52] utilized symmetric and fixed formations to navigate based on the collaborative estimation of gradient and Hessian information using noisy measurements while Tan [53] provided a comprehensive survey of “plume tracing and mapping via swarm robots” which is classified as extrema finding since it involves agents moving towards a source.

Contour Following: When extended to 3D, 2D contour/level set tracking becomes navigation on an isosurface. Matveev et al. [54] explored this with a single vehicle simulation by assuming a simplified kinematic robot, no external disturbances, and minimal system noise. Their proposed gradient-free strategy utilized the vehicle’s pose, instantaneous scalar field reading, and the time derivative of that signal to generate yaw control for a nonholonomic agent; allowing the vehicle to orbit a desired isosurface while cycling vertically between two predefined altitudes at a constant speed.

3D multivehicle contour / level set tracking (e.g., isosurface navigation) has been simulated assuming negligible robot dynamics and limited external disturbances. Using a leader-follower strategy initially published in Moore et al. [17], Han et al. [55] simulated the task of 2D planar contour mapping on a highly simplified 3D radiation field (i.e., an extruded ellipse) where the 3D aspect involved holding each of the two UAVs at different fixed altitudes and performing concurrent contour mapping in parallel planes.

Jacobi transform based formation control (to decouple the dynamics of the formation's center from its shape) was simulated by Wu et al. [56] utilizing Hessian information to move a robot group along noisy 3D level set curves to track principal lines of surface curvature. With two vessels in a fixed formation, in which one agent functioned as the sensor and the other as the tracker, Wang et al. [57] simulated dynamic plume front (i.e., time-varying level set) tracking while assuming ideal sensor measurements and an exponentially decaying external disturbance.

1.3 Contributions to 3D Scalar Field Adaptive Navigation

This dissertation presents initial results of a comprehensive program to develop, verify, and experimentally implement mission-level AN capabilities in three-dimensions. This builds upon prior SCU work and contributes to 3D scalar field AN by:

- demonstrating a wide range of diverse 3D AN capabilities using a single, unified, multilayer control architecture;
- defining new four- and nine-robot formation geometries specific to the needs of the new 3D AN control primitives, to include the derivation of their associated forward and inverse kinematic transforms;
- extending SCU's 2D multivehicle scalar field AN control primitives to 3D space in order to implement local extrema finding, navigation on isosurfaces, and navigation along plumes;
- introducing the state-based sequencing of different AN control primitives to execute mission-level capabilities such as isosurface mapping and downstream plume following;
- verifying the functionality of these aforementioned capabilities using high-fidelity simulations which account for vehicle dynamics, outdoor wind gust disturbances, position sensor inaccuracy, and scalar field sensor noise.

Because the focus in this work is to present new 3D AN capabilities, this work makes a number of simple implementation choices. For example, the proposed control laws are composed of basic proportional control terms. Furthermore, this work uses minimally-sized groups of robots that allow instantaneous characteristics of the scalar field to be computed and guidance decisions to be made for the proof of concept capabilities presented. Without question, more sophisticated controller approaches are possible in order to improve performance; such improvements are proposed as future work.

Section 2 describes the individual vehicle-level dynamic model, external wind disturbance model, and sensor noise models that the high-fidelity simulations are built upon. Section 3 discusses the multilayered control architecture and provides details of each layer. Section 4 presents the control architecture used to implement local extrema seeking (both static and time-varying), the definition of the required multivehicle cluster, the gradient estimation method, the proposed control primitive, and high-fidelity simulation results. Section 5 presents the implementation of isosurface navigation and mapping within the architecture, a description of the proposed control primitives, the state machine used to perform state-based sequencing for “mission-level” functions, and high-fidelity simulation results. Section 6 presents the downstream plume following (away from a source) implementation, the definition of the required multivehicle cluster, the proposed control primitive, the state machine used to perform state-based sequencing for “mission-level” functions, and high-fidelity simulation results. Finally, section 7 discusses broad questions, issues and challenges while section 8 summarizes this work and discusses ongoing and future research efforts.

2.0 Vehicle Dynamics and Control

The work presented in this dissertation simulates a cluster of AR.Drone quadcopters, which are consumer-grade UAVs manufactured by Parrot, Inc. A single Parrot® AR.Drone quadcopter is shown in Figure 2-1.



Figure 2-1: Parrot® AR.Drone

Given the use of an onboard autopilot for platform stabilization with limited pitch and roll angles, this work considers only the remaining degrees of freedom (DOF) with the relevant position and velocity state of the i th quadcopter as defined in (1), where x_i , y_i , z_i , and ψ_i are the three position coordinates and yaw of vehicle i , respectively.

$$\vec{R}_i = \begin{bmatrix} x_i \\ y_i \\ z_i \\ \psi_i \end{bmatrix}, \quad \dot{\vec{R}}_i = \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{z}_i \\ \dot{\psi}_i \end{bmatrix} \quad (1)$$

The four DOF dynamic model for an individual quadcopter is shown in (2), where the diagonal matrices K_u and K_v contain proportionality constants experimentally determined by Santos et al. [58], Santana et al. [59][60] and wind disturbance forces have been included. M is the vehicle mass (0.42 kg), $\dot{\vec{R}}_{i\,com}$ is the vector of commanded velocities, and $[D_x, D_y, D_z]^T$ are the drag force components. Additional details of the UAV dynamic model and its validation are presented in Appendix A.

$$\ddot{\vec{R}}_i = K_u \dot{\vec{R}}_{i\,com} - K_v \dot{\vec{R}}_i + \frac{1}{M} \begin{bmatrix} D_x \\ D_y \\ D_z \\ 0 \end{bmatrix} \quad (2)$$

A Von Kármán turbulence model with a mean wind speed of 5 m/s is used to induce normal forces (i.e., ram/momentum drag) on each UAV propeller per Selig [61] as a function of mass flow rate m_f and wind velocity $[V_x, V_y, V_z]^T$. Equation (3) illustrates the drag computation with model parameters from Selig [61] summarized in Table 2-1 and with typical wind gust time histories shown in Fig. 2-2. Note that in the high-fidelity simulations, each vehicle is subjected to different external wind gust disturbance models (i.e, different random noise seeds).

$$\begin{bmatrix} D_x \\ D_y \\ D_z \end{bmatrix} = P \cdot m_f \cdot \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} = P \cdot k \cdot \left(\rho A \sqrt{\frac{\tau}{2\rho A}} \right) \cdot \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} \quad (3)$$

TABLE 2-1: Von Kármán Wind Turbulence Model Parameters

Number of Propellers	$P = 4$
Correction Coefficient (non-shrouded propeller)	$k = 0.8$
Air Density (sea level @ +15 °C)	$\rho = 1.2260 \text{ kg/m}^3$
Propeller Disc Area	$A = 0.0314 \text{ m}^2$
Propeller Axial Thrust (hovering)	$\tau = 1.03 \text{ N}$

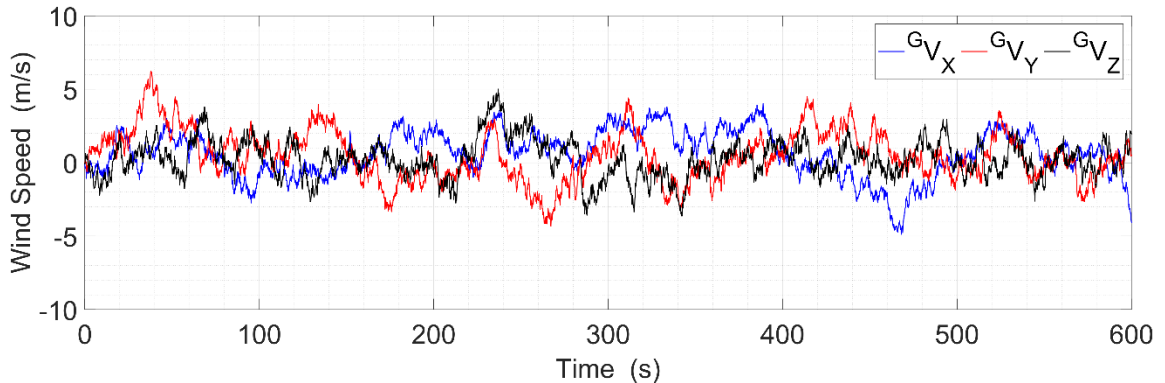


Figure 2-2: Typical Von Kármán Turbulence Model Simulated Wind Gusts

Sensor noise is simulated using band-limited white noise to generate normally distributed random numbers which are scaled by a constant factor. For the vehicle position sensors, manufacturer supplied RMS accuracy values can be approximated as one standard deviation σ when there is zero mean error per GPS World [62] and the National Research Council [63].

Scaling the simulated noise to attain a standard deviation equal to the RMS value allows us to achieve a 98% Accuracy Distribution (AD) per GPS World [64] as shown in (4).

$$98\% (AD) = 2.8 \cdot (RMS) = 2.8 \cdot \sigma \quad (4)$$

Since typical UAV mounted consumer-grade GPS receivers advertise positional accuracies from 1.0 m to 1.8 m RMS [65][66], the higher end value is used as the simulated position sensor noise standard deviation target. Similarly, electronic noise for vehicle mounted air pollutant sensors (e.g., NO₂, SO₂, CO) use a scaling factor based on the sensor manufacturers' provided standard deviation [67][68]. For comparison, this method generates larger sensor noise amplitudes compared to those used in other high-fidelity simulations from Matveev et al. [54], Wang et al. [57], Santana et al. [59], and Yang et al. [69].

3.0 Multirobot Control Architecture

“Mission-level” multirobot AN was implemented using the multilayer control architecture shown in Fig. 3-1. This general architecture is applicable to different numbers and types of robots, which are operating in different domains and performing different types of adaptive navigation functions. This section discusses the overall control architecture and provides details of the modeling and control implemented in each layer for multiple UAV clusters performing 3D AN.

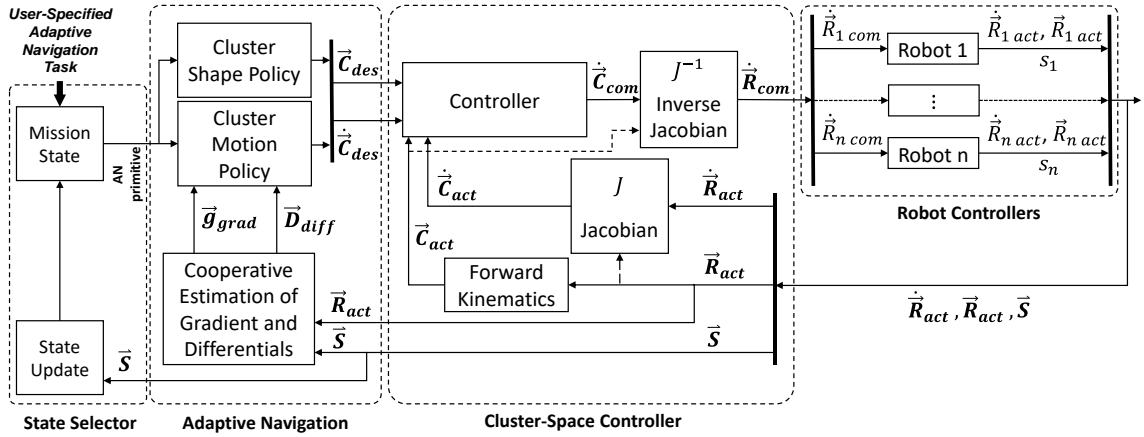


Figure 3-1: Multilayered Control Architecture Block Diagram

3.1 Cluster-Space Formation Control

Within the architecture, the Cluster-Space Controller layer responds to commands regarding the desired geometry and motion of the cluster by issuing translational and rotational velocity commands to each vehicle in the system. The controller uses a full-DOF operational-space framework, managing the motion and geometry of the group of robots as a virtual articulating mechanism whose shape, size, and orientation can be reconfigured in-flight per Kitts et al. [70]. The ability to implement a particular geometry allows for precise, spatially dependent scalar field measurements, tunable for (in)sensitivity to certain spatial frequencies.

For a system comprising n robots, “robot-space” pose and velocity vectors are represented as a concatenation of the individual robot position and velocity with respect to the global frame per (5).

$$\vec{R} = \begin{bmatrix} \vec{R}_1 \\ \vdots \\ \vec{R}_i \\ \vdots \\ \vec{R}_n \end{bmatrix}, \quad \dot{\vec{R}} = \begin{bmatrix} \dot{\vec{R}}_1 \\ \vdots \\ \dot{\vec{R}}_i \\ \vdots \\ \dot{\vec{R}}_n \end{bmatrix} \quad (5)$$

Since the Cluster-Space Controller layer uses a formation-specific representation of the motion state of the system, “cluster-space” positions, represented by the pose vector \vec{C} , typically include the position and orientation of the overall cluster, variables describing the cluster geometry, and variables that specify the relative orientation of each vehicle. Note that the cluster velocity vector $\dot{\vec{C}}$ is the time derivative of \vec{C} . A set of kinematic relationships relate the robot-space and cluster-space vectors as presented in (6) and (7), where n is the number of robots in the cluster, m is the number of DOF per robot, and J is the system’s Jacobian matrix.

$$\vec{C} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{mn} \end{bmatrix} = KIN(\vec{R}) = \begin{bmatrix} f_1(r_1, r_2, \dots, r_{mn}) \\ f_2(r_1, r_2, \dots, r_{mn}) \\ \vdots \\ f_{mn}(r_1, r_2, \dots, r_{mn}) \end{bmatrix} \quad (6)$$

$$\dot{\vec{C}} = \begin{bmatrix} \dot{c}_1 \\ \dot{c}_2 \\ \vdots \\ \dot{c}_{mn} \end{bmatrix} = J(\vec{R})\dot{\vec{R}} = \begin{bmatrix} \frac{\partial f_1}{\partial r_1} & \frac{\partial f_1}{\partial r_2} & \dots & \frac{\partial f_1}{\partial r_{mn}} \\ \frac{\partial f_2}{\partial r_1} & \frac{\partial f_2}{\partial r_2} & \dots & \frac{\partial f_2}{\partial r_{mn}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_{mn}}{\partial r_1} & \frac{\partial f_{mn}}{\partial r_2} & \dots & \frac{\partial f_{mn}}{\partial r_{mn}} \end{bmatrix} \dot{\vec{R}} \quad (7)$$

Cluster-space position variables are generally nonlinear functions of robot position variables, while the cluster velocity vector is an instantaneously linear function of robot velocities as expressed by a Jacobian matrix. The exact nature of these functions depend on the specific cluster definitions used.² For the work presented in this dissertation, two different cluster definitions are used and detailed in sections 4.1 and 6.1. Note also that the J and J^{-1} matrices do not need to be updated at the same rate as the cluster control loop.

The Cluster-Space Controller layer receives instantaneous cluster mobility and geometry setpoint commands in the form of desired values of \vec{C}_{des} and $\dot{\vec{C}}_{des}$, which are provided by the Adaptive Navigation control layer. Cluster-space error signals are generated by subtracting estimates of realtime cluster-space values

² The cluster designer has a great deal of flexibility in defining the “location” of the cluster. Fundamentally, the position and orientation of the cluster frame is defined as a function of robot positions and orientations, and these definitions constitute several of the cluster pose functions in the forward kinematic functions in (6). The frame can be assigned to be coincident with one particular robot, or it could be defined as a function of two or more of the cluster’s robots. There are ramifications to this choice in terms of the level of (de)centralization that can be achieved in controlling the cluster.

from these setpoints. These cluster-space estimates are generated by applying the kinematic transforms to robot-space position and velocity commands.³

For this study, the control law itself is a simple, error-driven, resolved-rate, proportional controller that computes compensation commands in the cluster-space. This conversion is computed by use of the cluster's inverse Jacobian transform and robot commands consisting of instantaneous translational and rotational velocity setpoint commands.⁴

Overall, the set of instantaneous robot-space velocity commands for each vehicle is computed as specified by (8), where K_p is a diagonal matrix of proportional gains appropriate for the variables being controlled. For this study, these gains were empirically developed to achieve the desired performance; prior SCU work demonstrated the use of a control law design process using a partitioned, nonlinear, model-based method.

$$\begin{aligned}\dot{\vec{R}}_{com} &= J^{-1} \cdot \dot{\vec{C}}_{com} \\ &= J^{-1} \cdot \left[K_p \cdot \left[Q \cdot (\vec{C}_{des} - \vec{C}_{act}) + Q' \cdot (\dot{\vec{C}}_{des} - \dot{\vec{C}}_{act}) \right] \right]\end{aligned}\quad (8)$$

3.2 Adaptive Navigation Control

The next layer in the control architecture, the Adaptive Navigation layer, issues cluster shape and mobility commands to the Cluster-Space Controller layer. It does this based on the position and measured local scalar field value from each robot as well as the selected control “primitive” AN strategy of interest. As shown in Fig. 3-2, these control primitives use robot positions \vec{R} and sensor readings \vec{S} (vector of the measured scalar field values from Robots 1 through n) to compute relevant characteristics of the local scalar field, such as the gradient or differential measurements across various baselines within the cluster. Based on these characteristics, the primitive control law provides instantaneous setpoint commands for the cluster's position \vec{C}_{des} and velocity $\dot{\vec{C}}_{des}$ in order to guide the cluster to/along scalar field features of interest.

³ Alternatively, it is possible to make direct measurements of some or all of the cluster-space variables, such as the distance between vehicles. Although this study does not use that approach, SCU has implemented such relative sensing in prior work.

⁴ In prior work, SCU has used a variety of other cluster-space control laws, including full PID and nonlinear control laws, dynamic controllers that issue force/torque commands, etc. SCU has also defined stability criteria and demonstrated the incorporation of robust collision avoidance behaviors and singularity handling strategies.

More specifically, the AN controller provides a setpoint command for the position or velocity (e.g., not both) relating to each cluster-space parameter. In implementing each AN control primitive, diagonal control selector matrices, Q and Q' , are used to designate which cluster pose variables are either position- or velocity-controlled. For each cluster-space variable, a value of 1 or 0 on the diagonal of the Q and Q' matrices specifies position or velocity control, respectively, such that $Q = I - Q'$ where I is the identity matrix.

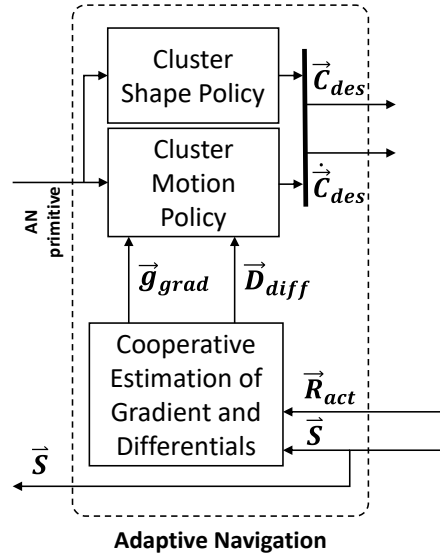


Figure 3-2: Adaptive Navigation Layer

For 2D AN, SCU demonstrated “primitive” controllers capable of implementing extrema finding, contour following, saddle point positioning, ridge descent/trench ascent, and front following per Kitts et al. [1]. This study introduces primitive AN controllers for 3D local extrema finding, isosurface navigation, and downstream plume following.

3.3 Mission-Specific State Control

Adaptive navigation control primitives are designed to perform a very specific navigation task. In some cases, such as with local extrema seeking, the control primitive might be all that is required for certain applications, such as finding a local signal source. More complex tasks, however, may demand the execution of a single control primitive, requiring a) the combination of an AN primitive with another controller to achieve parallel objectives, b) the cycling of different control parameter values over time for a single AN primitive,

and/or c) the sequencing of different AN primitives, based on the objective and desired performance. As shown in Fig. 3-1, the control architecture implements such complex functionality through the use of a state machine.

This work utilizes specific ‘missions’ of interest involving 3D scalar fields to showcase each of the three aforementioned cases of tailoring usage of the available AN control primitives. Simulations of these missions are provided in sections 5.4, 6.4, and 6.6.

4.0 Local Extrema Seeking

Real world applications of extrema seeking include tasks such as finding emitters, sources of pollution or leaks, anoxic areas in bodies of water, and so on. This work implements local extrema seeking using a single AN control primitive. Taking distributed synchronized sensor measurements from each vehicle allows computation of an estimate of the local field gradient for use as a reference direction for motion. By commanding the cluster to move in the direction of the gradient, the cluster moves towards a local maximum/source within the field. Conversely, traveling in the opposite direction of the gradient guides a cluster to a local minimum/sink in the field.

The work presented in this dissertation uses a tetrahedron-style four-robot cluster formation; a shape with the minimum number of robots required to compute an instantaneous 3D field gradient. This enables AN implementation without requiring extraneous motion (i.e., dithering) to characterize the nature of the local field. Note that the cluster is sized based on the considerations discussed in section 7 and extensively detailed in Kitts et al. [1].

This section presents the four-robot cluster definition, gradient estimation strategy, and gradient-based AN control law; which provides all necessary information to implement the local extrema finding capability. Functionality of this control primitive is demonstrated with high-fidelity simulations for a static scalar field as well as a time-varying field that is moving in a complex manner.

4.1 Four-Robot Tetrahedron Cluster

A nominal tetrahedron shape of the four-robot cluster is shown in Figure 4-1, where $R1$ through $R4$ represent origins of the individual vehicle frames. Note that this definition is used for the local extrema seeking AN control primitive as well as isosurface navigation and mapping in section 5. Robot-space position and orientation variables listed in Table 4-1 represent the pose of each robot in the figure. The robot-space pose vector of the entire group, \vec{R}_{4R} , contains all of the variables for each of the four robots. Note that the robot-space velocity vector, $\dot{\vec{R}}_{4R}$, is the time derivative of \vec{R}_{4R} .

The cluster definition used for this study locates the origin (Point B) of the cluster frame C_{4R} at the centroid of the triangle formed by Robots 1 to 3, where \hat{Z}_C is perpendicular to that plane and \hat{X}_C is pointed

towards Robot 1. The six cluster shape variables, which include three distances and three angles that define the cluster's geometry, are shown in Fig. 4-1 and listed in Table 4-2; this includes a side-angle-side description of the base triangle as well as a distance-azimuth-declination description of Robot 4 with respect to the cluster origin. The cluster-space position vector \vec{C}_{4R} consists of the variables in Table 4-2 while the cluster-space velocity vector $\dot{\vec{C}}_{4R}$ is the time derivative of \vec{C}_{4R} . The full listing of the vectors \vec{R}_{4R} and \vec{C}_{4R} are provided in Appendix B.

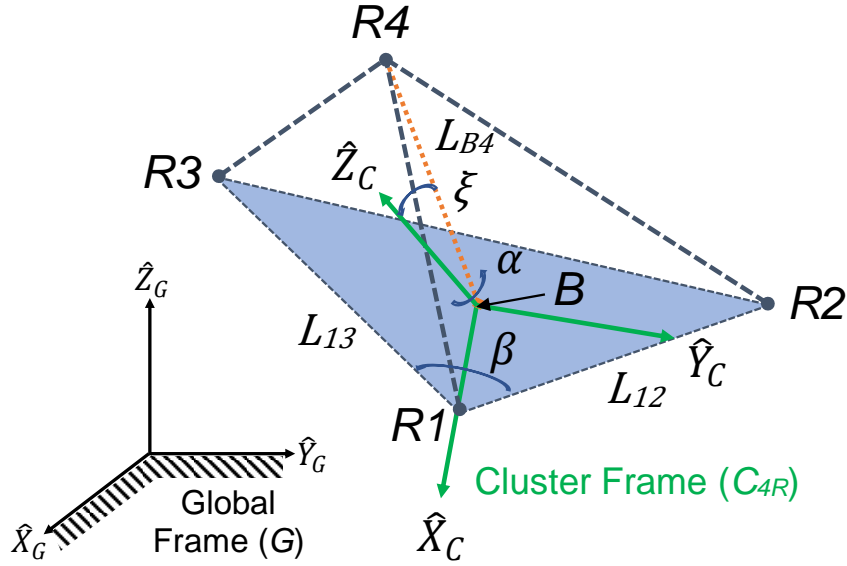


Figure 4-1: Four-Robot Virtual Structure Lengths and Internal Angles

TABLE 4-1: Four-Robot Formation Robot-Space Variables

Robot position variables, $n = 1,2,3,4$	${}^G x_n$	${}^G y_n$	${}^G z_n$
Robot yaw angles, $n = 1,2,3,4$	${}^G \psi_n$		

TABLE 4-2: Four-Robot Formation Cluster-Space Variables

Cluster position variables	${}^G X_B$	${}^G Y_B$	${}^G Z_B$
Cluster orientation variables	${}^G \phi_C$	${}^G \theta_C$	${}^G \psi_C$
Cluster shape variables, distances	L_{12}	L_{13}	L_{B4}
Cluster shape variables, angles	α	β	ξ
Robot relative yaw variables, $n = 1,2,3,4$	${}^C \psi_n$		

The forward position kinematic equations of the form described in (6) which transform “robot-space” pose vector \vec{R}_{4R} to the “cluster-space” vector \vec{C}_{4R} are presented in Appendix B. These equations are used in the Cluster-Space Controller layer to estimate realtime cluster pose, which is an input to the cluster controller. The

inverse Jacobian relationship J^{-1} is used to transform cluster control velocity commands to individual robot commands and is presented in Appendix E.

4.2 Gradient Estimation

Equation (9) generates the cooperative estimated gradient per Li et al. [25] at the robot formation's geometric center where ${}^G x_i, {}^G y_i, {}^G z_i,$ are the i th robot position coordinates, s_1 through s_4 are the measured scalar field values from Robots 1 through 4, and g_x, g_y, g_z are the components of the gradient vector as presented in (10).

$$\begin{bmatrix} g_x \\ g_y \\ g_z \\ 1 \end{bmatrix} = \begin{bmatrix} {}^G x_1 & {}^G y_1 & {}^G z_1 & 1 \\ {}^G x_2 & {}^G y_2 & {}^G z_2 & 1 \\ {}^G x_3 & {}^G y_3 & {}^G z_3 & 1 \\ {}^G x_4 & {}^G y_4 & {}^G z_4 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix} \quad (9)$$

$$\vec{g}_{grad} = \begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix} \quad (10)$$

4.3 Local Extrema Seeking Control Primitive

The gradient-following control primitive, shown in (11), generates translational cluster velocity commands to guide the cluster at a speed S in the direction of or opposite to the gradient by setting $d_{extrema}$ to +1 or -1, respectively.

$$\dot{\tilde{C}}_{4R}(1:3) = \begin{bmatrix} \dot{X}_{B des} \\ \dot{Y}_{B des} \\ \dot{Z}_{B des} \end{bmatrix} = S \cdot d_{extrema} \cdot \begin{bmatrix} \frac{g_x}{\|\vec{g}_{grad}\|} \\ \frac{g_y}{\|\vec{g}_{grad}\|} \\ \frac{g_z}{\|\vec{g}_{grad}\|} \end{bmatrix} \quad (11)$$

Cluster translational velocity commands are continuously varied based on the estimate of the local field gradient while all other cluster orientation and geometry variables designated by $\dot{\tilde{C}}_{4R}(4:16)$ are controlled to specified position setpoints using a proportional position controller. This achieves blended proportional velocity and position control with the resolved-rate cluster-space controller expressed in (8), given the selector matrix $Q' = \text{diag}(1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0)$ and $Q = I - Q'$, where I is the identity matrix.

Given the empirical tuning used in this study, this fully specifies the control law for the four-robot cluster in order to move along the local gradient of a field in order to navigate to the local minimum or maximum of the field.

4.4 Source Seeking Simulation Results

This section presents high-fidelity flight simulation results demonstrating the capabilities of the 3D AN gradient-following control primitive to perform local extrema finding behavior. These simulation results incorporate the verified vehicle dynamics model presented in (2), outdoor wind gust disturbances, position sensor inaccuracy, and scalar field sensor noise. The velocity command for each vehicle in the four-robot cluster is specified by (8), where $\vec{C}_{4R\ des}$ and $\dot{\vec{C}}_{4R\ des}$ are described in section 4.3, the gradient estimate is computed as described in section 4.2, and the four-robot formation and its kinematic transforms used to specify the instantaneous vehicle velocity commands are described in section 4.1.

Plots of the cluster’s 3D flight paths, time histories of relevant variables, and error metrics serve to illustrate the system performance during mission execution. The simulated scalar fields in this section are documented in Appendix D.

For these simulations, the four-robot cluster is held in a tetrahedron formation with shape variables shown in Table 4-3 and with fixed relative cluster attitude angles. Cluster translational speed was 3 m/s, and navigation occurred in a cubic workspace with 600 m length sides.

TABLE 4-3: Four-Robot Formation Shape Variables

L_{12}	25 m	α	0°
L_{13}	25 m	β	60°
L_{B4}	20.4 m	ξ	15°

4.4.1 Stationary Source Seeking in a Static Field Simulation

In the first demonstration, a four-robot cluster navigates to a stationary maximum point within a time-invariant scalar field simulated as a vertically oriented directional field with a broad beam width. Note that in Lee et al. [71], SCU demonstrated time-invariant source seeking without dynamic robot models, wind disturbances, or system noise. For this new work, Fig. 4-2 illustrates the resulting cluster paths for three different

trials, each with a different starting point. Per the figure, the cluster successfully navigates to the source in each trial.

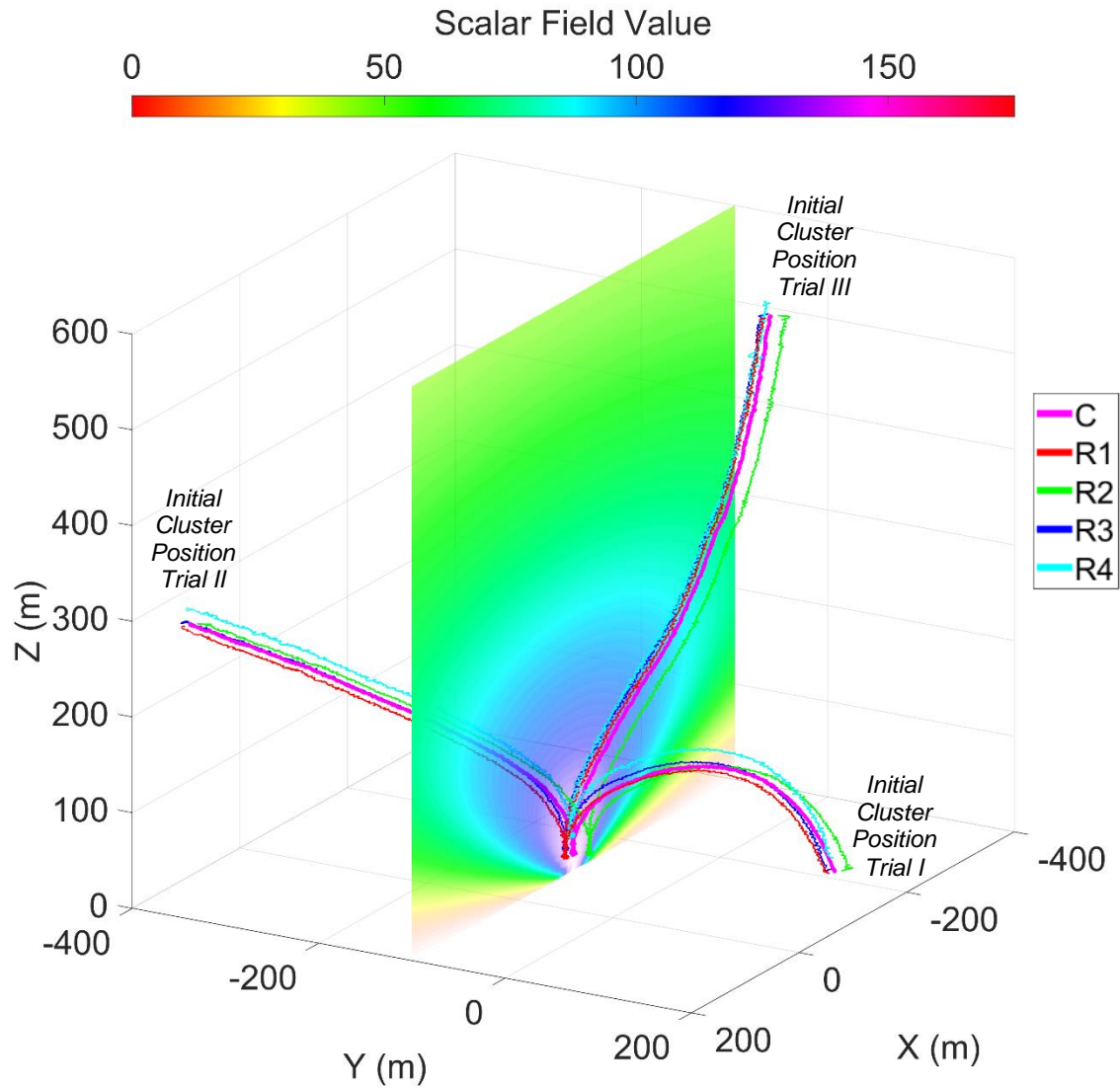


Figure 4-2: Multiple simulation runs showing flight paths of a four-robot cluster seeking a stationary and time-invariant source, each starting from different initial cluster locations.

Using trial III to illustrate performance, Fig. 4-3 shows the average sensed scalar field value of the formation increasing as expected over time, while the cluster navigates to the source. Given that the AN law directs the cluster to move in the direction of the gradient, Fig. 4-4 shows the angular deviation of these vectors over time; the RMS error for this period was 0.24 radians, until the cluster effectively reached the source. Furthermore, given that the controller is working to hold formation geometry as the cluster navigates, the RMS

errors for the cluster size parameters L_{I2} , L_{I3} , and L_{B4} were 1.9 m, 1.7 m and 1.0 m, respectively; which are within 0.5-1.5 times the standard deviation of the simulated position sensing error, indicating acceptable formation control.

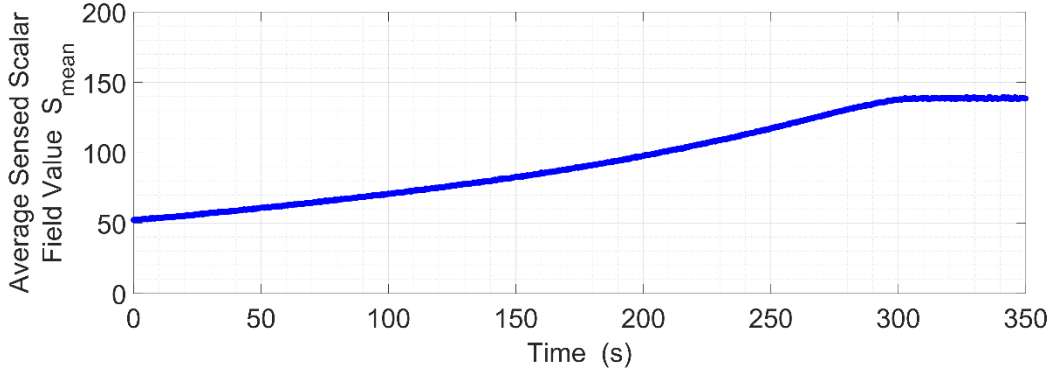


Figure 4-3: Time history of average sensed scalar field value from cluster initial position trial III.

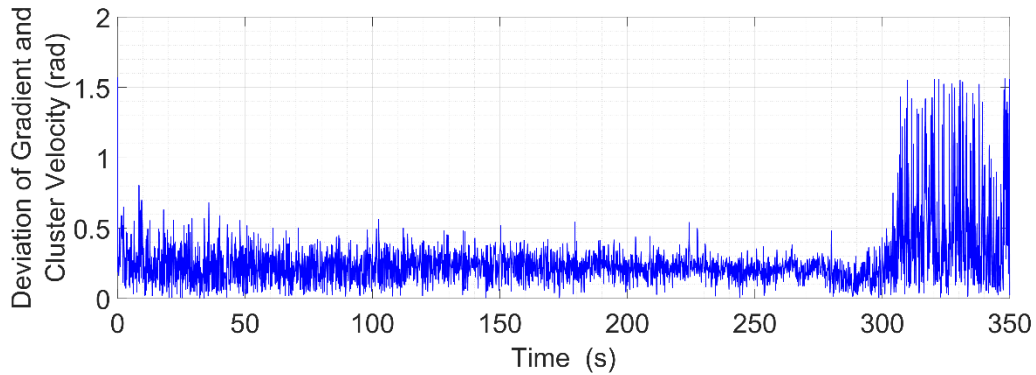


Figure 4-4: Angular deviation time history of cluster velocity from estimated gradient starting from initial position trial III. Cluster reaches the local maximum at approximately 295 s and dithers in the vicinity of the maximum without a motion termination condition.

4.4.2 Moving Source Seeking and Tracking in a Time-Varying Field Simulation

The local extrema seeking AN control primitive also enables a cluster to track a moving source in a dynamic field, provided that the cluster speed is greater than the source's speed and the evolution of the field is well-behaved. Without fully characterizing the limitations of this capability, this work offers a simple scenario of navigation in which the cluster locates a moving source within a time-varying field. Multipart Fig. 4-5 shows a cluster performing this function. The multipart figure shows how the source (which lies in the $z = 0$ plane) moves over time in an exponentially growing sinusoidal path, while the intensity of the field

increases linearly over time. Cluster shape and speed as well as the size of the navigation region were the same as used in section 4.4.1. For this scenario, the speed of the source is approximately 75% that of the cluster. The multimedia file associated with this dissertation provides an animation of this maneuver. Note that earlier work, Lee et al. [71] demonstrated moving source seeking and tracking of a time-invariant intensity plume without including realistic system dynamics, external disturbances, and system/sensor noise.

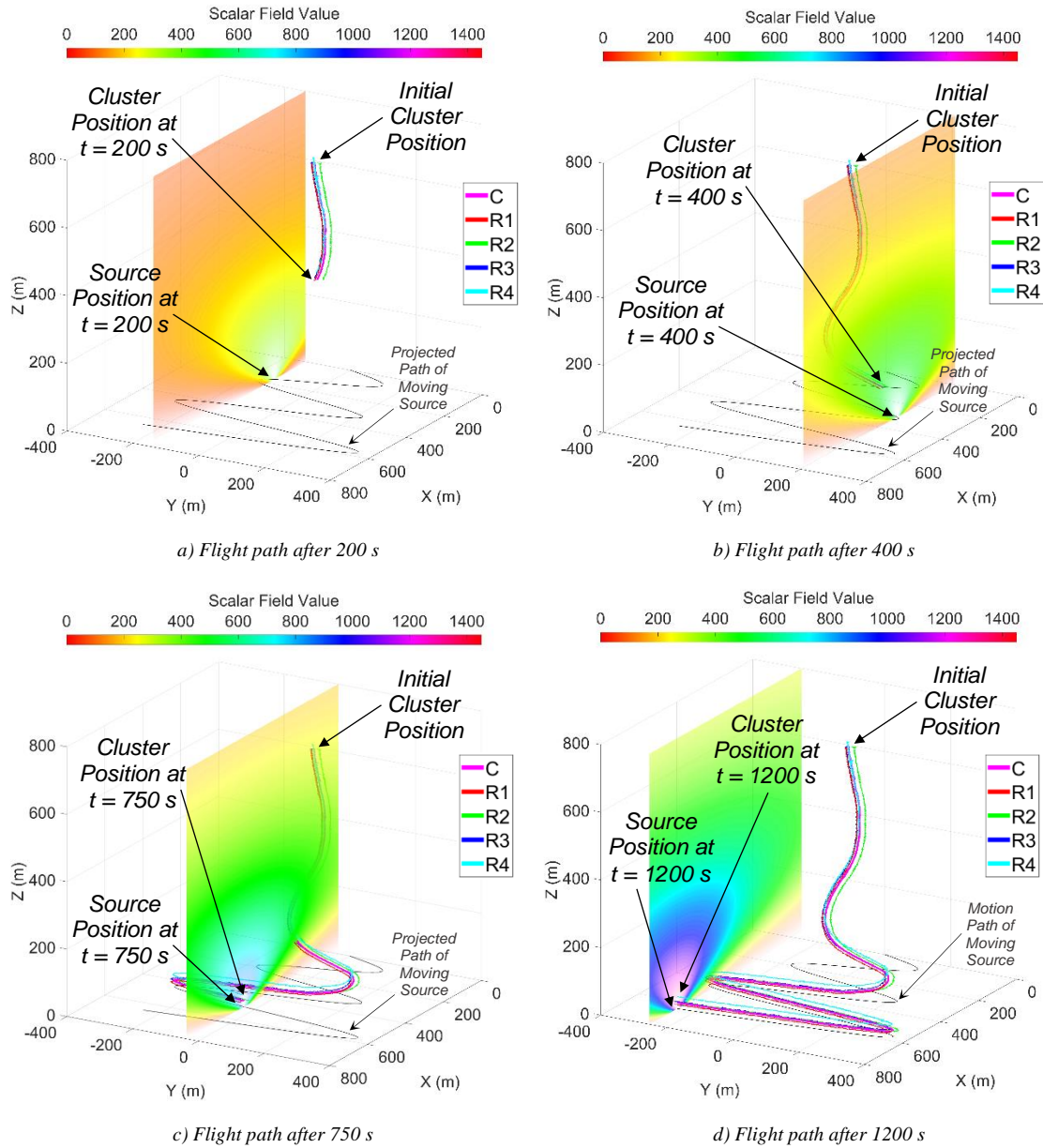


Figure 4-5: Flight path of a four-robot cluster seeking/tracking a moving and time-varying source.

To characterize performance, Fig. 4-6 shows the average sensed scalar value for the formation increasing as expected over time, as the cluster navigates toward the moving source. Fig. 4-7 shows the angular deviation of the cluster motion and gradient vectors over time, given that they should ideally be aligned; the RMS error for the shown maneuver was 0.19 radians. Note that at approximately $t = 420$ s, the cluster “catches up” with the moving source so that small changes in tracking position result in large angular deviations between the estimated gradient and cluster velocity vector. As for formation control performance as the cluster navigates, the RMS errors for the cluster size parameters L_{I2} , L_{I3} , and L_{B4} were 1.8 m, 1.7 m and 1.1 m, respectively; these are nearly identical to the results shown in section 4.4.1 and are in a range of 0.5-1.5 times the standard deviation of the simulated position sensor error, indicating acceptable formation control.

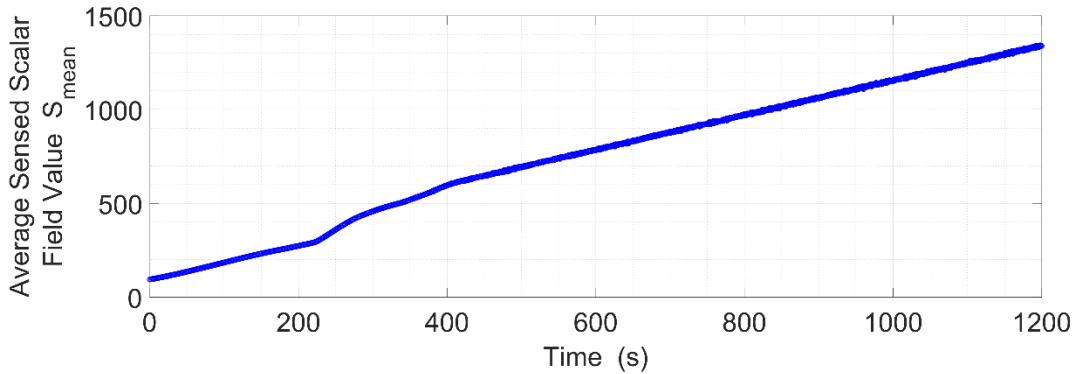


Figure 4-6: Time history of average sensed scalar field values while seeking and tracking a moving and time-varying field.

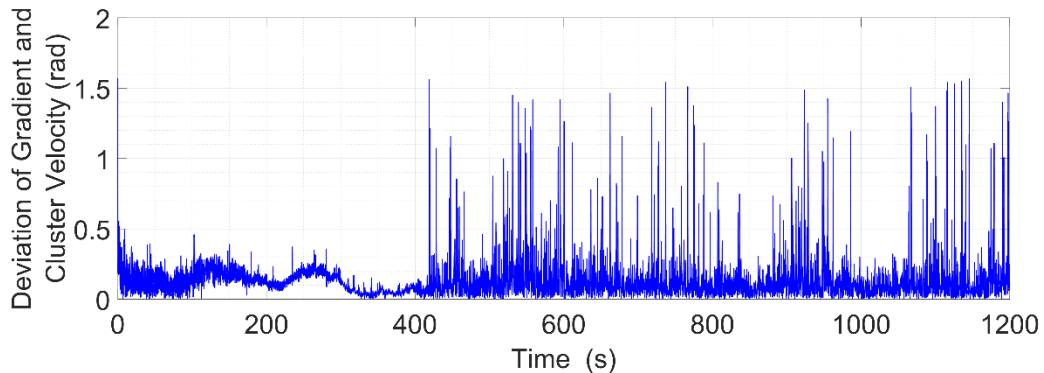


Figure 4-7: Angular deviation time history of cluster velocity from estimated gradient. Note that at approximately $t = 420$ s, the cluster “catches up” with the moving source so that small changes in tracking position result in large angular deviations between the estimated gradient and cluster velocity vector.

5.0 Isosurface Navigation and Mapping

The extension of 2D contour following to 3D scalar fields is movement along a scalar isosurface. Accordingly, this AN primitive controller moves a cluster to and maintains a cluster's position on an isosurface. However, motion within the isosurface is unconstrained thereby allowing for other control objectives to be met. One such control objective is structured isosurface mapping which will be discussed in sections 5.3 and 5.4.

To move a cluster to and then maintain its position on an isosurface, this section utilizes the same four-robot cluster definition previously discussed in section 4.1. Sensed scalar readings from the robots are used to compute the local field gradient per (9). Sections 5.1 and 5.2 propose the basic isosurface AN control primitive and simulations of its performance, respectively. Finally, sections 5.3 and 5.4 show the performance of a mission-level state machine to implement structured isosurface mapping.

5.1 Basic Isosurface Navigation Control Primitive

The proposed isosurface AN control strategy establishes a default direction of travel that is perpendicular to the gradient vector and therefore roughly parallel to the local surface tangent plane; a corrective vector term directed towards the desired surface and proportional to the scalar error is added to this default vector. The net vector is used to specify the direction of travel, and a constant speed is used to set the magnitude of the vector. This strategy is a 3D extension of the contour following strategy used in Kitts et al. [1], which in turn was originally based on a field-implemented planar path-following controller presented in Kitts et al. [72].

Equations (12) and (13) specify the cluster translational velocity commands to implement this motion control strategy. In (12), the first term directs the cluster in a direction that is tangential to the desired surface given the cross product with the gradient vector \vec{g}_{grad} ; the specific direction is dictated by the choice of the navigation reference vector ${}^G\hat{n}$ and d_{orbit} which is set to +1 and -1 for counterclockwise and clockwise travel relative to ${}^G\hat{n}$, respectively. The second term in (12) guides the cluster along the gradient towards the desired surface in the event that it is not already on the surface. As part of this term, K_{surf} is a proportional corrective gain, s_{des} is the scalar value of the desired surface, and $s_{cluster}$ is the mean value of the cluster's scalar measurement s_j through s_4 .

$$\mathbf{v}_{b-isosurface} = d_{orbit} \cdot \frac{(\vec{g}_{grad} \times {}^G\hat{n})}{\|\vec{g}_{grad} \times {}^G\hat{n}\|} + K_{surf} \cdot (s_{des} - s_{cluster}) \cdot \vec{g}_{grad} \quad (12)$$

$$\dot{\hat{C}}_{4R}(1:3) = \begin{bmatrix} \dot{X}_B des \\ \dot{Y}_B des \\ \dot{Z}_B des \end{bmatrix} = S \cdot \frac{v_{b-isosurface}}{\|v_{b-isosurface}\|} \quad (13)$$

While cluster translational velocity commands are continuously varied based on the estimate of the local field gradient, all other cluster orientation and geometry variables, designated by $\dot{\hat{C}}_{4R}(4:16)$, are controlled to specified position setpoints through the use of a proportional position controller. This blending of proportional velocity and position control is achieved by the resolved-rate cluster-space controller expressed by (8) given the selector matrix $Q' = \text{diag}(1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0)$ and $Q = I - Q'$, where I is the identity matrix.

It is noted that the navigation reference vector ${}^G \hat{n}$ is a user specified travel reference vector, chosen to not be collinear with the local gradient vector. Section 5.4 discusses how this vector may be selected in order to support specific mission-level objectives.

5.2 Basic Isosurface Navigation Simulation

This section demonstrates high-fidelity simulations of isosurface navigation for two different navigation reference vector orientations as shown in Fig. 5-1. For each trial, the UAV cluster starts in the same location and travels to and then along the isosurface defined by $s = 8$ scalar units, a speed of $S = 3$ m/s, and a direction setting of $d_{orbit} = -1$. In trial A, a vertical navigation reference vector ${}^G \hat{n} = {}^G \hat{Z} = [0 \ 0 \ 1]^T$ is used, leading to a navigation path that orbits this vector in a clockwise fashion once the isosurface is reached. In trial B, a similar result is achieved for a horizontal navigation reference vector ${}^G \hat{n} = {}^G \hat{Y} = [0 \ 1 \ 0]^T$, leading to a navigation path that orbits this vector in a clockwise fashion once the isosurface is reached.

It is critical to note that this primitive controller executes closed loop control to move the cluster to and hold it along the desired scalar surface; however, motion within that surface is open loop. Hence, for both of these cases, motion of the cluster once it arrives at the isosurface wanders in the direction of navigation reference vector ${}^G \hat{n}$ while orbiting that vector.

To demonstrate performance for trial A in Fig. 5-1, Fig. 5-2 shows the average sensed scalar value of the formation increasing over time and settling at the desired isosurface scalar value as the cluster navigates to and then travels along the surface. Fig. 5-3 shows the alignment of the cluster's direction of motion with the

gradient vector, indicating how navigation transitioned from moving toward to moving along the surface as it neared the surface.

Formation control performance is indicated by noting that the RMS errors for the cluster size parameters L_{12} , L_{13} , and L_{B4} were 1.7 m, 1.8 m and 1.0 m, respectively; these are all in a range of 0.5-1.5 times the standard deviation of simulated position sensor error, indicating acceptable formation control.

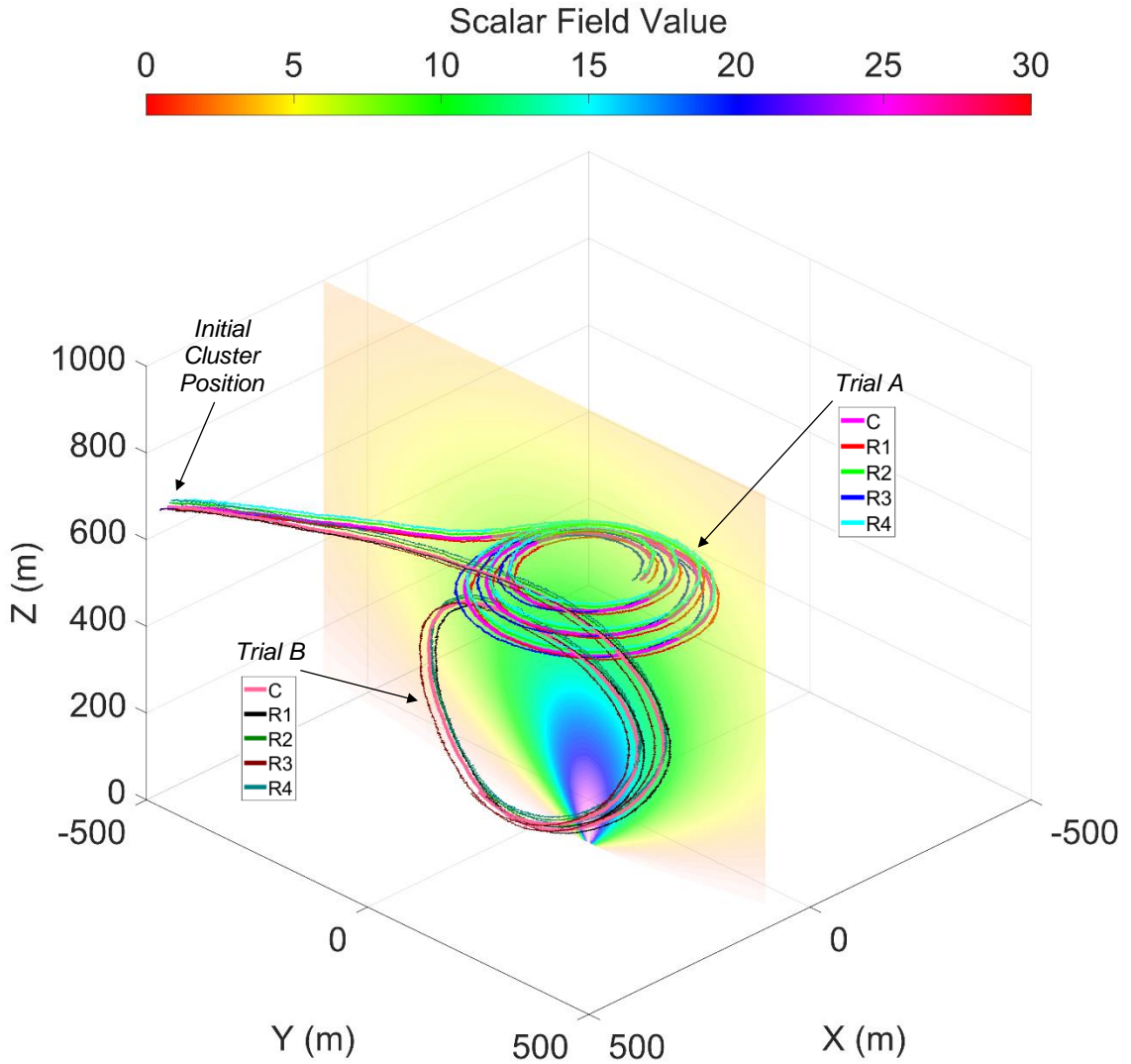


Figure 5-1: Basic isosurface AN with navigation reference vector set to ${}^c\hat{n} = {}^c\hat{z} = [0 \ 0 \ 1]^T$ and ${}^c\hat{n} = {}^c\hat{x} = [1 \ 0 \ 0]^T$ for trials A and B, respectively. The primitive controller executes closed loop control to move the cluster to and hold it along the desired scalar surface; however motion within the surface is not controlled.

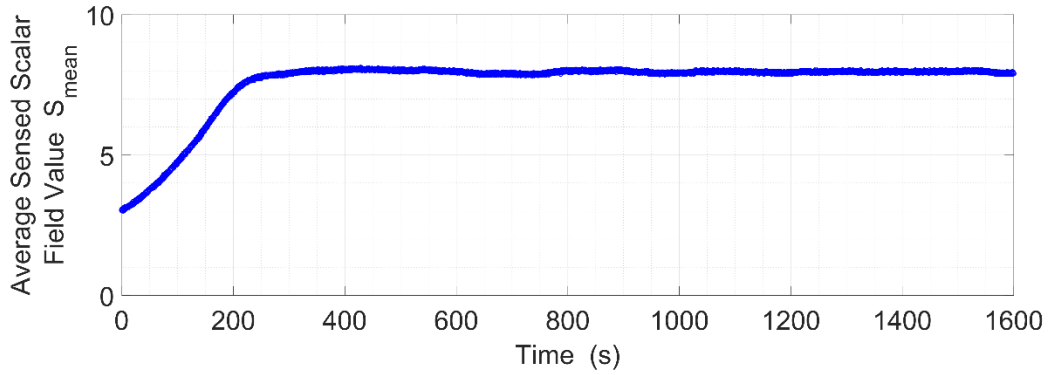


Figure 5-2: Time history of average sensed scalar field values during trial A as the formation navigates to and then travels along the surface

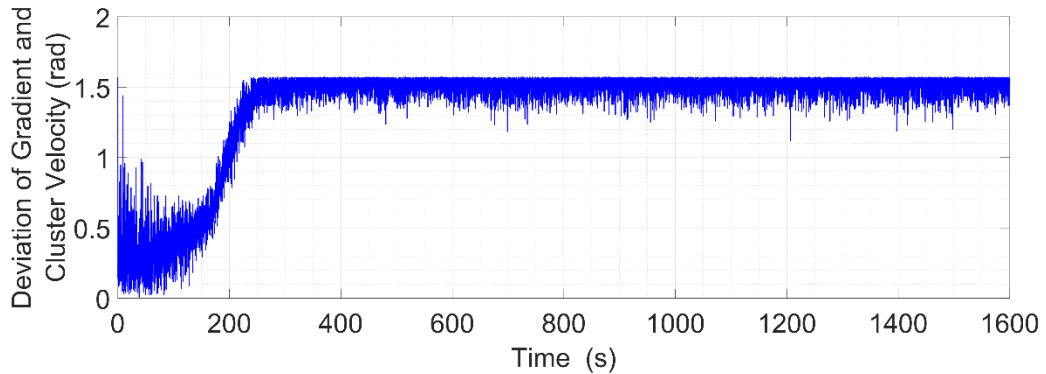


Figure 5-3: Angular deviation time history of cluster velocity from estimated gradient for trial A. As the cluster reaches the surface, the two vectors are perpendicular to each other, as is seen at approximately $t = 240$ s.

5.3 Isosurface Mapping State Machine

Isosurface mapping consists of navigating along the surface of interest in order to characterize its structure through a systematic ‘rendering’ of the surface made by contour traces in periodically spaced planar slices of the surface; this concept is notionally depicted in Fig. 5-4. Navigating along a contour in a given plane consists of simultaneously performing isosurface navigation while constrained to a plane that intersects the surface. Once that contour has been circumnavigated, the cluster moves to another parallel plane and repeats the process.

To achieve this, isosurface mapping requires the specification of a set of mission parameters. This is done by defining the desired isosurface scalar value s_{des} , a navigation reference vector perpendicular to the

desired planar slices ${}^G\hat{n}$, and the distance between planes Δn (along axis defined by ${}^G\hat{n}$ vector). The parameter d_{orbit} is used to specify the direction of travel about the navigation vector when contour following. This typically does not matter for this particular mission type, but for purposes of demonstrating functionality, this work uses +1 and -1, indicating counterclockwise and clockwise directions of travel, respectively. Finally, the desired cluster translational speed S must be specified.

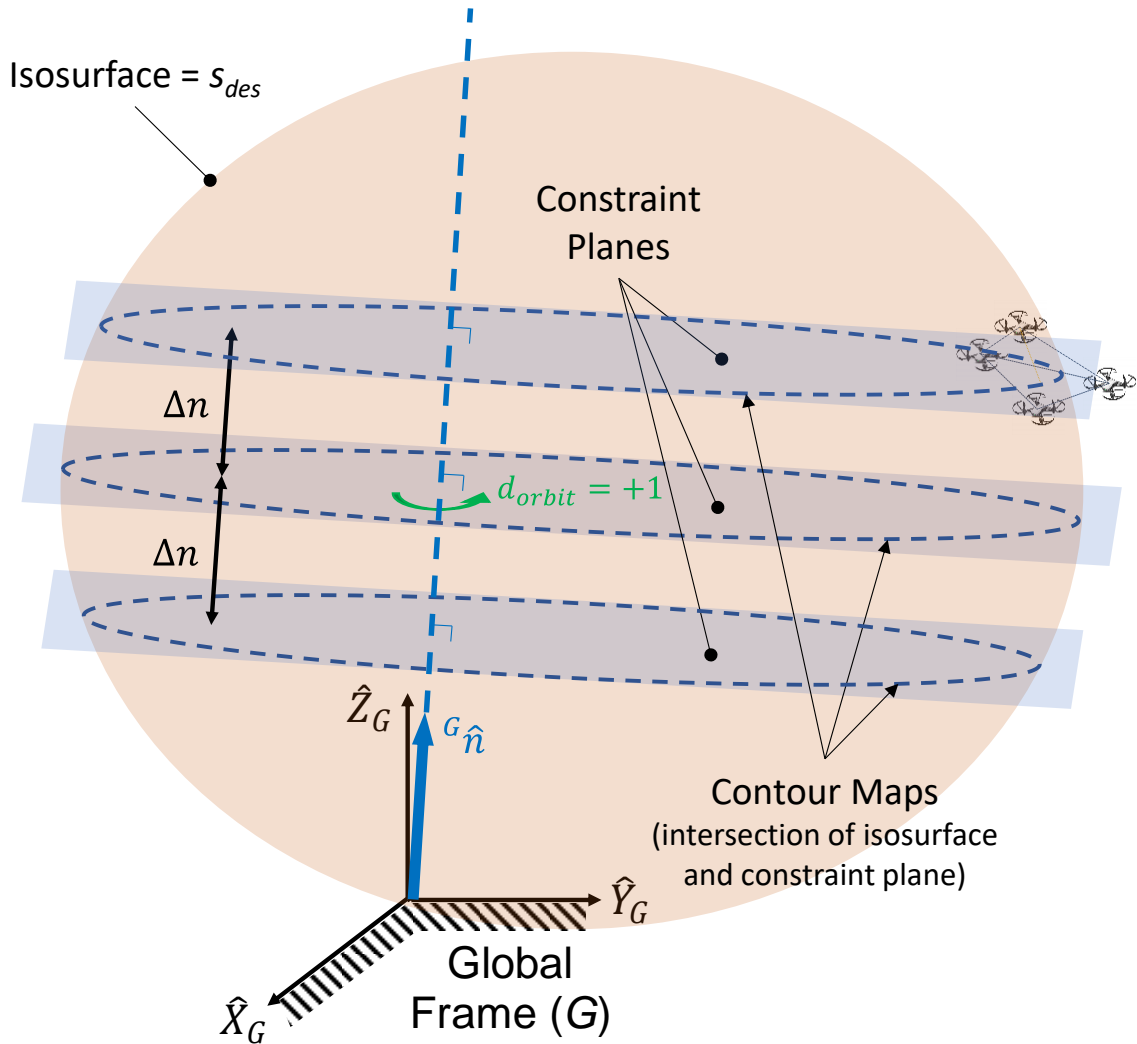


Figure 5-4: Isosurface mapping while constrained to a plane which intersects the surface

The control states that constitute the state machine for this mission make use of these mission parameters. State transitions indicate transitions between states as well as the change of parameter values for a given state.

Control States: Although isosurface mapping can be implemented in a variety of ways, the state diagram shown in Fig. 5-5 illustrates how it has been implemented for the results reported in this work.

The state machine sequences the cluster through two control states. The first control state, “State 1: Locate Surface”, uses only the basic isosurface navigation primitive discussed in section 5.1 to guide the cluster to the surface with a scalar value of interest s_{des} . The specific controller in this state is the isosurface controller provided in (12) and (13).

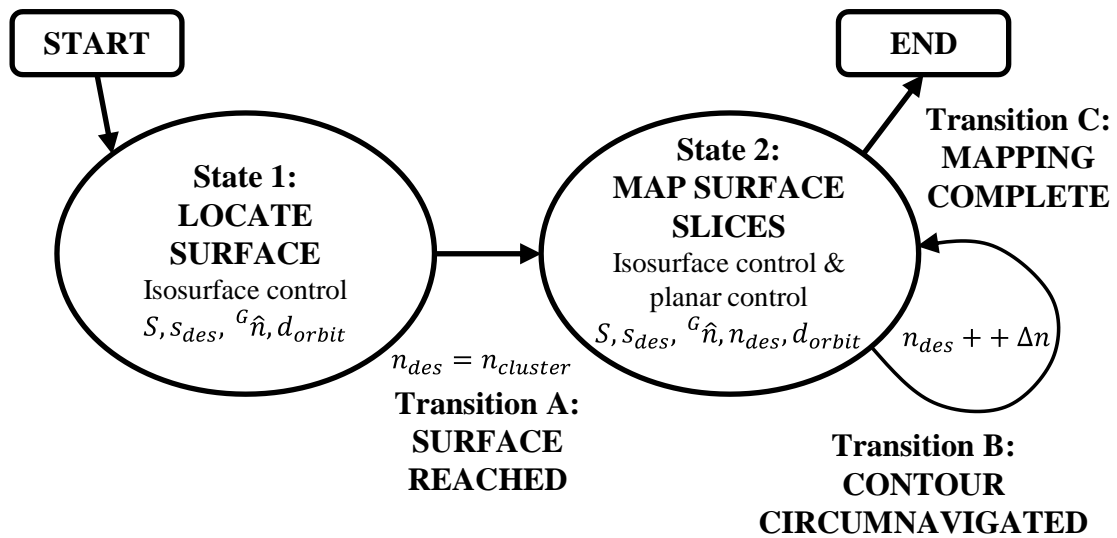


Figure 5-5: State machine for structured isosurface mapping where each of the two states consist of one or more AN-level controllers with specific control parameters. The n_{des} control parameter is updated at each state transition.

Once the surface is reached, “State 2: Map Surface Slices” enables navigation along the contour at the intersection of the desired surface and a specified plane; this is accomplished by a control equation that couples isosurface navigation as per (12) with a control term that guides the cluster to the intersection of the desired surface with a desired plane that intersects the surface. This multi-objective control equation for cluster

translational velocity is shown in (14). As with (12), the first two terms in (14) direct the cluster to move within the desired scalar surface. A new third term adds a corrective velocity component that exploits a degree of freedom of motion within the isosurface in order to guide the cluster to the contour line at the intersection of the isosurface and the desired plane. Within the third term, K_n is a proportional corrective gain and n_{des} is the planar location along the navigation reference vector ${}^G\hat{n}$.

$$v_{b-iso-mapping} = d_{orbit} \cdot \frac{(\vec{g}_{grad} \times {}^G\hat{n})}{\|\vec{g}_{grad} \times {}^G\hat{n}\|} + K_{surf} \cdot (s_{des} - s_{cluster}) \cdot \vec{g}_{grad} + K_n \cdot (n_{des} - P_{B_proj}) \cdot {}^G\hat{n} \quad (14)$$

As defined in (15), P_{B_proj} is the projection of the cluster point B position vector onto ${}^G\hat{n}$.

$$P_{B_proj} = \text{dot}(\vec{P}_{B_act}, {}^G\hat{n}) \quad (15)$$

The result is then used in (16) to generate desired velocity commands for cluster frame translational control, where S is the user specified translational speed.

$$\dot{\tilde{C}}_{4R}(1:3) = \begin{bmatrix} \dot{X}_{B_des} \\ \dot{Y}_{B_des} \\ \dot{Z}_{B_des} \end{bmatrix} = S \cdot \frac{v_{b-iso-mapping}}{\|v_{b-iso-mapping}\|} \quad (16)$$

While cluster translational velocity commands are continuously varied based on the estimate of the local field gradient, all other cluster orientation and geometry variables designated by $\dot{\tilde{C}}_{4R}(4:16)$, are controlled to specified position setpoints through the use of a proportional position controller. This blending of proportional velocity and position control is achieved by the resolved-rate cluster-space controller expressed by (8) given the selector matrix $Q' = \text{diag}(1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0)$ and $Q = I - Q'$, where I is the identity matrix.

For the initial execution of “State 2: Map Surface Slices”, n_{des} is set to be the value of P_{B_proj} upon arrival at the surface. Once a contour is completely circumnavigated within the desired plane, the state machine resets control in the same state with a new value of n_{des} that has been incremented by the specified value of Δn .

State Transition Criteria: Formal definitions of the state machine transition criteria consist of the following. The “Transition A: Surface Reached” criteria that transitions between the first and second states is

based on having $s_{cluster}$, the cluster's average scalar value, settle to within a prescribed threshold to the value s_{des} , as shown in (17).

$$\|s_{des} - s_{cluster}\| < s_{threshold} \quad (17)$$

Once in “State 2: Map Surface Slices”, the “Transition B: Contour Circumnavigated” criteria is used to change the planar slice setpoint n_{des} . To implement this, the criteria first requires the cluster to settle on the desired plane to within a prescribed threshold to n_{des} as shown in (18).

$$\|n_{des} - P_{B-proj}\| < n_{threshold} \quad (18)$$

Typically, during the first execution of the “State 2: Map Surface Slices” state, this criteria is immediately true given that n_{des} is set to the value of $n_{cluster}$ at the time of the state transition. However, for subsequent executions of the “State 2: Map Surface Slices” state, there is typically a transient period since the cluster must navigate from one plane to the next through a distance of Δn . Once (17) and (18) are satisfied, the state machine checks for a full revolution of travel about the ${}^G\hat{n}$ vector for each contour map slice. When the magnitude of the differences between the contour map start and end positions (δ for distance and ρ for angle) are within the prescribed thresholds shown in (19) and (20), the state machine increments the planar slice setpoint n_{des} per (21).

$$\|\delta_{start} - \delta_{end}\| < \delta_{threshold} \quad (19)$$

$$\|\rho_{start} - \rho_{end}\| < \rho_{threshold} \quad (20)$$

$$n_{des\ next} = n_{des\ current} + \Delta n \quad (21)$$

For this work, isosurface mapping continues until the cluster reaches the “end” of the surface in the given direction at which point, the estimated gradient and navigation reference vectors align with each other (other termination criteria are certainly possible). The “Transition C: Mapping Complete” termination condition is defined as the angle between the gradient \vec{g}_{grad} and the navigation reference vector ${}^G\hat{n}$ falling below a prescribed threshold as shown in (22).

$$\cos^{-1}\left(\frac{\vec{g}_{grad} \cdot {}^G\hat{n}}{\|\vec{g}_{grad}\| \|{}^G\hat{n}\|}\right) < \gamma_{threshold} \quad (22)$$

5.4 Isosurface Mapping Mission Simulation

Execution of isosurface mapping missions using a four-robot cluster is demonstrated using the state-based strategy described in section 5.3. The technique is demonstrated in three scenarios, each “slicing” the isosurface in a different planar orientation, with those planes perpendicular to the ${}^G\hat{X}$, ${}^G\hat{Y}$, and ${}^G\hat{Z}$ axes. The four-robot cluster was held in the shape of a tetrahedron with edges on the order of 25 to 35 m, cluster translational speeds of 3 to 5 m/s (selected to represent Parrot AR.Drone UAV cruising speed range), and both clockwise and counterclockwise circumnavigation. Navigation occurred in workspaces with sides up to 1,000 m in length. The simulated scalar fields in this section are documented in Appendix D.

Scenario A:

Scenario A involves mapping a symmetric vertical plume using horizontal contour slices. State machine control parameters used for this scenario are summarized in Table 5-1. Fig. 5-6 shows cluster motion, which starts using the basic State 1 navigation control primitive to locate the $s = 7$ scalar units isosurface. Upon arriving at the isosurface, n_{des} is initialized at the current value of 711 m, defining the first constraint plane perpendicular to the vertical navigation reference vector, ${}^G\hat{n}$. The cluster then navigates using the State 2 control law in order to circumnavigate the contour defined by the intersection of the isosurface and the constraint plane. The cluster then repeats this circumnavigation process with incremental planar offsets of $\Delta n = -110$ m until the termination condition is met.

Performance is characterized by the time histories in Fig. 5-7 which show a) the incremental change in altitude as isosurface slices are mapped, and b) the ability of the cluster to navigate on the desired surface, with transient deviations each time the cluster moves to a new plane. Formation control performance, as the cluster navigates, can be evaluated based on the RMS errors of the cluster size parameters. RMS errors for L_{12} , L_{13} , and L_{B4} were 2.5 m, 2.4 m and 2.9 m, respectively. These results are in a range of 1.0-2.0 times the standard deviation of position error, indicating acceptable formation control given the levels of sensor noise and the size of the cluster.

TABLE 5-1: State Machine Control Parameters for Scenario A

$S = 5 \text{ m/s}$
$s_{des} = 7 \text{ units}$
${}^G\hat{n} = [0 \ 0 \ 1]^T$
$\Delta n = -110 \text{ m}$
$d_{orbit} = -1$
$s_{threshold} = 0.25 \text{ units}$
$n_{threshold} = 10 \text{ m}$
$\delta_{threshold} = 50 \text{ m}$
$\rho_{threshold} = 10^\circ$
$\gamma_{threshold} = 10^\circ$

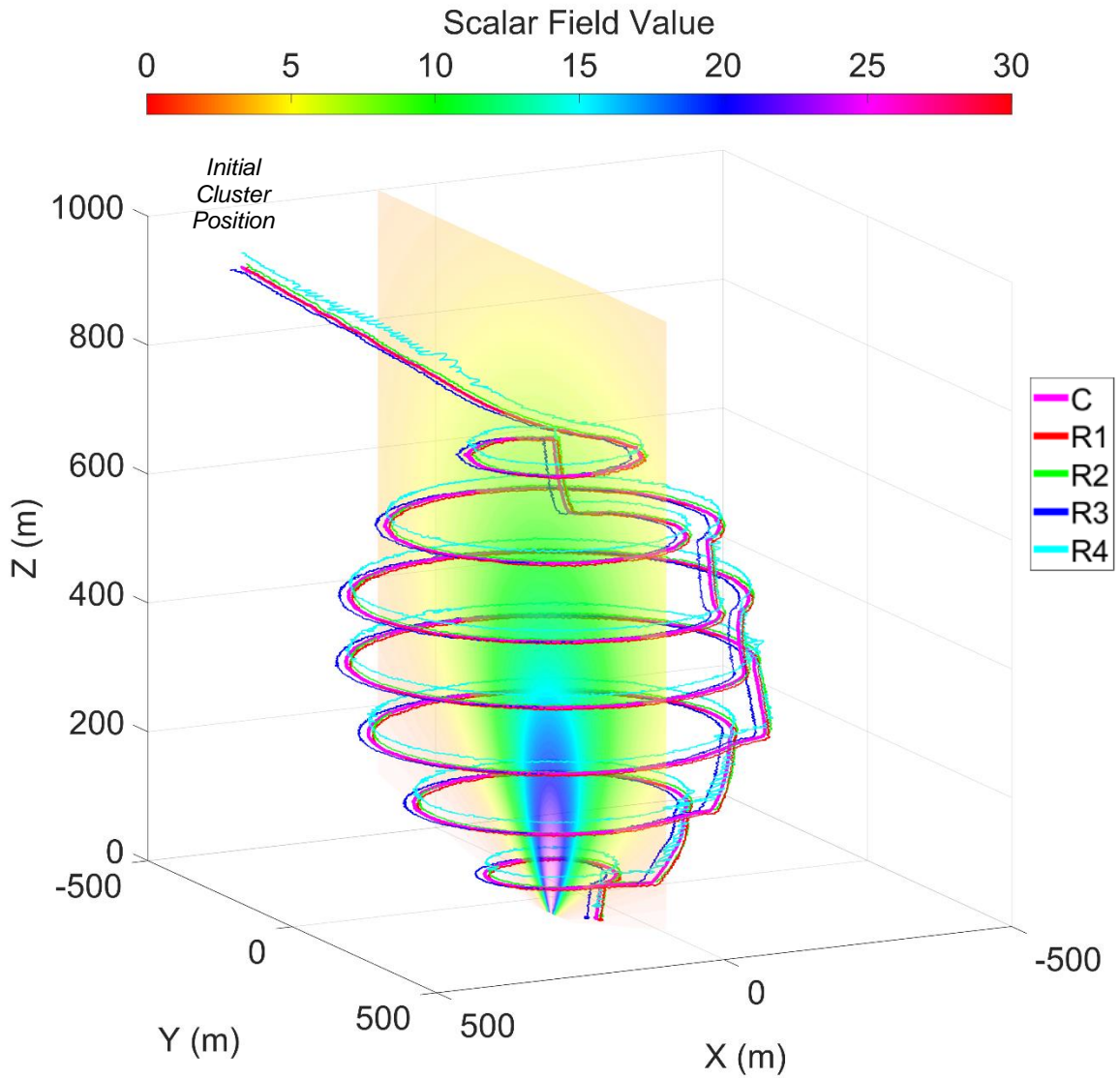


Figure 5-6: Isosurface mapping of symmetric plume with navigation reference vector set to ${}^G\hat{n} = {}^G\hat{z} = [0 \ 0 \ 1]^T$ and constraint planes 110 m apart that are parallel to the ground.

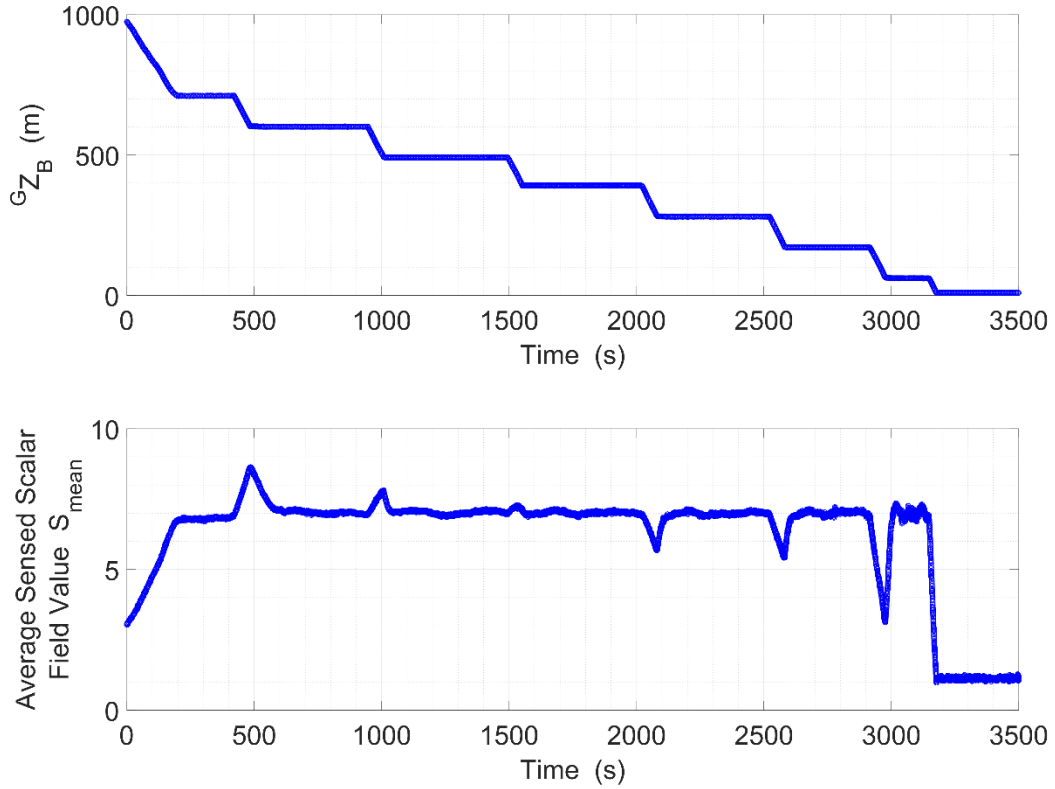


Figure 5-7: Time histories of the cluster Z-coordinate and average measured scalar field level.

Scenario B:

Scenario B involves mapping a more complex, asymmetric plume using vertical contour slices that are perpendicular to the ${}^G \hat{Y}$ axis. State machine control parameters used for this scenario are summarized in Table 5-2. Fig. 5-8 shows cluster motion starting with the basic State 1 navigation control primitive to locate the $s = 200$ scalar units isosurface. Upon arrival at the isosurface, n_{des} is initialized to the current value of 18 m which defines the first constraint plane perpendicular to the horizontal navigation reference vector, ${}^G \hat{n}$. The cluster then navigates using the State 2 control law in order to circumnavigate the contour defined by the intersection of the isosurface and the constraint plane. The cluster repeats this circumnavigation process with incremental planar offsets of $\Delta n = 60$ m until the termination condition is met.

Performance is characterized by the time histories in Fig. 5-9 which show a) the incremental change in altitude as isosurface slices are mapped, and b) the ability of the cluster to navigate on the desired surface, with transient deviations each time the cluster moves to a new plane. Formation control performance, as the cluster navigates, can be evaluated based on the RMS errors of the cluster size parameters. RMS errors for

TABLE 5-2: State Machine Control Parameters for Scenario B

$S = 3 \text{ m/s}$
$s_{des} = 200 \text{ units}$
${}^G\hat{n} = [0 \ 1 \ 0]^T$
$\Delta n = 60 \text{ m}$
$d_{orbit} = +1$
$s_{threshold} = 0.5 \text{ units}$
$n_{threshold} = 10 \text{ m}$
$\delta_{threshold} = 25 \text{ m}$
$\rho_{threshold} = 10^\circ$
$\gamma_{threshold} = 10^\circ$

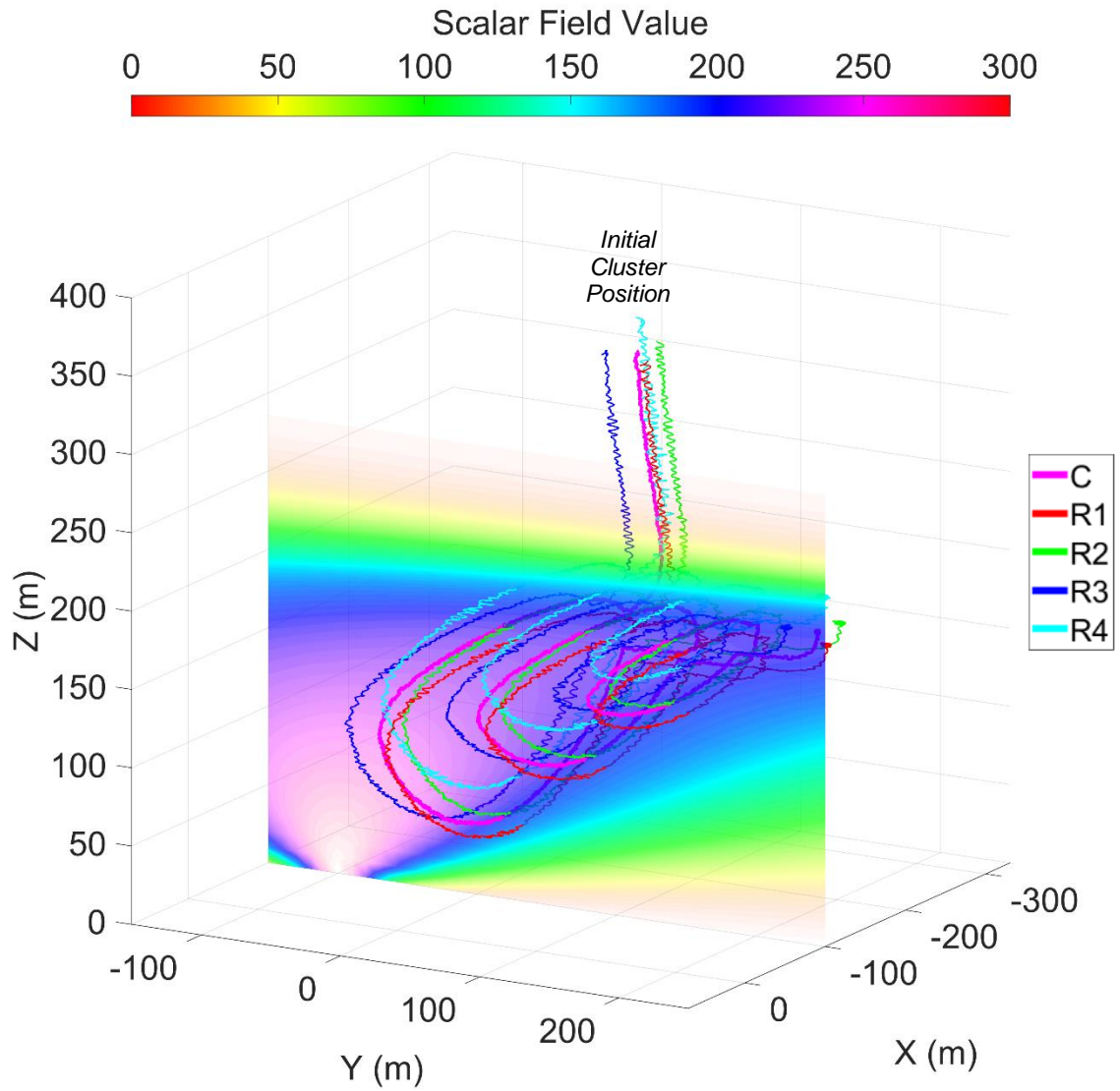


Figure 5-8: Isosurface mapping of an asymmetric plume with navigation reference vector set to ${}^G\hat{n} = {}^G\hat{v} = [0 \ 1 \ 0]^T$ and constraint planes 60 m apart that are perpendicular to the ${}^G\hat{v}$ axis.

L_{12} , L_{13} , and L_{B4} were 1.7 m, 2.3 m and 1.2 m, respectively. These results are in a range of 0.5-1.5 times the standard deviation of position error, indicating acceptable formation control given the levels of sensor noise and the size of the cluster.

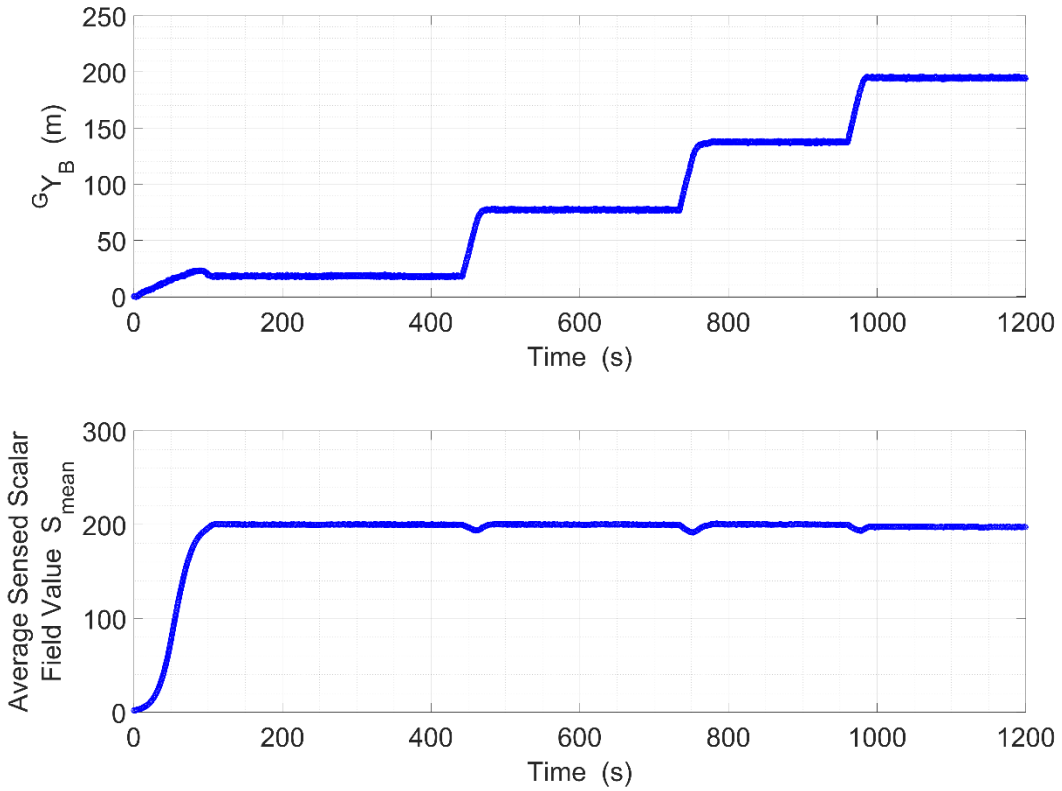


Figure 5-9: Time histories of the cluster Y-coordinate and average measured scalar field level.

Scenario C:

Scenario C further increases scalar field complexity through the use of overlapping scalar signals. Table 5-3 summarizes the state machine control parameters for scenario C1 while Fig. 5-10 shows planar slices that are spaced every 100 m and are perpendicular to the ${}^G \hat{X}$ axis, with a desired isosurface scalar field value of $s = 8$ scalar units. Fig. 5-11 contains time histories of the cluster’s position along the ${}^G \hat{X}$ axis and the average measured scalar value. RMS errors for L_{12} , L_{13} , and L_{B4} were 2.3 m, 2.3 m and 2.1 m, respectively. These results are in a range of 1.0-1.5 times the standard deviation of position error, indicating acceptable formation control given the levels of sensor noise and the size of the cluster.

TABLE 5-3: State Machine Control Parameters for Scenario C1

$S = 4 \text{ m/s}$
$s_{des} = 8 \text{ units}$
${}^G\hat{n} = [1 \ 0 \ 0]^T$
$\Delta n = 100 \text{ m}$
$d_{orbit} = -1$
$s_{threshold} = 0.5 \text{ units}$
$n_{threshold} = 10 \text{ m}$
$\delta_{threshold} = 60 \text{ m}$
$\rho_{threshold} = 10^\circ$
$\gamma_{threshold} = 10^\circ$

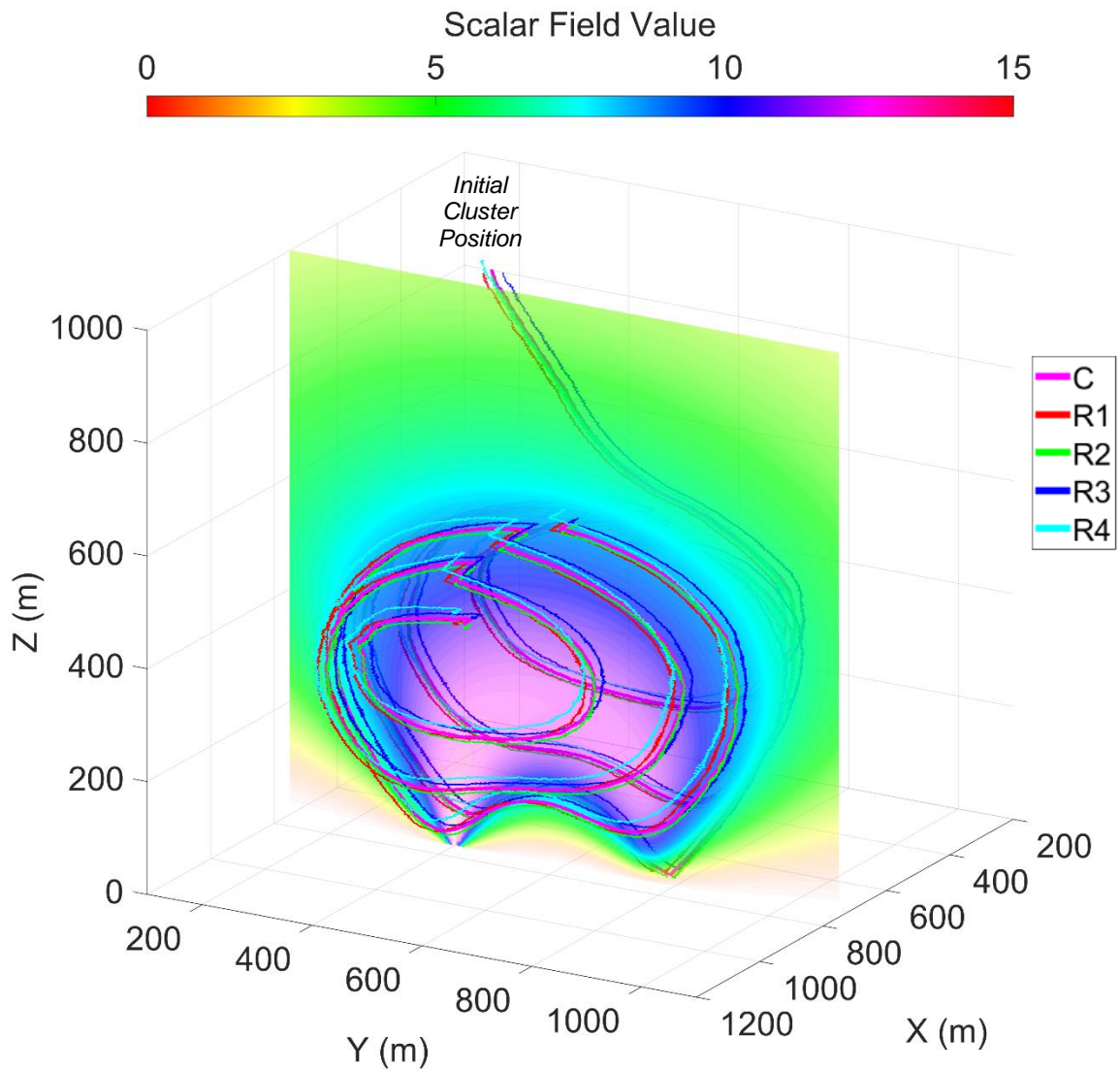


Figure 5-10: Isosurface mapping of overlapping plumes with navigation reference vector set to ${}^G\hat{n} = {}^G\hat{X} = [1 \ 0 \ 0]^T$ and constraint planes 100 m apart that are perpendicular to the ${}^G\hat{X}$ axis.

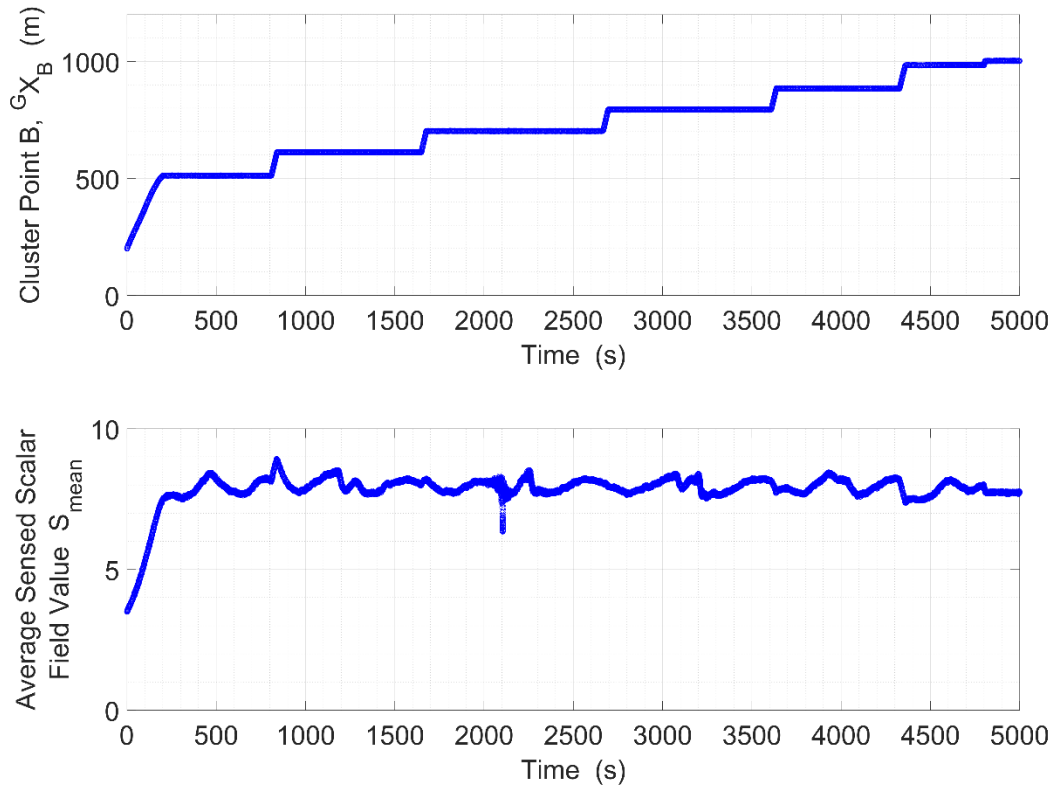


Figure 5-11: Time histories of the cluster X-coordinate and average measured scalar field level.

For the same set of scalar field overlapping plumes, scenario C2 demonstrates structured isosurface mapping with planar slices perpendicular to the ${}^G \hat{Y}$ axis. Table 5-4 summarizes the state machine control parameters where the desired isosurface scalar field value of $s = 8$ scalar units is unchanged from scenario C1. Fig 5-12 shows the planar slices which are spaced every 100 m and perpendicular to the ${}^G \hat{Y}$ axis. Formation length RMS errors for L_{12} , L_{13} , and L_{B4} were 2.0 m, 2.0 m and 1.8 m, respectively and in-family with previous trials.

TABLE 5-4: State Machine Control Parameters for Scenario C2

$S = 4 \text{ m/s}$
$s_{des} = 8 \text{ units}$
${}^G\hat{n} = [0 \ 1 \ 0]^T$
$\Delta n = 100 \text{ m}$
$d_{orbit} = +1$
$s_{threshold} = 0.5 \text{ units}$
$n_{threshold} = 10 \text{ m}$
$\delta_{threshold} = 60 \text{ m}$
$\rho_{threshold} = 10^\circ$
$\gamma_{threshold} = 10^\circ$

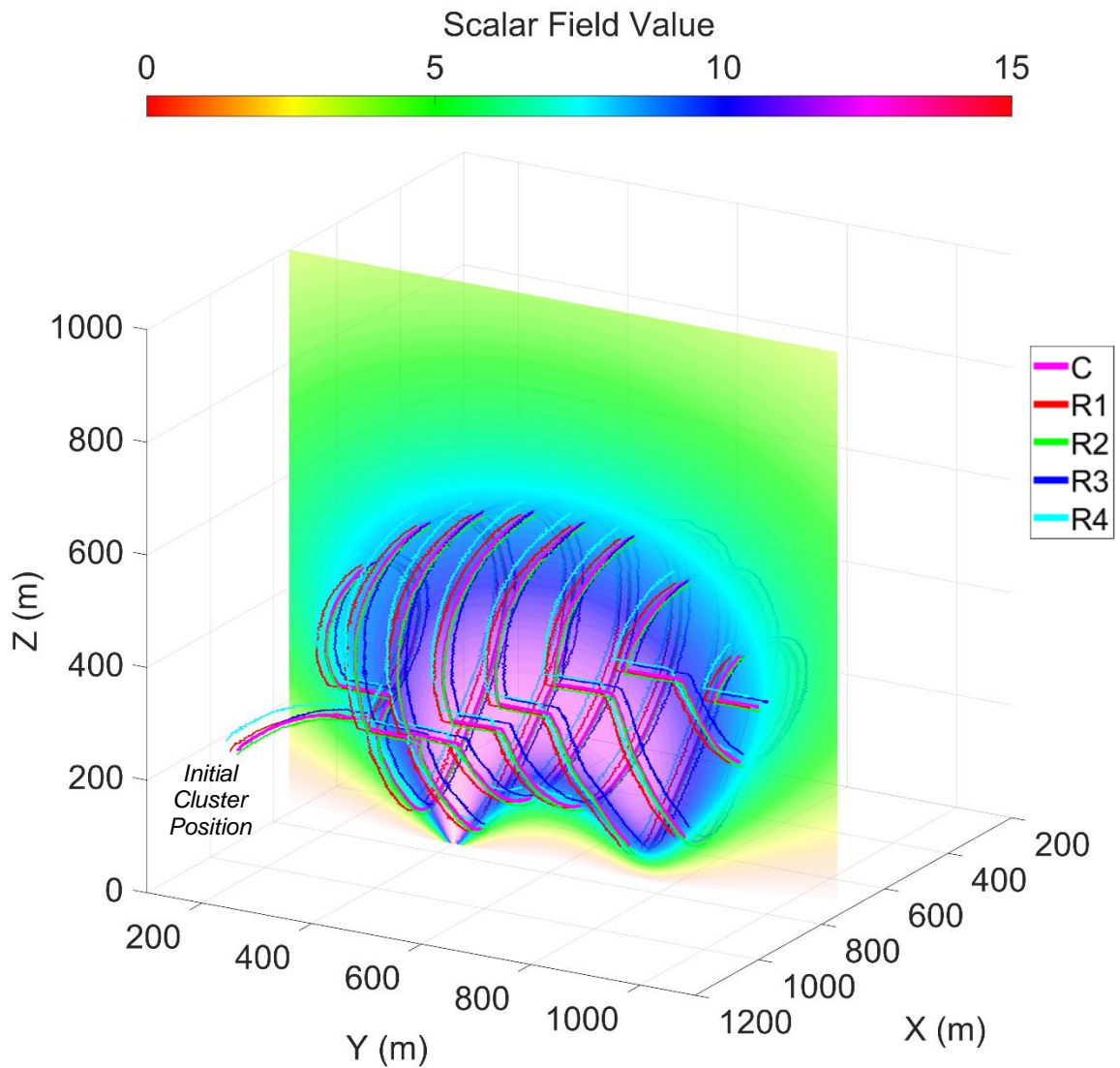


Figure 5-12: Isosurface mapping of overlapping plumes with navigation reference vector set to ${}^G\hat{n} = {}^G\hat{Y} = [0 \ 1 \ 0]^T$ and constraint planes 100 m apart that are perpendicular to the ${}^G\hat{Y}$ axis.

6.0 Downstream Plume Following

Being able to follow a plume outward, away from its source, has real-world applications such as determining the impact zone from a pollution source. In this work, plume following outward with a differential-based control strategy is implemented using synchronized and spatially relevant sensor measurements from each vehicle in the cluster. It is important to note that following a plume inwards/towards a source can be accomplished via a gradient-ascent extrema seeking control primitive, such as the one presented in section 4.3; however, following a plume outward, away from the source, is not possible using a gradient-descent algorithm.

The strategy for 3D downstream plume following is an extension of 2D ridge descent in Kitts et al. [1], a depiction of which is shown in Fig. 6-1. For the 2D control primitive, a ridge straddling strategy is used. In particular, assuming that the cluster is already straddling the ridge, differential scalar signals are generated by Robots 2 through 5 to position and orient the cluster with respect to the ridge.

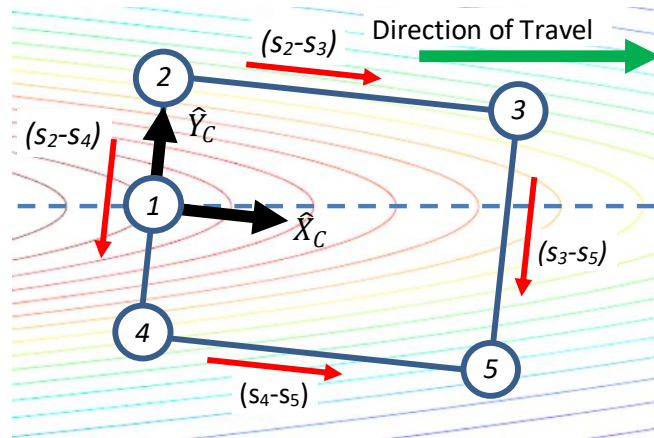


Figure 6-1: Differential drive compensation signals for ridge following in 2D with a five-robot cluster

The differential between Robots 2 and 4 and Robots 3 and 5 are used for lateral positioning and orientation control, while the differentials between Robots 2 and 3 and Robots 4 and 5 set the direction of travel. Robot 1 is used to ensure that the cluster is, in fact, well-positioned on the ridge with the expectation that its sensed scalar value is greater than those of Robots 2 and 4.

A ridge in 2D space becomes a plume in a 3D space; a simple example of this may be visualized by rotating the contours of Fig. 6-1 about the blue dashed line in that figure. Accordingly, the planar ridge

descent strategy is extended by complementing the five-robot formation in Fig. 6-1 with an additional formation that is oriented by rotating the formation by 90° about its x_c unit vector. Since Robot 1 in each portion of the formation is coincident, only one robot is used in this position. This approach gives rise to a nine-robot formation which, when properly positioned on a plume, “straddles” the plume in perpendicular planes. This allows AN to be implemented without requiring extraneous motions necessary to characterize the nature of the local field. The cluster is sized based on the considerations discussed in section 7 and detailed in Kitts et al. [1]. Fig. 6-2 shows a simple representation of a formation properly straddling a plume.

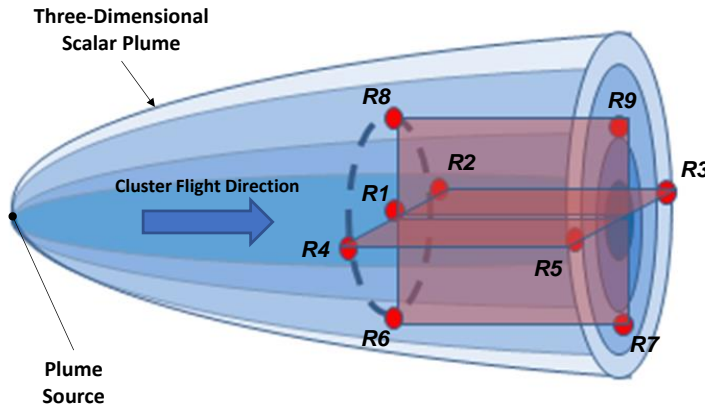


Figure 6-2: Plume following outward (i.e., ridge descent in 3D) using a nine-robot cluster in a right-rectangular prism formation

Given this plume following approach, this section presents the nine-robot cluster definition and a differential-based AN control law assuming the cluster is straddling the plume. A mission-level state machine is then described to first lead the cluster to the plume and then compel the cluster to follow the plume. Finally, high-fidelity simulations are used to show a nine-UAV cluster successfully navigating several plumes.

6.1 Nine-Robot Prismatic Cluster

The nine-robot cluster can be thought of as two intersecting, planar five-robot formations that share one of the robots, as shown in Fig. 6-3. The pose of each robot in the figure is represented by its robot-space position and orientation variables listed in Table 6-1. The robot-space pose vector of the group \vec{R}_{9R} contains these variables for each of the nine robots. Note that the robot-space velocity vector $\dot{\vec{R}}_{9R}$ is the time derivative of \vec{R}_{9R} .

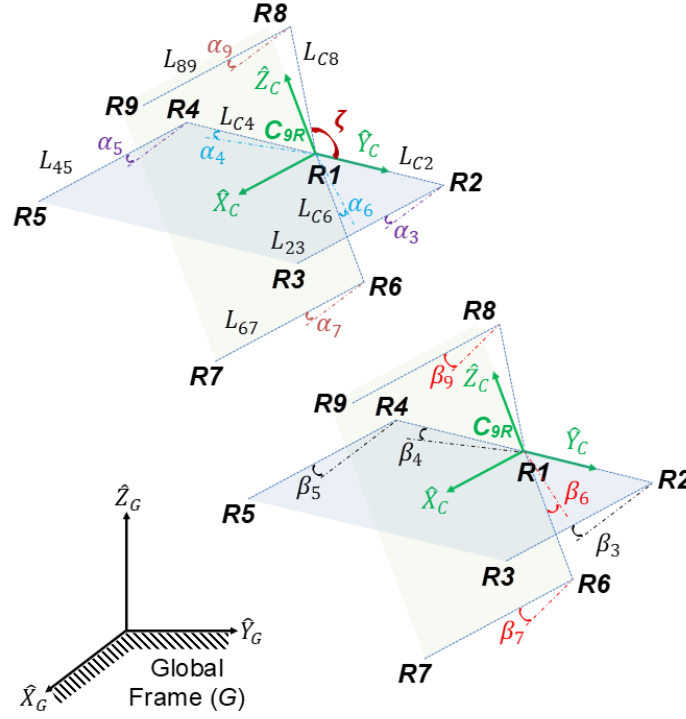


Figure 6-3: Nine-robot formation lengths & internal angles, α is perpendicular to $\hat{X}_c\hat{Y}_c$ or $\hat{X}_c\hat{Z}_c$ planes and β is parallel to $\hat{X}_c\hat{Y}_c$ or $\hat{X}_c\hat{Z}_c$ planes. Variables shown in separate formations for clarity.

For this study, the origins of cluster frame C_{9R} and Robot 1 frame $R1$ are coincident and 21 shape variables are used to define the geometries of the four, two-robot serial chains that kinematically extend from Robot 1, as shown in Fig. 6-3 and listed in Table 6-2. The cluster-space position vector \vec{C}_{9R} consists of the variables in Table 6-2 while the cluster-space velocity vector $\dot{\vec{C}}_{9R}$ is the time derivative of \vec{C}_{9R} . The full listing of vectors \vec{R}_{9R} and $\dot{\vec{C}}_{9R}$ are provided in Appendix C.

TABLE 6-1: Nine-Robot Formation Robot-Space Variables

Robot positions, $n = 1-9$	Gx_n Gy_n Gz_n
Robot yaw angles, $n = 1-9$	${}^G\psi_n$

TABLE 6-2: Nine-Robot Formation Cluster-Space Variables

Cluster position variables	GX_c GY_c GZ_c
Cluster orientation variables	${}^G\phi_c$ ${}^G\theta_c$ ${}^G\psi_c$
Robot relative headings, $n = 1-9$	${}^C\psi_n$
Cluster shape variables, distances	L_{C2} L_{23} L_{C4} L_{45} L_{C6} L_{67} L_{C8} L_{89}
Cluster shape variables, angles	α_3 α_4 α_5 α_6 α_7 α_9 β_3 β_4 β_5 β_6 β_7 β_9
Cluster shape, angle btw \hat{Y}_c & L_{C8}	ζ

from Robots 2, 4, 6, and 8. Furthermore, this work typically requires that Robot 1's value exceeds that of Robots 2, 4, 6, and 8 by a certain threshold to ensure not only proper positioning of the cluster, but also that the plume feature is "prominent" enough to be worth following. This prominence threshold criteria is formally expressed in (27).

$$(s_1 - \max[s_2, s_4, s_6, s_8]) > s_{prom\ threshold} \quad (27)$$

6.3 Plume Location and Descent State Machine

Autonomous plume location and descent consists of finding the source of a plume and following it out to the impact area while being able to reacquire the plume should the cluster inadvertently maneuver "off" the feature. Execution of this mission requires the specification of a set of mission parameters, such as the minimum plume prominence value. As before, the user specifies desired constant cluster translational and rotational speeds, S and R , respectively. In general, a right-rectangular prism formation is used to obtain distributed data in all three dimensions in order to attain accurate differential signals between robots as illustrated in Fig. 6-3. The size of the prism is selected to appropriately straddle a plume with a particular size of interest.

Control States: The state diagram shown in Fig. 6-4 illustrates the plume location and descent approach that has been implemented for the results reported in this work.

The primary state in this strategy is "State 2: Follow Plume Outward", in which the cluster is properly straddling the plume and implementing the plume following AN control primitive described in section 6.2.

The first state, "State 1: Locate Plume", is used to initially move to and straddle the plume. This state uses the plume following AN control primitive in section 6.2 with the exception that the X-component in (23) is negative, which enables gradient ascent in that dimension. This strategy moves the cluster towards a local maximum while aligning the cluster with a plume stemming from that source, if one exists. If the cluster successfully locates a plume that meets the prespecified minimum prominence threshold, the state machine proceeds to "State 2: Follow Plume Outward".

The third state, "State 3: Reacquire Plume", is used if the cluster inadvertently maneuvers "off" the plume while descending it. This is when the cluster is still straddling and aligned with the plume, but not centered in it. The strategy adopted in this case is to reverse the longitudinal direction of motion, as done in

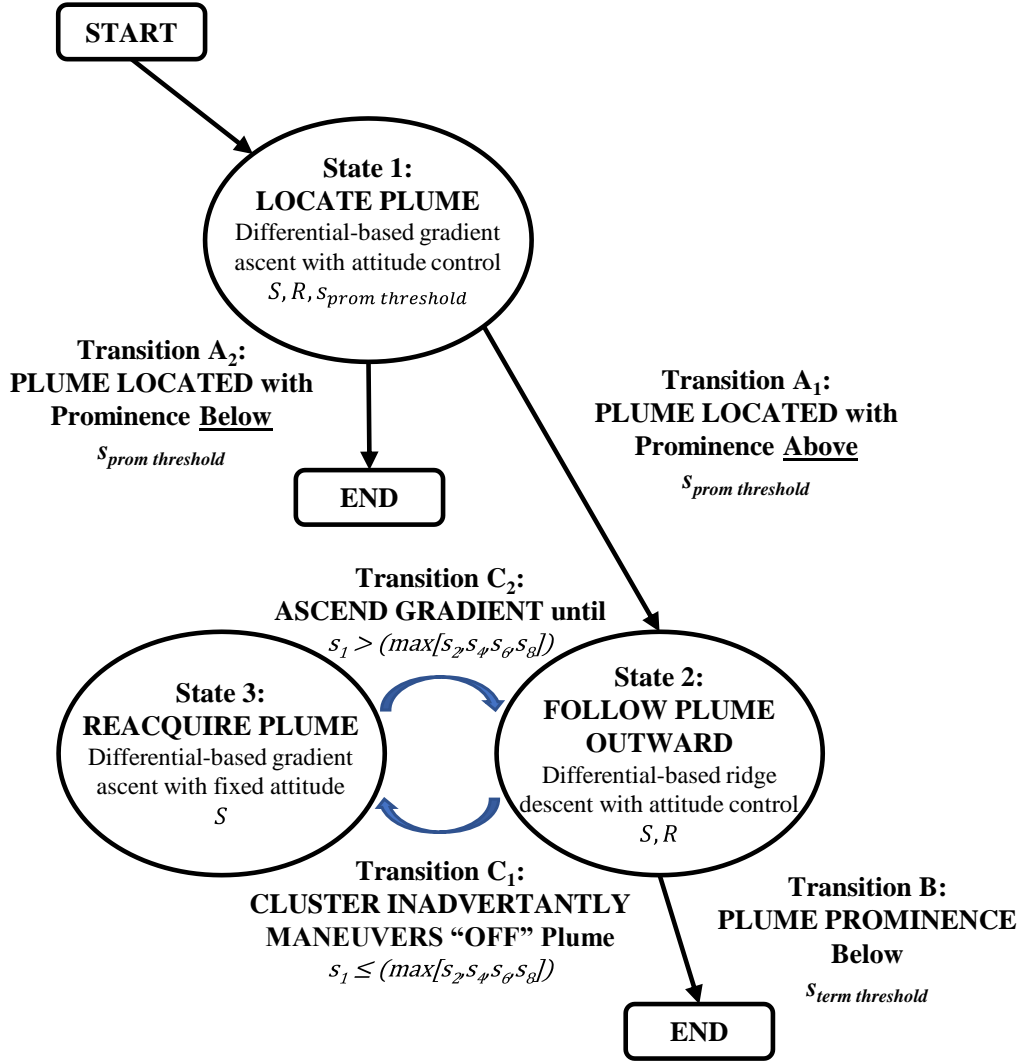


Figure 6-4: State machine for plume location and descent. Each of the three states consist of one or more AN-level controllers with specific control parameters

“State 1: Locate Plume”, by making the X-component in (23) negative and holding the cluster orientation constant. As shown in (28), cluster rotation is computed using a simple proportional controller, where K_{rot_hold} is the rotational gain, to hold the current orientation of the cluster in lieu of (24).

$$\vec{D}_{rot_hold} = \begin{bmatrix} 0 \\ K_{rot_hold} \cdot ({}^G\theta_{C_hold} - {}^G\theta_{C_act}) \\ K_{rot_hold} \cdot ({}^G\Psi_{C_hold} - {}^G\Psi_{C_act}) \end{bmatrix} \quad (28)$$

State Transition Criteria: Formal definitions of the state machine transition criteria consist of the following. The criteria for “Transition A₁” that must be satisfied to move between “State 1: Locate Plume” and

“State 2: Follow Plume Outward” are based on the magnitude of the differentials being below a prescribed threshold as shown in (29), the difference in the measured scalar field value from Robot 1 and the maximum from Robots 2, 4, 6, and 8 by a prescribed minimum threshold previously shown in (27), and the cluster heading alignment with the downstream direction of the plume to within a prescribed threshold as shown in (30) to (33). Note that (27) defines the minimum plume “prominence” which functions as a “go” or “no go” condition (i.e., if the plume contains a feature of interest) as to whether the cluster should proceed with following the plume outward. For compactness, (33) uses the notation C(#) and S(#) for cosine and sine functions, respectively.

$$\left\| \left[\begin{array}{c} \left(\frac{s_3+s_5}{2} \right) - \left(\frac{s_2+s_4}{2} \right) \\ \left(\frac{s_2+s_3}{2} \right) - \left(\frac{s_4+s_5}{2} \right) \\ \left(\frac{s_8+s_9}{2} \right) - \left(\frac{s_6+s_7}{2} \right) \end{array} \right] \right\| < \Delta_{threshold} \quad (29)$$

$$\| {}^G\Psi_{C des} - {}^G\Psi_{C act} \| < \Psi_{C threshold} \quad (30)$$

where
$${}^G\Psi_{C des} = ATAN2\left(\frac{-D_y diff}{-D_x diff}\right) \quad (31)$$

$$\vec{D}_{diff} = \begin{bmatrix} D_x diff \\ D_y diff \\ D_z diff \end{bmatrix} = {}^C R \cdot \begin{bmatrix} (s_3 - s_2) + (s_5 - s_4) \\ (s_2 - s_4) + (s_3 - s_5) \\ (s_8 - s_6) + (s_9 - s_7) \end{bmatrix} \quad (32)$$

$${}^C R = \begin{bmatrix} C\Psi_c C\theta_c & C\Psi_c S\theta_c S\phi_c - C\phi_c S\Psi_c & C\phi_c S\theta_c C\Psi_c + S\Psi_c S\phi_c \\ S\Psi_c C\theta_c & S\phi_c S\theta_c S\Psi_c + C\Psi_c C\phi_c & S\Psi_c S\theta_c C\phi_c - S\phi_c C\Psi_c \\ -S\theta_c & C\theta_c S\phi_c & C\theta_c C\phi_c \end{bmatrix} \quad (33)$$

While in “State 2: Follow Plume Outward”, in the event that the cluster inadvertently maneuvers “off” the feature, the state machine “Transition C₁” criteria to move to “State 3: Reacquire Plume” is shown in (34).

$$s_1 \leq (\max[s_2, s_4, s_6, s_8]) \quad (34)$$

While in “State 3: Reacquire Plume”, state machine “Transition C₂” criteria to move back to “State 2: Follow Plume Outward” is shown in (35).

$$s_1 > (\max[s_2, s_4, s_6, s_8]) \quad (35)$$

Following the plume outward continues until the cluster reaches a point where the plume prominence falls below a prescribed threshold with “Transition B” criteria shown in (36).

$$s_{term threshold} \leq (\max[s_2, s_4, s_6, s_8] - s_1) \quad (36)$$

6.4 Downstream Plume Following Mission Simulation

This section demonstrates execution of mission scenarios to locate and descend a plume using a nine-robot cluster using the state-based strategy described in section 6.3. As previously discussed, the objective is to have the cluster locate a plume (of unknown size, location, and intensity) with a minimum prominence value and to move along it in the direction away from the source, reacquire the plume if it inadvertently maneuvers “off” the feature, and to hold its position and attitude when it reaches a point where the prominence falls below a prescribed threshold. The nine-robot cluster was held in the shape of a right-rectangular prism with edges on the order of 25 to 35 m, cluster translational speed of 5 m/s, and cluster rotational speed of 5 deg/s. Navigation occurred in workspaces with sides up to 400 m in length. The simulated scalar fields in this section are documented in Appendix D.

