2009

Control Of Nonh=holonomic Systems

Hongliang Yuan
University of Central Florida

Control of Nonholonomic Systems

by

Hongliang Yuan
B.E. University of Science & Technology of China, 2002
M.S. University of Central Florida, 2007

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
from the School of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Summer Term
2009

Major Professor: Zhihua Qu

Abstract

Many real-world electrical and mechanical systems have velocity-dependent constraints in their dynamic models. For example, car-like robots, unmanned aerial vehicles, autonomous underwater vehicles and hopping robots, etc. Most of these systems can be transformed into a chained form, which is considered as a canonical form of these nonholonomic systems. Hence, study of chained systems ensure their wide applicability. This thesis studied the problem of continuous feed-back control of the chained systems while pursuing inverse optimality and exponential convergence rates, as well as the feed-back stabilization problem under input saturation constraints. These studies are based on global singularity-free state transformations and controls are synthesized from resulting linear systems. Then, the application of optimal motion planning and dynamic tracking control of nonholonomic autonomous underwater vehicles is considered. The obtained trajectories satisfy the boundary conditions and the vehicles' kinematic model, hence it is smooth and feasible. A collision avoidance criteria is set up to handle the dynamic environments. The resulting controls are in closed forms and suitable for real-time implementations. Further, dynamic tracking controls are developed through the Lyapunov second method and back-stepping technique based on a NPS AUV II model. In what follows, the application of cooperative surveillance and formation control of a group of nonholonomic robots is investigated. A designing

scheme is proposed to achieves a rigid formation along a circular trajectory or any arbitrary trajectories. The controllers are decentralized and are able to avoid internal and external collisions. Computer simulations are provided to verify the effectiveness of these designs.

To My Family.

Acknowledgments

I am thankful to Dr. Zhihua Qu, my supervisor, for his supports on this research. I am also thankful to Dr. Jing Wang and Jian Yang for their suggestions and assistance. I also give thanks to all the graduate students in the robotics/control lab for helpful discussions. I thank my family for their patience and *love*.

TABLE OF CONTENTS

LIST OF FIGURES . . . . . **xii**

LIST OF TABLES . . . . . **xv**

CHAPTER 1 INTRODUCTION TO NONHOLONOMIC SYSTEMS . . **1**

1.1 Definition Of Nonholonomic Systems 1

1.2 Some Examples Of Nonholonomic Systems 3

1.2.1 The Unicycle or UAV Kinematic Model 3

1.2.2 Car-like Robots 4

1.2.3 Hopping Robots 6

1.2.4 The Origin of Nonholonomy 7

1.3 Canonical Forms Of Nonholonomic Systems 8

CHAPTER 2 CONTROLLABILITY OF NONHOLONOMIC SYSTEMS 11

2.1 Nonlinear Controllability Analysis Based On Lie Bracket 12

2.2 Interpretation Of Lie Brackets From Control Viewpoint 14

2.3 Controllability Of Chained Systems 16

2.4 Difficulties In Nonholonomic Controls 17

CHAPTER 3 REVIEW OF NONHOLONOMIC CONTROLS 19

3.1 Open Loop Controls 21

3.2 Discontinuous Feedback Controls 24

3.3 Time-Varying Continuous Controls 26

CHAPTER 4 SMOOTH PURE FEEDBACK STABILIZATION OF CHAINED NONHOLONOMIC SYSTEMS 28

4.1 Problem Formulation 28

4.2 Global State Scaling Transformation And Control Design Scheme 31

4.2.1 Design of Control Component u_1 31

4.2.2 A Global State Transformation 33

4.2.3 Design of Control Component u_2 35

4.3 Optimal Performance 39

4.4 Design Examples 42

4.5 Simulations And Comparisons With Other Existing Controls 43

4.6 Conclusion 47

CHAPTER 5 SATURATED CONTROL OF CHAINED NONHOLONOMIC SYSTEMS 51

5.1 Problem Formulation 53

5.2 The Saturated Control Design 54

5.2.1 The Control Design u_1 and u_2 57

5.2.2 Choice of k and d 62

5.3 Simulations 64

5.4 Conclusion 67

CHAPTER 6 OPTIMAL REAL-TIME COLLISION-FREE MOTION PLANNING FOR NONHOLONOMIC AUVS IN A 3D UNDERWATER SPACE 68

6.1 Problem Formulation 73

6.1.1 The Kinematic Model 73

6.1.2 The Trajectory Planning Problem 74

6.2 Real-Time Trajectory Planning For AUVs 77

6.2.1 Trajectory Planning without Obstacles 78

6.2.2 Trajectory Planning with Obstacles 81

6.2.3 Optimal Solution of Candidate Trajectories 83

6.2.4 Solution and Solvability 86

6.3 Simulation Results 90

6.3.1 Single Obstacle 90

6.3.2 Multiple Obstacles 92

6.4 Torque Level Tracking Control Of 3D trajectories 95

6.4.1 The Kinematic Tracking Controller 96

6.4.2 The Dynamic Tracking Control Design 100

6.4.3 Simulation Results 105

6.5 Conclusion 106

CHAPTER 7 COORDINATED EXPLORATION AND FORMATION CONTROL FOR MULTIPLE UNMANNED AERIAL VEHICLES (UAVS) . . . 109

7.1 Problem Formulation 110

7.2 Motion Planning 112

7.2.1 Parametric Feasible Trajectories 112

7.2.2 Motion Planning for Avoiding Static/Dynamic Obstacles 115

7.3 Cooperative Formation Controls 118

7.3.1 Formation Control of Multiple UAVs 119

7.3.2 Adaptive Cooperative Formation Controls 121

7.3.3 Circular Trajectories and Arbitrary Trajectories 122

7.3.4 Internal and External Collision Avoidance 124

7.4 Simulations 125

7.4.1 Simulation Settings 127

7.4.2 Simulation Results 128

7.5 Conclusion 131

CHAPTER 8 CONCLUSION AND FUTURE WORK 133

LIST OF REFERENCES 138

LIST OF FIGURES

1.1 The Unicycle Model 3

1.2 The Car-like Robot Model 5

1.3 The Hopping Robot Model 6

2.1 Lie Bracket Motion Effects 14

4.1 Simulation Results of The Proposed Controls. (a),(c) State and Control with u_2' in (4.2.15). (b),(d) State and Control with u_2 in (4.2.8). (e) Model Difference for u_2' and u_2. 49

4.2 Control Effects for Various Other Control Designs. (a),(b) State and Control of Discontinuous Control. (c),(d) States and Controls for The Ordinary Periodic Time-varying Feedback Design and ρ-exponential Stabilizer. 50

5.1 Two Cases of Controls: (a), $t_1 \leq t_d$; (b), $t_1 > t_d$. 59

5.2 State and Controls for The Case $t_1 \leq t_d$. (a), State; (b), Control. 65

5.3 State and Control for The Case $t_1 > t_d$. (a), State; (b), Control. 66

6.1 A Control Block Diagram for An Underwater Vehicle 69

6.2 One AUV Model 73

6.3 AUV Moving in an Unknown Environment 75

6.4 Solution of (b_4^k, c_4^k) 87

6.5 The Optimal Collision Free Trajectory 91

6.6 The Linear Velocity 93

6.7 Angular Velocities 93

6.8 Orientation Angles 94

6.9 The Trajectory Avoids Obstacles 94

6.10 The Simulink Platform for Dynamic Tracking Controls 106

6.11 Trajectory Tracking Simulation Results. (a), Desired and Actual Trajectory. (b), Tracking of x. (c), Tracking of y. (d), Tracking of z. (e), Tracking of ϕ. (f), Tracking of ψ. (g), Tracking of θ. (h), Torque Control. (i), Rudder Control. (j), Bow Plane Angle. (k), Stern Plane Angle. 108

7.1 Block Diagram of UAS Control Loops. 111

7.2 Control Software Modules. 112

7.3 A UAV in the Presence Obstacles 116

7.4 A Formation Defined in the Frenet Frame 120

7.5 Collision avoidance for UAV formations 124

7.6 Flow-chart of the Simulation Platform 126

7.7 Waypoints for Each UAV 127

7.8 UAVs in Searching. 129

7.9 UAVs Traveling through Waypoints with A Two-column Formation. 130

7.10 UAVs Traveling through Waypoints with A Triangular Formation. 130

7.11 UAVs Patrolling in Triangular Formations. 131

LIST OF TABLES

4.1 Summary of Various Control Approaches 47

6.1 Settings of Single-Obstacle Scenario 90

6.2 Settings of Multi-Obstacle Scenario 95

6.3 Settings of The Obstacles 96

7.1 Map Coordinates 128

7.2 Initial Configuration of UAVs 128

7.3 Static Obstacles 129

CHAPTER 1
INTRODUCTION TO NONHOLONOMIC SYSTEMS

1.1 Definition Of Nonholonomic Systems

In general, nonholonomic systems are defined to be those systems with nonintegrable constraints on their velocities. The configuration of a mechanical or electrical system can be uniquely described by an n-dimensional vector:

$$q = [q_1 \;\; q_2 \;\; \cdots \;\; q_n]^T.$$

Normally, the configuration space Q is an n-dimensional smooth manifold, locally diffeomorphic to the Euclidean space $\Re^n$. The generalized velocity at a generic point of a trajectory $q(t) \subset Q$ is described by its tangent vector:

$$\dot{q} = [\dot{q}_1 \;\; \dot{q}_2 \;\; \cdots \;\; \dot{q}_n]^T.$$

Two types of constraints may be applied to the system, one is pure geometric constraints, which can be described as:

$$h_i(q) = 0, \quad i = 1, 2, \;\; \cdots, \;\; k.$$

The other type of constraints are velocity-dependent constraints, involving generalized coordinates and their derivatives, e.g. first-order kinematic constraints are:

$$a_i(q, \dot{q}) = 0, \quad i = 1, 2, \;\; \cdots, \;\; k.$$

In most cases, the kinematic constraints are linear in the velocities, which is called affine in velocity or Praffian, hence they can be described as:

$$a_i^T(q)\dot{q} = 0, \quad i = 1, 2, \ \cdots, \ k, \ \ or \ \ A^T(q)\dot{q} = 0. \tag{1.1.1}$$

Kinematic constraints may be integrable. That is, there may be k functions h_i such that

$$\frac{\partial h_i(q(t))}{\partial q} = a_i^T(q), \quad i = 1, 2, \ \cdots, \ k. \tag{1.1.2}$$

If we put (1.1.2) into (1.1.1), we would have:

$$\frac{\partial h_i(q(t))}{\partial q}\dot{q} = 0, \quad i = 1, 2, \ \cdots, \ k,$$

which yields,

$$h_i(q) = c_i, \quad i = 1, 2, \ \cdots, \ k.$$

It shows that through integration, the kinematic constraints are reduced to pure geometric constraints. In this case, the kinematic constraints are called to be holonomic. If the solution to the partial differential equation (1.1.2) does not exists, then the kinematic constraints can not be integrated, which means these constraints are indeed imposed on generalized velocities. In this case, these constraints are nonholonomic, and the system is a nonholonomic system.

A feasible way to check whether the kinematic constraints is nonholonomic can be done as following:

Suppose $h(q) = 0$ exists, then $\frac{\partial h}{\partial q}\dot{q} = 0$. Denote

$$(\frac{\partial h}{\partial q})_{ij} = \frac{\partial h_i}{\partial q_j},$$

then we know

$$\frac{\partial(\frac{\partial h}{\partial q})_{ij}}{\partial q_k} = \frac{\partial^2 h_i}{\partial q_j \partial q_k} = \frac{\partial^2 h_i}{\partial q_k \partial q_j} = \frac{\partial(\frac{\partial h}{\partial q})_{ik}}{\partial q_j}.$$

Hence for $A^T(q)$ in equation (1.1.1), if the following property does not holds, it must be a nonholonomic system,

$$\frac{\partial A_{ij}}{\partial q_k} = \frac{\partial A_{ik}}{\partial q_j}. \tag{1.1.3}$$

1.2 Some Examples Of Nonholonomic Systems

In this section, we will elaborate some mechanical systems that have nonholonomic constraints, then we will discuss the origin of nonholonomy and present a commonly used canonical form that is used for control design.

1.2.1 The Unicycle or UAV Kinematic Model

The following figure shows a Unicycle model:

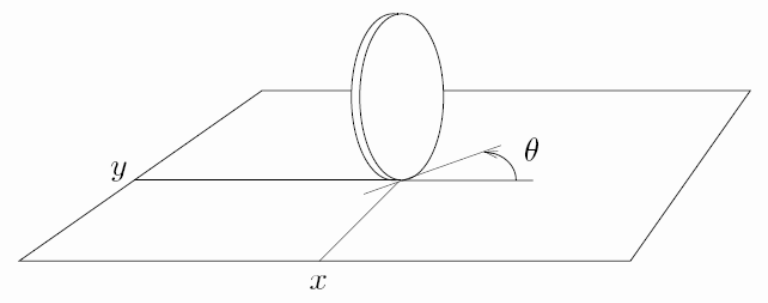


Figure 1.1: The Unicycle Model

On a plane, UAV or Unicycle share the same kinematic model, which is represented by:

$$\begin{aligned} \dot{x} &= v_1 \cos\theta \\ \dot{y} &= v_1 \sin\theta \\ \dot{\theta} &= v_2 \end{aligned} \tag{1.2.1}$$

where $q = [x, y, \theta]$ is the generalized coordinates. (x, y) is the world coordinates and θ is the heading angle. v_1 is forward linear velocity, v_2 is turning rate. The model (1.2.1) can be rewritten as:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix} v_1 + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} v_2.$$

It is straightforward to identify that $\dot{q} = [\dot{x} \;\; \dot{y} \;\; \dot{\theta}]^T$ has an one dimensional null space, which represents a kinematic constraint on the model:

$$[-\sin\theta \;\; \cos\theta \;\; 0]\dot{q} = 0,$$

where $A^T(q) = [-\sin\theta \;\; \cos\theta \;\; 0]$. It is straightforward to verify that equation (1.1.3) does not hold for this $A^T(q)$, thus the model is nonholonomic.

1.2.2 Car-like Robots

Fig. 1.2 illustrated a rear-driven car-like robot model. Its generalized coordinates is

$$q = [x, y, \theta, \phi]^T,$$

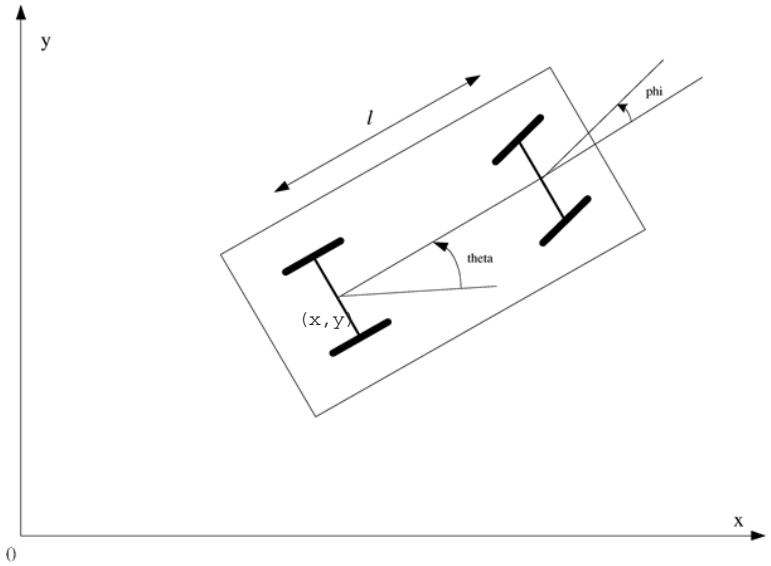


Figure 1.2: The Car-like Robot Model

where (x, y) is world coordinates, refers to the middle of the rear axle. θ is heading angle, ϕ is steering angle. v_1 is forward linear velocity, v_2 is steering rate.

The kinematic model is represented by:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \\ \frac{1}{l}\tan\phi \\ 0 \end{bmatrix} v_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} v_2.$$

The null space is 2-dimensional, represents 2 kinematic constraints, which are:

$$\begin{bmatrix} \sin\theta & -\cos\theta & 0 & 0 \\ \sin(\theta+\phi) & -\cos(\theta+\phi) & -l\cos\phi & 0 \end{bmatrix} \dot{q} = 0$$

Again, it can be determined that the kinematic constraints are nonholonomic by using equation (1.1.3). For a front-driven car-like robot, the model is similar except that the term $\frac{1}{l}\tan\phi$ is replaced by $\frac{1}{l}\sin\phi$, and it is also a nonholonomic model.

1.2.3 Hopping Robots

Fig. 1.3 shows a hopping robot model in flight phase. Its generalized coordinates is:

$$q = [\theta, l_l, \phi]^T,$$

where θ is the angle of the hip of the hopping robot with respect to its body, l_l is the length of the lower leg, ϕ is the angle of leg of the robot with respect to the horizontal axis.

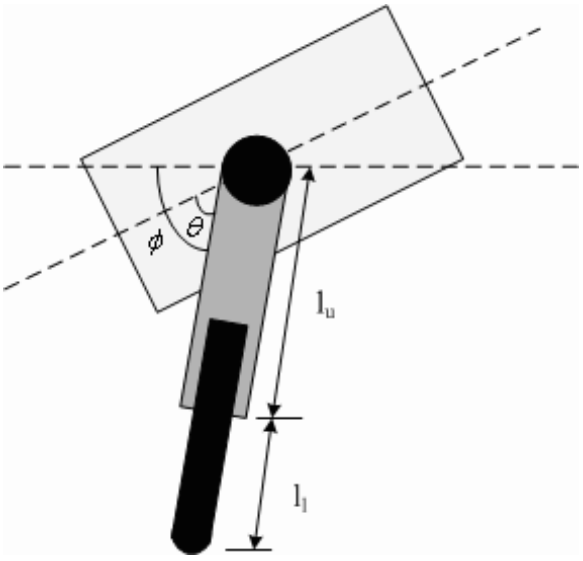


Figure 1.3: The Hopping Robot Model

The following kinematics model could be obtained:

$$\begin{bmatrix} \dot{\theta} \\ \dot{l}_l \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \frac{I}{I+m(l_u+l_l)^2} \end{bmatrix} v_1 + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} v_2,$$

where v_1 is the rate change of the angle between the body and leg, v_2 is the rate change of the length of the lower leg. I is the moment of inertia for the body, m is the mass for the leg which is concentrated at the foot, l_u is the length for the upper leg. The null space of the generalized velocity is one dimensional, means it has one kinematic constraint, which is:

$$[-I \;\; 0 \;\; I + m(l_u + l_l)^2]\dot{q} = 0,$$

and it is a nonholonomic constraint by checking equation (1.1.3).

1.2.4 The Origin of Nonholonomy

The origin of nonholonomy can be divided into two classes:

- Bodies of motion are in contact with each other and they roll/move without slippage.
- Conservation of moments in a multi-body system associated with under-actuated control.

For the Unicycle example in section 1.2.1, the nonholonomy arises because at the touching point between disk and surface, the velocity are confined to be aligned with the heading angle, no slippage is allowed. For UAV model, since the engine thrust is always aligned

with body's longitudinal direction, it can be considered approximately that there is no side slippage. For the car-like model presented in section 1.2.2, the two nonholonomic constraints arise because there are no side slippage at both front and rear wheel. For the hopping robot model presented in section 1.2.3, the nonholonomy arises because when it flies in the air, the angular moment is conserved since there is no external force applied to the system.

1.3 Canonical Forms Of Nonholonomic Systems

The existence of a canonical form for nonholonomic systems is essential for the systematic development of both open-loop and closed-loop controls. The most useful canonical form is the chained form [49]. Many practical mechanical and electrical systems can be converted into the following $(2, n)$ driftless model through diffeomorphic state and control transformations:

$$\begin{aligned} \dot{x}_1 &= u_1 \\ \dot{x}_2 &= u_2 \\ \dot{x}_3 &= x_2 u_1 \\ &\vdots \\ \dot{x}_n &= x_{n-1} u_1. \end{aligned} \tag{1.3.1}$$

More general study would involve multiple chains, which could be extended from the $(2, n)$ form.

For example, the unicycle model in section 1.2.1 can be converted into:

$$\begin{aligned}\dot{z}_1 &= u_1 \\ \dot{z}_2 &= u_2 \\ \dot{z}_3 &= z_2 u_1,\end{aligned}$$

through the transformation:

$$z_1 = x, \quad z_2 = \tan\theta, \quad z_3 = y, \quad v_1 = u_1 \sec\theta, \quad v_2 = u_2 \cos^2\theta.$$

And the car-like robot model in section 1.2.2 can be converted into:

$$\begin{aligned}\dot{z}_1 &= u_1 \\ \dot{z}_2 &= u_2 \\ \dot{z}_3 &= z_2 u_1 \\ \dot{z}_4 &= z_3 u_1,\end{aligned}$$

through the transformation:

$$\begin{aligned}z_1 &= x \\ z_2 &= \frac{\tan\phi}{l\cos^3\theta} \\ z_3 &= \tan\theta \\ z_4 &= y \\ v_1 &= \frac{u_1}{\cos\theta} \\ v_2 &= -\frac{3\sin\theta}{l\cos^2\theta}\sin^2\phi u_1 + l\cos^3\theta\cos^2\phi u_2.\end{aligned}$$

Based on chain-form, if we apply another transformation:

$$\begin{aligned}
\xi_1 &= x_1 \\
\xi_2 &= x_2 \\
\xi_3 &= -x_3 + x_1 x_2 \\
\xi_4 &= x_4 - x_1 x_3 + \frac{1}{2} x_1^2 x_2 \\
&\vdots \\
\xi_n &= (-1)^n x_n + \sum_{i=2}^{n-1} (-1)^i \frac{1}{(n-i)!} \xi_1^{n-i} \xi_i,
\end{aligned}$$

we would get another canonical form named power form, which is:

$$\begin{aligned}
\dot{\xi}_1 &= u_1 \\
\dot{\xi}_2 &= u_2 \\
\dot{\xi}_3 &= \xi_1 u_2 \\
\dot{\xi}_4 &= \frac{1}{2} \xi_1^2 u_2 \\
&\vdots \\
\dot{\xi}_n &= \frac{1}{(n-2)!} \xi_1^{n-2} u_2
\end{aligned} \tag{1.3.2}$$

Therefore, design of nonholonomic control usually starts from the canonical form (1.3.1), which deals with a class of systems, instead of for a specific physical system.

CHAPTER 2
CONTROLLABILITY OF NONHOLONOMIC SYSTEMS

For the kinematic constraints given in (1.1.1), their implications can be conveniently studied through a dual approach. That is to study the directions in which motion is permitted rather than directions in which motion is prohibited. (1.1.1) essentially implies that motion of configurations are in the null space of constraints $a_i(q),\ i = 1, \cdots, k$, i.e. a set of vector fields $g_j(q)$ can be defined such that

$$a_i^T(q)g_j(q) = 0,\ i = 1, \cdots, k,\ \ j = 1, \cdots, n-k.$$

Or in matrix form,

$$A^T(q)G(q) = 0.$$

The feasible trajectories of the systems are solutions $q(t)$:

$$\dot{q}(t) = \sum_{j=1}^{m} g_j(q)u_j = G(q)u \tag{2.0.1}$$

for some input $u(t) \in R^m,\ m = n - k$. Sometimes u is also called pseudo velocities. System (2.0.1) sometimes is also called driftless in the sense that when there is no control input, the states stay at any configuration.

2.1 Nonlinear Controllability Analysis Based On Lie Bracket

The controllability of system (2.0.1) is determined by the properties of the set of vector fields $g_j(q),\ j = 1, \cdots, m$. In order to reveal these properties, lets first introduce some concepts from differential geometry.

Definition 1. *A set of vector fields* $\{g_1, \cdots, g_m\}$ *in* $\Re^n$ *is said to be linearly independent if* $\alpha_1 g_1 + \cdots + \alpha_m g_m = 0$ *implies* $\alpha_1 = \alpha_2 = \cdots = \alpha_m = 0$. *The set of vector fields is linearly dependent if it is not linearly independent.*

Definition 2. *For vector fields* $f(q)$ *and* $g(q)$, *the operation of Lie Bracket is defined to be*

$$[f, g] = \frac{\partial g}{\partial q} f - \frac{\partial f}{\partial q} g.$$

It is straightforward to verify the following identities of Lie Bracket:

$$[f, g] = -[g, f], \quad (skew - symmetry)$$

$$[f, [g, h]] + [h, [f, g]] + [g, [h, f]] = 0, \quad (Jacobi\ identity)$$

Definition 3. *A set of linearly independent vector fields* $\{g_1, \cdots, g_m\}$ *in* R^n *is said to be involutive if* $\{g1, \cdots, g_m, [g_i, g_j]\}$ *is linearly dependent for any choice of* g_i *and* g_j *with* $i \neq j$.

Definition 4. *For a set of vector fields* $\{g_j(q), j = 1, \cdots, m\}$, $\Delta = span\{g_1, \cdots, g_m\}$ *is called the distribution of the set of vector fields.*

Definition 5. *For a set of vector fields* $\{g_j(q), j = 1, \cdots, m\}$, *its distribution is regular if* Δ *does not change with* q.

Definition 6. *$\overline{\Delta}$ is called the involutive closure of Δ if it is the smallest distribution containing Δ and if $f, g \in \overline{\Delta}$, then $[f, g] \in \overline{\Delta}$. $\overline{\Delta}_q$ denotes the involutive closure evaluated at a point q.*

Definition 6 implies that:

- $\overline{\Delta}$ is a Lie algebra.
- $\overline{\Delta}$ contains all linear combinations of g_1 up to g_m, their Lie Brackets, and all combinations of those as well.

The controllability of driftless system (2.0.1) is defined as:

Definition 7. *The system is controllable if for any pair of initial condition $q_0 \in \Re^n$ and final condition $q_f \in \Re^n$, there exists a $T > 0$ and $u : [0, T] \in R^m$ such that $q(0) = q_0$ and $q(T) = q_f$.*

Obviously, for system (2.0.1), if the motion is allowed in every direction of configuration space, it will be controllable. However, due to the existence of nonholonomic constraints, motion is confined in the null space of the constraints. Hence, dimension of tangent space is less than the dimension of configuration space ($m < n$). In this case, controllability depends on whether or not new linearly independent control directions can be generated through maneuvering controls along those allowed directions. Chow's theorem states that

Theorem 1. *If $\overline{\Delta}_q = \Re^n$ for all $q \in Q$, then system (2.0.1) is controllable on Q.*

A straightforward interpretation of Chow's theorem is that, for the set of vector fields $\{g_1, \cdots, g_m\}$ in system (2.0.1), if the union of its distribution and those subspace composed

of their Lie Bracket has same dimension as its configuration space at all points in the configuration space, then it is controllable. It implies that the Lie Bracket of the motions on allowed directions contribute to the motion on those restricted directions, through which controllability can be recovered. Hence the Lie Bracket operation is important in determine the controllability of driftless systems.

2.2 Interpretation Of Lie Brackets From Control Viewpoint

A useful interpretation of the effect of Lie Bracket operation on two vector fields is illustrated in Fig. 2.1.

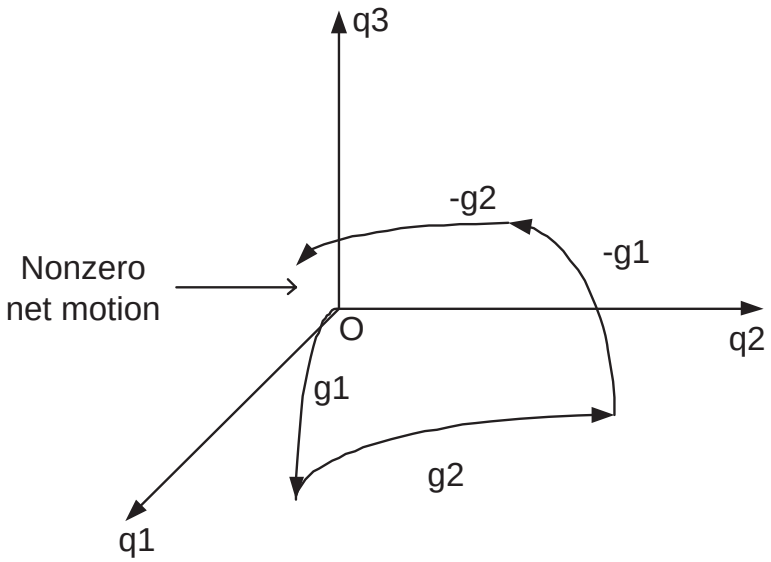


Figure 2.1: Lie Bracket Motion Effects

Let $\phi_t^f : U \to \Re^n$ denote the flow of a vector field f for time t, and consider the sequence of flows depicted in Fig. 2.1. The resulting solution is:

$$q(4\epsilon) = \phi_\epsilon^{-g_2} \circ \phi_\epsilon^{-g_1} \circ \phi_\epsilon^{g_2} \circ \phi_\epsilon^{g_1}$$

It follows that

$$\begin{aligned} q(\epsilon) &= \phi_\epsilon^{g_1} \\ &= q(0) + \epsilon\dot{q}(0) + \frac{1}{2}\epsilon^2\ddot{x}(0) + O(\epsilon^3) \\ &= q(0) + \epsilon g_1(0) + \frac{1}{2}\epsilon^2 \frac{\partial g_1}{\partial q}|_{x=0} + O(\epsilon^3). \end{aligned}$$

Similarly,

$$\begin{aligned} q(2\epsilon) &= q(\epsilon) + \epsilon\dot{q}(\epsilon) + \frac{1}{2}\epsilon^2\ddot{q}(\epsilon) + O(\epsilon^3) \\ &= q(0) + \epsilon g_1(0) + \frac{1}{2}\epsilon^2 \frac{\partial g_1}{\partial q}|_{q=0} + \epsilon g_2(q(\epsilon)) + \frac{1}{2}\epsilon^2 \frac{\partial g_2}{\partial q}|_{q=q(\epsilon)} + O(\epsilon^3) \end{aligned}$$

On the other hand, by Taylor's expansion, we have

$$\begin{aligned} g_2(q(\epsilon)) &= g_2(q(0)) + \frac{\partial g_2}{\partial q}|_{q=q(0)} \times [\epsilon g_1(0) + \frac{1}{2}\epsilon^2 \frac{\partial g_1}{\partial q}|_{q=q(0)}] + O(\epsilon^3) \\ &= g_2(q(0)) + \epsilon g_1(0) \frac{\partial g_2}{\partial q}|_{q=q(0)} + O(\epsilon^3). \end{aligned}$$

Similarly, $\frac{\partial g_2}{\partial q}|_{q=q(\epsilon)}$ can be expanded, hence,

$$q(2\epsilon) = q(0) + \epsilon(g_1(0) + g_2(0)) + \frac{1}{2}\epsilon^2[\frac{\partial g_1}{\partial q}g_1(0) + \frac{\partial g_2}{\partial q}g_2(0) + 2\frac{\partial g_2}{\partial q}g_1(0)] + O(\epsilon^3).$$

Accordingly, we have

$$q(3\epsilon) = q(0) + \epsilon g_2(0) + \frac{\epsilon^2}{2}[\frac{\partial g_2}{\partial q}g_2(0) + 2\frac{\partial g_2}{\partial q}g_1(0) - 2\frac{\partial g_1}{\partial q}g_2(0)] + O(\epsilon^3)$$

$$
\begin{aligned}
q(4\epsilon) &= q(0) + \epsilon^2[\frac{\partial g_2}{\partial q} g_1(0) - \frac{\partial g_1}{\partial q} g_2(0)] + O(\epsilon^3) \\
&= q(0) + \epsilon^2 [g_1, g_2]|_{q=q(0)} + O(\epsilon^3) \qquad (2.2.1)
\end{aligned}
$$

Equation (2.2.1) shows that after the series of motion described by $\phi_\epsilon^{-g_2} \circ \phi_\epsilon^{-g_1} \circ \phi_\epsilon^{g_2} \circ \phi_\epsilon^{g_1}$, the net outcome of the configuration $q(t)$ is moving along a direction that is obtained by Lie Bracket operation of allowed direction g_1 and g_2. This implies that the Lie Bracket of any pair of allowed moving direction potentially contributes to the reachable space as long as the generated vector fields are linearly independent to those of already exist. Hence, if the involutive closure of system (2.0.1) has same dimension as its configuration space, the system is controllable.

