DYNAMICAL FORMULATIONS AND CONTROL OF AN AUTOMATIC RETARGETING SYSTEM

A Thesis

by

MICHAEL CHARLES SOVINSKY

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

December 2005

Major Subject: Aerospace Engineering

DYNAMICAL FORMULATIONS AND CONTROL OF AN AUTOMATIC RETARGETING SYSTEM

A Thesis

by

MICHAEL CHARLES SOVINSKY

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,	John Hurtado
Committee Members,	Srinvas Vadali
	John Valasak
	Reza Langari
Head of Department,	Helen Reed

December 2005

Major Subject: Aerospace Engineering

ABSTRACT

Dynamical Formulations and Control of an Automatic Retargeting System.

(December 2005)

Michael Charles Sovinsky, B.S., Texas A&M University

Chair of Advisory Committee: Dr. John E. Hurtado

The Poincaré equations, also known as Lagrange's equations in quasi coordinates, are revisited with special attention focused on a diagonal form. The diagonal form stems from a special choice of quasi velocities that were first introduced by Georg Hamel nearly a century ago. The form has been largely ignored because the quasi velocities create so-called Hamel coefficients that appear in the governing equations and are based on the partial derivative of the mass matrix factorization. Consequently, closed-form expressions for the Hamel coefficients can be difficult to obtain and relying on finite-dimensional, numerical methods are unattractive. In this thesis we use a newly developed operator overloading technique to automatically generate the Hamel coefficients through exact partial differentiation together with numerical evaluation. The equations can then be numerically integrated for system simulation. These special Poincaré equations are called the Hamel Form and their usefulness in dynamic modeling and control is investigated.

Coordinated control algorithms for an automatic retargeting system are developed in an attempt to protect an area against direct assaults. The scenario is for a few weapon systems to suddenly be faced with many hostile targets appearing together. The weapon systems must decide which weapon system will attack which target and in whatever order deemed sufficient to defend the protected area. This must be performed in a real-time environment, where every second is crucial. Four different control methods in this thesis are developed. They are tested against each other in computer simulations to determine the survivability and thought process of the control algorithms. An auction based control algorithm finding targets of opportunity achieved the best results.

To Craig and Pattie Sovinsky, the greatest parents imaginable.

ACKNOWLEDGMENTS

First and foremost I would like to thank my advisor, Dr. John E. Hurtado. His guidance and support throughout my graduate career here at Texas A&M is immeasurable. The rest of my committee and other professors, including Dr. Valasek, Dr. Vadali, Dr. Langari, Dr. Junkins and Dr. Bhattacharyya, all opened their doors and offered assistance whenever necessary.

Of course my friends and colleagues here at A&M, who are too many to name, created a fabulous environment to enjoy this work day in and day out. A very special thanks is extended towards the Spacecraft Technology Center, especially to Mrs. Diane Hurtado. Through their funding and support I was able to participate in work that was interesting and enjoyable.

And finally, I would not be who I am today without Vanessa, who brings an insurmountable joy into my life which continues to grow day after day.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Dynamics and control of mechanical systems are important issues across a wide field of applications. A complete understanding of a system requires both topics to be explored. An adequate control algorithm requires that the dynamics of the system be understood. And if the dynamics of these systems can be understood clearer and computed with less error and effort, this will allow more precise control. This thesis will examine both aspects of these systems.

The dynamics portion will examine a formulation to simplify the equations of motion of mechanical systems, named the Hamel Form. Simplified equations of motion can lead to many benefits. A better physical understanding of the system and a reduction in integration error are a few examples of these benefits. The end result of this formulation is a diagonalized form of equations of motion. Traditionally, the mass matrix is a populated, configuration-dependent matrix which is then inverted. This diagonalized form eliminates the need to perform this costly inversion. An automatic differentiation tool, OCEA, will also be utilized to aid in the creation of the Hamel Form.

The second part of this thesis will analyze cooperative control techniques applied to an automatic retargeting system. A scenario is created where only a few weapon systems are faced with the task of eliminating many targets. This control algorithm needs to autonomously determine which weapon system follows which target, while preventing targets from passing beyond the weapon systems. This is to be accomplished in a real-time environment with attention also paid to a minimal amount of

The journal model is *IEEE Transactions on Automatic Control*.

communication necessary between the weapon systems.

A subset of this scenario is target identification using one or more pan and tilt cameras. Due to physical constraints with moving the actual camera system, a set of mirrors will be used to pan and tilt the image beam into the camera. A control algorithm will be developed to successfully point the camera system at a specified point.

CHAPTER II

INTRODUCTION TO HAMEL FORM

From the beginning, scholars have sought transformations that simplify the governing equations of motion of mechanical systems. Analysis and experience has shown that transformations performed at the velocity level, as opposed to coordinate level, are most fruitful. When considering transformations at the velocity level one is naturally lead to so-called *quasi velocities* [1, 2]. The adjective "quasi" serves to remind one that the integration of such velocities will lead to coordinates (quasi coordinates) that can not be used to describe the system configuration. This is unlike *true* or generalized velocities, which are nothing more than the first time derivative of true or generalized coordinates: the integration of true velocities yield the system configuration.

Perhaps the most famous quasi velocities are the components of the rigid-body angular-velocity vector when coordinatized along body-fixed axes. Euler discovered these and developed his equations for rigid-body rotational motion. His equations are sleek whereas a traditional Lagrangian approach that uses, for example, a set of Euler angles and their derivatives as true coordinates and velocities leads to a highly nonlinear, unattractive set of equations. Gibbs, Volterra, Poincaré, and others [2] deeply investigated the role and implications of quasi velocities in dynamic formulations, and Kane's method [3] uses quasi velocities (he prefers the term *generalized speeds*) to a great extent to generate simple-looking governing equations.

One motivation for seeking simplified equations of motion is to reduce the appearance of complex nonlinear terms: the reasons are twofold. For low-dimensioned systems, reducing the appearance of complex nonlinear terms sometimes allows one to gain insight into the resulting motion using analytical techniques. For high-dimensioned systems, reducing the appearance of complex nonlinear terms can allow faster simulation run times [4].

One approach to obtaining simplified equations of motion is to generate *diagonalized forms*, i.e., forms that produce a diagonal mass matrix. When the governing equations of motion are developed for a general nonlinear mechanical system using the traditional Lagrangian treatment, a symmetric, positive-definite, configuration-dependent mass matrix is generated, which must be inverted at each time step when using traditional explicit integration techniques like Runge-Kutta methods. Diagonalized forms alleviate the costly mass-matrix inversion, but always introduce another calculation difficulty.

A. Other Diagonalized Forms

Two separate but similar diagonalized forms have been proposed recently by Jain and Rodriguez [5] and Junkins and Schaub [6]. Their investigations are highlighted below.

1. Mass Matrix Factorization

Jain and Rodriguez achieve a diagonal form for their quasi velocities via a special mass matrix factorization.

$$M = (I + H\phi K)D(I + H\phi K)^T \equiv m(q)\, m(q)^T \tag{2.1}$$

In this matrix equation, I represents the identity matrix, whereas H, K, D and ϕ are spatial operators that are found recursively by spatial filtering and smoothing algorithms. Jain and Rodriguez have traded the need to invert the mass matrix with having to solve a recursive set of equations. Moreover, their technique applies only to tree-like, articulated multibody systems and other kinematically recursive topologies.

Jain and Rodriguez only consider natural systems, wherein the kinetic-energy function is a quadratic function of the generalized velocities only. Their quasi velocities are defined via $\nu = m(q)\,\dot{q}$ and their resulting equations of motion take a simple form.

$$\dot{\nu}_k + c_k(q, \nu) = \epsilon_k \tag{2.2}$$

Here, $c(q, \nu)$ is their *Coriolis force vector* and depends on the mass matrix factorization. The elements of c are given by the following expression.

$$c_k(q, \nu) = \ell_{kr} \left(\dot{m}_{ri}\nu_i - \frac{1}{2}\frac{\partial M_{ij}}{\partial q_k}\,\dot{q}_i\dot{q}_j \right) \tag{2.3}$$

The matrix $\ell(q)$ appearing here is the inverse of $m(q)$.

Jain and Rodriguez go on to demonstrate several important facets of their development, viz., that the Coriolis force vector depends quadratically on the quasi velocities ν, and that the Coriolis force vector does no mechanical work.

2. Eigenstructure Quasi Velocity Formulation

Junkins and Schaub achieve a diagonal form for their quasi velocities via a spectral decomposition of the mass matrix.

$$M = C^T DC; \quad C^T C = 1; \quad D = \mathrm{diag}(\lambda) \equiv S^T S \tag{2.4}$$

Their quasi velocities are defined via $\eta = SC\,\dot{q}$. Junkins and Schaub have essentially traded the need to invert the mass matrix with having to solve one additional matrix differential equation, viz., $\dot{C} = -\Omega C$. This new differential equation is based on the development of their new quasi velocities, which are related to the eigenvalues of the mass matrix. Some difficulties arise when the distinct eigenvalues are near each other.

Their resulting equations of motion take the following matrix form.

$$\dot{\eta} + S^{-1}(\Omega S + \dot{S})\eta - S^{-1}C\left(\frac{1}{2}\dot{q}^T\frac{\partial M}{\partial q}\dot{q} - \frac{\partial G^T}{\partial q}\dot{q}\right) = S^{-1}CF \tag{2.5}$$

Unlike Jain and Rodriguez, Junkins and Schaub consider more general nonnatural systems. The vector function G appearing in equation (2.5) is the coefficient vector that multiplies the generalized velocities linearly in the kinetic-energy function, and F is composed of potential forces, nonpotential forces, $\dot{G}$, and that part of the kinetic-energy function that is independent of generalized velocities. Of course, with a suitable definition of variables, the form of Junkins and Schaub could be written as simple-looking as the form of Jain and Rodriguez.