Scenario A:

Downstream plume following simulation scenario A utilizes a simulated plume with a “tail” of several hundred meters and a constant commanded cluster geometry held in the shape of a right-rectangular prism with formation variables shown in Table 6-3. State machine control parameters for this simulation are summarized in Table 6-4. Fig. 6-5 illustrates the cluster flight paths for four different trials, where each trail has a different starting point. For clarity, only the cluster origin positions are shown for each trial. The figure shows the cluster successfully navigating to and then moving along the plume in the direction away from the source in each trial.

TABLE 6-3: Nine-Robot Formation Shape Variables for Scenario A

$L_{C2}, L_{23}, L_{C4}, L_{45}$	35 m
$L_{C6}, L_{67}, L_{C8}, L_{89}$	35 m
ζ	90°
$\alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_9$	0°
$\beta_3, \beta_4, \beta_5, \beta_6, \beta_7, \beta_9$	0°

TABLE 6-4: State Machine Control Parameters for Scenario A

Cluster Translational Speed	$S = 5 \text{ m/s}$
Cluster Rotational Speed	$R = 5 \text{ deg/s}$
Plume Prominence Threshold (Start)	$S_{prom \text{ threshold}} = 20 \text{ units}$
Differentials Threshold (plume location)	$\Delta_{\text{threshold}} = 10 \text{ units}$
Heading Alignment Threshold (prior to following plume outward)	$\Psi_{C \text{ threshold}} = 15^\circ$
Plume Prominence Threshold (End)	$S_{term \text{ threshold}} = 2 \text{ units}$

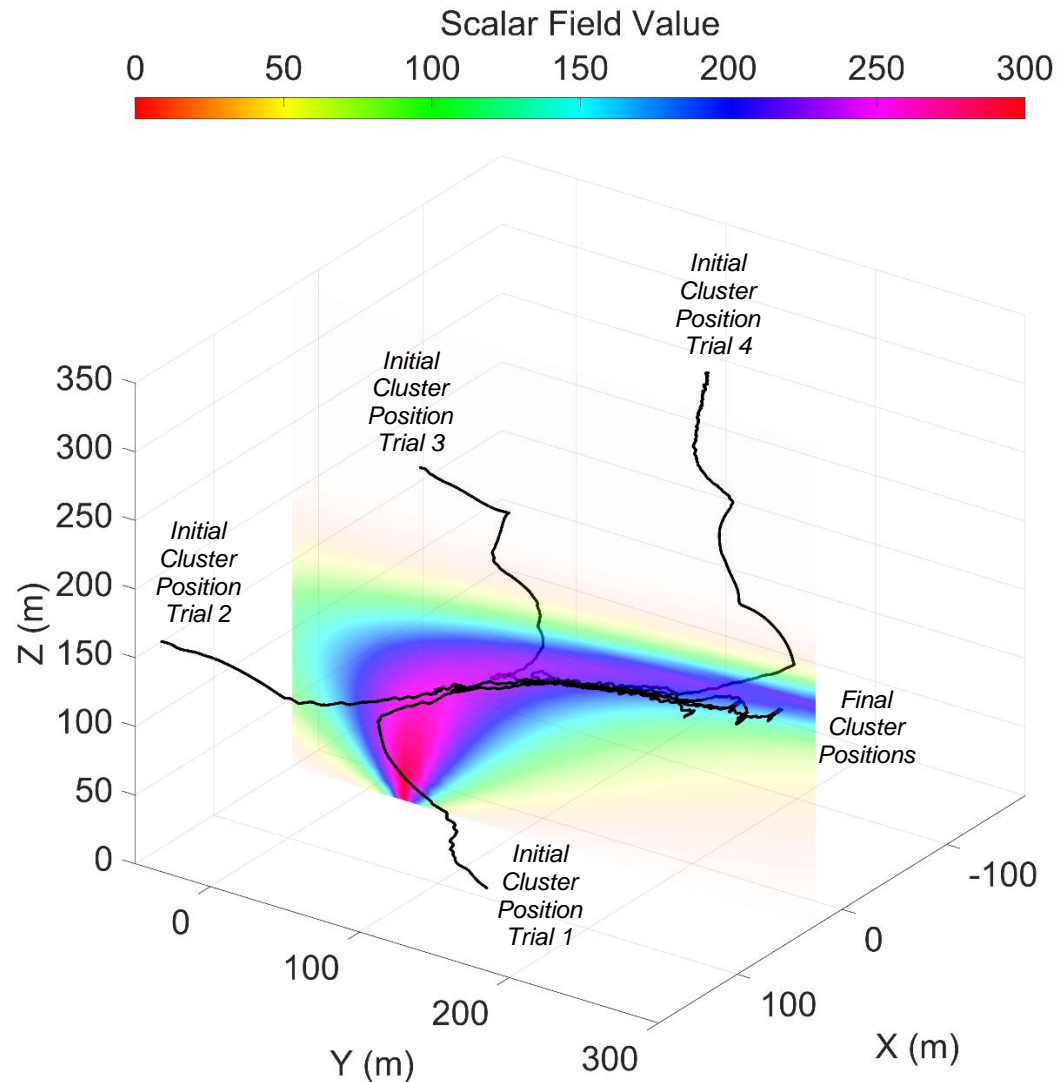
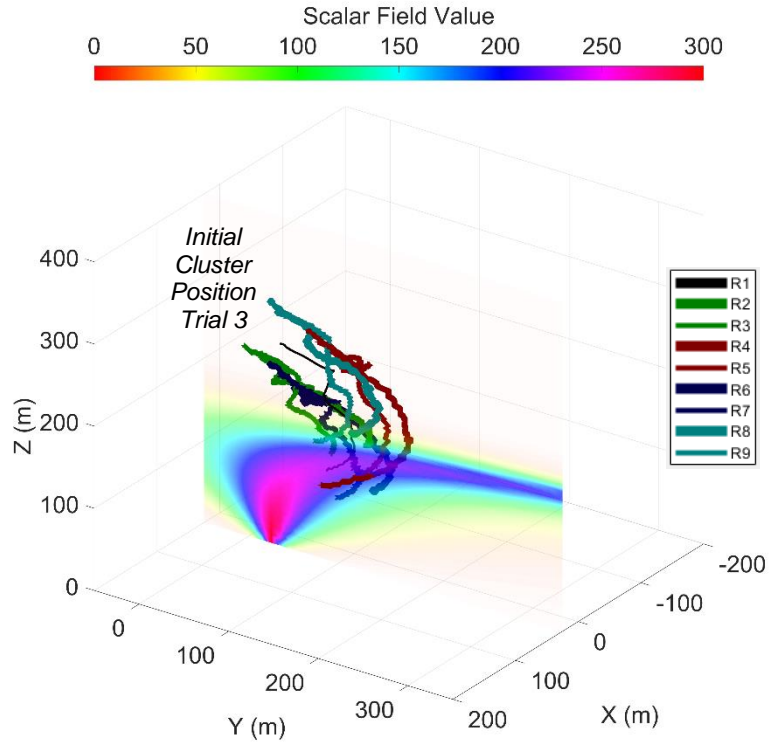
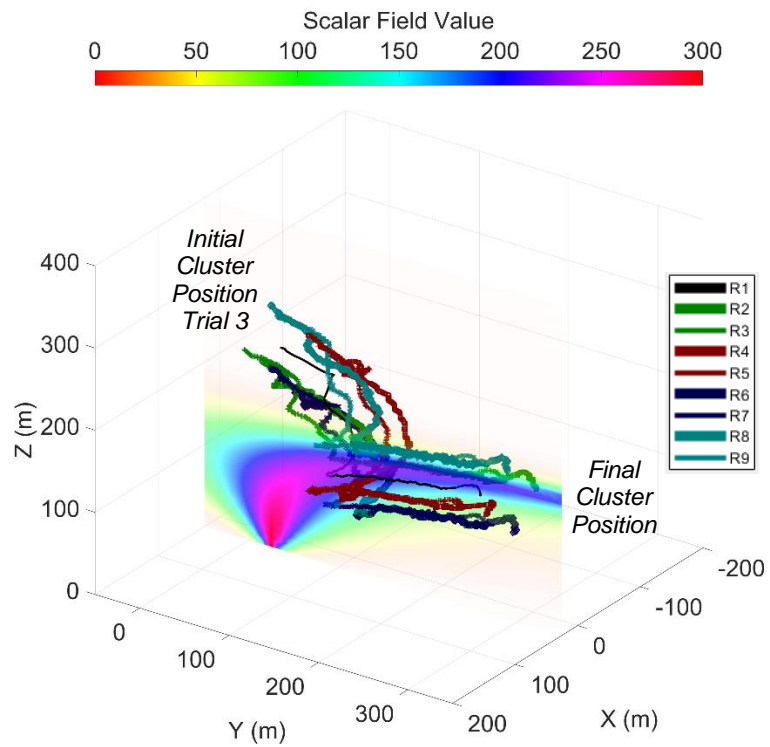


Figure 6-5: Scenario A multiple simulation trials showing flight paths of nine-robot clusters that locate and then follow a scalar plume outward. Only cluster origins are shown for clarity.

To provide a more detailed description of functionality, the motion of each individual UAV in the cluster of trial 3 is shown in multipart Fig. 6-6. In part (a), the cluster operates in the “Locate Plume” control state, moving to and straddling the plume, a process that takes the first 79 s. The controller then switches to the “Follow Plume Outward” state, shown in part (b), where the cluster successfully moves down the plume until $t = 175$ s, at which point the plume prominence falls below the prescribed threshold. Fig. 6-7 contains time histories of the measured scalar field values from the robots located in the cluster’s “back” plane which includes Robots 1, 2, 4, 6, and 8. These time histories show several indications that the cluster is properly



a) Flight path after 79 s



b) Flight path after 200 s

Figure 6-6: Scenario A, trial 3 flight path of a nine-robot cluster as it a) locates a plume, and then b) descends/follows the plume away from the source

aligned once the cluster locates and straddles the plume at $t = 99$ s. First, Robot 1's scalar reading is higher than all other robots in that plane. Second, the scalar values of the robot pairs 2-4 and 6-8 converge, indicating that the controller has successfully nulled the differential values in each of the cluster's dimensions. Fig. 6-8 contains time histories of the measured scalar field values from the robots located in the cluster's "front" plane which includes Robots 3, 5, 7, and 9 for lateral positioning and rotational differentials used to align the cluster with the plume. Furthermore, given that the controller is working to hold formation geometry as the cluster navigates, control performance is indicated by the RMS errors for the cluster size parameters L_{C_i} and $L_{i,i+1}$ for $i = 2,4,6,8$ which were 2.8 m, 3.1 m, 2.8 m, 3.4 m, 2.6 m, 2.9 m, 2.6 m, 3.5 m, respectively. These are in a range of 1.5-2.0 times the standard deviation of position error, indicating acceptable formation control.

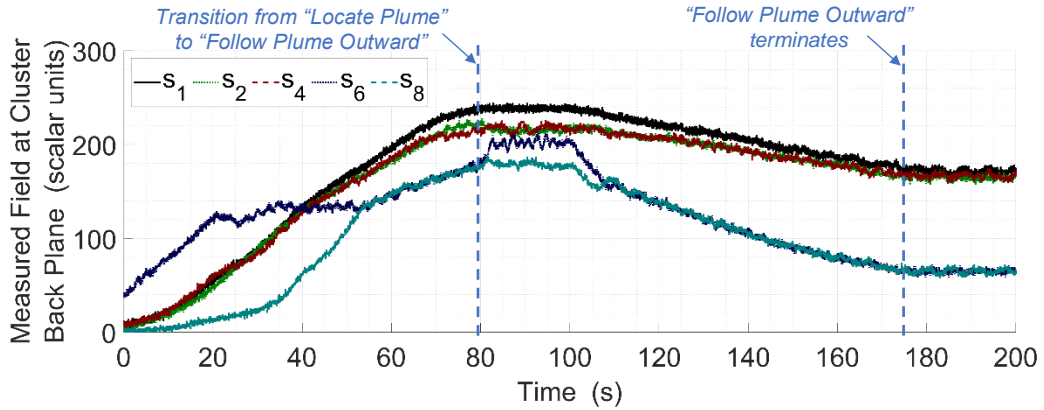


Figure 6-7: Scenario A, trial 3 time histories of measured scalar field values from robots in the cluster's "back" plane show that s_1 (Robot 1) remains higher, than the values measured by the surrounding robots, while following the plume outward and away from the source.

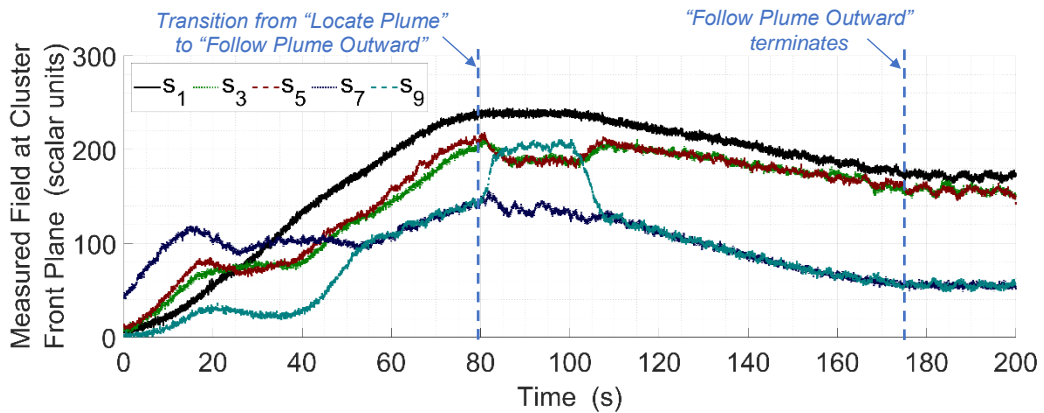


Figure 6-8: Scenario A, trial 3 time histories of measured scalar field values from robots in the cluster's "front" plane for lateral positioning and rotational differentials used to align the cluster with the plume.

Scenario B:

As in the previous case, downstream plume following simulation scenario B utilizes a simulated plume with a “tail” of several hundred meters, but with a different downwind direction and higher scalar field magnitude. A constant commanded cluster geometry held in the shape of a right-rectangular prism is used with formation variables shown in Table 6-5. State machine control parameters used in this simulation are summarized in Table 6-6. The multipart Fig. 6-9 shows the motion of each individual UAV in the cluster as it first operates in the “Locate Plume” state while moving toward and aligning itself with the plume, which completes at about $t = 88$ s. The cluster then switches to the “Follow Plume Outward” state until the plume prominence falls below the prescribed termination threshold at $t = 208$ s.

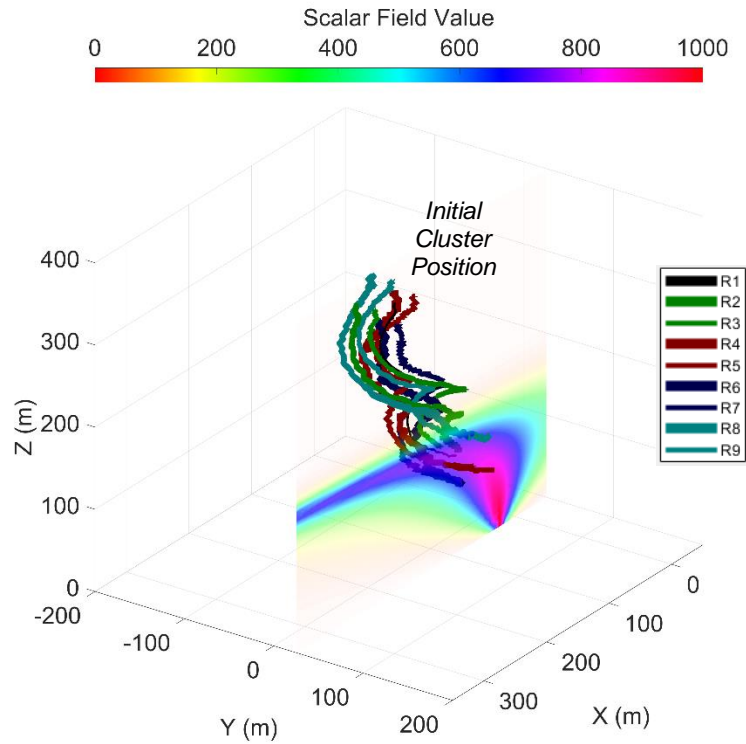
Time histories of the scalar values of the cluster “back” plane robots are shown in Fig. 6-10. The values are as expected while following the plume, with Robot 1 having the highest scalar value and the values converging for robot pairs 2-4 and 6-8. RMS errors for the cluster size parameters L_{C_i} and $L_{i,i+1}$ for $i = 2,4,6,8$ were 2.8 m, 3.0 m, 2.8 m, 3.1 m, 2.7 m, 2.8 m, 2.6 m, 2.8 m, respectively. Since the values are in a range of 1.5-2.0 times the standard deviation of position error, this indicates acceptable formation control. When following the plume, the values are as expected, with Robot 1 having the highest scalar value and with values converging for robot pairs 2-4 and 6-8. RMS errors for the cluster size parameters L_{C_i} and $L_{i,i+1}$ for $i = 2,4,6,8$ were 2.8 m, 3.0 m, 2.8 m, 3.1 m, 2.7 m, 2.8 m, 2.6 m, 2.8 m, respectively, are all in a range of 1.5-2.0 times the standard deviation of position error, indicating acceptable formation control.

TABLE 6-5: Nine-Robot Formation Shape Variables for Scenario B

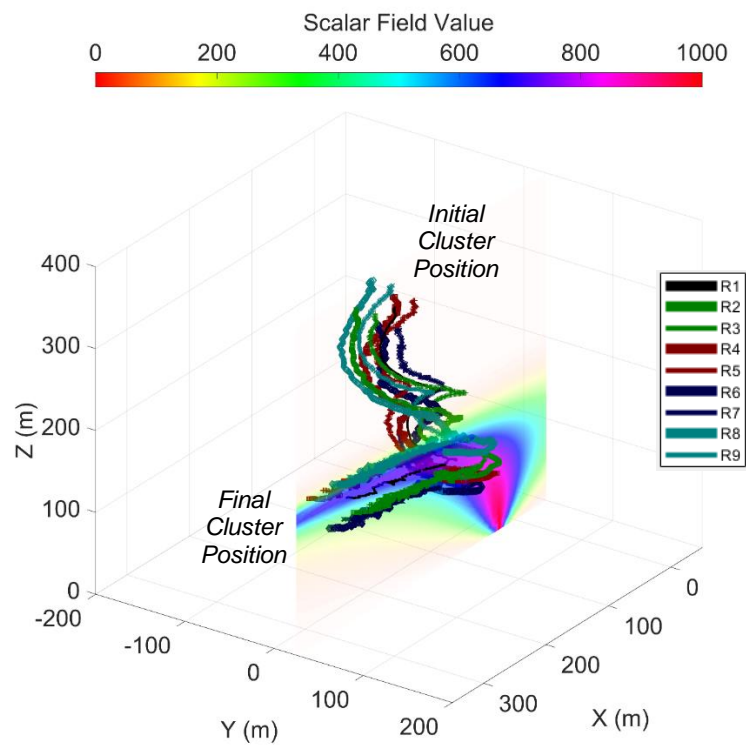
$L_{C2}, L_{23}, L_{C4}, L_{45}$	25 m
$L_{C6}, L_{67}, L_{C8}, L_{89}$	25 m
ζ	90°
$\alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_9$	0°
$\beta_3, \beta_4, \beta_5, \beta_6, \beta_7, \beta_9$	0°

TABLE 6-6: State Machine Control Parameters for Scenario B

Cluster Translational Speed	$S = 5 \text{ m/s}$
Cluster Rotational Speed	$R = 5 \text{ deg/s}$
Plume Prominence Threshold (Start)	$S_{prom \text{ threshold}} = 20 \text{ units}$
Differentials Threshold (plume location)	$\Delta_{threshold} = 20 \text{ units}$
Heading Alignment Threshold (prior to following plume outward)	$\Psi_{C \text{ threshold}} = 15^\circ$
Plume Prominence Threshold (End)	$S_{term \text{ threshold}} = 2 \text{ units}$



a) Flight path after 88 s



b) Flight path after 250 s

Figure 6-9: Scenario B flight path of a nine-robot cluster as it a) locates a plume, and then b) descends/follows the plume away from the source

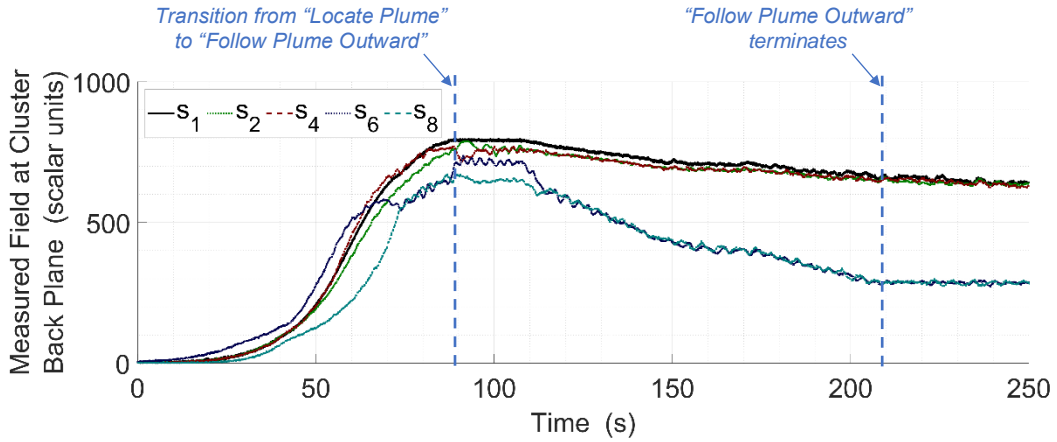


Figure 6-10: Scenario B time histories of measured scalar field values from robots in the cluster’s “back” plane show that s_1 (Robot 1) remains higher than the values measured by the surrounding robots, while following the plume outward and away from the source.

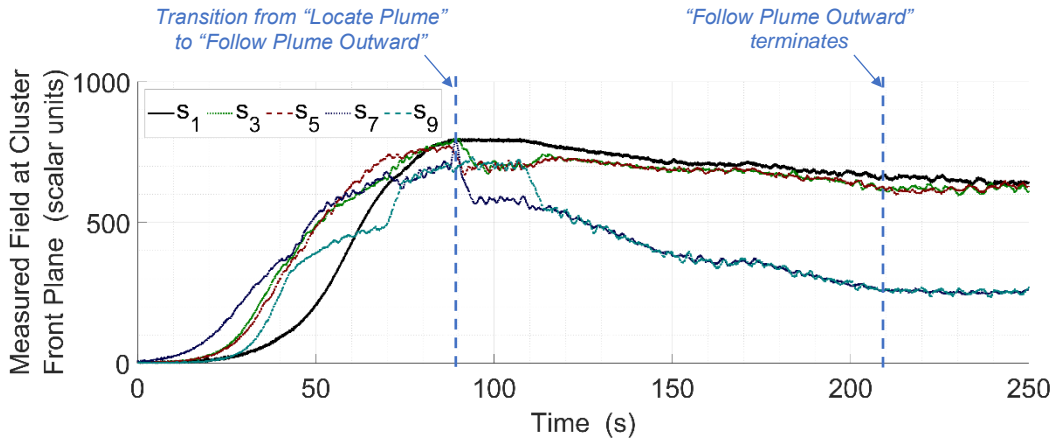


Figure 6-11: Scenario B time histories of measured scalar field values from robots in the cluster’s “front” plane for lateral positioning and rotational differentials used to align the cluster with the plume.

6.5 Plume Location and Descent with Cluster Resizing State Machine

Building upon the work presented in section 6.3, this section presents preliminary effort to demonstrate autonomous plume location and descent while resizing the cluster to maintain a desired sensed scalar value as it follows the plume outward and away from the source. This task simulates a UAV cluster following a pollutant from a source and outward to the impact area while adjusting the size of the cluster to maintain a particular subset of vehicles at a constant, desired sensed scalar field value. As in section 6.3, the state machine enables the cluster to reacquire the plume should it inadvertently maneuver “off” the feature.

Execution of this mission requires the specification of a set of parameters similar to section 6.3 with the addition of a user-specified, desired sensed scalar field value s_{target} that the “arms” of the cluster will track via expansion/contraction of the cluster size.

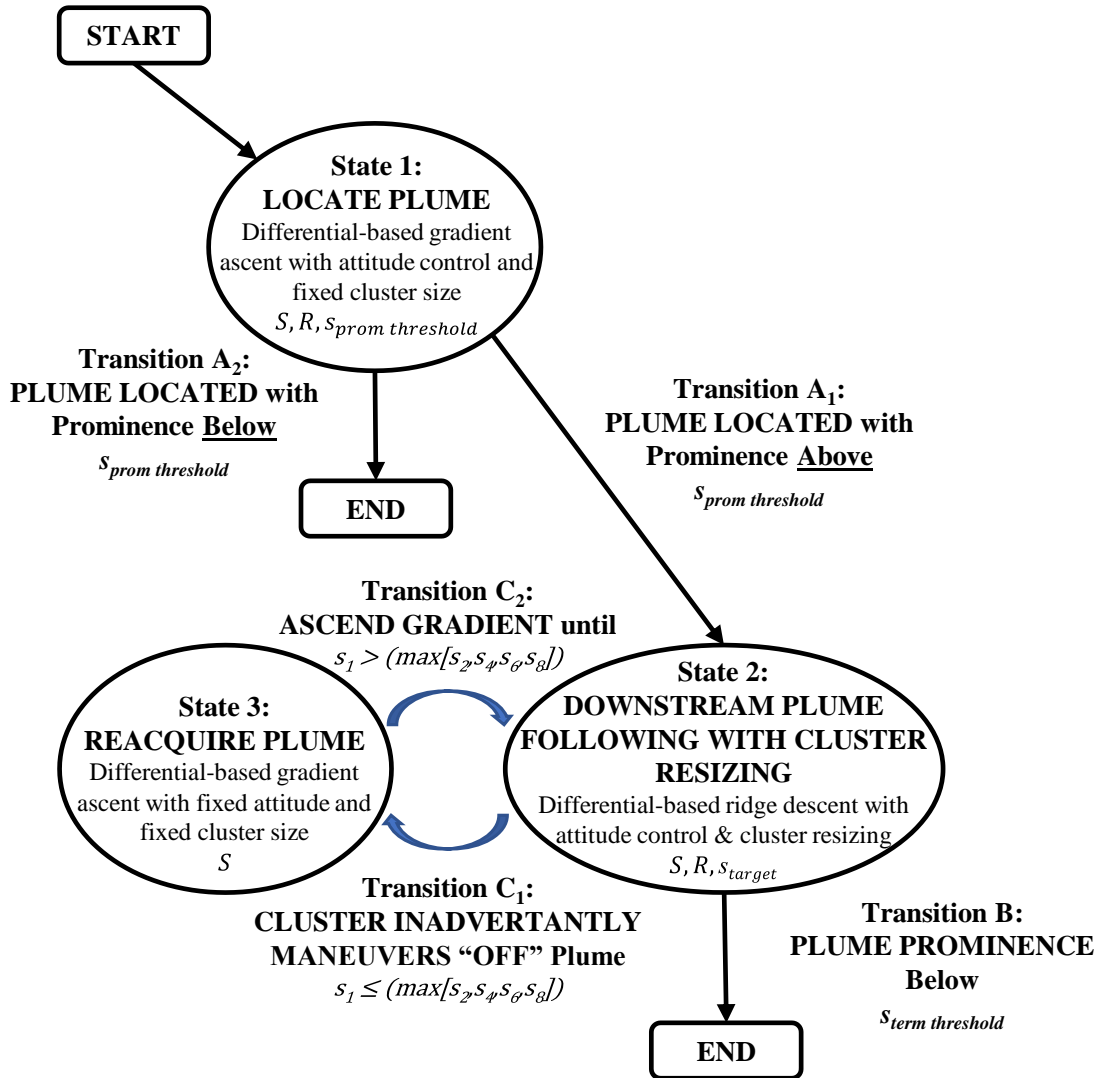


Figure 6-12: State machine for plume location and descent while resizing the cluster. Each of the three states consist of one or more AN-level controllers with specific control parameters

Control States: The state diagram shown in Fig. 6-12 illustrates the strategy, for plume location and descent while resizing the cluster, that has been implemented for the results reported in this work.

The primary state in this strategy is “State 2: Downstream Plume Following with Cluster Resizing”, in which the cluster is properly straddling the plume and implementing the plume following AN control primitive which issues corrective cluster translational and rotational control commands as shown in (37) to (40).

Note that the forward differential gain is K_x , lateral differential gains are K_y and K_z , pitch differential gain is K_{pitch} , and yaw differential gain is K_{yaw} .

$$\vec{D}_{tran} = \begin{bmatrix} K_x \cdot \text{sign}\{(s_2 - s_3) + (s_4 - s_5)\} \\ K_y \cdot \text{sign}\{(L_{C2} - L_{C4})\} \\ K_z \cdot \text{sign}\{(L_{C8} - L_{C6})\} \end{bmatrix} \quad (37)$$

$$\vec{D}_{rot} = \begin{bmatrix} 0 \\ K_{pitch} \cdot \text{sign}\{(s_7 - s_9) - (s_6 - s_8)\} \\ K_{yaw} \cdot \text{sign}\{(s_3 - s_5) - (s_2 - s_4)\} \end{bmatrix} \quad (38)$$

$$\dot{\vec{C}}_{9R}(1:3) = \begin{bmatrix} \dot{X}_{C des} \\ \dot{Y}_{C des} \\ \dot{Z}_{C des} \end{bmatrix} = S \cdot \frac{\vec{D}_{tran}}{\|\vec{D}_{tran}\|} \quad (39)$$

$$\dot{\vec{C}}_{9R}(4:6) = \begin{bmatrix} \dot{\phi}_{C des} \\ \dot{\theta}_{C des} \\ \dot{\psi}_{C des} \end{bmatrix} = R \cdot \frac{\vec{D}_{rot}}{\|\vec{D}_{rot}\|} \quad (40)$$

Cluster geometry resizing control commands for cluster lengths L_{C2} , L_{C4} , L_{C6} , L_{C8} are shown in (41) to (44), respectively, where s_{target} is the desired sensed scalar field value.

$$\dot{\vec{C}}_{9R}(7) = [\dot{L}_{C2}] = S \cdot \text{sign}\{(s_2 - s_{target})\} \quad (41)$$

$$\dot{\vec{C}}_{9R}(9) = [\dot{L}_{C4}] = S \cdot \text{sign}\{(s_4 - s_{target})\} \quad (42)$$

$$\dot{\vec{C}}_{9R}(11) = [\dot{L}_{C6}] = S \cdot \text{sign}\{(s_6 - s_{target})\} \quad (43)$$

$$\dot{\vec{C}}_{9R}(13) = [\dot{L}_{C8}] = S \cdot \text{sign}\{(s_8 - s_{target})\} \quad (44)$$

The first state, “State 1: Locate Plume”, is used to initially move to and straddle the plume. This state uses the plume following AN control primitive in section 6.2 with the exception that the X-component in (23) is negative, which enables gradient ascent in that dimension. This strategy moves the cluster towards a local maximum while aligning the cluster with a plume stemming from that source, if one exists. If the cluster successfully locates a plume that meets the prespecified minimum prominence threshold, the state machine proceeds to “State 2: Downstream Plume Following with Cluster Resizing”.

The third state, “State 3: Reacquire Plume”, is used if the cluster inadvertently maneuvers “off” the plume while descending it. This is when the cluster is still straddling and aligned with the plume, but not centered in it. The strategy adopted in this case is to reverse the longitudinal direction of motion, as done in

“State 1: Locate Plume”, by making the X-component in (23) negative and holding the cluster orientation and geometry constant. Cluster rotation is computed using a simple proportional controller per (28) to hold the current orientation of the cluster in lieu of (24).

State Transition Criteria: Formal definitions of the state machine transition criteria consist of the following. The criteria for “Transition A₁” that must be satisfied to move between “State 1: Locate Plume” and “State 2: Downstream Plume Following with Cluster Resizing” are based on the magnitude of the differentials being below a prescribed threshold per (29), the difference in the measured scalar field value from Robot 1 and the maximum from Robots 2, 4, 6, and 8 by a prescribed minimum threshold per (27), and the cluster heading alignment with the downstream direction of the plume to within a prescribed threshold per (30) to (33). Note that (27) defines the minimum plume “prominence” which functions as a “go” or “no go” condition (i.e., if the plume contains a feature of interest) as to whether the cluster should proceed with following the plume outward.

While in “State 2: Downstream Plume Following with Cluster Resizing”, in the event that the cluster inadvertently maneuvers “off” the feature, the state machine “Transition C₁” criteria to move to “State 3: Reacquire Plume” is per (34).

While in “State 3: Reacquire Plume”, state machine “Transition C₂” criteria to move back to “State 2: Downstream Plume Following with Cluster Resizing” is per (35). Following the plume outward continues until the cluster reaches a point where the plume prominence falls below a prescribed threshold with “Transition B” criteria per (36).

6.6 Downstream Plume Following with Cluster Resizing Mission Simulation

This section demonstrates execution of mission scenarios to locate and descend a plume using a nine-robot cluster while resizing the formation geometry to maintain a desired sensed scalar value using the state-based strategy described in section 6.5. As previously discussed, the objective is to have the cluster locate an unknown plume with a minimum prominence value, follow a pollutant from the source outward to the impact area while adjusting the size of the cluster to maintain a subset of vehicles (Robots 2, 4, 6, 8) at a constant desired scalar field value, reacquire the plume if it inadvertently maneuvers “off” the feature, and to hold its position and attitude when it reaches a point where the prominence is below a prescribed threshold.

Scenario C:

Downstream plume following with cluster resizing simulation scenario C utilizes a simulated plume with a “tail” of several hundred meters, initial constant commanded cluster geometry held in the shape of a right-rectangular prism with formation variables shown in Table 6-7. State machine control parameters used in this simulation are summarized in Table 6-8.

TABLE 6-7: Nine-Robot Formation Shape Variables for Scenario C

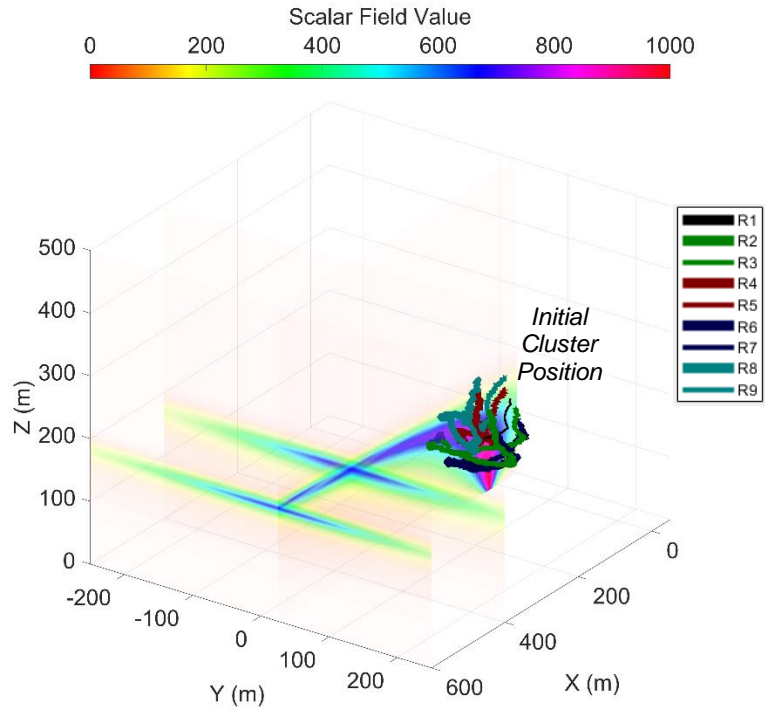
$L_{C2}, L_{23}, L_{C4}, L_{45}$	35 m
$L_{C6}, L_{67}, L_{C8}, L_{89}$	35 m
ζ	90°
$\alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_9$	0°
$\beta_3, \beta_4, \beta_5, \beta_6, \beta_7, \beta_9$	0°

TABLE 6-8: State Machine Control Parameters for Scenario C

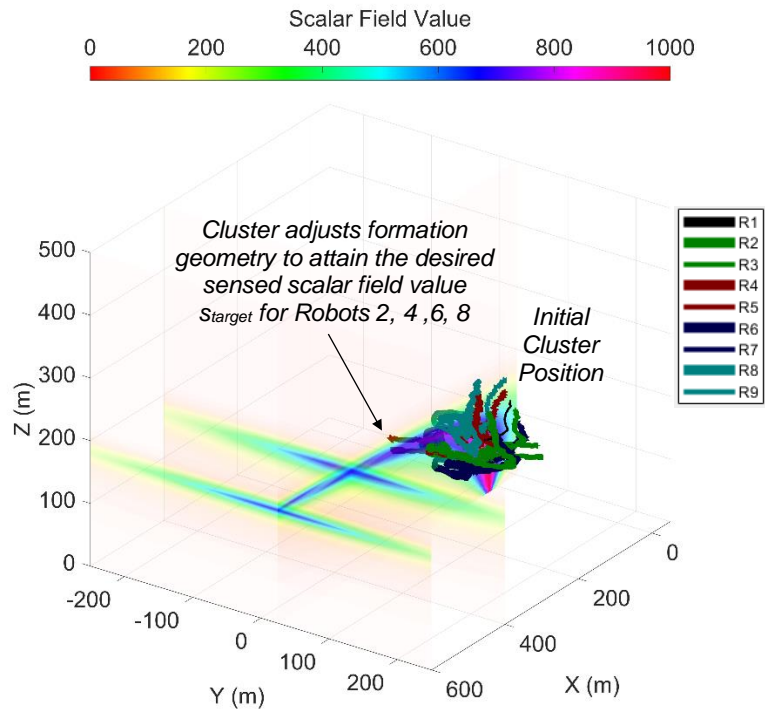
Cluster Translational Speed	$S = 4 \text{ m/s}$
Cluster Rotational Speed	$R = 5 \text{ deg/s}$
Plume Prominence Threshold (Start)	$s_{prom \text{ threshold}} = 20 \text{ units}$
Differentials Threshold (plume location)	$\Delta_{threshold} = 30 \text{ units}$
Heading Alignment Threshold (prior to following plume outward)	$\Psi_{C \text{ threshold}} = 15^\circ$
Plume Prominence Threshold (End)	$s_{term \text{ threshold}} = 25 \text{ units}$
Desired sensed scalar field value	$s_{target} = 500 \text{ units}$

Multipart Fig. 6-13 illustrates the motion path of each individual UAV in the cluster as it a) locates a plume, b) adjusts the geometry so that Robots 2, 4, 6, 8 are at the desired sensed scalar field value, c) follows the plume outward while modifying the formation geometry to maintain the desired sensed scalar field value, and d) holds the position, attitude, and formation geometry upon meeting the termination condition.

Figure 6-14 contains time histories of the measured scalar field values from the robots located in the cluster’s “back” plane which includes Robots 1, 2, 4, 6, and 8. These time histories indicate that the cluster is properly aligned once the cluster locates and straddles the plume at approximately $t = 97 \text{ s}$ since Robot 1’s scalar reading s_l is higher than all other robots in that plane. At $t = 137 \text{ s}$, the cluster completes the adjustment of the formation geometry so that the scalar values of Robots 2, 4, 6, 8 converge to the desired sensed scalar

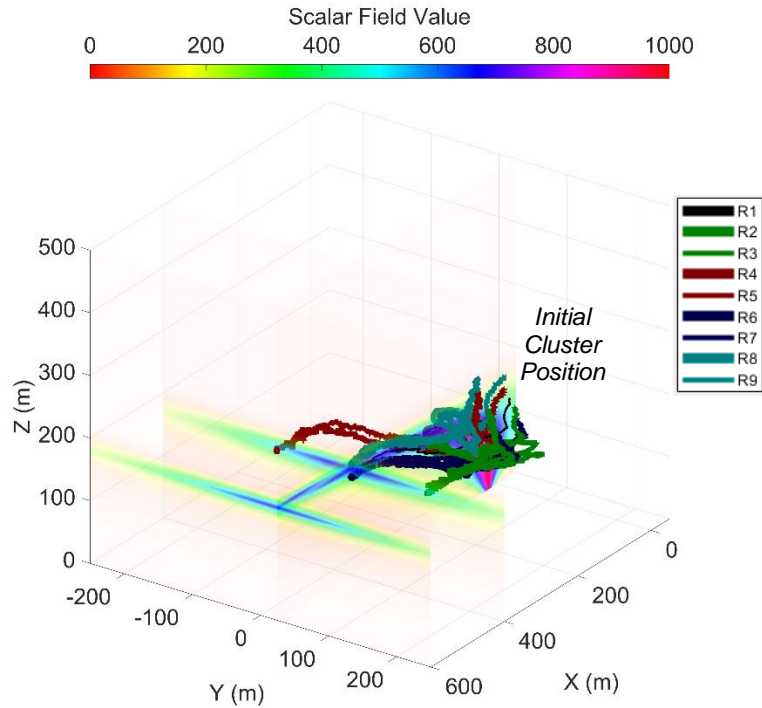


a) Flight path after 97 s as cluster locates plume

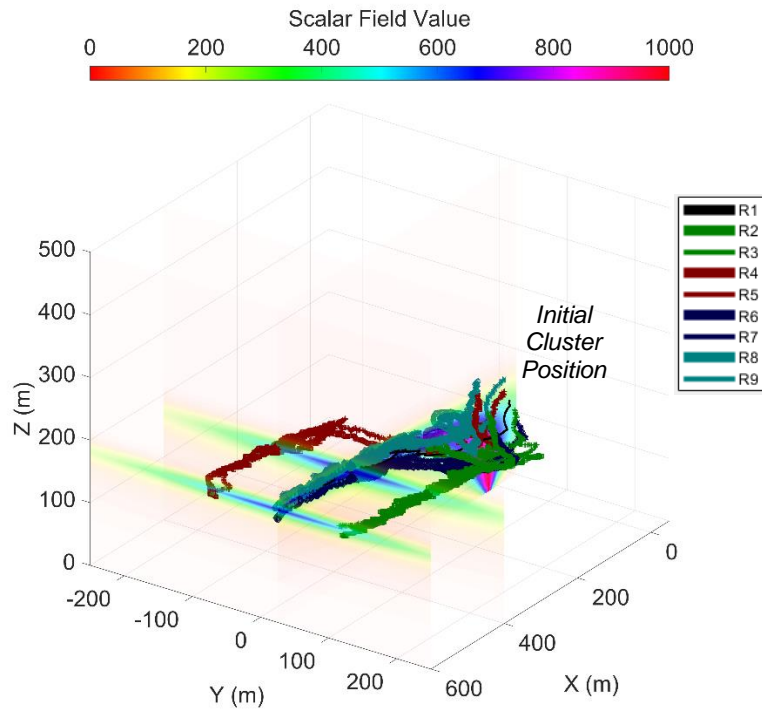


b) Flight path after 137 s as cluster adjusts geometry to attain desired sensed scalar value

Figure 6-13: Scenario C flight path of a nine-robot cluster as it a) locates plume, b) adjusts the geometry so that Robots 2, 4, 6, 8 are at the desired sensed scalar value, c) follows the plume outward while modifying the geometry to maintain the desired sensed scalar value, and d) holds position, attitude, and formation geometry upon meeting the termination condition.



c) Flight path after 230 s as cluster follows plume outward while adjusting geometry to maintain desired sensed scalar value



d) Flight path after 652 s when cluster stops upon meeting termination condition

Figure 6-13 (continued): Scenario C flight path of a nine-robot cluster as it a) locates plume, b) adjusts the geometry so that Robots 2, 4, 6, 8 are at the desired sensed scalar value, c) follows the plume outward while modifying the geometry to maintain the desired sensed scalar value, and d) holds position, attitude, and formation geometry upon meeting the termination condition.

field value of $s_{target} = 500$ units. Note that at approximately $t = 600$ s, the less pronounced plume prominence farther away from the source results in computed differentials which are less distinguishable from the scalar field sensor and position sensor noise. Finally, at $t = 652$ s, the state machine directs the cluster to hold its position, attitude, and formation geometry as the sensed plume prominence falls below the termination threshold $s_{term\ threshold} = 25$ units.

Furthermore, given that the controller is working to hold the remaining dimensions of the formation geometry constant as the cluster navigates, control performance is indicated by the RMS errors for the cluster size parameters $L_{i,i+1}$ for $i = 2,4,6,8$ which were 3.2 m, 2.9 m, 3.1 m, 3.1 m, respectively. As in the prior simulations, these fall in a range of 1.5-2.0 times the standard deviation of position error, indicating acceptable formation control.

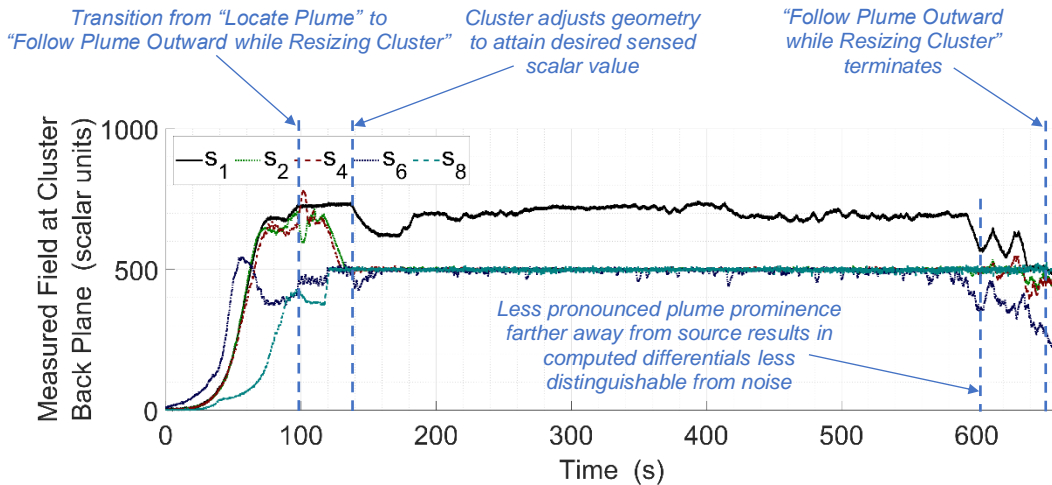


Figure 6-14: Scenario C time histories of measured scalar field values from robots in the cluster’s “back” plane show that s_1 (Robot 1) remains higher than the values measured by the surrounding robots and the sensed scalar field values for the other vehicles converge to s_{target} as the cluster adjusts the formation geometry while following the plume outward and away from the source.

Scenario D:

The goal of scenario D is to compare performance of downstream plume following using a fixed formation (i.e., no cluster resizing) against a cluster formation that can resize. As in the previous simulations, the scalar field was simulated using a plume with a “tail” of several hundred meters.

The first part of scenario D demonstrates downstream plume following using a fixed formation (i.e., no cluster resizing). The commanded cluster geometry is held constant in the shape of a right-rectangular prism with formation variables shown in Table 6-9. State machine control parameters for this simulation are

summarized in Table 6-10. Fig. 6-15 shows the motion of each individual UAV in the cluster as it first operates in the “Locate Plume” state while moving toward and aligning itself with the plume, which completes at about $t = 151$ s. The cluster then switches to the “Follow Plume Outward” state until the plume prominence falls below the prescribed termination threshold at $t = 170$ s.

TABLE 6-9: Nine-Robot Formation Shape Variables for Scenario D with Fixed Formation

$L_{C2}, L_{23}, L_{C4}, L_{45}$	35 m
$L_{C6}, L_{67}, L_{C8}, L_{89}$	35 m
ζ	90°
$\alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_9$	0°
$\beta_3, \beta_4, \beta_5, \beta_6, \beta_7, \beta_9$	0°

TABLE 6-10: State Machine Control Parameters for Scenario D with Fixed Formation

Cluster Translational Speed	$S = 4 \text{ m/s}$
Cluster Rotational Speed	$R = 5 \text{ deg/s}$
Plume Prominence Threshold (Start)	$S_{prom \text{ threshold}} = 15 \text{ units}$
Differentials Threshold (plume location)	$\Delta_{threshold} = 20 \text{ units}$
Heading Alignment Threshold (prior to following plume outward)	$\Psi_{C \text{ threshold}} = 15^\circ$
Plume Prominence Threshold (End)	$S_{term \text{ threshold}} = 3 \text{ units}$

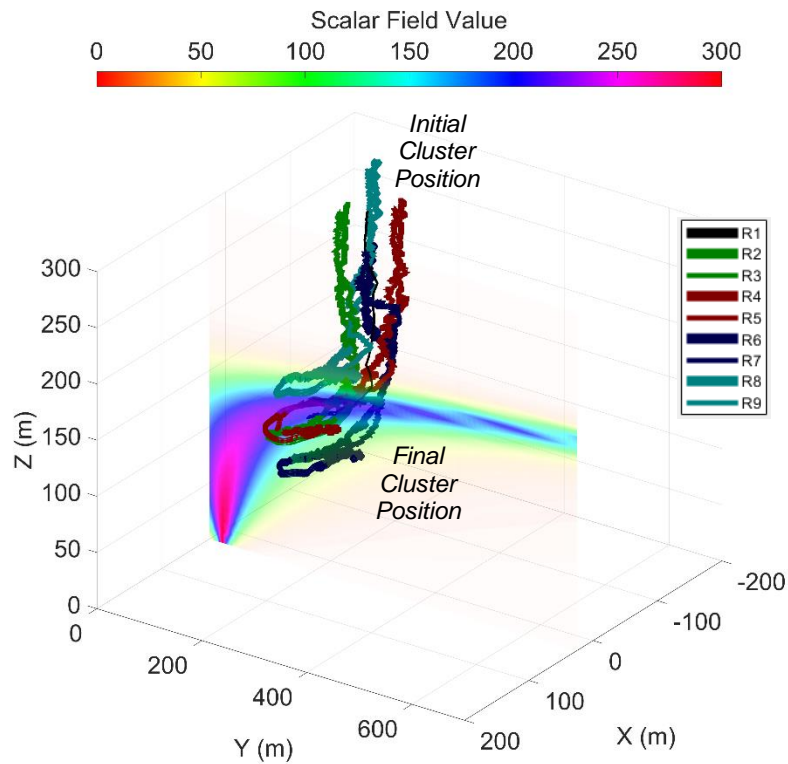


Figure 6-15: Scenario D with fixed formation flight path of a nine-robot cluster as it a) locates a plume and then b) descends/follows the plume away from the source.

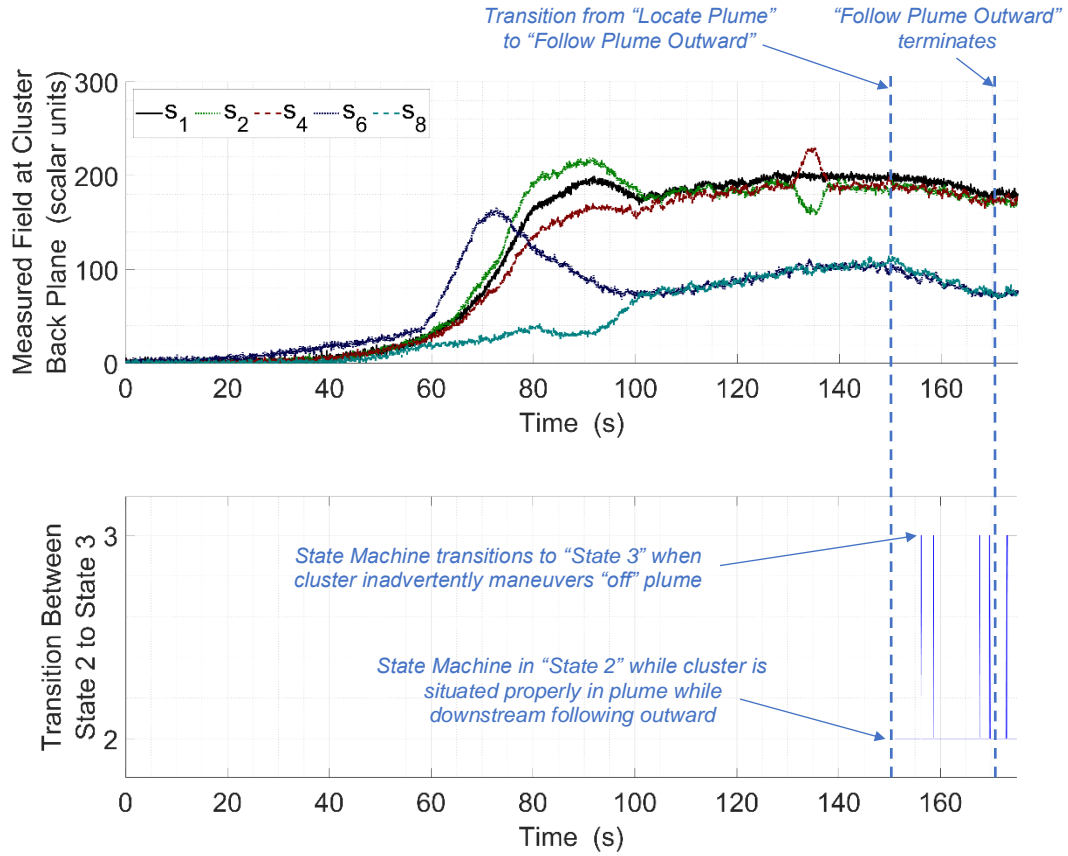


Figure 6-16: Scenario D with fixed formation time histories of measured scalar field values (upper plot) from robots in the cluster’s “back” plane show that s_1 (Robot 1) remains higher than the values measured by the surrounding robots, while following the plume outward and away from the source. State machine transitions between states 2 and 3 (lower plot) show where the cluster inadvertently maneuvers “off” the plume and performs corrective motion.

The second part of scenario D demonstrates downstream plume following with cluster resizing and its ability to follow a plume farther out and away from the source compared with using a fixed formation. The initial cluster geometry is in the shape of a right-rectangular prism with formation variables shown in Table 6-11. State machine control parameters for this simulation are summarized in Table 6-12. Note that the formation shape variables and state machine control parameters are set to the same values as the fixed formation case, where applicable. Fig. 6-17 illustrates the motion path of each individual UAV in the cluster as it a) locates a plume, b) adjusts the geometry so that Robots 2, 4, 6, 8 are at the desired sensed scalar field value, c) follows the plume outward while modifying the formation geometry to maintain the desired sensed scalar field value, and d) holds the position, attitude, and formation geometry upon meeting the termination condition.

TABLE 6-11: Nine-Robot Formation Shape Variables for Scenario D with Resizable Cluster

$L_{C2}, L_{23}, L_{C4}, L_{45}$	35 m
$L_{C6}, L_{67}, L_{C8}, L_{89}$	35 m
ζ	90°
$\alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_9$	0°
$\beta_3, \beta_4, \beta_5, \beta_6, \beta_7, \beta_9$	0°

TABLE 6-12: State Machine Control Parameters for Scenario D with Resizable Cluster

Cluster Translational Speed	$S = 4 \text{ m/s}$
Cluster Rotational Speed	$R = 5 \text{ deg/s}$
Plume Prominence Threshold (Start)	$S_{prom \text{ threshold}} = 15 \text{ units}$
Differentials Threshold (plume location)	$\Delta_{threshold} = 20 \text{ units}$
Heading Alignment Threshold (prior to following plume outward)	$\Psi_{C \text{ threshold}} = 15^\circ$
Plume Prominence Threshold (End)	$S_{term \text{ threshold}} = 3 \text{ units}$
Desired sensed scalar field value	$S_{target} = 155 \text{ units}$

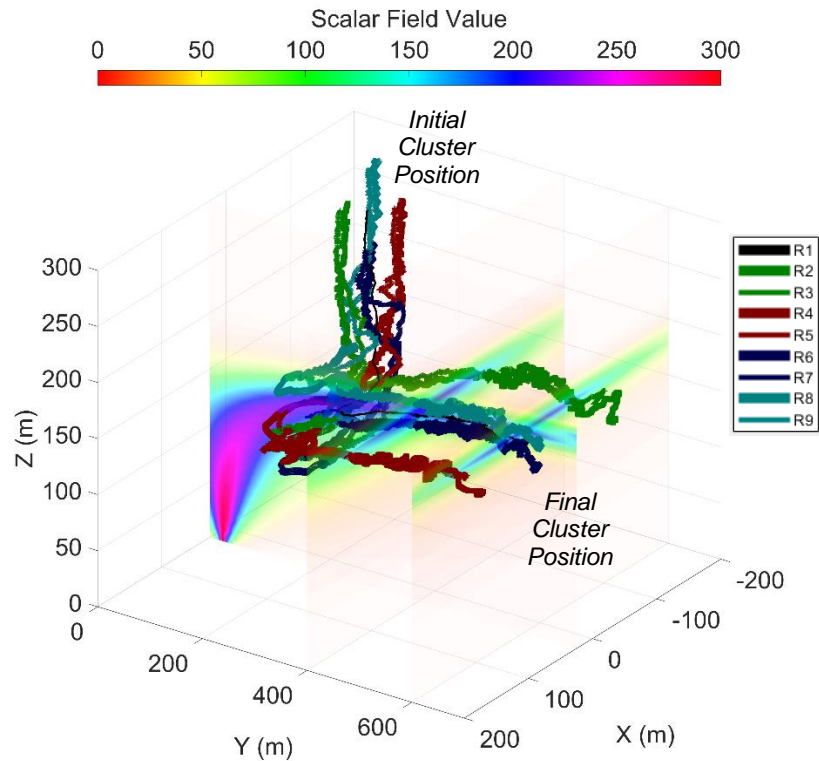


Figure 6-17: Scenario D with resizable formation flight path of a nine-robot cluster as it a) locates plume, b) adjusts the geometry so that Robots 2, 4, 6, 8 are at the desired sensed scalar value, c) follows the plume outward while modifying the geometry to maintain the desired sensed scalar value, and d) holds position, attitude, and formation geometry upon meeting the termination condition.

Figure 6-18 contains time histories of the measured scalar field values from the robots located in the cluster's "back" plane which includes Robots 1, 2, 4, 6, and 8. These time histories indicate that the cluster is properly aligned once the cluster locates and straddles the plume at approximately $t = 151$ s since Robot 1's scalar reading s_1 is higher than all other robots in that plane. At $t = 170$ s, the cluster completes the adjustment of the formation geometry so that the scalar values of Robots 2, 4, 6, 8 converge to the desired sensed scalar field value of $s_{target} = 155$ units. Finally, at $t = 435$ s, the state machine directs the cluster to hold its position, attitude, and formation geometry as the sensed plume prominence falls below the termination threshold $s_{term\ threshold} = 3$ units.

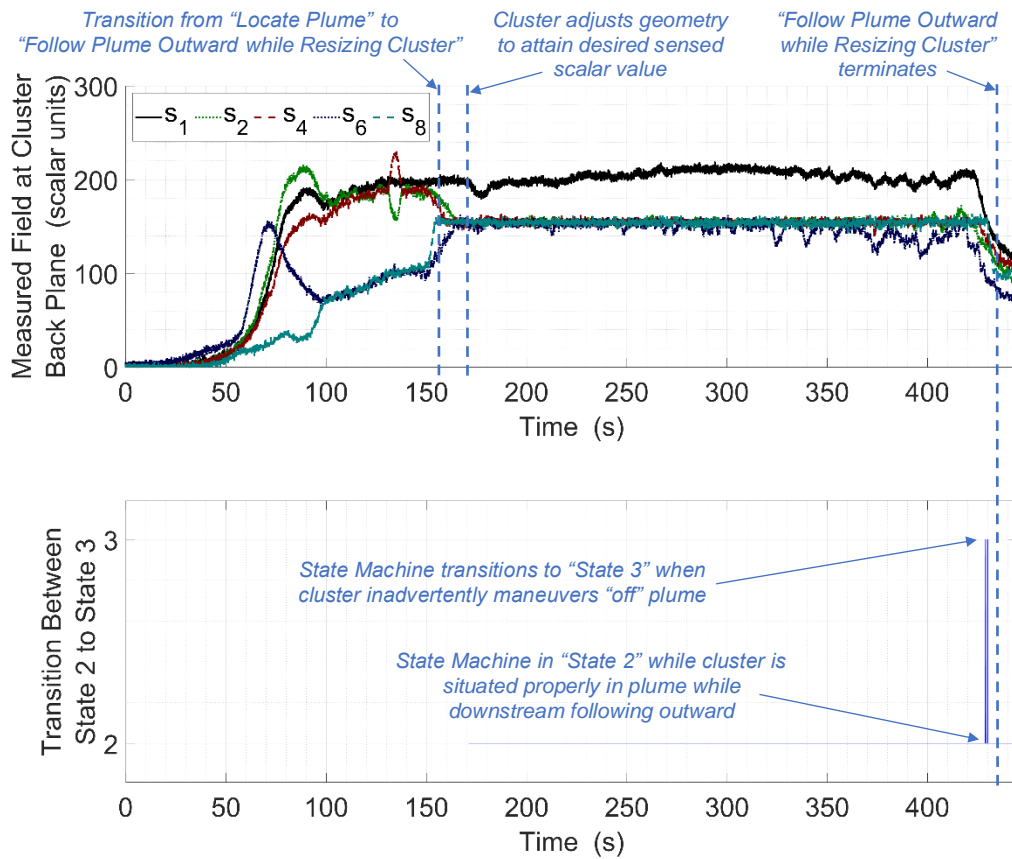


Figure 6-18: Scenario D with resizable formation time histories of measured scalar field values (upper plot) from robots in the cluster's "back" plane show that s_1 (Robot 1) remains higher than the values measured by the surrounding robots. The sensed scalar field values for the other vehicles converge to s_{target} as the cluster adjusts the formation geometry while following the plume outward and away from the source. State machine transitions between states 2 and 3 (lower plot) show where the cluster inadvertently maneuvers "off" plume and performs corrective motion.

This preliminary study appears to indicate that plume following outward with formation resizing allows the cluster to follow a plume downstream farther out than with a fixed formation. The results show that allowing the formation to change its size (so that the cluster “back” plane Robots 2, 4, 6, 8, move outwards to a sensed scalar field value lower than that of Robot 1) enables the cluster to be able to better distinguish the plume prominence while farther away from the source. This observation is also supported by the smaller number of state machine transitions between states 2 and 3 in the resizable formation case.

Furthermore, given that the controller is working to hold the remaining dimensions of the formation geometry constant as the cluster navigates, control performance is indicated by the RMS errors for the cluster size parameters $L_{i,i+1}$ for $i = 2,4,6,8$ which were 3.0 m, 2.9 m, 3.0 m, 3.1 m, respectively. As in the prior simulations, these fall in a range of 1.5-2.0 times the standard deviation of position error, indicating acceptable formation control.

7.0 Summary, Discussion, and Future Work

Summary:

This dissertation presents a multilayer control architecture for implementing multirobot AN capabilities to explore 3D scalar fields using UAVs. Verification of these capabilities is provided through high-fidelity simulations that incorporate real vehicle flight dynamics, wind gust disturbances, vehicle position sensor inaccuracy, and scalar field sensor noise. Specific scenarios demonstrating these capabilities include:

- seeking/tracking a local source, both stationary as well as moving within a time-varying field;
- acquisition of a desired isosurface within a field and then performing structured mapping of single and overlapping signals, with an option for the orientation and spacing of planar slices;
- acquisition and following of a plume outward (i.e., away from a source).

Building on SCU's prior work in multirobot AN, the contributions presented in this dissertation significantly extend the current state of the art in terms of AN functionality for exploring 3D scalar fields.

Distinct aspects of the presented work include:

- the use of a unified control architecture to demonstrate a wide range of AN missions and control primitives for clusters of varying sizes and shapes;
- the kinematic definition of new four- and nine-robot clusters appropriate for the AN tasks of interest;
- the extension of SCU's existing 2D multirobot AN control primitives to 3D fields in order to implement local extrema finding, isosurface navigation, and downstream plume following;
- the use of state-based sequencing of AN control primitives and parameters to implement mission-level capabilities such as characterizing the structure of isosurfaces and acquiring and following plumes;
- the verification of the proposed AN techniques via high-fidelity simulations that include vehicle-level dynamics, wind gust disturbances, vehicle position sensor inaccuracy, and scalar field sensor noise.

Discussion:

Several implementation issues pertaining to the use of cluster-space and AN control methodologies have been explored in detail in other publications, but are worthy of mention here.

Within the cluster-space formation control methodology, there is significant flexibility in how a cluster may be defined. These choices influence the degree of interdependency among the cluster pose state variables, which in turn, affects issues such as the amount of computation in the servo loop, the existence of singularities, and the level of (de)centralization per Kitts et al. [70]. For singularities, as an example, several approaches have been successfully demonstrated, such as dynamically switching to new cluster pose definitions and the use of dual quaternions per Mas et al. [73]. Regarding (de)centralization, while the examples used for this study were implemented in a centralized manner, decentralized cluster definitions at varying levels are possible, to include the ability to implement swarm-like capabilities. However, the technique includes managerial overhead (like the explicit enumeration of vehicles) that is unnecessary if there is no desire to take advantage of cluster-space control benefits (such as explicit formation control, well-behaved motion in the cluster-space, etc.). In separate research effort, SCU is exploring formalized swarm approaches that support the comprehensive AN capabilities being explored in this work.

With respect to AN, a significant issue is selecting the size of the cluster given that it drives AN performance by trading aperture size with the spatial resolution of the cluster. Given the field characterization techniques (local gradient, differential over a baseline, etc.) used by the AN control primitives, spatial filtering occurs per Kitts et al. [1]. For example, the gradient approach used in isosurface mapping assumes that the size of the cluster is “small” compared to the spatial variation of interest for the field (e.g., perhaps at most $\frac{1}{4}$ of the spatial wavelength of the scalar features of interest). Scalar features with smaller wavelengths are spatially smoothed, thus decreasing the responsiveness of AN motion. Because the cluster size is a specifiable mission attribute, the operator may select a size suitable for the application at hand, typically trading spatial resolution, gradient/differential signal amplification, noise suppression, and the breadth of field exploration for a given number of robots. As a practical example, previous work in using a cluster of automated boats to find large scale bathymetric scalar features used cluster sizes in the range of 10-20 m per Kitts et al. [72]. This allowed for navigation with respect to geologic features on the order of 50+ meters, while effectively filtering out “noise” created by small rocks and boulders.

With regard to edge cases, there can be scenarios involving “flat” or uniform scalar fields where there is not enough variation in the field for the AN control primitives to react in a meaningful way. In McDonald et al. [46], SCU used an additional state at the mission layer that reacts to this condition by using a more conventional navigation strategy such as “mowing the lawn,” until a significant signal is observed.

Future Work:

Near term future work is experimental verification of the items summarized above that have been demonstrated in simulation. A single, stationary radio frequency transmitter can be mounted to a slowly driven ground vehicle to experimentally demonstrate 3D local source seeking and tracking. The same stationary RF transmitter can be used to generate a scalar field to demonstrate isosurface mapping while two such transmitters can be used to form a more complex field. Finally, placing a directional antenna on the RF transmitter can effectively generate a scalar field to allow for downstream plume following.

This work lays the foundation for the design and verification of the control architecture that SCU is developing for an initial cluster of four octocopter vehicles. The system uses a commercial Pixhawk 4 autopilot on each UAV for executing vehicle-level translation and heading commands while maintaining platform stability for pitch and roll. The MAVLINK protocol is used to wirelessly exchange command and telemetry data with a central control station. This allows pilots to gracefully constitute and decompose UAV cluster formations by selecting the control of each UAV to be provided by joystick input, position hold executed by the on-board autopilot, or automated control as computed by a Matlab/Simulink-based controller running on the control computer. This automated controller will implement the cluster, AN, and state machine control layers as described in this paper. Xbee receivers on each vehicle measure the signal strength of periodic transmissions at their location and relay these measurements back to the automated flight controller. The development of this multi-UAV flight system shown in Fig. 7-1 has been accomplished largely through a senior capstone project by Cameron et al. [74].

In addition, since 3D scalar fields can also exist in underwater environments, SCU has the capability to perform experimental verification of the items proposed in this dissertation using the multiple submersible vehicles testbed documented in Kitts et al. [75].



Figure 7-1: Drone cluster utilized by Santa Clara University to demonstrate Adaptive Navigation.
[Photo Courtesy of R. Rasay]

Beyond experimental verification, SCU's ongoing and future work in multirobot AN has several objectives. One objective is to formally characterize how cluster shape and size affects navigation performance given desired mission objectives and the nature of a scalar field. This knowledge will enable autonomous exploration of unknown scalar fields with adaptive formation reshaping and resizing to allow for the tuning of cluster responsiveness to features of various sizes. Other objectives include exploring benefits and costs of cluster-space control with larger numbers of vehicles, implementation of more sophisticated control laws, and identifying and then addressing additional edge cases beyond "flat" or uniform scalar fields.

Additional research with regard to isosurface mapping can be performed where the navigation reference vector ${}^G\hat{n}$ is set to the gradient vector \vec{g}_{grad} after reaching the isosurface. This strategy would enable the cluster to navigate to the isosurface and then perform isosurface mapping from that point to the opposite end versus heading in a predefined direction (as in this work) or being limited to travel between two arbitrary altitudes as in Matveev et al. [54]. Downstream plume following and isosurface mapping can also be combined by "rolling" the formation while following a plume outward. This can be accomplished either by continuously rotating the cluster about its roll axis \hat{X}_C at a slow angular rate or pausing forward motion at regular intervals and performing 90° rotations.

Motivated by the needs of clients with specific AN missions in mind, SCU is pursuing broader extensions of multirobot AN and its cluster-space control paradigm. This work includes development of decentralized swarm controllers (currently focused on 2D fields), exploration of vector fields, extending techniques for discrete sampling strategies, and control refinement to address turbulent and discontinuous fields. The ultimate goal is to demonstrate a wide range of robust and practical AN capabilities through real missions in the field.

REFERENCES

- [1] C.A. Kitts, R.T. McDonald, and M.A. Neumann, "Adaptive Navigation Control Primitives for Multirobot Clusters: Extrema Finding, Contour Following, Ridge/Trench Following, and Saddle Point Station Keeping," *IEEE Access*, Vol 6, pp.17625-17642, 2018.
- [2] C. Zhang, D. Arnold, N. Ghods, A. Siranosian, and M. Krstic, "Source Seeking with Non-Holonomic Unicycle without Position Measurement and with Tuning of Forward Velocity," in *Systems & Control Letters*, Vol.56, No.3, pp.245-252, 2007.
- [3] C.G. Mayhew, R.G. Sanfelice, and A.R. Teel, "Robust Source-Seeking Hybrid Controllers for Nonholonomic Vehicles," in *Proceedings of Annual American Control Conference*, pp.2722-2727, Jun. 2008.
- [4] A.S. Matveev, H. Teimoori and A.V. Savkin, "Navigation of a Non-Holonomic Vehicle for Gradient Climbing and Source Seeking without Gradient Estimation," in *Proceedings of Annual American Control Conference*, pp.219-223, Jun. 2010.
- [5] A.S. Matveev, H. Teimoori and A.V. Savkin, "Navigation of a Unicycle-Like Mobile Robot for Environmental Extremum Seeking," *Automatica*, Vol.47, pp.85-91, 2011.
- [6] S. Azuma, M.S. Sakar, and G.J. Pappas, "Stochastic Source Seeking by Mobile Robots," *IEEE Transactions on Automatic Control*, Vol.57, No.9, pp.2308-2321, Sep. 2012.
- [7] A. Kashyap and D. Ghose, "Pursuing a Time Varying and Moving Source Signal Using a Sensor Equipped UAV," *2017 International Conference on Unmanned Aircraft Systems*, pp.506-515, Jun. 2017.
- [8] A. Kamthe and S. Ghosh, "Gradient-based Augmentation to Maxima Turn Switching Strategy for Source-Seeking using Sensor-Equipped UAVs," *2020 International Conference on Unmanned Aircraft Systems*, pp.1525-1532, Sep. 2020.
- [9] C.G. Mayhew, R.G. Sanfelice, and A.R. Teel, "Robust Hybrid Source-Seeking Algorithms Based on Directional Derivatives and Their Approximations," in *Proceedings of IEEE Conference on Decision and Control*, pp.1735-1740, Dec. 2008.
- [10] S. Shigaki, Y. Shiota, D. Kurabayashi, and R. Kanzaki, "Modeling of the Adaptive Chemical Plume Tracing Algorithm of an Insect Using Fuzzy Inference," *IEEE Transactions on Fuzzy Systems*, Vol.28, No.1, pp.72-84, Jan. 2020.
- [11] P.P. Neumann, V.H. Bennetts, A.J. Lilienthal, M. Bartholmai, and J.H. Schiller, "Gas Source Localization with a Micro-Drone Using Bio-inspired and Particle Filter-Based Algorithms," *Advanced Robotics*, Vol.27, No.9, pp.725-738, Apr. 2013.
- [12] P. Ogren, E. Fiorelli, and N.E. Leonard, "Cooperative Control of Mobile Sensor Networks: Adaptive Gradient Climbing in a Distributed Environment," *IEEE Transactions on Automatic Control*, Vol.49, No.8, pp.1292-1302, 2004.
- [13] S. Zhuo, "Source Seeking of Multi-UAV Based on Extremum Seeking Algorithm," in *Proceedings of 17th International Conference on Control, Automation and Systems (ICCAS)*, pp.1062-1067, Oct. 2017.
- [14] S. Eu and K.M. Yap, "Chemical Plume Tracing: A Three-Dimensional Technique for Quadrotors by Considering the Altitude Control of the Robot in the Casting Stage," *International Journal of Advanced Robotic Systems*, Vol.15, No.1, 2018.

- [15] E. Biyik and M. Arcak, "Gradient Climbing in Formation via Extremum Seeking and Passivity-Based Coordination Rules," in *Proceedings of IEEE Conference on Decision and Control*, pp.3133-3138, Dec. 2007.
- [16] S. Zhu, D. Wang, and C.B. Low, "Cooperative Control of Multiple UAVs for Moving Source Seeking," in *Proceedings of International Conference on Unmanned Aircraft Systems*, pp.193-202, May 2013.
- [17] B.J. Moore and C. Canudas-de-Wit, "Source Seeking via Collaborative Measurements by a Circular Formation of Agents," in *Proceedings of Annual American Control Conference*, pp.6417-6422, Jun. 2010.
- [18] A. Renzaglia and L. Briñón-Arranz. "Search and Localization of a Weak Source with a Multi-Robot Formation," *Journal of Intelligent & Robotic Systems*, Vol.97, pp.623-634, Apr. 2019.
- [19] V. Gazi, B. Fidan, L. Marques, and R. Ordonez, "Robot Swarms: Dynamics and Control," *Mobile Robots for Dynamic Environments*, E. F. Kececi and M. Ceccarelli, Eds. New York, NY, USA: ASME, Ch.4, pp.79-107, 2015.
- [20] A. Turgeman and H. Werner, "Multiple Source Seeking using Glowworm Swarm Optimization and Distributed Gradient Estimation," in *Proceedings of American Control Conference*, pp.3558-3563, Jun. 2018.
- [21] Z. Li, K. You, and S. Song, "Cooperative Source Seeking via Networked Multi-vehicle Systems," *Automatica*, May, 2020.
- [22] N. A. Atanasov, J.L. Ny, and G. J. Pappas, "Distributed Algorithms for Stochastic Source Seeking with Mobile Robot Networks," *Journal on Dynamic Systems, Measurement & Control*, Vol.173, No.3, pp.1-9, 2015.
- [23] R.G. Li and H.N. Wu, "Multi-Robot Source Location of Scalar Fields by a Novel Swarm Search Mechanism With Collision/Obstacle Avoidance," *IEEE Transactions on Intelligent Transportation Systems*, pp.1-16, Jul. 2020.
- [24] Z. Yuan, B. Wu, J. He, X. Fu, and H. Chen, "Leader-Follower Formation Source Seeking Control of Multiple Ships Using Sliding Mode Active Disturbance Rejection Observer," *Measurement and Control*, Vol.1, pp.1-11, May. 2020.
- [25] S. Li, R. Kong, and Y. Guo, "Cooperative Distributed Source Seeking by Multiple Robots: Algorithms and Experiments," *IEEE Transactions on Mechatronics*, Vol.19, No.6, pp.1810-1820, 2014.
- [26] W. Wu, I.D. Couzin, and F. Zhang, "Bio-inspired Source Seeking with No Explicit Gradient Estimation," in *Proceedings of IFAC Workshop on Distributed Estimation and Control in Networked Systems*, pp.240-245, Sep. 2012.
- [27] J.R. Bourne, E.R. Pardyjak, and K.K. Leang, "Coordinated Bayesian-Based Bioinspired Plume Source Term Estimation and Source Seeking for Mobile Robots," *IEEE Transactions on Robotics*, Vol. 35, No.4, pp.967-986, Aug. 2019.
- [28] W. Fu, J. Qin, W.X. Zheng, Y. Chen, and Y. Kang., "Resilient Cooperative Source Seeking of Double-Integrator Multi-robot Systems Under Deception Attacks," *IEEE Transactions on Industrial Electronics*, Apr. 2020.

- [29] A.S. Matveev, M.C. Hoy, A.M. Anisimov, and A.V. Savkin, "Tracking of Deforming Environmental Level Sets of Dynamic Fields by a Nonholonomic Robot without Gradient Estimation," in *Proceedings of IEEE ICCA*, pp.1754-1759, Jun. 2013.
- [30] A. Newaz, S. Jeong, H. Lee, and H. Ryu, "UAV-Based Multiple Source Localization and Contour Mapping of Radiation Fields," *Robotics and Autonomous Systems*, Vol.85, pp.12-25, 2016.
- [31] A.S. Matveev and M.S. Nikolaev, "Hybrid Control for Tracking Environmental Level Sets by Nonholonomic Robots in Maze-Like Environments," *Nonlinear Analysis: Hybrid Systems*, Vol.39, pp.1-20, Feb. 2021.
- [32] S. Srinivasan, K. Ramamritham, and P. Kulkarni, "ACE in the Hole: Adaptive Contour Estimation using Collaborating Mobile Sensors," in *Proceedings of Int. Conference on Information Processing in Sensor Networks*, pp.147-158, Apr. 2008.
- [33] W. Li, J. A. Farrell, S. Pang, and R. M. Arrieta, "Moth-Inspired Chemical Plume Tracing on an Autonomous Underwater Vehicle," *IEEE Transactions on Robotics*, Vol.22, No.2, pp.292-307, Apr. 2006.
- [34] Y. Tian, W. Li, A. Zhang, J. Yu, Q. Zhang, and Y. Li, "From Simulation to Validation - Moth-Inspired Chemical Plume Tracing with an Autonomous Underwater Vehicle," in *Proceedings of IEEE Oceans Conference*, St. John's, pp.1-10, Sep. 2014.
- [35] A.S. Matveev, M.C. Hoy, K. Ovchinnikov, A. Anisimov, and A.V. Savkin, "Robot Navigation for Monitoring Unsteady Environmental Boundaries without Field Gradient Estimation," *Automatica*, Vol.62, pp.227-235, Dec. 2015.
- [36] Z. Shen, Z. He, S. Li, Q. Wang, and Z. Shao, "A Multi-Quadcopter Cooperative Cyber-Physical System for Timely Air Pollution Localization," *ACM Transactions on Embedded Computing Systems*, Vol. 16, No. 3, Art. 70, Apr. 2017.
- [37] Z. Jin and A. L. Bertozzi, "Environmental Boundary Tracking and Estimation Using Multiple Autonomous Vehicles," in *Proceedings of IEEE Conference on Decision and Control*, pp.4918-4923, Dec. 2007.
- [38] F. Zhang and N. E. Leonard, "Cooperative Filters and Control for Cooperative Exploration," *IEEE Transactions on Automatic Control*, Vol.55, No.3, pp.650-663, Mar. 2010.
- [39] X. Kang and W. Li, "Moth-Inspired Plume Tracing via Multiple Autonomous Vehicles Under Formation Control," *Adaptive Behavior*, Vol.20, No.2, pp.131-142, Apr. 2012.
- [40] J. Han and Y. Chen, "Cooperative Source Seeking and Contour Mapping of a Diffusive Signal Field by Formations of Multiple UAVs," in *Proc of International Conf on Unmanned Aircraft Systems*, pp.35-40, May 2013.
- [41] J. Han and Y. Chen, "Multiple UAV Formations for Cooperative Source Seeking and Contour Mapping of a Radiative Signal Field," *Journal of Intelligent and Robotic Systems*, Vol.74, pp.323-332, 2014.
- [42] T. Paul, T.R. Krogstad, and J.T. Gravdahl, "UAV Formation Flight Using 3D Potential Field," *2008 16th Mediterranean Conference on Control and Automation*, pp.1240-1245, Jun. 2008.
- [43] Z. Cook, M. Kazemeini, A. Barzilov, and W. Yim, "Low Altitude Contour Mapping of Radiation Fields Using UAS Swarm," *Intelligent Service Robotics*, Vol. 12, pp.219-230, Apr. 2019.

- [44] A. Datar, P. Paulsen, and H. Werner. "Flocking Towards the Source: Indoor Experiments with Quadrotors," *European Control Conference (ECC)*, May. 2020.
- [45] R.T. McDonald, C.A. Kitts, and M.A. Neumann,, "Experimental Implementation and Verification of Scalar Field Ridge, Trench, and Saddle Point Maneuvers Using Multirobot Adaptive Navigation," *IEEE Access*, Vol. 7, pp.62950-62961, 2019.
- [46] R.T. McDonald, M. Condino, M.A. Neumann, and C.A. Kitts, "Navigation of Scalar Fronts with Multirobot Clusters in Simulation and Experiment," *IEEE Systems Journal*, Vol. 14, No. 3, pp.3755-3766, Sep. 2020.
- [47] J. Cochran, N. Ghods, and M. Krstic, "3D Nonholonomic Source Seeking Without Position Measurement," in *Proceedings of Annual American Control Conference*, pp.3518-3523, Jun. 2008.
- [48] J. Cochran, A. Siranosian, N. Ghods, and M. Krstic, "3-D Source Seeking for Underactuated Vehicles Without Position Measurement," *IEEE Transactions on Robotics*, Vol.25, No.1, pp.117-129, Feb. 2009.
- [49] A.S. Matveev, M.C. Hoy, and A.A. Semakova, "3D Environmental Extremum Seeking Navigation of a Nonholonomic Mobile Robot," *Automatica*, Vol.50, pp.1802-1815, Jun. 2014.
- [50] J. Lin, S. Song, K. You, and C. Wu, "3-D Velocity Regulation for Nonholonomic Source Seeking Without Position Measurement," *IEEE Transactions on Control Systems Technology*, Vol.24, No.2, pp.711-718, Mar. 2016.
- [51] S. Al-Abri, W. Wu, and F. Zhang, "A Gradient-Free 3-dimensional Source Seeking Strategy with Robustness Analysis," *IEEE Transactions on Automatic Control*, Vol.66, No.8, pp.3439-3446, Aug. 2019.
- [52] L. Briñón-Arranz, A. Renzaglia, and L. Schenato, "Multirobot Symmetric Formations for Gradient and Hessian Estimation with Application to Source Seeking," *IEEE Transactions on Robotics*, Vol.35, No.3, pp.782-789, Jun. 2019.
- [53] Y. Tan, "Chemical Plume Tracing and Mapping via Swarm Robots," *Handbook of Research on Design, Control, and Modeling of Swarm Robotics*, Ch.16, pp.421-455, 2016.
- [54] A.S. Matveev and A.A. Semakova, "Gradient-Free Navigation of a Nonholonomic Robot for Tracking Unsteady Environmental Boundaries in 3D," in *Proceedings of 22nd International Conference on System Theory, Control, and Computing*, pp.670-676, 2018.
- [55] J. Han, Y. Xu, L. Di, and Y. Chen, "Low-cost Multi-UAV Technologies for Contour Mapping of Nuclear Radiation Field," *Journal of Intelligent and Robotic Systems*, Vol.70, pp.401-410, Apr. 2013.
- [56] W. Wu and F. Zhang, "Cooperative Exploration of Level Surfaces of Three Dimensional Scalar Fields," *Automatica*, Vol.47, No.9, pp.2044-2051, 2011.
- [57] J.W. Wang, Y. Guo, M. Fahad, and B. Bingham, "Dynamic Plume Tracking by Cooperative Robots," *IEEE/ASME Transactions on Mechatronics*, Vol.24, No.2, pp.609-620, Apr. 2019.
- [58] M. Santos, C.D. Rosales, M. Sarcinelli-Filho, and R. Carelli, "A Novel Null-Space-Based UAV Trajectory Tracking Controller with Collision Avoidance," *IEEE/ASME Transactions on Mechatronics*, Vol.22, No. 6, pp.2543-2553, Dec. 2017.
- [59] L.V. Santana, A.S. Brandão, and M. Sarcinelli-Filho, "Navigation and Cooperative Control Using the AR Drone Quadrotor," *Journal of Intelligent and Robotic Systems*, Vol.84, pp.327-350, Dec. 2016.

- [60] L.V. Santana, A.S. Brandão, M. Sarcinelli-Filho, and R. Carelli, "A Trajectory Tracking and 3D Positioning Controller for AR.Drone Quadrotor," in *Proceedings of International Conference on Unmanned Aircraft Systems*, pp.756-767, May. 2014.
- [61] M.S. Selig, "Modeling Propeller Aerodynamics and Slipstream Effects on Small UAVs in Realtime", in *Proceedings of AIAA Atmospheric Flight Mechanics Conference*, Toronto, Canada, pp.7938-7961, Aug. 2010.
- [62] GPS World Staff, "GPS Accuracy: Lies, Damn Lies, and Statistics," *GPS World*, Vol.1, Jan. 1998.
- [63] National Research Council. *The Global Positioning System: A Shared National Asset*. DC: The National Academies Press, 1995, Append D.
- [64] GPS World Staff, "Update: GNSS Accuracy: Lies, Damn Lies, and Statistics," *GPS World*, Vol.1, Jan. 2007.
- [65] Trimble Navigation GPS Receiver data sheets, <http://trl.trimble.com/docushare/dsweb/Get/Document-873444/>
- [66] NovAtel GPS Receiver data sheets, <https://hexagondownloads.blob.core.windows.net/public/Novatel/assets/Documents/Papers/OEM7600-Product-Sheet/OEM7600-Product-Sheet.pdf>
- [67] G.M. Bolla, M. Casagrande, A. Comazzetto, R.D. Moro, M. Destro, E. Fantin, G. Colombatti, A. Aboudan, and E.C. Lorenzini, "ARIA: Air Pollutants Monitoring Using UAVs," in *Proceedings of 5th IEEE International Workshop on Metrology for AeroSpace (MetroAeroSpace)*, pp.225-229, 2018.
- [68] Alphasense Air Quality Sensor data sheets, <http://www.alphasense.com/index.php/air/downloads/>
- [69] L. Yang and S. Liu, "Distributed Stochastic Source Seeking for Multiple Vehicles over Fixed Topology," *Journal of Systems Science and Complexity*, Vol.33, pp.652-671, Jun. 2020.
- [70] C.A. Kitts and I. Mas, "Cluster space specification and control of mobile multirobot systems," *IEEE/ASME Transactions on Mechatronics*, Vol.14, No.2, pp.207-218, Apr. 2009.
- [71] R.K. Lee, C. Kitts, M. Neumann, and R. McDonald, "3-D Adaptive Navigation: Multirobot Formation Control for Seeking and Tracking of a Moving Source," in *Proceedings of 41st IEEE Aerospace Conference*, Mar. 2020.
- [72] C. Kitts, P. Mahacek, T. Adamek, K. Rasal, V. Howard, S. Li, and A. Badaoui, "Field Operation of a Robotic Small Waterplane Area Twin Hull Boat for Shallow-Water Bathymetric Characterization," *Journal of Field Robotics*, Vol.29, No.6, pp.924-938, Nov. 2012.
- [73] I. Mas and C.A. Kitts, "Dynamic Control of Mobile Multirobot Systems - The Cluster Space Formation," *IEEE Access*, Vol.2, pp.558-570, May. 2014.
- [74] Z. Cameron, B. Engh, T. Kambe, and A. Krishnan, "Adaptive Navigation Utilizing a Drone Cluster," *Interdisciplinary Design Senior Theses*. No.59, Santa Clara University, Santa Clara, CA, USA, Jun. 2020.
- [75] C. Kitts, T. Adamek, M. Vlahos, A. Mahacek, K. Poore, J. Guerra, M. Neumann, M. Chin, and M. Rasay, "An Underwater Robotic Testbed for Multi-vehicle Control," *2014 IEEE/OES Autonomous Underwater Vehicles (AUV)*, pp.1-8, Oct. 2014.

- [76] R. K. Lee, C. A. Kitts, M. A. Neumann, and R. T. McDonald, "Multiple UAV Adaptive Navigation for Three-Dimensional Scalar Fields," in *IEEE Access*, Vol. 9, pp.122626-122654, 2021.

APPENDIX A

(UAV Dynamic Model)

(This section contains 1 page)

The simulations presented in this work utilize the simplified dynamic model for a consumer-grade UAV, initially proposed by Santana et al. [60] with an onboard autopilot for platform stabilization with limited pitch and roll angles, in lieu of a full quadrotor dynamic model. Using Parrot Inc's AR.Drone, model parameters were determined by measuring responses for various input signals. They used the simplified model and experimentally demonstrated precise positioning and trajectory tracking in 3D. Santana et al. [59] extended the work to experimentally demonstrate leader-follower positioning and trajectory tracking with two vehicles in 3D. Santos et al. [58] validated the dynamic model through comprehensive experiments for parameter identification and performed precise trajectory tracking with obstacle avoidance in 3D using a single vehicle. Equations (A1) and (A2) contain the K_u and K_v matrices used in this work, which are from Santos et al. [58].

$$K_u = \begin{bmatrix} 12.63 & 0 & 0 & 0 \\ 0 & 7.61 & 0 & 0 \\ 0 & 0 & 6.63 & 0 \\ 0 & 0 & 0 & 1.89 \end{bmatrix} \quad (A1)$$

$$K_v = \begin{bmatrix} 1.43 & 0 & 0 & 0 \\ 0 & 0.84 & 0 & 0 \\ 0 & 0 & 7.56 & 0 \\ 0 & 0 & 0 & 0.54 \end{bmatrix} \quad (A2)$$

APPENDIX B

(Forward Kinematics for Four-Robot Cluster)

(This section contains 3 pages)

Equations (B1) and (B2) define the cluster-space vector \vec{C}_{4R} and robot-space vector \vec{R}_{4R} , respectively, for the four-robot cluster.

$$\vec{C}_{4R} = \begin{bmatrix} {}^G X_B \\ {}^G Y_B \\ {}^G Z_B \\ {}^G \Phi_C \\ {}^G \Theta_C \\ {}^G \Psi_C \\ L_{12} \\ L_{13} \\ L_{B4} \\ \alpha \\ \beta \\ \xi \\ {}^C \Psi_1 \\ {}^C \Psi_2 \\ {}^C \Psi_3 \\ {}^C \Psi_4 \end{bmatrix} \begin{array}{l} \text{cluster } X \text{ coordinate w. r. t. global frame} \\ \text{cluster } Y \text{ coordinate w. r. t. global frame} \\ \text{cluster } Z \text{ coordinate w. r. t. global frame} \\ \text{cluster Roll angle w. r. t. global frame} \\ \text{cluster Pitch angle w. r. t. global frame} \\ \text{cluster Yaw angle w. r. t. global frame} \\ \text{cluster length from Robot 1 and 2} \\ \text{cluster length from Robot 1 and 3} \\ \text{cluster length from Point B and Robot 4} \\ \text{cluster shape angle for } L_{B4} \\ \text{cluster shape angle between } L_{12} \text{ and } L_{13} \\ \text{cluster shape angle for } L_{B4} \\ \text{Robot 1 Yaw angle w. r. t. cluster frame} \\ \text{Robot 2 Yaw angle w. r. t. cluster frame} \\ \text{Robot 3 Yaw angle w. r. t. cluster frame} \\ \text{Robot 4 Yaw angle w. r. t. cluster frame} \end{array} \quad (\text{B1})$$

$$\vec{R}_{4R} = \begin{bmatrix} {}^G x_1 \\ {}^G y_1 \\ {}^G z_1 \\ {}^G \psi_1 \\ {}^G x_2 \\ {}^G y_2 \\ {}^G z_2 \\ {}^G \psi_2 \\ {}^G x_3 \\ {}^G y_3 \\ {}^G z_3 \\ {}^G \psi_3 \\ {}^G x_4 \\ {}^G y_4 \\ {}^G z_4 \\ {}^G \psi_4 \end{bmatrix} \begin{array}{l} \text{Robot 1 } X \text{ coordinate w. r. t. global frame} \\ \text{Robot 1 } Y \text{ coordinate w. r. t. global frame} \\ \text{Robot 1 } Z \text{ coordinate w. r. t. global frame} \\ \text{Robot 1 Yaw angle w. r. t. global frame} \\ \text{Robot 2 } X \text{ coordinate w. r. t. global frame} \\ \text{Robot 2 } Y \text{ coordinate w. r. t. global frame} \\ \text{Robot 2 } Z \text{ coordinate w. r. t. global frame} \\ \text{Robot 2 Yaw angle w. r. t. global frame} \\ \text{Robot 3 } X \text{ coordinate w. r. t. global frame} \\ \text{Robot 3 } Y \text{ coordinate w. r. t. global frame} \\ \text{Robot 3 } Z \text{ coordinate w. r. t. global frame} \\ \text{Robot 3 Yaw angle w. r. t. global frame} \\ \text{Robot 4 } X \text{ coordinate w. r. t. global frame} \\ \text{Robot 4 } Y \text{ coordinate w. r. t. global frame} \\ \text{Robot 4 } Z \text{ coordinate w. r. t. global frame} \\ \text{Robot 4 Yaw angle w. r. t. global frame} \end{array} \quad (\text{B2})$$

Equations (B3) through (B12) are used to compute the cluster frame coordinates and attitude angles.

$${}^G X_B = ({}^G x_1 + {}^G x_2 + {}^G x_3)/3 \quad (\text{B3})$$

$${}^G Y_B = ({}^G y_1 + {}^G y_2 + {}^G y_3)/3 \quad (\text{B4})$$

$${}^G Z_B = ({}^G z_1 + {}^G z_2 + {}^G z_3)/3 \quad (\text{B5})$$

$${}^G \Psi_C = \text{ATAN2} \left(\frac{{}^G y_1 - {}^G y_B}{{}^G x_1 - {}^G x_B} \right) \quad (\text{B6})$$

$${}^G \Theta_C = \sin^{-1} \left(- \frac{({}^G z_1 - {}^G z_B)}{\sqrt{({}^G x_1 - {}^G x_B)^2 + ({}^G y_1 - {}^G y_B)^2 + ({}^G z_1 - {}^G z_B)^2}} \right) \quad (\text{B7})$$

$${}^G \Phi_C = \text{ATAN2} \left(\frac{{}^G x_C a_1 - {}^G x_C a_2}{a_3} \right) \quad (\text{B8})$$

where

$$\begin{bmatrix} {}^G x_C \\ {}^G y_C \\ {}^G z_C \end{bmatrix} = \begin{bmatrix} \frac{({}^G x_1 - {}^G x_B)}{\sqrt{({}^G x_1 - {}^G x_B)^2 + ({}^G y_1 - {}^G y_B)^2 + ({}^G z_1 - {}^G z_B)^2}} \\ \frac{({}^G y_1 - {}^G y_B)}{\sqrt{({}^G x_1 - {}^G x_B)^2 + ({}^G y_1 - {}^G y_B)^2 + ({}^G z_1 - {}^G z_B)^2}} \\ \frac{({}^G z_1 - {}^G z_B)}{\sqrt{({}^G x_1 - {}^G x_B)^2 + ({}^G y_1 - {}^G y_B)^2 + ({}^G z_1 - {}^G z_B)^2}} \end{bmatrix} \quad (\text{B9})$$

$$a_1 = [({}^G y_2 - {}^G y_1)({}^G z_3 - {}^G z_1) - ({}^G y_3 - {}^G y_1)({}^G z_2 - {}^G z_1)] \quad (\text{B10})$$

$$a_2 = [({}^G x_3 - {}^G x_1)({}^G z_2 - {}^G z_1) - ({}^G x_2 - {}^G x_1)({}^G z_3 - {}^G z_1)] \quad (\text{B11})$$

$$a_3 = [({}^G x_2 - {}^G x_1)({}^G y_3 - {}^G y_1) - ({}^G x_3 - {}^G x_1)({}^G y_2 - {}^G y_1)] \quad (\text{B12})$$

Equations (B13) through (B15) are used to compute the cluster formation lengths.

$$L_{12} = \sqrt{({}^G x_2 - {}^G x_1)^2 + ({}^G y_2 - {}^G y_1)^2 + ({}^G z_2 - {}^G z_1)^2} \quad (\text{B13})$$

$$L_{13} = \sqrt{({}^G x_3 - {}^G x_1)^2 + ({}^G y_3 - {}^G y_1)^2 + ({}^G z_3 - {}^G z_1)^2} \quad (\text{B14})$$

$$L_{B4} = \sqrt{({}^Gx_4 - {}^GX_B)^2 + ({}^Gy_4 - {}^GY_B)^2 + ({}^Gz_4 - {}^GZ_B)^2} \quad (B15)$$

Equations (B16) through (B20) are used to compute the internal angles of the cluster formation.

$$\alpha = ATAN2({}^cL_{B4}/{}^xL_{B4}) \quad (B16)$$

$$\beta = \cos^{-1} \left(\frac{({}^Gx_2 - {}^Gx_1)({}^Gx_3 - {}^Gx_1) + ({}^Gy_2 - {}^Gy_1)({}^Gy_3 - {}^Gy_1) + ({}^Gz_2 - {}^Gz_1)({}^Gz_3 - {}^Gz_1)}{L_{12} * L_{13}} \right) \quad (B17)$$

$$\xi = ATAN2 \left(\sqrt{{}^xL_{B4}^2 + {}^yL_{B4}^2} / {}^zL_{B4} \right) \quad (B18)$$

where

$$\begin{bmatrix} {}^cL_{B4} \\ {}^xL_{B4} \\ {}^yL_{B4} \\ {}^zL_{B4} \end{bmatrix} = {}^cR \cdot \begin{bmatrix} ({}^Gx_4 - {}^GX_B) \\ ({}^Gy_4 - {}^GY_B) \\ ({}^Gz_4 - {}^GZ_B) \end{bmatrix} \quad (B19)$$

$${}^cR = \begin{bmatrix} C\Psi_c C\theta_c & S\Psi_c C\theta_c & -S\theta_c \\ C\Psi_c S\theta_c S\phi_c - C\phi_c S\Psi_c & S\phi_c S\theta_c S\Psi_c + C\Psi_c C\phi_c & C\theta_c S\phi_c \\ C\phi_c S\theta_c C\Psi_c + S\Psi_c S\phi_c & S\Psi_c S\theta_c C\phi_c - S\phi_c C\Psi_c & C\theta_c C\phi_c \end{bmatrix} \quad (B20)$$

For compactness, (B20) uses the notation C(#) and S(#) for cosine and sine functions, respectively. Equations (B21) through (B23) are used to compute individual robot yaw angles relative to frame C, for Robots 1 to 4 where $n = 1, \dots, 4$.

$${}^c\Psi_n = ATAN2[{}^cX_n / {}^cX_n] \quad (B21)$$

where

$${}^G_RnR = \begin{bmatrix} {}^cX_n & {}^cY_n & {}^cZ_n \\ {}^xX_n & {}^yY_n & {}^zZ_n \\ {}^zX_n & {}^zY_n & {}^zZ_n \end{bmatrix} = {}^cR \cdot {}^G_RnR \quad (B22)$$

$${}^G_RnR = \begin{bmatrix} C\psi_n C\theta_n & C\psi_n S\theta_n S\phi_n - C\phi_n S\psi_n & C\phi_n S\theta_n C\psi_n + S\psi_n S\phi_n \\ S\psi_n C\theta_n & S\phi_n S\theta_n S\psi_n + C\psi_n C\phi_n & S\psi_n S\theta_n C\phi_n - S\phi_n C\psi_n \\ -S\theta_n & C\theta_n S\phi_n & C\theta_n C\phi_n \end{bmatrix} \quad (B23)$$

APPENDIX C

(Forward Kinematics for Nine-Robot Cluster)

(This section contains 6 pages)

Equations (C1) and (C2) define the cluster-space vector \vec{C}_{9R} and robot-space vector \vec{R}_{9R} , respectively, for the nine-robot cluster. Equations (C3) through (C10) are used to compute the cluster frame coordinates and attitude angles.

$$\vec{C}_{9R} = \begin{bmatrix} {}^G X_C \\ {}^G Y_C \\ {}^G Z_C \\ {}^G \Phi_C \\ {}^G \Theta_C \\ {}^G \Psi_C \\ L_{C2} \\ L_{23} \\ L_{C4} \\ L_{45} \\ L_{C6} \\ L_{67} \\ L_{C8} \\ L_{89} \\ \zeta \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \\ \alpha_7 \\ \alpha_9 \\ \beta_3 \\ \beta_4 \\ \beta_5 \\ \beta_6 \\ \beta_7 \\ \beta_9 \\ c\psi_1 \\ c\psi_2 \\ c\psi_3 \\ c\psi_4 \\ c\psi_5 \\ c\psi_6 \\ c\psi_7 \\ c\psi_8 \\ c\psi_9 \end{bmatrix} \begin{array}{l} \text{cluster } X \text{ coordinate w.r. t. global frame} \\ \text{cluster } Y \text{ coordinate w.r. t. global frame} \\ \text{cluster } Z \text{ coordinate w.r. t. global frame} \\ \text{cluster Roll angle w.r. t. global frame} \\ \text{cluster Pitch angle w.r. t. global frame} \\ \text{cluster Yaw angle w.r. t. global frame} \\ \text{cluster length from Point C and Robot 2} \\ \text{cluster length from Robot 2 and Robot 3} \\ \text{cluster length from Point C and Robot 4} \\ \text{cluster length from Robot 4 and Robot 5} \\ \text{cluster length from Point C and Robot 6} \\ \text{cluster length from Robot 6 and Robot 7} \\ \text{cluster length from Point C and Robot 8} \\ \text{cluster length from Robot 8 and Robot 9} \\ \text{cluster shape angle between } \hat{Y}_C \text{ and } L_{C8} \\ \text{cluster shape angle } \perp \text{ to } \hat{X}_C \hat{Y}_C \text{ plane} \\ \text{cluster shape angle } \perp \text{ to } \hat{X}_C \hat{Y}_C \text{ plane} \\ \text{cluster shape angle } \perp \text{ to } \hat{X}_C \hat{Y}_C \text{ plane} \\ \text{cluster shape angle } \perp \text{ to } \hat{X}_C \hat{Z}_C \text{ plane} \\ \text{cluster shape angle } \perp \text{ to } \hat{X}_C \hat{Z}_C \text{ plane} \\ \text{cluster shape angle } \perp \text{ to } \hat{X}_C \hat{Y}_C \text{ plane} \\ \text{cluster shape angle } \parallel \text{ to } \hat{X}_C \hat{Y}_C \text{ plane} \\ \text{cluster shape angle } \parallel \text{ to } \hat{X}_C \hat{Y}_C \text{ plane} \\ \text{cluster shape angle } \parallel \text{ to } \hat{X}_C \hat{Y}_C \text{ plane} \\ \text{cluster shape angle } \parallel \text{ to } \hat{X}_C \hat{Z}_C \text{ plane} \\ \text{cluster shape angle } \parallel \text{ to } \hat{X}_C \hat{Z}_C \text{ plane} \\ \text{cluster shape angle } \parallel \text{ to } \hat{X}_C \hat{Z}_C \text{ plane} \\ \text{Robot 1 Yaw angle w.r. t. cluster frame} \\ \text{Robot 2 Yaw angle w.r. t. cluster frame} \\ \text{Robot 3 Yaw angle w.r. t. cluster frame} \\ \text{Robot 4 Yaw angle w.r. t. cluster frame} \\ \text{Robot 5 Yaw angle w.r. t. cluster frame} \\ \text{Robot 6 Yaw angle w.r. t. cluster frame} \\ \text{Robot 7 Yaw angle w.r. t. cluster frame} \\ \text{Robot 8 Yaw angle w.r. t. cluster frame} \\ \text{Robot 9 Yaw angle w.r. t. cluster frame} \end{array} \quad (C1)$$

$$\bar{R}_{9R} = \begin{bmatrix} {}^Gx_1 & \text{Robot 1 X coordinate w.r.t. global frame} \\ {}^Gy_1 & \text{Robot 1 Y coordinate w.r.t. global frame} \\ {}^Gz_1 & \text{Robot 1 Z coordinate w.r.t. global frame} \\ {}^G\psi_1 & \text{Robot 1 Yaw angle w.r.t. global frame} \\ {}^Gx_2 & \text{Robot 2 X coordinate w.r.t. global frame} \\ {}^Gy_2 & \text{Robot 2 Y coordinate w.r.t. global frame} \\ {}^Gz_2 & \text{Robot 2 Z coordinate w.r.t. global frame} \\ {}^G\psi_2 & \text{Robot 2 Yaw angle w.r.t. global frame} \\ {}^Gx_3 & \text{Robot 3 X coordinate w.r.t. global frame} \\ {}^Gy_3 & \text{Robot 3 Y coordinate w.r.t. global frame} \\ {}^Gz_3 & \text{Robot 3 Z coordinate w.r.t. global frame} \\ {}^G\psi_3 & \text{Robot 3 Yaw angle w.r.t. global frame} \\ {}^Gx_4 & \text{Robot 4 X coordinate w.r.t. global frame} \\ {}^Gy_4 & \text{Robot 4 Y coordinate w.r.t. global frame} \\ {}^Gz_4 & \text{Robot 4 Z coordinate w.r.t. global frame} \\ {}^G\psi_4 & \text{Robot 4 Yaw angle w.r.t. global frame} \\ {}^Gx_5 & \text{Robot 5 X coordinate w.r.t. global frame} \\ {}^Gy_5 & \text{Robot 5 Y coordinate w.r.t. global frame} \\ {}^Gz_5 & \text{Robot 5 Z coordinate w.r.t. global frame} \\ {}^G\psi_5 & \text{Robot 5 Yaw angle w.r.t. global frame} \\ {}^Gx_6 & \text{Robot 6 X coordinate w.r.t. global frame} \\ {}^Gy_6 & \text{Robot 6 Y coordinate w.r.t. global frame} \\ {}^Gz_6 & \text{Robot 6 Z coordinate w.r.t. global frame} \\ {}^G\psi_6 & \text{Robot 6 Yaw angle w.r.t. global frame} \\ {}^Gx_7 & \text{Robot 7 X coordinate w.r.t. global frame} \\ {}^Gy_7 & \text{Robot 7 Y coordinate w.r.t. global frame} \\ {}^Gz_7 & \text{Robot 7 Z coordinate w.r.t. global frame} \\ {}^G\psi_7 & \text{Robot 7 Yaw angle w.r.t. global frame} \\ {}^Gx_8 & \text{Robot 8 X coordinate w.r.t. global frame} \\ {}^Gy_8 & \text{Robot 8 Y coordinate w.r.t. global frame} \\ {}^Gz_8 & \text{Robot 8 Z coordinate w.r.t. global frame} \\ {}^G\psi_8 & \text{Robot 8 Yaw angle w.r.t. global frame} \\ {}^Gx_9 & \text{Robot 9 X coordinate w.r.t. global frame} \\ {}^Gy_9 & \text{Robot 9 Y coordinate w.r.t. global frame} \\ {}^Gz_9 & \text{Robot 9 Z coordinate w.r.t. global frame} \\ {}^G\psi_9 & \text{Robot 9 Yaw angle w.r.t. global frame} \end{bmatrix} \quad (C2)$$

$$\begin{bmatrix} {}^GX_C & {}^GY_C & {}^GZ_C \end{bmatrix}^T = \begin{bmatrix} {}^Gx_1 & {}^Gy_1 & {}^Gz_1 \end{bmatrix}^T \quad (C3)$$

$${}^G\Psi_C = ATAN2\left(\frac{-{}^Gx_1 {}^Gz_2 + {}^Gx_2 {}^Gz_1 + {}^Gx_1 {}^Gz_8 - {}^Gx_8 {}^Gz_1 - {}^Gx_2 {}^Gz_8 + {}^Gx_8 {}^Gz_2}{{}^Gy_1 {}^Gz_2 - {}^Gy_2 {}^Gz_1 - {}^Gy_1 {}^Gz_8 + {}^Gy_8 {}^Gz_1 + {}^Gy_2 {}^Gz_8 - {}^Gy_8 {}^Gz_2}\right) \quad (C4)$$

$${}^G\Theta_C = \sin^{-1} \left(\frac{-{}^Gx_1 {}^Gy_2 + {}^Gx_2 {}^Gy_1 + {}^Gx_1 {}^Gy_8 - {}^Gx_8 {}^Gy_1 - {}^Gx_2 {}^Gy_8 + {}^Gx_8 {}^Gy_2}{b_1 * b_2} \right) \quad (C5)$$

$${}^G\Phi_C = \text{ATAN2} \left(\frac{({}^Gz_2 - {}^Gz_1)(b_1 * b_2)}{({}^Gx_2 - {}^Gx_1)b_3 + ({}^Gy_2 - {}^Gy_1)b_4} \right) \quad (C6)$$

where

$$b_1 = \sqrt{({}^Gx_2 - {}^Gx_1)^2 + ({}^Gy_2 - {}^Gy_1)^2 + ({}^Gz_2 - {}^Gz_1)^2} \quad (C7)$$

$$b_2 = \sqrt{({}^Gx_8 - {}^Gx_1)^2 + ({}^Gy_8 - {}^Gy_1)^2 + ({}^Gz_8 - {}^Gz_1)^2} \quad (C8)$$

$$b_3 = ({}^Gx_1 {}^Gz_2 - {}^Gx_2 {}^Gz_1 - {}^Gx_1 {}^Gz_8 + {}^Gx_8 {}^Gz_1 + {}^Gx_2 {}^Gz_8 - {}^Gx_8 {}^Gz_2) \quad (C9)$$

$$b_4 = ({}^Gy_1 {}^Gz_2 - {}^Gy_2 {}^Gz_1 - {}^Gy_1 {}^Gz_8 + {}^Gy_8 {}^Gz_1 + {}^Gy_2 {}^Gz_8 - {}^Gy_8 {}^Gz_2) \quad (C10)$$

Equations (C11) through (C18) are used to compute the cluster formation lengths.

$$L_{C2} = \sqrt{({}^Gx_2 - {}^GX_C)^2 + ({}^Gy_2 - {}^GY_C)^2 + ({}^Gz_2 - {}^GZ_C)^2} \quad (C11)$$

$$L_{23} = \sqrt{({}^Gx_3 - {}^Gx_2)^2 + ({}^Gy_3 - {}^Gy_2)^2 + ({}^Gz_3 - {}^Gz_2)^2} \quad (C12)$$

$$L_{C4} = \sqrt{({}^Gx_4 - {}^GX_C)^2 + ({}^Gy_4 - {}^GY_C)^2 + ({}^Gz_4 - {}^GZ_C)^2} \quad (C13)$$

$$L_{45} = \sqrt{({}^Gx_5 - {}^Gx_4)^2 + ({}^Gy_5 - {}^Gy_4)^2 + ({}^Gz_5 - {}^Gz_4)^2} \quad (C14)$$

$$L_{C6} = \sqrt{({}^Gx_6 - {}^GX_C)^2 + ({}^Gy_6 - {}^GY_C)^2 + ({}^Gz_6 - {}^GZ_C)^2} \quad (C15)$$

$$L_{67} = \sqrt{({}^Gx_7 - {}^Gx_6)^2 + ({}^Gy_7 - {}^Gy_6)^2 + ({}^Gz_7 - {}^Gz_6)^2} \quad (C16)$$

$$L_{c8} = \sqrt{({}^Gx_8 - {}^GX_C)^2 + ({}^Gy_8 - {}^GY_C)^2 + ({}^Gz_8 - {}^GZ_C)^2} \quad (C17)$$

$$L_{89} = \sqrt{({}^Gx_9 - {}^Gx_8)^2 + ({}^Gy_9 - {}^Gy_8)^2 + ({}^Gz_9 - {}^Gz_8)^2} \quad (C18)$$

Equations (C19) through (C37) compute the internal angles of the cluster formation.

$$\zeta = \cos^{-1} \left(\frac{\begin{bmatrix} ({}^Gx_2 - {}^GX_C) \\ ({}^Gy_2 - {}^GY_C) \\ ({}^Gz_2 - {}^GZ_C) \end{bmatrix}}{L_{C2}} \cdot \frac{\begin{bmatrix} ({}^Gx_8 - {}^GX_C) \\ ({}^Gy_8 - {}^GY_C) \\ ({}^Gz_8 - {}^GZ_C) \end{bmatrix}}{L_{C8}} \right) \quad (C19)$$

$$\alpha_4 = -\sin^{-1} \left[\frac{({}^Gy_1 - {}^Gy_4)(C {}^G\Psi_C S {}^G\Phi_C - C {}^G\Phi_C S {}^G\Psi_C S {}^G\Theta_C)}{L_{C4}} + \dots \right. \\ \left. \frac{({}^Gx_4 - {}^Gx_1)(S {}^G\Psi_C S {}^G\Phi_C + C {}^G\Phi_C C {}^G\Psi_C S {}^G\Theta_C)}{L_{C4}} + \frac{({}^Gz_4 - {}^Gz_1)(C {}^G\Phi_C C {}^G\Theta_C)}{L_{C4}} \right] \quad (C20)$$

$$\alpha_6 = \sin^{-1} \left[\frac{({}^Gx_1 - {}^Gx_6)(S {}^G\Psi_C C {}^G\Phi_C - S {}^G\Phi_C C {}^G\Psi_C S {}^G\Theta_C)}{L_{C6}} + \dots \right. \\ \left. \frac{({}^Gy_6 - {}^Gy_1)(C {}^G\Psi_C C {}^G\Phi_C + S {}^G\Phi_C S {}^G\Psi_C S {}^G\Theta_C)}{L_{C6}} + \frac{({}^Gz_6 - {}^Gz_1)(S {}^G\Phi_C C {}^G\Theta_C)}{L_{C6}} \right] \quad (C21)$$

For h = 3, 5:

$$\alpha_h = -\sin^{-1} \left[\frac{({}^Gy_{h-1} - {}^Gy_h)(C {}^G\Psi_C S {}^G\Phi_C - C {}^G\Phi_C S {}^G\Psi_C S {}^G\Theta_C)}{L_{(h-1)(h)}} + \dots \right. \\ \left. \frac{({}^Gx_h - {}^Gx_{h-1})(S {}^G\Psi_C S {}^G\Phi_C + C {}^G\Phi_C C {}^G\Psi_C S {}^G\Theta_C)}{L_{(h-1)(h)}} + \frac{({}^Gz_h - {}^Gz_{h-1})(C {}^G\Phi_C C {}^G\Theta_C)}{L_{(h-1)(h)}} \right] \quad (C22)$$

For j = 7, 9:

$$\alpha_j = \sin^{-1} \left[\frac{({}^G x_{j-1} - {}^G x_j)(S {}^G \Psi_C C {}^G \Phi_C - S {}^G \Phi_C C {}^G \Psi_C S {}^G \Theta_C)}{L_{(j-1)(j)}} + \dots \right. \\ \left. \frac{({}^G y_j - {}^G y_{j-1})(C {}^G \Psi_C C {}^G \Phi_C + S {}^G \Phi_C S {}^G \Psi_C S {}^G \Theta_C)}{L_{(j-1)(j)}} + \frac{({}^G z_j - {}^G z_{j-1})(S {}^G \Phi_C C {}^G \Theta_C)}{L_{(j-1)(j)}} \right] \quad (C23)$$

$$\beta_4 = ATAN2[q_1/(b_5 + b_6 + b_7)] \quad (C24)$$

where

$$q_1 = C {}^G \Psi_C C {}^G \Theta_C ({}^G x_4 - {}^G x_1) + S {}^G \Psi_C C {}^G \Theta_C ({}^G y_4 - {}^G y_1) + S {}^G \Theta_C ({}^G z_1 - {}^G z_4) \quad (C25)$$

$$b_5 = ({}^G x_4 - {}^G x_1)(S {}^G \Psi_C C {}^G \Phi_C - S {}^G \Phi_C C {}^G \Psi_C S {}^G \Theta_C) \quad (C26)$$

$$b_6 = ({}^G y_4 - {}^G y_1)(C {}^G \Psi_C C {}^G \Phi_C + S {}^G \Phi_C S {}^G \Psi_C S {}^G \Theta_C) \quad (C27)$$

$$b_7 = ({}^G z_1 - {}^G z_4)(S {}^G \Phi_C C {}^G \Theta_C) \quad (C28)$$

$$\beta_6 = ATAN2[q_2/(b_8 + b_9 + b_{10})] \quad (C29)$$

where

$$q_2 = C {}^G \Psi_C C {}^G \Theta_C ({}^G x_1 - {}^G x_6) + S {}^G \Psi_C C {}^G \Theta_C ({}^G y_1 - {}^G y_6) + S {}^G \Theta_C ({}^G z_6 - {}^G z_1) \quad (C30)$$

$$b_8 = ({}^G x_1 - {}^G x_6)(S {}^G \Psi_C S {}^G \Phi_C + C {}^G \Phi_C C {}^G \Psi_C S {}^G \Theta_C) \quad (C31)$$

$$b_9 = ({}^G y_6 - {}^G y_1)(C {}^G \Psi_C S {}^G \Phi_C - C {}^G \Phi_C S {}^G \Psi_C S {}^G \Theta_C) \quad (C32)$$

$$b_{10} = ({}^G z_1 - {}^G z_6)(C {}^G \Phi_C C {}^G \Theta_C) \quad (C33)$$

For $p = 3, 5$:

$$\beta_p = ATAN2 \left[\frac{({}^G x_{p-1} - {}^G x_p)(S {}^G \Psi_C C {}^G \Phi_C - S {}^G \Phi_C C {}^G \Psi_C S {}^G \Theta_C)}{b_{11}} + \dots \right]$$

$$\left[\frac{({}^G y_p - {}^G y_{p-1})(C {}^G \Psi_C C {}^G \Phi_C + S {}^G \Phi_C S {}^G \Psi_C S {}^G \theta_C)}{b_{11}} + \frac{({}^G z_p - {}^G z_{p-1})(S {}^G \Phi_C C {}^G \theta_C)}{b_{11}} \right] \quad (C34)$$

where

$$b_{11} = C {}^G \Psi_C C {}^G \theta_C ({}^G x_p - {}^G x_{p-1}) + S {}^G \Psi_C C {}^G \theta_C ({}^G y_p - {}^G y_{p-1}) + S {}^G \theta_C ({}^G z_{p-1} - {}^G z_p) \quad (C35)$$

For r = 7, 9:

$$\beta_r = ATAN2 \left[\frac{({}^G x_{r-1} - {}^G x_r)(S {}^G \Psi_C S {}^G \Phi_C + C {}^G \Phi_C C {}^G \Psi_C S {}^G \theta_C)}{b_{12}} + \dots \right. \\ \left. \frac{({}^G y_r - {}^G y_{r-1})(C {}^G \Psi_C S {}^G \Phi_C - C {}^G \Phi_C S {}^G \Psi_C S {}^G \theta_C)}{b_{12}} + \frac{({}^G z_{r-1} - {}^G z_r)(C {}^G \Phi_C C {}^G \theta_C)}{b_{12}} \right] \quad (C36)$$

where

$$b_{12} = C {}^G \Psi_C C {}^G \theta_C ({}^G x_r - {}^G x_{r-1}) + S {}^G \Psi_C C {}^G \theta_C ({}^G y_r - {}^G y_{r-1}) + S {}^G \theta_C ({}^G z_{r-1} - {}^G z_r) \quad (C37)$$

Equations (B21) through (B23) are used to compute individual robot yaw angles relative to frame C, for Robots 1 to 9 where n = 1, ..., 9.

