2013

Nonlinear Control for Dual Quaternion Systems

William D. Price
Embry-Riddle Aeronautical University - Daytona Beach

NONLINEAR CONTROL FOR DUAL QUATERNION SYSTEMS

by

William D. Price

A Dissertation Submitted to the Physical Sciences Department
in Partial Fulfillment of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

(Engineering Physics)

Embry-Riddle Aeronautical University

Daytona Beach, FL 32114

2013

NONLINEAR CONTROL FOR DUAL QUATERNION SYSTEMS

By

William D. Price

This Dissertation was prepared under the direction of the candidate's Dissertation Committee Chair, Dr. Sergey V. Drakunov and has been approved by the members of his dissertation committee. It was submitted to the College of Arts and Sciences and was accepted in partial fulfillment of the requirements for the

Degree of
Doctor of Philosophy in Engineering Physics

Dr. Sergey V. Drakunov, Ph.D
Committee Chair

Dr. Yechiel Crispin, Ph.D
Committee Member

Dr. William MacKunis, Ph.D
Committee Member

Dr. Gregory S. Spradlin, Ph.D
Committee Member

Terry R Oswalt

Dr. Terry Oswalt, Ph.D
Department Chair, Physical Sciences

William Grams

Dr. William Grams, Ph.D
Dean, College of Arts and Sciences

M. Hickey

Dr. Michael Hickey, Ph.D
Dean of Research and Graduate Studies

15th November 2013

Date

Acknowledgments

I would like to thank my advisor, Dr. Sergey Drakunov, for providing me with this opportunity, and for his dedication, enthusiasm, and expertise towards the subject matter and completion of this thesis.

Abstract

The motion of rigid bodies includes three degrees of freedom (DOF) for rotation, generally referred to as roll, pitch and yaw, and 3 DOF for translation, generally described as motion along the x, y and z axis, for a total of 6 DOF. Many complex mechanical systems exhibit this type of motion, with constraints, such as complex humanoid robotic systems, multiple ground vehicles, unmanned aerial vehicles (UAVs), multiple spacecraft vehicles, and even quantum mechanical systems. These motions historically have been analyzed independently, with separate control algorithms being developed for rotation and translation. The goal of this research is to study the full 6 DOF of rigid body motion together, developing control algorithms that will affect both rotation and translation simultaneously. This will prove especially beneficial in complex systems in the aerospace and robotics area where translational motion and rotational motion are highly coupled, such as when spacecraft have body fixed thrusters.

A novel mathematical system known as dual quaternions provide an efficient method for mathematically modeling rigid body transformations, expressing both rotation and translation. Dual quaternions can be viewed as a representation of the special Euclidean group $SE(3)$. An eight dimensional representation of screw theory (combining dual numbers with traditional quaternions), dual quaternions allow for the development of control techniques for 6 DOF motion simultaneously. In this work variable structure nonlinear control methods are developed for dual quaternion systems. These techniques include use of sliding mode control. In particular, sliding mode methods are developed for use in dual quaternion systems with unknown control direction. This method, referred to as self-reconfigurable control, is based on the creation of multiple equilibrium surfaces for the system in the extended state space. Also in this work, the control problem for a class of driftless nonlinear systems is addressed via coordinate transformation. It is shown that driftless nonlinear systems that do not meet Brockett's conditions for coordinate transformation can be augmented such that they can be transformed into the Brockett's canonical form, which is nonholonomic. It is also shown that the kinematics for quaternion systems can be represented by a nonholonomic integrator. Then, a discontinuous controller designed for nonholonomic systems is applied. Examples of various applications for dual quaternion systems are given including spacecraft attitude and position control and robotics.

Contents

List of Tables

List of Figures

Nomenclature

Not Bold Letters	a, b, θ	Real scalars in $\mathbb{R}$
	$\vec{\omega}, \vec{v}, \vec{p}$	Real vectors in $\mathbb{R}^3$
	$\hat{a}, \hat{\theta}$	Dual scalars in $\mathbb{DR}$
	$\hat{\vec{\omega}}, \hat{\vec{n}}$	Dual vectors in $\mathbb{DR}^3$
Bold Lowercase Letters	$\mathbf{q}, \boldsymbol{\omega}, \mathbf{p}$	Quaternions in $\mathbb{H}$
	$\hat{\mathbf{q}}, \hat{\boldsymbol{\omega}}, \hat{\mathbf{p}}$	Dual quaternions in $\mathbb{DH}$
Bold Uppercase Letters	$\mathbf{A}, \mathbf{M}, \mathbf{I}$	Real matrices in $\mathbb{R}^{n \times n}$
	$\hat{\mathbf{A}}, \hat{\mathbf{M}}$	Dual matrices in $\mathbb{DR}^{n \times n}$

Chapter 1

Introduction

1.1 Dual Quaternions

The quaternion formulation of rotational kinematics has certain advantages in their application to mechanical systems, namely in robotics, spacecraft control and others . For example, quaternions allow for the easy design of control algorithms by using the quaternion error. This technique can be extended to a special number system known as dual-quaternions. Dual quaternions, a representation of the special Euclidean group $SE(3)$, are an efficient method for representing rigid body transformations, expressing both rotation and translation. They have already been useful in several areas such as developing rigid body dynamics (Brodsky & Shoham, 1999; Dooley, 1991), development of strapdown inertial navigation algorithms (Wu, Hu, & Hu, 2005), and development of a logarithmic proportional control law for rigid body kinematics using dual quaternions (D.-P. Han, Wei, & Li, 2008). This is further developed for control of rigid body kinematics and dynamics using a logarithm proportional derivative controller (D. Han, Wei, & Li, 2008). Dual quaternions also are now used in spacecraft control, specifically using dual quaternions to calculate spacecraft relative position and attitude (Li, Yuan, Yue, & Fang, 2007) as well as developing sliding mode controllers for the spacecraft rendezvous problem (J. Wang, Liang, & Sun, 2012).

The dynamics in this case are represented via dual-numbers and/or dual-vectors by introducing into the real numbers the dual element ε satisfying the property $\varepsilon^2 = 0$. The space of dual-quaternions is actually a Clifford algebra. The models of mechanical systems that include many rotational and translational parts as well as actuator and other dynamics in this case have a multidimensional state space such that each of the dimensions is represented by dual quaternions. By combining a dual quaternion-based dynamic representation with a variable structure approach you can achieve simultaneous rotation and translation control for spatial rigid body systems where the dynamics contain multiple sources of uncertainty and unmodeled effects.

Lyapunov-based techniques have proven to be very effective in nonlinear control design. Recently, these techniques have been extended to dual quaternion based model formulations such as providing a new formulation for the control of spatial rigid bodies (D. Han, Wei, & Li, 2008; D.-P. Han et al., 2008). After revealing the geometric structure of dual quaternions, logarithmic feedbacks are used to derive control laws in both kinematic design and dynamic design. The ideas are extended to regulation and tracking problems.

The high-cost of spacecraft mission design creates a strong interest in the use of new control methods. These high dollar costs include the high expense associated with attitude control equipment (either the value of the equipment itself or added launch weight needed to place larger, heavier equipment in space). Majority of the previous studies of spacecraft maneuvers have analyzed attitude and translation as separate problems.

One way costs can be minimized on a spacecraft is to reduce the number of actuators onboard, providing motivation to study the underactuated spacecraft problem. An underactuated spacecraft is one that has fewer actuators than total degrees-of-freedom. This could be the result of the spacecraft's initial design or because of a failure mode, allowing redundancy to be built into the control algorithm rather in actual hardware. Methods have been used to control both position and attitude of an underactuated satellite using only four coplanar thrusters (Yoshimura, Matsuno, & Hokamoto, 2011). Stability, accessibility, and controllability of a spacecraft has been examined using only two control torques provided by thrusters about two of the principle axis (Krishnan, 1992; Krishnan, Reyhanoglu, & McClamroch, 1994). A feedback control law for controlling an underactuated spacecraft has been developed with constraints on the control inputs (Tsiotras & Luo, 2000). Sliding control has been used to control an underactuated multibody spacecraft (Ashrafiuon & Erwin, 2004) and a sliding mode controller for use in maintaining a formation of underactuated satellites (Mcvittie, Kumar, Liu, & Candidate, 2010). Control algorithms have been developed using perturbation analysis and Lie group theory to design small-amplitude forces to control underactuated mechanical systems (Nordkvist, Bullo, & Member, 2008).

1.2 Self-Reconfigurable Control

Self-reconfigurable, variable structure approaches are widely used for the problems of dynamic systems control and observation due to their characteristics of finite time convergence, robustness to uncertainties, and insensitivity to external disturbances especially in sliding mode (DeCarlo, Zak, & Drakunov, 2011).

The main thrust of the sliding mode control research for many years has been in designing an appropriate sliding manifold to stabilize a nonlinear system. In the majority of cases, only one-component manifolds were considered, described by:

$$\sigma_i(t,x) = 0 \tag{1.1}$$

where $\sigma = col(\sigma_1, ..., \sigma_m)$ and the goal of the design was to make the system reach their intersection

$$\{\sigma(t,x) = 0\} = \bigcap_{i=1}^{m} \{\sigma_i(t,x)\} = 0 \tag{1.2}$$

The present work concentrates instead on the design of equilibrium sets in the state space with a more complicated structure than just the intersection of several one-component manifolds. The families of sliding (or potentially sliding) surfaces provide new opportunities for designing robust systems with new interesting properties.

Robust control of various classes of uncertain nonlinear systems has been widely researched in controls literature (Corless & Leitmann, 1981; Gutman, 1979; Qu, 1992, 1993; Marino & Tomei, 1993; Slotine & Hedrick, 1993; Kaloust & Qu, 1995). A particularly challenging class of uncertain systems are those containing uncertainty in the control sign (S. Drakunov, 1993). The control sign in this context represents the control direction - the control force or torque direction, for example, under any given control command. While many of the systems addressed in previous controls literature concern systems with known time-varying control direction, control design for systems with unknown control direction is a much more challenging task.

Sliding mode-based approach has been used to develop a robust controller in order to globally stabilize a system with unknown control direction (S. Drakunov, 1993; S. V. Drakunov, 1994). Applying this method is especially useful for mechanical systems acting in an unpredictable, uncertain environment. One of the successful examples is the design of the ABS system for the ground vehicles and landing aircrafts (S. V. Drakunov, Ozguner, Dix, & Ashrafi, 1995). Another approach used the so-called shifting laws, which are updated via online identification of the unknown control direction (Kaloust & Qu, 1995). Another approach proposed is based on monitoring functions to mitigate the difficulty in control direction uncertainty (Oliveira, Peixoto, & Liu, 2007, 2010). Another approach using monitoring functions employs a monitoring function-driven switching mechanism, which adjusts the control sign, assuming no a priori knowledge of the control direction (Hsu, Oliveira, & Peixoto, 2006).

The method that was presented in (S. Drakunov, 1993), uses a purely robust feedback technique to achieve finite-time convergence to a sliding manifold in the presence of unknown control direction. This robust feedback control design is much less computationally intensive because it requires no monitoring functions, function approximators, or online adaptive laws. By applying this computationally efficient control scheme with a compact dual-quaternion-based dynamic parametrization, an effective and versatile control method can be developed to achieve simultaneous translational and rotational control of a spatial rigid body.

To achieve computationally efficient six degrees of freedom (DOF) control (i.e., simultaneous translational and rotational control) of a spatial rigid-body system, choosing a proper position and orientation vector parametrization is critical. Three-element orientation vectors such as Euler angles can provide a unified representation of position and orientation; however, the Euler parametrization has inherent singularities in the parametrization. The unit quaternion has the benefit that it does not suffer from singularities. However, it has been shown that quaternion-based controller design can be complicated by the fact that the position and orientation errors are calculated separately (Pham, Perdereau, Adorno, & Fraisse, 2010; Xian, Dawson, de Queiroz, & Chen, 2004). To eliminate the need for control designs using two separate loops for controlling rotation and translation, dual quaternions are widely regarded as the most compact and efficient means for simultaneously representing translational and rotational motion (Aspragathos & Dimitros, 1998; Funda, Taylor, & Paul, 1990; Pham et al., 2010). The dual quaternion is an effective tool that it used in many applications, including robot manipulators (Aspragathos & Dimitros, 1998; Funda et al., 1990; Pham et al., 2010) inertial navigation (Wu et al., 2005), computer vision, and control of spatial rigid bodies (D. Han, Fang, & Wei, 2008).

By using a control architecture similar to that in (S. Drakunov, 1993), a sliding mode control approach is used in this work to achieve finite-time convergence to a sliding manifold for a class of dual-quaternion-based systems with unknown control input direction. The result is a robust control law that is rigorously proven to achieve finite-time convergence to a sliding manifold in the presence of unknown control direction without the use of function approximators or online parameter adaptation.

1.3 Nonlinear Control for Nonholonomic Driftless Systems

Also in this work control algorithms for a class of nonlinear driftless systems are investigated, originally introduced by Brockett (Brockett, 1981):

$$\dot{x} = B(x)u \tag{1.3}$$

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$. For example, a unicycle with direction and speed control inputs is prototypical system and the mobile robots are generalized unicycle systems with added kinematic constraints. Both systems are modeled by (5.1) and studied for control solutions in the literature (J. Wang et al., 2012; Medina-Garciadiego & Leonessa, 2011; Pathak & Agrawal, 2004; Zenkov, Bloch, & Marsden, 2002). The controllability condition for such systems is well established and can be easily tested via the accessibility algebra and the accessibility distribution (Isidori, 1997). An interesting class of controllable systems with the structure of (5.1) is the so-called kinematic nonholonomic system. Those systems are characterized by specific relation between the dimension of the system state x and the control input u. This current work relaxes this condition by extending the system space and designing control for the augmented system.

The method presented here addresses the possibility of such augmentation for systems that do not permit a Brockett's canonical form. The augmented system must be controllable and satisfy the hypotheses of the Brocket theorem listed in the next section. Therefore, the system can be controlled by various developed nonholonomic system controllers. The control architecture considered here is from (A. Bloch, Drakunov, & Kinyon, 1997) and (A. M. Bloch, Drakunov, & Kinyon, 2000).

This work discusses an example of a driftless nonholonomic system that can be augmented for the conversion to Brocket's canonical form and generalization of the proposed method. To show the proposed design framework, the work presents a prototypical unicycle system, including the example of controlling multiple moving robots with one of the controls being common for the whole swarm, and its simulations. Also presented is the example of multiple underactuated spacecraft represented using dual quaternions.

Chapter 2

Rigid Body Motion

In order to discuss rigid body motion using dual quaternions, it is important first to establish a foundation. A single dimensionless particle's position can be described using a Cartesian frame, usually consisting of a point projected onto three orthonormal axes $(x, y, z) \in \mathbb{R}^3$. The motion of a particle can then be represented by a continuous curve $p(t) = (x(t), y(t), z(t)) \in \mathbb{R}^3$. This concept is extended from a single particle to a completely "undistortable" collection of particles, referred to as a rigid body, where the distance between any two particles remains constant. If p and q represent any two particles of a rigid body, then the definition of a rigid body can be defined mathematically as:

$$\|p(t) - q(t)\| = \|p(0) - q(0)\| = constant \tag{2.1}$$

for all $t \geq 0$ (Murray, Li, & Sastry, 1994).

Rigid motion, consisting of both rotation and translation, is the continuous movement of a rigid body, meaning the distance between any two particles remains constant throughout the motion. A rigid body can be represented by the subset O where $O \subset \mathbb{R}^3$ and rigid motion represented using a family of continuous mappings $g(t) : O \to \mathbb{R}^3$ describing the motion of the individual particles in a fixed Cartesian coordinate frame. For the mapping $g(t) : O \to \mathbb{R}^3$ to describe rigid motion, a necessary condition must be that the mapping maintains the distance between points of the rigid body. However, this condition is not sufficient since it does not account for internal reflections, i.e. the mapping $(x, y, z) \to (x, y, -z)$ maintains distances between points but reflects points inside the rigid body about the xy plane. Therefore, a second necessary condition that the cross product of vectors between points in the rigid body be preserved is required. A rigid transformation is defined as a mapping $g : \mathbb{R}^3 \to \mathbb{R}^3$ satisfying these two conditions (Murray et al., 1994):

1. Distance is preserved: $\|g(p) - g(q)\| = \|p - q\|$ for all points $p, q \in O \subset \mathbb{R}^3$.
2. The cross product is preserved: $g_*(v \times w) = g_*(v) \times g_*(w)$ for all $v, w \in \mathbb{R}^3$ and where $v = p_1 - q_1$, $w = p_2 - q_2$, and $p_1, p_2, q_1, q_2 \in O \subset \mathbb{R}^3$, and $g_*(v) = g(p) - g(q)$.

2.1 Rotations

The definition of a rigid body rotation allows for particles in a rigid body to rotate but not translate with respect to each other. To describe this rotation, the orientation between a fixed coordinate frame attached to the rigid body and a fixed inertial frame is used. In Fig.2.1, reference frame A, consisting of three orthogonal unit vectors $\{\vec{a}_1, \vec{a}_2, \vec{a}_3\}$, refers to a fixed inertial frame and reference frame B, consisting of the unit vectors $\{\vec{b}_1, \vec{b}_2, \vec{b}_3\}$, refers to the body frame. The unit vectors of B can be defined in terms of frame A by:

$$\begin{aligned} \vec{b}_1 &= R_{11}\vec{a}_1 + R_{12}\vec{a}_2 + R_{13}\vec{a}_3 \qquad (2.2) \\ \vec{b}_2 &= R_{21}\vec{a}_1 + R_{22}\vec{a}_2 + R_{23}\vec{a}_3 \\ \vec{b}_3 &= R_{31}\vec{a}_1 + R_{32}\vec{a}_2 + R_{33}\vec{a}_3 \end{aligned}$$

where $R_{ij} \equiv \vec{b}_i \cdot \vec{a}_j$ is the cosine of the angle between $\vec{b}_i$ and $\vec{a}_j$, referred to as the direction cosine. A rotation matrix, $\mathbf{R}^{B/A} \in \mathbb{R}^{3\times 3}$ is also referred to as a direction cosine matrix and can be used to express the relation in (2.2) where $\mathbf{R}^{B/A} \equiv [R_{ij}]$ (Wie, 1998). Rotational matrices formed in this manner are orthogonal with $\det \mathbf{R} = \pm 1$ and when defined in right hand coordinate systems the $\det \mathbf{R} = +1$.

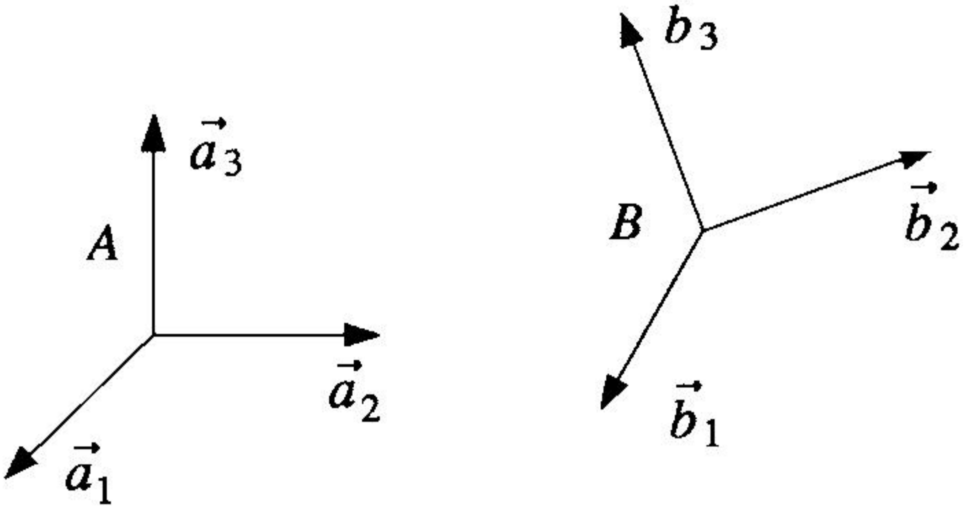

Figure 2.1: Fixed reference frame A and rotated frame B (Wie, 1998).

The space of rotation matrices in $\mathbb{R}^{n\times n}$ can be defined by the Lie group $SO(n) \subset$

$\mathbb{R}^{n\times n}$ as:

$$SO(n) = \left\{\mathbf{R} \in \mathbb{R}^{n\times n} : \mathbf{R}^T\mathbf{R} = \mathbf{I}, \det \mathbf{R} = +1\right\} \tag{2.3}$$

where SO refers to *special orthogonal* group, with *special* meaning that $\det \mathbf{R} = +1$. The group $SO(3) \subset \mathbb{R}^{3\times 3}$ describes the rotational group of $\mathbb{R}^3$ with the following properties (Murray et al., 1994):

1. *Closure:* If $\mathbf{R}_1, \mathbf{R}_2 \in SO(3)$, then $\mathbf{R}_1\mathbf{R}_2 \in SO(3)$.
2. *Identity:* The identity element of the group $SO(3)$ is the identity matrix $\mathbf{I} \in \mathbb{R}^{3\times 3}$.
3. *Inverse:* The inverse of $\mathbf{R} \in SO(3)$ is $\mathbf{R}^T \in SO(3)$, i.e. $\mathbf{R}\mathbf{R}^T = \mathbf{R}^T\mathbf{R} = \mathbf{I}$.
4. *Associativity:* The associativity of group $SO(3)$ follows from the associativity of matrix multiplication, i.e. $(\mathbf{R}_1\mathbf{R}_2)\,\mathbf{R}_3 = \mathbf{R}_1\,(\mathbf{R}_2\mathbf{R}_3)$.

Similarly, the group $SO(n)$ of $n \times n$ orthogonal matrices with $\det = 1$ is a matrix Lie group.

Definition 1. In general, a Lie group is a differentiable manifold G which is also a group such that the group product $G \times G \to G$ and the inverse map $g \to g^{-1}$ are differentiable (Murray et al., 1994).

Definition 2. A matrix Lie group is any group G with the property that if $A_m \in G$ and A_m converges to some matrix A that either $A \in G$ or A is not invertible (Hall, 2003).

$SO(n)$ is a differential manifold where $SO(n) \subset GL(n;\mathbb{R})$ where $GL(n;\mathbb{R})$ is the *general linear* matrix Lie group consisting of all $n \times n$ invertible matrices with real entries (Hall, 2003).

Euler Angles as a representation of $SO(3)$

A common method for representing $SO(3)$ rotations is the use of Euler angles. Euler angles consist of three successive angles of rotation, each about one of the axes of the rotated body-fixed reference frame. The first rotation is about any axes, the second rotation is about one of two axes not used in the first rotation, and the third rotation is about of the two axes not used in the second rotation. There are 12 possible combinations of axes rotations that can be used, each represented by three letters corresponding the axes used and the order in which they are used (eg: ZYZ, XYZ, ZYX, etc.) (Wie, 1998). The ZYX Euler angles are commonly used in the aerospace field, with the x-axis rotation referred to as roll (ϕ), the y-axis rotation referred to as pitch (θ), and the z-axis rotation referred to as yaw (ψ), shown in Fig. 2.2.

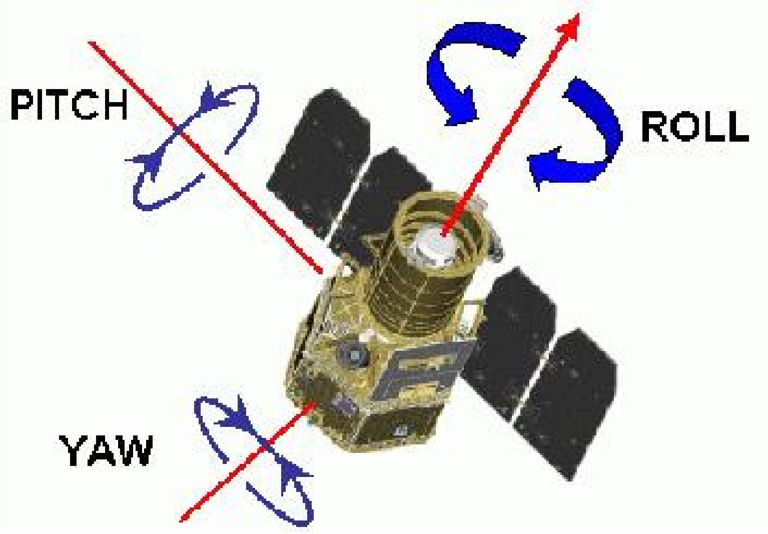

Figure 2.2: Euler angles associated with roll (ϕ), pitch (θ), and yaw (ψ) (NASA JPL, 2004).

Three separate rotation matrices, each representing a single rotation about a single principle axis, are combined to form a full 3-dimensional rotation matrix $\mathbf{R}^{B/A}$. The combined rotations for the ZYX Euler representation is shown in (2.4a).

$$\mathbf{R}^{B/A} = \mathbf{R}_z(\psi)\mathbf{R}_y(\theta)\mathbf{R}_x(\phi) \tag{2.4a}$$

$$\mathbf{R}^{B/A} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \tag{2.4b}$$

$$\mathbf{R}^{B/A} = \begin{bmatrix} \cos\psi\cos\theta & \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi & \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi \\ \cos\theta\sin\psi & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & \sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix} \tag{2.4c}$$

The Euler angle representation is surjective to $SO(3)$ allowing the Euler angles to be computed from a given rotation matrix $\mathbf{R} \in SO(3)$. Using (2.4c) the solutions for the ZYX rotation are given in (2.5).

$$\phi = \arctan_2{}^{1} \frac{R_{32}}{R_{33}} \tag{2.5a}$$

$$\theta = \arctan_2 \frac{-R_{31}}{\sqrt{R_{11}^2 + R_{21}^2}} \tag{2.5b}$$

$$\psi = \arctan_2 \frac{R_{21}}{R_{11}} \tag{2.5c}$$

It is a fundamental feature of $SO(3)$ that any three dimensional representation will have a singularity for a particular rotation. In the case of ZYX Euler angles, it occurs

[1] The $\arctan_2 \frac{a}{b}$ is similar to $\tan^{-1} \frac{a}{b}$ but uses the sign of a and b to determine the quadrant of the resultant angle.

at $\theta = \pm\frac{\pi}{2}$ (Murray et al., 1994). This is related to the fact that $SO(3)$ is not simply-connected. Another way to visualize this singularity is to examine the Euler angle kinematics, shown in (2.6).

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \frac{1}{\cos\theta} \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \cos\theta\sin\psi & \cos\theta\cos\psi & 0 \\ -\sin\theta\cos\psi & \sin\theta\sin\psi & \cos\theta \end{bmatrix} \vec{\omega} \tag{2.6}$$

The vector $\vec{\omega}$ represents the angular rate vector.

Definition 3. For a matrix Lie group G, the Lie algebra of G, denoted $\mathfrak{g}$, is the set of all matrices X such that e^{tX} is in G for all real numbers t (Hall, 2003).

The Lie algebra $so(n)$ associated with the matrix Lie group $SO(n)$ can be used to represent rotations using a single unit vector $\omega \in \mathbb{R}^3$ which specifies the direction of rotation and $\theta \in \mathbb{R}$, representing the angle of rotation in radians.

$SO(n)$ in general can be expressed in the form $e^{\omega^\times \theta}$ where $\omega^\times \in so(n)$ is a skew-symmetric matrix satisfying $\omega^{\times^T} = -\omega^\times$ whose elements come from $\vec{\omega}$ by the following relation:

$$\omega^\times = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \tag{2.7}$$

where $\vec{\omega} = [\omega_1 \; \omega_2 \; \omega_3]^T$. This $so(n)$ space is defined as

$$so(n) = \left\{ \omega^\times \in \mathbb{R}^{n\times n} : \omega^{\times^T} = -\omega^\times \right\}. \tag{2.8}$$

Specifically, the set $so(3) \subset \mathbb{R}^{3\times 3}$ is a vector space with respect of standard matrix summation and multiplication by a scalar since the sum of two elements of $so(3)$ is an element of $so(3)$ and the scalar multiple of any element of $so(3)$ is also an element of $so(3)$. Any element of $so(3)$ is represented by a unit skew-symmetric matrix ($\omega^\times \in so(3), \|\vec{\omega}\| = 1$) and a real number $\theta \in \mathbb{R}$ (Murray et al., 1994). Using this representation, the matrix exponential of $so(3)$ can be reduced as follows:

$$\begin{aligned} e^{\omega^\times \theta} &= I + \left(\theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} - \cdots\right)\omega^\times + \left(\frac{\theta^2}{2!} - \frac{\theta^4}{4!} + \frac{\theta^6}{6!} - \cdots\right)\omega^{\times^2} \\ e^{\omega^\times \theta} &= I + \omega^\times \sin\theta + \omega^{\times^2}(1 - \cos\theta) \qquad (2.9) \end{aligned}$$

where (2.9) is referred to as Rodrigues' Formula (Murray et al., 1994). In summary any rotation matrix $\mathbf{R} \in SO(3)$ can be expressed by a single rotation vector $\vec{\omega} \in \mathbb{R}^3, \|\omega\| = 1$ and a single rotation angle $\theta \in \mathbb{R}$ as shown in Fig. 2.3. This is also known as Euler's Eigenaxis Theorem (Wie, 1998) which states that any rotation can be represented by a single rotation (θ) about a single Euler Eigenaxis ($\vec{\omega}$). The rotation matrix is expressed in the form $\mathbf{R} = e^{\omega^\times \theta}$ where $\omega^\times \in so(3) \subset \mathbb{R}^{3\times 3}$.

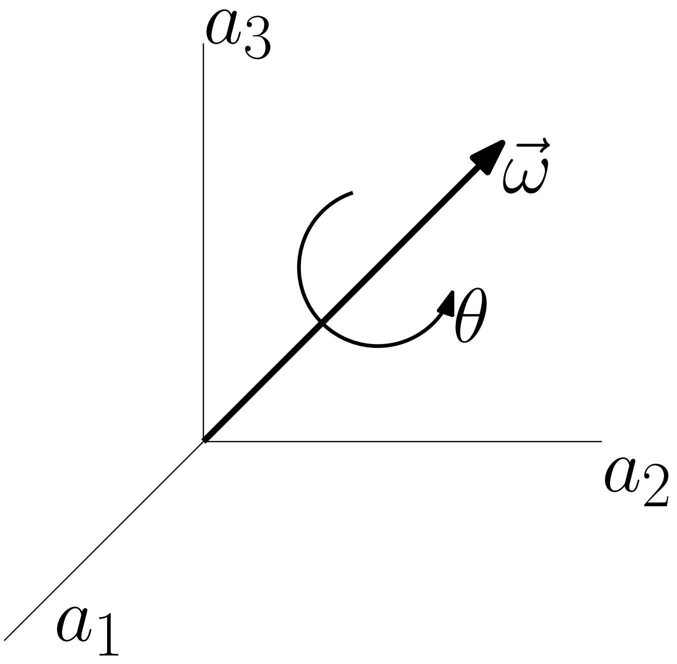

Figure 2.3: Representation of Euler's Eigenaxis Theorem.