CHAPTER III

THE HAMEL FORM

The kinetic-energy function for a general mechanical system can be expressed as the addition of three terms.

$$T(q,\dot{q},t) = T_2(q,\dot{q}) + T_1(q,\dot{q},t) + T_0(q,t) \tag{3.1}$$

The leading term is a quadratic function of the n-dimensional true velocity vector, $\dot{q}$, whereas the middle term is linear in $\dot{q}$ and the final term is independent of $\dot{q}$.

$$T_2 = \frac{1}{2}M_{ij}(q)\,\dot{q}_i\dot{q}_j; \quad T_1 = N_k(q,t)\,\dot{q}_k; \quad T_0 = m_0(q,t) \tag{3.2}$$

It is straightforward to adopt $n+1$ coordinates where $\dot{q}_{n+1}$ =dt/dt = 1 so that the kinetic-energy function can be written as a quadratic function of a $(n+1)$-dimensional augmented vector.

$$2T = \begin{bmatrix}\dot{q}_1 & \cdots & \dot{q}_n & 1\end{bmatrix} \times \begin{bmatrix} M_{11} & \cdots & M_{1n} & N_1 \\ \vdots & \ddots & & \vdots \\ M_{n1} & \cdots & M_{nn} & N_n \\ N_1 & \cdots & N_n & m_0 \end{bmatrix} \begin{bmatrix}\dot{q}_1 \\ \vdots \\ \dot{q}_n \\ 1\end{bmatrix} \tag{3.3}$$

This equation can be written more compactly.

$$2T(q,\dot{q},t) = M_{\alpha\beta}(t,q)\,\dot{q}_\alpha\dot{q}_\beta \tag{3.4}$$

Now consider a nonsingular linear transformation from the true velocities to a set of quasi velocities, ω.

$$\dot{q}_i = A_{ij}(q,t)\,\omega_j + a_i(q,t) \tag{3.5}$$

This equation represents the forward mapping, whereas the inverse mapping is given as follows.

$$\omega_j = B_{ji}(q,t)\,\dot{q}_i + b_j(q,t) \tag{3.6}$$

These mappings can be written in a compact manner using the augmented vector.

$$\dot{q}_\alpha = A_{\alpha\beta}(q,t)\,\omega_\beta; \qquad \omega_\alpha = B_{\alpha\beta}(q,t)\,\dot{q}_\beta; \tag{3.7}$$

Using the first of these equations leads to a kinetic-energy function dependent on the true coordinates, q, the quasi velocities, ω, and time.

$$2T^*(q,\omega,t) = M_{\alpha\beta}A_{\alpha\nu}A_{\beta\mu}\,\omega_\nu\,\omega_\mu \tag{3.8}$$

The particular set of quasi velocities have not yet been specified, that is, $A_{\alpha\beta}$ have not yet specified. Consequently, a special set is sought. It is desired that the quasi velocities to be such that $2T^*$ is independent of the true coordinates, q, which implies the following $(n+1)$-dimensional matrix equation.

$$A^T M A = 1 \tag{3.9}$$

This matrix equation can be satisfied if the Cholesky decomposition [7] of M is used to define the quasi velocities. That is, suppose $M = B^T B$, where B is an $(n+1)$-dimensional, upper-triangular matrix, and suppose this matrix B is used to define a set of quasi velocities via the second expression of equation (3.7). As a consequence of this choice, equation (3.9) will be satisfied and a simple form for the kinetic-energy function is obtained.

$$2T^* = \omega_\mu\,\omega_\mu \quad \text{or} \quad 2T^* = \omega^T\omega \tag{3.10}$$

At this point return to Lagrange's equations of motion in terms of quasi variables, which are also known as the Poincaré equations of motion. [8]

$$\frac{d}{dt}\left(\frac{\partial T^*}{\partial \omega_\nu}\right) + \gamma^\beta_{\nu\alpha}\omega_\alpha \frac{\partial T^*}{\partial \omega_\beta} - A_{\beta\nu}\frac{\partial T^*}{\partial q_\beta} = \pi_\nu \tag{3.11}$$

Note that our special set of quasi velocities simplifies this equation because $\partial T^*/\partial q = 0$. Furthermore, rewriting this equation in terms of the original n-dimensional vector components instead of the $(n+1)$-dimensional augmented vector components, and using the special form of T^*, leads to the following kinetic equations of motion.

$$\dot{\omega}_k + \gamma^j_{ki}\,\omega_i\omega_j + \gamma^j_k\omega_j = \pi_k \tag{3.12}$$

In these equations, π_k, which appears on the right-hand side, is the kth nonholonomic impressed force [2] and is composed of potential and nonpotential forces. The three-index symbol γ^j_{ki} represents the *three-index Hamel coefficients* and may be defined via the true and quasi velocity transformation matrices.

$$\gamma^r_{ka} \equiv \left(\frac{\partial B_{rj}}{\partial q_i} - \frac{\partial B_{ri}}{\partial q_j}\right) A_{jk}A_{i\alpha} \tag{3.13}$$

These coefficients are also known as the *Hamel-Volterra transitivity coefficients*, or the *Ricci-Boltzmann-Hamel three-index symbols* [2, 9]. The two-index symbol γ^j_k are *two-index Hamel coefficients* and are also related to the true and quasi velocity transformation matrices. [2]

$$\gamma^j_k \equiv \left(\frac{\partial B_{jb}}{\partial q_c} - \frac{\partial B_{jc}}{\partial q_b}\right) A_{bk}a_c + \left(\frac{\partial B_{jb}}{\partial t} - \frac{\partial b_j}{\partial q_b}\right) A_{bk} \tag{3.14}$$

A. Control Using Hamel Form

The Hamel equations of motion may be used to design stabilizing and regulating controls. For a stabilizing control, let the system kinetic-energy function be a Lyapunov function. Then, according to the work/energy-rate principle [10], $\dot{V} = \dot{T} = \dot{q}^T F$, where F are the generalized (or holonomic impressed) forces. In terms of quasi velocities and nonholonomic impressed forces, this reads as $\dot{V} = \omega^T \pi$, so choosing the nonholonomic impressed forces as the negative of the quasi velocities will stabilize the system.

For a regulating control (e.g., to the origin), let a Lyapunov function be composed of the system kinetic-energy function and a suitable (possibly fictitious) potential function: $V(q, \omega) = T(q, \omega) + U(q)$. The time derivative of V gives $\dot{V} = \omega^T(\pi + A^T(\partial U/\partial q)\,)$. Choosing the nonholonomic impressed forces so that the parenthetical factor equals the negative of the quasi velocities will regulate the system, provided that the only solution to $\partial U/\partial q = 0$ is the desired regulation point.

B. Observations and Remarks over Hamel Form

The complete motion of the dynamical system is governed by the kinetic equations, equation (3.12), and the kinematic equations, equation (3.5). These $2n$ first-order, ordinary differential equations replace the traditional n second-order, ordinary differential equations that result from Lagrange's equations of motion in terms of q and $\dot{q}$.

The special choice of quasi velocities, and the resulting form, were first introduced by Georg Hamel nearly a century ago [2, 11]. He called his equations *Lagrange-Euler equations* and we refer to them here as *the Hamel Form.* This choice of quasi velocities has been largely ignored because the quasi velocities create Hamel coefficients that are based on the partial derivative of the mass matrix factorization, see equation (3.13).

The Hamel coefficients arise solely because of the "nongenuine" nature of *quasi coordinates*, σ (the time derivative of the quasi coordinates equal the quasi velocities).

Locally, the differentials of quasi coordinates are related to the differentials of true coordinates via $d\sigma = B\,dq$. A necessary and sufficient condition for σ to be true coordinates is the satisfaction of integrability conditions given by the theory of Pfaffian forms [9].

$$(\partial B_{rj}/\partial q_i - \partial B_{ri}/\partial q_j) = 0 \tag{3.15}$$

When these integrability conditions are *not* satisfied, then no finite σ exist, only the differentials $d\sigma$. The left side of equation (3.15) appears in the definition of the three-index Hamel coefficients, and hence the dissatisfaction of the integrability conditions, i.e., the nongenuine nature of the quasi coordinates, gives rise to the Hamel coefficients. The integrability conditions make clear the following: quasi coordinates implies nonzero Hamel coefficients; true coordinates implies zero Hamel coefficients.

It is straightforward to show that the three-index Hamel coefficients are skew-symmetric in the lower indices: $\gamma^r_{ak} = -\gamma^r_{ka}$. The skew-symmetry property can be used to show that the three-index Hamel coefficient term (Hamel's Coriolis force vector) is orthogonal to the quasi velocities and is therefore nonworking.

$$\begin{aligned} \omega_k \gamma^j_{ki}\, \omega_i \omega_j &= \frac{1}{2}\omega_k \left(\gamma^j_{ki} - \gamma^j_{ik}\right) \omega_i \omega_j \\ &= \frac{1}{2}\omega_k \gamma^j_{ki} \omega_i \omega_j - \frac{1}{2}\omega_i \gamma^j_{ik} \omega_k \omega_j \\ &= 0 \end{aligned} \tag{3.16}$$

(The indices within each term are repeated and therefore their labels may be freely changed.) It is well known that the Coriolis force term in the Lagrangian treatment does mechanical work according to

$$\dot{\theta}^T C(\theta, \dot{\theta}) = \frac{1}{2}\dot{\theta}^T \left[\dot{\theta}^T \frac{\partial M}{\partial \theta}\, \dot{\theta}\right] \tag{3.17}$$

For *catastatic* systems [1] (i.e., $a_j = b_j = 0$), the two-index Hamel coefficients simplify (equation (3.14)):

$$\gamma_k^j = A_{bk} \left(\partial B_{jb} / \partial t \right) \tag{3.18}$$

whereas for *scleronomic* systems[1] (i.e., $a_j = b_j = 0$ and $\partial B_{jb}/\partial t = \partial A_{jb}/\partial t = 0$), the two-index Hamel coefficients vanish altogether:

$$\gamma_k^j = 0 \tag{3.19}$$

The special set of quasi velocities depend on the Cholesky decomposition of $M_{\alpha\beta}$. This decomposition is guaranteed to exist because $M_{\alpha\beta}$ is symmetric, positive definite. The Cholesky decomposition takes advantage of the symmetry of the mass matrix and requires $n^3/6$ (addition and multiplication) mathematical operations. The inversion of the upper-triangular matrix B (it's needed to compute the Hamel coefficients) requires an additional $n^3/6$ operations [12]. The eigenstructure quasi velocity formulation of Junkins and Schaub uses the spectral decomposition of the mass matrix, which is also requires $\mathcal{O}(n^3)$ operations. The approach of Jain and Rodriguez requires $\mathcal{O}(n)$ operations, but their method is tailored for tree-like multibody systems.