APPENDIX D

(Scalar Fields Plumes)

(This section contains 1 page)

Plume source coordinates x_p and y_p with respect to G, are constant for stationary plumes. Variables x , y , z are coordinates of Robot n with respect to G. Equation (D1) generates the scalar field plumes in the sections listed below.

$$S_n = p_1 * \frac{e^{-[|0.001 \cdot z| * (\text{sign}(z) + 1)]}}{\left(\frac{\sqrt{(x - (x_p + p_2))^2 + (y - (y_p + p_3))^2}}{p_4 * (0.1 * z + 1)} \right)^2 + 1} \quad (\text{D1})$$

Section 4.4.1: symmetric, stationary, and time- invariant plume where $p_1 = 150$, $p_2 = 0$, $p_3 = 0$, and $p_4 = 25$

Section 4.4.2: symmetric, moving, and time-varying plume where $p_1 = (250 + t)$, $p_2 = 0$, $p_3 = 0$, and $p_4 = 25$

Section 5.2 and 5.4: symmetric, stationary, and time-invariant plume where $p_1 = 30$, $p_2 = 0$, $p_3 = 0$, and $p_4 = 8$

Section 5.4: asymmetric, stationary, and time-invariant plume where $p_1 = 300$, $p_2 = 0$, $p_3 = e^{0.035z}$, and $p_4 = 30$

Section 5.4: overlapping, symmetric, stationary, and time-invariant plumes where $p_1 = 15$, $p_2 = 0$, $p_3 = 0$, and $p_4 = 10$

Section 6.4: asymmetric, stationary, and time-invariant plume where $p_1 = 300$, $p_2 = 0$, $p_3 = e^{0.035z}$, and $p_4 = 10$

Section 6.4: high scalar magnitude, asymmetric, stationary, and time-invariant plume where $p_1 = 1000$, $p_2 = e^{0.035z}$, $p_3 = 0$, and $p_4 = 10$

Section 6.6: high scalar magnitude, asymmetric, stationary, and time-invariant plume where $p_1 = 1000$, $p_2 = e^{0.035z}$, $p_3 = 0$, and $p_4 = 10$

Section 6.6: asymmetric, stationary, and time-invariant plume where $p_1 = 300$, $p_2 = 0$, $p_3 = e^{0.035z}$, and $p_4 = 10$

APPENDIX E

(Inverse Jacobian Matrix for Four-Robot Cluster as Implemented in Simulation)

(This section contains 23 pages)

Velocity Mapping from Cluster-Space to Robot-Space as Implemented in Simulation

Robot-Space Velocities		Cluster-Space Velocities		Cluster-Space Velocities		Robot-Space Velocities
$\begin{bmatrix} {}^G\dot{x}_1 \\ {}^G\dot{x}_2 \\ {}^G\dot{x}_3 \\ {}^G\dot{x}_4 \\ {}^G\dot{y}_1 \\ {}^G\dot{y}_2 \\ {}^G\dot{y}_3 \\ {}^G\dot{y}_4 \\ {}^G\dot{z}_1 \\ {}^G\dot{z}_2 \\ {}^G\dot{z}_3 \\ {}^G\dot{z}_4 \\ {}^G\dot{\psi}_1 \\ {}^G\dot{\theta}_1 \\ {}^G\dot{\phi}_1 \\ {}^G\dot{\psi}_2 \\ {}^G\dot{\theta}_2 \\ {}^G\dot{\phi}_2 \\ {}^G\dot{\psi}_3 \\ {}^G\dot{\theta}_3 \\ {}^G\dot{\phi}_3 \\ {}^G\dot{\psi}_4 \\ {}^G\dot{\theta}_4 \\ {}^G\dot{\phi}_4 \end{bmatrix}$	$= J^{-1} \cdot$	$\begin{bmatrix} {}^G\dot{X}_B \\ {}^G\dot{Y}_B \\ {}^G\dot{Z}_B \\ \dot{L}_{12} \\ \dot{L}_{13} \\ \dot{L}_{B4} \\ {}^G\dot{\psi}_C \\ {}^G\dot{\theta}_C \\ {}^G\dot{\phi}_C \\ \dot{\alpha} \\ \dot{\beta} \\ \dot{\xi} \\ {}^C\dot{\psi}_1 \\ {}^C\dot{\theta}_1 \\ {}^C\dot{\phi}_1 \\ {}^C\dot{\psi}_2 \\ {}^C\dot{\theta}_2 \\ {}^C\dot{\phi}_2 \\ {}^C\dot{\psi}_3 \\ {}^C\dot{\theta}_3 \\ {}^C\dot{\phi}_3 \\ {}^C\dot{\psi}_4 \\ {}^C\dot{\theta}_4 \\ {}^C\dot{\phi}_4 \end{bmatrix}$		$\begin{bmatrix} {}^G\dot{X}_B \\ {}^G\dot{Y}_B \\ {}^G\dot{Z}_B \\ \dot{L}_{12} \\ \dot{L}_{13} \\ \dot{L}_{B4} \\ {}^G\dot{\psi}_C \\ {}^G\dot{\theta}_C \\ {}^G\dot{\phi}_C \\ \dot{\alpha} \\ \dot{\beta} \\ \dot{\xi} \\ {}^C\dot{\psi}_1 \\ {}^C\dot{\theta}_1 \\ {}^C\dot{\phi}_1 \\ {}^C\dot{\psi}_2 \\ {}^C\dot{\theta}_2 \\ {}^C\dot{\phi}_2 \\ {}^C\dot{\psi}_3 \\ {}^C\dot{\theta}_3 \\ {}^C\dot{\phi}_3 \\ {}^C\dot{\psi}_4 \\ {}^C\dot{\theta}_4 \\ {}^C\dot{\phi}_4 \end{bmatrix}$	$= J \cdot$	$\begin{bmatrix} {}^G\dot{x}_1 \\ {}^G\dot{x}_2 \\ {}^G\dot{x}_3 \\ {}^G\dot{x}_4 \\ {}^G\dot{y}_1 \\ {}^G\dot{y}_2 \\ {}^G\dot{y}_3 \\ {}^G\dot{y}_4 \\ {}^G\dot{z}_1 \\ {}^G\dot{z}_2 \\ {}^G\dot{z}_3 \\ {}^G\dot{z}_4 \\ {}^G\dot{\psi}_1 \\ {}^G\dot{\theta}_1 \\ {}^G\dot{\phi}_1 \\ {}^G\dot{\psi}_2 \\ {}^G\dot{\theta}_2 \\ {}^G\dot{\phi}_2 \\ {}^G\dot{\psi}_3 \\ {}^G\dot{\theta}_3 \\ {}^G\dot{\phi}_3 \\ {}^G\dot{\psi}_4 \\ {}^G\dot{\theta}_4 \\ {}^G\dot{\phi}_4 \end{bmatrix}$

```
% Inverse Jacobian Matrix for Four-Robot Cluster
```

```
IJ = [...  
IJ_1_1  IJ_1_2  IJ_1_3  IJ_1_4  IJ_1_5  IJ_1_6  IJ_1_7  IJ_1_8  IJ_1_9  IJ_1_10  
IJ_1_11 IJ_1_12...  
IJ_1_13 IJ_1_14 IJ_1_15 IJ_1_16 IJ_1_17 IJ_1_18 IJ_1_19 IJ_1_20 IJ_1_21 IJ_1_22  
IJ_1_23 IJ_1_24...  
  
IJ_2_1  IJ_2_2  IJ_2_3  IJ_2_4  IJ_2_5  IJ_2_6  IJ_2_7  IJ_2_8  IJ_2_9  IJ_2_10  
IJ_2_11 IJ_2_12...  
IJ_2_13 IJ_2_14 IJ_2_15 IJ_2_16 IJ_2_17 IJ_2_18 IJ_2_19 IJ_2_20 IJ_2_21 IJ_2_22  
IJ_2_23 IJ_2_24...  
  
IJ_3_1  IJ_3_2  IJ_3_3  IJ_3_4  IJ_3_5  IJ_3_6  IJ_3_7  IJ_3_8  IJ_3_9  IJ_3_10  
IJ_3_11 IJ_3_12...  
IJ_3_13 IJ_3_14 IJ_3_15 IJ_3_16 IJ_3_17 IJ_3_18 IJ_3_19 IJ_3_20 IJ_3_21 IJ_3_22  
IJ_3_23 IJ_3_24...  
  
IJ_4_1  IJ_4_2  IJ_4_3  IJ_4_4  IJ_4_5  IJ_4_6  IJ_4_7  IJ_4_8  IJ_4_9  IJ_4_10  
IJ_4_11 IJ_4_12...  
IJ_4_13 IJ_4_14 IJ_4_15 IJ_4_16 IJ_4_17 IJ_4_18 IJ_4_19 IJ_4_20 IJ_4_21 IJ_4_22  
IJ_4_23 IJ_4_24...  
  
IJ_5_1  IJ_5_2  IJ_5_3  IJ_5_4  IJ_5_5  IJ_5_6  IJ_5_7  IJ_5_8  IJ_5_9  IJ_5_10  
IJ_5_11 IJ_5_12...  
IJ_5_13 IJ_5_14 IJ_5_15 IJ_5_16 IJ_5_17 IJ_5_18 IJ_5_19 IJ_5_20 IJ_5_21 IJ_5_22  
IJ_5_23 IJ_5_24...  
  
IJ_6_1  IJ_6_2  IJ_6_3  IJ_6_4  IJ_6_5  IJ_6_6  IJ_6_7  IJ_6_8  IJ_6_9  IJ_6_10  
IJ_6_11 IJ_6_12...  
IJ_6_13 IJ_6_14 IJ_6_15 IJ_6_16 IJ_6_17 IJ_6_18 IJ_6_19 IJ_6_20 IJ_6_21 IJ_6_22  
IJ_6_23 IJ_6_24...  
  
IJ_7_1  IJ_7_2  IJ_7_3  IJ_7_4  IJ_7_5  IJ_7_6  IJ_7_7  IJ_7_8  IJ_7_9  IJ_7_10  
IJ_7_11 IJ_7_12...  
IJ_7_13 IJ_7_14 IJ_7_15 IJ_7_16 IJ_7_17 IJ_7_18 IJ_7_19 IJ_7_20 IJ_7_21 IJ_7_22  
IJ_7_23 IJ_7_24...  
  
IJ_8_1  IJ_8_2  IJ_8_3  IJ_8_4  IJ_8_5  IJ_8_6  IJ_8_7  IJ_8_8  IJ_8_9  IJ_8_10  
IJ_8_11 IJ_8_12...  
IJ_8_13 IJ_8_14 IJ_8_15 IJ_8_16 IJ_8_17 IJ_8_18 IJ_8_19 IJ_8_20 IJ_8_21 IJ_8_22  
IJ_8_23 IJ_8_24...  
  
IJ_9_1  IJ_9_2  IJ_9_3  IJ_9_4  IJ_9_5  IJ_9_6  IJ_9_7  IJ_9_8  IJ_9_9  IJ_9_10  
IJ_9_11 IJ_9_12...  
IJ_9_13 IJ_9_14 IJ_9_15 IJ_9_16 IJ_9_17 IJ_9_18 IJ_9_19 IJ_9_20 IJ_9_21 IJ_9_22  
IJ_9_23 IJ_9_24...  
  
IJ_10_1 IJ_10_2 IJ_10_3 IJ_10_4 IJ_10_5 IJ_10_6 IJ_10_7 IJ_10_8 IJ_10_9  
IJ_10_10 IJ_10_11 IJ_10_12...  
IJ_10_13 IJ_10_14 IJ_10_15 IJ_10_16 IJ_10_17 IJ_10_18 IJ_10_19 IJ_10_20 IJ_10_21  
IJ_10_22 IJ_10_23 IJ_10_24...  
  
IJ_11_1 IJ_11_2 IJ_11_3 IJ_11_4 IJ_11_5 IJ_11_6 IJ_11_7 IJ_11_8 IJ_11_9  
IJ_11_10 IJ_11_11 IJ_11_12...  
IJ_11_13 IJ_11_14 IJ_11_15 IJ_11_16 IJ_11_17 IJ_11_18 IJ_11_19 IJ_11_20 IJ_11_21  
IJ_11_22 IJ_11_23 IJ_11_24...  
  
IJ_12_1 IJ_12_2 IJ_12_3 IJ_12_4 IJ_12_5 IJ_12_6 IJ_12_7 IJ_12_8 IJ_12_9  
IJ_12_10 IJ_12_11 IJ_12_12...  
IJ_12_13 IJ_12_14 IJ_12_15 IJ_12_16 IJ_12_17 IJ_12_18 IJ_12_19 IJ_12_20 IJ_12_21  
IJ_12_22 IJ_12_23 IJ_12_24...  
  
IJ_13_1 IJ_13_2 IJ_13_3 IJ_13_4 IJ_13_5 IJ_13_6 IJ_13_7 IJ_13_8 IJ_13_9  
IJ_13_10 IJ_13_11 IJ_13_12...  
IJ_13_13 IJ_13_14 IJ_13_15 IJ_13_16 IJ_13_17 IJ_13_18 IJ_13_19 IJ_13_20 IJ_13_21  
IJ_13_22 IJ_13_23 IJ_13_24...  
  
IJ_14_1 IJ_14_2 IJ_14_3 IJ_14_4 IJ_14_5 IJ_14_6 IJ_14_7 IJ_14_8 IJ_14_9  
IJ_14_10 IJ_14_11 IJ_14_12...
```

IJ_14_13	IJ_14_14	IJ_14_15	IJ_14_16	IJ_14_17	IJ_14_18	IJ_14_19	IJ_14_20	IJ_14_21
IJ_14_22	IJ_14_23	IJ_14_24...						
IJ_15_1	IJ_15_2	IJ_15_3	IJ_15_4	IJ_15_5	IJ_15_6	IJ_15_7	IJ_15_8	IJ_15_9
IJ_15_10	IJ_15_11	IJ_15_12...						
IJ_15_13	IJ_15_14	IJ_15_15	IJ_15_16	IJ_15_17	IJ_15_18	IJ_15_19	IJ_15_20	IJ_15_21
IJ_15_22	IJ_15_23	IJ_15_24...						
IJ_16_1	IJ_16_2	IJ_16_3	IJ_16_4	IJ_16_5	IJ_16_6	IJ_16_7	IJ_16_8	IJ_16_9
IJ_16_10	IJ_16_11	IJ_16_12...						
IJ_16_13	IJ_16_14	IJ_16_15	IJ_16_16	IJ_16_17	IJ_16_18	IJ_16_19	IJ_16_20	IJ_16_21
IJ_16_22	IJ_16_23	IJ_16_24...						
IJ_17_1	IJ_17_2	IJ_17_3	IJ_17_4	IJ_17_5	IJ_17_6	IJ_17_7	IJ_17_8	IJ_17_9
IJ_17_10	IJ_17_11	IJ_17_12...						
IJ_17_13	IJ_17_14	IJ_17_15	IJ_17_16	IJ_17_17	IJ_17_18	IJ_17_19	IJ_17_20	IJ_17_21
IJ_17_22	IJ_17_23	IJ_17_24...						
IJ_18_1	IJ_18_2	IJ_18_3	IJ_18_4	IJ_18_5	IJ_18_6	IJ_18_7	IJ_18_8	IJ_18_9
IJ_18_10	IJ_18_11	IJ_18_12...						
IJ_18_13	IJ_18_14	IJ_18_15	IJ_18_16	IJ_18_17	IJ_18_18	IJ_18_19	IJ_18_20	IJ_18_21
IJ_18_22	IJ_18_23	IJ_18_24...						
IJ_19_1	IJ_19_2	IJ_19_3	IJ_19_4	IJ_19_5	IJ_19_6	IJ_19_7	IJ_19_8	IJ_19_9
IJ_19_10	IJ_19_11	IJ_19_12...						
IJ_19_13	IJ_19_14	IJ_19_15	IJ_19_16	IJ_19_17	IJ_19_18	IJ_19_19	IJ_19_20	IJ_19_21
IJ_19_22	IJ_19_23	IJ_19_24...						
IJ_20_1	IJ_20_2	IJ_20_3	IJ_20_4	IJ_20_5	IJ_20_6	IJ_20_7	IJ_20_8	IJ_20_9
IJ_20_10	IJ_20_11	IJ_20_12...						
IJ_20_13	IJ_20_14	IJ_20_15	IJ_20_16	IJ_20_17	IJ_20_18	IJ_20_19	IJ_20_20	IJ_20_21
IJ_20_22	IJ_20_23	IJ_20_24...						
IJ_21_1	IJ_21_2	IJ_21_3	IJ_21_4	IJ_21_5	IJ_21_6	IJ_21_7	IJ_21_8	IJ_21_9
IJ_21_10	IJ_21_11	IJ_21_12...						
IJ_21_13	IJ_21_14	IJ_21_15	IJ_21_16	IJ_21_17	IJ_21_18	IJ_21_19	IJ_21_20	IJ_21_21
IJ_21_22	IJ_21_23	IJ_21_24...						
IJ_22_1	IJ_22_2	IJ_22_3	IJ_22_4	IJ_22_5	IJ_22_6	IJ_22_7	IJ_22_8	IJ_22_9
IJ_22_10	IJ_22_11	IJ_22_12...						
IJ_22_13	IJ_22_14	IJ_22_15	IJ_22_16	IJ_22_17	IJ_22_18	IJ_22_19	IJ_22_20	IJ_22_21
IJ_22_22	IJ_22_23	IJ_22_24...						
IJ_23_1	IJ_23_2	IJ_23_3	IJ_23_4	IJ_23_5	IJ_23_6	IJ_23_7	IJ_23_8	IJ_23_9
IJ_23_10	IJ_23_11	IJ_23_12...						
IJ_23_13	IJ_23_14	IJ_23_15	IJ_23_16	IJ_23_17	IJ_23_18	IJ_23_19	IJ_23_20	IJ_23_21
IJ_23_22	IJ_23_23	IJ_23_24...						
IJ_24_1	IJ_24_2	IJ_24_3	IJ_24_4	IJ_24_5	IJ_24_6	IJ_24_7	IJ_24_8	IJ_24_9
IJ_24_10	IJ_24_11	IJ_24_12...						
IJ_24_13	IJ_24_14	IJ_24_15	IJ_24_16	IJ_24_17	IJ_24_18	IJ_24_19	IJ_24_20	IJ_24_21
IJ_24_22	IJ_24_23	IJ_24_24...						

};

```

% Inverse Jacobian Matrix Elements
IJ_1_1 = 1;
IJ_1_2 = 0;
IJ_1_3 = 0;
IJ_1_4=0.333333E0.*(L12+L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)) ...
.^(-1/2).*cos(PHI_C).*cos(THETA_C);
IJ_1_5=0.333333E0.*(L13+L12.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)) ...
.^(-1/2).*cos(PHI_C).*cos(THETA_C);
IJ_1_6 = 0;
IJ_1_7=(-0.333333E0).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(1/2).*cos( ...
THETA_C).*sin(PHI_C);
IJ_1_8=(-0.333333E0).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(1/2).*cos(PHI_C) ...
.*sin(THETA_C);
IJ_1_9 = 0;
IJ_1_10 = 0;
IJ_1_11=(-0.333333E0).*L12.*L13.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1/2) ...
.*cos(PHI_C).*cos(THETA_C).*sin(BETA);
IJ_1_12 = 0;
IJ_1_13 = 0;
IJ_1_14 = 0;
IJ_1_15 = 0;
IJ_1_16 = 0;
IJ_1_17 = 0;
IJ_1_18 = 0;
IJ_1_19 = 0;
IJ_1_20 = 0;
IJ_1_21 = 0;
IJ_1_22 = 0;
IJ_1_23 = 0;
IJ_1_24 = 0;

IJ_2_1 = 1;
IJ_2_2 = 0;
IJ_2_3 = 0;
IJ_2_4=(-1/2).*(2.*L12+2.*L13.*cos(BETA)).*((-0.6666667E0).*L12.^2+0.333333E0.* ...
L13.^2+(-0.333333E0).*L12.*L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.* ...
cos(BETA)).^(-3/2).*cos(PHI_C).*cos(THETA_C)+((-0.133333E1).*L12+( ...
-0.333333E0).*L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^( ...
-1/2).*cos(PHI_C).*cos(THETA_C)+(-0.1666667E0).*(4.*L12.^2+L13.^2+(-4).* ...
L12.*L13.*cos(BETA)).^(1/2).*(2.*L12+2.*L13.*cos(BETA)).*(4.*L12.^2+ ...
L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+ ...
1.E0.*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^( ...
-2)+(-2).*(0.4E1.*L12+1.E0).*L13.*cos(BETA)).*(4.*L12.^2+L13.^2+(-4).* ...
L12.*L13.*cos(BETA)).^(-1).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.* ...
L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)+(8.*L12+( ...
-4).*L13.*cos(BETA)).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-2) ...
.*(0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*(L12.^2+ ...
L13.^2+2.*L12.*L13.*cos(BETA)).^(-1).*(1+(-1)).*(4.*L12.^2+L13.^2+(-4).* ...
L12.*L13.*cos(BETA)).^(-1).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.* ...
L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^( ...
-1/2).*(cos(PHI_C).*sin(PHI_C)+(-0.1E1).*cos(PHI_C).*sin(PHI_C)).*sin( ...
THETA_C))+(-0.1666667E0).*(8.*L12+(-4).*L13.*cos(BETA)).*(4.*L12.^2+ ...
L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1/2).*(1+(-1)).*(4.*L12.^2+L13.^2+( ...
-4).*L12.*L13.*cos(BETA)).^(-1).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.* ...
L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1) ...
.^(1/2).*(cos(PHI_C).*sin(PHI_C)+(-0.1E1).*cos(PHI_C).*sin(PHI_C)).*sin( ...
THETA_C));
IJ_2_5=(-1/2).*(2.*L13+2.*L12.*cos(BETA)).*((-0.6666667E0).*L12.^2+0.333333E0.* ...
L13.^2+(-0.333333E0).*L12.*L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.* ...
cos(BETA)).^(-3/2).*cos(PHI_C).*cos(THETA_C)+(0.6666667E0).*L13+( ...
-0.333333E0).*L12.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^( ...
-1/2).*cos(PHI_C).*cos(THETA_C)+(-0.1666667E0).*(4.*L12.^2+L13.^2+(-4).* ...
L12.*L13.*cos(BETA)).^(1/2).*(2.*L13+2.*L12.*cos(BETA)).*(4.*L12.^2+ ...
L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+ ...
1.E0.*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^( ...
-2)+(-2).*((-0.2E1).*L13+1.E0).*L12.*cos(BETA)).*(4.*L12.^2+L13.^2+(-4).* ...
L12.*L13.*cos(BETA)).^(-1).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.* ...
L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)+(2.*L13+( ...
-4).*L12.*cos(BETA)).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-2) ...
.*(0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*(L12.^2+ ...
L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).*(1+(-1)).*(4.*L12.^2+L13.^2+(-4).* ...
L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).*(1+(-1)).*(4.*L12.^2+L13.^2+(-4).* ...

```

```

L12.*L13.*cos(BETA)).^(-1).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.* ...
L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^( ...
-1/2).*(cos(PHI_C).*sin(PSI_C)+(-0.1E1).*cos(PSI_C).*sin(PHI_C).*sin( ...
THETA_C))+(-0.166667E0).*(2.*L13+(-4).*L12.*cos(BETA)).*(4.*L12.^2+ ...
L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1/2).*(1+(-1).*(4.*L12.^2+L13.^2+ ...
-4).*L12.*L13.*cos(BETA)).^(-1).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.* ...
L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1) ...
.^(1/2).*(cos(PHI_C).*sin(PSI_C)+(-0.1E1).*cos(PHI_C).*sin(PHI_C).*sin( ...
THETA_C));
IJ_2_6 = 0;
IJ_2_7=(-1).*((-0.666667E0).*L12.^2+0.333333E0.*L13.^2+(-0.333333E0).*L12.* ...
L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1/2).*cos( ...
THETA_C).*sin(PHI_C)+(-0.333333E0).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.* ...
cos(BETA)).^(1/2).*(1+(-1).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)) ...
.^(-1).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*( ...
L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^(1/2).*(cos(PHI_C).*cos( ...
PSI_C)+0.1E1.*sin(PHI_C).*sin(PHI_C).*sin(THETA_C));
IJ_2_8=0.333333E0.*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(1/2).*(1+(-1) ...
.*4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*(0.2E1.*L12.^2+ ...
-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.* ...
L13.*cos(BETA)).^(-1)).^(1/2).*cos(PHI_C).*cos(THETA_C).*sin(PHI_C)+(-1) ...
.*((-0.666667E0).*L12.^2+0.333333E0.*L13.^2+(-0.333333E0).*L12.*L13.* ...
cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1/2).*cos(PHI_C).* ...
sin(THETA_C);
IJ_2_9=(-0.333333E0).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(1/2).*(1+( ...
-1).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*(0.2E1.*L12.^2+ ...
-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.* ...
L13.*cos(BETA)).^(-1)).^(1/2).*((-1).*sin(PHI_C).*sin(PHI_C)+(-0.1E1).* ...
cos(PHI_C).*cos(PHI_C).*sin(THETA_C));
IJ_2_10 = 0;
IJ_2_11=L12.*L13.*((-0.666667E0).*L12.^2+0.333333E0.*L13.^2+(-0.333333E0).*L12.* ...
L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-3/2).*cos( ...
PSI_C).*cos(THETA_C).*sin(BETA)+0.333333E0.*L12.*L13.*(L12.^2+L13.^2+2.* ...
L12.*L13.*cos(BETA)).^(-1/2).*cos(PHI_C).*cos(THETA_C).*sin(BETA)+( ...
-0.666667E0).*L12.*L13.*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1) ...
-1/2).*(1+(-1).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*( ...
0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*(L12.^2+ ...
L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^(1/2).*sin(BETA).*(cos(PHI_C).* ...
sin(PHI_C)+(-0.1E1).*cos(PHI_C).*sin(PHI_C).*sin(THETA_C))+(-0.166667E0) ...
.*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(1/2).*(1+(-1).*(4.* ...
L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*(0.2E1.*L12.^2+(-1.E0).* ...
L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos( ...
BETA)).^(-1)).^(-1/2).*((-2).*L12.*L13.*(4.*L12.^2+L13.^2+(-4).*L12.* ...
L13.*cos(BETA)).^(-1).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.* ...
cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-2).*sin(BETA)+ ...
0.2E1.*L12.*L13.*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*( ...
0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).*(L12.^2+ ...
L13.^2+2.*L12.*L13.*cos(BETA)).^(-1).*sin(BETA)+4.*L12.*L13.*(4.*L12.^2+ ...
L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-2).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+ ...
1.E0.*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^( ...
-1).*sin(BETA).*(cos(PHI_C).*sin(PHI_C)+(-0.1E1).*cos(PHI_C).*sin( ...
PHI_C).*sin(THETA_C));
IJ_2_12 = 0;
IJ_2_13 = 0;
IJ_2_14 = 0;
IJ_2_15 = 0;
IJ_2_16 = 0;
IJ_2_17 = 0;
IJ_2_18 = 0;
IJ_2_19 = 0;
IJ_2_20 = 0;
IJ_2_21 = 0;
IJ_2_22 = 0;
IJ_2_23 = 0;
IJ_2_24 = 0;

IJ_3_1 = 1;
IJ_3_2 = 0;
IJ_3_3 = 0;
IJ_3_4=(-1/2).*(2.*L12+2.*L13.*cos(BETA)).*(0.333333E0.*L12.^2+(-0.666667E0).* ...
L13.^2+(-0.333333E0).*L12.*L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.* ...

```

```

cos(BETA)).^(-3/2).*cos(PHI_C).*cos(THETA_C)+(0.666667E0.*L12+( ...
-0.333333E0).*L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^( ...
-1/2).*cos(PHI_C).*cos(THETA_C)+0.166667E0.*(L12.^2+4.*L13.^2+(-4).* ...
L12.*L13.*cos(BETA)).^(1/2).*(2.*L12+2.*L13.*cos(BETA)).*(L12.^2+4.* ...
L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*(0.1E1.*L12.^2+(-0.2E1).* ...
L13.^2+(-0.1E1)).*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.* ...
cos(BETA)).^(-2)+(-2).*(0.2E1.*L12+(-0.1E1)).*L13.*cos(BETA)).*(L12.^2+ ...
4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*(0.1E1.*L12.^2+(-0.2E1).* ...
L13.^2+(-0.1E1)).*L12.*L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos( ...
BETA)).^(-1)+(2.*L12+(-4).*L13.*cos(BETA)).*(L12.^2+4.*L13.^2+(-4).* ...
L12.*L13.*cos(BETA)).^(-2).*(0.1E1.*L12.^2+(-0.2E1)).*L13.^2+(-0.1E1).* ...
L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).* ...
(1+(-1).*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*(0.1E1.* ...
L12.^2+(-0.2E1)).*L13.^2+(-0.1E1)).*L12.*L13.*cos(BETA)).^2.*(L12.^2+ ...
L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^(1/2).*(cos(PHI_C)).*sin(PHI_C)+ ...
-0.1E1).*cos(PHI_C)).*sin(PHI_C)).*sin(THETA_C));
IJ_3_5=(-1/2).*(2.*L13+2.*L12.*cos(BETA)).*(0.333333E0.*L12.^2+(-0.666667E0).* ...
L13.^2+(-0.333333E0)).*L12.*L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.* ...
cos(BETA)).^(-3/2).*cos(PHI_C)).*cos(THETA_C)+((-0.133333E1).*L13+( ...
-0.333333E0).*L12.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^( ...
-1/2).*cos(PHI_C)).*cos(THETA_C)+0.166667E0.*(L12.^2+4.*L13.^2+(-4).* ...
L12.*L13.*cos(BETA)).^(1/2).*(2.*L13+2.*L12.*cos(BETA)).*(L12.^2+4.* ...
L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*(0.1E1.*L12.^2+(-0.2E1).* ...
L13.^2+(-0.1E1)).*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.* ...
cos(BETA)).^(-2)+(-2).*((-0.4E1).*L13+(-0.1E1)).*L12.*cos(BETA)).*( ...
L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*(0.1E1.*L12.^2+ ...
-0.2E1)).*L13.^2+(-0.1E1)).*L12.*L13.*cos(BETA)).*(L12.^2+L13.^2+2.* ...
L13.*cos(BETA)).^(-1)+(8.*L13+(-4)).*L12.*cos(BETA)).*(L12.^2+4.*L13.^2+ ...
-4).*L12.*L13.*cos(BETA)).^(-2).*(0.1E1.*L12.^2+(-0.2E1)).*L13.^2+ ...
-0.1E1)).*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)) ...
.^(-1)).*(1+(-1).*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1)).*( ...
0.1E1.*L12.^2+(-0.2E1)).*L13.^2+(-0.1E1)).*L12.*L13.*cos(BETA)).^2.*( ...
L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^(1/2).*(cos(PHI_C)).*sin( ...
PSI_C)+(-0.1E1)).*cos(PHI_C)).*sin(PHI_C)).*sin(THETA_C))+0.166667E0.*(8.* ...
L13+(-4)).*L12.*cos(BETA)).*(L12.^2+4.*L13.^2+(-4)).*L12.*L13.*cos(BETA)) ...
.^(-1/2).*(1+(-1).*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1)).*( ...
0.1E1.*L12.^2+(-0.2E1)).*L13.^2+(-0.1E1)).*L12.*L13.*cos(BETA)).^2.*( ...
L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^(1/2).*(cos(PHI_C)).*sin( ...
PSI_C)+(-0.1E1)).*cos(PHI_C)).*sin(PHI_C)).*sin(THETA_C));
IJ_3_6 = 0;
IJ_3_7=(-1).*(0.333333E0.*L12.^2+(-0.666667E0)).*L13.^2+(-0.333333E0)).*L12.* ...
L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1/2)).*cos( ...
THETA_C)).*sin(PHI_C))+0.333333E0.*(L12.^2+4.*L13.^2+(-4)).*L12.*L13.*cos( ...
BETA)).^(1/2).*(1+(-1).*(L12.^2+4.*L13.^2+(-4)).*L12.*L13.*cos(BETA)).^( ...
-1)).*(0.1E1.*L12.^2+(-0.2E1)).*L13.^2+(-0.1E1)).*L12.*L13.*cos(BETA)).^2.* ...
(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^(1/2).*(cos(PHI_C)).*cos( ...
PSI_C)+0.1E1.*sin(PHI_C)).*sin(PHI_C)).*sin(THETA_C));
IJ_3_8=(-0.333333E0)).*(L12.^2+4.*L13.^2+(-4)).*L12.*L13.*cos(BETA)).^(1/2).*(1+( ...
-1)).*(L12.^2+4.*L13.^2+(-4)).*L12.*L13.*cos(BETA)).^(-1).*(0.1E1.*L12.^2+ ...
(-0.2E1)).*L13.^2+(-0.1E1)).*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.* ...
L12.*L13.*cos(BETA)).^(-1)).^(1/2).*(cos(PHI_C)).*cos(THETA_C)).*sin(PHI_C) ...
+(-1)).*(0.333333E0).*L12.^2+(-0.666667E0)).*L13.^2+(-0.333333E0)).*L12.* ...
L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1/2)).*cos( ...
PSI_C)).*sin(THETA_C));
IJ_3_9=0.333333E0).*(L12.^2+4.*L13.^2+(-4)).*L12.*L13.*cos(BETA)).^(1/2).*(1+(-1) ...
).*(L12.^2+4.*L13.^2+(-4)).*L12.*L13.*cos(BETA)).^(-1).*(0.1E1.*L12.^2+ ...
-0.2E1)).*L13.^2+(-0.1E1)).*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.* ...
L12.*L13.*cos(BETA)).^(-1)).^(1/2).*((-1)).*sin(PHI_C)).*sin(PSI_C)+ ...
-0.1E1)).*cos(PHI_C)).*cos(PSI_C)).*sin(THETA_C));
IJ_3_10 = 0;
IJ_3_11=L12.*L13.*(0.333333E0.*L12.^2+(-0.666667E0)).*L13.^2+(-0.333333E0)).*L12.* ...
L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-3/2)).*cos( ...
PSI_C)).*cos(THETA_C)).*sin(BETA))+0.333333E0.*L12.*L13.*(L12.^2+L13.^2+2.* ...
L12.*L13.*cos(BETA)).^(-1/2)).*cos(PSI_C)).*cos(THETA_C)).*sin(BETA)+ ...
0.666667E0.*L12.*L13.*(L12.^2+4.*L13.^2+(-4)).*L12.*L13.*cos(BETA)).^( ...
-1/2).*(1+(-1).*(L12.^2+4.*L13.^2+(-4)).*L12.*L13.*cos(BETA)).^(-1)).*( ...

```



```

0.1E1.*L12.^2+(-0.2E1).*L13.^2+(-0.1E1).*L12.*L13.*cos(BETA)).^2.*( ...
L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^(1/2).*sin(BETA).*(cos( ...
PHI_C).*sin(PSI_C)+(-0.1E1).*cos(PSI_C).*sin(PHI_C).*sin(THETA_C))+ ...
0.166667E0.*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^(1/2).*(1+(-1) ...
.*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1)).*(0.1E1.*L12.^2+ ...
-0.2E1).*L13.^2+(-0.1E1).*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.* ...
L12.*L13.*cos(BETA)).^(-1)).^(-1/2)).*((-2).*L12.*L13.*(L12.^2+4.*L13.^2+ ...
(-4).*L12.*L13.*cos(BETA)).^(-1)).*(0.1E1.*L12.^2+(-0.2E1).*L13.^2+ ...
-0.1E1).*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)) ...
.^(2).*sin(BETA)+(-0.2E1).*L12.*L13.*(L12.^2+4.*L13.^2+(-4).*L12.*L13.* ...
cos(BETA)).^(-1)).*(0.1E1.*L12.^2+(-0.2E1).*L13.^2+(-0.1E1).*L12.*L13.* ...
cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).sin(BETA)+4.* ...
L12.*L13.*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-2)).*(0.1E1.* ...
L12.^2+(-0.2E1).*L13.^2+(-0.1E1).*L12.*L13.*cos(BETA)).^2.*(L12.^2+ ...
L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).sin(BETA)).*(cos(PHI_C).*sin( ...
PSI_C)+(-0.1E1).*cos(PSI_C).*sin(PHI_C).*sin(THETA_C));
IJ_3_12 = 0;
IJ_3_13 = 0;
IJ_3_14 = 0;
IJ_3_15 = 0;
IJ_3_16 = 0;
IJ_3_17 = 0;
IJ_3_18 = 0;
IJ_3_19 = 0;
IJ_3_20 = 0;
IJ_3_21 = 0;
IJ_3_22 = 0;
IJ_3_23 = 0;
IJ_3_24 = 0;

IJ_4_1 = 1;
IJ_4_2 = 0;
IJ_4_3 = 0;
IJ_4_4 = 0;
IJ_4_5 = 0;
IJ_4_6=cos(XI).*(sin(PHI_C).*sin(PSI_C)+cos(PHI_C).*cos(PSI_C).*sin(THETA_C)) ...
+(cos(ALPHA).*cos(PSI_C).*cos(THETA_C)+sin(ALPHA).*((-1).*cos(PHI_C).* ...
sin(PSI_C)+cos(PSI_C).*sin(PHI_C).*sin(THETA_C))).*sin(XI);
IJ_4_7=LB4.*cos(XI).*(cos(PSI_C).*sin(PHI_C)+(-1).*cos(PHI_C).*sin(PSI_C).* ...
sin(THETA_C))+LB4.*((-1).*cos(ALPHA).*cos(THETA_C).*sin(PSI_C)+sin( ...
ALPHA).*((-1).*cos(PHI_C).*cos(PSI_C)+(-1).*sin(PHI_C).*sin(PSI_C).*sin( ...
THETA_C))).*sin(XI);
IJ_4_8=LB4.*cos(PHI_C).*cos(PSI_C).*cos(THETA_C).*cos(XI)+LB4.*(cos(PSI_C).* ...
cos(THETA_C)).*sin(ALPHA).*sin(PHI_C)+(-1).*cos(ALPHA).*cos(PSI_C).*sin( ...
THETA_C)).*sin(XI);
IJ_4_9=LB4.*cos(XI).*(cos(PHI_C).*sin(PSI_C)+(-1).*cos(PSI_C).*sin(PHI_C).* ...
sin(THETA_C))+LB4.*sin(ALPHA).*(sin(PHI_C).*sin(PSI_C)+cos(PHI_C).*cos( ...
PSI_C)).*sin(THETA_C)).*sin(XI);
IJ_4_10=LB4.*((-1).*cos(PSI_C).*cos(THETA_C).*sin(ALPHA)+cos(ALPHA).*((-1).*cos( ...
PHI_C).*sin(PSI_C)+cos(PSI_C).*sin(PHI_C).*sin(THETA_C))).*sin(XI);
IJ_4_11 = 0;
IJ_4_12=LB4.*cos(XI).*(cos(ALPHA).*cos(PSI_C).*cos(THETA_C)+sin(ALPHA).*((-1) ...
.*cos(PHI_C).*sin(PSI_C)+cos(PSI_C).*sin(PHI_C).*sin(THETA_C)))+(-1).* ...
LB4.*(sin(PHI_C).*sin(PSI_C)+cos(PHI_C).*cos(PSI_C).*sin(THETA_C)).*sin( ...
XI);
IJ_4_13 = 0;
IJ_4_14 = 0;
IJ_4_15 = 0;
IJ_4_16 = 0;
IJ_4_17 = 0;
IJ_4_18 = 0;
IJ_4_19 = 0;
IJ_4_20 = 0;
IJ_4_21 = 0;
IJ_4_22 = 0;
IJ_4_23 = 0;
IJ_4_24 = 0;

IJ_5_1 = 0;
IJ_5_2 = 1;
IJ_5_3 = 0;

```

```

IJ_5_4=0.333333E0.*(L12+L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)) ...
.^(-1/2).*cos(THETA_C).*sin(PSI_C);
IJ_5_5=0.333333E0.*(L13+L12.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)) ...
.^(-1/2).*cos(THETA_C).*sin(PSI_C);
IJ_5_6 = 0;
IJ_5_7=0.333333E0.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(1/2).*cos(PSI_C).* ...
cos(THETA_C);
IJ_5_8=(-0.333333E0).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(1/2).*sin(PSI_C) ...
.*sin(THETA_C);
IJ_5_9 = 0;
IJ_5_10 = 0;
IJ_5_11=(-0.333333E0).*L12.*L13.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1/2) ...
.*cos(THETA_C).*sin(BETA).*sin(PSI_C);
IJ_5_12 = 0;
IJ_5_13 = 0;
IJ_5_14 = 0;
IJ_5_15 = 0;
IJ_5_16 = 0;
IJ_5_17 = 0;
IJ_5_18 = 0;
IJ_5_19 = 0;
IJ_5_20 = 0;
IJ_5_21 = 0;
IJ_5_22 = 0;
IJ_5_23 = 0;
IJ_5_24 = 0;

IJ_6_1 = 0;
IJ_6_2 = 1;
IJ_6_3 = 0;
IJ_6_4=(-1/2).*(2.*L12+2.*L13.*cos(BETA)).*((-0.666667E0).*L12.^2+0.333333E0.* ...
L13.^2+(-0.333333E0).*L12.*L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.* ...
cos(BETA)).^(-3/2).*cos(THETA_C).*sin(PSI_C)+((-0.133333E1).*L12+( ...
-0.333333E0).*L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^( ...
-1/2).*cos(THETA_C).*sin(PSI_C)+0.166667E0.*(4.*L12.^2+L13.^2+(-4).* ...
L12.*L13.*cos(BETA)).^(1/2).*(2.*L12+2.*L13.*cos(BETA)).*(4.*L12.^2+ ...
L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+ ...
1.E0.*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^( ...
-2)+(-2).*(0.4E1.*L12+1.E0).*L13.*cos(BETA)).*(4.*L12.^2+L13.^2+(-4).* ...
L2.*L13.*cos(BETA)).^(-1).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.* ...
L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)+(8.*L12+( ...
-4).*L13.*cos(BETA)).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-2) ...
.*(0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*(L12.^2+ ...
L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).*(1+(-1)).*(4.*L12.^2+L13.^2+(-4).* ...
L12.*L13.*cos(BETA)).^(-1).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.* ...
L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^( ...
-1/2).*(cos(PHI_C).*cos(PSI_C)+sin(PHI_C).*sin(PSI_C).*sin(THETA_C))+ ...
0.166667E0.*(8.*L12+(-4).*L13.*cos(BETA)).*(4.*L12.^2+L13.^2+(-4).*L12.* ...
L13.*cos(BETA)).^(-1/2).*(1+(-1)).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos( ...
BETA)).^(-1).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)) ...
.^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^(1/2).*(cos(PHI_C).* ...
cos(PSI_C)+sin(PHI_C).*sin(PSI_C)).*sin(THETA_C));
IJ_6_5=(-1/2).*(2.*L13+2.*L12.*cos(BETA)).*((-0.666667E0).*L12.^2+0.333333E0.* ...
L13.^2+(-0.333333E0).*L12.*L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.* ...
cos(BETA)).^(-3/2).*cos(THETA_C).*sin(PSI_C)+(0.666667E0).*L13+( ...
-0.333333E0).*L12.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^( ...
-1/2).*cos(THETA_C).*sin(PSI_C)+0.166667E0.*(4.*L12.^2+L13.^2+(-4).* ...
L12.*L13.*cos(BETA)).^(1/2).*(2.*L13+2.*L12.*cos(BETA)).*(4.*L12.^2+ ...
L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+ ...
1.E0.*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^( ...
-2)+(-2).*(0.2E1).*L13+1.E0.*L12.*cos(BETA)).*(4.*L12.^2+L13.^2+(-4).* ...
L12.*L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)+(2.*L13+( ...
-4).*L12.*cos(BETA)).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-2) ...
.*(0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*(L12.^2+ ...
L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).*(1+(-1)).*(4.*L12.^2+L13.^2+(-4).* ...
L12.*L13.*cos(BETA)).^(-1).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.* ...
L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^( ...
-1/2).*(cos(PHI_C).*cos(PSI_C)+sin(PHI_C).*sin(PSI_C)).*sin(THETA_C))+ ...
0.166667E0.*(2.*L13+(-4).*L12.*cos(BETA)).*(4.*L12.^2+L13.^2+(-4).*L12.* ...
L13.*cos(BETA)).^(-1/2).*(1+(-1)).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos( ...

```

```

BETA)).^(-1).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)) ...
.^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^(1/2).*(cos(PHI_C).* ...
cos(PHI_C)+sin(PHI_C).*sin(PHI_C).*sin(THETA_C));
IJ_6_6 = 0;
IJ_6_7=(((-0.666667E0).*L12.^2+0.333333E0.*L13.^2+(-0.333333E0).*L12.*L13.*cos( ...
BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1/2).*cos(PHI_C).*cos( ...
THETA_C)+0.333333E0.*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(1/2) ...
.*(1+(-1).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*(0.2E1.* ...
L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.* ...
L12.*L13.*cos(BETA)).^(-1)).^(1/2).*((-1).*cos(PHI_C).*sin(PHI_C)+cos( ...
PSI_C).*sin(PHI_C).*sin(THETA_C));
IJ_6_8=0.333333E0.*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(1/2).*(1+(-1) ...
.*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*(0.2E1.*L12.^2+ ...
-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.* ...
L13.*cos(BETA)).^(-1)).^(1/2).*cos(THETA_C).*sin(PHI_C).*sin(PSI_C)+(-1) ...
.*((-0.666667E0).*L12.^2+0.333333E0.*L13.^2+(-0.333333E0).*L12.*L13.* ...
cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1/2).*sin(PSI_C).* ...
sin(THETA_C);
IJ_6_9=0.333333E0.*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(1/2).*(1+(-1) ...
.*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*(0.2E1.*L12.^2+ ...
-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.* ...
L13.*cos(BETA)).^(-1)).^(1/2).*((-1).*cos(PSI_C).*sin(PHI_C)+cos(PHI_C) ...
.*sin(PSI_C).*sin(THETA_C));
IJ_6_10 = 0;
IJ_6_11=L12.*L13.*((-0.666667E0).*L12.^2+0.333333E0.*L13.^2+(-0.333333E0).*L12.* ...
L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-3/2).*cos( ...
THETA_C).*sin(BETA).*sin(PSI_C)+0.333333E0.*L12.*L13.*(L12.^2+L13.^2+2.* ...
L12.*L13.*cos(BETA)).^(-1/2).*cos(THETA_C).*sin(BETA).*sin(PSI_C)+ ...
0.666667E0.*L12.*L13.*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^( ...
-1/2).*(1+(-1).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*( ...
0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*(L12.^2+ ...
L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^(1/2).*sin(BETA).*(cos(PHI_C).* ...
cos(PSI_C)+sin(PHI_C).*sin(PSI_C).*sin(THETA_C))+0.166667E0.*(4.*L12.^2+ ...
L13.^2+(-4).*L12.*L13.*cos(BETA)).^(1/2).*(1+(-1).*(4.*L12.^2+L13.^2+ ...
-4).*L12.*L13.*cos(BETA)).^(-1).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.* ...
L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1) ...
.^(-1/2).*((-2).*L12.*L13.*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)) ...
.^(-1).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*( ...
L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-2).*sin(BETA)+0.2E1.*L12.*L13.* ...
(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*(0.2E1.*L12.^2+ ...
-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.* ...
cos(BETA)).^(-1).*sin(BETA)+4.*L12.*L13.*(4.*L12.^2+L13.^2+(-4).*L12.* ...
L13.*cos(BETA)).^(-2).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.* ...
cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1).*sin(BETA) ...
.*(cos(PHI_C).*cos(PSI_C)+sin(PHI_C).*sin(PSI_C).*sin(THETA_C));
IJ_6_12 = 0;
IJ_6_13 = 0;
IJ_6_14 = 0;
IJ_6_15 = 0;
IJ_6_16 = 0;
IJ_6_17 = 0;
IJ_6_18 = 0;
IJ_6_19 = 0;
IJ_6_20 = 0;
IJ_6_21 = 0;
IJ_6_22 = 0;
IJ_6_23 = 0;
IJ_6_24 = 0;

IJ_7_1 = 0;
IJ_7_2 = 1;
IJ_7_3 = 0;
IJ_7_4=(-1/2).*(2.*L12+2.*L13.*cos(BETA)).*(0.333333E0.*L12.^2+(-0.666667E0).* ...
L13.^2+(-0.333333E0).*L12.*L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.* ...
cos(BETA)).^(-3/2).*cos(THETA_C).*sin(PSI_C)+(0.666667E0.*L12+ ...
-0.333333E0).*L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^( ...
-1/2).*cos(THETA_C).*sin(PSI_C)+(-0.166667E0).*(L12.^2+4.*L13.^2+(-4).* ...
L12.*L13.*cos(BETA)).^(1/2).*((2.*L12+2.*L13.*cos(BETA)).*(L12.^2+4.* ...
L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*(0.1E1.*L12.^2+(-0.2E1).* ...
L13.^2+(-0.1E1).*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.* ...
cos(BETA)).^(-2)+(-2).*(0.2E1.*L12+(-0.1E1).*L13.*cos(BETA)).*(L12.^2+ ...

```

$$4 * L13.^2+(-4) * L12 * L13 * \cos(\text{BETA}) .^(-1) * (0.1E1 * L12.^2+(-0.2E1) * \dots$$

$$L13.^2+(-0.1E1) * L12 * L13 * \cos(\text{BETA})) * (L12.^2+L13.^2+2 * L12 * L13 * \cos(\dots$$

$$\text{BETA})) .^(-1) + (2 * L12+(-4) * L13 * \cos(\text{BETA})) * (L12.^2+4 * L13.^2+(-4) * \dots$$

$$L12 * L13 * \cos(\text{BETA})) .^(-2) * (0.1E1 * L12.^2+(-0.2E1) * L13.^2+(-0.1E1) * \dots$$

$$L12 * L13 * \cos(\text{BETA})) .^2 * (L12.^2+L13.^2+2 * L12 * L13 * \cos(\text{BETA})) .^(-1)) * \dots$$

$$(1+(-1) * (L12.^2+4 * L13.^2+(-4) * L12 * L13 * \cos(\text{BETA})) .^(-1) * (0.1E1 * \dots$$

$$L12.^2+(-0.2E1) * L13.^2+(-0.1E1) * L12 * L13 * \cos(\text{BETA})) .^2 * (L12.^2+ \dots$$

$$L13.^2+2 * L12 * L13 * \cos(\text{BETA})) .^(-1)) .^(-1/2) * (\cos(\text{PHI}_C) * \cos(\text{PSI}_C) + \dots$$

$$\sin(\text{PHI}_C) * \sin(\text{PSI}_C) * \sin(\text{THETA}_C)) + (-0.166667E0) * (2 * L12+(-4) * L13 * \dots$$

$$\cos(\text{BETA})) * (L12.^2+4 * L13.^2+(-4) * L12 * L13 * \cos(\text{BETA})) .^(-1/2) * (1+(\dots$$

$$-1) * (L12.^2+4 * L13.^2+(-4) * L12 * L13 * \cos(\text{BETA})) .^(-1) * (0.1E1 * L12.^2+ \dots$$

$$(-0.2E1) * L13.^2+(-0.1E1) * L12 * L13 * \cos(\text{BETA})) .^2 * (L12.^2+L13.^2+2 * \dots$$

$$L12 * L13 * \cos(\text{BETA})) .^(-1)) .^ (1/2) * (\cos(\text{PHI}_C) * \cos(\text{PSI}_C) + \sin(\text{PHI}_C) * \dots$$

$$\sin(\text{PSI}_C) * \sin(\text{THETA}_C));$$

$$\text{IJ}_7_5 = (-1/2) * (2 * L13+2 * L12 * \cos(\text{BETA})) * (0.333333E0 * L12.^2+(-0.666667E0) * \dots$$

$$L13.^2+(-0.333333E0) * L12 * L13 * \cos(\text{BETA})) * (L12.^2+L13.^2+2 * L12 * L13 * \dots$$

$$\cos(\text{BETA})) .^(-3/2) * \cos(\text{THETA}_C) * \sin(\text{PSI}_C) + ((-0.133333E1) * L13+ (\dots$$

$$-0.333333E0) * L12 * \cos(\text{BETA})) * (L12.^2+L13.^2+2 * L12 * L13 * \cos(\text{BETA})) .^ (\dots$$

$$-1/2) * \cos(\text{THETA}_C) * \sin(\text{PSI}_C) + (-0.166667E0) * (L12.^2+4 * L13.^2+(-4) * \dots$$

$$L12 * L13 * \cos(\text{BETA})) .^ (1/2) * ((2 * L13+2 * L12 * \cos(\text{BETA})) * (L12.^2+4 * \dots$$

$$L13.^2+(-4) * L12 * L13 * \cos(\text{BETA})) .^(-1) * (0.1E1 * L12.^2+(-0.2E1) * \dots$$

$$L13.^2+(-0.1E1) * L12 * L13 * \cos(\text{BETA})) .^2 * (L12.^2+L13.^2+2 * L12 * L13 * \dots$$

$$\cos(\text{BETA})) .^(-2) + (-2) * ((-0.4E1) * L13+(-0.1E1) * L12 * \cos(\text{BETA})) * (\dots$$

$$L12.^2+4 * L13.^2+(-4) * L12 * L13 * \cos(\text{BETA})) .^(-1) * (0.1E1 * L12.^2+ (\dots$$

$$-0.2E1) * L13.^2+(-0.1E1) * L12 * L13 * \cos(\text{BETA})) * (L12.^2+L13.^2+2 * L12 * \dots$$

$$L13 * \cos(\text{BETA})) .^(-1) + (8 * L13+(-4) * L12 * \cos(\text{BETA})) * (L12.^2+4 * L13.^2+ (\dots$$

$$-4) * L12 * L13 * \cos(\text{BETA})) .^(-2) * (0.1E1 * L12.^2+(-0.2E1) * L13.^2+ (\dots$$

$$-0.1E1) * L12 * L13 * \cos(\text{BETA})) .^2 * (L12.^2+L13.^2+2 * L12 * L13 * \cos(\text{BETA})) \dots$$

$$. ^(-1)) * (1+(-1) * (L12.^2+4 * L13.^2+(-4) * L12 * L13 * \cos(\text{BETA})) .^(-1) * (\dots$$

$$0.1E1 * L12.^2+(-0.2E1) * L13.^2+(-0.1E1) * L12 * L13 * \cos(\text{BETA})) .^2 * (\dots$$

$$L12.^2+L13.^2+2 * L12 * L13 * \cos(\text{BETA})) .^(-1)) .^(-1/2) * (\cos(\text{PHI}_C) * \cos(\dots$$

$$\text{PSI}_C) + \sin(\text{PHI}_C) * \sin(\text{PSI}_C) * \sin(\text{THETA}_C)) + (-0.166667E0) * (8 * L13+(-4) \dots$$

$$* L12 * \cos(\text{BETA})) * (L12.^2+4 * L13.^2+(-4) * L12 * L13 * \cos(\text{BETA})) .^(-1/2) \dots$$

$$* (1+(-1) * (L12.^2+4 * L13.^2+(-4) * L12 * L13 * \cos(\text{BETA})) .^(-1) * (0.1E1 * \dots$$

$$L12.^2+(-0.2E1) * L13.^2+(-0.1E1) * L12 * L13 * \cos(\text{BETA})) .^2 * (L12.^2+ \dots$$

$$L13.^2+2 * L12 * L13 * \cos(\text{BETA})) .^(-1)) .^ (1/2) * (\cos(\text{PHI}_C) * \cos(\text{PSI}_C) + \dots$$

$$\sin(\text{PHI}_C) * \sin(\text{PSI}_C) * \sin(\text{THETA}_C));$$

$$\text{IJ}_7_6 = 0;$$

$$\text{IJ}_7_7 = (0.333333E0 * L12.^2+(-0.666667E0) * L13.^2+(-0.333333E0) * L12 * L13 * \cos(\dots$$

$$\text{BETA})) * (L12.^2+L13.^2+2 * L12 * L13 * \cos(\text{BETA})) .^(-1/2) * \cos(\text{PSI}_C) * \cos(\dots$$

$$\text{THETA}_C) + (-0.333333E0) * (L12.^2+4 * L13.^2+(-4) * L12 * L13 * \cos(\text{BETA})) .^ (\dots$$

$$1/2) * (1+(-1) * (L12.^2+4 * L13.^2+(-4) * L12 * L13 * \cos(\text{BETA})) .^(-1) * (\dots$$

$$0.1E1 * L12.^2+(-0.2E1) * L13.^2+(-0.1E1) * L12 * L13 * \cos(\text{BETA})) .^2 * (\dots$$

$$L12.^2+L13.^2+2 * L12 * L13 * \cos(\text{BETA})) .^(-1)) .^ (1/2) * ((-1) * \cos(\text{PHI}_C) * \dots$$

$$\sin(\text{PSI}_C) + \cos(\text{PSI}_C) * \sin(\text{PHI}_C) * \sin(\text{THETA}_C));$$

$$\text{IJ}_7_8 = (-0.333333E0) * (L12.^2+4 * L13.^2+(-4) * L12 * L13 * \cos(\text{BETA})) .^ (1/2) * (1+ (\dots$$

$$-1) * (L12.^2+4 * L13.^2+(-4) * L12 * L13 * \cos(\text{BETA})) .^(-1) * (0.1E1 * L12.^2+ \dots$$

$$(-0.2E1) * L13.^2+(-0.1E1) * L12 * L13 * \cos(\text{BETA})) .^2 * (L12.^2+L13.^2+2 * \dots$$

$$L12 * L13 * \cos(\text{BETA})) .^(-1)) .^ (1/2) * \cos(\text{THETA}_C) * \sin(\text{PHI}_C) * \sin(\text{PSI}_C) \dots$$

$$+ (-1) * (0.333333E0 * L12.^2+(-0.666667E0) * L13.^2+(-0.333333E0) * L12 * \dots$$

$$L13 * \cos(\text{BETA})) * (L12.^2+L13.^2+2 * L12 * L13 * \cos(\text{BETA})) .^(-1/2) * \sin(\dots$$

$$\text{PSI}_C) * \sin(\text{THETA}_C);$$

$$\text{IJ}_7_9 = (-0.333333E0) * (L12.^2+4 * L13.^2+(-4) * L12 * L13 * \cos(\text{BETA})) .^ (1/2) * (1+ (\dots$$

$$-1) * (L12.^2+4 * L13.^2+(-4) * L12 * L13 * \cos(\text{BETA})) .^(-1) * (0.1E1 * L12.^2+ \dots$$

$$(-0.2E1) * L13.^2+(-0.1E1) * L12 * L13 * \cos(\text{BETA})) .^2 * (L12.^2+L13.^2+2 * \dots$$

$$L12 * L13 * \cos(\text{BETA})) .^(-1)) .^ (1/2) * ((-1) * \cos(\text{PSI}_C) * \sin(\text{PHI}_C) + \cos(\dots$$

$$\text{PHI}_C) * \sin(\text{PSI}_C) * \sin(\text{THETA}_C));$$

$$\text{IJ}_7_{10} = 0;$$

$$\text{IJ}_7_{11} = L12 * L13 * (0.333333E0 * L12.^2+(-0.666667E0) * L13.^2+(-0.333333E0) * L12 * \dots$$

$$L13 * \cos(\text{BETA})) * (L12.^2+L13.^2+2 * L12 * L13 * \cos(\text{BETA})) .^(-3/2) * \cos(\dots$$

$$\text{THETA}_C) * \sin(\text{BETA}) * \sin(\text{PSI}_C) + 0.333333E0 * L12 * L13 * (L12.^2+L13.^2+2 * \dots$$

$$L12 * L13 * \cos(\text{BETA})) .^(-1/2) * \cos(\text{THETA}_C) * \sin(\text{BETA}) * \sin(\text{PSI}_C) + (\dots$$

$$-0.666667E0) * L12 * L13 * (L12.^2+4 * L13.^2+(-4) * L12 * L13 * \cos(\text{BETA})) .^ (\dots$$

$$-1/2) * (1+(-1) * (L12.^2+4 * L13.^2+(-4) * L12 * L13 * \cos(\text{BETA})) .^(-1) * (\dots$$

$$0.1E1 * L12.^2+(-0.2E1) * L13.^2+(-0.1E1) * L12 * L13 * \cos(\text{BETA})) .^2 * (\dots$$

$$L12.^2+L13.^2+2 * L12 * L13 * \cos(\text{BETA})) .^(-1)) .^ (1/2) * \sin(\text{BETA}) * (\cos(\dots$$

$$\text{PHI}_C) * \cos(\text{PSI}_C) + \sin(\text{PHI}_C) * \sin(\text{PSI}_C) * \sin(\text{THETA}_C)) + (-0.166667E0) * \dots$$

$$(L12.^2+4 * L13.^2+(-4) * L12 * L13 * \cos(\text{BETA})) .^ (1/2) * (1+(-1) * (L12.^2+ \dots$$

$$4 * L13.^2+(-4) * L12 * L13 * \cos(\text{BETA})) .^(-1) * (0.1E1 * L12.^2+(-0.2E1) * \dots$$

$$L13.^2+(-0.1E1) * L12 * L13 * \cos(\text{BETA})) .^2 * (L12.^2+L13.^2+2 * L12 * L13 * \dots$$

$$\cos(\text{BETA})) .^(-1)) .^(-1/2) * ((-2) * L12 * L13 * (L12.^2+4 * L13.^2+(-4) * \dots$$

```

L12.*L13.*cos(BETA)).^(-1).*(0.1E1.*L12.^2+(-0.2E1).*L13.^2+(-0.1E1).* ...
L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-2).* ...
sin(BETA)+(-0.2E1).*L12.*L13.*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos( ...
BETA)).^(-1).*(0.1E1.*L12.^2+(-0.2E1).*L13.^2+(-0.1E1).*L12.*L13.*cos( ...
BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1).*sin(BETA)+4.*L12.* ...
L13.*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-2).*(0.1E1.*L12.^2+ ...
(-0.2E1).*L13.^2+(-0.1E1).*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.* ...
L12.*L13.*cos(BETA)).^(-1).*sin(BETA)).*(cos(PHI_C).*cos(PHI_C)+sin( ...
PHI_C).*sin(PHI_C).*sin(THETA_C));
IJ_7_12 = 0;
IJ_7_13 = 0;
IJ_7_14 = 0;
IJ_7_15 = 0;
IJ_7_16 = 0;
IJ_7_17 = 0;
IJ_7_18 = 0;
IJ_7_19 = 0;
IJ_7_20 = 0;
IJ_7_21 = 0;
IJ_7_22 = 0;
IJ_7_23 = 0;
IJ_7_24 = 0;

IJ_8_1 = 0;
IJ_8_2 = 1;
IJ_8_3 = 0;
IJ_8_4 = 0;
IJ_8_5 = 0;
IJ_8_6=cos(PHI_C).*((-1).*cos(XI).*sin(PHI_C)+cos(PHI_C).*sin(ALPHA).*sin( ...
XI))+sin(PHI_C).*(cos(PHI_C).*cos(XI).*sin(THETA_C)+(cos(ALPHA).* ...
cos(THETA_C)+sin(ALPHA).*sin(PHI_C).*sin(THETA_C)).*sin(XI));
IJ_8_7=(-1).*LB4.*sin(PHI_C).*((-1).*cos(XI).*sin(PHI_C)+cos(PHI_C).*sin( ...
ALPHA).*sin(XI))+LB4.*cos(PHI_C).*(cos(PHI_C).*cos(XI).*sin(THETA_C) ...
+(cos(ALPHA).*cos(THETA_C)+sin(ALPHA).*sin(PHI_C).*sin(THETA_C)).*sin( ...
XI));
IJ_8_8=LB4.*sin(PHI_C).*(cos(PHI_C).*cos(THETA_C).*cos(XI)+(cos(THETA_C).* ...
sin(ALPHA).*sin(PHI_C)+(-1).*cos(ALPHA).*sin(THETA_C)).*sin(XI));
IJ_8_9=LB4.*cos(PHI_C).*((-1).*cos(PHI_C).*cos(XI)+(-1).*sin(ALPHA).*sin( ...
PHI_C).*sin(XI))+LB4.*sin(PHI_C).*((-1).*cos(XI).*sin(PHI_C).*sin( ...
THETA_C)+cos(PHI_C).*sin(ALPHA).*sin(THETA_C).*sin(XI));
IJ_8_10=LB4.*cos(ALPHA).*cos(PHI_C).*cos(PHI_C).*sin(XI)+LB4.*sin(PHI_C).*(( ...
-1).*cos(THETA_C).*sin(ALPHA)+cos(ALPHA).*sin(PHI_C).*sin(THETA_C)).* ...
sin(XI);
IJ_8_11 = 0;
IJ_8_12=LB4.*cos(PHI_C).*(cos(PHI_C).*cos(XI).*sin(ALPHA)+sin(PHI_C).*sin( ...
XI))+LB4.*sin(PHI_C).*(cos(XI).*(cos(ALPHA).*cos(THETA_C)+sin(ALPHA) ...
.*sin(PHI_C).*sin(THETA_C))+(-1).*cos(PHI_C).*sin(THETA_C).*sin(XI)); ...
IJ_8_13 = 0;
IJ_8_14 = 0;
IJ_8_15 = 0;
IJ_8_16 = 0;
IJ_8_17 = 0;
IJ_8_18 = 0;
IJ_8_19 = 0;
IJ_8_20 = 0;
IJ_8_21 = 0;
IJ_8_22 = 0;
IJ_8_23 = 0;
IJ_8_24 = 0;

IJ_9_1 = 0;
IJ_9_2 = 0;
IJ_9_3 = 1;
IJ_9_4=(-0.333333E0).*(L12+L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos( ...
BETA)).^(-1/2).*sin(THETA_C);
IJ_9_5=(-0.333333E0).*(L13+L12.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos( ...
BETA)).^(-1/2).*sin(THETA_C);
IJ_9_6 = 0;
IJ_9_7 = 0;
IJ_9_8=(-0.333333E0).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(1/2).*cos( ...
THETA_C);

```

```

IJ_9_9 = 0;
IJ_9_10 = 0;
IJ_9_11=0.333333E0.*L12.*L13.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1/2).* ...
    sin(BETA).*sin(THETA_C);
IJ_9_12 = 0;
IJ_9_13 = 0;
IJ_9_14 = 0;
IJ_9_15 = 0;
IJ_9_16 = 0;
IJ_9_17 = 0;
IJ_9_18 = 0;
IJ_9_19 = 0;
IJ_9_20 = 0;
IJ_9_21 = 0;
IJ_9_22 = 0;
IJ_9_23 = 0;
IJ_9_24 = 0;

IJ_10_1 = 0;
IJ_10_2 = 0;
IJ_10_3 = 1;
IJ_10_4=0.166667E0.*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(1/2).*((2.* ...
    L12+2.*L13.*cos(BETA)).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^( ...
    -1).*((0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*( ...
    L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-2)+(-2).*((0.4E1.*L12+1.E0.* ...
    L13.*cos(BETA)).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1)).*( ...
    0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).*(L12.^2+ ...
    L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)+(8.*L12+(-4).*L13.*cos(BETA)).*(4.* ...
    L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-2).*((0.2E1.*L12.^2+(-1.E0).* ...
    L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos( ...
    BETA)).^(-1)).*(1+(-1)).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^( ...
    -1)).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*( ...
    L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^(1/2).*cos(THETA_C).*sin( ...
    PHI_C)+0.166667E0.*(8.*L12+(-4).*L13.*cos(BETA)).*(4.*L12.^2+L13.^2+(-4) ...
    .*L12.*L13.*cos(BETA)).^(1/2)).*(1+(-1)).*(4.*L12.^2+L13.^2+(-4).*L12.* ...
    L13.*cos(BETA)).^(-1)).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.* ...
    cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^(1/2).* ...
    cos(THETA_C).*sin(PHI_C)+(-1/2)).*(2.*L12+2.*L13.*cos(BETA)).*( ...
    0.666667E0.*L12.^2+(-0.333333E0).*L13.^2+0.333333E0.*L12.*L13.*cos(BETA) ...
    ).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-3/2).*sin(THETA_C)+( ...
    0.133333E1.*L12+0.333333E0.*L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.* ...
    L13.*cos(BETA)).^(-1/2).*sin(THETA_C);
IJ_10_5=0.166667E0.*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(1/2).*((2.* ...
    L13+2.*L12.*cos(BETA)).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^( ...
    -1).*((0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*( ...
    L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-2)+(-2).*((-0.2E1).*L13+1.E0.* ...
    L12.*cos(BETA)).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1)).*( ...
    0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).*(L12.^2+ ...
    L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)+(2.*L13+(-4).*L12.*cos(BETA)).*(4.* ...
    L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-2).*((0.2E1.*L12.^2+(-1.E0).* ...
    L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos( ...
    BETA)).^(-1)).*(1+(-1)).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^( ...
    -1)).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*( ...
    L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^(1/2).*cos(THETA_C).*sin( ...
    PHI_C)+0.166667E0.*(2.*L13+(-4).*L12.*cos(BETA)).*(4.*L12.^2+L13.^2+(-4) ...
    .*L12.*L13.*cos(BETA)).^(1/2)).*(1+(-1)).*(4.*L12.^2+L13.^2+(-4).*L12.* ...
    L13.*cos(BETA)).^(-1)).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.* ...
    cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^(1/2).* ...
    cos(THETA_C).*sin(PHI_C)+(-1/2)).*(2.*L13+2.*L12.*cos(BETA)).*( ...
    0.666667E0.*L12.^2+(-0.333333E0).*L13.^2+0.333333E0.*L12.*L13.*cos(BETA) ...
    ).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-3/2).*sin(THETA_C)+( ...
    -0.666667E0).*L13+0.333333E0.*L12.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.* ...
    L13.*cos(BETA)).^(-1/2).*sin(THETA_C);
IJ_10_6 = 0;
IJ_10_7 = 0;
IJ_10_8=(0.666667E0.*L12.^2+(-0.333333E0).*L13.^2+0.333333E0.*L12.*L13.*cos( ...
    BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(1/2).*cos(THETA_C)+( ...
    -0.333333E0)).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(1/2)).*(1+( ...
    -1)).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1)).*(0.2E1.*L12.^2+ ...
    (-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.* ...
    L13.*cos(BETA)).^(-1)).^(1/2).*sin(PHI_C).*sin(THETA_C);

```

```

IJ_10_9=0.333333E0.*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(1/2).*(1+(-1) ...
.*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*(0.2E1.*L12.^2+( ...
-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.* ...
L13.*cos(BETA)).^(-1)).^(1/2).*cos(PHI_C).*cos(THETA_C);
IJ_10_10 = 0;
IJ_10_11=0.666667E0.*L12.*L13.*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^( ...
-1/2).*(1+(-1).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*( ...
0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*(L12.^2+ ...
L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^(1/2).*cos(THETA_C).*sin(BETA).* ...
sin(PHI_C)+0.166667E0.*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^( ...
1/2).*(1+(-1).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*( ...
0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*(L12.^2+ ...
L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^(1/2).*cos(THETA_C).*((-2).* ...
L12.*L13.*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*(0.2E1.* ...
L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.* ...
L12.*L13.*cos(BETA)).^(-2).*sin(BETA)+0.2E1.*L12.*L13.*(4.*L12.^2+ ...
L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+ ...
1.E0.*L12.*L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1) ...
.*sin(BETA)+4.*L12.*L13.*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^( ...
-2).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*( ...
L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1).*sin(BETA)).*sin(PHI_C)+ ...
L12.*L13.*(0.666667E0.*L12.^2+(-0.333333E0).*L13.^2+0.333333E0.*L12.* ...
L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-3/2).*sin( ...
BETA).*sin(THETA_C)+(-0.333333E0).*L12.*L13.*(L12.^2+L13.^2+2.*L12.* ...
L13.*cos(BETA)).^(-1/2).*sin(BETA).*sin(THETA_C);
IJ_10_12 = 0;
IJ_10_13 = 0;
IJ_10_14 = 0;
IJ_10_15 = 0;
IJ_10_16 = 0;
IJ_10_17 = 0;
IJ_10_18 = 0;
IJ_10_19 = 0;
IJ_10_20 = 0;
IJ_10_21 = 0;
IJ_10_22 = 0;
IJ_10_23 = 0;
IJ_10_24 = 0;

IJ_11_1 = 0;
IJ_11_2 = 0;
IJ_11_3 = 1;
IJ_11_4=(-0.166667E0).*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^(1/2).*(( ...
2.*L12+2.*L13.*cos(BETA)).*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)) ...
.^(-1).*(0.1E1.*L12.^2+(-0.2E1).*L13.^2+(-0.1E1).*L12.*L13.*cos(BETA)) ...
.^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-2)+(-2).*(0.2E1.*L12+( ...
-0.1E1).*L13.*cos(BETA)).*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)) ...
.^(-1).*(0.1E1.*L12.^2+(-0.2E1).*L13.^2+(-0.1E1).*L12.*L13.*cos(BETA)).* ...
(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)+(2.*L12+(-4).*L13.*cos( ...
BETA)).*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-2).*(0.1E1.* ...
L12.^2+(-0.2E1).*L13.^2+(-0.1E1).*L12.*L13.*cos(BETA)).^2.*(L12.^2+ ...
L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).*(1+(-1).*(L12.^2+4.*L13.^2+(-4).* ...
L12.*L13.*cos(BETA)).^(-1).*(0.1E1.*L12.^2+(-0.2E1).*L13.^2+(-0.1E1).* ...
L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1) ...
.^(-1/2).*cos(THETA_C).*sin(PHI_C)+(-0.166667E0).*(2.*L12+(-4).*L13.* ...
cos(BETA)).*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1/2).*(1+( ...
-1).*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*(0.1E1.*L12.^2+ ...
(-0.2E1).*L13.^2+(-0.1E1).*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.* ...
L12.*L13.*cos(BETA)).^(-1)).^(1/2).*cos(THETA_C).*sin(PHI_C)+(-1/2).*( ...
2.*L12+2.*L13.*cos(BETA)).*((-0.333333E0).*L12.^2+0.666667E0.*L13.^2+ ...
0.333333E0.*L12.*L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)) ...
.^(-3/2).*sin(THETA_C)+((-0.666667E0).*L12+0.333333E0.*L13.*cos(BETA)).* ...
(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1/2).*sin(THETA_C);
IJ_11_5=(-0.166667E0).*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^(1/2).*(( ...
2.*L13+2.*L12.*cos(BETA)).*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)) ...
.^(-1).*(0.1E1.*L12.^2+(-0.2E1).*L13.^2+(-0.1E1).*L12.*L13.*cos(BETA)) ...
.^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-2)+(-2).*((-0.4E1).*L13+( ...
-0.1E1).*L12.*cos(BETA)).*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)) ...
.^(-1).*(0.1E1.*L12.^2+(-0.2E1).*L13.^2+(-0.1E1).*L12.*L13.*cos(BETA)).* ...
(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)+(8.*L13+(-4).*L12.*cos( ...
BETA)).*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-2).*(0.1E1.* ...

```

```

L12.^2+(-0.2E1).*L13.^2+(-0.1E1).*L12.*L13.*cos(BETA)).^2.*(L12.^2+ ...
L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).*(1+(-1)).*(L12.^2+4.*L13.^2+(-4).* ...
L12.*L13.*cos(BETA)).^(-1)).*(0.1E1.*L12.^2+(-0.2E1).*L13.^2+(-0.1E1).* ...
L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)) ...
.^(-1/2).*cos(THETA_C).*sin(PHI_C)+(-0.166667E0).* (8.*L13+(-4).*L12.* ...
cos(BETA)).*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1/2)).*(1+( ...
-1)).*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1)).*(0.1E1.*L12.^2+ ...
(-0.2E1).*L13.^2+(-0.1E1).*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.* ...
L12.*L13.*cos(BETA)).^(-1)).^(1/2).*cos(THETA_C).*sin(PHI_C)+(-1/2)).*( ...
2.*L13+2.*L12.*cos(BETA)).*((-0.333333E0).*L12.^2+0.666667E0.*L13.^2+ ...
0.333333E0.*L12.*L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)) ...
.^(-3/2)).*sin(THETA_C)+(0.133333E1.*L13+0.333333E0.*L12.*cos(BETA)).*( ...
L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1/2)).*sin(THETA_C);
IJ_11_6 = 0;
IJ_11_7 = 0;
IJ_11_8=( (-0.333333E0).*L12.^2+0.666667E0.*L13.^2+0.333333E0.*L12.*L13.*cos( ...
BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1/2)).*cos(THETA_C)+ ...
0.333333E0.*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^(1/2)).*(1+(-1) ...
.*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1)).*(0.1E1.*L12.^2+ ...
-0.2E1).*L13.^2+(-0.1E1).*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.* ...
L12.*L13.*cos(BETA)).^(-1)).^(1/2)).*sin(PHI_C).*sin(THETA_C);
IJ_11_9=(-0.333333E0).* (L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^(1/2)).*(1+( ...
-1)).*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1)).*(0.1E1.*L12.^2+ ...
(-0.2E1).*L13.^2+(-0.1E1).*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.* ...
L12.*L13.*cos(BETA)).^(-1)).^(1/2)).*cos(PHI_C).*cos(THETA_C);
IJ_11_10 = 0;
IJ_11_11=(-0.666667E0).*L12.*L13.*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^( ...
-1/2)).*(1+(-1)).*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1)).*( ...
0.1E1.*L12.^2+(-0.2E1).*L13.^2+(-0.1E1).*L12.*L13.*cos(BETA)).^2.*( ...
L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^(1/2)).*cos(THETA_C).*sin( ...
BETA).*sin(PHI_C)+(-0.166667E0).* (L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos( ...
BETA)).^(1/2)).*(1+(-1)).*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^( ...
-1)).*(0.1E1.*L12.^2+(-0.2E1).*L13.^2+(-0.1E1).*L12.*L13.*cos(BETA)).^2.* ...
(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^(1/2)).*cos(THETA_C).*(( ...
-2)).*L12.*L13.*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1)).*( ...
0.1E1.*L12.^2+(-0.2E1).*L13.^2+(-0.1E1).*L12.*L13.*cos(BETA)).^2.*( ...
L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-2)).*sin(BETA)+(-0.2E1).*L12.* ...
L13.*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1)).*(0.1E1.*L12.^2+ ...
(-0.2E1).*L13.^2+(-0.1E1).*L12.*L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.* ...
L13.*cos(BETA)).^(-1)).*sin(BETA)+4.*L12.*L13.*(L12.^2+4.*L13.^2+(-4).* ...
L12.*L13.*cos(BETA)).^(-2)).*(0.1E1.*L12.^2+(-0.2E1).*L13.^2+(-0.1E1).* ...
L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).* ...
sin(BETA)).*sin(PHI_C)+L12.*L13.*((-0.333333E0).*L12.^2+0.666667E0.* ...
L13.^2+0.333333E0.*L12.*L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.* ...
cos(BETA)).^(-3/2)).*sin(BETA).*sin(THETA_C)+(-0.333333E0).*L12.*L13.*( ...
L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1/2)).*sin(BETA).*sin(THETA_C); ...
IJ_11_12 = 0;
IJ_11_13 = 0;
IJ_11_14 = 0;
IJ_11_15 = 0;
IJ_11_16 = 0;
IJ_11_17 = 0;
IJ_11_18 = 0;
IJ_11_19 = 0;
IJ_11_20 = 0;
IJ_11_21 = 0;
IJ_11_22 = 0;
IJ_11_23 = 0;
IJ_11_24 = 0;

IJ_12_1 = 0;
IJ_12_2 = 0;
IJ_12_3 = 1;
IJ_12_4 = 0;
IJ_12_5 = 0;
IJ_12_6=cos(PHI_C).*cos(THETA_C).*cos(XI)+(cos(THETA_C).*sin(ALPHA).*sin( ...
PHI_C)+(-1)).*cos(ALPHA).*sin(THETA_C)).*sin(XI);
IJ_12_7 = 0;
IJ_12_8=(-1).*LB4.*cos(PHI_C).*cos(XI).*sin(THETA_C)+LB4.*((-1).*cos(ALPHA).* ...
cos(THETA_C)+(-1)).*sin(ALPHA).*sin(PHI_C).*sin(THETA_C)).*sin(XI);
IJ_12_9=(-1).*LB4.*cos(THETA_C).*cos(XI).*sin(PHI_C)+LB4.*cos(PHI_C).*cos( ...

```



```

    THETA_C).*sin(ALPHA).*sin(XI);
IJ_12_10=LB4.*(cos(ALPHA).*cos(THETA_C).*sin(PHI_C)+sin(ALPHA).*sin(THETA_C)).* ...
sin(XI);
IJ_12_11 = 0;
IJ_12_12=LB4.*cos(XI).(cos(THETA_C).*sin(ALPHA).*sin(PHI_C)+(-1).*cos(ALPHA).* ...
sin(THETA_C))+(-1).*LB4.*cos(PHI_C).*cos(THETA_C).*sin(XI);
IJ_12_13 = 0;
IJ_12_14 = 0;
IJ_12_15 = 0;
IJ_12_16 = 0;
IJ_12_17 = 0;
IJ_12_18 = 0;
IJ_12_19 = 0;
IJ_12_20 = 0;
IJ_12_21 = 0;
IJ_12_22 = 0;
IJ_12_23 = 0;
IJ_12_24 = 0;

IJ_13_1 = 0;
IJ_13_2 = 0;
IJ_13_3 = 0;
IJ_13_4 = 0;
IJ_13_5 = 0;
IJ_13_6 = 0;
IJ_13_7 = 1;
IJ_13_8=(cos(PHI_C).*cos(THETA_1).*sin(PHI_C).*sin(THETA_1)).*(cos( ...
PHI_C).*cos(THETA_C).*sin(THETA_1)+cos(THETA_1).*((-1).*cos(THETA_C).* ...
sin(PHI_C).*sin(PHI_C)+cos(PHI_C).*sin(THETA_C)).*(cos(PHI_C).^2.*cos( ...
THETA_1).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C).* ...
sin(PHI_C).*sin(2.*THETA_1)+2.*cos(PHI_C).*cos(THETA_1).*cos(THETA_C).* ...
cos(THETA_1).*sin(PHI_C).*sin(PHI_C)+(-1).*cos(PHI_C).*sin(THETA_1)).* ...
sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_1).^2+cos(THETA_1).^2.*sin(PHI_1) ...
.^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_1).^2.*sin(PHI_1).^2+ ...
sin(THETA_1).^2.*sin(THETA_C).^2)).^(-1);
IJ_13_9=(1/4).**(4.*cos(PHI_1).*cos(THETA_1).*cos(THETA_C).*((-1).*cos(THETA_1).* ...
sin(PHI_C).*sin(PHI_C)+cos(PHI_C).*sin(THETA_1))+((-3)+cos(2.*PSI_1)+2.* ...
cos(PHI_1).^2.*cos(2.*THETA_1)).*sin(THETA_C)).*(cos(PHI_1).^2.*cos( ...
THETA_1).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C).* ...
sin(PHI_1).*sin(2.*THETA_1)+2.*cos(PHI_1).*cos(THETA_1).*cos(THETA_C).* ...
cos(THETA_1).*sin(PHI_C).*sin(PHI_1)+(-1).*cos(PHI_C).*sin(THETA_1)).* ...
sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_1).^2+cos(THETA_1).^2.*sin(PHI_1) ...
.^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_1).^2.*sin(PHI_1).^2+ ...
sin(THETA_1).^2.*sin(THETA_C).^2)).^(-1);
IJ_13_10 = 0;
IJ_13_11 = 0;
IJ_13_12 = 0;
IJ_13_13=(1/2).**(2.*cos(PHI_C).*cos(THETA_1).^2.*cos(THETA_C)+cos(THETA_C).*sin( ...
PHI_C).*sin(PHI_1).*sin(2.*THETA_1)+(-2).*cos(PHI_1).*cos(THETA_1).*sin( ...
THETA_1).*sin(THETA_C)).*(cos(PHI_1).^2.*cos(THETA_1).^2.*cos(THETA_C) ...
.^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C).*sin(PHI_1).*sin(2.*THETA_1) ...
+2.*cos(PHI_1).*cos(THETA_1).*cos(THETA_C).*(cos(THETA_1).*sin(PHI_C).* ...
sin(PHI_1)+(-1).*cos(PHI_C).*sin(THETA_1)).*sin(THETA_C)+sin(PHI_C).^2.* ...
(sin(THETA_1).^2+cos(THETA_1).^2.*sin(PHI_1).^2.*sin(THETA_C).^2)+cos( ...
PHI_C).^2.*(cos(THETA_1).^2.*sin(PHI_1).^2+sin(THETA_1).^2.*sin(THETA_C) ...
.^2)).^(-1);
IJ_13_14=(cos(PHI_1).*cos(THETA_C).*sin(PHI_C)+sin(PHI_1).*sin(THETA_C)).*(cos( ...
PSI_1).^2.*cos(THETA_1).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C) ...
.^2.*sin(PHI_C).*sin(PHI_1).*sin(2.*THETA_1)+2.*cos(PHI_1).*cos(THETA_1) ...
.*cos(THETA_C).*(cos(THETA_1).*sin(PHI_C).*sin(PHI_1)+(-1).*cos(PHI_C).* ...
sin(THETA_1)).*sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_1).^2+cos(THETA_1) ...
.^2.*sin(PHI_1).^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_1).^2.* ...
sin(PHI_1).^2+sin(THETA_1).^2.*sin(THETA_C).^2)).^(-1);
IJ_13_15 = 0;
IJ_13_16 = 0;
IJ_13_17 = 0;
IJ_13_18 = 0;
IJ_13_19 = 0;
IJ_13_20 = 0;
IJ_13_21 = 0;
IJ_13_22 = 0;

```

```

IJ_13_23 = 0;
IJ_13_24 = 0;

IJ_14_1 = 0;
IJ_14_2 = 0;
IJ_14_3 = 0;
IJ_14_4 = 0;
IJ_14_5 = 0;
IJ_14_6 = 0;
IJ_14_7 = 0;
IJ_14_8=(cos(PHI_1)*cos(THETA_1)*cos(THETA_C)+(cos(THETA_1)*sin(PHI_C)*sin( ...
    PSI_1)+(-1)*cos(PHI_C)*sin(THETA_1))*sin(THETA_C))* (1+(-1)*(cos( ...
    PHI_C)*cos(THETA_C)*sin(THETA_1)+cos(THETA_1)*((-1)*cos(THETA_C)* ...
    sin(PHI_C)*sin(PSI_1)+cos(PSI_1)*sin(THETA_C)))^2)^(-1/2);
IJ_14_9=(-1)*cos(THETA_C)*(cos(PHI_C)*cos(THETA_1)*sin(PSI_1)+sin(PHI_C)* ...
    sin(THETA_1))* (1+(-1)*(cos(PHI_C)*cos(THETA_C)*sin(THETA_1)+cos( ...
    THETA_1)*((-1)*cos(THETA_C)*sin(PHI_C)*sin(PSI_1)+cos(PSI_1)*sin( ...
    THETA_C)))^2)^(-1/2);
IJ_14_10 = 0;
IJ_14_11 = 0;
IJ_14_12 = 0;
IJ_14_13=(-1)*cos(THETA_1)*(cos(PSI_1)*cos(THETA_C)*sin(PHI_C)+sin(PSI_1)* ...
    sin(THETA_C))* (1+(-1)*(cos(PHI_C)*cos(THETA_C)*sin(THETA_1)+cos( ...
    THETA_1)*((-1)*cos(THETA_C)*sin(PHI_C)*sin(PSI_1)+cos(PSI_1)*sin( ...
    THETA_C)))^2)^(-1/2);
IJ_14_14=(cos(PHI_C)*cos(THETA_1)*cos(THETA_C)+sin(THETA_1)*(cos(THETA_C)* ...
    sin(PHI_C)*sin(PSI_1)+(-1)*cos(PSI_1)*sin(THETA_C))* (1+(-1)*(cos( ...
    PHI_C)*cos(THETA_C)*sin(THETA_1)+cos(THETA_1)*((-1)*cos(THETA_C)* ...
    sin(PHI_C)*sin(PSI_1)+cos(PSI_1)*sin(THETA_C)))^2)^(-1/2);
IJ_14_15 = 0;
IJ_14_16 = 0;
IJ_14_17 = 0;
IJ_14_18 = 0;
IJ_14_19 = 0;
IJ_14_20 = 0;
IJ_14_21 = 0;
IJ_14_22 = 0;
IJ_14_23 = 0;
IJ_14_24 = 0;

IJ_15_1 = 0;
IJ_15_2 = 0;
IJ_15_3 = 0;
IJ_15_4 = 0;
IJ_15_5 = 0;
IJ_15_6 = 0;
IJ_15_7 = 0;
IJ_15_8=(cos(PHI_C)*cos(THETA_1)*sin(PSI_1)+sin(PHI_C)*sin(THETA_1))*(cos( ...
    PHI_C)^2*cos(THETA_1)^2*cos(THETA_C)^2+2*cos(PHI_C)*cos(THETA_1) ...
    *cos(THETA_C)*sin(THETA_1)*(cos(THETA_C)*sin(PHI_C)*sin(PSI_1)+(-1) ...
    *cos(PSI_1)*sin(THETA_C))+sin(PSI_1)^2*(cos(THETA_C)^2*sin(PHI_C) ...
    ^2*sin(THETA_1)^2+sin(THETA_C)^2)+cos(PSI_1)^2*(cos(THETA_C)^2* ...
    sin(PHI_C)^2+sin(THETA_1)^2*sin(THETA_C)^2)+cos(PSI_1)*cos(THETA_1) ...
    ^2*sin(PHI_C)*sin(PSI_1)*sin(2*THETA_C))^(-1);
IJ_15_9=(1/2)*(2*cos(PSI_1)*cos(THETA_1)*cos(THETA_C)^2+(-2)*cos(PHI_C)* ...
    cos(THETA_C)*sin(THETA_1)*sin(THETA_C)+cos(THETA_1)*sin(PHI_C)*sin( ...
    PSI_1)*sin(2*THETA_C))*(cos(PHI_C)^2*cos(THETA_1)^2*cos(THETA_C) ...
    ^2+2*cos(PHI_C)*cos(THETA_1)*cos(THETA_C)*sin(THETA_1)*(cos( ...
    THETA_C)*sin(PHI_C)*sin(PSI_1)+(-1)*cos(PSI_1)*sin(THETA_C))+sin( ...
    PSI_1)^2*(cos(THETA_C)^2*sin(PHI_C)^2*sin(THETA_1)^2+sin(THETA_C) ...
    ^2)+cos(PSI_1)^2*(cos(THETA_C)^2*sin(PHI_C)^2+sin(THETA_1)^2* ...
    sin(THETA_C)^2)+cos(PSI_1)*cos(THETA_1)^2*sin(PHI_C)*sin(PSI_1)* ...
    sin(2*THETA_C))^(-1);
IJ_15_10 = 0;
IJ_15_11 = 0;
IJ_15_12 = 0;
IJ_15_13=(1/4)*(cos(PHI_C)^2*cos(THETA_1)^2*cos(THETA_C)^2+2*cos(PHI_C)* ...
    cos(THETA_1)*cos(THETA_C)*sin(THETA_1)*(cos(THETA_C)*sin(PHI_C)* ...
    sin(PSI_1)+(-1)*cos(PSI_1)*sin(THETA_C))+sin(PSI_1)^2*(cos(THETA_C) ...
    ^2*sin(PHI_C)^2*sin(THETA_1)^2+sin(THETA_C)^2)+cos(PSI_1)^2*( ...
    cos(THETA_C)^2*sin(PHI_C)^2+sin(THETA_1)^2*sin(THETA_C)^2)+cos( ...

```

```

PSI_1).*cos(THETA_1).^2.*sin(PHI_C).*sin(PSI_1).*sin(2.*THETA_C)).^(-1) ...
.*((-3)+cos(2.*PHI_C)+2.*cos(PHI_C).^2.*cos(2.*THETA_C)).*sin(THETA_1)+ ...
cos(THETA_1).*((-2).*cos(THETA_C).^2.*sin(2.*PHI_C).*sin(PSI_1)+2.*cos( ...
PHI_C).*cos(PSI_1).*sin(2.*THETA_C)));
IJ_15_14=(cos(PSI_1).*cos(THETA_C).*sin(PHI_C)+sin(PSI_1).*sin(THETA_C)).*(cos( ...
PHI_C).*cos(THETA_C).*sin(THETA_1)+cos(THETA_1).*((-1).*cos(THETA_C)).* ...
sin(PHI_C).*sin(PSI_1)+cos(PSI_1).*sin(THETA_C)).*(cos(PHI_C).^2.*cos( ...
THETA_1).^2.*cos(THETA_C).^2+2.*cos(PHI_C).*cos(THETA_1).*cos(THETA_C).* ...
sin(THETA_1).(cos(THETA_C).*sin(PHI_C).*sin(PSI_1)+(-1).*cos(PSI_1)).* ...
sin(THETA_C))+sin(PSI_1).^2.*(cos(THETA_C).^2.*sin(PHI_C).^2.*sin( ...
THETA_1).^2+sin(THETA_C).^2)+cos(PSI_1).^2.*(cos(THETA_C).^2.*sin(PHI_C) ...
.^2+sin(THETA_1).^2.*sin(THETA_C).^2)+cos(PSI_1).*cos(THETA_1).^2.*sin( ...
PHI_C).*sin(PSI_1).*sin(2.*THETA_C)).^(-1);
IJ_15_15 = 1;
IJ_15_16 = 0;
IJ_15_17 = 0;
IJ_15_18 = 0;
IJ_15_19 = 0;
IJ_15_20 = 0;
IJ_15_21 = 0;
IJ_15_22 = 0;
IJ_15_23 = 0;
IJ_15_24 = 0;

IJ_16_1 = 0;
IJ_16_2 = 0;
IJ_16_3 = 0;
IJ_16_4 = 0;
IJ_16_5 = 0;
IJ_16_6 = 0;
IJ_16_7 = 1;
IJ_16_8=(cos(PHI_C).*cos(THETA_2).*sin(PSI_2)+sin(PHI_C).*sin(THETA_2)).*(cos( ...
PHI_C).*cos(THETA_C).*sin(THETA_2)+cos(THETA_2).*((-1).*cos(THETA_C)).* ...
sin(PHI_C).*sin(PSI_2)+cos(PSI_2).*sin(THETA_C)).*(cos(PSI_2).^2.*cos( ...
THETA_2).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C).* ...
sin(PSI_2).^2.*sin(2.*THETA_2)+2.*cos(PSI_2).*cos(THETA_2).*cos(THETA_C).* ...
cos(THETA_2).*sin(PHI_C).*sin(PSI_2)+(-1).*cos(PHI_C).*sin(THETA_2)).* ...
sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_2).^2+cos(THETA_2).^2.*sin(PSI_2) ...
.^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_2).^2.*sin(PSI_2).^2+ ...
sin(THETA_2).^2.*sin(THETA_C).^2)).^(-1);
IJ_16_9=(1/4).*(4.*cos(PSI_2).*cos(THETA_2).*cos(THETA_C).*((-1).*cos(THETA_2)).* ...
sin(PHI_C).*sin(PSI_2)+cos(PHI_C).*sin(THETA_2))+((-3)+cos(2.*PSI_2)+2.* ...
cos(PSI_2).^2.*cos(2.*THETA_2)).*sin(THETA_C)).*(cos(PSI_2).^2.*cos( ...
THETA_2).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C).* ...
sin(PSI_2).*sin(2.*THETA_2)+2.*cos(PSI_2).*cos(THETA_2).*cos(THETA_C).* ...
cos(THETA_2).*sin(PHI_C).*sin(PSI_2)+(-1).*cos(PHI_C).*sin(THETA_2)).* ...
sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_2).^2+cos(THETA_2).^2.*sin(PSI_2) ...
.^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_2).^2.*sin(PSI_2).^2+ ...
sin(THETA_2).^2.*sin(THETA_C).^2)).^(-1);
IJ_16_10 = 0;
IJ_16_11 = 0;
IJ_16_12 = 0;
IJ_16_13 = 0;
IJ_16_14 = 0;
IJ_16_15 = 0;
IJ_16_16=(1/2).*(2.*cos(PHI_C).*cos(THETA_2).^2.*cos(THETA_C)+cos(THETA_C).*sin( ...
PHI_C).*sin(PSI_2).*sin(2.*THETA_2)+(-2).*cos(PSI_2).*cos(THETA_2).*sin( ...
THETA_2).*sin(THETA_C)).*(cos(PSI_2).^2.*cos(THETA_2).^2.*cos(THETA_C) ...
.^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C).*sin(PSI_2).*sin(2.*THETA_2) ...
+2.*cos(PSI_2).^2.*cos(THETA_2).*cos(THETA_C)).*(cos(THETA_2).*sin(PHI_C).* ...
sin(PSI_2)+(-1).*cos(PHI_C).*sin(THETA_2)).*sin(THETA_C)+sin(PHI_C).^2.* ...
(sin(THETA_2).^2+cos(THETA_2).^2.*sin(PSI_2).^2.*sin(THETA_C).^2)+cos( ...
PHI_C).^2.*(cos(THETA_2).^2.*sin(PSI_2).^2+sin(THETA_2).^2.*sin(THETA_C) ...
.^2)).^(-1);
IJ_16_17=(cos(PSI_2).*cos(THETA_C).*sin(PHI_C)+sin(PSI_2).*sin(THETA_C)).*(cos( ...
PSI_2).^2.*cos(THETA_2).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C) ...
.^2.*sin(PHI_C).*sin(PSI_2).*sin(2.*THETA_2)+2.*cos(PSI_2).*cos(THETA_2) ...
.*cos(THETA_C)).*(cos(THETA_2).*sin(PHI_C).*sin(PSI_2)+(-1).*cos(PHI_C).* ...
sin(THETA_2)).*sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_2).^2+cos(THETA_2) ...
.^2.*sin(PSI_2).^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_2).^2.* ...
sin(PSI_2).^2+sin(THETA_2).^2.*sin(THETA_C).^2)).^(-1);

```

```

IJ_16_18 = 0;
IJ_16_19 = 0;
IJ_16_20 = 0;
IJ_16_21 = 0;
IJ_16_22 = 0;
IJ_16_23 = 0;
IJ_16_24 = 0;

IJ_17_1 = 0;
IJ_17_2 = 0;
IJ_17_3 = 0;
IJ_17_4 = 0;
IJ_17_5 = 0;
IJ_17_6 = 0;
IJ_17_7 = 0;
IJ_17_8=(cos(PHI_2).*cos(THETA_2).*cos(THETA_C)+(cos(THETA_2).*sin(PHI_C).*sin( ...
    PSI_2)+(-1).*cos(PHI_C).*sin(THETA_2)).*sin(THETA_C)).*(1+(-1).*cos( ...
    PHI_C).*cos(THETA_C).*sin(THETA_2)+cos(THETA_2).*((-1).*cos(THETA_C).* ...
    sin(PHI_C).*sin(PSI_2)+cos(PSI_2).*sin(THETA_C)).^2).^(-1/2);
IJ_17_9=(-1).*cos(THETA_C).*cos(PHI_C).*cos(THETA_2).*sin(PSI_2)+sin(PHI_C).* ...
    sin(THETA_2)).*(1+(-1).*cos(PHI_C).*cos(THETA_C).*sin(THETA_2)+cos( ...
    THETA_2).*((-1).*cos(THETA_C).*sin(PHI_C).*sin(PSI_2)+cos(PSI_2).*sin( ...
    THETA_C)).^2).^(-1/2);
IJ_17_10 = 0;
IJ_17_11 = 0;
IJ_17_12 = 0;
IJ_17_13 = 0;
IJ_17_14 = 0;
IJ_17_15 = 0;
IJ_17_16=(-1).*cos(THETA_2).*cos(PSI_2).*cos(THETA_C).*sin(PHI_C)+sin(PSI_2).* ...
    sin(THETA_C)).*(1+(-1).*cos(PHI_C).*cos(THETA_C).*sin(THETA_2)+cos( ...
    THETA_2).*((-1).*cos(THETA_C).*sin(PHI_C).*sin(PSI_2)+cos(PSI_2).*sin( ...
    THETA_C)).^2).^(-1/2);
IJ_17_17=(cos(PHI_C).*cos(THETA_2).*cos(THETA_C)+sin(THETA_2).*cos(THETA_C).* ...
    sin(PHI_C).*sin(PSI_2)+(-1).*cos(PSI_2).*sin(THETA_C)).*(1+(-1).*cos( ...
    PHI_C).*cos(THETA_C).*sin(THETA_2)+cos(THETA_2).*((-1).*cos(THETA_C).* ...
    sin(PHI_C).*sin(PSI_2)+cos(PSI_2).*sin(THETA_C)).^2).^(-1/2);
IJ_17_18 = 0;
IJ_17_19 = 0;
IJ_17_20 = 0;
IJ_17_21 = 0;
IJ_17_22 = 0;
IJ_17_23 = 0;
IJ_17_24 = 0;

IJ_18_1 = 0;
IJ_18_2 = 0;
IJ_18_3 = 0;
IJ_18_4 = 0;
IJ_18_5 = 0;
IJ_18_6 = 0;
IJ_18_7 = 0;
IJ_18_8=(cos(PHI_C).*cos(THETA_2).*sin(PSI_2)+sin(PHI_C).*sin(THETA_2)).*(cos( ...
    PHI_C).^2.*cos(THETA_2).^2.*cos(THETA_C).^2+2.*cos(PHI_C).*cos(THETA_2) ...
    .*cos(THETA_C).*sin(THETA_2)).*(cos(THETA_C).*sin(PHI_C).*sin(PSI_2)+(-1) ...
    .*cos(PSI_2).*sin(THETA_C))+sin(PSI_2).^2.*(cos(THETA_C).^2.*sin(PHI_C) ...
    ^2.*sin(THETA_2).^2+sin(THETA_C).^2)+cos(PSI_2).^2.*(cos(THETA_C).^2.* ...
    sin(PHI_C).^2+sin(THETA_2).^2.*sin(THETA_C).^2)+cos(PSI_2).*cos(THETA_2) ...
    ^2.*sin(PHI_C).*sin(PSI_2).*sin(2.*THETA_C)).^(-1);
IJ_18_9=(1/2).*(2.*cos(PSI_2).*cos(THETA_2).*cos(THETA_C).^2+(-2).*cos(PHI_C).* ...
    cos(THETA_C).*sin(THETA_2)).*(sin(THETA_C)+cos(THETA_2)).*(sin(PHI_C).*sin( ...
    PSI_2).*sin(2.*THETA_C)).*(cos(PHI_C).^2.*cos(THETA_2).^2.*cos(THETA_C) ...
    ^2+2.*cos(PHI_C).*cos(THETA_2).*cos(THETA_C).*sin(THETA_2)).*(cos( ...
    THETA_C).*sin(PHI_C).*sin(PSI_2)+(-1).*cos(PSI_2).*sin(THETA_C))+sin( ...
    PSI_2).^2.*(cos(THETA_C).^2.*sin(PHI_C).^2.*sin(THETA_2).^2+sin(THETA_C) ...
    ^2)+cos(PSI_2).^2.*(cos(THETA_C).^2.*sin(PHI_C).^2+sin(THETA_2).^2.* ...
    sin(THETA_C).^2)+cos(PSI_2).*cos(THETA_2).^2.*sin(PHI_C).*sin(PSI_2).* ...
    sin(2.*THETA_C)).^(-1);
IJ_18_10 = 0;
IJ_18_11 = 0;
IJ_18_12 = 0;

```

```

IJ_18_13 = 0;
IJ_18_14 = 0;
IJ_18_15 = 0;
IJ_18_16=(1/4).*(cos(PHI_C).^2.*cos(THETA_2).^2.*cos(THETA_C).^2+2.*cos(PHI_C).* ...
cos(THETA_2).*cos(THETA_C).*sin(THETA_2).*(cos(THETA_C).*sin(PHI_C).* ...
sin(PSI_2)+(-1).*cos(PSI_2).*sin(THETA_C))+sin(PSI_2).^2.*(cos(THETA_C) ...
.^2.*sin(PHI_C).^2.*sin(THETA_2).^2+sin(THETA_C).^2)+cos(PSI_2).^2.*( ...
cos(THETA_C).^2.*sin(PHI_C).^2+sin(THETA_2).^2.*sin(THETA_C).^2)+cos( ...
PSI_2).*cos(THETA_2).^2.*sin(PHI_C).*sin(PSI_2).*sin(2.*THETA_C)).^(-1) ...
.*((-3)+cos(2.*PHI_C)+2.*cos(PHI_C).^2.*cos(2.*THETA_C)).*sin(THETA_2)+ ...
cos(THETA_2).*((-2).*cos(THETA_C).^2.*sin(2.*PHI_C).*sin(PSI_2)+2.*cos( ...
PHI_C).*cos(PSI_2).*sin(2.*THETA_C)));
IJ_18_17=(cos(PSI_2).*cos(THETA_C).*sin(PHI_C)+sin(PSI_2).*sin(THETA_C)).*(cos( ...
PHI_C).*cos(THETA_C).*sin(THETA_2)+cos(THETA_2).*((-1).*cos(THETA_C).* ...
sin(PHI_C).*sin(PSI_2)+cos(PSI_2).*sin(THETA_C))).*(cos(PHI_C).^2.*cos( ...
THETA_2).^2.*cos(THETA_C).^2+2.*cos(PHI_C).*cos(THETA_2).*cos(THETA_C).* ...
sin(THETA_2).*(cos(THETA_C).*sin(PHI_C).*sin(PSI_2)+(-1).*cos(PSI_2).* ...
sin(THETA_C))+sin(PSI_2).^2.*(cos(THETA_C).^2.*sin(PHI_C).^2.*sin( ...
THETA_2).^2+sin(THETA_C).^2)+cos(PSI_2).^2.*(cos(THETA_C).^2.*sin(PHI_C) ...
.^2+sin(THETA_2).^2.*sin(THETA_C).^2)+cos(PSI_2).*cos(THETA_2).^2.*sin( ...
PHI_C).*sin(PSI_2).*sin(2.*THETA_C)).^(-1);
IJ_18_18 = 1;
IJ_18_19 = 0;
IJ_18_20 = 0;
IJ_18_21 = 0;
IJ_18_22 = 0;
IJ_18_23 = 0;
IJ_18_24 = 0;

IJ_19_1 = 0;
IJ_19_2 = 0;
IJ_19_3 = 0;
IJ_19_4 = 0;
IJ_19_5 = 0;
IJ_19_6 = 0;
IJ_19_7 = 1;
IJ_19_8=(cos(PHI_C).*cos(THETA_3).*sin(PSI_3)+sin(PHI_C).*sin(THETA_3)).*(cos( ...
PHI_C).*cos(THETA_C).*sin(THETA_3)+cos(THETA_3).*((-1).*cos(THETA_C).* ...
sin(PHI_C).*sin(PSI_3)+cos(PSI_3).*sin(THETA_C))).*(cos(PSI_3).^2.*cos( ...
THETA_3).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C).* ...
sin(PSI_3).*sin(2.*THETA_3)+2.*cos(PSI_3).*cos(THETA_3).*cos(THETA_C).*( ...
cos(THETA_3).*sin(PHI_C).*sin(PSI_3)+(-1).*cos(PHI_C).*sin(THETA_3))).*( ...
sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_3).^2+cos(THETA_3).^2.*sin(PSI_3) ...
.^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_3).^2.*sin(PSI_3).^2+ ...
sin(THETA_3).^2.*sin(THETA_C).^2)).^(-1);
IJ_19_9=(1/4).*(4.*cos(PSI_3).*cos(THETA_3).*cos(THETA_C).*((-1).*cos(THETA_3).* ...
sin(PHI_C).*sin(PSI_3)+cos(PHI_C).*sin(THETA_3))+((-3)+cos(2.*PSI_3)+2.* ...
cos(PSI_3).^2.*cos(2.*THETA_3)).*sin(THETA_C)).*(cos(PSI_3).^2.*cos( ...
THETA_3).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C).* ...
sin(PSI_3).*sin(2.*THETA_3)+2.*cos(PSI_3).*cos(THETA_3).*cos(THETA_C).*( ...
cos(THETA_3).*sin(PHI_C).*sin(PSI_3)+(-1).*cos(PHI_C).*sin(THETA_3))).*( ...
sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_3).^2+cos(THETA_3).^2.*sin(PSI_3) ...
.^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_3).^2.*sin(PSI_3).^2+ ...
sin(THETA_3).^2.*sin(THETA_C).^2)).^(-1);
IJ_19_10 = 0;
IJ_19_11 = 0;
IJ_19_12 = 0;
IJ_19_13 = 0;
IJ_19_14 = 0;
IJ_19_15 = 0;
IJ_19_16 = 0;
IJ_19_17 = 0;
IJ_19_18 = 0;
IJ_19_19=(1/2).*(2.*cos(PHI_C).*cos(THETA_3).^2.*cos(THETA_C)+cos(THETA_C).*sin( ...
PHI_C).*sin(PSI_3).*sin(2.*THETA_3)+(-2).*cos(PSI_3).*cos(THETA_3).*sin( ...
THETA_3).*sin(THETA_C)).*(cos(PSI_3).^2.*cos(THETA_3).^2.*cos(THETA_C) ...
.^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C).*sin(PSI_3).*sin(2.*THETA_3) ...
+2.*cos(PSI_3).*cos(THETA_3).*cos(THETA_C).*(cos(THETA_3).*sin(PHI_C).* ...
sin(PSI_3)+(-1).*cos(PHI_C).*sin(THETA_3))).*sin(THETA_C)+sin(PHI_C).^2.* ...
(sin(THETA_3).^2+cos(THETA_3).^2.*sin(PSI_3).^2.*sin(THETA_C).^2)+cos( ...
PHI_C).^2.*(cos(THETA_3).^2.*sin(PSI_3).^2+sin(THETA_3).^2.*sin(THETA_C) ...