2.3 Controllability Of Chained Systems

From the discussion of previous section, the controllability of chained nonholonomic system can be studied as follows. The results can be extended to m input nonholonomic system. (1.3.1) can be rewritten as:

$$\dot{x} = g_1(x)u_1 + g_2(x)u_2$$

where $g_1 = [1\ 0\ x_2\ \cdots\ x_{n-1}]^T$ and $g_2 = [0\ 1\ 0\ \cdots\ 0]^T$. Denote repeated Lie brackets as $ad_{g_1}^k g_2 = [g_1, ad_{g_1}^{k-1} g_2]$, one has

$$
\begin{aligned}
ad_{g_1} g_2 &= [g_1\ g_2] \\
ad_{g_1}^k g_2 &= [g_1, ad_{g_1}^{k-1} g_2] \\
&= [0\ \cdots\ (-1)_k\ \cdots\ 0]^T, \quad k = 2, 3, \cdots, n-2
\end{aligned}
$$

where subscription k denotes kth entry in the vector field. It shows that the involutive inclosure of the distribution $\overline{\Delta} = \{g_1, g_2, \cdots, ad_{g_1}^k g_2, \cdots\},\ k = 1, 2, \cdots, n-2.$ has dimension n which is identical to the configuration space, hence the chained nonholonomic system is nonlinearly controllable, and the degree of nonholonomy is $n - 1$.

2.4 Difficulties In Nonholonomic Controls

In Section 4.1, we have mentioned the difficulties in designing feedback controls of chained nonholonomic systems. After discussing its nonlinear controllability and defining the math tools of Lie brackets, we are able to review these difficulties in a more detailed fashion with a mathematical viewpoint.

There exists no continuous control $u = u(x)$ that stabilizes the chained system. A necessary condition for such a control exist was given in [9]. That is, if the system $\dot{x} = f(x, u)$ is locally asymptotically C^1-stabilizable at x_e, then the image of map $f : M \times U \rightarrow \Re^n$ should contain some neighborhood of x_e. For the chained system, $\mathbf{0}$ is the equilibrium point. Let e_0 be a nonzero vector linearly independent from $g_1(0), g_2(0), \cdots, g_m(0)$. By continuity, there is an $\epsilon > 0$ such that for all $(x, u_1, \cdots, u_m)$ with $\|x\| < \epsilon$, the vector $\sum_{k=1}^{m} u_k g_k(x)$ is different from λe_0 for any λ in $\Re$. Therefore the map

$$(x, u_1, \cdots, u_m) \rightarrow \sum_{k=1}^{m} u_k g_k(x)$$

does not map the neighborhood $[-\epsilon, \epsilon]^{n+m}$ of $\mathbf{0}$ in $\Re^{n+m}$ into a neighborhood of $\mathbf{0}$ in $\Re^n$. Hence the necessary condition is violated, therefor a static feedback control does not exist

for the chained system. The chained system is not feedback linearizable. Suppose such a linearization exists then we would have the results $\dot{z} = Az+Bv$, where $z = T_1(x)$, $v = T_2(x, u)$ with T_1 a diffeomorphism state transformation. However since the original system is driftless, so must be the transformed system, therefor we have $A = 0$. And since the states are of higher dimension than the input u, one would not be able to find a transformation T_2 results in a constant matrix B. Pointwise linearization is not applicable as well, because at the origin, the system is not linearly controllable.

CHAPTER 3
REVIEW OF NONHOLONOMIC CONTROLS

In this chapter, we will brief review the evolution of the nonholonomic control problem. Controls of chained systems and more general nonholonomic systems are very active fields of research in the last decades. The topics are on motion planning, tracking and stabilizing. Both open-loop and close-loop approach are developed. The motion planning problem was introduced by [37], who proved that a car-like robot with one nonholonomic constraint is controllable. Open loop planners for low-dimensional mobile robots have been proposed in [38, 5, 39]. Other open-loop strategies have explored control theoretic approaches using differential geometry tools. Sinusoids were proposed by [49] to stabilize in open-loop nonholonomic system on canonical forms. Later on, the sinusoids methods were generalized through a given level of Lie-brackets of the input vectors in [34, 35, 22, 47]. [22] also proposed other open-loop controls such as piece-wise constant inputs and polynomial inputs.

In early as 1980's, feedback linearization technique has been prevailing. Sufficient and necessary conditions for exact feedback linearization of large classes of affine nonlinear systems were explicitly set up by using of differential geometry methods [21, 52]. Later on, the renewed interests on Lyapunov methods become dominant with the invention of the notion of control Lyapunov function and recursive designs such as backstepping [27, 32] in order to deal with more large classes of nonlinear systems with unmatched and/or generalized matched uncertainties [55]. While those conventional nonlinear control designs are

broadly applicable, there exist some classes of inherently nonlinear systems, such as nonlinear systems with uncontrollable linearization [3], which do not admit any smooth (or even continuous) pure state feedback controls as observed in the seminal chapter [9]. Therefore make the standard feedback linearization technique and Lyapunov direct method no longer straightforwardly applicable. Such a typical class of systems is the nonholonomic systems [30], which is not feedback linearizable and their feedback stabilization problem is challenging due to Brockett's necessary condition [9].

It is well known that chained systems are canonical forms of many nonholonomic mechanical and electrical systems such as autonomous underwater vehicles, car-like mobile robots, unmanned aerial vehicles and hopping robots, which can be transformed into the chained form by state and input transformations. Apparently, chained system does not satisfy Brockett's necessary condition, discontinuous or time-varying feedback controls have to be sought for its stabilization. During the past decades, extensive studies have been performed and a great deal of solutions have been obtained following the lines of using discontinuous control method and time-varying control method [30]. In general, discontinuous controls can render exponential stability [6, 24, 42, 43], while time-varying controls lead to asymptotic stability [54, 58, 64, 69]. More recent study has also seen the results of $\rho-$exponential stability of chained system using time-varying homogeneous feedback controls [46]. While the existing controls provide elegant solutions, there is still a desire of searching global singular-free transformations that map the chained systems into controllable linear systems. The motivation comes from the simple discontinuous controls proposed in [2, 24, 42] in which σ-process

based state scaling transformation is used. In such a method, a state scaling transformation

$$\xi_i = \frac{z_i}{x_1^{n-i}}, \quad 1 \leq i \leq n-1$$

is defined on a non-singular subspace $\Omega = \{x \in \Re^n : \ x_1 \neq 0\}$. The obvious shortcoming is that the resulting controls are discontinuous by nature, and a switching control is required to keep the state from the singularity hyperplane of $x_1 = 0$. Improvements were made in [36, 70], in which dynamic extension for control component u_1 was introduced to bypass the possible singularity due to singular initial conditions. The proposed methods are quasi-smooth and achieve quasi-exponential stability.

3.1 Open Loop Controls

The open loop control strategies include sinusoidal inputs, piecewise constants, and polynomial inputs. The basic idea of sinusoidal input is to steering every state one by one using sinusoids. The steps for sinusoidal control are listed as following: 1. Find u_1 such that x_1 goes from $x_1(t_0)$ to $x_1(t_f)$.

2. Find u_2 such that x_2 goes from $x_2(t_0)$ to $x_2(t_f)$.

3. Choose $u_1 = \alpha_1 \sin \omega t$ and $u_2 = \beta_1 \cos \omega t$ to steer x_3 from $x_3(t_0)$ to $x_3(t_f)$.

4. Choose $u_1 = \alpha_2 \sin \omega t$ and $u_2 = \beta_2 \cos 2\omega t$ to steer x_4 from $x_4(t_0)$ to $x_4(t_f)$.

$\vdots$ n. Choose $u_1 = \alpha_{n-2} \sin(\omega t)$ and $u_2 = \beta_{n-2} \cos(n-2)\omega t$ to steer x_n from $x_n(t_0)$ to $x_n(t_f)$.

Proof. Step 1 and step 2 is quite straightforward, one can simply pick $u_1 = \frac{x_1(t_f)-x_1(t_0)}{t_f-t_0}$ and $u_2 = \frac{x_2(t_f)-x_2(t_0)}{t_f-t_0}$. For step 3, it follows that,

$$\begin{aligned} x_1(t) &= x_1(t_0) + \int_{t_0}^{t} \alpha_1 \sin\omega\tau d\tau \\ &= x_1(t_0) + \frac{\alpha_1}{\omega}(\cos\omega t_0 - \cos\omega t) \\ &= x_1(0) + \frac{\alpha_1}{\omega}(1 - \cos\omega t), \end{aligned}$$

without losing generality, we put $t_0 = 0$ here. Similarly,

$$\begin{aligned} x_2(t) &= x_2(0) + \int_{0}^{t} \beta_1 \sin\omega\tau d\tau \\ &= x_2(0) + \frac{\beta_1}{\omega}\sin\omega t \end{aligned}$$

It indicates that at $t = k\frac{2\pi}{\omega},\ k = 1, 2, \cdots$, $x_1(t) = x_1(0)$ and $x_2(t) = x_2(0)$, which means x_1 and x_2 won't be changed. It follows that:

$$\dot{x}_3 = u_1 x_2 = [x_2(0) + \frac{\beta_1}{\omega}\sin\omega t]\alpha_1 \sin\omega t.$$

Hence at $t = k\frac{2\pi}{\omega}$,

$$x_3(t) = x_3(0) + \frac{k\alpha_1\beta_1}{\omega^2}.$$

This shows that while x_1 and x_2 were unchanged, x_3 can be moved to any desired value by adjust α_1, β_1.

Similarly, for step 4 to step n, the input would be:

$$u_1 = \alpha_{l-2}\sin\omega t,\ \ u_2 = \beta_{l-2}\cos(l-2)\omega t,\ \ l = 4, 5, \cdots, n.$$

At $t = k\frac{2\pi}{\omega}$, we have:

$$\begin{aligned} x_1(t) &= x_1(0) \\ &\vdots \\ x_{l-1}(t) &= x_{l-1}(0) \\ x_l(t) &= x_l(0) + \frac{k\alpha_{l-2}^{l-2}\beta_{l-2}}{(l-2)!(2\omega)^{l-2}}. \end{aligned}$$

It shows that only x_l is steered while others are intact. Hence after n steps, all states are steered. □

In the piecewise constant inputs approach, the total maneuvering time T is equally divided into subintervals. The length of each subinterval is δ, in which constant inputs are applied.

$$\begin{aligned} u_1(\tau) &= u_{1,k} \\ u_2(\tau) &= u_{2,k} \end{aligned}$$

for $\tau \in [(k-1)\delta, k\delta)$. Without losing generality, u_1 can be chosen as a constant such that $u_1 = \frac{x_{1f}-x_{10}}{T}$. Divide the total time T into $n-1$ subintervals such that $T = (n-1)\delta$. Assign the $n-1$ constant values of input u_2 as:

$$u_{2,1},\ u_{2,2},\ \cdots,\ u_{2,n-1},$$

which can be solved from a set of linear algebraic equations resulting from the integration of the model equations with u_2 applied ($n-1$ variables, $n-1$ equations).

The approach of polynomial inputs is similar to the approach of piecewise constant inputs, but with improved smoothness properties. The control are chosen as:

$$\begin{aligned} u_1 &= \text{sign}(x_{1f} - x_{10}) \\ u_2 &= c_0 + c_1 t + \cdots + c_{n-2} t^{n-2}, \end{aligned}$$

where $T = x_{1f} - x_{10}$. The coefficients $c_0, \cdots, c_{n-2}$ can be obtained by solving the set of linear algebraic equations resulting from the closed-form integration of the model equations

$$M(T) \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-2} \end{pmatrix} + m(x_i, T) = \begin{pmatrix} x_{2f} \\ x_{3f} \\ \vdots \\ x_{nf} \end{pmatrix}$$

Because polynomial inputs are smooth, they are more favorable than piecewise constant inputs. The reason is that the kinematic models are controlled at velocity levels, and if a torque level control is sought, the control signal needs to be differentiated one more time.

3.2 Discontinuous Feedback Controls

Among the methods to stabilize the chained systems, discontinuous controls are more straightforward than their time-varying counterparts. However, these approaches have a singular manifold because of the transformation they incorporate. The key idea is to switch control laws after system states leave the singular manifold, hence it avoids the difficulty to design a single continuous but time-varying control. The σ-process proposed by Astolfi [2] is a common representative of such a discontinuous control design. Consider the chained system in (1.3.1), the following state transformation is valid for all $x_1 \neq 0$,

$$\xi_1 = x_1, \;\; \xi_2 = x_2, \;\; \xi_i = \frac{x_i}{x_1^{i-2}}, \;\; i = 3, \cdots, n.$$

Letting $u_1 = -k\xi_1$, the ξ-system transforms to:

$$\dot{\xi} = \begin{bmatrix} -k & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & -k & k & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & (n-2)k \end{bmatrix} \xi + \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} u_2.$$

It is a stabilizable linear system, and u_2 is the new input. Typically, one can choose the following linear control law

$$u_2 = p_2\xi_2 + p_3\xi_3 + \cdots + p_n\xi_n$$

to assign the eigenvalues in left half of the complex plane, therefore the closed-loop system (in ξ-coordinates) is globally exponentially stable. However the linear control u_2 is not defined in the set

$$D = \{x \in R^n : x_1 = 0\},$$

because the transformation is no longer valid in the set. To handle this problem, it is proposed to first apply some open-loop controls for an apriori fixed time t_s in order to steer the state away from the singularity and then switch back to the linear feedback control law [2][10][62], hence there is discontinuity in the control.

3.3 Time-Varying Continuous Controls

There exist two type of time-varying continuous control. One is periodic, proposed in [54] and [69]. The periodic control is based on power form (1.3.2). The controller is given by:

$$\begin{aligned} u_1 &= -\xi_1 - (\sum_{j=1}^{n-2} \xi_{j+2}^2)(\sin(t) - \cos(t)) \\ u_2 &= -\xi_2 - (\sum_{j=1}^{n-2} c_j \xi_{j+2}) \cos(jt) \end{aligned}$$

Later on, to improve its convergence rate, [45] proposed homogeneous feedback approach. The control and simulation results will be discussed in detail lately when compare to our solution.

On the other hand, design of aperiodic time-varying feedback control was explored in [58] and [70]. [58] adopted a dynamic control,

$$\dot{u}_1 = -(k_1 + \zeta)u_1 - k_1\zeta x_1, \;\; u_1(t_0) = c_u\|x(t_0)\|.$$

Based on the dynamic control, a virtual output was constructed

$$y_d \triangleq \frac{k_1 x_1 + u_1}{k_1 - \zeta}.$$

Using the property $\dot{y}_d = -\zeta y_d$, y_d is applied in state scaling. Noting that the solution of y_d is $y_d = ce^{-\zeta t}$, the undergoing transformation is similar to the transformation proposed in this chapter. [70] obtained u_1 by augmenting the first subsystem to:

$$\dot{x}_0 = x_1, \;\; \dot{x}_1 = u_1.$$

Let α be the greater eigenvalue and β be the smaller one of the augmented system, then $u_1 = e^{-\beta t} f(t)$, where

$$f(t) = \beta^2 \frac{\alpha x_0(0) + x_1(0)}{\alpha - \beta} - \alpha^2 \frac{\beta x_0(0) + x_1(0)}{\alpha - \beta} e^{-(\alpha-\beta)t}.$$

And $z(t) = e^{-\beta t}$ is used in the state scaling transformation. The advantages of these two controls are that the state response and controls are all smooth, exponentially converging fast (similar rate with the approach in this chapter) with no oscillations. However their disadvantage is, as illustrated in the control equations, the successful control relies on proper tuning of some controller parameters that related to the system's initial conditions, making it fail to be a pure state feedback control, hence is less favorable.

CHAPTER 4
SMOOTH PURE FEEDBACK STABILIZATION OF CHAINED NONHOLONOMIC SYSTEMS

In this chapter, a smooth pure feedback control design is proposed and a novel feedback design scheme is proposed, which renders a smooth, time-varying, aperiodic, pure feedback control with exponential convergence rates. There are three main advantages with the proposed design: 1) In general, time-varying designs are mostly periodic and render asymptotic stability, whereas the proposed approach is aperiodic and have exponential convergent rates; 2) A novel state scaling transformation is proposed. It shows that even though u_1 vanishes in regulation problems, the controllability of chained systems can be regain by judiciously designing the input u_1 and by applying state transformations; 3) A class of memory functions is introduced into the control design, the controller dependency on the system's initial conditions in our previous work is removed and the control is a pure feedback. Moreover, the design is shown to be inversely optimal. Simulations and comparisons are carried through to verify the effectiveness of the proposed designs.

4.1 Problem Formulation

The feedback control design is to be studied based on the canonical forms obtained in previous sections. Exponential convergent rate is pursued. The chained system (1.3.1) can be rewritten into the following form if we reorder the states, with the initial condition

$x(t_0)$:

$$\dot{x}_1 = u_1, \quad \dot{x}_2 = x_3 u_1, \cdots, \dot{x}_{n-1} = x_n u_1 \quad \dot{x}_n = u_2, \tag{4.1.1}$$

where $x = [x_1, \cdots, x_n]^T \in \Re^n$ is the state, $u = [u_1, \ u_2]^T \in \Re^2$ is the control input.

System (4.1.1) can be partitioned into the following two subsystems:

$$\dot{x}_1 = u_1, \tag{4.1.2}$$

and

$$\dot{z} = u_1 A z + B u_2, \tag{4.1.3}$$

where $z = [z_1 \ z_2 \ \cdots \ z_{n-1}]^T \triangleq [x_2 \ x_3 \ \cdots \ x_n]^T$, and

$$A \triangleq \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & & 1 \\ 0 & 0 & \cdots & & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}.$$

As has been shown by (4.1.2) and (4.1.3), it is well recognized that the chained systems have some good properties:

1. Subsystem (4.1.2) is linear, and u_1 may be easily designed to stabilize x_1.

2. Subsystem (4.1.3) is a linear time-varying system, whose time varying components only exist in the matrix A. Specifically, it is a chain of integrators with weight u_1.

3. System (4.1.1) is nonlinearly controllable everywhere because the Lie brackets argument on its vector fields has full rank.

Although chained systems have the above nice properties, it remains to be interesting research subjects because of the following negative properties.

1. They are not linearly controllable around the origin.

2. Topologically, chained systems cannot be stabilized by any continuous feedback control $u = u(x)$ because of its nonlinear characteristics.

3. Chained systems are not globally feedback linearizable. Although local feedback incarnation is possible such as the σ-process, there is singularity manifold remains in the neighborhood around the origin.

To overcome these difficulties, discontinuous switching control (resulted from local state feedback linearization such as σ-process) and time-varying feedback control has been explored in literature. In general, time-varying feedback controls are periodic and has slow asymptotic convergence, while discontinuous controls can easily achieve an exponential convergence rate at the cost of sacrificing its continuity. But are there continuous feedback controls that are aperiodic and have exponential convergence rates? A straightforward thinking to answer this question is to search for a global singularity free transformation that transforms the chained systems into a controllable linear form, then to synthesize the controls in the transformed domain.

4.2 Global State Scaling Transformation And Control Design Scheme

This section presents the feedback design of u_1. A global state-scaling transformation is proposed to overcome the singularity problem of existing transformations. This novel transformation enables the designer to regain uniform controllability of the chained systems and to design a class of smooth, time-varying, aperiodic, pure feedback controls that are inverse optimal and have exponential stability.

4.2.1 Design of Control Component u_1

Before presenting the design of u_1, a set of memory functions is defined as:

Definition 8. *For a time set:*

$$\mathcal{T} = [t_0\ t], \quad t \geq t_0 \geq 0,$$

a set of memory function is defined to be:

$$\mathcal{M}_{\mathcal{F}} = \{f : \Re^n \times \Re \to \Re^m | f(x(\eta), \eta),\ \forall \eta \in \mathcal{T};\ n, m \in \aleph\}.$$

From the definition, it is clear that the output of a memory function not only relates to its current variables, but also relates to the history of its variables.

The proposed control for component $u_1(t)$ is:

$$u_1(t) = -\alpha x_1 + g(z,t)e^{-\beta t}, \tag{4.2.1}$$

where $\alpha > \beta > 0$. To be a pure state feedback and non-switching control, $g(z,t)$ is required to have the following two properties:

1. $g(z,t)$ is smooth, uniformly bounded by $c > g(z,t) \geq \underline{g} \geq 0$ for some constants $c > \underline{g} \geq 0$.

2. In case of $\|z(t_0)\| = 0$, there should be $\underline{g} = 0$ and $g(z,t) \equiv 0$ for all $t > t_0$, i.e. if the subsystem (4.1.3) is initially at the origin, control u_1 reduces to a regular negative state feedback. In case of $\|z(t_0)\| \neq 0$, $g(z,t)$ should monotone converge to c from $\underline{g}$ and $(c - g(z,t)) \in L^2[t_0\ \infty)$.

Property 2 requires that if $\|z(t_0)\| = 0$, then $g(z,t) \equiv 0$. However, in case of $\|z(t_0)\| \neq 0$, there is $\lim_{t\to\infty} \|x(t)\| = 0$, which implies $\lim_{t\to\infty} \|z(t)\| = 0$, but now $\lim_{t\to\infty} g(z,t) = c \neq 0$. From this contradiction, one can conclude that if $g(z,t)$ is to meet the requirements for both cases, it can only be a memory function, i.e. $g(z,t) \in \mathcal{M}_{\mathcal{F}}$. The second property also implies that if $x(t_0)$ is in the singular manifold $\{x|x_1 = 0, \|z\| \neq 0\}$, then $g(z,t)$ is able to yield a nonzero number so that x_1 and u_1 will deviate from zero. Then the controllability of subsystem (4.1.3) can be recovered in the subsequent design through state scaling transformations.

Remark 4.2.1. *Though the first property sets $c > g(z,t) \geq \underline{g} \geq 0$, the design scheme is also valid if $c < g(z,t) \leq \overline{g} \leq 0$, with $\overline{g} \leq 0$ to be some constant and corresponding changes are made in property 2.*

4.2.2 A Global State Transformation

For the subsystem (4.1.3), the following novel state scaling transformation is proposed: for $i = 1, \cdots, n-1$,

$$\xi_i = \begin{cases} 0 & if \ \|z(t_0)\| = 0 \\ \frac{z_i}{e^{-(n-1-i)\beta t}} & if \ else \end{cases}. \tag{4.2.2}$$

In the case that $\|z(t_0)\| = 0$, the ξ-system wouldn't move. In the case that $\|z(t_0)\| \neq 0$, for $i = 1, \cdots, n-2$, the new dynamic equations are:

$$\begin{aligned} \dot{\xi}_i &= \frac{\dot{z}_i}{e^{-(n-1-i)\beta t}} - \frac{-\beta(n-1-i)e^{-\beta t}}{e^{-(n-i)\beta t}} z_i \\ &= \frac{u_1}{e^{-\beta t}} \xi_{i+1} + \beta(n-1-i)\xi_i. \end{aligned} \tag{4.2.3}$$

For $i = n-1$, since $\xi_i = z_i$, it follows that:

$$\dot{\xi}_{n-1} = u_2. \tag{4.2.4}$$

Combine (4.2.3) and (4.2.4) into a matrix form and put together with the case that $\|z(t_0)\| = 0$, the following dynamic model in transformed space is established:

$$\dot{\xi} = \begin{cases} 0 & if \ \|z(t_0)\| = 0 \\ F(z,t)\xi + Bu_2 & if \ else \end{cases}, \tag{4.2.5}$$

where

$$F(z,t) = diag\{\beta(n-2), \beta(n-3), \cdots, \beta, 0\} + [g(z,t) - \alpha \frac{x_1}{e^{-\beta t}}]A.$$

The uniform complete controllability of the transformed system $\{F(z,t), B\}$ is established in the following theorem.

Theorem 2. *If $g(z,t)$ has the properties given in section 4.2.1, then the transformed system $\{F(z,t), B\}$ is uniformly completely controllable.*

Proof. Simple derivation shows that:

$$\frac{d}{dt}\frac{x_1(t)}{e^{-\beta t}} = -(\alpha-\beta)\frac{x_1}{e^{-\beta t}} + g(z,t).$$

Therefore, $\frac{x_1(t)}{e^{-\beta t}}$ can be solved as:

$$\frac{x_1(t)}{e^{-\beta t}} = \frac{x_1(t_0)}{e^{-\beta t_0}}e^{-(\alpha-\beta)(t-t_0)} + \int_{t_0}^{t} e^{-(\alpha-\beta)(t-\tau)}g(z,\tau)d\tau. \tag{4.2.6}$$

Since $\lim_{t->\infty} g(z,t) = c$, $\lim_{t->\infty}\frac{x_1(t)}{e^{-\beta t}} = \frac{c}{\alpha-\beta}$. therefore, we can obtain:

$$\lim_{t->\infty}[g(z,t) - \alpha\frac{x_1}{e^{-\beta t}}] = \lim_{t->\infty} g(z,t) - \alpha\lim_{t->\infty}\frac{x_1}{e^{-\beta t}} = -\frac{c\beta}{\alpha-\beta}.$$

It follows that the time-varying system $\{F(z,t),\ B\}$ can be partitioned into a nominal component and a time-varying component:

$$F(z,t) = F_0 + \frac{c\beta}{\alpha-\beta}A + [g(z,t) - \alpha\frac{x_1}{e^{-\beta t}}]A = F_0 + F_t(z,t),$$

where

$$F_0 = diag\{\beta(n-2), \cdots, \beta, 0\} - \frac{c\beta}{\alpha-\beta}A,$$

and

$$F_t(z,t) = [g(z,t) - \alpha\frac{x_1}{e^{-\beta t}} + \frac{c\beta}{\alpha-\beta}]A. \tag{4.2.7}$$

It is clear that the time-varying component $F_t(z,t)$ vanishes, hence the transformed system $\{F(z,t), B\}$ converges to its nominal system $\{F_0, B\}$. By the design properties of $g(z,t)$, there is $c \neq 0$. Hence the pair $\{F_0, B\}$ is completely controllable, which implies the time-varying system $\{F(z,t), B\}$ is uniformly completely controllable.

□

4.2.3 Design of Control Component u_2

Control component u_2 is designed to be:

$$u_2(t) = -R_2^{-1} B^T \hat{P}(t) \xi, \tag{4.2.8}$$

where $\hat{P}(t) > 0$ is symmetric, uniformly bounded, and satisfies the following matrix differential Riccati equation with $\hat{P}(\infty) > 0$.

$$\dot{\hat{P}}(t) + \hat{P}(t)\hat{F}(t) + \hat{F}^T(t)\hat{P}(t) + Q_2 - \hat{P}(t)BR_2^{-1}B^T\hat{P}(t) = 0, \tag{4.2.9}$$

where

$$\hat{F}(t) = F_0 + [\underline{g} + \frac{\beta c}{\alpha - \beta} + \alpha(\underline{g} - c)(t - t_0)]e^{-(\alpha-\beta)(t-t_0)}A,$$

and $Q_2 \in \Re^{n-1 \times n-1}$, $R_2 \in \Re$ are constant and positive definite matrices. By a procedure similar to theorem 2, the uniform complete controllability of the pair $\{\hat{F}(t), B\}$ can be verified, hence such a $\hat{P}(t)$ can always be found.

Lemma 1. *Let*

$$\hat{F}_t(t) = [\underline{g} + \frac{\beta c}{\alpha - \beta} + \alpha(\underline{g} - c)(t - t_0)]e^{-(\alpha-\beta)(t-t_0)}A, \tag{4.2.10}$$

then the norm of difference $\|F_t(z,t) - \hat{F}_t(t)\| \in L^2[t_0\ \infty)$.

Proof. It follows from (4.2.6) and (4.2.7) that:

$$\begin{aligned} F_t(z,t) &= [g(z,t) - \alpha\frac{x_1(t)}{e^{-\beta t}} + \frac{c\beta}{\alpha - \beta}]A \\ &= \{[g(z,t) - c] - \gamma e^{-(\alpha-\beta)(t-t_0)} \\ &\quad -\alpha\int_{t_0}^{t} e^{-(\alpha-\beta)(t-\tau)}[g(z,\tau) - c]d\tau\}A, \end{aligned} \tag{4.2.11}$$

where $\gamma = \alpha[\frac{x_1(t_0)}{e^{-\beta t_0}} - \frac{c\beta}{\alpha-\beta}]$. In the above equation, by the design properties of $g(z,t)$, $(g(z,t)-c)$ is L^2, and the second term is also L^2. For the third term, it can be treated as the input response of an exponential stable, linear time-invariant dynamic system with a L^2 input, hence this term has to be L^2 [27]. Therefore $\|F_t(z,t)\|$ is L^2. Moreover, from (4.2.10), $\|\hat{F}_t(t)\|$ is L^2. Since:

$$\|F_t(z,t) - \hat{F}_t(t)\| \leq \|F_t(z,t)\| + \|\hat{F}_t(t)\|,$$

there must be $\|F_t(z,t) - \hat{F}_t(t)\| \in L^2[t_0\ \infty)$.

□

The convergence property of the closed-loop system (4.1.1) under control (4.2.1) and (4.2.8) is presented in the following theorem.

Theorem 3. *For any $g(z,t)$ that has the properties presented in section 4.2.1, the control (4.2.1) and (4.2.8) globally asymptotic stabilize the system (4.1.1) with exponential convergence rates.*

Proof. It is clear from (4.2.1) and (4.2.8) that if $\|z(t_0)\| = 0$, then $u_2 \equiv 0$ and $u_1 = -x_1$, therefore system (4.1.1) is exponentially stabilized. Consider the case that $\|z(t_0)\| \neq 0$. For subsystem (4.1.2), take the following Lyapunov function candidates $V_1(x_1) = \frac{1}{2}x_1^2$, and $V_2(\xi) = \xi^T \hat{P}(t)\xi$. It follows that:

$$\begin{aligned} \dot{V}(x_1) &= x_1\dot{x}_1 \\ &= -\alpha x_1^2 + x_1 g(z,t)e^{-\beta t} \\ &\leq -\alpha x_1^2 + ce^{-\beta t_0}|x_1|, \end{aligned} \tag{4.2.12}$$

(4.2.12) shows that x_1 is uniformly ultimately bounded by the set

$$\Omega \stackrel{\triangle}{=} \{x_1 : |x_1| \leq \frac{ce^{-\beta t_0}}{\alpha}\}.$$

If $x_1(t_0) \in \Omega$, $x_1(t)$ remains in Ω for $t \geq t_0$. If $x_1(t_0) \notin \Omega$, $|x_1|$ monotone decreases into Ω. Therefore, a uniform bound for $x_1(t)$ is:

$$\delta \triangleq \max\{|x_1(t_0)|, \frac{ce^{-\beta t_0}}{\alpha}\}.$$

Then, (4.2.12) becomes

$$\dot{V}_1(x_1) \leq -2\alpha V_1 + \delta c e^{-\beta t}.$$

Hence subsystem (4.1.2) is globally exponentially attractive by lemma 2.19 of [55]. Therefore subsystem (4.1.2) is asymptotic stable with exponential convergence. The closed loop system of (4.2.5) is

$$\begin{aligned}
\dot{\xi} &= F(z,t)\xi - BR_2^{-1}B^T\hat{P}(t)\xi \\
&= [F_n - BR_2^{-1}B^T\hat{P}(t) + F_t(z,t)]\xi \\
&= [\hat{F}(t) - BR_2^{-1}B^T\hat{P}(t) + F_t(z,t) - \hat{F}_t(t)]\xi,
\end{aligned}$$

where $F_t(z,t)$ is defined in (4.2.7). It follows that:

$$\begin{aligned}
\dot{V}_2(\xi) &= \xi^T\{\dot{\hat{P}}(t) + [\hat{F}(t) - BR_2^{-1}B^T\hat{P}(t) + F_t(z,t) - \hat{F}_t(t)]^T\hat{P}(t) \\
&\quad + \hat{P}(t)[\hat{F}(t) - BR_2^{-1}B^T\hat{P}(t) + F_t(z,t) - \hat{F}_t(t)]\}\xi \\
&= \xi^T[\dot{\hat{P}}(t) + \hat{F}(t)^T\hat{P}(t) + \hat{P}(t)\hat{F}(t) - 2\hat{P}(t)BR_2^{-1}B^T\hat{P}(t) + N(z,t)]\xi \\
&= -\xi^T[Q_2 + \hat{P}(t)BR_2^{-1}B^T\hat{P}(t) - N(z,t)]\xi \\
&\leq [-\frac{c_2}{c_3} + \frac{\sum_{i=1}^{n-1}|\lambda_i(N(z,t))|}{c_4}]V_2, \qquad (4.2.13)
\end{aligned}$$

where $\lambda_i(\cdot)$ denotes the ith eigenvalue of a square matrix, $c_2,\ c_3,\ c_4$ are constants that satisfy:

$$c_1 I > Q_2 + \hat{P}(t)BR_2^{-1}B^T\hat{P}(t) > c_2 I > 0, \quad c_3 I > \hat{P}(t) > c_4 I > 0,$$

and

$$N(z,t) = [\hat{P}(t)(F_t(z,t) - \hat{F}_t(t)) + (F_t(z,t) - \hat{F}_t(t))^T\hat{P}(t)] \in \Re^{n-1\times n-1}. \qquad (4.2.14)$$

Since Q_2, R_2 are constant matrices, hence $\hat{P}(t)$ is uniformly bounded and constants $c_1,\ c_2,\ c_3,\ c_4$ can be found.