CHAPTER IV

COMPUTING HAMEL COEFFICIENTS

A diagonal form for the equations of motion of general mechanical systems is achieved, as explained in the previous chapter, using quasi velocities that are based on a Cholesky decomposition of the mass matrix. Unfortunately, like the forms of Jain and Rodriguez and Junkins and Schaub, the quasi velocities create coefficients that are based on the partial derivative of the mass matrix factorization. Consequently, closed-form expressions for the coefficients can be difficult to obtain. Unlike the spatial filtering and smoothing method of Jain and Rodriguez or the additional matrix differential equation approach (and the associated concern with crossing eigenvalues) of Junkins and Schaub, a newly developed operator overloading technique to automatically generate the coefficients through exact partial differentiation together with numerical evaluation will be used.

This newly developed automatic differentiation program is called OCEA (Object-Oriented, Coordinate Embedding Approach). The OCEA package, currently programmed as a FORTRAN90 (F90) extension, is an object-oriented, automatic differentiation manipulation package [13].

The strength and benefit of automatic differentiation is that it is an approach that invokes the chain rule automatically and takes place in the background, without user intervention. The key to the OCEA method is that certain variables are declared as *embedded.* These variables represent abstract data types, where hidden dimensions (background arrays) are used for storing and manipulating partial derivative calculations. For example, in a "second-order" version, OCEA replaces each scalar variable,

f, that is a function of an embedded variable, x, with a differential n-tuple.

$$\begin{aligned} f(x) &= \begin{bmatrix} f(x) & \partial f/\partial x & \partial^2 f/\partial x^2 \end{bmatrix} \\ &= \begin{bmatrix} f(x) & \nabla f(x) & \nabla^2 f(x) \end{bmatrix} \end{aligned} \tag{4.1}$$

The introduction of the abstract differential n-tuple allows the computer to continue to manipulate each scalar variable in a conventional way, even though the first- and higher-order partial derivatives are attached to the scalar variable in a hidden way. The individual objects of f are easily extracted by proper variable dimensioning. For example, if Df is dimensioned as a $n \times 1$ variable, then the statement $Df = f$ automatically extracts the gradient part of f, i.e., ∇f.

The automatic computation of the partial derivatives is achieved by operator-overloading methodologies that redefine the intrinsic mathematical operators and functions using the rules of calculus. For example, addition and multiplication are redefined as follows.

$$a(x) + b(x) = \begin{bmatrix} a + b & \nabla a + \nabla b & \nabla^2 a + \nabla^2 b \end{bmatrix} \tag{4.2}$$

$$a(x) * b(x) = \begin{bmatrix} a * b & \nabla(a * b) & \nabla^2(a * b) \end{bmatrix} \tag{4.3}$$

The addition and multiplication operators are overloaded so that coding the left-side expressions of the above equations causes all of the right-side computations to be carried out. Moreover, if $z_1 = a+b$ and $z_2 = a*b$, then computing $z_3 = z_1 + z_2$ causes the previous results to be propagated efficiently in the background for all subsequent computations.

$$z_3 = [a + b + a * b \quad \nabla z_3 \quad \nabla^2 z_3] \tag{4.4}$$

Additional operations for the standard mathematical library functions, such as exponential and trigonometric functions, are redefined to account for the known rules of differentiation.

The true power of OCEA, indeed the "exact partial differentiation together with numerical evaluation", is best seen when one deals with composite functions. For example, consider a function $g(x)$.

$$g(x) = [g(x) \quad \nabla g(x) \quad \nabla^2 g(x)] \tag{4.5}$$

Then when one defines a function $f(g)$, OCEA exactly performs the partial differentiations.

$$f(g) = [f(g) \quad \tfrac{\partial f}{\partial g}\nabla g(x) \quad \tfrac{\partial^2 f}{\partial g^2}\nabla g(x) \nabla g(x)^T + \tfrac{\partial f}{\partial g}\nabla^2 g(x)] \tag{4.6}$$

The *numerical evaluations* of $g(x)$, $\nabla g(x)$, and $\nabla^2 g(x)$ (from equation (4.5)) all participate in computing the *exact* higher-order partial derivatives of $f(g)$. In essence, this approach pre-codes, once and for all, the partial derivatives required for any problem and the chain rule is implemented automatically in background operations that the user neither derives nor codes. At compile time, and without user intervention, the OCEA-based approach links the subroutines and functions required for evaluating the partial derivative models.

With regard to computing the Hamel coefficients (and focusing on scleronomic systems), the generalized coordinates are declared as embedded variables, and the mass matrix is formed. For example, the mass matrix of a rigid two-link manipulator with parameters denoted as `M1, M2`, etc., could be coded as follows.

$$\mathtt{MASS(1,1) = (M1 + M2) * L1 * *2}$$

$$\mathtt{MASS(1,2) = M2 * L1 * L2 * COS(q(2) - q(1))}$$

$$\mathtt{MASS(2,1) = M2 * L1 * L2 * COS(q(2) - q(1))}$$

$$\mathtt{MASS(2,2) = M2 * L2 * *2}$$

The variables `q(1)` and `q(2)` are the embedded variables. The Cholesky decomposition of the mass matrix can be computed in a straightforward way, for example, according to the following statements [7]

$$\mathtt{B(i,i)} = \mathtt{SQRT}\left(\mathtt{(M(i,i)}-\sum_{j=1}^{i-1}\mathtt{B(j,i)}^2\right);\ i=1,\ldots,n \tag{4.7}$$

$$\mathtt{B(i,k)} = \frac{1}{\mathtt{B(i,i)}}\left(\mathtt{M(i,k)}-\sum_{j=1}^{i-1}\mathtt{B(j,i)B(j,k)}\right); \tag{4.8}$$

$$k=i+1,\ldots,n;\quad i=1,\ldots,n$$

When computing the Cholesky decomposition, these equations must be used alternately.

Now, because the elements of the Cholesky decomposition depend on the elements of the mass matrix, which in turn depend on the embedded variables, the partial derivatives of B with respect to the embedded variables (e.g., ∇B) are automatically computed and evaluated as the elements of B are computed. Consequently, if DB is dimensioned as an $n\times n\times n$ array, then the statement $DB = B$ automatically extracts $\nabla B \equiv \partial B_{ij}/\partial q_k$, which are terms that help compose the Hamel coefficients given by equation (3.13).

The key point is that, via OCEA, once the Cholesky decomposition of the mass matrix is carried out, the partial differentiations that compose the Hamel coefficients are automatically determined.

CHAPTER V

HAMEL FORM EXAMPLES

A. Example 1

This first example is used to demonstrate the Hamel Form. Consider the planar motion of a point mass, m. Let r and θ be the generalized coordinates, where r is the radial distance from the origin and θ is the angle between the x-axis and a vector directed to the point mass from the origin: $q_1 = r$; $q_2 = \theta$. Suppose the point mass is acted upon by a radial force F. The second-order Lagrange equations of motion are as follows:

$$m\ddot{q}_1 - mq_1\dot{q}_2^2 = F \tag{5.1}$$

$$mq_1^2\ddot{q}_2 + 2mq_1\dot{q}_1\dot{q}_2 = 0 \tag{5.2}$$

The mass matrix for this choice of generalized coordinates is diagonal, and so is its Cholesky decomposition.

$$B = \text{diag}\left(\begin{array}{cc} \sqrt{m} & q_1\sqrt{m} \end{array} \right) \tag{5.3}$$

This leads to the following nonzero Hamel coefficients and the Hamel Form of dynamic equations.

$$\gamma_{21}^2 = -\gamma_{12}^2 = \frac{1}{q_1\sqrt{m}} \tag{5.4}$$

$$\dot{\omega}_1 - \frac{\omega_2^2}{q_1\sqrt{m}} = \frac{F}{\sqrt{m}} \tag{5.5}$$

$$\dot{\omega}_2 + \frac{\omega_1\omega_2}{q_1\sqrt{m}} = 0 \tag{5.6}$$

Note that for this simple problem, the eigenstructure quasi velocity formulation of Junkins and Schaub gives the same quasi velocities and hence the same dynamic equations.

$$M = C^T S^T S C; \qquad C = 1 \tag{5.7}$$

$$S = \text{diag}\left(\sqrt{m} \quad q_1\sqrt{m}\right); \qquad B = SC \tag{5.8}$$

B. Example 2

This second example is used to show that different generalized coordinates lead to different quasi velocities, and hence, different Hamel coefficients. Consider the planar motion of a rigid body of mass m and mass moment of inertia, J. The body has three degrees of freedom. Let q_1 be the radial distance from the origin of a fixed reference frame, O, to the mass center, C; let q_2 be the angle between the x-axis and a line OC; and let q_3 be the angle between OC and a reference axis in the body. For this choice of generalized coordinates, the mass matrix, its Cholesky decomposition, and the nonzero Hamel coefficients can all be computed.

$$M = \begin{bmatrix} m & 0 & 0 \\ 0 & J + mq_1^2 & J \\ 0 & J & J \end{bmatrix} \tag{5.9}$$

$$B = \begin{bmatrix} \sqrt{m} & 0 & 0 \\ 0 & \sqrt{J + mq_1^2} & \frac{J}{\sqrt{J+mq_1^2}} \\ 0 & 0 & \frac{q_1\sqrt{mJ}}{\sqrt{J+mq_1^2}} \end{bmatrix} \tag{5.10}$$

$$\gamma_{21}^{2} = -\gamma_{12}^{2} = \frac{q_1\sqrt{m}}{\sqrt{J + mq_1^2}} \tag{5.11}$$

$$\gamma_{13}^{2} = -\gamma_{31}^{2} = \frac{2\sqrt{J}}{\sqrt{J + mq_1^2}} \tag{5.12}$$

$$\gamma_{31}^{3} = -\gamma_{13}^{3} = \frac{J}{q_1\sqrt{m(J + mq_1^2)}} \tag{5.13}$$

If q_3 is replaced with the generalized coordinate $\phi = q_2 + q_3$, then different quasi velocities and nonzero Hamel coefficients are obtained.

$$B = \text{diag}\begin{pmatrix} \sqrt{m} & q_1\sqrt{m} & \sqrt{J} \end{pmatrix} \tag{5.14}$$

$$\gamma_{21}^{2} = -\gamma_{12}^{2} = \frac{1}{q_1\sqrt{m}} \tag{5.15}$$

C. Example 3

This final example is used to compare results from the Hamel Form with those obtained from a traditional second-order Lagrangian formulation. Consider a problem presented in the dissertation of Schaub [14], discussed on pp. 80-84. The mechanism is a two-link manipulator whose shoulder is inertially fixed but free to rotate. A linear spring is connected to the tip of the second link and to the inertial position $(x, y) = (0, 4)$. Each link has a tip mass. The generalized coordinates are the angles between the x-axis and the links. This is shown in Figure 1 [14]. The mechanism properties and initial conditions are shown in Table I. A constant step size, fourth-order, Runga-Kutta method within OCEA was used to numerically integrate the two-link system. The angular histories in Figure 2 are identical to the results of Schaub, which used the eigenstructure quasi velocity formulation of Junkins and Schaub. The angular histories were obtained via integration of the Hamel dynamic equations within the OCEA enviroment.