2.1.1 Quaternions

Quaternions were invented by Hamilton in 1843 as an extension of the complex numbers to a hypercomplex numbers ($\mathbf{q} = q_0 + q_1 i + q_2 j + q_3 k \in \mathbb{H}$) consisting of a real part (q_0) and a three dimensional complex part ($q_1 i + q_2 j + q_3 k$) (Hamilton, 1844). Hamilton invented the quaternions to allow for division algebra among Euclidean vectors. Multiplication of the individual complex components uses the following rules

$$
\begin{aligned}
i^2 = j^2 = k^2 = ijk = -1 \qquad (2.10)\\
ij = -ji = k \qquad (2.11)\\
jk = -jk = i\\
ki = -ki = j.
\end{aligned}
$$

Quaternions can be also be represented as a combination of a scalar and three dimension vector as follows

$$\mathbf{q} = [s, \vec{\mathbf{q}}], \qquad (2.12)$$

where $s = q_0$ is the scalar part and $\vec{\mathbf{q}} = q_1 i + q_2 j + q_3 k$ is the vector part with imaginary unit elements. Basic operations of quaternions are given as follows

$$
\begin{aligned}
\mathbf{q_a} + \mathbf{q_b} &= [s_a + s_b, \vec{\mathbf{q}}_\mathbf{a} + \vec{\mathbf{q}}_\mathbf{b}] \qquad (2.13)\\
\lambda \mathbf{q} &= [\lambda s, \lambda \vec{\mathbf{q}}]\\
\mathbf{q_a}\mathbf{q_b} &= \left[s_a s_b - \vec{\mathbf{q}}_\mathbf{a}^\mathbf{T} \vec{\mathbf{q}}_\mathbf{b}, s_b \vec{\mathbf{q}}_\mathbf{a} + s_a \vec{\mathbf{q}}_\mathbf{b} + \vec{\mathbf{q}}_\mathbf{a} \times \vec{\mathbf{q}}_\mathbf{b}\right],
\end{aligned}
$$

where λ is a scalar. Of note, a feature of quaternions is that the product of two quaternions is noncommutative. The conjugate of $\mathbf{q}$ is defined as $\mathbf{q}^* = [s, -\vec{\mathbf{q}}]$ and the norm is $\|\mathbf{q}\| = \sqrt{\mathbf{q}\mathbf{q}^*}$. In summary, the algebra of $\mathbb{H}$ is a noncommutative associative division algebra over the reals as well as a normed division algebra.

Quaternions can be used in lieu of other methods such as direction cosine matrixes to represent attitude rotations. Unit quaternions, $\mathbb{H}_u$, are quaternions where $\|\mathbf{q}\| = \mathbf{1}$, and can used to represent rotation. Utilizing Euler's Eigenaxis Theorem, instead of using multiple angular rotations to describe a rotation, a single rotation angle θ about a unit eigenaxis $\vec{\mathbf{n}}$ can be used. A example rotation in the 3D ijk space is shown in Fig. 2.4. The red arrows show how i is rotated into j, j into k and k into i. The rotational unit quaternion, defined as

$$\mathbf{q} = \left[\cos\left(\frac{\theta}{2}\right), \sin\left(\frac{\theta}{2}\right)\vec{\mathbf{n}}\right] \tag{2.14}$$

relates a fixed $\vec{p}$ in an original frame B to a new frame A by

$$\mathbf{p_a} = \mathbf{q}^*\mathbf{p_b}\mathbf{q} \tag{2.15}$$

where $\mathbf{p_a}$ and $\mathbf{p_b}$ are vector quaternions with zero scalar part, i.e. $\mathbf{p} = [0, \vec{p}]$.

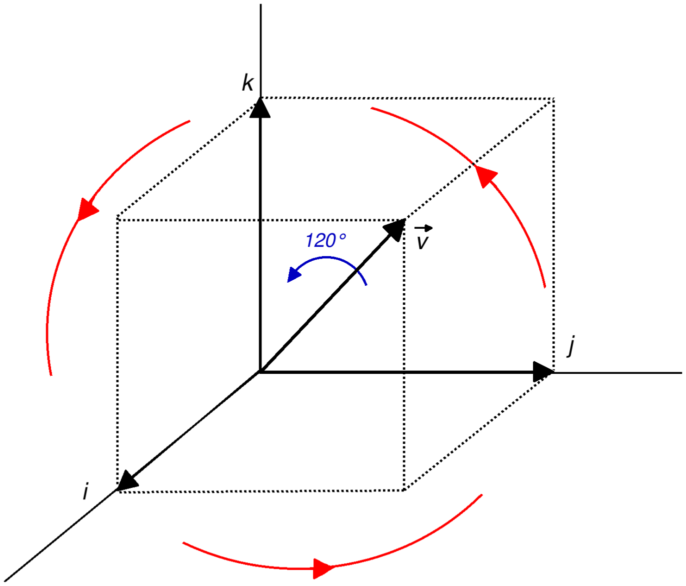

Figure 2.4: A single 120^o quaternion rotation about the first diagonal in the 3D ijk space (MathsPoetry, 2009).

The unit quaternions $\mathbb{H}_u$ are a representation of the *spin* Lie group S^3 which forms a 3-sphere in $\mathbb{R}^4$ which is isomorphic to the matrix Lie group $SU(2)$. SU refers to the *special unitary* matrix Lie group which consists of the space of matrices $\mathbf{U} \in \mathbb{C}^{n \times n}$ defined by (Hall, 2003):

$$SU(n) = \left\{ \mathbf{U} \in \mathbb{C}^{n \times n} : \mathbf{U}^\dagger \mathbf{U} = \mathbf{I}, \det \mathbf{U} = +1 \right\}. \tag{2.16}$$

Quaternions can be represented as a pair of complex numbers $\alpha = q_0 + q_1 i$ and $\beta = q_2 + q_3 i$ such that

$$\mathbf{q} = \alpha + \beta j = (q_0 + q_1 i) + (q_2 + q_3 i)j. \tag{2.17}$$

where $|\alpha|^2 + |\beta|^2 = 1$. This pair of complex numbers for quaternions can than be easily represented by $SU(2)$ as (D. Han, Fang, & Wei, 2008):

$$SU(2) = \begin{bmatrix} \alpha & \bar{\beta} \\ -\bar{\beta} & \bar{\alpha} \end{bmatrix} . \mathbb{H}_u \tag{2.18}$$

The group $SU(2)$ is simply-connected, and therefore no singularities appear for any orientation. $SU(2)$ is *almost* isomorphic to $SO(3)$, $SU(2)$ can be mapped onto $SO(3)$, but it is two to one (Hall, 2003). The direction cosine matrix $\mathbf{R} \in SO(3)$ can be obtained from the unit quaternion $\mathbf{q}$ by (Wie, 1998):

$$\mathbf{R} = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_3 q_0) & 2(q_1 q_3 + q_2 q_0) \\ 2(q_1 q_2 + q_3 q_0) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 - q_1 q_0) \\ 2(q_1 q_3 - q_2 q_0) & 2(q_2 q_3 + q_1 q_0) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \tag{2.19}$$

The fact that $SU(2)$ has a two to one to correspondence with $SO(3)$ creates some ambiguity (i.e. there are two unit quaternions, $\mathbf{q}$ and $-\mathbf{q}$, that represent the same physical orientation). In order to create a one to one correspondence from $\mathbb{H}_u$ to $SO(3)$, a *normalization* can be performed such that

$$\mathcal{N}(\mathbf{q}) = \begin{cases} \mathbf{q}, & \text{if } q_0 \geq 0 \\ -\mathbf{q}, & \text{otherwise} \end{cases} \tag{2.20}$$

where $\mathcal{N}(\mathbf{q})$ is the *normalization* of $\mathbf{q}$. This new set $\{\mathcal{N}(\mathbf{q})\}$ does have a one to one correspondence with $SO(3)$ (D. Han, Fang, & Wei, 2008).

The rotational kinematic equation for quaternions, which has no singularities, is given as

$$\dot{\mathbf{q}} = \frac{1}{2}\mathbf{q}\boldsymbol{\omega}, \tag{2.21}$$

where $\boldsymbol{\omega}$ is the quaternion of the angular rate vector ($\boldsymbol{\omega} = [0, \vec{\omega}]$) of the body frame relative to an inertial frame expressed in the body frame (Wie, 1998).

A useful function for quaternions is the logarithm (Kim, Kim, & Shin, 1996). The general logarithm for quaternions is defined as

$$\log \mathbf{q} = \left[0, \frac{\cos^{-1} s}{\sqrt{1-s^2}}\vec{\mathbf{q}}\right], s \neq 1. \tag{2.22}$$

Using 2.14, the logarithm of a unit quaternion is:

$$\log \mathbf{q} = \left[0, \frac{\theta}{2}\vec{\mathbf{n}}\right]. \tag{2.23}$$

This leads to the exponential form of $\mathbb{H}_u$ which forms the Lie algebra for unit quaternions and is of the form:

$$\mathbf{q} = e^{\left[0, \frac{\theta}{2}\mathbf{n}\right]}. \tag{2.24}$$

The unit quaternion logarithm in (2.23) is a vector quaternion (i.e. it's real scalar term is zero), so an approximation is used for simplicity:

$$\log \mathbf{q} = \frac{\theta}{2}\vec{\mathbf{n}} \tag{2.25}$$

yielding a normal $\mathbb{R}^3$ vector. When $\theta = 0, \mathbf{q} = \left[1, \vec{\mathbf{0}}\right]$ or when $\theta = 2\pi, \mathbf{q} = \left[-1, \vec{\mathbf{0}}\right]$ the $\log \mathbf{q} = \vec{\mathbf{0}}$ by convention.

2.2 Translation

Rigid body transformations that consist purely of translation are simple compared to transformations that consist purely of rotation. Given a point $q \in \mathbb{R}^3$, represented in a translated B frame as $q_b \in \mathbb{R}^3$, to be represented in the fixed frame A as $q_a \in \mathbb{R}^3$ is as straightforward as:

$$q_a = \vec{p}_{ab} + q_b \tag{2.26}$$

where $\vec{p}_{ab} \in \mathbb{R}^3$ represents a position vector of the origin of the translated frame B from the origin of the fixed frame A. This relationship is shown in Fig.2.5. Reference frame B has been translated from the fixed reference frame A by $\vec{p}_{ab}$. Point q can be represented in frame A using (5.40).

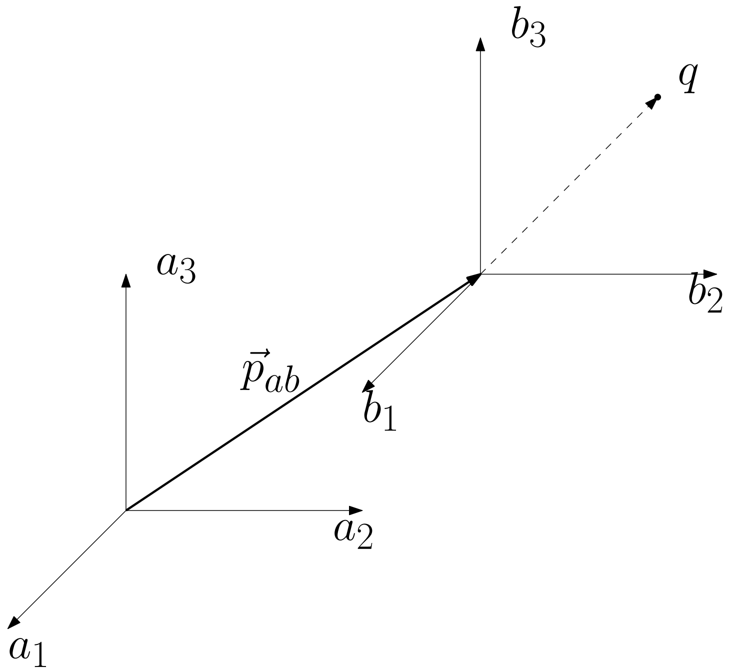

Figure 2.5: Representation of translation in $\mathbb{R}^3$.

The transformation for translation in (5.40) can be combined with the transformation for rotation in (2.2) to yield a transformation for for translation and rotation, shown in Fig.2.6. Reference frame B has been translated from the fixed reference frame A by $\vec{p}_{ab}$ and rotated by $\mathbf{R}^{A/B}$. Point q can be represented in frame A using (2.27). The transformation can be defined by the following:

$$q_a = \vec{p}_{ab} + \mathbf{R}^{A/B} q_b \tag{2.27}$$

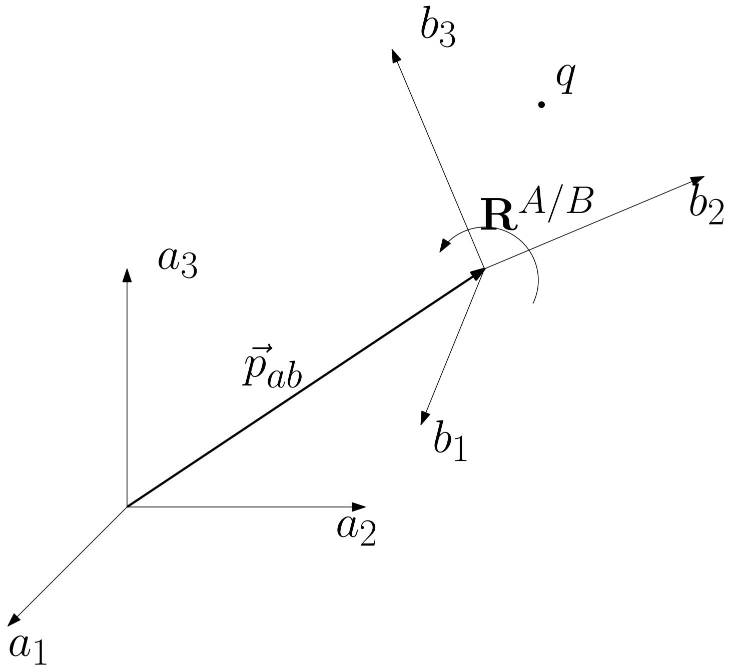

Figure 2.6: Representation of both translation and rotation in $\mathbb{R}^3$.

The *special Euclidean* matrix Lie group $SE(n) \subset \mathbb{R}^{(n+1)\times(n+1)}$ can be used to represent rigid body transformation by using the product space of $\mathbb{R}^n$ with $SO(n)$ where $SE(n) \triangleq \mathbb{R}^n \times SO(n)$. For the specific case of 3-dimensional rigid body motion:

$$SE(3) \triangleq \left\{(\vec{p}, \mathbf{R}) : \vec{p} \in \mathbb{R}^3, \mathbf{R} \in SO(3) \subset \mathbb{R}^{3\times3}\right\} = \mathbb{R}^3 \times SO(3). \tag{2.28}$$

For example, the transformation described in (2.27) and shown in Fig.2.6 can be represent by the homogenous transformation matrix(Murray et al., 1994):

$$\mathbf{T}^{A/B} = \begin{bmatrix} \mathbf{R}^{A/B} & \vec{p}_{ab} \\ \mathbf{0}^{1\times3} & 1 \end{bmatrix} \tag{2.29a}$$

$$\begin{bmatrix} q_{a_1} \\ q_{a_2} \\ q_{a_3} \\ 1 \end{bmatrix} = \mathbf{T}^{A/B} \begin{bmatrix} q_{b_1} \\ q_{b_2} \\ q_{b_3} \\ 1 \end{bmatrix} \tag{2.29b}$$

In order to transform points in $\mathbb{R}^3$, the 3×1 vector used to represent the point must be augmented to a 4×1 vector by adding a 1 in the fourth element, as shown in (2.29b). Vectors can also be transformed in a similar matter by augmenting their fourth element with a 0. The identity element for $SE(3)$ is $\mathbf{I} \in \mathbb{R}^{4\times4}$ and the inverse is defined as:

$$\mathbf{T}^{B/A} = \begin{bmatrix} \mathbf{R}^{B/A} & -\mathbf{R}^{B/A}\vec{p}_{ab} \\ \mathbf{0}^{1\times3} & 1 \end{bmatrix} \tag{2.30}$$

where $\mathbf{R}^{B/A} = \mathbf{R}^{A/B^T}$.

Chasles Theorem, also referred to as screw theory, which states that every rigid body transformation can be realized by a rotation about an axis combined with a translation parallel to that axis (Murray et al., 1994). Fig. 2.7 illustrates the concept of Screw theory, which states that a rigid body transformation can be described by a single rotation (θ) about an axis ($\vec{\omega}$) combined with a translation ($\vec{d}$) parallel to that axis.

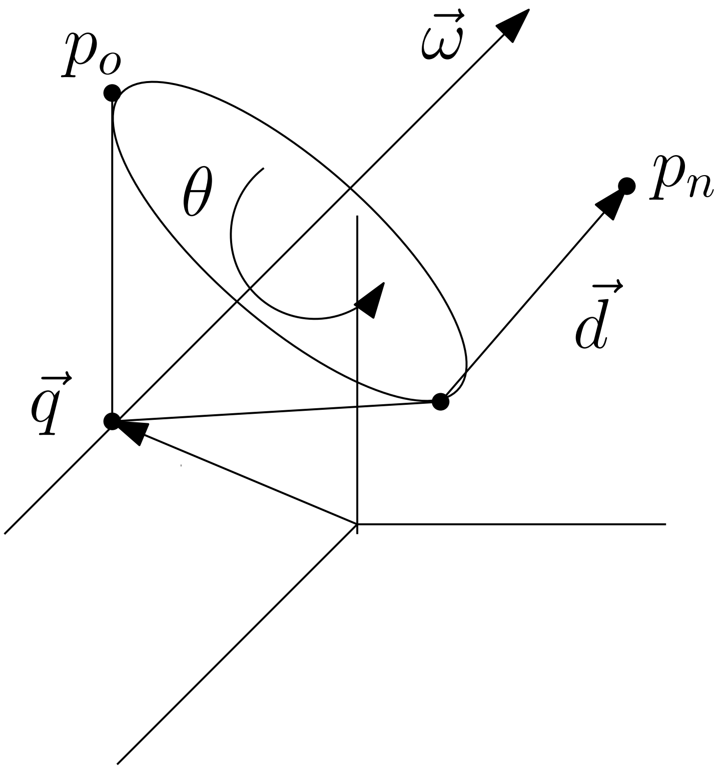

Figure 2.7: Representation of Chasles Theorem or Screw theory.

The Lie algebra for $SE(3)$, defined as the following, can be used to describe this concept:

$$se(3) \triangleq \left\{ \left(\vec{v}, \omega^{\times}\right) : \vec{v} \in \mathbb{R}^{3}, \omega^{\times} \in so(3) \subset \mathbb{R}^{3\times3} \right\}. \tag{2.31}$$

In homogenous coordinates, an element of $se(3)$ is represented as:

$$\xi^{\times} = \begin{bmatrix} \omega^{\times} & \vec{v} \\ \mathbf{0}^{1\times3} & 0 \end{bmatrix} \in \mathbb{R}^{4\times4} \tag{2.32}$$

where $\omega^{\times}$ is the skew-symmetric representation of the rotation axis $\vec{\omega}$ and $\vec{v} = -\vec{\omega} \times \vec{q}$, shown in Fig. 2.7. The exponential mapping of the Lie algebra $se(3)$ onto its Lie group $SE(3)$ is as follows:

$$e^{\xi^\times \theta} = \begin{bmatrix} e^{\omega^\times \theta} & (\mathbf{I} - e^{\omega^\times \theta})(\vec{\omega} \times \vec{v}) + \vec{\omega}\vec{\omega}^T \vec{v}\theta \\ \mathbf{0}^{1\times 3} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \vec{p} \\ \mathbf{0}^{1\times 3} & 1 \end{bmatrix} \tag{2.33}$$

where $\vec{\omega}$ and θ come from the $SO(3)$ exponential form $\mathbf{R} = e^{\omega^\times \theta}$ (Murray et al., 1994). This leaves solving the equation

$$\mathbf{A}\vec{v} = \vec{p} \tag{2.34a}$$

$$\mathbf{A} = (\mathbf{I} - e^{\omega^\times \theta})\omega^\times + \vec{\omega}\vec{\omega}^T \theta \tag{2.34b}$$

for $\vec{v}$ to determine all the elements in (2.32). The matrix $\mathbf{A}$ in (2.34b) is nonsingular for all $\theta \in \mathbb{R}$.

2.2.1 Dual Numbers

Dual numbers were invented by Clifford (Clifford, 1873) in 1873 and further developed for the use in rigid body motion by Study (Study, 1891). One can formally extend the set of real numbers $\mathbb{R}$ by adding a *dual factor* ϵ with nilpotent property $\epsilon^2 = 0$ while $\epsilon \neq 0$. The set of such numbers $\mathbb{DR}$ is called a set of dual numbers defined as

$$\hat{x} = x + \epsilon x'. \tag{2.35}$$

The dual element can also be thought of as a linear operator

$$\epsilon = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \tag{2.36}$$

giving the dual number $\hat{x}$ from (2.35) the form

$$\hat{x} = \begin{bmatrix} x & 0 \\ x' & x \end{bmatrix}. \tag{2.37}$$

The dual numbers form a two-dimensional associative commutative algebra, satisfying the following basic operations

$$\hat{a} + \hat{b} = (a + b) + \epsilon\,(a' + b') \tag{2.38a}$$

$$\lambda\hat{a} = \lambda a + \epsilon\lambda a' \tag{2.38b}$$

$$\hat{a}\hat{b} = ab + \epsilon\,(ba' + ab') \tag{2.38c}$$

$$\hat{a}^* = a - \epsilon a' \tag{2.38d}$$

$$\frac{\hat{a}}{\hat{b}} = \frac{\hat{a}\hat{b}^*}{\hat{b}\hat{b}^*} = \frac{a}{b} + \epsilon\left(\frac{a'}{b} - \frac{ab'}{b^2}\right),\ b \neq 0 \tag{2.38e}$$

where λ is a scalar (Wu & Hu, 2006). As seen from (2.38e), division for dual numbers is not well defined and only exists when the real term of the divisor does not equal zero.

Dual vectors are vectors whose elements are dual numbers or, alternatively, are dual numbers whose real and dual parts are both vectors.

$$\hat{\vec{v}} = \vec{v} + \epsilon \vec{v}' \tag{2.39}$$

Unit dual vectors provide a method of representing lines in free space, ($\hat{\vec{l}} = \vec{v} + \epsilon\vec{m}$). The real term represents the unit direction vector ($\vec{v}$) of the line and the dual part is the line moment ($\vec{m} = \vec{q} \times \vec{v}$) where the vector $\vec{q}$ is vector from the reference frame origin to a point on the vector $\vec{v}$, and the moment $\vec{m}$ is normal to the plane passing through the origin and $\vec{v}$ (Wu et al., 2005). This representation is also referred as a Plücker coordinate for a line, shown in Fig.2.8. Plücker Coordinate for a line consists of a dual vector ($\hat{\vec{l}} = \vec{v} + \epsilon\vec{m}$) where the real component ($\vec{v}$) represents the unit direction vector of the line and the dual component is the line moment ($\vec{m} = \vec{q} \times \vec{v}$).

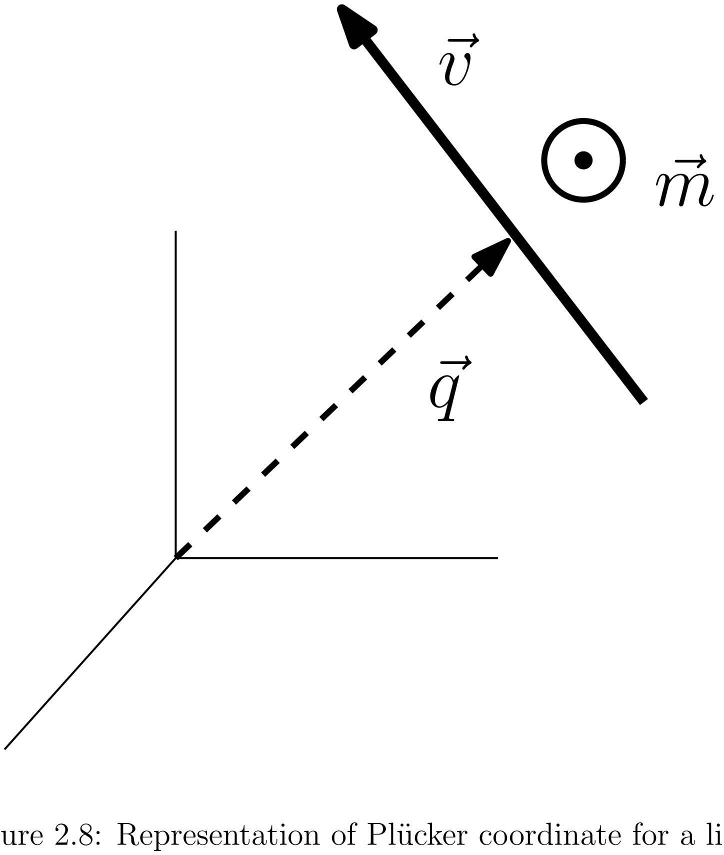

Figure 2.8: Representation of Plücker coordinate for a line.

These Plücker coordinates can be used to define the screw axis from (2.32) as $\hat{\vec{\xi}} = \vec{\omega} + \epsilon\vec{v}$, shown in Fig. 2.9. The screw axis can be represented by a Plücker line coordinate, $\hat{\vec{\xi}} = \vec{\omega} + \epsilon\vec{v}$ where $\vec{v} = -\vec{\omega} \times \vec{q}$. The dual angle $\hat{\theta} = \theta + \epsilon d$ describes the motion about the screw axis.

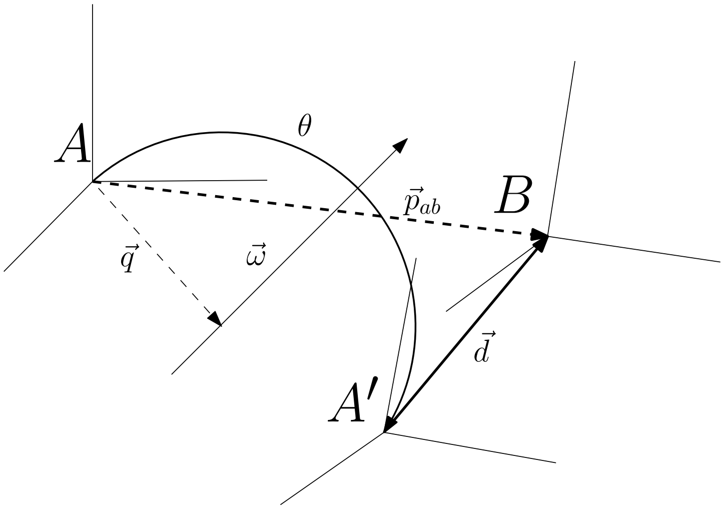

Figure 2.9: Plücker coordinate representation of screw axis and rotation.

The basic operation for dual vectors are as follows

$$\hat{\vec{v}}_1 = \vec{v}_1 + \epsilon\vec{v}_1' \qquad\qquad \hat{\vec{v}}_2 = \vec{v}_2 + \epsilon\vec{v}_2'$$

$$\hat{\vec{v}}_1^T\hat{\vec{v}}_2 = \vec{v}_1^T\vec{v}_2 + \epsilon\left(\vec{v}_1^T\vec{v}_2' + \vec{v}_1'^T\vec{v}_2\right) \tag{2.40a}$$

$$\hat{\vec{v}}_1 \times \hat{\vec{v}}_2 = \vec{v}_1 \times \vec{v}_2 + \epsilon\left(\vec{v}_1 \times \vec{v}_2' + \vec{v}_1' \times \vec{v}_2\right). \tag{2.40b}$$

These vector operations can be applied to Plücker lines to develop useful relationship between screws. The dual vector dot product from (2.40a) is equivalent to the cosine of the dual angle $\hat{\theta} = \theta + \epsilon d$

$$\hat{\vec{l}}_1^T\hat{\vec{l}}_2 = \cos\hat{\theta} \tag{2.41}$$

where $\hat{\vec{l}}_1 = \vec{l}_1 + \epsilon\vec{m}_1$, $\hat{\vec{l}}_2 = \vec{l}_2 + \epsilon\vec{m}_2$, θ is the crossing angle and d the perpendicular distance between the Plücker lines, shown in Fig.2.10. The cross product between two Plücker lines, defined as

$$\hat{\vec{l}}_1 \times \hat{\vec{l}}_2 = \sin\hat{\theta}\hat{\vec{n}} \tag{2.42}$$

produces a new Plücker line $\hat{\vec{n}}$ which represents the common perpendicularly intersecting lines in the direction of $\vec{l}_1 \times \vec{l}_2$, also shown in Fig. 2.10. The dot product and cross product of two Plücker lines $(\hat{\vec{l}}_1, \hat{\vec{l}}_2)$ can be used to represent the relationship between the two lines. θ is the crossing angle, d is the perpendicular distance between the lines, and the dual vector $\hat{\vec{n}}$ is the common perpendicularly intersecting line in the direction of $\vec{l}_1 \times \vec{l}_2$ (Wu et al., 2005).

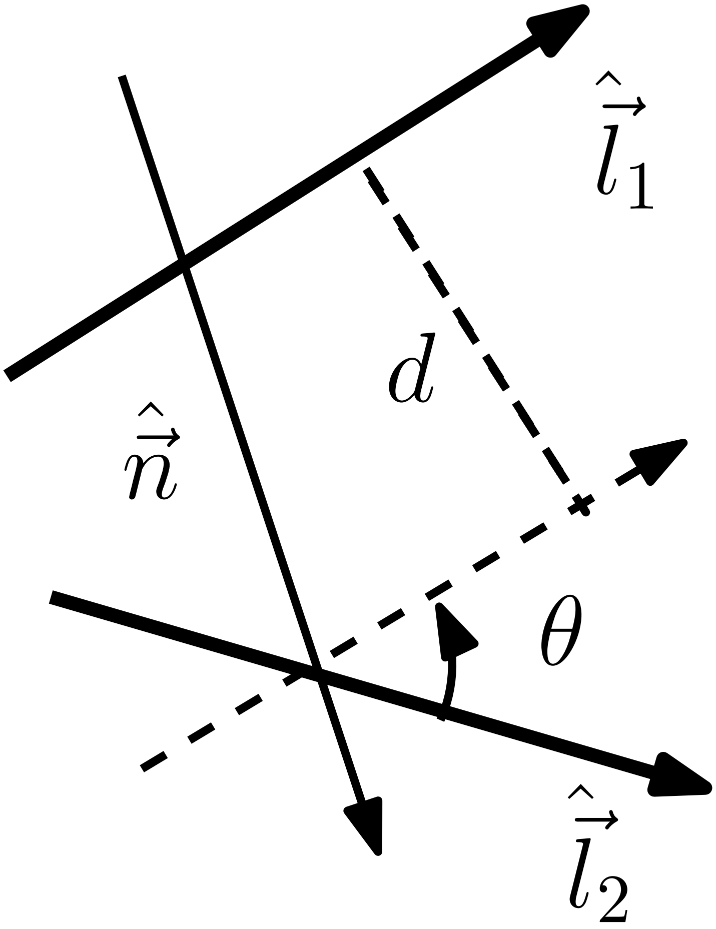

Figure 2.10: Representation of dual vector dot and cross products.