Note that,

$$|\lambda_i(N(z,t))| \leq \|N(z,t)\| \leq 2\|\hat{P}(t)\|\|F_t(z,t) - \hat{F}_t(t)\|.$$

Since $\hat{P}(t)$ is uniformly bounded, and by lemma 1, $\|F_t(z,t) - \hat{F}_t(t)\| \in L^2[t_0\ \infty)$, both $\|N(z,t)\|$ and $|\lambda_i(N(z,t))|$ are L^2. Then treating (4.2.13) as a scalar dynamic system, V_2 is exponentially stabilized by invoking Lemma 2.2 of [51] and comparison principle. It follows that the ξ-systems is exponentially stabilized, which implies the z-system is exponentially stabilized according to the transformation (4.2.2). After combining the results for subsystems (4.1.2) and (4.1.3), it is concluded that the overall system has asymptotic stability with exponential convergence rates. Since the argument is globally valid, the stability results is global.

□

The control u_2 in (4.2.8) shows that the underlying idea is that using the pure time function $\hat{F}_t(t)$ in (4.2.10) to approximate the time-varying component $F_t(z,t)$ of $F(z,t)$, which is given in (4.2.11). The goal is to remove the state variable z from the system matrix, hence the control u_2 could be synthesized from the linear time-varying system $\{\hat{F}(t), B\}$. This approximation assumes that $g(z,t)$ converges to c exponentially, i.e.

$$g(z,t) - c \approx (\underline{g} - c)e^{-(\alpha-\beta)(t-t_0)}.$$

In this case, the model difference $\|F_t(z,t) - F_t(t)\|$ is L^2 by Lemma 1, which guarantees the exponential stability.

Note that in limit, both $F(z,t)$ and $\hat{F}(t)$ reduce to their nominal system F_0. Hence by solving $P > 0$ from the following algebraic Riccati equation (ARE):

$$F_0^T P + P F_0 + Q_2 - P B R_2^{-1} B^T P = 0,$$

the control

$$u_2'(t) = -R_2^{-1} B^T P \xi \tag{4.2.15}$$

is also a stabilizing control, since this case is equivalent to take $\hat{F}_t(t) \equiv 0$, and the model difference is $\|F_t(z,t)\|$, which by itself is L^2 as shown in Lemma 1. In simulations, we compared control effects for both u_2 and u_2'. It shows that the performance of u_2 with $\hat{F}_t(t)$ in (4.2.10) is much better.

4.3 Optimal Performance

The following theorem indicates that the proposed control (4.2.1) and (4.2.8) is optimal with respect to some quadratic performance index.

Theorem 4. *For system (4.1.1), the feedback controls (4.2.1) and (4.2.8) are optimal with respect to performance index $J = J_1 + J_2$, where*

$$J_1(t, u_1(t)) = \int_t^\infty \{ \begin{bmatrix} x_1 & y \end{bmatrix} Q_1(t) \begin{bmatrix} x_1 \\ y \end{bmatrix} + u_1^2 \} dt$$

and

$$J_2(t, u_2(t)) = \int_t^\infty [\xi^T \hat{Q}_2(t) \xi + u_2 R_2 u_2] dt,$$

where $y = e^{-\beta t}$ is the augmented state,

$$Q_1(t) = \begin{bmatrix} \alpha^2 & \dot{g} - (\alpha+\beta)g \\ \dot{g} - (\alpha+\beta)g & 2k\beta + g^2 \end{bmatrix},$$

with k chosen to satisfy:

$$k > \max\{\frac{c^2}{\alpha}, \frac{\dot{g}^2 + g^2\beta(4\alpha+\beta) - 2g\dot{g}(\beta+2\alpha)}{4\alpha^2\beta}, \frac{\dot{g}^2 + g^2\beta(2\alpha+\beta) - 2g\dot{g}(\beta+\alpha)}{2\alpha^2\beta}\},$$

and $\hat{Q}_2(t) = Q_2 - N(z,t)$, with $N(z,t)$ defined in (4.2.14).

Proof. By design properties of $g(z,t)$, g is monotone and uniformly bounded, therefore g must be uniformly continuous, hence $\dot{g}$ is uniformly bounded. Therefore, such a k can always be found and by the specified choice of k, $Q_1(t)$ is positive definite.

Under control (4.2.1), the closed loop system of subsystem (4.1.2) is:

$$\dot{x}_1 = u_1 = -\alpha x_1 + g(z,t)y. \tag{4.3.1}$$

We first show that

$$V_1'(x_1, y) \stackrel{\triangle}{=} \alpha x_1^2 - 2g x_1 y + k y^2$$

is a Lyapunov function of the augmented system (4.3.1). It is straightforward that by the specified choice of k, V_1' is positive definite. It follows that:

$$\begin{aligned} \dot{V}_1' &= 2\alpha x_1 \dot{x}_1 + 2ky\dot{y} - 2\dot{g}x_1 y - 2g\dot{x}_1 y - 2g x_1 \dot{y} \\ &= -2\alpha^2 x_1^2 - 2(k\beta + g^2)y^2 + (4\alpha g + 2g\beta - 2\dot{g})x_1 y \\ &= -\begin{bmatrix} x_1 & y \end{bmatrix} \begin{bmatrix} 2\alpha^2 & \dot{g} - g(2\alpha+\beta) \\ \dot{g} - g(2\alpha+\beta) & 2k\beta + 2g^2 \end{bmatrix} \begin{bmatrix} x_1 \\ y \end{bmatrix}. \end{aligned}$$

$\dot{V}_1'$ is negative definite, hence V_1' is a Lyapunov function of the augmented system. To show the optimality of u_1 w.r.t. J_1, substitute control u_1 in (4.2.1) with an incremental

term Δu_1 into J_1, i.e. $u_1(t) = -\alpha x_1 + g(z,t)y + \Delta u_1$. Evaluate $\dot{V}_1'$ along the system's new trajectory with the perturbed control, we have:

$$\dot{V}_1' = -\begin{bmatrix} x_1 & y \end{bmatrix} \begin{bmatrix} 2\alpha^2 & \dot{g} - g(2\alpha+\beta) \\ \dot{g} - g(2\alpha+\beta) & 2k\beta + 2g^2 \end{bmatrix} \begin{bmatrix} x_1 \\ y \end{bmatrix} + 2u_1\Delta u_1.$$

It follows that the performance index J_1 for the perturbed system is:

$$\begin{aligned} J_1 &= \int_t^\infty \{ \begin{bmatrix} x_1 & y \end{bmatrix} Q_1(t) \begin{bmatrix} x_1 \\ y \end{bmatrix} + (u_1 + \Delta u_1)^2 \} dt \\ &= \int_t^\infty \{ \begin{bmatrix} x_1 & y \end{bmatrix} Q_1(t) \begin{bmatrix} x_1 \\ y \end{bmatrix} + u_1^2 + 2u_1\Delta u_1 + \Delta u_1^2 \} dt \\ &= -\int_t^\infty dV_1' + \int_t^\infty \Delta u_1^2 dt \\ &= V_1'(x_1(t), y(t)) + \int_t^\infty \Delta u_1^2 dt, \end{aligned}$$

which is minimized by $\Delta u_1 = 0$, hence u_1 is optimal with respect to J_1.

For system (4.2.5), it is straightforward to verify that the following matrix differential equation holds:

$$\dot{\hat{P}}(t) + \hat{P}(t)F(z,t) + F(z,t)\hat{P}(t) + \hat{Q}_2(t) - \hat{P}(t)BR_2^{-1}B^T\hat{P}(t) = 0. \tag{4.3.2}$$

To show the optimality of system (4.2.5) with respect to J_2, substituting u_2 in (4.2.8) with an incremental term Δu_2 (that is, $u_2 = -R_2^{-1}B^T\hat{P}\xi + \Delta u_2$) into J_2, and evaluate $\dot{V}_2$ along the new state trajectory and control:

$$\dot{V}_2 = -\xi^T[\hat{Q}_2(t) + \hat{P}(t)BR_2^{-1}B^T\hat{P}(t)]\xi + 2\xi^T\hat{P}(t)B\Delta u_2.$$

It follows that the performance index J_2 for the perturbed system is:

$$\begin{aligned} J_2 &= \int_t^\infty [\xi^T \hat{Q}_2(t)\xi + \xi^T \hat{P}(t)BR_2^{-1}B^T\hat{P}(t)\xi - 2\xi^T\hat{P}(t)B\Delta u_2 + \Delta u_2^T R_2 \Delta u_2]dt \\ &= -\int_t^\infty dV_2 + \int_t^\infty \Delta u_2^T R_2 \Delta u_2 dt \\ &= V_2(\xi(t)) + \int_t^\infty \Delta u_2^T R_2 \Delta u_2 dt. \end{aligned}$$

Here, (4.3.2) is used. It is clear that J_2 is minimized by $\Delta u_2 = 0$. Note that Q_2 is positive definite and in theorem 3, we have shown $\|N(z,t)\|$ is L^2, therefore $N(z,t)$ vanishes. Hence in some cases, $\hat{Q}_2(t)$ might need a finite period to be positive definite. But by the above Lyapunov argument, the performance index J_2 would be always positive. To this end, the overall system has been shown to be optimal with respect to J.

□

4.4 Design Examples

In this section, examples of applying the proposed design scheme are provided. Examples of nontrivial memory functions in $\mathcal{M}_\mathcal{F}$ include, for instance,

$$\int_{t_0}^t l(\|z(\tau)\|)d\tau, \quad \min_{t_0\le\eta\le t} l(\|z(\eta)\|), \quad \max_{t_0\le\eta\le t} l(\|z(\eta)\|),$$

where $l(\cdot)$ is a function. For example, we design $g(z,t)$ to be:

$$g(z,t) = \frac{t\int_{t_0}^t \|z(\tau)\|d\tau}{1 + t\int_{t_0}^t \|z(\tau)\|d\tau}. \tag{4.4.1}$$

According to theorem 3, to show the stability, one only needs to show that $g(z,t)$ in (4.4.1) has the three properties given in section 4.2.1.

It is straightforward to verify that the closed loop systems of (4.1.2) and (4.2.5) under control (4.2.1) and (4.2.8) are globally Lipschitz. Therefore the solution x_1 and ξ exists and

is unique, hence by transformation (4.2.2), solution z exists. For property 1, clearly $g(z,t)$ is differentiable everywhere for $t \geq t_0$ and uniformly bounded by $\underline{g} = 0$ and $c = 1$. For property 2, if $\|z(t_0)\| = 0$, then $u_2(t_0) = 0$. Subsystem (4.1.3) wouldn't move, hence $z(t) \equiv 0$, which in turn yields $g(z,t) \equiv 0$. In case of $\|z(t_0)\| \neq 0$, there is $\lim_{t\to\infty} g(z(t),t) = c = 1$. Moreover,

$$c - g(z,t) = \frac{1}{1 + t\int_{t_0}^{t} \|z(\tau)\| d\tau} > 0.$$

It is clear that whether or not z is exponential convergent, $(c - g(z,t)) \in L^2[t_0\ \infty)$.

4.5 Simulations And Comparisons With Other Existing Controls

In the simulation, a 3rd order chained system is studied. $g(z,t)$ in (4.4.1) is used. The design parameters are set to be $\alpha = 1$, $\beta = 0.5$, $Q_2 = I$ and $R_2 = 1$. To verify the effectiveness of avoiding singularity, initial condition of the state is set to be $x(t_0) = [0\ 0\ 1]^T$.

The results in Fig. 4.1 verify that the proposed stabilizing control is successful. Fig. 4.1(a) and Fig. 4.1(c) illustrate the state and control for u_2' in (4.2.15). Fig. 4.1(b) and Fig. 4.1(d) illustrate the control effects for u_2 in (4.2.8). Fig. 4.1(e) shows the model difference for the two cases. Clearly, In both cases, despite of $x_1(t_0) = 0$, asymptotic stability and exponential convergence rates are achieved and both states and controls are aperiodic. When u_2' is used, $\hat{F}_t(t) \equiv 0$, the model difference is $\|F_t(z,t)\|$, its transient is larger and converges slower. Fig. 4.1(e) shows that by applying (4.2.10), the model difference $\|F_t(z,t) - \hat{F}_t(t)\|$ is smaller, hence the transient response is improved.

For the same system with the same initial condition, simulations for discontinuous controls [43], ordinary periodic time-varying feedback controls [54] and ρ-exponential stabilizer [46] are also conducted. Fig. 4.2(a) and Fig. 4.2(b) show the state and control of the discontinuous control. Fig. 4.2(c) shows the states for an ordinary periodic time-varying feedback control and the ρ-exponential stabilizer. Fig. 4.2(d) shows the controls for an ordinary periodic time-varying feedback control and the ρ-exponential stabilizer.

Fig. 4.2(a) and 4.2(b) illustrate the simulation results for the discontinuous control. Fig. 4.2(b) shows that the controls are discontinuous when it is switched at time t_s (in the simulation, $t_s = 0.5$). Therefore the state response is not smooth at t_s as can be seen in Fig. 4.2(a). From t_s, linear control law were applied, system states and control converge to origin exponentially. However, with the apriori determined t_s, the transitory period and the open-loop control remains important regardless of the closeness of the initial conditions to the origin, therefore the closed-loop system is not Lyapunov stable and its performance is not guaranteed.

In addition to the discontinuous control design, researchers also proposed various types of smooth time-varying feedback control, either periodic or aperiodic. [54] has proposed the following design of aperiodic time-varying control. E.g. for the system:

$$\dot{x}_1 = u_1, \;\; \dot{x}_2 = u_2, \;\; \dot{x}_3 = x_2 u_1,$$

[54] proposed the following control:

$$\begin{cases} u_1(t,x) = -x_1 + x_3\cos(t) \\ u_2(t,x) = -x_2 + x_3^2\sin(t) \end{cases}. \tag{4.5.1}$$

Another time-varying, periodic feedback control is proposed in [69], which is based on power form (1.3.2). The controller is given by

$$\begin{cases} u_1 = -\xi_1 - (\sum_{j=1}^{n-2}\xi_{j+2}^2)(\sin(t) - \cos(t)) \\ u_2 = -\xi_2 - (\sum_{j=1}^{n-2} c_j x_{j+2})\cos(jt) \end{cases}.$$

Asymptotic stability of the closed-loop system for the control (4.5.1) can be illustrated by the following Lyapunov function:

$$\begin{aligned} V(t,x) &= (x_1 - \frac{x_3}{2}(\cos(t) + \sin(t)))^2 + (x_2 \\ &\quad -\frac{x_3^2}{2}(\sin(t) - \cos(t)))^2 + x_3^2. \end{aligned}$$

Later on, to improve its convergence rate, [45] introduced so-called ρ-exponential stabilizer using homogeneous feedback, i.e. the control changes to:

$$\begin{cases} u_1(t,x) = -x_1 + \lambda x_3\cos(t) \\ u_2(t,x) = -x_2 + \lambda^3 x_3^2\sin(t) \end{cases}, \tag{4.5.2}$$

where λ is obtained from

$$V(t, \Delta_\lambda x) = C, \tag{4.5.3}$$

with $\Delta_\lambda x = (\lambda x_1,\ \lambda x_2,\ \lambda^2 x_3)$ and C is a constant.

The simulation results for these two controls are illustrated in Fig. 4.2(c) and Fig. 4.2(d). Fig. 4.2(c)(a) and Fig. 4.2(d) show that the convergent rate of both state response

and control of (4.5.1) is unfavorably slow, while the ρ-exponential stabilizer does much better in Fig. 4.2(c) and Fig. 4.2(d). However its setting time (around 15 sec) is still much larger than the proposed approach (around 9 sec) and has more oscillations before converging. One drawback of ρ-exponential stabilizer is its performance is critically determined by the level set value C in equation (4.5.3), however there is no systematic way to determine what C should be except numerical tests.

On the other hand, design of aperiodic time-varying feedback control was explored in [58] and [70]. [58] adopted a dynamic control,

$$\dot{u}_1 = -(k_1 + \zeta)u_1 - k_1\zeta x_1, \;\; u_1(t_0) = c_u\|x(t_0)\|.$$

Based on the dynamic control, a virtual output was constructed

$$y_d \triangleq \frac{k_1 x_1 + u_1}{k_1 - \zeta}.$$

Using the property $\dot{y}_d = -\zeta y_d$, y_d is applied in state scaling. Noting that the solution of y_d is $y_d = ce^{-\zeta t}$, the undergoing transformation is similar to the proposed transformation.

In [70], subsystem (4.1.2) is augmented to:

$$\dot{x}_0 = x_1, \;\; \dot{x}_1 = u_1.$$

Let α be the greater eigenvalue and β be the smaller one of the augmented system, then $u_1 = e^{-\beta t}f(t)$, where

$$f(t) = \beta^2\frac{\alpha x_0(0) + x_1(0)}{\alpha - \beta} - \alpha^2\frac{\beta x_0(0) + x_1(0)}{\alpha - \beta}e^{-(\alpha-\beta)t}.$$

And $z(t) = e^{-\beta t}$ is used in the state scaling transformation. The advantages of these two controls are that the state response and controls are all smooth, exponentially converging fast (similar rate with the approach in this chapter) with no oscillations. However their disadvantage is, as illustrated in the control equations, the successful control relies on proper tuning of some controller parameters that related to the system's initial conditions, hence is not a pure feed-back control.

The characteristics of the aforementioned controls and our proposed control are summarized in Table 4.1, and their differences are easily seen.

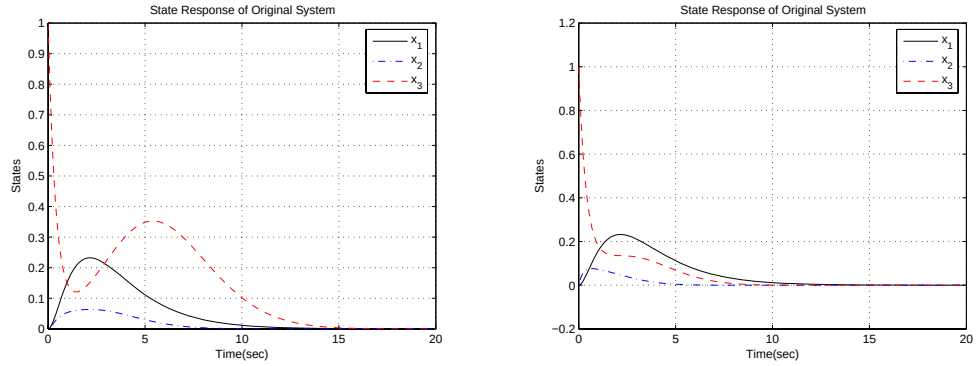
Table 4.1: Summary of Various Control Approaches

	switching	ordinary time-varying	ρ-exponential	our control
continuity	discontinuous	smooth	smooth	smooth
convergence	exponential	asymptotic	exponential	exponential
oscillation	aperiodic	periodic	periodic	aperiodic
stability region	global	global	global	global

4.6 Conclusion

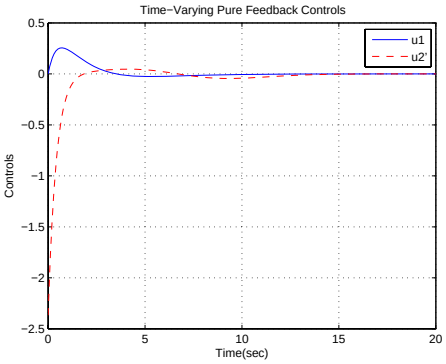
In this chapter, feedback stabilization problem of chained nonholonomic systems is studied by investigating its uniform controllability. It is illustrated that linear controllability does not hold for stabilizing the chained system but can be reestablished by a state scaling transformation. Based on this idea, we proposed a new design methodology and implemented one particular control. The procedure is systematic and straightforward. The controls are inverse

optimal with respect to some quadratic performance index. By simulations and comparisons with other existing controls, the proposed control is shown to be effective and exhibited advantages in smoothness, convergent rates, oscillations, and being pure state feedback.

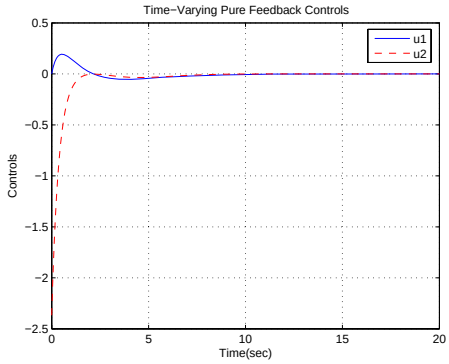


(a)

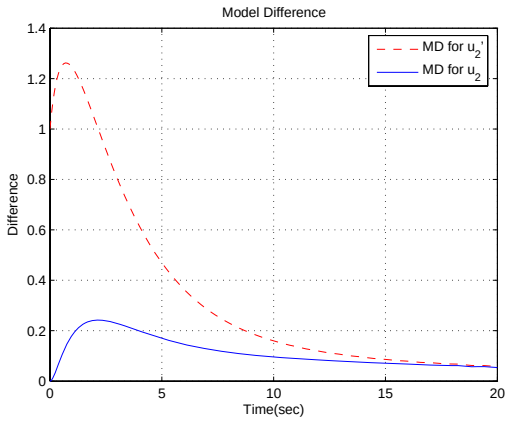
(b)



(c)

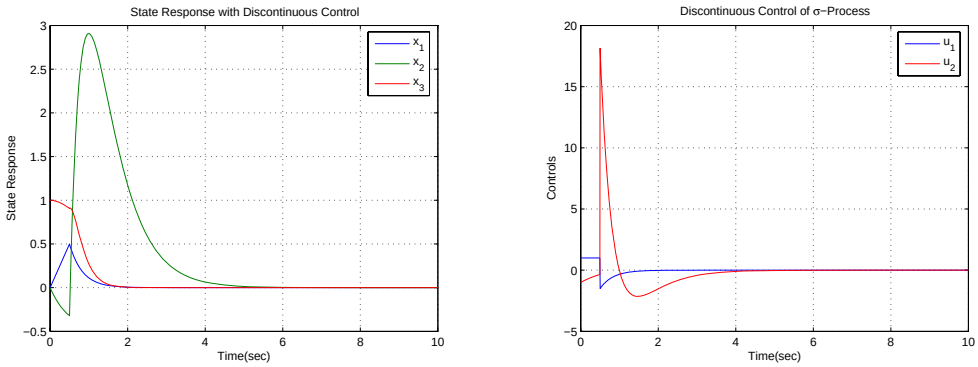


(d)



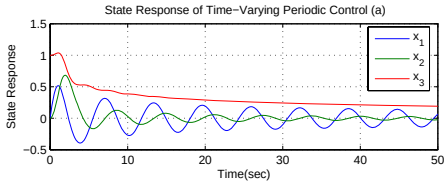
(e)

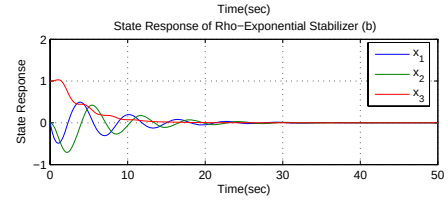
Figure 4.1: Simulation Results of The Proposed Controls. (a),(c) State and Control with u_2' in (4.2.15). (b),(d) State and Control with u_2 in (4.2.8). (e) Model Difference for u_2' and u_2.



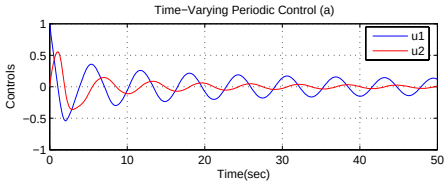
(a)

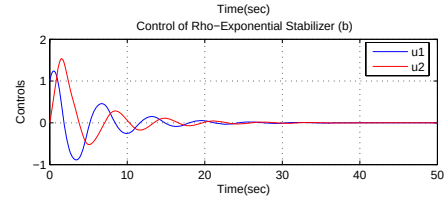
(b)





(c)





(d)

Figure 4.2: Control Effects for Various Other Control Designs. (a),(b) State and Control of Discontinuous Control. (c),(d) States and Controls for The Ordinary Periodic Time-varying Feedback Design and ρ-exponential Stabilizer.

CHAPTER 5
SATURATED CONTROL OF CHAINED NONHOLONOMIC SYSTEMS

In past decades, plenty of effort has been devoted to the stabilization and tracking control of chained systems [2, 43, 54, 45, 57, 58, 70, 79, 25, 42]. It is well known that the chained form is a canonical form for many nonholonomic mechanical systems, hence control designs based on chained systems ensure their wide applicability. Since chained systems do not satisfy Brockett's necessary condition [9], discontinuous or time-varying feedback controls have to be sought for their stabilization. In the literature, a great deal of solutions have been obtained following the lines of using discontinuous control method or time-varying control method [30]. In general, discontinuous controls can render exponential stability [43, 42, 6, 24], while time-varying controls lead to asymptotic stability [54, 69, 64]. More recent study has also seen the results of $\rho-$exponential stability of chained system using time-varying periodic feedback controls [45]. In [58, 70, 79], exponential convergence rates are also reported for continuous time-varying aperiodic design.

Despite these extensive studies on feedback control design, the problem of stabilization with input saturation effect is rarely addressed. In this section, we focus on designing such a control with constrained inputs. When actuator saturation is applied to the inputs, usually, there could be two types of treatments. One is to handle the saturation effect implicitly (or a posteriori), through the so-called anti-windup strategies [19, 31, 16]. The

other treatment is to handle the saturation explicitly (or a priori), pursuing one of the following two techniques. The first one is the saturation avoidance method which prevents the saturation from taking place. Therefore the resulted controller always operates in the linear region of saturation nonlinearities. The second approach is the saturation allowance approach which allows the saturation to take place and take saturation effects into account from the outset of control design. The existing designs for nonholonomic systems have been following the second approach mentioned above. In [25], the saturated stabilization and tracking controls are directly synthesized from a unicycle-type robot model by using passivity theory and Lyapunov argument. However, the design was not generalized to nonholonomic systems in the chained form. In [42], the authors proposed a discontinuous control design, seeking to remedy the excessively large control inputs near the singular manifold resulting from the σ-process [2]. The state space is decomposed into two separate 'good' or 'bad' regions. In the 'good' region, the control inputs are typically small. In the 'bad' region, the controller uses the so called linear-dominant function (L.D.F) to scale down the magnitude of the control inputs while forcing the trajectories to get into the 'good' region. This section proposes a novel switching control design. The chained system is divided into two subsystems controlled by u_1 and u_2, respectively. The key idea is to make u_1 piecewise constant, which renders the other subsystem a chain of integrators. Then, the multiple-integrator system is transformed into a linear system with an upper triangular system matrix and control u_2 is synthesized.

In this chapter, $\|x\|$ denotes the Euclidean norm of a vector x, $\min\{a, b\}$ and $\max\{a, b\}$ define the minimum and maximum of parameters a and b. The sign functions are defined as:

$$\text{sign}(x) = \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases}.$$

The saturation functions are defined as $\text{sat}_\phi(x) = \text{sign}(x)\min\{|x|, \phi\}$, where ϕ is the saturation bound. Moreover, $\text{sat}_1(x)$ is written as $\text{sat}(x)$ for short.

5.1 Problem Formulation

The objective of this chapter is to present a control design strategy which globally stabilizes the chained nonholonomic system under saturation conditions. Consider the nth order chained system (4.1.1) which is subject to the following saturation constraint:

$$-\delta_i \leq u_i \leq \delta_i, \quad i = 1, 2, \ \delta_i > 0. \tag{5.1.1}$$

The control design follows the second aforementioned approach, i.e. the saturation effect is taken into consideration at the design phase. Clearly, subsystem (4.1.2) only contains x_1 and is independent of the rest of the states. It can be easily stabilized with or without saturation. Subsystem (4.1.3) is a linear time-varying (LTV) system, which is very structurally similar to a multiple-integrator system, except that it is weighted by one of the control inputs. Naturally, one would think of manipulating u_1 to gain advantages in controlling subsystem (4.1.3). A straightforward way is to create a piecewise constant u_1 that meets the saturation condition as well as stabilize the subsystem (4.1.2). Then subsystem (4.1.3) becomes a

constant-weighted multiple-integrator systems whose saturation control is studied in [68, 67, 44, 81].

5.2 The Saturated Control Design

Before proceeding with the control design, we first study the saturated control of a scalar system from [81].

Lemma 2. *Consider the following scalar system:*

$$\dot{\zeta} = u, \;\; u = -\epsilon \,\mathrm{sat}_\delta(\frac{\lambda\zeta}{\epsilon}) + \rho(t), \;\; t \geq t_0,$$

where $\rho(t) : [t_0, \; \infty) \rightarrow \Re$ *is uniformly bounded, and* $\epsilon\delta > |\rho(t)|, \;\; t \geq t_0$*, then there exists constant* $T > t_0$ *such that for* $\forall t > T$*,* $|\zeta| \leq \frac{\epsilon}{\lambda}\delta$ *holds. Moreover, the input* u *can be simplified as* $u = -\lambda\zeta + \rho(t)$*.*

Proof. Choose the Lyapunov function $V(\zeta) = \frac{1}{2}\zeta^2$. It follows that:

$$\dot{V}(\zeta) = \zeta\dot{\zeta} = \zeta[-\epsilon \,\mathrm{sat}_\delta(\frac{\lambda\zeta}{\epsilon}) + \rho(t)].$$

Since $\epsilon\delta > |\rho(t)|$, if $|\zeta| > \frac{\epsilon}{\lambda}\delta$, it follows that:

$$\mathrm{sign}(-\epsilon \,\mathrm{sat}_\delta(\frac{\lambda\zeta}{\epsilon}) + \rho(t)) = -\mathrm{sign}(x),$$

which implies that $\dot{V} < 0$. Consequently, $|\zeta|$ is uniformly bounded by $\frac{\epsilon}{\lambda}\delta$, in which there is no saturation and u is simplified. □

The result of the above scalar system can be extended to the following vector case:

Theorem 5. *Let $\lambda_i,\ i = 1, \cdots, n$ be a series of positive constants. Consider the follow linear system with input constraint $-u_{\max} \le u \le u_{\max}$ with $u_{\max} > 0$:*

$$\dot{\xi} = A_n \xi + b_n u_n, \tag{5.2.1}$$

where $\xi = [\xi_1\ \xi_2\ \cdots\ \xi_n]$ and

$$A_n = \begin{bmatrix} 0 & \lambda_2 & \cdots & \lambda_{n-1} & \lambda_n \\ 0 & 0 & \ddots & \vdots & \vdots \\ \vdots & \vdots & \cdots & \lambda_{n-1} & \lambda_n \\ 0 & 0 & \cdots & 0 & \lambda_n \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix}, b_n = \begin{bmatrix} 1 \\ \vdots \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

The nonlinear control:

$$u_n = -\sum_{i=1}^{n} \epsilon_i \,\mathrm{sat}(\frac{\lambda_i \xi_i}{\epsilon_i}), \tag{5.2.2}$$

where ϵ_i satisfies:

$$\begin{cases} \epsilon_1 > 0 \\ \epsilon_j > \sum_{i=1}^{j-1} \epsilon_i, \quad j = 2, 3, \cdots, n. \\ \sum_{i=1}^{n} \epsilon_i \le u_{\max} \end{cases} \tag{5.2.3}$$

is a globally stabilizing control that satisfies the input constraint. Furthermore, the closed loop system will operate in a linear region in finite time with eigenvalues $-\lambda_i,\ i = 1, \cdots, n$.