Figure 3 shows the time histories of the Hamel Form quasi velocities. The time

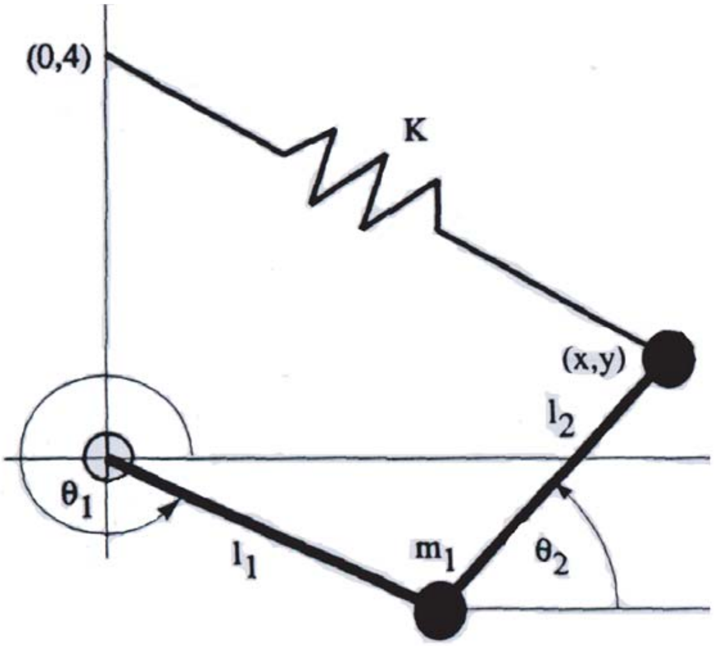

Fig. 1. Two-Link Manipulator Example.

histories are smooth and well behaved. A "snap-through" condition, characterized by large changes in the angular accelerations ($\ddot{\theta}_1$, $\ddot{\theta}_2$) can occur for this system when $\theta_1 \approx \theta_2$, and this can lead to numerical integration error. When the snap-throughs occur, the changes in the first time derivative of the quasi velocities are not nearly as much as the changes in the angular accelerations. Consequently, less error will appear in the results from integrating the Hamel variables (θ, ω) than the traditional Lagrangian variables (θ, $\dot{\theta}$).

This system is conservative and so the total system energy is constant. As a result, one can use the difference between the current energy and initial energy as a measure of integration error. This is shown in the energy plots of Figures 4 and 5. The Hamel Form shows consistently less error than the Lagrangian treatment ("Brute") throughout the integration, especially during the snap-through periods, when $\theta_1 \approx \theta_2$.

Table I. Mechanism Properties and Initial Conditions

Links:	$l_1 = 1/2;$	$l_2 = 1/\sqrt{2}$
Tip masses:	$m_1 = m_2 = 1.0$	
Spring constant:	$k = 1.0$	
Initial positions:	$\theta_1 = 0;$	$\theta_2 = 60\,\mathrm{deg}$
Initial velocities:	$\dot{\theta}_1 = \dot{\theta}_2 = 0$	

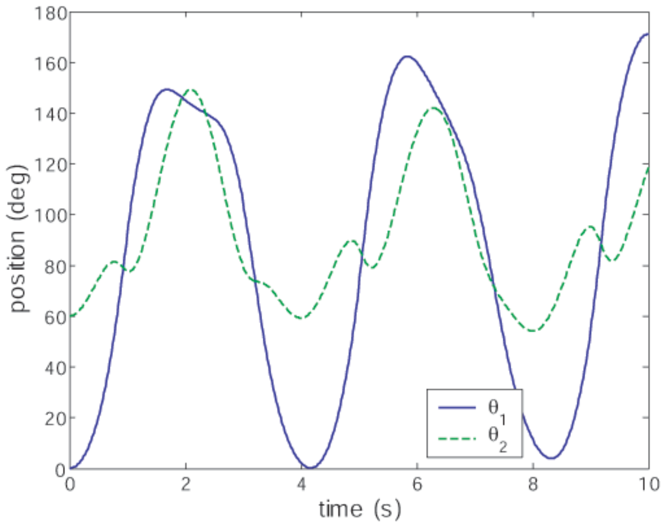

Fig. 2. Angular Time Histories.

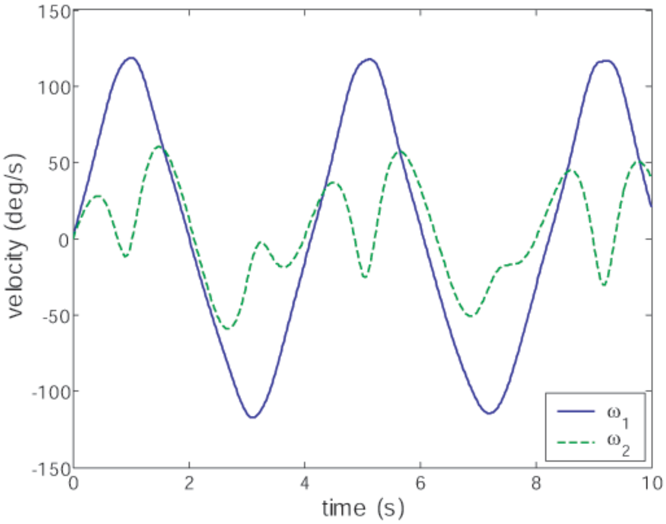

Fig. 3. Time Histories of the Quasi Velocities.

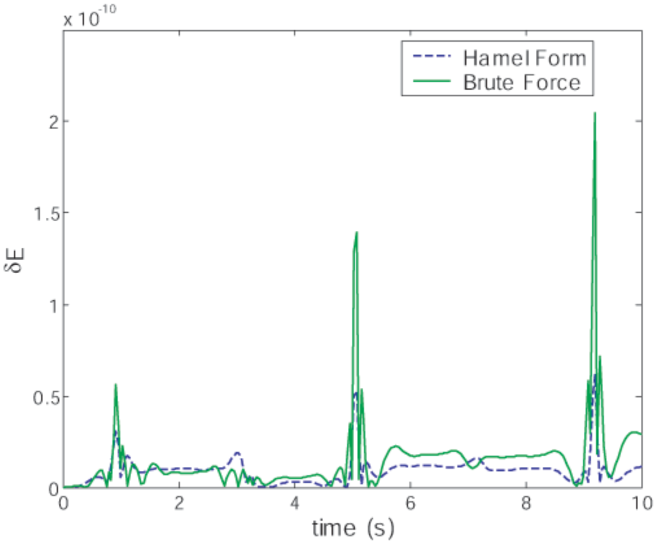

Fig. 4. Difference Between the Current Energy and Initial Energy for Ten Seconds.

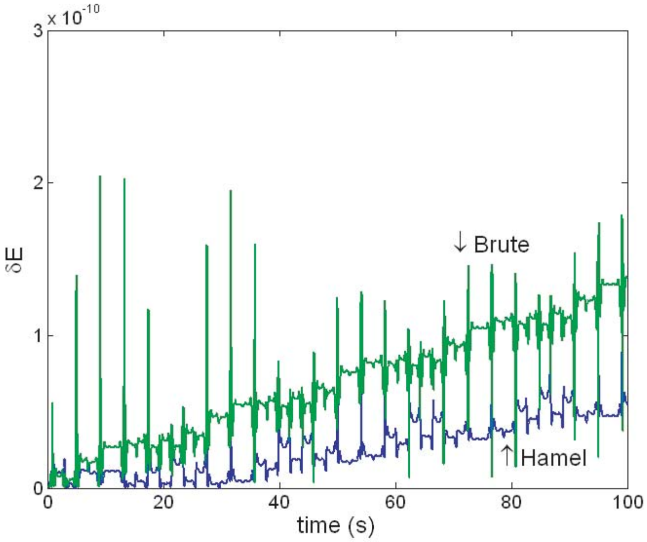

Fig. 5. Difference Between the Current Energy and Initial Energy When Integrated for 100 Seconds.

CHAPTER VI

COORDINATED RETARGETING AND IDENTIFICATION SYSTEM

Bases, camps, outposts, etc. are critical to effective military operations in hostile combat zones. Most of these locations, airfields as an example, tend to be at a fixed location throughout the duration of the conflict. The loss of these locations can have detrimental effects to almost all aspects of military campaigns. And these adverse effects typically take considerable manpower and resources to correct, if they are reversible at all. Therefore, these locations are high priority targets for an engaging force. Accordingly, they are also critical positions to be defended. Governmental buildings can also be threatened, especially with today's para-military and terrorist threats. The same priorities could be applied to these areas during tense or unrestful political situations.

The use of automated, unmanned systems provides many benefits to the protection of a military/governmental system. First and foremost is the reduction in risk to human life. A sentry force can be augmented or even eliminated with the addition of unmanned weapon systems. Another benefit is that these systems are always "on alert", regardless of the time of day or duration of deployment time. A third benefit is the reduction of base resources. These systems do not need to be fed, clothed or have quarters, although the trade off is they need to be powered and maintained. Intentional decoys of weapon systems placed in visible locations can fool electronic and even human reconnaissance into believing that system is genuine. This can be a deterrent to further aggression. Also, depending on the size of the systems, they could possibly be hidden from sight as well. The sudden appearance of these systems could surprise, confuse, and impede an attacking force. They can be used as the front-line defense and absorb the initial attack and give the defense ample time to coordinate.

A. Outline of Retargeting Scenerio

The goal of the proposed scenario is to minimize damage by targets to the area being protected while also attempting to keep communications between the fully automated weapon systems low. Secondary goals are to maximize the number of targets eliminated and achieve the finished scenario in minimum time. Multiple weapon systems are deployed and are stationary, but have the ability to pan and tilt. The opposition consists of multiple targets that clearly outnumber the weapon systems. Additionally, further targets will continue to appear throughout the exercise that were not present initially.

Each target will start randomly within the field of operation. They all follow trajectories towards the protected area. They travel at different speeds, and each target has a different "damage level" which it inflicts on the protected area. A target is considered to damage the area to be protected when it has passed through the defensive line of weapon systems. The actions and decisions of the weapon systems will be completed in real-time. The weapon systems have no prior knowledge of any target data.