A dual matrix consists of a real and dual matrix, or a matrix of dual numbers. Multiplication of dual vectors and matrices are performed as follows

$$\hat{\mathbf{A}}_\mathbf{1} = \mathbf{A_1} + \epsilon \mathbf{A'_1} \qquad \hat{\mathbf{A}}_\mathbf{2} = \mathbf{A_2} + \epsilon \mathbf{A'_2}$$

$$\hat{\mathbf{A}}_\mathbf{1}\hat{\mathbf{A}}_\mathbf{2} = \mathbf{A_1 A_2} + \epsilon \left(\mathbf{A_1 A'_2} + \mathbf{A'_1 A_2}\right) \tag{2.43a}$$

$$\hat{\mathbf{A}}_\mathbf{1}\hat{\mathbf{v}}_\mathbf{1} = \mathbf{A_1}\vec{\mathbf{v}}_\mathbf{1} + \epsilon \left(\mathbf{A'_1}\vec{\mathbf{v}}_\mathbf{1} + \mathbf{A_1}\vec{\mathbf{v}}'_\mathbf{1}\right) \tag{2.43b}$$

where multiplication of dual matrices is associative, but not necessarily commutative. For nonsingular dual matrices the inverse is defined as $\hat{\mathbf{A}}^{-1} = \mathbf{A}^{-1} - \epsilon\left(\mathbf{A}^{-1}\mathbf{A}'\mathbf{A}^{-1}\right)$ which satisfies $\hat{\mathbf{A}}^{-1}\hat{\mathbf{A}} = \hat{\mathbf{A}}\hat{\mathbf{A}}^{-1} = \mathbf{I}$ where $\mathbf{I}$ is a real identity matrix (J. Wang et al., 2012).

2.2.2 Dual Quaternions

In a similar way dual numbers can be used to extend $\mathbb{R}^n$ and $\mathbb{C}^n$ thus obtaining the sets of dual vectors $\mathbb{DR}^n$ and $\mathbb{DC}^n$, the quaternions $\mathbb{H}$ can also be extended to the dual quaternions $\mathbb{DH}$. Namely, dual quaternions are defined as

$$\hat{\mathbf{q}} = \mathbf{q} + \epsilon\mathbf{q}' \tag{2.44a}$$

or

$$\hat{\mathbf{q}} = \left[\hat{q}_0, \hat{\vec{q}}\right] \tag{2.44b}$$

where $\mathbf{q} \in \mathbb{H}$ and $\mathbf{q}' \in \mathbb{H}$ are quaternions (Wu et al., 2005), $\hat{q}_0$ is a dual number and $\hat{\vec{q}}$ is a dual vector representing the scalar and vector components of the dual quaternion respectively. The following is some basic operations of dual quaternions:

$$\hat{\mathbf{q}}_1 + \hat{\mathbf{q}}_2 = \left[\hat{q}_{1_0} + \hat{q}_{2_0}, \hat{\vec{q}}_1 + \hat{\vec{q}}_1\right] \tag{2.45a}$$

$$\hat{\mathbf{q}}_1\hat{\mathbf{q}}_2 = \mathbf{q}_1\mathbf{q}_2 + \epsilon(\mathbf{q}_2\mathbf{q}_1' + \mathbf{q}_1\mathbf{q}_2') \tag{2.45b}$$

$$\hat{\mathbf{q}}^* = \left[\hat{q}_0, -\hat{\vec{q}}\right] \tag{2.45c}$$

$$\|\hat{\mathbf{q}}\|^2 = \hat{\mathbf{q}}\hat{\mathbf{q}}^* \tag{2.45d}$$

where $\hat{\mathbf{q}}^* = \mathbf{q}^* + \epsilon \mathbf{q}_{\prime *}$ is the conjugate of $\hat{\mathbf{q}}$. Of note, the magnitude squared of $\hat{\mathbf{q}}$ from (2.45d) yields a dual number.

Dual quaternions can be used to represent screw motion, using quaternions rather than angles to represent rotation. A 6-DOF transformation consisting of a rotation $\mathbf{q} \in \mathbb{H}_u$ followed by a translation $\vec{p} \in \mathbb{R}^3$ can be represented by a dual quaternion by setting

$$\mathbf{q}' = \frac{1}{2}\mathbf{q}[0, \vec{p}]. \tag{2.46}$$

A dual quaternion defined in this matter has the characteristic $\mathbf{q} \cdot \mathbf{q}' = 0$ in addition to the unit quaternion criterion $\|\mathbf{q}\| = 1$ and is considered normalized and referred to as $\mathbb{DH}_u$. $\mathbb{DH}_u$ forms a Lie group, consisting of a manifold with 3 dual dimensions (X. Wang, Han, Yu, & Zheng, 2012). The identity element of $\mathbb{DH}_u$ is $\hat{\mathbf{I}} \in \mathbb{DH}_u$ and is defined as $\hat{\mathbf{I}} = (1, 0, 0, 0) + \epsilon(0, 0, 0, 0)$. Unit dual quaternions can be used to define the screw motion shown in Fig. 2.9 by the following

$$\hat{\mathbf{q}} = \left[\cos\frac{\hat{\theta}}{2}, \sin\frac{\hat{\theta}}{2}\hat{\vec{\xi}}\right] = \mathbf{q} + \epsilon\frac{1}{2}\mathbf{q}\mathbf{p}_{ab} \tag{2.47}$$

where $\hat{\theta} = \theta + \epsilon d$ is the dual angle about the screw axis $\hat{\vec{\xi}} = \vec{\omega} + \epsilon\vec{v}$ (X. Wang et al., 2012). Elements of $\mathbb{DH}_u$ can be expressed in exponential form by its Lie algebra $\hat{\mathbf{h}}$, defined as (X. Wang et al., 2012)

$$\hat{\mathbf{h}} = \left[\hat{0}, \frac{1}{2}\hat{\theta}\hat{\vec{\xi}}\right] \tag{2.48a}$$

$$\hat{\mathbf{q}} = e^{\frac{\theta}{2}\hat{\mathbf{h}}} \tag{2.48b}$$

The screw motion in Fig. 2.9 can be represented by this Lie algebra as (X. Wang et al., 2012)

$$\begin{aligned}\hat{\mathbf{h}} &= \left[\hat{0}, \hat{\vec{h}}\right] \\ \hat{\vec{h}} &= \vec{\omega} + \epsilon\left(\vec{q} \times \vec{\omega} + \frac{\theta}{d}\vec{\omega}\right).\end{aligned} \tag{2.49}$$

The logarithm of a dual quaternion can then be defined as

$$\log \hat{\mathbf{q}} = \frac{\theta}{2}\hat{\mathbf{J}} = \left[\hat{0}, \frac{1}{2}\hat{\theta}\hat{\vec{\xi}}\right]. \tag{2.50}$$

Since this is not a intuitive result for representing transformations in $SE(3)$, the logarithm can be approximated by a dual vector, exploiting the geometry in Fig. 2.9 as

$$\log \hat{\mathbf{q}} = \frac{1}{2}(\theta\vec{\omega} + \epsilon\vec{p}_{ab}) \tag{2.51}$$

where $\theta \in \mathbb{R}$ and $\vec{\omega} \in \mathbb{R}^3$ are respectively the rotational angle and eigenaxis of rotation, and $\vec{p}_{ab} \in \mathbb{R}^3$ is the translation vector (X. Wang et al., 2012). By definition, the $\log\left(\pm\hat{\mathbf{I}}\right)$ is the dual null vector $\hat{\vec{0}} = [0,0,0] + \epsilon\,[0,0,0]$ (D. Han, Wei, & Li, 2008).

The error difference between two dual quaternions $\hat{q}_1, \hat{q}_2 \in \mathbb{DH}$ is defined as

$$\hat{\mathbf{e}}_{12} = \hat{\mathbf{q}}_1^* \circ \hat{\mathbf{q}}_2. \tag{2.52}$$

A norm for dual quaternions, $\mathfrak{N}(\hat{\mathbf{q}})$, can be defined using the inner product of the logarithm in (2.51) as follows

$$\mathfrak{N}(\hat{\mathbf{q}}) = 2\|\log\hat{\mathbf{q}}\| = 2\sqrt{\langle\log\hat{\mathbf{q}}, \log\hat{\mathbf{q}}\rangle} \tag{2.53}$$

where

$$\langle\log\hat{\mathbf{q}}, \log\hat{\mathbf{q}}\rangle = (\log\hat{\mathbf{q}})^T \cdot \log\hat{\mathbf{q}} = \frac{|\theta|^2 + \epsilon(2\theta p_{ab}\cos\alpha)}{4}$$

$$\Rightarrow \mathfrak{N}(\hat{q}) = |\theta| + \epsilon(\theta p_{ab}\cos\alpha)$$

where α is the angle between $\vec{\omega}$ and $\vec{p}_{ab}$ and p_{ab} is the magnitude of $\vec{p}_{ab}$. $\mathfrak{N}(\hat{\mathbf{q}})$ is a dual positive number (i.e. both the real and dual elements are positive) and $\mathfrak{N}(\pm\hat{\mathbf{I}}) = \hat{0}$. The distance between two dual quaternions $\mathfrak{D}(\hat{\mathbf{q}}_1, \hat{\mathbf{q}}_2)$ is defined using (2.52) and (**??**) as $\mathfrak{D}(\hat{\mathbf{q}}_1, \hat{\mathbf{q}}_2) = \mathfrak{N}(\hat{\mathbf{e}}_{12})$ (D. Han, Fang, & Wei, 2008).

2.3 Dual Quaternion Rigid Body Motion Model

For a single rigid body the kinematic equation describing simultaneously rotation and translation is

$$\dot{\hat{\mathbf{q}}} = \frac{1}{2}\hat{\mathbf{q}}\hat{\boldsymbol{\omega}} \tag{2.54a}$$

$$\hat{\boldsymbol{\omega}} = \left[\hat{0}, \hat{\vec{\omega}}\right] \tag{2.54b}$$

where the dual vector $\hat{\vec{\omega}} \in \mathbb{DR}^3$, called a twist, is defined as

$$\hat{\vec{\omega}} = \vec{\omega} + \epsilon\vec{v} = \omega + \epsilon(\dot{\vec{p}} + \vec{\omega} \times \vec{p}). \tag{2.55}$$

where $\vec{\omega} \in \mathbb{R}^3$ is the angular velocity, $\vec{p}$ refers to position, $\dot{\vec{p}}$ is the velocity with respect to the moving frame and the vector $\vec{v} \in \mathbb{R}^3$ refers to the inertial velocity, defined in the moving frame (J. Wang et al., 2012).

Rigid body dynamics are traditionally defined by:

$$\dot{\vec{v}} = \frac{\vec{f}}{m} \tag{2.56a}$$

$$\dot{\vec{\omega}} = -\mathbf{J}^{-1}\left(\vec{\omega} \times \mathbf{J}\vec{\omega}\right) + \mathbf{J}^{-1}\vec{\tau} \tag{2.56b}$$

where $\vec{v} \in \mathbb{R}^3$ is the translational velocity, $m \in \mathbb{R}$ is the translational inertia or mass, $\vec{f} \in \mathbb{R}^3$ is the translational force applied, $\vec{\omega} \in \mathbb{R}^3$ is the rotational velocity, $\mathbf{J} \in \mathbb{R}^{3\times3}$ is the rotational inertia matrix, and $\vec{\tau} \in \mathbb{R}^3$ is the torque applied.

Dual vectors can be used to describe the same rigid body dynamics in a single equation. To do so, a dual inertia matrix is defined as

$$\begin{aligned}\hat{\mathbf{M}} &= m\frac{d}{d\epsilon}\mathbf{I} + \epsilon\mathbf{J} \qquad (2.57)\\ &= \begin{bmatrix} m\frac{d}{d\epsilon} + \epsilon J_{xx} & \epsilon J_{xy} & \epsilon J_{xz} \\ \epsilon J_{xy} & m\frac{d}{d\epsilon} + \epsilon J_{yy} & \epsilon J_{yz} \\ \epsilon J_{xz} & \epsilon J_{yz} & m\frac{d}{d\epsilon} + \epsilon J_{zz} \end{bmatrix}\end{aligned}$$

where m is mass, $\mathbf{J} \in \mathbb{R}^{3\times3}$ is the inertia matrix, and $\mathbf{I} \in \mathbb{R}^{3\times3}$ the real identity matrix (Brodsky & Shoham, 1999). The operator $\frac{d}{d\epsilon}$ is complimentary to the element ϵ. The operations ϵ and $\frac{d}{d\epsilon}$ are defined as follows

$$\begin{aligned} \epsilon\hat{\mathbf{v}} &= \epsilon(\mathbf{v} + \epsilon\mathbf{v}') = \epsilon\mathbf{v} \\ \frac{d}{d\epsilon}\hat{\mathbf{v}} &= \frac{d}{d\epsilon}(\mathbf{v} + \epsilon\mathbf{v}') = \mathbf{v}'. \end{aligned} \tag{2.58}$$

The inverse of the dual inertia matrix is defined as $\hat{\mathbf{M}}^{\mathbf{-1}} = \mathbf{J}^{\mathbf{-1}}\frac{d}{d\epsilon} + \epsilon\frac{1}{m}\mathbf{I}$ (J. Wang et al., 2012).

The dynamics of a rigid body is then defined as

$$\dot{\hat{\vec{\omega}}} = -\hat{\mathbf{M}}^{\mathbf{-1}}(\hat{\vec{\omega}} \times \hat{\mathbf{M}}\hat{\vec{\omega}}) + \hat{\mathbf{M}}^{\mathbf{-1}}\hat{\mathbf{f}} \tag{2.59}$$

where $\hat{\mathbf{f}} = \mathbf{f} + \epsilon\ \tau \in \mathbb{DR}^3$ is a dual vector called the force motor with $\mathbf{f} \in \mathbb{R}^3$ and $\tau \in \mathbb{R}^3$ being the force and torque vectors in the body frame (J. Wang et al., 2012).

Chapter 3

Sliding Mode Control

The sliding mode control approach has long been recognized as a particularly suitable method for handling nonlinear systems with uncertain dynamics and disturbances. It is is one of the most powerful of contemporary control methods. Probably the first work was done by Irmgard Flügge-Lotz (Flügge-Lotz, 1953). Further development on sliding mode control was done by S.V. Emelyanov and V. I. Utkin (Utkin, 1978).

The main idea of the method is to switch the control in such a way so that the system from any possible state was forced to reach a certain manifold in the state space, i.e. to keep some relation between the systems' internal variables. This relation (manifold) is chosen in such a way that the system fulfills a desired task under that constraint.

The major advantage of Sliding Mode controllers is an inherent low sensitivity to parameter variations and disturbances since after the reaching phase the state is kept very robustly on the manifold in spite of the parameter variations and external disturbances. This chapter provides an introduction into the sliding mode control theory and applications.

3.1 Ordinary Differential Equations with Sliding Modes

This section discusses the main results of the theory of ordinary differential equations with a discontinuous right hand side. The solution of this equations does not exist in classical sense so a generalization of the concept of solution is needed. First recall classical theorems for existence and uniqueness of ordinary differential equation solutions. Consider the initial value problem for ordinary differential equation:

$$\dot{x} = f(x,t) \tag{3.1}$$

where $x(0) = x_0$. Classical conditions for existence of solutions to this problem are the Carathéodory conditions (Coddington & Levinson, 1955):

1. $f(x,t)$ is continuous with respect to x for *almost* all t.
2. $f(x,t)$ is measurable for all $(t,x) \in \mathcal{D}$.
3. $f(x,t) \leq g(t)$ (bounded).

Theorem 1. *If Carathéodory conditions are met, then the solution of the initial value problem (IVP) exists such that*

$$t \in [-\delta,\delta], \exists\ U \ni x_0, \exists\ x(t) \tag{3.2}$$

Carathéodory conditions guarantee existence but not uniqueness of the solution. Consider the following example.

Example:

$$\dot{x} = f(x,t),\ x \in \mathbb{R}^n \tag{3.3a}$$

$$\dot{x} = \sqrt{|x|} \tag{3.3b}$$

$$x(0) = 0 \tag{3.3c}$$

There are two solutions, $x(t) = 0$ and $x(t) = \frac{t^2}{4}$.

Uniqueness requires stronger conditions for $f(x,t)$. One of the most known is the Lipshitz condition (Khalil, 2002):

$$|f(x,t) - f(t,y)| \leq l(t)|x - y| \tag{3.4}$$

Theorem 2. *If $f(x,t)$ is Lipschitz then the solution of the initial value problem is unique on some interval $[-\delta, \delta] \ni t$.*

Consider the simple first order equation:

$$\dot{x} = -M \cdot \operatorname{sgn}(x) \tag{3.5}$$

where $M > 0$. For any $x(0) \neq 0$ the solution reaches the origin in finite time t_1. After that it is "natural" to assume that $x(t) \equiv 0$ for $t > t_1$. In this case the solution exists but if $x(t_o) = 0$ it is unique only for the "forward" interval $t \in [t_o, \infty)$, but not unique for $t \in (-\infty, t_o)$. For example $x_1(t) = M(t_0 - t) \neq x_2(t) = M(t - t_0)$ for $t < t_0$ but both are solutions of (3.5) and $x_1(t) \equiv x_2(t) \equiv 0$ when $t \geq t_0$.

3.1.1 Differential Inclusions

More general case when the right hand side of the ordinary differential equation is discontinuous was rigorously studied by Filippov (Filippov, 1988). Consider the general system $\dot{x} = f(t,x)$ where $f(t,x)$ is discontinuous at the point (t,x^*) and $x \in \mathbb{R}$. By examining the potentially resulting vectors in the vicinity ϵ away from the point (t,x^*) as ϵ approaches zero, the remaining vectors must be a combination of the vectors in set $F(t,x)$, where $\dot{x} \in F(t,x)$.

The function $x(t)$ is the solution to the discontinuous differential equation $\dot{x} = f(t,x)$ if and only if

1. The function $x(t)$ is differentiable almost everywhere (discontinuities at a point are allowed), and
2. $\dot{x}(t)$ remains an element of the set $F(t,x)$ for all t.

This definition is the most general case for the solution of an differential equation with a discontinuous right–hand side. In itself, it does not provide an exact solution, but a set of vectors.

Mathematically, the elements of this set of vectors F must be within the convex hull of the resulting vectors. Let M be the set composed of the discontinuous points in (t,x). The convex hull of a set M is defined as:

$$co^1\ M = \{\alpha_1 x_1 + \alpha_2 x_2 + \ldots + \alpha_k x_k\} \tag{3.6}$$

where $x_1, x_2, \ldots x_n, \in R$, $\alpha_1, \alpha_2, \ldots, \alpha_k \geq 0$, and $\sum_{i=1}^{k} \alpha_i = 1$. If such elements in the convex hull exist where all of the α_i are between zero and one, then by the

[1] co denotes the convex hull of the corresponding set $\{\}$.

Fillipov Definition this set of α_i's will form a set of valid $\dot{x}$ vectors which will solve the differential equation $\dot{x} = f(t,x)$.

3.1.2 Sliding Motion Description

Classical Filippov Definition

For a smooth switching manifold $\sigma = 0$, the Fillipov Definition selects a subset of vectors from the convex closure which will be tangent to the switching manifold. Given the convex hull exists for which includes vectors from the trajectory tangent to σ, the differential equation can be replaced by the differential inclusion $\dot{x} \in F(t,x)$. The classical Fillipov definition for right–hand side discontinuous differential equation $\dot{x} = f(t,x)$, for $(t,x) \in M$ is stated as (Filippov, 1988):

$$F(t,x^*) = co\{lim_{\epsilon\to 0}\{f(t,x)|x \in V_\epsilon(x^*) \text{ excluding } M\}\} \tag{3.7}$$

where x^* is a point of discontinuity on the right–hand side of the differential equation and $V_\epsilon(x^*)$ is the vicinity ϵ away from x^* (i.e., $V_\epsilon(x^*) = \{x| \, \|x - x^*\| < \epsilon\}$) and $F(t,x)$ is a set of vector fields in $\mathbb{R}^n$.

The resulting set $F(t,x)$ lies within the differential inclusion and within the convex hull. Moreover, these vectors $F(t,x)$ will be tangential to the smooth switching manifold $\sigma = 0$. $F(t,x) \perp \nabla\sigma$ solution yields a smaller set of possible vectors but it is still not unique, but it is a necessary condition to remain on the manifold σ. Let us consider what the Filippov definition gives in application to (3.5): **Example 1:**

$$\dot{x} = -\operatorname{sgn} x$$

Notice this is a first order system with the set of discontinuities M only composed of the point $x = 0$, so $\sigma = x = 0$. Observing the behavior near the origin, if x is greater than zero, the resulting vector $\dot{x}$ is equal to negative one and vice versa. Replacing the differential equation with a differential inclusion:

$$\begin{gathered} \dot{x} \in F(t, x) \\ \text{where :} \\ F(t, x) = co\{-1, +1\} \\ = \alpha_1(-1) + \alpha_2(+1) \end{gathered}$$

where $\alpha_1, \alpha_2 \geq 0$ and $\alpha_1 + \alpha_2 = 1$. By substituting for α_2 into the convex hull:

$$-\alpha_1 + 1 + \alpha_1 = 1 - 2\alpha_1$$

where $\alpha_1 \geq 0$. If the trajectory were directed to remain on the manifold once it arrives, the vector $\dot{x}$ must equal zero. Given the differential equation was replaced by a differential inclusion, and solving for $\dot{x} = 0$ results in $\alpha_1 = \frac{1}{2}$ and $\alpha_2 =1 - \alpha_1 =\frac{1}{2}$.

Example 2: A mass–spring system with mass unit mass and spring constant is described by the differential equation.

$$\ddot{x} + x = u$$

can be rewritten by choosing $x_1 = x$ and $x_2 = \dot{x}$ as:

$$\dot{x_1} = x_2$$

$$\dot{x_2} = -x_1 + u$$

Choosing the discontinuous control signal of the form $u = -M\,\mathrm{sgn}(x_2 + x_1)$, the equations at the points of discontinuity become:

$$x_2^* + x_1^* = 0$$

$$F(t, x^*) = co\left\{\begin{bmatrix} x_2^* \\ -x_1^* - M \end{bmatrix}, \begin{bmatrix} x_2^* \\ -x_1^* + M \end{bmatrix}\right\}$$

$$\alpha \begin{bmatrix} x_2^* \\ -x_1^* - M \end{bmatrix} + (1-\alpha) \begin{bmatrix} x_2^* \\ -x_1^* + M \end{bmatrix} = \begin{bmatrix} x_2^* \\ -x_1^* + (1-2\alpha)M \end{bmatrix}$$

For the vector to be tangent to the manifold $\sigma = x_1 + x_2 = 0$, it must be orthogonal to the vector normal to $\sigma = 0$, which means orthogonal to $\nabla\sigma = [1\ 1]^T$. Therefore their inner product must be zero resulting in:

$$x_2^* + [-x_1^* + (1 - 2\alpha]M] = 0$$

$$\alpha = \frac{x_2^* - x_1^* + M}{2M}$$

Substituting α into $\dot{x}$ yields

$$\begin{bmatrix} x_2^* \\ -x_1^* + M - x_2^* + x_1^* - M \end{bmatrix} = \begin{bmatrix} x_2^* \\ -x_2^* \end{bmatrix}$$

where the value $\alpha(x_1, x_2)$ is constrained between zero and one.

The affine system is given as follow:

$$\dot{x} = f(t, x) \tag{3.8}$$

Outside the switching manifold, the trajectories are described by

$$f = \begin{cases} f^+, & \sigma > 0 \\ f^-, & \sigma < 0 \end{cases}, \tag{3.9}$$

while on the manifold $\sigma = x_1 + x_2 = 0$ and since $\nabla\sigma = [1,\ 1]^T$ the system equations are as follows:

$$\dot{x}_1 = x_2 \tag{3.10a}$$

$$\dot{x}_2 = -x_2 \tag{3.10b}$$

or $F = [x_2,\ -x_2]^T$. The normal vector of the manifold is denoted as:

$$\vec{n} = \left[\frac{\partial\sigma}{\partial x}\right]^T \frac{\partial\sigma}{\partial x} \cdot \left(\alpha f^+ + (1-\alpha) f^-\right) = 0 \tag{3.11a}$$

Since normal vector perpendiculars to the tangential vector of the manifold, the value of α can be calculated by setting the product of two vectors equal to zero as in the above equation. Using the manifold equation we have $x_2 = -x_1$ so the system is given by

$$\dot{x}_1 = -x_1 \tag{3.12a}$$

$$\dot{x}_2 = -x_2. \tag{3.12b}$$

Thus, $x_1 \to 0$ and $x_2 \to 0$ as $t \to \infty$.

Generalized Filippov Definition

For a system is given as:

$$\dot{x} = f(t, x, u) \tag{3.13}$$

where $u \in U(t, x)$. $\dot{x}$ is in the convex hull of the $f(t, x, U(t, x))$ function which satisfies the Filippov definition:

$$\dot{x}{\in}F(t, x) = \lim \epsilon \to 0co\left\{f(t, x, U(t, x))|\|x - x^*\| < \varepsilon\right\} \tag{3.14}$$

where $inU(t, x)$.

Equivalent Control Method

Consider the system:

$$\dot{x} = f(t, x, u) \quad x \in \mathbb{R}^n, u \in \mathbb{R} \tag{3.15}$$

The control variables are described in the smooth switching manifold $\sigma(t,x) = 0$:

$$u = \begin{cases} u^+(t,x), & \sigma > 0 \\ u^-(t,x), & \sigma < 0 \end{cases} \tag{3.16}$$

The vectors f, defined as

$$f^+(t,x) = f(t,x,u^+(t,x)) \tag{3.17a}$$

$$f^-(t,x) = f(t,x,u^-(t,x)) \tag{3.17b}$$

are assumed to satisfy Lipschitz condition. Since in sliding mode $\sigma(t,x) = 0$ then at the "average" $\dot{\sigma} = 0$, therefore:

$$\frac{\partial \sigma}{\partial t} + \frac{\partial \sigma}{\partial x} f(t,x,u_{eq}) = 0 \tag{3.18}$$

where the solution u_{eq} of (3.18) is called the equivalent control.

For a system described by an equation affine in the control variable u

$$\dot{x} = f(t,x) + B(t,x)u \tag{3.19}$$

the equivalent control method gives the same solution as Filippov definition. In this case the equation describing system behavior on the manifold is:

$$\dot{x} = f(t,x) + B(t,x)u_{eq} \tag{3.20}$$

Solving (3.18) for u_{eq} in case of the system (3.19) we obtain

$$\sigma(t,x) = 0 \tag{3.21a}$$

$$\dot{\sigma} = \frac{\partial\sigma}{\partial t} + \frac{\partial\sigma}{\partial x}\cdot(f + Bu_{eq}) = 0 \tag{3.21b}$$

$$\Rightarrow u_{eq} = -\left(\frac{\partial\sigma}{\partial x}B\right)^{-1}\left(\frac{\partial\sigma}{\partial t} + \frac{\partial\sigma}{\partial x}f\right) \tag{3.21c}$$

This case is illustrated in Fig. 3.1.

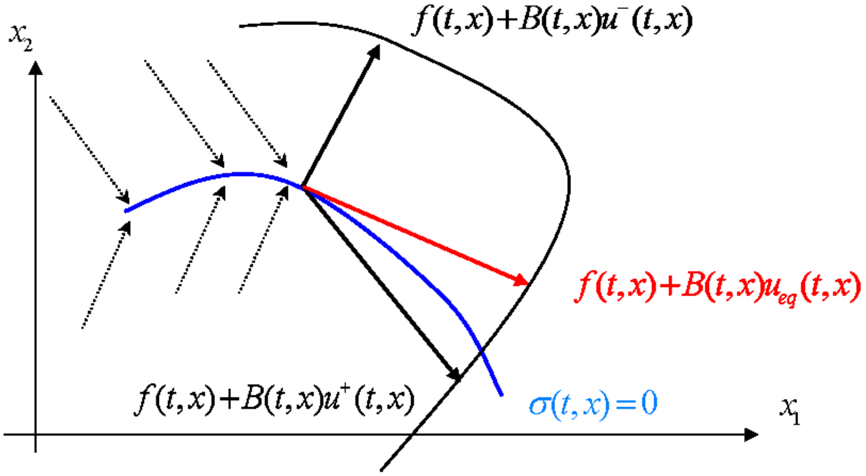

Figure 3.1: Sliding Mode Equivalent Control u_{eq}.

Many of the equations considered in this work are of the following

$$\dot{x} = B(x)u \tag{3.22}$$

where

$$u_{eq} = -\left(\frac{\partial\sigma}{\partial x}B(x)\right)^{-1}\frac{\partial\sigma}{\partial t}. \tag{3.23}$$

Chapter 4

Self-Reconfigurable Control

Sliding mode control for many years has been used in designing an appropriate sliding manifold to stabilize nonlinear systems (S. V. Drakunov & Utkin, 1992; S. V. Drakunov & DeCarlo, 1994). The sliding mode control approach is used for a class of systems

$$\dot{x} = f(t,x) + B(t,x)u \tag{4.1}$$

with uncertain functions f and B. The objective is to make $\sigma(x) = 0$ to be a stable sliding manifold where σ is a given function, under the assumption that f and B belong to wide classes of functions. In majority of cases only one-component manifolds were considered described by the equations

$$\sigma_i(t,x) = 0 \tag{4.2}$$

where $\sigma = col(\sigma_1, ..., \sigma_m)$ and the goal of the design was to make the system reach their intersection

$$\{\sigma(t,x) = 0\} = \bigcap_{i=1}^{m} \{\sigma_i(t,x)\} = 0 \tag{4.3}$$

These techniques were extended to a larger class of systems using the addition of multiple manifolds (S. Drakunov, 1993; S. V. Drakunov, 1994).

This present work, some of which has been previously presented in (Price, Ton, MacKunis, & Drakunov, 2013), concentrates on the design of equilibrium sets in the state space with more complicated structure than just the intersection of several one-component manifolds. The families of sliding (or potentially sliding) surfaces provide new opportunities for designing robust systems with new interesting properties. The control design is based on the creation of multiple sliding surfaces for the system in the extended state space. Each sliding surface in the extended space corresponds to the stability of the origin in σ-space of the given systems, i.e. sliding surface on $\sigma(x) = 0$. In general, these points are different for different f and B.

If areas of attraction for the multiple sliding surfaces covers the entire state space, the proposed control allows one to achieve the goal by a control law which is universal for the class of systems. Being adaptive by its behavior such control automatically changes its structure if the system structure changes during the regulation process. By combining a dual quaternion-based dynamic representation with sliding mode control approach, simultaneous rotation and translation control can be achieved for spatial rigid body systems, where the dynamics contain multiple sources of uncertainty and unmodeled effects. In this work dual quaternion models are used in combination with self-configurable variable structure/sliding-mode control, extending previous work (S. Drakunov, 1993).