Proof. It follows that:

$$|u_n| \le \sum_{i=1}^{n} \epsilon_i \le u_{\max},$$

Therefore the saturation condition holds. u_n can be rewritten as:

$$u_n = -\epsilon_n \,\mathrm{sat}(\frac{\lambda_n \xi_n}{\epsilon_n}) - \sum_{i=1}^{n-1} \epsilon_i \,\mathrm{sat}(\frac{\lambda_i \xi_i}{\epsilon_i}).$$

If we look at the last state ξ_n, by applying Lemma 2, we know that ξ_n will enter a linear region $|\xi_n| \le \frac{\epsilon_n}{\lambda_n}$, The same happens to the other $n-1$ states one by one from ξ_{n-1} to ξ_1, with

the linear region $|\xi_i| \leq \frac{\epsilon_i}{\lambda_i}, \quad i = 1, \cdots, n-1$. After all states get into their linear regions, the closed loop system matrix becomes:

$$\begin{bmatrix} -\lambda_1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ -\lambda_1 & \ddots & -\lambda_{n-1} & 0 \\ -\lambda_1 & -\lambda_2 & \cdots & -\lambda_n \end{bmatrix},$$

which is stable and the eigenvalues are: $-\lambda_i,\ i = 1, \cdots, n$. □

The following corollary can be obtained from Theorem 5.

Corollary 1. *Once control (5.2.2) gets into its linear operate region, its saturation elements will not be saturated again, i.e. the control becomes a linear control law afterward.*

Proof. It can be deduced from (5.2.2) that the linear operate region can be explicitly given by the following set:

$$\Omega = \{\xi : |\xi_1| \leq \frac{\epsilon_1}{\lambda_1}, |\xi_2| \leq \frac{\epsilon_2}{\lambda_2}, \cdots, |\xi_n| \leq \frac{\epsilon_n}{\lambda_n}\}.$$

Suppose at certain moment, control (5.2.2) is in its linear region Ω, then it can be rewritten as:

$$u_n = -\lambda_n \xi_n + u_{n-1}.$$

Consider the last state equation:

$$\dot{\xi}_n = u_n = -\lambda_n \xi_n + u_{n-1}.$$

Take the Lyapunov function candidate $V_n = \frac{1}{2}\xi_n^2$. It follows that:

$$\begin{aligned} \dot{V}_n &= \xi_n \dot{\xi}_n \\ &= -\lambda_n \xi_n^2 + \xi_n u_{n-1} \\ &\leq -\lambda_n |\xi_n|^2 + |\xi_n||u_{n-1}| \\ &\leq -\lambda_n |\xi_n|^2 + |\xi_n| \epsilon_n. \end{aligned}$$

It shows that ξ_n is ultimately bounded by $|\xi_n| \leq \frac{\epsilon_n}{\lambda_n}$. Consider the second to last state equation:

$$\dot{\xi}_{n-1} = \lambda_n \xi_n + u_n = -\lambda_{n-1}\xi_{n-1} + u_{n-2}.$$

Take the Lyapunov function candidate $V_{n-1} = \frac{1}{2}\xi_{n-1}^2$. A similar process would show that ξ_{n-1} is ultimately bounded by $|\xi_{n-1}| \leq \frac{\epsilon_{n-1}}{\lambda_{n-1}}$. Repeating the same process for the state $\xi_1,\ \xi_2, \cdots, \xi_{n-2}$, one would have:

$$|\xi_i| \leq \frac{\epsilon_i}{\lambda_i}, \quad i = 1, 2, \cdots, n-2.$$

It shows that the state ξ is confined in the same set of Ω, which indicates that once the state gets into Ω, it cannot leave Ω, where control (5.2.2) is linear. □

5.2.1 The Control Design u_1 and u_2

For the chained system (4.1.1), we propose the following control of u_1:

$$u_1(t) = \begin{cases} -\text{sign}(x_1(t_0))k\delta_1, & t_0 \leq t \leq t_1 \\ -\text{sign}(x_1(t_1))k\delta_1, & t_1 < t \leq t_2 \\ 0, & t > t_2 \end{cases}, \tag{5.2.4}$$

where $0 < k \leq 1$ is the control gain, t_1 is the moment when the control of subsystem (4.1.3), i.e. u_2, starts working in its linear region, which satisfies either of the following two sets of conditions:

$$\begin{cases} t_0 \leq t_1 \leq t_d, & w(t_1) \in \Omega_1, & x_1(t_1) \geq d \\ t_1 > t_d, & w'(t_1) \in \Omega_2, & x_1(t_1) \leq -d \end{cases}, \tag{5.2.5}$$

where t_d, d, w, w', Ω_1, Ω_2 will be defined later, and t_2 is the time when the control goal is considered to be accomplished and it can be quantified as:

$$t_2 = t_1 + \frac{|x_1(t_1)|}{k\delta_1}.$$

The control design of u_2 for the case of $x_1(t_0) \geq 0$ is to be discussed and stability results will be proved. For the case of $x_1(t_0) < 0$, one can always make it positive by redefining the following coordinate system $x_i'(t) = (-1)^i x_i(t),\ i = 1, 2, \cdots, n$, resulting in a new chained system. Under this condition, control u_2 is proposed to be:

$$u_2(t) = \begin{cases} -\sum_{i=1}^{n-1} \epsilon_i \,\mathrm{sat}(\frac{k\delta_1 w_i}{\epsilon_i}), & t_0 \leq t \leq t_1 \\ -\sum_{i=1}^{n-1} \epsilon_i \,\mathrm{sat}(\frac{k\delta_1 w_i}{\epsilon_i}), & t_1 < t \leq t_2,\ t_1 \leq t_d \\ -\sum_{i=1}^{n-1} \epsilon_i \,\mathrm{sat}(\frac{k\delta_1 w_i'}{\epsilon_i}), & t_1 < t \leq t_2,\ t_1 > t_d \\ 0, & t > t_2 \end{cases}, \tag{5.2.6}$$

where

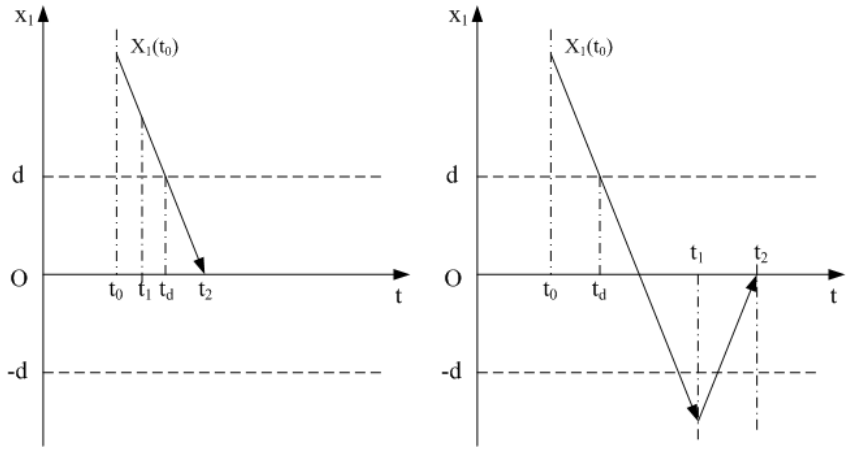
$$w_{n-1-i} = \sum_{j=0}^{i} \begin{pmatrix} i \\ j \end{pmatrix} y_{n-1-j}, \quad i = 0, 1, \cdots, n-2, \tag{5.2.7}$$

with $y_j = (-1)^{n-1-j} z_j,\ j = 1, 2, \cdots, n-1$, and

$$w'_{n-1-i} = \sum_{j=0}^{i} \begin{pmatrix} i \\ j \end{pmatrix} z_{n-1-j}, \quad i = 0, 1, \cdots, n-2. \tag{5.2.8}$$

Also, ϵ_i satisfies condition (5.2.3) with $i = 1, 2, \cdots, n-1$ and $u_{\max} = \delta_2$. The definition of t_d is illustrated in the following figures.

In Fig. 5.1, a buffer zone $-d \leq x_1 \leq d$ is created, with $d > 0$ as a design parameter. The purpose of creating this buffer zone is to ensure a sufficient amount of time so that subsystem



(a) (b)

Figure 5.1: Two Cases of Controls: (a), $t_1 \leq t_d$; (b), $t_1 > t_d$.

(4.1.3) can maneuver after u_2 and gets into its linear region. Because in the linear region, subsystem (4.1.3) converges with fixed eigenvalues. t_d is defined to be the time when x_1 first gets into the buffer zone, and it is quantified by:

$$t_d = \begin{cases} \frac{x_1(t_0)-d}{k\delta_1}, & x_1(t_0) > d \\ 0, & 0 \leq x_1(t_0) \leq d \end{cases}.$$

t_1 is determined by the following process. Define two sets:

$$\Omega_1 = \{w : |w_1| \leq \frac{\epsilon_1}{k\delta_1}, |w_2| \leq \frac{\epsilon_2}{k\delta_1}, \cdots, |w_{n-1}| \leq \frac{\epsilon_{n-1}}{k\delta_1}\},$$

and

$$\Omega_2 = \{w' : |w'_1| \leq \frac{\epsilon_1}{k\delta_1}, |w'_2| \leq \frac{\epsilon_2}{k\delta_1}, \cdots, |w'_{n-1}| \leq \frac{\epsilon_{n-1}}{k\delta_1}\}.$$

t_1 is defined to be the earliest time when either condition of (5.2.5) is met. If the first condition of (5.2.5) is satisfied, then the second condition is ignored. Otherwise, continue to apply the control (4.2.1) and (4.2.8) until the second set of conditions of (5.2.5) are met.

The existence of a finite t_1 will be proved in Theorem 7. Note that if the initial condition is $0 \leq x_1(t_0) < d$, then the first condition of (5.2.5) is not possible to be meet. The asymptotic stability of the proposed control is proved in the following theorem.

Theorem 6. *Control (4.2.1) and (4.2.8) are asymptotic stable controls for the chained system (4.1.1) while satisfying the bound condition (5.1.1).*

Proof. Consider subsystem (4.1.2), since $0 < k \leq 1$, obviously u_1 satisfies $|u_1| \leq \delta_1$. Moreover, no matter where $x_1(t_1)$ is,

$$\begin{aligned} x_1(t_2) &= x_1(t_1) + u_1 \times (t_2 - t_1) \\ &= x_1(t_1) - \text{sign}(x_1(t_1))k\delta_1 \times \frac{|x_1(t_1)|}{k\delta_1} \\ &= 0. \end{aligned}$$

Under the choice of u_1 in (4.2.1), subsystem (4.1.3) becomes a chain of constant weighted integrators:

$$\begin{cases} \dot{z}_1 &= -k\delta_1 z_2 \\ \dot{z}_2 &= -k\delta_1 z_3 \\ &\vdots \\ \dot{z}_{n-2} &= -k\delta_1 z_{n-1} \\ \dot{z}_{n-1} &= u_2 \end{cases}.$$

Or it can be expressed as y-system:

$$\begin{cases} \dot{y}_1 &= k\delta_1 y_2 \\ \dot{y}_2 &= k\delta_1 y_3 \\ &\vdots \\ \dot{y}_{n-2} &= k\delta_1 y_{n-1} \\ \dot{y}_{n-1} &= u_2 \end{cases}. \tag{5.2.9}$$

For the y-system, by the transformation (5.2.7), the resulted system is:

$$\dot{w} = A_y w + B_y u_2, \tag{5.2.10}$$

where

$$A_y = \begin{bmatrix} 0 & k\delta_1 & \cdots & k\delta_1 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & k\delta_1 \\ 0 & \cdots & \cdots & 0 \end{bmatrix}, \quad B_y = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}.$$

By Theorem 5, u_2 for $t \in [t_0\ t_1]$ is a stabilizing control for system (5.2.10), which satisfies $|u_2| \leq \delta_2$. Therefore $\lim_{t\to\infty} w(t) \to 0$. If $w(t_1) \in \Omega_1$ for some $t_1 \in [t_0\ t_d]$, it indicates that u_2 reaches its linear region before $x_1(t)$ gets into the buffer zone, and remains linear thereafter by Corollary 1. Since the linear system is converging with fixed eigenvalues, by choosing a relatively large d would ensure subsystem (2) to be stabilized. If $w(t) \notin \Omega_1$ for all $t \in [t_0\ t_d]$, this means $t_1 > t_d$. Then the current controls are kept until $x_1(t_1) \leq -d$ and $w'(t_1) \in \Omega_2$. In this case, control u_1 and subsystem (4.1.3) for $t_1 < t \leq t_2$ becomes:

$$u_1 = k\delta_1, \tag{5.2.11}$$

and

$$\begin{cases} \dot{z}_1 &= k\delta_1 z_2 \\ \dot{z}_2 &= k\delta_1 z_3 \\ &\vdots \\ \dot{z}_{n-2} &= k\delta_1 z_{n-1} \\ \dot{z}_{n-1} &= u_2 \end{cases}. \tag{5.2.12}$$

By transformation (5.2.8), subsystem (4.1.3) becomes:

$$\dot{w}' = A_y w' + B_y u_2, \tag{5.2.13}$$

By Theorem 5, control u_2 is a stabilizing control and since $w'(t_1) \in \Omega_2$, the control is linear for $t \geq t_1$. The choice of d will guarantee the closeness to the origin at t_2. □

Theorem 7. *A finite t_1 always exists for the control (4.2.1) and (4.2.8) that satisfies the condition (5.2.5).*

Proof. It is obtained that:

$$\begin{aligned} x_1(t_1) &= x_1(t_0) - k\delta_1 \times (t_1 - t_0) \\ &= x_1(t_0) + k\delta_1 \times t_0 - k\delta_1 \times t_1 \end{aligned}$$

Under control (4.2.8), suppose there is no limitation of t_1, then $\lim_{t\to\infty} \|w(t)\| \to 0$, hence there is a finite t_1 such that $w(t_1) \in \Omega_1$. If this t_1 satisfies $t_0 \leq t_1 \leq t_d$, then the first set of conditions of (5.2.5) are satisfied. Otherwise, The condition for $x_1(t_1) \leq -d$ is:

$$t_1 \geq t'_d \triangleq t_0 + \frac{x_1(t_0) + d}{k\delta_1}.$$

Since $\lim_{t\to\infty} \|w(t)\| \to 0$, therefore $\lim_{t\to\infty} \|z(t)\| \to 0$. By transformation (5.2.8), $\lim_{t\to\infty} \|w'(t)\| \to 0$, hence there exists a finite time t'_w such that $w'(t'_w) \in \Omega_2$. Then t_1 can be chosen as: $t_1 = \max\{t'_d,\ t'_w\}$. □

5.2.2 Choice of k and d

In order to meet the saturation condition, the design parameter k is restricted by $0 < k \leq 1$. Intuitively, k should be chosen large. Because with a larger k, the connections among the states of subsystem (4.1.3) are stronger and the magnitude of control u_2 tends to be larger (within the saturation bound). This contributes to a faster convergence rate before the controller reaches the linear region. Moreover, in the linear operation region Ω_1 and Ω_2, the

closed loop system of subsystem (4.1.3) in transformed space becomes:

$$\dot{w} = A_c w, \tag{5.2.14}$$

where

$$A_c = \begin{bmatrix} -k\delta_1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ -k\delta_1 & \ddots & -k\delta_1 & 0 \\ -k\delta_1 & -k\delta_1 & \cdots & -k\delta_1 \end{bmatrix}.$$

It shows that $-k\delta_1$ is the $(n-1)$th order eigenvalue, k also decides the convergence rate when the controls work in the linear operation regions. So, where convergence speed is concerned, k needs to be chosen as large as possible, i.e. $k = 1$.

For the choice of d, notice that when $|x_1| \leq d$, the w or w'-system must be in its linear region. The primary concern is d should be large enough that when x_1 reaches 0, w (or w') is small. It follows from (5.2.14) that the Laplace transformation of the state transition matrix is:

$$\mathcal{L}(e^{A_c t}) = (sI - A_c)^{-1} = \begin{bmatrix} \frac{1}{s+k\delta_1} & 0 & \cdots & 0 & 0 \\ \frac{-k\delta_1}{(s+k\delta_1)^2} & \frac{1}{s+k\delta_1} & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ \frac{-k\delta_1 s^{n-4}}{(s+k\delta_1)^{n-2}} & \ddots & \ddots & \frac{1}{s+k\delta_1} & 0 \\ \frac{-k\delta_1 s^{n-3}}{(s+k\delta_1)^{n-1}} & \frac{-k\delta_1 s^{n-4}}{(s+k\delta_1)^{n-2}} & \cdots & \frac{-k\delta_1}{(s+k\delta_1)^2} & \frac{1}{s+k\delta_1} \end{bmatrix}.$$

Therefore,

$$e^{A_c t} = \begin{bmatrix} e^{-k\delta_1 t} & 0 & \cdots & 0 & 0 \\ -k\delta_1 t e^{-k\delta_1 t} & e^{-k\delta_1 t} & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ \sum_{i=2}^{n-2}(-1)^{i-1}\frac{(k\delta_1 t)^{i-1}}{(i-1)!}e^{-k\delta_1 t} & \ddots & \ddots & e^{-k\delta_1 t} & 0 \\ \sum_{i=2}^{n-1}(-1)^{i-1}\frac{(k\delta_1 t)^{i-1}}{(i-1)!}e^{-k\delta_1 t} & \sum_{i=2}^{n-2}(-1)^{i-1}\frac{(k\delta_1 t)^{i-1}}{(i-1)!}e^{-k\delta_1 t} & \cdots & -k\delta_1 t e^{-k\delta_1 t} & e^{-k\delta_1 t} \end{bmatrix}.$$

With the information of k and δ_1, one can solve for the time T_m that is needed for maneuvering in the linear region. Then d is obtained by $d \geq k\delta_1 T_m$. For example, with the choice $k = 1$ and the saturation bound $\delta_1 = \delta_2 = 1$, the state transition matrix for a chained system with $n = 3$ is:

$$e^{A_c t} = \begin{bmatrix} e^{-t} & 0 \\ -te^{-t} & e^{-t} \end{bmatrix}.$$

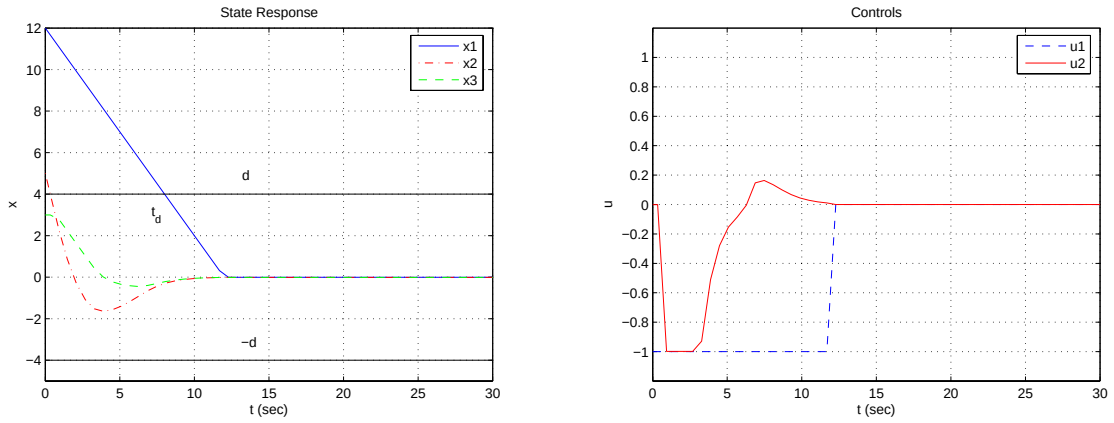
If one chooses $T_m = 4$ or $T_m = 5$, the final state is around 7% or 3% of its value when the state entered in the linear operation region Ω_1 and Ω_2.

5.3 Simulations

In this section, simulation results for the proposed control are presented. The simulation is conducted on a chained system with $n = 3$. The saturation limit is chosen to be $\delta_1 = \delta_2 = 1$, the gain parameter for u_1 is $k = 1$ and d is set to be $d = 4$ as discussed in Section 5.2.2. Satisfying the condition (5.2.3), ϵ_1 and ϵ_2 are chosen to be $\epsilon_1 = 0.499$ and $\epsilon_2 = 0.5$. To illustrate the two types of control actions, two sets of initial conditions are selected in the

simulation. The results for both cases show that the proposed control is successful under the saturation condition.

In the first case, the initial condition is set to be $x(t_0) = [12\ 5\ 3]$. Then, it can be obtained that $t_d = 8$. By running the simulation, it is obtained that $t_1 = 6.2146$ and $t_2 = 12$. The simulation results for this case are shown in Fig. 5.2. Fig. 5.2(a) shows the state response, since $t_1 < t_d$, subsystem (4.1.3) reaches the linear region Ω_1 before x_1 gets into the buffer zone $[-d\ d]$. Therefore, the controller knows the time for maneuvering subsystem (4.1.3) is sufficient. Hence when x_1 reaches 0, the controls stop.

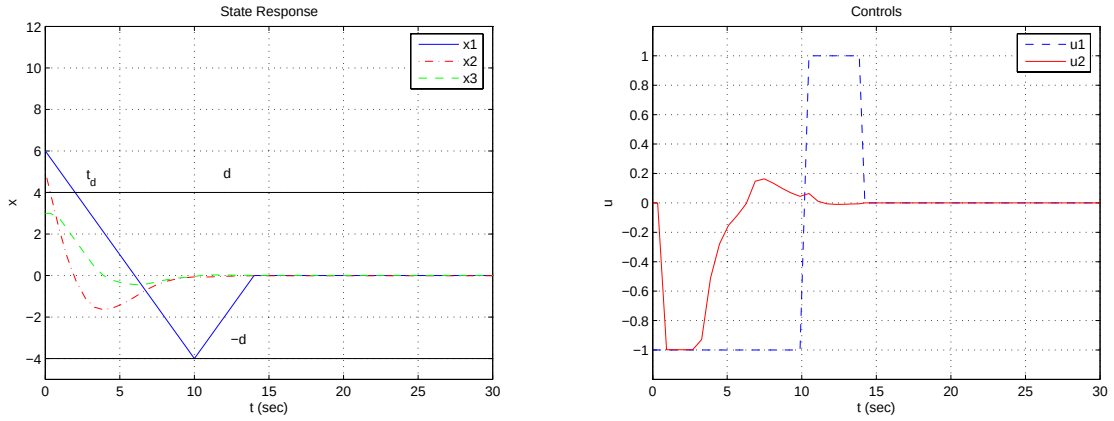


(a) (b)

Figure 5.2: State and Controls for The Case $t_1 \leq t_d$. (a), State; (b), Control.

In the second case, the initial condition is set to be $x(t_0) = [6\ 5\ 3]$. Therefore $t_d = 2$. It is obtained from the simulation that $t_1 = 10.0012$ and $t_2 = 14.0024$. The simulation results for this case are shown in Fig. 5.3. Fig. 5.3(a) shows the state response, since $t_1 > t_d$, subsystem (4.1.3) reaches the linear region Ω_1 later than x_1 gets into the buffer zone $[-d\ d]$.

Therefore, the controller thinks the time for maneuvering subsystem (4.1.3) is not sufficient. Hence it steers x_1 cross 0 until w' gets into the linear region Ω_2 then steers x_1 back to 0.



(a) (b)

Figure 5.3: State and Control for The Case $t_1 > t_d$. (a), State; (b), Control.

Remark 5.3.1. *The controls proposed in (4.2.1) and (4.2.8) can be roughly verified by the daily experience of parking a car. A car is a 4th order nonholonomic system. x_1 is the displacement from the parking position and u_1 relates to its linear velocity. Subsystem (4.1.3) is its orientation and u_2 is its angular control. When the car's initial position is far away from the parking position, one usually can drive directly to the parking position. The car's body angle can be aligned without difficulties and no more maneuvers are needed. However, when the car's initial position is close to the parking position, it might not be feasible to get to the parking position while aligning the car's body angle at the same time. Therefore a straightforward solution would be to slightly get beyond the parking position for aligning the body angle and then back into the parking position.*

5.4 Conclusion

In this chapter, we studied the feedback stabilization problem of chained nonholonomic systems with input constraints, and a switching control design scheme is proposed. The essential idea is that by making u_1 to be piece-wise constant, subsystem (4.1.3) becomes multiple integrators that have a constant weight u_1. Then, a state transformation is applied to convert the multiple-integrator system into a linear system with an upper triangular system matrix, based on which the saturated control is obtained. Simulation study shows the effectiveness of the proposed control.

CHAPTER 6
OPTIMAL REAL-TIME COLLISION-FREE MOTION PLANNING FOR NONHOLONOMIC AUVS IN A 3D UNDERWATER SPACE

This chapter presents one approach to designing an optimal real-time collision-free trajectory for autonomous underwater vehicles (AUVs) that move in a 3D unknown underwater space. By explicitly considering the kinematic model of AUVs, a class of feasible trajectories is derived in a closed form, and is expressed in terms of two adjustable parameters for the purpose of collision avoidance. Then, a collision avoidance criteria is derived to determine a class of collision-free trajectories. Finally, a performance index is established to find an optimal trajectory from the class. All the steps can be implemented in real-time. The advantages of the proposed approach are: 1) The 3D motion planning problem is reduced to a 2D problem. Instead of directly searching in a 3D space, one only needs to determine two parameters in their plane. Therefore, computational efforts are greatly reduced, which is suitable for real-time implementation; 2) The vehicle's kinematic model is explicitly considered, and all boundary conditions are met. After the parameters are determined, the trajectory and controls are explicitly solved in closed forms.

The study of unmanned, untethered, free-swimming autonomous vehicles has been an active research topic in recent years due to their wide practical applications, such as ocean observations, deep-sea rescue, mineral and oil exploration, bathymetric surveys, sunken ship salvage, protection and cultivation of fishery resources [50, 1, 80, 13, 4, 71].

In many practical applications, it is desirable that the vehicles are able to explore within an uncertain environment, as complete environmental information cannot be assumed a priori. Under this background, the motion planning is of key importance for vehicles to successfully carry out various missions. The goal of motion planning is to generate the desired trajectory to be fed into the motion control system so that the vehicle executes or tracks the desired trajectory. Fig. 6.1 illustrates how this functionality can be implemented for an AUV. The higher level mission management module usually supplies waypoint information to the motion planner remotely or pre-stores it into the onboard system. The motion planner retrieves the waypoints, generates a desired trajectory, which includes the desired position profile X_d and velocity profile V_d. X_d and V_d, is fed into the kinematic controller to obtain a reference velocity V_r, then the torque-level tracking control can be designed through backstepping techniques [71, 60]. In this chapter, we will focus on designing an optimal, real-time,

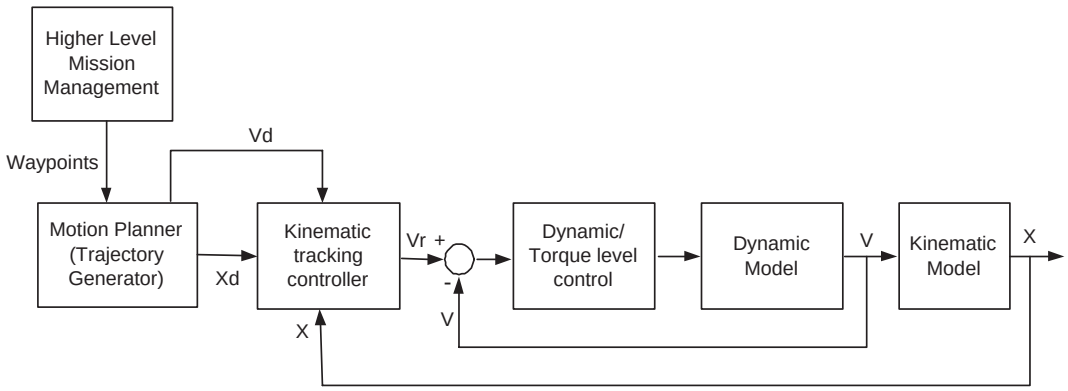


Figure 6.1: A Control Block Diagram for An Underwater Vehicle

and collision-free motion planner for AUVs that operate in an unknown underwater space.

The popular methods for motion planning include reactive approach, trajectory parameterizations and exhaustive search.

1. The reactive approach was pioneered in [28], its basic idea is to assign potential fields to obstacles to expel the trajectory away from obstacles and bring the trajectory to the final destination. To illustrate this idea, consider the repulsive potential field:

$$U(r) = \frac{1}{r^2}.$$

The attractive potential field is defined as:

$$U(r') = r'^2,$$

where r, r' are the corresponding distances to the obstacle and goal. A robot is to reach its goal along the gradient direction of its overall potential, that is,

$$U(r, r') = U(r) + U(r') = \frac{1}{r^2} + r'^2.$$

This scalar field has local minimums close to the goal point. If the robot approaches the minima, it will become stuck. When multiple obstacles are injected into the scenario, the potential becomes more complicated. Follow-up work can be found in [13, 4, 73, 20, 33, 74, 11]. These results only address the 2D problem. While 3D planning can be similarly done, it requires much more computation efforts. Also, the reactive approach generally suffers from local minima.

2. The parametric methods includes [26, 56, 75, 76]. In [26], a set of splines are adopted to form a path through a sequence of waypoints. However, prior information of the

waypoints might not be available because the environment could be unknown. Moreover, the kinematic constraints of the robots are not taken into consideration in the splines. Therefore, a trajectory may not be applicable for a specific robot. A common cubic spline method, each section of the path could be described by the parametric equations:

$$\begin{aligned} x(u) &= a_x u^3 + b_x u^2 + c_x u + d_x \\ y(u) &= a_y u^3 + b_y u^2 + c_y u + d_y, \end{aligned}$$

where $u \in [0 \;\; 1]$. This type of parameterization concentrates on the smooth property at the connection of various segments, rather than the kinematic constraints of the robot. The trajectory obtained by this method in this situation may not be feasible for specific types of robots. In [56], trajectories are parameterized by polynomials, then the coefficients are determined by fitting the kinematic model and boundary conditions. In [75], an optimal solution of [56] is discussed. Also, only 2D cases are addressed in [26, 56, 75].

3. In search based methods, A* (proposed in [53]) utilizes a heuristic function to guide the search direction to the goal, thus making it more efficient than the Dijkstra algorithm and guarantees an optimal path from the starting position to the ending position can be found, if one exists. However it requires the complete map information. To handle the dynamic environments, it needs a complete recalculation every time the map information is updated, causing it inefficient. A typical heuristic index used in

A* is:

$$f(n) = h(n) + g(n),$$

where $f(n)$ is the overall cost for a node, $h(n)$ is the cost already spent from the initial node to the current node, and $g(n)$ is the estimated cost from the current node to the end node. Generally $g(n)$ can be taken as the Euclidean distance between the current and end nodes. One improvement of the A* approach is found as D* (presented in [65] and [66]). The D* searching algorithm does not require the complete map information. It starts with an apriori map and at each time the map data is updated, it invokes a localized A* search to make incremental changes to the path. Its performance is compromised relative to the performance of the A* search. Both searching algorithms require much computational resources and do not take kinematic models into consideration.

By acknowledging the limitations of the existing techniques, we can improve on these methods by leveraging this information and creating a motion planning approach for an AUV in a 3D space. The trajectories are parameterized by polynomials. By allowing two parameters to be adjustable, the 3D problem is reduced to a 2D problem. The vehicle's kinematic model is explicitly taken into consideration and controls can be solved analytically, which are suitable for real-time implementation. Moreover, as long as collision does not occur at the boundary conditions the collision avoidance condition is always solvable.

6.1 Problem Formulation

A torpedo-shaped AUV model is shown in Fig. 6.2. Two reference frames are set up, one is the world frame $O - XYZ$, the other is a body frame $c - xyz$ which is attached to the center of gravity (CG) of the AUV with x axis along the longitudinal direction and z axis pointing along OZ direction. Both frames follow the right-hand rule.