```

```

.^2)).^(-1);
IJ_19_20=(cos(PHI_3).*cos(THETA_C).*sin(PHI_C)+sin(PHI_3).*sin(THETA_C)).*(cos( ...
PSI_3).^2.*cos(THETA_3).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C) ...
.^2.*sin(PHI_C).*sin(PSI_3).*sin(2.*THETA_3)+2.*cos(PSI_3).*cos(THETA_3) ...
.*cos(THETA_C)).*(cos(THETA_3).*sin(PHI_C).*sin(PSI_3)+(-1).*cos(PHI_C).* ...
sin(THETA_3)).*sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_3).^2+cos(THETA_3) ...
.^2.*sin(PSI_3).^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_3).^2.* ...
sin(PSI_3).^2+sin(THETA_3).^2.*sin(THETA_C).^2)).^(-1);
IJ_19_21 = 0;
IJ_19_22 = 0;
IJ_19_23 = 0;
IJ_19_24 = 0;

IJ_20_1 = 0;
IJ_20_2 = 0;
IJ_20_3 = 0;
IJ_20_4 = 0;
IJ_20_5 = 0;
IJ_20_6 = 0;
IJ_20_7 = 0;
IJ_20_8=(cos(PSI_3).*cos(THETA_3).*cos(THETA_C)+(cos(THETA_3).*sin(PHI_C)).*sin( ...
PSI_3)+(-1).*cos(PHI_C).*sin(THETA_3)).*sin(THETA_C)).*(1+(-1).*cos( ...
PHI_C).*cos(THETA_C).*sin(THETA_3)+cos(THETA_3).*((-1).*cos(THETA_C)).* ...
sin(PHI_C)).*sin(PSI_3)+cos(PSI_3).*sin(THETA_C)).^2).^(-1/2);
IJ_20_9=(-1).*cos(THETA_C)).*(cos(PHI_C).*cos(THETA_3).*sin(PSI_3)+sin(PHI_C)).* ...
sin(THETA_3)).*(1+(-1).*cos(PHI_C).*cos(THETA_C).*sin(THETA_3)+cos( ...
THETA_3).*((-1).*cos(THETA_C)).*sin(PHI_C)).*sin(PSI_3)+cos(PSI_3)).*sin( ...
THETA_C)).^2).^(-1/2);
IJ_20_10 = 0;
IJ_20_11 = 0;
IJ_20_12 = 0;
IJ_20_13 = 0;
IJ_20_14 = 0;
IJ_20_15 = 0;
IJ_20_16 = 0;
IJ_20_17 = 0;
IJ_20_18 = 0;
IJ_20_19=(-1).*cos(THETA_3)).*(cos(PSI_3).*cos(THETA_C)).*sin(PHI_C)+sin(PSI_3)).* ...
sin(THETA_C)).*(1+(-1).*cos(PHI_C).*cos(THETA_C)).*sin(THETA_3)+cos( ...
THETA_3)).*((-1).*cos(THETA_C)).*sin(PHI_C)).*sin(PSI_3)+cos(PSI_3)).*sin( ...
THETA_C)).^2).^(-1/2);
IJ_20_20=(cos(PHI_C).*cos(THETA_3).*cos(THETA_C)+sin(THETA_3)).*(cos(THETA_C)).* ...
sin(PHI_C)).*sin(PSI_3)+(-1).*cos(PSI_3)).*sin(THETA_C)).*(1+(-1).*cos( ...
PHI_C).*cos(THETA_C)).*sin(THETA_3)+cos(THETA_3)).*((-1).*cos(THETA_C)).* ...
sin(PHI_C)).*sin(PSI_3)+cos(PSI_3)).*sin(THETA_C)).^2).^(-1/2);
IJ_20_21 = 0;
IJ_20_22 = 0;
IJ_20_23 = 0;
IJ_20_24 = 0;

IJ_21_1 = 0;
IJ_21_2 = 0;
IJ_21_3 = 0;
IJ_21_4 = 0;
IJ_21_5 = 0;
IJ_21_6 = 0;
IJ_21_7 = 0;
IJ_21_8=(cos(PHI_C).*cos(THETA_3)).*sin(PSI_3)+sin(PHI_C)).*sin(THETA_3)).*(cos( ...
PHI_C).^2.*cos(THETA_3).^2.*cos(THETA_C).^2+2.*cos(PHI_C).*cos(THETA_3) ...
.*cos(THETA_C)).*sin(THETA_3)).*(cos(THETA_C)).*sin(PHI_C)).*sin(PSI_3)+(-1) ...
.*cos(PSI_3)).*sin(THETA_C))+sin(PSI_3).^2.*(cos(THETA_C).^2.*sin(PHI_C) ...
.^2.*sin(THETA_3).^2+sin(THETA_C).^2)+cos(PSI_3).^2.*(cos(THETA_C).^2.* ...
sin(PHI_C).^2+sin(THETA_3).^2.*sin(THETA_C).^2)+cos(PSI_3).*cos(THETA_3) ...
.^2.*sin(PHI_C)).*sin(PSI_3)).*sin(2.*THETA_C)).^(-1);
IJ_21_9=(1/2)).*(2.*cos(PSI_3)).*cos(THETA_3)).*cos(THETA_C).^2+(-2).*cos(PHI_C)).* ...
cos(THETA_C)).*sin(THETA_3)).*sin(THETA_C)+cos(THETA_3)).*sin(PHI_C)).*sin( ...
PSI_3)).*sin(2.*THETA_C)).*(cos(PHI_C).^2.*cos(THETA_3).^2.*cos(THETA_C) ...
.^2+2.*cos(PHI_C)).*cos(THETA_3)).*cos(THETA_C)).*sin(THETA_3)).*(cos( ...
THETA_C)).*sin(PHI_C)).*sin(PSI_3)+(-1).*cos(PSI_3)).*sin(THETA_C))+sin( ...
PSI_3).^2.*(cos(THETA_C).^2.*sin(PHI_C).^2.*sin(THETA_3).^2+sin(THETA_C) ...
.^2)+cos(PSI_3).^2.*(cos(THETA_C).^2.*sin(PHI_C).^2+sin(THETA_3).^2.* ...

```

```

sin(THETA_C).^2+cos(PHI_3).*cos(THETA_3).^2.*sin(PHI_C).*sin(PHI_3).* ...
sin(2.*THETA_C)).^(-1);
IJ_21_10 = 0;
IJ_21_11 = 0;
IJ_21_12 = 0;
IJ_21_13 = 0;
IJ_21_14 = 0;
IJ_21_15 = 0;
IJ_21_16 = 0;
IJ_21_17 = 0;
IJ_21_18 = 0;
IJ_21_19=(1/4).*cos(PHI_C).^2.*cos(THETA_3).^2.*cos(THETA_C).^2+2.*cos(PHI_C).* ...
cos(THETA_3).*cos(THETA_C).*sin(THETA_3).*cos(THETA_C).*sin(PHI_C).* ...
sin(PHI_3)+(-1).*cos(PHI_3).*sin(THETA_C))+sin(PHI_3).^2.*(cos(THETA_C) ...
.^2.*sin(PHI_C).^2.*sin(THETA_3).^2+sin(THETA_C).^2)+cos(PHI_3).^2.*( ...
cos(THETA_C).^2.*sin(PHI_C).^2+sin(THETA_3).^2.*sin(THETA_C).^2)+cos( ...
PHI_3).*cos(THETA_3).^2.*sin(PHI_C).*sin(PHI_3).*sin(2.*THETA_C)).^(-1) ...
.*((-3)+cos(2.*PHI_C)+2.*cos(PHI_C).^2.*cos(2.*THETA_C)).*sin(THETA_3)+ ...
cos(THETA_3).*((-2).*cos(THETA_C).^2.*sin(2.*PHI_C).*sin(PHI_3)+2.*cos( ...
PHI_C).*cos(PHI_3).*sin(2.*THETA_C)));
IJ_21_20=(cos(PHI_3).*cos(THETA_C).*sin(PHI_C)+sin(PHI_3).*sin(THETA_C)).*(cos( ...
PHI_C).*cos(THETA_C).*sin(THETA_3)+cos(THETA_3).*((-1).*cos(THETA_C).* ...
sin(PHI_C).*sin(PHI_3)+cos(PHI_3).*sin(THETA_C))).*(cos(PHI_C).^2.*cos( ...
THETA_3).^2.*cos(THETA_C).^2+2.*cos(PHI_C).*cos(THETA_3).*cos(THETA_C).* ...
sin(THETA_3).*cos(THETA_C).*sin(PHI_C).*sin(PHI_3)+(-1).*cos(PHI_3).* ...
sin(THETA_C))+sin(PHI_3).^2.*(cos(THETA_C).^2.*sin(PHI_C).^2.*sin( ...
THETA_3).^2+sin(THETA_C).^2)+cos(PHI_3).^2.*(cos(THETA_C).^2.*sin(PHI_C) ...
.^2+sin(THETA_3).^2.*sin(THETA_C).^2)+cos(PHI_3).*cos(THETA_3).^2.*sin( ...
PHI_C).*sin(PHI_3).*sin(2.*THETA_C)).^(-1);
IJ_21_21 = 1;
IJ_21_22 = 0;
IJ_21_23 = 0;
IJ_21_24 = 0;

IJ_22_1 = 0;
IJ_22_2 = 0;
IJ_22_3 = 0;
IJ_22_4 = 0;
IJ_22_5 = 0;
IJ_22_6 = 0;
IJ_22_7 = 1;
IJ_22_8=(cos(PHI_C).*cos(THETA_4).*sin(PHI_4)+sin(PHI_C).*sin(THETA_4)).*(cos( ...
PHI_C).*cos(THETA_C).*sin(THETA_4)+cos(THETA_4).*((-1).*cos(THETA_C).* ...
sin(PHI_C).*sin(PHI_4)+cos(PHI_4).*sin(THETA_C))).*(cos(PHI_4).^2.*cos( ...
THETA_4).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C).* ...
sin(PHI_4).*sin(2.*THETA_4)+2.*cos(PHI_4).*cos(THETA_4).*cos(THETA_C).* ...
cos(THETA_4).*sin(PHI_C).*sin(PHI_4)+(-1).*cos(PHI_C).*sin(THETA_4)).* ...
sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_4).^2+cos(THETA_4).^2.*sin(PHI_4) ...
.^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_4).^2.*sin(PHI_4).^2+ ...
sin(THETA_4).^2.*sin(THETA_C).^2)).^(-1);
IJ_22_9=(1/4).*cos(PHI_4).*cos(THETA_4).*cos(THETA_C).*((-1).*cos(THETA_4).* ...
sin(PHI_C).*sin(PHI_4)+cos(PHI_C).*sin(THETA_4))+((-3)+cos(2.*PHI_4)+2.* ...
cos(PHI_4).^2.*cos(2.*THETA_4)).*sin(THETA_C)).*(cos(PHI_4).^2.*cos( ...
THETA_4).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C).* ...
sin(PHI_4).*sin(2.*THETA_4)+2.*cos(PHI_4).*cos(THETA_4).*cos(THETA_C).* ...
cos(THETA_4).*sin(PHI_C).*sin(PHI_4)+(-1).*cos(PHI_C).*sin(THETA_4)).* ...
sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_4).^2+cos(THETA_4).^2.*sin(PHI_4) ...
.^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_4).^2.*sin(PHI_4).^2+ ...
sin(THETA_4).^2.*sin(THETA_C).^2)).^(-1);
IJ_22_10 = 0;
IJ_22_11 = 0;
IJ_22_12 = 0;
IJ_22_13 = 0;
IJ_22_14 = 0;
IJ_22_15 = 0;
IJ_22_16 = 0;
IJ_22_17 = 0;
IJ_22_18 = 0;
IJ_22_19 = 0;
IJ_22_20 = 0;
IJ_22_21 = 0;

```

```

IJ_22_22=(1/2).*(2.*cos(PHI_C).*cos(THETA_4).^2.*cos(THETA_C)+cos(THETA_C).*sin( ...
PHI_C).*sin(PSI_4).*sin(2.*THETA_4)+(-2).*cos(PSI_4).*cos(THETA_4).*sin( ...
THETA_4).*sin(THETA_C)).*(cos(PSI_4).^2.*cos(THETA_4).^2.*cos(THETA_C) ...
.^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C).*sin(PSI_4).*sin(2.*THETA_4) ...
+2.*cos(PSI_4).*cos(THETA_4).*cos(THETA_C).*(cos(THETA_4).*sin(PHI_C).* ...
sin(PSI_4)+(-1).*cos(PHI_C).*sin(THETA_4)).*sin(THETA_C)+sin(PHI_C).^2.* ...
(sin(THETA_4).^2+cos(THETA_4).^2.*sin(PSI_4).^2.*sin(THETA_C).^2)+cos( ...
PHI_C).^2.*(cos(THETA_4).^2.*sin(PSI_4).^2+sin(THETA_4).^2.*sin(THETA_C) ...
.^2)).^(-1);
IJ_22_23=(cos(PSI_4).*cos(THETA_C).*sin(PHI_C)+sin(PSI_4).*sin(THETA_C)).*(cos( ...
PSI_4).^2.*cos(THETA_4).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C) ...
.^2.*sin(PHI_C).*sin(PSI_4).*sin(2.*THETA_4)+2.*cos(PSI_4).*cos(THETA_4) ...
.*cos(THETA_C).*(cos(THETA_4).*sin(PHI_C).*sin(PSI_4)+(-1).*cos(PHI_C).* ...
sin(THETA_4)).*sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_4).^2+cos(THETA_4) ...
.^2.*sin(PSI_4).^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_4).^2.* ...
sin(PSI_4).^2+sin(THETA_4).^2.*sin(THETA_C).^2)).^(-1);
IJ_22_24 = 0;

IJ_23_1 = 0;
IJ_23_2 = 0;
IJ_23_3 = 0;
IJ_23_4 = 0;
IJ_23_5 = 0;
IJ_23_6 = 0;
IJ_23_7 = 0;
IJ_23_8=(cos(PSI_4).*cos(THETA_4).*cos(THETA_C)+(cos(THETA_4).*sin(PHI_C).*sin( ...
PSI_4)+(-1).*cos(PHI_C).*sin(THETA_4)).*sin(THETA_C)).*(1+(-1).*(cos( ...
PHI_C).*cos(THETA_C).*sin(THETA_4)+cos(THETA_4).*((-1).*cos(THETA_C).* ...
sin(PHI_C).*sin(PSI_4)+cos(PSI_4).*sin(THETA_C))).^2).^(-1/2);
IJ_23_9=(-1).*cos(THETA_C).*(cos(PHI_C).*cos(THETA_4).*sin(PSI_4)+sin(PHI_C).* ...
sin(THETA_4)).*(1+(-1).*(cos(PHI_C).*cos(THETA_C).*sin(THETA_4)+cos( ...
THETA_4).*((-1).*cos(THETA_C).*sin(PHI_C).*sin(PSI_4)+cos(PSI_4).*sin( ...
THETA_C))).^2).^(-1/2);
IJ_23_10 = 0;
IJ_23_11 = 0;
IJ_23_12 = 0;
IJ_23_13 = 0;
IJ_23_14 = 0;
IJ_23_15 = 0;
IJ_23_16 = 0;
IJ_23_17 = 0;
IJ_23_18 = 0;
IJ_23_19 = 0;
IJ_23_20 = 0;
IJ_23_21 = 0;
IJ_23_22=(-1).*cos(THETA_4).*(cos(PSI_4).*cos(THETA_C).*sin(PHI_C)+sin(PSI_4).* ...
sin(THETA_C)).*(1+(-1).*(cos(PHI_C).*cos(THETA_C).*sin(THETA_4)+cos( ...
THETA_4).*((-1).*cos(THETA_C).*sin(PHI_C).*sin(PSI_4)+cos(PSI_4).*sin( ...
THETA_C))).^2).^(-1/2);
IJ_23_23=(cos(PHI_C).*cos(THETA_4).*cos(THETA_C)+sin(THETA_4).*(cos(THETA_C).* ...
sin(PHI_C).*sin(PSI_4)+(-1).*cos(PSI_4).*sin(THETA_C))).*(1+(-1).*(cos( ...
PHI_C).*cos(THETA_C).*sin(THETA_4)+cos(THETA_4).*((-1).*cos(THETA_C).* ...
sin(PHI_C).*sin(PSI_4)+cos(PSI_4).*sin(THETA_C))).^2).^(-1/2);
IJ_23_24 = 0;

IJ_24_1 = 0;
IJ_24_2 = 0;
IJ_24_3 = 0;
IJ_24_4 = 0;
IJ_24_5 = 0;
IJ_24_6 = 0;
IJ_24_7 = 0;
IJ_24_8=(cos(PHI_C).*cos(THETA_4).*sin(PSI_4)+sin(PHI_C).*sin(THETA_4)).*(cos( ...
PHI_C).^2.*cos(THETA_4).^2.*cos(THETA_C).^2+2.*cos(PHI_C).*cos(THETA_4) ...
.*cos(THETA_C).*sin(THETA_4).*(cos(THETA_C).*sin(PHI_C).*sin(PSI_4)+(-1) ...
.*cos(PSI_4).*sin(THETA_C))+sin(PSI_4).^2.*(cos(THETA_C).^2.*sin(PHI_C) ...
.^2.*sin(THETA_4).^2+sin(THETA_C).^2)+cos(PSI_4).^2.*(cos(THETA_C).^2.* ...
sin(PHI_C).^2+sin(THETA_4).^2.*sin(THETA_C).^2)+cos(PSI_4).*cos(THETA_4) ...
.^2.*sin(PHI_C).*sin(PSI_4).*sin(2.*THETA_C)).^(-1);
IJ_24_9=(1/2).*(2.*cos(PSI_4).*cos(THETA_4).*cos(THETA_C).^2+(-2).*cos(PHI_C).* ...
cos(THETA_C).*sin(THETA_4).*sin(THETA_C)+cos(THETA_4).*sin(PHI_C).*sin( ...

```



```

PSI_4).*sin(2.*THETA_C)).*(cos(PHI_C).^2.*cos(THETA_4).^2.*cos(THETA_C) ...
.^2+2.*cos(PHI_C).*cos(THETA_4).*cos(THETA_C).*sin(THETA_4).*(cos( ...
THETA_C).*sin(PHI_C).*sin(PSI_4)+(-1).*cos(PSI_4).*sin(THETA_C))+sin( ...
PSI_4).^2.*(cos(THETA_C).^2.*sin(PHI_C).^2.*sin(THETA_4).^2+sin(THETA_C) ...
.^2)+cos(PSI_4).^2.*(cos(THETA_C).^2.*sin(PHI_C).^2+sin(THETA_4).^2.* ...
sin(THETA_C).^2)+cos(PSI_4).*cos(THETA_4).^2.*sin(PHI_C).*sin(PSI_4).* ...
sin(2.*THETA_C)).^(-1);
IJ_24_10 = 0;
IJ_24_11 = 0;
IJ_24_12 = 0;
IJ_24_13 = 0;
IJ_24_14 = 0;
IJ_24_15 = 0;
IJ_24_16 = 0;
IJ_24_17 = 0;
IJ_24_18 = 0;
IJ_24_19 = 0;
IJ_24_20 = 0;
IJ_24_21 = 0;
IJ_24_22=(1/4).* (cos(PHI_C).^2.*cos(THETA_4).^2.*cos(THETA_C).^2+2.*cos(PHI_C).* ...
cos(THETA_4).*cos(THETA_C).*sin(THETA_4).*(cos(THETA_C).*sin(PHI_C).* ...
sin(PSI_4)+(-1).*cos(PSI_4).*sin(THETA_C))+sin(PSI_4).^2.*(cos(THETA_C) ...
.^2.*sin(PHI_C).^2.*sin(THETA_4).^2+sin(THETA_C).^2)+cos(PSI_4).^2.*( ...
cos(THETA_C).^2.*sin(PHI_C).^2+sin(THETA_4).^2.*sin(THETA_C).^2)+cos( ...
PSI_4).*cos(THETA_4).^2.*sin(PHI_C).*sin(PSI_4).*sin(2.*THETA_C)).^(-1) ...
.*((-3)+cos(2.*PHI_C)+2.*cos(PHI_C).^2.*cos(2.*THETA_C)).*sin(THETA_4)+ ...
cos(THETA_4).*((-2).*cos(THETA_C).^2.*sin(2.*PHI_C).*sin(PSI_4)+2.*cos( ...
PHI_C).*cos(PSI_4).*sin(2.*THETA_C));
IJ_24_23=(cos(PSI_4).*cos(THETA_C).*sin(PHI_C)+sin(PSI_4).*sin(THETA_C)).*(cos( ...
PHI_C).*cos(THETA_C).*sin(THETA_4)+cos(THETA_4).*((-1).*cos(THETA_C)).* ...
sin(PHI_C).*sin(PSI_4)+cos(PSI_4).*sin(THETA_C)).*(cos(PHI_C).^2.*cos( ...
THETA_4).^2.*cos(THETA_C).^2+2.*cos(PHI_C).*cos(THETA_4).*cos(THETA_C).* ...
sin(THETA_4).*(cos(THETA_C).*sin(PHI_C).*sin(PSI_4)+(-1).*cos(PSI_4)).* ...
sin(THETA_C))+sin(PSI_4).^2.*(cos(THETA_C).^2.*sin(PHI_C).^2.*sin( ...
THETA_4).^2+sin(THETA_C).^2)+cos(PSI_4).^2.*(cos(THETA_C).^2.*sin(PHI_C) ...
.^2+sin(THETA_4).^2.*sin(THETA_C).^2)+cos(PSI_4).*cos(THETA_4).^2.*sin( ...
PHI_C).*sin(PSI_4).*sin(2.*THETA_C)).^(-1);
IJ_24_24 = 1;

```

APPENDIX F

(Inverse Jacobian Matrix for Nine-Robot Cluster as Implemented in Simulation)

(This section contains 64 pages)

Velocity Mapping from Cluster-Space to Robot-Space as Implemented in Simulation

Robot-Space Velocities	Cluster-Space Velocities
$\begin{bmatrix} G\dot{x}_1 \\ G\dot{y}_1 \\ G\dot{z}_1 \\ G\dot{x}_2 \\ G\dot{y}_2 \\ G\dot{z}_2 \\ G\dot{x}_3 \\ G\dot{y}_3 \\ G\dot{z}_3 \\ G\dot{x}_4 \\ G\dot{y}_4 \\ G\dot{z}_4 \\ G\dot{x}_5 \\ G\dot{y}_5 \\ G\dot{z}_5 \\ G\dot{x}_6 \\ G\dot{y}_6 \\ G\dot{z}_6 \\ G\dot{x}_7 \\ G\dot{y}_7 \\ G\dot{z}_7 \\ G\dot{x}_8 \\ G\dot{y}_8 \\ G\dot{z}_8 \\ G\dot{x}_9 \\ G\dot{y}_9 \\ G\dot{z}_9 \\ \vdots \end{bmatrix}$	$\begin{bmatrix} G\dot{X}_C \\ G\dot{Y}_C \\ G\dot{Z}_C \\ G\dot{\Psi}_C \\ G\dot{\Theta}_C \\ G\dot{\Phi}_C \\ C\dot{\Psi}_1 \\ C\dot{\Theta}_1 \\ C\dot{\Phi}_1 \\ C\dot{\Psi}_2 \\ C\dot{\Theta}_2 \\ C\dot{\Phi}_2 \\ C\dot{\Psi}_3 \\ C\dot{\Theta}_3 \\ C\dot{\Phi}_3 \\ C\dot{\Psi}_4 \\ C\dot{\Theta}_4 \\ C\dot{\Phi}_4 \\ C\dot{\Psi}_5 \\ C\dot{\Theta}_5 \\ C\dot{\Phi}_5 \\ C\dot{\Psi}_6 \\ C\dot{\Theta}_6 \\ C\dot{\Phi}_6 \\ C\dot{\Psi}_7 \\ C\dot{\Theta}_7 \\ C\dot{\Phi}_7 \\ \vdots \end{bmatrix}$
$= J^{-1} \cdot$	
$\begin{bmatrix} \vdots \\ G\dot{\Psi}_1 \\ G\dot{\Theta}_1 \\ G\dot{\Phi}_1 \\ G\dot{\Psi}_2 \\ G\dot{\Theta}_2 \\ G\dot{\Phi}_2 \\ G\dot{\Psi}_3 \\ G\dot{\Theta}_3 \\ G\dot{\Phi}_3 \\ G\dot{\Psi}_4 \\ G\dot{\Theta}_4 \\ G\dot{\Phi}_4 \\ G\dot{\Psi}_5 \\ G\dot{\Theta}_5 \\ G\dot{\Phi}_5 \\ G\dot{\Psi}_6 \\ G\dot{\Theta}_6 \\ G\dot{\Phi}_6 \\ G\dot{\Psi}_7 \\ G\dot{\Theta}_7 \\ G\dot{\Phi}_7 \\ G\dot{\Psi}_8 \\ G\dot{\Theta}_8 \\ G\dot{\Phi}_8 \\ G\dot{\Psi}_9 \\ G\dot{\Theta}_9 \\ G\dot{\Phi}_9 \end{bmatrix}$	$\begin{bmatrix} \vdots \\ C\dot{\Psi}_8 \\ C\dot{\Theta}_8 \\ C\dot{\Phi}_8 \\ C\dot{\Psi}_9 \\ C\dot{\Theta}_9 \\ C\dot{\Phi}_9 \\ \dot{L}_{C2} \\ \dot{L}_{23} \\ \dot{L}_{C4} \\ \dot{L}_{45} \\ \dot{L}_{C6} \\ \dot{L}_{67} \\ \dot{L}_{C8} \\ \dot{L}_{89} \\ \zeta \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \\ \alpha_7 \\ \alpha_9 \\ \beta_3 \\ \beta_4 \\ \beta_5 \\ \beta_6 \\ \beta_7 \\ \beta_9 \end{bmatrix}$

```
% Inverse Jacobian Matrix for Nine-Robot Cluster
```

```
IJ = [...  
IJ_1_1  IJ_1_2  IJ_1_3  IJ_1_4  IJ_1_5  IJ_1_6  IJ_1_7  IJ_1_8  IJ_1_9  IJ_1_10  
IJ_1_11 IJ_1_12...  
IJ_1_13 IJ_1_14 IJ_1_15 IJ_1_16 IJ_1_17 IJ_1_18 IJ_1_19 IJ_1_20 IJ_1_21 IJ_1_22  
IJ_1_23 IJ_1_24...  
IJ_1_25 IJ_1_26 IJ_1_27 IJ_1_28 IJ_1_29 IJ_1_30 IJ_1_31 IJ_1_32 IJ_1_33 IJ_1_34  
IJ_1_35 IJ_1_36...  
IJ_1_37 IJ_1_38 IJ_1_39 IJ_1_40 IJ_1_41 IJ_1_42 IJ_1_43 IJ_1_44 IJ_1_45 IJ_1_46  
IJ_1_47 IJ_1_48...  
IJ_1_49 IJ_1_50 IJ_1_51 IJ_1_52 IJ_1_53 IJ_1_54 ;...  
  
IJ_2_1  IJ_2_2  IJ_2_3  IJ_2_4  IJ_2_5  IJ_2_6  IJ_2_7  IJ_2_8  IJ_2_9  IJ_2_10  
IJ_2_11 IJ_2_12...  
IJ_2_13 IJ_2_14 IJ_2_15 IJ_2_16 IJ_2_17 IJ_2_18 IJ_2_19 IJ_2_20 IJ_2_21 IJ_2_22  
IJ_2_23 IJ_2_24...  
IJ_2_25 IJ_2_26 IJ_2_27 IJ_2_28 IJ_2_29 IJ_2_30 IJ_2_31 IJ_2_32 IJ_2_33 IJ_2_34  
IJ_2_35 IJ_2_36...  
IJ_2_37 IJ_2_38 IJ_2_39 IJ_2_40 IJ_2_41 IJ_2_42 IJ_2_43 IJ_2_44 IJ_2_45 IJ_2_46  
IJ_2_47 IJ_2_48...  
IJ_2_49 IJ_2_50 IJ_2_51 IJ_2_52 IJ_2_53 IJ_2_54 ;...  
  
IJ_3_1  IJ_3_2  IJ_3_3  IJ_3_4  IJ_3_5  IJ_3_6  IJ_3_7  IJ_3_8  IJ_3_9  IJ_3_10  
IJ_3_11 IJ_3_12...  
IJ_3_13 IJ_3_14 IJ_3_15 IJ_3_16 IJ_3_17 IJ_3_18 IJ_3_19 IJ_3_20 IJ_3_21 IJ_3_22  
IJ_3_23 IJ_3_24...  
IJ_3_25 IJ_3_26 IJ_3_27 IJ_3_28 IJ_3_29 IJ_3_30 IJ_3_31 IJ_3_32 IJ_3_33 IJ_3_34  
IJ_3_35 IJ_3_36...  
IJ_3_37 IJ_3_38 IJ_3_39 IJ_3_40 IJ_3_41 IJ_3_42 IJ_3_43 IJ_3_44 IJ_3_45 IJ_3_46  
IJ_3_47 IJ_3_48...  
IJ_3_49 IJ_3_50 IJ_3_51 IJ_3_52 IJ_3_53 IJ_3_54 ;...  
  
IJ_4_1  IJ_4_2  IJ_4_3  IJ_4_4  IJ_4_5  IJ_4_6  IJ_4_7  IJ_4_8  IJ_4_9  IJ_4_10  
IJ_4_11 IJ_4_12...  
IJ_4_13 IJ_4_14 IJ_4_15 IJ_4_16 IJ_4_17 IJ_4_18 IJ_4_19 IJ_4_20 IJ_4_21 IJ_4_22  
IJ_4_23 IJ_4_24...  
IJ_4_25 IJ_4_26 IJ_4_27 IJ_4_28 IJ_4_29 IJ_4_30 IJ_4_31 IJ_4_32 IJ_4_33 IJ_4_34  
IJ_4_35 IJ_4_36...  
IJ_4_37 IJ_4_38 IJ_4_39 IJ_4_40 IJ_4_41 IJ_4_42 IJ_4_43 IJ_4_44 IJ_4_45 IJ_4_46  
IJ_4_47 IJ_4_48...  
IJ_4_49 IJ_4_50 IJ_4_51 IJ_4_52 IJ_4_53 IJ_4_54 ;...  
  
IJ_5_1  IJ_5_2  IJ_5_3  IJ_5_4  IJ_5_5  IJ_5_6  IJ_5_7  IJ_5_8  IJ_5_9  IJ_5_10  
IJ_5_11 IJ_5_12...  
IJ_5_13 IJ_5_14 IJ_5_15 IJ_5_16 IJ_5_17 IJ_5_18 IJ_5_19 IJ_5_20 IJ_5_21 IJ_5_22  
IJ_5_23 IJ_5_24...  
IJ_5_25 IJ_5_26 IJ_5_27 IJ_5_28 IJ_5_29 IJ_5_30 IJ_5_31 IJ_5_32 IJ_5_33 IJ_5_34  
IJ_5_35 IJ_5_36...  
IJ_5_37 IJ_5_38 IJ_5_39 IJ_5_40 IJ_5_41 IJ_5_42 IJ_5_43 IJ_5_44 IJ_5_45 IJ_5_46  
IJ_5_47 IJ_5_48...  
IJ_5_49 IJ_5_50 IJ_5_51 IJ_5_52 IJ_5_53 IJ_5_54 ;...  
  
IJ_6_1  IJ_6_2  IJ_6_3  IJ_6_4  IJ_6_5  IJ_6_6  IJ_6_7  IJ_6_8  IJ_6_9  IJ_6_10  
IJ_6_11 IJ_6_12...  
IJ_6_13 IJ_6_14 IJ_6_15 IJ_6_16 IJ_6_17 IJ_6_18 IJ_6_19 IJ_6_20 IJ_6_21 IJ_6_22  
IJ_6_23 IJ_6_24...  
IJ_6_25 IJ_6_26 IJ_6_27 IJ_6_28 IJ_6_29 IJ_6_30 IJ_6_31 IJ_6_32 IJ_6_33 IJ_6_34  
IJ_6_35 IJ_6_36...  
IJ_6_37 IJ_6_38 IJ_6_39 IJ_6_40 IJ_6_41 IJ_6_42 IJ_6_43 IJ_6_44 IJ_6_45 IJ_6_46  
IJ_6_47 IJ_6_48...  
IJ_6_49 IJ_6_50 IJ_6_51 IJ_6_52 IJ_6_53 IJ_6_54 ;...  
  
IJ_7_1  IJ_7_2  IJ_7_3  IJ_7_4  IJ_7_5  IJ_7_6  IJ_7_7  IJ_7_8  IJ_7_9  IJ_7_10  
IJ_7_11 IJ_7_12...  
IJ_7_13 IJ_7_14 IJ_7_15 IJ_7_16 IJ_7_17 IJ_7_18 IJ_7_19 IJ_7_20 IJ_7_21 IJ_7_22  
IJ_7_23 IJ_7_24...  
IJ_7_25 IJ_7_26 IJ_7_27 IJ_7_28 IJ_7_29 IJ_7_30 IJ_7_31 IJ_7_32 IJ_7_33 IJ_7_34  
IJ_7_35 IJ_7_36...  
IJ_7_37 IJ_7_38 IJ_7_39 IJ_7_40 IJ_7_41 IJ_7_42 IJ_7_43 IJ_7_44 IJ_7_45 IJ_7_46  
IJ_7_47 IJ_7_48...
```

IJ_7_49 IJ_7_50 IJ_7_51 IJ_7_52 IJ_7_53 IJ_7_54 ;...
 IJ_8_1 IJ_8_2 IJ_8_3 IJ_8_4 IJ_8_5 IJ_8_6 IJ_8_7 IJ_8_8 IJ_8_9 IJ_8_10
 IJ_8_11 IJ_8_12...
 IJ_8_13 IJ_8_14 IJ_8_15 IJ_8_16 IJ_8_17 IJ_8_18 IJ_8_19 IJ_8_20 IJ_8_21 IJ_8_22
 IJ_8_23 IJ_8_24...
 IJ_8_25 IJ_8_26 IJ_8_27 IJ_8_28 IJ_8_29 IJ_8_30 IJ_8_31 IJ_8_32 IJ_8_33 IJ_8_34
 IJ_8_35 IJ_8_36...
 IJ_8_37 IJ_8_38 IJ_8_39 IJ_8_40 IJ_8_41 IJ_8_42 IJ_8_43 IJ_8_44 IJ_8_45 IJ_8_46
 IJ_8_47 IJ_8_48...
 IJ_8_49 IJ_8_50 IJ_8_51 IJ_8_52 IJ_8_53 IJ_8_54 ;...
 IJ_9_1 IJ_9_2 IJ_9_3 IJ_9_4 IJ_9_5 IJ_9_6 IJ_9_7 IJ_9_8 IJ_9_9 IJ_9_10
 IJ_9_11 IJ_9_12...
 IJ_9_13 IJ_9_14 IJ_9_15 IJ_9_16 IJ_9_17 IJ_9_18 IJ_9_19 IJ_9_20 IJ_9_21 IJ_9_22
 IJ_9_23 IJ_9_24...
 IJ_9_25 IJ_9_26 IJ_9_27 IJ_9_28 IJ_9_29 IJ_9_30 IJ_9_31 IJ_9_32 IJ_9_33 IJ_9_34
 IJ_9_35 IJ_9_36...
 IJ_9_37 IJ_9_38 IJ_9_39 IJ_9_40 IJ_9_41 IJ_9_42 IJ_9_43 IJ_9_44 IJ_9_45 IJ_9_46
 IJ_9_47 IJ_9_48...
 IJ_9_49 IJ_9_50 IJ_9_51 IJ_9_52 IJ_9_53 IJ_9_54 ;...
 IJ_10_1 IJ_10_2 IJ_10_3 IJ_10_4 IJ_10_5 IJ_10_6 IJ_10_7 IJ_10_8 IJ_10_9
 IJ_10_10 IJ_10_11 IJ_10_12...
 IJ_10_13 IJ_10_14 IJ_10_15 IJ_10_16 IJ_10_17 IJ_10_18 IJ_10_19 IJ_10_20 IJ_10_21
 IJ_10_22 IJ_10_23 IJ_10_24...
 IJ_10_25 IJ_10_26 IJ_10_27 IJ_10_28 IJ_10_29 IJ_10_30 IJ_10_31 IJ_10_32 IJ_10_33
 IJ_10_34 IJ_10_35 IJ_10_36...
 IJ_10_37 IJ_10_38 IJ_10_39 IJ_10_40 IJ_10_41 IJ_10_42 IJ_10_43 IJ_10_44 IJ_10_45
 IJ_10_46 IJ_10_47 IJ_10_48...
 IJ_10_49 IJ_10_50 IJ_10_51 IJ_10_52 IJ_10_53 IJ_10_54 ;...
 IJ_11_1 IJ_11_2 IJ_11_3 IJ_11_4 IJ_11_5 IJ_11_6 IJ_11_7 IJ_11_8 IJ_11_9
 IJ_11_10 IJ_11_11 IJ_11_12...
 IJ_11_13 IJ_11_14 IJ_11_15 IJ_11_16 IJ_11_17 IJ_11_18 IJ_11_19 IJ_11_20 IJ_11_21
 IJ_11_22 IJ_11_23 IJ_11_24...
 IJ_11_25 IJ_11_26 IJ_11_27 IJ_11_28 IJ_11_29 IJ_11_30 IJ_11_31 IJ_11_32 IJ_11_33
 IJ_11_34 IJ_11_35 IJ_11_36...
 IJ_11_37 IJ_11_38 IJ_11_39 IJ_11_40 IJ_11_41 IJ_11_42 IJ_11_43 IJ_11_44 IJ_11_45
 IJ_11_46 IJ_11_47 IJ_11_48...
 IJ_11_49 IJ_11_50 IJ_11_51 IJ_11_52 IJ_11_53 IJ_11_54 ;...
 IJ_12_1 IJ_12_2 IJ_12_3 IJ_12_4 IJ_12_5 IJ_12_6 IJ_12_7 IJ_12_8 IJ_12_9
 IJ_12_10 IJ_12_11 IJ_12_12...
 IJ_12_13 IJ_12_14 IJ_12_15 IJ_12_16 IJ_12_17 IJ_12_18 IJ_12_19 IJ_12_20 IJ_12_21
 IJ_12_22 IJ_12_23 IJ_12_24...
 IJ_12_25 IJ_12_26 IJ_12_27 IJ_12_28 IJ_12_29 IJ_12_30 IJ_12_31 IJ_12_32 IJ_12_33
 IJ_12_34 IJ_12_35 IJ_12_36...
 IJ_12_37 IJ_12_38 IJ_12_39 IJ_12_40 IJ_12_41 IJ_12_42 IJ_12_43 IJ_12_44 IJ_12_45
 IJ_12_46 IJ_12_47 IJ_12_48...
 IJ_12_49 IJ_12_50 IJ_12_51 IJ_12_52 IJ_12_53 IJ_12_54 ;...
 IJ_13_1 IJ_13_2 IJ_13_3 IJ_13_4 IJ_13_5 IJ_13_6 IJ_13_7 IJ_13_8 IJ_13_9
 IJ_13_10 IJ_13_11 IJ_13_12...
 IJ_13_13 IJ_13_14 IJ_13_15 IJ_13_16 IJ_13_17 IJ_13_18 IJ_13_19 IJ_13_20 IJ_13_21
 IJ_13_22 IJ_13_23 IJ_13_24...
 IJ_13_25 IJ_13_26 IJ_13_27 IJ_13_28 IJ_13_29 IJ_13_30 IJ_13_31 IJ_13_32 IJ_13_33
 IJ_13_34 IJ_13_35 IJ_13_36...
 IJ_13_37 IJ_13_38 IJ_13_39 IJ_13_40 IJ_13_41 IJ_13_42 IJ_13_43 IJ_13_44 IJ_13_45
 IJ_13_46 IJ_13_47 IJ_13_48...
 IJ_13_49 IJ_13_50 IJ_13_51 IJ_13_52 IJ_13_53 IJ_13_54 ;...
 IJ_14_1 IJ_14_2 IJ_14_3 IJ_14_4 IJ_14_5 IJ_14_6 IJ_14_7 IJ_14_8 IJ_14_9
 IJ_14_10 IJ_14_11 IJ_14_12...
 IJ_14_13 IJ_14_14 IJ_14_15 IJ_14_16 IJ_14_17 IJ_14_18 IJ_14_19 IJ_14_20 IJ_14_21
 IJ_14_22 IJ_14_23 IJ_14_24...
 IJ_14_25 IJ_14_26 IJ_14_27 IJ_14_28 IJ_14_29 IJ_14_30 IJ_14_31 IJ_14_32 IJ_14_33
 IJ_14_34 IJ_14_35 IJ_14_36...
 IJ_14_37 IJ_14_38 IJ_14_39 IJ_14_40 IJ_14_41 IJ_14_42 IJ_14_43 IJ_14_44 IJ_14_45
 IJ_14_46 IJ_14_47 IJ_14_48...
 IJ_14_49 IJ_14_50 IJ_14_51 IJ_14_52 IJ_14_53 IJ_14_54 ;...

IJ_15_1	IJ_15_2	IJ_15_3	IJ_15_4	IJ_15_5	IJ_15_6	IJ_15_7	IJ_15_8	IJ_15_9
IJ_15_10	IJ_15_11	IJ_15_12...						
IJ_15_13	IJ_15_14	IJ_15_15	IJ_15_16	IJ_15_17	IJ_15_18	IJ_15_19	IJ_15_20	IJ_15_21
IJ_15_22	IJ_15_23	IJ_15_24...						
IJ_15_25	IJ_15_26	IJ_15_27	IJ_15_28	IJ_15_29	IJ_15_30	IJ_15_31	IJ_15_32	IJ_15_33
IJ_15_34	IJ_15_35	IJ_15_36...						
IJ_15_37	IJ_15_38	IJ_15_39	IJ_15_40	IJ_15_41	IJ_15_42	IJ_15_43	IJ_15_44	IJ_15_45
IJ_15_46	IJ_15_47	IJ_15_48...						
IJ_15_49	IJ_15_50	IJ_15_51	IJ_15_52	IJ_15_53	IJ_15_54	;...		
IJ_16_1	IJ_16_2	IJ_16_3	IJ_16_4	IJ_16_5	IJ_16_6	IJ_16_7	IJ_16_8	IJ_16_9
IJ_16_10	IJ_16_11	IJ_16_12...						
IJ_16_13	IJ_16_14	IJ_16_15	IJ_16_16	IJ_16_17	IJ_16_18	IJ_16_19	IJ_16_20	IJ_16_21
IJ_16_22	IJ_16_23	IJ_16_24...						
IJ_16_25	IJ_16_26	IJ_16_27	IJ_16_28	IJ_16_29	IJ_16_30	IJ_16_31	IJ_16_32	IJ_16_33
IJ_16_34	IJ_16_35	IJ_16_36...						
IJ_16_37	IJ_16_38	IJ_16_39	IJ_16_40	IJ_16_41	IJ_16_42	IJ_16_43	IJ_16_44	IJ_16_45
IJ_16_46	IJ_16_47	IJ_16_48...						
IJ_16_49	IJ_16_50	IJ_16_51	IJ_16_52	IJ_16_53	IJ_16_54	;...		
IJ_17_1	IJ_17_2	IJ_17_3	IJ_17_4	IJ_17_5	IJ_17_6	IJ_17_7	IJ_17_8	IJ_17_9
IJ_17_10	IJ_17_11	IJ_17_12...						
IJ_17_13	IJ_17_14	IJ_17_15	IJ_17_16	IJ_17_17	IJ_17_18	IJ_17_19	IJ_17_20	IJ_17_21
IJ_17_22	IJ_17_23	IJ_17_24...						
IJ_17_25	IJ_17_26	IJ_17_27	IJ_17_28	IJ_17_29	IJ_17_30	IJ_17_31	IJ_17_32	IJ_17_33
IJ_17_34	IJ_17_35	IJ_17_36...						
IJ_17_37	IJ_17_38	IJ_17_39	IJ_17_40	IJ_17_41	IJ_17_42	IJ_17_43	IJ_17_44	IJ_17_45
IJ_17_46	IJ_17_47	IJ_17_48...						
IJ_17_49	IJ_17_50	IJ_17_51	IJ_17_52	IJ_17_53	IJ_17_54	;...		
IJ_18_1	IJ_18_2	IJ_18_3	IJ_18_4	IJ_18_5	IJ_18_6	IJ_18_7	IJ_18_8	IJ_18_9
IJ_18_10	IJ_18_11	IJ_18_12...						
IJ_18_13	IJ_18_14	IJ_18_15	IJ_18_16	IJ_18_17	IJ_18_18	IJ_18_19	IJ_18_20	IJ_18_21
IJ_18_22	IJ_18_23	IJ_18_24...						
IJ_18_25	IJ_18_26	IJ_18_27	IJ_18_28	IJ_18_29	IJ_18_30	IJ_18_31	IJ_18_32	IJ_18_33
IJ_18_34	IJ_18_35	IJ_18_36...						
IJ_18_37	IJ_18_38	IJ_18_39	IJ_18_40	IJ_18_41	IJ_18_42	IJ_18_43	IJ_18_44	IJ_18_45
IJ_18_46	IJ_18_47	IJ_18_48...						
IJ_18_49	IJ_18_50	IJ_18_51	IJ_18_52	IJ_18_53	IJ_18_54	;...		
IJ_19_1	IJ_19_2	IJ_19_3	IJ_19_4	IJ_19_5	IJ_19_6	IJ_19_7	IJ_19_8	IJ_19_9
IJ_19_10	IJ_19_11	IJ_19_12...						
IJ_19_13	IJ_19_14	IJ_19_15	IJ_19_16	IJ_19_17	IJ_19_18	IJ_19_19	IJ_19_20	IJ_19_21
IJ_19_22	IJ_19_23	IJ_19_24...						
IJ_19_25	IJ_19_26	IJ_19_27	IJ_19_28	IJ_19_29	IJ_19_30	IJ_19_31	IJ_19_32	IJ_19_33
IJ_19_34	IJ_19_35	IJ_19_36...						
IJ_19_37	IJ_19_38	IJ_19_39	IJ_19_40	IJ_19_41	IJ_19_42	IJ_19_43	IJ_19_44	IJ_19_45
IJ_19_46	IJ_19_47	IJ_19_48...						
IJ_19_49	IJ_19_50	IJ_19_51	IJ_19_52	IJ_19_53	IJ_19_54	;...		
IJ_20_1	IJ_20_2	IJ_20_3	IJ_20_4	IJ_20_5	IJ_20_6	IJ_20_7	IJ_20_8	IJ_20_9
IJ_20_10	IJ_20_11	IJ_20_12...						
IJ_20_13	IJ_20_14	IJ_20_15	IJ_20_16	IJ_20_17	IJ_20_18	IJ_20_19	IJ_20_20	IJ_20_21
IJ_20_22	IJ_20_23	IJ_20_24...						
IJ_20_25	IJ_20_26	IJ_20_27	IJ_20_28	IJ_20_29	IJ_20_30	IJ_20_31	IJ_20_32	IJ_20_33
IJ_20_34	IJ_20_35	IJ_20_36...						
IJ_20_37	IJ_20_38	IJ_20_39	IJ_20_40	IJ_20_41	IJ_20_42	IJ_20_43	IJ_20_44	IJ_20_45
IJ_20_46	IJ_20_47	IJ_20_48...						
IJ_20_49	IJ_20_50	IJ_20_51	IJ_20_52	IJ_20_53	IJ_20_54	;...		
IJ_21_1	IJ_21_2	IJ_21_3	IJ_21_4	IJ_21_5	IJ_21_6	IJ_21_7	IJ_21_8	IJ_21_9
IJ_21_10	IJ_21_11	IJ_21_12...						
IJ_21_13	IJ_21_14	IJ_21_15	IJ_21_16	IJ_21_17	IJ_21_18	IJ_21_19	IJ_21_20	IJ_21_21
IJ_21_22	IJ_21_23	IJ_21_24...						
IJ_21_25	IJ_21_26	IJ_21_27	IJ_21_28	IJ_21_29	IJ_21_30	IJ_21_31	IJ_21_32	IJ_21_33
IJ_21_34	IJ_21_35	IJ_21_36...						
IJ_21_37	IJ_21_38	IJ_21_39	IJ_21_40	IJ_21_41	IJ_21_42	IJ_21_43	IJ_21_44	IJ_21_45
IJ_21_46	IJ_21_47	IJ_21_48...						
IJ_21_49	IJ_21_50	IJ_21_51	IJ_21_52	IJ_21_53	IJ_21_54	;...		

IJ_22_1	IJ_22_2	IJ_22_3	IJ_22_4	IJ_22_5	IJ_22_6	IJ_22_7	IJ_22_8	IJ_22_9
IJ_22_10	IJ_22_11	IJ_22_12...						
IJ_22_13	IJ_22_14	IJ_22_15	IJ_22_16	IJ_22_17	IJ_22_18	IJ_22_19	IJ_22_20	IJ_22_21
IJ_22_22	IJ_22_23	IJ_22_24...						
IJ_22_25	IJ_22_26	IJ_22_27	IJ_22_28	IJ_22_29	IJ_22_30	IJ_22_31	IJ_22_32	IJ_22_33
IJ_22_34	IJ_22_35	IJ_22_36...						
IJ_22_37	IJ_22_38	IJ_22_39	IJ_22_40	IJ_22_41	IJ_22_42	IJ_22_43	IJ_22_44	IJ_22_45
IJ_22_46	IJ_22_47	IJ_22_48...						
IJ_22_49	IJ_22_50	IJ_22_51	IJ_22_52	IJ_22_53	IJ_22_54	;...		
IJ_23_1	IJ_23_2	IJ_23_3	IJ_23_4	IJ_23_5	IJ_23_6	IJ_23_7	IJ_23_8	IJ_23_9
IJ_23_10	IJ_23_11	IJ_23_12...						
IJ_23_13	IJ_23_14	IJ_23_15	IJ_23_16	IJ_23_17	IJ_23_18	IJ_23_19	IJ_23_20	IJ_23_21
IJ_23_22	IJ_23_23	IJ_23_24...						
IJ_23_25	IJ_23_26	IJ_23_27	IJ_23_28	IJ_23_29	IJ_23_30	IJ_23_31	IJ_23_32	IJ_23_33
IJ_23_34	IJ_23_35	IJ_23_36...						
IJ_23_37	IJ_23_38	IJ_23_39	IJ_23_40	IJ_23_41	IJ_23_42	IJ_23_43	IJ_23_44	IJ_23_45
IJ_23_46	IJ_23_47	IJ_23_48...						
IJ_23_49	IJ_23_50	IJ_23_51	IJ_23_52	IJ_23_53	IJ_23_54	;...		
IJ_24_1	IJ_24_2	IJ_24_3	IJ_24_4	IJ_24_5	IJ_24_6	IJ_24_7	IJ_24_8	IJ_24_9
IJ_24_10	IJ_24_11	IJ_24_12...						
IJ_24_13	IJ_24_14	IJ_24_15	IJ_24_16	IJ_24_17	IJ_24_18	IJ_24_19	IJ_24_20	IJ_24_21
IJ_24_22	IJ_24_23	IJ_24_24...						
IJ_24_25	IJ_24_26	IJ_24_27	IJ_24_28	IJ_24_29	IJ_24_30	IJ_24_31	IJ_24_32	IJ_24_33
IJ_24_34	IJ_24_35	IJ_24_36...						
IJ_24_37	IJ_24_38	IJ_24_39	IJ_24_40	IJ_24_41	IJ_24_42	IJ_24_43	IJ_24_44	IJ_24_45
IJ_24_46	IJ_24_47	IJ_24_48...						
IJ_24_49	IJ_24_50	IJ_24_51	IJ_24_52	IJ_24_53	IJ_24_54	;...		
IJ_25_1	IJ_25_2	IJ_25_3	IJ_25_4	IJ_25_5	IJ_25_6	IJ_25_7	IJ_25_8	IJ_25_9
IJ_25_10	IJ_25_11	IJ_25_12...						
IJ_25_13	IJ_25_14	IJ_25_15	IJ_25_16	IJ_25_17	IJ_25_18	IJ_25_19	IJ_25_20	IJ_25_21
IJ_25_22	IJ_25_23	IJ_25_24...						
IJ_25_25	IJ_25_26	IJ_25_27	IJ_25_28	IJ_25_29	IJ_25_30	IJ_25_31	IJ_25_32	IJ_25_33
IJ_25_34	IJ_25_35	IJ_25_36...						
IJ_25_37	IJ_25_38	IJ_25_39	IJ_25_40	IJ_25_41	IJ_25_42	IJ_25_43	IJ_25_44	IJ_25_45
IJ_25_46	IJ_25_47	IJ_25_48...						
IJ_25_49	IJ_25_50	IJ_25_51	IJ_25_52	IJ_25_53	IJ_25_54	;...		
IJ_26_1	IJ_26_2	IJ_26_3	IJ_26_4	IJ_26_5	IJ_26_6	IJ_26_7	IJ_26_8	IJ_26_9
IJ_26_10	IJ_26_11	IJ_26_12...						
IJ_26_13	IJ_26_14	IJ_26_15	IJ_26_16	IJ_26_17	IJ_26_18	IJ_26_19	IJ_26_20	IJ_26_21
IJ_26_22	IJ_26_23	IJ_26_24...						
IJ_26_25	IJ_26_26	IJ_26_27	IJ_26_28	IJ_26_29	IJ_26_30	IJ_26_31	IJ_26_32	IJ_26_33
IJ_26_34	IJ_26_35	IJ_26_36...						
IJ_26_37	IJ_26_38	IJ_26_39	IJ_26_40	IJ_26_41	IJ_26_42	IJ_26_43	IJ_26_44	IJ_26_45
IJ_26_46	IJ_26_47	IJ_26_48...						
IJ_26_49	IJ_26_50	IJ_26_51	IJ_26_52	IJ_26_53	IJ_26_54	;...		
IJ_27_1	IJ_27_2	IJ_27_3	IJ_27_4	IJ_27_5	IJ_27_6	IJ_27_7	IJ_27_8	IJ_27_9
IJ_27_10	IJ_27_11	IJ_27_12...						
IJ_27_13	IJ_27_14	IJ_27_15	IJ_27_16	IJ_27_17	IJ_27_18	IJ_27_19	IJ_27_20	IJ_27_21
IJ_27_22	IJ_27_23	IJ_27_24...						
IJ_27_25	IJ_27_26	IJ_27_27	IJ_27_28	IJ_27_29	IJ_27_30	IJ_27_31	IJ_27_32	IJ_27_33
IJ_27_34	IJ_27_35	IJ_27_36...						
IJ_27_37	IJ_27_38	IJ_27_39	IJ_27_40	IJ_27_41	IJ_27_42	IJ_27_43	IJ_27_44	IJ_27_45
IJ_27_46	IJ_27_47	IJ_27_48...						
IJ_27_49	IJ_27_50	IJ_27_51	IJ_27_52	IJ_27_53	IJ_27_54	;...		
IJ_28_1	IJ_28_2	IJ_28_3	IJ_28_4	IJ_28_5	IJ_28_6	IJ_28_7	IJ_28_8	IJ_28_9
IJ_28_10	IJ_28_11	IJ_28_12...						
IJ_28_13	IJ_28_14	IJ_28_15	IJ_28_16	IJ_28_17	IJ_28_18	IJ_28_19	IJ_28_20	IJ_28_21
IJ_28_22	IJ_28_23	IJ_28_24...						
IJ_28_25	IJ_28_26	IJ_28_27	IJ_28_28	IJ_28_29	IJ_28_30	IJ_28_31	IJ_28_32	IJ_28_33
IJ_28_34	IJ_28_35	IJ_28_36...						
IJ_28_37	IJ_28_38	IJ_28_39	IJ_28_40	IJ_28_41	IJ_28_42	IJ_28_43	IJ_28_44	IJ_28_45
IJ_28_46	IJ_28_47	IJ_28_48...						
IJ_28_49	IJ_28_50	IJ_28_51	IJ_28_52	IJ_28_53	IJ_28_54	;...		

IJ_29_1	IJ_29_2	IJ_29_3	IJ_29_4	IJ_29_5	IJ_29_6	IJ_29_7	IJ_29_8	IJ_29_9
IJ_29_10	IJ_29_11	IJ_29_12...						
IJ_29_13	IJ_29_14	IJ_29_15	IJ_29_16	IJ_29_17	IJ_29_18	IJ_29_19	IJ_29_20	IJ_29_21
IJ_29_22	IJ_29_23	IJ_29_24...						
IJ_29_25	IJ_29_26	IJ_29_27	IJ_29_28	IJ_29_29	IJ_29_30	IJ_29_31	IJ_29_32	IJ_29_33
IJ_29_34	IJ_29_35	IJ_29_36...						
IJ_29_37	IJ_29_38	IJ_29_39	IJ_29_40	IJ_29_41	IJ_29_42	IJ_29_43	IJ_29_44	IJ_29_45
IJ_29_46	IJ_29_47	IJ_29_48...						
IJ_29_49	IJ_29_50	IJ_29_51	IJ_29_52	IJ_29_53	IJ_29_54	;...		
IJ_30_1	IJ_30_2	IJ_30_3	IJ_30_4	IJ_30_5	IJ_30_6	IJ_30_7	IJ_30_8	IJ_30_9
IJ_30_10	IJ_30_11	IJ_30_12...						
IJ_30_13	IJ_30_14	IJ_30_15	IJ_30_16	IJ_30_17	IJ_30_18	IJ_30_19	IJ_30_20	IJ_30_21
IJ_30_22	IJ_30_23	IJ_30_24...						
IJ_30_25	IJ_30_26	IJ_30_27	IJ_30_28	IJ_30_29	IJ_30_30	IJ_30_31	IJ_30_32	IJ_30_33
IJ_30_34	IJ_30_35	IJ_30_36...						
IJ_30_37	IJ_30_38	IJ_30_39	IJ_30_40	IJ_30_41	IJ_30_42	IJ_30_43	IJ_30_44	IJ_30_45
IJ_30_46	IJ_30_47	IJ_30_48...						
IJ_30_49	IJ_30_50	IJ_30_51	IJ_30_52	IJ_30_53	IJ_30_54	;...		
IJ_31_1	IJ_31_2	IJ_31_3	IJ_31_4	IJ_31_5	IJ_31_6	IJ_31_7	IJ_31_8	IJ_31_9
IJ_31_10	IJ_31_11	IJ_31_12...						
IJ_31_13	IJ_31_14	IJ_31_15	IJ_31_16	IJ_31_17	IJ_31_18	IJ_31_19	IJ_31_20	IJ_31_21
IJ_31_22	IJ_31_23	IJ_31_24...						
IJ_31_25	IJ_31_26	IJ_31_27	IJ_31_28	IJ_31_29	IJ_31_30	IJ_31_31	IJ_31_32	IJ_31_33
IJ_31_34	IJ_31_35	IJ_31_36...						
IJ_31_37	IJ_31_38	IJ_31_39	IJ_31_40	IJ_31_41	IJ_31_42	IJ_31_43	IJ_31_44	IJ_31_45
IJ_31_46	IJ_31_47	IJ_31_48...						
IJ_31_49	IJ_31_50	IJ_31_51	IJ_31_52	IJ_31_53	IJ_31_54	;...		
IJ_32_1	IJ_32_2	IJ_32_3	IJ_32_4	IJ_32_5	IJ_32_6	IJ_32_7	IJ_32_8	IJ_32_9
IJ_32_10	IJ_32_11	IJ_32_12...						
IJ_32_13	IJ_32_14	IJ_32_15	IJ_32_16	IJ_32_17	IJ_32_18	IJ_32_19	IJ_32_20	IJ_32_21
IJ_32_22	IJ_32_23	IJ_32_24...						
IJ_32_25	IJ_32_26	IJ_32_27	IJ_32_28	IJ_32_29	IJ_32_30	IJ_32_31	IJ_32_32	IJ_32_33
IJ_32_34	IJ_32_35	IJ_32_36...						
IJ_32_37	IJ_32_38	IJ_32_39	IJ_32_40	IJ_32_41	IJ_32_42	IJ_32_43	IJ_32_44	IJ_32_45
IJ_32_46	IJ_32_47	IJ_32_48...						
IJ_32_49	IJ_32_50	IJ_32_51	IJ_32_52	IJ_32_53	IJ_32_54	;...		
IJ_33_1	IJ_33_2	IJ_33_3	IJ_33_4	IJ_33_5	IJ_33_6	IJ_33_7	IJ_33_8	IJ_33_9
IJ_33_10	IJ_33_11	IJ_33_12...						
IJ_33_13	IJ_33_14	IJ_33_15	IJ_33_16	IJ_33_17	IJ_33_18	IJ_33_19	IJ_33_20	IJ_33_21
IJ_33_22	IJ_33_23	IJ_33_24...						
IJ_33_25	IJ_33_26	IJ_33_27	IJ_33_28	IJ_33_29	IJ_33_30	IJ_33_31	IJ_33_32	IJ_33_33
IJ_33_34	IJ_33_35	IJ_33_36...						
IJ_33_37	IJ_33_38	IJ_33_39	IJ_33_40	IJ_33_41	IJ_33_42	IJ_33_43	IJ_33_44	IJ_33_45
IJ_33_46	IJ_33_47	IJ_33_48...						
IJ_33_49	IJ_33_50	IJ_33_51	IJ_33_52	IJ_33_53	IJ_33_54	;...		
IJ_34_1	IJ_34_2	IJ_34_3	IJ_34_4	IJ_34_5	IJ_34_6	IJ_34_7	IJ_34_8	IJ_34_9
IJ_34_10	IJ_34_11	IJ_34_12...						
IJ_34_13	IJ_34_14	IJ_34_15	IJ_34_16	IJ_34_17	IJ_34_18	IJ_34_19	IJ_34_20	IJ_34_21
IJ_34_22	IJ_34_23	IJ_34_24...						
IJ_34_25	IJ_34_26	IJ_34_27	IJ_34_28	IJ_34_29	IJ_34_30	IJ_34_31	IJ_34_32	IJ_34_33
IJ_34_34	IJ_34_35	IJ_34_36...						
IJ_34_37	IJ_34_38	IJ_34_39	IJ_34_40	IJ_34_41	IJ_34_42	IJ_34_43	IJ_34_44	IJ_34_45
IJ_34_46	IJ_34_47	IJ_34_48...						
IJ_34_49	IJ_34_50	IJ_34_51	IJ_34_52	IJ_34_53	IJ_34_54	;...		
IJ_35_1	IJ_35_2	IJ_35_3	IJ_35_4	IJ_35_5	IJ_35_6	IJ_35_7	IJ_35_8	IJ_35_9
IJ_35_10	IJ_35_11	IJ_35_12...						
IJ_35_13	IJ_35_14	IJ_35_15	IJ_35_16	IJ_35_17	IJ_35_18	IJ_35_19	IJ_35_20	IJ_35_21
IJ_35_22	IJ_35_23	IJ_35_24...						
IJ_35_25	IJ_35_26	IJ_35_27	IJ_35_28	IJ_35_29	IJ_35_30	IJ_35_31	IJ_35_32	IJ_35_33
IJ_35_34	IJ_35_35	IJ_35_36...						
IJ_35_37	IJ_35_38	IJ_35_39	IJ_35_40	IJ_35_41	IJ_35_42	IJ_35_43	IJ_35_44	IJ_35_45
IJ_35_46	IJ_35_47	IJ_35_48...						
IJ_35_49	IJ_35_50	IJ_35_51	IJ_35_52	IJ_35_53	IJ_35_54	;...		

IJ_36_1	IJ_36_2	IJ_36_3	IJ_36_4	IJ_36_5	IJ_36_6	IJ_36_7	IJ_36_8	IJ_36_9
IJ_36_10	IJ_36_11	IJ_36_12...						
IJ_36_13	IJ_36_14	IJ_36_15	IJ_36_16	IJ_36_17	IJ_36_18	IJ_36_19	IJ_36_20	IJ_36_21
IJ_36_22	IJ_36_23	IJ_36_24...						
IJ_36_25	IJ_36_26	IJ_36_27	IJ_36_28	IJ_36_29	IJ_36_30	IJ_36_31	IJ_36_32	IJ_36_33
IJ_36_34	IJ_36_35	IJ_36_36...						
IJ_36_37	IJ_36_38	IJ_36_39	IJ_36_40	IJ_36_41	IJ_36_42	IJ_36_43	IJ_36_44	IJ_36_45
IJ_36_46	IJ_36_47	IJ_36_48...						
IJ_36_49	IJ_36_50	IJ_36_51	IJ_36_52	IJ_36_53	IJ_36_54	;...		
IJ_37_1	IJ_37_2	IJ_37_3	IJ_37_4	IJ_37_5	IJ_37_6	IJ_37_7	IJ_37_8	IJ_37_9
IJ_37_10	IJ_37_11	IJ_37_12...						
IJ_37_13	IJ_37_14	IJ_37_15	IJ_37_16	IJ_37_17	IJ_37_18	IJ_37_19	IJ_37_20	IJ_37_21
IJ_37_22	IJ_37_23	IJ_37_24...						
IJ_37_25	IJ_37_26	IJ_37_27	IJ_37_28	IJ_37_29	IJ_37_30	IJ_37_31	IJ_37_32	IJ_37_33
IJ_37_34	IJ_37_35	IJ_37_36...						
IJ_37_37	IJ_37_38	IJ_37_39	IJ_37_40	IJ_37_41	IJ_37_42	IJ_37_43	IJ_37_44	IJ_37_45
IJ_37_46	IJ_37_47	IJ_37_48...						
IJ_37_49	IJ_37_50	IJ_37_51	IJ_37_52	IJ_37_53	IJ_37_54	;...		
IJ_38_1	IJ_38_2	IJ_38_3	IJ_38_4	IJ_38_5	IJ_38_6	IJ_38_7	IJ_38_8	IJ_38_9
IJ_38_10	IJ_38_11	IJ_38_12...						
IJ_38_13	IJ_38_14	IJ_38_15	IJ_38_16	IJ_38_17	IJ_38_18	IJ_38_19	IJ_38_20	IJ_38_21
IJ_38_22	IJ_38_23	IJ_38_24...						
IJ_38_25	IJ_38_26	IJ_38_27	IJ_38_28	IJ_38_29	IJ_38_30	IJ_38_31	IJ_38_32	IJ_38_33
IJ_38_34	IJ_38_35	IJ_38_36...						
IJ_38_37	IJ_38_38	IJ_38_39	IJ_38_40	IJ_38_41	IJ_38_42	IJ_38_43	IJ_38_44	IJ_38_45
IJ_38_46	IJ_38_47	IJ_38_48...						
IJ_38_49	IJ_38_50	IJ_38_51	IJ_38_52	IJ_38_53	IJ_38_54	;...		
IJ_39_1	IJ_39_2	IJ_39_3	IJ_39_4	IJ_39_5	IJ_39_6	IJ_39_7	IJ_39_8	IJ_39_9
IJ_39_10	IJ_39_11	IJ_39_12...						
IJ_39_13	IJ_39_14	IJ_39_15	IJ_39_16	IJ_39_17	IJ_39_18	IJ_39_19	IJ_39_20	IJ_39_21
IJ_39_22	IJ_39_23	IJ_39_24...						
IJ_39_25	IJ_39_26	IJ_39_27	IJ_39_28	IJ_39_29	IJ_39_30	IJ_39_31	IJ_39_32	IJ_39_33
IJ_39_34	IJ_39_35	IJ_39_36...						
IJ_39_37	IJ_39_38	IJ_39_39	IJ_39_40	IJ_39_41	IJ_39_42	IJ_39_43	IJ_39_44	IJ_39_45
IJ_39_46	IJ_39_47	IJ_39_48...						
IJ_39_49	IJ_39_50	IJ_39_51	IJ_39_52	IJ_39_53	IJ_39_54	;...		
IJ_40_1	IJ_40_2	IJ_40_3	IJ_40_4	IJ_40_5	IJ_40_6	IJ_40_7	IJ_40_8	IJ_40_9
IJ_40_10	IJ_40_11	IJ_40_12...						
IJ_40_13	IJ_40_14	IJ_40_15	IJ_40_16	IJ_40_17	IJ_40_18	IJ_40_19	IJ_40_20	IJ_40_21
IJ_40_22	IJ_40_23	IJ_40_24...						
IJ_40_25	IJ_40_26	IJ_40_27	IJ_40_28	IJ_40_29	IJ_40_30	IJ_40_31	IJ_40_32	IJ_40_33
IJ_40_34	IJ_40_35	IJ_40_36...						
IJ_40_37	IJ_40_38	IJ_40_39	IJ_40_40	IJ_40_41	IJ_40_42	IJ_40_43	IJ_40_44	IJ_40_45
IJ_40_46	IJ_40_47	IJ_40_48...						
IJ_40_49	IJ_40_50	IJ_40_51	IJ_40_52	IJ_40_53	IJ_40_54	;...		
IJ_41_1	IJ_41_2	IJ_41_3	IJ_41_4	IJ_41_5	IJ_41_6	IJ_41_7	IJ_41_8	IJ_41_9
IJ_41_10	IJ_41_11	IJ_41_12...						
IJ_41_13	IJ_41_14	IJ_41_15	IJ_41_16	IJ_41_17	IJ_41_18	IJ_41_19	IJ_41_20	IJ_41_21
IJ_41_22	IJ_41_23	IJ_41_24...						
IJ_41_25	IJ_41_26	IJ_41_27	IJ_41_28	IJ_41_29	IJ_41_30	IJ_41_31	IJ_41_32	IJ_41_33
IJ_41_34	IJ_41_35	IJ_41_36...						
IJ_41_37	IJ_41_38	IJ_41_39	IJ_41_40	IJ_41_41	IJ_41_42	IJ_41_43	IJ_41_44	IJ_41_45
IJ_41_46	IJ_41_47	IJ_41_48...						
IJ_41_49	IJ_41_50	IJ_41_51	IJ_41_52	IJ_41_53	IJ_41_54	;...		
IJ_42_1	IJ_42_2	IJ_42_3	IJ_42_4	IJ_42_5	IJ_42_6	IJ_42_7	IJ_42_8	IJ_42_9
IJ_42_10	IJ_42_11	IJ_42_12...						
IJ_42_13	IJ_42_14	IJ_42_15	IJ_42_16	IJ_42_17	IJ_42_18	IJ_42_19	IJ_42_20	IJ_42_21
IJ_42_22	IJ_42_23	IJ_42_24...						
IJ_42_25	IJ_42_26	IJ_42_27	IJ_42_28	IJ_42_29	IJ_42_30	IJ_42_31	IJ_42_32	IJ_42_33
IJ_42_34	IJ_42_35	IJ_42_36...						
IJ_42_37	IJ_42_38	IJ_42_39	IJ_42_40	IJ_42_41	IJ_42_42	IJ_42_43	IJ_42_44	IJ_42_45
IJ_42_46	IJ_42_47	IJ_42_48...						
IJ_42_49	IJ_42_50	IJ_42_51	IJ_42_52	IJ_42_53	IJ_42_54	;...		

IJ_43_1	IJ_43_2	IJ_43_3	IJ_43_4	IJ_43_5	IJ_43_6	IJ_43_7	IJ_43_8	IJ_43_9
IJ_43_10	IJ_43_11	IJ_43_12...						
IJ_43_13	IJ_43_14	IJ_43_15	IJ_43_16	IJ_43_17	IJ_43_18	IJ_43_19	IJ_43_20	IJ_43_21
IJ_43_22	IJ_43_23	IJ_43_24...						
IJ_43_25	IJ_43_26	IJ_43_27	IJ_43_28	IJ_43_29	IJ_43_30	IJ_43_31	IJ_43_32	IJ_43_33
IJ_43_34	IJ_43_35	IJ_43_36...						
IJ_43_37	IJ_43_38	IJ_43_39	IJ_43_40	IJ_43_41	IJ_43_42	IJ_43_43	IJ_43_44	IJ_43_45
IJ_43_46	IJ_43_47	IJ_43_48...						
IJ_43_49	IJ_43_50	IJ_43_51	IJ_43_52	IJ_43_53	IJ_43_54	;...		
IJ_44_1	IJ_44_2	IJ_44_3	IJ_44_4	IJ_44_5	IJ_44_6	IJ_44_7	IJ_44_8	IJ_44_9
IJ_44_10	IJ_44_11	IJ_44_12...						
IJ_44_13	IJ_44_14	IJ_44_15	IJ_44_16	IJ_44_17	IJ_44_18	IJ_44_19	IJ_44_20	IJ_44_21
IJ_44_22	IJ_44_23	IJ_44_24...						
IJ_44_25	IJ_44_26	IJ_44_27	IJ_44_28	IJ_44_29	IJ_44_30	IJ_44_31	IJ_44_32	IJ_44_33
IJ_44_34	IJ_44_35	IJ_44_36...						
IJ_44_37	IJ_44_38	IJ_44_39	IJ_44_40	IJ_44_41	IJ_44_42	IJ_44_43	IJ_44_44	IJ_44_45
IJ_44_46	IJ_44_47	IJ_44_48...						
IJ_44_49	IJ_44_50	IJ_44_51	IJ_44_52	IJ_44_53	IJ_44_54	;...		
IJ_45_1	IJ_45_2	IJ_45_3	IJ_45_4	IJ_45_5	IJ_45_6	IJ_45_7	IJ_45_8	IJ_45_9
IJ_45_10	IJ_45_11	IJ_45_12...						
IJ_45_13	IJ_45_14	IJ_45_15	IJ_45_16	IJ_45_17	IJ_45_18	IJ_45_19	IJ_45_20	IJ_45_21
IJ_45_22	IJ_45_23	IJ_45_24...						
IJ_45_25	IJ_45_26	IJ_45_27	IJ_45_28	IJ_45_29	IJ_45_30	IJ_45_31	IJ_45_32	IJ_45_33
IJ_45_34	IJ_45_35	IJ_45_36...						
IJ_45_37	IJ_45_38	IJ_45_39	IJ_45_40	IJ_45_41	IJ_45_42	IJ_45_43	IJ_45_44	IJ_45_45
IJ_45_46	IJ_45_47	IJ_45_48...						
IJ_45_49	IJ_45_50	IJ_45_51	IJ_45_52	IJ_45_53	IJ_45_54	;...		
IJ_46_1	IJ_46_2	IJ_46_3	IJ_46_4	IJ_46_5	IJ_46_6	IJ_46_7	IJ_46_8	IJ_46_9
IJ_46_10	IJ_46_11	IJ_46_12...						
IJ_46_13	IJ_46_14	IJ_46_15	IJ_46_16	IJ_46_17	IJ_46_18	IJ_46_19	IJ_46_20	IJ_46_21
IJ_46_22	IJ_46_23	IJ_46_24...						
IJ_46_25	IJ_46_26	IJ_46_27	IJ_46_28	IJ_46_29	IJ_46_30	IJ_46_31	IJ_46_32	IJ_46_33
IJ_46_34	IJ_46_35	IJ_46_36...						
IJ_46_37	IJ_46_38	IJ_46_39	IJ_46_40	IJ_46_41	IJ_46_42	IJ_46_43	IJ_46_44	IJ_46_45
IJ_46_46	IJ_46_47	IJ_46_48...						
IJ_46_49	IJ_46_50	IJ_46_51	IJ_46_52	IJ_46_53	IJ_46_54	;...		
IJ_47_1	IJ_47_2	IJ_47_3	IJ_47_4	IJ_47_5	IJ_47_6	IJ_47_7	IJ_47_8	IJ_47_9
IJ_47_10	IJ_47_11	IJ_47_12...						
IJ_47_13	IJ_47_14	IJ_47_15	IJ_47_16	IJ_47_17	IJ_47_18	IJ_47_19	IJ_47_20	IJ_47_21
IJ_47_22	IJ_47_23	IJ_47_24...						
IJ_47_25	IJ_47_26	IJ_47_27	IJ_47_28	IJ_47_29	IJ_47_30	IJ_47_31	IJ_47_32	IJ_47_33
IJ_47_34	IJ_47_35	IJ_47_36...						
IJ_47_37	IJ_47_38	IJ_47_39	IJ_47_40	IJ_47_41	IJ_47_42	IJ_47_43	IJ_47_44	IJ_47_45
IJ_47_46	IJ_47_47	IJ_47_48...						
IJ_47_49	IJ_47_50	IJ_47_51	IJ_47_52	IJ_47_53	IJ_47_54	;...		
IJ_48_1	IJ_48_2	IJ_48_3	IJ_48_4	IJ_48_5	IJ_48_6	IJ_48_7	IJ_48_8	IJ_48_9
IJ_48_10	IJ_48_11	IJ_48_12...						
IJ_48_13	IJ_48_14	IJ_48_15	IJ_48_16	IJ_48_17	IJ_48_18	IJ_48_19	IJ_48_20	IJ_48_21
IJ_48_22	IJ_48_23	IJ_48_24...						
IJ_48_25	IJ_48_26	IJ_48_27	IJ_48_28	IJ_48_29	IJ_48_30	IJ_48_31	IJ_48_32	IJ_48_33
IJ_48_34	IJ_48_35	IJ_48_36...						
IJ_48_37	IJ_48_38	IJ_48_39	IJ_48_40	IJ_48_41	IJ_48_42	IJ_48_43	IJ_48_44	IJ_48_45
IJ_48_46	IJ_48_47	IJ_48_48...						
IJ_48_49	IJ_48_50	IJ_48_51	IJ_48_52	IJ_48_53	IJ_48_54	;...		
IJ_49_1	IJ_49_2	IJ_49_3	IJ_49_4	IJ_49_5	IJ_49_6	IJ_49_7	IJ_49_8	IJ_49_9
IJ_49_10	IJ_49_11	IJ_49_12...						
IJ_49_13	IJ_49_14	IJ_49_15	IJ_49_16	IJ_49_17	IJ_49_18	IJ_49_19	IJ_49_20	IJ_49_21
IJ_49_22	IJ_49_23	IJ_49_24...						
IJ_49_25	IJ_49_26	IJ_49_27	IJ_49_28	IJ_49_29	IJ_49_30	IJ_49_31	IJ_49_32	IJ_49_33
IJ_49_34	IJ_49_35	IJ_49_36...						
IJ_49_37	IJ_49_38	IJ_49_39	IJ_49_40	IJ_49_41	IJ_49_42	IJ_49_43	IJ_49_44	IJ_49_45
IJ_49_46	IJ_49_47	IJ_49_48...						
IJ_49_49	IJ_49_50	IJ_49_51	IJ_49_52	IJ_49_53	IJ_49_54	;...		

IJ_50_1	IJ_50_2	IJ_50_3	IJ_50_4	IJ_50_5	IJ_50_6	IJ_50_7	IJ_50_8	IJ_50_9
IJ_50_10	IJ_50_11	IJ_50_12...						
IJ_50_13	IJ_50_14	IJ_50_15	IJ_50_16	IJ_50_17	IJ_50_18	IJ_50_19	IJ_50_20	IJ_50_21
IJ_50_22	IJ_50_23	IJ_50_24...						
IJ_50_25	IJ_50_26	IJ_50_27	IJ_50_28	IJ_50_29	IJ_50_30	IJ_50_31	IJ_50_32	IJ_50_33
IJ_50_34	IJ_50_35	IJ_50_36...						
IJ_50_37	IJ_50_38	IJ_50_39	IJ_50_40	IJ_50_41	IJ_50_42	IJ_50_43	IJ_50_44	IJ_50_45
IJ_50_46	IJ_50_47	IJ_50_48...						
IJ_50_49	IJ_50_50	IJ_50_51	IJ_50_52	IJ_50_53	IJ_50_54	;...		
IJ_51_1	IJ_51_2	IJ_51_3	IJ_51_4	IJ_51_5	IJ_51_6	IJ_51_7	IJ_51_8	IJ_51_9
IJ_51_10	IJ_51_11	IJ_51_12...						
IJ_51_13	IJ_51_14	IJ_51_15	IJ_51_16	IJ_51_17	IJ_51_18	IJ_51_19	IJ_51_20	IJ_51_21
IJ_51_22	IJ_51_23	IJ_51_24...						
IJ_51_25	IJ_51_26	IJ_51_27	IJ_51_28	IJ_51_29	IJ_51_30	IJ_51_31	IJ_51_32	IJ_51_33
IJ_51_34	IJ_51_35	IJ_51_36...						
IJ_51_37	IJ_51_38	IJ_51_39	IJ_51_40	IJ_51_41	IJ_51_42	IJ_51_43	IJ_51_44	IJ_51_45
IJ_51_46	IJ_51_47	IJ_51_48...						
IJ_51_49	IJ_51_50	IJ_51_51	IJ_51_52	IJ_51_53	IJ_51_54	;...		
IJ_52_1	IJ_52_2	IJ_52_3	IJ_52_4	IJ_52_5	IJ_52_6	IJ_52_7	IJ_52_8	IJ_52_9
IJ_52_10	IJ_52_11	IJ_52_12...						
IJ_52_13	IJ_52_14	IJ_52_15	IJ_52_16	IJ_52_17	IJ_52_18	IJ_52_19	IJ_52_20	IJ_52_21
IJ_52_22	IJ_52_23	IJ_52_24...						
IJ_52_25	IJ_52_26	IJ_52_27	IJ_52_28	IJ_52_29	IJ_52_30	IJ_52_31	IJ_52_32	IJ_52_33
IJ_52_34	IJ_52_35	IJ_52_36...						
IJ_52_37	IJ_52_38	IJ_52_39	IJ_52_40	IJ_52_41	IJ_52_42	IJ_52_43	IJ_52_44	IJ_52_45
IJ_52_46	IJ_52_47	IJ_52_48...						
IJ_52_49	IJ_52_50	IJ_52_51	IJ_52_52	IJ_52_53	IJ_52_54	;...		
IJ_53_1	IJ_53_2	IJ_53_3	IJ_53_4	IJ_53_5	IJ_53_6	IJ_53_7	IJ_53_8	IJ_53_9
IJ_53_10	IJ_53_11	IJ_53_12...						
IJ_53_13	IJ_53_14	IJ_53_15	IJ_53_16	IJ_53_17	IJ_53_18	IJ_53_19	IJ_53_20	IJ_53_21
IJ_53_22	IJ_53_23	IJ_53_24...						
IJ_53_25	IJ_53_26	IJ_53_27	IJ_53_28	IJ_53_29	IJ_53_30	IJ_53_31	IJ_53_32	IJ_53_33
IJ_53_34	IJ_53_35	IJ_53_36...						
IJ_53_37	IJ_53_38	IJ_53_39	IJ_53_40	IJ_53_41	IJ_53_42	IJ_53_43	IJ_53_44	IJ_53_45
IJ_53_46	IJ_53_47	IJ_53_48...						
IJ_53_49	IJ_53_50	IJ_53_51	IJ_53_52	IJ_53_53	IJ_53_54	;...		
IJ_54_1	IJ_54_2	IJ_54_3	IJ_54_4	IJ_54_5	IJ_54_6	IJ_54_7	IJ_54_8	IJ_54_9
IJ_54_10	IJ_54_11	IJ_54_12...						
IJ_54_13	IJ_54_14	IJ_54_15	IJ_54_16	IJ_54_17	IJ_54_18	IJ_54_19	IJ_54_20	IJ_54_21
IJ_54_22	IJ_54_23	IJ_54_24...						
IJ_54_25	IJ_54_26	IJ_54_27	IJ_54_28	IJ_54_29	IJ_54_30	IJ_54_31	IJ_54_32	IJ_54_33
IJ_54_34	IJ_54_35	IJ_54_36...						
IJ_54_37	IJ_54_38	IJ_54_39	IJ_54_40	IJ_54_41	IJ_54_42	IJ_54_43	IJ_54_44	IJ_54_45
IJ_54_46	IJ_54_47	IJ_54_48...						
IJ_54_49	IJ_54_50	IJ_54_51	IJ_54_52	IJ_54_53	IJ_54_54];		

```
% Inverse Jacobian Matrix Elements
```

```
IJ_1_1=1;  
IJ_1_2=0;  
IJ_1_3=0;  
IJ_1_4=0;  
IJ_1_5=0;  
IJ_1_6=0;  
IJ_1_7=0;  
IJ_1_8=0;  
IJ_1_9=0;  
IJ_1_10=0;  
IJ_1_11=0;  
IJ_1_12=0;  
IJ_1_13=0;  
IJ_1_14=0;  
IJ_1_15=0;  
IJ_1_16=0;  
IJ_1_17=0;  
IJ_1_18=0;  
IJ_1_19=0;  
IJ_1_20=0;  
IJ_1_21=0;  
IJ_1_22=0;  
IJ_1_23=0;  
IJ_1_24=0;  
IJ_1_25=0;  
IJ_1_26=0;  
IJ_1_27=0;  
IJ_1_28=0;  
IJ_1_29=0;  
IJ_1_30=0;  
IJ_1_31=0;  
IJ_1_32=0;  
IJ_1_33=0;  
IJ_1_34=0;  
IJ_1_35=0;  
IJ_1_36=0;  
IJ_1_37=0;  
IJ_1_38=0;  
IJ_1_39=0;  
IJ_1_40=0;  
IJ_1_41=0;  
IJ_1_42=0;  
IJ_1_43=0;  
IJ_1_44=0;  
IJ_1_45=0;  
IJ_1_46=0;  
IJ_1_47=0;  
IJ_1_48=0;  
IJ_1_49=0;  
IJ_1_50=0;  
IJ_1_51=0;  
IJ_1_52=0;  
IJ_1_53=0;  
IJ_1_54=0;
```

```
IJ_2_1=0;  
IJ_2_2=1;  
IJ_2_3=0;  
IJ_2_4=0;  
IJ_2_5=0;  
IJ_2_6=0;  
IJ_2_7=0;  
IJ_2_8=0;  
IJ_2_9=0;  
IJ_2_10=0;  
IJ_2_11=0;  
IJ_2_12=0;  
IJ_2_13=0;  
IJ_2_14=0;  
IJ_2_15=0;
```

IJ_2_16=0;
IJ_2_17=0;
IJ_2_18=0;
IJ_2_19=0;
IJ_2_20=0;
IJ_2_21=0;
IJ_2_22=0;
IJ_2_23=0;
IJ_2_24=0;
IJ_2_25=0;
IJ_2_26=0;
IJ_2_27=0;
IJ_2_28=0;
IJ_2_29=0;
IJ_2_30=0;
IJ_2_31=0;
IJ_2_32=0;
IJ_2_33=0;
IJ_2_34=0;
IJ_2_35=0;
IJ_2_36=0;
IJ_2_37=0;
IJ_2_38=0;
IJ_2_39=0;
IJ_2_40=0;
IJ_2_41=0;
IJ_2_42=0;
IJ_2_43=0;
IJ_2_44=0;
IJ_2_45=0;
IJ_2_46=0;
IJ_2_47=0;
IJ_2_48=0;
IJ_2_49=0;
IJ_2_50=0;
IJ_2_51=0;
IJ_2_52=0;
IJ_2_53=0;
IJ_2_54=0;

IJ_3_1=0;
IJ_3_2=0;
IJ_3_3=1;
IJ_3_4=0;
IJ_3_5=0;
IJ_3_6=0;
IJ_3_7=0;
IJ_3_8=0;
IJ_3_9=0;
IJ_3_10=0;
IJ_3_11=0;
IJ_3_12=0;
IJ_3_13=0;
IJ_3_14=0;
IJ_3_15=0;
IJ_3_16=0;
IJ_3_17=0;
IJ_3_18=0;
IJ_3_19=0;
IJ_3_20=0;
IJ_3_21=0;
IJ_3_22=0;
IJ_3_23=0;
IJ_3_24=0;
IJ_3_25=0;
IJ_3_26=0;
IJ_3_27=0;
IJ_3_28=0;
IJ_3_29=0;
IJ_3_30=0;
IJ_3_31=0;

```

IJ_3_32=0;
IJ_3_33=0;
IJ_3_34=0;
IJ_3_35=0;
IJ_3_36=0;
IJ_3_37=0;
IJ_3_38=0;
IJ_3_39=0;
IJ_3_40=0;
IJ_3_41=0;
IJ_3_42=0;
IJ_3_43=0;
IJ_3_44=0;
IJ_3_45=0;
IJ_3_46=0;
IJ_3_47=0;
IJ_3_48=0;
IJ_3_49=0;
IJ_3_50=0;
IJ_3_51=0;
IJ_3_52=0;
IJ_3_53=0;
IJ_3_54=0;

IJ_4_1=1;
IJ_4_2=0;
IJ_4_3=0;
IJ_4_4=(-1).*LC2.*(cos(PHI_C).*cos(PSI_C)+sin(PHI_C).*sin(PSI_C).*sin(THETA_C)) ...
;
IJ_4_5=LC2.*cos(PSI_C).*cos(THETA_C).*sin(PHI_C);
IJ_4_6=LC2.*(sin(PHI_C).*sin(PSI_C)+cos(PHI_C).*cos(PSI_C).*sin(THETA_C));
IJ_4_7=0;
IJ_4_8=0;
IJ_4_9=0;
IJ_4_10=0;
IJ_4_11=0;
IJ_4_12=0;
IJ_4_13=0;
IJ_4_14=0;
IJ_4_15=0;
IJ_4_16=0;
IJ_4_17=0;
IJ_4_18=0;
IJ_4_19=0;
IJ_4_20=0;
IJ_4_21=0;
IJ_4_22=0;
IJ_4_23=0;
IJ_4_24=0;
IJ_4_25=0;
IJ_4_26=0;
IJ_4_27=0;
IJ_4_28=0;
IJ_4_29=0;
IJ_4_30=0;
IJ_4_31=0;
IJ_4_32=0;
IJ_4_33=0;
IJ_4_34=(-1).*cos(PHI_C).*sin(PSI_C)+cos(PSI_C).*sin(PHI_C).*sin(THETA_C);
IJ_4_35=0;
IJ_4_36=0;
IJ_4_37=0;
IJ_4_38=0;
IJ_4_39=0;
IJ_4_40=0;
IJ_4_41=0;
IJ_4_42=0;
IJ_4_43=0;
IJ_4_44=0;
IJ_4_45=0;
IJ_4_46=0;

```

```

IJ_4_47=0;
IJ_4_48=0;
IJ_4_49=0;
IJ_4_50=0;
IJ_4_51=0;
IJ_4_52=0;
IJ_4_53=0;
IJ_4_54=0;

IJ_5_1=0;
IJ_5_2=1;
IJ_5_3=0;
IJ_5_4=LC2.*((-1).*cos(PHI_C).*sin(PSI_C)+cos(PSI_C).*sin(PHI_C).*sin(THETA_C)) ...
;
IJ_5_5=LC2.*cos(THETA_C).*sin(PHI_C).*sin(PSI_C);
IJ_5_6=LC2.*((-1).*cos(PSI_C).*sin(PHI_C)+cos(PHI_C).*sin(PSI_C).*sin(THETA_C)) ...
;
IJ_5_7=0;
IJ_5_8=0;
IJ_5_9=0;
IJ_5_10=0;
IJ_5_11=0;
IJ_5_12=0;
IJ_5_13=0;
IJ_5_14=0;
IJ_5_15=0;
IJ_5_16=0;
IJ_5_17=0;
IJ_5_18=0;
IJ_5_19=0;
IJ_5_20=0;
IJ_5_21=0;
IJ_5_22=0;
IJ_5_23=0;
IJ_5_24=0;
IJ_5_25=0;
IJ_5_26=0;
IJ_5_27=0;
IJ_5_28=0;
IJ_5_29=0;
IJ_5_30=0;
IJ_5_31=0;
IJ_5_32=0;
IJ_5_33=0;
IJ_5_34=cos(PHI_C).*cos(PSI_C)+sin(PHI_C).*sin(PSI_C).*sin(THETA_C);
IJ_5_35=0;
IJ_5_36=0;
IJ_5_37=0;
IJ_5_38=0;
IJ_5_39=0;
IJ_5_40=0;
IJ_5_41=0;
IJ_5_42=0;
IJ_5_43=0;
IJ_5_44=0;
IJ_5_45=0;
IJ_5_46=0;
IJ_5_47=0;
IJ_5_48=0;
IJ_5_49=0;
IJ_5_50=0;
IJ_5_51=0;
IJ_5_52=0;
IJ_5_53=0;
IJ_5_54=0;

IJ_6_1=0;
IJ_6_2=0;
IJ_6_3=1;
IJ_6_4=0;
IJ_6_5=(-1).*LC2.*sin(PHI_C).*sin(THETA_C);

```

```

IJ_6_6=LC2.*cos(PHI_C).*cos(THETA_C);
IJ_6_7=0;
IJ_6_8=0;
IJ_6_9=0;
IJ_6_10=0;
IJ_6_11=0;
IJ_6_12=0;
IJ_6_13=0;
IJ_6_14=0;
IJ_6_15=0;
IJ_6_16=0;
IJ_6_17=0;
IJ_6_18=0;
IJ_6_19=0;
IJ_6_20=0;
IJ_6_21=0;
IJ_6_22=0;
IJ_6_23=0;
IJ_6_24=0;
IJ_6_25=0;
IJ_6_26=0;
IJ_6_27=0;
IJ_6_28=0;
IJ_6_29=0;
IJ_6_30=0;
IJ_6_31=0;
IJ_6_32=0;
IJ_6_33=0;
IJ_6_34=cos(THETA_C).*sin(PHI_C);
IJ_6_35=0;
IJ_6_36=0;
IJ_6_37=0;
IJ_6_38=0;
IJ_6_39=0;
IJ_6_40=0;
IJ_6_41=0;
IJ_6_42=0;
IJ_6_43=0;
IJ_6_44=0;
IJ_6_45=0;
IJ_6_46=0;
IJ_6_47=0;
IJ_6_48=0;
IJ_6_49=0;
IJ_6_50=0;
IJ_6_51=0;
IJ_6_52=0;
IJ_6_53=0;
IJ_6_54=0;

IJ_7_1=1;
IJ_7_2=0;
IJ_7_3=0;
IJ_7_4=(-1).*L23.*cos(ALPHA3).*cos(BETA3).*cos(THETA_C).*sin(PSI_C)+L23.*sin( ...
    ALPHA3).*((-1).*cos(PSI_C).*sin(PHI_C)+cos(PHI_C).*sin(PSI_C).*sin( ...
    THETA_C))+(-1).*(LC2+L23.*cos(ALPHA3).*sin(BETA3)).*(cos(PHI_C).*cos( ...
    PSI_C)+sin(PHI_C).*sin(PSI_C).*sin(THETA_C));
IJ_7_5=cos(PSI_C).*((-1).*L23.*cos(PHI_C).*cos(THETA_C).*sin(ALPHA3)+cos( ...
    THETA_C).*(LC2+L23.*cos(ALPHA3).*sin(BETA3)).*sin(PHI_C))+(-1).*L23.*cos( ...
    ALPHA3).*cos(BETA3).*sin(THETA_C));
IJ_7_6=(LC2+L23.*cos(ALPHA3).*sin(BETA3)).*(sin(PHI_C).*sin(PSI_C)+cos(PHI_C).* ...
    cos(PSI_C).*sin(THETA_C))+L23.*sin(ALPHA3).*((-1).*cos(PHI_C).*sin( ...
    PSI_C)+cos(PSI_C).*sin(PHI_C).*sin(THETA_C));
IJ_7_7=0;
IJ_7_8=0;
IJ_7_9=0;
IJ_7_10=0;
IJ_7_11=0;
IJ_7_12=0;
IJ_7_13=0;
IJ_7_14=0;

```



```

IJ_7_15=0;
IJ_7_16=0;
IJ_7_17=0;
IJ_7_18=0;
IJ_7_19=0;
IJ_7_20=0;
IJ_7_21=0;
IJ_7_22=0;
IJ_7_23=0;
IJ_7_24=0;
IJ_7_25=0;
IJ_7_26=0;
IJ_7_27=0;
IJ_7_28=0;
IJ_7_29=0;
IJ_7_30=0;
IJ_7_31=0;
IJ_7_32=0;
IJ_7_33=0;
IJ_7_34=(-1).*cos(PHI_C).*sin(PSI_C)+cos(PSI_C).*sin(PHI_C).*sin(THETA_C);
IJ_7_35=(-1).*sin(ALPHA3).*sin(PHI_C).*sin(PSI_C)+cos(PHI_C).*cos(PSI_C).*sin( ...
    THETA_C))+cos(ALPHA3).*cos(BETA3).*cos(PSI_C).*cos(THETA_C)+sin(BETA3) ...
    .*((-1).*cos(PHI_C).*sin(PSI_C)+cos(PSI_C).*sin(PHI_C).*sin(THETA_C))); ...
IJ_7_36=0;
IJ_7_37=0;
IJ_7_38=0;
IJ_7_39=0;
IJ_7_40=0;
IJ_7_41=0;
IJ_7_42=0;
IJ_7_43=(-1).*L23.*(cos(BETA3).*cos(PSI_C).*cos(THETA_C).*sin(ALPHA3)+(-1).*cos( ...
    PHI_C).*sin(ALPHA3).*sin(BETA3).*sin(PSI_C)+cos(ALPHA3).*sin(PHI_C).* ...
    sin(PSI_C)+cos(PSI_C).*cos(ALPHA3).*cos(PHI_C)+sin(ALPHA3).*sin(BETA3) ...
    .*sin(PHI_C)).*sin(THETA_C));
IJ_7_44=0;
IJ_7_45=0;
IJ_7_46=0;
IJ_7_47=0;
IJ_7_48=0;
IJ_7_49=(-1).*L23.*cos(ALPHA3).*cos(BETA3).*cos(PHI_C).*sin(PSI_C)+cos(PSI_C).* ...
    (cos(THETA_C).*sin(BETA3)+(-1).*cos(BETA3).*sin(PHI_C).*sin(THETA_C))); ...
IJ_7_50=0;
IJ_7_51=0;
IJ_7_52=0;
IJ_7_53=0;
IJ_7_54=0;

IJ_8_1=0;
IJ_8_2=1;
IJ_8_3=0;
IJ_8_4=L23.*cos(ALPHA3).*cos(BETA3).*cos(PSI_C).*cos(THETA_C)+(-1).*L23.*sin( ...
    ALPHA3).*sin(PHI_C).*sin(PSI_C)+cos(PHI_C).*cos(PSI_C).*sin(THETA_C))+ ...
    LC2+L23.*cos(ALPHA3).*sin(BETA3)).*((-1).*cos(PHI_C).*sin(PSI_C)+cos( ...
    PSI_C).*sin(PHI_C).*sin(THETA_C));
IJ_8_5=sin(PSI_C).*((-1).*L23.*cos(PHI_C).*cos(THETA_C).*sin(ALPHA3)+cos( ...
    THETA_C)).*(LC2+L23.*cos(ALPHA3).*sin(BETA3)).*sin(PHI_C)+(-1).*L23.*cos( ...
    ALPHA3).*cos(BETA3).*sin(THETA_C));
IJ_8_6=(LC2+L23.*cos(ALPHA3).*sin(BETA3)).*((-1).*cos(PSI_C).*sin(PHI_C)+cos( ...
    PHI_C).*sin(PSI_C).*sin(THETA_C))+L23.*sin(ALPHA3).*cos(PHI_C).*cos( ...
    PSI_C)+sin(PHI_C).*sin(PSI_C).*sin(THETA_C));
IJ_8_7=0;
IJ_8_8=0;
IJ_8_9=0;
IJ_8_10=0;
IJ_8_11=0;
IJ_8_12=0;
IJ_8_13=0;
IJ_8_14=0;
IJ_8_15=0;
IJ_8_16=0;
IJ_8_17=0;

```

```

IJ_8_18=0;
IJ_8_19=0;
IJ_8_20=0;
IJ_8_21=0;
IJ_8_22=0;
IJ_8_23=0;
IJ_8_24=0;
IJ_8_25=0;
IJ_8_26=0;
IJ_8_27=0;
IJ_8_28=0;
IJ_8_29=0;
IJ_8_30=0;
IJ_8_31=0;
IJ_8_32=0;
IJ_8_33=0;
IJ_8_34=cos(PHI_C).*cos(PSI_C)+sin(PHI_C).*sin(PSI_C).*sin(THETA_C);
IJ_8_35=sin(ALPHA3).*(cos(PSI_C).*sin(PHI_C)+(-1).*cos(PHI_C).*sin(PSI_C).*sin( ...
    THETA_C))+cos(ALPHA3).*(cos(PHI_C).*cos(PSI_C).*sin(BETA3)+sin(PSI_C).*( ...
    cos(BETA3)).*cos(THETA_C)+sin(BETA3).*sin(PHI_C).*sin(THETA_C));
IJ_8_36=0;
IJ_8_37=0;
IJ_8_38=0;
IJ_8_39=0;
IJ_8_40=0;
IJ_8_41=0;
IJ_8_42=0;
IJ_8_43=(-1).*L23.*((-1).*cos(ALPHA3).*cos(PSI_C).*sin(PHI_C)+sin(ALPHA3).*sin( ...
    PSI_C).*(cos(BETA3).*cos(THETA_C)+sin(BETA3).*sin(PHI_C).*sin(THETA_C))+ ...
    cos(PHI_C).*(cos(PSI_C).*sin(ALPHA3).*sin(BETA3)+cos(ALPHA3).*sin(PSI_C) ...
    .*sin(THETA_C)));
IJ_8_44=0;
IJ_8_45=0;
IJ_8_46=0;
IJ_8_47=0;
IJ_8_48=0;
IJ_8_49=L23.*cos(ALPHA3).*((-1).*cos(THETA_C).*sin(BETA3).*sin(PSI_C)+cos(BETA3) ...
    *(cos(PHI_C).*cos(PSI_C)+sin(PHI_C).*sin(PSI_C).*sin(THETA_C)));
IJ_8_50=0;
IJ_8_51=0;
IJ_8_52=0;
IJ_8_53=0;
IJ_8_54=0;

IJ_9_1=0;
IJ_9_2=0;
IJ_9_3=1;
IJ_9_4=0;
IJ_9_5=(-1).*L23.*cos(ALPHA3).*cos(BETA3).*cos(THETA_C)+L23.*cos(PHI_C).*sin( ...
    ALPHA3).*sin(THETA_C)+(-1).*(LC2+L23.*cos(ALPHA3).*sin(BETA3)).*sin( ...
    PHI_C).*sin(THETA_C);
IJ_9_6=cos(THETA_C).*(cos(PHI_C).*(LC2+L23.*cos(ALPHA3).*sin(BETA3))+L23.*sin( ...
    ALPHA3).*sin(PHI_C));
IJ_9_7=0;
IJ_9_8=0;
IJ_9_9=0;
IJ_9_10=0;
IJ_9_11=0;
IJ_9_12=0;
IJ_9_13=0;
IJ_9_14=0;
IJ_9_15=0;
IJ_9_16=0;
IJ_9_17=0;
IJ_9_18=0;
IJ_9_19=0;
IJ_9_20=0;
IJ_9_21=0;
IJ_9_22=0;
IJ_9_23=0;
IJ_9_24=0;

```

```

IJ_9_25=0;
IJ_9_26=0;
IJ_9_27=0;
IJ_9_28=0;
IJ_9_29=0;
IJ_9_30=0;
IJ_9_31=0;
IJ_9_32=0;
IJ_9_33=0;
IJ_9_34=cos(THETA_C).*sin(PHI_C);
IJ_9_35=(-1).*cos(PHI_C).*cos(THETA_C).*sin(ALPHA3)+cos(ALPHA3).*(cos(THETA_C).* ...
    sin(BETA3).*sin(PHI_C)+(-1).*cos(BETA3).*sin(THETA_C));
IJ_9_36=0;
IJ_9_37=0;
IJ_9_38=0;
IJ_9_39=0;
IJ_9_40=0;
IJ_9_41=0;
IJ_9_42=0;
IJ_9_43=(-1).*L23.*cos(THETA_C).*(cos(ALPHA3).*cos(PHI_C)+sin(ALPHA3).*sin( ...
    BETA3).*sin(PHI_C))+L23.*cos(BETA3).*sin(ALPHA3).*sin(THETA_C);
IJ_9_44=0;
IJ_9_45=0;
IJ_9_46=0;
IJ_9_47=0;
IJ_9_48=0;
IJ_9_49=L23.*cos(ALPHA3).*(cos(BETA3).*cos(THETA_C).*sin(PHI_C)+sin(BETA3).*sin( ...
    THETA_C));
IJ_9_50=0;
IJ_9_51=0;
IJ_9_52=0;
IJ_9_53=0;
IJ_9_54=0;

IJ_10_1=1;
IJ_10_2=0;
IJ_10_3=0;
IJ_10_4=LC4.*(sin(ALPHA4).*((-1).*cos(PHI_C).*sin(PHI_C)+cos(PHI_C).*sin(PHI_C) ...
    .*sin(THETA_C))+cos(ALPHA4).*((-1).*cos(THETA_C).*sin(BETA4).*sin(PHI_C) ...
    +cos(BETA4).*(cos(PHI_C).*cos(PHI_C)+sin(PHI_C).*sin(PHI_C).*sin( ...
    THETA_C))));
IJ_10_5=(-1).*LC4.*cos(PHI_C).*(cos(PHI_C).*cos(THETA_C).*sin(ALPHA4)+cos( ...
    ALPHA4).*(cos(BETA4).*cos(THETA_C).*sin(PHI_C)+sin(BETA4).*sin(THETA_C)) ...
    );
IJ_10_6=(-1).*LC4.*((cos(PHI_C).*sin(ALPHA4)+cos(ALPHA4).*cos(BETA4).*sin(PHI_C) ...
    ).*sin(PHI_C)+cos(PHI_C).*(cos(ALPHA4).*cos(BETA4).*cos(PHI_C)+(-1).* ...
    sin(ALPHA4).*sin(PHI_C)).*sin(THETA_C));
IJ_10_7=0;
IJ_10_8=0;
IJ_10_9=0;
IJ_10_10=0;
IJ_10_11=0;
IJ_10_12=0;
IJ_10_13=0;
IJ_10_14=0;
IJ_10_15=0;
IJ_10_16=0;
IJ_10_17=0;
IJ_10_18=0;
IJ_10_19=0;
IJ_10_20=0;
IJ_10_21=0;
IJ_10_22=0;
IJ_10_23=0;
IJ_10_24=0;
IJ_10_25=0;
IJ_10_26=0;
IJ_10_27=0;
IJ_10_28=0;
IJ_10_29=0;
IJ_10_30=0;

```

```

IJ_10_31=0;
IJ_10_32=0;
IJ_10_33=0;
IJ_10_34=0;
IJ_10_35=0;
IJ_10_36=(-1).*sin(ALPHA4).*sin(PHI_C).*sin(PSI_C)+cos(PHI_C).*cos(PSI_C).*sin( ...
    THETA_C))+cos(ALPHA4).*cos(BETA4).*cos(PHI_C).*sin(PSI_C)+cos(PSI_C).*cos( ...
    cos(THETA_C).*sin(BETA4)+(-1).*cos(BETA4).*sin(PHI_C).*sin(THETA_C));
IJ_10_37=0;
IJ_10_38=0;
IJ_10_39=0;
IJ_10_40=0;
IJ_10_41=0;
IJ_10_42=0;
IJ_10_43=0;
IJ_10_44=(-1).*LC4.*((cos(BETA4).*cos(PHI_C).*sin(ALPHA4)+cos(ALPHA4).*sin(PHI_C) ...
    ).*sin(PSI_C)+cos(PSI_C).*cos(THETA_C).*sin(ALPHA4).*sin(BETA4)+cos( ...
    ALPHA4).*cos(PHI_C)+(-1).*cos(BETA4).*sin(ALPHA4).*sin(PHI_C)).*sin( ...
    THETA_C));
IJ_10_45=0;
IJ_10_46=0;
IJ_10_47=0;
IJ_10_48=0;
IJ_10_49=0;
IJ_10_50=LC4.*cos(ALPHA4).*cos(BETA4).*cos(PSI_C).*cos(THETA_C)+sin(BETA4).*(( ...
    -1).*cos(PHI_C).*sin(PSI_C)+cos(PSI_C).*sin(PHI_C).*sin(THETA_C));
IJ_10_51=0;
IJ_10_52=0;
IJ_10_53=0;
IJ_10_54=0;

IJ_11_1=0;
IJ_11_2=1;
IJ_11_3=0;
IJ_11_4=LC4.*((-1).*sin(ALPHA4).*sin(PHI_C).*sin(PSI_C)+cos(PHI_C).*cos(PSI_C) ...
    .*sin(THETA_C))+cos(ALPHA4).*cos(BETA4).*cos(PHI_C).*sin(PSI_C)+cos( ...
    PSI_C).*cos(THETA_C).*sin(BETA4)+(-1).*cos(BETA4).*sin(PHI_C).*sin( ...
    THETA_C));
IJ_11_5=(-1).*LC4.*sin(PSI_C).*cos(PHI_C).*cos(THETA_C).*sin(ALPHA4)+cos( ...
    ALPHA4).*cos(BETA4).*cos(THETA_C).*sin(PHI_C)+sin(BETA4).*sin(THETA_C) ...
    );
IJ_11_6=LC4.*(cos(PHI_C).*cos(PSI_C).*sin(ALPHA4)+(-1).*cos(ALPHA4).*cos(BETA4) ...
    .*sin(PSI_C).*sin(THETA_C))+sin(PHI_C).*cos(ALPHA4).*cos(BETA4).*cos( ...
    PSI_C)+sin(ALPHA4).*sin(PSI_C).*sin(THETA_C));
IJ_11_7=0;
IJ_11_8=0;
IJ_11_9=0;
IJ_11_10=0;
IJ_11_11=0;
IJ_11_12=0;
IJ_11_13=0;
IJ_11_14=0;
IJ_11_15=0;
IJ_11_16=0;
IJ_11_17=0;
IJ_11_18=0;
IJ_11_19=0;
IJ_11_20=0;
IJ_11_21=0;
IJ_11_22=0;
IJ_11_23=0;
IJ_11_24=0;
IJ_11_25=0;
IJ_11_26=0;
IJ_11_27=0;
IJ_11_28=0;
IJ_11_29=0;
IJ_11_30=0;
IJ_11_31=0;
IJ_11_32=0;
IJ_11_33=0;

```

```

IJ_11_34=0;
IJ_11_35=0;
IJ_11_36=sin(ALPHA4) .* (cos(PHI_C) .* sin(PHI_C) + (-1) .* cos(PHI_C) .* sin(PHI_C) .* sin( ...
    THETA_C)) + cos(ALPHA4) .* (cos(THETA_C) .* sin(BETA4) .* sin(PHI_C) + (-1) .* cos( ...
    BETA4) .* (cos(PHI_C) .* cos(PHI_C) + sin(PHI_C) .* sin(PHI_C) .* sin(THETA_C))); ...
IJ_11_37=0;
IJ_11_38=0;
IJ_11_39=0;
IJ_11_40=0;
IJ_11_41=0;
IJ_11_42=0;
IJ_11_43=0;
IJ_11_44=LC4 .* ((-1) .* cos(THETA_C) .* sin(ALPHA4) .* sin(BETA4) .* sin(PHI_C) + cos( ...
    ALPHA4) .* (cos(PHI_C) .* sin(PHI_C) + (-1) .* cos(PHI_C) .* sin(PHI_C) .* sin( ...
    THETA_C)) + cos(BETA4) .* sin(ALPHA4) .* (cos(PHI_C) .* cos(PHI_C) + sin(PHI_C) .* ...
    sin(PHI_C) .* sin(THETA_C)));
IJ_11_45=0;
IJ_11_46=0;
IJ_11_47=0;
IJ_11_48=0;
IJ_11_49=0;
IJ_11_50=LC4 .* cos(ALPHA4) .* (cos(PHI_C) .* cos(PHI_C) .* sin(BETA4) + sin(PHI_C) .* (cos( ...
    BETA4) .* cos(THETA_C) + sin(BETA4) .* sin(PHI_C) .* sin(THETA_C)));
IJ_11_51=0;
IJ_11_52=0;
IJ_11_53=0;
IJ_11_54=0;

IJ_12_1=0;
IJ_12_2=0;
IJ_12_3=1;
IJ_12_4=0;
IJ_12_5=LC4 .* (cos(PHI_C) .* sin(ALPHA4) .* sin(THETA_C) + cos(ALPHA4) .* ((-1) .* cos( ...
    THETA_C) .* sin(BETA4) + cos(BETA4) .* sin(PHI_C) .* sin(THETA_C)));
IJ_12_6=LC4 .* cos(THETA_C) .* ((-1) .* cos(ALPHA4) .* cos(BETA4) .* cos(PHI_C) + sin( ...
    ALPHA4) .* sin(PHI_C));
IJ_12_7=0;
IJ_12_8=0;
IJ_12_9=0;
IJ_12_10=0;
IJ_12_11=0;
IJ_12_12=0;
IJ_12_13=0;
IJ_12_14=0;
IJ_12_15=0;
IJ_12_16=0;
IJ_12_17=0;
IJ_12_18=0;
IJ_12_19=0;
IJ_12_20=0;
IJ_12_21=0;
IJ_12_22=0;
IJ_12_23=0;
IJ_12_24=0;
IJ_12_25=0;
IJ_12_26=0;
IJ_12_27=0;
IJ_12_28=0;
IJ_12_29=0;
IJ_12_30=0;
IJ_12_31=0;
IJ_12_32=0;
IJ_12_33=0;
IJ_12_34=0;
IJ_12_35=0;
IJ_12_36=(-1) .* cos(PHI_C) .* cos(THETA_C) .* sin(ALPHA4) + (-1) .* cos(ALPHA4) .* (cos( ...
    BETA4) .* cos(THETA_C) .* sin(PHI_C) + sin(BETA4) .* sin(THETA_C));
IJ_12_37=0;
IJ_12_38=0;
IJ_12_39=0;
IJ_12_40=0;

```

```

IJ_12_41=0;
IJ_12_42=0;
IJ_12_43=0;
IJ_12_44=LC4.*((-1).*cos(ALPHA4).*cos(PHI_C).*cos(THETA_C)+sin(ALPHA4).*cos( ...
    BETA4).*cos(THETA_C).*sin(PHI_C)+sin(BETA4).*sin(THETA_C));
IJ_12_45=0;
IJ_12_46=0;
IJ_12_47=0;
IJ_12_48=0;
IJ_12_49=0;
IJ_12_50=LC4.*cos(ALPHA4).*cos(THETA_C).*sin(BETA4).*sin(PHI_C)+(-1).*cos(BETA4) ...
    .*sin(THETA_C);
IJ_12_51=0;
IJ_12_52=0;
IJ_12_53=0;
IJ_12_54=0;

IJ_13_1=1;
IJ_13_2=0;
IJ_13_3=0;
IJ_13_4=(-1).*cos(THETA_C).*(L45.*cos(ALPHA5).*cos(BETA5)+LC4.*cos(ALPHA4).*sin( ...
    BETA4)).*sin(PHI_C)+(-1).*cos(PHI_C).*sin(PHI_C).*sin(THETA_C))+LC4.* ...
    cos(ALPHA4).*cos(BETA4)+(-1).*L45.*cos(ALPHA5).*sin(BETA5)).*(cos(PHI_C) ...
    .*cos(PHI_C)+sin(PHI_C).*sin(PHI_C).*sin(THETA_C));
IJ_13_5=(-1).*cos(PHI_C).*(cos(PHI_C).*cos(THETA_C).*(LC4.*sin(ALPHA4)+L45.*sin( ...
    ALPHA5))+L45.*cos(ALPHA5).*(-1).*cos(THETA_C).*sin(BETA5).*sin(PHI_C)+ ...
    cos(BETA5).*sin(THETA_C))+LC4.*cos(ALPHA4).*(cos(BETA4).*cos(THETA_C).* ...
    sin(PHI_C)+sin(BETA4).*sin(THETA_C)));
IJ_13_6=(-1).*LC4.*cos(ALPHA4).*cos(BETA4)+L45.*cos(ALPHA5).*sin(BETA5)).*(sin( ...
    PHI_C).*sin(PHI_C)+cos(PHI_C).*cos(PHI_C).*sin(THETA_C))+(-1).*LC4.* ...
    sin(ALPHA4)+L45.*sin(ALPHA5)).*(cos(PHI_C).*sin(PHI_C)+(-1).*cos(PHI_C) ...
    .*sin(PHI_C).*sin(THETA_C));
IJ_13_7=0;
IJ_13_8=0;
IJ_13_9=0;
IJ_13_10=0;
IJ_13_11=0;
IJ_13_12=0;
IJ_13_13=0;
IJ_13_14=0;
IJ_13_15=0;
IJ_13_16=0;
IJ_13_17=0;
IJ_13_18=0;
IJ_13_19=0;
IJ_13_20=0;
IJ_13_21=0;
IJ_13_22=0;
IJ_13_23=0;
IJ_13_24=0;
IJ_13_25=0;
IJ_13_26=0;
IJ_13_27=0;
IJ_13_28=0;
IJ_13_29=0;
IJ_13_30=0;
IJ_13_31=0;
IJ_13_32=0;
IJ_13_33=0;
IJ_13_34=0;
IJ_13_35=0;
IJ_13_36=(-1).*sin(ALPHA4).*(sin(PHI_C).*sin(PHI_C)+cos(PHI_C).*cos(PHI_C).*sin( ...
    THETA_C))+cos(ALPHA4).*(cos(BETA4).*cos(PHI_C).*sin(PHI_C)+cos(PHI_C)).*( ...
    cos(THETA_C).*sin(BETA4)+(-1).*cos(BETA4).*sin(PHI_C).*sin(THETA_C)));
IJ_13_37=(-1).*sin(ALPHA5).*(sin(PHI_C).*sin(PHI_C)+cos(PHI_C).*cos(PHI_C).*sin( ...
    THETA_C))+cos(ALPHA5).*(cos(BETA5).*cos(PHI_C).*cos(THETA_C)+sin(BETA5) ...
    .*((-1).*cos(PHI_C).*sin(PHI_C)+cos(PHI_C).*sin(PHI_C).*sin(THETA_C))); ...
IJ_13_38=0;
IJ_13_39=0;
IJ_13_40=0;

```

```

IJ_13_41=0;
IJ_13_42=0;
IJ_13_43=0;
IJ_13_44=(-1).*LC4.*((cos(BETA4).*cos(PHI_C).*sin(ALPHA4)+cos(ALPHA4).*sin(PHI_C) ...
).*sin(PHI_C)+cos(PHI_C).*cos(THETA_C).*sin(ALPHA4).*sin(BETA4)+cos( ...
ALPHA4).*cos(PHI_C)+(-1).*cos(BETA4).*sin(ALPHA4).*sin(PHI_C)).*sin( ...
THETA_C));
IJ_13_45=(-1).*L45.*(cos(BETA5).*cos(PHI_C).*cos(THETA_C).*sin(ALPHA5)+(-1).*cos( ...
PHI_C).*sin(ALPHA5).*sin(BETA5).*sin(PHI_C)+cos(ALPHA5).*sin(PHI_C).* ...
sin(PHI_C)+cos(PHI_C).*cos(ALPHA5).*cos(PHI_C)+sin(ALPHA5).*sin(BETA5) ...
.*sin(PHI_C)).*sin(THETA_C));
IJ_13_46=0;
IJ_13_47=0;
IJ_13_48=0;
IJ_13_49=0;
IJ_13_50=LC4.*cos(ALPHA4).*cos(BETA4).*cos(PHI_C).*cos(THETA_C)+sin(BETA4).*(( ...
-1).*cos(PHI_C).*sin(PHI_C)+cos(PHI_C).*sin(PHI_C).*sin(THETA_C));
IJ_13_51=(-1).*L45.*cos(ALPHA5).*cos(BETA5).*cos(PHI_C).*sin(PHI_C)+cos(PHI_C).* ...
(cos(THETA_C).*sin(BETA5)+(-1).*cos(BETA5).*sin(PHI_C).*sin(THETA_C))); ...
IJ_13_52=0;
IJ_13_53=0;
IJ_13_54=0;

IJ_14_1=0;
IJ_14_2=1;
IJ_14_3=0;
IJ_14_4=cos(PHI_C).*cos(THETA_C).*cos(ALPHA5).*cos(BETA5)+LC4.*cos(ALPHA4) ...
.*sin(BETA4)+(-1).*cos(PHI_C).*sin(PHI_C).*sin(THETA_C));
IJ_14_5=(-1).*L45.*cos(ALPHA5).*sin(BETA5).*cos(PHI_C).*sin(PHI_C)+ ...
cos(BETA5).*sin(THETA_C)+LC4.*cos(ALPHA4).*cos(BETA4).*cos(THETA_C).* ...
sin(PHI_C)+sin(BETA4).*sin(THETA_C));
IJ_14_6=(LC4.*cos(ALPHA4).*cos(BETA4)+(-1).*L45.*cos(ALPHA5).*sin(BETA5)).*cos( ...
PHI_C).*sin(PHI_C)+(-1).*cos(PHI_C).*sin(PHI_C).*sin(THETA_C))+LC4.* ...
sin(ALPHA4)+L45.*sin(ALPHA5)).*cos(PHI_C).*cos(PHI_C)+sin(PHI_C).*sin( ...
PHI_C).*sin(THETA_C));
IJ_14_7=0;
IJ_14_8=0;
IJ_14_9=0;
IJ_14_10=0;
IJ_14_11=0;
IJ_14_12=0;
IJ_14_13=0;
IJ_14_14=0;
IJ_14_15=0;
IJ_14_16=0;
IJ_14_17=0;
IJ_14_18=0;
IJ_14_19=0;
IJ_14_20=0;
IJ_14_21=0;
IJ_14_22=0;
IJ_14_23=0;
IJ_14_24=0;
IJ_14_25=0;
IJ_14_26=0;
IJ_14_27=0;
IJ_14_28=0;
IJ_14_29=0;
IJ_14_30=0;
IJ_14_31=0;
IJ_14_32=0;
IJ_14_33=0;
IJ_14_34=0;
IJ_14_35=0;
IJ_14_36=sin(ALPHA4).*cos(PHI_C).*sin(PHI_C)+(-1).*cos(PHI_C).*sin(PHI_C).*sin( ...
THETA_C))+cos(ALPHA4).*cos(THETA_C).*sin(BETA4).*sin(PHI_C)+(-1).*cos( ...
BETA4).*cos(PHI_C).*cos(PHI_C)+sin(PHI_C).*sin(PHI_C).*sin(THETA_C)); ...