B. Target Identification

One more important aspect of the overall system is target identification. A set of cameras with a small and more detailed field of view are required. These cameras need to return readable pictures from large distances and highly detailed pictures from short distances. This can have uses with the retargeting system. One use is for the retargeting system to identify the targets themselves from a preset set of images. Or a human user, who does not need to be onsite, can control the camera system as well. This will allow the user to examine the pertinent area and possibly make decisions based on the images.

Because of the detail need, these cameras will have large lenses attached to front of the cameras. It is impractical to physically move the camera and its large lens system in a pan and tilt motion. However, a mirror system can be used to alleviate this problem. Two mirrors will be used to redirected the beam where commanded. Each will rotate about one axis, representing the pan and tilt directions.

CHAPTER VII

CONTROL ALGORITHMS FOR RETARGETING SYSTEM

To accomplish the scenario established in the previous chapter, the weapon systems must operate autonomously. While developing the scenario, certain assumptions and guidelines were set up to demonstrate the simulation. They are listed below.

- Targets are randomly placed in the region. Their trajectories are also random to the extent that they all travel in the general direction towards the protected area. The targets velocities are random as well.
- The targets velocities are constant, and they travel in straight lines
- The targets damage capabilities are randomly assigned.
- Additional targets can appear in the zone of interest after the start time and follow the above items.
- The weapon systems are in fixed locations, randomly chosen – though staggered to somewhat evenly span the vicinity.
- The weapon systems know the current position and velocity of the targets.
- The weapon systems have unlimited ammunition. Once the weapon system and target are aligned, the target is considered destroyed.
- The weapon system's two degrees of freedom, pan and tilt, can move at different rates.

There are also specific properties of the scenario to be explained.

- The protected area starts with a “health” of 150.

- The targets damage rating is between 0 and 200 and time to impact ranges from five to forty seconds.
- The weapon systems have a maximum slew rate of ten degrees per second.

The next sections will outline the developed control algorithms.

A. Method 1: Closest Target

In this method, each weapon system determines which target it can reach first and chooses that target. This is typically the closest target with respect to the angle between the turret and the target. In some cases, however, it is not. A target can be moving away from the weapon system, thus increasing its slew time. Therefore a farther target moving towards the weapon system could be targeted sooner. The only information passed between the weapon systems is the declaration of target, to prevent multiple assignments of targets.

The computing needs of the weapon systems are very minimal. As a result, the communication requirements are very minimal. The weapon systems do not have a waiting period to either send or receive a communication. They also do not need a response from their outgoing messages to proceed.

However, this is far from an optimal approach. Situations can easily be conceived to show the shortcomings of this control method. The one advantage of this control technique is speed. Always choosing the closest target allows the weapons system to routinely eliminate more targets than the other algorithms. The major problem is that the weapon systems can not differentiate between high and low damage targets. They do not have any awareness if a target is critical to the protected area or not. Thus, there is a dependence on luck if critical targets are targeted.

Another problem is choosing targets at a distance rather than nearby. This algorithm only analyzes which target has the shortest slew time. Thus the weapons systems have no ability to distinguish depth of targets. Just like with the previous problem, the weapon systems are just as likely to aim for a target thirty seconds out compared to five seconds away.

When a new target appears within the time frame, each weapon system will determine if the new target has a shorter slew time. If the new target has a shorter slew time, the weapon system will disengage its current target and proceed to slew towards the new target.

Because of the minimal communication and computing power required, this algorithm is robust in a chaotic military environment. This algorithm would be the backup plan in the case that communication with the other weapon systems became erratic or non-existent.

B. Method 2: Auction Control

This control method has an overseer or an "auctioneer" to assign targets to weapon systems. The auctioneer does not have to be a separate entity; it can be one of the weapon systems. This auction system alleviates the problems of the previous control system of determining target priorities.

When a weapon system(s) signals that it is idle, the auction process begins [15]. The auctioneer determines the target with the highest priority, and then relays this information to the weapon systems. Each weapon system returns a bid to the auctioneer. The winning weapon system gets the target, and the process continues with the auctioneer determining the next high value target and bidding until all the weapon systems are busy.

How the auctioneer determines the highest priority targets is dependant on a set of metrics defined beforehand [16]. For this scenario, the damage rating of the target and it's time to impact are the key properties of the targets. Time to impact is more informational than strictly velocity or distance from the protected area alone.

In a simpler case, one can just auction off the quickest to impact or the most powerful targets first. However, the goal is to prevent as much damage as possible to the protected area. Just choosing the target with the highest damage could force the weapon systems to aim at targets far way and let closer ones pass through. Alternatively, always choosing the quickest to impact could send the weapon systems chasing targets with minimal damage properties while harder hitting targets slip past the weapon systems.

A mix of these two properties is sought after. Below are the metrics used to determine which target will be auctioned.

Step one Search for all targets within fifteen seconds of impact that do at least ten percent damage to the protected area. This step allows the auctioneer to find the closest targets but cautions against chasing weaker targets. If more than one target appears in this list, the auctioneer will auction off the target that does the most damage.

Step two If no targets are found in step one, find any target within twenty-five seconds that will do eighty percent or more damage to the protected area. This is a look-ahead feature. It is possible for weaker targets to be by-passed by step one. Again, this will caution against chasing weaker targets. It is more imperative to knock out critical damaging targets. If more than one target appears in this list, auction off the closest.

Step three Find any available target. If no targets appear in step one or two, this step will guide the weapon systems after all remaining targets. This is the first step the auctioneer could possibly look for targets past twenty-five seconds to impact. If more than one target is on this list, target the closest.

Step four Assign a target to multiple weapon systems. This step will be reached once the number of weapon systems is greater than the number targets. The auctioneer will simply tell the weapon system to find the target with the shortest slew time and engage.

Once the auctioneer determines the target with highest priority, the weapon systems must place a bid. For this scenario, this bid is simply the estimated time for the weapon system to slew to the target. This is the only property of this scenario that is pertinent to the abilities of the weapon systems. Whichever weapon system has the shortest slew time, wins the bid.

This control algorithm also has the ability for preemption. If a new target appears, the auctioneer could recalculate the priorities of the current targets. If the new target is more of a threat than current targets, the auctioneer will disengage all weapon systems targeting a lower prioritized target. The auctioneer would then bid off the new target, followed by the previous targets until all the weapon systems are active again.

Once again, the computing power need for the auctioneer is minimal. The other weapon systems computing power needs are even less. These weapon systems will need to communicate more often than the first method, and the communications are more detailed (declaring idle, getting target, sending bid, getting response). However, each weapon system is only communicating with one entity. There is no need for the weapon systems to be able to receive commands from each other.

From an operational point of view, having the entire decision making going through one entity is precarious. If for some reason the auctioneer is disabled, even with the weapon systems automatically switching to the closest target method, the results could be disastrous. Thus, to ensure stability, other computers (either separate machines or other weapon systems) need to be able to take control in case of loss. This increases the complexity of the programming required in the weapon systems. It is still true that each weapon system will still only be communicating with one auctioneer, but they also now need the ability to also communicate with the backups as well.

C. Method 3: Targets of Opportunity

An area of improvement with the auction method is the ability to hit additional targets of opportunity along the path to the main target. For example, when a weapon system wins a bid for a target, the weapon system will slew towards the intended target. It is quite possible along the way towards the intended target, the weapon system will pass another target. The weapon system could possibly neutralize that target also and still hit its initially assigned target. This could greatly reduce the time needed to eliminate targets and increase the survivability of the protected area.

Having the weapon systems find targets of opportunity will not greatly increase the workload of the weapon systems. The first step of this new algorithm is the same as the last method, a weapon systems will find their priority target. Now, during the weapon system's the initial movement, the weapon systems will attempt to find extra targets. However, one cannot allow the weapon systems to look at all the available targets. It is time consuming, and a majority of the targets are not reasonably nearby. Therefore, the weapon system will only look for targets that are in the same direction of the priority targets, which is a logical subset of targets.

Unfortunately, applying this in a straightforward manner actually reduces the ability to minimize damage to the base. An example is shown in the figure below.

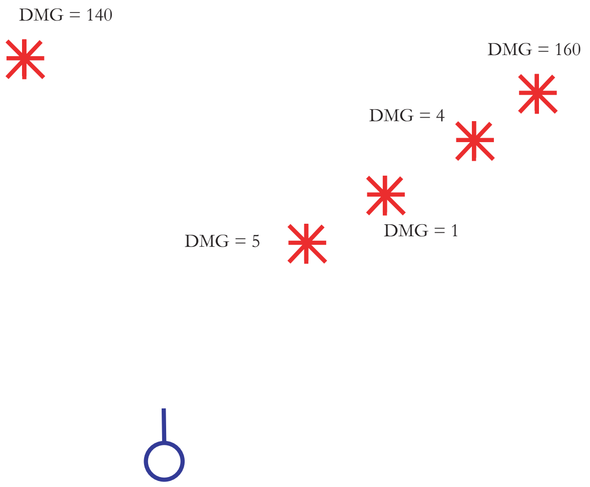

Fig. 6. Proposed Target Scenario.

Using the auction method, the weapon system would aim toward the target with the 160 damage rating, and then next slew to the target with the 140 damage rating. The other three targets would hit the protected area and result in a total damage to the area of 10. Without any restraints on finding targets of opportunity, the weapon system would hit all three targets on the way to the 160 target. However, the 140 target would reach the protected area before the weapon system could swing back. While an extra target was neutralized compared to the first case, 131 extra damage was applied to the protected area.

Even though this is a more extreme example, it demonstrates a valid point. More often than not, the targets of opportunity are not the mission critical targets. If these targets were more important, they would already have been auctioned off. So while this action maximized the number of targets hit, it did not minimize damage to the protected area. And minimizing damage to the protected area is vitally more important than the maximum number or targets eliminated.

Therefore a limit must be put on the number of targets a weapon system will queue before its priority target. Remembering that these targets of opportunity are not deemed the most critical targets, this limit will be set to only one additional target. There are a few criteria which need to be met for the weapon system to decide to target and additional target.

The weapon system will estimate its time to slew to the new target, and then re-estimate its slew time to the original target. Of course, there is a degree of uncertainty in propagating the future re-estimate of the original target. Therefore, if the new estimate of reaching the original target occurs when the original target is less than ten seconds from reaching the protected area, the target of opportunity will not be targeted.