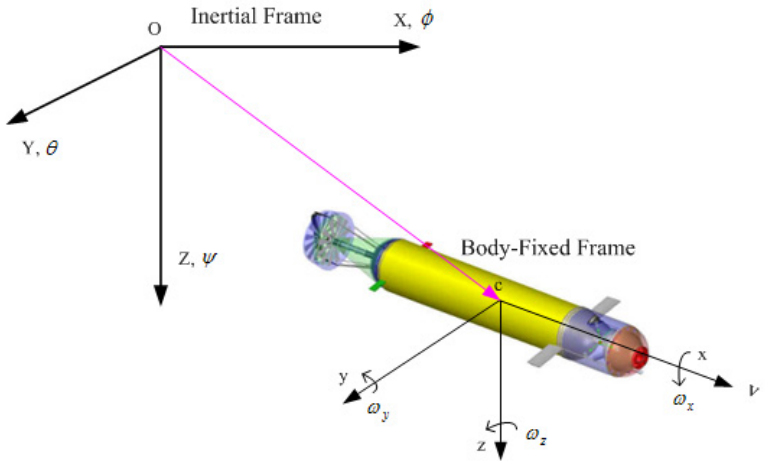


Figure 6.2: One AUV Model

6.1.1 The Kinematic Model

In the inertial frame, the configuration vector $\mathbf{q} = (x,\ y,\ z,\ \phi,\ \theta,\ \psi)$ is used to specify the position and orientation of the AUV where (x, y, z) is the vehicle's CG and (ψ, θ, ϕ) are the $Z - Y - X$ Euler angles, with ψ as the yaw, θ as the pitch, and ϕ as the roll. The kinematic model is given by the following two equations:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} v\cos\theta\cos\psi \\ v\cos\theta\sin\psi \\ -v\sin\theta \end{bmatrix}, \tag{6.1.1}$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}, \tag{6.1.2}$$

where $\theta \neq \pm\frac{\pi}{2} + h\pi,\ h = 0, 1, \cdots$. v is the longitudinal velocity, ω_x, ω_y and ω_z are angular velocities around the body's three fixed axises. The motion planning is based on the kinematic model (6.1.1) and (6.1.2), v, ω_x, ω_y, and ω_z are considered to be the kinematic level reference controls which are explicitly solved in equations (6.2.4) and (6.2.5).

6.1.2 The Trajectory Planning Problem

Fig. 6.3 shows a 3D underwater space with depth d. One AUV is moving from its initial condition $\mathbf{q}_0 = (x_0,\ y_0,\ z_0,\ \phi_0,\ \theta_0,\ \psi_0)$ at time t_0 with speed v_0 to the terminal condition $\mathbf{q}_f = (x_f,\ y_f,\ z_f,\ \phi_f,\ \theta_f,\ \psi_f)$ at time t_f with speed v_f. In general, the geometrical model of the AUV can be any shape that can be analytically described or a composite of multiple such shapes. For simplicity of derivation, we consider it to be the smallest sphere that contains the AUV. The center of the sphere is at its CG, and the radius is r_0. In fact, the size of the vehicles can be taken into obstacles, by modeling the obstacles to be a little"larger".

Therefore, one only needs the vehicle's CG position and the obstacles' positions to determine whether there will be collisions. The effective range of the onboard sensor is also modeled as a sphere centered at the CG with radius R_s. The trajectory planning problem is to search for a collision-free trajectory that satisfies the vehicle's boundary conditions and its kinematic model. The following assumptions are made to ensure the solvability and to simplify the technical development.

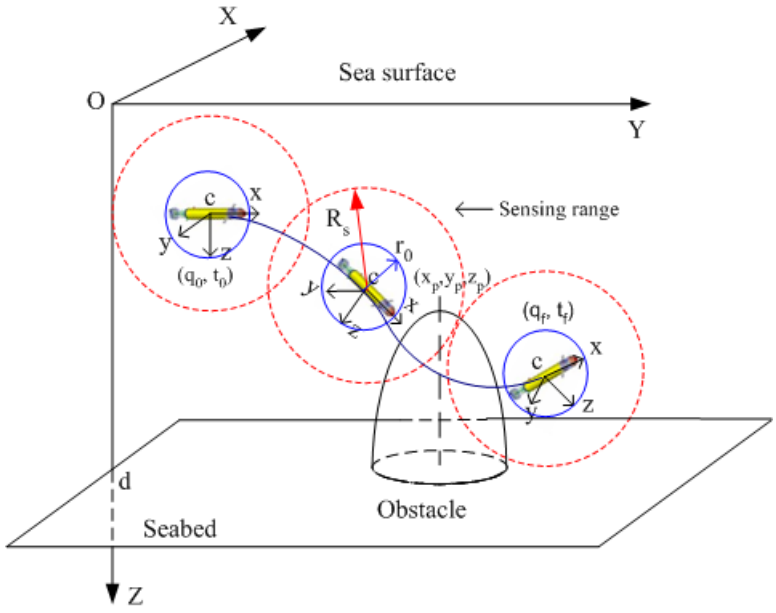


Figure 6.3: AUV Moving in an Unknown Environment

Assumption 1. *All obstacles have convex shapes.*

Assumption 2. *Boundary points do not locate inside any obstacle.*

Assumption 3. $x_f > x_0$, $-\frac{\pi}{2} < \psi_0$, $\psi_f < \frac{\pi}{2}$, *and the pitch angle is maintained in* $-\frac{\pi}{2} + \rho < \theta < \frac{\pi}{2} - \rho$ *with* $0 < \rho << 1$.

Assumption 1 is needed because if an obstacle is concave, and a point locates close to its concave area, it would be equivalent that this point is inside a convex object which is adopted to model this concave obstacle. Assumption 2 ensures the solvability of collision avoidance conditions, which will be discussed in Section 6.2.4. In Assumption 3, $x_f > x_0$, $-\frac{\pi}{2} < \psi_0,\ \psi_f < \frac{\pi}{2}$ can always be satisfied by inserting middle points and/or defining new coordinate systems. $-\frac{\pi}{2} < \theta < \frac{\pi}{2}$ is needed to avoid the representation singularity in (6.1.2), it is reasonable because it is very rare that an AUV would have its pitch angle exceed $\pm\frac{\pi}{2}$.

In underwater environments, an obstacle could be a hill growing from the seabed. It can be modeled by an elliptical paraboloid. In the inertial frame, the equation is:

$$z - z_p = \frac{(x - x_p)^2}{m^2} + \frac{(y - y_p)^2}{n^2}$$

where (x_p, y_p, z_p) is the peak position and m, n are parameters that could be modified to change its size. Define:

$$s(x, y, z) = \frac{(x - x_p)^2}{m^2} + \frac{(y - y_p)^2}{n^2} - (z - z_p).$$

Obviously, $s(x, y, z) > 0$ indicates that the point (x, y, z) locates outside an obstacle, while $s(x, y, z) \leq 0$ implies it locates inside or on the surface of an obstacle. Hence, a collision avoidance criterion could be:

$$s(x, y, z) > 0. \tag{6.1.3}$$

Suppose the vehicle's trajectory can be parameterized as $\mathbf{q} = \mathbf{q}(t, \mathbf{w})$, where $\mathbf{w}$ is a decision vector that can be chosen. Moreover, a performance index $J(\mathbf{q}, \dot{\mathbf{q}})$ can be established. It could be a measure of the length of the trajectory or the energy cost to maneuver on

the trajectory, etc. Therefore, the trajectory planning problem is finally formulated as the following optimization problem:

$$\begin{aligned} &\min_{\mathbf{w}} J(\mathbf{q}, \dot{\mathbf{q}}) \\ &s.t. \quad \mathbf{q}(t_0) = \mathbf{q}_0,\ \mathbf{q}(t_f) = \mathbf{q}_f \\ &\qquad s(x, y, z) > 0, \quad \forall t \in [t_0\ t_f]\ , \\ &\qquad M(\mathbf{q}, \dot{\mathbf{q}}) = 0 \end{aligned}$$

where $M(\mathbf{q}, \dot{\mathbf{q}})$ denotes the vehicle's kinematic model.

6.2 Real-Time Trajectory Planning For AUVs

In this section, the real-time trajectory planning problem is solved in three steps. In Section 6.2.1, trajectory planning is considered without obstacles. In Section 6.2.2, the parameterization is made piecewise and a condition to avoid the obstacles is developed. Section 6.2.3 gives an optimal solution to the parameters and Section 6.2.4 discusses the solution and solvability.

6.2.1 Trajectory Planning without Obstacles

The kinematic model given by equation (6.1.1) and (6.1.2) can be rewritten as

$$\begin{cases} \dot{x} &= v\cos\theta\cos\psi \\ \dot{y} &= v\cos\theta\sin\psi \\ \dot{z} &= -v\sin\theta \\ \dot{\phi} &= \omega_x + \dot{\psi}\sin\theta \\ \dot{\theta} &= \omega_y\cos\phi - \omega_z\sin\phi \\ \dot{\psi} &= \omega_y\sin\phi\sec\theta + \omega_z\cos\phi\sec\theta \end{cases} . \tag{6.2.1}$$

From boundary conditions, the following quantities can be determined: $x(t_0)$, $\dot{x}(t_0)$, $x(t_f)$, $\dot{x}(t_f)$, $y(t_0)$, $\frac{dy}{dx}|_{t_0}$, $y(t_f)$, $\frac{dy}{dx}|_{t_f}$, $z(t_0)$, $\dot{z}(t_0)$, $z(t_f)$, $\dot{z}(t_f)$, i.e. there are 4 boundary conditions available for each of x, y, and z. Hence, when polynomial parameterization is used, each polynomial at least needs 4 free coefficients (3rd order). If a higher order is chosen, redundant coefficients can be treated as the decision vector, which provides the freedom to choose trajectories. We parameterize the desired trajectory to be:

$$\begin{cases} x(t) &= a_0 + a_1 t + a_2 t^2 + a_3 t^3 \\ y(x) &= b_0 + b_1 x + b_2 x^2 + b_3 x^3 + b_4 x^4 \\ z(t) &= c_0 + c_1 t + c_2 t^2 + c_3 t^3 + c_4 t^4 \end{cases} . \tag{6.2.2}$$

By meeting the boundary conditions of model (6.2.1), and put the equations into a matrix form, the coefficients can be solved as:

$$\begin{cases} [a_0\ a_1\ a_2\ a_3]^T &= (\mathbf{B}_1)^{-1}\mathbf{Y}_1 \\ [b_0\ b_1\ b_2\ b_3]^T &= (\mathbf{B}_2)^{-1}(\mathbf{Y}_2 - \mathbf{A}_2 b_4) \\ [c_0\ c_1\ c_2\ c_3]^T &= (\mathbf{B}_3)^{-1}(\mathbf{Y}_3 - \mathbf{A}_3 c_4) \end{cases}, \tag{6.2.3}$$

where

$$\mathbf{B}_1 = \begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 \\ 0 & 1 & 2t_0 & 3t_0^2 \\ 1 & t_f & t_f^2 & t_f^3 \\ 0 & 1 & 2t_f & 3t_f^2 \end{bmatrix}, \quad \mathbf{A}_2 = \begin{bmatrix} (x_0)^4 \\ 4(x_0)^3 \\ (x_f)^4 \\ 4(x_f)^3 \end{bmatrix},$$

$$\mathbf{B}_2 = \begin{bmatrix} 1 & x_0 & (x_0)^2 & (x_0)^3 \\ 0 & 1 & 2x_0 & 3(x_0)^2 \\ 1 & x_f & (x_f)^2 & (x_f)^3 \\ 0 & 1 & 2x_f & 3(x_f)^2 \end{bmatrix}, \quad \mathbf{B}_3 = \mathbf{B}_1,$$

$$\mathbf{Y}_1 = \begin{bmatrix} x_0 \\ v_0 cos\theta_0 \cos\psi_0 \\ x_f \\ v_f cos\theta_f \cos\psi_f \end{bmatrix}, \quad \mathbf{Y}_2 = \begin{bmatrix} y_0 \\ \tan\psi_0 \\ y_f \\ \tan\psi_f \end{bmatrix},$$

$$\mathbf{Y}_3 = \begin{bmatrix} z_0 \\ v_0 \sin\theta_0 \\ z_f \\ v_f \sin\theta_f \end{bmatrix}, \quad \mathbf{A}_3 = \begin{bmatrix} (t_0)^4 \\ 4(t_0)^3 \\ (t_f)^4 \\ 4(t_f)^3 \end{bmatrix}.$$

It is clear from (6.2.3) that in the parameterization, we have chosen $\mathbf{w} = (b_4, c_4)$ to be the decision vector. By Assumption 3, $x_0 \neq x_f$, and in a practical mission, there would be $t_f > t_0$, hence the matrices $\mathbf{B}_1$, $\mathbf{B}_2$, $\mathbf{B}_3$ are all nonsingular and invertible. Therefore the coefficients of (6.2.2) are solvable.

Theorem 8. *The reference controls given by*

$$v = \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2} \tag{6.2.4}$$

and

$$\begin{cases} \omega_x &= -k\phi_0 e^{-kt} - \frac{(\ddot{y}\dot{x} - \dot{y}\ddot{x})\dot{z}}{(\dot{x}^2+\dot{y}^2)\sqrt{\dot{x}^2+\dot{y}^2+\dot{z}^2}} \\ \omega_y &= \frac{\ddot{z}(\dot{x}^2+\dot{y}^2) - \dot{z}(\dot{x}\ddot{x}+\dot{y}\ddot{y})}{\sqrt{\dot{x}^2+\dot{y}^2}(\dot{x}^2+\dot{y}^2+\dot{z}^2)} \cos(\phi_0 e^{-kt}) \\ & \quad + \frac{\ddot{y}\dot{x} - \dot{y}\ddot{x}}{\sqrt{(\dot{x}^2+\dot{y}^2)(\dot{x}^2+\dot{y}^2+\dot{z}^2)}} \sin(\phi_0 e^{-kt}) \\ \omega_z &= \frac{\ddot{y}\dot{x} - \dot{y}\ddot{x}}{\sqrt{(\dot{x}^2+\dot{y}^2)(\dot{x}^2+\dot{y}^2+\dot{z}^2)}} \cos(\phi_0 e^{-kt}) \\ & \quad - \frac{\ddot{z}(\dot{x}^2+\dot{y}^2) - \dot{z}(\dot{x}\ddot{x}+\dot{y}\ddot{y})}{\sqrt{\dot{x}^2+\dot{y}^2}(\dot{x}^2+\dot{y}^2+\dot{z}^2)} \sin(\phi_0 e^{-kt}) \end{cases}, \tag{6.2.5}$$

where $k > 0$ is a constant, steer the AUV along the trajectory (6.2.2).

Proof. It follows from equation (6.2.1) that

$$v = \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}.$$

Let the AUV always restore itself to be upright, then one can design:

$$\dot{\phi} = -k\phi,$$

hence:

$$\phi = \phi_0 e^{-kt}. \tag{6.2.6}$$

Under Assumption 3, the slope of polynomial trajectory $y(x)$ can not exceed $\pm\frac{\pi}{2}$, hence $\psi(t) \in (-\frac{\pi}{2}\ \frac{\pi}{2})$, therefore:

$$\psi = \arctan\frac{\dot{y}}{\dot{x}}. \tag{6.2.7}$$

The pitch angle is always maintained in $(-\frac{\pi}{2}\ \frac{\pi}{2})$, hence:

$$\theta = \arctan\frac{\dot{z}}{\sqrt{\dot{x}^2 + \dot{y}^2}}. \tag{6.2.8}$$

Substitute (6.2.6), (6.2.7) and (6.2.8) into (6.2.1) leads to (6.2.5).

□

6.2.2 Trajectory Planning with Obstacles

Exploring in an unknown environment requires the AUV to implement its trajectory planning algorithm in real-time to update the controls. This requirement can be implemented by a piecewise-constant parameterization. Suppose the total operation time is T from the initial configuration q^0 to its final configuration q^f, and the sampling period is T_s, so that $\bar{k} = T/T_s$ is an integer. For $k = 0$, the initial condition is q^0. For $\bar{k} > k > 0$, the initial condition is $q^k = (x^k, y^k, \theta^k, v^k)$, the final condition is always q^f. The path planning method described in the previous subsections can be applied by using the boundary conditions $q_k,\ k = 0, 1, \cdots$ and q_f for real-time replanning as k increases. In the following parts of this chapter, notations

with superscript k or subscript k indicate they are versions of the corresponding variables at the kth sampling period.

In parameterization (6.2.2), $x(t)$ can be determined uniquely by the boundary conditions, while the coefficients of $y(t)$ and $z(t)$ are in terms of b_4^k and c_4^k, hence a collision free trajectory can be obtained by adjusting the two parameters according to the collision avoidance criterion.

By substituting the trajectory (6.2.2) into (6.1.3), the collision avoidance criterion becomes:

$$\begin{aligned} f_3 c_4^k \;\; \leq \;\; & \frac{f_1^2}{n^2}(b_4^k)^2 + \frac{2f_1(f_2 - y_p)}{n^2} b_4^k + \frac{(f_2 - y_p)^2}{n^2} \\ & - \frac{(x - x_p)^2}{m^2} + z_p - f_4, \;\; \forall t \in [\underline{t}\ \bar{t}], \end{aligned} \tag{6.2.9}$$

where

$$\begin{aligned} f_1 &= x^4 - [1\ x\ x^2\ x^3](\mathbf{B}_2^k)^{-1}\mathbf{A}_2^k, \\ f_2 &= [1\ x\ x^2\ x^3](\mathbf{B}_2^k)^{-1}\mathbf{Y}_2^k, \\ f_3 &= t^4 - [1\ t\ t^2\ t^3](\mathbf{B}_3^k)^{-1}\mathbf{A}_3^k, \\ f_4 &= [1\ t\ t^2\ t^3](\mathbf{B}_3^k)^{-1}\mathbf{Y}_3^k. \end{aligned}$$

It is not necessary to check the collision avoidance condition (6.2.9) in all time domains. By projecting a 2D image of the obstacle onto the seabed, the largest potential collision region can be given, which is elliptical:

$$d - z_p = \frac{(x - x_p)^2}{m^2} + \frac{(y - y_p)^2}{n^2}.$$

Its solution is:

$$\begin{cases} x &= x_p + m\sqrt{d - z_p}\cos\alpha \\ y &= y_p + n\sqrt{d - z_p}\sin\alpha \end{cases},$$

where $\alpha \in [0,\ 2\pi]$. It indicates that $x \in [x_p - m\sqrt{d - z_p}\ x_p + m\sqrt{d - z_p}]$ and $y \in [y_p - n\sqrt{d - z_p}\quad y_p + n\sqrt{d - z_p}]$. Then the time interval $[\underline{t}\ \bar{t}] \subset [t_0\ t_f]$ could be solved when $x(t) \in [x_p - m\sqrt{d - z_p}\quad x_p + m\sqrt{d - z_p}]$. Only in this interval, the collision avoidance criterion needs to be checked.

Since the AUV cannot go beyond the sea surface or below the seabed, the following constraint could always be applied,

$$0 \le z(t) \le d, \quad \forall t \in [t_0\ t_f],$$

from which, the following inequality about c_4^k can be derived:

$$-f_4 \le f_3 c_4^k \le d - f_4, \quad \forall t \in [t_k\ t_f]. \tag{6.2.10}$$

It shows that if (b_4^k, c_4^k) is solvable, the choice is not unique, which would yield a family of trajectories.

6.2.3 Optimal Solution of Candidate Trajectories

Equation (6.2.2) parameterizes a family of trajectories by making (b_4^k, c_4^k) variable. Nevertheless, some choice of (b_4^k, c_4^k) may generate long detoured paths. A suitable performance index (PI) needs to be established to find an optimal choice of (b_4^k, c_4^k) that minimizes the

trajectory length. A straightforward PI is the arc length, which is:

$$J_k^{\circ}(b_4^k, c_4^k) = \int_{x_k}^{x_f} \sqrt{1 + \left(\frac{dy}{dx}\right)^2 + \left(\frac{dz}{dx}\right)^2} dx \tag{6.2.11}$$

However, no optimal solution of (b_4^k, c_4^k) can be solved analytically from (6.2.11). One has to search the (b_4^k, c_4^k) plane without any prior information and integrate the arc length numerically, which requires huge computational efforts. Here, we introduce an 'initial straight line' (ISL), which is the line segment that connects the starting position at kth sampling period and the goal. Its equation in a 3D space is given by:

$$\begin{cases} y_l &= K_y(x_l - x_k) + y_k \\ z_l &= K_z(x_l - x_k) + z_k \end{cases},$$

where (x_l, y_l, z_l) are coordinates of the ISL, with $x_k \leq x_l \leq x_f$, and

$$K_y = \frac{y_f - y_k}{x_f - x_k}, \quad K_z = \frac{z_f - z_k}{x_f - x_k}.$$

The PI can be established as:

$$J_k(b_4^k, c_4^k) = \int_{x_k}^{x_f} [(x - x_l))^2 + (y - y_l)^2 + (z - z_l)^2] dx, \tag{6.2.12}$$

where $x_l = x(t)$ is set. In essence, PI (6.2.12) measures the closeness of the trajectory to a straight line trajectory.

Theorem 9. *Under PI (6.2.12), the optimal solution of* (b_4^k, c_4^k) *is given by:*

$$\begin{cases} b_4^{k*} &= -\frac{p_2}{2p_1} \\ c_4^{k*} &= -\frac{p_4}{2p_3} \end{cases}, \tag{6.2.13}$$

and the optimal performance index is:

$$J_k^*(b_4^k, c_4^k) = p_5 - \frac{p_2^2}{4p_1} - \frac{p_4^2}{4p_3},$$

where

$$\begin{aligned}
p_1 &= \int_{x_k}^{x_f} (f_1)^2 dx, \\
p_2 &= 2\int_{x_k}^{x_f} [f_1 f_2 - f_1 y_k - f_1 K_y(x - x_k)] dx, \\
p_3 &= \int_{x_k}^{x_f} (f_3)^2 dx, \\
p_4 &= 2\int_{x_k}^{x_f} [f_3 f_4 - f_3 z_k - f_3 K_z(x - x_k)] dx, \\
p_5 &= \int_{x_k}^{x_f} [f_2 - K_y(x - x_k) - y_k]^2 dx \\
&\quad + \int_{x_k}^{x_f} [f_4 - K_z(x - x_k) - z_k]^2 dx.
\end{aligned}$$

Proof. It follows from (6.2.12) that:

$$\begin{aligned}
J_k(b_4^k, c_4^k) &= \int_{x_k}^{x_f} (y - y_l)^2 dx + \int_{x_k}^{x_f} (z - z_l)^2 dx \\
&= \int_{x_k}^{x_f} [f_1 b_4^k + f_2 - K_y(x - x_k) - y_k]^2 dx \\
&\quad + \int_{x_k}^{x_f} [f_3 b_4^k + f_4 - K_z(x - x_k) - z_k]^2 dx \\
&= p_1 (b_4^k)^2 + p_2 b_4^k + p_3 (c_4^k)^2 + p_4 c_4^k + p_5 \\
&= p_1 (b_4^k + \frac{p_2}{2p_1})^2 + p_3 (c_4^k + \frac{p_4}{2p_3})^2 + p_5 - \frac{p_2^2}{4p_1} - \frac{p_4^2}{4p_3}. \qquad (6.2.14)
\end{aligned}$$

By Assumption 3, $x_f > x_0$, therefore $p_1, p_3 > 0$ (note that f_1, f_3 are both polynomials of finite order, the number of roots to zero is finite, hence $f_1 \equiv 0$ or $f_3 \equiv 0$ would not happen). Therefore the optimal PI is obtained at $b_4^{k*} = -\frac{p_2}{2p_1}$, $c_4^{k*} = -\frac{p_4}{2p_3}$, and its value is $J_k^*(b_4^k, c_4^k) = p_5 - \frac{p_2^2}{4p_1} - \frac{p_4^2}{4p_3}$. Note that by definition of PI (6.2.12), $J_k^*(b_4^k, c_4^k) \geq 0$. □

More detailed investigation shows that the contour of the PI is a series ellipses centered at (6.2.13). We already knew that $J_k^*(b_4^k, c_4^k)$ is the minimum, and it represents the point (b_4^{k*}, c_4^{k*}). Then for non-optimal (b_4^k, c_4^k), based on $J_k^*(b_4^k, c_4^k)$, we can introduce an incremental term $i\delta$, where $i = 1, 2, 3\cdots$, and $\delta > 0$ is a step size that can be chosen. It follows that:

$$J_k^*(b_4^k, c_4^k) + i\delta = p_1(b_4^k + \frac{p_2}{2p_1})^2 + p_3(c_4^k + \frac{p_4}{2p_3})^2 + p_5 - \frac{p_2^2}{4p_1} - \frac{p_4^2}{4p_3}.$$

It reduces to:

$$i\delta = p_1(b_4^k + \frac{p_2}{2p_1})^2 + p_3(c_4^k + \frac{p_4}{2p_3})^2. \tag{6.2.15}$$

Clearly, this is an elliptical equation, and its center is (b_4^{k*}, c_4^{k*}). All points on the ellipse has the same PI $J_k^* + i\delta$, so we call these ellipses contours, and by increasing i, the contours are expanded. Equation (6.2.15) can be transformed into the following form to calculate (b_4^k, c_4^k):

$$\begin{cases} b_4^k &= -\frac{p_2}{2p_1} + \sqrt{\frac{i\delta}{p_1}}\cos\alpha \\ c_4^k &= -\frac{p_4}{2p_3} + \sqrt{\frac{i\delta}{p_3}}\sin\alpha \end{cases}, \tag{6.2.16}$$

where $\alpha \in [0\ 2\pi)$ is an angle parameter, for example, in implementation, it can be supplied as $\alpha = \frac{j\pi}{180},\ j = 0, 1, 2, \cdots, 359$.

6.2.4 Solution and Solvability

In summary of the discussions in Section 6.2.1, 6.2.2, and 6.2.3, the steps to obtain a solution of (b_4^k, c_4^k) can be given as follows:

- **Step 1:** By solving equation (6.2.13), obtain the optimal solution of (b_4^{k*}, c_4^{k*}) without considering obstacles.

- **Step 2:** Consider the obstacle avoidance condition (6.2.9). In the (b_4^k, c_4^k) plane as shown in Fig. 6.4, only the points outside the parabola boundary satisfy (6.2.9). Substitute (b_4^{k*}, c_4^{k*}) in (6.2.9). If it holds, then the given solution is not only optimal, but also avoids the obstacle. In this case, the optimal point is illustrated by "*" in. If (6.2.9) does not hold, recall that the contour of the PI is a series of ellipses centered at (b_4^{k*}, c_4^{k*}), we expand the contour until the first point $(b_4^{k'}, c_4^{k'})$ that satisfies (6.2.9) is found, which is a suboptimal point with relatively low PI value. In this case, the optimal point and suboptimal point are marked by "x" and "+" respectively.

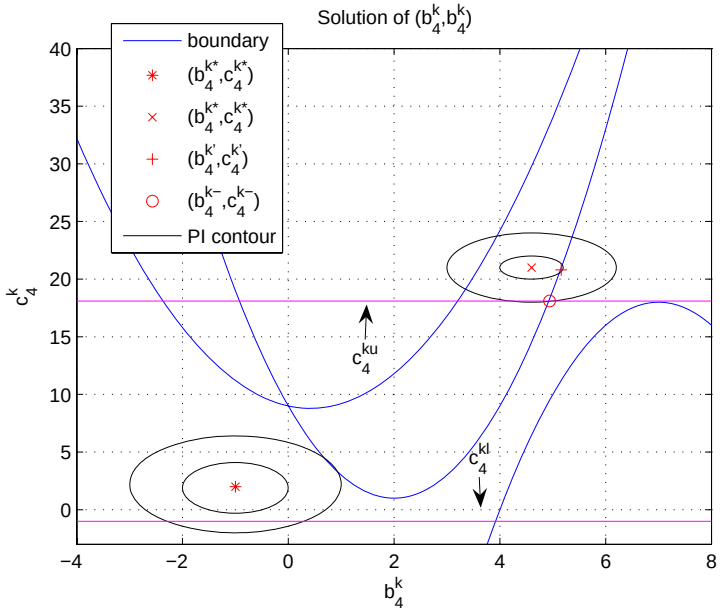


Figure 6.4: Solution of (b_4^k, c_4^k)

- **Step 3:** Check the constraint (6.2.10). It shows that no matter what sign f_3 is, the solution set of c_4^k is a closed interval. The intersection of solution sets over time could be empty or nonempty. In the case that it is nonempty, let it be $[c_4^{kl} \;\; c_4^{ku}]$. It denotes

a strip in the (b_4^k, c_4^k) plane. If the solution obtained in previous steps, i.e. (b_4^{k*}, c_4^{k*}) or $(b_4^{k'}, c_4^{k'})$, locates in the strip, then no further modification is needed. Otherwise the PI contour needs be enlarged again to find the point (b_4^{k-}, c_4^{k-}) which first gets into the strip. This point is the new solution of (b_4^k, c_4^k), and is marked by "o" in Fig. 6.4.

On the other hand, if the solution set is empty, then no matter what c_4^k is chosen, part of the trajectory is going to be above the sea surface or below the seabed. The case happens when the boundary points are above the sea surface or below the seabed, or the AUV stays too close to the sea surface while pointing upward or too close to the seabed while pointing downward. The only thing left that can be done is correcting the boundary conditions before planning a trajectory, i.e. moving the boundary points into the sea or adjusting the orientation angles so that the AUV points away from the sea surface or the seabed.

Theorem 10. *Under Assumption 2, the collision avoidance condition is always solvable, i.e. a collision-free trajectory always exists.*

Proof. Note that in (6.2.9), $\frac{f_1^2}{n^2} \geq 0$. If $f_1 \neq 0$, (6.2.9) represents all of the points outside a parabola in the (b_4^k, c_4^k) plane. In this case, a solution to (b_4^k, c_4^k) always exists for any single obstacle. In the presence of multiple obstacles, each obstacle imposes a constraint as (6.2.9), the final solution is the intersection of all solutions to every single object. It always yields at least one finite solution.

On the other hand, if $f_1 = 0$, b_4^k is removed from (6.2.9), then b_4^k no longer affects the collision avoidance. To understand the condition of $f_1 = 0$ and its implications, consider the

simpler case that $\bar{k} = 1$, it follows that:

$$\begin{aligned} y(x) &= b_0 + b_1 x + b_2 x^2 + b_3 x^3 + b_4 x^4 \\ &= [1\ x\ x^2\ x^3]\mathbf{B}_2^{-1}\mathbf{Y}_2 + b_4(x^4 - [1\ x\ x^2\ x^3]\mathbf{B}_2^{-1}\mathbf{A}_2) \\ &= [1\ x\ x^2\ x^3]\mathbf{B}_2^{-1}\mathbf{Y}_2 + b_4 f_1. \end{aligned}$$

The equation indicates that no matter what b_4 is picked, a trajectory reduces to a fixed cubic polynomial. This is impossible unless at the boundary points which do not vary with b_4. Hence $f_1 = 0$ only occurs at the boundary points. Also, it can be derived that $z(t) = [1\ t\ t^2\ t^3]\mathbf{B}_3^{-1}\mathbf{Y}_3 + c_4 f_3$. At the boundary points, $z(t_0)$ or $z(t_f)$ does not change with c_4, hence f_3 has to be 0. Then in the case of $f_1 = 0$ (i.e. at boundary points), there are only constant terms in (6.2.9). Under Assumption 2, the inequality must hold at the boundary points since there is no collision. Therefore, combined with the discussion for the case $f_1 \neq 0$, we know that collision avoidance condition (6.2.9) is always solvable. □

Remark 6.2.1. *If multiple obstacles are detected simultaneously, every obstacle imposes its own version of constraint (6.2.9) on the choice of* (b_4^k, c_4^k)*. Then if the optimal solution* (b_4^{k*}, c_4^{k*}) *locates inside any parabola area, the PI contour needs be expanded to find suboptimal* (b_4^k, c_4^k)*, and Theorem 10 guarantees that at least one finite* (b_4^k, c_4^k) *can be found to satisfy all inequalities (6.2.9) imposed by obstacles.*

In summary, the point denoted by "*" in Fig. 6.4 represents an ideal solution. It is optimal, collision free, and stays within the sea all the time. While "+" or "o" represents a suboptimal solution. It avoids the obstacle, stays within the sea, but the PI is enlarged, i.e. its trajectory length is longer.

6.3 Simulation Results

In this section, simulation results are presented for the proposed approach. The simulation is conducted for two scenarios, one has a single obstacle while the other has multiple obstacles.