```

```

IJ_14_37=sin(ALPHA5).*cos(PHI_C).*sin(PHI_C)+(-1).*cos(PHI_C).*sin(PHI_C).*sin( ...
  THETA_C))+cos(ALPHA5).*cos(PHI_C).*cos(PHI_C).*sin(BETA5)+sin(PHI_C).*sin( ...
  cos(BETA5).*cos(THETA_C)+sin(BETA5).*sin(PHI_C).*sin(THETA_C));
IJ_14_38=0;
IJ_14_39=0;
IJ_14_40=0;
IJ_14_41=0;
IJ_14_42=0;
IJ_14_43=0;
IJ_14_44=LC4.*((-1).*cos(THETA_C).*sin(ALPHA4).*sin(BETA4).*sin(PHI_C)+cos( ...
  ALPHA4).*cos(PHI_C).*sin(PHI_C)+(-1).*cos(PHI_C).*sin(PHI_C).*sin( ...
  THETA_C))+cos(BETA4).*sin(ALPHA4).*cos(PHI_C).*cos(PHI_C)+sin(PHI_C).* ...
  sin(PHI_C).*sin(THETA_C));
IJ_14_45=(-1).*L45.*((-1).*cos(ALPHA5).*cos(PHI_C).*sin(PHI_C)+sin(ALPHA5).*sin( ...
  PHI_C).*cos(BETA5).*cos(THETA_C)+sin(BETA5).*sin(PHI_C).*sin(THETA_C))+ ...
  cos(PHI_C).*cos(PHI_C).*sin(ALPHA5).*sin(BETA5)+cos(ALPHA5).*sin(PHI_C) ...
  .*sin(THETA_C));
IJ_14_46=0;
IJ_14_47=0;
IJ_14_48=0;
IJ_14_49=0;
IJ_14_50=LC4.*cos(ALPHA4).*cos(PHI_C).*cos(PHI_C).*sin(BETA4)+sin(PHI_C).*cos( ...
  BETA4).*cos(THETA_C)+sin(BETA4).*sin(PHI_C).*sin(THETA_C));
IJ_14_51=L45.*cos(ALPHA5).*((-1).*cos(THETA_C).*sin(BETA5).*sin(PHI_C)+cos(BETA5) ...
  .*cos(PHI_C).*cos(PHI_C)+sin(PHI_C).*sin(PHI_C).*sin(THETA_C));
IJ_14_52=0;
IJ_14_53=0;
IJ_14_54=0;

IJ_15_1=0;
IJ_15_2=0;
IJ_15_3=1;
IJ_15_4=0;
IJ_15_5=cos(PHI_C).(LC4.*sin(ALPHA4)+L45.*sin(ALPHA5)).*sin(THETA_C)+LC4.*cos( ...
  ALPHA4).*((-1).*cos(THETA_C).*sin(BETA4)+cos(BETA4).*sin(PHI_C).*sin( ...
  THETA_C))+(-1).*L45.*cos(ALPHA5).*cos(BETA5).*cos(THETA_C)+sin(BETA5).* ...
  sin(PHI_C).*sin(THETA_C));
IJ_15_6=cos(THETA_C).*((-1).*LC4.*cos(ALPHA4).*cos(BETA4).*cos(PHI_C)+L45.*cos( ...
  ALPHA5).*cos(PHI_C).*sin(BETA5)+(LC4.*sin(ALPHA4)+L45.*sin(ALPHA5)).* ...
  sin(PHI_C));
IJ_15_7=0;
IJ_15_8=0;
IJ_15_9=0;
IJ_15_10=0;
IJ_15_11=0;
IJ_15_12=0;
IJ_15_13=0;
IJ_15_14=0;
IJ_15_15=0;
IJ_15_16=0;
IJ_15_17=0;
IJ_15_18=0;
IJ_15_19=0;
IJ_15_20=0;
IJ_15_21=0;
IJ_15_22=0;
IJ_15_23=0;
IJ_15_24=0;
IJ_15_25=0;
IJ_15_26=0;
IJ_15_27=0;
IJ_15_28=0;
IJ_15_29=0;
IJ_15_30=0;
IJ_15_31=0;
IJ_15_32=0;
IJ_15_33=0;
IJ_15_34=0;
IJ_15_35=0;
IJ_15_36=(-1).*cos(PHI_C).*cos(THETA_C).*sin(ALPHA4)+(-1).*cos(ALPHA4).*cos( ...
  BETA4).*cos(THETA_C).*sin(PHI_C)+sin(BETA4).*sin(THETA_C));

```



```

IJ_15_37=(-1).*cos(PHI_C).*cos(THETA_C).*sin(ALPHA5)+cos(ALPHA5).*(cos(THETA_C).*...
sin(BETA5).*sin(PHI_C)+(-1).*cos(BETA5).*sin(THETA_C));
IJ_15_38=0;
IJ_15_39=0;
IJ_15_40=0;
IJ_15_41=0;
IJ_15_42=0;
IJ_15_43=0;
IJ_15_44=LC4.*((-1).*cos(ALPHA4).*cos(PHI_C).*cos(THETA_C)+sin(ALPHA4).*(cos(...
BETA4).*cos(THETA_C).*sin(PHI_C)+sin(BETA4).*sin(THETA_C)));
IJ_15_45=(-1).*L45.*cos(THETA_C).*(cos(ALPHA5).*cos(PHI_C)+sin(ALPHA5).*sin(...
BETA5).*sin(PHI_C))+L45.*cos(BETA5).*sin(ALPHA5).*sin(THETA_C);
IJ_15_46=0;
IJ_15_47=0;
IJ_15_48=0;
IJ_15_49=0;
IJ_15_50=LC4.*cos(ALPHA4).*(cos(THETA_C).*sin(BETA4).*sin(PHI_C)+(-1).*cos(BETA4)...
.*sin(THETA_C));
IJ_15_51=L45.*cos(ALPHA5).*(cos(BETA5).*cos(THETA_C).*sin(PHI_C)+sin(BETA5).*sin(...
THETA_C));
IJ_15_52=0;
IJ_15_53=0;
IJ_15_54=0;

IJ_16_1=1;
IJ_16_2=0;
IJ_16_3=0;
IJ_16_4=(-1).*LC6.*(cos(PHI_C).*cos(PSI_C).*sin(ALPHA6)+cos(ALPHA6).*cos(BETA6)...
.*cos(PSI_C).*sin(PHI_C)+(-1).*cos(ALPHA6).*cos(THETA_C).*sin(BETA6).*...
sin(PSI_C)+((-1).*cos(ALPHA6).*cos(BETA6).*cos(PHI_C)+sin(ALPHA6).*sin(...
PHI_C)).*sin(PSI_C).*sin(THETA_C));
IJ_16_5=LC6.*cos(PSI_C).*(cos(THETA_C).*sin(ALPHA6).*sin(PHI_C)+cos(ALPHA6).*((...
-1).*cos(BETA6).*cos(PHI_C).*cos(THETA_C)+sin(BETA6).*sin(THETA_C)));
IJ_16_6=LC6.*(sin(ALPHA6).*sin(PHI_C).*sin(PSI_C)+cos(PHI_C).*cos(PSI_C).*sin(...
THETA_C))+cos(ALPHA6).*cos(BETA6).*((-1).*cos(PHI_C).*sin(PSI_C)+cos(...
PSI_C).*sin(PHI_C).*sin(THETA_C));
IJ_16_7=0;
IJ_16_8=0;
IJ_16_9=0;
IJ_16_10=0;
IJ_16_11=0;
IJ_16_12=0;
IJ_16_13=0;
IJ_16_14=0;
IJ_16_15=0;
IJ_16_16=0;
IJ_16_17=0;
IJ_16_18=0;
IJ_16_19=0;
IJ_16_20=0;
IJ_16_21=0;
IJ_16_22=0;
IJ_16_23=0;
IJ_16_24=0;
IJ_16_25=0;
IJ_16_26=0;
IJ_16_27=0;
IJ_16_28=0;
IJ_16_29=0;
IJ_16_30=0;
IJ_16_31=0;
IJ_16_32=0;
IJ_16_33=0;
IJ_16_34=0;
IJ_16_35=0;
IJ_16_36=0;
IJ_16_37=0;
IJ_16_38=sin(ALPHA6).*((-1).*cos(PHI_C).*sin(PSI_C)+cos(PSI_C).*sin(PHI_C).*sin(...
THETA_C))+(-1).*cos(ALPHA6).*(cos(BETA6).*sin(PHI_C).*sin(PSI_C)+cos(...
PSI_C).*(cos(THETA_C).*sin(BETA6)+cos(BETA6).*cos(PHI_C).*sin(THETA_C)))...
;

```

```

IJ_16_39=0;
IJ_16_40=0;
IJ_16_41=0;
IJ_16_42=0;
IJ_16_43=0;
IJ_16_44=0;
IJ_16_45=0;
IJ_16_46=LC6.*((-1).*cos(ALPHA6).*cos(PHI_C)+cos(BETA6).*sin(ALPHA6).*sin(PHI_C) ...
).*sin(PHI_C)+cos(PHI_C).*(cos(THETA_C).*sin(ALPHA6).*sin(BETA6)+(cos( ...
BETA6).*cos(PHI_C).*sin(ALPHA6)+cos(ALPHA6).*sin(PHI_C)).*sin(THETA_C))) ...
;
IJ_16_47=0;
IJ_16_48=0;
IJ_16_49=0;
IJ_16_50=0;
IJ_16_51=0;
IJ_16_52=LC6.*cos(ALPHA6).*((-1).*cos(BETA6).*cos(PHI_C).*cos(THETA_C)+sin(BETA6) ...
).*(sin(PHI_C).*sin(PHI_C)+cos(PHI_C).*cos(PHI_C).*sin(THETA_C));
IJ_16_53=0;
IJ_16_54=0;

IJ_17_1=0;
IJ_17_2=1;
IJ_17_3=0;
IJ_17_4=(-1).*LC6.*(sin(ALPHA6).*(cos(PHI_C).*sin(PHI_C)+(-1).*cos(PHI_C).*sin( ...
PHI_C).*sin(THETA_C))+cos(ALPHA6).*(cos(BETA6).*sin(PHI_C).*sin(PHI_C)+ ...
cos(PHI_C).*(cos(THETA_C).*sin(BETA6)+cos(BETA6).*cos(PHI_C).*sin( ...
THETA_C))));
IJ_17_5=LC6.*sin(PHI_C).*(cos(THETA_C).*sin(ALPHA6).*sin(PHI_C)+cos(ALPHA6).*(( ...
-1).*cos(BETA6).*cos(PHI_C).*cos(THETA_C)+sin(BETA6).*sin(THETA_C)));
IJ_17_6=LC6.*(sin(ALPHA6).*((-1).*cos(PHI_C).*sin(PHI_C)+cos(PHI_C).*sin(PHI_C) ...
).*sin(THETA_C))+cos(ALPHA6).*cos(BETA6).*(cos(PHI_C).*cos(PHI_C)+sin( ...
PHI_C).*sin(PHI_C).*sin(THETA_C));
IJ_17_7=0;
IJ_17_8=0;
IJ_17_9=0;
IJ_17_10=0;
IJ_17_11=0;
IJ_17_12=0;
IJ_17_13=0;
IJ_17_14=0;
IJ_17_15=0;
IJ_17_16=0;
IJ_17_17=0;
IJ_17_18=0;
IJ_17_19=0;
IJ_17_20=0;
IJ_17_21=0;
IJ_17_22=0;
IJ_17_23=0;
IJ_17_24=0;
IJ_17_25=0;
IJ_17_26=0;
IJ_17_27=0;
IJ_17_28=0;
IJ_17_29=0;
IJ_17_30=0;
IJ_17_31=0;
IJ_17_32=0;
IJ_17_33=0;
IJ_17_34=0;
IJ_17_35=0;
IJ_17_36=0;
IJ_17_37=0;
IJ_17_38=cos(PHI_C).*cos(PHI_C).*sin(ALPHA6)+cos(ALPHA6).*cos(BETA6).*cos(PHI_C) ...
).*sin(PHI_C)+(-1).*cos(ALPHA6).*cos(THETA_C).*sin(BETA6).*sin(PHI_C)+(( ...
-1).*cos(ALPHA6).*cos(BETA6).*cos(PHI_C)+sin(ALPHA6).*sin(PHI_C)).*sin( ...
PHI_C).*sin(THETA_C);
IJ_17_39=0;
IJ_17_40=0;
IJ_17_41=0;

```

```

IJ_17_42=0;
IJ_17_43=0;
IJ_17_44=0;
IJ_17_45=0;
IJ_17_46=LC6.*(cos(ALPHA6).*(cos(PHI_C).*cos(PSI_C)+sin(PHI_C).*sin(PSI_C).*sin( ...
  THETA_C))+sin(ALPHA6).*(cos(THETA_C).*sin(BETA6).*sin(PSI_C)+cos(BETA6) ...
  .*(-1).*cos(PSI_C).*sin(PHI_C)+cos(PHI_C).*sin(PSI_C).*sin(THETA_C))); ...
IJ_17_47=0;
IJ_17_48=0;
IJ_17_49=0;
IJ_17_50=0;
IJ_17_51=0;
IJ_17_52=(-1).*LC6.*cos(ALPHA6).*(cos(PSI_C).*sin(BETA6).*sin(PHI_C)+sin(PSI_C).* ...
  (cos(BETA6).*cos(THETA_C)+(-1).*cos(PHI_C).*sin(BETA6).*sin(THETA_C))); ...
IJ_17_53=0;
IJ_17_54=0;

IJ_18_1=0;
IJ_18_2=0;
IJ_18_3=1;
IJ_18_4=0;
IJ_18_5=LC6.*((-1).*sin(ALPHA6).*sin(PHI_C).*sin(THETA_C)+cos(ALPHA6).*(cos( ...
  THETA_C).*sin(BETA6)+cos(BETA6).*cos(PHI_C).*sin(THETA_C)));
IJ_18_6=LC6.*cos(THETA_C).*(cos(PHI_C).*sin(ALPHA6)+cos(ALPHA6).*cos(BETA6).* ...
  sin(PHI_C));
IJ_18_7=0;
IJ_18_8=0;
IJ_18_9=0;
IJ_18_10=0;
IJ_18_11=0;
IJ_18_12=0;
IJ_18_13=0;
IJ_18_14=0;
IJ_18_15=0;
IJ_18_16=0;
IJ_18_17=0;
IJ_18_18=0;
IJ_18_19=0;
IJ_18_20=0;
IJ_18_21=0;
IJ_18_22=0;
IJ_18_23=0;
IJ_18_24=0;
IJ_18_25=0;
IJ_18_26=0;
IJ_18_27=0;
IJ_18_28=0;
IJ_18_29=0;
IJ_18_30=0;
IJ_18_31=0;
IJ_18_32=0;
IJ_18_33=0;
IJ_18_34=0;
IJ_18_35=0;
IJ_18_36=0;
IJ_18_37=0;
IJ_18_38=cos(THETA_C).*sin(ALPHA6).*sin(PHI_C)+cos(ALPHA6).*(-1).*cos(BETA6).* ...
  cos(PHI_C).*cos(THETA_C)+sin(BETA6).*sin(THETA_C));
IJ_18_39=0;
IJ_18_40=0;
IJ_18_41=0;
IJ_18_42=0;
IJ_18_43=0;
IJ_18_44=0;
IJ_18_45=0;
IJ_18_46=LC6.*(cos(BETA6).*cos(PHI_C).*cos(THETA_C).*sin(ALPHA6)+cos(ALPHA6).* ...
  cos(THETA_C).*sin(PHI_C)+(-1).*sin(ALPHA6).*sin(BETA6).*sin(THETA_C));
IJ_18_47=0;
IJ_18_48=0;
IJ_18_49=0;
IJ_18_50=0;

```

```

IJ_18_51=0;
IJ_18_52=LC6.*cos(ALPHA6).*(cos(PHI_C).*cos(THETA_C).*sin(BETA6)+cos(BETA6).*sin( ...
    THETA_C));
IJ_18_53=0;
IJ_18_54=0;

IJ_19_1=1;
IJ_19_2=0;
IJ_19_3=0;
IJ_19_4=cos(THETA_C).*((-1).*L67.*cos(ALPHA7).*cos(BETA7)+LC6.*cos(ALPHA6).*sin( ...
    BETA6)).*sin(PSI_C)+(-1).*(LC6.*cos(ALPHA6).*cos(BETA6)+L67.*cos(ALPHA7) ...
    .*sin(BETA7)).*(cos(PSI_C).*sin(PHI_C)+(-1).*cos(PHI_C).*sin(PSI_C).* ...
    sin(THETA_C))+(-1).*(LC6.*sin(ALPHA6)+L67.*sin(ALPHA7)).*(cos(PHI_C).* ...
    cos(PSI_C)+sin(PHI_C).*sin(PSI_C).*sin(THETA_C));
IJ_19_5=cos(PSI_C).*(cos(THETA_C).*(LC6.*sin(ALPHA6)+L67.*sin(ALPHA7)).*sin( ...
    PHI_C)+(-1).*L67.*cos(ALPHA7).*(cos(PHI_C).*cos(THETA_C).*sin(BETA7)+ ...
    cos(BETA7).*sin(THETA_C))+LC6.*cos(ALPHA6).*((-1).*cos(BETA6).*cos( ...
    PHI_C).*cos(THETA_C)+sin(BETA6).*sin(THETA_C)));
IJ_19_6=(LC6.*sin(ALPHA6)+L67.*sin(ALPHA7)).*(sin(PHI_C).*sin(PSI_C)+cos(PHI_C) ...
    .*cos(PSI_C).*sin(THETA_C))+(-1).*(LC6.*cos(ALPHA6).*cos(BETA6)+L67.* ...
    cos(ALPHA7).*sin(BETA7)).*(cos(PHI_C).*sin(PSI_C)+(-1).*cos(PSI_C).*sin( ...
    PHI_C).*sin(THETA_C));
IJ_19_7=0;
IJ_19_8=0;
IJ_19_9=0;
IJ_19_10=0;
IJ_19_11=0;
IJ_19_12=0;
IJ_19_13=0;
IJ_19_14=0;
IJ_19_15=0;
IJ_19_16=0;
IJ_19_17=0;
IJ_19_18=0;
IJ_19_19=0;
IJ_19_20=0;
IJ_19_21=0;
IJ_19_22=0;
IJ_19_23=0;
IJ_19_24=0;
IJ_19_25=0;
IJ_19_26=0;
IJ_19_27=0;
IJ_19_28=0;
IJ_19_29=0;
IJ_19_30=0;
IJ_19_31=0;
IJ_19_32=0;
IJ_19_33=0;
IJ_19_34=0;
IJ_19_35=0;
IJ_19_36=0;
IJ_19_37=0;
IJ_19_38=sin(ALPHA6).*((-1).*cos(PHI_C).*sin(PSI_C)+cos(PSI_C).*sin(PHI_C).*sin( ...
    THETA_C))+(-1).*cos(ALPHA6).*(cos(BETA6).*sin(PHI_C).*sin(PSI_C)+cos( ...
    PSI_C).*(cos(THETA_C).*sin(BETA6)+cos(BETA6).*cos(PHI_C).*sin(THETA_C))) ...
    ;
IJ_19_39=sin(ALPHA7).*((-1).*cos(PHI_C).*sin(PSI_C)+cos(PSI_C).*sin(PHI_C).*sin( ...
    THETA_C))+cos(ALPHA7).*(cos(BETA7).*cos(PSI_C).*cos(THETA_C)+(-1).*sin( ...
    BETA7).*(sin(PHI_C).*sin(PSI_C)+cos(PHI_C).*cos(PSI_C).*sin(THETA_C))); ...
IJ_19_40=0;
IJ_19_41=0;
IJ_19_42=0;
IJ_19_43=0;
IJ_19_44=0;
IJ_19_45=0;
IJ_19_46=LC6.*(((-1).*cos(ALPHA6).*cos(PHI_C)+cos(BETA6).*sin(ALPHA6).*sin(PHI_C) ...
    ).*sin(PSI_C)+cos(PSI_C).*(cos(THETA_C).*sin(ALPHA6).*sin(BETA6)+cos( ...
    BETA6).*cos(PHI_C).*sin(ALPHA6)+cos(ALPHA6).*sin(PHI_C)).*sin(THETA_C))) ...
    ;
IJ_19_47=L67.*((-1).*cos(BETA7).*cos(PSI_C).*cos(THETA_C).*sin(ALPHA7)+sin( ...

```

```

ALPHA7).*sin(BETA7).*sin(PHI_C).*sin(PSI_C)+cos(PHI_C).*cos(PSI_C).* ...
sin(THETA_C))+cos(ALPHA7).*((-1).*cos(PHI_C).*sin(PSI_C)+cos(PSI_C).* ...
sin(PHI_C).*sin(THETA_C)));
IJ_19_48=0;
IJ_19_49=0;
IJ_19_50=0;
IJ_19_51=0;
IJ_19_52=LC6.*cos(ALPHA6).*((-1).*cos(BETA6).*cos(PSI_C).*cos(THETA_C)+sin(BETA6) ...
.*sin(PHI_C).*sin(PSI_C)+cos(PHI_C).*cos(PSI_C).*sin(THETA_C)));
IJ_19_53=(-1).*L67.*cos(ALPHA7).*cos(BETA7).*sin(PHI_C).*sin(PSI_C)+cos(PSI_C).* ...
(cos(THETA_C).*sin(BETA7)+cos(BETA7).*cos(PHI_C).*sin(THETA_C)));
IJ_19_54=0;

IJ_20_1=0;
IJ_20_2=1;
IJ_20_3=0;
IJ_20_4=cos(PSI_C).*cos(THETA_C).(L67.*cos(ALPHA7).*cos(BETA7)+(-1).*LC6.*cos( ...
ALPHA6).*sin(BETA6))+(-1).(LC6.*cos(ALPHA6).*cos(BETA6)+L67.*cos( ...
ALPHA7).*sin(BETA7)).*sin(PHI_C).*sin(PSI_C)+cos(PHI_C).*cos(PSI_C).* ...
sin(THETA_C))+LC6.*sin(ALPHA6)+L67.*sin(ALPHA7)).*((-1).*cos(PHI_C).* ...
sin(PSI_C)+cos(PSI_C).*sin(PHI_C).*sin(THETA_C)));
IJ_20_5=sin(PSI_C).(cos(THETA_C).(LC6.*sin(ALPHA6)+L67.*sin(ALPHA7)).*sin( ...
PHI_C)+(-1).*L67.*cos(ALPHA7).(cos(PHI_C).*cos(THETA_C).*sin(BETA7)+ ...
cos(BETA7).*sin(THETA_C))+LC6.*cos(ALPHA6)).*((-1).*cos(BETA6).*cos( ...
PHI_C).*cos(THETA_C)+sin(BETA6).*sin(THETA_C)));
IJ_20_6=(LC6.*sin(ALPHA6)+L67.*sin(ALPHA7)).*((-1).*cos(PSI_C).*sin(PHI_C)+cos( ...
PHI_C).*sin(PSI_C).*sin(THETA_C))+LC6.*cos(ALPHA6).*cos(BETA6)+L67.* ...
cos(ALPHA7).*sin(BETA7)).*cos(PHI_C).*cos(PSI_C)+sin(PHI_C).*sin(PSI_C) ...
.*sin(THETA_C));
IJ_20_7=0;
IJ_20_8=0;
IJ_20_9=0;
IJ_20_10=0;
IJ_20_11=0;
IJ_20_12=0;
IJ_20_13=0;
IJ_20_14=0;
IJ_20_15=0;
IJ_20_16=0;
IJ_20_17=0;
IJ_20_18=0;
IJ_20_19=0;
IJ_20_20=0;
IJ_20_21=0;
IJ_20_22=0;
IJ_20_23=0;
IJ_20_24=0;
IJ_20_25=0;
IJ_20_26=0;
IJ_20_27=0;
IJ_20_28=0;
IJ_20_29=0;
IJ_20_30=0;
IJ_20_31=0;
IJ_20_32=0;
IJ_20_33=0;
IJ_20_34=0;
IJ_20_35=0;
IJ_20_36=0;
IJ_20_37=0;
IJ_20_38=cos(PHI_C).*cos(PSI_C).*sin(ALPHA6)+cos(ALPHA6).*cos(BETA6).*cos(PSI_C) ...
.*sin(PHI_C)+(-1).*cos(ALPHA6).*cos(THETA_C).*sin(BETA6).*sin(PSI_C)+(( ...
-1).*cos(ALPHA6).*cos(BETA6).*cos(PHI_C)+sin(ALPHA6).*sin(PHI_C)).*sin( ...
PSI_C).*sin(THETA_C));
IJ_20_39=cos(PHI_C).*cos(PSI_C).*sin(ALPHA7)+cos(ALPHA7).*cos(PSI_C).*sin(BETA7) ...
.*sin(PHI_C)+cos(ALPHA7).*cos(BETA7).*cos(THETA_C).*sin(PSI_C)+((-1).* ...
cos(ALPHA7).*cos(PHI_C).*sin(BETA7)+sin(ALPHA7).*sin(PHI_C)).*sin(PSI_C) ...
.*sin(THETA_C));
IJ_20_40=0;
IJ_20_41=0;
IJ_20_42=0;

```

```

IJ_20_43=0;
IJ_20_44=0;
IJ_20_45=0;
IJ_20_46=LC6.*(cos(ALPHA6).*(cos(PHI_C).*cos(PSI_C)+sin(PHI_C).*sin(PSI_C).*sin( ...
    THETA_C))+sin(ALPHA6).*(cos(THETA_C).*sin(BETA6).*sin(PSI_C)+cos(BETA6) ...
    .*(-1).*cos(PSI_C).*sin(PHI_C)+cos(PHI_C).*sin(PSI_C).*sin(THETA_C))); ...
IJ_20_47=L67.*(cos(ALPHA7).*(cos(PHI_C).*cos(PSI_C)+sin(PHI_C).*sin(PSI_C).*sin( ...
    THETA_C))+(-1).*sin(ALPHA7).*(cos(PSI_C).*sin(BETA7).*sin(PHI_C)+sin( ...
    PSI_C).*(cos(BETA7).*cos(THETA_C)+(-1).*cos(PHI_C).*sin(BETA7).*sin( ...
    THETA_C)));
IJ_20_48=0;
IJ_20_49=0;
IJ_20_50=0;
IJ_20_51=0;
IJ_20_52=(-1).*LC6.*cos(ALPHA6).*(cos(PSI_C).*sin(BETA6).*sin(PHI_C)+sin(PSI_C).* ...
    (cos(BETA6).*cos(THETA_C)+(-1).*cos(PHI_C).*sin(BETA6).*sin(THETA_C))); ...
IJ_20_53=(-1).*L67.*cos(ALPHA7).*(cos(THETA_C).*sin(BETA7).*sin(PSI_C)+cos(BETA7) ...
    .*(-1).*cos(PSI_C).*sin(PHI_C)+cos(PHI_C).*sin(PSI_C).*sin(THETA_C))); ...
IJ_20_54=0;

IJ_21_1=0;
IJ_21_2=0;
IJ_21_3=1;
IJ_21_4=0;
IJ_21_5=(-1).*(LC6.*sin(ALPHA6)+L67.*sin(ALPHA7)).*sin(PHI_C).*sin(THETA_C)+ ...
    LC6.*cos(ALPHA6).*(cos(THETA_C).*sin(BETA6)+cos(BETA6).*cos(PHI_C).*sin( ...
    THETA_C))+L67.*cos(ALPHA7).*(-1).*cos(BETA7).*cos(THETA_C)+cos(PHI_C).* ...
    sin(BETA7).*sin(THETA_C));
IJ_21_6=cos(THETA_C).*(cos(PHI_C).*(LC6.*sin(ALPHA6)+L67.*sin(ALPHA7)))+(LC6.* ...
    cos(ALPHA6).*cos(BETA6)+L67.*cos(ALPHA7).*sin(BETA7)).*sin(PHI_C));
IJ_21_7=0;
IJ_21_8=0;
IJ_21_9=0;
IJ_21_10=0;
IJ_21_11=0;
IJ_21_12=0;
IJ_21_13=0;
IJ_21_14=0;
IJ_21_15=0;
IJ_21_16=0;
IJ_21_17=0;
IJ_21_18=0;
IJ_21_19=0;
IJ_21_20=0;
IJ_21_21=0;
IJ_21_22=0;
IJ_21_23=0;
IJ_21_24=0;
IJ_21_25=0;
IJ_21_26=0;
IJ_21_27=0;
IJ_21_28=0;
IJ_21_29=0;
IJ_21_30=0;
IJ_21_31=0;
IJ_21_32=0;
IJ_21_33=0;
IJ_21_34=0;
IJ_21_35=0;
IJ_21_36=0;
IJ_21_37=0;
IJ_21_38=cos(THETA_C).*sin(ALPHA6).*sin(PHI_C)+cos(ALPHA6).*(-1).*cos(BETA6).* ...
    cos(PHI_C).*cos(THETA_C)+sin(BETA6).*sin(THETA_C));
IJ_21_39=cos(THETA_C).*sin(ALPHA7).*sin(PHI_C)+(-1).*cos(ALPHA7).*(cos(PHI_C).* ...
    cos(THETA_C).*sin(BETA7)+cos(BETA7).*sin(THETA_C));
IJ_21_40=0;
IJ_21_41=0;
IJ_21_42=0;
IJ_21_43=0;
IJ_21_44=0;
IJ_21_45=0;

```

```

IJ_21_46=LC6.*(cos(BETA6).*cos(PHI_C).*cos(THETA_C).*sin(ALPHA6)+cos(ALPHA6).* ...
cos(THETA_C).*sin(PHI_C)+(-1).*sin(ALPHA6).*sin(BETA6).*sin(THETA_C));
IJ_21_47=L67.*(cos(PHI_C).*cos(THETA_C).*sin(ALPHA7).*sin(BETA7)+cos(ALPHA7).* ...
cos(THETA_C).*sin(PHI_C)+cos(BETA7).*sin(ALPHA7).*sin(THETA_C));
IJ_21_48=0;
IJ_21_49=0;
IJ_21_50=0;
IJ_21_51=0;
IJ_21_52=LC6.*cos(ALPHA6).*cos(PHI_C).*cos(THETA_C).*sin(BETA6)+cos(BETA6).*sin( ...
THETA_C));
IJ_21_53=L67.*cos(ALPHA7).*((-1).*cos(BETA7).*cos(PHI_C).*cos(THETA_C)+sin(BETA7) ...
.*sin(THETA_C));
IJ_21_54=0;

IJ_22_1=1;
IJ_22_2=0;
IJ_22_3=0;
IJ_22_4=(-1).*LC8.*(cos(PHI_C).*cos(PHI_C+ZETA)+sin(PHI_C).*sin(THETA_C).*sin( ...
PHI_C+ZETA));
IJ_22_5=LC8.*cos(PHI_C).*cos(THETA_C).*sin(PHI_C+ZETA);
IJ_22_6=LC8.*(cos(PHI_C).*cos(PHI_C+ZETA).*sin(THETA_C)+sin(PHI_C).*sin(PHI_C+ ...
ZETA));
IJ_22_7=0;
IJ_22_8=0;
IJ_22_9=0;
IJ_22_10=0;
IJ_22_11=0;
IJ_22_12=0;
IJ_22_13=0;
IJ_22_14=0;
IJ_22_15=0;
IJ_22_16=0;
IJ_22_17=0;
IJ_22_18=0;
IJ_22_19=0;
IJ_22_20=0;
IJ_22_21=0;
IJ_22_22=0;
IJ_22_23=0;
IJ_22_24=0;
IJ_22_25=0;
IJ_22_26=0;
IJ_22_27=0;
IJ_22_28=0;
IJ_22_29=0;
IJ_22_30=0;
IJ_22_31=0;
IJ_22_32=0;
IJ_22_33=0;
IJ_22_34=0;
IJ_22_35=0;
IJ_22_36=0;
IJ_22_37=0;
IJ_22_38=0;
IJ_22_39=0;
IJ_22_40=(-1).*cos(PHI_C+ZETA).*sin(PHI_C)+cos(PHI_C).*sin(THETA_C).*sin(PHI_C+ ...
ZETA);
IJ_22_41=0;
IJ_22_42=LC8.*(cos(PHI_C).*cos(PHI_C+ZETA).*sin(THETA_C)+sin(PHI_C).*sin(PHI_C+ ...
ZETA));
IJ_22_43=0;
IJ_22_44=0;
IJ_22_45=0;
IJ_22_46=0;
IJ_22_47=0;
IJ_22_48=0;
IJ_22_49=0;
IJ_22_50=0;
IJ_22_51=0;
IJ_22_52=0;
IJ_22_53=0;

```

```

IJ_22_54=0;

IJ_23_1=0;
IJ_23_2=1;
IJ_23_3=0;
IJ_23_4=(-1).*LC8.*cos(PHI_C+ZETA).*sin(PSI_C)+LC8.*cos(PSI_C).*sin(THETA_C).* ...
    sin(PHI_C+ZETA);
IJ_23_5=LC8.*cos(THETA_C).*sin(PSI_C).*sin(PHI_C+ZETA);
IJ_23_6=LC8.*(cos(PHI_C+ZETA).*sin(PSI_C).*sin(THETA_C)+(-1).*cos(PSI_C).*sin( ...
    PHI_C+ZETA));
IJ_23_7=0;
IJ_23_8=0;
IJ_23_9=0;
IJ_23_10=0;
IJ_23_11=0;
IJ_23_12=0;
IJ_23_13=0;
IJ_23_14=0;
IJ_23_15=0;
IJ_23_16=0;
IJ_23_17=0;
IJ_23_18=0;
IJ_23_19=0;
IJ_23_20=0;
IJ_23_21=0;
IJ_23_22=0;
IJ_23_23=0;
IJ_23_24=0;
IJ_23_25=0;
IJ_23_26=0;
IJ_23_27=0;
IJ_23_28=0;
IJ_23_29=0;
IJ_23_30=0;
IJ_23_31=0;
IJ_23_32=0;
IJ_23_33=0;
IJ_23_34=0;
IJ_23_35=0;
IJ_23_36=0;
IJ_23_37=0;
IJ_23_38=0;
IJ_23_39=0;
IJ_23_40=cos(PSI_C).*cos(PHI_C+ZETA)+sin(PSI_C).*sin(THETA_C).*sin(PHI_C+ZETA);
IJ_23_41=0;
IJ_23_42=LC8.*(cos(PHI_C+ZETA).*sin(PSI_C).*sin(THETA_C)+(-1).*cos(PSI_C).*sin( ...
    PHI_C+ZETA));
IJ_23_43=0;
IJ_23_44=0;
IJ_23_45=0;
IJ_23_46=0;
IJ_23_47=0;
IJ_23_48=0;
IJ_23_49=0;
IJ_23_50=0;
IJ_23_51=0;
IJ_23_52=0;
IJ_23_53=0;
IJ_23_54=0;

IJ_24_1=0;
IJ_24_2=0;
IJ_24_3=1;
IJ_24_4=0;
IJ_24_5=(-1).*LC8.*sin(THETA_C).*sin(PHI_C+ZETA);
IJ_24_6=LC8.*cos(THETA_C).*cos(PHI_C+ZETA);
IJ_24_7=0;
IJ_24_8=0;
IJ_24_9=0;
IJ_24_10=0;
IJ_24_11=0;

```



```

IJ_24_12=0;
IJ_24_13=0;
IJ_24_14=0;
IJ_24_15=0;
IJ_24_16=0;
IJ_24_17=0;
IJ_24_18=0;
IJ_24_19=0;
IJ_24_20=0;
IJ_24_21=0;
IJ_24_22=0;
IJ_24_23=0;
IJ_24_24=0;
IJ_24_25=0;
IJ_24_26=0;
IJ_24_27=0;
IJ_24_28=0;
IJ_24_29=0;
IJ_24_30=0;
IJ_24_31=0;
IJ_24_32=0;
IJ_24_33=0;
IJ_24_34=0;
IJ_24_35=0;
IJ_24_36=0;
IJ_24_37=0;
IJ_24_38=0;
IJ_24_39=0;
IJ_24_40=cos (THETA_C) .*sin (PHI_C+ZETA);
IJ_24_41=0;
IJ_24_42=LC8 .*cos (THETA_C) .*cos (PHI_C+ZETA);
IJ_24_43=0;
IJ_24_44=0;
IJ_24_45=0;
IJ_24_46=0;
IJ_24_47=0;
IJ_24_48=0;
IJ_24_49=0;
IJ_24_50=0;
IJ_24_51=0;
IJ_24_52=0;
IJ_24_53=0;
IJ_24_54=0;

IJ_25_1=1;
IJ_25_2=0;
IJ_25_3=0;
IJ_25_4=(-1) .*L89 .*cos (ALPHA9) .*cos (BETA9) .*cos (THETA_C) .*sin (PSI_C)+(-1) .* ( ...
    LC8 .*cos (ZETA)+L89 .*sin (ALPHA9)) .* (cos (PHI_C) .*cos (PSI_C)+sin (PHI_C) .* ...
    sin (PSI_C) .*sin (THETA_C)) +(cos (PSI_C) .*sin (PHI_C)+(-1) .*cos (PHI_C) .*sin ( ...
    PSI_C) .*sin (THETA_C)) .* ((-1) .*L89 .*cos (ALPHA9) .*sin (BETA9)+LC8 .*sin ( ...
    ZETA));
IJ_25_5=cos (PSI_C) .* ((-1) .*L89 .*cos (ALPHA9) .* (cos (PHI_C) .*cos (THETA_C) .*sin ( ...
    BETA9)+cos (BETA9) .*sin (THETA_C))+cos (THETA_C) .* (L89 .*sin (ALPHA9) .*sin ( ...
    PHI_C)+LC8 .*sin (PHI_C+ZETA));
IJ_25_6=(LC8 .*cos (ZETA)+L89 .*sin (ALPHA9)) .* (sin (PHI_C) .*sin (PSI_C)+cos (PHI_C) .* ...
    cos (PSI_C) .*sin (THETA_C)) +(cos (PHI_C) .*sin (PSI_C)+(-1) .*cos (PSI_C) .*sin ( ...
    PHI_C) .*sin (THETA_C)) .* ((-1) .*L89 .*cos (ALPHA9) .*sin (BETA9)+LC8 .*sin ( ...
    ZETA));
IJ_25_7=0;
IJ_25_8=0;
IJ_25_9=0;
IJ_25_10=0;
IJ_25_11=0;
IJ_25_12=0;
IJ_25_13=0;
IJ_25_14=0;
IJ_25_15=0;
IJ_25_16=0;
IJ_25_17=0;
IJ_25_18=0;

```

```

IJ_25_19=0;
IJ_25_20=0;
IJ_25_21=0;
IJ_25_22=0;
IJ_25_23=0;
IJ_25_24=0;
IJ_25_25=0;
IJ_25_26=0;
IJ_25_27=0;
IJ_25_28=0;
IJ_25_29=0;
IJ_25_30=0;
IJ_25_31=0;
IJ_25_32=0;
IJ_25_33=0;
IJ_25_34=0;
IJ_25_35=0;
IJ_25_36=0;
IJ_25_37=0;
IJ_25_38=0;
IJ_25_39=0;
IJ_25_40=(-1).*cos(PHI_C+ZETA).*sin(PSI_C)+cos(PSI_C).*sin(THETA_C).*sin(PHI_C+ ...
ZETA);
IJ_25_41=sin(ALPHA9).*((-1).*cos(PHI_C).*sin(PSI_C)+cos(PSI_C).*sin(PHI_C).*sin( ...
THETA_C))+cos(ALPHA9).(cos(BETA9).*cos(PSI_C).*cos(THETA_C)+(-1).*sin( ...
BETA9).*sin(PHI_C).*sin(PSI_C)+cos(PHI_C).*cos(PSI_C).*sin(THETA_C)); ...
IJ_25_42=LC8.*(cos(PSI_C).*cos(PHI_C+ZETA).*sin(THETA_C)+sin(PSI_C).*sin(PHI_C+ ...
ZETA));
IJ_25_43=0;
IJ_25_44=0;
IJ_25_45=0;
IJ_25_46=0;
IJ_25_47=0;
IJ_25_48=L89.*((-1).*cos(BETA9).*cos(PSI_C).*cos(THETA_C).*sin(ALPHA9)+sin( ...
ALPHA9).*sin(BETA9).*sin(PHI_C).*sin(PSI_C)+cos(PHI_C).*cos(PSI_C).* ...
sin(THETA_C))+cos(ALPHA9).*((-1).*cos(PHI_C).*sin(PSI_C)+cos(PHI_C).* ...
sin(PHI_C).*sin(THETA_C));
IJ_25_49=0;
IJ_25_50=0;
IJ_25_51=0;
IJ_25_52=0;
IJ_25_53=0;
IJ_25_54=(-1).*L89.*cos(ALPHA9).(cos(BETA9).*sin(PHI_C).*sin(PSI_C)+cos(PHI_C).* ...
(cos(THETA_C).*sin(BETA9)+cos(BETA9).*cos(PHI_C).*sin(THETA_C)));

IJ_26_1=0;
IJ_26_2=1;
IJ_26_3=0;
IJ_26_4=L89.*cos(ALPHA9).*cos(BETA9).*cos(PSI_C).*cos(THETA_C)+(LC8.*cos(ZETA)+ ...
L89.*sin(ALPHA9)).*((-1).*cos(PHI_C).*sin(PSI_C)+cos(PHI_C).*sin(PHI_C) ...
.*sin(THETA_C))+sin(PHI_C).*sin(PSI_C)+cos(PHI_C).*cos(PSI_C).*sin( ...
THETA_C)).*((-1).*L89.*cos(ALPHA9).*sin(BETA9)+LC8.*sin(ZETA));
IJ_26_5=sin(PHI_C).*((-1).*L89.*cos(ALPHA9)).*(cos(PHI_C).*cos(THETA_C).*sin( ...
BETA9)+cos(BETA9).*sin(THETA_C))+cos(THETA_C).*(L89.*sin(ALPHA9).*sin( ...
PHI_C)+LC8.*sin(PHI_C+ZETA));
IJ_26_6=(LC8.*cos(ZETA)+L89.*sin(ALPHA9)).*((-1).*cos(PHI_C).*sin(PHI_C)+cos( ...
PHI_C).*sin(PSI_C).*sin(THETA_C))+cos(PHI_C).*cos(PHI_C)+sin(PHI_C).* ...
sin(PHI_C).*sin(THETA_C)).*(L89.*cos(ALPHA9).*sin(BETA9)+(-1).*LC8.*sin( ...
ZETA));
IJ_26_7=0;
IJ_26_8=0;
IJ_26_9=0;
IJ_26_10=0;
IJ_26_11=0;
IJ_26_12=0;
IJ_26_13=0;
IJ_26_14=0;
IJ_26_15=0;
IJ_26_16=0;
IJ_26_17=0;
IJ_26_18=0;

```

```

IJ_26_19=0;
IJ_26_20=0;
IJ_26_21=0;
IJ_26_22=0;
IJ_26_23=0;
IJ_26_24=0;
IJ_26_25=0;
IJ_26_26=0;
IJ_26_27=0;
IJ_26_28=0;
IJ_26_29=0;
IJ_26_30=0;
IJ_26_31=0;
IJ_26_32=0;
IJ_26_33=0;
IJ_26_34=0;
IJ_26_35=0;
IJ_26_36=0;
IJ_26_37=0;
IJ_26_38=0;
IJ_26_39=0;
IJ_26_40=cos(PHI_C)*cos(PHI_C+ZETA)+sin(PHI_C)*sin(THETA_C)*sin(PHI_C+ZETA);
IJ_26_41=cos(PHI_C)*cos(PHI_C)*sin(ALPHA9)+cos(ALPHA9)*cos(PHI_C)*sin(BETA9) ...
    *sin(PHI_C)+cos(ALPHA9)*cos(BETA9)*cos(THETA_C)*sin(PHI_C)+(-1)* ...
    cos(ALPHA9)*cos(PHI_C)*sin(BETA9)+sin(ALPHA9)*sin(PHI_C)*sin(PHI_C) ...
    *sin(THETA_C);
IJ_26_42=LC8*(cos(PHI_C+ZETA)*sin(PHI_C)*sin(THETA_C)+(-1)*cos(PHI_C)*sin( ...
    PHI_C+ZETA));
IJ_26_43=0;
IJ_26_44=0;
IJ_26_45=0;
IJ_26_46=0;
IJ_26_47=0;
IJ_26_48=L89*(cos(ALPHA9)*(cos(PHI_C)*cos(PHI_C)+sin(PHI_C)*sin(PHI_C)*sin( ...
    THETA_C))+(-1)*sin(ALPHA9)*(cos(PHI_C)*sin(BETA9)*sin(PHI_C)+sin( ...
    PHI_C)*(cos(BETA9)*cos(THETA_C)+(-1)*cos(PHI_C)*sin(BETA9)*sin( ...
    THETA_C)));
IJ_26_49=0;
IJ_26_50=0;
IJ_26_51=0;
IJ_26_52=0;
IJ_26_53=0;
IJ_26_54=(-1)*L89*cos(ALPHA9)*(cos(THETA_C)*sin(BETA9)*sin(PHI_C)+cos(BETA9) ...
    *(-1)*cos(PHI_C)*sin(PHI_C)+cos(PHI_C)*sin(PHI_C)*sin(THETA_C)); ...

IJ_27_1=0;
IJ_27_2=0;
IJ_27_3=1;
IJ_27_4=0;
IJ_27_5=L89*cos(ALPHA9)*((-1)*cos(BETA9)*cos(THETA_C)+cos(PHI_C)*sin(BETA9) ...
    *sin(THETA_C))+(-1)*sin(THETA_C)*(L89*sin(ALPHA9)*sin(PHI_C)+LC8* ...
    sin(PHI_C+ZETA));
IJ_27_6=cos(THETA_C)*(cos(PHI_C)*(LC8*cos(ZETA)+L89*sin(ALPHA9))+sin(PHI_C) ...
    *(L89*cos(ALPHA9)*sin(BETA9)+(-1)*LC8*sin(ZETA)));
IJ_27_7=0;
IJ_27_8=0;
IJ_27_9=0;
IJ_27_10=0;
IJ_27_11=0;
IJ_27_12=0;
IJ_27_13=0;
IJ_27_14=0;
IJ_27_15=0;
IJ_27_16=0;
IJ_27_17=0;
IJ_27_18=0;
IJ_27_19=0;
IJ_27_20=0;
IJ_27_21=0;
IJ_27_22=0;
IJ_27_23=0;

```

```

IJ_27_24=0;
IJ_27_25=0;
IJ_27_26=0;
IJ_27_27=0;
IJ_27_28=0;
IJ_27_29=0;
IJ_27_30=0;
IJ_27_31=0;
IJ_27_32=0;
IJ_27_33=0;
IJ_27_34=0;
IJ_27_35=0;
IJ_27_36=0;
IJ_27_37=0;
IJ_27_38=0;
IJ_27_39=0;
IJ_27_40=cos(THETA_C).*sin(PHI_C+ZETA);
IJ_27_41=cos(THETA_C).*sin(ALPHA9).*sin(PHI_C)+(-1).*cos(ALPHA9).*cos(PHI_C).*...
cos(THETA_C).*sin(BETA9)+cos(BETA9).*sin(THETA_C));
IJ_27_42=LC8.*cos(THETA_C).*cos(PHI_C+ZETA);
IJ_27_43=0;
IJ_27_44=0;
IJ_27_45=0;
IJ_27_46=0;
IJ_27_47=0;
IJ_27_48=L89.*(cos(PHI_C).*cos(THETA_C).*sin(ALPHA9).*sin(BETA9)+cos(ALPHA9).*...
cos(THETA_C).*sin(PHI_C)+cos(BETA9).*sin(ALPHA9).*sin(THETA_C));
IJ_27_49=0;
IJ_27_50=0;
IJ_27_51=0;
IJ_27_52=0;
IJ_27_53=0;
IJ_27_54=L89.*cos(ALPHA9).*((-1).*cos(BETA9).*cos(PHI_C).*cos(THETA_C)+sin(BETA9)...
.*sin(THETA_C));

IJ_28_1=0;
IJ_28_2=0;
IJ_28_3=0;
IJ_28_4=1;
IJ_28_5=(cos(PHI_C).*cos(THETA_1).*sin(PHI_C)+sin(PHI_C).*sin(THETA_1)).*(cos(...
PHI_C).*cos(THETA_C).*sin(THETA_1)+cos(THETA_1).*((-1).*cos(THETA_C)).*...
sin(PHI_C).*sin(PHI_C)+cos(PHI_C).*sin(THETA_C)).*(cos(PHI_C).^2.*cos(...
THETA_1).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C).*...
sin(PHI_C).*sin(2.*THETA_1)+2.*cos(PHI_C).*cos(THETA_1).*cos(THETA_C)).*(...
cos(THETA_1).*sin(PHI_C).*sin(PHI_C)+(-1).*cos(PHI_C).*sin(THETA_1)).*...
sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_1).^2+cos(THETA_1).^2.*sin(PHI_C)...
.^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_1).^2.*sin(PHI_C).^2+...
sin(THETA_1).^2.*sin(THETA_C).^2)).^(-1);
IJ_28_6=(1/4).* (4.*cos(PHI_C).*cos(THETA_1).*cos(THETA_C).*((-1).*cos(THETA_1)).*...
sin(PHI_C).*sin(PHI_C)+cos(PHI_C).*sin(THETA_1))+((-3)+cos(2.*PSI_1)+2.*...
cos(PSI_1).^2.*cos(2.*THETA_1)).*sin(THETA_C)).*(cos(PSI_1).^2.*cos(...
THETA_1).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C)).*...
sin(PSI_1).*sin(2.*THETA_1)+2.*cos(PSI_1).*cos(THETA_1).*cos(THETA_C)).*(...
cos(THETA_1).*sin(PHI_C).*sin(PHI_C)+(-1).*cos(PHI_C).*sin(THETA_1)).*...
sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_1).^2+cos(THETA_1).^2.*sin(PHI_C)...
.^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_1).^2.*sin(PHI_C).^2+...
sin(THETA_1).^2.*sin(THETA_C).^2)).^(-1);
IJ_28_7=(1/2).* (2.*cos(PHI_C).*cos(THETA_1).^2.*cos(THETA_C)+cos(THETA_C)).*sin(...
PHI_C).*sin(PSI_1).*sin(2.*THETA_1)+(-2).*cos(PSI_1).*cos(THETA_1).*sin(...
THETA_1).*sin(THETA_C)).*(cos(PSI_1).^2.*cos(THETA_1).^2.*cos(THETA_C)...
.^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C).*sin(PSI_1).*sin(2.*THETA_1)...
+2.*cos(PSI_1).*cos(THETA_1).*cos(THETA_C)).*(cos(THETA_1).*sin(PHI_C)).*...
sin(PSI_1)+(-1).*cos(PHI_C).*sin(THETA_1)).*sin(THETA_C)+sin(PHI_C).^2.*...
(sin(THETA_1).^2+cos(THETA_1).^2.*sin(PHI_C).^2.*sin(THETA_C).^2)+cos(...
PHI_C).^2.*(cos(THETA_1).^2.*sin(PHI_C).^2+sin(THETA_1).^2.*sin(THETA_C)...
.^2)).^(-1);
IJ_28_8=(cos(PSI_1).*cos(THETA_C).*sin(PHI_C)+sin(PSI_1).*sin(THETA_C)).*(cos(...
PSI_1).^2.*cos(THETA_1).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C)...
.^2.*sin(PHI_C).*sin(PSI_1).*sin(2.*THETA_1)+2.*cos(PSI_1).*cos(THETA_1)...
.*cos(THETA_C)).*(cos(THETA_1).*sin(PHI_C).*sin(PSI_1)+(-1).*cos(PHI_C)).*...
sin(THETA_1)).*sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_1).^2+cos(THETA_1)...

```

```

.^2.*sin(PHI_C).^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_1).^2.* ...
sin(PHI_C).^2+sin(THETA_1).^2.*sin(THETA_C).^2).^(-1);
IJ_28_9=0;
IJ_28_10=0;
IJ_28_11=0;
IJ_28_12=0;
IJ_28_13=0;
IJ_28_14=0;
IJ_28_15=0;
IJ_28_16=0;
IJ_28_17=0;
IJ_28_18=0;
IJ_28_19=0;
IJ_28_20=0;
IJ_28_21=0;
IJ_28_22=0;
IJ_28_23=0;
IJ_28_24=0;
IJ_28_25=0;
IJ_28_26=0;
IJ_28_27=0;
IJ_28_28=0;
IJ_28_29=0;
IJ_28_30=0;
IJ_28_31=0;
IJ_28_32=0;
IJ_28_33=0;
IJ_28_34=0;
IJ_28_35=0;
IJ_28_36=0;
IJ_28_37=0;
IJ_28_38=0;
IJ_28_39=0;
IJ_28_40=0;
IJ_28_41=0;
IJ_28_42=0;
IJ_28_43=0;
IJ_28_44=0;
IJ_28_45=0;
IJ_28_46=0;
IJ_28_47=0;
IJ_28_48=0;
IJ_28_49=0;
IJ_28_50=0;
IJ_28_51=0;
IJ_28_52=0;
IJ_28_53=0;
IJ_28_54=0;

IJ_29_1=0;
IJ_29_2=0;
IJ_29_3=0;
IJ_29_4=0;
IJ_29_5=(cos(PHI_C).*cos(THETA_1).*cos(THETA_C)+(cos(THETA_1).*sin(PHI_C).*sin( ...
PSI_1)+(-1).*cos(PHI_C).*sin(THETA_1)).*sin(THETA_C)).*(1+(-1).*cos( ...
PHI_C).*cos(THETA_C).*sin(THETA_1)+cos(THETA_1)).*(-1).*cos(THETA_C).* ...
sin(PHI_C).*sin(PHI_C)+cos(PHI_C).*sin(THETA_C)).^2).^(-1/2);
IJ_29_6=(-1).*cos(THETA_C).(cos(PHI_C).*cos(THETA_1).*sin(PHI_C)+sin(PHI_C).* ...
sin(THETA_1)).*(1+(-1).*cos(PHI_C).*cos(THETA_C).*sin(THETA_1)+cos( ...
THETA_1)).*(-1).*cos(THETA_C).*sin(PHI_C).*sin(PHI_C)+cos(PHI_C).*sin( ...
THETA_C)).^2).^(-1/2);
IJ_29_7=(-1).*cos(THETA_1).(cos(PHI_C).*cos(THETA_C).*sin(PHI_C)+sin(PHI_C).* ...
sin(THETA_C)).*(1+(-1).*cos(PHI_C).*cos(THETA_C).*sin(THETA_1)+cos( ...
THETA_1)).*(-1).*cos(THETA_C).*sin(PHI_C).*sin(PHI_C)+cos(PHI_C).*sin( ...
THETA_C)).^2).^(-1/2);
IJ_29_8=(cos(PHI_C).*cos(THETA_1).*cos(THETA_C)+sin(THETA_1).(cos(THETA_C).* ...
sin(PHI_C).*sin(PHI_C)+(-1).*cos(PHI_C).*sin(THETA_C)).*(1+(-1).*cos( ...
PHI_C).*cos(THETA_C).*sin(THETA_1)+cos(THETA_1)).*(-1).*cos(THETA_C).* ...
sin(PHI_C).*sin(PHI_C)+cos(PHI_C).*sin(THETA_C)).^2).^(-1/2);
IJ_29_9=0;
IJ_29_10=0;

```

```

IJ_29_11=0;
IJ_29_12=0;
IJ_29_13=0;
IJ_29_14=0;
IJ_29_15=0;
IJ_29_16=0;
IJ_29_17=0;
IJ_29_18=0;
IJ_29_19=0;
IJ_29_20=0;
IJ_29_21=0;
IJ_29_22=0;
IJ_29_23=0;
IJ_29_24=0;
IJ_29_25=0;
IJ_29_26=0;
IJ_29_27=0;
IJ_29_28=0;
IJ_29_29=0;
IJ_29_30=0;
IJ_29_31=0;
IJ_29_32=0;
IJ_29_33=0;
IJ_29_34=0;
IJ_29_35=0;
IJ_29_36=0;
IJ_29_37=0;
IJ_29_38=0;
IJ_29_39=0;
IJ_29_40=0;
IJ_29_41=0;
IJ_29_42=0;
IJ_29_43=0;
IJ_29_44=0;
IJ_29_45=0;
IJ_29_46=0;
IJ_29_47=0;
IJ_29_48=0;
IJ_29_49=0;
IJ_29_50=0;
IJ_29_51=0;
IJ_29_52=0;
IJ_29_53=0;
IJ_29_54=0;

IJ_30_1=0;
IJ_30_2=0;
IJ_30_3=0;
IJ_30_4=0;
IJ_30_5=(cos(PHI_C).*cos(THETA_1).*sin(PSI_1)+sin(PHI_C).*sin(THETA_1)).*(cos( ...
    PHI_C).^2.*cos(THETA_1).^2.*cos(THETA_C).^2+2.*cos(PHI_C).*cos(THETA_1) ...
    .*cos(THETA_C).*sin(THETA_1).*(cos(THETA_C).*sin(PHI_C).*sin(PSI_1)+(-1) ...
    .*cos(PSI_1).*sin(THETA_C))+sin(PSI_1).^2.*(cos(THETA_C).^2.*sin(PHI_C) ...
    .^2.*sin(THETA_1).^2+sin(THETA_C).^2)+cos(PSI_1).^2.*(cos(THETA_C).^2.* ...
    sin(PHI_C).^2+sin(THETA_1).^2.*sin(THETA_C).^2)+cos(PSI_1).*cos(THETA_1) ...
    .^2.*sin(PHI_C).*sin(PSI_1).*sin(2.*THETA_C)).^(-1);
IJ_30_6=(1/2).*(2.*cos(PSI_1).*cos(THETA_1).*cos(THETA_C).^2+(-2).*cos(PHI_C).* ...
    cos(THETA_C).*sin(THETA_1).*sin(THETA_C)+cos(THETA_1).*sin(PHI_C).*sin( ...
    PSI_1).*sin(2.*THETA_C)).*(cos(PHI_C).^2.*cos(THETA_1).^2.*cos(THETA_C) ...
    .^2+2.*cos(PHI_C).*cos(THETA_1).*cos(THETA_C).*sin(THETA_1).*(cos( ...
    THETA_C).*sin(PHI_C).*sin(PSI_1)+(-1).*cos(PSI_1).*sin(THETA_C))+sin( ...
    PSI_1).^2.*(cos(THETA_C).^2.*sin(PHI_C).^2.*sin(THETA_1).^2+sin(THETA_C) ...
    .^2)+cos(PSI_1).^2.*(cos(THETA_C).^2.*sin(PHI_C).^2+sin(THETA_1).^2.* ...
    sin(THETA_C).^2)+cos(PSI_1).*cos(THETA_1).^2.*sin(PHI_C).*sin(PSI_1).* ...
    sin(2.*THETA_C)).^(-1);
IJ_30_7=(1/4).*(cos(PHI_C).^2.*cos(THETA_1).^2.*cos(THETA_C).^2+2.*cos(PHI_C).* ...
    cos(THETA_1).*cos(THETA_C).*sin(THETA_1).*(cos(THETA_C).*sin(PHI_C).* ...
    sin(PSI_1)+(-1).*cos(PSI_1).*sin(THETA_C))+sin(PSI_1).^2.*(cos(THETA_C) ...
    .^2.*sin(PHI_C).^2.*sin(THETA_1).^2+sin(THETA_C).^2)+cos(PSI_1).^2.*( ...
    cos(THETA_C).^2.*sin(PHI_C).^2+sin(THETA_1).^2.*sin(THETA_C).^2)+cos( ...
    PSI_1).*cos(THETA_1).^2.*sin(PHI_C).*sin(PSI_1).*sin(2.*THETA_C)).^(-1) ...

```

```

.*((-3)+cos(2.*PHI_C)+2.*cos(PHI_C).^2.*cos(2.*THETA_C)).*sin(THETA_1)+ ...
cos(THETA_1).*(-2).*cos(THETA_C).^2.*sin(2.*PHI_C).*sin(PSI_1)+2.*cos( ...
PHI_C).*cos(PSI_1).*sin(2.*THETA_C));
IJ_30_8=(cos(PSI_1).*cos(THETA_C).*sin(PHI_C)+sin(PSI_1).*sin(THETA_C)).*(cos( ...
PHI_C).*cos(THETA_C).*sin(THETA_1)+cos(THETA_1).*(-1).*cos(THETA_C).* ...
sin(PHI_C).*sin(PSI_1)+cos(PSI_1).*sin(THETA_C)).*(cos(PHI_C).^2.*cos( ...
THETA_1).^2.*cos(THETA_C).^2+2.*cos(PHI_C).*cos(THETA_1).*cos(THETA_C).* ...
sin(THETA_1).*(cos(THETA_C).*sin(PHI_C).*sin(PSI_1)+(-1).*cos(PSI_1).* ...
sin(THETA_C))+sin(PSI_1).^2.*(cos(THETA_C).^2.*sin(PHI_C).^2.*sin( ...
THETA_1).^2+sin(THETA_C).^2)+cos(PSI_1).^2.*(cos(THETA_C).^2.*sin(PHI_C) ...
.^2+sin(THETA_1).^2.*sin(THETA_C).^2)+cos(PSI_1).*cos(THETA_1).^2.*sin( ...
PHI_C).*sin(PSI_1).*sin(2.*THETA_C)).^(-1);
IJ_30_9=1;
IJ_30_10=0;
IJ_30_11=0;
IJ_30_12=0;
IJ_30_13=0;
IJ_30_14=0;
IJ_30_15=0;
IJ_30_16=0;
IJ_30_17=0;
IJ_30_18=0;
IJ_30_19=0;
IJ_30_20=0;
IJ_30_21=0;
IJ_30_22=0;
IJ_30_23=0;
IJ_30_24=0;
IJ_30_25=0;
IJ_30_26=0;
IJ_30_27=0;
IJ_30_28=0;
IJ_30_29=0;
IJ_30_30=0;
IJ_30_31=0;
IJ_30_32=0;
IJ_30_33=0;
IJ_30_34=0;
IJ_30_35=0;
IJ_30_36=0;
IJ_30_37=0;
IJ_30_38=0;
IJ_30_39=0;
IJ_30_40=0;
IJ_30_41=0;
IJ_30_42=0;
IJ_30_43=0;
IJ_30_44=0;
IJ_30_45=0;
IJ_30_46=0;
IJ_30_47=0;
IJ_30_48=0;
IJ_30_49=0;
IJ_30_50=0;
IJ_30_51=0;
IJ_30_52=0;
IJ_30_53=0;
IJ_30_54=0;

IJ_31_1=0;
IJ_31_2=0;
IJ_31_3=0;
IJ_31_4=1;
IJ_31_5=(cos(PHI_C).*cos(THETA_2).*sin(PSI_2)+sin(PHI_C).*sin(THETA_2)).*(cos( ...
PHI_C).*cos(THETA_C).*sin(THETA_2)+cos(THETA_2).*(-1).*cos(THETA_C).* ...
sin(PHI_C).*sin(PSI_2)+cos(PSI_2).*sin(THETA_C)).*(cos(PSI_2).^2.*cos( ...
THETA_2).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C).* ...
sin(PSI_2).*sin(2.*THETA_2)+2.*cos(PSI_2).*cos(THETA_2).*cos(THETA_C).* ...
cos(THETA_2).*sin(PHI_C).*sin(PSI_2)+(-1).*cos(PHI_C).*sin(THETA_2)).*( ...
sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_2).^2+cos(THETA_2).^2.*sin(PSI_2) ...
.^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_2).^2.*sin(PSI_2).^2+ ...

```

```

sin(THETA_2).^2.*sin(THETA_C).^2).^(-1);
IJ_31_6=(1/4).*(4.*cos(PSI_2).*cos(THETA_2).*cos(THETA_C).*((-1).*cos(THETA_2)).* ...
sin(PHI_C).*sin(PSI_2)+cos(PHI_C).*sin(THETA_2))+((-3)+cos(2.*PSI_2)+2.* ...
cos(PSI_2).^2.*cos(2.*THETA_2)).*sin(THETA_C)).*(cos(PSI_2).^2.*cos( ...
THETA_2).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C)).* ...
sin(PSI_2).*sin(2.*THETA_2)+2.*cos(PSI_2).*cos(THETA_2).*cos(THETA_C)).*( ...
cos(THETA_2).*sin(PHI_C).*sin(PSI_2)+(-1).*cos(PHI_C).*sin(THETA_2)).* ...
sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_2).^2+cos(THETA_2).^2.*sin(PSI_2) ...
.^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_2).^2.*sin(PSI_2).^2+ ...
sin(THETA_2).^2.*sin(THETA_C).^2)).^(-1);
IJ_31_7=0;
IJ_31_8=0;
IJ_31_9=0;
IJ_31_10=(1/2).*(2.*cos(PHI_C).*cos(THETA_2).^2.*cos(THETA_C)+cos(THETA_C).*sin( ...
PHI_C).*sin(PSI_2).*sin(2.*THETA_2)+(-2).*cos(PSI_2).*cos(THETA_2).*sin( ...
THETA_2).*sin(THETA_C)).*(cos(PSI_2).^2.*cos(THETA_2).^2.*cos(THETA_C) ...
.^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C).*sin(PSI_2).*sin(2.*THETA_2) ...
+2.*cos(PSI_2).*cos(THETA_2).*cos(THETA_C)).*(cos(THETA_2).*sin(PHI_C)).* ...
sin(PSI_2)+(-1).*cos(PHI_C).*sin(THETA_2)).*sin(THETA_C)+sin(PHI_C).^2.* ...
(sin(THETA_2).^2+cos(THETA_2).^2.*sin(PSI_2).^2.*sin(THETA_C).^2)+cos( ...
PHI_C).^2.*(cos(THETA_2).^2.*sin(PSI_2).^2+sin(THETA_2).^2.*sin(THETA_C) ...
.^2)).^(-1);
IJ_31_11=(cos(PSI_2).*cos(THETA_C).*sin(PHI_C)+sin(PSI_2).*sin(THETA_C)).*(cos( ...
PSI_2).^2.*cos(THETA_2).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C) ...
.^2.*sin(PHI_C).*sin(PSI_2).*sin(2.*THETA_2)+2.*cos(PSI_2).*cos(THETA_2) ...
.*cos(THETA_C)).*(cos(THETA_2).*sin(PHI_C).*sin(PSI_2)+(-1).*cos(PHI_C)).* ...
sin(THETA_2)).*sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_2).^2+cos(THETA_2) ...
.^2.*sin(PSI_2).^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_2).^2.* ...
sin(PSI_2).^2+sin(THETA_2).^2.*sin(THETA_C).^2)).^(-1);
IJ_31_12=0;
IJ_31_13=0;
IJ_31_14=0;
IJ_31_15=0;
IJ_31_16=0;
IJ_31_17=0;
IJ_31_18=0;
IJ_31_19=0;
IJ_31_20=0;
IJ_31_21=0;
IJ_31_22=0;
IJ_31_23=0;
IJ_31_24=0;
IJ_31_25=0;
IJ_31_26=0;
IJ_31_27=0;
IJ_31_28=0;
IJ_31_29=0;
IJ_31_30=0;
IJ_31_31=0;
IJ_31_32=0;
IJ_31_33=0;
IJ_31_34=0;
IJ_31_35=0;
IJ_31_36=0;
IJ_31_37=0;
IJ_31_38=0;
IJ_31_39=0;
IJ_31_40=0;
IJ_31_41=0;
IJ_31_42=0;
IJ_31_43=0;
IJ_31_44=0;
IJ_31_45=0;
IJ_31_46=0;
IJ_31_47=0;
IJ_31_48=0;
IJ_31_49=0;
IJ_31_50=0;
IJ_31_51=0;
IJ_31_52=0;
IJ_31_53=0;

```



```

IJ_31_54=0;

IJ_32_1=0;
IJ_32_2=0;
IJ_32_3=0;
IJ_32_4=0;
IJ_32_5=(cos(PHI_2).*cos(THETA_2).*cos(THETA_C)+(cos(THETA_2).*sin(PHI_C).*sin( ...
    PSI_2)+(-1).*cos(PHI_C).*sin(THETA_2)).*sin(THETA_C)).*(1+(-1).*cos( ...
    PHI_C).*cos(THETA_C).*sin(THETA_2)+cos(THETA_2).*(-1).*cos(THETA_C).* ...
    sin(PHI_C).*sin(PSI_2)+cos(PSI_2).*sin(THETA_C)).^2).^(-1/2);
IJ_32_6=(-1).*cos(THETA_C).*(cos(PHI_C).*cos(THETA_2).*sin(PSI_2)+sin(PHI_C).* ...
    sin(THETA_2)).*(1+(-1).*cos(PHI_C).*cos(THETA_C).*sin(THETA_2)+cos( ...
    THETA_2).*(-1).*cos(THETA_C).*sin(PHI_C).*sin(PSI_2)+cos(PSI_2).*sin( ...
    THETA_C)).^2).^(-1/2);
IJ_32_7=0;
IJ_32_8=0;
IJ_32_9=0;
IJ_32_10=(-1).*cos(THETA_2).*(cos(PSI_2).*cos(THETA_C).*sin(PHI_C)+sin(PSI_2).* ...
    sin(THETA_C)).*(1+(-1).*cos(PHI_C).*cos(THETA_C).*sin(THETA_2)+cos( ...
    THETA_2).*(-1).*cos(THETA_C).*sin(PHI_C).*sin(PSI_2)+cos(PSI_2).*sin( ...
    THETA_C)).^2).^(-1/2);
IJ_32_11=(cos(PHI_C).*cos(THETA_2).*cos(THETA_C)+sin(THETA_2).*(cos(THETA_C).* ...
    sin(PHI_C).*sin(PSI_2)+(-1).*cos(PSI_2).*sin(THETA_C)).*(1+(-1).*cos( ...
    PHI_C).*cos(THETA_C).*sin(THETA_2)+cos(THETA_2).*(-1).*cos(THETA_C).* ...
    sin(PHI_C).*sin(PSI_2)+cos(PSI_2).*sin(THETA_C)).^2).^(-1/2);
IJ_32_12=0;
IJ_32_13=0;
IJ_32_14=0;
IJ_32_15=0;
IJ_32_16=0;
IJ_32_17=0;
IJ_32_18=0;
IJ_32_19=0;
IJ_32_20=0;
IJ_32_21=0;
IJ_32_22=0;
IJ_32_23=0;
IJ_32_24=0;
IJ_32_25=0;
IJ_32_26=0;
IJ_32_27=0;
IJ_32_28=0;
IJ_32_29=0;
IJ_32_30=0;
IJ_32_31=0;
IJ_32_32=0;
IJ_32_33=0;
IJ_32_34=0;
IJ_32_35=0;
IJ_32_36=0;
IJ_32_37=0;
IJ_32_38=0;
IJ_32_39=0;
IJ_32_40=0;
IJ_32_41=0;
IJ_32_42=0;
IJ_32_43=0;
IJ_32_44=0;
IJ_32_45=0;
IJ_32_46=0;
IJ_32_47=0;
IJ_32_48=0;
IJ_32_49=0;
IJ_32_50=0;
IJ_32_51=0;
IJ_32_52=0;
IJ_32_53=0;
IJ_32_54=0;

IJ_33_1=0;
IJ_33_2=0;

```

```

IJ_33_3=0;
IJ_33_4=0;
IJ_33_5=(cos(PHI_C)*cos(THETA_2)*sin(PSI_2)+sin(PHI_C)*sin(THETA_2))*(cos( ...
    PHI_C).^2*cos(THETA_2).^2*cos(THETA_C).^2+2*cos(PHI_C)*cos(THETA_2) ...
    *cos(THETA_C)*sin(THETA_2)*(cos(THETA_C)*sin(PHI_C)*sin(PSI_2)+(-1) ...
    *cos(PSI_2)*sin(THETA_C))+sin(PSI_2).^2*(cos(THETA_C).^2*sin(PHI_C) ...
    ^2.*sin(THETA_2).^2+sin(THETA_C).^2)+cos(PSI_2).^2*(cos(THETA_C).^2.*
    sin(PHI_C).^2+sin(THETA_2).^2.*sin(THETA_C).^2)+cos(PSI_2)*cos(THETA_2) ...
    ^2.*sin(PHI_C)*sin(PSI_2)*sin(2.*THETA_C)).^(-1);
IJ_33_6=(1/2)*(2*cos(PSI_2)*cos(THETA_2)*cos(THETA_C).^2+(-2)*cos(PHI_C)* ...
    cos(THETA_C)*sin(THETA_2)*sin(THETA_C)+cos(THETA_2)*sin(PHI_C)*sin( ...
    PSI_2)*sin(2.*THETA_C))*(cos(PHI_C).^2*cos(THETA_2).^2*cos(THETA_C) ...
    ^2+2*cos(PHI_C)*cos(THETA_2)*cos(THETA_C)*sin(THETA_2)*(cos( ...
    THETA_C)*sin(PHI_C)*sin(PSI_2)+(-1)*cos(PSI_2)*sin(THETA_C))+sin( ...
    PSI_2).^2*(cos(THETA_C).^2.*sin(PHI_C).^2.*sin(THETA_2).^2+sin(THETA_C) ...
    ^2)+cos(PSI_2).^2*(cos(THETA_C).^2.*sin(PHI_C).^2+sin(THETA_2).^2.* ...
    sin(THETA_C).^2)+cos(PSI_2)*cos(THETA_2).^2.*sin(PHI_C)*sin(PSI_2)* ...
    sin(2.*THETA_C)).^(-1);
IJ_33_7=0;
IJ_33_8=0;
IJ_33_9=0;
IJ_33_10=(1/4)*(cos(PHI_C).^2*cos(THETA_2).^2*cos(THETA_C).^2+2*cos(PHI_C)* ...
    cos(THETA_2)*cos(THETA_C)*sin(THETA_2)*(cos(THETA_C)*sin(PHI_C)* ...
    sin(PSI_2)+(-1)*cos(PSI_2)*sin(THETA_C))+sin(PSI_2).^2*(cos(THETA_C) ...
    ^2.*sin(PHI_C).^2.*sin(THETA_2).^2+sin(THETA_C).^2)+cos(PSI_2).^2*( ...
    cos(THETA_C).^2.*sin(PHI_C).^2+sin(THETA_2).^2.*sin(THETA_C).^2)+cos( ...
    PSI_2)*cos(THETA_2).^2.*sin(PHI_C)*sin(PSI_2)*sin(2.*THETA_C)).^(-1) ...
    *((-3)+cos(2.*PHI_C)+2*cos(PHI_C).^2*cos(2.*THETA_C))*sin(THETA_2)+ ...
    cos(THETA_2)*((-2)*cos(THETA_C).^2.*sin(2.*PHI_C)*sin(PSI_2)+2*cos( ...
    PHI_C)*cos(PSI_2)*sin(2.*THETA_C));
IJ_33_11=(cos(PSI_2)*cos(THETA_C)*sin(PHI_C)+sin(PSI_2)*sin(THETA_C))*(cos( ...
    PHI_C)*cos(THETA_C)*sin(THETA_2)+cos(THETA_2)*((-1)*cos(THETA_C)* ...
    sin(PHI_C)*sin(PSI_2)+cos(PSI_2)*sin(THETA_C))*(cos(PHI_C).^2*cos( ...
    THETA_2).^2*cos(THETA_C).^2+2*cos(PHI_C)*cos(THETA_2)*cos(THETA_C)* ...
    sin(THETA_2)*(cos(THETA_C)*sin(PHI_C)*sin(PSI_2)+(-1)*cos(PSI_2)* ...
    sin(THETA_C))+sin(PSI_2).^2*(cos(THETA_C).^2.*sin(PHI_C).^2.*sin( ...
    THETA_2).^2+sin(THETA_C).^2)+cos(PSI_2).^2*(cos(THETA_C).^2.*sin(PHI_C) ...
    ^2+sin(THETA_2).^2.*sin(THETA_C).^2)+cos(PSI_2)*cos(THETA_2).^2.*sin( ...
    PHI_C)*sin(PSI_2)*sin(2.*THETA_C)).^(-1);
IJ_33_12=1;
IJ_33_13=0;
IJ_33_14=0;
IJ_33_15=0;
IJ_33_16=0;
IJ_33_17=0;
IJ_33_18=0;
IJ_33_19=0;
IJ_33_20=0;
IJ_33_21=0;
IJ_33_22=0;
IJ_33_23=0;
IJ_33_24=0;
IJ_33_25=0;
IJ_33_26=0;
IJ_33_27=0;
IJ_33_28=0;
IJ_33_29=0;
IJ_33_30=0;
IJ_33_31=0;
IJ_33_32=0;
IJ_33_33=0;
IJ_33_34=0;
IJ_33_35=0;
IJ_33_36=0;
IJ_33_37=0;
IJ_33_38=0;
IJ_33_39=0;
IJ_33_40=0;
IJ_33_41=0;
IJ_33_42=0;
IJ_33_43=0;

```

```

IJ_33_44=0;
IJ_33_45=0;
IJ_33_46=0;
IJ_33_47=0;
IJ_33_48=0;
IJ_33_49=0;
IJ_33_50=0;
IJ_33_51=0;
IJ_33_52=0;
IJ_33_53=0;
IJ_33_54=0;

IJ_34_1=0;
IJ_34_2=0;
IJ_34_3=0;
IJ_34_4=1;
IJ_34_5=(cos(PHI_C).*cos(THETA_3).*sin(PSI_3)+sin(PHI_C).*sin(THETA_3)).*(cos( ...
    PHI_C).*cos(THETA_C).*sin(THETA_3)+cos(THETA_3).*((-1).*cos(THETA_C)).* ...
    sin(PHI_C).*sin(PSI_3)+cos(PSI_3).*sin(THETA_C)).*(cos(PSI_3).^2.*cos( ...
    THETA_3).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C).* ...
    sin(PSI_3).*sin(2.*THETA_3)+2.*cos(PSI_3).*cos(THETA_3).*cos(THETA_C)).*( ...
    cos(THETA_3).*sin(PHI_C).*sin(PSI_3)+(-1).*cos(PHI_C).*sin(THETA_3)).* ...
    sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_3).^2+cos(THETA_3).^2.*sin(PSI_3) ...
    .^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_3).^2.*sin(PSI_3).^2+ ...
    sin(THETA_3).^2.*sin(THETA_C).^2)).^(-1);
IJ_34_6=(1/4).* (4.*cos(PSI_3).*cos(THETA_3).*cos(THETA_C).*((-1).*cos(THETA_3)).* ...
    sin(PHI_C).*sin(PSI_3)+cos(PHI_C).*sin(THETA_3))+((-3)+cos(2.*PSI_3)+2.* ...
    cos(PSI_3).^2.*cos(2.*THETA_3)).*sin(THETA_C)).*(cos(PSI_3).^2.*cos( ...
    THETA_3).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C)).* ...
    sin(PSI_3).*sin(2.*THETA_3)+2.*cos(PSI_3).*cos(THETA_3).*cos(THETA_C)).*( ...
    cos(THETA_3).*sin(PHI_C).*sin(PSI_3)+(-1).*cos(PHI_C).*sin(THETA_3)).* ...
    sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_3).^2+cos(THETA_3).^2.*sin(PSI_3) ...
    .^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_3).^2.*sin(PSI_3).^2+ ...
    sin(THETA_3).^2.*sin(THETA_C).^2)).^(-1);
IJ_34_7=0;
IJ_34_8=0;
IJ_34_9=0;
IJ_34_10=0;
IJ_34_11=0;
IJ_34_12=0;
IJ_34_13=(1/2).* (2.*cos(PHI_C).*cos(THETA_3).^2.*cos(THETA_C)+cos(THETA_C).*sin( ...
    PHI_C).*sin(PSI_3).*sin(2.*THETA_3)+(-2).*cos(PSI_3).*cos(THETA_3).*sin( ...
    THETA_3).*sin(THETA_C)).*(cos(PSI_3).^2.*cos(THETA_3).^2.*cos(THETA_C) ...
    .^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C).*sin(PSI_3).*sin(2.*THETA_3) ...
    +2.*cos(PSI_3).*cos(THETA_3).*cos(THETA_C)).*(cos(THETA_3).*sin(PHI_C)).* ...
    sin(PSI_3)+(-1).*cos(PHI_C).*sin(THETA_3)).*sin(THETA_C)+sin(PHI_C).^2.* ...
    (sin(THETA_3).^2+cos(THETA_3).^2.*sin(PSI_3).^2.*sin(THETA_C).^2)+cos( ...
    PHI_C).^2.*(cos(THETA_3).^2.*sin(PSI_3).^2+sin(THETA_3).^2.*sin(THETA_C) ...
    .^2)).^(-1);
IJ_34_14=(cos(PSI_3).*cos(THETA_C).*sin(PHI_C)+sin(PSI_3).*sin(THETA_C)).*(cos( ...
    PSI_3).^2.*cos(THETA_3).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C) ...
    .^2.*sin(PHI_C).*sin(PSI_3).*sin(2.*THETA_3)+2.*cos(PSI_3).*cos(THETA_3) ...
    .*cos(THETA_C)).*(cos(THETA_3).*sin(PHI_C).*sin(PSI_3)+(-1).*cos(PHI_C)).* ...
    sin(THETA_3)).*sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_3).^2+cos(THETA_3) ...
    .^2.*sin(PSI_3).^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_3).^2.* ...
    sin(PSI_3).^2+sin(THETA_3).^2.*sin(THETA_C).^2)).^(-1);
IJ_34_15=0;
IJ_34_16=0;
IJ_34_17=0;
IJ_34_18=0;
IJ_34_19=0;
IJ_34_20=0;
IJ_34_21=0;
IJ_34_22=0;
IJ_34_23=0;
IJ_34_24=0;
IJ_34_25=0;
IJ_34_26=0;
IJ_34_27=0;
IJ_34_28=0;
IJ_34_29=0;

```

```

IJ_34_30=0;
IJ_34_31=0;
IJ_34_32=0;
IJ_34_33=0;
IJ_34_34=0;
IJ_34_35=0;
IJ_34_36=0;
IJ_34_37=0;
IJ_34_38=0;
IJ_34_39=0;
IJ_34_40=0;
IJ_34_41=0;
IJ_34_42=0;
IJ_34_43=0;
IJ_34_44=0;
IJ_34_45=0;
IJ_34_46=0;
IJ_34_47=0;
IJ_34_48=0;
IJ_34_49=0;
IJ_34_50=0;
IJ_34_51=0;
IJ_34_52=0;
IJ_34_53=0;
IJ_34_54=0;

IJ_35_1=0;
IJ_35_2=0;
IJ_35_3=0;
IJ_35_4=0;
IJ_35_5=(cos(PHI_3)*cos(THETA_3)*cos(THETA_C)+(cos(THETA_3)*sin(PHI_C)*sin(
    PSI_3)+(-1)*cos(PHI_C)*sin(THETA_3))*sin(THETA_C))* (1+(-1)*cos(
    PHI_C)*cos(THETA_C)*sin(THETA_3)+cos(THETA_3)*((-1)*cos(THETA_C)*
    sin(PHI_C)*sin(PSI_3)+cos(PSI_3)*sin(THETA_C))^2)^(-1/2);
IJ_35_6=(-1)*cos(THETA_C)*(cos(PHI_C)*cos(THETA_3)*sin(PSI_3)+sin(PHI_C)*
    sin(THETA_3))* (1+(-1)*cos(PHI_C)*cos(THETA_C)*sin(THETA_3)+cos(
    THETA_3)*((-1)*cos(THETA_C)*sin(PHI_C)*sin(PSI_3)+cos(PSI_3)*sin(
    THETA_C))^2)^(-1/2);
IJ_35_7=0;
IJ_35_8=0;
IJ_35_9=0;
IJ_35_10=0;
IJ_35_11=0;
IJ_35_12=0;
IJ_35_13=(-1)*cos(THETA_3)*(cos(PSI_3)*cos(THETA_C)*sin(PHI_C)+sin(PSI_3)*
    sin(THETA_C))* (1+(-1)*cos(PHI_C)*cos(THETA_C)*sin(THETA_3)+cos(
    THETA_3)*((-1)*cos(THETA_C)*sin(PHI_C)*sin(PSI_3)+cos(PSI_3)*sin(
    THETA_C))^2)^(-1/2);
IJ_35_14=(cos(PHI_C)*cos(THETA_3)*cos(THETA_C)+sin(THETA_3)*(cos(THETA_C)*
    sin(PHI_C)*sin(PSI_3)+(-1)*cos(PSI_3)*sin(THETA_C)))* (1+(-1)*cos(
    PHI_C)*cos(THETA_C)*sin(THETA_3)+cos(THETA_3)*((-1)*cos(THETA_C)*
    sin(PHI_C)*sin(PSI_3)+cos(PSI_3)*sin(THETA_C))^2)^(-1/2);
IJ_35_15=0;
IJ_35_16=0;
IJ_35_17=0;
IJ_35_18=0;
IJ_35_19=0;
IJ_35_20=0;
IJ_35_21=0;
IJ_35_22=0;
IJ_35_23=0;
IJ_35_24=0;
IJ_35_25=0;
IJ_35_26=0;
IJ_35_27=0;
IJ_35_28=0;
IJ_35_29=0;
IJ_35_30=0;
IJ_35_31=0;
IJ_35_32=0;
IJ_35_33=0;

```

```

IJ_35_34=0;
IJ_35_35=0;
IJ_35_36=0;
IJ_35_37=0;
IJ_35_38=0;
IJ_35_39=0;
IJ_35_40=0;
IJ_35_41=0;
IJ_35_42=0;
IJ_35_43=0;
IJ_35_44=0;
IJ_35_45=0;
IJ_35_46=0;
IJ_35_47=0;
IJ_35_48=0;
IJ_35_49=0;
IJ_35_50=0;
IJ_35_51=0;
IJ_35_52=0;
IJ_35_53=0;
IJ_35_54=0;

IJ_36_1=0;
IJ_36_2=0;
IJ_36_3=0;
IJ_36_4=0;
IJ_36_5=(cos(PHI_C)*cos(THETA_3)*sin(PSI_3)+sin(PHI_C)*sin(THETA_3))*(cos( ...
    PHI_C).^2.*cos(THETA_3).^2.*cos(THETA_C).^2+2.*cos(PHI_C)*cos(THETA_3) ...
    .*cos(THETA_C)*sin(THETA_3)*(cos(THETA_C)*sin(PHI_C)*sin(PSI_3)+(-1) ...
    .*cos(PSI_3)*sin(THETA_C))+sin(PSI_3).^2.*(cos(THETA_C).^2.*sin(PHI_C) ...
    .*sin(THETA_3).^2+sin(THETA_C).^2)+cos(PSI_3).^2.*(cos(THETA_C).^2.* ...
    sin(PHI_C).^2+sin(THETA_3).^2.*sin(THETA_C).^2)+cos(PSI_3)*cos(THETA_3) ...
    .*sin(PHI_C)*sin(PSI_3)*sin(2.*THETA_C)).^(-1);
IJ_36_6=(1/2).* (2.*cos(PSI_3)*cos(THETA_3)*cos(THETA_C).^2+(-2).*cos(PHI_C)* ...
    cos(THETA_C)*sin(THETA_3)*sin(THETA_C)+cos(THETA_3)*sin(PHI_C)*sin( ...
    PHI_C).*sin(2.*THETA_C)).*(cos(PHI_C).^2.*cos(THETA_3).^2.*cos(THETA_C) ...
    .*sin(THETA_3)*cos(THETA_C)*sin(THETA_3)*(cos( ...
    THETA_C)*sin(PHI_C)*sin(PSI_3)+(-1).*cos(PSI_3)*sin(THETA_C))+sin( ...
    PHI_C).^2.*(cos(THETA_C).^2.*sin(PHI_C).^2.*sin(THETA_3).^2+sin(THETA_C) ...
    .*sin(THETA_3).^2+sin(THETA_C).^2.*sin(PHI_C).^2+sin(THETA_3).^2.* ...
    sin(THETA_C).^2)+cos(PSI_3)*cos(THETA_3).^2.*sin(PHI_C)*sin(PSI_3)* ...
    sin(2.*THETA_C)).^(-1);
IJ_36_7=0;
IJ_36_8=0;
IJ_36_9=0;
IJ_36_10=0;
IJ_36_11=0;
IJ_36_12=0;
IJ_36_13=(1/4).* (cos(PHI_C).^2.*cos(THETA_3).^2.*cos(THETA_C).^2+2.*cos(PHI_C)* ...
    cos(THETA_3)*cos(THETA_C)*sin(THETA_3)*(cos(THETA_C)*sin(PHI_C)* ...
    sin(PSI_3)+(-1).*cos(PSI_3)*sin(THETA_C))+sin(PSI_3).^2.*(cos(THETA_C) ...
    .*sin(PHI_C).^2.*sin(THETA_3).^2+sin(THETA_C).^2)+cos(PSI_3).^2.*( ...
    cos(THETA_C).^2.*sin(PHI_C).^2+sin(THETA_3).^2.*sin(THETA_C).^2)+cos( ...
    PSI_3)*cos(THETA_3).^2.*sin(PHI_C)*sin(PSI_3)*sin(2.*THETA_C)).^(-1) ...
    .* ((-3)+cos(2.*PHI_C)+2.*cos(PHI_C).^2.*cos(2.*THETA_C)).*sin(THETA_3)+ ...
    cos(THETA_3)*((-2).*cos(THETA_C).^2.*sin(2.*PHI_C)*sin(PSI_3)+2.*cos( ...
    PHI_C)*cos(PSI_3)*sin(2.*THETA_C));
IJ_36_14=(cos(PSI_3)*cos(THETA_C)*sin(PHI_C)+sin(PSI_3)*sin(THETA_C))*(cos( ...
    PHI_C)*cos(THETA_C)*sin(THETA_3)+cos(THETA_3)*((-1).*cos(THETA_C)* ...
    sin(PHI_C)*sin(PSI_3)+cos(THETA_3)*sin(THETA_C)).*(cos(PHI_C).^2.*cos( ...
    THETA_3).^2.*cos(THETA_C).^2+2.*cos(PHI_C)*cos(THETA_3)*cos(THETA_C)* ...
    sin(THETA_3)*(cos(THETA_C)*sin(PHI_C)*sin(PSI_3)+(-1).*cos(PSI_3)* ...
    sin(THETA_C))+sin(PSI_3).^2.*(cos(THETA_C).^2.*sin(PHI_C).^2.*sin( ...
    THETA_3).^2+sin(THETA_C).^2)+cos(PSI_3).^2.*(cos(THETA_C).^2.*sin(PHI_C) ...
    .*sin(THETA_3).^2+sin(THETA_C).^2.*sin(PHI_C).^2+sin(THETA_3).^2.*sin( ...
    PHI_C)*sin(PSI_3)*sin(2.*THETA_C)).^(-1);
IJ_36_15=1;
IJ_36_16=0;
IJ_36_17=0;
IJ_36_18=0;
IJ_36_19=0;

```

```

IJ_36_20=0;
IJ_36_21=0;
IJ_36_22=0;
IJ_36_23=0;
IJ_36_24=0;
IJ_36_25=0;
IJ_36_26=0;
IJ_36_27=0;
IJ_36_28=0;
IJ_36_29=0;
IJ_36_30=0;
IJ_36_31=0;
IJ_36_32=0;
IJ_36_33=0;
IJ_36_34=0;
IJ_36_35=0;
IJ_36_36=0;
IJ_36_37=0;
IJ_36_38=0;
IJ_36_39=0;
IJ_36_40=0;
IJ_36_41=0;
IJ_36_42=0;
IJ_36_43=0;
IJ_36_44=0;
IJ_36_45=0;
IJ_36_46=0;
IJ_36_47=0;
IJ_36_48=0;
IJ_36_49=0;
IJ_36_50=0;
IJ_36_51=0;
IJ_36_52=0;
IJ_36_53=0;
IJ_36_54=0;

IJ_37_1=0;
IJ_37_2=0;
IJ_37_3=0;
IJ_37_4=1;
IJ_37_5=(cos(PHI_C).*cos(THETA_4).*sin(PSI_4)+sin(PHI_C).*sin(THETA_4)).*(cos( ...
    PHI_C).*cos(THETA_C).*sin(THETA_4)+cos(THETA_4).*((-1).*cos(THETA_C)).* ...
    sin(PHI_C).*sin(PSI_4)+cos(PSI_4).*sin(THETA_C))).*(cos(PSI_4).^2.*cos( ...
    THETA_4).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C)).* ...
    sin(PSI_4).*sin(2.*THETA_4)+2.*cos(PSI_4).*cos(THETA_4).*cos(THETA_C)).*( ...
    cos(THETA_4).*sin(PHI_C).*sin(PSI_4)+(-1).*cos(PHI_C).*sin(THETA_4)).* ...
    sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_4).^2+cos(THETA_4).^2.*sin(PSI_4) ...
    .^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_4).^2.*sin(PSI_4).^2+ ...
    sin(THETA_4).^2.*sin(THETA_C).^2)).^(-1);
IJ_37_6=(1/4).* (4.*cos(PSI_4).*cos(THETA_4).*cos(THETA_C).*((-1).*cos(THETA_4)).* ...
    sin(PHI_C).*sin(PSI_4)+cos(PHI_C).*sin(THETA_4))+((-3)+cos(2.*PSI_4)+2.* ...
    cos(PSI_4).^2.*cos(2.*THETA_4)).*sin(THETA_C)).*(cos(PSI_4).^2.*cos( ...
    THETA_4).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C)).* ...
    sin(PSI_4).*sin(2.*THETA_4)+2.*cos(PSI_4).*cos(THETA_4).*cos(THETA_C)).*( ...
    cos(THETA_4).*sin(PHI_C).*sin(PSI_4)+(-1).*cos(PHI_C).*sin(THETA_4)).* ...
    sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_4).^2+cos(THETA_4).^2.*sin(PSI_4) ...
    .^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_4).^2.*sin(PSI_4).^2+ ...
    sin(THETA_4).^2.*sin(THETA_C).^2)).^(-1);
IJ_37_7=0;
IJ_37_8=0;
IJ_37_9=0;
IJ_37_10=0;
IJ_37_11=0;
IJ_37_12=0;
IJ_37_13=0;
IJ_37_14=0;
IJ_37_15=0;
IJ_37_16=(1/2).* (2.*cos(PHI_C).*cos(THETA_4).^2.*cos(THETA_C)+cos(THETA_C).*sin( ...
    PHI_C).*sin(PSI_4).*sin(2.*THETA_4)+(-2).*cos(PSI_4).*cos(THETA_4).*sin( ...
    THETA_4).*sin(THETA_C)).*(cos(PSI_4).^2.*cos(THETA_4).^2.*cos(THETA_C) ...
    .^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C).*sin(PSI_4).*sin(2.*THETA_4) ...

```

```

+2.*cos(PSI_4).*cos(THETA_4).*cos(THETA_C).*cos(THETA_4).*sin(PHI_C).* ...
sin(PSI_4)+(-1).*cos(PHI_C).*sin(THETA_4)).*sin(THETA_C)+sin(PHI_C).^2.* ...
(sin(THETA_4).^2+cos(THETA_4).^2.*sin(PSI_4).^2.*sin(THETA_C).^2)+cos( ...
PHI_C).^2.*(cos(THETA_4).^2.*sin(PSI_4).^2+sin(THETA_4).^2.*sin(THETA_C) ...
.^2)).^(-1);
IJ_37_17=(cos(PSI_4).*cos(THETA_C).*sin(PHI_C)+sin(PSI_4).*sin(THETA_C)).*(cos( ...
PSI_4).^2.*cos(THETA_4).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C) ...
.^2.*sin(PHI_C).*sin(PSI_4).*sin(2.*THETA_4)+2.*cos(PSI_4).*cos(THETA_4) ...
.*cos(THETA_C).*cos(THETA_4).*sin(PHI_C).*sin(PSI_4)+(-1).*cos(PHI_C).* ...
sin(THETA_4)).*sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_4).^2+cos(THETA_4) ...
.^2.*sin(PSI_4).^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_4).^2.* ...
sin(PSI_4).^2+sin(THETA_4).^2.*sin(THETA_C).^2)).^(-1);
IJ_37_18=0;
IJ_37_19=0;
IJ_37_20=0;
IJ_37_21=0;
IJ_37_22=0;
IJ_37_23=0;
IJ_37_24=0;
IJ_37_25=0;
IJ_37_26=0;
IJ_37_27=0;
IJ_37_28=0;
IJ_37_29=0;
IJ_37_30=0;
IJ_37_31=0;
IJ_37_32=0;
IJ_37_33=0;
IJ_37_34=0;
IJ_37_35=0;
IJ_37_36=0;
IJ_37_37=0;
IJ_37_38=0;
IJ_37_39=0;
IJ_37_40=0;
IJ_37_41=0;
IJ_37_42=0;
IJ_37_43=0;
IJ_37_44=0;
IJ_37_45=0;
IJ_37_46=0;
IJ_37_47=0;
IJ_37_48=0;
IJ_37_49=0;
IJ_37_50=0;
IJ_37_51=0;
IJ_37_52=0;
IJ_37_53=0;
IJ_37_54=0;

IJ_38_1=0;
IJ_38_2=0;
IJ_38_3=0;
IJ_38_4=0;
IJ_38_5=(cos(PSI_4).*cos(THETA_4).*cos(THETA_C)+(cos(THETA_4).*sin(PHI_C)).*sin( ...
PSI_4)+(-1).*cos(PHI_C).*sin(THETA_4)).*sin(THETA_C)).*(1+(-1)).*(cos( ...
PHI_C).*cos(THETA_C).*sin(THETA_4)+cos(THETA_4)).*((-1).*cos(THETA_C)).* ...
sin(PHI_C).*sin(PSI_4)+cos(PSI_4).*sin(THETA_C))).^2).^(-1/2);
IJ_38_6=(-1).*cos(THETA_C).*cos(PHI_C).*cos(THETA_4).*sin(PSI_4)+sin(PHI_C).* ...
sin(THETA_4)).*(1+(-1)).*(cos(PHI_C).*cos(THETA_C).*sin(THETA_4)+cos( ...
THETA_4)).*((-1).*cos(THETA_C)).*sin(PHI_C)).*sin(PSI_4)+cos(PSI_4)).*sin( ...
THETA_C))).^2).^(-1/2);
IJ_38_7=0;
IJ_38_8=0;
IJ_38_9=0;
IJ_38_10=0;
IJ_38_11=0;
IJ_38_12=0;
IJ_38_13=0;
IJ_38_14=0;
IJ_38_15=0;

```

```

IJ_38_16=(-1).*cos(THETA_4).*(cos(PSI_4).*cos(THETA_C).*sin(PHI_C)+sin(PSI_4).* ...
sin(THETA_C)).*(1+(-1).*(cos(PHI_C).*cos(THETA_C).*sin(THETA_4)+cos( ...
THETA_4).*(-1).*cos(THETA_C).*sin(PHI_C).*sin(PSI_4)+cos(PSI_4).*sin( ...
THETA_C)).^2).^(-1/2);
IJ_38_17=(cos(PHI_C).*cos(THETA_4).*cos(THETA_C)+sin(THETA_4).*(cos(THETA_C).* ...
sin(PHI_C).*sin(PSI_4)+(-1).*cos(PSI_4).*sin(THETA_C))).*(1+(-1).*(cos( ...
PHI_C).*cos(THETA_C).*sin(THETA_4)+cos(THETA_4).*(-1).*cos(THETA_C).* ...
sin(PHI_C).*sin(PSI_4)+cos(PSI_4).*sin(THETA_C))).^2).^(-1/2);
IJ_38_18=0;
IJ_38_19=0;
IJ_38_20=0;
IJ_38_21=0;
IJ_38_22=0;
IJ_38_23=0;
IJ_38_24=0;
IJ_38_25=0;
IJ_38_26=0;
IJ_38_27=0;
IJ_38_28=0;
IJ_38_29=0;
IJ_38_30=0;
IJ_38_31=0;
IJ_38_32=0;
IJ_38_33=0;
IJ_38_34=0;
IJ_38_35=0;
IJ_38_36=0;
IJ_38_37=0;
IJ_38_38=0;
IJ_38_39=0;
IJ_38_40=0;
IJ_38_41=0;
IJ_38_42=0;
IJ_38_43=0;
IJ_38_44=0;
IJ_38_45=0;
IJ_38_46=0;
IJ_38_47=0;
IJ_38_48=0;
IJ_38_49=0;
IJ_38_50=0;
IJ_38_51=0;
IJ_38_52=0;
IJ_38_53=0;
IJ_38_54=0;

IJ_39_1=0;
IJ_39_2=0;
IJ_39_3=0;
IJ_39_4=0;
IJ_39_5=(cos(PHI_C).*cos(THETA_4).*sin(PSI_4)+sin(PHI_C).*sin(THETA_4)).*(cos( ...
PHI_C).^2.*cos(THETA_4).^2.*cos(THETA_C).^2+2.*cos(PHI_C).*cos(THETA_4) ...
.*cos(THETA_C).*sin(THETA_4).*(cos(THETA_C).*sin(PHI_C).*sin(PSI_4)+(-1) ...
.*cos(PSI_4).*sin(THETA_C))+sin(PSI_4).^2.*(cos(THETA_C).^2.*sin(PHI_C) ...
.^2.*sin(THETA_4).^2+sin(THETA_C).^2)+cos(PSI_4).^2.*(cos(THETA_C).^2.* ...
sin(PHI_C).^2+sin(THETA_4).^2.*sin(THETA_C).^2)+cos(PSI_4).*cos(THETA_4) ...
.^2.*sin(PHI_C).*sin(PSI_4).*sin(2.*THETA_C)).^(-1);
IJ_39_6=(1/2).*(2.*cos(PSI_4).*cos(THETA_4).*cos(THETA_C).^2+(-2).*cos(PHI_C).* ...
cos(THETA_C).*sin(THETA_4).*sin(THETA_C)+cos(THETA_4).*sin(PHI_C).*sin( ...
PSI_4).*sin(2.*THETA_C)).*(cos(PHI_C).^2.*cos(THETA_4).^2.*cos(THETA_C) ...
.^2+2.*cos(PHI_C).*cos(THETA_4).*cos(THETA_C).*sin(THETA_4).*(cos( ...
THETA_C).*sin(PHI_C).*sin(PSI_4)+(-1).*cos(PSI_4).*sin(THETA_C))+sin( ...
PSI_4).^2.*(cos(THETA_C).^2.*sin(PHI_C).^2.*sin(THETA_4).^2+sin(THETA_C) ...
.^2)+cos(PSI_4).^2.*(cos(THETA_C).^2.*sin(PHI_C).^2+sin(THETA_4).^2.* ...
sin(THETA_C).^2)+cos(PSI_4).*cos(THETA_4).^2.*sin(PHI_C).*sin(PSI_4).* ...
sin(2.*THETA_C)).^(-1);
IJ_39_7=0;
IJ_39_8=0;
IJ_39_9=0;
IJ_39_10=0;
IJ_39_11=0;

```



```

IJ_39_12=0;
IJ_39_13=0;
IJ_39_14=0;
IJ_39_15=0;
IJ_39_16=(1/4).*cos(PHI_C).^2.*cos(THETA_4).^2.*cos(THETA_C).^2+2.*cos(PHI_C).*...
cos(THETA_4).*cos(THETA_C).*sin(THETA_4).*cos(THETA_C).*sin(PHI_C).*...
sin(PSI_4)+(-1).*cos(PSI_4).*sin(THETA_C))+sin(PSI_4).^2.*(cos(THETA_C)...
.^2.*sin(PHI_C).^2.*sin(THETA_4).^2+sin(THETA_C).^2)+cos(PSI_4).^2.*( ...
cos(THETA_C).^2.*sin(PHI_C).^2+sin(THETA_4).^2.*sin(THETA_C).^2)+cos( ...
PSI_4).*cos(THETA_4).^2.*sin(PHI_C).*sin(PSI_4).*sin(2.*THETA_C)).^(-1) ...
.*((-3)+cos(2.*PHI_C)+2.*cos(PHI_C).^2.*cos(2.*THETA_C)).*sin(THETA_4)+ ...
cos(THETA_4).*((-2).*cos(THETA_C).^2.*sin(2.*PHI_C).*sin(PSI_4)+2.*cos( ...
PHI_C).*cos(PSI_4).*sin(2.*THETA_C)));
IJ_39_17=(cos(PSI_4).*cos(THETA_C).*sin(PHI_C)+sin(PSI_4).*sin(THETA_C)).*(cos( ...
PHI_C).*cos(THETA_C).*sin(THETA_4)+cos(THETA_4).*((-1).*cos(THETA_C)).*...
sin(PHI_C).*sin(PSI_4)+cos(PSI_4).*sin(THETA_C)).*(cos(PHI_C).^2.*cos( ...
THETA_4).^2.*cos(THETA_C).^2+2.*cos(PHI_C).*cos(THETA_4).*cos(THETA_C)).*...
sin(THETA_4).*cos(THETA_C).*sin(PHI_C).*sin(PSI_4)+(-1).*cos(PSI_4).*...
sin(THETA_C))+sin(PSI_4).^2.*(cos(THETA_C).^2.*sin(PHI_C).^2.*sin( ...
THETA_4).^2+sin(THETA_C).^2)+cos(PSI_4).^2.*(cos(THETA_C).^2.*sin(PHI_C) ...
.^2+sin(THETA_4).^2.*sin(THETA_C).^2)+cos(PSI_4).*cos(THETA_4).^2.*sin( ...
PHI_C).*sin(PSI_4).*sin(2.*THETA_C)).^(-1);
IJ_39_18=1;
IJ_39_19=0;
IJ_39_20=0;
IJ_39_21=0;
IJ_39_22=0;
IJ_39_23=0;
IJ_39_24=0;
IJ_39_25=0;
IJ_39_26=0;
IJ_39_27=0;
IJ_39_28=0;
IJ_39_29=0;
IJ_39_30=0;
IJ_39_31=0;
IJ_39_32=0;
IJ_39_33=0;
IJ_39_34=0;
IJ_39_35=0;
IJ_39_36=0;
IJ_39_37=0;
IJ_39_38=0;
IJ_39_39=0;
IJ_39_40=0;
IJ_39_41=0;
IJ_39_42=0;
IJ_39_43=0;
IJ_39_44=0;
IJ_39_45=0;
IJ_39_46=0;
IJ_39_47=0;
IJ_39_48=0;
IJ_39_49=0;
IJ_39_50=0;
IJ_39_51=0;
IJ_39_52=0;
IJ_39_53=0;
IJ_39_54=0;

IJ_40_1=0;
IJ_40_2=0;
IJ_40_3=0;
IJ_40_4=1;
IJ_40_5=(cos(PHI_C).*cos(THETA_5).*sin(PSI_5)+sin(PHI_C).*sin(THETA_5)).*(cos( ...
PHI_C).*cos(THETA_C).*sin(THETA_5)+cos(THETA_5).*((-1).*cos(THETA_C)).*...
sin(PHI_C).*sin(PSI_5)+cos(PSI_5).*sin(THETA_C)).*(cos(PSI_5).^2.*cos( ...
THETA_5).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C)).*...
sin(PSI_5).*sin(2.*THETA_5)+2.*cos(PSI_5).*cos(THETA_5).*cos(THETA_C)).*( ...
cos(THETA_5).*sin(PHI_C).*sin(PSI_5)+(-1).*cos(PHI_C).*sin(THETA_5)).*...
sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_5).^2+cos(THETA_5).^2.*sin(PSI_5) ...