If there is more than one target along the path to the auctioned target, the weapon system will choose the target of opportunity which predicts the quickest to reach the intended target. This choice will ensure that the weapon system will not waste extra time chasing an alternate target.

This new algorithm will increase the computing power of the weapon systems, but they are only minor calculations. The large increase will come in communications. Each weapon system will have to relay its new target of opportunity to the other weapon systems and to the auctioneer. This will eliminate two or more weapon systems aiming at the same target.

D. Method 4: Initial Decisions

The scenario is defined for a multitude of targets to appear suddenly all at once. Then as time progresses, additional targets appear, but not nearly with the magnitude of the first wave. One area to possibly increase the efficiency and optimality of the targeting system are the initial choices for targets. The method would determine the angular optimal choice for the first two targets for each weapon system. Determining the order that n number of weapon systems will target the $2n$ targets and their respective order is a challenging task to be completed in a real-time environment. The reason for an angular optimal solution over a more favorable time optimal solution is the real-time constraint. In the auction process, the weapon systems create their bid of slew time to the target by propagating the future. This single operation alone does not take any substantial amount of time. However, with three weapon systems and six targets are chosen to sort, that could require up to ninety different combinations of have slew times computed. This can not be done in a real-time environment, when seconds matter greatly. On the other hand, the instant angular distance between the weapon systems and the targets can be computed promptly.

This technique can only be used for two or three weapon systems. Four or more weapon systems can not be computed within reasonable time limits. With three weapon systems, six targets will be chosen for targeting. Worst case possibility, there are 600 different possible paths for the weapon systems to take that would need to be searched. This can still be accomplished in under half of a second. If there are four weapon systems, eight targets would be chosen. Worst case now leads to 1680 possible solutions. This leads to difficulty in computing the optimal solution in short time.

The first step is to identify the $2n$ targets to be considered. The auction process already defined earlier will determine these targets. Once these targets are defined, the next step is finding the cost of all the possible combinations for the first round of targets and sort them. For this problem, the cost of the operation will be the angular distance needed for the weapon system to slew to the target. For the two weapon system scenario, the possible paths are for weapon system one to aim at target one and weapon system two to aim at target two. Or targets 1 and 3, or 1 and 4, or 2 and 1 etc.

To efficiently sort the various costs, a binary heap sorting process will be used [17]. A traditional sorting routine where each value is ranked lowest to highest requires $O(n)$ computations to sort, add or delete items from list. A binary heap will have the lowest value at the top of the sort, but all the values below are not sorted from the next lowest to highest. This suits this application, since only the lowest cost at the present time is needed. Organizing a binary heap only takes $O(log_2 n)$ computations.

The next step is to remove the lowest cost path from the top of the heap, and expand it to find costs to the second set of targets. These new costs are now put back into the heap and organized accordingly. This process of removing the lowest cost and expanding the possible paths will continue until the lowest cost available contains both levels of targets. This solution is guaranteed to involve the least amount of angular movement of the weapon systems at that exact time.

Unfortunately, the targets are moving. Thus, finding the lowest angular distance does not guarantee this path will eliminate the targets in the shortest time. For example, target A could be farther away from the weapon system than target B. Therefore, this algorithm would choose the weapon system to pursue target A. However, target B is moving towards the weapon system, whereas target A is moving away. The weapon system would actually hit target B first, even though it was farther away initially.

Once the path is found, this information is relayed to the weapon systems, and the rest of the operation would be completed using the "targets of opportunity" method. The algorithm would essentially have the same amount of communications required as the "targets of opportunity" method.

E. Control of Weapon Systems

The control of the weapon systems uses a simple kinematic exponential control based on the error between the angle where the weapon system is currently pointing and the desired target. A rate limited is put onto the slew to more accurately model a dynamical system. Figure 7 displays a weapon system and its intented target. Let the error, ϵ be defined as the current angle minus the desired angle

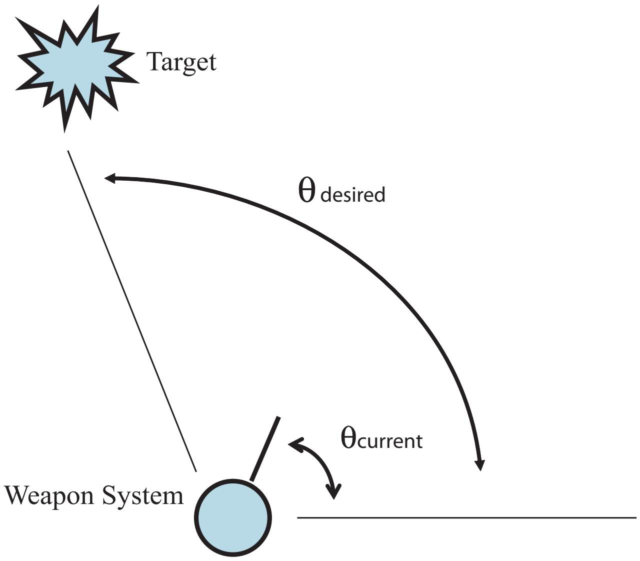

Fig. 7. Diagram of Weapon System and Target.

$$\epsilon = \theta_{cur} - \theta_{des} \tag{7.1}$$

Taking the time derivative and rearranging equation (7.1) gives

$$\dot{\theta}_{cur} = \dot{\epsilon} + \dot{\theta}_{des} \tag{7.2}$$

Now lets define the weapon system's slew rate ω as

$$\omega = \dot{\theta}_{des} - k\epsilon \tag{7.3}$$

ω is the same term as $\dot{\theta}_{cur}$, thus substituting equation (7.3) into equation (7.2)

$$\dot{\epsilon} = -k\epsilon \tag{7.4}$$

The solution to equation (7.4) is

$$\epsilon(t) = \epsilon_o e^{-kt} \tag{7.5}$$

Therefore the error will always be driven to zero. The update equation for control of the weapon systems is as follows

$$\dot{\theta}_{cur} = \omega = \dot{\theta}_{des} - k\left(\theta_{cur} - \theta_{des}\right) \tag{7.6}$$

CHAPTER VIII

CONTROL ALGORITHM TRADE STUDIES

A. Example 1

Table II. Retargeting Simulation 1

3 weapon systems and 6 targets				
	Gun 1	Gun 2	Gun 3	Time to Complete
Method 1	3	2	1	10.2 seconds
	5	6	4	
Method 2	3	5	6	10.6 seconds
	1	2	4	
Method 3	3	2*	6	9.2 seconds
	1	5	4	
Method 4	6	3	2	9.6 seconds
	5	1	4	

The first simulation shown involves three weapon systems and only six targets. This is a simple case and does not follow the design of the scenario involving a clear majority of targets. Even in this simple case however, all four methods chose a different order to eliminate the targets, as shown in Table II. Notice that method 4, the angular optimal choice, actually was not the fastest time to complete. If the targets were stationary, then method four would have the shortest time. However, because the targets are all moving, the final slew distances will be different than the original amount. Method 3 delivered the shortest completion time. The star on the target signifies that it was a target of opportunity, and chose that target before its priority target.

B. Example 2

Table III. Retargeting Simulation 2

3 weapon systems and 18 targets		
	Time to Complete	Final Base Health
Method 1	15.0 sec.	-252
Method 2	34.4 sec.	-14
Method 3	34.8 sec.	40
Method 4	32.4 sec.	-150

Table III details a simulation with three weapon systems and eighteen targets. This simulation is noteworthy as method 3 was the only method to successfully complete the mission. Method 4 failed because of its initial choices, ironically. Two of the weapon systems could not actually slew to their second targets before the targets reached the protected area. Once again, method 4 does not account for the targets moving towards or away from the weapon systems. Therefore, there is no guarantee that the weapon systems could actually reach the target in time, just a guarantee that at the initial time, the choice delivers the least angular distance required. Table IV shows the decisions made by the weapon systems in deciding which targets to target.

Method 3 was able to eliminate one extra target, and that the was difference between success and failure for the scenario. The starred numbers represent that target being a target of opportunity. Table V highlights the target information.

Table IV. Weapon Systems Target Choices

Method 2			**Method 3**		
Gun 1	Gun 2	Gun 3	Gun 1	Gun 2	Gun 3
17	6	5	17	6	5
12	8	16	11*	8	16
7	13	3	12	13	3
	10	9		7	9
	1	11		2	10
		2			14*
		4			1
					4

C. Example 3

This example ran through 250 cases of sixteen targets and three weapon systems. Their averages are shown in Table VI. Method 3 on average had the best results of the three methods tested. The average health column shows the average ending base health, with the protected area starting with 150. The next column shows the average time to complete the scenario and the last column shows how many times the scenario was completed successfully (final base health greater than zero).

D. Conclusion

The algorithm that achieved the best results was the auction control looking for targets of opportunity. Finding targets of opportunity greatly enhanced the survivability of the weapon system over the basic auction method. While on some occasions, the angular optimal algorithm achieved better results, as in example 3. Most instances though, it would not even achieve better results than the simple auction control. Even in the simple case of example 1, the angular optimal choice actually was not the fastest algorithm to complete the task. This lies in the fact that the algorithm does not take advantage of all the information available to the weapon systems.

The target of opportunity method could be expanded to look at more than one target before its priority target. Careful consideration should be heeded to ensure that the weapon systems do not spend time chasing less important targets.

Table V. Target Data for Example 2

Target Number	**Damage Level**	**Time to Hit**	**Target Number**	**Damage Level**	**Time to Hit**
1	109	38.4	10	48	29.1
2	95	32.5	11	167	36.7
3	121	29.2	12	165	32.6
4	66	34.9	13	43	25.4
5	102	12.3	14	54	31.9
6	171	12.1	15	55	32.2
7	112	31.1	16	194	19.2
8	48	12.4	17	81	10.1
9	44	26.4	18	55	33.8

Table VI. Averaged Results for 250 Simulations

3 weapon systems and 16 targets			
	Ave. Health	Ave. Time	Successful Scenarios
Method 1	-86.03	17.23	82
Method 2	27.18	28.69	147
Method 3	47.06	36.12	189
Method 4	-12.74	29.12	125

CHAPTER IX

TARGET IDENTIFICATION

Another area of interest within the retargeting scenario is target identification. Positioning cameras in the scene will allow either the computers or users to identify targets and make decisions about the target. These decisions could be as simple as friend or foe, or the images can be used to determine a specific point of attack at a target. Thus, these cameras need to be able to pan, tilt and zoom. A two mirror system was deemed necessary to control the pan and tilt of the camera. This reasoning is explained below.