6.3.1 Single Obstacle

A single-obstacle scenario is illustrated in Fig. 6.5. Settings of the single obstacle scenario are listed in Table 6.1. The simulation results are presented in Fig. 6.5 to Fig. 6.8.

Table 6.1: Settings of Single-Obstacle Scenario

Operation time (sec)	40
Sampling Period (sec)	1
Sensing Range (meter)	20
Scenario Scale (meter)	$100 \times 100 \times 100$
Initial Position (x_0, y_0, z_0)	$(10, 10, 10)$
Initial Attitude $(\phi_0, \theta_0, \psi_0)$	$(\pi/6, \pi/6, \pi/4)$
Initial Velocity (meter/sec)	5
Final Position (x_f, y_f, z_f)	$(90, 90, 90)$
Final Attitude $(\phi_f, \theta_f, \psi_f)$	$(0, -\pi/6, \pi/3)$
Final Velocity (meter/sec)	4

The obstacle settings are:

$$x_p = 50, \;\; y_p = 50, \;\; z_p = 36, \;\; m = 4, \;\; n = 2.$$

Fig. 6.5 shows two trajectories. The initial trajectory is planned without the knowledge of obstacles, hence, it passes through the obstacle. At the 11th second, the obstacle is detected, and by checking the collision avoidance condition, the AUV knows there will be collisions. Hence a new trajectory is replanned. The outcome of the proposed algorithm is:

$$\begin{cases} b_4^{0*} &= 3.217 \times 10^{-6} \\ c_4^{0*} &= -1.0616 \times 10^{-4} \end{cases}, \begin{cases} b_4^{11'} &= 9.4517 \times 10^{-6} \\ c_4^{11'} &= -7.5544 \times 10^{-4} \end{cases},$$

for the initial and replanned trajectories respectively.

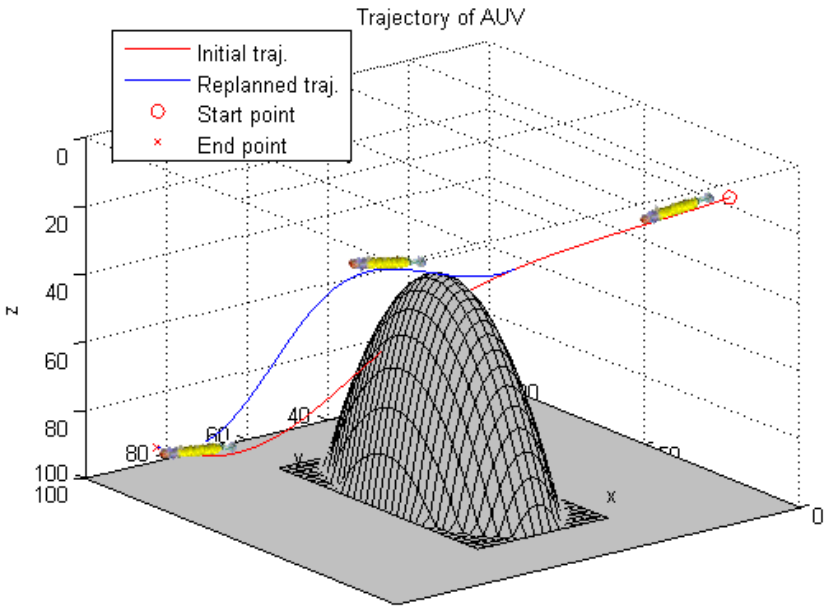


Figure 6.5: The Optimal Collision Free Trajectory

Remark 6.3.1. *Although the sampling period is* 1 *second, it does not mean the trajectory has to be replanned every second. The trajectory needs to be updated only when new obstacles are*

detected, and by checking the collision avoidance condition (6.2.9) imposed by each obstacle, a collision will happen if the AUV keeps the current trajectory.

Fig. 6.6 to Fig. 6.8 illustrate the state variables and inputs during the motion. The AUV successfully moved from its starting position to the ending position. The controls are piecewise continuous. The discontinuity comes from the second order derivatives in (6.2.5), as the kinematic model only ensures the continuity of the first order derivatives at the point when a trajectory is changed. As a result, the orientation angles are continuous but not differentiable at the moment when the trajectory is switched. The linear velocity is continuous in all time domains since it only relates to the first order derivatives of the trajectory. When backstepping technique is applied to find an exponential tracking control for the reference trajectory, it is anticipated that the torque level control is also a switching control.

6.3.2 Multiple Obstacles

A multiple-obstacle scenario is shown in Fig. 6.9, scenario settings are listed in Table 6.2. The obstacles' information is listed in Table 6.3.

The proposed method yields the trajectory illustrated in Fig. 6.9. It is composed of 3 segments. The AUV initially knew obstacle 1 and planned the first trajectory. While it was moving along segment 1, at the 12th second, obstacle 2 was detected. By checking the collision avoidance condition (6.2.9), the AUV knew there was going to be a collision, so it replanned the second trajectory and switched to segment 2, the rest of the first trajectory

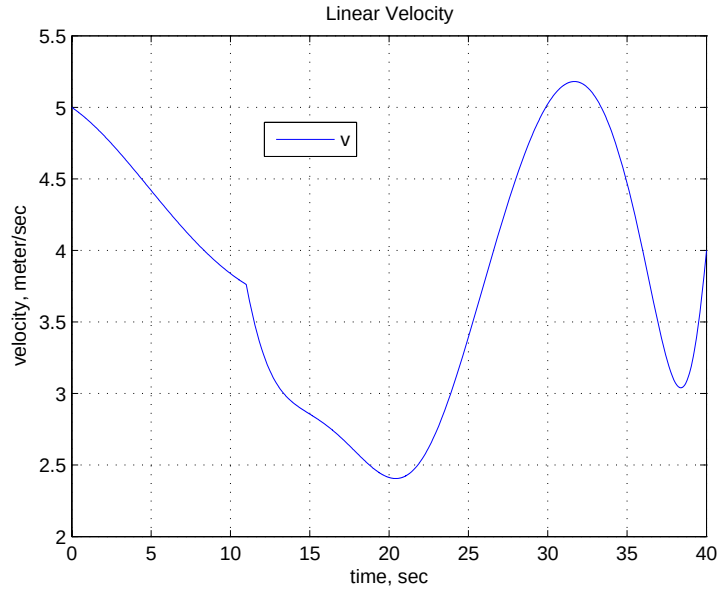


Figure 6.6: The Linear Velocity

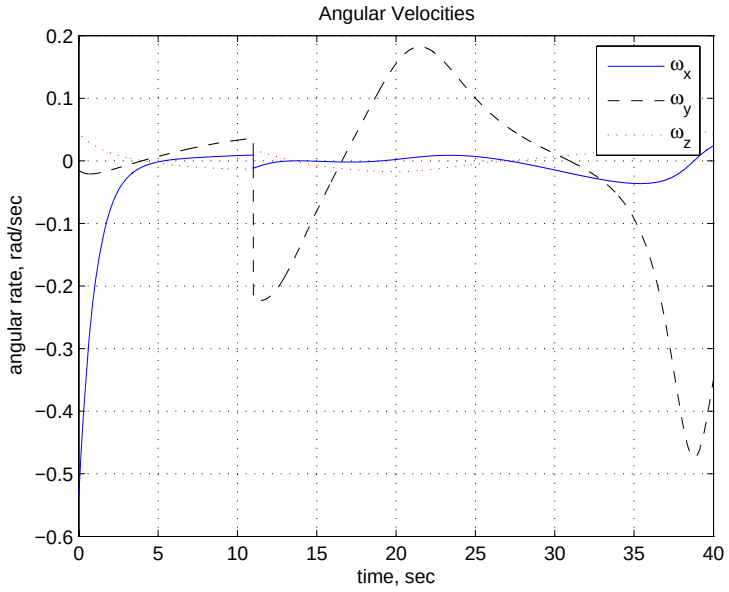


Figure 6.7: Angular Velocities

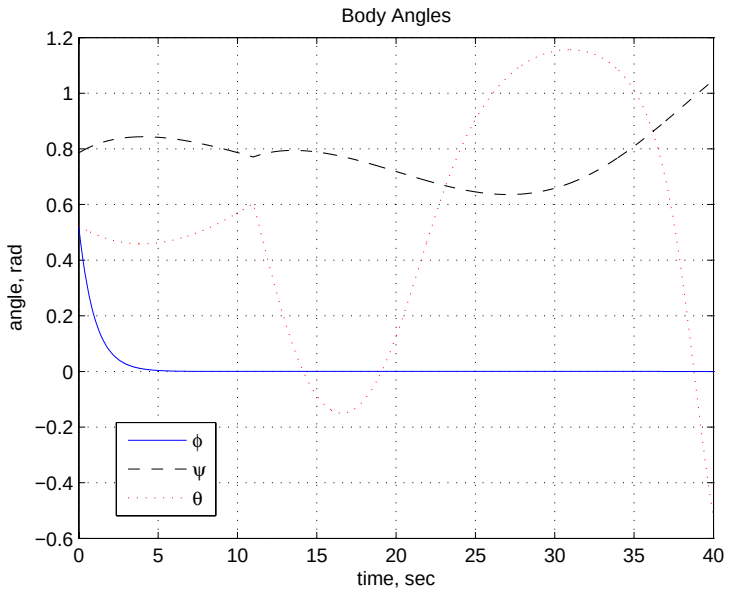


Figure 6.8: Orientation Angles

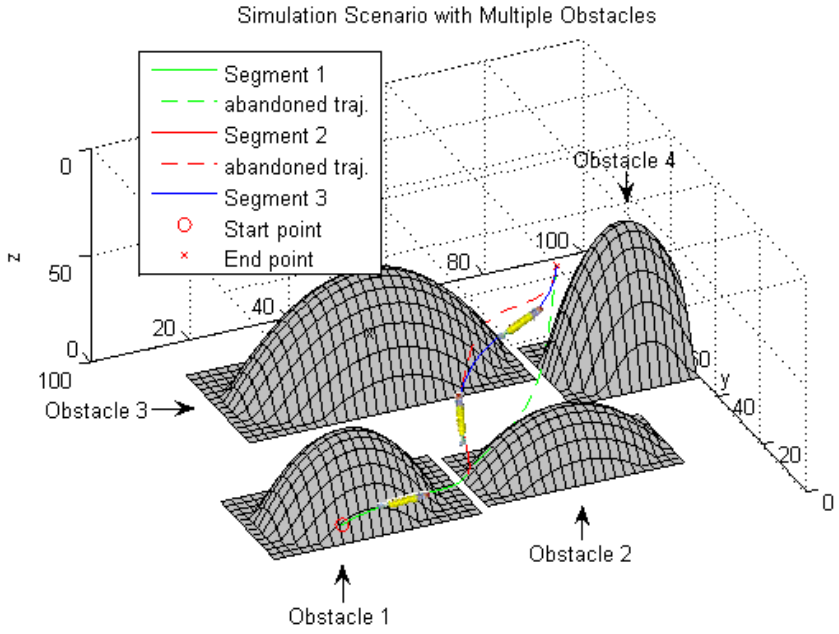


Figure 6.9: The Trajectory Avoids Obstacles

Table 6.2: Settings of Multi-Obstacle Scenario

Operation time (sec)	40
Sampling Period (sec)	1
Sensing Range (meter)	10
Scenario Scale (meter)	$100 \times 100 \times 100$
Initial Position (x_0, y_0, z_0)	$(10, 10, 80)$
Initial Attitude $(\phi_0, \theta_0, \psi_0)$	$(\pi/6, \pi/6, \pi/4)$
Initial Velocity (meter/sec)	5
Final Position (x_f, y_f, z_f)	$(90, 90, 90)$
Final Attitude $(\phi_f, \theta_f, \psi_f)$	$(0, -\pi/6, \pi/3)$
Final Velocity (meter/sec)	4

was abandoned. At the 25th second, it detected obstacle 3, also, it decided that a collision would happen, so it switched to segment 3 and abandoned the rest of the second trajectory. When obstacle 4 was detected, it decided that there was no threat, so no action was taken. The three trajectories yielded by the proposed approach are:

$$\left\{ \begin{array}{rcl} b_4^{0'} &=& -5.8844 \times 10^{-6} \\ c_4^{0'} &=& -1.5321 \times 10^{-4} \end{array} \right., \left\{ \begin{array}{rcl} b_4^{12'} &=& 2.423 \times 10^{-5} \\ c_4^{12'} &=& -0.0019 \end{array} \right., \left\{ \begin{array}{rcl} b_4^{25'} &=& 3.1149 \times 10^{-5} \\ c_4^{25'} &=& -1.9185 \times 10^{-4} \end{array} \right. .$$

6.4 Torque Level Tracking Control Of 3D trajectories

In this section we will discuss the dynamic trajectory tracking control for the planned optimal real-time collision free trajectory. The design is divided into two steps. In the first step, we

Table 6.3: Settings of The Obstacles

Obstacle	x_p	y_p	z_p	m	n
1	20	30	64	3	2
2	64	30	75	4	2
3	44	74	51	4	2
4	90	60	36	2	1.5

introduce a kinematic level tracking controller. In the second step, we propose a torque level control design via backstepping design.

6.4.1 The Kinematic Tracking Controller

In this section, a kinematic trajectory tracking control design is introduced from [50]. The reference trajectory to be tracked by the AUV is generated by the trajectory planning algorithm presented in Section 6.2. The reference variables are all solved in equations (6.2.2), (6.2.4), (6.2.5). The kinematic model given in (6.1.1) and (6.1.2) can be reorganized into the following form:

$$\begin{bmatrix} \dot{\bar{x}} \\ \dot{\bar{\theta}} \end{bmatrix} = \begin{bmatrix} S_1(\bar{\theta}) & 0_{3\times 3} \\ 0_{3\times 1} & S_2(\bar{\theta}) \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \tag{6.4.1}$$

where $\bar{x} = [x\ y\ z]^T$, $\bar{\theta} = [\phi\ \theta\ \psi]^T$, $u_1 = v$, $u_2 = [\omega_x\ \omega_y\ \omega_z]^T$, and

$$S_1(\bar{\theta}) = \begin{bmatrix} \cos\theta\cos\psi \\ \cos\theta\sin\psi \\ -\sin\theta \end{bmatrix}$$

$$S_2(\bar{\theta}) = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix}.$$

Let a feasible reference trajectory given by: $\bar{x}_d(t) = [x_d\ y_d\ z_d]^T$, $\bar{\theta}_d(t) = [\phi_d\ \theta_d\ \psi_d]^T$, and $u_{1d},\ u_{2d}$. It is feasible means $\bar{x}_d,\ \bar{\theta}_d,\ u_{1d},\ u_{2d}$ satisfies the kinematic model (6.4.1). Define the following error vectors, $\bar{x}_e$ is the difference between $\bar{x}_d$ and $\bar{x}$ as seen from the local frame;whereas $\bar{\theta}_e$ is simply as the difference between $\bar{\theta}_d(t)$ and $\bar{\theta}$. Thus:

$$\bar{x}_e = R^T(\bar{\theta})(\bar{x}_d - \bar{x})$$

$$\bar{\theta}_e = \bar{\theta}_d - \bar{\theta} = [\phi_e\ \theta_e\ \psi_e]^T$$

$$R(\bar{\theta}) = \begin{bmatrix} c\psi c\theta & c\psi s\theta s\phi - s\psi c\phi & c\psi s\theta c\phi + s\psi s\phi \\ s\psi c\theta & s\psi s\theta s\phi + c\psi c\phi & s\psi s\theta c\phi - c\psi s\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix}.$$

Then the error dynamic equations are:

$$\dot{x}_e = R^T(\theta)\dot{x}_d - R^T(\theta)\dot{x} + \dot{R}^T(x_d - x). \tag{6.4.2}$$

The following relationship can be easily verified:

$$R^T(\bar{\theta})S_1(\bar{\theta}) = R^T(\bar{\theta}_e)S_1(\bar{\theta}_e) = R^T(\bar{\theta}_d)S_1(\bar{\theta}_d) = [1\ 0\ 0]^T.$$

Since $R(\cdot)$ is orthonormal, we have:

$$S_1(\bar{\theta}_e) = R(\bar{\theta}_e)R^T(\bar{\theta}_d)S_1(\bar{\theta}_d) = [R(\bar{\theta}_e)R^T(\bar{\theta}_d)R(\bar{\theta})]R^T(\bar{\theta})S_1(\bar{\theta}_d).$$

If the error of attitude angle is small, then $R(\bar{\theta}_e)R^T(\bar{\theta}_d)R(\bar{\theta})$ can be approximated by an identity matrix and we have:

$$S_1(\bar{\theta}_e) = R^T(\bar{\theta})S_1(\bar{\theta}_d). \tag{6.4.3}$$

The physical implication of this approximation can be interpreted as following: if the desired vehicle attitude $R(\bar{\theta}_d)$ is not too different from the vehicle's current attitude $R(\bar{\theta})$, then the desired orientation as seen from the body frame ($R^T(\bar{\theta})R^T(\bar{\theta}_d)$) is approximately equal to the orientation $R(\bar{\theta}_e)$ obtained from the error of the Euler angles. Note that in a 2D case, $R^T(\bar{\theta})R^T(\bar{\theta}_d)$ is exactly same to $R(\bar{\theta}_e)$ and hence the relationship expressed by equation (6.4.3) is exact. From this discussion, and from equation (6.4.2), the error dynamics are:

$$\begin{cases} \dot{\bar{x}}_e &= S_1(\bar{\theta}_e)u_{1d} - R^T(\bar{\theta})S_1(\bar{\theta})u_1 - u_2 \times \bar{x}_e \\ \dot{\bar{\theta}}_e &= \dot{\bar{\theta}}_d - \dot{\bar{\theta}} = S_2(\bar{\theta}_d)u_{2d} - S_2(\bar{\theta})u_2 \end{cases} \tag{6.4.4}$$

We propose the following feedback tracking control:

$$\begin{cases} u_1 &= u_{1d} + u_{1d}(\cos\psi_e \cos\theta_e - 1) + \gamma^2 x_e \\ u_2 &= u_{2d} + S_2^{-1}(\bar{\theta})\{q + [S_2(\bar{\theta}_d) - S_2(\bar{\theta})]u_{2d} + p\} \end{cases}, \tag{6.4.5}$$

where the constant γ should be chosen appropriately, and

$$\begin{aligned} q &= [0 \;\; \frac{-z_e u_{1d}}{k_2} \;\; \frac{y_e u_{1d}\cos\theta_e}{k_3}]^T \\ p &= [k_1 \sin\phi_e \;\; k_2 \sin\theta_e \;\; k_3 \sin\psi_e]^T, \end{aligned}$$

with $k_1, k_2, k_3 > 0$. To show the convergence under control (6.4.5), take the Lyapunov function candidate:

$$V = \frac{1}{2}\bar{x}_e^T \bar{x}_e + \bar{k}^T f(\bar{\theta}_e),$$

where $\bar{k} = [k_1\ k_2\ k_3]^T$, and

$$f(\bar{\theta}_e) = [1 - \cos\phi_e\ 1 - \cos\theta_e\ 1 - \cos\psi_e]^T.$$

It follows that

$$\dot{V} = \bar{x}_e^T \dot{\bar{x}}_e + \bar{k}^T \frac{df}{d\bar{\theta}_e}\dot{\bar{\theta}}_e,$$

where

$$\frac{df}{d\bar{\theta}_e} = \begin{bmatrix} \sin\phi_e & 0 & 0 \\ 0 & \sin\theta_e & 0 \\ 0 & 0 & \sin\psi_e \end{bmatrix}.$$

Noting that $\bar{x}_e$ is orthogonal to $u_2 \times \bar{x}_e$, and $R^T(\bar{\theta})S_1(\bar{\theta}) = [1\ 0\ 0]^T$, $\dot{V}$ becomes:

$$\begin{aligned} \dot{V} &= p^T\{q + S_2(\bar{\theta}_d)u_{2d} - S_2(\bar{\theta})u_2\} - \gamma^2 x_e^2 \\ &= -p^T p - \gamma^2 x_e^2 \\ &\leq 0. \end{aligned}$$

The error dynamic system (6.4.4) is time-varying due to the existence of reference signal, therefore the Lasalle's invariance theorem can not be directly applied, however it can be easily made to be time invariant by introducing a new state and the following dynamics.

$$\dot{\tau} = 1.$$

And all the time varying terms are changed to be the functions of τ. According to the Lasalle's invariance theorem, any bounded trajectory must go to the largest invariant set, which is:

$$\Omega = \{\bar{x}_e, \bar{\theta}_e | x_e = 0, \phi_e = 0, \theta_e = 0, \psi_e = 0\}.$$

Then the dynamics for y_e, z_e are:

$$\begin{cases} \dot{y}_e &= -\omega_x z_e \\ \dot{z}_e &= \omega_x y_e \end{cases},$$

which is stable.

6.4.2 The Dynamic Tracking Control Design

This section presents the dynamic trajectory tracking control design. A dynamic model of AUVs is given in [14]. It is a simplified model derived for control design purpose, and captures the main dynamics of a flat-fish shaped AUV. The vehicle is underactuated, i.e., it has less control inputs than the number of degree of freedom (DOF). Specifically, the three controls are surge propulsion T, rudder angle δ_r for yaw rotation, and stern and bow plane angles $\delta_s = -\delta_b$ for pitch rotation. Since the vehicles are considered to be nonholonomic, the dynamics of sway and heave are neglected, i.e. the sway velocity and heave velocity are

always zero. The dynamic equations of motion are:

$$\begin{cases} (m - r_3 X_{\dot{v}})\dot{v} &= r_2 X_{vv} v^2 + T \\ (I_x - r_5 K_{\dot{\omega}_x})\dot{\omega}_x &= r_5 K_{\omega_y\omega_z}\omega_y\omega_z + r_4 K_{\omega_x} v\omega_x + z_{CB} B\cos\theta\sin\phi \\ (I_y - r_5 M_{\dot{\omega}_y})\dot{\omega}_y &= (r_5 M_{\omega_x\omega_z} + I_z - I_x)\omega_x\omega_z + r_4 M_{v\omega_y} v\omega_y \\ & \quad + r_3 v^2(M_{ds}\delta_s + 2M_{db}\delta_b) + z_{CB} B\sin\theta \\ (I_z - r_5 N_{\dot{\omega}_z})\dot{\omega}_z &= (r_5 N_{\omega_x\omega_y} + I_x - I_y)\omega_x\omega_y + r_4 N_{\omega_z} v\omega_z + r_3 v^2 N_{dr}\delta_r \end{cases} . \quad (6.4.6)$$

An explanation of the terms and the values of the main entries in (6.4.6) is as follows: $m = 5454.54kg$ is the vehicle's mass, and $I_x = 2038Nms^2$, $I_y = 13587Nms^2$, and $I_z = 13587Nms^2$ are the moments of inertia about the body axes respectively. The term B is the buoyancy force applying on the center of buoyancy (CB). The term z_{CB} is the z-coordinate of the CB in the body frame.

$$r_i = \frac{\rho}{2} L^i, \quad i = 1, 2, 3, 4, 5,$$

where ρ is the water density and $L = 5.3m$ the AUVs length. $X_{\dot{v}}$ is the added mass term and $K_{\dot{\omega}_x}$, $M_{\dot{\omega}_y}$, $N_{\dot{\omega}_z}$ are added moments of inertia terms. $K_{\omega_y\omega_z}$, $M_{\omega_x\omega_z}$, and $N_{\omega_x\omega_y}$ are added mass cross terms. $X_{\omega_x\omega_x}$, K_{ω_x}, $M_{v\omega_y}$, M_{ds}, M_{db}, N_{ω_z}, and $N_{d\omega_z}$ are drag and lift, force and moment terms. More detailed descriptions and values of the model parameters can be found in [14].

The dynamical error model (6.4.6) can be partially linearized to be:

$$\begin{cases} \dot{v} &= \tau_v \\ \dot{\omega}_x &= \frac{r_4 K_{\omega_x}}{I_x - r_5 K_{\dot{\omega}_x}} v\omega_x + \epsilon \\ \dot{\omega}_y &= \tau_y \\ \dot{\omega}_z &= \tau_z \end{cases}, \tag{6.4.7}$$

where

$$\begin{aligned} T &= -r_2 X_{vv} v^2 + (m - r_3 X_{\dot{v}})\tau_v \\ \delta_s &= [\frac{1}{r_3 v^2}(M_{ds} - 2M_{db})][(I_x - I_z - r_5 M_{\omega_x\omega_z})\omega_x\omega_z - r_4 M_{v\omega_y} v\omega_y \\ &\quad -r_3 v^2(M_{ds}\delta_s - 2M_{db}\delta_b) - z_{CB} B\sin\theta + (I_y - r_5 M_{\dot{\omega}_y})\tau_y] \\ \delta_r &= \frac{1}{r_3 v^2 N_{dr}}[(I_y - I_x - r_5 N_{\omega_x\omega_y})\omega_x\omega_y - r_4 N_{\omega_z} v\omega_z + (I_z - r_5 N_{\dot{\omega}_z})\tau_z] \\ \epsilon &= [r_5 K_{\omega_y\omega_z}\omega_y\omega_z + z_{CB} B\cos\theta\sin\phi]/(I_x - r_5 K_{\dot{\omega}_x}). \end{aligned}$$

We will use backstepping approach to design the torque control τ_v, τ_y, and τ_z. Let the kinematic control u_1, u_2 obtained in (6.4.5) be the desired linear and angular velocity to be tracked, i.e.:

$$v_d = u_1, \quad \bar{\omega}_d = [\omega_x^d \;\; \omega_y^d \;\; \omega_z^d]^T = u_2.$$

Define the dynamic error states to be:

$$v_e = v - v_d, \;\; \bar{\omega}_e = [\omega_x^e \;\; \omega_y^e \;\; \omega_z^e]^T = [\omega_x - \omega_x^d \;\; \omega_y - \omega_y^d \;\; \omega_z - \omega_z^d]^T.$$

The error dynamics are:

$$\begin{cases} \dot{v}_e &= -\dot{v}_d + \tau_v \\ \dot{\omega}_x^e &= \frac{r_4 K_{\omega_x}}{I_x - r_5 K_{\dot{\omega}_x}} v_e \omega_x + \epsilon' \\ \dot{\omega}_y^e &= -\dot{\omega}_y^d + \tau_y \\ \dot{\omega}_z^e &= -\dot{\omega}_z^d + \tau_z \end{cases},$$

where $\epsilon' = -\dot{\omega}_x^d + [r_4 K_{\omega_x} v_d \omega_x + r_5 K_{\omega_y \omega_z} \omega_y \omega_z + z_{CB} B \cos\theta \sin\phi]/(I_x - r_5 K_{\dot{\omega}_x})$. Design the following control:

$$\begin{cases} \tau_v &= \dot{v}_d - h_1 v_e + \tau_v' \\ \tau_y &= \dot{\omega}_y^d - h_2 \omega_y^e + \tau_y' \\ \tau_z &= \dot{\omega}_z^d - h_3 \omega_z^e + \tau_z' \end{cases}, \tag{6.4.8}$$

where $h_1, h_2, h_3 > 0$ are control gains and τ_v', τ_y', τ_z' are supplemental terms to be determined.

Define the Lyapunov Function Candidate:

$$V_c = V + \frac{1}{2}(v_e^2 + \omega_x^{e2} + \omega_y^{e2} + \omega_y^{e2}).$$

Then the time derivative along the closed loop system is:

$$\begin{aligned}
\dot{V}_c &= -p^T p - \gamma^2 x_e^2 - v_e x_e - p^T S_2(\bar{\theta})\bar{\omega}_e + v_e \dot{v}_e + \bar{\omega}_e^T \dot{\bar{\omega}}_e \\
&= -p^T p - \gamma^2 x_e^2 - h_1 v_e^2 - h_2 \omega_y^{e2} - h_3 \omega_z^{e2} + v_e(\tau_v' - x_e + \frac{r_4 K_{\omega_x}}{I_x - r_5 K_{\dot{\omega}_x}} \omega_x \omega_x^e) \\
&\quad + \omega_y^e(\tau_y' - k_1 \sin\phi_e \sin\phi \tan\theta - k_2 \sin\theta_e \cos\phi - k_3 \sin\psi_e \sin\phi \sec\theta) \\
&\quad + (\epsilon' - k_1 \sin\phi_e)\omega_x^e + \omega_z^e(\tau_z' - k_1 \sin\phi_e \cos\phi \tan\theta + k_2 \sin\theta_e \sin\phi \\
&\quad - k_3 \sin\psi_e \cos\phi \sec\theta) \\
&= -p^T p - \gamma^2 x_e^2 - h_1 v_e^2 - h_2 \omega_y^{e2} - h_3 \omega_z^{e2} + (\epsilon' - k_1 \sin\phi_e)\omega_x^e \\
&= -p^T p - h_1 v_e^2 - h_2 \omega_y^{e2} - h_3 \omega_z^{e2} - [(\gamma^2 - \eta_1)\omega_x^e - \frac{\eta_2 - k_1 \sin\phi_e}{2}]^2 \\
&\quad + (\frac{\eta_2 - k_1 \sin\phi_e}{2})^2, \qquad (6.4.9)
\end{aligned}$$

where,

$$\begin{aligned}
\eta_1 &= r_4 K_{wx} V_d \\
\eta_2 &= -\dot{\omega}_x^d + [r_4 K_{\omega_x} v_d \omega_x^d + r_5 K_{\omega_y \omega_z} \omega_y \omega_z + z_{CB} B \cos\theta \sin\phi]/(I_x - r_5 K_{\dot{\omega}_x}).
\end{aligned}$$

and

$$\begin{aligned}
\tau_v' &= x_e - \frac{r_4 K_{\omega_x}}{I_x - r_5 K_{\dot{\omega}_x}} \omega_x \omega_x^e \\
\tau_y' &= k_1 \sin\phi_e \sin\phi \tan\theta + k_2 \sin\theta_e \cos\phi + k_3 \sin\psi_e \sin\phi \sec\theta \\
\tau_z' &= k_1 \sin\phi_e \cos\phi \tan\theta - k_2 \sin\theta_e \sin\phi + k_3 \sin\psi_e \cos\phi \sec\theta.
\end{aligned}$$

By assumption 3, τ_y' and τ_z' are uniformly bounded, therefore w_y, w_z are uniformly bounded. Hence η_2 is uniformly bounded. Therefore the right hand side of (6.4.9) shows the tracking error is bounded, and the bound can be tuned by k_1.

6.4.3 Simulation Results

The simulation is conducted using Matlab simulink. The simulink platform is shown in the Fig. 6.10. The AUV dynamic model (6.4.6) and hydrodynamic parameters are taken from the NPS AUV II model from [14]. Specifically,

$$m = 5454.54kg,\ \ I_x = 2038Nms^2,\ \ I_y = 13587Nms^2,\ \ I_z = 13587Nms^2,$$

$$B = 53.4kN,\ \ \rho = 1000kg/m^3,\ \ L = 5.3m,\ \ X_{\dot{v}} = -0.0076,\ \ X_{vv} = 0.053,$$

$$K_{\omega_x} = -0.011,\ \ K_{\dot{\omega}_x} = -0.001,\ \ K_{\omega_y\omega_z} = 0.017,\ \ M_{\dot{\omega}_y} = -0.017,\ \ z_{CB} = 0,$$

$$M_{v\omega_y} = -0.068,\ \ M_{ds} = -0.041,\ \ M_{db} = 0.0035,\ \ M_{\omega_x\omega_z} = 0.005,$$

$$N_{\omega_z} = -0.016,\ \ N_{\omega_x\omega_y} = -0.021,\ \ N_{\dot{\omega}_z} = -0.0034,\ \ N_{d\omega_z} = -0.013.$$

Initial conditions of the planned trajectory are:

$$x_0 = 10,\ y_0 = 10,\ z_0 = 10,\ \phi_0 = 0,\ \theta_0 = \pi/6,\ \psi_0 = \pi/4,\ v_0 = 5.$$

Final conditions of the planned trajectory are:

$$x_f = 90,\ y_f = 90,\ z_f = 90,\ \phi_f = 0,\ \theta_f = -\pi/6,\ \psi_f = \pi/3,\ v_f = 4.$$

The actual initial position and orientation of the AUV are:

$$x_i = 40,\ y_i = 30,\ z_i = 0,\ \phi_i = 0,\ \theta_i = \pi/6,\ \psi_i = \pi/4,\ v_i = 0.$$

The choice of control gains are: $k_1 = k_2 = k_3 = 1$ and $h_1 = h_2 = h_3 = 5$.