```

```

.^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_5).^2.*sin(PSI_5).^2+ ...
sin(THETA_5).^2.*sin(THETA_C).^2)).^(-1);
IJ_40_6=(1/4).*(4.*cos(PSI_5).*cos(THETA_5).*cos(THETA_C).*(-1).*cos(THETA_5).* ...
sin(PHI_C).*sin(PSI_5)+cos(PHI_C).*sin(THETA_5))+((-3)+cos(2.*PSI_5)+2.* ...
cos(PSI_5).^2.*cos(2.*THETA_5)).*sin(THETA_C).*(cos(PSI_5).^2.*cos( ...
THETA_5).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C).* ...
sin(PSI_5).^2.*sin(2.*THETA_5)+2.*cos(PSI_5).*cos(THETA_5).*cos(THETA_C).*( ...
cos(THETA_5).*sin(PHI_C).*sin(PSI_5)+(-1).*cos(PHI_C).*sin(THETA_5)).* ...
sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_5).^2+cos(THETA_5).^2.*sin(PSI_5) ...
.^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_5).^2.*sin(PSI_5).^2+ ...
sin(THETA_5).^2.*sin(THETA_C).^2)).^(-1);
IJ_40_7=0;
IJ_40_8=0;
IJ_40_9=0;
IJ_40_10=0;
IJ_40_11=0;
IJ_40_12=0;
IJ_40_13=0;
IJ_40_14=0;
IJ_40_15=0;
IJ_40_16=0;
IJ_40_17=0;
IJ_40_18=0;
IJ_40_19=(1/2).*(2.*cos(PHI_C).*cos(THETA_5).^2.*cos(THETA_C)+cos(THETA_C).*sin( ...
PHI_C).*sin(PSI_5).*sin(2.*THETA_5)+(-2).*cos(PSI_5).*cos(THETA_5).*sin( ...
THETA_5).*sin(THETA_C)).*(cos(PSI_5).^2.*cos(THETA_5).^2.*cos(THETA_C) ...
.^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C).*sin(PSI_5).*sin(2.*THETA_5) ...
+2.*cos(PSI_5).*cos(THETA_5).*cos(THETA_C).*(cos(THETA_5).*sin(PHI_C).* ...
sin(PSI_5)+(-1).*cos(PHI_C).*sin(THETA_5)).*sin(THETA_C)+sin(PHI_C).^2.* ...
(sin(THETA_5).^2+cos(THETA_5).^2.*sin(PSI_5).^2.*sin(THETA_C).^2)+cos( ...
PHI_C).^2.*(cos(THETA_5).^2.*sin(PSI_5).^2+sin(THETA_5).^2.*sin(THETA_C) ...
.^2)).^(-1);
IJ_40_20=(cos(PSI_5).*cos(THETA_C).*sin(PHI_C)+sin(PSI_5).*sin(THETA_C)).*(cos( ...
PSI_5).^2.*cos(THETA_5).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C) ...
.^2.*sin(PHI_C).*sin(PSI_5).*sin(2.*THETA_5)+2.*cos(PSI_5).*cos(THETA_5) ...
.*cos(THETA_C).*(cos(THETA_5).*sin(PHI_C).*sin(PSI_5)+(-1).*cos(PHI_C).* ...
sin(THETA_5)).*sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_5).^2+cos(THETA_5) ...
.^2.*sin(PSI_5).^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_5).^2.* ...
sin(PSI_5).^2+sin(THETA_5).^2.*sin(THETA_C).^2)).^(-1);
IJ_40_21=0;
IJ_40_22=0;
IJ_40_23=0;
IJ_40_24=0;
IJ_40_25=0;
IJ_40_26=0;
IJ_40_27=0;
IJ_40_28=0;
IJ_40_29=0;
IJ_40_30=0;
IJ_40_31=0;
IJ_40_32=0;
IJ_40_33=0;
IJ_40_34=0;
IJ_40_35=0;
IJ_40_36=0;
IJ_40_37=0;
IJ_40_38=0;
IJ_40_39=0;
IJ_40_40=0;
IJ_40_41=0;
IJ_40_42=0;
IJ_40_43=0;
IJ_40_44=0;
IJ_40_45=0;
IJ_40_46=0;
IJ_40_47=0;
IJ_40_48=0;
IJ_40_49=0;
IJ_40_50=0;
IJ_40_51=0;
IJ_40_52=0;

```

```

IJ_40_53=0;
IJ_40_54=0;

IJ_41_1=0;
IJ_41_2=0;
IJ_41_3=0;
IJ_41_4=0;
IJ_41_5=(cos(PHI_5).*cos(THETA_5).*cos(THETA_C)+(cos(THETA_5).*sin(PHI_C).*sin( ...
    PSI_5)+(-1).*cos(PHI_C).*sin(THETA_5)).*sin(THETA_C)).*(1+(-1).*cos( ...
    PHI_C).*cos(THETA_C).*sin(THETA_5)+cos(THETA_5).*(-1).*cos(THETA_C).* ...
    sin(PHI_C).*sin(PSI_5)+cos(PSI_5).*sin(THETA_C)).^2).^(-1/2);
IJ_41_6=(-1).*cos(THETA_C).*(cos(PHI_C).*cos(THETA_5).*sin(PSI_5)+sin(PHI_C).* ...
    sin(THETA_5)).*(1+(-1).*cos(PHI_C).*cos(THETA_C).*sin(THETA_5)+cos( ...
    THETA_5).*(-1).*cos(THETA_C).*sin(PHI_C).*sin(PSI_5)+cos(PSI_5).*sin( ...
    THETA_C)).^2).^(-1/2);
IJ_41_7=0;
IJ_41_8=0;
IJ_41_9=0;
IJ_41_10=0;
IJ_41_11=0;
IJ_41_12=0;
IJ_41_13=0;
IJ_41_14=0;
IJ_41_15=0;
IJ_41_16=0;
IJ_41_17=0;
IJ_41_18=0;
IJ_41_19=(-1).*cos(THETA_5).*(cos(PSI_5).*cos(THETA_C).*sin(PHI_C)+sin(PSI_5).* ...
    sin(THETA_C)).*(1+(-1).*cos(PHI_C).*cos(THETA_C).*sin(THETA_5)+cos( ...
    THETA_5).*(-1).*cos(THETA_C).*sin(PHI_C).*sin(PSI_5)+cos(PSI_5).*sin( ...
    THETA_C)).^2).^(-1/2);
IJ_41_20=(cos(PHI_C).*cos(THETA_5).*cos(THETA_C)+sin(THETA_5).*(cos(THETA_C).* ...
    sin(PHI_C).*sin(PSI_5)+(-1).*cos(PSI_5).*sin(THETA_C))).*(1+(-1).*cos( ...
    PHI_C).*cos(THETA_C).*sin(THETA_5)+cos(THETA_5).*(-1).*cos(THETA_C).* ...
    sin(PHI_C).*sin(PSI_5)+cos(PSI_5).*sin(THETA_C)).^2).^(-1/2);
IJ_41_21=0;
IJ_41_22=0;
IJ_41_23=0;
IJ_41_24=0;
IJ_41_25=0;
IJ_41_26=0;
IJ_41_27=0;
IJ_41_28=0;
IJ_41_29=0;
IJ_41_30=0;
IJ_41_31=0;
IJ_41_32=0;
IJ_41_33=0;
IJ_41_34=0;
IJ_41_35=0;
IJ_41_36=0;
IJ_41_37=0;
IJ_41_38=0;
IJ_41_39=0;
IJ_41_40=0;
IJ_41_41=0;
IJ_41_42=0;
IJ_41_43=0;
IJ_41_44=0;
IJ_41_45=0;
IJ_41_46=0;
IJ_41_47=0;
IJ_41_48=0;
IJ_41_49=0;
IJ_41_50=0;
IJ_41_51=0;
IJ_41_52=0;
IJ_41_53=0;
IJ_41_54=0;

IJ_42_1=0;

```

```

IJ_42_2=0;
IJ_42_3=0;
IJ_42_4=0;
IJ_42_5=(cos(PHI_C).*cos(THETA_5).*sin(PSI_5)+sin(PHI_C).*sin(THETA_5)).*(cos( ...
    PHI_C).^2.*cos(THETA_5).^2.*cos(THETA_C).^2+2.*cos(PHI_C).*cos(THETA_5) ...
    .*cos(THETA_C).*sin(THETA_5).*(cos(THETA_C).*sin(PHI_C).*sin(PSI_5)+(-1) ...
    .*cos(PSI_5).*sin(THETA_C))+sin(PSI_5).^2.*(cos(THETA_C).^2.*sin(PHI_C) ...
    .^2.*sin(THETA_5).^2+sin(THETA_C).^2)+cos(PSI_5).^2.*(cos(THETA_C).^2.* ...
    sin(PHI_C).^2+sin(THETA_5).^2.*sin(THETA_C).^2)+cos(PSI_5).*cos(THETA_5) ...
    .^2.*sin(PHI_C).*sin(PSI_5).*sin(2.*THETA_C)).^(-1);
IJ_42_6=(1/2).*(2.*cos(PSI_5).*cos(THETA_5).*cos(THETA_C).^2+(-2).*cos(PHI_C).* ...
    cos(THETA_C).*sin(THETA_5).*sin(THETA_C)+cos(THETA_5).*sin(PHI_C).*sin( ...
    PSI_5).*sin(2.*THETA_C)).*(cos(PHI_C).^2.*cos(THETA_5).^2.*cos(THETA_C) ...
    .^2+2.*cos(PHI_C).*cos(THETA_5).*cos(THETA_C).*sin(THETA_5).*(cos( ...
    THETA_C).*sin(PHI_C).*sin(PSI_5)+(-1).*cos(PSI_5).*sin(THETA_C))+sin( ...
    PSI_5).^2.*(cos(THETA_C).^2.*sin(PHI_C).^2.*sin(THETA_5).^2+sin(THETA_C) ...
    .^2)+cos(PSI_5).^2.*(cos(THETA_C).^2.*sin(PHI_C).^2+sin(THETA_5).^2.* ...
    sin(THETA_C).^2)+cos(PSI_5).*cos(THETA_5).^2.*sin(PHI_C).*sin(PSI_5).* ...
    sin(2.*THETA_C)).^(-1);
IJ_42_7=0;
IJ_42_8=0;
IJ_42_9=0;
IJ_42_10=0;
IJ_42_11=0;
IJ_42_12=0;
IJ_42_13=0;
IJ_42_14=0;
IJ_42_15=0;
IJ_42_16=0;
IJ_42_17=0;
IJ_42_18=0;
IJ_42_19=(1/4).*(cos(PHI_C).^2.*cos(THETA_5).^2.*cos(THETA_C).^2+2.*cos(PHI_C).* ...
    cos(THETA_5).*cos(THETA_C).*sin(THETA_5).*(cos(THETA_C).*sin(PHI_C).* ...
    sin(PSI_5)+(-1).*cos(PSI_5).*sin(THETA_C))+sin(PSI_5).^2.*(cos(THETA_C) ...
    .^2.*sin(PHI_C).^2.*sin(THETA_5).^2+sin(THETA_C).^2)+cos(PSI_5).^2.*( ...
    cos(THETA_C).^2.*sin(PHI_C).^2+sin(THETA_5).^2.*sin(THETA_C).^2)+cos( ...
    PSI_5).*cos(THETA_5).^2.*sin(PHI_C).*sin(PSI_5).*sin(2.*THETA_C)).^(-1) ...
    .*((-3)+cos(2.*PHI_C)+2.*cos(PHI_C).^2.*cos(2.*THETA_C)).*sin(THETA_5)+ ...
    cos(THETA_5).*((-2).*cos(THETA_C).^2.*sin(2.*PHI_C).*sin(PSI_5)+2.*cos( ...
    PHI_C).*cos(PSI_5).*sin(2.*THETA_C));
IJ_42_20=(cos(PSI_5).*cos(THETA_C).*sin(PHI_C)+sin(PSI_5).*sin(THETA_C)).*(cos( ...
    PHI_C).*cos(THETA_C).*sin(THETA_5)+cos(THETA_5).*((-1).*cos(THETA_C).* ...
    sin(PHI_C).*sin(PSI_5)+cos(PSI_5).*sin(THETA_C))).*(cos(PHI_C).^2.*cos( ...
    THETA_5).^2.*cos(THETA_C).^2+2.*cos(PHI_C).*cos(THETA_5).*cos(THETA_C).* ...
    sin(THETA_5).*(cos(THETA_C).*sin(PHI_C).*sin(PSI_5)+(-1).*cos(PSI_5).* ...
    sin(THETA_C))+sin(PSI_5).^2.*(cos(THETA_C).^2.*sin(PHI_C).^2.*sin( ...
    THETA_5).^2+sin(THETA_C).^2)+cos(PSI_5).^2.*(cos(THETA_C).^2.*sin(PHI_C) ...
    .^2+sin(THETA_5).^2.*sin(THETA_C).^2)+cos(PSI_5).*cos(THETA_5).^2.*sin( ...
    PHI_C).*sin(PSI_5).*sin(2.*THETA_C)).^(-1);
IJ_42_21=1;
IJ_42_22=0;
IJ_42_23=0;
IJ_42_24=0;
IJ_42_25=0;
IJ_42_26=0;
IJ_42_27=0;
IJ_42_28=0;
IJ_42_29=0;
IJ_42_30=0;
IJ_42_31=0;
IJ_42_32=0;
IJ_42_33=0;
IJ_42_34=0;
IJ_42_35=0;
IJ_42_36=0;
IJ_42_37=0;
IJ_42_38=0;
IJ_42_39=0;
IJ_42_40=0;
IJ_42_41=0;
IJ_42_42=0;

```

```

IJ_42_43=0;
IJ_42_44=0;
IJ_42_45=0;
IJ_42_46=0;
IJ_42_47=0;
IJ_42_48=0;
IJ_42_49=0;
IJ_42_50=0;
IJ_42_51=0;
IJ_42_52=0;
IJ_42_53=0;
IJ_42_54=0;

IJ_43_1=0;
IJ_43_2=0;
IJ_43_3=0;
IJ_43_4=1;
IJ_43_5=(cos(PHI_C).*cos(THETA_6).*sin(PSI_6)+sin(PHI_C).*sin(THETA_6)).*(cos( ...
    PHI_C).*cos(THETA_C).*sin(THETA_6)+cos(THETA_6).*((-1).*cos(THETA_C)).* ...
    sin(PHI_C).*sin(PSI_6)+cos(PSI_6).*sin(THETA_C)).*(cos(PSI_6).^2.*cos( ...
    THETA_6).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C)).* ...
    sin(PSI_6).*sin(2.*THETA_6)+2.*cos(PSI_6).*cos(THETA_6).*cos(THETA_C)).*( ...
    cos(THETA_6).*sin(PHI_C).*sin(PSI_6)+(-1).*cos(PHI_C).*sin(THETA_6)).* ...
    sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_6).^2+cos(THETA_6).^2.*sin(PSI_6) ...
    .^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_6).^2.*sin(PSI_6).^2+ ...
    sin(THETA_6).^2.*sin(THETA_C).^2)).^(-1);
IJ_43_6=(1/4).*(4.*cos(PSI_6).*cos(THETA_6).*cos(THETA_C).*((-1).*cos(THETA_6)).* ...
    sin(PHI_C)).*sin(PSI_6)+cos(PHI_C).*sin(THETA_6))+((-3)+cos(2.*PSI_6)+2.* ...
    cos(PSI_6).^2.*cos(2.*THETA_6)).*sin(THETA_C)).*(cos(PSI_6).^2.*cos( ...
    THETA_6).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C)).* ...
    sin(PSI_6).*sin(2.*THETA_6)+2.*cos(PSI_6).*cos(THETA_6).*cos(THETA_C)).*( ...
    cos(THETA_6).*sin(PHI_C).*sin(PSI_6)+(-1).*cos(PHI_C).*sin(THETA_6)).* ...
    sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_6).^2+cos(THETA_6).^2.*sin(PSI_6) ...
    .^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_6).^2.*sin(PSI_6).^2+ ...
    sin(THETA_6).^2.*sin(THETA_C).^2)).^(-1);
IJ_43_7=0;
IJ_43_8=0;
IJ_43_9=0;
IJ_43_10=0;
IJ_43_11=0;
IJ_43_12=0;
IJ_43_13=0;
IJ_43_14=0;
IJ_43_15=0;
IJ_43_16=0;
IJ_43_17=0;
IJ_43_18=0;
IJ_43_19=0;
IJ_43_20=0;
IJ_43_21=0;
IJ_43_22=(1/2).*(2.*cos(PHI_C).*cos(THETA_6).^2.*cos(THETA_C)+cos(THETA_C).*sin( ...
    PHI_C)).*sin(PSI_6).*sin(2.*THETA_6)+(-2).*cos(PSI_6).*cos(THETA_6).*sin( ...
    THETA_6).*sin(THETA_C)).*(cos(PSI_6).^2.*cos(THETA_6).^2.*cos(THETA_C) ...
    .^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C)).*sin(PSI_6).*sin(2.*THETA_6) ...
    +2.*cos(PSI_6).*cos(THETA_6).*cos(THETA_C)).*(cos(THETA_6).*sin(PHI_C)).* ...
    sin(PSI_6)+(-1).*cos(PHI_C)).*sin(THETA_6)).*sin(THETA_C)+sin(PHI_C).^2.* ...
    (sin(THETA_6).^2+cos(THETA_6).^2.*sin(PSI_6).^2.*sin(THETA_C).^2)+cos( ...
    PHI_C).^2.*(cos(THETA_6).^2.*sin(PSI_6).^2+sin(THETA_6).^2.*sin(THETA_C) ...
    .^2)).^(-1);
IJ_43_23=(cos(PSI_6).*cos(THETA_C).*sin(PHI_C)+sin(PSI_6).*sin(THETA_C)).*(cos( ...
    PSI_6).^2.*cos(THETA_6).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C) ...
    .^2.*sin(PHI_C)).*sin(PSI_6).*sin(2.*THETA_6)+2.*cos(PSI_6).*cos(THETA_6) ...
    .*cos(THETA_C)).*(cos(THETA_6).*sin(PHI_C)).*sin(PSI_6)+(-1).*cos(PHI_C)).* ...
    sin(THETA_6)).*sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_6).^2+cos(THETA_6) ...
    .^2.*sin(PSI_6).^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_6).^2.* ...
    sin(PSI_6).^2+sin(THETA_6).^2.*sin(THETA_C).^2)).^(-1);
IJ_43_24=0;
IJ_43_25=0;
IJ_43_26=0;
IJ_43_27=0;
IJ_43_28=0;

```

```

IJ_43_29=0;
IJ_43_30=0;
IJ_43_31=0;
IJ_43_32=0;
IJ_43_33=0;
IJ_43_34=0;
IJ_43_35=0;
IJ_43_36=0;
IJ_43_37=0;
IJ_43_38=0;
IJ_43_39=0;
IJ_43_40=0;
IJ_43_41=0;
IJ_43_42=0;
IJ_43_43=0;
IJ_43_44=0;
IJ_43_45=0;
IJ_43_46=0;
IJ_43_47=0;
IJ_43_48=0;
IJ_43_49=0;
IJ_43_50=0;
IJ_43_51=0;
IJ_43_52=0;
IJ_43_53=0;
IJ_43_54=0;

IJ_44_1=0;
IJ_44_2=0;
IJ_44_3=0;
IJ_44_4=0;
IJ_44_5=(cos(PSI_6).*cos(THETA_6).*cos(THETA_C)+(cos(THETA_6).*sin(PHI_C).*sin( ...
    PSI_6)+(-1).*cos(PHI_C).*sin(THETA_6)).*sin(THETA_C)).*(1+(-1).*cos( ...
    PHI_C).*cos(THETA_C).*sin(THETA_6)+cos(THETA_6).*((-1).*cos(THETA_C).* ...
    sin(PHI_C).*sin(PSI_6)+cos(PSI_6).*sin(THETA_C))).^2).^(-1/2);
IJ_44_6=(-1).*cos(THETA_C).*cos(PHI_C).*cos(THETA_6).*sin(PSI_6)+sin(PHI_C).* ...
    sin(THETA_6)).*(1+(-1).*cos(PHI_C).*cos(THETA_C).*sin(THETA_6)+cos( ...
    THETA_6).*((-1).*cos(THETA_C).*sin(PHI_C).*sin(PSI_6)+cos(PSI_6).*sin( ...
    THETA_C))).^2).^(-1/2);
IJ_44_7=0;
IJ_44_8=0;
IJ_44_9=0;
IJ_44_10=0;
IJ_44_11=0;
IJ_44_12=0;
IJ_44_13=0;
IJ_44_14=0;
IJ_44_15=0;
IJ_44_16=0;
IJ_44_17=0;
IJ_44_18=0;
IJ_44_19=0;
IJ_44_20=0;
IJ_44_21=0;
IJ_44_22=(-1).*cos(THETA_6).*cos(PSI_6).*cos(THETA_C).*sin(PHI_C)+sin(PSI_6).* ...
    sin(THETA_C)).*(1+(-1).*cos(PHI_C).*cos(THETA_C).*sin(THETA_6)+cos( ...
    THETA_6).*((-1).*cos(THETA_C).*sin(PHI_C).*sin(PSI_6)+cos(PSI_6).*sin( ...
    THETA_C))).^2).^(-1/2);
IJ_44_23=(cos(PHI_C).*cos(THETA_6).*cos(THETA_C)+sin(THETA_6).*cos(THETA_C).* ...
    sin(PHI_C).*sin(PSI_6)+(-1).*cos(PSI_6).*sin(THETA_C)).*(1+(-1).*cos( ...
    PHI_C).*cos(THETA_C).*sin(THETA_6)+cos(THETA_6).*((-1).*cos(THETA_C).* ...
    sin(PHI_C).*sin(PSI_6)+cos(PSI_6).*sin(THETA_C))).^2).^(-1/2);
IJ_44_24=0;
IJ_44_25=0;
IJ_44_26=0;
IJ_44_27=0;
IJ_44_28=0;
IJ_44_29=0;
IJ_44_30=0;
IJ_44_31=0;
IJ_44_32=0;

```

```

IJ_44_33=0;
IJ_44_34=0;
IJ_44_35=0;
IJ_44_36=0;
IJ_44_37=0;
IJ_44_38=0;
IJ_44_39=0;
IJ_44_40=0;
IJ_44_41=0;
IJ_44_42=0;
IJ_44_43=0;
IJ_44_44=0;
IJ_44_45=0;
IJ_44_46=0;
IJ_44_47=0;
IJ_44_48=0;
IJ_44_49=0;
IJ_44_50=0;
IJ_44_51=0;
IJ_44_52=0;
IJ_44_53=0;
IJ_44_54=0;

IJ_45_1=0;
IJ_45_2=0;
IJ_45_3=0;
IJ_45_4=0;
IJ_45_5=(cos(PHI_C).*cos(THETA_6).*sin(PSI_6)+sin(PHI_C).*sin(THETA_6)).*(cos( ...
    PHI_C).^2.*cos(THETA_6).^2.*cos(THETA_C).^2+2.*cos(PHI_C).*cos(THETA_6) ...
    .*cos(THETA_C).*sin(THETA_6).*(cos(THETA_C).*sin(PHI_C).*sin(PSI_6)+(-1) ...
    .*cos(PSI_6).*sin(THETA_C))+sin(PSI_6).^2.*(cos(THETA_C).^2.*sin(PHI_C) ...
    ^2.*sin(THETA_6).^2+sin(THETA_C).^2)+cos(PSI_6).^2.*(cos(THETA_C).^2.*
    sin(PHI_C).^2+sin(THETA_6).^2.*sin(THETA_C).^2)+cos(PSI_6).*cos(THETA_6) ...
    ^2.*sin(PHI_C).*sin(PSI_6).*sin(2.*THETA_C)).^(-1);
IJ_45_6=(1/2).* (2.*cos(PSI_6).*cos(THETA_6).*cos(THETA_C).^2+(-2).*cos(PHI_C).* ...
    cos(THETA_C).*sin(THETA_6).*sin(THETA_C))+cos(THETA_6).*sin(PHI_C).*sin( ...
    PSI_6).*sin(2.*THETA_C)).*(cos(PHI_C).^2.*cos(THETA_6).^2.*cos(THETA_C) ...
    ^2+2.*cos(PHI_C).*cos(THETA_6).*cos(THETA_C).*sin(THETA_6).*(cos( ...
    THETA_C).*sin(PHI_C).*sin(PSI_6)+(-1).*cos(PSI_6).*sin(THETA_C))+sin( ...
    PSI_6).^2.*(cos(THETA_C).^2.*sin(PHI_C).^2.*sin(THETA_6).^2+sin(THETA_C) ...
    ^2)+cos(PSI_6).^2.*(cos(THETA_C).^2.*sin(PHI_C).^2+sin(THETA_6).^2.* ...
    sin(THETA_C).^2)+cos(PSI_6).*cos(THETA_6).^2.*sin(PHI_C).*sin(PSI_6).* ...
    sin(2.*THETA_C)).^(-1);
IJ_45_7=0;
IJ_45_8=0;
IJ_45_9=0;
IJ_45_10=0;
IJ_45_11=0;
IJ_45_12=0;
IJ_45_13=0;
IJ_45_14=0;
IJ_45_15=0;
IJ_45_16=0;
IJ_45_17=0;
IJ_45_18=0;
IJ_45_19=0;
IJ_45_20=0;
IJ_45_21=0;
IJ_45_22=(1/4).* (cos(PHI_C).^2.*cos(THETA_6).^2.*cos(THETA_C).^2+2.*cos(PHI_C).* ...
    cos(THETA_6).*cos(THETA_C).*sin(THETA_6).*(cos(THETA_C).*sin(PHI_C).* ...
    sin(PSI_6)+(-1).*cos(PSI_6).*sin(THETA_C))+sin(PSI_6).^2.*(cos(THETA_C) ...
    ^2.*sin(PHI_C).^2.*sin(THETA_6).^2+sin(THETA_C).^2)+cos(PSI_6).^2.*( ...
    cos(THETA_C).^2.*sin(PHI_C).^2+sin(THETA_6).^2.*sin(THETA_C).^2)+cos( ...
    PSI_6).*cos(THETA_6).^2.*sin(PHI_C).*sin(PSI_6).*sin(2.*THETA_C)).^(-1) ...
    .*((-3)+cos(2.*PHI_C)+2.*cos(PHI_C).^2.*cos(2.*THETA_C)).*sin(THETA_6)+ ...
    cos(THETA_6).*((-2).*cos(THETA_C).^2.*sin(2.*PHI_C).*sin(PSI_6)+2.*cos( ...
    PHI_C).*cos(PSI_6).*sin(2.*THETA_C)));
IJ_45_23=(cos(PSI_6).*cos(THETA_C).*sin(PHI_C)+sin(PSI_6).*sin(THETA_C)).*(cos( ...
    PHI_C).*cos(THETA_C).*sin(THETA_6)+cos(THETA_6).*((-1).*cos(THETA_C)).* ...
    sin(PHI_C).*sin(PSI_6)+cos(PSI_6).*sin(THETA_C)).*(cos(PHI_C).^2.*cos( ...
    THETA_6).^2.*cos(THETA_C).^2+2.*cos(PHI_C).*cos(THETA_6).*cos(THETA_C).* ...

```

```

sin(THETA_6).*(cos(THETA_C).*sin(PHI_C).*sin(PSI_6)+(-1).*cos(PSI_6)).* ...
sin(THETA_C))+sin(PSI_6).^2.*(cos(THETA_C).^2.*sin(PHI_C).^2.*sin( ...
THETA_6).^2+sin(THETA_C).^2)+cos(PSI_6).^2.*(cos(THETA_C).^2.*sin(PHI_C) ...
).^2+sin(THETA_6).^2.*sin(THETA_C).^2)+cos(PSI_6).*cos(THETA_6).^2.*sin( ...
PHI_C).*sin(PSI_6).*sin(2.*THETA_C)).^(-1);
IJ_45_24=1;
IJ_45_25=0;
IJ_45_26=0;
IJ_45_27=0;
IJ_45_28=0;
IJ_45_29=0;
IJ_45_30=0;
IJ_45_31=0;
IJ_45_32=0;
IJ_45_33=0;
IJ_45_34=0;
IJ_45_35=0;
IJ_45_36=0;
IJ_45_37=0;
IJ_45_38=0;
IJ_45_39=0;
IJ_45_40=0;
IJ_45_41=0;
IJ_45_42=0;
IJ_45_43=0;
IJ_45_44=0;
IJ_45_45=0;
IJ_45_46=0;
IJ_45_47=0;
IJ_45_48=0;
IJ_45_49=0;
IJ_45_50=0;
IJ_45_51=0;
IJ_45_52=0;
IJ_45_53=0;
IJ_45_54=0;

IJ_46_1=0;
IJ_46_2=0;
IJ_46_3=0;
IJ_46_4=1;
IJ_46_5=(cos(PHI_C).*cos(THETA_7).*sin(PSI_7)+sin(PHI_C).*sin(THETA_7)).*(cos( ...
PHI_C).*cos(THETA_C).*sin(THETA_7)+cos(THETA_7).*((-1).*cos(THETA_C)).* ...
sin(PHI_C).*sin(PSI_7)+cos(PSI_7).*sin(THETA_C)).*(cos(PSI_7).^2.*cos( ...
THETA_7).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C).* ...
sin(PSI_7).*sin(2.*THETA_7)+2.*cos(PSI_7).*cos(THETA_7).*cos(THETA_C).*( ...
cos(THETA_7).*sin(PHI_C).*sin(PSI_7)+(-1).*cos(PHI_C).*sin(THETA_7)).* ...
sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_7).^2+cos(THETA_7).^2.*sin(PSI_7) ...
).^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_7).^2.*sin(PSI_7).^2+ ...
sin(THETA_7).^2.*sin(THETA_C).^2)).^(-1);
IJ_46_6=(1/4).*(4.*cos(PSI_7).*cos(THETA_7).*cos(THETA_C).*((-1).*cos(THETA_7)).* ...
sin(PHI_C).*sin(PSI_7)+cos(PHI_C).*sin(THETA_7))+((-3)+cos(2.*PSI_7)+2.* ...
cos(PSI_7).^2.*cos(2.*THETA_7)).*sin(THETA_C)).*(cos(PSI_7).^2.*cos( ...
THETA_7).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C).* ...
sin(PSI_7).*sin(2.*THETA_7)+2.*cos(PSI_7).*cos(THETA_7).*cos(THETA_C).*( ...
cos(THETA_7).*sin(PHI_C).*sin(PSI_7)+(-1).*cos(PHI_C).*sin(THETA_7)).* ...
sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_7).^2+cos(THETA_7).^2.*sin(PSI_7) ...
).^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_7).^2.*sin(PSI_7).^2+ ...
sin(THETA_7).^2.*sin(THETA_C).^2)).^(-1);
IJ_46_7=0;
IJ_46_8=0;
IJ_46_9=0;
IJ_46_10=0;
IJ_46_11=0;
IJ_46_12=0;
IJ_46_13=0;
IJ_46_14=0;
IJ_46_15=0;
IJ_46_16=0;
IJ_46_17=0;
IJ_46_18=0;

```



```

IJ_46_19=0;
IJ_46_20=0;
IJ_46_21=0;
IJ_46_22=0;
IJ_46_23=0;
IJ_46_24=0;
IJ_46_25=(1/2).*(2.*cos(PHI_C).*cos(THETA_7).^2.*cos(THETA_C)+cos(THETA_C).*sin( ...
  PHI_C).*sin(PSI_7).*sin(2.*THETA_7)+(-2).*cos(PSI_7).*cos(THETA_7).*sin( ...
  THETA_7).*sin(THETA_C)).*(cos(PSI_7).^2.*cos(THETA_7).^2.*cos(THETA_C) ...
  .^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C).*sin(PSI_7).*sin(2.*THETA_7) ...
  +2.*cos(PSI_7).*cos(THETA_7).*cos(THETA_C)).*(cos(THETA_7).*sin(PHI_C).* ...
  sin(PSI_7)+(-1).*cos(PHI_C).*sin(THETA_7)).*sin(THETA_C)+sin(PHI_C).^2.* ...
  (sin(THETA_7).^2+cos(THETA_7).^2.*sin(PSI_7).^2.*sin(THETA_C).^2)+cos( ...
  PHI_C).^2.*cos(THETA_7).^2.*sin(PSI_7).^2+sin(THETA_7).^2.*sin(THETA_C) ...
  .^2).^(-1);
IJ_46_26=(cos(PSI_7).*cos(THETA_C).*sin(PHI_C)+sin(PSI_7).*sin(THETA_C)).*(cos( ...
  PSI_7).^2.*cos(THETA_7).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C) ...
  .^2.*sin(PHI_C).*sin(PSI_7).*sin(2.*THETA_7)+2.*cos(PSI_7).*cos(THETA_7) ...
  .*cos(THETA_C)).*(cos(THETA_7).*sin(PHI_C).*sin(PSI_7)+(-1).*cos(PHI_C).* ...
  sin(THETA_7)).*sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_7).^2+cos(THETA_7) ...
  .^2.*sin(PSI_7).^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_7).^2.* ...
  sin(PSI_7).^2+sin(THETA_7).^2.*sin(THETA_C).^2).^(-1);
IJ_46_27=0;
IJ_46_28=0;
IJ_46_29=0;
IJ_46_30=0;
IJ_46_31=0;
IJ_46_32=0;
IJ_46_33=0;
IJ_46_34=0;
IJ_46_35=0;
IJ_46_36=0;
IJ_46_37=0;
IJ_46_38=0;
IJ_46_39=0;
IJ_46_40=0;
IJ_46_41=0;
IJ_46_42=0;
IJ_46_43=0;
IJ_46_44=0;
IJ_46_45=0;
IJ_46_46=0;
IJ_46_47=0;
IJ_46_48=0;
IJ_46_49=0;
IJ_46_50=0;
IJ_46_51=0;
IJ_46_52=0;
IJ_46_53=0;
IJ_46_54=0;

IJ_47_1=0;
IJ_47_2=0;
IJ_47_3=0;
IJ_47_4=0;
IJ_47_5=(cos(PSI_7).*cos(THETA_7).*cos(THETA_C)+cos(THETA_7).*sin(PHI_C).*sin( ...
  PSI_7)+(-1).*cos(PHI_C).*sin(THETA_7)).*sin(THETA_C)).*(1+(-1).*(cos( ...
  PHI_C).*cos(THETA_C).*sin(THETA_7)+cos(THETA_7).*(-1).*cos(THETA_C).* ...
  sin(PHI_C).*sin(PSI_7)+cos(PSI_7).*sin(THETA_C))).^2).^(-1/2);
IJ_47_6=(-1).*cos(THETA_C).*(cos(PHI_C).*cos(THETA_7).*sin(PSI_7)+sin(PHI_C).* ...
  sin(THETA_7)).*(1+(-1).*(cos(PHI_C).*cos(THETA_C).*sin(THETA_7)+cos( ...
  THETA_7).*(-1).*cos(THETA_C).*sin(PHI_C).*sin(PSI_7)+cos(PSI_7).*sin( ...
  THETA_C))).^2).^(-1/2);
IJ_47_7=0;
IJ_47_8=0;
IJ_47_9=0;
IJ_47_10=0;
IJ_47_11=0;
IJ_47_12=0;
IJ_47_13=0;
IJ_47_14=0;

```

```

IJ_47_15=0;
IJ_47_16=0;
IJ_47_17=0;
IJ_47_18=0;
IJ_47_19=0;
IJ_47_20=0;
IJ_47_21=0;
IJ_47_22=0;
IJ_47_23=0;
IJ_47_24=0;
IJ_47_25=(-1).*cos(THETA_7).*cos(PSI_7).*cos(THETA_C).*sin(PHI_C)+sin(PSI_7).* ...
sin(THETA_C)).*(1+(-1).*cos(PHI_C).*cos(THETA_C).*sin(THETA_7)+cos( ...
THETA_7)).*(-1).*cos(THETA_C).*sin(PHI_C).*sin(PSI_7)+cos(PSI_7).*sin( ...
THETA_C)).^2).^(-1/2);
IJ_47_26=(cos(PHI_C).*cos(THETA_7).*cos(THETA_C)+sin(THETA_7)).*(cos(THETA_C).* ...
sin(PHI_C).*sin(PSI_7)+(-1).*cos(PSI_7).*sin(THETA_C))).*(1+(-1).*cos( ...
PHI_C).*cos(THETA_C).*sin(THETA_7)+cos(THETA_7)).*(-1).*cos(THETA_C).* ...
sin(PHI_C).*sin(PSI_7)+cos(PSI_7).*sin(THETA_C))).^2).^(-1/2);
IJ_47_27=0;
IJ_47_28=0;
IJ_47_29=0;
IJ_47_30=0;
IJ_47_31=0;
IJ_47_32=0;
IJ_47_33=0;
IJ_47_34=0;
IJ_47_35=0;
IJ_47_36=0;
IJ_47_37=0;
IJ_47_38=0;
IJ_47_39=0;
IJ_47_40=0;
IJ_47_41=0;
IJ_47_42=0;
IJ_47_43=0;
IJ_47_44=0;
IJ_47_45=0;
IJ_47_46=0;
IJ_47_47=0;
IJ_47_48=0;
IJ_47_49=0;
IJ_47_50=0;
IJ_47_51=0;
IJ_47_52=0;
IJ_47_53=0;
IJ_47_54=0;

IJ_48_1=0;
IJ_48_2=0;
IJ_48_3=0;
IJ_48_4=0;
IJ_48_5=(cos(PHI_C).*cos(THETA_7).*sin(PSI_7)+sin(PHI_C).*sin(THETA_7)).*(cos( ...
PHI_C).^2.*cos(THETA_7).^2.*cos(THETA_C).^2+2.*cos(PHI_C).*cos(THETA_7) ...
.*cos(THETA_C).*sin(THETA_7)).*(cos(THETA_C).*sin(PHI_C).*sin(PSI_7)+(-1) ...
.*cos(PSI_7).*sin(THETA_C))+sin(PSI_7).^2.*cos(THETA_C).^2.*sin(PHI_C) ...
.^2.*sin(THETA_7).^2+sin(THETA_C).^2)+cos(PSI_7).^2.*cos(THETA_C).^2.* ...
sin(PHI_C).^2+sin(THETA_7).^2.*sin(THETA_C).^2)+cos(PSI_7).*cos(THETA_7) ...
.^2.*sin(PHI_C).*sin(PSI_7).*sin(2.*THETA_C)).^(-1);
IJ_48_6=(1/2).*cos(PSI_7).*cos(THETA_7).*cos(THETA_C).^2+(-2).*cos(PHI_C).* ...
cos(THETA_C).*sin(THETA_7)).*(sin(THETA_C)+cos(THETA_7)).*(sin(PHI_C).*sin( ...
PSI_7).*sin(2.*THETA_C)).*(cos(PHI_C).^2.*cos(THETA_7).^2.*cos(THETA_C) ...
.^2+2.*cos(PHI_C).*cos(THETA_7).*cos(THETA_C).*sin(THETA_7)).*(cos( ...
THETA_C).*sin(PHI_C).*sin(PSI_7)+(-1).*cos(PSI_7).*sin(THETA_C))+sin( ...
PSI_7).^2.*cos(THETA_C).^2.*sin(PHI_C).^2.*sin(THETA_7).^2+sin(THETA_C) ...
.^2)+cos(PSI_7).^2.*cos(THETA_C).^2.*sin(PHI_C).^2+sin(THETA_7).^2.* ...
sin(THETA_C).^2)+cos(PSI_7).*cos(THETA_7).^2.*sin(PHI_C).*sin(PSI_7)).* ...
sin(2.*THETA_C)).^(-1);
IJ_48_7=0;
IJ_48_8=0;
IJ_48_9=0;
IJ_48_10=0;

```

```

IJ_48_11=0;
IJ_48_12=0;
IJ_48_13=0;
IJ_48_14=0;
IJ_48_15=0;
IJ_48_16=0;
IJ_48_17=0;
IJ_48_18=0;
IJ_48_19=0;
IJ_48_20=0;
IJ_48_21=0;
IJ_48_22=0;
IJ_48_23=0;
IJ_48_24=0;
IJ_48_25=(1/4).*(cos(PHI_C).^2.*cos(THETA_7).^2.*cos(THETA_C).^2+2.*cos(PHI_C).* ...
cos(THETA_7).*cos(THETA_C).*sin(THETA_7).*(cos(THETA_C).*sin(PHI_C).* ...
sin(PSI_7)+(-1).*cos(PSI_7).*sin(THETA_C))+sin(PSI_7).^2.*(cos(THETA_C) ...
.^2.*sin(PHI_C).^2.*sin(THETA_7).^2+sin(THETA_C).^2)+cos(PSI_7).^2.*( ...
cos(THETA_C).^2.*sin(PHI_C).^2+sin(THETA_7).^2.*sin(THETA_C).^2)+cos( ...
PSI_7).^2.*cos(THETA_7).^2.*sin(PHI_C).*sin(PSI_7).*sin(2.*THETA_C)).^(-1) ...
.*((-3)+cos(2.*PHI_C)+2.*cos(PHI_C).^2.*cos(2.*THETA_C)).*sin(THETA_7)+ ...
cos(THETA_7).*((-2).*cos(THETA_C).^2.*sin(2.*PHI_C).*sin(PSI_7)+2.*cos( ...
PHI_C).*cos(PSI_7).*sin(2.*THETA_C)));
IJ_48_26=(cos(PSI_7).*cos(THETA_C).*sin(PHI_C)+sin(PSI_7).*sin(THETA_C)).*(cos( ...
PHI_C).*cos(THETA_C).*sin(THETA_7)+cos(THETA_7).*((-1).*cos(THETA_C)).* ...
sin(PHI_C).*sin(PSI_7)+cos(PSI_7).*sin(THETA_C)).*(cos(PHI_C).^2.*cos( ...
THETA_7).^2.*cos(THETA_C).^2+2.*cos(PHI_C).*cos(THETA_7).*cos(THETA_C).* ...
sin(THETA_7).*(cos(THETA_C).*sin(PHI_C).*sin(PSI_7)+(-1).*cos(PSI_7)).* ...
sin(THETA_C))+sin(PSI_7).^2.*(cos(THETA_C).^2.*sin(PHI_C).^2.*sin( ...
THETA_7).^2+sin(THETA_C).^2)+cos(PSI_7).^2.*(cos(THETA_C).^2.*sin(PHI_C) ...
.^2+sin(THETA_7).^2.*sin(THETA_C).^2)+cos(PSI_7).*cos(THETA_7).^2.*sin( ...
PHI_C).*sin(PSI_7).*sin(2.*THETA_C)).^(-1);
IJ_48_27=1;
IJ_48_28=0;
IJ_48_29=0;
IJ_48_30=0;
IJ_48_31=0;
IJ_48_32=0;
IJ_48_33=0;
IJ_48_34=0;
IJ_48_35=0;
IJ_48_36=0;
IJ_48_37=0;
IJ_48_38=0;
IJ_48_39=0;
IJ_48_40=0;
IJ_48_41=0;
IJ_48_42=0;
IJ_48_43=0;
IJ_48_44=0;
IJ_48_45=0;
IJ_48_46=0;
IJ_48_47=0;
IJ_48_48=0;
IJ_48_49=0;
IJ_48_50=0;
IJ_48_51=0;
IJ_48_52=0;
IJ_48_53=0;
IJ_48_54=0;

IJ_49_1=0;
IJ_49_2=0;
IJ_49_3=0;
IJ_49_4=1;
IJ_49_5=(cos(PHI_C).*cos(THETA_8).*sin(PSI_8)+sin(PHI_C).*sin(THETA_8)).*(cos( ...
PHI_C).*cos(THETA_C).*sin(THETA_8)+cos(THETA_8).*((-1).*cos(THETA_C)).* ...
sin(PHI_C).*sin(PSI_8)+cos(PSI_8).*sin(THETA_C)).*(cos(PSI_8).^2.*cos( ...
THETA_8).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C)).* ...
sin(PSI_8).*sin(2.*THETA_8)+2.*cos(PSI_8).*cos(THETA_8).*cos(THETA_C)).*( ...
cos(THETA_8).*sin(PHI_C).*sin(PSI_8)+(-1).*cos(PHI_C).*sin(THETA_8)).* ...

```

```

sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_8).^2+cos(THETA_8).^2.*sin(PSI_8) ...
.^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_8).^2.*sin(PSI_8).^2+ ...
sin(THETA_8).^2.*sin(THETA_C).^2)).^(-1);
IJ_49_6=(1/4).*(4.*cos(PSI_8).*cos(THETA_8).*cos(THETA_C).*(-1).*cos(THETA_8).* ...
sin(PHI_C).*sin(PSI_8)+cos(PHI_C).*sin(THETA_8))+((-3)+cos(2.*PSI_8)+2.* ...
cos(PSI_8).^2.*cos(2.*THETA_8)).*sin(THETA_C)).*(cos(PSI_8).^2.*cos( ...
THETA_8).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C).* ...
sin(PSI_8).*sin(2.*THETA_8)+2.*cos(PSI_8).*cos(THETA_8).*cos(THETA_C).*( ...
cos(THETA_8).*sin(PHI_C).*sin(PSI_8)+(-1).*cos(PHI_C).*sin(THETA_8)).* ...
sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_8).^2+cos(THETA_8).^2.*sin(PSI_8) ...
.^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_8).^2.*sin(PSI_8).^2+ ...
sin(THETA_8).^2.*sin(THETA_C).^2)).^(-1);
IJ_49_7=0;
IJ_49_8=0;
IJ_49_9=0;
IJ_49_10=0;
IJ_49_11=0;
IJ_49_12=0;
IJ_49_13=0;
IJ_49_14=0;
IJ_49_15=0;
IJ_49_16=0;
IJ_49_17=0;
IJ_49_18=0;
IJ_49_19=0;
IJ_49_20=0;
IJ_49_21=0;
IJ_49_22=0;
IJ_49_23=0;
IJ_49_24=0;
IJ_49_25=0;
IJ_49_26=0;
IJ_49_27=0;
IJ_49_28=(1/2).*(2.*cos(PHI_C).*cos(THETA_8).^2.*cos(THETA_C)+cos(THETA_C).*sin( ...
PHI_C).*sin(PSI_8).*sin(2.*THETA_8)+(-2).*cos(PSI_8).*cos(THETA_8).*sin( ...
THETA_8).*sin(THETA_C)).*(cos(PSI_8).^2.*cos(THETA_8).^2.*cos(THETA_C) ...
.^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C).*sin(PSI_8).*sin(2.*THETA_8) ...
+2.*cos(PSI_8).*cos(THETA_8).*cos(THETA_C).*(cos(THETA_8).*sin(PHI_C).* ...
sin(PSI_8)+(-1).*cos(PHI_C).*sin(THETA_8)).*sin(THETA_C)+sin(PHI_C).^2.* ...
(sin(THETA_8).^2+cos(THETA_8).^2.*sin(PSI_8).^2.*sin(THETA_C).^2)+cos( ...
PHI_C).^2.*(cos(THETA_8).^2.*sin(PSI_8).^2+sin(THETA_8).^2.*sin(THETA_C) ...
.^2)).^(-1);
IJ_49_29=(cos(PSI_8).*cos(THETA_C).*sin(PHI_C)+sin(PSI_8).*sin(THETA_C)).*(cos( ...
PSI_8).^2.*cos(THETA_8).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C) ...
.^2.*sin(PHI_C).*sin(PSI_8).*sin(2.*THETA_8)+2.*cos(PSI_8).*cos(THETA_8) ...
.*cos(THETA_C).*(cos(THETA_8).*sin(PHI_C).*sin(PSI_8)+(-1).*cos(PHI_C).* ...
sin(THETA_8)).*sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_8).^2+cos(THETA_8) ...
.^2.*sin(PSI_8).^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_8).^2.* ...
sin(PSI_8).^2+sin(THETA_8).^2.*sin(THETA_C).^2)).^(-1);
IJ_49_30=0;
IJ_49_31=0;
IJ_49_32=0;
IJ_49_33=0;
IJ_49_34=0;
IJ_49_35=0;
IJ_49_36=0;
IJ_49_37=0;
IJ_49_38=0;
IJ_49_39=0;
IJ_49_40=0;
IJ_49_41=0;
IJ_49_42=0;
IJ_49_43=0;
IJ_49_44=0;
IJ_49_45=0;
IJ_49_46=0;
IJ_49_47=0;
IJ_49_48=0;
IJ_49_49=0;
IJ_49_50=0;
IJ_49_51=0;

```

```

IJ_49_52=0;
IJ_49_53=0;
IJ_49_54=0;

IJ_50_1=0;
IJ_50_2=0;
IJ_50_3=0;
IJ_50_4=0;
IJ_50_5=(cos(PHI_8).*cos(THETA_8).*cos(THETA_C)+(cos(THETA_8).*sin(PHI_C).*sin( ...
    PSI_8)+(-1).*cos(PHI_C).*sin(THETA_8)).*sin(THETA_C)).*(1+(-1).*cos( ...
    PHI_C).*cos(THETA_C).*sin(THETA_8)+cos(THETA_8).*((-1).*cos(THETA_C).* ...
    sin(PHI_C).*sin(PSI_8)+cos(PSI_8).*sin(THETA_C))).^2).^(-1/2);
IJ_50_6=(-1).*cos(THETA_C).*(cos(PHI_C).*cos(THETA_8).*sin(PSI_8)+sin(PHI_C).* ...
    sin(THETA_8)).*(1+(-1).*cos(PHI_C).*cos(THETA_C).*sin(THETA_8)+cos( ...
    THETA_8)).*((-1).*cos(THETA_C).*sin(PHI_C).*sin(PSI_8)+cos(PSI_8).*sin( ...
    THETA_C))).^2).^(-1/2);
IJ_50_7=0;
IJ_50_8=0;
IJ_50_9=0;
IJ_50_10=0;
IJ_50_11=0;
IJ_50_12=0;
IJ_50_13=0;
IJ_50_14=0;
IJ_50_15=0;
IJ_50_16=0;
IJ_50_17=0;
IJ_50_18=0;
IJ_50_19=0;
IJ_50_20=0;
IJ_50_21=0;
IJ_50_22=0;
IJ_50_23=0;
IJ_50_24=0;
IJ_50_25=0;
IJ_50_26=0;
IJ_50_27=0;
IJ_50_28=(-1).*cos(THETA_8).*(cos(PSI_8).*cos(THETA_C).*sin(PHI_C)+sin(PSI_8).* ...
    sin(THETA_C)).*(1+(-1).*cos(PHI_C).*cos(THETA_C).*sin(THETA_8)+cos( ...
    THETA_8)).*((-1).*cos(THETA_C).*sin(PHI_C).*sin(PSI_8)+cos(PSI_8).*sin( ...
    THETA_C))).^2).^(-1/2);
IJ_50_29=(cos(PHI_C).*cos(THETA_8).*cos(THETA_C)+sin(THETA_8).*(cos(THETA_C).* ...
    sin(PHI_C).*sin(PSI_8)+(-1).*cos(PSI_8).*sin(THETA_C))).*(1+(-1).*cos( ...
    PHI_C).*cos(THETA_C).*sin(THETA_8)+cos(THETA_8).*((-1).*cos(THETA_C).* ...
    sin(PHI_C).*sin(PSI_8)+cos(PSI_8).*sin(THETA_C))).^2).^(-1/2);
IJ_50_30=0;
IJ_50_31=0;
IJ_50_32=0;
IJ_50_33=0;
IJ_50_34=0;
IJ_50_35=0;
IJ_50_36=0;
IJ_50_37=0;
IJ_50_38=0;
IJ_50_39=0;
IJ_50_40=0;
IJ_50_41=0;
IJ_50_42=0;
IJ_50_43=0;
IJ_50_44=0;
IJ_50_45=0;
IJ_50_46=0;
IJ_50_47=0;
IJ_50_48=0;
IJ_50_49=0;
IJ_50_50=0;
IJ_50_51=0;
IJ_50_52=0;
IJ_50_53=0;
IJ_50_54=0;

```

```

IJ_51_1=0;
IJ_51_2=0;
IJ_51_3=0;
IJ_51_4=0;
IJ_51_5=(cos(PHI_C).*cos(THETA_8).*sin(PSI_8)+sin(PHI_C).*sin(THETA_8)).*(cos( ...
PHI_C).^2.*cos(THETA_8).^2.*cos(THETA_C).^2+2.*cos(PHI_C).*cos(THETA_8) ...
.*cos(THETA_C).*sin(THETA_8).*(cos(THETA_C).*sin(PHI_C).*sin(PSI_8)+(-1) ...
.*cos(PSI_8).*sin(THETA_C))+sin(PSI_8).^2.*(cos(THETA_C).^2.*sin(PHI_C) ...
.^2.*sin(THETA_8).^2+sin(THETA_C).^2)+cos(PSI_8).^2.*(cos(THETA_C).^2.* ...
sin(PHI_C).^2+sin(THETA_8).^2.*sin(THETA_C).^2)+cos(PSI_8).*cos(THETA_8) ...
.^2.*sin(PHI_C).*sin(PSI_8).*sin(2.*THETA_C)).^(-1);
IJ_51_6=(1/2).*(2.*cos(PSI_8).*cos(THETA_8).*cos(THETA_C).^2+(-2).*cos(PHI_C).* ...
cos(THETA_C).*sin(THETA_8).*sin(THETA_C)+cos(THETA_8).*sin(PHI_C).*sin( ...
PSI_8).*sin(2.*THETA_C)).*(cos(PHI_C).^2.*cos(THETA_8).^2.*cos(THETA_C) ...
.^2+2.*cos(PHI_C).*cos(THETA_8).*cos(THETA_C).*sin(THETA_8).*(cos( ...
THETA_C).*sin(PHI_C).*sin(PSI_8)+(-1).*cos(PSI_8).*sin(THETA_C))+sin( ...
PSI_8).^2.*(cos(THETA_C).^2.*sin(PHI_C).^2.*sin(THETA_8).^2+sin(THETA_C) ...
.^2)+cos(PSI_8).^2.*(cos(THETA_C).^2.*sin(PHI_C).^2+sin(THETA_8).^2.* ...
sin(THETA_C).^2)+cos(PSI_8).*cos(THETA_8).^2.*sin(PHI_C).*sin(PSI_8).* ...
sin(2.*THETA_C)).^(-1);
IJ_51_7=0;
IJ_51_8=0;
IJ_51_9=0;
IJ_51_10=0;
IJ_51_11=0;
IJ_51_12=0;
IJ_51_13=0;
IJ_51_14=0;
IJ_51_15=0;
IJ_51_16=0;
IJ_51_17=0;
IJ_51_18=0;
IJ_51_19=0;
IJ_51_20=0;
IJ_51_21=0;
IJ_51_22=0;
IJ_51_23=0;
IJ_51_24=0;
IJ_51_25=0;
IJ_51_26=0;
IJ_51_27=0;
IJ_51_28=(1/4).*(cos(PHI_C).^2.*cos(THETA_8).^2.*cos(THETA_C).^2+2.*cos(PHI_C).* ...
cos(THETA_8).*cos(THETA_C).*sin(THETA_8).*(cos(THETA_C).*sin(PHI_C).* ...
sin(PSI_8)+(-1).*cos(PSI_8).*sin(THETA_C))+sin(PSI_8).^2.*(cos(THETA_C) ...
.^2.*sin(PHI_C).^2.*sin(THETA_8).^2+sin(THETA_C).^2)+cos(PSI_8).^2.*( ...
cos(THETA_C).^2.*sin(PHI_C).^2+sin(THETA_8).^2.*sin(THETA_C).^2)+cos( ...
PSI_8).*cos(THETA_8).^2.*sin(PHI_C).*sin(PSI_8).*sin(2.*THETA_C)).^(-1) ...
.*((-3)+cos(2.*PHI_C)+2.*cos(PHI_C).^2.*cos(2.*THETA_C)).*sin(THETA_8)+ ...
cos(THETA_8).*((-2).*cos(THETA_C).^2.*sin(2.*PHI_C).*sin(PSI_8)+2.*cos( ...
PHI_C).*cos(PSI_8).*sin(2.*THETA_C)));
IJ_51_29=(cos(PSI_8).*cos(THETA_C).*sin(PHI_C)+sin(PSI_8).*sin(THETA_C)).*(cos( ...
PHI_C).*cos(THETA_C).*sin(THETA_8)+cos(THETA_8).*((-1).*cos(THETA_C).* ...
sin(PHI_C).*sin(PSI_8)+cos(PSI_8).*sin(THETA_C))).*(cos(PHI_C).^2.*cos( ...
THETA_8).^2.*cos(THETA_C).^2+2.*cos(PHI_C).*cos(THETA_8).*cos(THETA_C).* ...
sin(THETA_8).*(cos(THETA_C).*sin(PHI_C).*sin(PSI_8)+(-1).*cos(PSI_8).* ...
sin(THETA_C))+sin(PSI_8).^2.*(cos(THETA_C).^2.*sin(PHI_C).^2.*sin( ...
THETA_8).^2+sin(THETA_C).^2)+cos(PSI_8).^2.*(cos(THETA_C).^2.*sin(PHI_C) ...
.^2+sin(THETA_8).^2.*sin(THETA_C).^2)+cos(PSI_8).*cos(THETA_8).^2.*sin( ...
PHI_C).*sin(PSI_8).*sin(2.*THETA_C)).^(-1);
IJ_51_30=1;
IJ_51_31=0;
IJ_51_32=0;
IJ_51_33=0;
IJ_51_34=0;
IJ_51_35=0;
IJ_51_36=0;
IJ_51_37=0;
IJ_51_38=0;
IJ_51_39=0;
IJ_51_40=0;
IJ_51_41=0;

```

```

IJ_51_42=0;
IJ_51_43=0;
IJ_51_44=0;
IJ_51_45=0;
IJ_51_46=0;
IJ_51_47=0;
IJ_51_48=0;
IJ_51_49=0;
IJ_51_50=0;
IJ_51_51=0;
IJ_51_52=0;
IJ_51_53=0;
IJ_51_54=0;

IJ_52_1=0;
IJ_52_2=0;
IJ_52_3=0;
IJ_52_4=1;
IJ_52_5=(cos(PHI_C).*cos(THETA_9).*sin(PSI_9)+sin(PHI_C).*sin(THETA_9)).*(cos( ...
    PHI_C).*cos(THETA_C).*sin(THETA_9)+cos(THETA_9).*((-1).*cos(THETA_C)).* ...
    sin(PHI_C).*sin(PSI_9)+cos(PSI_9).*sin(THETA_C))).*(cos(PSI_9).^2.*cos( ...
    THETA_9).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C)).* ...
    sin(PSI_9).*sin(2.*THETA_9)+2.*cos(PSI_9).*cos(THETA_9).*cos(THETA_C)).*( ...
    cos(THETA_9).*sin(PHI_C).*sin(PSI_9)+(-1).*cos(PHI_C).*sin(THETA_9)).* ...
    sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_9).^2+cos(THETA_9).^2.*sin(PSI_9) ...
    .^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_9).^2.*sin(PSI_9).^2+ ...
    sin(THETA_9).^2.*sin(THETA_C).^2)).^(-1);
IJ_52_6=(1/4).**(4.*cos(PSI_9).*cos(THETA_9).*cos(THETA_C).*((-1).*cos(THETA_9)).* ...
    sin(PHI_C).*sin(PSI_9)+cos(PHI_C).*sin(THETA_9))+((-3)+cos(2.*PSI_9)+2.* ...
    cos(PSI_9).^2.*cos(2.*THETA_9)).*sin(THETA_C)).*(cos(PSI_9).^2.*cos( ...
    THETA_9).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C)).* ...
    sin(PSI_9).*sin(2.*THETA_9)+2.*cos(PSI_9).*cos(THETA_9).*cos(THETA_C)).*( ...
    cos(THETA_9).*sin(PHI_C).*sin(PSI_9)+(-1).*cos(PHI_C).*sin(THETA_9)).* ...
    sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_9).^2+cos(THETA_9).^2.*sin(PSI_9) ...
    .^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_9).^2.*sin(PSI_9).^2+ ...
    sin(THETA_9).^2.*sin(THETA_C).^2)).^(-1);
IJ_52_7=0;
IJ_52_8=0;
IJ_52_9=0;
IJ_52_10=0;
IJ_52_11=0;
IJ_52_12=0;
IJ_52_13=0;
IJ_52_14=0;
IJ_52_15=0;
IJ_52_16=0;
IJ_52_17=0;
IJ_52_18=0;
IJ_52_19=0;
IJ_52_20=0;
IJ_52_21=0;
IJ_52_22=0;
IJ_52_23=0;
IJ_52_24=0;
IJ_52_25=0;
IJ_52_26=0;
IJ_52_27=0;
IJ_52_28=0;
IJ_52_29=0;
IJ_52_30=0;
IJ_52_31=(1/2).**(2.*cos(PHI_C).*cos(THETA_9).^2.*cos(THETA_C)+cos(THETA_C).*sin( ...
    PHI_C).*sin(PSI_9).*sin(2.*THETA_9)+(-2).*cos(PSI_9).*cos(THETA_9).*sin( ...
    THETA_9).*sin(THETA_C)).*(cos(PSI_9).^2.*cos(THETA_9).^2.*cos(THETA_C) ...
    .^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C).*sin(PSI_9).*sin(2.*THETA_9) ...
    +2.*cos(PSI_9).*cos(THETA_9).*cos(THETA_C)).*(cos(THETA_9).*sin(PHI_C)).* ...
    sin(PSI_9)+(-1).*cos(PHI_C).*sin(THETA_9)).*sin(THETA_C)+sin(PHI_C).^2.* ...
    (sin(THETA_9).^2+cos(THETA_9).^2.*sin(PSI_9).^2.*sin(THETA_C).^2)+cos( ...
    PHI_C).^2.*(cos(THETA_9).^2.*sin(PSI_9).^2+sin(THETA_9).^2.*sin(THETA_C) ...
    .^2)).^(-1);
IJ_52_32=(cos(PSI_9).*cos(THETA_C).*sin(PHI_C)+sin(PSI_9).*sin(THETA_C)).*(cos( ...
    PSI_9).^2.*cos(THETA_9).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C) ...

```

```

.^2.*sin(PHI_C).*sin(PHI_9).*sin(2.*THETA_9)+2.*cos(PHI_9).*cos(THETA_9) ...
.*cos(THETA_C).*cos(THETA_9).*sin(PHI_C).*sin(PHI_9)+(-1).*cos(PHI_C).* ...
sin(THETA_9)).*sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_9).^2+cos(THETA_9) ...
.^2.*sin(PHI_9).^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_9).^2.* ...
sin(PHI_9).^2+sin(THETA_9).^2.*sin(THETA_C).^2)).^(-1);
IJ_52_33=0;
IJ_52_34=0;
IJ_52_35=0;
IJ_52_36=0;
IJ_52_37=0;
IJ_52_38=0;
IJ_52_39=0;
IJ_52_40=0;
IJ_52_41=0;
IJ_52_42=0;
IJ_52_43=0;
IJ_52_44=0;
IJ_52_45=0;
IJ_52_46=0;
IJ_52_47=0;
IJ_52_48=0;
IJ_52_49=0;
IJ_52_50=0;
IJ_52_51=0;
IJ_52_52=0;
IJ_52_53=0;
IJ_52_54=0;

IJ_53_1=0;
IJ_53_2=0;
IJ_53_3=0;
IJ_53_4=0;
IJ_53_5=(cos(PHI_9).*cos(THETA_9).*cos(THETA_C)+(cos(THETA_9).*sin(PHI_C).*sin( ...
PSI_9)+(-1).*cos(PHI_C).*sin(THETA_9)).*sin(THETA_C)).*(1+(-1).*cos( ...
PHI_C).*cos(THETA_C).*sin(THETA_9)+cos(THETA_9).*(-1).*cos(THETA_C).* ...
sin(PHI_C).*sin(PHI_9)+cos(PHI_9).*sin(THETA_C)).^2).^(-1/2);
IJ_53_6=(-1).*cos(THETA_C).*cos(PHI_C).*cos(THETA_9).*sin(PHI_9)+sin(PHI_C).* ...
sin(THETA_9)).*(1+(-1).*cos(PHI_C).*cos(THETA_C).*sin(THETA_9)+cos( ...
THETA_9).*(-1).*cos(THETA_C).*sin(PHI_C).*sin(PHI_9)+cos(PHI_9).*sin( ...
THETA_C)).^2).^(-1/2);
IJ_53_7=0;
IJ_53_8=0;
IJ_53_9=0;
IJ_53_10=0;
IJ_53_11=0;
IJ_53_12=0;
IJ_53_13=0;
IJ_53_14=0;
IJ_53_15=0;
IJ_53_16=0;
IJ_53_17=0;
IJ_53_18=0;
IJ_53_19=0;
IJ_53_20=0;
IJ_53_21=0;
IJ_53_22=0;
IJ_53_23=0;
IJ_53_24=0;
IJ_53_25=0;
IJ_53_26=0;
IJ_53_27=0;
IJ_53_28=0;
IJ_53_29=0;
IJ_53_30=0;
IJ_53_31=(-1).*cos(THETA_9).*cos(PHI_9).*cos(THETA_C).*sin(PHI_C)+sin(PHI_9).* ...
sin(THETA_C)).*(1+(-1).*cos(PHI_C).*cos(THETA_C).*sin(THETA_9)+cos( ...
THETA_9).*(-1).*cos(THETA_C).*sin(PHI_C).*sin(PHI_9)+cos(PHI_9).*sin( ...
THETA_C)).^2).^(-1/2);
IJ_53_32=(cos(PHI_C).*cos(THETA_9).*cos(THETA_C)+sin(THETA_9).*cos(THETA_C).* ...
sin(PHI_C).*sin(PHI_9)+(-1).*cos(PHI_9).*sin(THETA_C)).*(1+(-1).*cos( ...
PHI_C).*cos(THETA_C).*sin(THETA_9)+cos(THETA_9).*(-1).*cos(THETA_C).* ...

```



```

sin(PHI_C).*sin(PHI_9)+cos(PHI_9).*sin(THETA_C)).^2).^(-1/2);
IJ_53_33=0;
IJ_53_34=0;
IJ_53_35=0;
IJ_53_36=0;
IJ_53_37=0;
IJ_53_38=0;
IJ_53_39=0;
IJ_53_40=0;
IJ_53_41=0;
IJ_53_42=0;
IJ_53_43=0;
IJ_53_44=0;
IJ_53_45=0;
IJ_53_46=0;
IJ_53_47=0;
IJ_53_48=0;
IJ_53_49=0;
IJ_53_50=0;
IJ_53_51=0;
IJ_53_52=0;
IJ_53_53=0;
IJ_53_54=0;

IJ_54_1=0;
IJ_54_2=0;
IJ_54_3=0;
IJ_54_4=0;
IJ_54_5=(cos(PHI_C).*cos(THETA_9).*sin(PHI_9)+sin(PHI_C).*sin(THETA_9)).*(cos( ...
    PHI_C).^2.*cos(THETA_9).^2.*cos(THETA_C).^2+2.*cos(PHI_C).*cos(THETA_9) ...
    .*cos(THETA_C).*sin(THETA_9).*cos(THETA_C).*sin(PHI_C).*sin(PHI_9)+(-1) ...
    .*cos(PHI_9).*sin(THETA_C))+sin(PHI_9).^2.*(cos(THETA_C).^2.*sin(PHI_C) ...
    .*sin(THETA_9).^2+sin(THETA_C).^2)+cos(PHI_9).^2.*(cos(THETA_C).^2.* ...
    sin(PHI_C).^2+sin(THETA_9).^2.*sin(THETA_C).^2)+cos(PHI_9).*cos(THETA_9) ...
    .*sin(PHI_C).*sin(PHI_9).*sin(2.*THETA_C)).^(-1);
IJ_54_6=(1/2).*(2.*cos(PHI_9).*cos(THETA_9).*cos(THETA_C).^2+(-2).*cos(PHI_C).* ...
    cos(THETA_C).*sin(THETA_9).*sin(THETA_C)+cos(THETA_C).*sin(PHI_C).*sin( ...
    PHI_9).*sin(2.*THETA_C)).*(cos(PHI_C).^2.*cos(THETA_9).^2.*cos(THETA_C) ...
    .*sin(THETA_9).*cos(THETA_C).*sin(THETA_9).*cos( ...
    PHI_C).*sin(PHI_C).*sin(PHI_9)+(-1).*cos(PHI_9).*sin(THETA_C))+sin( ...
    PHI_9).^2.*(cos(THETA_C).^2.*sin(PHI_C).^2.*sin(THETA_9).^2+sin(THETA_C) ...
    .*sin(THETA_9).^2)+cos(PHI_9).^2.*(cos(THETA_C).^2.*sin(PHI_C).^2+sin(THETA_9).^2.* ...
    sin(THETA_C).^2)+cos(PHI_9).*cos(THETA_9).^2.*sin(PHI_C).*sin(PHI_9).* ...
    sin(2.*THETA_C)).^(-1);
IJ_54_7=0;
IJ_54_8=0;
IJ_54_9=0;
IJ_54_10=0;
IJ_54_11=0;
IJ_54_12=0;
IJ_54_13=0;
IJ_54_14=0;
IJ_54_15=0;
IJ_54_16=0;
IJ_54_17=0;
IJ_54_18=0;
IJ_54_19=0;
IJ_54_20=0;
IJ_54_21=0;
IJ_54_22=0;
IJ_54_23=0;
IJ_54_24=0;
IJ_54_25=0;
IJ_54_26=0;
IJ_54_27=0;
IJ_54_28=0;
IJ_54_29=0;
IJ_54_30=0;
IJ_54_31=(1/4).*(cos(PHI_C).^2.*cos(THETA_9).^2.*cos(THETA_C).^2+2.*cos(PHI_C).* ...
    cos(THETA_9).*cos(THETA_C).*sin(THETA_9).*cos(THETA_C).*sin(PHI_C).* ...
    sin(PHI_9)+(-1).*cos(PHI_9).*sin(THETA_C))+sin(PHI_9).^2.*(cos(THETA_C) ...

```

```

.^2.*sin(PHI_C).^2.*sin(THETA_9).^2+sin(THETA_C).^2)+cos(PHI_9).^2.*( ...
cos(THETA_C).^2.*sin(PHI_C).^2+sin(THETA_9).^2.*sin(THETA_C).^2)+cos( ...
PSI_9).*cos(THETA_9).^2.*sin(PHI_C).*sin(PSI_9).*sin(2.*THETA_C)).^(-1) ...
.*((-3)+cos(2.*PHI_C)+2.*cos(PHI_C).^2.*cos(2.*THETA_C)).*sin(THETA_9)+ ...
cos(THETA_9).*(-2).*cos(THETA_C).^2.*sin(2.*PHI_C).*sin(PSI_9)+2.*cos( ...
PHI_C).*cos(PSI_9).*sin(2.*THETA_C));
IJ_54_32=(cos(PSI_9).*cos(THETA_C).*sin(PHI_C)+sin(PSI_9).*sin(THETA_C)).*(cos( ...
PHI_C).*cos(THETA_C).*sin(THETA_9)+cos(THETA_9).*(-1).*cos(THETA_C).* ...
sin(PHI_C).*sin(PSI_9)+cos(PSI_9).*sin(THETA_C)).*(cos(PHI_C).^2.*cos( ...
THETA_9).^2.*cos(THETA_C).^2+2.*cos(PHI_C).*cos(THETA_9).*cos(THETA_C).* ...
sin(THETA_9)).*(cos(THETA_C).*sin(PHI_C).*sin(PSI_9)+(-1).*cos(PSI_9)).* ...
sin(THETA_C))+sin(PSI_9).^2.*(cos(THETA_C).^2.*sin(PHI_C).^2.*sin( ...
THETA_9).^2+sin(THETA_C).^2)+cos(PSI_9).^2.*(cos(THETA_C).^2.*sin(PHI_C) ...
.^2+sin(THETA_9).^2.*sin(THETA_C).^2)+cos(PSI_9).*cos(THETA_9).^2.*sin( ...
PHI_C).*sin(PSI_9).*sin(2.*THETA_C)).^(-1);
IJ_54_33=1;
IJ_54_34=0;
IJ_54_35=0;
IJ_54_36=0;
IJ_54_37=0;
IJ_54_38=0;
IJ_54_39=0;
IJ_54_40=0;
IJ_54_41=0;
IJ_54_42=0;
IJ_54_43=0;
IJ_54_44=0;
IJ_54_45=0;
IJ_54_46=0;
IJ_54_47=0;
IJ_54_48=0;
IJ_54_49=0;
IJ_54_50=0;
IJ_54_51=0;
IJ_54_52=0;
IJ_54_53=0;
IJ_54_54=0;

```

APPENDIX G

(Simulation Files and Descriptions)

(This section contains 7 pages)

Stationary Source Seeking in a Static Field Simulation

Simulation Execution Files ('Run' files in consecutive order)

<u>Filename</u>	<u>Description</u>
A_Four_Robot_Cluster_Trajectory_Generator_Trial_i.m	User-specified initial and final values of cluster-space variables to generate trajectories using cubic spline User-specified total simulation time/duration User-specified simulation time step size User-specified cluster frame origin initial coordinates w.r.t. G User-option to plot cluster-space variable trajectories
B_Four_Robot_Cluster_Simulation_Parameters_Trial_i.m	User-specified initial "actual" values of cluster-space variables (does not have to equal initial value in "A_Four_Robot_Cluster_Trajectory_Generator_Trial_i.m" to allow for differences in "desired" and "actual" cluster-space variables) User-specified cluster translational speed User-specified gains for resolved-rate proportional control law User-specified selector matrices for position/velocity control within resolved-rate proportional control law User-specified sensor and position noise magnitudes User-specified wind model input parameters User-specified time constant for 1 st order UAV model (optional)
C1_Four_Robot_Cluster_Inverse_Position_Kinematics.m	Transformation for position variables from cluster-space to "robot-space" for simulation initialization Plots simple representation of cluster and robot frames in 3D
C2_Generate_Constant_Speed_Trajectory_of_Moving_Source.m	Generate trajectory of moving source (file needs to be run even for stationary source)
D_Dynamic_Plume_Tracking_AN.slx	Simulation SIMULINK model

Simulation Support Files

<u>Filename</u>	<u>Description</u>
arclength.m	Compute arc length of space curve represented as sequence of points (for moving scalar field plume)
interparc.m	Interpolate points along curve (for moving scalar field plume)
Gradient_Ascent_3D.m	Compute 3D gradient (file 1 of 2)
gradient_calc_3D.m	Compute 3D gradient (file 2 of 2)
Plume_Moving.m	Simulated scalar field plume User-specified scalar field maximum value, half-strength distance, and shape function

Post-Processing Scripts

<u>Filename</u>	<u>Description</u>
E_Four_Robot_Cluster_Scalar_Field_and_Origin_Plot_Routine.m	Plots average of sensed scalar field and cluster point B coordinates w.r.t. G
E_Four_Robot_Cluster_Error_Plot_Routine_for_IEEE_Paper.m	Plots cluster-space variable errors and computes cluster formation RMS errors
F_Four_Robot_Cluster_3D_Flight_Path_Plot_Routine.m	Plots flight path of cluster in 3D (origin point for cluster frame and individual robot frames)
F_3D_Plot_Routine_for_Multiple_Cases.m	Plots flight paths of clusters in 3D for multiple trials (origin point for cluster frame and individual robot frames)

Moving Source Seeking and Tracking in a Time-Varying Field Simulation

Simulation Execution Files ('Run' files in consecutive order)

<u>Filename</u>	<u>Description</u>
A_Four_Robot_Cluster_Trajectory_Generator.m	<p>User-specified initial and final values of cluster-space variables to generate trajectories using cubic spline</p> <p>User-specified total simulation time/duration</p> <p>User-specified simulation time step size</p> <p>User-specified cluster frame origin initial coordinates w.r.t. G</p> <p>User-option to plot cluster-space variable trajectories</p>
B_Four_Robot_Cluster_Simulation_Parameters.m	<p>User-specified initial "actual" values of cluster-space variables (does not have to equal initial value in "A_Four_Robot_Cluster_Trajectory_Generator.m" to allow for differences in "desired" and "actual" cluster-space variables)</p> <p>User-specified cluster translational speed</p> <p>User-specified gains for resolved-rate proportional control law</p> <p>User-specified selector matrices for position/velocity control within resolved-rate proportional control law</p> <p>User-specified sensor and position noise magnitudes</p> <p>User-specified wind model input parameters</p> <p>User-specified time constant for 1st order UAV model (optional)</p>
C1_Four_Robot_Cluster_Inverse_Position_Kinematics.m	<p>Transformation for position variables from cluster-space to "robot-space" for simulation initialization</p> <p>Plots simple representation of cluster and robot frames in 3D</p>
C2_Generate_Constant_Speed_Trajectory_of_Moving_Source.m	Generate trajectory of moving source
D_Dynamic_Plume_Tracking_AN.slx	Simulation SIMULINK model

Simulation Support Files

<u>Filename</u>	<u>Description</u>
arclength.m	Compute arc length of space curve represented as sequence of points (for moving scalar field plume)
interparc.m	Interpolate points along curve (for moving scalar field plume)
Gradient_Ascent_3D.m	Compute 3D gradient (file 1 of 2)
gradient_calc_3D.m	Compute 3D gradient (file 2 of 2)
Plume_Moving.m	<p>Simulated scalar field plume</p> <p>User-specified scalar field maximum value, half-strength distance, and shape function</p>

Post-Processing Scripts

<u>Filename</u>	<u>Description</u>
E_Four_Robot_Cluster_Scalar_Field_and-Origin_Plot_Routine.m	Plots average of sensed scalar field and cluster point B coordinates w.r.t. G
E_Four_Robot_Cluster_Error_Plot_Routine_for_IEEE_Paper.m	Plots cluster-space variable errors and computes cluster formation RMS errors
F_Four_Robot_Cluster_3D_Flight_Path_Plot_Routine.m	Plots flight path of cluster in 3D (origin point for cluster frame and individual robot frames)

Basic Isosurface Navigation Simulation

Simulation Execution Files ('Run' files in consecutive order)

<u>Filename</u>	<u>Description</u>
A_Four_Robot_Cluster_Trajectory_Generator.m	<p>User-specified initial and final values of cluster-space variables to generate trajectories using cubic spline</p> <p>User-specified total simulation time/duration</p> <p>User-specified simulation time step size</p> <p>User-specified cluster frame origin initial coordinates w.r.t. G</p> <p>User-option to plot cluster-space variable trajectories</p>
B_Four_Robot_Cluster_Simulation_Parameters_NRV_j.m	<p>User-specified initial "actual" values of cluster-space variables (does not have to equal initial value in "A_Four_Robot_Cluster_Trajectory_Generator.m" to allow for differences in "desired" and "actual" cluster-space variables)</p> <p>User-specified cluster translational speed</p> <p>User-specified gains for resolved-rate proportional control law</p> <p>User-specified selector matrices for position/velocity control within resolved-rate proportional control law</p> <p>User-specified sensor and position noise magnitudes</p> <p>User-specified wind model input parameters</p> <p>User-specified navigation reference vector, ${}^G\hat{n}$</p> <p>User-specified scalar error correction gain</p> <p>User-specified isosurface mapping direction</p> <p>User-specified scalar value of isosurface to be mapped</p> <p>User-specified time constant for 1st order UAV model (optional)</p>
C_Four_Robot_Cluster_Inverse_Position_Kinematics.m	<p>Transformation for position variables from cluster-space to "robot-space" for simulation initialization</p> <p>Plots simple representation of cluster and robot frames in 3D</p>
D_Isosurface_Basic_AN.slx	Simulation SIMULINK model

Simulation Support Files

<u>Filename</u>	<u>Description</u>
Gradient_Ascent_3D.m	Compute 3D gradient (file 1 of 2)
gradient_calc_3D.m	Compute 3D gradient (file 2 of 2)
Plume_Moving.m	<p>Simulated scalar field plume</p> <p>User-specified scalar field maximum value, half-strength distance, and shape function</p>

Post-Processing Scripts

<u>Filename</u>	<u>Description</u>
E_Measured_Scalars_and_Cluster_Point_B_Plot_Routine.m	Plots average of sensed scalar field and cluster point B coordinates w.r.t. G
E_Four_Robot_Cluster_Error_Plot_Routine_for_IEEE_Paper.m	Plots cluster-space variable errors and computes cluster formation RMS errors
F_Four_Robot_Cluster_3D_Flight_Path_Plot_Routine.m	Plots flight path of cluster in 3D (origin point for cluster frame and individual robot frames)
F_3D_Plot_Routine_for_Two_Datasets.m	Plots flight paths of clusters in 3D for two trials (origin point for cluster frame and individual robot frames)

Isosurface Mapping Mission Simulation (Simple Plume)

Simulation Execution Files ('Run' files in consecutive order)

<u>Filename</u>	<u>Description</u>
A_Four_Robot_Cluster_Trajectory_Generator_NRV_PZ.m	<p>User-specified initial and final values of cluster-space variables to generate trajectories using cubic spline</p> <p>User-specified total simulation time/duration</p> <p>User-specified simulation time step size</p> <p>User-specified cluster frame origin initial coordinates w.r.t. G</p> <p>User-option to plot cluster-space variable trajectories</p>
B_Four_Robot_Cluster_Simulation_Parameters_NRV_PZ.m	<p>User-specified initial "actual" values of cluster-space variables (does not have to equal initial value in "A_Four_Robot_Cluster_Trajectory_Generator_NRV_PZ.m" to allow for differences in "desired" and "actual" cluster-space variables)</p> <p>User-specified cluster translational speed</p> <p>User-specified gains for resolved-rate proportional control law</p> <p>User-specified selector matrices for position/velocity control within resolved-rate proportional control law</p> <p>User-specified sensor and position noise magnitudes</p> <p>User-specified wind model input parameters</p> <p>User-specified navigation reference vector, ${}^G\hat{n}$</p> <p>User-specified scalar error correction gain and constraint plane positional error correction gain</p> <p>User-specified distance between isosurface map planes</p> <p>User-specified isosurface mapping direction</p> <p>User-specified scalar value of isosurface to be mapped</p> <p>User-specified isosurface mapping tolerance parameters</p> <p>User-specified time constant for 1st order UAV model (optional)</p>
C_Four_Robot_Cluster_Inverse_Position_Kinematics.m	<p>Transformation for position variables from cluster-space to "robot-space" for simulation initialization</p> <p>Plots simple representation of cluster and robot frames in 3D</p>
D_Isosurface_Advanced_AN.slx	Simulation SIMULINK model

Simulation Support Files

<u>Filename</u>	<u>Description</u>
Gradient_Ascent_3D.m	Compute 3D gradient (file 1 of 2)
gradient_calc_3D.m	Compute 3D gradient (file 2 of 2)
Plume_Moving.m	<p>Simulated scalar field plume</p> <p>User-specified scalar field maximum value, half-strength distance, and shape function</p>

Post-Processing Scripts

<u>Filename</u>	<u>Description</u>
E_Measured_Scalars_and_Cluster_Point_B_Plot_Routine.m	Plots average of sensed scalar field and cluster point B coordinates w.r.t. G
E_Four_Robot_Cluster_Error_Plot_Routine_for_IEEE_Paper.m	Plots cluster-space variable errors and computes cluster formation RMS errors
F_Four_Robot_Cluster_3D_Flight_Path_Plot_Routine.m	Plots flight path of cluster in 3D (origin point for cluster frame and individual robot frames)

Isosurface Mapping Mission Simulation (Asymmetric Plume)

Simulation Execution Files ('Run' files in consecutive order)

<u>Filename</u>	<u>Description</u>
A_Four_Robot_Cluster_Trajectory_Generator_NRV_PY.m	<p>User-specified initial and final values of cluster-space variables to generate trajectories using cubic spline</p> <p>User-specified total simulation time/duration</p> <p>User-specified simulation time step size</p> <p>User-specified cluster frame origin initial coordinates w.r.t. G</p> <p>User-option to plot cluster-space variable trajectories</p>
B_Four_Robot_Cluster_Simulation_Parameters_NRV_PY.m	<p>User-specified initial "actual" values of cluster-space variables (does not have to equal initial value in "A_Four_Robot_Cluster_Trajectory_Generator_NRV_PY.m" to allow for differences in "desired" and "actual" cluster-space variables)</p> <p>User-specified cluster translational speed</p> <p>User-specified gains for resolved-rate proportional control law</p> <p>User-specified selector matrices for position/velocity control within resolved-rate proportional control law</p> <p>User-specified sensor and position noise magnitudes</p> <p>User-specified wind model input parameters</p> <p>User-specified navigation reference vector, ${}^G\hat{n}$</p> <p>User-specified scalar error correction gain and constraint plane positional error correction gain</p> <p>User-specified distance between isosurface map planes</p> <p>User-specified isosurface mapping direction</p> <p>User-specified scalar value of isosurface to be mapped</p> <p>User-specified isosurface mapping tolerance parameters</p> <p>User-specified time constant for 1st order UAV model (optional)</p>
C_Four_Robot_Cluster_Inverse_Position_Kinematics.m	<p>Transformation for position variables from cluster-space to "robot-space" for simulation initialization</p> <p>Plots simple representation of cluster and robot frames in 3D</p>
D_Isosurface_Advanced_AN.slx	Simulation SIMULINK model

Simulation Support Files

<u>Filename</u>	<u>Description</u>
Gradient_Ascent_3D.m	Compute 3D gradient (file 1 of 2)
gradient_calc_3D.m	Compute 3D gradient (file 2 of 2)
Plume_Moving.m	<p>Simulated scalar field plume</p> <p>User-specified scalar field maximum value, half-strength distance, and shape function</p>

Post-Processing Scripts

<u>Filename</u>	<u>Description</u>
E_Measured_Scalars_and_Cluster_Point_B_Plot_Routine.m	Plots average of sensed scalar field and cluster point B coordinates w.r.t. G
E_Four_Robot_Cluster_Error_Plot_Routine_for_IEEE_Paper.m	Plots cluster-space variable errors and computes cluster formation RMS errors
F_Four_Robot_Cluster_3D_Flight_Path_Plot_Routine.m	Plots flight path of cluster in 3D (origin point for cluster frame and individual robot frames)

Isosurface Mapping Mission Simulation (Overlapping Plumes)

Simulation Execution Files ('Run' files in consecutive order)

<u>Filename</u>	<u>Description</u>
A_Four_Robot_Cluster_Trajectory_Generator_NRV_Pj.m	<p>User-specified initial and final values of cluster-space variables to generate trajectories using cubic spline</p> <p>User-specified total simulation time/duration</p> <p>User-specified simulation time step size</p> <p>User-specified cluster frame origin initial coordinates w.r.t. G</p> <p>User-option to plot cluster-space variable trajectories</p>
B_Four_Robot_Cluster_Simulation_Parameters_NRV_Pj.m	<p>User-specified initial "actual" values of cluster-space variables (does not have to equal initial value in "A_Four_Robot_Cluster_Trajectory_Generator_NRV_Pj.m" to allow for differences in "desired" and "actual" cluster-space variables)</p> <p>User-specified cluster translational speed</p> <p>User-specified gains for resolved-rate proportional control law</p> <p>User-specified selector matrices for position/velocity control within resolved-rate proportional control law</p> <p>User-specified sensor and position noise magnitudes</p> <p>User-specified wind model input parameters</p> <p>User-specified navigation reference vector, ${}^G\hat{n}$</p> <p>User-specified scalar error correction gain and constraint plane positional error correction gain</p> <p>User-specified distance between isosurface map planes</p> <p>User-specified isosurface mapping direction</p> <p>User-specified scalar value of isosurface to be mapped</p> <p>User-specified isosurface mapping tolerance parameters</p> <p>User-specified time constant for 1st order UAV model (optional)</p>
C_Four_Robot_Cluster_Inverse_Position_Kinematics.m	<p>Transformation for position variables from cluster-space to "robot-space" for simulation initialization</p> <p>Plots simple representation of cluster and robot frames in 3D</p>
D_Isosurface_Advanced_AN.slx	Simulation SIMULINK model

Simulation Support Files

<u>Filename</u>	<u>Description</u>
Gradient_Ascent_3D.m	Compute 3D gradient (file 1 of 2)
gradient_calc_3D.m	Compute 3D gradient (file 2 of 2)
Plume_Moving.m	<p>Simulated scalar field plume</p> <p>User-specified scalar field maximum value, half-strength distance, and shape function</p>

Post-Processing Scripts

<u>Filename</u>	<u>Description</u>
E_Measured_Scalars_and_Cluster_Point_B_Plot_Routine.m	Plots average of sensed scalar field and cluster point B coordinates w.r.t. G
E_Four_Robot_Cluster_Error_Plot_Routine_for_IEEE_Paper.m	Plots cluster-space variable errors and computes cluster formation RMS errors
F_3D_Flight_Path_Plot_Routine_Multiple_Plumes.m	Plots flight path of cluster in 3D (origin point for cluster frame and individual robot frames)

Downstream Plume Following Mission Simulation

Simulation Execution Files ('Run' files in consecutive order)

<u>Filename</u>	<u>Description</u>
A_Nine_Robot_Cluster_Trajectory_Generator_Case_i.m	<p>User-specified initial and final values of cluster-space variables to generate trajectories using cubic spline</p> <p>User-specified total simulation time/duration</p> <p>User-specified simulation time step size</p> <p>User-specified cluster frame origin initial coordinates w.r.t. G</p> <p>User-option to plot cluster-space variable trajectories</p>
B_Nine_Robot_Cluster_Simulation_Parameters_Case_i.m	<p>User-specified initial "actual" values of cluster-space variables (does not have to equal initial value in "A_Nine_Robot_Cluster_Trajectory_Generator_Case_i.m" to allow for differences in "desired" and "actual" cluster-space variables)</p> <p>User-specified cluster translational and rotational speeds</p> <p>User-specified gains for resolved-rate proportional control law</p> <p>User-specified selector matrices for position/velocity control within resolved-rate proportional control law</p> <p>User-specified sensor and position noise magnitudes</p> <p>User-specified wind model input parameters</p> <p>User-specified downstream plume following thresholds</p> <p>User-specified time constant for 1st order UAV model (optional)</p> <p>User-specified plume origin coordinates</p>
C_Nine_Robot_Cluster_Inverse_Position_Kinematics.m	<p>Transformation for position variables from cluster-space to "robot-space" for simulation initialization</p> <p>Plots simple representation of cluster and robot frames in 3D</p>
D_Plume_Following_Outward.slx	Simulation SIMULINK model

Simulation Support Files

<u>Filename</u>	<u>Description</u>
Plume_Moving.m	<p>Simulated scalar field plume</p> <p>User-specified scalar field maximum value, half-strength distance, and shape function</p>

Post-Processing Scripts

<u>Filename</u>	<u>Description</u>
E Scalars_Differentials_Cluster_Variables_for_IEEE_Paper.m	Plots of sensed scalar field, differential signals, and cluster-space variables
E_Nine_Robot_Cluster_Error_Plot_Routine_for_IEEE_Paper.m	Plots cluster-space variable errors and computes cluster formation RMS errors
F_Nine_Robot_Cluster_3D_Flight_Path_Plot_Routine.m	Plots flight path of cluster in 3D (origin point for cluster frame and individual robot frames)

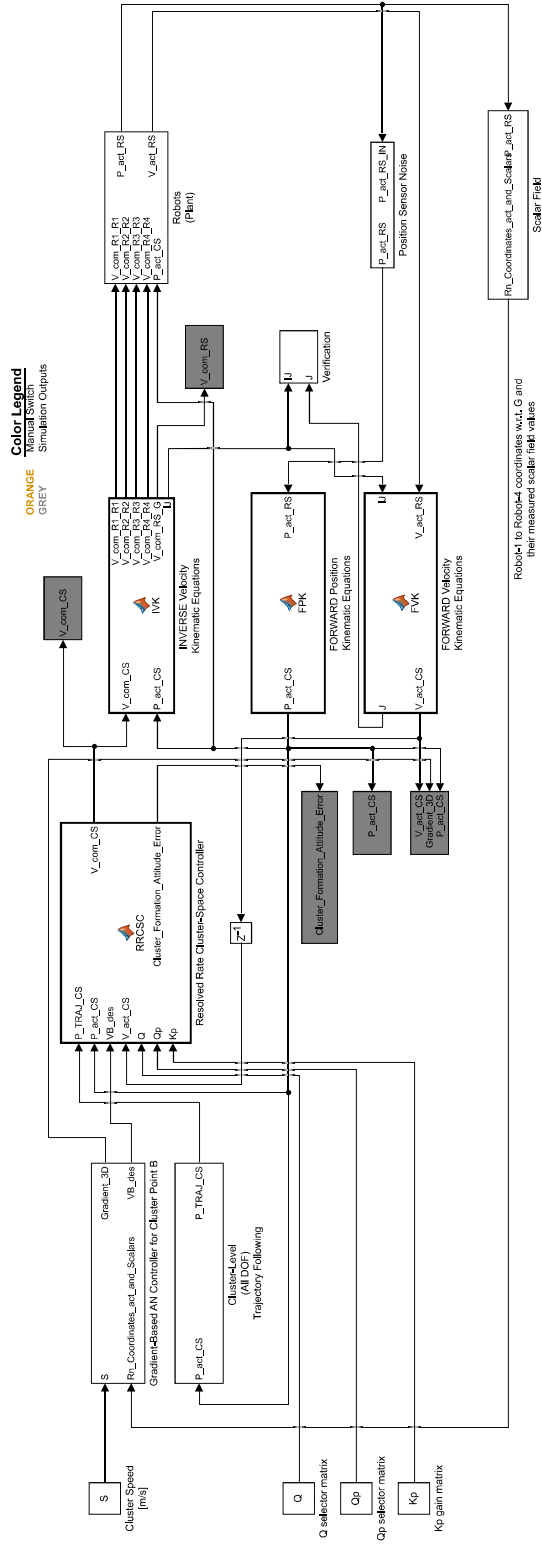
SAMPLE OF MATLAB/SIMULINK MODEL

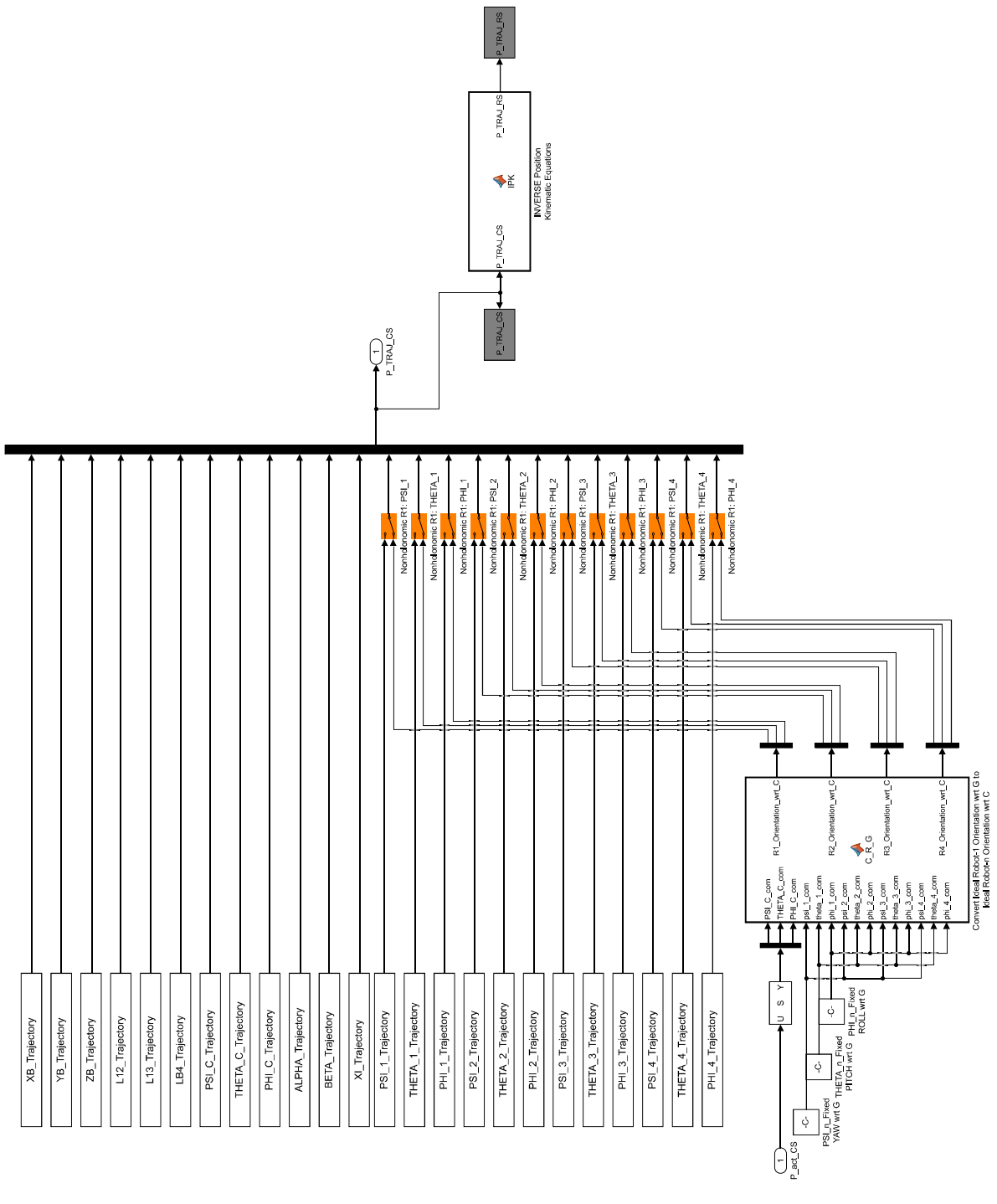
(Stationary Source Seeking in a Static Field Simulation)

(This section contains 79 pages)

Gradient-Based Adaptive Navigation for Source Seeking and Tracking with Four-Robot Cluster

Robert K. Lee
 Santa Clara University
 Department of Mechanical Engineering






```

%=====
%=====  

% Robot-3 Orientation wrt C =====  

% Robot-3 Frame (R3) Orientation expressed in G  

G_R_R3P = ...  

    [ cos(psi_3_com)   -sin(psi_3_com)   0   ; ...  

      sin(psi_3_com)   cos(psi_3_com)   0   ; ...  

      0                 0                 1   ];  

R3P_R_R3PP = ...  

    [ cos(theta_3_com)   0   sin(theta_3_com); ...  

      0                   1   0   ; ...  

      -sin(theta_3_com)  0   cos(theta_3_com)];  

R3PP_R_R3 = ...  

    [ 1   0   0   ; ...  

      0   cos(phi_3_com) -sin(phi_3_com) ; ...  

      0   sin(phi_3_com)  cos(phi_3_com)  ];  

G_R_R3 = G_R_R3P*R3P_R_R3PP*R3PP_R_R3; % [ Xhat_3 Yhat_3 Zhat_3 ]  

% Robot-3 Frame (R3) Orientation expressed in C  

C_R_R3 = transpose(G_R_C)*G_R_R3; % [ C_Xhat_3 C_Yhat_3 C_Zhat_3 ]  

%=====  

PSI_3_com = atan2(C_R_R3(2,1),C_R_R3(1,1)); % R3 Yaw [rad] wrt C  

THETA_3_com = asin(-C_R_R3(3,1)); % R3 Pitch [rad] wrt C  

PHI_3_com = atan2(C_R_R3(3,2),C_R_R3(3,3)); % R3 Roll [rad] wrt C  

R3_Orientation_wrt_C = [ PSI_3_com ; THETA_3_com ; PHI_3_com ];  

%=====  

%=====  

%=====  

% Robot-4 Orientation wrt C =====  

% Robot-4 Frame (R4) Orientation expressed in G  

G_R_R4P = ...  

    [ cos(psi_4_com)   -sin(psi_4_com)   0   ; ...  

      sin(psi_4_com)   cos(psi_4_com)   0   ; ...  

      0                 0                 1   ];  

R4P_R_R4PP = ...  

    [ cos(theta_4_com)   0   sin(theta_4_com); ...  

      0                   1   0   ; ...  

      -sin(theta_4_com)  0   cos(theta_4_com)];  

R4PP_R_R4 = ...  

    [ 1   0   0   ; ...  

      0   cos(phi_4_com) -sin(phi_4_com) ; ...  

      0   sin(phi_4_com)  cos(phi_4_com)  ];  

G_R_R4 = G_R_R4P*R4P_R_R4PP*R4PP_R_R4; % [ Xhat_4 Yhat_4 Zhat_4 ]  

% Robot-4 Frame (R4) Orientation expressed in C  

C_R_R4 = transpose(G_R_C)*G_R_R4; % [ C_Xhat_4 C_Yhat_4 C_Zhat_4 ]  

%=====  

PSI_4_com = atan2(C_R_R4(2,1),C_R_R4(1,1)); % R4 Yaw [rad] wrt C  

THETA_4_com = asin(-C_R_R4(3,1)); % R4 Pitch [rad] wrt C  

PHI_4_com = atan2(C_R_R4(3,2),C_R_R4(3,3)); % R4 Roll [rad] wrt C  

R4_Orientation_wrt_C = [ PSI_4_com ; THETA_4_com ; PHI_4_com ];  

%=====  

%=====  

%=====  


```

```

function P_TRAJ_RS = IPK(P_TRAJ_CS)

% Decompose P_TRAJ_CS vector into Cluster-Space Position Variables:
XB = P_TRAJ_CS(1,1); YB = P_TRAJ_CS(2,1); ZB = P_TRAJ_CS(3,1);
L12 = P_TRAJ_CS(4,1); L13 = P_TRAJ_CS(5,1); LB4 = P_TRAJ_CS(6,1);
PSI_C = P_TRAJ_CS(7,1); THETA_C = P_TRAJ_CS(8,1); PHI_C = P_TRAJ_CS(9,1);
ALPHA = P_TRAJ_CS(10,1); BETA = P_TRAJ_CS(11,1); XI = P_TRAJ_CS(12,1);
PSI_1 = P_TRAJ_CS(13,1); THETA_1 = P_TRAJ_CS(14,1); PHI_1 = P_TRAJ_CS(15,1);
PSI_2 = P_TRAJ_CS(16,1); THETA_2 = P_TRAJ_CS(17,1); PHI_2 = P_TRAJ_CS(18,1);
PSI_3 = P_TRAJ_CS(19,1); THETA_3 = P_TRAJ_CS(20,1); PHI_3 = P_TRAJ_CS(21,1);
PSI_4 = P_TRAJ_CS(22,1); THETA_4 = P_TRAJ_CS(23,1); PHI_4 = P_TRAJ_CS(24,1);

%=====
% Form position vector with given Point B coordinates expressed in G
P_OB = [ XB ; YB ; ZB ];

% Scalar Distance: R2 to R3
L23 = sqrt(L12^2 + L13^2 - 2*L12*L13*cos(BETA));

% Scalar Distance: R1 to D
L1D = 0.5*sqrt(2*L12^2 + 2*L13^2 - L23^2);

% Scalar Distance: B to R1
LB1 = 2/3*L1D;

% Scalar Distance: R2 to E
L2E = 0.5*sqrt(2*L12^2 + 2*L23^2 - L13^2);

% Scalar Distance: B to R2
LB2 = 2/3*L2E;

% Scalar Distance: R3 to F
L3F = 0.5*sqrt(2*L13^2 + 2*L23^2 - L12^2);

% Scalar Distance: B to R3
LB3 = 2/3*L3F;

% Scalar Angle: between "LB1" to "LB2"
eta2 = acos((LB1^2 + LB2^2 - L12^2)/(2*LB1*LB2));

% Scalar Angle: between "LB1" to "LB3"
eta3 = -acos((LB1^2 + LB3^2 - L13^2)/(2*LB1*LB3));
%=====
% Cluster Frame (C) Orientation expressed in G
G_R_P = [ cos(PSI_C)   -sin(PSI_C)   0   ;...
          sin(PSI_C)   cos(PSI_C)   0   ;...
          0             0             1   ];
P_R_DP = [ cos(THETA_C)  0   sin(THETA_C);...
          0             1   0           ;...
          -sin(THETA_C)  0   cos(THETA_C)];
DP_R_C = [ 1             0             0   ;...
          0             cos(PHI_C)  -sin(PHI_C);...
          0             sin(PHI_C)  cos(PHI_C) ];
G_R_C = G_R_P*P_R_DP*DP_R_C; % [ Xhat_C Yhat_C Zhat_C ]
%=====
% Position Vector of Robot-1
C_P_B1 = [ LB1 ; 0 ; 0 ]; % [ C_x1 ; C_y1 ; C_z1 ]
P_O1 = P_OB + G_R_C*C_P_B1; % [ x1 ; y1 ; z1 ]
x1 = P_O1(1,1); y1 = P_O1(2,1); z1 = P_O1(3,1);

% Position Vector of Robot-2
C_P_B2 = [ LB2*cos(eta2) ; LB2*sin(eta2) ; 0 ]; % [ C_x2 ; C_y2 ; C_z2 ]
P_O2 = P_OB + G_R_C*C_P_B2; % [ x2 ; y2 ; z2 ]
x2 = P_O2(1,1); y2 = P_O2(2,1); z2 = P_O2(3,1);

% Position Vector of Robot-3
C_P_B3 = [ LB3*cos(eta3) ; LB3*sin(eta3) ; 0 ]; % [ C_x3 ; C_y3 ; C_z3 ]
P_O3 = P_OB + G_R_C*C_P_B3; % [ x3 ; y3 ; z3 ]
x3 = P_O3(1,1); y3 = P_O3(2,1); z3 = P_O3(3,1);

% Position Vector of Robot-4
C_P_B4 = [ LB4*sin(XI)*cos(ALPHA) ; ... % [ C_x4 ; C_y4 ; C_z4 ]
          LB4*sin(XI)*sin(ALPHA) ; LB4*cos(XI) ];
P_O4 = P_OB + G_R_C*C_P_B4; % [ x4 ; y4 ; z4 ]
x4 = P_O4(1,1); y4 = P_O4(2,1); z4 = P_O4(3,1);
%=====

```



```

%=====  

%=====  

% Robot-1 Orientation =====  

% Robot-1 Frame (R1) Orientation expressed in C  

C_r_R1P = ...  

[ cos(Psi_1)   -sin(Psi_1)   0   ;...  

  sin(Psi_1)   cos(Psi_1)   0   ;...  

  0             0             1   ];  

R1P_r_R1PP = ...  

[ cos(THETA_1)   0   sin(THETA_1);...  

  0               1   0           ;...  

 -sin(THETA_1)   0   cos(THETA_1)];  

R1PP_r_R1 = ...  

[ 1   0   0   ;...  

  0   cos(PHI_1) -sin(PHI_1) ;...  

  0   sin(PHI_1)  cos(PHI_1)  ];  

C_r_R1 = C_r_R1P*R1P_r_R1PP*R1PP_r_R1; % [ C_Xhat_1 C_Yhat_1 C_Zhat_1 ]  

% Robot-1 Frame (R1) Orientation expressed in G  

G_r_R1 = G_R_C*C_r_R1; % [ Xhat_1 Yhat_1 Zhat_1 ]  

%=====  

psi_1 = atan2(G_r_R1(2,1),G_r_R1(1,1)); % R1 Yaw [rad] wrt G  

theta_1 = asin(-G_r_R1(3,1)); % R1 Pitch [rad] wrt G  

phi_1 = atan2(G_r_R1(3,2),G_r_R1(3,3)); % R1 Roll [rad] wrt G  

%=====  

%=====  

%=====  

% Robot-2 Orientation =====  

% Robot-2 Frame (R2) Orientation expressed in C  

C_r_R2P = ...  

[ cos(Psi_2)   -sin(Psi_2)   0   ;...  

  sin(Psi_2)   cos(Psi_2)   0   ;...  

  0             0             1   ];  

R2P_r_R2PP = ...  

[ cos(THETA_2)   0   sin(THETA_2);...  

  0               1   0           ;...  

 -sin(THETA_2)   0   cos(THETA_2)];  

R2PP_r_R2 = ...  

[ 1   0   0   ;...  

  0   cos(PHI_2) -sin(PHI_2) ;...  

  0   sin(PHI_2)  cos(PHI_2)  ];  

C_r_R2 = C_r_R2P*R2P_r_R2PP*R2PP_r_R2; % [ C_Xhat_2 C_Yhat_2 C_Zhat_2 ]  

% Robot-2 Frame (R2) Orientation expressed in G  

G_r_R2 = G_R_C*C_r_R2; % [ Xhat_2 Yhat_2 Zhat_2 ]  

%=====  

psi_2 = atan2(G_r_R2(2,1),G_r_R2(1,1)); % R2 Yaw [rad] wrt G  

theta_2 = asin(-G_r_R2(3,1)); % R2 Pitch [rad] wrt G  

phi_2 = atan2(G_r_R2(3,2),G_r_R2(3,3)); % R2 Roll [rad] wrt G  

%=====  

%=====  

%=====  

% Robot-3 Orientation =====  

% Robot-3 Frame (R3) Orientation expressed in C  

C_r_R3P = ...  

[ cos(Psi_3)   -sin(Psi_3)   0   ;...  

  sin(Psi_3)   cos(Psi_3)   0   ;...  

  0             0             1   ];  

R3P_r_R3PP = ...  

[ cos(THETA_3)   0   sin(THETA_3);...  

  0               1   0           ;...  

 -sin(THETA_3)   0   cos(THETA_3)];  

R3PP_r_R3 = ...  

[ 1   0   0   ;...  

  0   cos(PHI_3) -sin(PHI_3) ;...  

  0   sin(PHI_3)  cos(PHI_3)  ];  

C_r_R3 = C_r_R3P*R3P_r_R3PP*R3PP_r_R3; % [ C_Xhat_3 C_Yhat_3 C_Zhat_3 ]  

% Robot-3 Frame (R3) Orientation expressed in G  

G_r_R3 = G_R_C*C_r_R3; % [ Xhat_3 Yhat_3 Zhat_3 ]  

%=====  

psi_3 = atan2(G_r_R3(2,1),G_r_R3(1,1)); % R3 Yaw [rad] wrt G  

theta_3 = asin(-G_r_R3(3,1)); % R3 Pitch [rad] wrt G  

phi_3 = atan2(G_r_R3(3,2),G_r_R3(3,3)); % R3 Roll [rad] wrt G  

%=====  


```

```

%===== Robot-3 Orientation =====
%

%===== Robot-4 Orientation =====
% Robot-4 Frame (R4) Orientation expressed in C
C_r_R4P = ...
    [ cos(Psi_4)   -sin(Psi_4)   0   ;...
      sin(Psi_4)   cos(Psi_4)   0   ;...
      0             0             1   ];

R4P_r_R4PP = ...
    [ cos(THETA_4)   0   sin(THETA_4);...
      0               1   0           ;...
      -sin(THETA_4)  0   cos(THETA_4)];

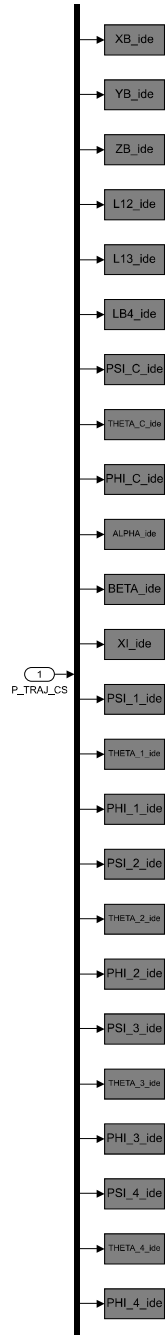
R4PP_r_R4 = ...
    [ 1             0             0   ;...
      0             cos(PHI_4)  -sin(PHI_4);...
      0             sin(PHI_4)  cos(PHI_4)  ];

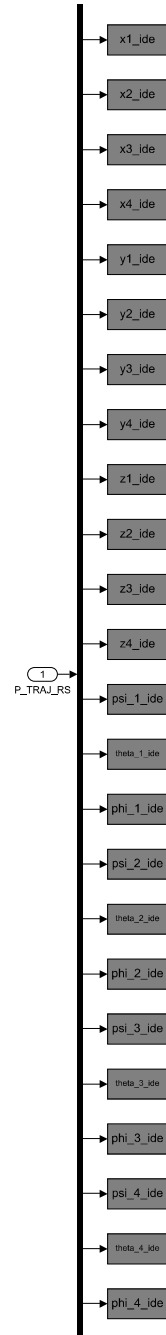
C_r_R4 = C_r_R4P*R4P_r_R4PP*R4PP_r_R4; % [ C_Xhat_4 C_Yhat_4 C_Zhat_4 ]

% Robot-4 Frame (R4) Orientation expressed in G
G_r_R4 = G_R_C*C_r_R4; % [ Xhat_4 Yhat_4 Zhat_4 ]
%=====
psi_4 = atan2(G_r_R4(2,1),G_r_R4(1,1)); % R4 Yaw [rad] wrt G
theta_4 = asin(-G_r_R4(3,1)); % R4 Pitch [rad] wrt G
phi_4 = atan2(G_r_R4(3,2),G_r_R4(3,3)); % R4 Roll [rad] wrt G
%===== Robot-4 Orientation =====
%

P_TRAJ_RS = [ x1 ; x2 ; x3 ; x4 ;...
              y1 ; y2 ; y3 ; y4 ;...
              z1 ; z2 ; z3 ; z4 ;...
              psi_1 ; theta_1 ; phi_1 ;...
              psi_2 ; theta_2 ; phi_2 ;...
              psi_3 ; theta_3 ; phi_3 ;...
              psi_4 ; theta_4 ; phi_4 ];

```





```

function P_act_CS = FPK(P_act_RS)

%Note: 'FORWARD Position Kinematic Equations' block contains "XI Singularity filter"

% Decompose P_act_RS vector in Actual Robot-Space Position Variables:
x1 = P_act_RS(1,1); x2 = P_act_RS(2,1); x3 = P_act_RS(3,1); x4 = P_act_RS(4,1);
y1 = P_act_RS(5,1); y2 = P_act_RS(6,1); y3 = P_act_RS(7,1); y4 = P_act_RS(8,1);
z1 = P_act_RS(9,1); z2 = P_act_RS(10,1); z3 = P_act_RS(11,1); z4 = P_act_RS(12,1);
psi_1 = P_act_RS(13,1); theta_1 = P_act_RS(14,1); phi_1 = P_act_RS(15,1);
psi_2 = P_act_RS(16,1); theta_2 = P_act_RS(17,1); phi_2 = P_act_RS(18,1);
psi_3 = P_act_RS(19,1); theta_3 = P_act_RS(20,1); phi_3 = P_act_RS(21,1);
psi_4 = P_act_RS(22,1); theta_4 = P_act_RS(23,1); phi_4 = P_act_RS(24,1);

%=====  

% Coordinates: Point B expressed in G  

XB = (x1 + x2 + x3)/3; YB = (y1 + y2 + y3)/3; ZB = (z1 + z2 + z3)/3;

% Position Vector: Point B to R1 expressed in G  

L_B1 = [ (x1 - XB) ; (y1 - YB) ; (z1 - ZB) ];  

% Scalar Distance: Point B to R1  

LB1 = norm(L_B1);

% Position Vector: Point R1 to R2 expressed in G  

L_12 = [ (x2 - x1) ; (y2 - y1) ; (z2 - z1) ];  

% Scalar Distance: Point R1 to R2  

L12 = norm(L_12);

% Position Vector: Point R1 to R3 expressed in G  

L_13 = [ (x3 - x1) ; (y3 - y1) ; (z3 - z1) ];  

% Scalar Distance: Point R1 to R3  

L13 = norm(L_13);

% Position Vector: Point B to R4 expressed in G  

L_B4 = [ (x4 - XB) ; (y4 - YB) ; (z4 - ZB) ];  

% Scalar Distance: Point B to R4  

LB4 = norm(L_B4);

%=====  

% Normal Vector: Normal to "Cluster Base Triangle" expressed in G  

n = cross(L_12,L_13);
%=====  

% Cluster Frame (C) Axes expressed in G  

Xhat_C = L_B1/norm(L_B1); % aka: G_Xhat_C  

Zhat_C = n/norm(n); % aka: G_Zhat_C  

Yhat_C = cross(Zhat_C,Xhat_C); % aka: G_Yhat_C
%=====  

PSI_C = atan2(Xhat_C(2,1),Xhat_C(1,1)); % Cluster Yaw [rad]  

THETA_C = asin(-Xhat_C(3,1)); % Cluster Pitch [rad]  

PHI_C = atan2(Yhat_C(3,1),Zhat_C(3,1)); % Cluster Roll [rad]
%=====  

% Cluster Frame (C) Orientation expressed in G  

G_R_P = [ cos(PSI_C) -sin(PSI_C) 0 ; ...  

sin(PSI_C) cos(PSI_C) 0 ; ...  

0 0 1 ];  

P_R_DP = [ cos(THETA_C) 0 sin(THETA_C); ...  

0 1 0 ; ...  

-sin(THETA_C) 0 cos(THETA_C) ];  

DP_R_C = [ 1 0 0 ; ...  

0 cos(PHI_C) -sin(PHI_C) ; ...  

0 sin(PHI_C) cos(PHI_C) ];  

G_R_C = G_R_P*P_R_DP*DP_R_C; % [ Xhat_C Yhat_C Zhat_C ]

% Position Vector: Point B to R4 expressed in C  

C_L_B4 = transpose(G_R_C)*L_B4;

% Logic to Filter Out Noise from ALPHA and XI Computation  

if abs(C_L_B4(1,1))<1e-5  

C_L_B4(1,1) = 0;  

end  

if abs(C_L_B4(2,1))<1e-5  

C_L_B4(2,1) = 0;  

end  

if abs(C_L_B4(3,1))<1e-5  

C_L_B4(3,1) = 0;  

end
%=====  

% Cluster Frame (C) Internal Angles  

ALPHA = atan2(C_L_B4(2,1),C_L_B4(1,1)); % Alpha [rad]

```

```

BETA = acos(dot(L_12,L_13)/(L12*L13)); % Beta [rad]
XI = ... % Zeta [rad]
atan2(sqrt((C_L_B4(1,1))^2+(C_L_B4(2,1))^2),C_L_B4(3,1));
%=====

%===== Robot-1 Orientation =====
% Robot-1 Frame (R1) Orientation expressed in G
G_R_R1P = ...
[ cos(psi_1) -sin(psi_1) 0 ;...
sin(psi_1) cos(psi_1) 0 ;...
0 0 1 ];

R1P_R_R1PP = ...
[ cos(theta_1) 0 sin(theta_1);...
0 1 0 ;...
-sin(theta_1) 0 cos(theta_1)];

R1PP_R_R1 = ...
[ 1 0 0 ;...
0 cos(phi_1) -sin(phi_1) ;...
0 sin(phi_1) cos(phi_1) 1];

G_R_R1 = G_R_R1P*R1P_R_R1PP*R1PP_R_R1; % [ Xhat_1 Yhat_1 Zhat_1 ]

% Robot-1 Frame (R1) Orientation expressed in C
C_R_R1 = transpose(G_R_C)*G_R_R1; % [ C_Xhat_1 C_Yhat_1 C_Zhat_1 ]
%=====

PSI_1 = atan2(C_R_R1(2,1),C_R_R1(1,1)); % R1 Yaw [rad] wrt C
THETA_1 = asin(-C_R_R1(3,1)); % R1 Pitch [rad] wrt C
PHI_1 = atan2(C_R_R1(3,2),C_R_R1(3,3)); % R1 Roll [rad] wrt C
%===== Robot-1 Orientation =====
%=====

%===== Robot-2 Orientation =====
% Robot-2 Frame (R2) Orientation expressed in G
G_R_R2P = ...
[ cos(psi_2) -sin(psi_2) 0 ;...
sin(psi_2) cos(psi_2) 0 ;...
0 0 1 ];

R2P_R_R2PP = ...
[ cos(theta_2) 0 sin(theta_2);...
0 1 0 ;...
-sin(theta_2) 0 cos(theta_2)];

R2PP_R_R2 = ...
[ 1 0 0 ;...
0 cos(phi_2) -sin(phi_2) ;...
0 sin(phi_2) cos(phi_2) 1];

G_R_R2 = G_R_R2P*R2P_R_R2PP*R2PP_R_R2; % [ Xhat_2 Yhat_2 Zhat_2 ]

% Robot-2 Frame (R2) Orientation expressed in C
C_R_R2 = transpose(G_R_C)*G_R_R2; % [ C_Xhat_2 C_Yhat_2 C_Zhat_2 ]
%=====

PSI_2 = atan2(C_R_R2(2,1),C_R_R2(1,1)); % R2 Yaw [rad] wrt C
THETA_2 = asin(-C_R_R2(3,1)); % R2 Pitch [rad] wrt C
PHI_2 = atan2(C_R_R2(3,2),C_R_R2(3,3)); % R2 Roll [rad] wrt C
%===== Robot-2 Orientation =====
%=====

%===== Robot-3 Orientation =====
% Robot-3 Frame (R3) Orientation expressed in G
G_R_R3P = ...
[ cos(psi_3) -sin(psi_3) 0 ;...
sin(psi_3) cos(psi_3) 0 ;...
0 0 1 ];

R3P_R_R3PP = ...
[ cos(theta_3) 0 sin(theta_3);...
0 1 0 ;...
-sin(theta_3) 0 cos(theta_3)];

R3PP_R_R3 = ...
[ 1 0 0 ;...
0 cos(phi_3) -sin(phi_3) ;...
0 sin(phi_3) cos(phi_3) 1];

G_R_R3 = G_R_R3P*R3P_R_R3PP*R3PP_R_R3; % [ Xhat_3 Yhat_3 Zhat_3 ]

% Robot-3 Frame (R3) Orientation expressed in C
C_R_R3 = transpose(G_R_C)*G_R_R3; % [ C_Xhat_3 C_Yhat_3 C_Zhat_3 ]

```

```

%=====
PSI_3 = atan2(C_R_R3(2,1),C_R_R3(1,1)); % R3 Yaw [rad] wrt C
THETA_3 = asin(-C_R_R3(3,1)); % R3 Pitch [rad] wrt C
PHI_3 = atan2(C_R_R3(3,2),C_R_R3(3,3)); % R3 Roll [rad] wrt C
%===== Robot-3 Orientation =====
%=====

%===== Robot-4 Orientation =====
% Robot-4 Frame (R4) Orientation expressed in G
G_R_R4P = ...
    [ cos(psi_4)  -sin(psi_4)  0  ;...
      sin(psi_4)  cos(psi_4)  0  ;...
      0           0          1  ];

R4P_R_R4PP = ...
    [ cos(theta_4)  0  sin(theta_4);...
      0             1  0           ;...
      -sin(theta_4) 0  cos(theta_4)];

R4PP_R_R4 = ...
    [ 1  0  0  ;...
      0  cos(phi_4) -sin(phi_4) ;...
      0  sin(phi_4)  cos(phi_4)  ];

G_R_R4 = G_R_R4P*R4P_R_R4PP*R4PP_R_R4; % [ Xhat_4 Yhat_4 Zhat_4 ]

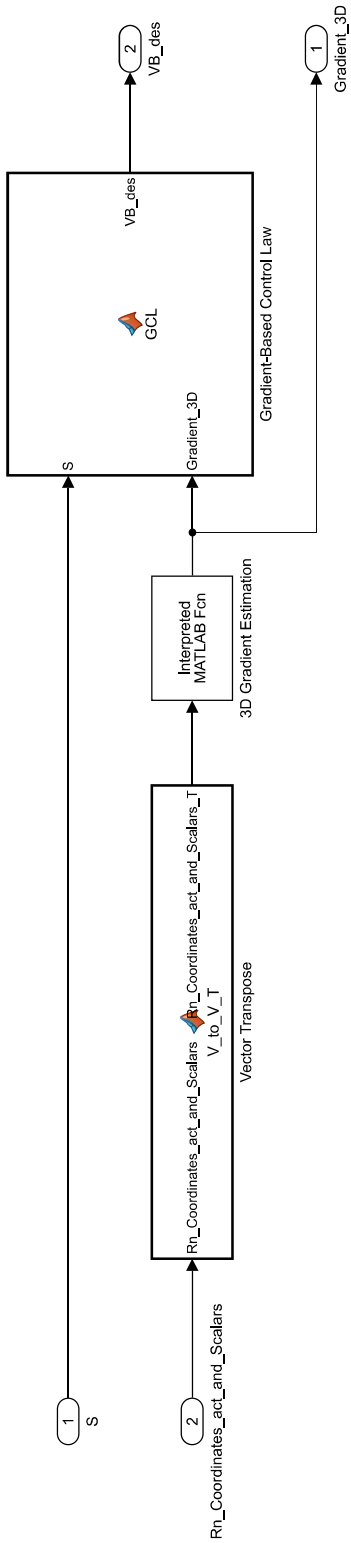
% Robot-4 Frame (R4) Orientation expressed in C
C_R_R4 = transpose(G_R_C)*G_R_R4; % [ C_Xhat_4 C_Yhat_4 C_Zhat_4 ]
%=====
PSI_4 = atan2(C_R_R4(2,1),C_R_R4(1,1)); % R4 Yaw [rad] wrt C
THETA_4 = asin(-C_R_R4(3,1)); % R4 Pitch [rad] wrt C
PHI_4 = atan2(C_R_R4(3,2),C_R_R4(3,3)); % R4 Roll [rad] wrt C
%===== Robot-4 Orientation =====
%=====

%===== XI Singularity Filter =====
if XI<0.1*pi()/180
    XI = 0.1*pi()/180;
end
%===== XI Singularity Filter =====
%=====

P_act_CS = [ XB ; YB ; ZB ; ...
             L12 ; L13 ; LB4 ; ...
             PSI_C ; THETA_C ; PHI_C ; ...
             ALPHA ; BETA ; XI ; ...
             PSI_1 ; THETA_1 ; PHI_1 ; ...
             PSI_2 ; THETA_2 ; PHI_2 ; ...
             PSI_3 ; THETA_3 ; PHI_3 ; ...
             PSI_4 ; THETA_4 ; PHI_4 ];

```

```
function [J,V_act_CS] = FVK(IJ,V_act_RS)
J = inv(IJ);
% Compute Actual Cluster-Space Velocities
V_act_CS = J*V_act_RS;
```

```
function VB_des = GCL(S, Gradient_3D)

% Parse out 3D Gradient Vector
g_x = Gradient_3D(1); g_y = Gradient_3D(2); g_z = Gradient_3D(3);

% Form Gradient Vector
g_gradient = [ g_x ; g_y ; g_z ];

% Computed 3D Gradient Vector Magnitude
g_gradient_norm = norm(g_gradient);

% Form Cluster DESIRED Translational Velocity Vector
VB_des = S*(g_gradient/g_gradient_norm);
```

```
function Rn_Coordinates_act_and Scalars_T = V_to_V_T(Rn_Coordinates_act_and Scalars)
Rn_Coordinates_act_and Scalars_T = transpose(Rn_Coordinates_act_and Scalars);
```

```

function [V_com_R1_R1,V_com_R2_R2,V_com_R3_R3,V_com_R4_R4,...
        V_com_RS_G,IJ_] = IVK(V_com_CS,P_act_CS)

% Decompose P_act_CS vector in Actual Cluster-Space Position Variables:
XB = P_act_CS(1,1); YB = P_act_CS(2,1); ZB = P_act_CS(3,1);
L12 = P_act_CS(4,1); L13 = P_act_CS(5,1); LB4 = P_act_CS(6,1);
PSI_C = P_act_CS(7,1); THETA_C = P_act_CS(8,1); PHI_C = P_act_CS(9,1);
ALPHA = P_act_CS(10,1); BETA = P_act_CS(11,1); XI = P_act_CS(12,1);
PSI_1 = P_act_CS(13,1); THETA_1 = P_act_CS(14,1); PHI_1 = P_act_CS(15,1);
PSI_2 = P_act_CS(16,1); THETA_2 = P_act_CS(17,1); PHI_2 = P_act_CS(18,1);
PSI_3 = P_act_CS(19,1); THETA_3 = P_act_CS(20,1); PHI_3 = P_act_CS(21,1);
PSI_4 = P_act_CS(22,1); THETA_4 = P_act_CS(23,1); PHI_4 = P_act_CS(24,1);

% Jacobian Matrix Elements
IJ_1_1 = 1;
IJ_1_2 = 0;
IJ_1_3 = 0;
IJ_1_4 = 0.333333E0.*(L12+L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)) ...
    .^(-1/2).*cos(PSI_C).*cos(THETA_C);
IJ_1_5 = 0.333333E0.*(L13+L12.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)) ...
    .^(-1/2).*cos(PSI_C).*cos(THETA_C);
IJ_1_6 = 0;
IJ_1_7 = (-0.333333E0).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(1/2).*cos( ...
    THETA_C).*sin(PSI_C);
IJ_1_8 = (-0.333333E0).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(1/2).*cos(PSI_C) ...
    .*sin(THETA_C);
IJ_1_9 = 0;
IJ_1_10 = 0;
IJ_1_11 = (-0.333333E0).*L12.*L13.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1/2) ...
    .*cos(PSI_C).*cos(THETA_C).*sin(BETA);
IJ_1_12 = 0;
IJ_1_13 = 0;
IJ_1_14 = 0;
IJ_1_15 = 0;
IJ_1_16 = 0;
IJ_1_17 = 0;
IJ_1_18 = 0;
IJ_1_19 = 0;
IJ_1_20 = 0;
IJ_1_21 = 0;
IJ_1_22 = 0;
IJ_1_23 = 0;
IJ_1_24 = 0;

IJ_2_1 = 1;
IJ_2_2 = 0;
IJ_2_3 = 0;
IJ_2_4 = (-1/2).*(2.*L12+2.*L13.*cos(BETA)).*((-0.666667E0).*L12.^2+0.333333E0.* ...
    L13.^2+(-0.333333E0).*L12.*L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.* ...
    cos(BETA)).^(-3/2).*cos(PSI_C).*cos(THETA_C)+((-0.133333E1).*L12+( ...
    -0.333333E0).*L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^( ...
    -1/2).*cos(PSI_C).*cos(THETA_C)+(-0.166667E0).*(4.*L12.^2+L13.^2+(-4).* ...
    L12.*L13.*cos(BETA)).^(1/2).*(2.*L12+2.*L13.*cos(BETA)).*(4.*L12.^2+ ...
    L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+ ...
    1.E0).*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^( ...
    -2)+(-2).*(0.4E1.*L12+1.E0).*L13.*cos(BETA)).*(4.*L12.^2+L13.^2+(-4).* ...
    L12.*L13.*cos(BETA)).^(-1).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0).*L12.* ...
    L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)+(8.*L12+( ...
    -4).*L13.*cos(BETA)).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-2) ...
    .* (0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0).*L12.*L13.*cos(BETA)).^2.*(L12.^2+ ...
    L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).*(1+(-1).*(4.*L12.^2+L13.^2+(-4).* ...
    L12.*L13.*cos(BETA)).^(-1)).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0).*L12.* ...
    L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^( ...
    -1/2).*(cos(PHI_C).*sin(PSI_C)+(-0.1E1).*cos(PSI_C)).*sin(PHI_C).*sin( ...
    THETA_C))+(-0.166667E0).*(8.*L12+(-4).*L13.*cos(BETA)).*(4.*L12.^2+ ...
    L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1/2).*(1+(-1).*(4.*L12.^2+L13.^2+( ...
    -4).*L12.*L13.*cos(BETA)).^(-1)).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.* ...
    L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)) ...
    .^(1/2).*(cos(PHI_C).*sin(PSI_C)+(-0.1E1).*cos(PSI_C)).*sin(PHI_C).*sin( ...
    THETA_C));
IJ_2_5 = (-1/2).*(2.*L13+2.*L12.*cos(BETA)).*((-0.666667E0).*L12.^2+0.333333E0.* ...
    L13.^2+(-0.333333E0).*L12.*L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.* ...
    cos(BETA)).^(-3/2).*cos(PSI_C).*cos(THETA_C)+(0.666667E0).*L13+( ...
    -0.333333E0).*L12.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^( ...
    -1/2).*cos(PSI_C).*cos(THETA_C)+(-0.166667E0).*(4.*L12.^2+L13.^2+(-4).* ...
    L12.*L13.*cos(BETA)).^(1/2).*(2.*L13+2.*L12.*cos(BETA)).*(4.*L12.^2+ ...
    L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+ ...