A. Two Mirror System

These cameras are required to see objects at an extreme distance. Therefore, large lenses are necessary on the camera. Since it is not possible to mount a rotation device to the lens device, it must be mounted to the camera. This will create a considerable moment of inertia to control. Using a mirror system to redirect the image into the camera instead of physically moving the camera alleviates many problems. The first problem eased is pan and tilt speed. It will be much faster to rotate a mirror with considerably less inertia than the whole camera system. Settling time could also be improved, for much of the same reason as pan and tilt speeds. Settling time also can be reduced because the smaller motors which do not have to deliver as much torque tend to be more responsive also. Further, the camera needs to be calibrated to be accurate, and physically moving the camera system greatly increases the risk of altering the camera internal properties. A two mirror system, each rotated in only one direction was chosen over one mirror system able to rotate in two directions. This decision was chiefly based on the available range of motion of the systems. A two-axis device is not able to pan and tilt as large of an angular range as a two mirror system would. An example of a camera and mirror system is shown in Figure 8. The camera would be mounted inside of a building or a protective box to reduce the risk of damage to the camera system.

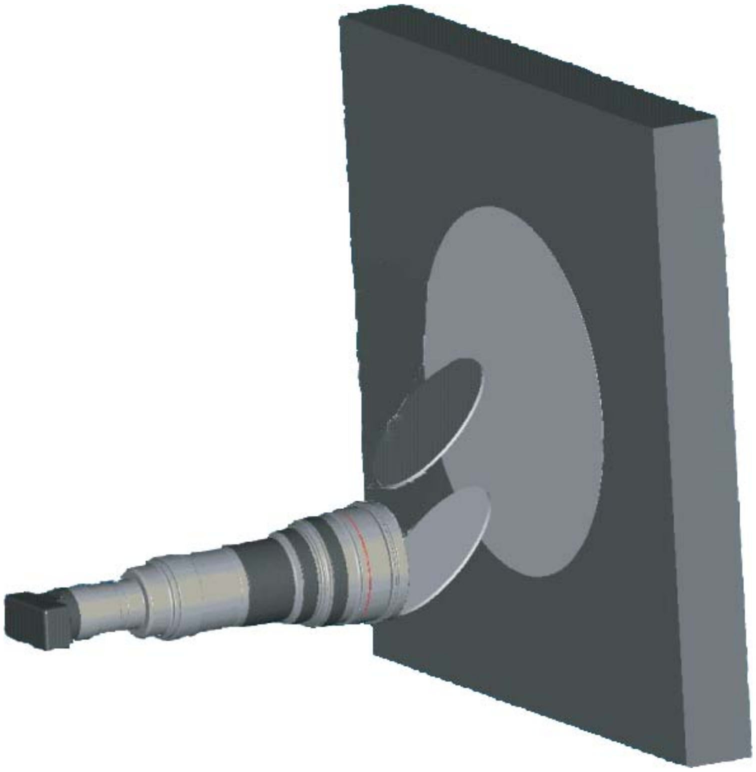

Fig. 8. Camera and Mirror Example.

B. Determination of Mirror Pan and Tilt Angles

The user will have some kind of interface, such as a joystick, to control the camera. The user would input a command as if he was physically controlling the camera. He might want the camera to pan to the right a certain distance. However, this does not map identically to just moving the mirror responsible for panning the same distance.

A relationship must be found to coordinate a move in the real world to the rotation of the mirrors. This section will explain how given a certain point in three dimensional space, what the ensuing mirrors angles need to be to capture that image point.

To begin with, a few coordinate systems need to be defined. These are mutually orthogonal systems and are depicted in Figure 9.

$\{\boldsymbol{c}_i\}$ (fixed) camera system/world system

$\{\boldsymbol{d}_i\}$ mirror 1 system

$\{\boldsymbol{e}_i\}$ mirror 2 system

The camera system is aligned such that $\boldsymbol{c}_2$ points straight out of the window. Each mirror system is aligned such that $\boldsymbol{d}_3$ and $\boldsymbol{e}_3$ is normal to the mirror plane and $\boldsymbol{d}_1$, $\boldsymbol{d}_2$, $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ lie in the mirror plane.

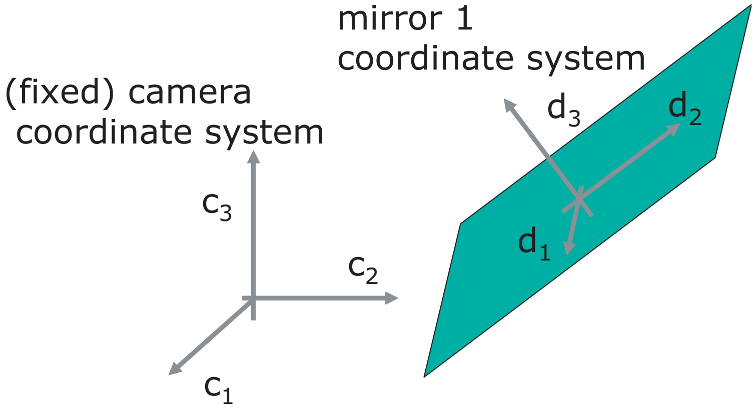

Fig. 9. Coordinate System Depiction.

Now consider the vector $\boldsymbol{u}$. This vector is coordinatized along the camera system axes as $\boldsymbol{u} = 0\boldsymbol{c}_1 + 1\boldsymbol{c}_2 + 0\boldsymbol{c}_3$. The column matrix representation of this vector is $[\boldsymbol{u}]_c = [0\ 1\ 0]^{\mathrm{T}}$. This vector represents the camera pointing straight out from its platform. Next, this vector needs to be represented in the mirror 1 coordinate system. Let $\boldsymbol{C}$ be the orthonormal rotation tensor that performs the transformation of a vector coordinatized along the camera system axes to a coordinatization along the mirror 1 axes.

$$[\boldsymbol{u}]_d = [\boldsymbol{C}][\boldsymbol{u}]_c \qquad ; \qquad [\boldsymbol{u}]_c = [\boldsymbol{C}]^{\mathrm{T}}[\boldsymbol{u}]_d \tag{9.1}$$

The vector $\boldsymbol{u}$ coordinatized along the mirror 1 axes is now $\boldsymbol{u} = C_{12}\boldsymbol{d}_1 + C_{22}\boldsymbol{d}_2 + C_{32}\boldsymbol{d}_3$, which in column matrix form is simply $[\boldsymbol{u}]_d = [C_{12} \quad C_{22} \quad C_{32}]^{\mathrm{T}}$.

This vector $\boldsymbol{u}$ that is now coordinatized along the mirror 1 axes is the incident vector on mirror 1. The reflected vector is also important to know. Figure 10 is an example of the incident and reflected vector relationship. The reflected vector from mirror 1 is given by the following equation, where $\boldsymbol{r}$ is the reflected vector and $\boldsymbol{n}$ is the vector normal to the (mirror) plane.

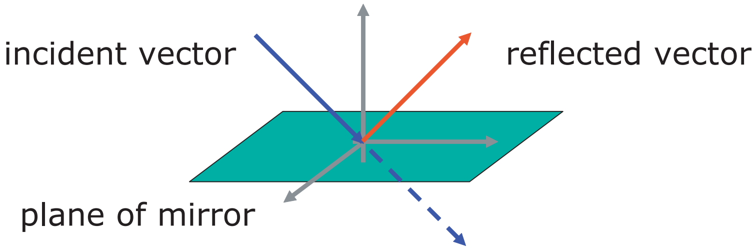

Fig. 10. Incident and Reflected Vector.

$$\boldsymbol{r} = \boldsymbol{u} - (2\boldsymbol{u} \cdot \boldsymbol{n})\,\boldsymbol{n} \tag{9.2}$$

This coordinatization of the reflected vector may be written in terms of the coordinatization of the incident vector using a reflection matrix $\boldsymbol{R}$.

$$[\boldsymbol{R}] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{9.3}$$

The reflected vector may be coordinatized along the camera system axes as such.

$$\begin{aligned}[\boldsymbol{r}]_c &= [\boldsymbol{C}]^{\mathrm{T}}[\boldsymbol{r}]_d \\ &= [\boldsymbol{C}]^{\mathrm{T}}[\boldsymbol{R}][\boldsymbol{u}]_d \\ &= [\boldsymbol{C}]^{\mathrm{T}}[\boldsymbol{R}][\boldsymbol{C}][\boldsymbol{u}]_c \end{aligned} \tag{9.4}$$

Let $\boldsymbol{T}$ be the orthonormal rotation tensor that performs the transformation of a vector coordinatized along the camera system axes to a coordinatization along the mirror 2 axes.

$$[\boldsymbol{r}]_e = [\boldsymbol{T}][\boldsymbol{r}]_c \qquad ; \qquad [\boldsymbol{r}]_c = [\boldsymbol{T}]^{\mathrm{T}}[\boldsymbol{r}]_e \tag{9.5}$$

Here the vector $[\boldsymbol{r}]_e$ is the reflected vector from mirror 1 coordinatized along the mirror 2 axes. Likewise, this vector also represents the incident vector onto mirror 2. They are the same vector. Similar to before, the reflected vector from mirror 2 can be related to the incident vector on mirror to as $[\boldsymbol{w}]_e = [\boldsymbol{R}][\boldsymbol{r}]_e$. Then the vector $\boldsymbol{w}$ can then be transformed from the mirror 2 reference frame to the camera system reference frame from equation (9.5). This vector, $[\boldsymbol{w}]_c$, represents the location that the user of the camera system would want the camera to be pointing towards. The final matrix result is obtained by combining all the previous transformations.

$$[\boldsymbol{w}]_c = [\boldsymbol{T}]^{\mathrm{T}}[\boldsymbol{R}][\boldsymbol{T}][\boldsymbol{C}]^{\mathrm{T}}[\boldsymbol{R}][\boldsymbol{C}][\boldsymbol{u}]_c \tag{9.6}$$

Equation (9.6) details how a vector $\boldsymbol{u}$ is redirected to vector $\boldsymbol{w}$ as a consequence of the orientations of mirrors 1 and 2.

1. Parameterization of the Transformations

Now that the transformation matrices have been defined, they must be parameterized. First, an overview of how the mirrors are to be rotated. Mirror 1, the mirror closest to the camera, will be responsible for panning. If looking directly out of the camera, this mirror would rotate left and right. Mirror 2 would thus rotate up and down. This is depicted in Figure 11 where the camera is on the left side of the picture.