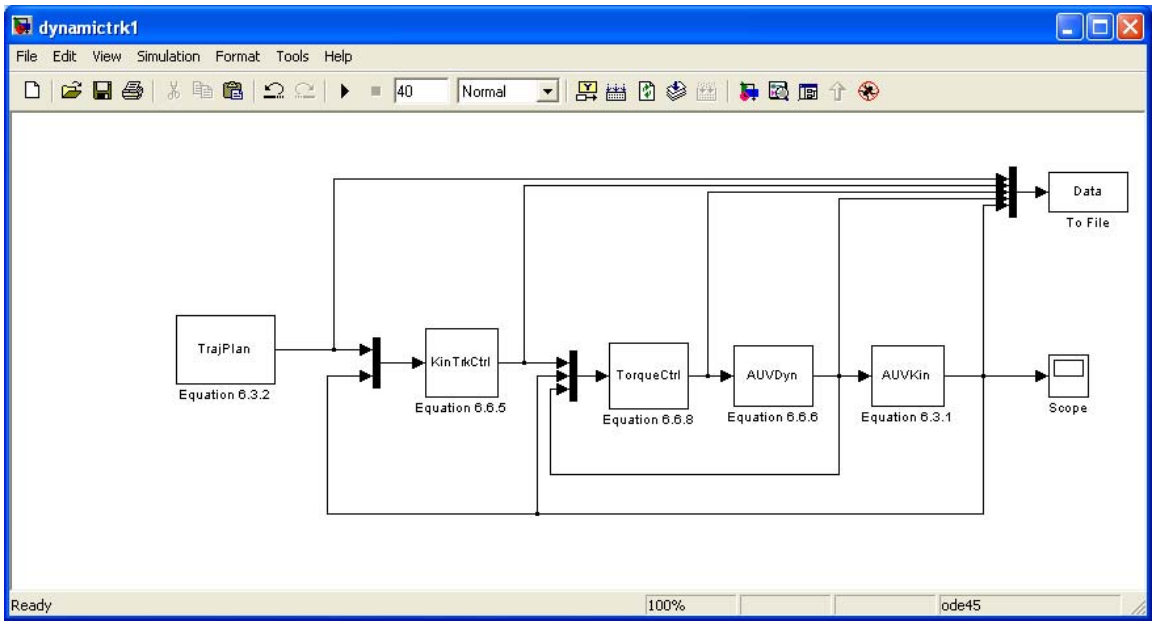


Figure 6.10: The Simulink Platform for Dynamic Tracking Controls

The simulation results are illustrated in Fig. 6.11. Fig. 6.11(a) to Fig. 6.11(d) shows the tracking control is successful, the trajectory of the AUV is able to track the the planned trajectory. However, there are oscillations in the signals which indicate there exists a bounded error for the trajectory and control.

6.5 Conclusion

In this chapter, an optimal real-time motion planning approach is proposed for AUVs operating in an unknown 3D underwater space. The 3D planning problem is reduced to a 2D problem. The vehicle's kinematic model was explicitly taken into consideration. Collision avoidance criteria is established based on a piecewise polynomial parameterization of feasible trajectories. By checking the condition in realtime, the proposed approach prevents

any collision and renders close-form control solutions with optimal performance. Moreover, a dynamic tracking controller is proposed which ensures the AUV to track the planned trajectory.

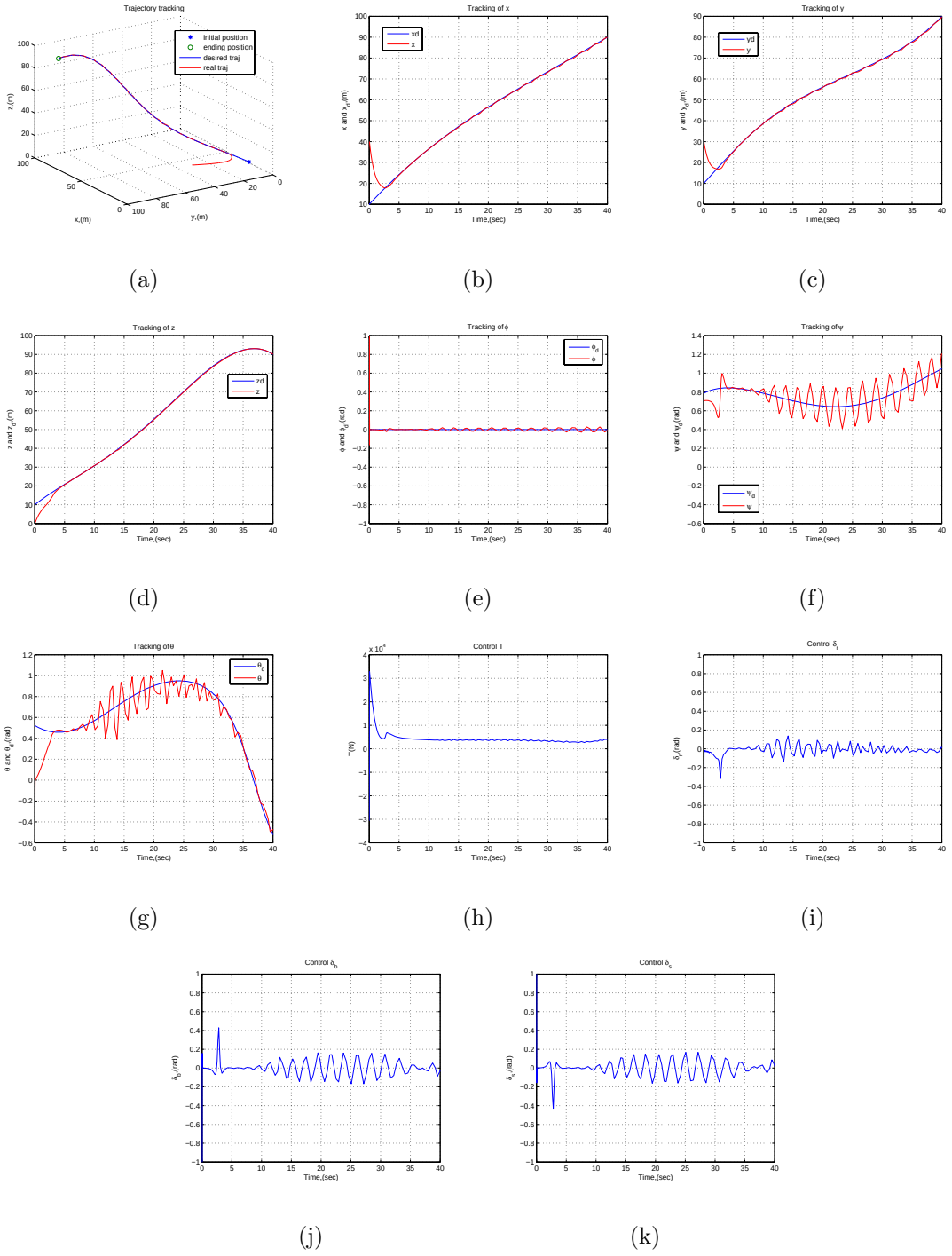


Figure 6.11: Trajectory Tracking Simulation Results. (a), Desired and Actual Trajectory. (b), Tracking of x. (c), Tracking of y. (d), Tracking of z. (e), Tracking of ϕ. (f), Tracking of ψ. (g), Tracking of θ. (h), Torque Control. (i), Rudder Control. (j), Bow Plane Angle. (k), Stern Plane Angle.

CHAPTER 7
COORDINATED EXPLORATION AND FORMATION CONTROL FOR MULTIPLE UNMANNED AERIAL VEHICLES (UAVS)

The study of unmanned aerial vehicles has been an active research topic in recent years due to the rapid growth of UAS real-world applications driven by the Global War on Terrorism (GWOT). The UAS is defined as a complete unmanned system including control station, data links, and vehicle. Currently UAVs operate individually, independent of neighboring UAVs and used primarily for surveillance. However, UAVs' tasks are expanding to the extent where UAV groups will work as cooperative autonomous units. The idea behind is that cooperatively controlled units have the ability to accomplish complicated missions with higher efficiency and failure tolerance, such as coordinated navigation, terrain surveillance and search/rescue tasks.

Inspired by the flocking behavior of flying birds, Reynolds conducted a computer simulation model for cohesion, separation, and alignment in [63]. Subsequently, a straightforward discrete-time model (Vicsek model) was presented in [72] for the heading adjustment of autonomous particles moving in a plane. Simulation results verified the Vicsek model. More recently, [23] presented a theoretical explanation of Vicsek's model by using graph theory and established conditions on the connectivity of undirected sensor graphs for the convergence of overall system. Later, [41, 48, 61] extended the condition to networks with directed sensor graphs. One recent result on synthesizing decentralized cooperative control is from

matrix theory. Less restrictive results have been established in [59]. Suppose there is a group of robots which can be feedback linearized and their sensing/communication matrices satisfy sequentially complete conditions. Then, the production of state transition matrices of the overall system results in a matrix with identical rows, hence all states of the group of robots will converge. Therefore, cooperative control objective can be accomplished. The cooperative control strategy has been widely applied in consensus controls and formation controls, such as [61, 77, 78, 59].

In this chapter, we consider a mission scenario that involves both motion planning and cooperative formation controls of UAS and provide our solutions. The motion planing approach is parametric that adapts to the UAV's kinematic model as well as achieve obstacle avoidance. The resulting controls are in closed forms, hence can be implemented easily in realtime applications. The cooperative formation controls are local decentralized controllers developed on the matrix theory for each UAV. A virtual leader is introduced into the group of UAVs to achieve asymptotic trajectory convergence and help interaction between the human-machine interface (HMI) and operators.

7.1 Problem Formulation

In this study, the following scenario is considered:

1. A predefined area (presumably rectangular) is to be surveilled.

2. A group of UAVs are launched separately from their base locations around the area.

3. If any target is found, neighboring UAVs come together as a formation and cooperatively fly over target locations.

4. UAVs should be able to avoid flying into some restricted areas (obstacles).

The following nonholonomic kinematic model of a UAV is adopted:

$$\begin{cases} \dot{x} &= v\cos(\theta) \\ \dot{y} &= v\sin(\theta) \\ \dot{\theta} &= \omega \end{cases}, \tag{7.1.1}$$

where (x, y) is the world coordinate of the UAV, θ is the heading angle, v is the longitudinal velocity, and ω is its rotation velocity. A block diagram of the UAS control scheme is shown in Fig. 7.1. The innermost loop handles guidance and motion control, the middle loop deals with navigation and obstacle avoidance, the outmost loop manages mission, payload and other high level configurations.

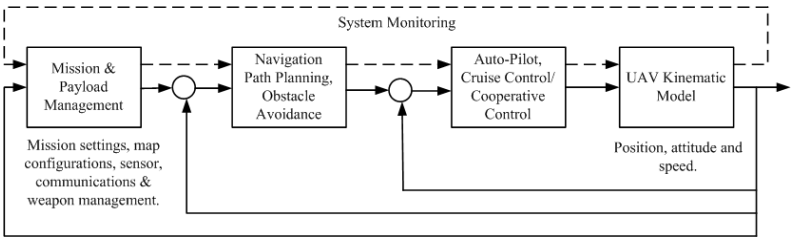


Figure 7.1: Block Diagram of UAS Control Loops.

Fig. 7.2 illustrates the modules should be implemented in the control software and its logic architecture. At top level are human-machine interface and dynamic environment. Control algorithms are implemented in the middle. The obtained controls are applied on the UAVs' kinematic model at the bottom level to achieve specified goals.

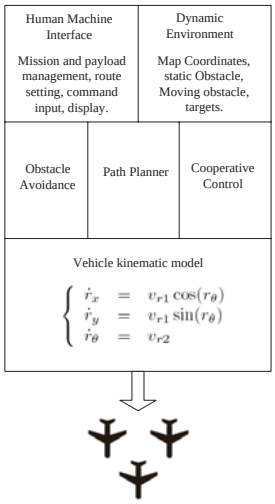


Figure 7.2: Control Software Modules.

7.2 Motion Planning

In a surveillance mission, each UAV may be required to pass multiple waypoints. The objective of motion planning is to find a feasible and smooth trajectory that leads a UAV from a starting waypoint to an ending waypoint. Every pair of consecutive waypoints are composed of a starting waypoint and an ending waypoint, through which the UAV is able to navigate all required waypoints.

7.2.1 Parametric Feasible Trajectories

By analyzing the kinematic model (7.1.1), one can establish that a UAV's path is some smooth function $y = f(x)$. Given initial and final conditions $q^0 = (x^0, y^0, \theta^0, v^0)$ at t_0 and

$q^f = (x^f, y^f, \theta^f, v^f)$ at t_f, the model has four constraints on the path. That is:

$$y^0 = f(x^0),\ \tan(\theta^0) = \frac{dy}{dx}\Big|_{t=t_0},\ y^f = f(x^f),\ \tan(\theta^f) = \frac{dy}{dx}\Big|_{t=t_f}.$$

Thus, if a path is to be parameterized by a finite dimensional polynomial, it should have at least four free coefficients to accommodate these constraints. To achieve a class of paths, more than four coefficients are needed. In this paper, the path is parameterized by a $4th$ order polynomial which has five coefficients. That is,

$$y = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4. \tag{7.2.1}$$

Given the boundary conditions q_0 and q_f the solution to the coefficients are:

$$[a_0\ a_1\ a_2\ a_3]^T = (B_1)^{-1}(Y_1 - A_1 a_4),$$

where

$$B_1 = \begin{bmatrix} 1 & x^0 & (x^0)^2 & (x^0)^3 \\ 0 & 1 & 2x^0 & 3(x^0)^2 \\ 1 & x^f & (x^f)^2 & (x^f)^3 \\ 0 & 1 & 2x^f & 3(x^f)^2 \end{bmatrix},\ Y_1 = \begin{bmatrix} y^0 \\ \tan(\theta^0) \\ y^f \\ \tan(\theta^f) \end{bmatrix},\ A_1 = \begin{bmatrix} (x^0)^4 \\ 4(x^0)^3 \\ (x^f)^4 \\ 4(x^f)^3 \end{bmatrix}.$$

It shows that the matrices B, Y, A are determined by boundary conditions. As long as $x^f \neq x^0$, B_1 is invertible, the coefficients a_0, a_1, a_2, a_3 are solvable in terms of a_4. Therefore (7.2.1) denotes a class of paths that take a_4 as its parameter and any path in the class would satisfy the boundary condition and is feasible to a UAV. By adjusting a_4, it is possible for one to seek a collision-free path.

Polynomial (7.2.1) is still not a trajectory yet since the timing information has not been incorporated. In order to do so, we propose the following motion:

$$x = b_0 + b_1 t + b_2 t^2 + b_3 t^3. \tag{7.2.2}$$

By meeting the boundary conditions, the coefficients can be uniquely determined as:

$$[b_0\ b_1\ b_2\ b_3]^T = (B_2)^{-1}(Y_2),$$

where

$$B_2 = \begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 \\ 0 & 1 & 2t_0 & 3t_0^2 \\ 1 & t_f & t_f^2 & t_f^3 \\ 0 & 1 & 2t_f & 3t_f^2 \end{bmatrix}, \quad Y_2 = \begin{bmatrix} x^0 \\ v^0\cos(\theta^0) \\ x^f \\ v^f\cos(\theta^f) \end{bmatrix}.$$

As long as $t_f > t_0$, B_2 is invertible and b_0, b_1, b_2, b_3 are solvable.

Theorem 11. *The control given by*

$$\begin{cases} v &= \sqrt{\dot{x}^2 + \dot{y}^2} \\ \omega &= (\ddot{y}\dot{x} - \dot{y}\ddot{x})/(\dot{x}^2 + \dot{y}^2) \end{cases} \tag{7.2.3}$$

steers a UAV modeled by (7.1.1) along the trajectory (7.2.1) and (7.2.2).

Proof. It is directly synthesized from (7.1.1) that

$$v = \sqrt{\dot{x}^2 + \dot{y}^2}.$$

Moreover, since $x^f \neq x^0$, the tangent of a polynomial is either in $(-\frac{\pi}{2}\ \frac{\pi}{2})$ or $(\frac{\pi}{2}\ \frac{3\pi}{2})$. Therefore:

$$\theta = \arctan\frac{\dot{y}}{\dot{x}} \ \text{ or } \ \theta = \arctan\frac{\dot{y}}{\dot{x}} + \pi.$$

It follows that,

$$\omega = \dot{\theta} = (\ddot{y}\dot{x} - \dot{y}\ddot{x})/(\dot{x}^2 + \dot{y}^2).$$

Therefore, trajectories of x and y satisfy the state equation (7.1.1). □

7.2.2 Motion Planning for Avoiding Static/Dynamic Obstacles

To handle a dynamic environment, when new obstacles' information is available, the parametric trajectory given by equation (7.2.1) may need updates. The update can be accomplished by a piecewise-constant polynomial parametrization. Suppose the total operation time is T from the initial configuration q^0 to its final configuration q^f, and the sampling period is T_s, so that $\bar{k} = T/T_s$ is an integer. For $k = 0$, the initial condition is q^0. For $\bar{k} > k > 0$, the initial condition is $q^k = (x^k, y^k, \theta^k, v^k)$, the final condition is always q^f. The path planning method described in the previous subsections can be applied by using the boundary conditions $q^k,\ k = 0, 1, \cdots$ and q^f for real-time replanning as k increases. In the following part of this paper, notations with superscription k represent the corresponding terms at the kth sampling period.

Fig. 7.3 illustrates a UAV moving from q^0 to q^f in the presence of obstacles. The UAV is represented by the smallest sphere that contains itself. In Fig. 7.3, the small circle with solid line is the UAV, its radius is r and velocity is v^k. The larger circle with solid line centered at (x_o^k, y_o^k) represents the obstacle, its radius is R and velocity is v_o^k. The circle with dashed line represents the sensing range of a UAV, within which, an obstacle can be detected, its radius is R_s. In kth sampling period, the trajectory equation (7.2.1) is rewritten as:

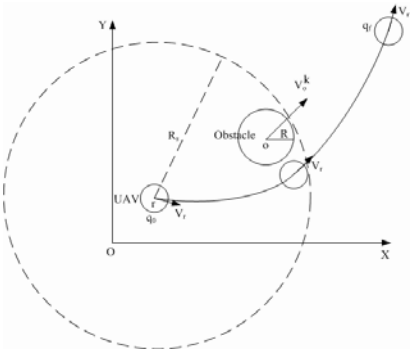


Figure 7.3: A UAV in the Presence Obstacles

$$y = a_0^k + a_1^k x + a_2^k x^2 + a_3^k x^3 + a_4^k x^4. \tag{7.2.4}$$

Clearly, for anytime $t \in [t_0 + kT_s,\ t_0 + T]$, it is desirable to have the distance between the UAV and obstacle greater than $r + R$ to avoid any possible collision. Therefore, the collision avoidance criterion is:

$$(y - y_o^k - v_{o,y}^k \tau)^2 + (x - x_o^k - v_{o,x}^k \tau)^2 \geq (r + R)^2, \tag{7.2.5}$$

where $v_{o,x}^k$ and $v_{o,y}^k$ are the obstacle's velocity along x and y directions, $\tau = t - (t_0 + kT_s)$ for $t \in [t_0 + kT_s,\ t_0 + T]$.

According to the results in section 7.2.1, the coefficients of (7.2.4) can be solved in terms of a_4^k,

$$[a_0^k \ \ a_1^k \ \ a_2^k \ \ a_3^k]^T = (B^k)^{-1}(Y^k - A^k a_4^k), \tag{7.2.6}$$

where

$$B^k = \begin{bmatrix} 1 & x^k & (x^k)^2 & (x^k)^3 \\ 0 & 1 & 2x^k & 3(x^k)^2 \\ 1 & x^f & (x^f)^2 & (x^f)^3 \\ 0 & 1 & 2x^f & 3(x^f)^2 \end{bmatrix}, \quad Y^k = \begin{bmatrix} y^k \\ \tan(\theta^k) \\ y^f \\ \tan(\theta^f) \end{bmatrix}, \quad A^k = \begin{bmatrix} (x^k)^4 \\ 4(x^k)^3 \\ (x^f)^4 \\ 4(x^f)^3 \end{bmatrix}.$$

Substituting (7.2.4) and (7.2.6) into (7.2.5), one obtains the following inequality:

$$g_2(x,k)(a_4^k)^2 + g_1(x,k,\tau)a_4^k + g_0(x,k,\tau)|_{\tau=t-t_0-kT_s} \geq 0 \tag{7.2.7}$$

for all $\tau \in [0,\ T - kT_s]$, where

$$\begin{aligned} g_2(x,k) &= [x^4 - h(x)(B^k)^{-1}A^k]^2 \\ g_1(x,k,\tau) &= 2[x^4 - h(x)(B^k)^{-1}A^k][h(x)(B^k)^{-1}Y^k - y_k - v_{o,y}^k\tau] \\ g_0(x,k,\tau) &= [h(x)(B^k)^{-1}Y^k - y_k - v_{o,y}^k\tau]^2 + (x - x_k - v_{o,x}^k\tau)^2 - (r+R)^2 \\ h(x) &= [1 \ \ x \ \ x^2 \ \ x^3] \end{aligned}.$$

Inequality (7.2.7) is about the adjustable coefficient a_4^k, as long as a_4^k is picked to satisfy this inequality, the obstacle is avoided.

Theorem 12. *The collision avoidance condition (7.2.7) is always solvable as long as the obstacle does not hold the ending position q^f infinitely long.*

Proof. Note that the left hand side of (7.2.7) is a parabola equation and $g_2(x,k) \geq 0$. In the case that $g_2(x,k) > 0$, (7.2.7) is always solvable. So one only needs to study what happens when $g_2(x,k) = 0$. A simple observation is that when $g_2(x,k) = 0$, there is also $g_1(x,k,\tau) = 0$, therefore a_4^k no longer appears in the inequality (7.2.7), which shows that no matter what a_4^k is chosen, it can't affect the collision avoidance. It further implies that the

polynomial parameterized trajectory (7.2.4) is not affected by a_4^k. This can only happen on the two boundary points of the trajectory, which are fixed. Other points in between always vary with respect to the choice of a_4^k. These can be verified as:

$$\begin{aligned} y &= a_0^k + a_1^k x + a_2^k x^2 + a_3^k x^3 + a_4^k x^4 \\ &= h(x)(B^k)^{-1}(Y^k - A^k a_4^k) + a_4^k x^4 \\ &= [1 \;\; x \;\; x^2 \;\; x^3](B^k)^{-1}Y^k + a_4^k[x^4 - h(x)(B^k)^{-1}A^k] \\ &= [1 \;\; x \;\; x^2 \;\; x^3](B^k)^{-1}Y^k + a_4^k g_2(x,k), \end{aligned} \tag{7.2.8}$$

(7.2.8) shows that (7.2.4) degenerated to a third-order polynomial no matter what a_4^k is. This is impossible unless at the boundary points where boundary constraints must hold for both third- and forth-order polynomials.

Moreover, in the case of $g_2(x,k) = 0$, (7.2.7) reduces to $g_0(x,k,\tau) \geq 0$, which equivalently requires the boundary points are free of obstacles. For the starting position, it would be always true. For the ending position, one can extend τ (which means the mission time is extended) until $g_0(x,k,\tau) \geq 0$ holds. As long as the obstacle does not occupy the ending position forever, $g_0(x,k,\tau) \geq 0$ would hold for large τ. □

If multiple moving obstacles present in the environment, every obstacle would impose a constraint similar to (7.2.7) on a_4^k. When a_4^k satisfies these constraints simultaneously, all obstacles are avoided.

7.3 Cooperative Formation Controls

The objective for cooperative control is to ensure a group of dynamical systems (or error systems) converge to the same steady state. In applications of the formation flying control, a

group of UAVs converge to a rigid formation when the error systems from a group of desired trajectories converge to zero.

7.3.1 Formation Control of Multiple UAVs

Consider the problem of controlling a group of q UAVs with model (7.1.1) to form a rigid formation during its fly. To synthesize the formation control, the first step is to feedback linearize model (7.1.1). In following paragraphs, subscription i is used to denote the state and controls of the ith UAV. Define the following diffeomorphic state and control transformations, for $i = 1, 2, \cdots, q$,

$$\phi_{i1} = x_i + L\cos(\theta_i), \quad \phi_{i2} = y_i + L\sin(\theta_i),$$

and

$$\begin{bmatrix} \gamma_{i1} \\ \gamma_{i2} \end{bmatrix} = \begin{bmatrix} \cos(\theta_i) & -L\sin(\theta_i) \\ \sin(\theta_i) & L\cos(\theta_i) \end{bmatrix} \begin{bmatrix} v_i \\ \omega_i \end{bmatrix}.$$

The UAV model can be casted into the following integrator model with stable internal dynamics

$$\dot{\phi}_i = \gamma_i, \tag{7.3.1}$$

where $\phi_i = [\phi_{i1}\ \phi_{i2}]^T$ and $\gamma_i = [\gamma_{i1}\ \gamma_{i2}]^T$.

Along an arbitrary trajectory H, a formation can be defined by its Frenet frame $F_H(t)$, which moves along the path. Let $e_1(t) \in \Re^2$ and $e_2(t) \in \Re^2$ be the orthonormal base of $F_H(t)$, and $\phi^d(t) = [x^d(t)\ y^d(t)]^T \in \Re^2$ be the origin of $F_H(t)$ that is on the trajectory. Fig. 7.4 illustrates a formation composed of three UAVs. A formation that is composed of q

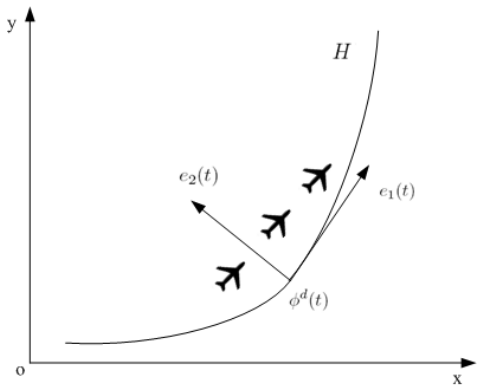


Figure 7.4: A Formation Defined in the Frenet Frame

UAVs in $F_H(t)$ can be denoted by $\{P_1(t), \cdots, P_q(t)\}$, where

$$P_i(t) = d_{i1}(t)e_1(t) + d_{i2}(t)e_2(t), \;\; i = 1, \cdots, q.$$

with $d_i(t) = [d_{i1}(t), d_{i2}(t)] \in \Re^2$ as the desired coordinates for the ith UAV in $F_H(t)$. Obviously, a rigid formation can be achieved by setting $d_i(t)$ to be constant. Therefore, the desired trajectory for the ith robot is:

$$\phi_i^d(t) = \phi^d(t) + P_i(t). \tag{7.3.2}$$

Further define the following state transformation:

$$\zeta_i(t) = \phi_i - \phi_i^d, \;\; \gamma_i = \dot{\phi}_i^d - \phi_i + \phi_i^d + u_i. \tag{7.3.3}$$

It follows that

$$\dot{\zeta}_i = A_i\zeta_i + B_iu_i, \quad \eta_i = C_i\zeta_i, \tag{7.3.4}$$

where u_i is the cooperative control for ith UAV, η_i is the output, and

$$A_i = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}, \quad B_i = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad C_i = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

(7.3.4) is in the canonical form of [59], therefore its cooperative control is given by:

$$u_i = \sum_{j=1}^{q} G_{ij}(t)[s_{ij}(t)\eta_j], \quad i = 1, \cdots, q, \tag{7.3.5}$$

where $s_{ij}(t)$ is the entry of sensing/communication matrices, G_{ij} is a 2×2 block that reflects the influence of jth output to the ith control in the gain matrix G. It can be obtained by the following formula:

$$G_{ij}(t) = \frac{s_{ij}(t)}{\sum_{k=1}^{q} s_{ik}(t)} K_c, \qquad j = 1, \cdots, q, \tag{7.3.6}$$

where the design parameter $K_c \in \Re^{2\times 2}$ is a constant, non-negative, and row stochastic matrix. The sensing/communication matrix is defined as:

$$S(t) = \begin{bmatrix} S_1(t) \\ S_2(t) \\ \vdots \\ S_q(t) \end{bmatrix} = \begin{bmatrix} s_{11} & s_{12}(t) & \cdots & s_{1q}(t) \\ s_{21}(t) & s_{22} & \cdots & s_{2q}(t) \\ \vdots & \vdots & \vdots & \vdots \\ s_{q1}(t) & s_{q2}(t) & \cdots & s_{qq} \end{bmatrix},$$

where $s_{ii} \equiv 1$; $s_{ij}(t) = 1$ if the outputs of jth UAV is known by the ith UAV at time t; otherwise $s_{ij}(t) = 0$.

7.3.2 Adaptive Cooperative Formation Controls

The formation control (7.3.5) does make the error states ζ_i converge, but the limit is not necessarily at the origin, i.e. the formation may be shifted from its desired trajectory [59]. In order to achieve asymptotic convergence to a desired trajectory, a virtual UAV needs be

adopted into the group. The virtual UAV possesses the following properties:

$$\zeta_0 \equiv 0, \quad \eta_0 \equiv 0,$$

which indicates the virtual UAV is always on the desired trajectory. The virtual UAV and other real UAVs interact through the following augmented sensing/communication matrix:

$$\overline{S}(t) = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ s_{10} & & & \\ \vdots & & S(t) & \\ s_{q0} & & & \end{bmatrix},$$

where the first row/column reflects effects of the virtual vehicle. Correspondingly, the cooperative formation controls become:

$$u_i = \sum_{j=0}^{q} \frac{s_{ij}(t)}{\sum_{k=0}^{q} s_{ik}(t)} K_c[s_{ij}(t)\eta_j], \quad i = 1, \cdots, q. \tag{7.3.7}$$

7.3.3 Circular Trajectories and Arbitrary Trajectories

The formation control design scheme presented in section 7.3.1 and 7.3.2 requires a desired trajectory to be specified. In this section, the methods of generating the desired trajectories are discussed.

For the simple case of a circular trajectory, it can be parameterized as:

$$\phi^d(t) = [x_c + R\cos(\omega t),\ y_c + R\sin(\omega t)]^T,$$

where (x_c, y_c) is the center of the trajectory. R is the radius of the trajectory and ω is the circling rate. The moving frame of the trajectory is:

$$e_1(t) = \begin{bmatrix} -\sin(\omega t) \\ \cos(\omega t) \end{bmatrix}, \quad e_2(t) = \begin{bmatrix} \cos(\omega t) \\ \sin(\omega t) \end{bmatrix}.$$

Therefore the desired trajectories for the whole formation are given by (7.3.2).

In most cases, it is desired that a group of UAVs fly through a set of specified waypoints. Suppose there are n waypoints $(x_j^w, y_j^w),\ j = 1, \cdots, n$. The following Lagrange interpolating polynomial can be adopted to determine a path:

$$y^d(x^d) = \sum_{j=1}^{n} y_j^w \frac{g(x^d)}{(x^d - x_j^w) g'(x_j^w)}, \tag{7.3.8}$$

where $g(x^d) = (x^d - x_1^w)(x^d - x_2^w) \cdots (x^d - x_n^w)$.

Assuming the formation has a desired cruise speed $V_s(t)$, and the desired trajectory starts from the waypoint (x_1^w, y_1^w) at time t_0, then the trajectory of the Frenet frame can be given as:

$$\begin{cases} x^d(t) &= x_1^w + \int_{t0}^{t} \frac{V_s(t)}{\sqrt{1+(dy^d/dx^d)^2}} dt \\ y^d(t) &= y_1^w + \int_{t0}^{t} \frac{V_s(t)}{\sqrt{1+(dx^d/dy^d)^2}} dt \end{cases},$$

which can be numericly integrated online in computer implementation. The orthonormal base of the Frenet frame can be given as:

$$e_1(t) = \begin{bmatrix} \frac{1}{\sqrt{1+(dy^d/dx^d)^2}} \\ \frac{dy^d/dx^d}{\sqrt{1+(dy^d/dx^d)^2}} \end{bmatrix}, \quad e_2(t) = \begin{bmatrix} \frac{-dy^d/dx^d}{\sqrt{1+(dy^d/dx^d)^2}} \\ \frac{1}{\sqrt{1+(dy^d/dx^d)^2}} \end{bmatrix}.$$

Then (7.3.2) gives the desired trajectories for the whole group of UAVs and the formation control can be designed through the procedure presented in Section 7.3.1 and Section 7.3.2.