```

```

1.E0.*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^( ...
-2)+(-2).*((-0.2E1).*L13+1.E0.*L12.*cos(BETA)).*(4.*L12.^2+L13.^2+(-4).* ...
L12.*L13.*cos(BETA)).^(-1).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.* ...
L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)+(2.*L13+( ...
-4).*L12.*cos(BETA)).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-2) ...
.*(0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*(L12.^2+ ...
L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).*(1+(-1).*(4.*L12.^2+L13.^2+(-4).* ...
L12.*L13.*cos(BETA)).^(-1).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.* ...
L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^( ...
-1/2)).*(cos(PHI_C).*sin(PHI_C)+(-0.1E1).*cos(PHI_C).*sin(PHI_C).*sin( ...
THETA_C))+(-0.166667E0).*(2.*L13+(-4).*L12.*cos(BETA)).*(4.*L12.^2+ ...
L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1/2).*(1+(-1).*(4.*L12.^2+L13.^2+( ...
-4).*L12.*L13.*cos(BETA)).^(-1).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.* ...
L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1) ...
.^(1/2)).*(cos(PHI_C).*sin(PHI_C)+(-0.1E1).*cos(PHI_C).*sin(PHI_C).*sin( ...
THETA_C));
IJ_2_6 = 0;
IJ_2_7 = (-1).*((-0.666667E0).*L12.^2+0.333333E0.*L13.^2+(-0.333333E0).*L12.* ...
L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1/2).*cos( ...
THETA_C).*sin(PHI_C)+(-0.333333E0).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.* ...
cos(BETA)).^(1/2).*(1+(-1).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)) ...
.^(-1).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*( ...
L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^(1/2)).*(cos(PHI_C).*cos( ...
PSI_C)+0.1E1.*sin(PHI_C).*sin(PHI_C).*sin(THETA_C));
IJ_2_8 = 0.333333E0.*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(1/2).*(1+(-1) ...
.^(-1).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*(0.2E1.*L12.^2+( ...
-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.* ...
L13.*cos(BETA)).^(-1)).^(1/2).*cos(PSI_C).*cos(THETA_C).*sin(PHI_C)+(-1) ...
.*(0.666667E0).*L12.^2+0.333333E0.*L13.^2+(-0.333333E0).*L12.*L13.* ...
cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1/2).*cos(PSI_C).* ...
sin(THETA_C);
IJ_2_9 = (-0.333333E0).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(1/2).*(1+( ...
-1).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*(0.2E1.*L12.^2+ ...
(-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.* ...
L13.*cos(BETA)).^(-1)).^(1/2).*((-1).*sin(PHI_C).*sin(PSI_C)+(-0.1E1).* ...
cos(PHI_C).*cos(PSI_C).*sin(THETA_C));
IJ_2_10 = 0;
IJ_2_11 = L12.*L13.*((-0.666667E0).*L12.^2+0.333333E0.*L13.^2+(-0.333333E0).*L12.* ...
L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-3/2).*cos( ...
PSI_C).*cos(THETA_C).*sin(BETA)+0.333333E0.*L12.*L13.*(L12.^2+L13.^2+2.* ...
L12.*L13.*cos(BETA)).^(-1/2).*cos(PSI_C).*cos(THETA_C).*sin(BETA)+ ...
-0.666667E0).*L12.*L13.*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1) ...
-1/2)).*(1+(-1).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*( ...
0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*(L12.^2+ ...
L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^(1/2).*sin(BETA).*(cos(PHI_C).* ...
sin(PSI_C)+(-0.1E1).*cos(PSI_C).*sin(PHI_C).*sin(THETA_C))+(-0.166667E0) ...
.*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(1/2).*(1+(-1).*(4.* ...
L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*(0.2E1.*L12.^2+(-1.E0).* ...
L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos( ...
BETA)).^(-1)).^(1/2).*((-1).*L12.*L13.*(4.*L12.^2+L13.^2+(-4).*L12.* ...
L13.*cos(BETA)).^(-1).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.* ...
cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).*( ...
0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).*(L12.^2+ ...
L13.^2+2.*L12.*L13.*cos(BETA)).^(-1).*sin(BETA)+4.*L12.*L13.*(4.*L12.^2+ ...
L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-2).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+ ...
1.E0.*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^( ...
-1).*sin(BETA)).*(cos(PHI_C).*sin(PSI_C)+(-0.1E1).*cos(PSI_C).*sin( ...
PHI_C).*sin(THETA_C));
IJ_2_12 = 0;
IJ_2_13 = 0;
IJ_2_14 = 0;
IJ_2_15 = 0;
IJ_2_16 = 0;
IJ_2_17 = 0;
IJ_2_18 = 0;
IJ_2_19 = 0;
IJ_2_20 = 0;
IJ_2_21 = 0;
IJ_2_22 = 0;
IJ_2_23 = 0;
IJ_2_24 = 0;
IJ_3_1 = 1;
IJ_3_2 = 0;
IJ_3_3 = 0;
IJ_3_4 = (-1/2).*(2.*L12+2.*L13.*cos(BETA)).*(0.333333E0.*L12.^2+(-0.666667E0).* ...

```

```

L13.^2+(-0.333333E0)*L12.*L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.* ...
cos(BETA)).^(-3/2).*cos(PSI_C).*cos(THETA_C)+(0.666667E0)*L12+( ...
-0.333333E0)*L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^( ...
-1/2).*cos(PSI_C).*cos(THETA_C)+0.166667E0.*(L12.^2+4.*L13.^2+(-4)* ...
L12.*L13.*cos(BETA)).^(1/2).*((2.*L12+2.*L13.*cos(BETA)).*(L12.^2+4.* ...
L13.^2+(-4)*L12.*L13.*cos(BETA)).^(-1)*(0.1E1.*L12.^2+(-0.2E1)* ...
L13.^2+(-0.1E1)*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.* ...
cos(BETA)).^(-2)+(-2)*(0.2E1.*L12+(-0.1E1)*L13.*cos(BETA)).*(L12.^2+ ...
4.*L13.^2+(-4)*L12.*L13.*cos(BETA)).^(-1)*(0.1E1.*L12.^2+(-0.2E1)* ...
L13.^2+(-0.1E1)*L12.*L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos( ...
BETA)).^(-1)+(2.*L12+(-4)*L13.*cos(BETA)).*(L12.^2+4.*L13.^2+(-4)* ...
L12.*L13.*cos(BETA)).^(-2)*(0.1E1.*L12.^2+(-0.2E1)*L13.^2+(-0.1E1)* ...
L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1) ...
(1+(-1)).*(L12.^2+4.*L13.^2+(-4)*L12.*L13.*cos(BETA)).^(-1)*(0.1E1.* ...
L12.^2+(-0.2E1)*L13.^2+(-0.1E1)*L12.*L13.*cos(BETA)).^2.*(L12.^2+ ...
L13.^2+(-4)*L12.*L13.*cos(BETA)).^(-1)).^(1/2)).*(cos(PHI_C)*sin(PSI_C)+( ...
-0.1E1)*cos(PSI_C)*sin(PHI_C)*sin(THETA_C))+0.166667E0.*(2.*L12+(-4) ...
.*L13.*cos(BETA)).*(L12.^2+4.*L13.^2+(-4)*L12.*L13.*cos(BETA)).^(-1/2) ...
.*(1+(-1)).*(L12.^2+4.*L13.^2+(-4)*L12.*L13.*cos(BETA)).^(-1)*(0.1E1.* ...
L12.^2+(-0.2E1)*L13.^2+(-0.1E1)*L12.*L13.*cos(BETA)).^2.*(L12.^2+ ...
L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^(1/2)).*(cos(PHI_C)*sin(PSI_C)+( ...
-0.1E1)*cos(PSI_C)*sin(PHI_C)*sin(THETA_C));
IJ 3 5=(-1/2).*(2.*L13+2.*L12.*cos(BETA)).*(0.333333E0)*L12.^2+(-0.666667E0)* ...
L13.^2+(-0.333333E0)*L12.*L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.* ...
cos(BETA)).^(-3/2).*cos(PSI_C).*cos(THETA_C)+(-0.133333E1)*L13+( ...
-0.333333E0)*L12.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^( ...
-1/2).*cos(PSI_C).*cos(THETA_C)+0.166667E0.*(L12.^2+4.*L13.^2+(-4)* ...
L12.*L13.*cos(BETA)).^(1/2).*((2.*L13+2.*L12.*cos(BETA)).*(L12.^2+4.* ...
L13.^2+(-4)*L12.*L13.*cos(BETA)).^(-1)*(0.1E1.*L12.^2+(-0.2E1)* ...
L13.^2+(-0.1E1)*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.* ...
cos(BETA)).^(-2)+(-2)*(0.4E1)*L13+(-0.1E1)*L12.*cos(BETA)).*( ...
L12.^2+4.*L13.^2+(-4)*L12.*L13.*cos(BETA)).^(-1)*(0.1E1.*L12.^2+ ...
-0.2E1)*L13.^2+(-0.1E1)*L12.*L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.* ...
L13.*cos(BETA)).^(-1)+(8.*L13+(-4)*L12.*cos(BETA)).*(L12.^2+4.*L13.^2+ ...
-4)*L12.*L13.*cos(BETA)).^(-2)*(0.1E1.*L12.^2+(-0.2E1)*L13.^2+ ...
-0.1E1)*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)) ...
.^(-1)).*(1+(-1)).*(L12.^2+4.*L13.^2+(-4)*L12.*L13.*cos(BETA)).^(-1) ...
0.1E1.*L12.^2+(-0.2E1)*L13.^2+(-0.1E1)*L12.*L13.*cos(BETA)).^2.*( ...
L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^(1/2)).*(cos(PHI_C)*sin( ...
PSI_C)+(-0.1E1)*cos(PSI_C)*sin(PHI_C)*sin(THETA_C))+0.166667E0.*(8.* ...
L13+(-4)*L12.*cos(BETA)).*(L12.^2+4.*L13.^2+(-4)*L12.*L13.*cos(BETA)) ...
.^(-1/2)).*(1+(-1)).*(L12.^2+4.*L13.^2+(-4)*L12.*L13.*cos(BETA)).^(-1) ...
0.1E1.*L12.^2+(-0.2E1)*L13.^2+(-0.1E1)*L12.*L13.*cos(BETA)).^2.*( ...
L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^(1/2)).*(cos(PHI_C)*sin( ...
PSI_C)+(-0.1E1)*cos(PSI_C)*sin(PHI_C)*sin(THETA_C));
IJ 3 6 = 0;
IJ 3 7=(-1).*(0.333333E0)*L12.^2+(-0.666667E0)*L13.^2+(-0.333333E0)*L12.* ...
L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1/2).*cos( ...
THETA_C)*sin(PSI_C)+0.333333E0)*(L12.^2+4.*L13.^2+(-4)*L12.*L13.*cos( ...
BETA)).^(1/2)).*(1+(-1)).*(L12.^2+4.*L13.^2+(-4)*L12.*L13.*cos(BETA)) ...
-1)*(0.1E1.*L12.^2+(-0.2E1)*L13.^2+(-0.1E1)*L12.*L13.*cos(BETA)).^2.* ...
(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^(1/2)).*(cos(PHI_C)*cos( ...
PSI_C)+0.1E1.*sin(PHI_C)*sin(THETA_C));
IJ 3 8=(-0.333333E0).*(L12.^2+4.*L13.^2+(-4)*L12.*L13.*cos(BETA)).^(1/2)).*(1+ ...
-1)).*(L12.^2+4.*L13.^2+(-4)*L12.*L13.*cos(BETA)).^(-1)*(0.1E1.*L12.^2+ ...
(-0.2E1)*L13.^2+(-0.1E1)*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.* ...
L12.*L13.*cos(BETA)).^(-1)).^(1/2)).*(cos(PSI_C)*cos(THETA_C)*sin(PHI_C) ...
+(-1)*(0.333333E0)*L12.^2+(-0.666667E0)*L13.^2+(-0.333333E0)*L12.* ...
L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1/2)).*(cos( ...
PSI_C)*sin(THETA_C));
IJ 3 9=0.333333E0.*L12.^2+4.*L13.^2+(-4)*L12.*L13.*cos(BETA)).^(1/2)).*(1+(-1) ...
).*(L12.^2+4.*L13.^2+(-4)*L12.*L13.*cos(BETA)).^(-1)*(0.1E1.*L12.^2+ ...
-0.2E1)*L13.^2+(-0.1E1)*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.* ...
L12.*L13.*cos(BETA)).^(-1)).^(1/2)).*(cos(PHI_C)*sin(PSI_C)+ ...
-0.1E1)*cos(PSI_C)*sin(THETA_C));
IJ 3 10 = 0;
IJ 3 11=L12.*L13.*(0.333333E0)*L12.^2+(-0.666667E0)*L13.^2+(-0.333333E0)*L12.* ...
L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-3/2)).*(cos( ...
PSI_C)*cos(THETA_C)*sin(BETA)+0.333333E0)*L12.*L13.*(L12.^2+L13.^2+2.* ...
L12.*L13.*cos(BETA)).^(-1/2)).*(cos(PSI_C)*cos(THETA_C)*sin(BETA)+ ...
0.666667E0)*L12.*L13.*(L12.^2+4.*L13.^2+(-4)*L12.*L13.*cos(BETA)).^( ...
-1/2)).*(1+(-1)).*(L12.^2+4.*L13.^2+(-4)*L12.*L13.*cos(BETA)).^(-1) ...
0.1E1.*L12.^2+(-0.2E1)*L13.^2+(-0.1E1)*L12.*L13.*cos(BETA)).^2.*( ...
L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^(1/2)).*(cos( ...
PHI_C)*sin(PSI_C)+(-0.1E1)*cos(PSI_C)*sin(PHI_C)*sin(THETA_C))+ ...
0.166667E0)*(L12.^2+4.*L13.^2+(-4)*L12.*L13.*cos(BETA)).^(1/2)).*(1+(-1) ...
).*(L12.^2+4.*L13.^2+(-4)*L12.*L13.*cos(BETA)).^(-1)).*(0.1E1.*L12.^2+ ...

```

```

-0.2E1).*L13.^2+(-0.1E1).*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.* ...
L12.*L13.*cos(BETA)).^(-1)).^(-1/2)).*((-2).*L12.*L13.*(L12.^2+4.*L13.^2+ ...
(-4).*L12.*L13.*cos(BETA)).^(-1)).*(0.1E1.*L12.^2+(-0.2E1).*L13.^2+( ...
-0.1E1).*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)) ...
.^(-2)).*sin(BETA)+(-0.2E1).*L12.*L13.*(L12.^2+4.*L13.^2+(-4).*L12.*L13.* ...
cos(BETA)).^(-1)).*(0.1E1.*L12.^2+(-0.2E1).*L13.^2+(-0.1E1).*L12.*L13.* ...
cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).*sin(BETA)+4.* ...
L12.*L13.*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-2)).*(0.1E1.* ...
L12.^2+(-0.2E1).*L13.^2+(-0.1E1).*L12.*L13.*cos(BETA)).^2.*(L12.^2+ ...
L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).*sin(BETA)).*(cos(PHI_C)).*sin( ...
PSI_C)+(-0.1E1).*cos(PSI_C)).*sin(PHI_C)).*sin(THETA_C));
IJ_3_12 = 0;
IJ_3_13 = 0;
IJ_3_14 = 0;
IJ_3_15 = 0;
IJ_3_16 = 0;
IJ_3_17 = 0;
IJ_3_18 = 0;
IJ_3_19 = 0;
IJ_3_20 = 0;
IJ_3_21 = 0;
IJ_3_22 = 0;
IJ_3_23 = 0;
IJ_3_24 = 0;

IJ_4_1 = 1;
IJ_4_2 = 0;
IJ_4_3 = 0;
IJ_4_4 = 0;
IJ_4_5 = 0;
IJ_4_6=cos(XI).*(sin(PHI_C)).*sin(PSI_C)+cos(PHI_C)).*cos(PSI_C)).*sin(THETA_C)) ...
+(cos(ALPHA)).*cos(PSI_C)).*cos(THETA_C)+sin(ALPHA)).*((-1)).*cos(PHI_C)).* ...
sin(PSI_C)+cos(PSI_C)).*sin(PHI_C)).*sin(THETA_C)).*sin(XI);
IJ_4_7=LB4.*cos(XI).*(cos(PSI_C)).*sin(PHI_C)+(-1)).*cos(PHI_C)).*sin(PSI_C)).* ...
sin(THETA_C))+LB4.*((-1)).*cos(ALPHA)).*cos(THETA_C)).*sin(PSI_C)+sin( ...
ALPHA)).*((-1)).*cos(PHI_C)).*cos(PSI_C)+(-1)).*sin(PHI_C)).*sin(PSI_C)).*sin( ...
THETA_C)).*sin(XI);
IJ_4_8=LB4.*cos(PHI_C)).*cos(PSI_C)).*cos(THETA_C)).*cos(XI)+LB4.*(cos(PSI_C)).* ...
cos(THETA_C)).*sin(ALPHA)).*sin(PHI_C)+(-1)).*cos(ALPHA)).*cos(PSI_C)).*sin( ...
THETA_C)).*sin(XI);
IJ_4_9=LB4.*cos(XI).*(cos(PHI_C)).*sin(PSI_C)+(-1)).*cos(PSI_C)).*sin(PHI_C)).* ...
sin(THETA_C))+LB4.*sin(ALPHA)).*sin(PHI_C)).*sin(PSI_C)+cos(PHI_C)).*cos( ...
PSI_C)).*sin(THETA_C)).*sin(XI);
IJ_4_10=LB4.*((-1)).*cos(PSI_C)).*cos(THETA_C)).*sin(ALPHA)+cos(ALPHA)).*((-1)).*cos( ...
PHI_C)).*sin(PSI_C)+cos(PSI_C)).*sin(PHI_C)).*sin(THETA_C)).*sin(XI);
IJ_4_11 = 0;
IJ_4_12=LB4.*cos(XI).*(cos(ALPHA)).*cos(PSI_C)).*cos(THETA_C)+sin(ALPHA)).*((-1)) ...
.*cos(PHI_C)).*sin(PSI_C)+cos(PSI_C)).*sin(PHI_C)).*sin(THETA_C))+(-1)).* ...
LB4.*(sin(PHI_C)).*sin(PSI_C)+cos(PHI_C)).*cos(PSI_C)).*sin(THETA_C)).*sin( ...
XI);
IJ_4_13 = 0;
IJ_4_14 = 0;
IJ_4_15 = 0;
IJ_4_16 = 0;
IJ_4_17 = 0;
IJ_4_18 = 0;
IJ_4_19 = 0;
IJ_4_20 = 0;
IJ_4_21 = 0;
IJ_4_22 = 0;
IJ_4_23 = 0;
IJ_4_24 = 0;

IJ_5_1 = 0;
IJ_5_2 = 1;
IJ_5_3 = 0;
IJ_5_4=0.333333E0.*(L12+L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)) ...
.^(-1/2)).*cos(THETA_C)).*sin(PSI_C);
IJ_5_5=0.333333E0.*(L13+L12.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)) ...
.^(-1/2)).*cos(THETA_C)).*sin(PSI_C);
IJ_5_6 = 0;
IJ_5_7=0.333333E0.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(1/2)).*cos(PSI_C)).* ...
cos(THETA_C);
IJ_5_8=(-0.333333E0).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(1/2)).*sin(PSI_C) ...
.*sin(THETA_C);
IJ_5_9 = 0;
IJ_5_10 = 0;

```

```

IJ_5_11=(-0.333333E0)*L12*L13*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1/2) ...
.*cos(THETA_C).*sin(BETA).*sin(PHI_C);
IJ_5_12 = 0;
IJ_5_13 = 0;
IJ_5_14 = 0;
IJ_5_15 = 0;
IJ_5_16 = 0;
IJ_5_17 = 0;
IJ_5_18 = 0;
IJ_5_19 = 0;
IJ_5_20 = 0;
IJ_5_21 = 0;
IJ_5_22 = 0;
IJ_5_23 = 0;
IJ_5_24 = 0;

IJ_6_1 = 0;
IJ_6_2 = 1;
IJ_6_3 = 0;
IJ_6_4=(-1/2).*(2.*L12+2.*L13.*cos(BETA)).*((-0.666667E0)*L12.^2+0.333333E0.* ...
L13.^2+(-0.333333E0)*L12.*L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.* ...
cos(BETA)).^(-3/2).*cos(THETA_C).*sin(PHI_C)+((-0.133333E1)*L12+( ...
-0.333333E0)*L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^( ...
-1/2).*cos(THETA_C).*sin(PHI_C)+0.166667E0.*(4.*L12.^2+L13.^2+(-4).* ...
L12.*L13.*cos(BETA)).^(1/2).*(2.*L12+2.*L13.*cos(BETA)).*(4.*L12.^2+ ...
L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*(0.2E1.*L12.^2+(-1.E0)*L13.^2+ ...
1.E0)*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^( ...
-2)+(-2).*(0.4E1.*L12+1.E0)*L13.*cos(BETA)).*(4.*L12.^2+L13.^2+(-4).* ...
L12.*L13.*cos(BETA)).^(-1).*(0.2E1.*L12.^2+(-1.E0)*L13.^2+1.E0)*L12.* ...
L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)+(8.*L12+( ...
-4).*L13.*cos(BETA)).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-2) ...
.*(0.2E1.*L12.^2+(-1.E0)*L13.^2+1.E0)*L12.*L13.*cos(BETA)).^2.*(L12.^2+ ...
L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).*(1+(-1).*(4.*L12.^2+L13.^2+(-4).* ...
L12.*L13.*cos(BETA)).^(-1).*(0.2E1.*L12.^2+(-1.E0)*L13.^2+1.E0)*L12.* ...
L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^( ...
-1/2).*(cos(PHI_C).*cos(PHI_C)+sin(PHI_C).*sin(PHI_C).*sin(THETA_C))+ ...
0.166667E0.*(8.*L12+(-4).*L13.*cos(BETA)).*(4.*L12.^2+L13.^2+(-4).*L12.* ...
L13.*cos(BETA)).^(-1/2).*(1+(-1).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos( ...
BETA)).^(-1).*(0.2E1.*L12.^2+(-1.E0)*L13.^2+1.E0)*L12.*L13.*cos(BETA)) ...
.^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^(1/2).*(cos(PHI_C)* ...
cos(PHI_C)+sin(PHI_C).*sin(PHI_C).*sin(THETA_C));
IJ_6_5=(-1/2).*(2.*L13+2.*L12.*cos(BETA)).*((-0.666667E0)*L12.^2+0.333333E0.* ...
L13.^2+(-0.333333E0)*L12.*L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.* ...
cos(BETA)).^(-3/2).*cos(THETA_C).*sin(PHI_C)+(0.666667E0)*L13+( ...
-0.333333E0)*L12.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^( ...
-1/2).*cos(THETA_C).*sin(PHI_C)+0.166667E0.*(4.*L12.^2+L13.^2+(-4).* ...
L12.*L13.*cos(BETA)).^(1/2).*(2.*L13+2.*L12.*cos(BETA)).*(4.*L12.^2+ ...
L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*(0.2E1.*L12.^2+(-1.E0)*L13.^2+ ...
1.E0)*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^( ...
-2)+(-2).*((-0.2E1)*L13+1.E0)*L12.*cos(BETA)).*(4.*L12.^2+L13.^2+(-4).* ...
L12.*L13.*cos(BETA)).^(-1).*(0.2E1.*L12.^2+(-1.E0)*L13.^2+1.E0)*L12.* ...
L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)+(2.*L13+( ...
-4).*L12.*cos(BETA)).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-2) ...
.*(0.2E1.*L12.^2+(-1.E0)*L13.^2+1.E0)*L12.*L13.*cos(BETA)).^2.*(L12.^2+ ...
L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).*(1+(-1).*(4.*L12.^2+L13.^2+(-4).* ...
L12.*L13.*cos(BETA)).^(-1).*(0.2E1.*L12.^2+(-1.E0)*L13.^2+1.E0)*L12.* ...
L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^( ...
-1/2).*(cos(PHI_C).*cos(PHI_C)+sin(PHI_C).*sin(PHI_C).*sin(THETA_C))+ ...
0.166667E0.*(2.*L13+(-4).*L12.*cos(BETA)).*(4.*L12.^2+L13.^2+(-4).*L12.* ...
L13.*cos(BETA)).^(-1/2).*(1+(-1).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos( ...
BETA)).^(-1).*(0.2E1.*L12.^2+(-1.E0)*L13.^2+1.E0)*L12.*L13.*cos(BETA)) ...
.^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^(1/2).*(cos(PHI_C)* ...
cos(PHI_C)+sin(PHI_C).*sin(PHI_C).*sin(THETA_C));
IJ_6_6 = 0;
IJ_6_7=(-0.666667E0)*L12.^2+0.333333E0.*L13.^2+(-0.333333E0)*L12.*L13.*cos( ...
BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1/2).*(cos(PHI_C).*cos( ...
THETA_C)+0.333333E0.*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(1/2) ...
.*(1+(-1).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*(0.2E1.* ...
L12.^2+(-1.E0)*L13.^2+1.E0)*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.* ...
L12.*L13.*cos(BETA)).^(-1)).^(1/2).*((-1).*cos(PHI_C).*sin(PHI_C)+cos( ...
PSI_C).*sin(PHI_C).*sin(THETA_C));
IJ_6_8=0.333333E0.*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(1/2).*(1+(-1) ...
).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*(0.2E1.*L12.^2+ ...
-1.E0)*L13.^2+1.E0)*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.* ...
L13.*cos(BETA)).^(-1)).^(1/2).*(cos(THETA_C).*sin(PHI_C).*sin(PHI_C)+(-1) ...
).*((-0.666667E0)*L12.^2+0.333333E0)*L13.^2+(-0.333333E0)*L12.*L13.* ...
cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1/2).*(sin(PHI_C)* ...

```



```

sin(THETA_C);
IJ_6_9=0.333333E0.*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(1/2).*(1+(-1) ...
.*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*(0.2E1.*L12.^2+ ...
-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.* ...
L13.*cos(BETA)).^(-1)).^(1/2).*((-1).*cos(PHI_C)).*sin(PHI_C)+cos(PHI_C) ...
.*sin(PHI_C).*sin(THETA_C);
IJ_6_10 = 0;
IJ_6_11=L12.*L13.*((-0.666667E0).*L12.^2+0.333333E0.*L13.^2+(-0.333333E0).*L12.* ...
L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-3/2).*cos( ...
THETA_C).*sin(BETA).*sin(PHI_C)+0.333333E0.*L12.*L13.*(L12.^2+L13.^2+2.* ...
L12.*L13.*cos(BETA)).^(-1/2).*cos(THETA_C).*sin(BETA).*sin(PHI_C)+ ...
0.666667E0.*L12.*L13.*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^( ...
-1/2).*(1+(-1)).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*( ...
0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*(L12.^2+ ...
L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^(1/2).*sin(BETA).*(cos(PHI_C).* ...
cos(PHI_C)+sin(PHI_C).*sin(PHI_C)).*sin(THETA_C)+0.166667E0.*(4.*L12.^2+ ...
L13.^2+(-4).*L12.*L13.*cos(BETA)).^(1/2).*(1+(-1)).*(4.*L12.^2+L13.^2+ ...
-4).*L12.*L13.*cos(BETA)).^(-1).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.* ...
L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1) ...
.^(-1/2).*((-2).*L12.*L13.*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)) ...
.^(-1).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.* ...
L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-2).*sin(BETA)+0.2E1.*L12.*L13.* ...
(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*(0.2E1.*L12.^2+ ...
-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.* ...
cos(BETA)).^(-1).*sin(BETA)+4.*L12.*L13.*(4.*L12.^2+L13.^2+(-4).*L12.* ...
L13.*cos(BETA)).^(-2).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.* ...
cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1).*sin(BETA) ...
.*(cos(PHI_C).*cos(PHI_C)+sin(PHI_C).*sin(PHI_C)).*sin(THETA_C);
IJ_6_12 = 0;
IJ_6_13 = 0;
IJ_6_14 = 0;
IJ_6_15 = 0;
IJ_6_16 = 0;
IJ_6_17 = 0;
IJ_6_18 = 0;
IJ_6_19 = 0;
IJ_6_20 = 0;
IJ_6_21 = 0;
IJ_6_22 = 0;
IJ_6_23 = 0;
IJ_6_24 = 0;

IJ_7_1 = 0;
IJ_7_2 = 1;
IJ_7_3 = 0;
IJ_7_4=(-1/2).*(2.*L12+2.*L13.*cos(BETA)).*(0.333333E0.*L12.^2+(-0.666667E0).* ...
L13.^2+(-0.333333E0).*L12.*L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.* ...
cos(BETA)).^(-3/2).*cos(THETA_C).*sin(PHI_C)+(0.666667E0.*L12+( ...
-0.333333E0).*L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^( ...
-1/2).*cos(THETA_C).*sin(PHI_C)+(-0.166667E0).*(L12.^2+4.*L13.^2+(-4).* ...
L12.*L13.*cos(BETA)).^(1/2).*((2.*L12+2.*L13.*cos(BETA)).*(L12.^2+4.* ...
L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*(0.1E1.*L12.^2+(-0.2E1).* ...
L13.^2+(-0.1E1).*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.* ...
cos(BETA)).^(-2)+(-2).*(0.2E1.*L12+(-0.1E1).*L13.*cos(BETA)).*(L12.^2+ ...
4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*(0.1E1.*L12.^2+(-0.2E1).* ...
L13.^2+(-0.1E1).*L12.*L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos( ...
BETA)).^(-1)+(2.*L12+(-4).*L13.*cos(BETA)).*(L12.^2+4.*L13.^2+(-4).* ...
L12.*L13.*cos(BETA)).^(-2).*(0.1E1.*L12.^2+(-0.2E1).*L13.^2+(-0.1E1).* ...
L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1) ...
(1+(-1).*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*(0.1E1.* ...
L12.^2+(-0.2E1).*L13.^2+(-0.1E1).*L12.*L13.*cos(BETA)).^2.*(L12.^2+ ...
L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^(1/2).*(cos(PHI_C).*cos(PHI_C)+ ...
sin(PHI_C).*sin(PHI_C)).*sin(THETA_C)+(-0.166667E0).*(2.*L12+(-4).*L13.* ...
cos(BETA)).*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1/2).*(1+ ...
-1).*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*(0.1E1.*L12.^2+ ...
(-0.2E1).*L13.^2+(-0.1E1).*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.* ...
L12.*L13.*cos(BETA)).^(-1)).^(1/2).*(cos(PHI_C).*cos(PHI_C)+sin(PHI_C).* ...
sin(PHI_C)).*sin(THETA_C);
IJ_7_5=(-1/2).*(2.*L13+2.*L12.*cos(BETA)).*(0.333333E0.*L12.^2+(-0.666667E0).* ...
L13.^2+(-0.333333E0).*L12.*L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.* ...
cos(BETA)).^(-3/2).*cos(THETA_C).*sin(PHI_C)+((-0.133333E1).*L13+( ...
-0.333333E0).*L12.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^( ...
-1/2).*cos(THETA_C).*sin(PHI_C)+(-0.166667E0).*(L12.^2+4.*L13.^2+(-4).* ...
L12.*L13.*cos(BETA)).^(1/2).*((2.*L13+2.*L12.*cos(BETA)).*(L12.^2+4.* ...
L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*(0.1E1.*L12.^2+(-0.2E1).* ...
L13.^2+(-0.1E1).*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.* ...
cos(BETA)).^(-2)+(-2).*((-0.4E1).*L13+(-0.1E1).*L12.*cos(BETA)).*( ...

```

```

L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*(0.1E1.*L12.^2+( ...
-0.2E1).*L13.^2+(-0.1E1).*L12.*L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*
L13.*cos(BETA)).^(-1)+(8.*L13.^2+(-4).*L12.*cos(BETA)).*(L12.^2+4.*L13.^2+( ...
-4).*L12.*L13.*cos(BETA)).^(-2).*(0.1E1.*L12.^2+(-0.2E1).*L13.^2+( ...
-0.1E1).*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)) ...
.^(-1)).*(1+(-1)).*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*( ...
0.1E1.*L12.^2+(-0.2E1).*L13.^2+(-0.1E1).*L12.*L13.*cos(BETA)).^2.*( ...
L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^(-1/2).*(cos(PHI_C).*cos( ...
PSI_C)+sin(PHI_C).*sin(PHI_C).*sin(THETA_C))+(-0.166667E0).*(8.*L13+(-4) ...
.*L12.*cos(BETA)).*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1/2) ...
.*(1+(-1)).*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*(0.1E1.*
L12.^2+(-0.2E1).*L13.^2+(-0.1E1).*L12.*L13.*cos(BETA)).^2.*(L12.^2+ ...
L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^(-1/2).*(cos(PHI_C).*cos(PSI_C)+ ...
sin(PHI_C).*sin(PSI_C).*sin(THETA_C));
IJ_7_6 = 0;
IJ_7_7 = (0.333333E0.*L12.^2+(-0.666667E0).*L13.^2+(-0.333333E0).*L12.*L13.*cos( ...
BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1/2).*cos(PSI_C).*cos( ...
THETA_C)+(-0.333333E0).*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^(- ...
1/2).*(1+(-1)).*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*( ...
0.1E1.*L12.^2+(-0.2E1).*L13.^2+(-0.1E1).*L12.*L13.*cos(BETA)).^2.*( ...
L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^(-1/2).*((-1).*cos(PHI_C).* ...
sin(PSI_C)+cos(PSI_C).*sin(PHI_C).*sin(THETA_C));
IJ_7_8 = (-0.333333E0).*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1/2).*(1+( ...
-1)).*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*(0.1E1.*L12.^2+ ...
(-0.2E1).*L13.^2+(-0.1E1).*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.* ...
L12.*L13.*cos(BETA)).^(-1)).^(-1/2).*cos(THETA_C).*sin(PHI_C).*sin(PSI_C) ...
+(-1)).*(0.333333E0.*L12.^2+(-0.666667E0).*L13.^2+(-0.333333E0).*L12.*
L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1/2).*sin( ...
PSI_C).*sin(THETA_C);
IJ_7_9 = (-0.333333E0).*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1/2).*(1+( ...
-1)).*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*(0.1E1.*L12.^2+ ...
(-0.2E1).*L13.^2+(-0.1E1).*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.* ...
L12.*L13.*cos(BETA)).^(-1)).^(-1/2).*((-1).*cos(PSI_C).*sin(PHI_C)+cos( ...
PHI_C).*sin(PSI_C).*sin(THETA_C));
IJ_7_10 = 0;
IJ_7_11 = L12.*L13.*(0.333333E0.*L12.^2+(-0.666667E0).*L13.^2+(-0.333333E0).*L12.*
L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-3/2).*cos( ...
THETA_C).*sin(BETA).*sin(PSI_C)+0.333333E0.*L12.*L13.*(L12.^2+L13.^2+2.*
L12.*L13.*cos(BETA)).^(-1/2).*cos(THETA_C).*sin(BETA).*sin(PSI_C)+( ...
-0.666667E0).*L12.*L13.*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^(- ...
1/2).*(1+(-1)).*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*( ...
0.1E1.*L12.^2+(-0.2E1).*L13.^2+(-0.1E1).*L12.*L13.*cos(BETA)).^2.*( ...
L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^(-1/2).*sin(BETA).*(cos( ...
PHI_C).*cos(PSI_C)+sin(PHI_C).*sin(PSI_C).*sin(THETA_C))+(-0.166667E0).*( ...
(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1/2)).*(1+(-1)).*(L12.^2+ ...
4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*(0.1E1.*L12.^2+(-0.2E1).*
L13.^2+(-0.1E1).*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*
cos(BETA)).^(-1)).^(-1/2).*((-2).*L12.*L13.*(L12.^2+4.*L13.^2+(-4).*
L12.*L13.*cos(BETA)).^(-1)).*(0.1E1.*L12.^2+(-0.2E1).*L13.^2+(-0.1E1).*
L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-2).*( ...
sin(BETA)+(-0.2E1).*L12.*L13.*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos( ...
BETA)).^(-1)).*(0.1E1.*L12.^2+(-0.2E1).*L13.^2+(-0.1E1).*L12.*L13.*cos( ...
BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1).*sin(BETA)+4.*L12.*
L13.*(L12.^2+4.*L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-2).*(0.1E1.*L12.^2+
(-0.2E1).*L13.^2+(-0.1E1).*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*
L12.*L13.*cos(BETA)).^(-1).*sin(BETA)).*(cos(PHI_C).*cos(PSI_C)+sin( ...
PHI_C).*sin(PSI_C).*sin(THETA_C));
IJ_7_12 = 0;
IJ_7_13 = 0;
IJ_7_14 = 0;
IJ_7_15 = 0;
IJ_7_16 = 0;
IJ_7_17 = 0;
IJ_7_18 = 0;
IJ_7_19 = 0;
IJ_7_20 = 0;
IJ_7_21 = 0;
IJ_7_22 = 0;
IJ_7_23 = 0;
IJ_7_24 = 0;

IJ_8_1 = 0;
IJ_8_2 = 1;
IJ_8_3 = 0;
IJ_8_4 = 0;
IJ_8_5 = 0;
IJ_8_6 = cos(PSI_C).*((-1).*cos(XI).*sin(PHI_C)+cos(PHI_C).*sin(ALPHA).*sin( ...

```

```

XI))+sin(PSI_C).*(cos(PHI_C).*cos(XI).*sin(THETA_C)+(cos(ALPHA).* ...
cos(THETA_C)+sin(ALPHA).*sin(PHI_C).*sin(THETA_C)).*sin(XI));
IJ_8_7=(-1)*LB4.*sin(PSI_C).*(-1).*cos(XI).*sin(PHI_C)+cos(PHI_C).*sin( ...
ALPHA).*sin(XI))+LB4.*cos(PSI_C).*(cos(PHI_C).*cos(XI).*sin(THETA_C) ...
+(cos(ALPHA).*cos(THETA_C)+sin(ALPHA).*sin(PHI_C).*sin(THETA_C)).*sin( ...
XI));
IJ_8_8=LB4.*sin(PSI_C).*(cos(PHI_C).*cos(THETA_C).*cos(XI)+(cos(THETA_C).* ...
sin(ALPHA).*sin(PHI_C)+(-1).*cos(ALPHA).*sin(THETA_C)).*sin(XI));
IJ_8_9=LB4.*cos(PSI_C).*(-1).*cos(PHI_C).*cos(XI)+(-1).*sin(ALPHA).*sin( ...
PHI_C).*sin(XI))+LB4.*sin(PSI_C).*(-1).*cos(XI).*sin(PHI_C).*sin( ...
THETA_C)+cos(PHI_C).*sin(ALPHA).*sin(THETA_C).*sin(XI));
IJ_8_10=LB4.*cos(ALPHA).*cos(PHI_C).*cos(PSI_C).*sin(XI)+LB4.*sin(PSI_C).*( ...
-1).*cos(THETA_C).*sin(ALPHA)+cos(ALPHA).*sin(PHI_C).*sin(THETA_C)).* ...
sin(XI);
IJ_8_11 = 0;
IJ_8_12=LB4.*cos(PSI_C).*(cos(PHI_C).*cos(XI).*sin(ALPHA)+sin(PHI_C).*sin( ...
XI))+LB4.*sin(PSI_C).*(cos(XI).*(cos(ALPHA).*cos(THETA_C)+sin(ALPHA) ...
.*sin(PHI_C).*sin(THETA_C))+(-1).*cos(PHI_C).*sin(THETA_C).*sin(XI)); ...
IJ_8_13 = 0;
IJ_8_14 = 0;
IJ_8_15 = 0;
IJ_8_16 = 0;
IJ_8_17 = 0;
IJ_8_18 = 0;
IJ_8_19 = 0;
IJ_8_20 = 0;
IJ_8_21 = 0;
IJ_8_22 = 0;
IJ_8_23 = 0;
IJ_8_24 = 0;

IJ_9_1 = 0;
IJ_9_2 = 0;
IJ_9_3 = 1;
IJ_9_4=(-0.333333E0).*(L12+L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos( ...
BETA)).^(-1/2).*sin(THETA_C);
IJ_9_5=(-0.333333E0).*(L13+L12.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos( ...
BETA)).^(-1/2).*sin(THETA_C);
IJ_9_6 = 0;
IJ_9_7 = 0;
IJ_9_8=(-0.333333E0).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(1/2).*cos( ...
THETA_C);
IJ_9_9 = 0;
IJ_9_10 = 0;
IJ_9_11=0.333333E0.*L12.*L13.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1/2).* ...
sin(BETA).*sin(THETA_C);
IJ_9_12 = 0;
IJ_9_13 = 0;
IJ_9_14 = 0;
IJ_9_15 = 0;
IJ_9_16 = 0;
IJ_9_17 = 0;
IJ_9_18 = 0;
IJ_9_19 = 0;
IJ_9_20 = 0;
IJ_9_21 = 0;
IJ_9_22 = 0;
IJ_9_23 = 0;
IJ_9_24 = 0;

IJ_10_1 = 0;
IJ_10_2 = 0;
IJ_10_3 = 1;
IJ_10_4=0.166667E0.*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(1/2).*(2.* ...
L12+2.*L13.*cos(BETA)).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^( ...
-1).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*( ...
L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-2)+(-2).*(0.4E1.*L12+1.E0.* ...
L13.*cos(BETA)).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*( ...
0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).*(L12.^2+ ...
L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)+(8.*L12+(-4).*L13.*cos(BETA)).*(4.* ...
L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-2).*(0.2E1.*L12.^2+(-1.E0).* ...
L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos( ...
BETA)).^(-1).*(1+(-1).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^( ...
-1).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*( ...
L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^(-1/2).*cos(THETA_C).*sin( ...
PHI_C)+0.166667E0.*(8.*L12+(-4).*L13.*cos(BETA)).*(4.*L12.^2+L13.^2+(-4) ...
.*L12.*L13.*cos(BETA)).^(-1/2).*(1+(-1).*(4.*L12.^2+L13.^2+(-4).*L12.* ...

```

```

L13.*cos(BETA)).^(-1).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.* ...
cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^(1/2).* ...
cos(THETA_C).*sin(PHI_C)+(-1/2).*(2.*L12+2.*L13.*cos(BETA)).*( ...
0.666667E0.*L12.^2+(-0.333333E0).*L13.^2+0.333333E0.*L12.*L13.*cos(BETA) ...
).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-3/2).*sin(THETA_C)+( ...
0.133333E1.*L12+0.333333E0.*L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.* ...
L13.*cos(BETA)).^(-1/2).*sin(THETA_C);
IJ_10_5=0.166667E0.*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(1/2).*(2.* ...
L13+2.*L12.*cos(BETA)).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^( ...
-1).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*( ...
L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-2)+(-2).*((-0.2E1).*L13+1.E0.* ...
L12.*cos(BETA)).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1).*( ...
0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).*(L12.^2+ ...
L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)+(2.*L13+(-4).*L12.*cos(BETA)).*(4.* ...
L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-2).*(0.2E1.*L12.^2+(-1.E0).* ...
L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos( ...
BETA)).^(-1)).*(1+(-1).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^( ...
-1)).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*( ...
L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^(1/2).*cos(THETA_C).*sin( ...
PHI_C)+0.166667E0.*(2.*L13+(-4).*L12.*cos(BETA)).*(4.*L12.^2+L13.^2+(-4) ...
.*L12.*L13.*cos(BETA)).^(-1/2).*(1+(-1).*(4.*L12.^2+L13.^2+(-4).*L12.* ...
L13.*cos(BETA)).^(-1)).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.* ...
cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^(1/2).* ...
cos(THETA_C).*sin(PHI_C)+(-1/2).*(2.*L13+2.*L12.*cos(BETA)).*( ...
0.666667E0.*L12.^2+(-0.333333E0).*L13.^2+0.333333E0.*L12.*L13.*cos(BETA) ...
).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-3/2).*sin(THETA_C)+( ...
-0.666667E0).*L13+0.333333E0.*L12.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.* ...
L13.*cos(BETA)).^(-1/2).*sin(THETA_C);
IJ_10_6 = 0;
IJ_10_7 = 0;
IJ_10_8=(0.666667E0.*L12.^2+(-0.333333E0).*L13.^2+0.333333E0.*L12.*L13.*cos( ...
BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1/2).*cos(THETA_C)+( ...
-0.333333E0).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(1/2).*(1+ ...
-1).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1)).*(0.2E1.*L12.^2+ ...
(-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.* ...
L13.*cos(BETA)).^(-1)).^(1/2).*sin(PHI_C).*sin(THETA_C);
IJ_10_9=0.333333E0.*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(1/2).*(1+(-1) ...
.*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1)).*(0.2E1.*L12.^2+ ...
-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.*L12.* ...
L13.*cos(BETA)).^(-1)).^(1/2).*cos(PHI_C).*cos(THETA_C);
IJ_10_10 = 0;
IJ_10_11=0.666667E0.*L12.*L13.*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^( ...
-1/2).*(1+(-1).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1)).*( ...
0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*(L12.^2+ ...
L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^(1/2).*cos(THETA_C).*sin(BETA).* ...
sin(PHI_C)+0.166667E0.*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^( ...
1/2).*(1+(-1).*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1)).*( ...
0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*(L12.^2+ ...
L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)).^(1/2).*cos(THETA_C).*( (-2).* ...
L12.*L13.*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1)).*(0.2E1.* ...
L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*(L12.^2+L13.^2+2.* ...
L12.*L13.*cos(BETA)).^(-2).*sin(BETA)+0.2E1.*L12.*L13.*(4.*L12.^2+ ...
L13.^2+(-4).*L12.*L13.*cos(BETA)).^(-1)).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+ ...
1.E0.*L12.*L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1) ...
.*sin(BETA)+4.*L12.*L13.*(4.*L12.^2+L13.^2+(-4).*L12.*L13.*cos(BETA)).^( ...
-2).*(0.2E1.*L12.^2+(-1.E0).*L13.^2+1.E0.*L12.*L13.*cos(BETA)).^2.*( ...
L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1).*sin(BETA)).*sin(PHI_C)+ ...
L12.*L13.*(0.666667E0.*L12.^2+(-0.333333E0).*L13.^2+0.333333E0.*L12.* ...
L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-3/2).*sin( ...
BETA).*sin(THETA_C)+(-0.333333E0).*L12.*L13.*(L12.^2+L13.^2+2.*L12.* ...
L13.*cos(BETA)).^(-1/2).*sin(BETA).*sin(THETA_C);
IJ_10_12 = 0;
IJ_10_13 = 0;
IJ_10_14 = 0;
IJ_10_15 = 0;
IJ_10_16 = 0;
IJ_10_17 = 0;
IJ_10_18 = 0;
IJ_10_19 = 0;
IJ_10_20 = 0;
IJ_10_21 = 0;
IJ_10_22 = 0;
IJ_10_23 = 0;
IJ_10_24 = 0;

IJ_11_1 = 0;
IJ_11_2 = 0;

```

```

IJ_11_3 = 1;
IJ_11_4 = (-0.166667E0) * (L12.^2+4.*L13.^2+(-4). *L12.*L13.*cos(BETA)).^(1/2) * (( ...
2.*L12+2.*L13.*cos(BETA)).*(L12.^2+4.*L13.^2+(-4). *L12.*L13.*cos(BETA)) ...
.^(-1) * (0.1E1.*L12.^2+(-0.2E1). *L13.^2+(-0.1E1). *L12.*L13.*cos(BETA)) ...
.^2 * (L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-2) + (-2) * (0.2E1.*L12+ ...
-0.1E1). *L13.*cos(BETA)).*(L12.^2+4.*L13.^2+(-4). *L12.*L13.*cos(BETA)) ...
.^(-1) * (0.1E1.*L12.^2+(-0.2E1). *L13.^2+(-0.1E1). *L12.*L13.*cos(BETA)) * ...
(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1) + (2.*L12+(-4). *L13.*cos( ...
BETA)).*(L12.^2+4.*L13.^2+(-4). *L12.*L13.*cos(BETA)).^(-2) * (0.1E1.* ...
L12.^2+(-0.2E1). *L13.^2+(-0.1E1). *L12.*L13.*cos(BETA)).^2 * (L12.^2+ ...
L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)) * (1+(-1). *L12.^2+4.*L13.^2+(-4). * ...
L12.*L13.*cos(BETA)).^(-1) * (0.1E1.*L12.^2+(-0.2E1). *L13.^2+(-0.1E1). * ...
L12.*L13.*cos(BETA)).^2 * (L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)) ...
.^(-1/2) * cos(THETA_C) * sin(PHI_C) + (-0.166667E0) * (2.*L12+(-4). *L13.* ...
cos(BETA)).*(L12.^2+4.*L13.^2+(-4). *L12.*L13.*cos(BETA)).^(-1/2) * (1+ ...
-1). * (L12.^2+4.*L13.^2+(-4). *L12.*L13.*cos(BETA)).^(-1) * (0.1E1.*L12.^2+ ...
(-0.2E1). *L13.^2+(-0.1E1). *L12.*L13.*cos(BETA)).^2 * (L12.^2+L13.^2+2.* ...
L12.*L13.*cos(BETA)).^(-1)) * cos(THETA_C) * sin(PHI_C) + (-1/2) * ( ...
2.*L12+2.*L13.*cos(BETA)).*(L12.^2+4.*L13.^2+(-4). *L12.*L13.*cos(BETA)) ...
0.333333E0 * L12.*L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)) ...
.^(-3/2) * sin(THETA_C) + ((-0.666667E0) * L12+0.333333E0 * L13.*cos(BETA)) * ...
(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1/2) * sin(THETA_C);
IJ_11_5 = (-0.166667E0) * (L12.^2+4.*L13.^2+(-4). *L12.*L13.*cos(BETA)).^(1/2) * (( ...
2.*L13+2.*L12.*cos(BETA)).*(L12.^2+4.*L13.^2+(-4). *L12.*L13.*cos(BETA)) ...
.^(-1) * (0.1E1.*L12.^2+(-0.2E1). *L13.^2+(-0.1E1). *L12.*L13.*cos(BETA)) ...
.^2 * (L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-2) + (-2) * ((-0.4E1). *L13+ ...
-0.1E1). *L12.*cos(BETA)).*(L12.^2+4.*L13.^2+(-4). *L12.*L13.*cos(BETA)) ...
.^(-1) * (0.1E1.*L12.^2+(-0.2E1). *L13.^2+(-0.1E1). *L12.*L13.*cos(BETA)) * ...
(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1) + (8.*L13+(-4). *L12.*cos( ...
BETA)).*(L12.^2+4.*L13.^2+(-4). *L12.*L13.*cos(BETA)).^(-2) * (0.1E1.* ...
L12.^2+(-0.2E1). *L13.^2+(-0.1E1). *L12.*L13.*cos(BETA)).^2 * (L12.^2+ ...
L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)) * (1+(-1). *L12.^2+4.*L13.^2+(-4). * ...
L12.*L13.*cos(BETA)).^(-1) * (0.1E1.*L12.^2+(-0.2E1). *L13.^2+(-0.1E1). * ...
L12.*L13.*cos(BETA)).^2 * (L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)) ...
.^(-1/2) * cos(THETA_C) * sin(PHI_C) + (-0.166667E0) * (8.*L13+(-4). *L12.* ...
cos(BETA)).*(L12.^2+4.*L13.^2+(-4). *L12.*L13.*cos(BETA)).^(-1/2) * (1+ ...
-1). * (L12.^2+4.*L13.^2+(-4). *L12.*L13.*cos(BETA)).^(-1) * (0.1E1.*L12.^2+ ...
(-0.2E1). *L13.^2+(-0.1E1). *L12.*L13.*cos(BETA)).^2 * (L12.^2+L13.^2+2.* ...
L12.*L13.*cos(BETA)).^(-1)) * cos(THETA_C) * sin(PHI_C) + (-1/2) * ( ...
2.*L13+2.*L12.*cos(BETA)).*(L12.^2+4.*L13.^2+(-4). *L12.*L13.*cos(BETA)) ...
0.333333E0 * L12.*L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)) ...
.^(-3/2) * sin(THETA_C) + (0.133333E1 * L13+0.333333E0 * L12.*cos(BETA)) * ( ...
L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1/2) * sin(THETA_C);
IJ_11_6 = 0;
IJ_11_7 = 0;
IJ_11_8 = ((-0.333333E0) * L12.^2+0.666667E0 * L13.^2+0.333333E0 * L12.*L13.*cos( ...
BETA)).*(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1/2) * cos(THETA_C) + ...
0.333333E0 * (L12.^2+4.*L13.^2+(-4). *L12.*L13.*cos(BETA)).^(1/2) * (1+(-1) ...
.* (L12.^2+4.*L13.^2+(-4). *L12.*L13.*cos(BETA)).^(-1) * (0.1E1.*L12.^2+ ...
-0.2E1). *L13.^2+(-0.1E1). *L12.*L13.*cos(BETA)).^2 * (L12.^2+L13.^2+2.* ...
L12.*L13.*cos(BETA)).^(-1)) * cos(THETA_C) * sin(PHI_C) * sin(THETA_C);
IJ_11_9 = (-0.333333E0) * (L12.^2+4.*L13.^2+(-4). *L12.*L13.*cos(BETA)).^(1/2) * (( ...
-1). * (L12.^2+4.*L13.^2+(-4). *L12.*L13.*cos(BETA)).^(-1) * (0.1E1.*L12.^2+ ...
(-0.2E1). *L13.^2+(-0.1E1). *L12.*L13.*cos(BETA)).^2 * (L12.^2+L13.^2+2.* ...
L12.*L13.*cos(BETA)).^(-1)) * cos(THETA_C) * sin(PHI_C);
IJ_11_10 = 0;
IJ_11_11 = (-0.666667E0) * L12.*L13.* (L12.^2+4.*L13.^2+(-4). *L12.*L13.*cos(BETA)).^( ...
-1/2) * (1+(-1). * (L12.^2+4.*L13.^2+(-4). *L12.*L13.*cos(BETA)).^(-1) * ( ...
0.1E1.*L12.^2+(-0.2E1). *L13.^2+(-0.1E1). *L12.*L13.*cos(BETA)).^2 * ( ...
L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)) * cos(THETA_C) * sin( ...
BETA) * sin(PHI_C) + (-0.166667E0) * (L12.^2+4.*L13.^2+(-4). *L12.*L13.*cos( ...
BETA)).^(1/2) * (1+(-1). * (L12.^2+4.*L13.^2+(-4). *L12.*L13.*cos(BETA)).^(-1) * ( ...
-1). * (0.1E1.*L12.^2+(-0.2E1). *L13.^2+(-0.1E1). *L12.*L13.*cos(BETA)).^2 * ...
(L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)) * cos(THETA_C) * (( ...
-2). *L12.*L13.* (L12.^2+4.*L13.^2+(-4). *L12.*L13.*cos(BETA)).^(-1) * ( ...
0.1E1.*L12.^2+(-0.2E1). *L13.^2+(-0.1E1). *L12.*L13.*cos(BETA)).^2 * ( ...
L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-2) * sin(BETA) + (-0.2E1). *L12.* ...
L13.* (L12.^2+4.*L13.^2+(-4). *L12.*L13.*cos(BETA)).^(-1) * (0.1E1.*L12.^2+ ...
(-0.2E1). *L13.^2+(-0.1E1). *L12.*L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.* ...
L13.*cos(BETA)).^(-1) * sin(BETA) + 4.*L12.*L13.* (L12.^2+4.*L13.^2+(-4). * ...
L12.*L13.*cos(BETA)).^(-2) * (0.1E1.*L12.^2+(-0.2E1). *L13.^2+(-0.1E1). * ...
L12.*L13.*cos(BETA)).^2 * (L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1)) * ...
sin(BETA) * sin(PHI_C) + L12.*L13.* ((-0.333333E0) * L12.^2+0.666667E0 * ...
L13.^2+0.333333E0 * L12.*L13.*cos(BETA)).*(L12.^2+L13.^2+2.*L12.*L13.* ...
cos(BETA)).^(-3/2) * sin(BETA) * sin(THETA_C) + (-0.333333E0) * L12.*L13.* ( ...
L12.^2+L13.^2+2.*L12.*L13.*cos(BETA)).^(-1/2) * sin(BETA) * sin(THETA_C); ...
IJ_11_12 = 0;

```

```

IJ_11_13 = 0;
IJ_11_14 = 0;
IJ_11_15 = 0;
IJ_11_16 = 0;
IJ_11_17 = 0;
IJ_11_18 = 0;
IJ_11_19 = 0;
IJ_11_20 = 0;
IJ_11_21 = 0;
IJ_11_22 = 0;
IJ_11_23 = 0;
IJ_11_24 = 0;

IJ_12_1 = 0;
IJ_12_2 = 0;
IJ_12_3 = 1;
IJ_12_4 = 0;
IJ_12_5 = 0;
IJ_12_6=cos(PHI_C).*cos(THETA_C).*cos(XI)+(cos(THETA_C).*sin(ALPHA).*sin( ...
    PHI_C)+(-1).*cos(ALPHA).*sin(THETA_C)).*sin(XI);
IJ_12_7 = 0;
IJ_12_8=(-1).*LB4.*cos(PHI_C).*cos(XI).*sin(THETA_C)+LB4.*((-1).*cos(ALPHA).* ...
    cos(THETA_C)+(-1).*sin(ALPHA).*sin(PHI_C).*sin(THETA_C)).*sin(XI);
IJ_12_9=(-1).*LB4.*cos(THETA_C).*cos(XI).*sin(PHI_C)+LB4.*cos(PHI_C).*cos( ...
    THETA_C).*sin(ALPHA).*sin(XI);
IJ_12_10=LB4.*(cos(ALPHA).*cos(THETA_C).*sin(PHI_C)+sin(ALPHA).*sin(THETA_C)).* ...
    sin(XI);
IJ_12_11 = 0;
IJ_12_12=LB4.*cos(XI).(cos(THETA_C).*sin(ALPHA).*sin(PHI_C)+(-1).*cos(ALPHA).* ...
    sin(THETA_C))+(-1).*LB4.*cos(PHI_C).*cos(THETA_C).*sin(XI);
IJ_12_13 = 0;
IJ_12_14 = 0;
IJ_12_15 = 0;
IJ_12_16 = 0;
IJ_12_17 = 0;
IJ_12_18 = 0;
IJ_12_19 = 0;
IJ_12_20 = 0;
IJ_12_21 = 0;
IJ_12_22 = 0;
IJ_12_23 = 0;
IJ_12_24 = 0;

IJ_13_1 = 0;
IJ_13_2 = 0;
IJ_13_3 = 0;
IJ_13_4 = 0;
IJ_13_5 = 0;
IJ_13_6 = 0;
IJ_13_7 = 1;
IJ_13_8=(cos(PHI_C).*cos(THETA_1).*sin(PHI_1)+sin(PHI_C).*sin(THETA_1)).*(cos( ...
    PHI_C).*cos(THETA_C).*sin(THETA_1)+cos(THETA_1)).*((-1).*cos(THETA_C)).* ...
    sin(PHI_C).*sin(PHI_1)+cos(PHI_1).*sin(THETA_C)).*(cos(PHI_1).^2.*cos( ...
    THETA_1).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C)).* ...
    sin(PHI_1).*sin(2.*THETA_1)+2.*cos(PHI_1).*cos(THETA_1).*cos(THETA_C)).*( ...
    cos(THETA_1).*sin(PHI_C).*sin(PHI_1)+(-1).*cos(PHI_C).*sin(THETA_1)).* ...
    sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_1).^2+cos(THETA_1).^2.*sin(PHI_1) ...
    .^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_1).^2.*sin(PHI_1).^2+ ...
    sin(THETA_1).^2.*sin(THETA_C).^2)).^(-1);
IJ_13_9=(1/4).*(4.*cos(PHI_1).*cos(THETA_1).*cos(THETA_C)).*((-1).*cos(THETA_1)).* ...
    sin(PHI_C).*sin(PHI_1)+cos(PHI_C).*sin(THETA_1))+((-3)+cos(2.*PHI_1)+2.* ...
    cos(PHI_1).^2.*cos(2.*THETA_1)).*sin(THETA_C)).*(cos(PHI_1).^2.*cos( ...
    THETA_1).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C)).* ...
    sin(PHI_1).*sin(2.*THETA_1)+2.*cos(PHI_1).*cos(THETA_1).*cos(THETA_C)).*( ...
    cos(THETA_1).*sin(PHI_C).*sin(PHI_1)+(-1).*cos(PHI_C).*sin(THETA_1)).* ...
    sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_1).^2+cos(THETA_1).^2.*sin(PHI_1) ...
    .^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_1).^2.*sin(PHI_1).^2+ ...
    sin(THETA_1).^2.*sin(THETA_C).^2)).^(-1);
IJ_13_10 = 0;
IJ_13_11 = 0;
IJ_13_12 = 0;
IJ_13_13=(1/2).*(2.*cos(PHI_C).*cos(THETA_1).^2.*cos(THETA_C)+cos(THETA_C).*sin( ...
    PHI_C).*sin(PHI_1).*sin(2.*THETA_1)+(-2).*cos(PHI_1).*cos(THETA_1).*sin( ...
    THETA_1).*sin(THETA_C)).*(cos(PHI_1).^2.*cos(THETA_1).^2.*cos(THETA_C) ...
    .^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C).*sin(PHI_1).*sin(2.*THETA_1) ...
    +2.*cos(PHI_1).*cos(THETA_1).*cos(THETA_C)).*(cos(THETA_1).*sin(PHI_C)).* ...
    sin(PHI_1)+(-1).*cos(PHI_C).*sin(THETA_1)).*sin(THETA_C)+sin(PHI_C).^2.* ...

```

```

(sin(THETA_1).^2+cos(THETA_1).^2.*sin(PSI_1).^2.*sin(THETA_C).^2)+cos( ...
PHI_C).^2.*cos(THETA_1).^2.*sin(PSI_1).^2+sin(THETA_1).^2.*sin(THETA_C) ...
.^2).^(-1);
IJ_13_14=(cos(PSI_1).*cos(THETA_C).*sin(PHI_C)+sin(PSI_1).*sin(THETA_C)).*(cos( ...
PSI_1).^2.*cos(THETA_1).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C) ...
.^2.*sin(PHI_C).*sin(PSI_1).*sin(2.*THETA_1)+2.*cos(PSI_1).*cos(THETA_1) ...
.*cos(THETA_C)).*(cos(THETA_1).*sin(PSI_1).*sin(PSI_1)+(-1).*cos(PHI_C)).* ...
sin(THETA_1)).*sin(THETA_C)+sin(PHI_C).^2.*sin(THETA_1).^2+cos(THETA_1) ...
.^2.*sin(PSI_1).^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_1).^2.* ...
sin(PSI_1).^2+sin(THETA_1).^2.*sin(THETA_C).^2)).^(-1);
IJ_13_15 = 0;
IJ_13_16 = 0;
IJ_13_17 = 0;
IJ_13_18 = 0;
IJ_13_19 = 0;
IJ_13_20 = 0;
IJ_13_21 = 0;
IJ_13_22 = 0;
IJ_13_23 = 0;
IJ_13_24 = 0;

IJ_14_1 = 0;
IJ_14_2 = 0;
IJ_14_3 = 0;
IJ_14_4 = 0;
IJ_14_5 = 0;
IJ_14_6 = 0;
IJ_14_7 = 0;
IJ_14_8=(cos(PSI_1).*cos(THETA_1).*cos(THETA_C)+(cos(THETA_1).*sin(PHI_C).*sin( ...
PSI_1)+(-1).*cos(PHI_C).*sin(THETA_1)).*sin(THETA_C)).*(1+(-1).*cos( ...
PHI_C).*cos(THETA_C).*sin(THETA_1)+cos(THETA_1).*((-1).*cos(THETA_C)).* ...
sin(PHI_C).*sin(PSI_1)+cos(PSI_1)).*sin(THETA_C)).^2).^(-1/2);
IJ_14_9=(-1).*cos(THETA_C).*cos(PHI_C).*cos(THETA_1).*sin(PSI_1)+sin(PHI_C).* ...
sin(THETA_1)).*(1+(-1).*cos(PHI_C).*cos(THETA_C).*sin(THETA_1)+cos( ...
THETA_1)).*((-1).*cos(THETA_C)).*sin(PHI_C).*sin(PSI_1)+cos(PSI_1)).*sin( ...
THETA_C)).^2).^(-1/2);
IJ_14_10 = 0;
IJ_14_11 = 0;
IJ_14_12 = 0;
IJ_14_13=(-1).*cos(THETA_1).*cos(PSI_1).*cos(THETA_C).*sin(PHI_C)+sin(PSI_1).* ...
sin(THETA_C)).*(1+(-1).*cos(PHI_C).*cos(THETA_C)).*sin(THETA_1)+cos( ...
THETA_1)).*((-1).*cos(THETA_C)).*sin(PHI_C).*sin(PSI_1)+cos(PSI_1)).*sin( ...
THETA_C)).^2).^(-1/2);
IJ_14_14=(cos(PHI_C).*cos(THETA_1).*cos(THETA_C)+sin(THETA_1)).*(cos(THETA_C)).* ...
sin(PHI_C).*sin(PSI_1)+(-1).*cos(PSI_1).*sin(THETA_C)).*(1+(-1).*cos( ...
PHI_C).*cos(THETA_C).*sin(THETA_1)+cos(THETA_1)).*((-1).*cos(THETA_C)).* ...
sin(PHI_C).*sin(PSI_1)+cos(PSI_1)).*sin(THETA_C)).^2).^(-1/2);
IJ_14_15 = 0;
IJ_14_16 = 0;
IJ_14_17 = 0;
IJ_14_18 = 0;
IJ_14_19 = 0;
IJ_14_20 = 0;
IJ_14_21 = 0;
IJ_14_22 = 0;
IJ_14_23 = 0;
IJ_14_24 = 0;

IJ_15_1 = 0;
IJ_15_2 = 0;
IJ_15_3 = 0;
IJ_15_4 = 0;
IJ_15_5 = 0;
IJ_15_6 = 0;
IJ_15_7 = 0;
IJ_15_8=(cos(PHI_C).*cos(THETA_1).*sin(PSI_1)+sin(PHI_C).*sin(THETA_1)).*(cos( ...
PHI_C).^2.*cos(THETA_1).^2.*cos(THETA_C).^2+2.*cos(PHI_C).*cos(THETA_1) ...
.*cos(THETA_C).*sin(THETA_1)).*(cos(THETA_C).*sin(PHI_C)).*sin(PSI_1)+(-1) ...
.*cos(PSI_1)).*sin(THETA_C)+sin(PSI_1).^2.*cos(THETA_C).^2.*sin(PHI_C) ...
.^2.*sin(THETA_1).^2+sin(THETA_1).^2.*sin(THETA_C).^2)+cos(PSI_1).^2.*cos(THETA_C).^2.* ...
sin(PHI_C).^2+sin(THETA_1).^2.*sin(THETA_C).^2)+cos(PSI_1).*cos(THETA_1) ...
.^2.*sin(PHI_C).*sin(PSI_1)).*sin(2.*THETA_C)).^(-1);
IJ_15_9=(1/2).*(2.*cos(PSI_1)).*cos(THETA_1).*cos(THETA_C).^2+(-2).*cos(PHI_C).* ...
cos(THETA_C)).*sin(THETA_1)).*sin(THETA_C)+cos(THETA_1)).*sin(PHI_C)).*sin( ...
PSI_1)).*sin(2.*THETA_C)).*(cos(PHI_C).^2.*cos(THETA_1).^2.*cos(THETA_C) ...
.^2+2.*cos(PHI_C).*cos(THETA_1).*cos(THETA_C)).*sin(THETA_1)).*(cos( ...
THETA_C)).*sin(PHI_C)).*sin(PSI_1)+(-1).*cos(PSI_1)).*sin(THETA_C))+sin( ...

```

```

PSI_1).^2.*(cos(THETA_C).^2.*sin(PHI_C).^2.*sin(THETA_1).^2+sin(THETA_C) ...
.^2)+cos(PSI_1).^2.*(cos(THETA_C).^2.*sin(PHI_C).^2+sin(THETA_1).^2.*...
sin(THETA_C).^2)+cos(PSI_1).*cos(THETA_1).^2.*sin(PHI_C).*sin(PSI_1).* ...
sin(2.*THETA_C)).^(-1);
IJ_15_10 = 0;
IJ_15_11 = 0;
IJ_15_12 = 0;
IJ_15_13=(1/4).*cos(PHI_C).^2.*cos(THETA_1).^2.*cos(THETA_C).^2+2.*cos(PHI_C).* ...
cos(THETA_1).*cos(THETA_C).*sin(THETA_1).*(cos(THETA_C).*sin(PHI_C).* ...
sin(PSI_1))+(-1).*cos(PSI_1).*sin(THETA_C))+sin(PSI_1).^2.*(cos(THETA_C) ...
.^2.*sin(PHI_C).^2.*sin(THETA_1).^2+sin(THETA_C).^2)+cos(PSI_1).^2.* ...
cos(THETA_C).^2.*sin(PHI_C).^2+sin(THETA_1).^2.*sin(THETA_C).^2)+cos( ...
PSI_1).*cos(THETA_1).^2.*sin(PHI_C).*sin(PSI_1).*sin(2.*THETA_C)).^(-1) ...
.*( (-3)+cos(2.*PHI_C)+2.*cos(PHI_C).^2.*cos(2.*THETA_C)).*sin(THETA_1)+ ...
cos(THETA_1).*( (-2)-.*cos(THETA_C).^2.*sin(2.*PHI_C).*sin(PSI_1)+2.*cos( ...
PHI_C).*cos(PSI_1).*sin(2.*THETA_C)));
IJ_15_14=(cos(PSI_1).*cos(THETA_C).*sin(PHI_C)+sin(PSI_1).*sin(THETA_C)).*(cos( ...
PHI_C).*cos(THETA_C).*sin(THETA_1)+cos(THETA_1).*( (-1).*cos(THETA_C)).* ...
sin(PHI_C).*sin(PSI_1)+cos(PSI_1).*sin(THETA_C)).*(cos(PHI_C).^2.*cos( ...
THETA_1).^2.*cos(THETA_C).^2+2.*cos(PHI_C).*cos(THETA_1).*cos(THETA_C).* ...
sin(THETA_1).*(cos(THETA_C).*sin(PHI_C).*sin(PSI_1))+(-1).*cos(PSI_1).* ...
sin(THETA_C))+sin(PSI_1).^2.*(cos(THETA_C).^2.*sin(PHI_C).^2.*sin( ...
THETA_1).^2+sin(THETA_C).^2)+cos(PSI_1).^2.*(cos(THETA_C).^2.*sin(PHI_C) ...
.^2+sin(THETA_1).^2.*sin(THETA_C).^2)+cos(PSI_1).^2.*cos(THETA_1).^2.*sin( ...
PHI_C).*sin(PSI_1).*sin(2.*THETA_C)).^(-1);
IJ_15_15 = 1;
IJ_15_16 = 0;
IJ_15_17 = 0;
IJ_15_18 = 0;
IJ_15_19 = 0;
IJ_15_20 = 0;
IJ_15_21 = 0;
IJ_15_22 = 0;
IJ_15_23 = 0;
IJ_15_24 = 0;

IJ_16_1 = 0;
IJ_16_2 = 0;
IJ_16_3 = 0;
IJ_16_4 = 0;
IJ_16_5 = 0;
IJ_16_6 = 0;
IJ_16_7 = 1;
IJ_16_8=(cos(PHI_C).*cos(THETA_2).*sin(PSI_2)+sin(PHI_C).*sin(THETA_2)).*(cos( ...
PHI_C).*cos(THETA_C).*sin(THETA_2)+cos(THETA_2).*( (-1).*cos(THETA_C)).* ...
sin(PHI_C).*sin(PSI_2)+cos(PSI_2).*sin(THETA_C)).*(cos(PSI_2).^2.*cos( ...
THETA_2).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C).* ...
sin(PSI_2).*sin(2.*THETA_2)+2.*cos(PSI_2).*cos(THETA_2).*cos(THETA_C).*( ...
cos(THETA_2).*sin(PHI_C).*sin(PSI_2))+(-1).*cos(PHI_C).*sin(THETA_2)).* ...
sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_2).^2+cos(THETA_2).^2.*sin(PSI_2) ...
).^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_2).^2.*sin(PSI_2).^2+ ...
sin(THETA_2).^2.*sin(THETA_C).^2)).^(-1);
IJ_16_9=(1/4).*cos(PSI_2).^2.*cos(THETA_2).*cos(THETA_C).*( (-1).*cos(THETA_2)).* ...
sin(PHI_C).*sin(PSI_2)+cos(PHI_C).*sin(THETA_2))+((-3)+cos(2.*PSI_2)+2.* ...
cos(PSI_2).^2.*cos(2.*THETA_2)).*sin(THETA_C)).*(cos(PSI_2).^2.*cos( ...
THETA_2).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C).* ...
sin(PSI_2).*sin(2.*THETA_2)+2.*cos(PSI_2).*cos(THETA_2).*cos(THETA_C).*( ...
cos(THETA_2).*sin(PHI_C).*sin(PSI_2))+(-1).*cos(PHI_C).*sin(THETA_2)).* ...
sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_2).^2+cos(THETA_2).^2.*sin(PSI_2) ...
).^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_2).^2.*sin(PSI_2).^2+ ...
sin(THETA_2).^2.*sin(THETA_C).^2)).^(-1);
IJ_16_10 = 0;
IJ_16_11 = 0;
IJ_16_12 = 0;
IJ_16_13 = 0;
IJ_16_14 = 0;
IJ_16_15 = 0;
IJ_16_16=(1/2).*cos(PHI_C).*cos(THETA_2).^2.*cos(THETA_C)+cos(THETA_C).*sin( ...
PHI_C).*sin(PSI_2).*sin(2.*THETA_2)+(-2).*cos(PSI_2).*cos(THETA_2).*sin( ...
THETA_2).*sin(THETA_C)).*(cos(PSI_2).^2.*cos(THETA_2).^2.*cos(THETA_C) ...
.^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C).*sin(PSI_2).*sin(2.*THETA_2) ...
+2.*cos(PSI_2).*cos(THETA_2).*cos(THETA_C)).*(cos(THETA_2).*sin(PHI_C)).* ...
sin(PSI_2)+(-1).*cos(PHI_C).*sin(THETA_2)).*sin(THETA_C)+sin(PHI_C).^2.* ...
sin(THETA_2).^2+cos(THETA_2).^2.*sin(PSI_2).^2.*sin(THETA_C).^2)+cos( ...
PHI_C).^2.*(cos(THETA_2).^2.*sin(PSI_2).^2+sin(THETA_2).^2.*sin(THETA_C) ...
).^2)).^(-1);
IJ_16_17=(cos(PSI_2).*cos(THETA_C).*sin(PHI_C)+sin(PSI_2).*sin(THETA_C)).*(cos( ...

```



```

PSI_2).^2.*cos(THETA_2).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C) ...
.^2.*sin(PHI_C).*sin(PSI_2).*sin(2.*THETA_2)+2.*cos(PSI_2).*cos(THETA_2) ...
.*cos(THETA_C).*cos(THETA_2).*sin(PHI_C).*sin(PSI_2)+(-1).*cos(PHI_C).* ...
sin(THETA_2)).*sin(THETA_C)+sin(PHI_C).^2.*sin(THETA_2).^2+cos(THETA_2) ...
.^2.*sin(PSI_2).^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_2).^2.* ...
sin(PSI_2).^2+sin(THETA_2).^2.*sin(THETA_C).^2)).^(-1);
IJ_16_18 = 0;
IJ_16_19 = 0;
IJ_16_20 = 0;
IJ_16_21 = 0;
IJ_16_22 = 0;
IJ_16_23 = 0;
IJ_16_24 = 0;

IJ_17_1 = 0;
IJ_17_2 = 0;
IJ_17_3 = 0;
IJ_17_4 = 0;
IJ_17_5 = 0;
IJ_17_6 = 0;
IJ_17_7 = 0;
IJ_17_8=(cos(PSI_2).*cos(THETA_2).*cos(THETA_C)+(cos(THETA_2).*sin(PHI_C).*sin( ...
PSI_2)+(-1).*cos(PHI_C).*sin(THETA_2)).*sin(THETA_C)).*(1+(-1).*cos( ...
PHI_C).*cos(THETA_C).*sin(THETA_2)+cos(THETA_2).*((-1).*cos(THETA_C).* ...
sin(PHI_C).*sin(PSI_2)+cos(PSI_2).*sin(THETA_C))).^2).^(-1/2);
IJ_17_9=(-1).*cos(THETA_C).*cos(PHI_C).*cos(THETA_2).*sin(PSI_2)+sin(PHI_C).* ...
sin(THETA_2)).*(1+(-1).*cos(PHI_C).*cos(THETA_C).*sin(THETA_2)+cos( ...
THETA_2)).*((-1).*cos(THETA_C).*sin(PHI_C).*sin(PSI_2)+cos(PSI_2).*sin( ...
THETA_C))).^2).^(-1/2);
IJ_17_10 = 0;
IJ_17_11 = 0;
IJ_17_12 = 0;
IJ_17_13 = 0;
IJ_17_14 = 0;
IJ_17_15 = 0;
IJ_17_16=(-1).*cos(THETA_2).*cos(PSI_2).*cos(THETA_C).*sin(PHI_C)+sin(PSI_2).* ...
sin(THETA_C)).*(1+(-1).*cos(PHI_C).*cos(THETA_C).*sin(THETA_2)+cos( ...
THETA_2)).*((-1).*cos(THETA_C).*sin(PHI_C).*sin(PSI_2)+cos(PSI_2).*sin( ...
THETA_C))).^2).^(-1/2);
IJ_17_17=(cos(PHI_C).*cos(THETA_2).*cos(THETA_C)+sin(THETA_2).*cos(THETA_C).* ...
sin(PHI_C).*sin(PSI_2)+(-1).*cos(PSI_2).*sin(THETA_C))).*(1+(-1).*cos( ...
PHI_C).*cos(THETA_C).*sin(THETA_2)+cos(THETA_2)).*((-1).*cos(THETA_C).* ...
sin(PHI_C).*sin(PSI_2)+cos(PSI_2).*sin(THETA_C))).^2).^(-1/2);
IJ_17_18 = 0;
IJ_17_19 = 0;
IJ_17_20 = 0;
IJ_17_21 = 0;
IJ_17_22 = 0;
IJ_17_23 = 0;
IJ_17_24 = 0;

IJ_18_1 = 0;
IJ_18_2 = 0;
IJ_18_3 = 0;
IJ_18_4 = 0;
IJ_18_5 = 0;
IJ_18_6 = 0;
IJ_18_7 = 0;
IJ_18_8=(cos(PHI_C).*cos(THETA_2).*sin(PSI_2)+sin(PHI_C).*sin(THETA_2)).*(cos( ...
PHI_C).^2.*cos(THETA_2).^2.*cos(THETA_C).^2+2.*cos(PHI_C).*cos(THETA_2) ...
.*cos(THETA_C).*sin(THETA_2)).*(cos(THETA_C).*sin(PHI_C).*sin(PSI_2)+(-1) ...
.*cos(PSI_2).*sin(THETA_C)+sin(PSI_2).^2.*(cos(THETA_C).^2.*sin(PHI_C) ...
.^2.*sin(THETA_2).^2+sin(THETA_C).^2)+cos(PSI_2).^2.*(cos(THETA_C).^2.* ...
sin(PHI_C).^2+sin(THETA_2).^2.*sin(THETA_C).^2)+cos(PSI_2).*cos(THETA_2) ...
.^2.*sin(PHI_C).*sin(PSI_2).*sin(2.*THETA_C))).^(-1);
IJ_18_9=(1/2).*(2.*cos(PSI_2).*cos(THETA_2).*cos(THETA_C).^2+(-2).*cos(PHI_C).* ...
cos(THETA_C).*sin(THETA_2).*sin(THETA_C)+cos(THETA_2).*sin(PHI_C).*sin( ...
PSI_2).*sin(2.*THETA_C)).*(cos(PHI_C).^2.*cos(THETA_2).^2.*cos(THETA_C) ...
.^2+2.*cos(PHI_C).*cos(THETA_2).*cos(THETA_C).*sin(THETA_2)).*(cos( ...
THETA_C).*sin(PHI_C).*sin(PSI_2)+(-1).*cos(PSI_2).*sin(THETA_C))+sin( ...
PSI_2).^2.*(cos(THETA_C).^2.*sin(PHI_C).^2.*sin(THETA_2).^2+sin(THETA_C) ...
.^2)+cos(PSI_2).^2.*(cos(THETA_C).^2.*sin(PHI_C).^2+sin(THETA_2).^2.* ...
sin(THETA_C).^2)+cos(PSI_2).*cos(THETA_2).^2.*sin(PHI_C).*sin(PSI_2)).* ...
sin(2.*THETA_C))).^(-1);
IJ_18_10 = 0;
IJ_18_11 = 0;
IJ_18_12 = 0;

```

```

IJ_18_13 = 0;
IJ_18_14 = 0;
IJ_18_15 = 0;
IJ_18_16=(1/4).*(cos(PHI_C).^2.*cos(THETA_2).^2.*cos(THETA_C).^2+2.*cos(PHI_C).* ...
cos(THETA_2).*cos(THETA_C).*sin(THETA_2).*(cos(THETA_C).*sin(PHI_C).* ...
sin(PHI_C).^2+(-1).*cos(PHI_C).^2.*sin(THETA_C)+sin(PHI_C).^2.*cos(THETA_C) ...
.^2.*sin(PHI_C).^2.*sin(THETA_2).^2+sin(THETA_C).^2+cos(PHI_C).^2.* ...
cos(THETA_C).^2.*sin(PHI_C).^2+sin(THETA_2).^2.*sin(THETA_C).^2+cos( ...
PSI_2).*cos(THETA_2).^2.*sin(PHI_C).*sin(PSI_2).*sin(2.*THETA_C).^(-1) ...
.*((-3)+cos(2.*PHI_C)+2.*cos(PHI_C).^2.*cos(2.*THETA_C)).*sin(THETA_2)+ ...
cos(THETA_2).*((-2).*cos(THETA_C).^2.*sin(2.*PHI_C).*sin(PSI_2)+2.*cos( ...
PHI_C).*cos(PSI_2).*sin(2.*THETA_C));
IJ_18_17=(cos(PSI_2).*cos(THETA_C).*sin(PHI_C)+sin(PSI_2).*sin(THETA_C)).*(cos( ...
PHI_C).*cos(THETA_C).*sin(THETA_2)+cos(THETA_2).*((-1).*cos(THETA_C)).* ...
sin(PHI_C).*sin(PSI_2)+cos(PSI_2).*sin(THETA_C)).*(cos(PHI_C).^2.*cos( ...
THETA_2).^2.*cos(THETA_C).^2+2.*cos(PHI_C).*cos(THETA_2).*cos(THETA_C).* ...
sin(THETA_2).*(cos(THETA_C).*sin(PHI_C).*sin(PSI_2)+(-1).*cos(PSI_2)).* ...
sin(THETA_C)+sin(PSI_2).^2.*cos(THETA_C).^2.*sin(PHI_C).^2.*sin( ...
THETA_2).^2+sin(THETA_C).^2+cos(PSI_2).^2.*cos(THETA_C).^2.*sin(PHI_C) ...
.^2+sin(THETA_2).^2.*sin(THETA_C).^2)+cos(PSI_2).^2.*cos(THETA_2).^2.*sin( ...
PHI_C).*sin(PSI_2).*sin(2.*THETA_C)).^(-1);
IJ_18_18 = 1;
IJ_18_19 = 0;
IJ_18_20 = 0;
IJ_18_21 = 0;
IJ_18_22 = 0;
IJ_18_23 = 0;
IJ_18_24 = 0;

IJ_19_1 = 0;
IJ_19_2 = 0;
IJ_19_3 = 0;
IJ_19_4 = 0;
IJ_19_5 = 0;
IJ_19_6 = 0;
IJ_19_7 = 1;
IJ_19_8=(cos(PHI_C).*cos(THETA_3).*sin(PSI_3)+sin(PHI_C).*sin(THETA_3)).*(cos( ...
PHI_C).*cos(THETA_C).*sin(THETA_3)+cos(THETA_3).*((-1).*cos(THETA_C)).* ...
sin(PHI_C).*sin(PSI_3)+cos(PSI_3).*sin(THETA_C)).*(cos(PSI_3).^2.*cos( ...
THETA_3).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_3).^2.*sin(PHI_C).* ...
sin(PSI_3).*sin(2.*THETA_3)+2.*cos(PSI_3).*cos(THETA_3).*cos(THETA_C).*( ...
cos(THETA_3).*sin(PHI_C).*sin(PSI_3)+(-1).*cos(PHI_C).*sin(THETA_3)).* ...
sin(THETA_C)+sin(PHI_C).^2.*sin(THETA_3).^2+cos(THETA_3).^2.*sin(PSI_3) ...
.^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*cos(THETA_3).^2.*sin(PSI_3).^2+ ...
sin(THETA_3).^2.*sin(THETA_C).^2)).^(-1);
IJ_19_9=(1/4).*(4.*cos(PSI_3).*cos(THETA_3).*cos(THETA_C).*((-1).*cos(THETA_3)).* ...
sin(PHI_C).*sin(PSI_3)+cos(PHI_C).*sin(THETA_3)+((-3)+cos(2.*PSI_3)+2.* ...
cos(PSI_3).^2.*cos(2.*THETA_3)).*sin(THETA_C)).*(cos(PSI_3).^2.*cos( ...
THETA_3).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C).* ...
sin(PSI_3).*sin(2.*THETA_3)+2.*cos(PSI_3).*cos(THETA_3).*cos(THETA_C).*( ...
cos(THETA_3).*sin(PHI_C).*sin(PSI_3)+(-1).*cos(PHI_C).*sin(THETA_3)).* ...
sin(THETA_C)+sin(PHI_C).^2.*sin(THETA_3).^2+cos(THETA_3).^2.*sin(PSI_3) ...
.^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*cos(THETA_3).^2.*sin(PSI_3).^2+ ...
sin(THETA_3).^2.*sin(THETA_C).^2)).^(-1);
IJ_19_10 = 0;
IJ_19_11 = 0;
IJ_19_12 = 0;
IJ_19_13 = 0;
IJ_19_14 = 0;
IJ_19_15 = 0;
IJ_19_16 = 0;
IJ_19_17 = 0;
IJ_19_18 = 0;
IJ_19_19=(1/2).*(2.*cos(PHI_C).*cos(THETA_3).^2.*cos(THETA_C)+cos(THETA_C).*sin( ...
PHI_C).*sin(PSI_3).*sin(2.*THETA_3)+(-2).*cos(PSI_3).*cos(THETA_3).*sin( ...
THETA_3).*sin(THETA_C)).*(cos(PSI_3).^2.*cos(THETA_3).^2.*cos(THETA_C) ...
.^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C).*sin(PSI_3).*sin(2.*THETA_3) ...
+2.*cos(PSI_3).*cos(THETA_3).*cos(THETA_C)).*(cos(THETA_3).*sin(PHI_C)).* ...
sin(PSI_3)+(-1).*cos(PHI_C).*sin(THETA_3)).*sin(THETA_C)+sin(PHI_C).^2.* ...
sin(THETA_3).^2.*cos(THETA_3).^2.*sin(PSI_3).^2.*sin(THETA_C).^2)+cos( ...
PHI_C).^2.*cos(THETA_3).^2.*sin(PSI_3).^2+sin(THETA_3).^2.*sin(THETA_C) ...
.^2)).^(-1);
IJ_19_20=(cos(PSI_3).*cos(THETA_C).*sin(PHI_C)+sin(PSI_3).*sin(THETA_C)).*(cos( ...
PSI_3).^2.*cos(THETA_3).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C) ...
.^2.*sin(PHI_C).*sin(PSI_3).*sin(2.*THETA_3)+2.*cos(PSI_3).*cos(THETA_3) ...
.*cos(THETA_C).*(cos(THETA_3).*sin(PHI_C).*sin(PSI_3)+(-1).*cos(PHI_C)).* ...
sin(THETA_3)).*sin(THETA_C)+sin(PHI_C).^2.*sin(THETA_3).^2+cos(THETA_3) ...
.^2.*sin(THETA_C).^2)+sin(PHI_C).^2.*sin(THETA_3).^2+cos(THETA_3) ...
.^2.*sin(THETA_C).^2)).^(-1);

```

```

.^2.*sin(PSI_3).^2.*sin(THETA_C).^2+cos(PHI_C).^2.*(cos(THETA_3).^2.* ...
sin(PSI_3).^2+sin(THETA_3).^2.*sin(THETA_C).^2).^(-1);
IJ_19_21 = 0;
IJ_19_22 = 0;
IJ_19_23 = 0;
IJ_19_24 = 0;

IJ_20_1 = 0;
IJ_20_2 = 0;
IJ_20_3 = 0;
IJ_20_4 = 0;
IJ_20_5 = 0;
IJ_20_6 = 0;
IJ_20_7 = 0;
IJ_20_8=(cos(PSI_3).*cos(THETA_3).*cos(THETA_C)+(cos(THETA_3).*sin(PHI_C).*sin( ...
PSI_3)+(-1).*cos(PHI_C).*sin(THETA_3)).*sin(THETA_C)).*(1+(-1).*(cos( ...
PHI_C).*cos(THETA_C).*sin(THETA_3)+cos(THETA_3)).*(-1).*cos(THETA_C).* ...
sin(PHI_C).*sin(PSI_3)+cos(PSI_3).*sin(THETA_C)).^2).^(-1/2);
IJ_20_9=(-1).*cos(THETA_C).*(cos(PHI_C).*cos(THETA_3).*sin(PSI_3)+sin(PHI_C).* ...
sin(THETA_3)).*(1+(-1).*(cos(PHI_C).*cos(THETA_C).*sin(THETA_3)+cos( ...
THETA_3)).*(-1).*cos(THETA_C).*sin(PHI_C).*sin(PSI_3)+cos(PSI_3).*sin( ...
THETA_C)).^2).^(-1/2);
IJ_20_10 = 0;
IJ_20_11 = 0;
IJ_20_12 = 0;
IJ_20_13 = 0;
IJ_20_14 = 0;
IJ_20_15 = 0;
IJ_20_16 = 0;
IJ_20_17 = 0;
IJ_20_18 = 0;
IJ_20_19=(-1).*cos(THETA_3).*(cos(PSI_3).*cos(THETA_C).*sin(PHI_C)+sin(PSI_3).* ...
sin(THETA_C)).*(1+(-1).*(cos(PHI_C).*cos(THETA_C).*sin(THETA_3)+cos( ...
THETA_3)).*(-1).*cos(THETA_C).*sin(PHI_C).*sin(PSI_3)+cos(PSI_3).*sin( ...
THETA_C)).^2).^(-1/2);
IJ_20_20=(cos(PHI_C).*cos(THETA_3).*cos(THETA_C)+sin(THETA_3)).*(cos(THETA_C).* ...
sin(PHI_C).*sin(PSI_3)+(-1).*cos(PSI_3).*sin(THETA_C)).*(1+(-1).*(cos( ...
PHI_C).*cos(THETA_C).*sin(THETA_3)+cos(THETA_3)).*(-1).*cos(THETA_C).* ...
sin(PHI_C).*sin(PSI_3)+cos(PSI_3).*sin(THETA_C)).^2).^(-1/2);
IJ_20_21 = 0;
IJ_20_22 = 0;
IJ_20_23 = 0;
IJ_20_24 = 0;

IJ_21_1 = 0;
IJ_21_2 = 0;
IJ_21_3 = 0;
IJ_21_4 = 0;
IJ_21_5 = 0;
IJ_21_6 = 0;
IJ_21_7 = 0;
IJ_21_8=(cos(PHI_C).*cos(THETA_3).*sin(PSI_3)+sin(PHI_C).*sin(THETA_3)).*(cos( ...
PHI_C).^2.*cos(THETA_3).^2.*cos(THETA_C).^2+2.*cos(PHI_C).*cos(THETA_3) ...
.*cos(THETA_C).*sin(THETA_3)).*(cos(THETA_C).*sin(PHI_C).*sin(PSI_3)+(-1) ...
.*cos(PSI_3).*sin(THETA_C)+sin(PSI_3).^2.*cos(THETA_C).^2.*sin(PHI_C) ...
.^2.*sin(THETA_3).^2+sin(THETA_C).^2)+cos(PSI_3).^2.*(cos(THETA_C).^2.* ...
sin(PHI_C).^2+sin(THETA_3).^2.*sin(THETA_C).^2)+cos(PSI_3).*cos(THETA_3) ...
.^2.*sin(PHI_C).*sin(PSI_3).*sin(2.*THETA_C)).^(-1);
IJ_21_9=(1/2).*(2.*cos(PSI_3).*cos(THETA_3).*cos(THETA_C).^2+(-2).*cos(PHI_C).* ...
cos(THETA_C).*sin(THETA_3)).*(cos(THETA_C)+cos(THETA_3)).*(cos(PHI_C).*sin( ...
PSI_3).*sin(2.*THETA_C)).*(cos(PHI_C).^2.*cos(THETA_3).^2.*cos(THETA_C) ...
.^2+2.*cos(PHI_C).*cos(THETA_3).*cos(THETA_C).*sin(THETA_3)).*(cos( ...
THETA_C).*sin(PHI_C).*sin(PSI_3)+(-1).*cos(PSI_3).*sin(THETA_C)+sin( ...
PSI_3).^2.*(cos(THETA_C).^2.*sin(PHI_C).^2.*sin(THETA_3).^2+sin(THETA_C) ...
.^2)+cos(PSI_3).^2.*(cos(THETA_C).^2.*sin(PHI_C).^2+sin(THETA_3).^2.* ...
sin(THETA_C).^2)+cos(PSI_3).*cos(THETA_3).^2.*sin(PHI_C).*sin(PSI_3).* ...
sin(2.*THETA_C)).^(-1);
IJ_21_10 = 0;
IJ_21_11 = 0;
IJ_21_12 = 0;
IJ_21_13 = 0;
IJ_21_14 = 0;
IJ_21_15 = 0;
IJ_21_16 = 0;
IJ_21_17 = 0;
IJ_21_18 = 0;
IJ_21_19=(1/4).*(cos(PHI_C).^2.*cos(THETA_3).^2.*cos(THETA_C).^2+2.*cos(PHI_C).* ...

```

```

cos(THETA_3).*cos(THETA_C).*sin(THETA_3).*(cos(THETA_C).*sin(PHI_C).* ...
sin(PSI_3)+(-1).*cos(PSI_3).*sin(THETA_C))+sin(PSI_3).^2.*(cos(THETA_C) ...
.^2.*sin(PHI_C).^2.*sin(THETA_3).^2+sin(THETA_C).^2+cos(PSI_3).^2.*{ ...
cos(THETA_C).^2.*sin(PHI_C).^2+sin(THETA_3).^2.*sin(THETA_C).^2+cos( ...
PSI_3).*cos(THETA_3).^2.*sin(PHI_C).*sin(PSI_3).*sin(2.*THETA_C).^(-1) ...
.*((-3)+cos(2.*PHI_C)+2.*cos(PHI_C).^2.*cos(2.*THETA_C)).*sin(THETA_3)+ ...
cos(THETA_3).*(-2).*cos(THETA_C).^2.*sin(2.*PHI_C).*sin(PSI_3)+2.*cos( ...
PHI_C).*cos(PSI_3).*sin(2.*THETA_C));
IJ_21_20=(cos(PSI_3).*cos(THETA_C).*sin(PHI_C)+sin(PSI_3).*sin(THETA_C)).*(cos( ...
PHI_C).*cos(THETA_C).*sin(THETA_3)+cos(THETA_3).*(-1).*cos(THETA_C).* ...
sin(PHI_C).*sin(PSI_3)+cos(PSI_3).*sin(THETA_C)).*(cos(PHI_C).^2.*cos( ...
THETA_3).^2.*cos(THETA_C).^2+2.*cos(PHI_C).*cos(THETA_3).*cos(THETA_C).* ...
sin(THETA_3).*(cos(THETA_C).*sin(PHI_C).^2.*sin(PSI_3)+(-1).*cos(PSI_3).* ...
sin(THETA_C))+sin(PSI_3).^2.*(cos(THETA_C).^2.*sin(PHI_C).^2.*sin( ...
THETA_3).^2+sin(THETA_C).^2+cos(PSI_3).^2.*(cos(THETA_C).^2.*sin(PHI_C) ...
.^2+sin(THETA_3).^2.*sin(THETA_C).^2)+cos(PSI_3).*cos(THETA_3).^2.*sin( ...
PHI_C).*sin(PSI_3).*sin(2.*THETA_C)).^(-1);
IJ_21_21 = 1;
IJ_21_22 = 0;
IJ_21_23 = 0;
IJ_21_24 = 0;

IJ_22_1 = 0;
IJ_22_2 = 0;
IJ_22_3 = 0;
IJ_22_4 = 0;
IJ_22_5 = 0;
IJ_22_6 = 0;
IJ_22_7 = 1;
IJ_22_8=(cos(PHI_C).*cos(THETA_4).*sin(PSI_4)+sin(PHI_C).*sin(THETA_4)).*(cos( ...
PHI_C).*cos(THETA_C).*sin(THETA_4)+cos(THETA_4).*(-1).*cos(THETA_C).* ...
sin(PHI_C).*sin(PSI_4)+cos(PSI_4).*sin(THETA_C)).*(cos(PSI_4).^2.*cos( ...
THETA_4).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C).* ...
sin(PSI_4).*sin(2.*THETA_4)+2.*cos(PSI_4).*cos(THETA_4).*cos(THETA_C).*( ...
cos(THETA_4).*sin(PHI_C).*sin(PSI_4)+(-1).*cos(PHI_C).*sin(THETA_4)).* ...
sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_4).^2+cos(THETA_4).^2.*sin(PSI_4) ...
.^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_4).^2.*sin(PSI_4).^2+ ...
sin(THETA_4).^2.*sin(THETA_C).^2).^(-1);
IJ_22_9=(1/4).*(4.*cos(PSI_4).*cos(THETA_4).*cos(THETA_C).*(-1).*cos(THETA_4).* ...
sin(PHI_C).*sin(PSI_4)+cos(PHI_C).*sin(THETA_4))+((-3)+cos(2.*PSI_4)+2.* ...
cos(PSI_4).^2.*cos(2.*THETA_4)).*sin(THETA_C)).*(cos(PSI_4).^2.*cos( ...
THETA_4).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C).* ...
sin(PSI_4).*sin(2.*THETA_4)+2.*cos(PSI_4).*cos(THETA_4).*cos(THETA_C).*( ...
cos(THETA_4).*sin(PHI_C).*sin(PSI_4)+(-1).*cos(PHI_C).*sin(THETA_4)).* ...
sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_4).^2+cos(THETA_4).^2.*sin(PSI_4) ...
.^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_4).^2.*sin(PSI_4).^2+ ...
sin(THETA_4).^2.*sin(THETA_C).^2).^(-1);
IJ_22_10 = 0;
IJ_22_11 = 0;
IJ_22_12 = 0;
IJ_22_13 = 0;
IJ_22_14 = 0;
IJ_22_15 = 0;
IJ_22_16 = 0;
IJ_22_17 = 0;
IJ_22_18 = 0;
IJ_22_19 = 0;
IJ_22_20 = 0;
IJ_22_21 = 0;
IJ_22_22=(1/2).*(2.*cos(PHI_C).*cos(THETA_4).^2.*cos(THETA_C)+cos(THETA_C).*sin( ...
PHI_C).*sin(PSI_4).*sin(2.*THETA_4)+(-2).*cos(PSI_4).*cos(THETA_4).*sin( ...
THETA_4).*sin(THETA_C)).*(cos(PSI_4).^2.*cos(THETA_4).^2.*cos(THETA_C) ...
.^2+cos(PHI_C).*cos(THETA_C).^2.*sin(PHI_C).*sin(PSI_4).*sin(2.*THETA_4) ...
+2.*cos(PSI_4).*cos(THETA_4).*cos(THETA_C).*(cos(THETA_4).*sin(PHI_C).* ...
sin(PSI_4)+(-1).*cos(PHI_C).*sin(THETA_4)).*sin(THETA_C)+sin(PHI_C).^2.* ...
(sin(THETA_4).^2+cos(THETA_4).^2.*sin(PSI_4).^2.*sin(THETA_C).^2)+cos( ...
PHI_C).^2.*(cos(THETA_4).^2.*sin(PSI_4).^2+sin(THETA_4).^2.*sin(THETA_C) ...
.^2).^(-1);
IJ_22_23=(cos(PSI_4).*cos(THETA_C).*sin(PHI_C)+sin(PSI_4).*sin(THETA_C)).*(cos( ...
PSI_4).^2.*cos(THETA_4).^2.*cos(THETA_C).^2+cos(PHI_C).*cos(THETA_C) ...
.^2.*sin(PHI_C).*sin(PSI_4).*sin(2.*THETA_4)+2.*cos(PSI_4).*cos(THETA_4) ...
.*cos(THETA_C).*(cos(THETA_4).*sin(PHI_C).^2.*sin(PSI_4)+(-1).*cos(PHI_C).* ...
sin(THETA_4)).*sin(THETA_C)+sin(PHI_C).^2.*(sin(THETA_4).^2+cos(THETA_4) ...
.^2.*sin(PSI_4).^2.*sin(THETA_C).^2)+cos(PHI_C).^2.*(cos(THETA_4).^2.* ...
sin(PSI_4).^2+sin(THETA_4).^2.*sin(THETA_C).^2).^(-1);
IJ_22_24 = 0;

```

```

IJ_23_1 = 0;
IJ_23_2 = 0;
IJ_23_3 = 0;
IJ_23_4 = 0;
IJ_23_5 = 0;
IJ_23_6 = 0;
IJ_23_7 = 0;
IJ_23_8=(cos(PHI_4).*cos(THETA_4).*cos(THETA_C)+(cos(THETA_4).*sin(PHI_C).*sin(...
    PSI_4)+(-1).*cos(PHI_C).*sin(THETA_4)).*sin(THETA_C)).*(1+(-1).*cos(...
    PHI_C).*cos(THETA_C).*sin(THETA_4)+cos(THETA_4).*(-1).*cos(THETA_C).*...
    sin(PHI_C).*sin(PSI_4)+cos(PSI_4).*sin(THETA_C)).^2).^(-1/2);
IJ_23_9=(-1).*cos(THETA_C).*cos(PHI_C).*cos(THETA_4).*sin(PSI_4)+sin(PHI_C).*...
    sin(THETA_4)).*(1+(-1).*cos(PHI_C).*cos(THETA_C).*sin(THETA_4)+cos(...
    THETA_4)).*(-1).*cos(THETA_C).*sin(PHI_C).*sin(PSI_4)+cos(PSI_4).*sin(...
    THETA_C)).^2).^(-1/2);
IJ_23_10 = 0;
IJ_23_11 = 0;
IJ_23_12 = 0;
IJ_23_13 = 0;
IJ_23_14 = 0;
IJ_23_15 = 0;
IJ_23_16 = 0;
IJ_23_17 = 0;
IJ_23_18 = 0;
IJ_23_19 = 0;
IJ_23_20 = 0;
IJ_23_21 = 0;
IJ_23_22=(-1).*cos(THETA_4).*cos(PSI_4).*cos(THETA_C).*sin(PHI_C)+sin(PSI_4).*...
    sin(THETA_C)).*(1+(-1).*cos(PHI_C).*cos(THETA_C).*sin(THETA_4)+cos(...
    THETA_4)).*(-1).*cos(THETA_C).*sin(PHI_C).*sin(PSI_4)+cos(PSI_4).*sin(...
    THETA_C)).^2).^(-1/2);
IJ_23_23=(cos(PHI_C).*cos(THETA_4).*cos(THETA_C)+sin(THETA_4)).*(cos(THETA_C).*...
    sin(PHI_C).*sin(PSI_4)+(-1).*cos(PSI_4).*sin(THETA_C)).*(1+(-1).*cos(...
    PHI_C).*cos(THETA_C).*sin(THETA_4)+cos(THETA_4)).*(-1).*cos(THETA_C).*...
    sin(PHI_C).*sin(PSI_4)+cos(PSI_4).*sin(THETA_C)).^2).^(-1/2);
IJ_23_24 = 0;

IJ_24_1 = 0;
IJ_24_2 = 0;
IJ_24_3 = 0;
IJ_24_4 = 0;
IJ_24_5 = 0;
IJ_24_6 = 0;
IJ_24_7 = 0;
IJ_24_8=(cos(PHI_C).*cos(THETA_4).*sin(PSI_4)+sin(PHI_C).*sin(THETA_4)).*(cos(...
    PHI_C).^2.*cos(THETA_4).^2.*cos(THETA_C).^2+2.*cos(PHI_C).*cos(THETA_4)...
    .*cos(THETA_C).*sin(THETA_4)).*(cos(THETA_C).*sin(PHI_C).*sin(PSI_4)+(-1)...
    .*cos(PSI_4).*sin(THETA_C)+sin(PSI_4).^2.*(cos(THETA_C).^2.*sin(PHI_C)...
    .*sin(THETA_4).^2+sin(THETA_C).^2)+cos(PSI_4).^2.*(cos(THETA_C).^2.*...
    sin(PHI_C).^2+sin(THETA_4).^2.*sin(THETA_C).^2)+cos(PSI_4).*cos(THETA_4)...
    .*sin(PHI_C).*sin(PSI_4).*sin(2.*THETA_C)).^(-1);
IJ_24_9=(1/2).*sin(2.*cos(PSI_4).*cos(THETA_4).*cos(THETA_C).^2+(-2).*cos(PHI_C).*...
    cos(THETA_C).*sin(THETA_4)).*(cos(THETA_C)+cos(THETA_4)).*(sin(PHI_C).*sin(...
    PSI_4).*sin(2.*THETA_C)).*(cos(PHI_C).^2.*cos(THETA_4).^2.*cos(THETA_C)...
    .*sin(2.*cos(PHI_C).*cos(THETA_4).*cos(THETA_C).*sin(THETA_4)).*(cos(...
    THETA_C).*sin(PHI_C).*sin(PSI_4)+(-1).*cos(PSI_4).*sin(THETA_C)+sin(...
    PSI_4).^2.*(cos(THETA_C).^2.*sin(PHI_C).^2.*sin(THETA_4).^2+sin(THETA_C)...
    .*sin(THETA_4).^2)+cos(PSI_4).^2.*(cos(THETA_C).^2.*sin(PHI_C).^2+sin(THETA_4).^2.*...
    sin(THETA_C).^2)+cos(PSI_4).*cos(THETA_4).^2.*sin(PHI_C).*sin(PSI_4)).*...
    sin(2.*THETA_C)).^(-1);
IJ_24_10 = 0;
IJ_24_11 = 0;
IJ_24_12 = 0;
IJ_24_13 = 0;
IJ_24_14 = 0;
IJ_24_15 = 0;
IJ_24_16 = 0;
IJ_24_17 = 0;
IJ_24_18 = 0;
IJ_24_19 = 0;
IJ_24_20 = 0;
IJ_24_21 = 0;
IJ_24_22=(1/4).*cos(PHI_C).^2.*cos(THETA_4).^2.*cos(THETA_C).^2+2.*cos(PHI_C).*...
    cos(THETA_4).*cos(THETA_C).*sin(THETA_4)).*(cos(THETA_C).*sin(PHI_C).*...
    sin(PSI_4)+(-1).*cos(PSI_4).*sin(THETA_C)+sin(PSI_4).^2.*(cos(THETA_C)...
    .*sin(PHI_C).^2.*sin(THETA_4).^2+sin(THETA_C).^2)+cos(PSI_4).^2.*...
    cos(THETA_C).^2.*sin(PHI_C).^2+sin(THETA_4).^2.*sin(THETA_C).^2)+cos(...