4.1 Dual Quaternion Sliding Surface

The dual quaternion kinematics (5.47) and dynamics (4.4b) for a single rigid body was presented in Chapter 2 as

$$\dot{\hat{\mathbf{q}}} = \frac{1}{2}\hat{\mathbf{q}}\hat{\boldsymbol{\omega}} \tag{4.4a}$$

$$\dot{\hat{\vec{\omega}}} = -\hat{\mathbf{M}}^{\mathbf{-1}}(\hat{\vec{\omega}} \times \hat{\mathbf{M}}\hat{\vec{\omega}}) + \hat{\mathbf{M}}^{\mathbf{-1}}\hat{\mathbf{f}} \tag{4.4b}$$

The kinematics and dynamics of a single rigid body can be expanded on and generalized to include multiple bodies:

$$\dot{\hat{\mathbf{q}}}^i = \frac{1}{2}\hat{\mathbf{q}}^i\hat{\boldsymbol{\omega}}^i \tag{4.5}$$

$$\dot{\hat{q}}^i = f\left(\hat{\mathbf{q}}^1, \hat{\mathbf{q}}^2, ..., \hat{\mathbf{q}}^n, \hat{\vec{\omega}}^1, \hat{\vec{\omega}}^2, ..., \hat{\vec{\omega}}^n, t\right) \tag{4.6}$$

$$\begin{aligned}\dot{\hat{\vec{\omega}}}^i = {} & -\hat{\mathbf{M}}^{\mathbf{-1}}\hat{\mathbf{g}}_i\left(\hat{\mathbf{q}}^1, \hat{\mathbf{q}}^2, ..., \hat{\mathbf{q}}^n, \hat{\vec{\omega}}^1, \hat{\vec{\omega}}^2, ..., \hat{\vec{\omega}}^n, t\right) \\ & + \hat{\mathbf{M}}^{\mathbf{-1}}\hat{\mathbf{h}}_i(\hat{\mathbf{q}}^1, \hat{\mathbf{q}}^2, ..., \hat{\mathbf{q}}^n, \hat{\vec{\omega}}^1, \hat{\vec{\omega}}^2, ..., \hat{\vec{\omega}}^n, t)\hat{\mathbf{f}}^i\end{aligned} \tag{4.7}$$

where the integer i represents a single rigid body and n represents the total number of rigid bodies. In these equations the functions $\hat{\mathbf{g}}_i$ and $\hat{\mathbf{h}}_i$ represent the internal forces/torques and the direction of the control force $\hat{\mathbf{f}}^i = \hat{\mathbf{u}}_i$, respectively.

Introducing the generalized position dual quaternion vector $\hat{\mathbf{Q}} = [\hat{\mathbf{q}}^1, \hat{\mathbf{q}}^2, \ldots, \hat{\mathbf{q}}^n]^T \in \mathbb{DH}^n$ and generalized dual velocities vector $\hat{\mathbf{\Omega}} = [\hat{\vec{\omega}}^1, \hat{\vec{\omega}}^2, \ldots, \hat{\vec{\omega}}^n]^T \in \mathbb{DR}^n$ the model can be written as

$$\dot{\hat{\mathbf{Q}}} = \frac{1}{2}\hat{\mathbf{Q}}[\hat{\mathbf{0}}, \hat{\mathbf{\Omega}}] \tag{4.8}$$

$$\dot{\hat{\mathbf{\Omega}}} = -\hat{\mathbf{M}}^{\mathbf{-1}}\hat{\mathbf{g}}(\hat{\mathbf{Q}}, \hat{\mathbf{\Omega}}, t) + \hat{\mathbf{M}}^{\mathbf{-1}}\hat{\mathbf{h}}(\hat{\mathbf{Q}}, \hat{\mathbf{\Omega}}, t)\hat{\mathbf{u}}. \tag{4.9}$$

If in the equations (4.5),(4.8) $\hat{\mathbf{\Omega}} = [\hat{\vec{\omega}}^1, \hat{\vec{\omega}}^2, \ldots, \hat{\vec{\omega}}^n]^T$ is considered as control, the logarithmic feedback law can be used to solve the kinematic regulation control problem (D. Han, Fang, & Wei, 2008) $\hat{\vec{\omega}}^i = -2k \log \lambda \hat{q}^i, \; k > 0$ or

$$\hat{\mathbf{\Omega}} = -2k \log \lambda \hat{\mathbf{Q}}, \tag{4.10}$$

where the log is the quaternion log approximation (2.51) and understood componentwise. The parameter λ is used to have the controller take the shorter path for the identical equilibrium positions $\hat{\mathbf{I}}$ and $-\hat{\mathbf{I}}$, where (D. Han, Fang, & Wei, 2008)

$$\lambda = \begin{cases} 1, & \text{if } \hat{\mathbf{q}}(0) \cdot \hat{\mathbf{I}} \geq 0 \\ -1, & \text{otherwise.} \end{cases} \tag{4.11}$$

Using (4.10), the sliding surface is described as

$$\hat{\sigma} = \hat{\mathbf{\Omega}} + 2k \log \lambda \hat{\mathbf{Q}} = \hat{\vec{0}}. \tag{4.12}$$

Another function that will be useful is $\mathbf{sgn}^\rho(\mathbf{x}) \in \mathbb{R}^n$ defined as (Zhang & Duan, 2011):

$$\mathbf{sgn}^\rho(\mathbf{x}) = [|x_1|^\rho \text{sgn}(x_1) ... |x_n|^\rho \text{sgn}(x_n)]^T \tag{4.13}$$

where $\mathbf{x} \in \mathbb{R}^n$ is an arbitrary vector and $\rho \in \mathbb{R}$. For a dual vector $\hat{\mathbf{x}} = \mathbf{x}_1 + \epsilon \mathbf{x}_2 \in \mathbb{DR}^n, \mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^n$ (Zhang & Duan, 2011):

$$\mathbf{sgn}^\rho(\hat{\mathbf{x}}) = \mathbf{sgn}^\rho(\mathbf{x}_1) + \epsilon \mathbf{sgn}^\rho(\mathbf{x}_2). \tag{4.14}$$

4.2 Self-Reconfigurable Control

In this section, a control algorithm is presented to solve stabilization problem to the sliding manifold

$$\mathcal{M} = \{[\hat{\mathbf{Q}}, \hat{\mathbf{\Omega}}] \in \mathbb{DH}^n \times \mathbb{DR}^n | \hat{\sigma}(\hat{\mathbf{Q}}, \hat{\mathbf{\Omega}}) = 0\} \tag{4.15}$$

introduced in the previous section. Note here, that $\hat{\sigma}$ is an n-dimensional dual vector belonging to $\mathbb{DR}^n$ which is a linear vector space[1]. It is assumed that the dynamic model (4.8), (4.9) contains an unknown, state- and time-varying input gain matrix, which causes unmodeled variations that manifest themselves as a priori unknown changes in the commanded control direction. Once the dual-quaternion-based dynamic model is expressed in the general form, a robust sliding mode controller will be presented, which will be proven to mitigate the unknown control direction based on the approach suggested in (S. Drakunov, 1993), (S. V. Drakunov et al., 1995) and achieve finite-time convergence to a sliding surface.

Differentiating (4.12) yields

$$\dot{\hat{\sigma}} = \hat{\mathbf{B}}(\hat{\mathbf{Q}}, \hat{\mathbf{\Omega}}, t)\hat{\mathbf{u}} + \hat{\mathbf{F}}(\hat{\mathbf{Q}}, \hat{\mathbf{\Omega}}, t) \tag{4.16}$$

where $\hat{\mathbf{B}} = \hat{\mathbf{M}}^{-\mathbf{1}}\hat{\mathbf{h}}(\hat{\mathbf{Q}}, \hat{\mathbf{\Omega}}, t) \in \mathbb{DR}^{n\times n}$ is a matrix defining direction of the control action in σ-space. The goal is to develop a control algorithm that does not require knowledge of $\hat{\mathbf{B}}$, but it can be assumed that this dual matrix satisfy natural conditions that follow from mechanical properties of the controlled system. $\hat{\mathbf{B}}$ is such that (i) it is nonsingular almost everywhere, and (ii) the corresponding quadratic form

[1] It is also a Banach space with corresponding norms. So that, for example, if $\hat{\sigma} = \sigma + \epsilon\sigma'$, then this dual vector p-norm is $\|\hat{\sigma}\|_p = \left[\|\sigma\|_p^p + \|\sigma'\|_p^p\right]^{\frac{1}{p}} = [\sum_{k=1}^n(|\sigma_k|^p + |\sigma'{}_k|^p)]^{\frac{1}{p}}$, $(p \geq 1)$.

$\xi^T \hat{\mathbf{B}}(\hat{\mathbf{Q}}, \hat{\mathbf{\Omega}})\xi$ is sign definite and a manifold (if such exists) in $\mathbb{DH}^n \times \mathbb{DR}^n$ space where this quadratic form can change its sign does not coincide with the desired sliding manifold $\mathcal{M}$ at least in some area of the state space $\mathbb{DH}^n \times \mathbb{DR}^n$ where the system trajectories evolve.

The main idea behind this control is in partitioning the $\hat{\sigma}$-subspace onto a grid comprised of concentric manifolds that are spheres defined by $\|\hat{\sigma}\|_p^p = \Delta(t)k$, where $\Delta(t) > 0$ is the variable grid step, k is a nonnegative integer and $\|\cdot\|_p$ is a p-norm. Inside each layer $\mathcal{L}_k$ between these manifolds $\mathcal{L}_k = \{\Delta(t)k \leq \|\hat{\sigma}\|_p^p \leq \Delta(t)(k+1)\}$ the control may be constant, but its sign alternates from one layer to another. We show that this control structure under a nonsingularity condition results in a set of stable equilibrium spheres in $\hat{\sigma}$-subspace. Then we choose the dynamics of $\Delta(t)$ so that eventually all spheres radii converge to zero, thus, stabilizing $\hat{\sigma}$ to the origin of the corresponding dual vector space.

The union of the concentric manifolds forms a switching manifold:

$$\mathcal{G} = \bigcup_{k=0,\pm1,\ldots}^{r} \mathcal{G}_k = \bigcup_{k=0,\pm1,\ldots}^{r} \left\{x : \|\hat{\sigma}(x)\|_p^p = \Delta(t)k\right\}. \tag{4.17}$$

Pick $\hat{\mathbf{u}}$ such that

$$\hat{\mathbf{u}} = \hat{\mathbf{U}}_0 \mathbf{sgn}\left[\sin\left(\pi \frac{\|\hat{\sigma}\|_p^p}{\Delta(t)}\right)\right] \mathbf{sgn}(\hat{\sigma}), \tag{4.18}$$

where $\hat{\mathbf{U}}_0$ is a dual matrix control gain of the form:

$$\hat{\mathbf{U}}_0 = \mathbf{K_f}\frac{d}{d\epsilon} + \epsilon \mathbf{K}_\tau \tag{4.19}$$

where $\mathbf{K_f}$ and $\mathbf{K}_\tau$ may be constant diagonal matrixes or state dependent diagonal matrixes related to the gains for translation and rotation respectively. The operators $\frac{d}{d\epsilon}$ and ϵ are included in the dual matrix control gain so that the real (force) and dual (torque) components of the force motor $\hat{\mathbf{f}}$ are associated with the dual (displacement) and real (rotation) components of the sliding surface $\hat{\sigma}$. Let us note that the term $\mathbf{sgn}\left[\sin\left(\pi\frac{\|\hat{\sigma}\|_p^p}{\Delta(t)}\right)\right]$ is a changing sign scalar and the last term $\mathbf{sgn}(\hat{\sigma})$ in (4.18) is a dual vector part of the control that alternate signs in the quadrants of the corresponding dual vector space $\mathbb{DR}^n$. It is needed to guarantee stability on one of the sliding manifolds $\mathcal{G}_k$.

The function $\Delta(t)$ is the following:

$$\Delta(t) = C - \mu\int_0^t \|\hat{\sigma}(\tau)\|_p^p d\tau, \tag{4.20}$$

where $C > 0$ is chosen from the area of initial conditions and $\mu > 0$ is a control parameter regulating spheres' radii convergence rate.

Fig 5.1 demonstrates the vector field of velocities and one possible scenario of convergence toward the sliding manifold (red line) using control (4.18). In this simulation experiment the matrix B and the initial conditions were chosen randomly.

Also note, that in (4.18) the $\sin(\pi x)$ function is used only for convenience and relate this control algorithm to the one described in (S. V. Drakunov, 1994). In fact, the main property that is required from this function is the alternating sign. It does not even have to be periodic. So the more general form of the control algorithm is

$$\hat{\mathbf{u}} = \hat{\mathbf{U}}_0\psi\left(\frac{\|\hat{\sigma}\|_p^p}{\Delta(t)}\right)\mathbf{sgn}(\hat{\sigma}),, \tag{4.21}$$

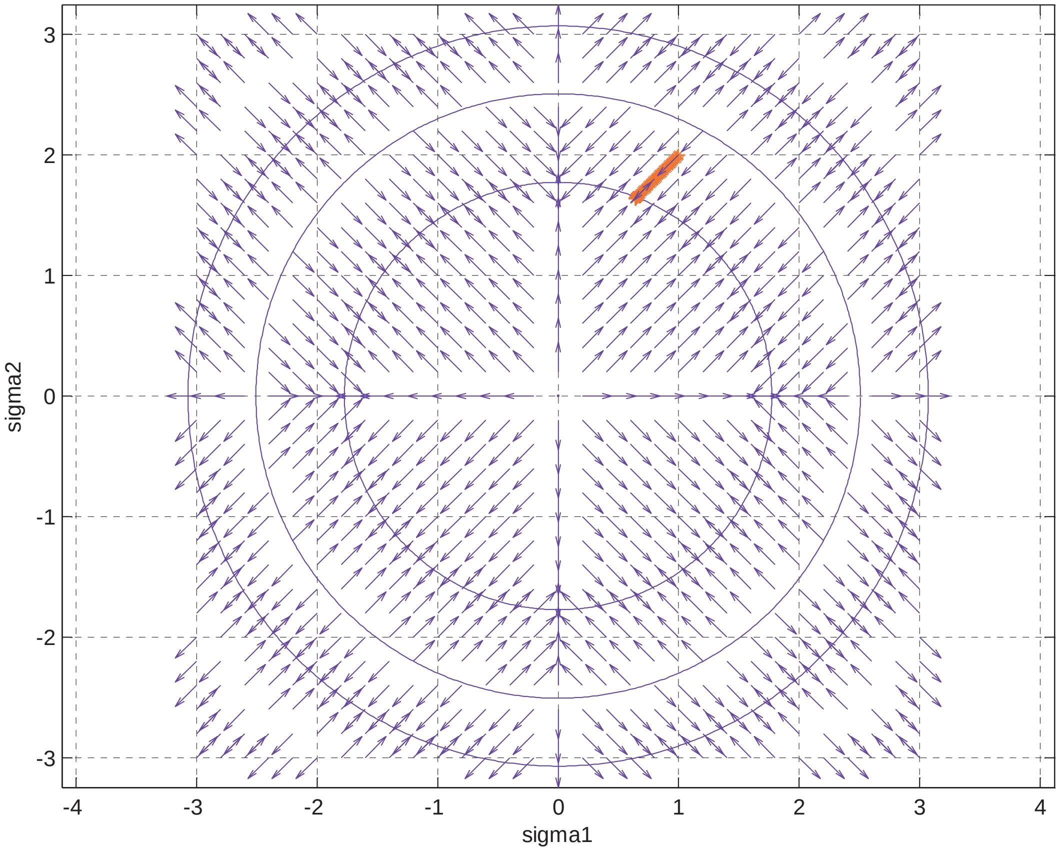

Figure 4.1: Equilibrium manifolds in $\hat{\sigma}$-space ($p = 2$).

where the function $\psi(x)$ is such that, for example, $\psi(x) = \operatorname{sgn} n(x)$ if $|x| \leq 1$, and $\psi(x) = -\operatorname{sgn}(x)$ otherwise.

Proof. Here a sketch of the proof is provided just to demonstrate the technique by considering the case of real $\hat{\mathbf{B}}$. The general situation is treated similarly by considering separately the real and dual part of the control (4.18). The values $p = 1$, and a real scalar $\hat{\mathbf{U}}_0$ are used. The idea behind the convergence proof is the following: consider a Lyapunov function with multiple zeroes:

$$V = \left| \sin\left(\pi \frac{\|\hat{\sigma}\|_1}{\Delta(t)} \right) \right| . \tag{4.22}$$

V is positive everywhere except it is zero at the points where

$$\|\hat{\sigma}\|_1 = \Delta(t)k \tag{4.23}$$

$k = 0, \pm 1, \pm 2, \ldots$.

The derivative of V along the system trajectories is

$$\dot{V} = \mathbf{sgn}\left[\sin\left(\pi \frac{\|\hat{\sigma}\|_1}{\Delta(t)} \right) \right] \cos\left(\pi \frac{\|\hat{\sigma}\|_1}{\Delta(t)} \right) \pi \frac{d}{dt} \left[\frac{\|\hat{\sigma}\|_1}{\Delta(t)} \right] . \tag{4.24}$$

Since the 1-norm can be represented as $\|\sigma\|_1 = (\mathbf{sgn}\sigma)^T \sigma$ the derivative of $\|\sigma\|_1$ using (4.16) can be written as

$$\frac{d\|\hat{\sigma}\|_1}{dt} = (\mathbf{sgn}\sigma)^T \dot{\sigma} = (\mathbf{sgn}\sigma)^T \hat{\mathbf{B}}\hat{\mathbf{u}} + (\mathbf{sgn}\sigma)^T \hat{\mathbf{F}}, \tag{4.25}$$

or using (4.18) written as

$$\begin{aligned} \frac{d\|\hat{\sigma}\|_1}{dt} &= \hat{\mathbf{U}}_0 (\mathbf{sgn}\sigma)^T \hat{\mathbf{B}} (\mathbf{sgn}\sigma) \mathbf{sgn}\left[\sin\left(\pi \frac{\|\hat{\sigma}\|_1}{\Delta(t)} \right) \right] \\ &+ (\mathbf{sgn}\sigma)^T \hat{\mathbf{F}}. \end{aligned} \tag{4.26}$$

Combining this with (4.24) the Lyapunov function derivative can be written as

$$\dot{V} = \hat{\mathbf{U}}_0 \frac{\pi}{\Delta(t)} (\mathbf{sgn}\sigma)^T \hat{\mathbf{B}} (\mathbf{sgn}\sigma) \cos\left(\pi \frac{\|\hat{\sigma}\|_1}{\Delta(t)} \right) + \mathbf{G}, \tag{4.27}$$

where all terms that don't depend on the control in the variable $\mathbf{G}$ are combined. By this assumption the quadratic form with the matrix $\hat{\mathbf{B}}$ in this expression is sign definite. On the other hand at the points where $V = 0$ ($\|\hat{\sigma}\|_1 = \Delta(t)k$) the cos is $+1$ or -1, so at every other point it is guaranteed that $\dot{V} < 0$ if of course, $\hat{\mathbf{U}}_0$ is big enough. This proves the stability some of the points (4.23). In fact, sliding mode will start at one of these points and $\|\hat{\sigma}\|_1 = \Delta(t)k$ will be true after some moment of time.

Now using the expression for Δ (4.20) and (4.23) yields

$$\|\hat{\sigma}\|_1 = \left[C - \mu \int_0^t \|\hat{\sigma}(\tau)\|_1 d\tau \right] k. \tag{4.28}$$

The latter is stable equation that guarantees $\|\hat{\sigma}\|_1 \to 0$ exponentially as $t \to \infty$. □

4.3 Numerical Examples

4.3.1 Planar Example

In the first numerical example the planar motion of the rigid body is considered:

$$\dot{x} = v_x \tag{4.29a}$$

$$\dot{y} = v_y \tag{4.29b}$$

$$\dot{\theta} = \omega \tag{4.29c}$$

$$\dot{v_x} = f_x \tag{4.29d}$$

$$\dot{v_y} = f_y \tag{4.29e}$$

$$\dot{\omega} = \tau \tag{4.29f}$$

where v_x, v_y are the velocity in the x and y direction, ω is the angular velocity, and $\vec{f} = [f_x, f_y, \tau]^T$ is the generalized force vector that defined in the fixed frame which depends on the control $\vec{u} \in \mathbb{R}^3$. $\vec{f}$ and $\vec{u}$ are related via an unknown 3×3 possibly state dependent matrix $B = B(X)$ $(X = [x, y, v_x, v_y, \theta, \omega]^T)$.

$$\vec{f} = B(X)\vec{u}. \tag{4.30}$$

Let the sliding surface be $\vec{\sigma} = [\sigma_1\ \sigma_2\ \sigma_3]^T$, where

$$\sigma_1 = k_x x + v_x \tag{4.31a}$$

$$\sigma_2 = k_y y + v_y \tag{4.31b}$$

$$\sigma_3 = k_\theta \theta + \omega. \tag{4.31c}$$

The objective is to drive the system (4.29) to the origin with an orientation of $\theta = 0$. Differentiating $\vec{\sigma}$ yields

$$\dot{\vec{\sigma}} = B(X)u + G. \tag{4.32}$$

For the purpose of this simulation the values in Table 4.1 were used.

Table 4.1: Values used for Self-Reconfigurable Control $\mathbb{R}^3$ example with $B = -\mathbf{I}$.

x_i	2
y_i	-2.5
θ_i	$\frac{\pi}{4}$
C	.3
μ	.05
k_x	.12
k_y	.12
k_θ	.1
$\mathbf{U}_0$	$.2\mathbf{I}$

Using the control algorithm in (4.18) the control $\vec{u}$ is defined as:

$$\vec{u} = .2\mathbf{I}\,\mathrm{sgn}\left[\sin\left(\pi\frac{\|\vec{\sigma}\|}{\Delta(t)}\right)\right]\mathrm{sgn}\left(\vec{\sigma}\right) \tag{4.33}$$

The first simulation used the matrix $B(X) = -\mathbf{I}$, effectively placing the control vector opposite of the desired direction. Figs. 4.2 and 4.3 demonstrate convergence to the sliding manifolds and the convergence for the variables x, y, θ. As can be seen in Fig. 4.2, the system moved away from the nearest sliding surface at $k = 1$ and but followed the next surface at $k = 2$.

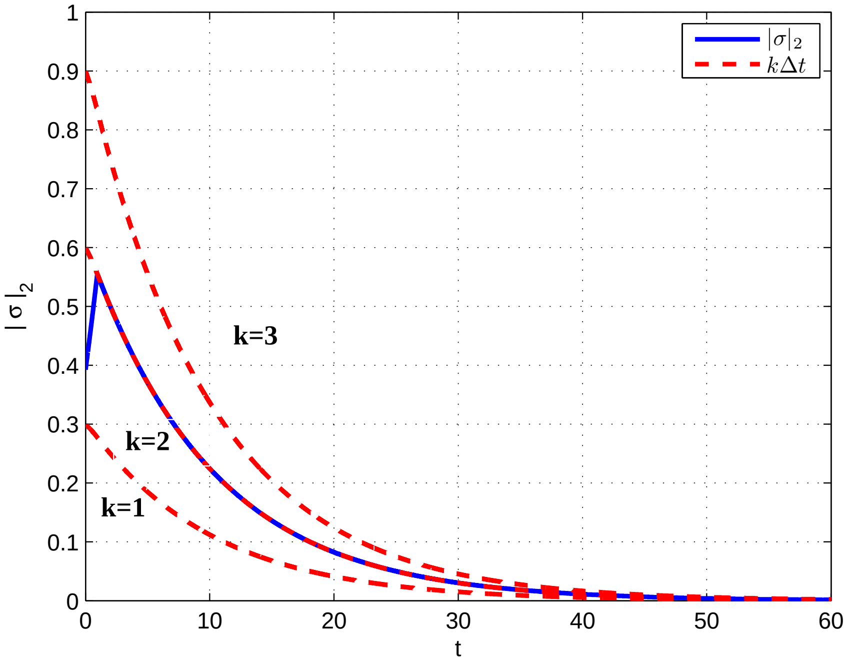

Figure 4.2: Convergence of $|\sigma|_2$ to $k\Delta t$ for $B = -\mathbf{I}$.

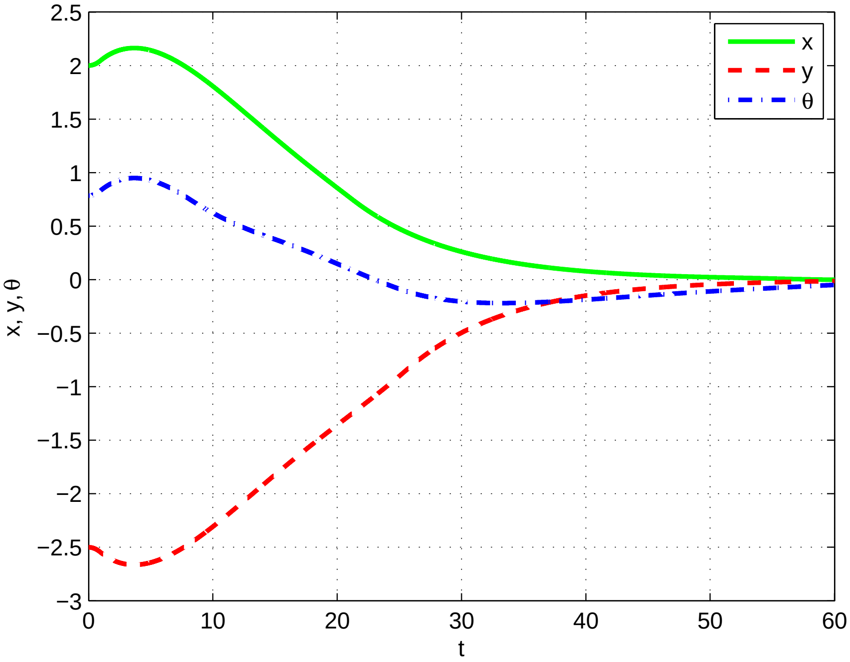

Figure 4.3: Convergence of positions x, y, θ for $B = -\mathbf{I}$.

The next simulation used a time varying, state dependent matrix for $B(X)$ defined as:

$$B(X) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{4.34}$$

This $B(X)$ has the effect of assuming that that translation control is in the fixed frame, while in actuality the control is in the rotating body frame. The values in Table 4.2 were used for this simulation.

Table 4.2: Values used for Self-Reconfigurable Control $\mathbb{R}^3$ example with state dependent $B(x)$.

x_i	.5
y_i	-1
θ_i	π
C	.5
μ	.015
k_x	.12
k_y	.12
k_θ	.6
$\mathbf{U}_0$	$.1\mathbf{I}$

Figs. 4.4 and 4.5 demonstrate convergence to the sliding manifolds and the convergence for the variables x, y, θ.

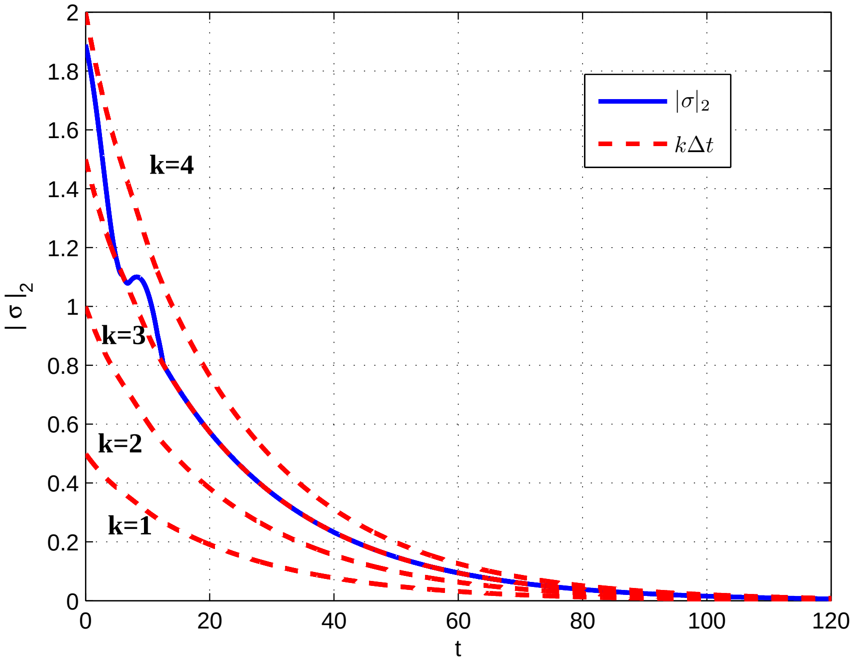

Figure 4.4: Convergence of $|\sigma|_2$ to $k\Delta t$ for state dependent $B(x)$.

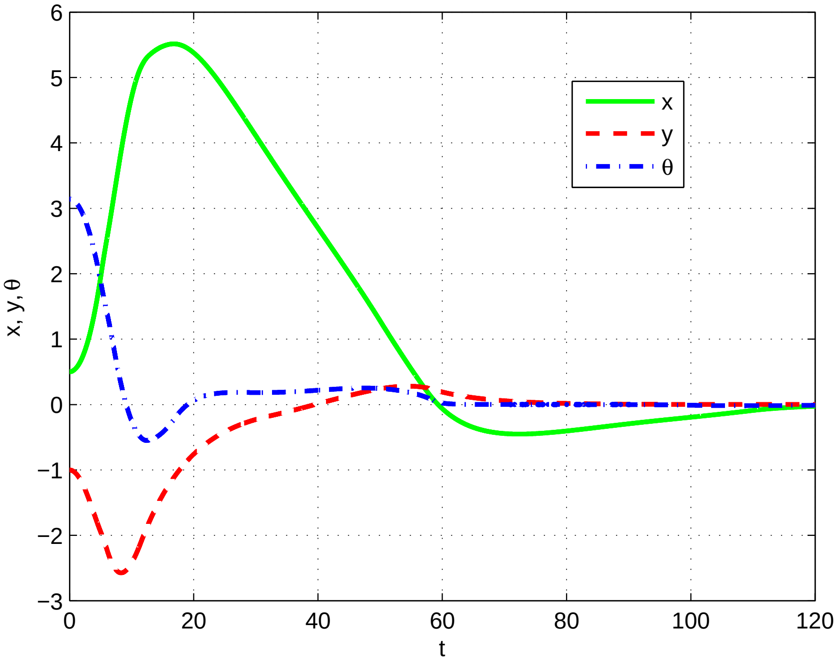

Figure 4.5: Convergence of positions x, y, θ for state dependent $B(x)$.

4.3.2 Dual Quaternion Example

In the second numerical example the 6-DOF motion of a rigid body using dual quaternions is considered:

$$\dot{\hat{\mathbf{q}}} = \frac{1}{2}\hat{\mathbf{q}}\hat{\boldsymbol{\omega}} \tag{4.35a}$$

$$\dot{\hat{\vec{\omega}}} = -\hat{\mathbf{M}}^{\mathbf{-1}}(\hat{\vec{\omega}} \times \hat{\mathbf{M}}\hat{\vec{\omega}}) + \hat{\mathbf{M}}^{\mathbf{-1}}\hat{\vec{f}} \tag{4.35b}$$

where $\hat{\mathbf{q}}$ is the dual quaternion, $\hat{\vec{\omega}}$ is the dual velocity vector and $\hat{\vec{f}} = [f_x, f_y, f_z]^T + \epsilon[\tau_x, \tau_y, \tau_z]^T$ is the generalized force dual vector that depends on the control $\hat{\vec{u}} \in \mathbb{DR}^3$. $\hat{\vec{f}}$ and $\hat{\vec{u}}$ are related via an unknown 3×3 possibly state dependent dual matrix

$\hat{C} = \hat{C}(\hat{\mathbf{q}}, \hat{\vec{\omega}})$ by

$$\hat{\vec{f}} = \hat{C}\left(\hat{\vec{u}}\right). \tag{4.36}$$

Let the sliding surface be $\hat{\vec{\sigma}}$, where

$$\hat{\vec{\sigma}} = (\vec{\omega} + k_\theta \theta \vec{n}) + \epsilon\left(\vec{v} + k_p \vec{p}\right) \tag{4.37}$$

The objective is to drive the system (4.35) to the origin with an orientation of $\theta = 0$ or $q = [1, 0, 0, 0]$.

For the purpose of this simulation the values in Table 4.3 were used.

Table 4.3: Values used for Self-Reconfigurable Control $\mathbb{DH}$ example.