7.3.4 Internal and External Collision Avoidance

In the formation flying control, one would always specify the desired position of each UAV in the formation in a way that they do not collide with each other. However, in transient, control (7.3.7) by itself cannot guarantee that there is no collision among UAVs. These collisions are referred as internal collisions in the formation. Also, the UAV formation may collide with obstacles, which is referred as external collisions. To handle the collision avoidance problem, the controls need be improved.

The collision avoidance scheme is adding supplemental terms to control (7.3.7). These terms are going to provide a negative feedback using the distances among all entities. The basic idea is to think all entities have a layer of elastic massless substance surrounded. So, when UAVs and obstacles get close, they resist each other to get closer. This can be illustrated by Fig. 7.5. Denote the collision-free formation control by $u_i',\ i = 1, \cdots, q$.

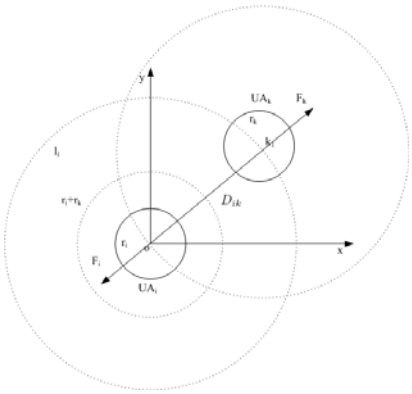


Figure 7.5: Collision avoidance for UAV formations

Correspondingly, in (7.3.3), u_i needs be replaced by u_i'. Consider the scenario of q UAVs with their radius of envelop $r_i,\ i = 1, \cdots, q$ and n obstacles with their radius of envelop

$R_i,\ i = 1, \cdots, n$ and center at $o_i = [x_{oi}\ y_{oi}]^T$. u_i' is proposed to be:

$$u_i' = u_i + \sum_{k=1, k\neq i}^{q} \rho_1(\|D_{ik}\|)\frac{D_{ik}}{\|D_{ik}\|} + \sum_{k=1}^{n} \rho_2(\|E_{ik}\|)\frac{E_{ik}}{\|E_{ik}\|}, \tag{7.3.9}$$

with $D_{ik} = \phi_i - \phi_k$, $E_{ik} = \phi_i - o_k$ and

$$\rho_1(\|D_{ik}\|) = \begin{cases} \alpha_i \frac{l_i - \|D_{ik}\|}{\|D_{ik}\| - (r_i + r_k)}, & \|D_{ik}\| < l_i \\ 0, & \|D_{ik}\| \geq l_i \end{cases},$$

$$\rho_2(\|E_{ik}\|) = \begin{cases} \alpha_i \frac{l_i - \|E_{ik}\|}{\|E_{ik}\| - (r_i + R_k)}, & \|E_{ik}\| < l_i \\ 0, & \|E_{ik}\| \geq l_i \end{cases},$$

where $\alpha_i > 0$ is the elastic coefficient of ith UAV, l_i is the range of ith UAV's elastic layer within which the resist force is available. Note that there must be $l_i > r_i + r_k$ and $l_i > r_i + R_k$, which means the resist force starts work before the UAV's collision.

7.4 Simulations

The simulation platform is developed by MSVC++ 6. The scenario has six UAVs in a rectangular region. In the first part of the simulation, the six UAVs coverage search the entire area. In the second part, The six UAVs will converge to a rigid formation and fly through some waypoints. Fig. 7.6 is a flow chart of the simulation platform.

In the first part of the simulation (searching), a minimum number of circles are placed the area. Each circle has the same size as the sensing range [17]. The combination of the circles covers the whole region. Centers of these circles are considered as waypoints to be traveled. Each UAV finds a set of waypoints to follow. This is accomplished by a

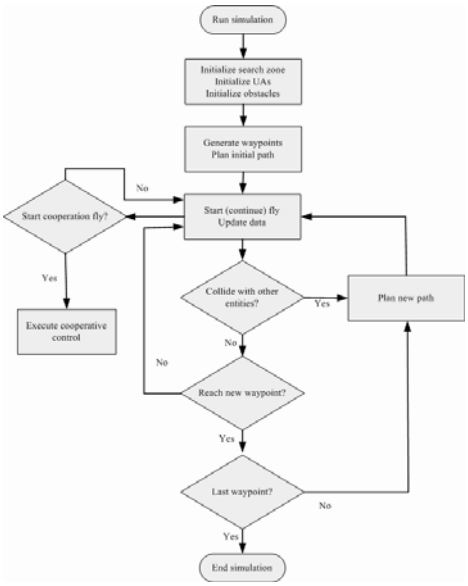


Figure 7.6: Flow-chart of the Simulation Platform

Voronoi diagram, which adds every waypoint to the nearest UAV. Lastly, every UAV selects the nearest waypoint in its set as its first waypoint to move, then travel to the nearest unvisited waypoint that is most clockwise relative to its previous one. This method renders a counterclockwise path. Fig. 7.7 shows the set waypoints assigned to each UAV, and line segments are connected to show a rough path.

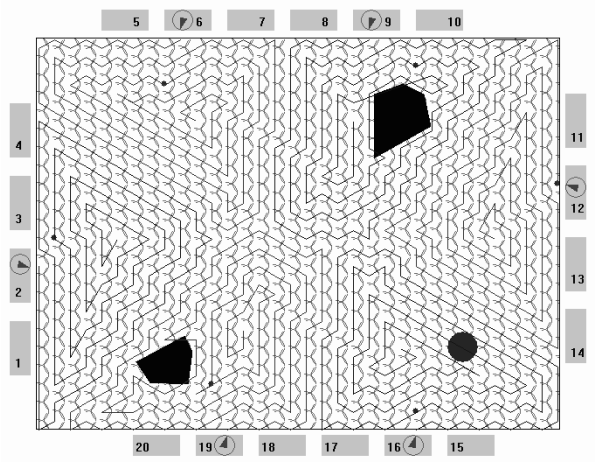


Figure 7.7: Waypoints for Each UAV

7.4.1 Simulation Settings

This section illustrates settings of the simulation scenario. The sensing/communication matrix randomly switch among the following matrices at each sampling period:

$$S_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}, S_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, S_3 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

The row stochastic matrix K_c in (7.3.6) is:

$$K_c = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

The coordinates of the operation region are listed in Table 7.1. The initial settings of UAVs are given in Table 7.2. The positions of static obstacles are given in Table 7.3.

Table 7.1: Map Coordinates

	Horizontal(miles)	Vertical(miles)
Bottom-left	0	0
Top-left	0	75
Top-right	100	75
Bottom-right	100	0

Table 7.2: Initial Configuration of UAVs

	Horizontal(miles)	Vertical(miles)	Heading(RAD)
UA 1	-3	31	$-\pi/12$
UA 2	28	78	$-7\pi/12$
UA 3	64	78	$-7\pi/12$
UA 4	103	46	$11\pi/12$
UA 5	72	-3	$5\pi/12$
UA 6	36	-3	$5\pi/12$

7.4.2 Simulation Results

Simulation results are illustrated in Fig. 7.8 to Fig. 7.11. Fig. 7.8 shows the searching part. Every UAV searches a subset of the region and render the region with a distinct color. If a target is identified, the corresponding circle becomes red and its coordinates are stored. Fig.

Table 7.3: Static Obstacles

	Static Obs. 1		Static Obs. 2	
Vertex 1	29	18	65	64
Vertex 2	19	13	64	52
Vertex 3	21	9	75	58
Vertex 4	29	9	74	64
Vertex 5	30	15	70	66

7.9 shows the UAVs are traveling through a set of waypoints to targets with a two-column formation. Fig. 7.10 shows the UAVs are traveling through a set of waypoints to targets with a triangular formation. Fig. 7.11 shows two groups of UAVs are patrolling around targets in rigid formations.

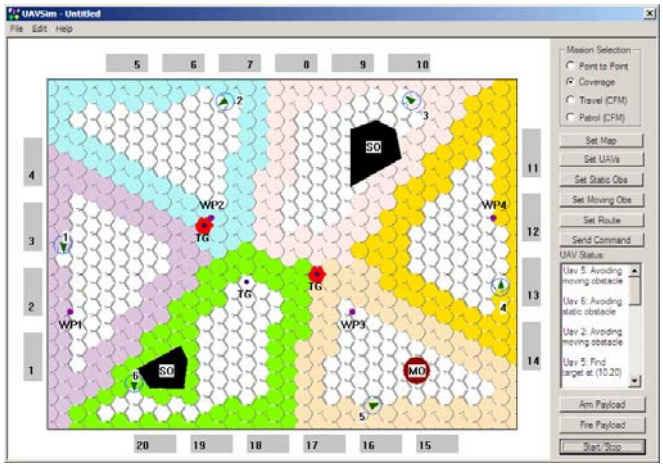


Figure 7.8: UAVs in Searching.

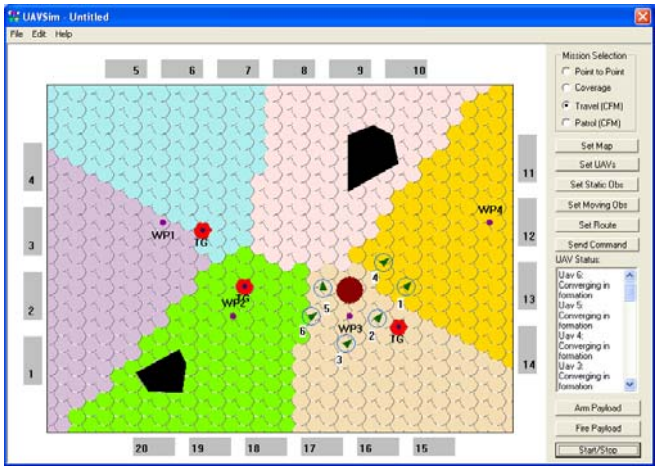


Figure 7.9: UAVs Traveling through Waypoints with A Two-column Formation.

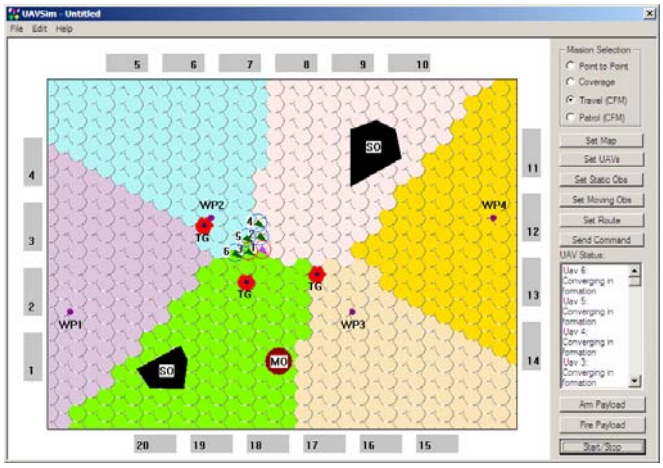


Figure 7.10: UAVs Traveling through Waypoints with A Triangular Formation.

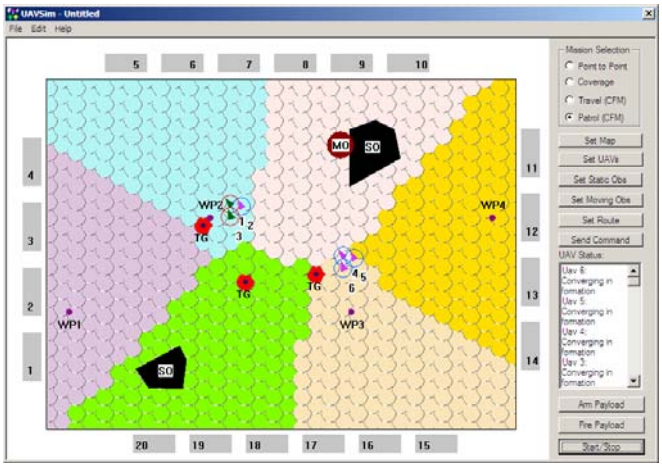


Figure 7.11: UAVs Patrolling in Triangular Formations.

7.5 Conclusion

This chapter proposed a solution for real-time motion planning and formation control for groups of UAVs. The trajectory planning is polynomial parametric. By satisfying boundary conditions, the trajectories can be expressed in terms of a design parameter, which can be chosen to find a collision-free trajectory. For the coverage searching mission, firstly, the operation area is divided into sub-areas according to the Voronoi diagram, i.e. each circle belongs to the closest UAV. Then each UAV works in its own region by repeat the trajectory planning approach between every pair of consecutive waypoints until all waypoints are visited. In the formation control part, for any arbitrary trajectory, its trajectory can be parameterized by the Lagrangian polynomial (7.3.8) and a formation is defined in local coordinate systems and the desired trajectory for each UAV are determined by (7.3.2), and the formation control is given by (7.3.9). Collision avoidance mechanisms are adopted

to avoid potential internal or external collisions. Simulations are conducted to verify the effectiveness of proposed approaches.

CHAPTER 8
CONCLUSION AND FUTURE WORK

In this dissertation, we studied nonholonomic systems for its control design and some applications on robotics. Specifically, the following subjects are investigated:

1. Smooth, time-varying, pure feedback regulation of chained nonholonomic systems. In this research subject, a new feedback design framework is proposed based on a novel global state-scaling transformation. The obtained controls are also inverse optimal with respect to certain quadratic performance index. This design is unique because it has all of the following nice properties at the same time: smooth, time-varying, pure feedback, and exponential converging without oscillations.

2. Feedback stabilization problem of chained nonholonomic systems with input constraints. In this research subject, a novel global asymptotic stable control law is proposed to deal with the input saturation constraints. In literature, this problem is rarely addressed, so this design is a contribution to the controls in this category.

3. Optimal, collision-free motion planning and tracking of nonholonomic robots. In this research subject, parametric trajectories are proposed to searching for an optimal and collision-free path and back-stepping technique is used to derive dynamic tracking controls. The merits of this design is to reduce a 3D planning problem to a 2D problem,

and a close form solution for trajectory and control can be solved by optimizing certain performance index.

4. Coordinated exploration and formation control of multiple unmanned aerial vehicles. In this research subject, coverage searching and formation control algorithms are developed via feedback linearization of the robot's dynamic models. Collision avoidance mechanisms are implemented and verified. The coordinated searching is optimized in the sense that the searching area is divided according to a Voronoi diagram. The collision avoidance is achieved by combining conventional controls and potential fields.

However, it is not a closure of this work. The popular topics on robot navigation and controls have been widely studied and plenty of results are obtained. Generally, motion planning approaches can be divided into the following classes: 1, Reactive approach based on potential fields. 2, Heuristic searching. 3, Parametric trajectory. 4, Some searching based algorithms combined with potential fields. Cooperative control of multi-agent systems is a relatively new subject, two types of general design framework have been developed in recent years, one is based on the graph theory and the other is based on the matrix theory. The two approaches are equivalent in the sense that their necessary and sufficient conditions for convergence are equivalent. However, most of these discussions are based on simplified system model and some ideal assumptions. While the solutions are sound in theory, there could be extra challenges if one considers more practical situations. Some typical such challenges can be summarized into the following cases:

1. A ground robot has a velocity limit and a curvature limit, for an unmanned aerial vehicle (UAV), it cannot fly backward or even too slow. What is the impact on collision avoidance tasks?

2. In a formation/consensus control case, how to incorporate a successful collision avoidance mechanism into existing frameworks to avoid collisions inside the formation as well as avoiding collisions with external obstacles, and based on this, how to deal with actuator saturation effects?

3. If multiple moving obstacles approach simultaneously, what conditions and strategies would be sufficed for robots to evade?

4. There could be communication imperfectness, parameter variations, sensor noise, uncertainties or neglected dynamics in the system, disturbance attenuation and robustness need be considered in practical applications.

My experience from previous researches indicates that the velocity constraint and curvature constraint add significant challenge to the task of real-time collision avoidance. In the case of navigation tasks, the existing techniques, such as the potential fields approach, has local minima problem, and if the number of entities is large, this problem becomes headache even not to consider the constraints. For heuristic searching approach such as A* and D*, one concern is the computational requirements which might not be met by on-board computers, the other concern is that robots dynamic models are not considered, therefore a collision free path may not meet these constraints at all. For some parametric approaches, the control

can be obtained in closed form, which is good for real-time implementation. However, there is no guarantee that the parametric trajectory will meet the constraints either, and the close form solution is fragile to sensor noises and uncertainties in dynamic models. In the case of formation or consensus controls, the cooperative controller is in closed form. One could incorporate the concept of potential fields to the robots and obstacles to obtain additional terms for the controls to handle the obstacles, however there is no guarantee that the obstacles can be avoided due to the velocity and curvature constraints, especially when multiple moving obstacles approach simultaneously. As each single existing technique couldn't solve the collision avoidance problem satisfactorily, innovative thinking is needed and more complex controllers needs be constructed. A primitive thought is to use them as a combination: to take advantage of their merits and avoid their weaknesses. Even more, one might take the artificial intelligence and computational geometry algorithms into the scenario and help to make decision in realtime. Considering current status of researches in the related fields and the existing challenges, in the near future, I would like to work on the following subjects:

1. Cooperation and coordination of multi-agent systems.

2. Collision avoidance mechanisms with actuator limitations for complex dynamic environments.

3. Design nonlinear and optimal regulation and tracking controllers.

4. Design robust and/or adaptive controllers to handle uncertainties, disturbance, parameter variations, neglected dynamics or communication imperfectness.

5. Implement optimal state estimation through Kalman filtering, H-infinity filtering or other nonlinear approaches.

6. Implement software and hardware platforms for experimental verification and validation purpose.

LIST OF REFERENCES

[1] Arinaga S, Nakajima S, Okabe H, Ono A, Kanayama Y. A Motion Planning Method for an Auv. Proc. Symposium on AUV Technology, Monterey, CA, USA, June 1996, pp. 477-484.

[2] Astolfi A. Discontinuous control of nonholonomic systems. Systems & Control Letters, 1996, 27:37-45.

[3] Bacciotti A. Local Stabilizability of Nonlinear Control Systems. World Scientific, Singapore, 1992.

[4] Barisic M, Vukic Z, Miskovic N. Kinematic Simulative Analysis of Virtual Potential Field Method for AUV Trajectory Planning. Proc. Mediterranean Conference on Control & Automation, Athens, Greece, July 2007, pp. 1-6.

[5] Barraquand J, Latombe JC. On nonholonomic mobile robots and optimal maneuvering. Revue d'Intelligence Artificielle vol.3, no.2, pp. 77-103, 1989.

[6] Bloch A, Reyhanoglu M, McClamroch NH. Control and stabilization of nonholonomic dynamic systems. IEEE Trans. on Auto Contr., 1992, 37:1746-1757.

[7] Bortoff S. Path planning for UAVs. Proc. 2000 American Control Conference, Chicago, IL, June 2000, pp. 364-368.

[8] Breivik M, Fossen T. Principles of Guidance-Based Path Following in 2D and 3D. Proc. 44th IEEE International Conference on Decision and Control, and the European Control Conference, Seville, Spain, Dec. 2005, pp. 627-634

[9] Brockett RW. Asymptotic stability and feedback stabilization. Differential Geometric Control Theory, 1983, pp. 181-191.

[10] Canudas de Witt C, Khennouf H. Quasi-continuous stabilizing controllers for nonholonomic systems: Design and robustness considerations. Proceedings of the European Control Conference, 1995.

[11] Chuang J. Potential-Based Modeling of Three-Dimensional Workspace for Obstacle Avoidance, IEEE Trans. on Robotics and Automation, 1998, 14, 5:778-785.

[12] Chung C, Saridis G. Path Planning for an Intelligent Robot by The Extended Vgraph Algorithm. Proc. IEEE international Symposium on Intelligent Control, Albany, NY, Sept. 1989, pp. 544-549.

[13] Ding F, Jiao P, Bian X, Wang H. AUV Local Path Planning Based on Virtual Potential Field. Proc. IEEE International Conference on Mechatronics and Automation, Niagara Falls, Canada, July 2006, pp. 1711-1716.

[14] Fossen TI. Guidance and Control of Ocean Vehicles. New York, Wiley, 1994.

[15] Fujimura K. A Hierarchical Strategy for Path Planning Among Moving Obstacles. IEEE Trans. on Robotics and Automation, 1989, 5:61-69.

[16] Grimm G, Teel AR, Zaccarian L. The l_2 anti-windup problem for discrete time linear systems: Definition and solutions. Systems & Control Letters, 2008; 57:356-364.

[17] Guo Y, Qu Z. Coverage control for a mobile robot patrolling a dynamic and uncertain environment. 5th World Congress on Intelligent Control and Automation, Hangzhou, China, Jan.15-19, 2004.

[18] Herman M. Fast, Three-Dimensional, Collision-Free Motion Planning. Proc. IEEE International Conference on Robotics and Automation, San Francisco, CA, Apr. 1986 , pp. 1056-1063.

[19] Hu T, Teel AR, Zaccarian L. Anti-windup synthesis for linear control systems with input saturation: Achieving regional, nonlinear performance. Automatica, 2008, 44:512-519.

[20] Hwang Y, and Ahuja N. A Potential Field Approach to Path Planning, IEEE Trans. on Robotics and Automation, 1992, 8, 1:23-32.

[21] Isidori A. Nonlinear Control Systems. Springer-Verlag, Berlin, 1995.

[22] Jacob G. Motion planning by piecewise constant or polynomial inputs. Proc. Nonlinear Contr. Syst. Design Symp., Bordeaux, France, June 1992, pp. 628-633.

[23] Jadbabaie A, Lin J, Morse, AS. Coordination of groups of mobile autonomous agents using nearest neighbor rules. IEEE Trans. on Automatic Control, 48:988-1001.

[24] Jiang ZP. Robust exponential regulation of nonholonomic systems with uncertainties. Automatica, 2000, 36:189-209.

[25] Jiang ZP, Lefeber E, Nijmeijer H. Saturated stabilization and tracking of a nonholonomic mobile robot. System & Control Letters, 2001, 42:327-332.

[26] Judd B, Mclain W. Spline Based Path Planning for Unmanned Air Vehicles. AIAA Guidance, Navigation, and Control Conference and Exhibit, Montreal, Canada, Aug 2001, AIAA-2001-4238

[27] Khalil H. Nonlinear Systems, 3rd ed. NJ: Prentice-Hall, Upper Saddle River, 2002.

[28] Khatib O. Real-time Obstacle Avoidance for Manipulators and Mobile Robots. International Journal of Robotics Research, 1986, 5, 1:90-98.

[29] Kitamura Y, Tanaka T, Kishino F, Yachida M. 3-D Path Planning in a Dynamic Environment Using an Octree and an Artificial Potential Field. IEEE International Conference on Intelligent Robots and Systems, Pittsburgh, USA, Aug 1995, pp. 474-481

[30] Kolmanovsky I, McClamroch H. Developments in nonholonomic control problems. IEEE Control Systems Mag. 1995, 6:20-36.

[31] Kothare MV, Campo PJ, Morari M, Nett CN. A unified framework for the study of anti-windup designs. Automatica, 1994, 30:1869-1883.

[32] Krstic M, Kanellakopoulos I, Kokotovic PV. Nonlinear and Adaptive Control Design. Wiley, New York, 1995.

[33] Kyriakopoulos K, Kakambouras P, Krikelis N. Potential Fields for Nonholonomic Vehicles. Proc. IEEE International Symposium on Intelligent Control, Monterey, CA, Aug. 1995, pp. 461-465

[34] Lafferriere G, Sussmann HJ. Motion planning for controllable systems without drift. Proc. IEEE Int. Conf. Robotics and Automation, Sacramento, California, Apr. 1991, pp. 1148-1153.

[35] Lafferriere G. A general strategy for computing controls of systems without drift. Proc. Proc. 30th Conf. Decis. Contr., Brighton, England, Dec. 1991, pp. 1115-1120.

[36] Laiou M, Astolfi A. Quasi-smooth control of chained systems. Proceeding of the American Control Conference, San Diego, pp. 3940-3944.

[37] Laumond JP. Feasible trajectories for mobile robots with kinematic and environment constraints. Proc. Int. Conf. Intelligent Autonomous Syst, Amsterdam, The Netherlands, 1986, pp. 346-354.

[38] Laumond JP. Finding collision-free smooth trajectories for a nonholonomic mobile robot. 10th Int. Joint Conf. Artificial Intelligence, Milano, Italy, 1987, pp. 1120-1123.

[39] Laumond JP, Taix M, Jacobs P. A motion planner for car-like robots based on a mixed global/local approach. Proc. Int. Conf. Intelligent Robots Syst, Japan, 1990, pp. 765-773.

[40] Lin W. Time-varying feedback control of nonaffine nonlinear systems without drift. System & Control Letters, 1996, 29, pp. 101-110.

[41] Lin Z, Brouchke M, Francis B. Local control strategies for groups of mobile autonomous agents. IEEE Trans. on Automatic Control, 49:622-629.

[42] Luo J, Tsiotras P. Control design for chained-form systems with bounded inputs. System & Control Letters, 2000, 39:123-131.

[43] March N, Alamir M. Discontinuous exponential stabilization of chained form systems. Automatica, 2003, 39:343-348.

[44] Marc N, Hably A. Global stabilization of multiple integrators with bounded controls. Automatica, 2005, 41:2147-2152.

[45] M'Closkey RT, Murray RM. Exponential Stabilization of Driftless Nonlinear Control System Using Homogeneous Feedback. IEEE Trans. on Automatic Control, 1997, 42:614-628.

[46] M'Closkey RT, Murray RM. Exponential stabilization of driftless nonlinear control system. Systems & Control Letters, 2000, 39:123-131.

[47] Monaco S, Normand-Cyrot D. An introduction to motion planning under multirate control. Proc. 31th Conf. Decis. Contr., Tucson, AZ, Dec. 1992, pp. 1780-1785.

[48] Moreau L. Leaderless coordination via bidirectional and unidirectional time-dependent communication. Proceedings of the 42nd IEEE Conference on Decision and Control, Maui, Hawaii.

[49] Murray RM and Sastry SS. Nonholonomic motion planning: Steering using sinusoids. IEEE Trans. on Automatic Control, 1993, 38:700-716.

[50] Nakamura Y, Savant S. Nonlinear Tracking Control of Autonomous Underwater Vehicles. Proc. IEEE International Conference on Robotics and Automation, Nice, France, 1992, pp. A4-A9

[51] Narendra KS, Annaswamy AM. Stable Adaptive Systems. Prentice Hall, Englewood Cliffs, NJ., 1989.

[52] Nijmeijer H, Van der Schaft AJ. Nonlinear Dynamical Control Systems. Springer, Berlin, 1990.

[53] Nilsson NJ, Principles of Artificial Intelligence, Tioga Publishing Company, 1980.

[54] Pomet JB. Explicit design of time-varying stabilizing control laws for a class of controllable systems without drift. Systems & Control Letters, 1992, 18:147-158.

[55] Qu Z. Robust Control of Nonlinear Uncertain Systems. Wiley-InterScience, 1998.

[56] Qu Z, Wang J, Plaisted CE. A new analytical solution to mobile robot trajectory generation in the presence of moving obstacles. IEEE Transactions on Robotics, 2004, 20:978-993.

[57] Qu Z, Wang J, Plaisted CE, Hull RA. Global-stabilizing near-optimal control design for nonholonomic chained systems. IEEE Transactions on Automatic Control, 2006, 51:1440-1456.

[58] Qu Z, Wang J, Hull RA, Martin J. Continuous and Inverse Optimal Control Design for Chained Systems. Optimal Control Application and Methods, 2008, 1:1-25.

[59] Qu Z, Wang J, Hull RA. Cooperative control of dynamical systems with application to autonomous vehicles. Submitted to IEEE Transactions on Automatic Control

[60] Repoulilas F, Papadopoulos E. Three Dimensional Trajectory Control of Underactuated AUVs. Proc. of the European Control Conference 2007, Kos, Greece, July, 2007, ppl 3492-3499

[61] Ren, W. Simulation and Experimental Study of Consensus Algorithms for Multiple Mobile Robots with Information Feedback. Intelligent Automation and Soft Computing, 1:73-87, 2008.

[62] Reyhanoglu M, Cho S, McClamroch N, Kolmanovsky I. Discontinuous feedback control a planar rigid body with an underactuated degree of freedom. Proceedings of the 37th Conference on Decision and Control, 1998, pp. 433-438.

[63] Reynolds CW. Flocks, herds, and schools: a distributed behavioral model. Computer Graphics (ACM SIGGRAPH 87 Conference Proceedings), 21(4):25-34.

[64] Samson C. Control of chained systems: Application to path following and time-varying point-stabilization of mobile robots. IEEE Trans. on Automatic Control, 1995, 40:64-77.

[65] Stentz A. Optimal and efficient path planning for partially-known environments. IEEE International Conference on Robotics and Automation, May 1994.

[66] Stentz A. The Focussed D* Algorithm for Real-Time Replanning. Proceedings of the International Joint Conference on Artificial Intelligence, August 1995.

[67] Sussmann HJ, Sontag ED, Yang Y. A general result on the stabilization of linear system using bounded controls. IEEE Transactions on Automatic Control, 1994, 12:2411-2425.

[68] Teel AR. Global stabilization and restricted tracking for multiple integrators with bounded controls. Systems & Control Letters, 1992, 18:165-171.

[69] Teel AR, Murray RM, Walsh G. Nonholonomic control systems: from steering to stabilization with sinusoids. Proceedings of the 31st IEEE Conference on Decision and Control, Tucson, Arizona, 1992, pp. 1603-1609.

[70] Tian YP, Li S. Smooth exponential stabilization of nonholonomic systems via time-varying feedback. Proceedings of the 39th IEEE Conference on Decision and Control, Sydney, 2000, pp. 1912-1917.

[71] Valenciaga F, Puleston P, Calvo O, Acosta G. Trajectory Tracking of the Cormoran AUV Based on a PI-MIMO Approach. Oceans'07, Aberdeen Scotland, June, 2007, pp. 1-6.

[72] Vicsek T, Czirok A, Jacob EB, Cohen I, Shochet O. Novel type of phase transition in a system of self-driven particles. Physical Review Letters, 75:1226-1229.

[73] Warren C. A Technique for Autonomous Underwater Vehicle Route Planning. IEEE Trans. of Oceanic Engineering, 1990, 15,3:199-204.

[74] Wang Y, Lane D. Subsea Vehicle Path Planning Using Nonlinear Programming and Constructive Solid Geometry. IEE proceedings-Control Theory and Applications, 1997, 2:143-152.

[75] Yang J, Daoui A, Qu Z, Wang J. An Optimal and Real-Time Solution to Parameterized Mobile Robot Trajectories. Proc. IEEE International Conference on Robotics and Automation, Barcelona, Spain, April. 2005, pp. 4412-4417.

[76] Yuan H, Yang J, Qu Z, Kaloust J. An Optimal Real-time Motion Planner for Vehicles with a Minimum Turning Radius. The 6th World Congress on Intelligent Control and Automation, Dalian, China, June 21-23, 2006.

[77] Yuan H, Gottesman V, Qu Z, Falash M, Pollak E, Chunyu J. Cooperative Formation Flying in Autonomous Unmanned Air Systems with Application to Training. The 7th International Conference on Cooperative Control and Optimization, Gainesville, Florida, Jan.31-Feb.2, 2007.

[78] Yuan H, Qu Z. Design of An Experimental Testbed for Supervisory Control of Multiple Cooperative Controlled Vehicles. 2008 IEEE International Conference on Distributed Human-Machine Systems, Athens, Greece, March 9-12, 2008.

[79] Yuan H, Qu Z. Continuous Time-Varying Pure Feedback Control for Chained Nonholonomic Systems with Exponential Convergent Rate. 17th IFAC World Congress, Seoul, Korea, 2008.

[80] Zhang Q. A Hierarchical Global Path Planning Approach for AUV Based on Genetic Algorithm. Proc. IEEE International Conference on Mechatronics and Automation, Luoyang, China, June 2006, pp. 1745-1750.

[81] Zhou B, Duan GR. Global stabilization of linear systems via bounded controls. Systems & Control Letters, 2009, 58:54-61.