```

```

PSI_4).*cos(THETA_4).^2.*sin(PHI_C).*sin(PSI_4).*sin(2.*THETA_C)).^(-1) ...
.*(7*(-3)+cos(2.*PHI_C)+2.*cos(PHI_C).^2.*cos(2.*THETA_C)).*sin(THETA_4)+ ...
cos(THETA_4).*( (-2).*cos(THETA_C).^2.*sin(2.*PHI_C).*sin(PSI_4)+2.*cos( ...
PHI_C).*cos(PSI_4).*sin(2.*THETA_C)));
IJ_24_23=(cos(PSI_4).*cos(THETA_C).*sin(PHI_C)+sin(PSI_4).*sin(THETA_C)).*(cos( ...
PHI_C).*cos(THETA_C).*sin(THETA_4)+cos(THETA_4).*( (-1).*cos(THETA_C)).* ...
sin(PHI_C).*sin(PSI_4)+cos(PSI_4).*sin(THETA_C))).*(cos(PHI_C).^2.*cos( ...
THETA_4).^2.*cos(THETA_C).^2+2.*cos(PHI_C).*cos(THETA_4).*cos(THETA_C).* ...
sin(THETA_4).*(cos(THETA_C).*sin(PHI_C).*sin(PSI_4)+(-1).*cos(PSI_4)).* ...
sin(THETA_C))+sin(PSI_4).^2.*cos(THETA_C).^2.*sin(PHI_C).^2.*sin( ...
THETA_4).^2+sin(THETA_C).^2)+cos(PSI_4).^2.*cos(THETA_C).^2.*sin(PHI_C) ...
.^2+sin(THETA_4).^2.*sin(THETA_C).^2)+cos(PSI_4).*cos(THETA_4).^2.*sin( ...
PHI_C).*sin(PSI_4).*sin(2.*THETA_C)).^(-1);
IJ_24_24 = 1;

% Form Inverse Jacobian Matrix
IJ = [...
IJ_1_1 IJ_1_2 IJ_1_3 IJ_1_4 IJ_1_5 IJ_1_6 IJ_1_7 IJ_1_8 IJ_1_9 IJ_1_10 IJ_1_11 I
IJ_1_13 IJ_1_14 IJ_1_15 IJ_1_16 IJ_1_17 IJ_1_18 IJ_1_19 IJ_1_20 IJ_1_21 IJ_1_22 IJ_1_23 I
IJ_2_1 IJ_2_2 IJ_2_3 IJ_2_4 IJ_2_5 IJ_2_6 IJ_2_7 IJ_2_8 IJ_2_9 IJ_2_10 IJ_2_11 I
IJ_2_13 IJ_2_14 IJ_2_15 IJ_2_16 IJ_2_17 IJ_2_18 IJ_2_19 IJ_2_20 IJ_2_21 IJ_2_22 IJ_2_23 I
IJ_3_1 IJ_3_2 IJ_3_3 IJ_3_4 IJ_3_5 IJ_3_6 IJ_3_7 IJ_3_8 IJ_3_9 IJ_3_10 IJ_3_11 I
IJ_3_13 IJ_3_14 IJ_3_15 IJ_3_16 IJ_3_17 IJ_3_18 IJ_3_19 IJ_3_20 IJ_3_21 IJ_3_22 IJ_3_23 I
IJ_4_1 IJ_4_2 IJ_4_3 IJ_4_4 IJ_4_5 IJ_4_6 IJ_4_7 IJ_4_8 IJ_4_9 IJ_4_10 IJ_4_11 I
IJ_4_13 IJ_4_14 IJ_4_15 IJ_4_16 IJ_4_17 IJ_4_18 IJ_4_19 IJ_4_20 IJ_4_21 IJ_4_22 IJ_4_23 I
IJ_5_1 IJ_5_2 IJ_5_3 IJ_5_4 IJ_5_5 IJ_5_6 IJ_5_7 IJ_5_8 IJ_5_9 IJ_5_10 IJ_5_11 I
IJ_5_13 IJ_5_14 IJ_5_15 IJ_5_16 IJ_5_17 IJ_5_18 IJ_5_19 IJ_5_20 IJ_5_21 IJ_5_22 IJ_5_23 I
IJ_6_1 IJ_6_2 IJ_6_3 IJ_6_4 IJ_6_5 IJ_6_6 IJ_6_7 IJ_6_8 IJ_6_9 IJ_6_10 IJ_6_11 I
IJ_6_13 IJ_6_14 IJ_6_15 IJ_6_16 IJ_6_17 IJ_6_18 IJ_6_19 IJ_6_20 IJ_6_21 IJ_6_22 IJ_6_23 I
IJ_7_1 IJ_7_2 IJ_7_3 IJ_7_4 IJ_7_5 IJ_7_6 IJ_7_7 IJ_7_8 IJ_7_9 IJ_7_10 IJ_7_11 I
IJ_7_13 IJ_7_14 IJ_7_15 IJ_7_16 IJ_7_17 IJ_7_18 IJ_7_19 IJ_7_20 IJ_7_21 IJ_7_22 IJ_7_23 I
IJ_8_1 IJ_8_2 IJ_8_3 IJ_8_4 IJ_8_5 IJ_8_6 IJ_8_7 IJ_8_8 IJ_8_9 IJ_8_10 IJ_8_11 I
IJ_8_13 IJ_8_14 IJ_8_15 IJ_8_16 IJ_8_17 IJ_8_18 IJ_8_19 IJ_8_20 IJ_8_21 IJ_8_22 IJ_8_23 I
IJ_9_1 IJ_9_2 IJ_9_3 IJ_9_4 IJ_9_5 IJ_9_6 IJ_9_7 IJ_9_8 IJ_9_9 IJ_9_10 IJ_9_11 I
IJ_9_13 IJ_9_14 IJ_9_15 IJ_9_16 IJ_9_17 IJ_9_18 IJ_9_19 IJ_9_20 IJ_9_21 IJ_9_22 IJ_9_23 I
IJ_10_1 IJ_10_2 IJ_10_3 IJ_10_4 IJ_10_5 IJ_10_6 IJ_10_7 IJ_10_8 IJ_10_9 IJ_10_10 I
IJ_10_13 IJ_10_14 IJ_10_15 IJ_10_16 IJ_10_17 IJ_10_18 IJ_10_19 IJ_10_20 IJ_10_21 IJ_10_22 I
IJ_11_1 IJ_11_2 IJ_11_3 IJ_11_4 IJ_11_5 IJ_11_6 IJ_11_7 IJ_11_8 IJ_11_9 IJ_11_10 I
IJ_11_13 IJ_11_14 IJ_11_15 IJ_11_16 IJ_11_17 IJ_11_18 IJ_11_19 IJ_11_20 IJ_11_21 IJ_11_22 I
IJ_12_1 IJ_12_2 IJ_12_3 IJ_12_4 IJ_12_5 IJ_12_6 IJ_12_7 IJ_12_8 IJ_12_9 IJ_12_10 I
IJ_12_13 IJ_12_14 IJ_12_15 IJ_12_16 IJ_12_17 IJ_12_18 IJ_12_19 IJ_12_20 IJ_12_21 IJ_12_22 I
IJ_13_1 IJ_13_2 IJ_13_3 IJ_13_4 IJ_13_5 IJ_13_6 IJ_13_7 IJ_13_8 IJ_13_9 IJ_13_10 I
IJ_13_13 IJ_13_14 IJ_13_15 IJ_13_16 IJ_13_17 IJ_13_18 IJ_13_19 IJ_13_20 IJ_13_21 IJ_13_22 I
IJ_14_1 IJ_14_2 IJ_14_3 IJ_14_4 IJ_14_5 IJ_14_6 IJ_14_7 IJ_14_8 IJ_14_9 IJ_14_10 I
IJ_14_13 IJ_14_14 IJ_14_15 IJ_14_16 IJ_14_17 IJ_14_18 IJ_14_19 IJ_14_20 IJ_14_21 IJ_14_22 I
IJ_15_1 IJ_15_2 IJ_15_3 IJ_15_4 IJ_15_5 IJ_15_6 IJ_15_7 IJ_15_8 IJ_15_9 IJ_15_10 I
IJ_15_13 IJ_15_14 IJ_15_15 IJ_15_16 IJ_15_17 IJ_15_18 IJ_15_19 IJ_15_20 IJ_15_21 IJ_15_22 I
IJ_16_1 IJ_16_2 IJ_16_3 IJ_16_4 IJ_16_5 IJ_16_6 IJ_16_7 IJ_16_8 IJ_16_9 IJ_16_10 I
IJ_16_13 IJ_16_14 IJ_16_15 IJ_16_16 IJ_16_17 IJ_16_18 IJ_16_19 IJ_16_20 IJ_16_21 IJ_16_22 I
IJ_17_1 IJ_17_2 IJ_17_3 IJ_17_4 IJ_17_5 IJ_17_6 IJ_17_7 IJ_17_8 IJ_17_9 IJ_17_10 I
IJ_17_13 IJ_17_14 IJ_17_15 IJ_17_16 IJ_17_17 IJ_17_18 IJ_17_19 IJ_17_20 IJ_17_21 IJ_17_22 I
IJ_18_1 IJ_18_2 IJ_18_3 IJ_18_4 IJ_18_5 IJ_18_6 IJ_18_7 IJ_18_8 IJ_18_9 IJ_18_10 I
IJ_18_13 IJ_18_14 IJ_18_15 IJ_18_16 IJ_18_17 IJ_18_18 IJ_18_19 IJ_18_20 IJ_18_21 IJ_18_22 I
IJ_19_1 IJ_19_2 IJ_19_3 IJ_19_4 IJ_19_5 IJ_19_6 IJ_19_7 IJ_19_8 IJ_19_9 IJ_19_10 I
IJ_19_13 IJ_19_14 IJ_19_15 IJ_19_16 IJ_19_17 IJ_19_18 IJ_19_19 IJ_19_20 IJ_19_21 IJ_19_22 I
IJ_20_1 IJ_20_2 IJ_20_3 IJ_20_4 IJ_20_5 IJ_20_6 IJ_20_7 IJ_20_8 IJ_20_9 IJ_20_10 I
IJ_20_13 IJ_20_14 IJ_20_15 IJ_20_16 IJ_20_17 IJ_20_18 IJ_20_19 IJ_20_20 IJ_20_21 IJ_20_22 I

```

```

IJ_21_1 IJ_21_2 IJ_21_3 IJ_21_4 IJ_21_5 IJ_21_6 IJ_21_7 IJ_21_8 IJ_21_9 IJ_21_10
IJ_21_13 IJ_21_14 IJ_21_15 IJ_21_16 IJ_21_17 IJ_21_18 IJ_21_19 IJ_21_20 IJ_21_21 IJ_21_22

IJ_22_1 IJ_22_2 IJ_22_3 IJ_22_4 IJ_22_5 IJ_22_6 IJ_22_7 IJ_22_8 IJ_22_9 IJ_22_10
IJ_22_13 IJ_22_14 IJ_22_15 IJ_22_16 IJ_22_17 IJ_22_18 IJ_22_19 IJ_22_20 IJ_22_21 IJ_22_22

IJ_23_1 IJ_23_2 IJ_23_3 IJ_23_4 IJ_23_5 IJ_23_6 IJ_23_7 IJ_23_8 IJ_23_9 IJ_23_10
IJ_23_13 IJ_23_14 IJ_23_15 IJ_23_16 IJ_23_17 IJ_23_18 IJ_23_19 IJ_23_20 IJ_23_21 IJ_23_22

IJ_24_1 IJ_24_2 IJ_24_3 IJ_24_4 IJ_24_5 IJ_24_6 IJ_24_7 IJ_24_8 IJ_24_9 IJ_24_10
IJ_24_13 IJ_24_14 IJ_24_15 IJ_24_16 IJ_24_17 IJ_24_18 IJ_24_19 IJ_24_20 IJ_24_21 IJ_24_22
];

% Compute Commanded Cluster-Space Velocities
V_com_RS = IJ*V_com_CS;

% Decompose V_com_RS into Commanded Robot-n Velocity Variables (wrt G)
x1d_com = V_com_RS(1,1); x2d_com = V_com_RS(2,1); x3d_com = V_com_RS(3,1); x4d_com = V_com_RS(4,1);
y1d_com = V_com_RS(5,1); y2d_com = V_com_RS(6,1); y3d_com = V_com_RS(7,1); y4d_com = V_com_RS(8,1);
z1d_com = V_com_RS(9,1); z2d_com = V_com_RS(10,1); z3d_com = V_com_RS(11,1); z4d_com = V_com_RS(12,1);
psi_1d_com = V_com_RS(13,1); theta_1d_com = V_com_RS(14,1); phi_1d_com = V_com_RS(15,1);
psi_2d_com = V_com_RS(16,1); theta_2d_com = V_com_RS(17,1); phi_2d_com = V_com_RS(18,1);
psi_3d_com = V_com_RS(19,1); theta_3d_com = V_com_RS(20,1); phi_3d_com = V_com_RS(21,1);
psi_4d_com = V_com_RS(22,1); theta_4d_com = V_com_RS(23,1); phi_4d_com = V_com_RS(24,1);

V_com_RS_G = ...
[ x1d_com ; x2d_com ; x3d_com ; x4d_com ;...
  y1d_com ; y2d_com ; y3d_com ; y4d_com ;...
  z1d_com ; z2d_com ; z3d_com ; z4d_com ;...
  psi_1d_com ; theta_1d_com ; phi_1d_com ;...
  psi_2d_com ; theta_2d_com ; phi_2d_com ;...
  psi_3d_com ; theta_3d_com ; phi_3d_com ;...
  psi_4d_com ; theta_4d_com ; phi_4d_com ];

%=====  

% Cluster Frame (C) Orientation expressed in G  

G_R_P = [ cos(PHI_C) -sin(PHI_C) 0 ;...
          sin(PHI_C) cos(PHI_C) 0 ;...
          0 0 1 ];
P_R_DP = [ cos(THETA_C) 0 sin(THETA_C);...
           0 1 0 ;...
           -sin(THETA_C) 0 cos(THETA_C)];
DP_R_C = [ 1 0 0 ;...
           0 cos(PHI_C) -sin(PHI_C) ;...
           0 sin(PHI_C) cos(PHI_C) ];
G_R_C = G_R_P*P_R_DP*DP_R_C; % [ Xhat_C Yhat_C Zhat_C ]
%=====  

%=====  

%=====  

% Robot-1 Frame (R1) Orientation expressed in C  

C_r_R1P =...
[ cos(PHI_1) -sin(PHI_1) 0 ;...
  sin(PHI_1) cos(PHI_1) 0 ;...
  0 0 1 ];
R1P_r_R1PP =...
[ cos(THETA_1) 0 sin(THETA_1);...
  0 1 0 ;...
  -sin(THETA_1) 0 cos(THETA_1)];
R1PP_r_R1 =...
[ 1 0 0 ;...
  0 cos(PHI_1) -sin(PHI_1) ;...
  0 sin(PHI_1) cos(PHI_1) ];
C_r_R1 = C_r_R1P*R1P_r_R1PP*R1PP_r_R1; % [ C_Xhat_1 C_Yhat_1 C_Zhat_1 ]

% Robot-1 Frame (R1) Orientation expressed in G  

G_r_R1 = G_R_C*C_r_R1; % [ Xhat_1 Yhat_1 Zhat_1 ]
R1_r_G = transpose(G_r_R1);
%=====  

%=====  

%=====  

%=====  

% Robot-2 Orientation  

% Robot-2 Frame (R2) Orientation expressed in C  

C_r_R2P =...
[ cos(PHI_2) -sin(PHI_2) 0 ;...
  sin(PHI_2) cos(PHI_2) 0 ;...
  0 0 1 ];

```

```

R2P_r_R2PP =...
    [ cos(THETA_2)      0      sin(THETA_2);...
      0                1      0      ;...
    -sin(THETA_2)    0      cos(THETA_2)];
R2PP_r_R2 =...
    [      1      0      0      ;...
      0      cos(PHI_2) -sin(PHI_2) ;...
      0      sin(PHI_2)  cos(PHI_2)  1];
C_r_R2 = C_r_R2P*R2P_r_R2PP*R2PP_r_R2; % [ C_Xhat_2 C_Yhat_2 C_Zhat_2 ]

% Robot-2 Frame (R2) Orientation expressed in G
G_r_R2 = G_R_C*C_r_R2; % [ Xhat_2 Yhat_2 Zhat_2 ]
R2_r_G = transpose(G_r_R2);
%===== Robot-2 Orientation =====
%=====

%===== Robot-3 Orientation =====
% Robot-3 Frame (R3) Orientation expressed in C
C_r_R3P =...
    [ cos(Psi_3)  -sin(Psi_3)  0      ;...
      sin(Psi_3)  cos(Psi_3)  0      ;...
      0          0          1      ];
R3P_r_R3PP =...
    [ cos(THETA_3)  0      sin(THETA_3);...
      0            1      0      ;...
    -sin(THETA_3)  0      cos(THETA_3)];
R3PP_r_R3 =...
    [      1      0      0      ;...
      0      cos(PHI_3) -sin(PHI_3) ;...
      0      sin(PHI_3)  cos(PHI_3)  1];
C_r_R3 = C_r_R3P*R3P_r_R3PP*R3PP_r_R3; % [ C_Xhat_3 C_Yhat_3 C_Zhat_3 ]

% Robot-3 Frame (R3) Orientation expressed in G
G_r_R3 = G_R_C*C_r_R3; % [ Xhat_3 Yhat_3 Zhat_3 ]
R3_r_G = transpose(G_r_R3);
%===== Robot-3 Orientation =====
%=====

%===== Robot-4 Orientation =====
% Robot-4 Frame (R4) Orientation expressed in C
C_r_R4P =...
    [ cos(Psi_4)  -sin(Psi_4)  0      ;...
      sin(Psi_4)  cos(Psi_4)  0      ;...
      0          0          1      ];
R4P_r_R4PP =...
    [ cos(THETA_4)  0      sin(THETA_4);...
      0            1      0      ;...
    -sin(THETA_4)  0      cos(THETA_4)];
R4PP_r_R4 =...
    [      1      0      0      ;...
      0      cos(PHI_4) -sin(PHI_4) ;...
      0      sin(PHI_4)  cos(PHI_4)  1];
C_r_R4 = C_r_R4P*R4P_r_R4PP*R4PP_r_R4; % [ C_Xhat_4 C_Yhat_4 C_Zhat_4 ]

% Robot-4 Frame (R4) Orientation expressed in G
G_r_R4 = G_R_C*C_r_R4; % [ Xhat_4 Yhat_4 Zhat_4 ]
R4_r_G = transpose(G_r_R4);
%===== Robot-4 Orientation =====
%=====

% Transform V_com R1 (wrt G) to V_com R1 R1 (wrt R1)
V_com_R1_R1_top = R1_r_G*[ x1d_com ; y1d_com ; z1d_com];
V_com_R1_R1_bot = R1_r_G*[psi_1d_com ; theta_1d_com ; phi_1d_com];
V_com_R1_R1 = vertcat(V_com_R1_R1_top,V_com_R1_R1_bot);

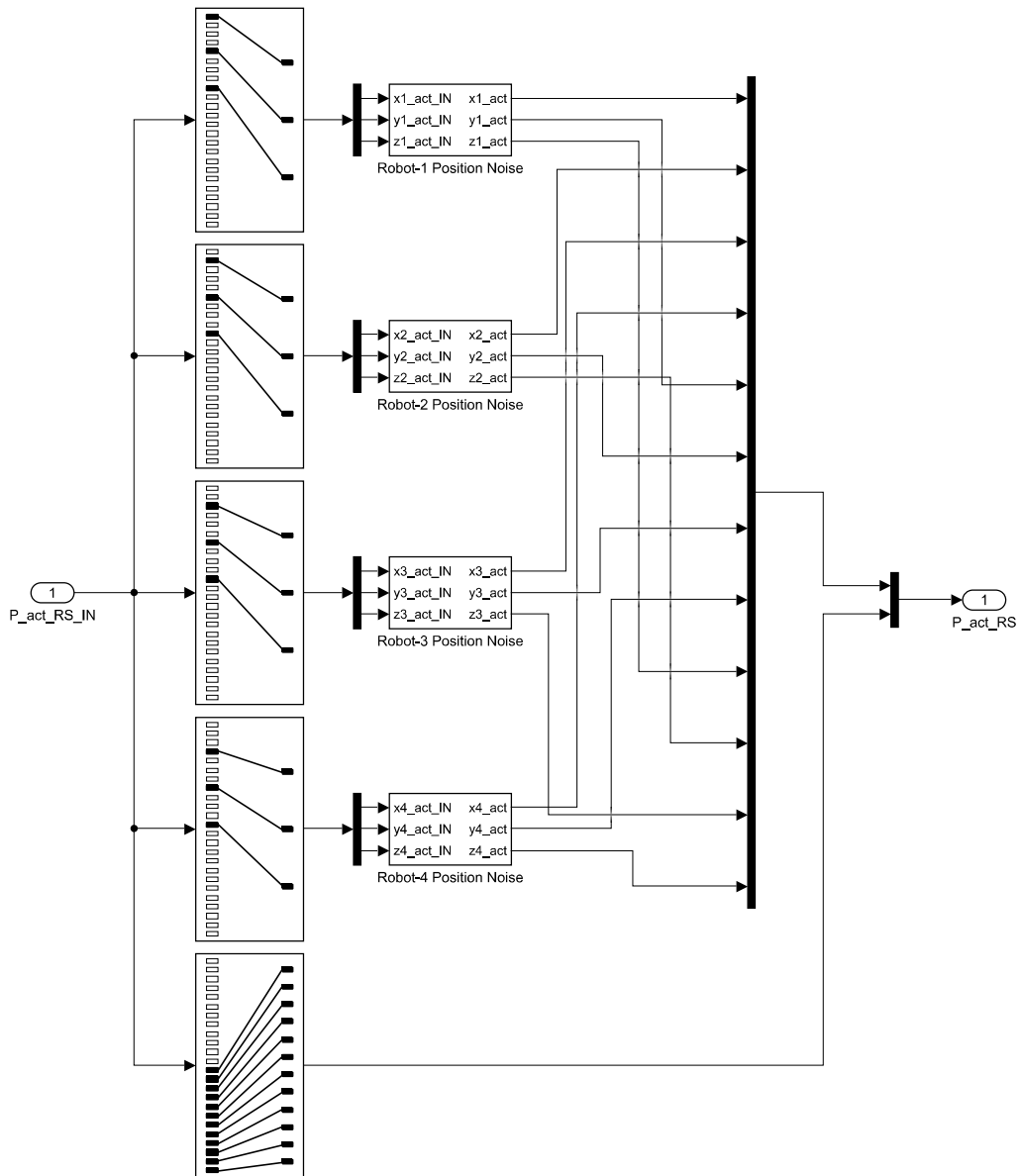
% Transform V_com R2 (wrt G) to V_com R2 R2 (wrt R2)
V_com_R2_R2_top = R2_r_G*[ x2d_com ; y2d_com ; z2d_com];
V_com_R2_R2_bot = R2_r_G*[psi_2d_com ; theta_2d_com ; phi_2d_com];
V_com_R2_R2 = vertcat(V_com_R2_R2_top,V_com_R2_R2_bot);

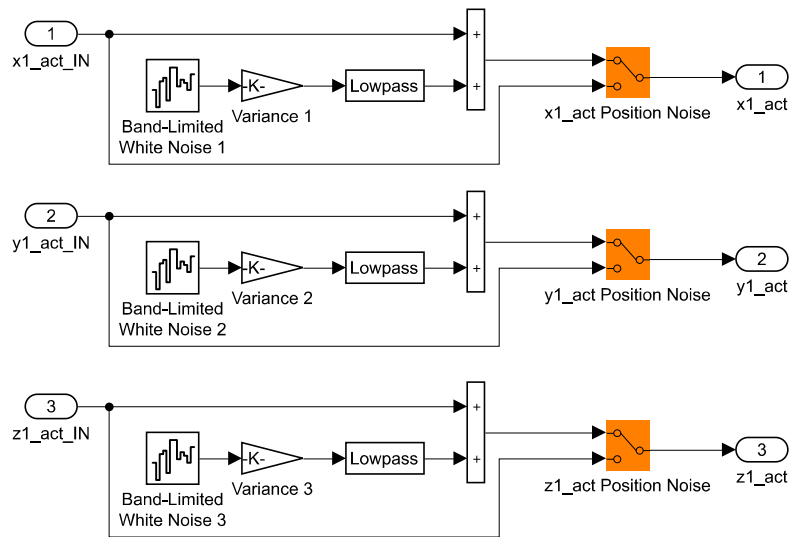
% Transform V_com R3 (wrt G) to V_com R3 R3 (wrt R3)
V_com_R3_R3_top = R3_r_G*[ x3d_com ; y3d_com ; z3d_com];
V_com_R3_R3_bot = R3_r_G*[psi_3d_com ; theta_3d_com ; phi_3d_com];
V_com_R3_R3 = vertcat(V_com_R3_R3_top,V_com_R3_R3_bot);

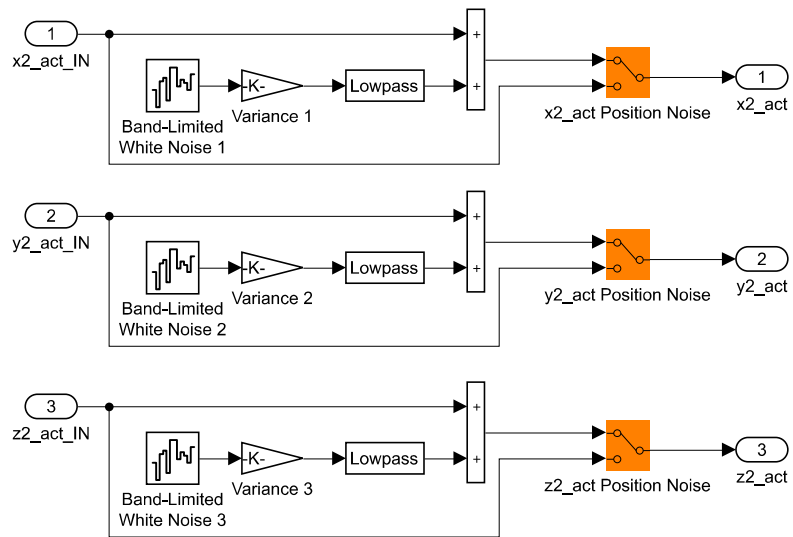
```

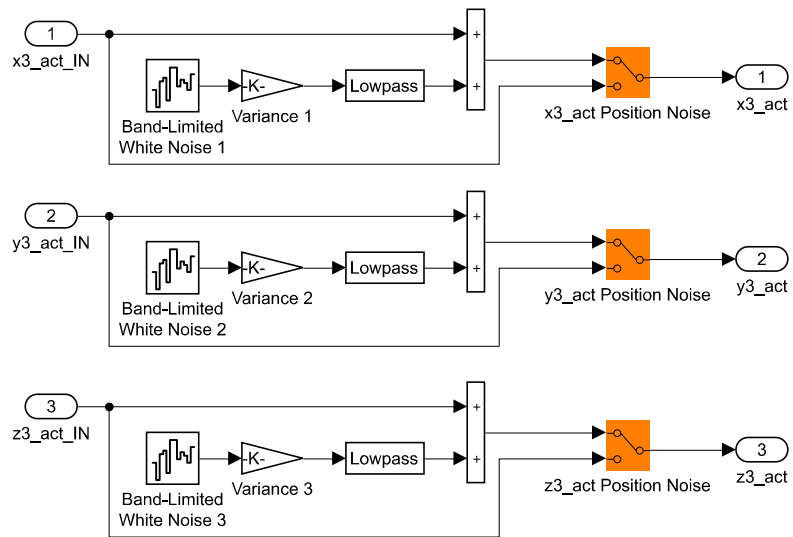


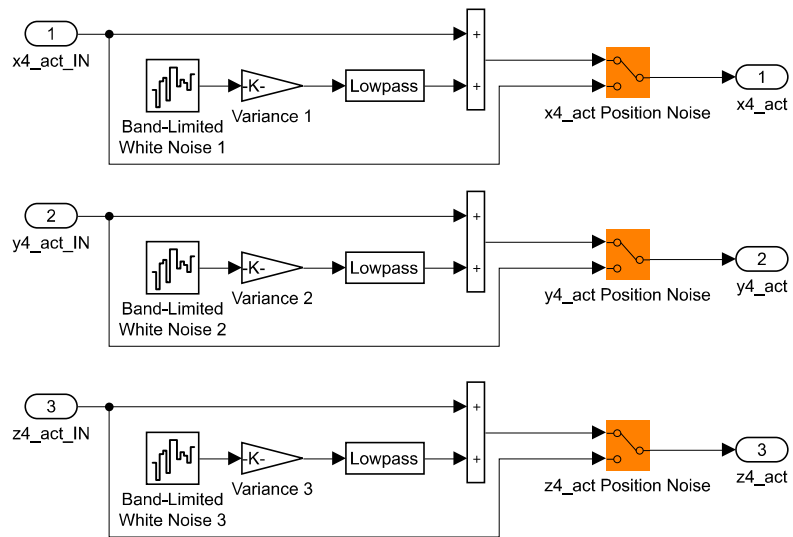
```
% Transform V_com_R4 (wrt G) to V_com_R4_R4 (wrt R4)
V_com_R4_R4_top = R4_x_G*[ x4d_com ; y4d_com ; z4d_com];
V_com_R4_R4_bot = R4_z_G*[psi_4d_com ; theta_4d_com ; phi_4d_com];
V_com_R4_R4 = vertcat(V_com_R4_R4_top,V_com_R4_R4_bot);
```











```

function [V_com_CS,Cluster_Formation_Attitude_Error] = RRCSC( P_TRAJ_CS,...
    P_act_CS,...
    VB_des,...
    V_act_CS,...
    Q,...
    Qp,...
    Kp)

% Decompose P_TRAJ_CS vector into Desired Cluster-Space Position Variables:
XB_des = P_TRAJ_CS(1,1);   YB_des = P_TRAJ_CS(2,1);   ZB_des = P_TRAJ_CS(3,1);
L12_des = P_TRAJ_CS(4,1);   L13_des = P_TRAJ_CS(5,1);   LB4_des = P_TRAJ_CS(6,1);
PSI_C_des = P_TRAJ_CS(7,1);   THETA_C_des = P_TRAJ_CS(8,1);   PHI_C_des = P_TRAJ_CS(9,1);
ALPHA_des = P_TRAJ_CS(10,1);   BETA_des = P_TRAJ_CS(11,1);   XI_des = P_TRAJ_CS(12,1);
PSI_1_des = P_TRAJ_CS(13,1);   THETA_1_des = P_TRAJ_CS(14,1);   PHI_1_des = P_TRAJ_CS(15,1);
PSI_2_des = P_TRAJ_CS(16,1);   THETA_2_des = P_TRAJ_CS(17,1);   PHI_2_des = P_TRAJ_CS(18,1);
PSI_3_des = P_TRAJ_CS(19,1);   THETA_3_des = P_TRAJ_CS(20,1);   PHI_3_des = P_TRAJ_CS(21,1);
PSI_4_des = P_TRAJ_CS(22,1);   THETA_4_des = P_TRAJ_CS(23,1);   PHI_4_des = P_TRAJ_CS(24,1);

% Decompose P_act_CS vector into Actual Cluster-Space Position Variables:
XB_act = P_act_CS(1,1);   YB_act = P_act_CS(2,1);   ZB_act = P_act_CS(3,1);
L12_act = P_act_CS(4,1);   L13_act = P_act_CS(5,1);   LB4_act = P_act_CS(6,1);
PSI_C_act = P_act_CS(7,1);   THETA_C_act = P_act_CS(8,1);   PHI_C_act = P_act_CS(9,1);
ALPHA_act = P_act_CS(10,1);   BETA_act = P_act_CS(11,1);   XI_act = P_act_CS(12,1);
PSI_1_act = P_act_CS(13,1);   THETA_1_act = P_act_CS(14,1);   PHI_1_act = P_act_CS(15,1);
PSI_2_act = P_act_CS(16,1);   THETA_2_act = P_act_CS(17,1);   PHI_2_act = P_act_CS(18,1);
PSI_3_act = P_act_CS(19,1);   THETA_3_act = P_act_CS(20,1);   PHI_3_act = P_act_CS(21,1);
PSI_4_act = P_act_CS(22,1);   THETA_4_act = P_act_CS(23,1);   PHI_4_act = P_act_CS(24,1);

% Decompose VB_des vector into Desired Cluster Point B Velocity Variables:
XBd_des = VB_des(1,1);   YBd_des = VB_des(2,1);   ZBd_des = VB_des(3,1);

% Decompose V_act_CS vector into Actual Cluster-Space Velocity Variables:
XBd_act = V_act_CS(1,1);   YBd_act = V_act_CS(2,1);   ZBd_act = V_act_CS(3,1);
L12d_act = V_act_CS(4,1);   L13d_act = V_act_CS(5,1);   LB4d_act = V_act_CS(6,1);
PSI_Cd_act = V_act_CS(7,1);   THETA_Cd_act = V_act_CS(8,1);   PHI_Cd_act = V_act_CS(9,1);
ALPHAd_act = V_act_CS(10,1);   BETAd_act = V_act_CS(11,1);   XId_act = V_act_CS(12,1);
PSI_1d_act = V_act_CS(13,1);   THETA_1d_act = V_act_CS(14,1);   PHI_1d_act = V_act_CS(15,1);
PSI_2d_act = V_act_CS(16,1);   THETA_2d_act = V_act_CS(17,1);   PHI_2d_act = V_act_CS(18,1);
PSI_3d_act = V_act_CS(19,1);   THETA_3d_act = V_act_CS(20,1);   PHI_3d_act = V_act_CS(21,1);
PSI_4d_act = V_act_CS(22,1);   THETA_4d_act = V_act_CS(23,1);   PHI_4d_act = V_act_CS(24,1);

% Compute Error Signals
E_XB = XB_des - XB_act; E_YB = YB_des - YB_act; E_ZB = ZB_des - ZB_act;
E_L12 = L12_des - L12_act; E_L13 = L13_des - L13_act; E_LB4 = LB4_des - LB4_act;
E_PSI_C = PSI_C_des - PSI_C_act; E_THETA_C = THETA_C_des - THETA_C_act; E_PHI_C = PHI_C_des - PHI_C_act;
E_ALPHA = ALPHA_des - ALPHA_act; E_BETA = BETA_des - BETA_act; E_XI = XI_des - XI_act;
E_PSI_1 = PSI_1_des - PSI_1_act; E_THETA_1 = THETA_1_des - THETA_1_act; E_PHI_1 = PHI_1_des - PHI_1_act;
E_PSI_2 = PSI_2_des - PSI_2_act; E_THETA_2 = THETA_2_des - THETA_2_act; E_PHI_2 = PHI_2_des - PHI_2_act;
E_PSI_3 = PSI_3_des - PSI_3_act; E_THETA_3 = THETA_3_des - THETA_3_act; E_PHI_3 = PHI_3_des - PHI_3_act;
E_PSI_4 = PSI_4_des - PSI_4_act; E_THETA_4 = THETA_4_des - THETA_4_act; E_PHI_4 = PHI_4_des - PHI_4_act;

E_XBd = XBd_des - XBd_act; E_YBd = YBd_des - YBd_act; E_ZBd = ZBd_des - ZBd_act;
E_L12d = L12d_des - L12d_act; E_L13d = L13d_des - L13d_act; E_LB4d = LB4d_des - LB4d_act;
E_PSI_Cd = PSI_Cd_des - PSI_Cd_act; E_THETA_Cd = THETA_Cd_des - THETA_Cd_act; E_PHI_Cd = PHI_Cd_des - PHI_Cd_act;
E_ALPHAd = ALPHAd_des - ALPHAd_act; E_BETAd = BETAd_des - BETAd_act; E_XId = XId_des - XId_act;
E_PSI_1d = PSI_1d_des - PSI_1d_act; E_THETA_1d = THETA_1d_des - THETA_1d_act; E_PHI_1d = PHI_1d_des - PHI_1d_act;
E_PSI_2d = PSI_2d_des - PSI_2d_act; E_THETA_2d = THETA_2d_des - THETA_2d_act; E_PHI_2d = PHI_2d_des - PHI_2d_act;
E_PSI_3d = PSI_3d_des - PSI_3d_act; E_THETA_3d = THETA_3d_des - THETA_3d_act; E_PHI_3d = PHI_3d_des - PHI_3d_act;
E_PSI_4d = PSI_4d_des - PSI_4d_act; E_THETA_4d = THETA_4d_des - THETA_4d_act; E_PHI_4d = PHI_4d_des - PHI_4d_act;

% Error Range Limiting Logic
if E_PSI_C < -pi
    E_PSI_C = E_PSI_C + 2*pi;
end
if E_PSI_C > pi
    E_PSI_C = E_PSI_C - 2*pi;
end
if E_PHI_C < -pi
    E_PHI_C = E_PHI_C + 2*pi;
end
if E_PHI_C > pi
    E_PHI_C = E_PHI_C - 2*pi;
end
if E_PSI_1 < -pi
    E_PSI_1 = E_PSI_1 + 2*pi;
end
if E_PSI_1 > pi

```

```

    E_PSI_1 = E_PSI_1 - 2*pi;
end
if E_PHI_1 < -pi
    E_PHI_1 = E_PHI_1 + 2*pi;
end
if E_PHI_1 > pi
    E_PHI_1 = E_PHI_1 - 2*pi;
end
if E_PSI_2 < -pi
    E_PSI_2 = E_PSI_2 + 2*pi;
end
if E_PSI_2 > pi
    E_PSI_2 = E_PSI_2 - 2*pi;
end
if E_PHI_2 < -pi
    E_PHI_2 = E_PHI_2 + 2*pi;
end
if E_PHI_2 > pi
    E_PHI_2 = E_PHI_2 - 2*pi;
end
if E_PSI_3 < -pi
    E_PSI_3 = E_PSI_3 + 2*pi;
end
if E_PSI_3 > pi
    E_PSI_3 = E_PSI_3 - 2*pi;
end
if E_PHI_3 < -pi
    E_PHI_3 = E_PHI_3 + 2*pi;
end
if E_PHI_3 > pi
    E_PHI_3 = E_PHI_3 - 2*pi;
end
if E_PSI_4 < -pi
    E_PSI_4 = E_PSI_4 + 2*pi;
end
if E_PSI_4 > pi
    E_PSI_4 = E_PSI_4 - 2*pi;
end
if E_PHI_4 < -pi
    E_PHI_4 = E_PHI_4 + 2*pi;
end
if E_PHI_4 > pi
    E_PHI_4 = E_PHI_4 - 2*pi;
end

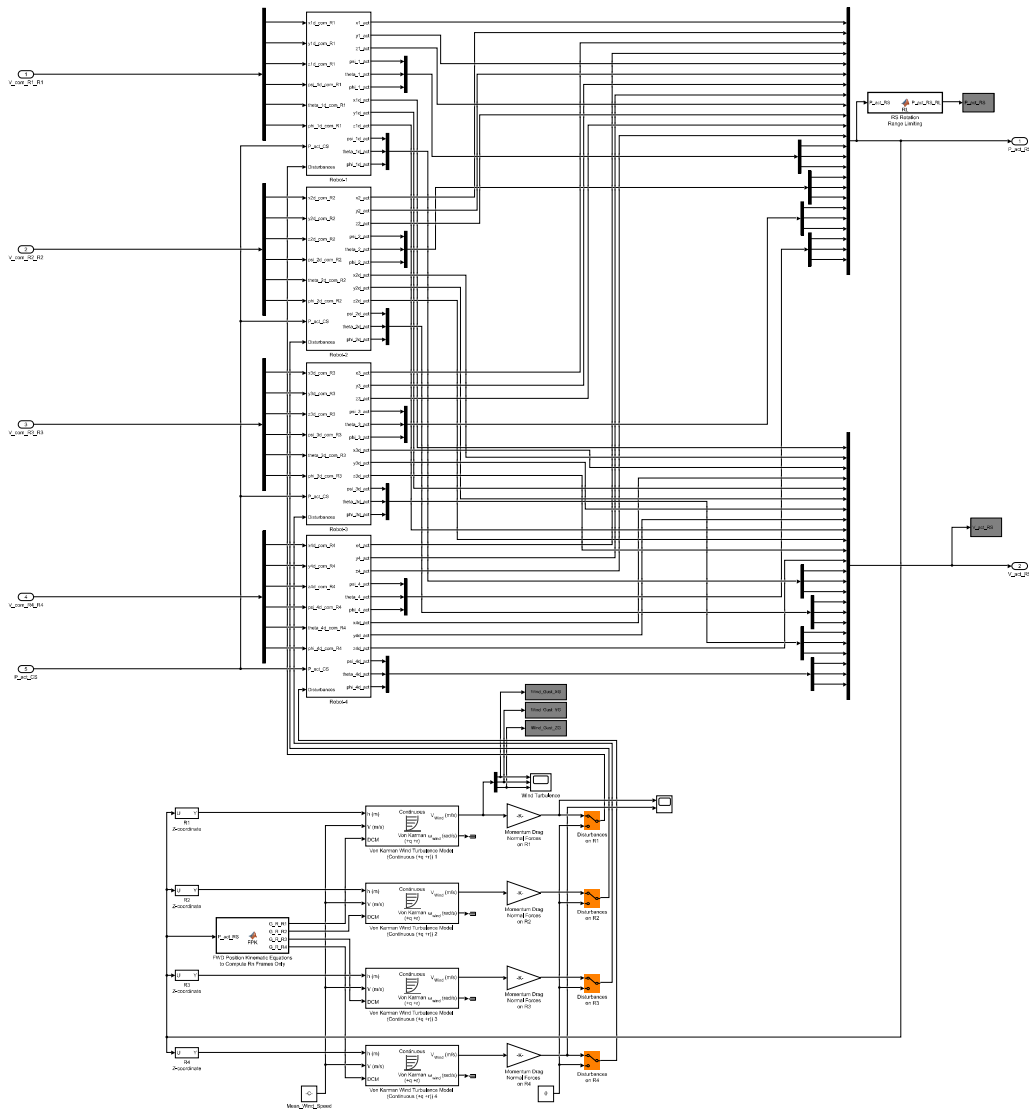
Error_P_CS = [ E_XB      ; E_YB      ; E_ZB      ;
               E_L12     ; E_L13     ; E_LB4     ;
               E_PSI_C   ; E_THETA_C ; E_PHI_C   ;
               E_ALPHA   ; E_BETA    ; E_XI     ;
               E_PSI_1   ; E_THETA_1 ; E_PHI_1   ;
               E_PSI_2   ; E_THETA_2 ; E_PHI_2   ;
               E_PSI_3   ; E_THETA_3 ; E_PHI_3   ;
               E_PSI_4   ; E_THETA_4 ; E_PHI_4   ];

Error_V_CS = [ E_XBd     ; E_YBd     ; E_ZBd     ;
               E_L12d    ; E_L13d    ; E_LB4d    ;
               E_PSI_Cd  ; E_THETA_Cd ; E_PHI_Cd  ;
               E_ALPHAd  ; E_BETAd   ; E_XId     ;
               E_PSI_1d  ; E_THETA_1d ; E_PHI_1d  ;
               E_PSI_2d  ; E_THETA_2d ; E_PHI_2d  ;
               E_PSI_3d  ; E_THETA_3d ; E_PHI_3d  ;
               E_PSI_4d  ; E_THETA_4d ; E_PHI_4d  ];

% Cluster-Space Velocity Control Commands
V_com_CS = Kp*( Q*(Error_P_CS) + Qp*(Error_V_CS));

% Cluster Formation and Attitude Error
Cluster_Formation_Attitude_Error = [ E_L12     ; E_L13     ; E_LB4     ; ...
                                     E_PSI_C   ; E_THETA_C ; E_PHI_C   ; ...
                                     E_ALPHA   ; E_BETA    ; E_XI     ; ];

```

```

function [G_R_R1,G_R_R2,G_R_R3,G_R_R4] = FPK(P_act_RS)

% Decompose P_act_RS vector in Actual Robot-Space Position Variables:
x1 = P_act_RS(1,1); x2 = P_act_RS(2,1); x3 = P_act_RS(3,1); x4 = P_act_RS(4,1);
y1 = P_act_RS(5,1); y2 = P_act_RS(6,1); y3 = P_act_RS(7,1); y4 = P_act_RS(8,1);
z1 = P_act_RS(9,1); z2 = P_act_RS(10,1); z3 = P_act_RS(11,1); z4 = P_act_RS(12,1);
psi_1 = P_act_RS(13,1); theta_1 = P_act_RS(14,1); phi_1 = P_act_RS(15,1);
psi_2 = P_act_RS(16,1); theta_2 = P_act_RS(17,1); phi_2 = P_act_RS(18,1);
psi_3 = P_act_RS(19,1); theta_3 = P_act_RS(20,1); phi_3 = P_act_RS(21,1);
psi_4 = P_act_RS(22,1); theta_4 = P_act_RS(23,1); phi_4 = P_act_RS(24,1);

%=====  

%=====  

% Robot-1 Orientation =====  

% Robot-1 Frame (R1) Orientation expressed in G  

G_R_R1P = ...  

[ cos(psi_1)   -sin(psi_1)   0   ;...  

  sin(psi_1)   cos(psi_1)   0   ;...  

  0             0             1   ];  

R1P_R_R1PP = ...  

[ cos(theta_1)   0   sin(theta_1);...  

  0             1   0             ;...  

  -sin(theta_1)  0   cos(theta_1)];  

R1PP_R_R1 = ...  

[ 1   0   0   ;...  

  0   cos(phi_1) -sin(phi_1) ;...  

  0   sin(phi_1)  cos(phi_1)  ];  

G_R_R1 = G_R_R1P*R1P_R_R1PP*R1PP_R_R1; % [ Xhat_1 Yhat_1 Zhat_1 ]  

%=====  

%=====  

%=====  

%=====  

% Robot-2 Orientation =====  

% Robot-2 Frame (R2) Orientation expressed in G  

G_R_R2P = ...  

[ cos(psi_2)   -sin(psi_2)   0   ;...  

  sin(psi_2)   cos(psi_2)   0   ;...  

  0             0             1   ];  

R2P_R_R2PP = ...  

[ cos(theta_2)   0   sin(theta_2);...  

  0             1   0             ;...  

  -sin(theta_2)  0   cos(theta_2)];  

R2PP_R_R2 = ...  

[ 1   0   0   ;...  

  0   cos(phi_2) -sin(phi_2) ;...  

  0   sin(phi_2)  cos(phi_2)  ];  

G_R_R2 = G_R_R2P*R2P_R_R2PP*R2PP_R_R2; % [ Xhat_2 Yhat_2 Zhat_2 ]  

%=====  

%=====  

%=====  

%=====  

% Robot-3 Orientation =====  

% Robot-3 Frame (R3) Orientation expressed in G  

G_R_R3P = ...  

[ cos(psi_3)   -sin(psi_3)   0   ;...  

  sin(psi_3)   cos(psi_3)   0   ;...  

  0             0             1   ];  

R3P_R_R3PP = ...  

[ cos(theta_3)   0   sin(theta_3);...  

  0             1   0             ;...  

  -sin(theta_3)  0   cos(theta_3)];  

R3PP_R_R3 = ...  

[ 1   0   0   ;...  

  0   cos(phi_3) -sin(phi_3) ;...  

  0   sin(phi_3)  cos(phi_3)  ];  

G_R_R3 = G_R_R3P*R3P_R_R3PP*R3PP_R_R3; % [ Xhat_3 Yhat_3 Zhat_3 ]  

%=====  

%=====  

%=====  

%=====  

% Robot-4 Orientation =====  

% Robot-4 Frame (R4) Orientation expressed in G  

G_R_R4P = ...  

[ cos(psi_4)   -sin(psi_4)   0   ;...  

  sin(psi_4)   cos(psi_4)   0   ;...  

  0             0             1   ];  

R4P_R_R4PP = ...  

[ cos(theta_4)   0   sin(theta_4);...

```

```

      0      1      0      ;...
R4PP_R_R4 =... [-sin(theta_4) 0 cos(theta_4)];
      [ 1      0      0      ;...
        0      cos(phi_4) -sin(phi_4) ;...
        0      sin(phi_4)  cos(phi_4)  ];
G_R_R4 = G_R_R4P*R4P_R_R4PP*R4PP_R_R4; % [ Xhat_4 Yhat_4 Zhat_4 ]
%===== Robot-4 Orientation =====
%
```

```

function P_act_RS_RL = RL(P_act_RS)

% Decompose P_act_RS vector in Actual Robot-Space Position Variables:
x1 = P_act_RS(1,1); x2 = P_act_RS(2,1); x3 = P_act_RS(3,1); x4 = P_act_RS(4,1);
y1 = P_act_RS(5,1); y2 = P_act_RS(6,1); y3 = P_act_RS(7,1); y4 = P_act_RS(8,1);
z1 = P_act_RS(9,1); z2 = P_act_RS(10,1); z3 = P_act_RS(11,1); z4 = P_act_RS(12,1);
psi_1 = P_act_RS(13,1); theta_1 = P_act_RS(14,1); phi_1 = P_act_RS(15,1);
psi_2 = P_act_RS(16,1); theta_2 = P_act_RS(17,1); phi_2 = P_act_RS(18,1);
psi_3 = P_act_RS(19,1); theta_3 = P_act_RS(20,1); phi_3 = P_act_RS(21,1);
psi_4 = P_act_RS(22,1); theta_4 = P_act_RS(23,1); phi_4 = P_act_RS(24,1);

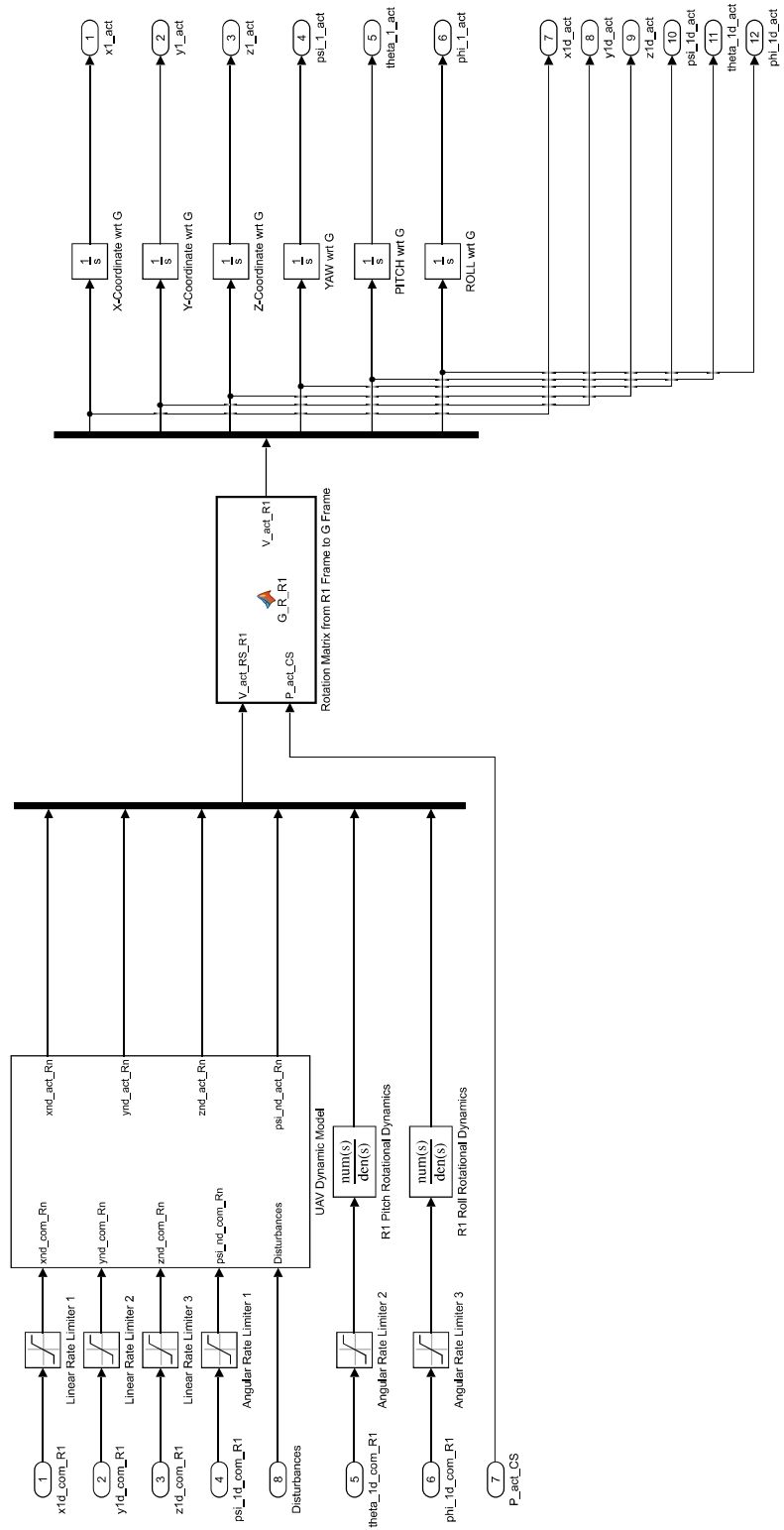
%-----
if (psi_1 > pi) % Logic for +PI/-PI discontinuity
    psi_1 = psi_1 - 2*pi;
    while psi_1 > pi
        psi_1 = psi_1 - 2*pi;
    end
end
if (psi_1 < -pi) % Logic for +PI/-PI discontinuity
    psi_1 = psi_1 + 2*pi;
    while psi_1 < -pi
        psi_1 = psi_1 + 2*pi;
    end
end
if (phi_1 > pi) % Logic for +PI/-PI discontinuity
    phi_1 = phi_1 - 2*pi;
    while phi_1 > pi
        phi_1 = phi_1 - 2*pi;
    end
end
if (phi_1 < -pi) % Logic for +PI/-PI discontinuity
    phi_1 = phi_1 + 2*pi;
    while phi_1 < -pi
        phi_1 = phi_1 + 2*pi;
    end
end
%-----
if (psi_2 > pi) % Logic for +PI/-PI discontinuity
    psi_2 = psi_2 - 2*pi;
    while psi_2 > pi
        psi_2 = psi_2 - 2*pi;
    end
end
if (psi_2 < -pi) % Logic for +PI/-PI discontinuity
    psi_2 = psi_2 + 2*pi;
    while psi_2 < -pi
        psi_2 = psi_2 + 2*pi;
    end
end
if (phi_2 > pi) % Logic for +PI/-PI discontinuity
    phi_2 = phi_2 - 2*pi;
    while phi_2 > pi
        phi_2 = phi_2 - 2*pi;
    end
end
if (phi_2 < -pi) % Logic for +PI/-PI discontinuity
    phi_2 = phi_2 + 2*pi;
    while phi_2 < -pi
        phi_2 = phi_2 + 2*pi;
    end
end
%-----
if (psi_3 > pi) % Logic for +PI/-PI discontinuity
    psi_3 = psi_3 - 2*pi;
    while psi_3 > pi
        psi_3 = psi_3 - 2*pi;
    end
end
if (psi_3 < -pi) % Logic for +PI/-PI discontinuity
    psi_3 = psi_3 + 2*pi;
    while psi_3 < -pi
        psi_3 = psi_3 + 2*pi;
    end
end
end

```

```

if (phi_3 > pi) % Logic for +PI/-PI discontinuity
    phi_3 = phi_3 - 2*pi;
    while phi_3 > pi
        phi_3 = phi_3 - 2*pi;
    end
end
if (phi_3 < -pi) % Logic for +PI/-PI discontinuity
    phi_3 = phi_3 + 2*pi;
    while phi_3 < -pi
        phi_3 = phi_3 + 2*pi;
    end
end
%-----
if (psi_4 > pi) % Logic for +PI/-PI discontinuity
    psi_4 = psi_4 - 2*pi;
    while psi_4 > pi
        psi_4 = psi_4 - 2*pi;
    end
end
if (psi_4 < -pi) % Logic for +PI/-PI discontinuity
    psi_4 = psi_4 + 2*pi;
    while psi_4 < -pi
        psi_4 = psi_4 + 2*pi;
    end
end
if (phi_4 > pi) % Logic for +PI/-PI discontinuity
    phi_4 = phi_4 - 2*pi;
    while phi_4 > pi
        phi_4 = phi_4 - 2*pi;
    end
end
if (phi_4 < -pi) % Logic for +PI/-PI discontinuity
    phi_4 = phi_4 + 2*pi;
    while phi_4 < -pi
        phi_4 = phi_4 + 2*pi;
    end
end
%-----
P_act_RS_RL = [ x1 ; x2 ; x3 ; x4 ;
                y1 ; y2 ; y3 ; y4 ;
                z1 ; z2 ; z3 ; z4 ;
                psi_1 ; theta_1 ; phi_1 ;
                psi_2 ; theta_2 ; phi_2 ;
                psi_3 ; theta_3 ; phi_3 ;
                psi_4 ; theta_4 ; phi_4 ];

```



```

function V_act_R1 = G_R_R1(V_act_RS_R1, P_act_CS)

% Decompose V_act_RS_R1 into Actual Robot-1 Velocity Variables (wrt R1)
xld_act_R1 = V_act_RS_R1(1,1); yld_act_R1 = V_act_RS_R1(2,1); zld_act_R1 = V_act_RS_R1(3,1);
psi_ld_act_R1 = V_act_RS_R1(4,1); theta_ld_act_R1 = V_act_RS_R1(5,1); phi_ld_act_R1 = V_act_RS_R1(6,1);

% Decompose P_act_CS vector into Actual Cluster-Space Position Variables:
XB = P_act_CS(1,1); YB = P_act_CS(2,1); ZB = P_act_CS(3,1);
L12 = P_act_CS(4,1); L13 = P_act_CS(5,1); LB4 = P_act_CS(6,1);
PSI_C = P_act_CS(7,1); THETA_C = P_act_CS(8,1); PHI_C = P_act_CS(9,1);
ALPHA = P_act_CS(10,1); BETA = P_act_CS(11,1); XI = P_act_CS(12,1);
PSI_1 = P_act_CS(13,1); THETA_1 = P_act_CS(14,1); PHI_1 = P_act_CS(15,1);
PSI_2 = P_act_CS(16,1); THETA_2 = P_act_CS(17,1); PHI_2 = P_act_CS(18,1);
PSI_3 = P_act_CS(19,1); THETA_3 = P_act_CS(20,1); PHI_3 = P_act_CS(21,1);
PSI_4 = P_act_CS(22,1); THETA_4 = P_act_CS(23,1); PHI_4 = P_act_CS(24,1);

%=====  

% Cluster Frame (C) Orientation expressed in G  

G_R_P = [ cos(PSI_C)   -sin(PSI_C)   0   ;...  

          sin(PSI_C)   cos(PSI_C)   0   ;...  

          0             0             1   ];  

P_R_DP = [ cos(THETA_C)  0   sin(THETA_C);...  

          0             1   0   ;...  

          -sin(THETA_C)  0   cos(THETA_C)];  

DP_R_C = [ 1             0             0   ;...  

          0             cos(PHI_C)  -sin(PHI_C);...  

          0             sin(PHI_C)  cos(PHI_C)  ];  

G_R_C = G_R_P*P_R_DP*DP_R_C; % [ Xhat_C Yhat_C Zhat_C ]  

%=====  

%=====  

%=====  

% Robot-1 Frame (R1) Orientation expressed in C  

C_r_R1P =...  

          [ cos(PSI_1)   -sin(PSI_1)   0   ;...  

          sin(PSI_1)   cos(PSI_1)   0   ;...  

          0             0             1   ];  

R1P_r_R1PP =...  

          [ cos(THETA_1)  0   sin(THETA_1);...  

          0             1   0   ;...  

          -sin(THETA_1)  0   cos(THETA_1)];  

R1PP_r_R1 =...  

          [ 1             0             0   ;...  

          0             cos(PHI_1)  -sin(PHI_1);...  

          0             sin(PHI_1)  cos(PHI_1)  ];  

C_r_R1 = C_r_R1P*R1P_r_R1PP*R1PP_r_R1; % [ C_Xhat_1 C_Yhat_1 C_Zhat_1 ]  

% Robot-1 Frame (R1) Orientation expressed in G  

G_r_R1 = G_R_C*C_r_R1; % [ Xhat_1 Yhat_1 Zhat_1 ]  

%=====  

%=====  

%=====  

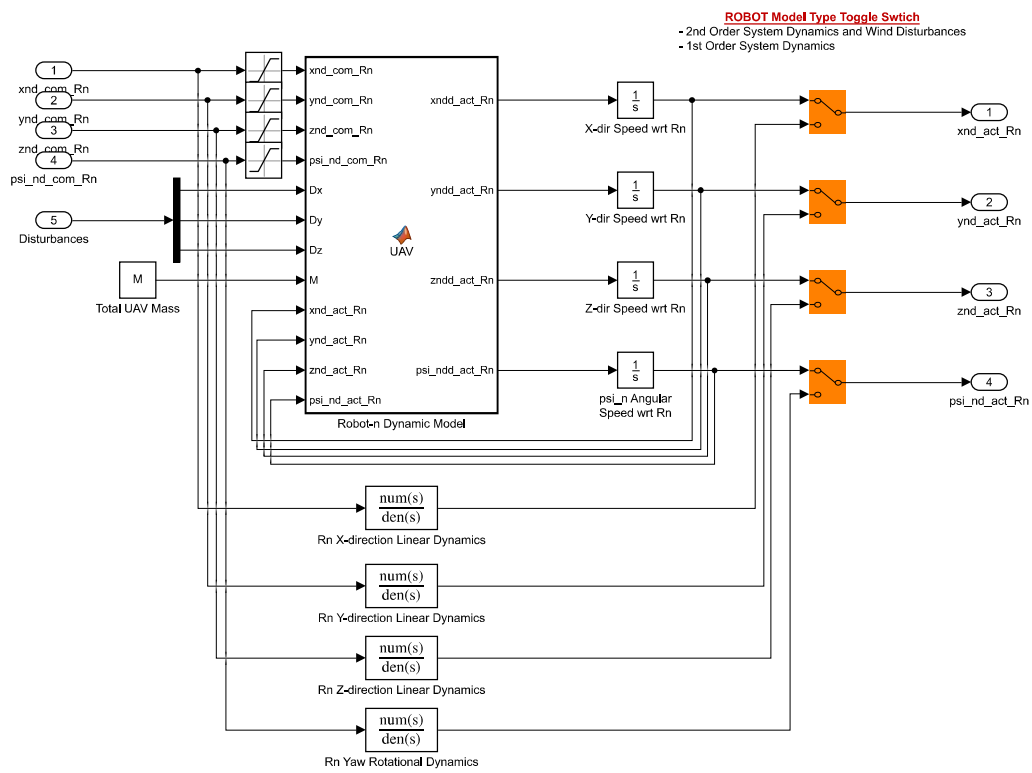
% Transform V_act_R1_R1 (wrt R1) to V_act_R1 (wrt G)  

V_act_R1_top = G_r_R1*[ xld_act_R1 ; yld_act_R1 ; zld_act_R1];  

V_act_R1_bot = G_r_R1*[psi_ld_act_R1 ; theta_ld_act_R1 ; phi_ld_act_R1];  

V_act_R1 = vertcat(V_act_R1_top,V_act_R1_bot);

```




```

function [xndd_act_Rn,yndd_act_Rn,zndd_act_Rn,psi_ndd_act_Rn] = ...
    UAV(xnd_com_Rn,ynd_com_Rn,znd_com_Rn,psi_nd_com_Rn,...
        Dx,Dy,Dz,M,xnd_act_Rn,ynd_act_Rn,znd_act_Rn,psi_nd_act_Rn)

% Decompose Disturbance vector into XYZ components wrt Robot frame:
D = [ Dx ; Dy ; Dz ; 0 ];

% User-Input: UAV Measured Parameters
k1 = 12.63; k2 = 1.43; k3 = 7.61; k4 = 0.84;
k5 = 6.63; k6 = 7.56; k7 = 1.89; k8 = 0.54;

% Robot-n Commanded Velocities wrt Rn
U = [ xnd_com_Rn ; ynd_com_Rn ; znd_com_Rn ; psi_nd_com_Rn ];

% Robot-n Actual Velocities wrt Rn
V = [ xnd_act_Rn ; ynd_act_Rn ; znd_act_Rn ; psi_nd_act_Rn ];

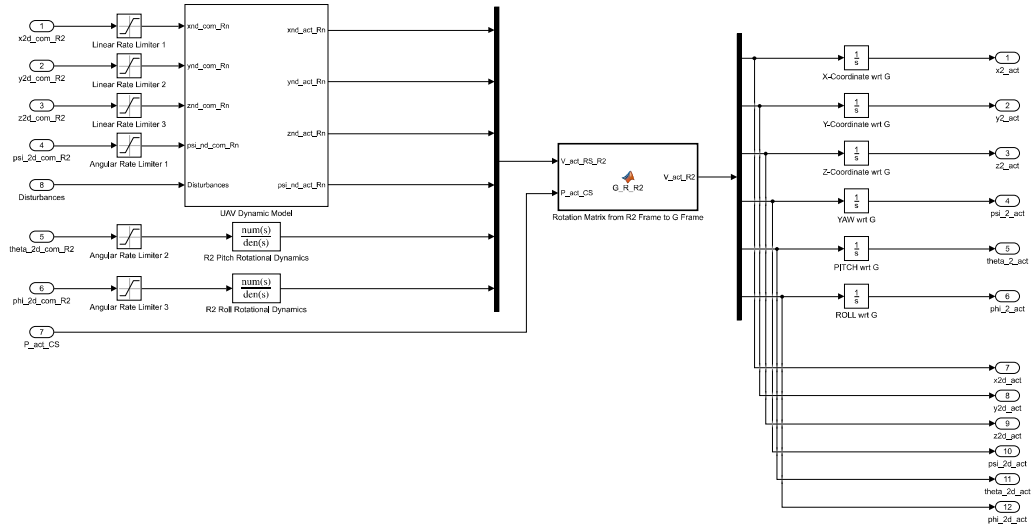
% Form K_u Matrix
K_u = [ k1 0 0 0 ;...
        0 k3 0 0 ;...
        0 0 k5 0 ;...
        0 0 0 k7 ];

% Form K_v Matrix
K_v = [ k2 0 0 0 ;...
        0 k4 0 0 ;...
        0 0 k6 0 ;...
        0 0 0 k8 ];

% Form Acceleration Equation
A = K_u*U - K_v*V + 1/M*D;

% Output Accelerations wrt R1
xndd_act_Rn = A(1,1); yndd_act_Rn = A(2,1); zndd_act_Rn = A(3,1); psi_ndd_act_Rn = A(4,1);

```



```

function V_act_R2 = G_R_R2(V_act_RS_R2, P_act_CS)

% Decompose V_act_RS_R2 into Actual Robot-2 Velocity Variables (wrt R2)
x2d_act_R2 = V_act_RS_R2(1,1); y2d_act_R2 = V_act_RS_R2(2,1); z2d_act_R2 = V_act_RS_R2(3,1);
psi_2d_act_R2 = V_act_RS_R2(4,1); theta_2d_act_R2 = V_act_RS_R2(5,1); phi_2d_act_R2 = V_act_RS_R2(6,1);

% Decompose P_act_CS vector into Actual Cluster-Space Position Variables:
XB = P_act_CS(1,1); YB = P_act_CS(2,1); ZB = P_act_CS(3,1);
L12 = P_act_CS(4,1); L13 = P_act_CS(5,1); LB4 = P_act_CS(6,1);
PSI_C = P_act_CS(7,1); THETA_C = P_act_CS(8,1); PHI_C = P_act_CS(9,1);
ALPHA = P_act_CS(10,1); BETA = P_act_CS(11,1); XI = P_act_CS(12,1);
PSI_1 = P_act_CS(13,1); THETA_1 = P_act_CS(14,1); PHI_1 = P_act_CS(15,1);
PSI_2 = P_act_CS(16,1); THETA_2 = P_act_CS(17,1); PHI_2 = P_act_CS(18,1);
PSI_3 = P_act_CS(19,1); THETA_3 = P_act_CS(20,1); PHI_3 = P_act_CS(21,1);
PSI_4 = P_act_CS(22,1); THETA_4 = P_act_CS(23,1); PHI_4 = P_act_CS(24,1);

%=====  

% Cluster Frame (C) Orientation expressed in G  

G_R_P = [ cos(PSI_C)   -sin(PSI_C)   0   ;...  

          sin(PSI_C)   cos(PSI_C)   0   ;...  

          0             0             1   ];  

P_R_DP = [ cos(THETA_C)  0   sin(THETA_C);...  

          0             1   0   ;...  

          -sin(THETA_C)  0   cos(THETA_C)];  

DP_R_C = [ 1             0             0   ;...  

          0             cos(PHI_C)  -sin(PHI_C);...  

          0             sin(PHI_C)  cos(PHI_C)  ];  

G_R_C = G_R_P*P_R_DP*DP_R_C; % [ Xhat_C Yhat_C Zhat_C ]  

%=====  

%=====  

%=====  

% Robot-2 Frame (R2) Orientation expressed in C  

C_r_R2P =...  

          [ cos(PSI_2)   -sin(PSI_2)   0   ;...  

          sin(PSI_2)   cos(PSI_2)   0   ;...  

          0             0             1   ];  

R2P_r_R2PP =...  

          [ cos(THETA_2)  0   sin(THETA_2);...  

          0             1   0   ;...  

          -sin(THETA_2)  0   cos(THETA_2)];  

R2PP_r_R2 =...  

          [ 1             0             0   ;...  

          0             cos(PHI_2)  -sin(PHI_2);...  

          0             sin(PHI_2)  cos(PHI_2)  ];  

C_r_R2 = C_r_R2P*R2P_r_R2PP*R2PP_r_R2; % [ C_Xhat_2 C_Yhat_2 C_Zhat_2 ]  

% Robot-2 Frame (R2) Orientation expressed in G  

G_r_R2 = G_R_C*C_r_R2; % [ Xhat_2 Yhat_2 Zhat_2 ]  

%=====  

%=====  

%=====  

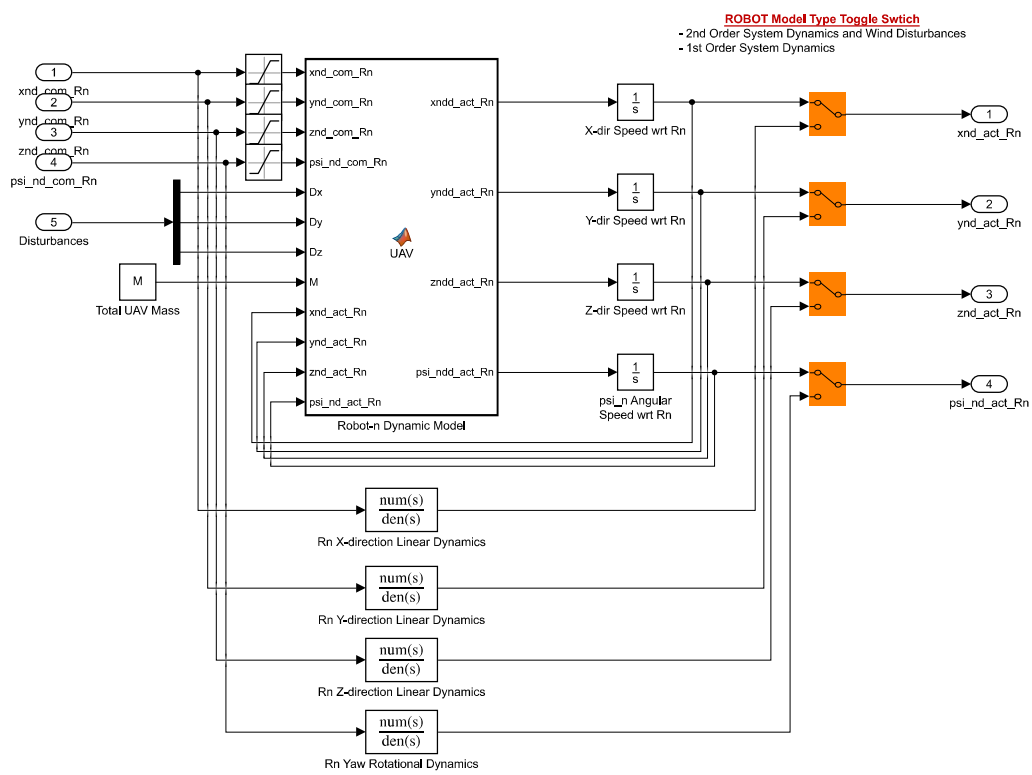
% Transform V_act_R2_R2 (wrt R2) to V_act_R2 (wrt G)  

V_act_R2_top = G_r_R2*[ x2d_act_R2 ; y2d_act_R2 ; z2d_act_R2];  

V_act_R2_bot = G_r_R2*[psi_2d_act_R2 ; theta_2d_act_R2 ; phi_2d_act_R2];  

V_act_R2 = vertcat(V_act_R2_top,V_act_R2_bot);

```



```

function [xndd_act_Rn,yndd_act_Rn,zndd_act_Rn,psi_ndd_act_Rn] = ...
    UAV(xnd_com_Rn,ynd_com_Rn,znd_com_Rn,psi_nd_com_Rn,...
        Dx,Dy,Dz,M,xnd_act_Rn,ynd_act_Rn,znd_act_Rn,psi_nd_act_Rn)

% Decompose Disturbance vector into XYZ components wrt Robot frame:
D = [ Dx ; Dy ; Dz ; 0 ];

% User-Input: UAV Measured Parameters
k1 = 12.63; k2 = 1.43; k3 = 7.61; k4 = 0.84;
k5 = 6.63; k6 = 7.56; k7 = 1.89; k8 = 0.54;

% Robot-n Commanded Velocities wrt Rn
U = [ xnd_com_Rn ; ynd_com_Rn ; znd_com_Rn ; psi_nd_com_Rn ];

% Robot-n Actual Velocities wrt Rn
V = [ xnd_act_Rn ; ynd_act_Rn ; znd_act_Rn ; psi_nd_act_Rn ];

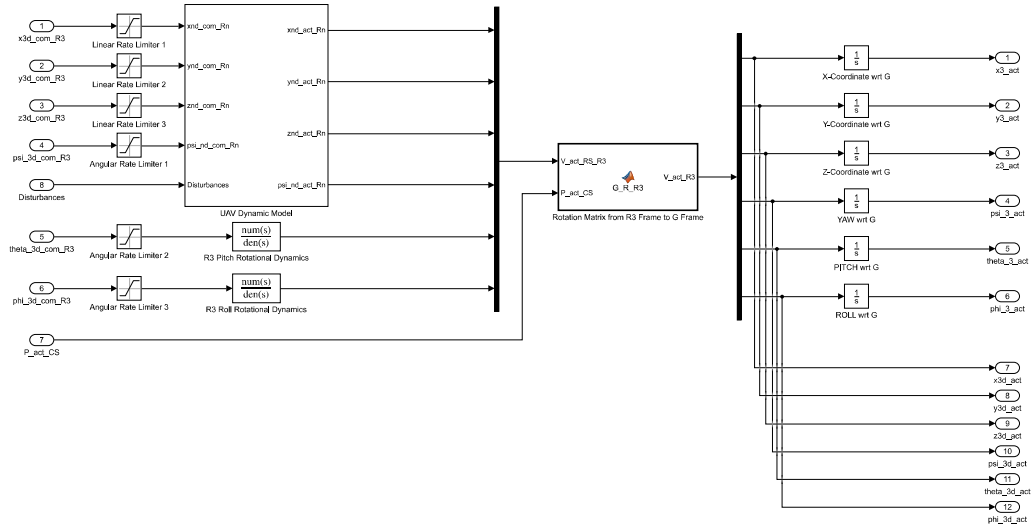
% Form K_u Matrix
K_u = [
    k1 0 0 0 ;...
    0 k3 0 0 ;...
    0 0 k5 0 ;...
    0 0 0 k7 ];

% Form K_v Matrix
K_v = [
    k2 0 0 0 ;...
    0 k4 0 0 ;...
    0 0 k6 0 ;...
    0 0 0 k8 ];

% Form Acceleration Equation
A = K_u*U - K_v*V + 1/M*D;

% Output Accelerations wrt R1
xndd_act_Rn = A(1,1); yndd_act_Rn = A(2,1); zndd_act_Rn = A(3,1); psi_ndd_act_Rn = A(4,1);

```



```

function V_act_R3 = G_R_R3(V_act_RS_R3, P_act_CS)

% Decompose V_act_RS_R3 into Actual Robot-3 Velocity Variables (wrt R3)
x3d_act_R3 = V_act_RS_R3(1,1); y3d_act_R3 = V_act_RS_R3(2,1); z3d_act_R3 = V_act_RS_R3(3,1);
psi_3d_act_R3 = V_act_RS_R3(4,1); theta_3d_act_R3 = V_act_RS_R3(5,1); phi_3d_act_R3 = V_act_RS_R3(6,1);

% Decompose P_act_CS vector into Actual Cluster-Space Position Variables:
XB = P_act_CS(1,1); YB = P_act_CS(2,1); ZB = P_act_CS(3,1);
L12 = P_act_CS(4,1); L13 = P_act_CS(5,1); LB4 = P_act_CS(6,1);
PSI_C = P_act_CS(7,1); THETA_C = P_act_CS(8,1); PHI_C = P_act_CS(9,1);
ALPHA = P_act_CS(10,1); BETA = P_act_CS(11,1); XI = P_act_CS(12,1);
PSI_1 = P_act_CS(13,1); THETA_1 = P_act_CS(14,1); PHI_1 = P_act_CS(15,1);
PSI_2 = P_act_CS(16,1); THETA_2 = P_act_CS(17,1); PHI_2 = P_act_CS(18,1);
PSI_3 = P_act_CS(19,1); THETA_3 = P_act_CS(20,1); PHI_3 = P_act_CS(21,1);
PSI_4 = P_act_CS(22,1); THETA_4 = P_act_CS(23,1); PHI_4 = P_act_CS(24,1);

%=====  

% Cluster Frame (C) Orientation expressed in G  

G_R_P = [ cos(PSI_C)   -sin(PSI_C)   0   ;...  

          sin(PSI_C)   cos(PSI_C)   0   ;...  

          0             0             1   ];  

P_R_DP = [ cos(THETA_C)  0   sin(THETA_C);...  

           0             1   0   ;...  

           -sin(THETA_C) 0   cos(THETA_C)];  

DP_R_C = [ 1   0   0   ;...  

           0   cos(PHI_C) -sin(PHI_C) ;...  

           0   sin(PHI_C)  cos(PHI_C)  ];  

G_R_C = G_R_P*P_R_DP*DP_R_C; % [ Xhat_C Yhat_C Zhat_C ]  

%=====

%=====  

%=====  

% Robot-3 Frame (R3) Orientation expressed in C  

C_r_R3P =...  

          [ cos(PSI_3)   -sin(PSI_3)   0   ;...  

          sin(PSI_3)   cos(PSI_3)   0   ;...  

          0             0             1   ];  

R3P_r_R3PP =...  

          [ cos(THETA_3)  0   sin(THETA_3);...  

           0             1   0   ;...  

           -sin(THETA_3) 0   cos(THETA_3)];  

R3PP_r_R3 =...  

          [ 1   0   0   ;...  

           0   cos(PHI_3) -sin(PHI_3) ;...  

           0   sin(PHI_3)  cos(PHI_3)  ];  

C_r_R3 = C_r_R3P*R3P_r_R3PP*R3PP_r_R3; % [ C_Xhat_3 C_Yhat_3 C_Zhat_3 ]

% Robot-3 Frame (R3) Orientation expressed in G  

G_r_R3 = G_R_C*C_r_R3; % [ Xhat_3 Yhat_3 Zhat_3 ]  

%=====  

%=====  

%=====  

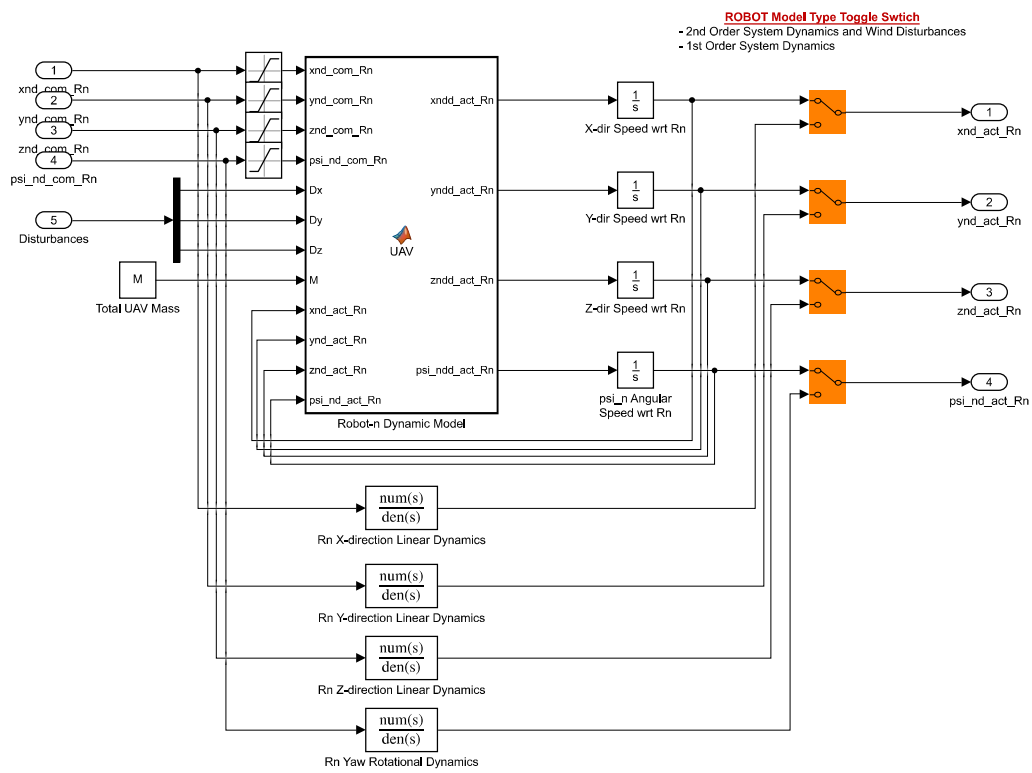
% Transform V_act_R3_R3 (wrt R3) to V_act_R3 (wrt G)  

V_act_R3_top = G_r_R3*[ x3d_act_R3 ; y3d_act_R3 ; z3d_act_R3];  

V_act_R3_bot = G_r_R3*[psi_3d_act_R3 ; theta_3d_act_R3 ; phi_3d_act_R3];  

V_act_R3 = vertcat(V_act_R3_top,V_act_R3_bot);

```




```

function [xndd_act_Rn,yndd_act_Rn,zndd_act_Rn,psi_ndd_act_Rn] = ...
    UAV(xnd_com_Rn,ynd_com_Rn,znd_com_Rn,psi_nd_com_Rn,...
        Dx,Dy,Dz,M,xnd_act_Rn,ynd_act_Rn,znd_act_Rn,psi_nd_act_Rn)

% Decompose Disturbance vector into XYZ components wrt Robot frame:
D = [ Dx ; Dy ; Dz ; 0 ];

% User-Input: UAV Measured Parameters
k1 = 12.63; k2 = 1.43; k3 = 7.61; k4 = 0.84;
k5 = 6.63; k6 = 7.56; k7 = 1.89; k8 = 0.54;

% Robot-n Commanded Velocities wrt Rn
U = [ xnd_com_Rn ; ynd_com_Rn ; znd_com_Rn ; psi_nd_com_Rn ];

% Robot-n Actual Velocities wrt Rn
V = [ xnd_act_Rn ; ynd_act_Rn ; znd_act_Rn ; psi_nd_act_Rn ];

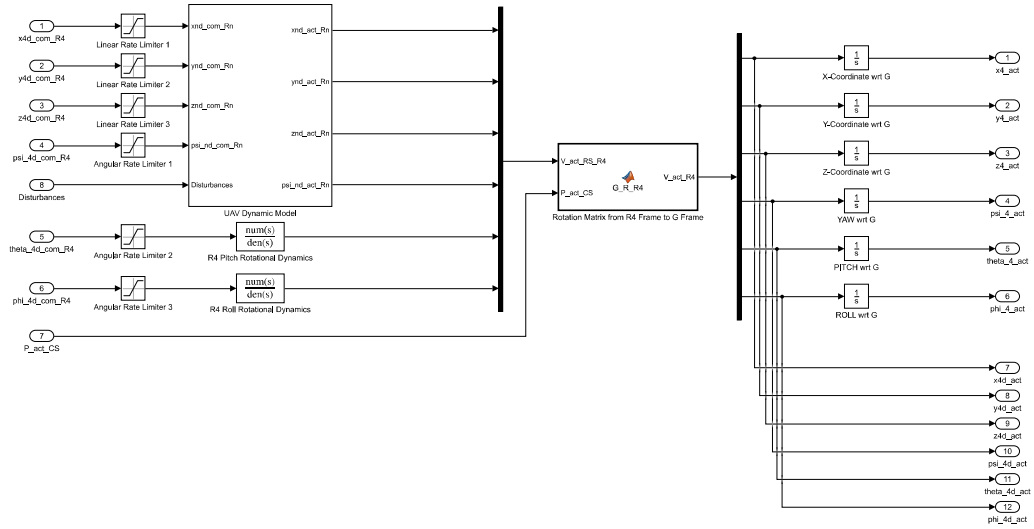
% Form K_u Matrix
K_u = [ k1 0 0 0 ;...
        0 k3 0 0 ;...
        0 0 k5 0 ;...
        0 0 0 k7 ];

% Form K_v Matrix
K_v = [ k2 0 0 0 ;...
        0 k4 0 0 ;...
        0 0 k6 0 ;...
        0 0 0 k8 ];

% Form Acceleration Equation
A = K_u*U - K_v*V + 1/M*D;

% Output Accelerations wrt R1
xndd_act_Rn = A(1,1); yndd_act_Rn = A(2,1); zndd_act_Rn = A(3,1); psi_ndd_act_Rn = A(4,1);

```



```

function V_act_R4 = G_R_R4(V_act_RS_R4, P_act_CS)

% Decompose V_act_RS_R4 into Actual Robot-4 Velocity Variables (wrt R4)
x4d_act_R4 = V_act_RS_R4(1,1); y4d_act_R4 = V_act_RS_R4(2,1); z4d_act_R4 = V_act_RS_R4(3,1);
psi_4d_act_R4 = V_act_RS_R4(4,1); theta_4d_act_R4 = V_act_RS_R4(5,1); phi_4d_act_R4 = V_act_RS_R4(6,1);

% Decompose P_act_CS vector into Actual Cluster-Space Position Variables:
XB = P_act_CS(1,1); YB = P_act_CS(2,1); ZB = P_act_CS(3,1);
L12 = P_act_CS(4,1); L13 = P_act_CS(5,1); LB4 = P_act_CS(6,1);
PSI_C = P_act_CS(7,1); THETA_C = P_act_CS(8,1); PHI_C = P_act_CS(9,1);
ALPHA = P_act_CS(10,1); BETA = P_act_CS(11,1); XI = P_act_CS(12,1);
PSI_1 = P_act_CS(13,1); THETA_1 = P_act_CS(14,1); PHI_1 = P_act_CS(15,1);
PSI_2 = P_act_CS(16,1); THETA_2 = P_act_CS(17,1); PHI_2 = P_act_CS(18,1);
PSI_3 = P_act_CS(19,1); THETA_3 = P_act_CS(20,1); PHI_3 = P_act_CS(21,1);
PSI_4 = P_act_CS(22,1); THETA_4 = P_act_CS(23,1); PHI_4 = P_act_CS(24,1);

%=====  

% Cluster Frame (C) Orientation expressed in G  

G_R_P = [ cos(PSI_C)   -sin(PSI_C)   0   ;...  

          sin(PSI_C)   cos(PSI_C)   0   ;...  

          0             0             1   ];  

P_R_DP = [ cos(THETA_C)  0   sin(THETA_C);...  

          0             1   0   ;...  

          -sin(THETA_C)  0   cos(THETA_C)];  

DP_R_C = [ 1   0   0   ;...  

          0   cos(PHI_C) -sin(PHI_C) ;...  

          0   sin(PHI_C) cos(PHI_C)  ];  

G_R_C = G_R_P*P_R_DP*DP_R_C; % [ Xhat_C Yhat_C Zhat_C ]  

%=====  

%=====  

%=====  

% Robot-4 Frame (R4) Orientation expressed in C  

C_r_R4P =...  

          [ cos(PSI_4)   -sin(PSI_4)   0   ;...  

          sin(PSI_4)   cos(PSI_4)   0   ;...  

          0             0             1   ];  

R4P_r_R4PP =...  

          [ cos(THETA_4)  0   sin(THETA_4);...  

          0             1   0   ;...  

          -sin(THETA_4)  0   cos(THETA_4)];  

R4PP_r_R4 =...  

          [ 1   0   0   ;...  

          0   cos(PHI_4) -sin(PHI_4) ;...  

          0   sin(PHI_4) cos(PHI_4)  ];  

C_r_R4 = C_r_R4P*R4P_r_R4PP*R4PP_r_R4; % [ C_Xhat_4 C_Yhat_4 C_Zhat_4 ]  

% Robot-4 Frame (R4) Orientation expressed in G  

G_r_R4 = G_R_C*C_r_R4; % [ Xhat_4 Yhat_4 Zhat_4 ]  

%=====  

%=====  

%=====  

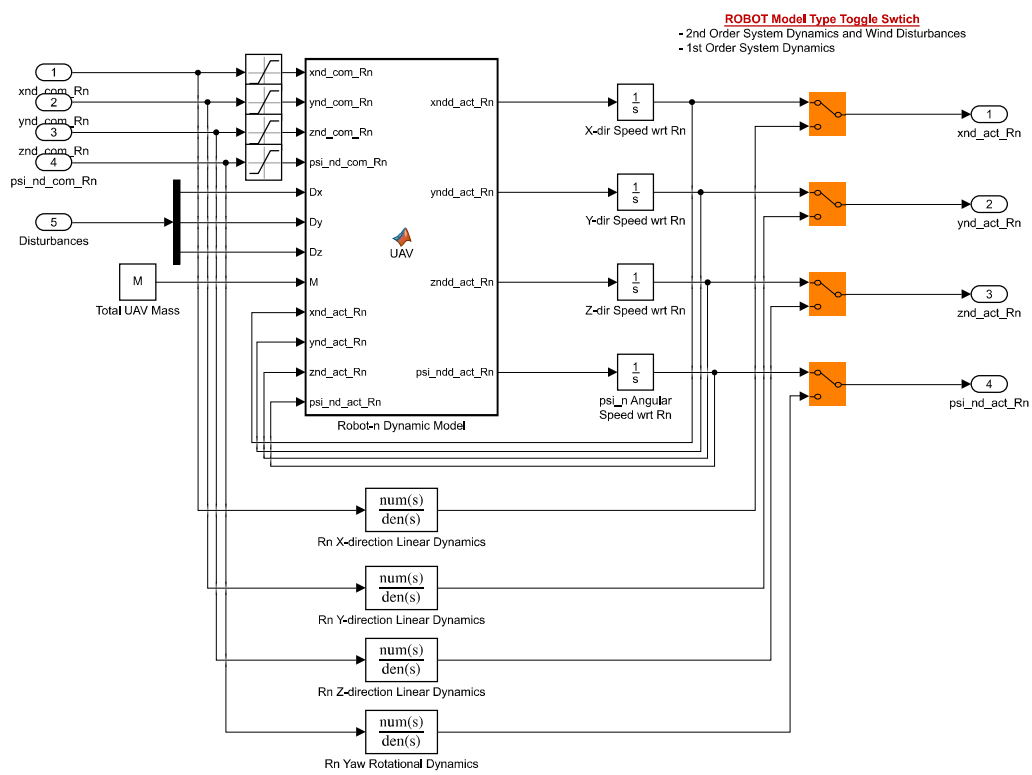
% Transform V_act_R4_R4 (wrt R4) to V_act_R4 (wrt G)  

V_act_R4_top = G_r_R4*[ x4d_act_R4 ; y4d_act_R4 ; z4d_act_R4];  

V_act_R4_bot = G_r_R4*[psi_4d_act_R4 ; theta_4d_act_R4 ; phi_4d_act_R4];  

V_act_R4 = vertcat(V_act_R4_top,V_act_R4_bot);

```



```

function [xndd_act_Rn,yndd_act_Rn,znnd_act_Rn,psi_ndd_act_Rn] = ...
    UAV(xnd_com_Rn,ynd_com_Rn,znd_com_Rn,psi_nd_com_Rn,...
        Dx,Dy,Dz,M,xnd_act_Rn,ynd_act_Rn,znd_act_Rn,psi_nd_act_Rn)

% Decompose Disturbance vector into XYZ components wrt Robot frame:
D = [ Dx ; Dy ; Dz ; 0 ];

% User-Input: UAV Measured Parameters
k1 = 12.63; k2 = 1.43; k3 = 7.61; k4 = 0.84;
k5 = 6.63; k6 = 7.56; k7 = 1.89; k8 = 0.54;

% Robot-n Commanded Velocities wrt Rn
U = [ xnd_com_Rn ; ynd_com_Rn ; znd_com_Rn ; psi_nd_com_Rn ];

% Robot-n Actual Velocities wrt Rn
V = [ xnd_act_Rn ; ynd_act_Rn ; znd_act_Rn ; psi_nd_act_Rn ];

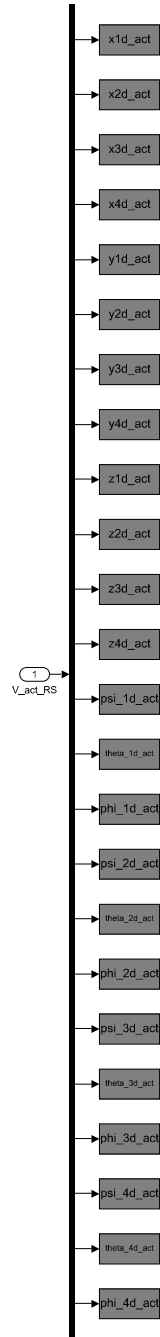
% Form K_u Matrix
K_u = [
    k1 0 0 0 ;...
    0 k3 0 0 ;...
    0 0 k5 0 ;...
    0 0 0 k7 ];

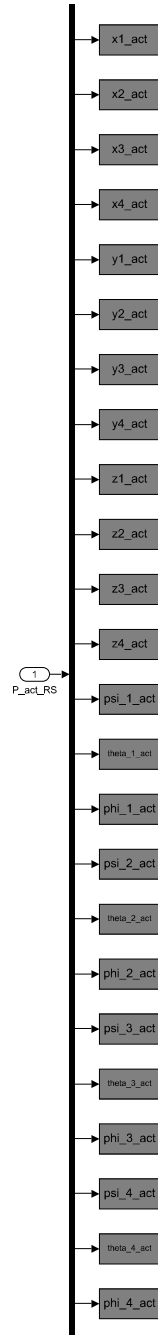
% Form K_v Matrix
K_v = [
    k2 0 0 0 ;...
    0 k4 0 0 ;...
    0 0 k6 0 ;...
    0 0 0 k8 ];

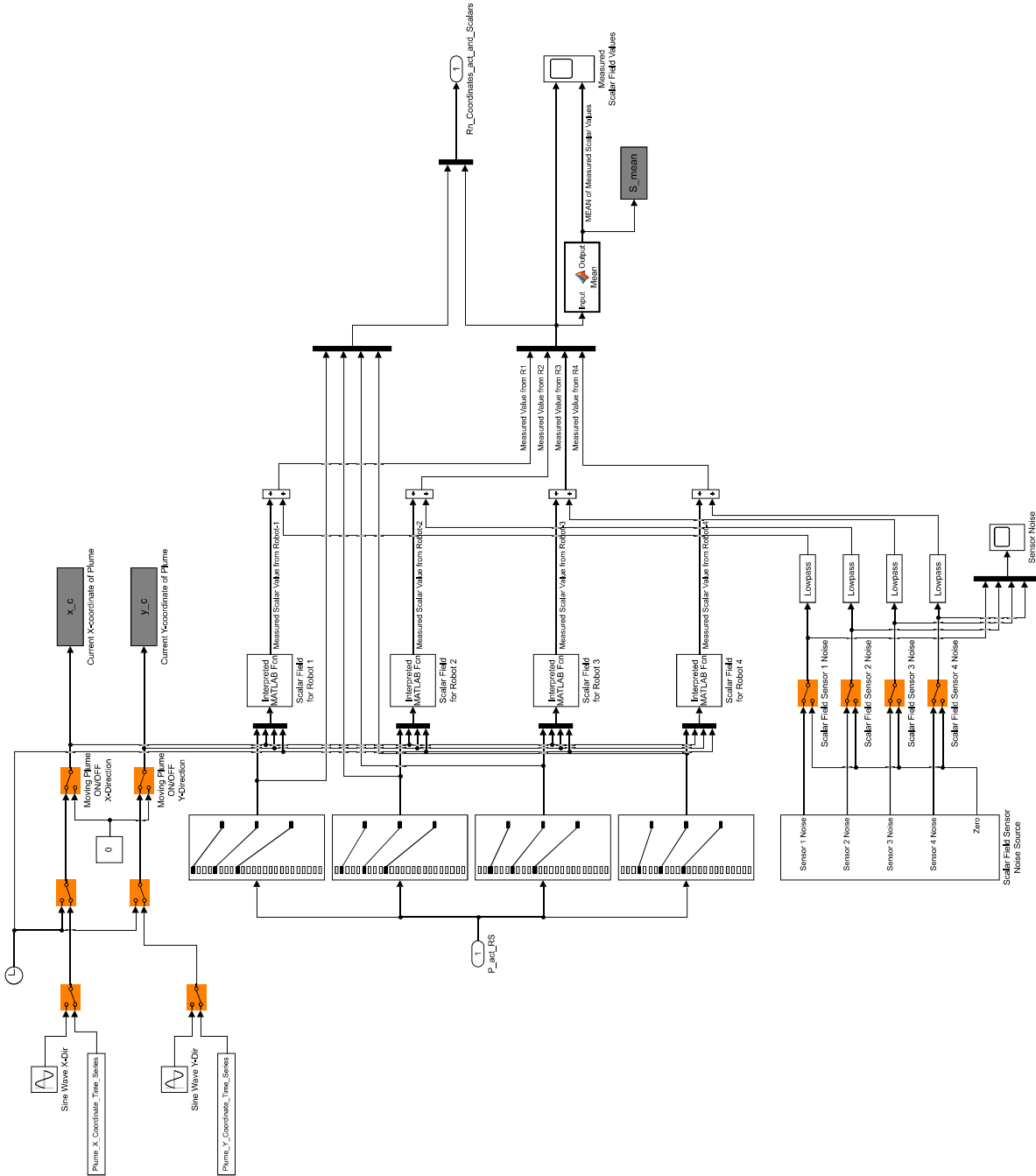
% Form Acceleration Equation
A = K_u*U - K_v*V + 1/M*D;

% Output Accelerations wrt R1
xndd_act_Rn = A(1,1); ynnd_act_Rn = A(2,1); znnd_act_Rn = A(3,1); psi_ndd_act_Rn = A(4,1);

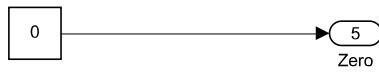
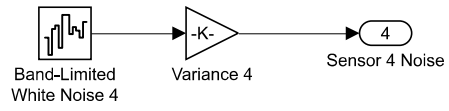
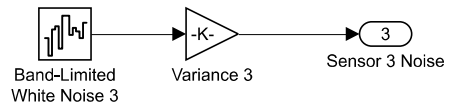
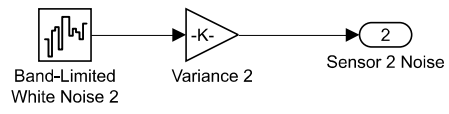
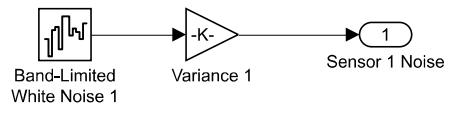
```

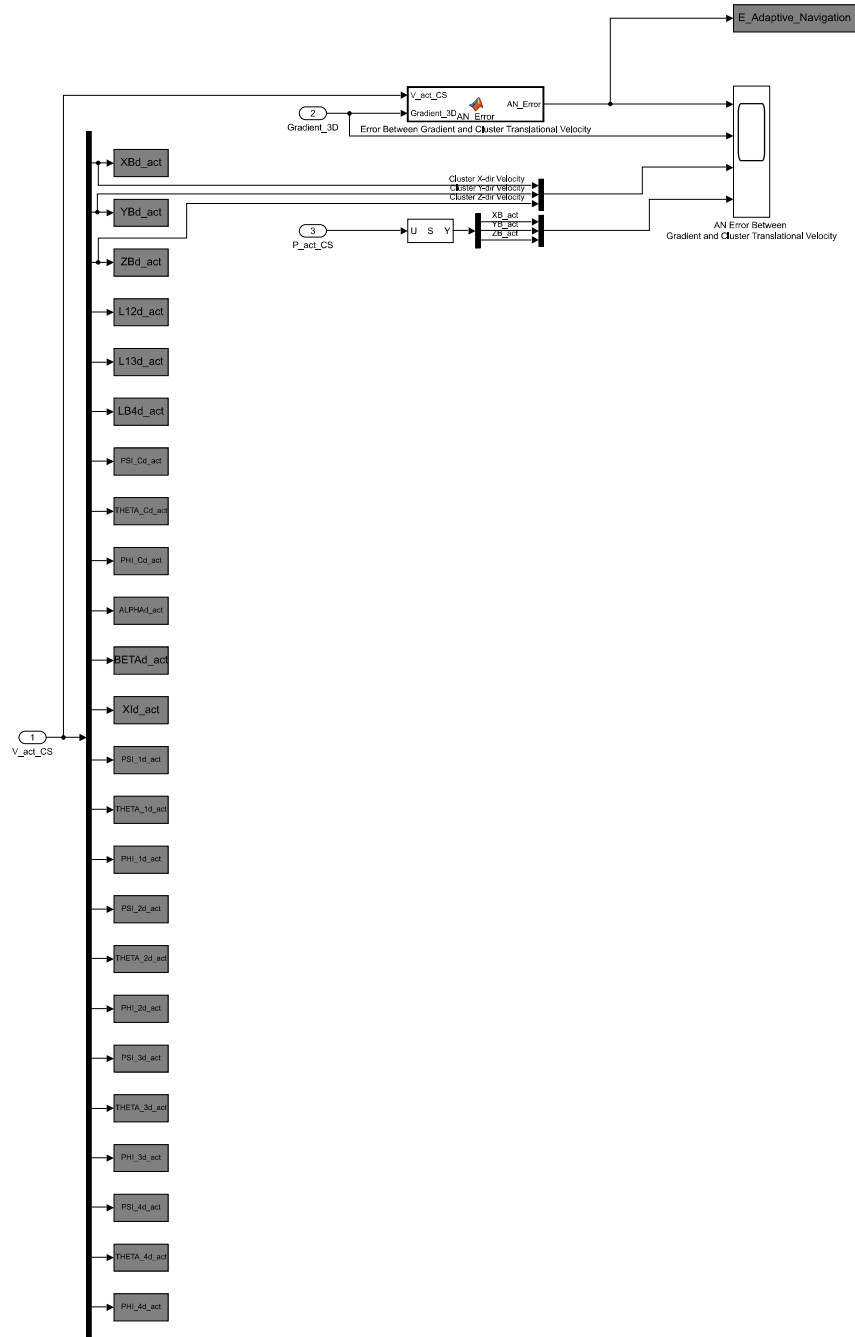







```
function Output = Mean(Input)
% Compute Mean Value of Input Vector
Output = mean(Input);
```



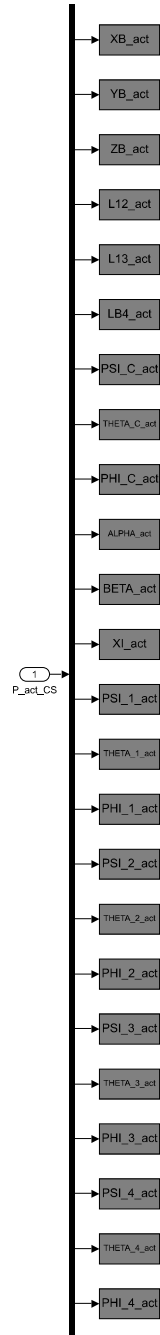


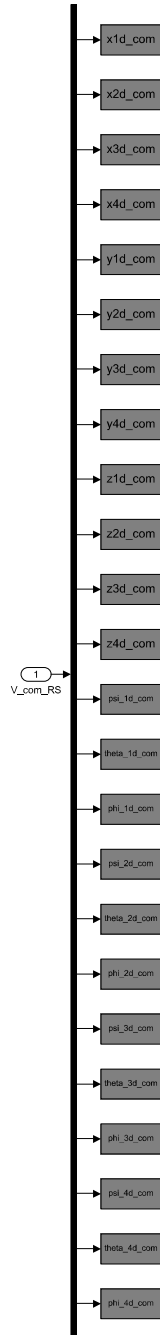
```
function AN_Error = AN_Error(V_act_CS, Gradient_3D)

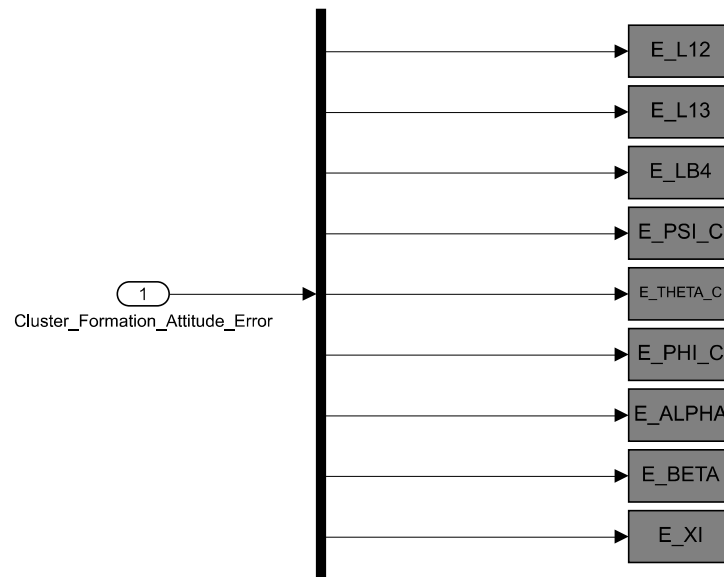
% Extract Cluster Velocity Components from V_act_CS
XBd_act = V_act_CS(1,1); YBd_act = V_act_CS(2,1); ZBd_act = V_act_CS(3,1);

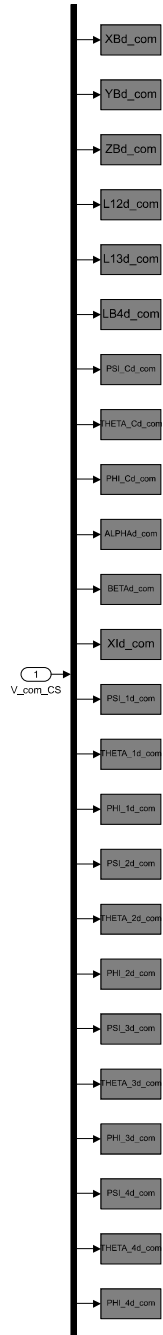
% Compute Cluster Translational Velocity Vector
Cluster_Translational_Velocity = [XBd_act ; YBd_act ; ZBd_act ];

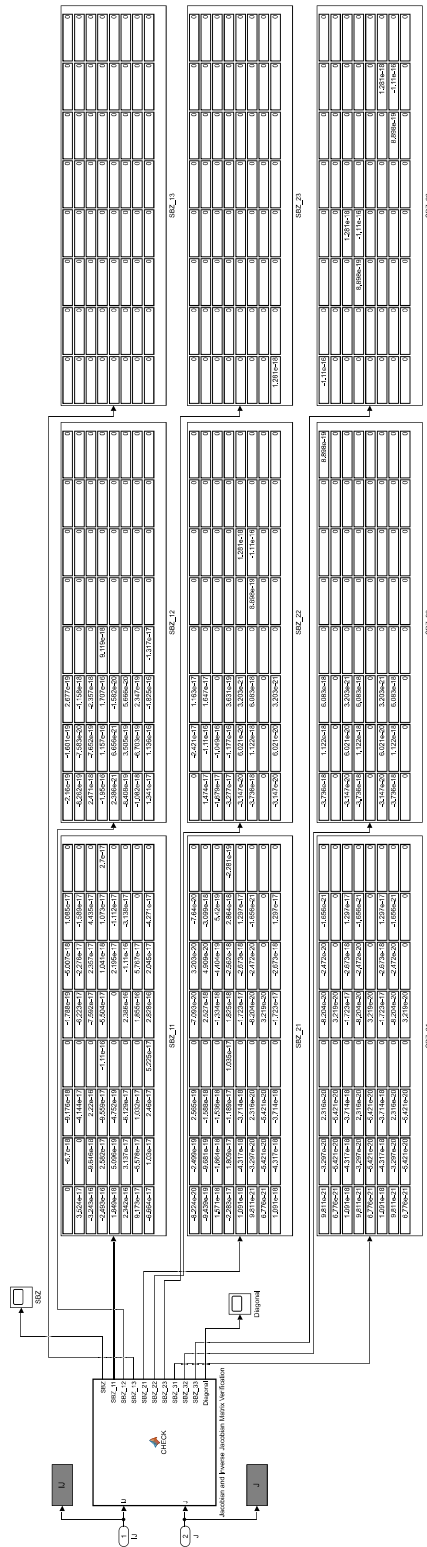
% Error between 3D-Gradient and Cluster Velocity Unit Vectors
AN_Error = subspace(Cluster_Translational_Velocity,Gradient_3D);
```











```

function [SBZ,...
        SBZ_11,SBZ_12,SBZ_13,...
        SBZ_21,SBZ_22,SBZ_23,...
        SBZ_31,SBZ_32,SBZ_33,Diagonal] = CHECK(IJ,J)

% Matrix Multiply: Inverse Jacobian and Jacobian Matrices
SBZ = (IJ*J)-eye(24);

Diagonal = diag(IJ*J);

SBZ_11 = SBZ(1:8,1:8);
SBZ_21 = SBZ(9:16,1:8);
SBZ_31 = SBZ(17:24,1:8);

SBZ_12 = SBZ(1:8,9:16);
SBZ_22 = SBZ(9:16,9:16);
SBZ_32 = SBZ(17:24,9:16);

SBZ_13 = SBZ(1:8,17:24);
SBZ_23 = SBZ(9:16,17:24);
SBZ_33 = SBZ(17:24,17:24);

```