$\mathbf{q}(0)$	$[.1739, .3392, -.8213, .4244]$
$\vec{p}(0)$	$[2.5, 1.5, -1]$
C	$.3$
μ	$.03$
k_θ	$.1$
k_p	$.2$
$\mathbf{K_f}$	$15\mathbf{I}$
$\mathbf{K}_\tau$	$15\mathbf{I}$

Fig. 4.6 shows the convergence of the system to the real and dual components of the sliding manifold. Fig. 4.7 and Fig. 4.8 demonstrate convergence for the rotational and translational position respectively.

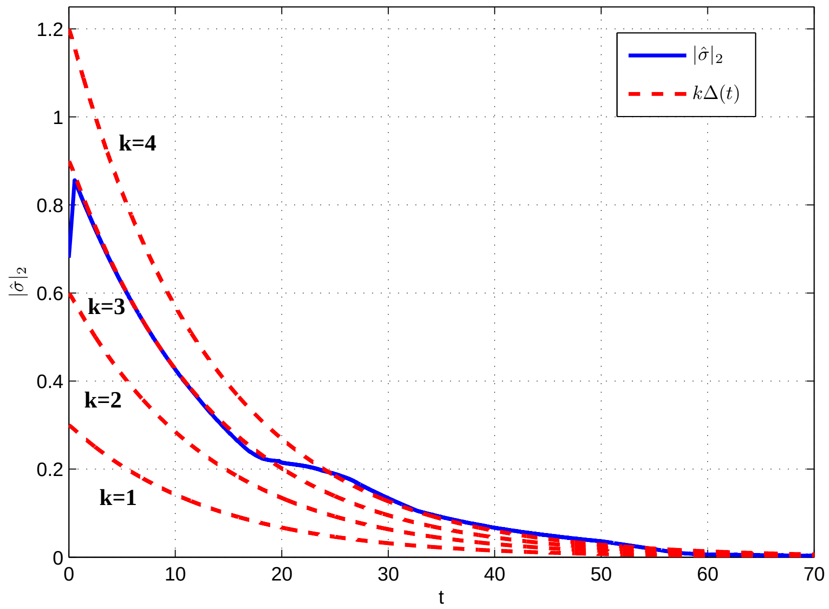

Figure 4.6: Convergence of $|\hat{\vec{\sigma}}|_2$ to $k\Delta t$

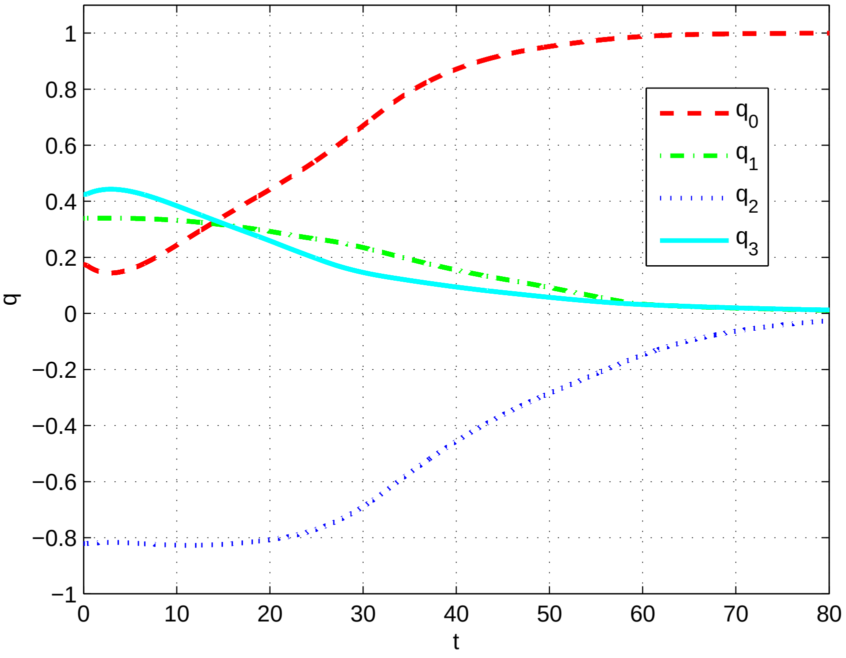

Figure 4.7: Convergence of attitude **q**

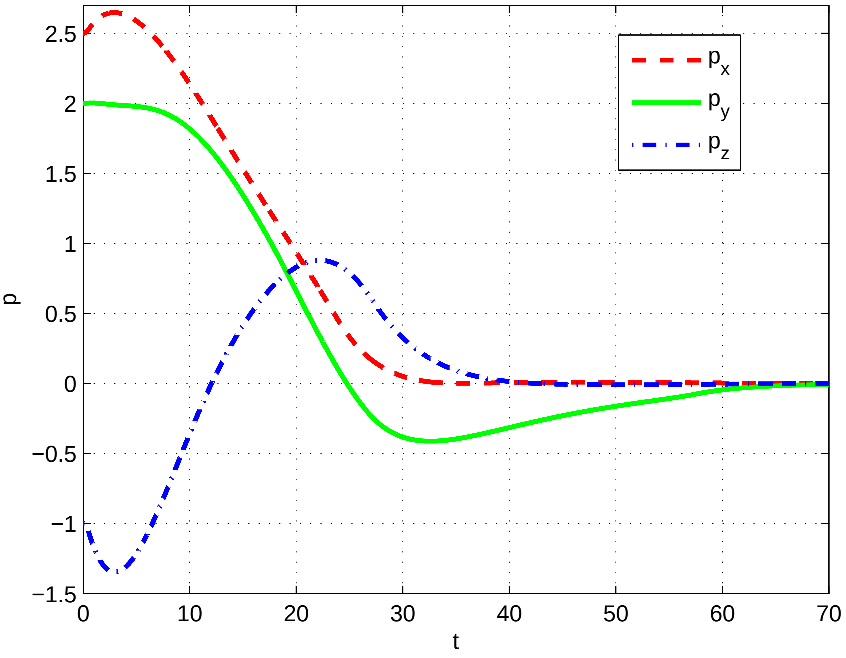

Figure 4.8: Convergence of positions x, y, z

Chapter 5

Nonlinear Driftless Systems

In this chapter[1] a control algorithm is developed for a class of nonlinear driftless systems originally introduced by Brockett (Brockett, 1981):

$$\dot{x} = B(x)u \tag{5.1}$$

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$. The controllability condition for such systems is well established and can be easily tested via the accessibility algebra and the accessibility distribution (Isidori, 1997). An interesting class of controllable systems with the structure of (5.1) is so-called a kinematic nonholonomic system. Those systems are characterized by specific relation between the dimension of the system state x and the control input u. This condition will be relaxed by extending the system space and designing control for the augmented system.

The conditions necessary for augmentation of such systems which do not permit

[1] Some of the material presented in this chapter was also presented in (Price, Seo, Kitchen-Mckinley, & Drakunov, 2014)

a Brockett's canonical form is examined. The augmented system must be controllable and satisfy the hypotheses of the Brocket theorem listed in the next section. Therefore, the system can be controlled by various developed nonholonomic system controllers. The control architecture considered here is from (A. Bloch et al., 1997) and (A. M. Bloch et al., 2000).

5.1 Approach

In (Brockett, 1981), Brockett presented a theorem stating that there exists a coordinate transformation which renders (5.1) to a canonical form with the following assumptions:

H 1. For $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$, it is assumed that $n = m(m+1)/2$.

H 2. $\mathrm{E}^{(1)}$ spans $\mathbb{R}^{m(m+1)/2}$, wherein $\mathrm{E}^{(1)}$ is defined by

$$\mathrm{E}^{(0)}_{\mathrm{x}} = \mathrm{span}\{B(x)\} \tag{5.2}$$

$$\mathrm{E}^{(1)} = \mathrm{span}\{\mathrm{E}^{(0)}_{\mathrm{x}} + [\mathrm{E}^{(0)}_{\mathrm{x}}, \mathrm{E}^{(0)}_{\mathrm{x}}]\} \tag{5.3}$$

with Lie bracket $[\cdot, \cdot]$ for vector fields.

Corresponding Brockett's theorem is

Theorem 3. *Given the system* (5.1) *satisfying both* H*1 and* H*2, there exist coordinates* $(x^1, x^2, \ldots, x^m, y^{1,2}, y^{1,3}, \ldots, y^{m-1,m})$ *in a neighborhood of a given point, say*

$x = 0$, so that the equations are represented by

$$\dot{x}^i = u^i + r^i, \quad i = 1, 2, \ldots, m \tag{5.4a}$$

$$\dot{y}^{ij} = u^i x^j - u^j x^i + r^{ij}, \quad i, j = 1, 2, \ldots, m,\ i < j \tag{5.4b}$$

where the r^i and r^{ij} have vanishing first partials with respect to x and y and in addition r^{ij} has vanishing second partials with respect to x^i and x^j.

The observations on the hypotheses 1 and 2 reveal that the certain class of controllable systems failing to satisfy those hypotheses may have an augmented system representation for Theorem 3 to be applicable. More specifically, if $n < m(m+1)/2$, then, there may exist $n_1 = m(m+1)/2 - n$ differential equations

$$\dot{\boldsymbol{x}}_s = B_s(\boldsymbol{x}_s, \boldsymbol{x})\boldsymbol{u} \quad \boldsymbol{x}_s \in \mathbb{R}^{n_1} \tag{5.5}$$

such that combining (5.1) and (5.5), i.e., $\boldsymbol{x}_a = [\boldsymbol{x}^T, \boldsymbol{x}_s^T]^T$ and $B_a = [B^T, B_s^T]^T$, we obtain the augmented system

$$\dot{\boldsymbol{x}}_a = B_a(\boldsymbol{x}_a)\boldsymbol{u} \quad \boldsymbol{x}_a \in \mathbb{R}^{m(m+1)/2}, \tag{5.6}$$

which satisfy $H2$ and thus the augmented system dynamics have the form of (5.1) with $H1$ and $H2$.

Then to the augmented system we apply control from (A. Bloch et al., 1997) and (A. M. Bloch et al., 2000) considering (5.5) as part of dynamical control law.

5.2 Controller Design

The general system studied is described as follows. Let $\mathfrak{g}$ be a Lie algebra with a direct sum decomposition $\mathfrak{g} = \mathfrak{m} + \mathfrak{h}$ such that $\mathfrak{h}$ is a Lie subalgebra, $[\mathfrak{h}, \mathfrak{m}] \subseteq \mathfrak{m}$, and $[\mathfrak{m}, \mathfrak{m}] \subseteq \mathfrak{h}$. Consider the following system in $\mathfrak{g}$:

$$\dot{x} = u \tag{5.7a}$$

$$\dot{Y} = [u, x] = xu^T - ux^T \tag{5.7b}$$

where $x, u \in \mathfrak{m}$, $Y \in \mathfrak{h}$, $\mathfrak{m} \in \mathbb{R}^n$, $\mathfrak{h} \in \mathfrak{so}(n)$, and $n \geq 2$. The control input u is generated by

$$u(x, Y) = -\alpha x + \beta[Y, x] \tag{5.8}$$

where $\alpha = \alpha(x, Y)$ and $\beta = \beta(x, Y)$ (A. Bloch et al., 1997). With this choice of u, (5.7) becomes:

$$\dot{x} = -\alpha x + \beta[Y, x] \tag{5.9a}$$

$$\dot{Y} = \beta\left[[Y, x], x\right] \tag{5.9b}$$

The following polynomials are used to develop the control strategy:

$$U = \frac{1}{2}\langle x, x\rangle = \frac{1}{2}(x^T x) \tag{5.10a}$$

$$V_1 = \frac{1}{2}\langle Y, Y\rangle = \frac{1}{2}trace(Y^T Y) \tag{5.10b}$$

$$V_2 = \frac{1}{2}\langle [x, Y], [x, Y]\rangle = \frac{1}{2}x^T\left(Y^T Y\right)x \tag{5.10c}$$

From (5.10) the following is computed:

$$\dot{U} = -2\alpha U \tag{5.11a}$$

$$\dot{V}_1 = -4\beta V_2 \tag{5.11b}$$

$$\dot{V}_2 = -2\alpha V_2 - 4\beta U V_2 \tag{5.11c}$$

The control strategy can now be described using the following steps:

(i) With the control $\alpha > 0$, $\beta = 0$ without changing Y, drive x to the eigenvector of $Y^T Y$ corresponding to its maximum nonzero eigenvalue $\lambda_{max} > 0$; if $\lambda_{max} = 0$ implement step (iv).

(ii) Implement the control with $\alpha = 0$, $\beta > 0$. This does not change the magnitude of x ($\|x\| =$ const.), but in the cases of interest will imply that the maximum eigenvalue of $Y^T Y$ strictly decreases. Continue until $V_2 \approx 0$.

(iii) Repeat Steps (i) and (ii) until $Y \approx 0$ or $V_1 \approx 0$.

(iv) Drive x to the origin using the control with $\alpha > 0$, $\beta = 0$.

Step (i) of the control strategy is implemented as follows:

(A) Drive x to the origin via the control $u = -\alpha x$ where $\alpha > 0$. This does not affect Y since $\dot{Y} = 0$.

(B) Drive x from 0 to the eigenvector x^* of $Y^T Y$ corresponding to λ_{max} by the control $u = -\alpha(x - x*)$ where $\alpha > 0$.

Theorem 4. *The algorithm (i)-(iv) represents a global stabilizing control for the system* (5.7)*.*

The stability proof is given in (A. Bloch et al., 1997) and (A. M. Bloch et al., 2000).

The tracking problem can be developed by defining the tracking error using the following:

$$\bar{x} = x - x_d \tag{5.12a}$$

$$\bar{Y} = Y - Y_d - x_d x^t + x x_d^T \tag{5.12b}$$

where x_d and Y_d are the desired states and $\bar{x}$ and $\bar{Y}$ are the state errors. The system in (5.7) can then modified to be

$$\dot{\bar{x}} = \dot{x} - \dot{x}_d \tag{5.13a}$$

$$\bar{u} = u - u_d \tag{5.13b}$$

$$\dot{\bar{Y}} = \bar{x}\bar{u}^T - \bar{u}\bar{x}^T + g, \quad g = 2\bar{x}u_d^T - 2u_d\bar{x}^T \tag{5.13c}$$

where u_d is the control required for the desired trajectory and $\bar{u}$ is the control error (A. Bloch & Drakunov, 1995). A modification to Step 4 of the control strategy is required to allow for a return to Step 1 in the case were $\bar{Y}$ increases beyond an unacceptable amount due to the extra term g found in (5.13c). With this change, the control strategy can then be used to ensure that the state errors are driven to zero and remain there.

In the low dimension case, important for various applications, when $x \in \mathbb{R}^2$ the system (5.7) is a nonholonomic integrator:

$$\dot{x}_1 = u_1 \tag{5.14a}$$

$$\dot{x}_2 = u_2 \tag{5.14b}$$

$$\dot{x}_3 = x_1 u_2 - u_1 x_2. \tag{5.14c}$$

The control law (5.8) in this case, as shown in (A. Bloch & Drakunov, 1996), is

$$u_1 = -\alpha x_1 + \beta\, x_2 \tag{5.15a}$$

$$u_2 = -\alpha x_2 - \beta\, x_1 \tag{5.15b}$$

where α, β can be chosen as

$$\alpha = \alpha_0 \ \operatorname{sgn}(x_1^2 + x_2^2 - |x_3|) \ \ (\alpha_0 > 0) \tag{5.16a}$$

$$\beta = \beta_0 \operatorname{sgn}(x_3) \ \ (\beta_0 > 0) \tag{5.16b}$$

In this case when outside the parabolic area $x_1^2+x_2^2 > |x_3|$, asymptotic convergence of x_3 is guaranteed. If the initial conditions are inside this area, $x_1^2 + x_2^2$ is increasing and reaches parabola in finite time, staying then in sliding mode on the surface of parabola, where

$$\dot{x}_3 = -\beta_0 x_3. \tag{5.17}$$

In fact, this control forms two sliding surfaces in the state space of the closed system: $\{x_3 = 0\}$ and $\{x_1^2 + x_2^2 = |x_3|\}$.

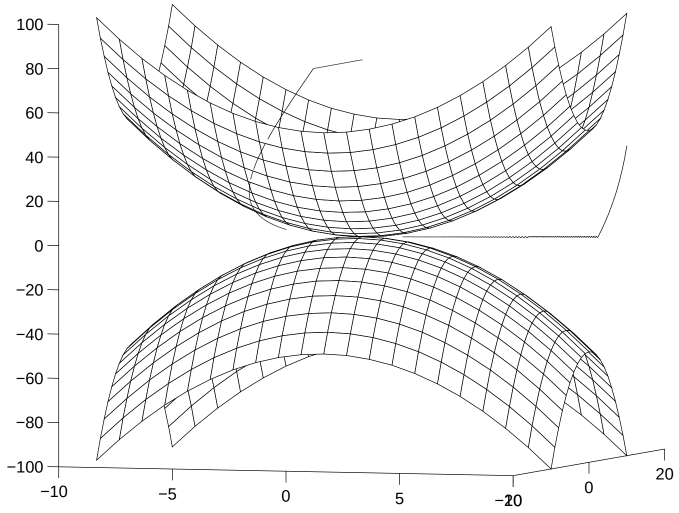

Figure 5.1: Stabilization of the nonholonomic integrator (A. Bloch & Drakunov, 1996).

5.2.1 Application of the Nonholonomic Feedback to Quaternion System

Let us show how to modify the above result in order to design a feedback control for very important case of quaternion system when only two controls are used. Consider the kinematic equation for unit quaternions $\mathbb{H}_u$:

$$\dot{\mathbf{q}} = \frac{1}{2}\mathbf{q}\boldsymbol{\omega} \tag{5.18a}$$

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \begin{bmatrix} 0 \\ \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \tag{5.18b}$$

Now consider the case where only two components of the angular velocity can be controlled, i.e. $\vec{\omega} = [\omega_1\ \omega_2\ 0]^T$, the kinematics can be simplified to (5.19a). Then the equations for q_1 and q_2 in (5.19a) are linearized around the equilibrium point $\mathbf{q} = [1\ 0\ 0\ 0]^T$ yielding (5.19b) with the equation for q_3 containing the nonholonomic constraint for q_1 and q_2.

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -q_1 & -q_2 \\ q_0 & -q_3 \\ q_3 & q_0 \\ -q_2 & q_1 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} \tag{5.19a}$$

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -q_1 & -q_2 \\ 1 & 0 \\ 0 & 1 \\ -q_2 & q_1 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} \tag{5.19b}$$

It can be seen when examining $\vec{q} = [q_1\ q_2\ q_3]^T$ in (5.19b) that this is a nonholonomic integrator where:

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} \tag{5.20a}$$

$$\dot{q}_3 = \frac{1}{2} \left(q_1\omega_2 - q_2\omega_1 \right) \tag{5.20b}$$

The control can be designed as:

$$\begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} = \gamma \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} + \kappa \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}. \tag{5.21}$$

Substituting (5.21) into (5.19a) yields

$$2 \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} = (\gamma q_0 + \kappa q_3) \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} + (\gamma q_3 - \kappa q_0) \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}. \tag{5.22}$$

As can be seen this expression is similar to (5.15) where

$$\alpha = \gamma q_0 + \kappa q_3 \tag{5.23a}$$

$$\beta = \gamma q_3 - \kappa q_0, \tag{5.23b}$$

or

$$\gamma = \frac{1}{\Delta}(\alpha q_0 + \beta q_3) \tag{5.24a}$$

$$\kappa = \frac{1}{\Delta}(\alpha q_3 - \beta q_0), \tag{5.24b}$$

where $\Delta = q_0^2 + q_3^2$ and similarly to (5.16)

$$\alpha = \alpha_0 \ \operatorname{sgn}(q_1^2 + q_2^2 - |q_3|) \ \ (\alpha_0 < 0) \tag{5.25a}$$

$$\beta = \beta_0 \operatorname{sgn}(q_3) \ \ (\beta_0 < 0). \tag{5.25b}$$

Singularity of such control does not create any problem since if $\Delta = 0$ any constant control that "pushes" the state from this manifold can be applied and then the algorithm (5.25) will stabilize the system.

Numerical Simulation for Nonholonomic Feedback to Quaternion System

For the simulation results, a single rotating rigid body with a $u \in \mathbb{R}^2$ control vector $u = [\omega_1, \omega_2]$ using the kinematics of (5.19a) and the control algorithm in (5.21)-(5.25) is used. In the first example, initial conditions where chosen such that they would lie outside the parabola $q_1^2 + q_2^2 = |q_3|$. Fig. 5.2 provides the results for the quaternion system, showing the system reaching the equilibrium point $\mathbf{q} = [1, 0, 0, 0]$ with initial conditions $\mathbf{q}_i = [.23, -.65, .69, -.23]$. Gains of $\alpha = -1$ and $\beta = -1$ were used for simplicity. As seen in Fig. 5.2, when the initial conditions lie outside the parabola $q_1^2 + q_2^2 = |q_3|$, the system proceeds to the sliding surface $q_3 = 0$ and then proceeds to the equilibrium point.

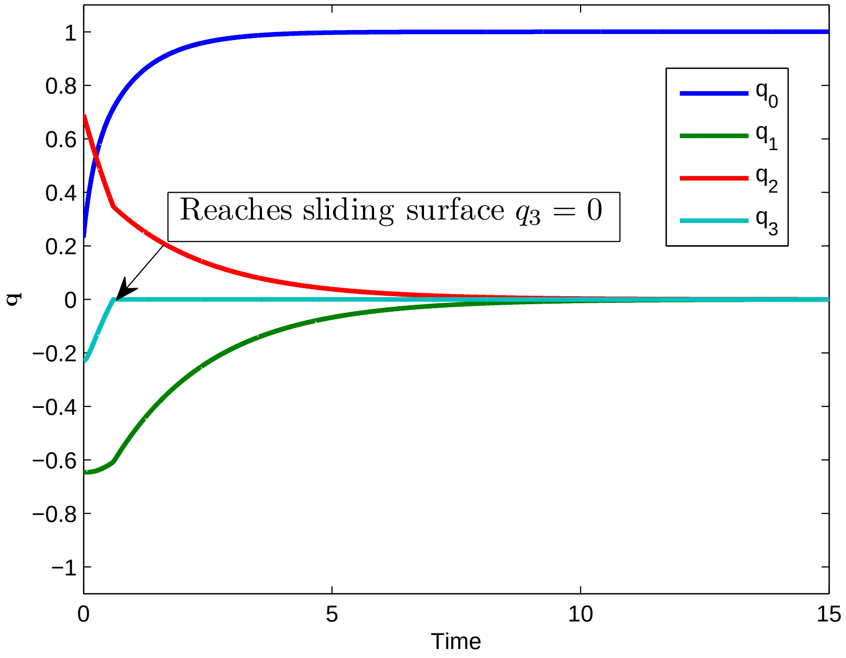

Figure 5.2: Results for quaternion system when $q_{1_o}^2 + q_{2_o}^2 > |q_{3_o}|$.

In the second example, initial conditions where chosen such that they would lie inside the parabola $q_1^2 + q_2^2 = |q_3|$. Fig. 5.3 provides the results for the quaternion system, showing the system reaching the equilibrium point $\mathbf{q} = [1, 0, 0, 0]$ with initial conditions $\mathbf{q}_i = [.26, -.26, .26, -.89]$. As seen in Fig. 5.3, when the initial conditions lie inside the parabola $q_1^2 + q_2^2 = |q_3|$, the system must first moves q_1 and q_2 outward until it reaches the parabolic surface $q_1^2 + q_2^2 = |q_3|$, then the system is able to proceed to the equilibrium point along the surface of the parabola.

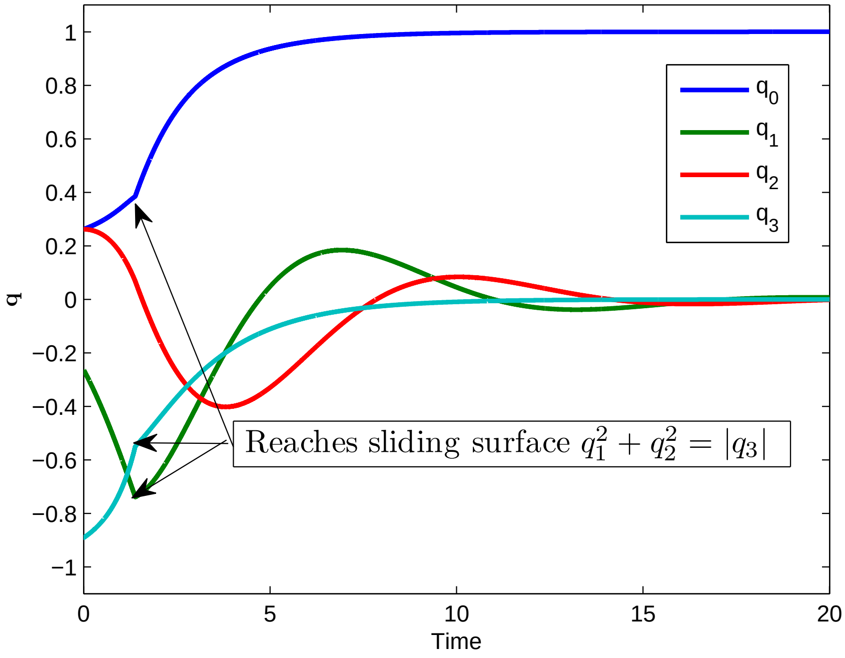

Figure 5.3: Results for quaternion system when $q_{1_o}^2 + q_{2_o}^2 < |q_{3_o}|$.

5.3 Brockett's Canonical Form

Consider a nonlinear driftless system represented by

$$\dot{\boldsymbol{x}} = B(\boldsymbol{x})\boldsymbol{u} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -x_2 & x_1 & 0 \\ x_1^2 & x_2^2 & 1 \end{bmatrix} \boldsymbol{u} \tag{5.26}$$

where $\boldsymbol{x} = [x_1, x_2, x_3, x_4]^T$ and $\boldsymbol{u} = [u_1, u_2, u_3]^T$. Since $\boldsymbol{x} \in \mathbb{R}^4$ and $\boldsymbol{u} \in \mathbb{R}^3$, the system (5.26) fails to satisfy Brockett's condition $H1$ for a canonical-form conversion.

Since the number of states is smaller than the required in $H1$, we may think of adding additional states without modifying the dynamics of $\boldsymbol{x}$. Thus, we define the augmented state vector $\boldsymbol{x_a} = [x_1, x_2, x_3, x_4, x_5, x_6]^T$ whose dynamics are given by

$$\begin{aligned} \dot{\boldsymbol{x}}_a &= B_a(\boldsymbol{x}_a)\boldsymbol{u} \qquad (5.27a) \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -x_2 & x_1 & 0 \\ x_1^2 & x_2^2 & 1 \\ \psi_{11}(\boldsymbol{x}_a) & \psi_{12}(\boldsymbol{x}_a) & \psi_{13}(\boldsymbol{x}_a) \\ \psi_{21}(\boldsymbol{x}_a) & \psi_{22}(\boldsymbol{x}_a) & \psi_{23}(\boldsymbol{x}_a) \end{bmatrix} \boldsymbol{u} \qquad (5.27b) \end{aligned}$$

wherein each ψ_{ij} will be determined to satisfy $H2$ later. Based on H2, we obtain $\mathrm{E}^{(1)}$ for $B_a(\boldsymbol{x}_a)$ as

$$E^{(1)} = \text{span}\{B_{a_1}, B_{a_2}, B_{a_3}, [B_{a_1}, B_{a_2}], [B_{a_1}, B_{a_3}], [B_{a_2}, B_{a_3}]\} \qquad (5.28)$$

where B_{a_i} is i-th column of B_a matrix; and the matrix representation of $\mathrm{E}^{(1)}$ must be nonsingular. Since the rank of $\mathrm{E}^{(1)}$ for $B(\boldsymbol{x})$ in (5.26) is 3, we have the matrix representation of $\mathrm{E}^{(1)}$ for $B_a(\boldsymbol{x}_a)$ as

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ -x_2 & x_1 & 0 & 2 & 0 & 0 \\ x_1^2 & x_2^2 & 1 & 0 & 0 & 0 \\ \psi_{11}(\boldsymbol{x}_a) & \psi_{12}(\boldsymbol{x}_a) & \psi_{13}(\boldsymbol{x}_a) & \zeta_{11} & \zeta_{12} & \zeta_{13} \\ \psi_{21}(\boldsymbol{x}_a) & \psi_{22}(\boldsymbol{x}_a) & \psi_{23}(\boldsymbol{x}_a) & \zeta_{21} & \zeta_{22} & \zeta_{23} \end{bmatrix} \tag{5.29}$$

where ζ_{ij} are given by

$$\begin{aligned} \zeta_{11} = &-\psi_{22}\psi_{11,6} + \psi_{21}\psi_{12,6} - \psi_{12}\psi_{11,5} + \psi_{11}\psi_{12,5} - x_2^2\psi_{11,4} \\ &+ x_1^2\psi_{12,4} - x_1\psi_{11,3} - x_2\psi_{12,3} - \psi_{11,2} + \psi_{12,1} \end{aligned} \tag{5.30a}$$

$$\begin{aligned} \zeta_{21} = &-\psi_{22}\psi_{21,6} + \psi_{21}\psi_{22,6} - \psi_{12}\psi_{21,5} + \psi_{11}\psi_{22,5} - x_2^2\psi_{21,4} \\ &+ x_1^2\psi_{22,4} - x_1\psi_{21,3} - x_2\psi_{22,3} - \psi_{21,2} + \psi_{22,1} \end{aligned} \tag{5.30b}$$

$$\begin{aligned} \zeta_{12} = &-\psi_{23}\psi_{11,6} + \psi_{21}\psi_{13,6} - \psi_{13}\psi_{11,5} + \psi_{11}\psi_{13,5} - \psi_{11,4} \\ &+ x_1^2\psi_{13,4} - x_2\psi_{13,3} + \psi_{13,1} \end{aligned} \tag{5.30c}$$

$$\begin{aligned} \zeta_{22} = &-\psi_{23}\psi_{21,6} + \psi_{21}\psi_{23,6} - \psi_{13}\psi_{21,5} + \psi_{11}\psi_{23,5} \\ &- \psi_{21,4} + x_1^2\psi_{23,4} - x_2\psi_{23,3} + \psi_{23,1} \end{aligned} \tag{5.30d}$$

$$\begin{aligned} \zeta_{13} = &-\psi_{23}\psi_{12,6} + \psi_{22}\psi_{13,6} - \psi_{13}\psi_{12,5} + \psi_{12}\psi_{13,5} \\ &- \psi_{12,4} + x_2^2\psi_{13,4} + x_1\psi_{13,3} + \psi_{13,2} \end{aligned} \tag{5.30e}$$

$$\begin{aligned} \zeta_{23} = &-\psi_{23}\psi_{2,6} + \psi_{22}\psi_{23,6} - \psi_{13}\psi_{22,5} + \psi_{12}\psi_{23,5} \\ &- \psi_{22,4} + x_2^2\psi_{23,4} + x_1\psi_{23,3} + \psi_{23,2} \end{aligned} \tag{5.30f}$$

In (5.30), arguments of the function are suppressed for notational simplicity and $\psi_{ij,k}$ is a notation for to $\partial\psi_{ij}/\partial x_k$. Although (5.30) looks complicated, one only needs to focus on $\zeta_{12}, \zeta_{13}, \zeta_{22}$, and ζ_{23} because these define the rank of $\mathrm{E}^{(1)}$ for B_a based on (5.29). Thus, the following ψ_{ij} are chosen:

$$\psi_{11} = 0 \tag{5.31a}$$

$$\psi_{12} = x_1^3 + x_2^3 - 6x_2 - 3x_4 \tag{5.31b}$$

$$\psi_{13} = 3x_2 \tag{5.31c}$$

$$\psi_{21} = x_1^3 - 3x_1^2 + x_2^3 - 3x_4 \tag{5.31d}$$

$$\psi_{22} = -3x_2^2 \tag{5.31e}$$

$$\psi_{23} = 3x_1 - 3 \tag{5.31f}$$

Therefore, the augmented system represented by (5.27) with (5.31) satisfies both $H1$ and $H2$ so that (5.27) can be transformed to Brockett's canonical form by Theorem 3. Accordingly, the coordinate transition for Brockett's canonical form is obtained as

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ -\frac{1}{3}(x_1^3 + x_2^3) + x_4 \\ x_3 \\ x_4 + \frac{x_6}{3} \\ x_2^2 + \frac{x_5}{3} \end{bmatrix} \tag{5.32}$$

In fact, the obtained transformation in (5.32) is a global diffeomorphism and thus Theorem 3 is valid about any $x_0 \in \mathbb{R}^n$ for (5.26). Transformed system is

$$\dot{\boldsymbol{z}} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_2 z_1 - u_1 z_2 \\ u_3 z_1 - u_1 z_3 \\ u_3 z_2 - u_2 z_3 \end{bmatrix} \tag{5.33}$$

Generalization of the described scheme (augmenting the system with additional states) to arbitrary nonlinear driftless system is given by the following theorem which is the extension of Theorem 3.