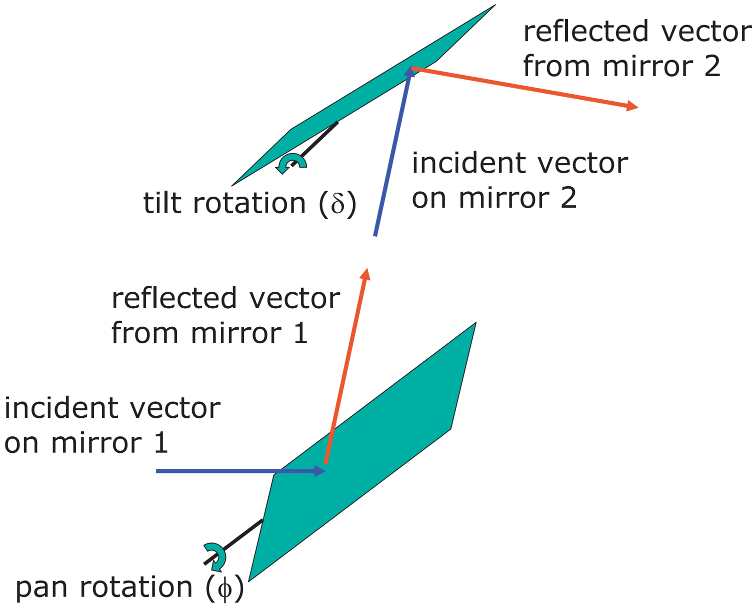

Fig. 11. Mirror Pan and Tilt Depiction.

Let $\boldsymbol{C}$, the orthonormal rotation tensor that performs the transformation of a vector coordinatized along the camera system axes to a coordinatization along the mirror 1 axes, be parameterized using Euler angles. The mirror 1 axes are tilted at a constant angle of forty-five degrees, which can be viewed as a **1**-axes Euler rotation. The variable pan angle, ϕ, can be viewed as a **2**-axes rotation.

$$[\boldsymbol{C}] = \begin{bmatrix} \cos\phi & 0 & -\sin\phi \\ 0 & 1 & 0 \\ \sin\phi & 0 & -\cos\phi \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\pi/4) & \sin(\pi/4) \\ 0 & -\sin(\pi/4) & \cos(\pi/4) \end{bmatrix} \tag{9.7}$$

Let $\boldsymbol{T}$, the orthonormal rotation tensor that performs the transformation of a vector coordinatized along the camera system axes to a coordinatization along the mirror 2 axes, be also parameterized using Euler angles. The mirror 2 axes are tilted at a variable tilt angle of θ, which can be viewed as a **1**-axes Euler rotation.

$$[\boldsymbol{T}] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) \\ 0 & -\sin(\theta) & \cos(\theta) \end{bmatrix} \tag{9.8}$$

Using these parameters together with $[\boldsymbol{u}]_c = [0\ 1\ 0]^{\mathrm{T}}$, equation (9.6) can be used to compute the reflected vector from mirror 2 coordinatized along the camera system axes.

$$[\boldsymbol{w}]_c = \begin{bmatrix} \left(\sqrt{2}/2\right)\sin 2\phi \\ \sin^2\phi\cos 2\theta + \cos^2\phi\sin 2\theta \\ \sin^2\phi\sin 2\theta - \cos^2\phi\cos 2\theta \end{bmatrix} \tag{9.9}$$

The nominal tilt angle of mirror 2 is forty-five degrees. Therefore, we can write $\theta = 45\ \text{deg} + \delta$. Now, equation (9.9) can be written in term of δ instead of θ.

$$[\boldsymbol{w}]_c = \begin{bmatrix} \left(\sqrt{2}/2\right)\sin 2\phi \\ -\sin^2\phi\sin 2\delta + \cos^2\phi\cos 2\delta \\ \sin^2\phi\cos 2\theta + \cos^2\phi\sin 2\delta \end{bmatrix} \tag{9.10}$$

Note that if $\phi = 0$ (no pan) and $\delta = 0$ (no additional tilt beyond the nominal forty-five degrees) then $[\boldsymbol{w}]_c = [0\ 1\ 0\ ]^{\mathrm{T}} = [\boldsymbol{u}]_c$.

Now suppose a desired point in the camera system axes is defined by the coordinates (ω_{1d}, ω_{3d}). The corresponding pan and tilt angles, ϕ and δ, can be computed using the elements of $\boldsymbol{\omega}$ coordinatized in the camera system axes as given by equation

(9.10).

$$\omega_{1d} = \left(\sqrt{2}/2\right) \sin 2\phi \tag{9.11}$$

$$\omega_{3d} = \sin^2 \phi \cos 2\theta + \cos^2 \phi \sin 2\delta \tag{9.12}$$

Equation (9.11) may be solved for the pan angle ϕ.

$$\phi = \frac{1}{2} \arcsin \left(\sqrt{2}\omega_{1d}\right) \tag{9.13}$$

Solving for the tilt angle, δ, is not so straightforward. Equation (9.12) is a transcendental equation. For small pan angles ϕ, the solution of equation (9.12) is nearly given by the second term. This can provide a good starting guess for δ

$$\delta \approx \frac{1}{2} \arcsin(\omega_{3d}/ \cos^2 \phi) \tag{9.14}$$

Using a standard non-linear least squares solving method [18], it usually only takes about two iterations to arrive at a solution.

CHAPTER X

SUMMARY

A. Hamel Form

A special form of the Poincaré equations, which is called the Hamel Form, was developed in this thesis. Through this special choice of quasi velocities, a diagonal form of the equations of motion can be achieved. The coefficients generated by the Hamel Form are based on the partial differentiation of the mass matrix, and thus closed form representations are difficult to obtain. However, a newly developed program called OCEA is used to automatically, and without any user intervention, generate these coefficients so that numerical integration can follow. Although the Hamel Form is a diagonal form similar in concept to those of Jain and Rodriguez and Junkins and Schaub, the quasi velocities of all three methods are generally different. One advantage, as displayed in example 3 from Chapter V, is that integration error with the Hamel form was less than the traditional Lagrange treatment. This arises because the quasi velocities generated are much smoother through the "snap-through" of the system, whereas the traditional accelerations have a more dramatic change in their values.

B. Coordinated Retargeting and Target Identification

An automated retargeting system provides many benefits to the user of this system. Most important of these is the almost complete reduction of human risk. This scenario involved greatly outnumbered weapon systems suddenly inundated with targets. An auction control system was developed to be operated in real-time to minimize damage to the protected area. Refining the auction control system by identifying targets of opportunity increased the survivability of the system. Another method attempted to optimally find the first two levels of targets for each weapon system, and then revert to the auction process afterwards. This process looked only at angular distance, and not estimated time to slew to target, because of the real-time constraints. Thus, in some cases this algorithm will achieve better results, but it can run into problems when targets are moving away from the weapon system's slew direction.

Target identification is another aspect within the retargeting system. Due to the range and clarity needed from these cameras, large lens are attached to the cameras. This creates problems controlling the pan and tilt ability of the cameras. This difficulty is alleviated by the use of a two mirror system to pan and tilt the image into the camera instead of physically moving the camera. An algorithm was developed which given a target point in the real world for the camera to look at, the mirrors would rotate accordingly.

REFERENCES

[1] L. Meirovitch, *Methods of Analytical Dynamics.* New York: McGraw-Hill, 1970.

[2] J. Papastavridis, *Analytical Mechanics.* New York: Oxford University Press, 2002.

[3] T. Kane and D. Levinson, *Dynamics: Theory and Applications.* New York: McGraw-Hill, 1985.

[4] P. Mitiguy and T. Kane, "Motion variables leading to efficient equations of motion," *The International Journal of Robotics Research*, vol. 15, no. 5, pp. 522–532, 1996.

[5] A. Jain and G. Rodriguez, "Diagonalized lagrangian robot dynamics," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 14, pp. 571–584, 1995.

[6] J. Junkins and H. Schaub, "An instantaneous eigenstructure quasivelocity formulation for nonlinear multibody dynamics," *The Journal of the Astronautical Sciences*, vol. 45, no. 3, pp. 279–295, 1997.

[7] L. Meirovitch, *Computational Methods in Structural Dynamics.* Rockville, MD: Sijthoff & Noordhoff, 1980.

[8] R. Talman, *Geometric Mechanics.* New York: John Wiley & Sons, 2000.

[9] J. Papastavridis, *Tensor Calculus and Analytical Dynamics.* Boca Raton, FL: CRC Press LLC, 1999.

[10] H. Oh, S. Vadali, and J. Junkins, "On the use of the work-energy rate principle for designing feedback control laws," *Journal of Guidance, Control, and Dynamics*, vol. 15, no. 1, pp. 275–277, 1992.

[11] G. Hamel, *Theoretische Mechanik.* Berlin: Springer-Verlag, 1967.

[12] G. Stewart, *Introduction to Matrix Computations.* Orlando, FL: Academic Press, 1973.

[13] J. Turner, “Automated generation of high-order partial derivative models,” *AIAA Journal*, vol. 41, no. 8, pp. 1590–1598, 2003.

[14] H. Schaub, “Novel coordinates for nonlinear multibody motion with applications to spacecraft dynamics and control,” Ph.D. dissertation, Texas A&M University, College Station, 1998.

[15] W. Vickrey, “Counterspeculation, auctions, and competitive sealed tenders,” *Journal of Finance*, vol. 16, pp. 8–37, March 1961.

[16] B. Gerkey and M. Matarić, “Sold! market methods for multi-robot control,” *IEEE Transactions on Robotics and Automation Special Issue on Multi-Robot Systems*, pp. 1–12, March 2001.

[17] S. Skiena, *The Algorithm Design Manuel.* New York: Springer-Verlag, 1997.

[18] J. Crassidis and J. Junkins, *Optimal Estimation of Dynamic Systems.* Boca Raton, FL: Chapman & Hall/CRC, 2004.

VITA

Michael Charles Sovinsky received his Bachelor of Science degree in aerospace engineering from Texas A&M University in May of 2003. In September of 2003, he entered the graduate program at Texas A&M University and graduated with a Master of Science degree in aerospace engineering in December of 2005. While in graduate school, his research areas focused on dynamics and control of dynamical systems.

Michael can be reached at Texas A&M University, Aerospace Engineering Dept., TAMU 3141, College Station, TX, 77843, or through email at msovinsky@tamu.edu.