Theorem 5. *Consider the system represented by* (5.1) *and* $\operatorname{rank}(E^{(1)}) = n$. *If* $n \leq m(m+1)/2$, *then there exist* $\psi_{ij}(\boldsymbol{x}_a)$ *with* $\boldsymbol{x}_a = [x_1, \ldots, x_{m(m+1)/2}]^T$, $i = 1, \ldots, m(m+1)/2 - n$ *and* $j = 1, \ldots, n$ *which augment the system* (5.1) *to be* (5.27a)*; and coordinate transitions* $x_a \mapsto (x^1, x^2, \ldots, x^m, y^{1,2}, y^{1,3}, \ldots, y^{m-1,m})$ *for* (5.27a) *in a neighborhood of a given point, say* $x = 0$, *so that the equations are represented by*

$$\dot{x}^i = u^i + r^i, \quad i = 1, 2, \ldots, m \tag{5.34a}$$

$$\dot{y}^{ij} = u^i x^j - u^j x^i + r^{ij}, \quad i, j = 1, 2, \ldots, m, \; i < j \tag{5.34b}$$

where the r^i *and* r^{ij} *have vanishing first partials with respect to* x *and* y *and in addition* r^{ij} *has vanishing second partials with respect to* x^i *and* x^j.

Proof. Since $\mathrm{rank}(\mathrm{E}^{(1)}) = n$, the matrix form of $\mathrm{E}^{(1)}$ can be represented as a lower triangular matrix with nonzero diagonal elements about $\boldsymbol{x} = 0$ without loss of generality as follows,

$$\mathrm{E}^{(1)} = [A|B] \tag{5.35}$$

where $A \in \mathbb{R}^{\frac{m(m+1)}{2} \times n}$ and $B \in \mathbb{R}^{\frac{m(m+1)}{2} \times \frac{m(m-1)}{2}}$ are given by

$$A = \begin{bmatrix} b_{1,1} & \dots & & & & 0 \\ \vdots & & \ddots & & & 0 \\ b_{n,1} & & & & & b_{n,n} \\ \psi_{1,1} & \dots & \psi_{1,m} & \zeta_{1,1} & \dots & \zeta_{1,n-m} \\ \vdots & \ddots & & & & \vdots \\ \psi_{n_1,1} & \dots & \psi_{n_1,m} & \zeta_{n_1,1} & \dots & \zeta_{n_1,n-m} \end{bmatrix} \tag{5.36}$$

$$B = \begin{bmatrix} 0 & \dots & 0 \\ \vdots & & \vdots \\ 0 & \dots & 0 \\ \zeta_{1,n-m+1} & \cdots & \zeta_{1,\frac{m(m-1)}{2}} \\ \vdots & \ddots & \vdots \\ \zeta_{n_1,n-m+1} & \cdots & \zeta_{n_1,\frac{m(m-1)}{2}} \end{bmatrix} \tag{5.37}$$

where $n_1 = m(m+1)/2 - n$ as in (5.5). For $\mathrm{E}^{(1)}$ of (5.35) to be a full rank matrix, the square matrix $B_s = [\zeta_{i,j}],\ i = 1, \dots, d,\ j = n-m+1, \dots, m(m-1)/2$ in B should be nonsingular. Since $\boldsymbol{x} \in \mathbb{R}^n$ is in the $\mathcal{C}^1$-manifold, $\mathcal{M}$, whose tangent space is spanned by column vectors of A and $\mathrm{rank}(A)$ is n, $\mathcal{M}$ can be obtained via an immersion from $\mathcal{M}_a$ where $\boldsymbol{x}_a$ resides. Therefore, we may conclude that there exist $\psi_{i,j}(\boldsymbol{x}_a)$ casting

B_s nonsingular matrix about $\boldsymbol{x} = 0$. This makes $B_a(\boldsymbol{x}_a)$ satisfy both $H1$ and $H2$. Therefore, by Theorem 3, there exists a necessary coordinate transition for Brockett's canonical form. □

Remark 1. In the previous example, (5.26) satisfies $\mathrm{rank}(\mathrm{E}^{(1)}) = \mathrm{n}$ with $n = 4$. Thus, there exists $\psi_{i,j}$ which renders the augmented system to be a controllable Brockett's canonical form. In order to find $\psi_{i,j}$, it is required to satisfy the full rank condition for the augmented system involving partial derivatives as shown in the motivating example. However, we may use the various control methods developed for Brockett's canonical system as a consequence of Theorem 5.

Remark 2. $\boldsymbol{x}$ defines the flow on n-manifold, $\mathcal{M}$, which is a submanifold of $m(m+1)/2$-dimensional manifold, $\mathcal{M}_a$, where $\boldsymbol{x}_a$ flow is defined. As a result, $\mathcal{M}$ can also be regarded as a projection of $\mathcal{M}_a$, and thus $\boldsymbol{x}_a$ need not be zero while $\boldsymbol{x}$ is zero in the motivating example for control purposes.

5.3.1 Unicycle Example

This section presents the implementation of Theorem 5 for unicycle systems starting from a single unicycle, shown in Fig. 5.4. In the case of the nonholonomic system consisting of two unicycles coupled by one common control input for heading, a conventional control strategy is also demonstrated for the comparison with the proposed generalized augmentation approach.

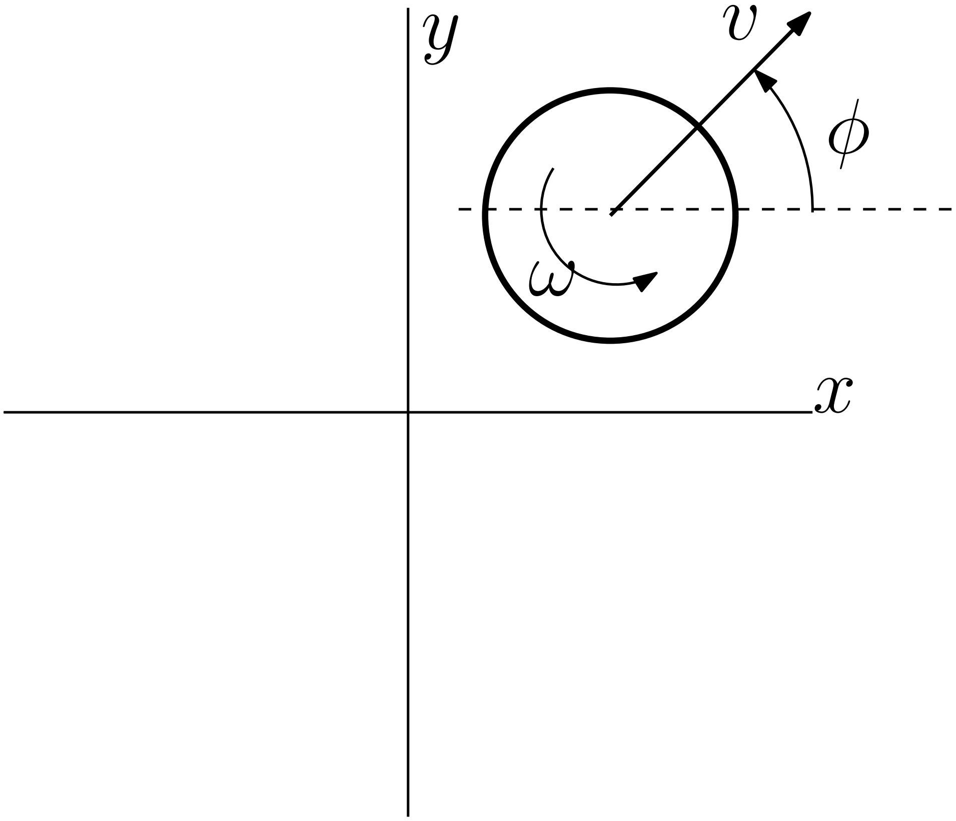

Figure 5.4: Unicycle system diagram.

A single simple unicycle vehicle model can be written as

$$\dot{x} = v \cos \phi \tag{5.38a}$$

$$\dot{y} = v \sin \phi \tag{5.38b}$$

$$\dot{\phi} = \omega \tag{5.38c}$$

where the forward velocity and heading velocity controls are v and ω respectively.

We can define a change of coordinates $F(\phi)$

$$\begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^T = F(\phi) \begin{bmatrix} x & y & \phi \end{bmatrix}^T \tag{5.39}$$

where

$$F(\phi) = \begin{bmatrix} 0 & 0 & 1 \\ \cos\phi & \sin\phi & 0 \\ \phi\cos\phi - 2\sin\phi & \phi\sin\phi + 2\cos\phi & 0 \end{bmatrix} \tag{5.40}$$

and a nonsingular state-dependent transformation of the controls

$$u_1 = \omega \tag{5.41a}$$

$$u_2 = v + \left(\frac{x_3}{2} - \frac{x_1 x_2}{2}\right)\omega \tag{5.41b}$$

yielding the system

$$\dot{x}_1 = u_1 \tag{5.42a}$$

$$\dot{x}_2 = u_2 \tag{5.42b}$$

$$\dot{Y} = \begin{bmatrix} 0 & x_1 u_2 - x_2 u_1 \\ -x_1 u_2 + x_2 u_1 & 0 \end{bmatrix} \tag{5.42c}$$

Two simple unicycle vehicles (vehicles a and b) with coupled steering (same ϕ and ω)

but independent velocity control can be written as:

$$\dot{x}_a = v_a \cos\phi \tag{5.43a}$$

$$\dot{y}_a = v_a \sin\phi \tag{5.43b}$$

$$\dot{x}_b = v_b \cos\phi \tag{5.43c}$$

$$\dot{y}_b = v_b \sin\phi \tag{5.43d}$$

$$\dot{\phi} = \omega \tag{5.43e}$$

and after applying the transformation in (5.40) the kinematics of (5.43) can be written in the form of (5.44):

$$\dot{x}_1 = u_1 \tag{5.44a}$$

$$\dot{x}_2 = u_2 \tag{5.44b}$$

$$\dot{x}_3 = u_3 \tag{5.44c}$$

$$\dot{x}_4 = x_2 u_3 - x_3 u_2 \tag{5.44d}$$

$$\dot{x}_5 = x_3 u_1 - x_1 u_3 \tag{5.44e}$$

$$\dot{x}_6 = x_1 u_2 - x_2 u_1 \tag{5.44f}$$

where u_2 and u_3 account for the independent velocity controls of the vehicles and u_1 accounts for the steering control for *both* vehicles. (5.44d) is added to satisfy $H1$. (5.44e and 5.44f) come naturally from the transformation of the original system in (5.43). (5.44d) is an additional constraint added to the kinematics that puts an added constraint linking the velocity of the vehicles together. While this added constraint is a more limited system, the control algorithm is able to stabilize the system. For N

vehicles, $\frac{N(N-1)}{2}$ additional equations must be added. The system can also be written as

$$\dot{\boldsymbol{x}} = \boldsymbol{u} \tag{5.45a}$$

$$\dot{\boldsymbol{Y}} = \boldsymbol{x}\boldsymbol{u}^T - \boldsymbol{u}\boldsymbol{x}^T = [\boldsymbol{u}, \boldsymbol{x}] \tag{5.45b}$$

where

$$\begin{aligned}
\boldsymbol{x} &= \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^T \\
\dot{\boldsymbol{Y}} &= \begin{bmatrix} 0 & x_1u_2 - x_2u_1 & x_3u_1 - x_1u_3 \\ -x_1u_2 + x_2u_1 & 0 & x_2u_3 - x_3u_2 \\ -x_3u_1 + x_1u_3 & -x_2u_3 + x_3u_2 & 0 \end{bmatrix} \\
\boldsymbol{u} &= \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix}^T
\end{aligned} \tag{5.46}$$

Multiple Unicycles Regulation Problem

For the simulation results, four unicycles were considered using the control strategy describe in section 5.2. Table 5.1 provides the initial conditions that were used for the four unicycles. Gains of $\alpha = 1, \beta = 1$ were used for simplicity.

Table 5.1: Initial conditions for unicycle regulation example.

	x_i	y_i	ϕ_i
Vehicle A	2	2	230°
Vehicle B	-2	1.5	230°
Vehicle C	-1.5	-2	230°
Vehicle D	2	-1	230°

Fig. 5.5 provides the results for the transformed coordinate system. These results correspond to the transformed coordinates vs. time, the top plot displaying the x-coordinates and the bottom displaying the y-coordinates. The step labels refer to areas in the results that correspond to the various steps of the control algorithm. The results for x_1 are not displayed in 5.5, but displayed as ϕ in Fig. 5.7. The y values that are displayed correspond to the elements of the matrix Y that correspond to the coordinates that come from the original system naturally, as opposed to the ones that are added in order to meet the Brockett conditions. Each step in the control strategy can be easily seen in the results. Initially the x-coordinates are seen to drive to the origin and then to another value corresponding to the initial eigenvector, all the while the values for the y-coordinates remain unchanged. Then, during step 2 of the control strategy, the y values are driven to zero. Since there is only one initial eigenvalue, the algorithm then proceeds to step 4 were the x values are driven to zero once again.

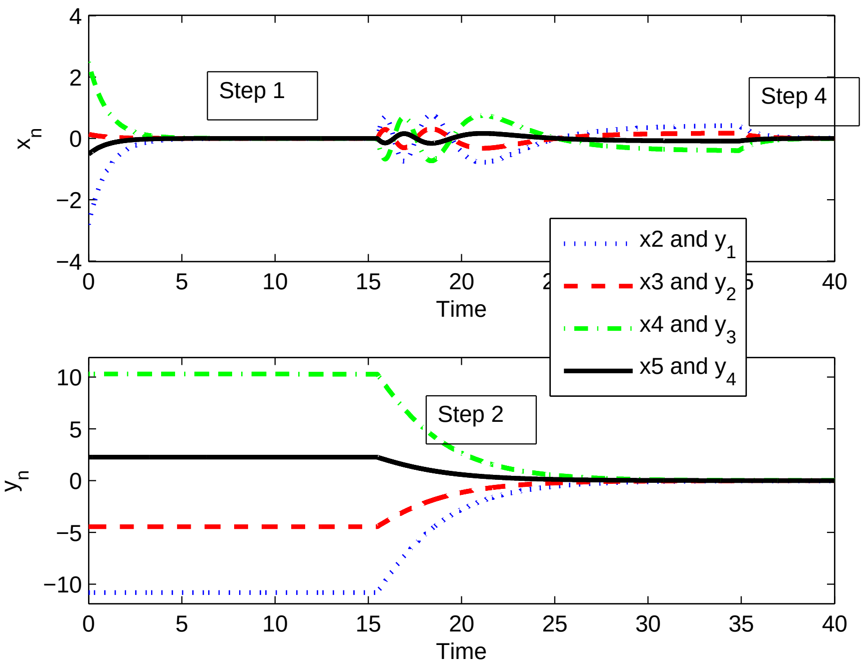

Figure 5.5: Results for translated system of a regulation example including four unicycles.

The various steps of the control strategy are also easily illustrate in Fig. 5.6 which displays the Lyapunov functions U, V_1 and V_2. These results correspond to the polynomial equations U, V_1, V_2 vs. time. The top plot displays $U = \frac{1}{2}\langle x, x\rangle$, the middle plot displays $V_1 = \frac{1}{2}\langle Y, Y\rangle$ and the bottom plot displays $V_2 = \frac{1}{2}\langle [x, Y], [x, Y]\rangle$. The step labels refer to areas in the results that correspond to the various steps of the control algorithm. Step 1 is broken up to subparts A and B to illustrate the controller first driving to the origin and then to the first eigenvector of Y^TY.

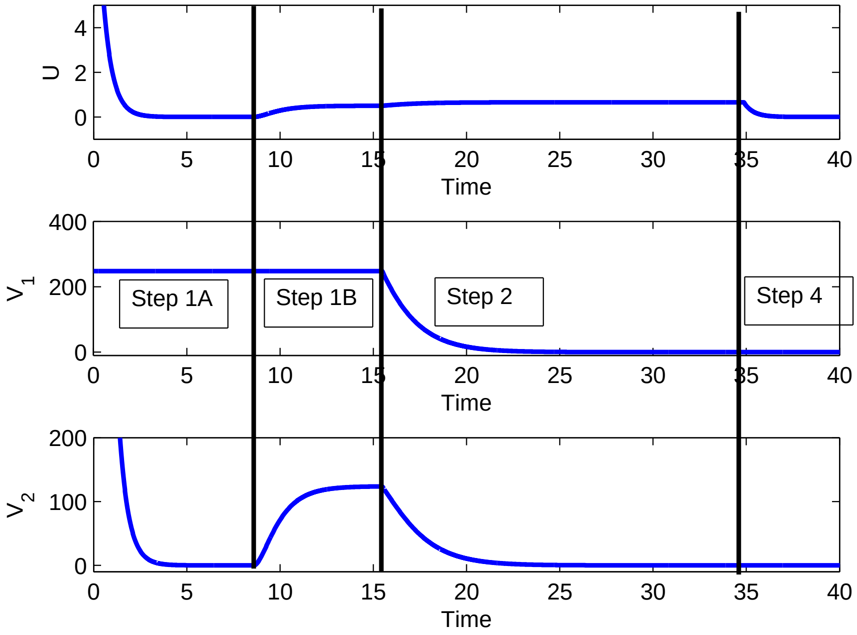

Figure 5.6: Results for Lyapunov functions of a regulation example including four unicycles.

Figs. 5.7 and 5.8 display the simulation results in terms of the original system. Fig. 5.7 provides the results vs. time. These results correspond to the original coordinates vs. time. The top plot displays the results for ϕ, which corresponds to the heading angle for all four unicycles. The middle and bottom plot displays the results for the x-coordinates and y-coordinates of the four unicycles respectively.

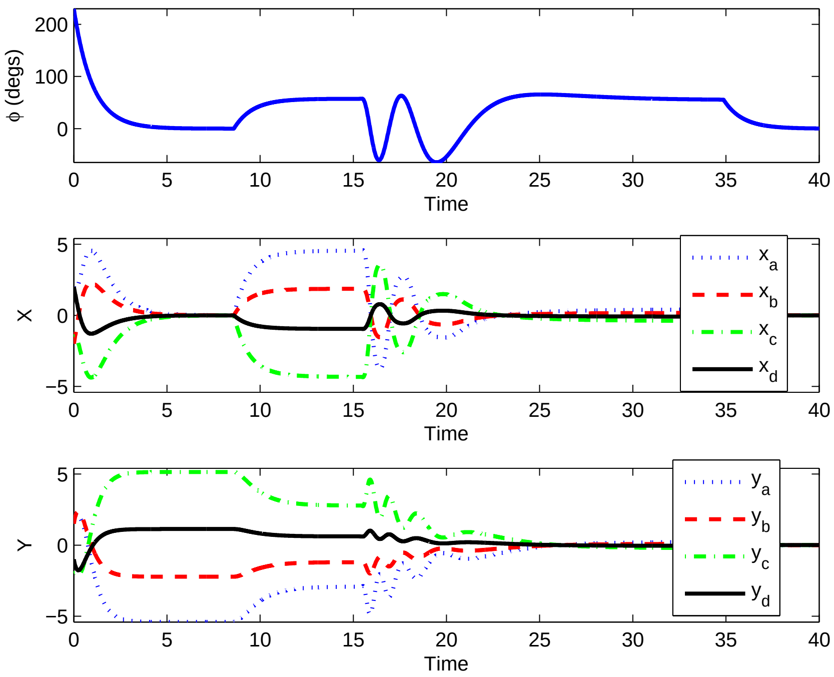

Figure 5.7: Results for original coordinates of a regulation example including four unicycles.

Fig. 5.8 is a plot of the $x - y$ coordinates, showing the paths that each of the four unicycles take. These results correspond to the original x and y coordinates. The x-coordinates are displayed along the horizontal axis with the y-coordinates displayed along the vertical axis.

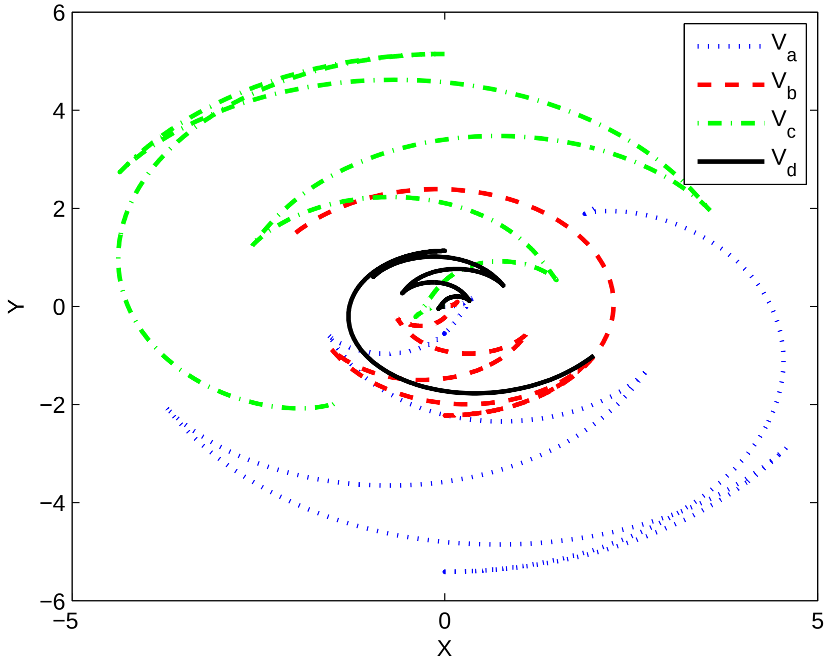

Figure 5.8: Results for X-Y coordinates of a regulation example including four unicycles.

Multiple Unicycles Tracking Problem

Simulation results are also provided to demonstrate the tracking ability of the control strategy in section 5.2. For this example only two unicycles are used to allow for displaying both the desired path and the actual path. The following provides the initial conditions that were used for the two unicycles plus the desired trajectory used. The desired trajectories for each of the vehicles is listed in Table 5.2 and their initial conditions are listed in Table 5.3. Gains of $\alpha = 1, \beta = 1$ were used for simplicity.

Table 5.2: Desired trajectories for unicycle tracking problem.

	Vehicle A	**Vehicle B**
x_{d_i}	0	0
y_{d_i}	1	-1
ϕ_{d_i}	0°	0°
$\dot{x}_d$	$.2\cos\phi$	$.2\cos\phi$
$\dot{y}_d$	$.2\sin\phi$	$.2\sin\phi$
ϕ_d	$\frac{\pi}{4}\sin\frac{\pi}{9}t$	$\frac{\pi}{4}\sin\frac{\pi}{9}t$

Table 5.3: Initial conditions for unicycle tracking problem.

	Vehicle A	**Vehicle B**
x_i	2	2
y_i	-1	-1.5
ϕ_i	230°	230°

The results provided in Fig.5.9 illustrate the error values $\bar{x}$ and $\bar{y}$ of the transformed coordinate system. These results correspond to the transformed error coordinates vs. time, the top plot displaying the $\bar{x}$-coordinates and the bottom displaying the $\bar{y}$-coordinates. Again the $\bar{y}$ values that are displayed correspond to the elements of the matrix $\bar{Y}$ that correspond to the coordinates that come from the original system naturally, as opposed to the ones that are added in order to meet the Brockett conditions.

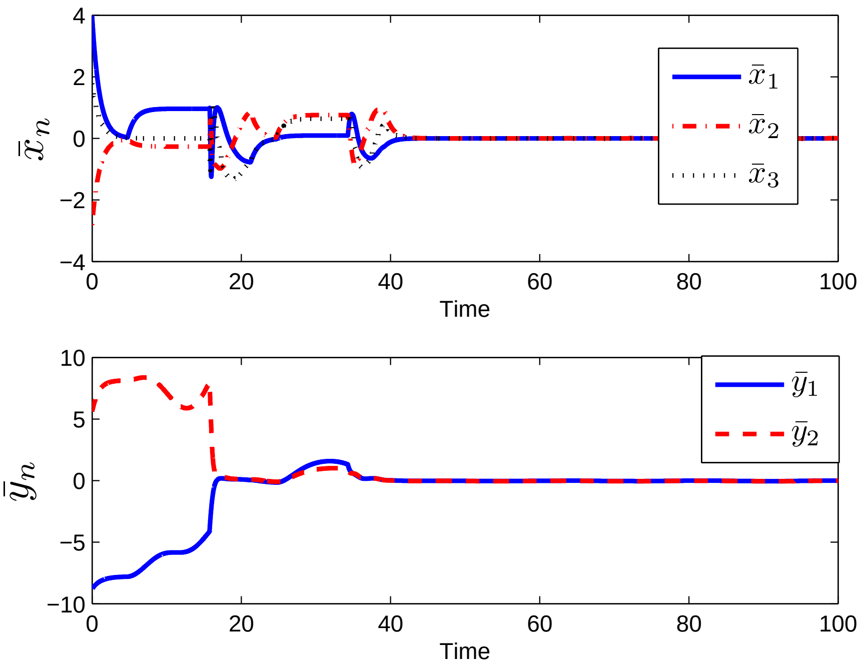

Figure 5.9: Results for translated error coordinates of a tracking example including two unicycles.

Fig. 5.10 displays the Lyapunov functions U, V_1 and V_2. These results correspond to the polynomial equations U, V_1, V_2 vs. time. The top plot displays $U = \frac{1}{2}\langle \bar{x}, \bar{x} \rangle$, the middle plot displays $V_1 = \frac{1}{2}\langle \bar{Y}, \bar{Y} \rangle$ and the bottom plot displays $V_2 = \frac{1}{2}\langle [\bar{x}, \bar{Y}], [\bar{x}, \bar{Y}] \rangle$. The various steps of the control strategy are more difficult to distinguish in the tracking problem, but in can be seen from the results for U in Fig.5.10 that in this example, there were two eigenvalues of $\bar{Y}^T\bar{Y}$ as $\bar{x}$ is seen to drive towards the first eigenvector around 15 secs and to the second eigenvector around 25 seconds.

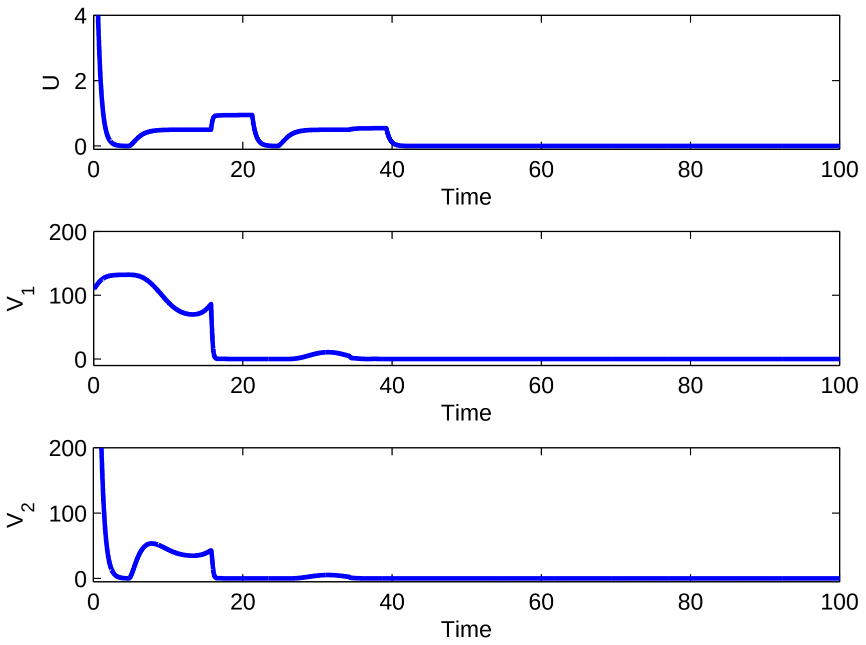

Figure 5.10: Results for Lyapunov functions for a tracking example including two unicycles.

Figs. 5.11 and 5.12 display the simulation results in terms of the original system as compared to the desired track path. Fig. 5.11 provides the results vs. time. These results correspond to the original coordinates vs. time. The top plot displays the results for ϕ and ϕ_d, which correspond to the heading angle and desired heading angle for both unicycles. The middle and bottom plot displays the results and desired trajectories for the x-coordinates and y-coordinates of the unicycles respectively.

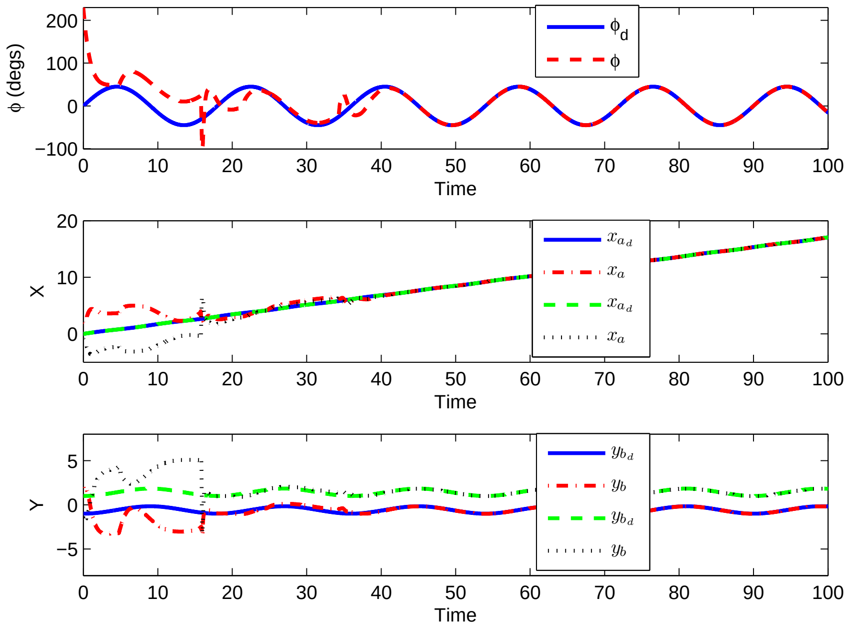

Figure 5.11: Results for original coordinates for a tracking example including two unicycles.

Fig.5.12 is a plot of the $x - y$ coordinates, showing the paths that each of the unicycles take plus the desired trajectories of both unicycles. These results correspond to the original x and y coordinates. The x-coordinates are displayed along the horizontal axis with the y-coordinates displayed along the vertical axis.

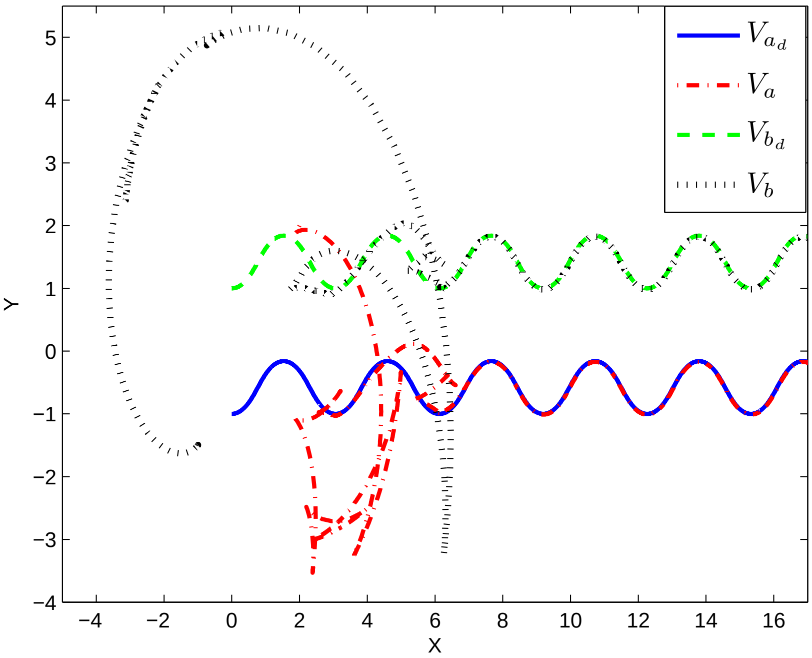

Figure 5.12: Results for X-Y coordinates for a tracking example including two unicycles plus their desired trajectories.

5.3.2 Dual Quaternion Example

This section presents the implementation of Theorem 5 for dual quaternion systems. The example that is considered is the kinematics for multiple spacecraft in free space. An interesting feature of the kinematics for dual quaternion systems is that they are easily modified to meet Brockett conditions. For review, the kinematics for dual quaternions are defined as:

$$\dot{\hat{\mathbf{q}}} = \frac{1}{2}\hat{\mathbf{q}}\hat{\omega}. \tag{5.47}$$

For this example it is assumed that the spacecraft is underactuated, using only 2 rotational controllers and one translational controller, where $\hat{\vec{\omega}} = [0, \omega_2, \omega_3]^T + \epsilon[v_x, 0, 0]^T$. The kinematics from (5.47) are reduced and written in matrix form to reflect this $\hat{\vec{\omega}}$.

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}'_0 \\ \dot{q}'_1 \\ \dot{q}'_2 \\ \dot{q}'_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -q_2 & -q_3 & 0 \\ -q_3 & q_2 & 0 \\ q_0 & -q_1 & 0 \\ q_1 & q_0 & 0 \\ -q'_2 & -q'_3 & -q_1 \\ -q'_3 & q'_2 & q_0 \\ q'_0 & -q'_1 & q_3 \\ q'_1 & q'_0 & -q_2 \end{bmatrix} \begin{bmatrix} \omega_2 \\ \omega_3 \\ v_x \end{bmatrix} \tag{5.48}$$

The kinematics of (5.48) for components q_2, q_3, and q'_1 are linearized about the point $\hat{\mathbf{q}} = [1, 0, 0, 0]^T + \epsilon[0, 0, 0, 0]^T$ and rearranged to match the canonical form in (5.7). The states q_0 and q'_0 are not used in the linearized model and assumed to be 1 and 0 respectively. This is an expression of the constraints characteristic of $\mathbb{DH}_u$ of $|\mathbf{q}| = 1$ and $\mathbf{q} \cdot \mathbf{q}' = 0$.

$$\begin{bmatrix} \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}'_1 \\ \dot{q}'_2 \\ \dot{q}'_3 \\ \dot{q}_1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -q'_1 & q_3 \\ q'_1 & 0 & -q_2 \\ -q_3 & q_2 & 0 \end{bmatrix} \begin{bmatrix} \omega_2 \\ \omega_3 \\ v_x \end{bmatrix} \tag{5.49}$$

As seen in (5.49), the linearized kinematics naturally satisfy $H1$ and $H2$ and can now be defined in the form of (5.7) where

$$x = [q_2, q_3, q_1']^T \tag{5.50a}$$

$$u = [\omega_2, \omega_3, v_x]^T \tag{5.50b}$$

$$Y = \begin{bmatrix} 0 & q_1 & -q_3' \\ -q_1 & 0 & q_2' \\ q_3' & -q_2' & 0 \end{bmatrix} \tag{5.50c}$$

The next step is to examine the kinematics for multiple space vehicles. As in the unicycle example, an additional constraint is added such that all the vehicles must have the same orientation. The kinematics of (5.48) are extended to include the dual kinematics for two additional vehicles, for a total of three vehicles (A, B, C). The real component of the dual quaternion kinematics, which represents the rotational kinematics, is assumed to be the same for each vehicle to reflect the constraint that they must all have the same attitude orientation.

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}'_{0_A} \\ \dot{q}'_{1_A} \\ \dot{q}'_{2_A} \\ \dot{q}'_{3_A} \\ \dot{q}'_{0_B} \\ \dot{q}'_{1_B} \\ \dot{q}'_{2_B} \\ \dot{q}'_{3_B} \\ \dot{q}'_{0_C} \\ \dot{q}'_{1_C} \\ \dot{q}'_{2_C} \\ \dot{q}'_{3_C} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -q_2 & -q_3 & 0 & 0 & 0 \\ -q_3 & q_2 & 0 & 0 & 0 \\ q_0 & -q_1 & 0 & 0 & 0 \\ q_1 & q_0 & 0 & 0 & 0 \\ -q'_{2_A} & -q'_{3_A} & -q_1 & 0 & 0 \\ -q'_{3_A} & q'_{2_A} & q_0 & 0 & 0 \\ q'_{0_A} & -q'_{1_A} & q_3 & 0 & 0 \\ q'_{1_A} & q'_{0_A} & -q_2 & 0 & 0 \\ -q'_{2_B} & -q'_{3_B} & 0 & -q_1 & 0 \\ -q'_{3_B} & q'_{2_B} & 0 & q_0 & 0 \\ q'_{0_B} & -q'_{1_B} & 0 & q_3 & 0 \\ q'_{1_B} & q'_{0_B} & 0 & -q_2 & 0 \\ -q'_{2_C} & -q'_{3_C} & 0 & 0 & -q_1 \\ -q'_{3_C} & q'_{2_C} & 0 & 0 & q_0 \\ q'_{0_C} & -q'_{1_C} & 0 & 0 & q_3 \\ q'_{1_C} & q'_{0_C} & 0 & 0 & -q_2 \end{bmatrix} \begin{bmatrix} \omega_2 \\ \omega_3 \\ v_{x_A} \\ v_{x_B} \\ v_{x_C} \end{bmatrix} \tag{5.51}$$

The subscripts A, B, and C refer to each of the 3 vehicles being considered. The same linearization and rearrangement that was done in (5.49) is done to (5.51). However, for the kinematics to satisfy $H1$, additional kinematic equations must be added. Again, the number of additional constraints that must be added is $\frac{N(N-1)}{2}$ where N is the number of vehicles. For three vehicles, three additional constraints are required, referred to as q_{AB}, q_{AC}, and q_{BC}. These added kinematic equations place an additional constraint relating the velocity of the vehicles together. The result is the following:

$$\begin{bmatrix} \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}'_{1_A} \\ \dot{q}'_{1_B} \\ \dot{q}'_{1_C} \\ \dot{q}'_{2_C} \\ \dot{q}'_{3_C} \\ \dot{q}'_{2_B} \\ \dot{q}'_{3_B} \\ \dot{q}'_{2_A} \\ \dot{q}'_{3_A} \\ \dot{q}_1 \\ \dot{q}_{AB} \\ \dot{q}_{AC} \\ \dot{q}_{BC} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & -q'_{1_C} & 0 & 0 & q_3 \\ q'_{1_C} & 0 & 0 & 0 & -q_2 \\ 0 & -q'_{1_B} & 0 & q_3 & 0 \\ q'_{1_B} & 0 & 0 & -q_2 & 0 \\ 0 & -q'_{1_A} & q_3 & 0 & 0 \\ q'_{1_A} & 0 & -q_2 & 0 & 0 \\ -q_3 & q_2 & 0 & 0 & 0 \\ 0 & 0 & q'_{1_B} & -q'_{1_A} & 0 \\ 0 & 0 & q'_{1_C} & 0 & -q'_{1_A} \\ 0 & 0 & 0 & q'_{1_C} & -q'_{1_B} \end{bmatrix} \begin{bmatrix} \omega_2 \\ \omega_3 \\ v_{x_A} \\ v_{x_B} \\ v_{x_C} \end{bmatrix} \tag{5.52}$$

The result in (5.52) now satisfies $H1$ and $H2$ and can be defined in the form of (5.7) where

$$x = \left[q_2, q_3, q'_{1_A}, q'_{1_B}, q'_{1_C}\right]^T \tag{5.53a}$$

$$u = \left[\omega_2, \omega_3, v_{x_A}, v_{x_B}, v_{x_C}\right]^T \tag{5.53b}$$

$$Y = \begin{bmatrix} 0 & q_1 & -q'_{3_A} & -q'_{3_B} & -q'_{3_C} \\ -q_1 & 0 & q'_{2_A} & q'_{2_B} & q'_{2_C} \\ q'_{3_A} & -q'_{2_A} & 0 & q_{AB} & q_{AC} \\ q'_{3_B} & -q'_{2_B} & -q_{AB} & 0 & q_{BC} \\ q'_{3_C} & -q'_{2_C} & -q_{AC} & -q_{BC} & 0 \end{bmatrix} \tag{5.53c}$$

Multiple Dual Quaternion Spacecraft Regulation Problem

For the simulation results, three spacecraft were considered using the control strategy in section 5.2. Table 5.4 provides the initial positions that were used for the three spacecraft. An initial attitude of $\mathbf{q} = [.50, -.40, .68, .35]^T$ was for all three spacecraft. Gains of $\alpha = 3, \beta = 3$ were used.

Table 5.4: Initial conditions for three spacecraft dual quaternion example.

	p_x	p_y	p_z
Vehicle A	2	1	-1.5
Vehicle B	-1.5	-1	2
Vehicle C	-1	-2	1.5

Fig. 5.13 provides the attitude $\mathbf{q}$ results for all three spacecraft. As expected, the vector portion of $\mathbf{q}$ is seen converging to 0, while the scalar portion q_0 is seen

converging to 1.

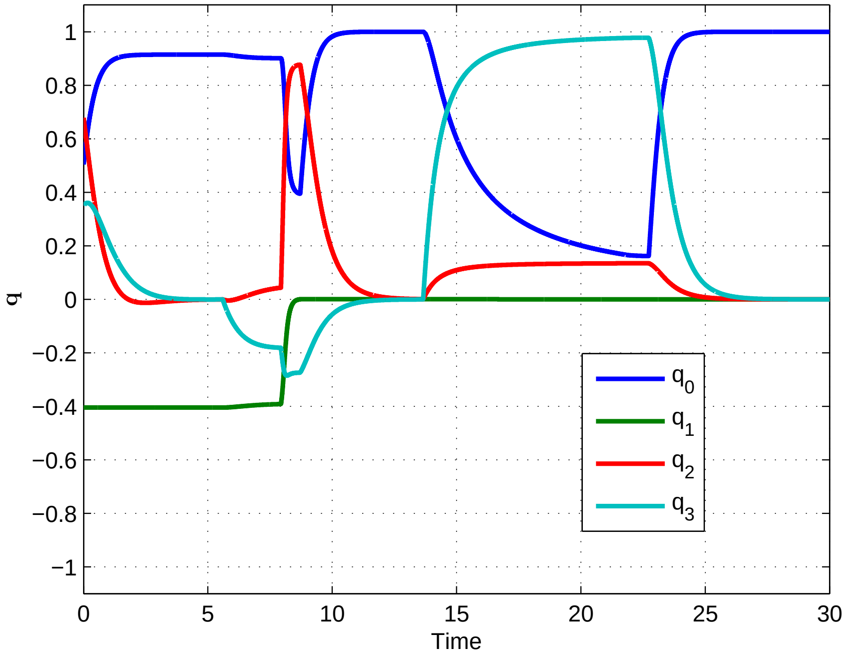

Figure 5.13: Attitude **q** results for all three spacecraft.

Figs. 5.14, 5.15, and 5.16 provide the translational $\vec{p}$ results for spacecrafts A, B, and C respectively. As expected, all three spacecraft results converge to 0.

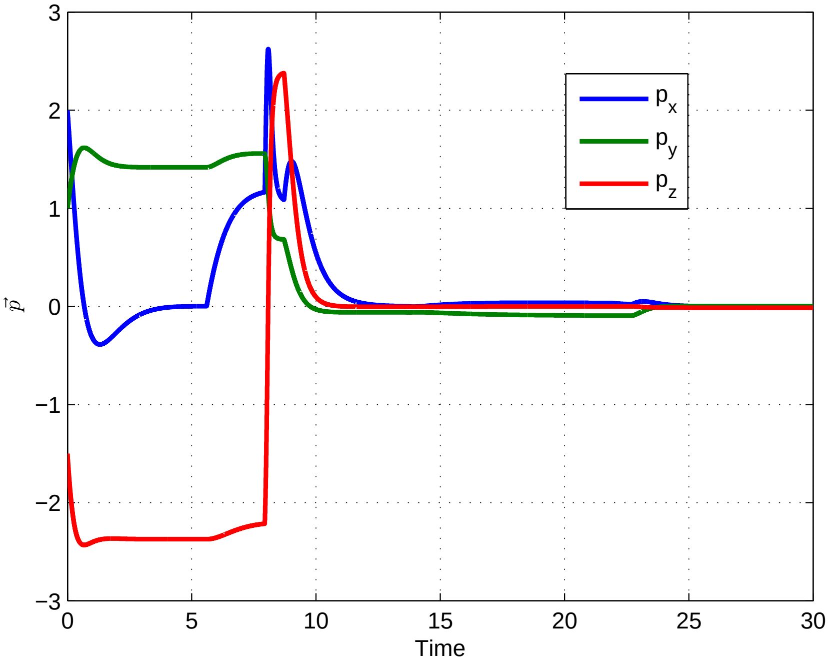

Figure 5.14: Translation $\vec{p}$ results for spacecraft A.

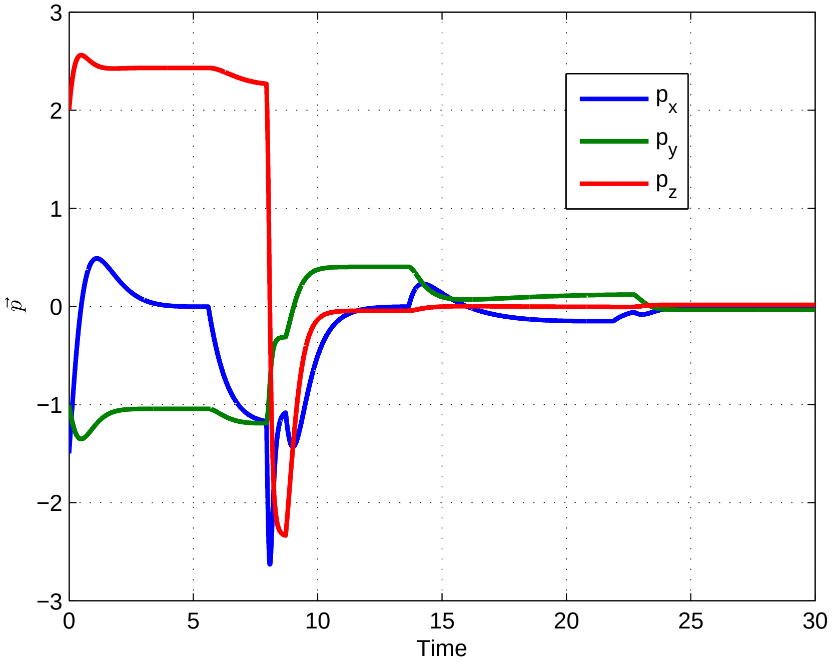

Figure 5.15: Translation $\vec{p}$ results for spacecraft B.

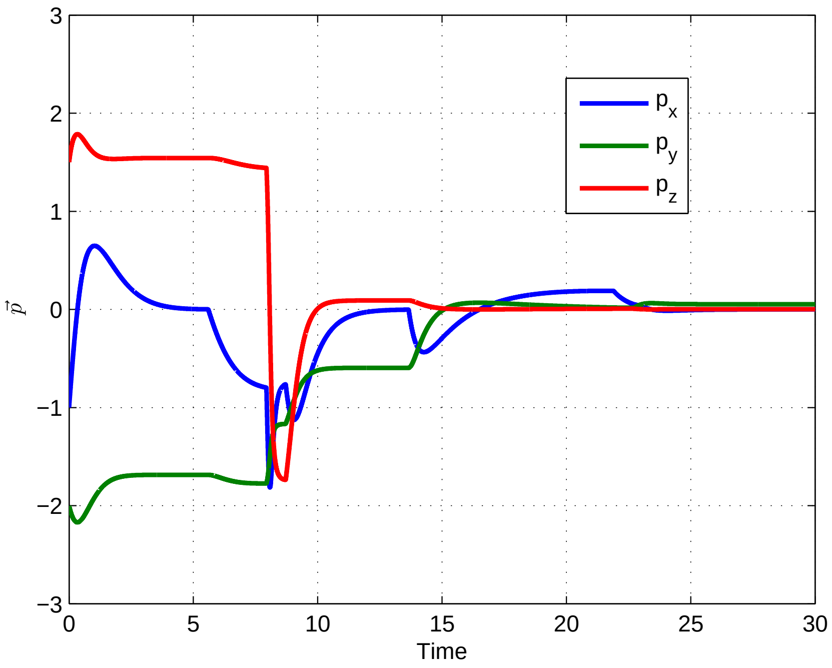

Figure 5.16: Translation $\vec{p}$ results for spacecraft C.

Fig. 5.17 provides the results for the Lyapunov functions U, V_1 and V_2. These results correspond to the polynomial equations U, V_1, V_2 vs. time. The top plot displays $U = \frac{1}{2}\langle x, x\rangle$, the middle plot displays $V_1 = \frac{1}{2}\langle Y, Y\rangle$ and the bottom plot displays $V_2 = \frac{1}{2}\langle [x, Y], [x, Y]\rangle$. The various steps of the control strategy are can be seen in Fig. 5.17.

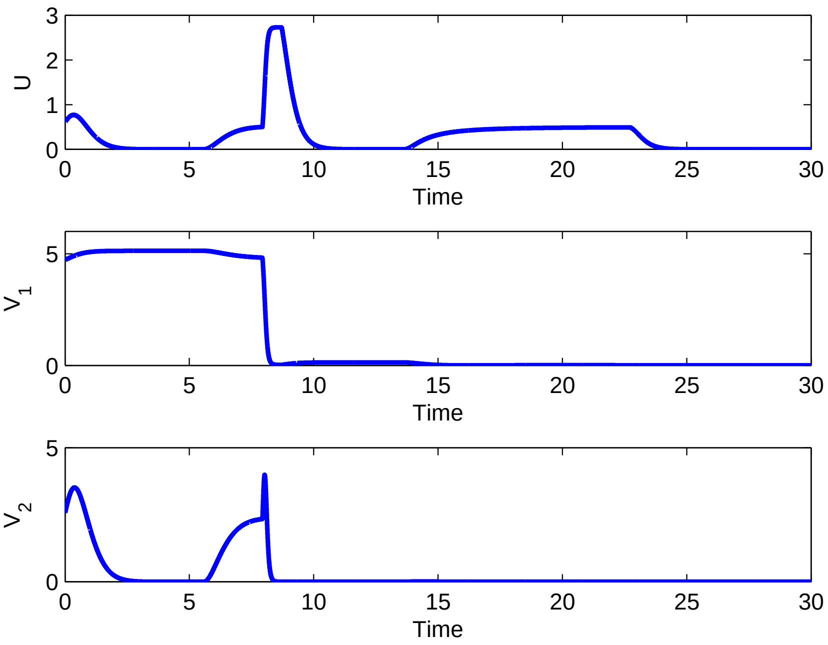

Figure 5.17: Lyapunov function results for dual quaternion example.

Fig. 5.18 provides the translation results for all three spacecraft together on a 3D x, y, z plot.

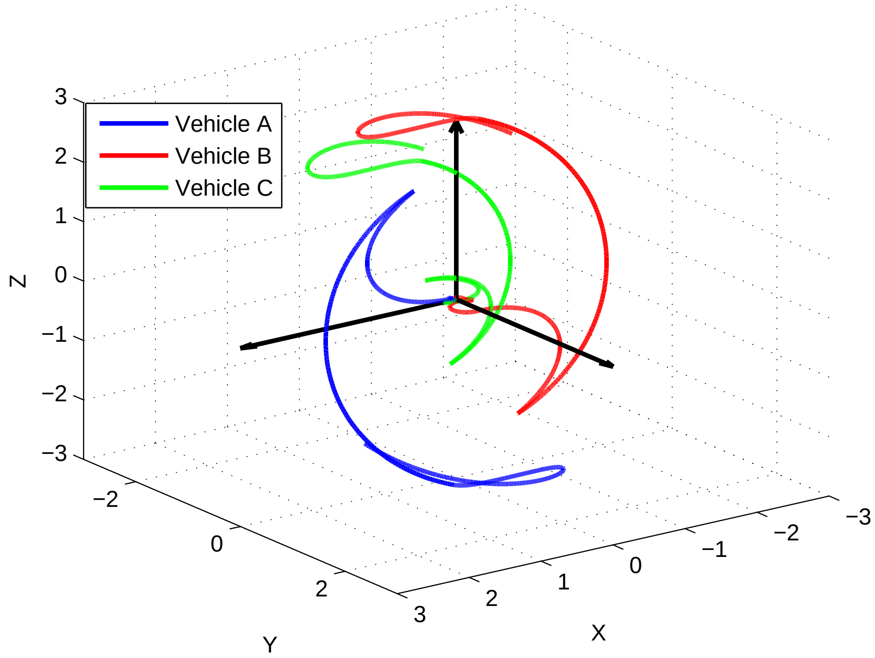

Figure 5.18: 3D translational results for all three spacecraft.

Chapter 6

Conclusion

The current work presents the study and feedback design for dynamical systems having dual quaternion representation. Mathematical methods utilizing dual quaternions allow extremely compact (and thus computationally efficient) description of complex systems in the aerospace and robotics area. From the control theory point of view the dissertation focuses on achieving closed loop stability under unknown control direction. Such type of model uncertainty represents serious challenges for conventional linear and nonlinear control. The main methodology of the suggested approach is based on sliding mode control with multiple equilibrium manifolds extended to dual quaternion systems. Control algorithms based on this method allow design of universal algorithms for a class of mathematical models where the control matrix B(t,x) is completely unknown or only partial information about this matrix is available.

Also in this work, nonholonomic control strategies for driftless nonlinear systems were studied. In particular, methods for augmenting the general class of driftless nonlinear systems which do not already satisfy the Brockett conditions for transformation to a canonical form were developed by the addition of new states. As a result, control algorithms that have been previous developed for nonholonomic systems meeting Brockett conditions can now be applied to an even larger class of nonlinear driftless systems. In this work two such control techniques, nonholonomic integrator and Bloch-Drakunov-Kinyon feedback algorithm, are used as examples. Specifically a nonholonomic integrator control algorithm is used for kinematic attitude control of a underactuated quaternion system. Also the Bloch-Drakunov-Kinyon feedback algorithm is used for the example of multiple unicycles as well as the dual quaternion example of multiple underactuated spacecraft. Both of these examples are nonholonomic systems which did not originally satisfy Brockett conditions. Numerous simulation experiments are presented that confirm and support the theoretical results of stability and robustness of the developed feedback algorithms.

In summary, the main results of the dissertation are:

- Extending the results of dual quaternion systems to systems described by multiple dual quaternions and dual vectors;
- Developing control methodology based on sliding mode control to dual quaternion systems;
- Designing specific control laws for dual quaternion systems in presence of uncertainty in the control matrix;
- Extending control Bloch-Drakunov-Kinyon feedback algorithm for nonholonomic systems to systems not satisfying Brocket conditions;
- Designing quaternion/dual quaternion based control for nonholonomic systems.

Future research on this work may include the following:

- extending the unknown control direction sliding mode control techniques developed for dual quaternion systems to n-dual quaternion systems,

- extending the Bloch-Drakunov-Kinyon feedback algorithm developed for dual quaternion systems to include the system dynamics as well as tracking,
- apply sliding mode control techniques to the Bloch-Drakunov-Kinyon feedback algorithm to compensate for various disturbances and/or uncertainties.

Appendix A

Matlab Code

The following code presents how control algorithms were implemented for the various numerical examples presented in this work. The code used to create the various plots have not been included for brevity.

A.1 Self Reconfigurable Control MATLAB Files

A.1.1 Planar Example

Code used in section 4.3.1.

selfreconfig2d.m

```
clc; clear all; close all;

%Initial Conditions
x = .5;
y = -1;
theta = 5*pi/4;
Vx = 0;
Vy = 0;
```

```
w = 0;

%Sim parameters
tmax = 120;
t = 0;
dt = .004;

%Control gains
global kx ky kt mu Uk C
kx = .12;
ky = .12;
kt = .6;
mu = .015;
Uk = diag(ones(3,1))*.1;
C = .5;
INT = 0;

X = [x Vx y Vy theta w INT]';
while t(end)<tmax
    Xi=rk4('ecc1eq',t(end),X(:,end),dt);
    t = [t;t(end)+dt];
    X = [X Xi];
end
```

self2deq.m

```
function[Xd]=ecc1eq(t,X)
global kx ky kt C mu Uk normsig

x = X(1);
Vx = X(2);
y = X(3);
Vy = X(4);
theta = X(5);
w = X(6);
INT = X(7);

dx = Vx;
dy = Vy;
dtheta = w;
```

```
%Sliding Manifold
sig1 = kx*l*x + Vx;
sig2 = ky*l*y + Vy;
sig3 = kt*l*theta + w;
sigma = [sig1 sig2 sig3]';
normsig = norm(sigma);

%Control Algorithm
dINT = mu*normsig;
DT = C - INT;
U = Uk*sign(sin(pi*normsig/DT))*sign(sigma);

%Unknown control direction matrix
B = [cos(theta) sin(theta) 0;...
    -sin(theta) cos(theta) 0;...
    0 0 1];

F=B*U;

%System kinematics
ddx = F(1);
ddy = F(2);
dw = F(3);

Xd=[dx;ddx;dy;ddy;dtheta;dw;dINT];
```

A.1.2 Dual Quaternion Example

Code used in section 4.3.2.

selfquat.m

```
clc; clear all; close all;
global kt kp mu Uk C J iJ l m B

%Initial Conditions
PARA.ATTI(1) = 45;              %Yaw (degs)
PARA.ATTI(2) = -145;             %Roll (degs)
PARA.ATTI(3) = 30;              %Pitch (degs)
PARA.ATTI=PARA.ATTI*pi/180;     %Convert to radians

s1              = sin(PARA.ATTI(1)/2);     %Conv to quaternion
s2              = sin(PARA.ATTI(2)/2);
s3              = sin(PARA.ATTI(3)/2);
c1              = cos(PARA.ATTI(1)/2);
c2              = cos(PARA.ATTI(2)/2);
c3              = cos(PARA.ATTI(3)/2);
q              = [c1*c2*c3+s1*s2*s3;...
                   s1*c2*c3-c1*s2*s3;...
                   c1*s2*c3+s1*c2*s3;...
                   c1*c2*s3-s1*s2*c3];

Q = [q(1) -q(2) -q(3) -q(4);...
    q(2) q(1) -q(4) q(3);...
    q(3) q(4) q(1) -q(2);...
    q(4) -q(3) q(2) q(1)];

p = 2*[2.5; 2; -1];
qp = .5*Q*[0;p];
DQ = [q;qp];

DW = [0; 0; 0; 0; 0; 0];


J = diag([200 300 400]);
iJ = inv(J);
m = 100;

%Sim parameters
tmax = 100;
t = 0;
dt = .008;

qrand=[.7,0,.2,0];
qrand=qrand/norm(qrand);
B=quat2dcm(qrand);
B=[B zeros(3);zeros(3) B];
%B=-1;
%Control gains
```

```
kt = .13;
kp = .10;
mu = .02;
Uk = diag([ones(3,1);ones(3,1)])*7;
C = .35;
INT = 0;
l = sign(q(1));

%System DQ's
X = [DQ; DW; INT];
while t(end)<tmax
    Xi=rk4('self3eq',t(end),X(:,end),dt);
    t = [t;t(end)+dt];
    X = [X Xi];
end

%Calcculation of position and sigma results
for k=1:length(t)
    q = X(1:4,k);
    qp = X(5:8,k);
    QC = [q(1)  q(2)  q(3)  q(4);...
         -q(2)  q(1)  q(4) -q(3);...
         -q(3) -q(4)  q(1)  q(2);...
         -q(4)  q(3) -q(2)  q(1)];
    P(:,k) = 2*QC*qp;
    s = acos(l*q(1));
if (1e6-round(abs(q(1))*1e6))==0
    logq = [0;0;0];
else
    logq = l*s*q(2:4)*(1-q(1)^2)^-.5;
end

logqp = P(2:4,k);
sigq(:,k) = X(9:11,k) + 2*kt*logq;
normsigq(k) = norm(sigq(:,k));
sigp(:,k) = X(12:14,k) + kp*logqp;
normsigp(k) = norm(sigp(:,k));
normsig(k) = norm([sigq(:,k), sigp(:,k)]);
end
```

selfdqeq.m

```
function[Xd]=self3eq(t,X)
global kt kp l C mu Uk normsig J iJ m B

q = X(1:4);
qp = X(5:8);
w = X(9:11);
v = X(12:14);
DW = X(9:14);
INT = X(15);

Q = [q(1) -q(2) -q(3) -q(4);...
     q(2)  q(1) -q(4)  q(3);...
     q(3)  q(4)  q(1) -q(2);...
     q(4) -q(3)  q(2)  q(1)];
QP = [qp(1) -qp(2) -qp(3) -qp(4);...
      qp(2)  qp(1) -qp(4)  qp(3);...
      qp(3)  qp(4)  qp(1) -qp(2);...
      qp(4) -qp(3)  qp(2)  qp(1)];
QC = [q(1)  q(2)  q(3)  q(4);...
     -q(2)  q(1)  q(4) -q(3);...
     -q(3) -q(4)  q(1)  q(2);...
     -q(4)  q(3) -q(2)  q(1)];


s = acos(l*q(1));
if (1e6-round(abs(q(1))*1e6))==0
    logq = [0;0;0];
else
    logq = l*s*q(2:4)*(1-q(1)^2)^-.5;
end

sp = QC*qp;

logqp = sp(2:4);
logDQ = [kp*logqp;kt*logq];

%Sliding surface
sig = [v;w] + 2*logDQ;
```

```
normsig = norm(sig);

DT = C - INT;
U = -Uk*sign(sin(pi*normsig/DT))*sign(sig);
dINT = mu*normsig;

iM = [zeros(3) iJ;(1/m)*eye(3) zeros(3)];
M = [zeros(3) m*eye(3);J zeros(3)];

DQ = [Q zeros(4);QP Q];
cw = [0 -w(3) w(2);...
      w(3) 0 -w(1);...
     -w(2) w(1) 0];
cv = [0 -v(3) v(2);...
      v(3) 0 -v(1);...
     -v(2) v(1) 0];
cDW=[cw zeros(3);cv cw];


%DQ Kinematics
dDQ = .5*DQ*[0;DW(1:3);0;DW(4:6)];
%DQ Dynamics
dDW = -iM*cDW*M*DW+iM*B*U;

Xd=[dDQ;dDW;dINT];
```

A.2 Nonholonomic Driftless System MATLAB Files

A.2.1 Nonholonomic Feedback to Quaternion System

Code used in section 5.2.1.

nonholq.m

```
clear all;clc;close all

%Random initial conditons
qi=[randn randn randn randn]’;
qi=qi/norm(qi);
```

```
%Simulation parameters

dt=.001;
t(1)=0;
tmax=20;

q(:,1)=qi;
while t(end)<tmax
    Xi=rk4('nonholqfun',t(end),q(:,end),dt);
    t=[t;t(end)+dt];
    q=[q Xi];
end
 plot(t,q,'LineWidth',2)
 axis([0 t(end) -1.1 1.1])
 legend('q_0','q_1','q_2','q_3')
 xlabel('Time')
 ylabel('$\mathbf{q}$','Interpreter','latex')
```

nonholfunq.m

```
function[dX]=nonholqfun(t,X)

q=X;

%Control gains
a0=-1;
b0=-1;

%Sliding surfaces
a=a0*sign(q(2)^2+q(3)^2-abs(q(4)));
b=b0*sign(q(4));
d=q(1)^2+q(4)^2;
if d<.0001
    d=.0001;
end


%Control algorithm
k1=(a*q(1)+b*q(4))/d;
k2=(a*q(4)-b*q(1))/d;
```

```
w=[0 0 0 0]';
w(2)=k1*q(2)+k2*q(3);
w(3)=k1*q(3)-k2*q(2);

Q=[q(1) -q(2) -q(3) -q(4);...
    q(2) q(1) -q(4) q(3);...
    q(3) q(4) q(1) -q(2);...
    q(4) -q(3) q(2) q(1)];

%System kinematics
dq=.5*Q*w;

dX=dq;
```

A.2.2 Unicycle Examples

Code used in section 5.3.1.
multiunicycleregulation.m

```
clear all;clc;close all

N=4;            %Number of unicycles

T=90;           %Simulation runtime
dt=0.01;        %Time intervals
eball=0.001;    %Error tolerance


%Variable Declarations
s(1:2*N) = zeros;
x(1:N+1,1) = zeros;
Y(N+1,N+1,1) = zeros;
u(1:N+1,1)=zeros;


%Starting heading
phi=2*pi*rand;
x(1,1)=phi;

%Randomizes starting positions and converts to translated system
```

```
for i=1:N
    s(2*i-1)=3*randn;
    s(2*i)=3*randn;
    x(i+1,1)=cos(phi)*s(2*i-1)+sin(phi)*s(2*i);
    Y(1,i+1,1)=(phi*cos(phi)-2*sin(phi))*s(2*i-1)+(phi*sin(phi)...
        ...+2*cos(phi))*s(2*i);
    Y(i+1,1,1)=-Y(1,i+1,1);
end


%Starting values
COUNT=0;
t(1)=0;
U(1)=0.5*(x'*x);
V1(1)=0.5*trace(Y'*Y);
V2(1)=0.5*x'*(Y'*Y)*x;
STEP=1;

for k=1:T/dt,

    t(k+1)=t(k)+dt;
    %control algorithm
    switch STEP
        case 1          %Step 1A of control algorithm
            normx=norm(x(:,k));
            alpha=1;
            beta=0;
            u(:,k)=-alpha*x(:,k);
            if normx<eball
                [v,d]=eigs(Y(:,:,k)'*Y(:,:,k));
                xe=v(:,1);
                STEP=2;
                COUNT=COUNT+1;
            end
        case 2          %Step 1B of control algorithm
            norme=norm(x(:,k)-xe);
            alpha=1;
            beta=0;
            u(:,k)=-alpha*(x(:,k)-xe);
            if norme<eball
```

```
                STEP=3;
            end
        case 3          %Step 2&3 of control algorithm
            alpha=0;
            beta=1/N;
            u(:,k)=beta*Y(:,:,k)*x(:,k);
            if V2(k)<eball/N
                STEP=1;
            end
            if V1(k)<eball
                STEP=4;
            end
        case 4          %Step 4 of control algorithm
            normx=norm(x(:,k));
            alpha=1;
            beta=0;
            u(:,k)=-alpha*x(:,k);
    end


    %System Kinematics
    x(:,k+1)=x(:,k)+dt*u(:,k);
    Y(:,:,k+1)=Y(:,:,k)+dt*(x(:,k)*u(:,k)'-u(:,k)*x(:,k)');
    Y(:,:,k+1)=triu(Y(:,:,k+1))-tril(Y(:,:,k+1)');
    U(k+1)=0.5*x(:,k+1)'*x(:,k+1);
    V1(k+1)=0.5*trace(Y(:,:,k+1)'*Y(:,:,k+1));
    V2(k+1)=0.5*x(:,k+1)'*Y(:,:,k+1)'*Y(:,:,k+1)*x(:,k+1);
end


%Transfers results back to original coordinates
p=x(1,:);
y(:,:)=Y(1,2:N+1,:);
for i=1:N
    S(2*i-1,:)=(cos(p)+p.*sin(p)/2).*x(i+1,:)-sin(p)/2.*y(i,:);
    S(2*i,:)=(sin(p)-p.*cos(p)/2).*x(i+1,:)+cos(p)/2.*y(i,:);
   % v(i,:) =uc(k,:)-(X(k+N,:)-X(1,:).*X(k,:))/2.*uc(1,:);
end
```

Code used in section 5.3.1.

multiunicycletracking.m

```
clear all;close all;clc


N=4;            %Number of unicycles

T=90;           %Simulation runtime
dt=0.01;        %Time intervals
eball=0.001;    %Error tolerance
t=[0:dt:T];

%Desired trajectory generation

% Heading Trajectory
A=pi/4;             %Amplitude
w=20*pi/180;        %Frequency
phid=A*sin(w*t);
wd=w*A*cos(w*t);

%Translation trajectory
V=.2;               %Velocity
xc=0;
yc=0;
xdd=V*cos(phid);
ydd=V*sin(phid);
Vd=linspace(V,V,T/dt+1);


%Variable declarations
s(1:2*N) = zeros;
x(1:N+1,1) = zeros;
Y(N+1,N+1,1) = zeros;
u(1:N+1,1)=zeros;


%Random starting heading
phi=2*pi*rand;
x(1,1)=phi;
```

```
%Formation spacing
r=1;            %radius
a=2*pi/(N);     %circular interval

%Randomize starting positions, translate starting positions and desired
%positions
for i=1:N
     s(2*i-1)=2*randn;
     s(2*i)=2*randn;
    x(i+1,1)=cos(phi)*s(2*i-1)+sin(phi)*s(2*i);
    Y(1,i+1,1)=(phi*cos(phi)-2*sin(phi))*s(2*i-1)+(phi*sin(phi)...
        ...+2*cos(phi))*s(2*i);
    Y(i+1,1,1)=-Y(1,i+1,1);
    xd(i,1)=r*sin(i*a)+xc;
    yd(i,1)=r*cos(i*a)+yc;
end


%Generate translated desired trajectory
for k=1:T/dt;
  xd(:,k+1)=xdd(k)*dt+xd(:,k);
  yd(:,k+1)=ydd(k)*dt+yd(:,k);
end
Xd(1,:)=phid;
ud(1,:)=wd;
for i=1:N
    Xd(i+1,:)=cos(phid).*xd(i,:)+sin(phid).*yd(i,:);
    Yd(1,i+1,:)=(phid.*cos(phid)-2*sin(phid)).*xd(i,:)+(phid.*sin(phid)...
        ...+2*cos(phid)).*yd(i,:);
    Yd(i+1,1,:)=-Yd(1,i+1,:);
    YD(:,:)=Yd(1,i+1,:);
    ud(i+1,:)=Vd+(YD'-Xd(1,:).*Xd(i+1,:))/2.*ud(1,:);
end


%Initialize values
YD=Yd(:,:,1);
COUNT=0;
t(1)=0;
STEP=1;
```

```
XE(:,1)=x(:,1)-Xd(:,1);
YE(:,:,1)=Y(:,:,1)-YD(:,:,1)-Xd(:,1)*x(:,1)'+x(:,1)*Xd(:,1)';
U(1)=0.5*XE(:,1)'*XE(:,1);
V1(1)=0.5*trace(YE(:,:,1)'*YE(:,:,1));
V2(1)=0.5*XE(:,1)'*YE(:,:,1)'*YE(:,:,1)*XE(:,1);

for k=1:T/dt,
    %control algorithm
    switch STEP
        case 1                  %Step 1A of control algorithm

            alpha=1;
            uh(:,k)=-alpha*XE(:,k);
            if U(k)<eball
                [v,d]=eigs(YE(:,:,k)'*YE(:,:,k));
                xe=v(:,1);
                STEP=2;
                COUNT=COUNT+1;
            end
        case 2                  %Step 2A of control algorithm
            alpha=1;
            norme=norm(XE(:,k)-xe);
            uh(:,k)=-alpha*(XE(:,k)-xe);
            if norme<eball
                STEP=3;
            end
        case 3                  %Step 2&3 of control algorithm
            beta=2;
            uh(:,k)=beta*YE(:,:,k)*XE(:,k);
            if V2(k)<eball/10
                STEP=1;
            end
            if V1(k)<eball*10
                STEP=4;
            end
        case 4                  %Step 4 of control algorithm
            alpha=1;
            beta=0;
            uh(:,k)=-alpha*XE(:,k);
            if V1(k)>eball*50
```

```
                STEP=1;
            end
    end


   %System Kinematics
    u(:,k)=uh(:,k)+ud(:,k);
    x(:,k+1)=x(:,k)+dt*u(:,k);
    Y(:,:,k+1)=Y(:,:,k)+dt*(x(:,k)*u(:,k)'-u(:,k)*x(:,k)');
    Y(:,:,k+1)=triu(Y(:,:,k+1))-tril(Y(:,:,k+1)');
    YD(:,:,k+1)=YD(:,:,k)+dt*(Xd(:,k)*ud(:,k)'-ud(:,k)*Xd(:,k)');
    YD(:,:,k+1)=triu(YD(:,:,k+1))-tril(YD(:,:,k+1)');
    XE(:,k+1)=x(:,k+1)-Xd(:,k+1);
    YE(:,:,k+1)=Y(:,:,k+1)-YD(:,:,k+1)-Xd(:,k+1)*x(:,k+1)'...
        ...+x(:,k+1)*Xd(:,k+1)';


    U(k+1)=0.5*XE(:,k+1)'*XE(:,k+1);
    V1(k+1)=0.5*trace(YE(:,:,k+1)'*YE(:,:,k+1));
    V2(k+1)=0.5*XE(:,k+1)'*YE(:,:,k+1)'*YE(:,:,k+1)*XE(:,k+1);
end

%Transfers results back to original coordinates
p=x(1,:);
y(:,:)=Y(1,2:N+1,:);
ye(:,:)=YE(1,2:N+1,:);
for i=1:N
    S(2*i-1,:)=(cos(p)+p.*sin(p)/2).*x(i+1,:)-sin(p)/2.*y(i,:);
    S(2*i,:)=(sin(p)-p.*cos(p)/2).*x(i+1,:)+cos(p)/2.*y(i,:);
   % v(i,:) =uc(k,:)-(X(k+N,:)-X(1,:).*X(k,:))/2.*uc(1,:);
end
```

A.2.3 Multiple Dual Quaternion Spacecraft Example

Code used in section 5.3.2.

nonholndq.m

```
clear all;
clc;close all
global dwu N
```

```
%Number of spacecraft
N=9;

%Random initial attitude
qi=[randn randn randn randn]';
qi=qi/norm(qi);
qi=sign(qi(1))*qi;
Qi=[qi(1) -qi(2) -qi(3) -qi(4);...
    qi(2)  qi(1) -qi(4)  qi(3);...
    qi(3)  qi(4)  qi(1) -qi(2);...
    qi(4) -qi(3)  qi(2)  qi(1)];
dqi=qi;

%Establish canonical system
x(1:2,1)=[dqi(3);dqi(4)];
Y(:,:,1)=zeros(N+2);
Yi=zeros(N+2);
Y(1,2,1)=dqi(2);
dqx=zeros(N*(N-1)/2,1);

k=0;
s=1;
for i=1:N
    pi(:,i)=[0 4*rand-2 4*rand-2 4*rand-2]';   %Randomize starting pos's
    qpi(:,i)=.5*Qi*pi(:,i);
    dqi=[dqi;qpi(:,i)];
    x(i+2,1)=dqi(4*i+2);
    Y(1:2,i+2)=[-dqi(4*i+4);dqi(4*i+3)];
    if i>1
        Y(3:i+1,i+2,1)=dqx(s:s+i-2);
        k=k+1;
        s=s+k;
    end
 end
Y=triu(Y)-triu(Y).';

%System parameters
dt=.005;
t(1)=0;
```

```
T=120;
eball=.005;



p(:,:,1)=pi;
dq(:,1)=[dqi;dqx];
COUNT=0;
Z=zeros(4);
U(1)=0.5*(x'*x);
V1(1)=0.5*trace(Y'*Y);
V2(1)=0.5*x'*(Y'*Y)*x;
STEP=1;

%Control gains
a=3;
b=3;


for k=1:T/dt,

    %control algorithm
    switch STEP
        case 1          %Step 1A of control algorithm
            normx=norm(x(:,k));
            alpha=a;
            beta=0;
            u(:,k)=-alpha*x(:,k);
            if normx<eball
                [v,d]=eigs(Y(:,:)'*Y(:,:));
                xe=v(:,1);
                STEP=2;
                COUNT=COUNT+1;
            end
        case 2          %Step 1B of control algorithm
            norme=norm(x(:,k)-xe);
            alpha=a;
            beta=0;
            u(:,k)=-alpha*(x(:,k)-xe);
            if norme<eball*5
                STEP=3;
            end
        case 3          %Step 2&3 of control algorithm
            alpha=0;
            beta=b;
            %normp=norm(p1(:,k))+norm(p2(:,k));
            u(:,k)=beta*Y(:,:)*x(:,k);
            if V2(k)<eball/10
                STEP=1;
            end
            if V1(k)<eball
                STEP=4;
            end
        case 4          %Step 4 of control algorithm
            %normx=norm(x(:,k));
            alpha=a;
            beta=0;
            u(:,k)=-alpha*x(:,k);
    end


    %System Kinematics
    dwu=u(:,k);
    Xi=rk4('nonholnedqfun',t(k),dq(:,k),dt);
    t(k+1)=t(k)+dt;
    dq(:,k+1)=Xi;
    x(1:2,k+1)=[dq(3,k+1);dq(4,k+1)];
    qc=[dq(1,k+1);-dq(2,k+1);-dq(3,k+1);-dq(4,k+1)];
    Qc=[qc(1) -qc(2) -qc(3) -qc(4);...
        qc(2)  qc(1) -qc(4)  qc(3);...
        qc(3)  qc(4)  qc(1) -qc(2);...
        qc(4) -qc(3)  qc(2 ) qc(1)];
    j=0;
    s=1;
    dqx(:,k+1)=dq(4*N+5:end,k+1);
    Yi(1,2)=dq(2,k+1);
    for i=1:N
        p(:,i,k+1)=2*Qc*dq(4*i+1:4*i+4,k+1);
        x(i+2,k+1)=dq(4*i+2,k+1);
        Yi(1:2,i+2)=[-dq(4*i+4,k+1);dq(4*i+3,k+1)];
```

```
        if i>1
            Yi(3:i+1,i+2)=dqx(s:s+i-2,k+1);
            j=j+1;
            s=s+j;
        end
    end
    Y(:,:)=triu(Yi)-triu(Yi).';

    U(k+1)=0.5*x(:,k+1)'*x(:,k+1);
    V1(k+1)=0.5*trace(Y(:,:)'*Y(:,:));
    V2(k+1)=0.5*x(:,k+1)'*Y(:,:)'*Y(:,:)*x(:,k+1);
end

q=dq(1:4,:);
```

nonholndqfun.m

```
function[dX]=nonholnedqfun(t,X)
global dwu N

dqx=X(N*4+5:end);
q=X(1:4);
Q=[q(1) -q(2) -q(3) -q(4);...
    q(2) q(1) -q(4) q(3);...
    q(3) q(4) q(1) -q(2);...
    q(4) -q(3) q(2) q(1)];

w=[0 0 dwu(1) dwu(2)]';

%attitude kinematics
ddq(1:4)=.5*Q*w;

ux=dwu(3:end);
for i=1:N
    qp=X(4*i+1:4*i+4);
    QP=[qp(1) -qp(2) -qp(3) -qp(4);...
        qp(2)  qp(1) -qp(4)  qp(3);...
        qp(3)  qp(4)  qp(1) -qp(2);...
        qp(4) -qp(3)  qp(2)  qp(1)];
    dw=[w;0;dwu(i+2);0;0];
    %Dual kinematics
    ddq(4*i+1:4*i+4)=.5*[QP Q]*dw;

    xd(i,1)=qp(2);
end

%additional constraints kinematics
Yd=xd*ux.'-ux*xd.';
k=0;
s=1;
for i=1:(N-1)
    ddqx(s:s+i-1,1)=Yd(1:i,i+1);
    k=k+1;
    s=s+k;
end

dX=[ddq';ddqx];
```

References

Ashrafiuon, H., & Erwin, R. S. (2004). Sliding control approach to underactuated multibody systems. In *Proc. american control conference.* (pp. 1283–1288). Boston, MA.

Aspragathos, N. A., & Dimitros, J. K. (1998). A comparative study of three methods for robot kinematics. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *28*(2), 135–145.

Bloch, A., & Drakunov, S. (1995). Tracking in nonholonomic dynamic systems via sliding modes. In *Proc. conf. on decision and control* (pp. 2103–2106). New Orleans, LA.

Bloch, A., & Drakunov, S. (1996). Stabilization and tracking in the nonholonomic integrator via sliding modes. *Systems & Control Letters*, *29*, 91–99. Retrieved from http://www.sciencedirect.com/science/article/pii/S0167691196000497

Bloch, A., Drakunov, S., & Kinyon, M. (1997, December). Stabilization of Brockett's generalized canonical driftless system. In *Decision and control, 1997., proceedings of the 36th ieee conference on* (Vol. 5, pp. 4260 –4265 vol.5). doi: 10.1109/CDC.1997.649505

Bloch, A. M., Drakunov, S. V., & Kinyon, M. K. (2000, January). Stabilization of Nonholonomic Systems Using Isospectral Flows. *SIAM Journal on Control and Optimization*, *38*(3), 855–874. Retrieved from http://epubs.siam.org/doi/abs/10.1137/S0363012998335607 doi: 10.1137/S0363012998335607

Brockett, R. W. (1981). *Control theory and singular Riemannian geometry* (P. J. Hilton & G. S. Young, Eds.). Springer-Verlag.

Brodsky, V., & Shoham, M. (1999, July). Dual numbers representation of rigid body dynamics. *Mechanism and Machine Theory*, *34*(5), 693–718. Retrieved from http://linkinghub.elsevier.com/retrieve/pii/S0094114X98000494 doi: 10.1016/S0094-114X(98)00049-4

Clifford, W. (1873). Preliminary Sketch of Bi-quaternions. *Proc. London Mathemat-*

ical Society, *4*, 381–395.

Coddington, E. A., & Levinson, N. (1955). *Theory of Ordinary Differential Equations.* New York: McGraw-Hill.

Corless, M., & Leitmann, G. (1981). Continuous state feedback guaranteeing uniform ultimate boundedness for uncertain dynamic systems. *IEEE Transactions on Automatic Control*, *26*(5), 1139–1144.

DeCarlo, R., Zak, S., & Drakunov, S. V. (2011). *Variable structure and sliding mode control.* In: The Control Handbook (W. S. Levine, ed.), CRC Press.

Dooley, J. R. (1991). Spatial rigid body dynamics using dual quaternion components. *Robotics and Automation, 1991. ...*(April), 90–95. Retrieved from `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=131559`

Drakunov, S. (1993). Sliding mode control of the systems with uncertain direction of control vector. *Decision and Control, 1993., Proceedings of the ...*, 6–7. Retrieved from `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=325642`

Drakunov, S. V. (1994). Sliding mode control with multiple equilibrium manifolds. In *Proceedings of me94 international congress and exposition (the winter annual meeting of asme)* (pp. 101–108). Chicago, IL.

Drakunov, S. V., & DeCarlo, R. (1994). Sliding Mode Control Design via Lyapunov Approach. In *Proceedings of 33rd ieee conference on decision and control (cdc)* (pp. 1925–1930). Orlando, FL.

Drakunov, S. V., Ozguner, U., Dix, P., & Ashrafi, B. (1995). ABS control using optimum search via sliding modes. *IEEE Transactions on Control Systems Technology*, *3*(1), 79–85.

Drakunov, S. V., & Utkin, V. (1992). Sliding Mode Control in Dynamic Systems. *International Journal of Control*, *55*, 1029–1037.

Filippov, A. F. (1988). *Differential Equations with Discontinuous Right-hand Sides.* Boston, MA: Kluwer Academic Publishers.

Flügge-Lotz, I. (1953). *Discontinuous Automatic Control.* Princeton Univeristy Press.

Funda, J., Taylor, R. H., & Paul, R. P. (1990). On homogeneous transforms, quaternions, and computational efficiency. *IEEE Transactions on Robotics and Automation*, *6*(3), 382–388.

Gutman, S. (1979, June). Uncertain dynamical systems–a {L}yapunov min-max approach. *IEEE Trans. Automat. Contr.*, *24*(3), 437–443.

Hall, B. C. (2003). *Lie Groups, Lie Algebras, and Representations.* Springer-Verlag New York, Inc.

Hamilton, W. R. (1844). On Quaternions; or on a New System of Imaginaries in Algebra. *The London, Edinburgh and Dublin Philosophical Magazine and J. Science*, *25*(3), 489–495.

Han, D., Fang, X., & Wei, Q. (2008, April). Rotation interpolation based on the geometric structure of unit quaternions. *2008 IEEE International Conference on Industrial Technology*(1), 1–6. doi: 10.1109/ICIT.2008.4608619

Han, D., Wei, Q., & Li, Z. (2008, April). A Dual-quaternion Method for Control of Spatial Rigid Body. *2008 IEEE International Conference on Networking, Sensing and Control*(5), 1–6. doi: 10.1109/ICNSC.2008.4525172

Han, D.-P., Wei, Q., & Li, Z.-X. (2008, July). Kinematic control of free rigid bodies using dual quaternions. *International Journal of Automation and Computing*, *5*(3), 319–324. Retrieved from `http://www.springerlink.com/index/10.1007/s11633-008-0319-1` doi: 10.1007/s11633-008-0319-1

Hsu, L., Oliveira, T. R., & Peixoto, A. J. (2006, June). Sliding Mode Control of Uncertain Nonlinear Systems with Arbitrary Relative Degree and Unknown Control Direction. *Int'l Workshop on Variable Structure Systems*, 178–183.

Isidori, A. (1997). *Nonlinear control systems.* Springer-Verlag New York, Inc.

Kaloust, J., & Qu, Z. (1995). Continuous robust control design for nonlinear uncertain systems without a priori knowledge of control direction. *IEEE Transactions on Automatic Control*, *40*(2), 276–282.

Khalil, H. K. (2002). *Nonlinear Systems* (3rd ed.). Upper Saddle River, NJ: Prentice Hall.

Kim, M., Kim, M., & Shin, S. (1996). A compact differential formula for the first derivative of a unit quaternion curve. *Journal of Visualization and Computer . . .*, *0*(0), 1–14.

Krishnan, H. (1992). Attitude stabilization of a rigid spacecraft using gas jet actuators operating in a failure mode. In *Decision and control, ...* Tucson, Arizona. Retrieved from `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=371454`

Krishnan, H., Reyhanoglu, M., & McClamroch, H. (1994, June). Attitude stabilization of a rigid spacecraft using two control torques: A nonlinear control approach based on the spacecraft attitude dynamics. *Automatica*, *30*(6), 1023–1027. Retrieved from `http://linkinghub.elsevier.com/retrieve/pii/0005109894901961` doi: 10.1016/0005-1098(94)90196-1

Li, K., Yuan, J., Yue, X., & Fang, Q. (2007). Autonomous navigation algorithm for spacecrafts based on dual quaternion. *Proceedings of SPIE*, *6795*, 67953K–67953K–6. Retrieved from `http://link.aip.org/link/PSISDG/v6795/i1/p67953K/s1&Agg=doi` doi: 10.1117/12.774862

Marino, R., & Tomei, P. (1993). Robust stabilization of feedback linearizable time-varying uncertain nonlinear systems. *Automatica*, *29*, 181–189.

MathsPoetry. (2009). *Diagonal rotation.* Retrieved 8 OCT 2013, from `http://en.wikipedia.org/wiki/File:Diagonal_rotation.png`

Mcvittie, G. R., Kumar, K. D., Liu, G., & Candidate, D. S. (2010). Reduced Input Control of an Underactuated Satellite Formation. In *Aiaa guidance, navigation, and control conference* (pp. 1–20). Toronto, Ontario Canada.

Medina-Garciadiego, V., & Leonessa, A. (2011). Tracking control of a nonholonomic ground vehicle. In *American control conference (acc), 2011* (pp. 1710–1713).

Murray, R. M., Li, Z., & Sastry, S. S. (1994). *A mathematical introduction to robotic manipulation.* Boca Raton, FL: CRC Press LLC.

NASA JPL. (2004). *Satellite attitude control.* Retrieved 10 Oct 2013, from `http://www.srl.caltech.edu/personnel/mseibert/galex/protected/test/htdocs-gal`

Nordkvist, N., Bullo, F., & Member, S. (2008). Control algorithms along relative equilibria of underactuated Lagrangian systems on Lie groups. *Automatic Control, IEEE Transactions on*, *53*(11), 2021–2026. Retrieved from `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4700860`

Oliveira, T. R., Peixoto, A. J., & Liu, H. (2007, July). Sliding Mode Output Tracking of Uncertain Nonlinear Systems with Unknown Control Direction. *American Control Conference*, 3831–3836. doi: 10.1109/ACC.2007.4282888

Oliveira, T. R., Peixoto, A. J., & Liu, H. (2010). Sliding Mode Control of Uncertain Multivariable Nonlinear Systems With Unknown Control Direction via Switching and Monitoring Function. *IEEE Transactions on Automatic Control*, *55*(4), 1028–1034. doi: 10.1109/TAC.2010.2041986

Pathak, K., & Agrawal, S. K. (2004). Planning and control of a nonholonomic unicycle using ring shaped local potential fields. In *American control conference, 2004. proceedings of the 2004* (Vol. 3, pp. 2368—-2373 vol.3).

Pham, H.-L., Perdereau, V., Adorno, B. V., & Fraisse, P. (2010). Position and orientation control of robot manipulators using dual quaternion feedback. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 658–663.

Price, W. D., Seo, D., Kitchen-Mckinley, S. J., & Drakunov, S. V. (2014). Stabilization of Driftless Nonlinear Systems with Nonholonomic Control Strategy. In *2014 european control conference (submitted).*

Price, W. D., Ton, C., MacKunis, W., & Drakunov, S. (2013). Self-Reconfigurable Control for Dual-Quaternion/Dual-Vector Systems. In *2013 european control conference* (pp. 860–865). Zurich, Switzerland.

Qu, Z. (1992). Global stabilization of nonlinear systems with a class of unmatched uncertainties. *Sys. Contr. Lett.*, *18*(3), 301–307.

Qu, Z. (1993). Robust control of nonlinear uncertain systems under generalized matching conditions. *Automatica*, *29*, 985–998.

Slotine, J. J. E., & Hedrick, K. (1993). Robust input-output feedback linearization. *lnt. J. Contr.*, *57*, 1133–1139.

Study, E. (1891). Von den Bewegungen und Umlegungen. *Mathematische Annalen*, *39*, 441–556.

Tsiotras, P., & Luo, J. (2000). Control of underactuated spacecraft with bounded inputs. *Automatica*, *36*, 1153–1169. Retrieved from `http://www.sciencedirect.com/science/article/pii/S000510980000025X`

Utkin, V. (1978). *Sliding Modes and Their Application in Variable Structure Systems.* Moscow: MIR.

Wang, J., Liang, H., & Sun, Z. (2012, February). Dual-quaternion-based finite-time control for spacecraft tracking in six degrees of freedom. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, *0*(0), 1–18. Retrieved from `http://pig.sagepub.com/lookup/doi/10.1177/0954410011434883` doi: 10.1177/0954410011434883

Wang, X., Han, D., Yu, C., & Zheng, Z. (2012, May). The geometric structure of unit dual quaternion with application in kinematic control. *Journal of Mathematical Analysis and Applications*, *389*(2), 1352–1364. Retrieved from `http://linkinghub.elsevier.com/retrieve/pii/S0022247X12000327` doi: 10.1016/j.jmaa.2012.01.016

Wie, B. (1998). *Space Vehicle Dynamics and Control* (J. S. Przemieniecki, Ed.). Reston, VA: AIAA.

Wu, Y., & Hu, X. (2006). Strapdown inertial navigation using dual quaternion algebra: error analysis. *Aerospace and Electronic Systems, . . .*, *42*(1), 259–266. Retrieved from `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1603421`

Wu, Y., Hu, X., & Hu, D. (2005). Strapdown inertial navigation system algorithms based on dual quaternions. *Aerospace and Electronic . . .*, *41*(1). Retrieved from `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1413751`

Xian, B., Dawson, D. M., de Queiroz, M. S., & Chen, J. (2004, July). A Continuous Asymptotic Tracking Control Strategy for Uncertain Nonlinear Systems. *{IEEE} Transactions on Automatic Control*, *49*(7), 1206–1211.

Yoshimura, Y., Matsuno, T., & Hokamoto, S. (2011). Position and Attitude Control of an Underactuated Satellite with Constant Thrust. In *Aiaa guidance, navigation, and control conference* (pp. 1–13). Portland, Oregon. Retrieved from `http://www.aric.or.kr/treatise/journal/content.asp?idx=144438`

Zenkov, D. V., Bloch, A. M., & Marsden, J. E. (2002). The {Lyapunov} Malkin theorem and stabilization of the unicycle with rider. *Systems & Control Letters*, *45*(4), 293–302. Retrieved from http://www.sciencedirect.com/science/article/pii/S0167691101001876 doi: 10.1016/S0167-6911(01)00187-6

Zhang, F., & Duan, G. (2011). Robust Integrated Translation and Rotation Finite-Time Maneuver of a Rigid Spacecraft Based on Dual Quaternion. In *Aiaa guidance, navigation, and control conference* (pp. 1–17). Portland, Oregon. Retrieved from http://www.aric.or.kr/treatise/journal/content.asp?idx=144484