Moving Mass Actuated Missile Control Using Convex Optimization Techniques

Sesha Sai Vaddi*

Optimal Synthesis, 868 San-Antonio Road, Palo Alto, CA, 94303, USA

This paper deals with the control of a moving mass actuated missile in the presence of state and control constraints. Quadratic programming which is a convex optimization formulation is employed for control computation. Planar equations of motion of a moving mass actuated missile are obtained in closed form using Lagrangian approach. These equations are analytically linearized and used for control purposes. Control is computed at each instant of time using the time-to-go information. Results obtained from closed loop nonlinear simulations demonstrate the effectiveness of the controller.

Nomenclature

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Moment of inertia of the missile</td>
</tr>
<tr>
<td>$M_b$</td>
<td>Mass of the missile</td>
</tr>
<tr>
<td>$m$</td>
<td>Moving mass</td>
</tr>
<tr>
<td>$\delta$</td>
<td>Position of the moving mass along its axis</td>
</tr>
<tr>
<td>$T$</td>
<td>Kinetic energy</td>
</tr>
<tr>
<td>$F_T$</td>
<td>Axial rocket motor thrust</td>
</tr>
<tr>
<td>$g$</td>
<td>Acceleration due to gravity</td>
</tr>
<tr>
<td>$X$</td>
<td>North position of the missile in an inertial NED frame</td>
</tr>
<tr>
<td>$Z$</td>
<td>Altitude of the missile</td>
</tr>
<tr>
<td>$\theta$</td>
<td>Attitude angle of the missile</td>
</tr>
<tr>
<td>$M_t$</td>
<td>Total mass</td>
</tr>
<tr>
<td>$M$</td>
<td>Mass matrix</td>
</tr>
<tr>
<td>$u$</td>
<td>Actuator force</td>
</tr>
<tr>
<td>$X$</td>
<td>State vector of the dynamic system</td>
</tr>
<tr>
<td>$Z$</td>
<td>State vector of the optimization problem</td>
</tr>
</tbody>
</table>

Subscript

<table>
<thead>
<tr>
<th>Subscript</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>Body</td>
</tr>
<tr>
<td>t</td>
<td>Total</td>
</tr>
<tr>
<td>c</td>
<td>Continuous</td>
</tr>
<tr>
<td>d</td>
<td>discrete</td>
</tr>
</tbody>
</table>

I. Introduction

Moving mass actuation system is a novel concept introduced recently for the control of missiles. These actuators are completely enclosed within the geometric envelope of the vehicle and are equally effective inside and outside the atmosphere. The objective of this research is to design a control system for a moving mass actuated missile.

The dynamic model of the system is nonlinear and has both actuator and state constraints. Control of nonlinear dynamical systems is more difficult than the control of their linear counterparts. The presence of

*Research Scientist, Optimal Synthesis, 868 San-Antonio Road, Palo Alto, CA, 94303.
state and actuator constraints adds up to the complexity of the problem. No general framework exists for control design of nonlinear systems with state and actuator constraints.

Current approach is based on obtaining a convex relaxation. Current controller design approach is based on quadratic programming. A linear discretised version of the nonlinear system is used for this purpose. The approach is particularly attractive for its ability to explicitly accommodate the before mentioned constraints.

A. Background

Moving mass actuation system was recently discussed by Menon et al. An integrated guidance-control system for target interception was designed in that paper. Numerical feedback linearization technique which is a feature of commercially distributed software Nonlinear Synthesis Toolbox is used for the controller design. State and actuator constraints arising from limited mass displacement and peak actuator force are not included in the design process. The controller is implemented on a simulation where the states and controls are saturated when these constraints are violated. Similar setting has been used for testing a controller in a recent publication that is designed using a combination of feedback linearization and finite-interval optimal control. Even though the controller works well in a wide range of scenarios it cannot prevent saturation. The current research is primarily motivated by the need to avoid saturation and thereby prevent any undesirable consequences associated with it. A controller formulation that can accommodate the state and actuator constraints is essential for achieving this.

Convex optimization formulations and solution methodologies have been gaining popularity in the recent past. A fundamental property of convex optimization problems is that any locally minima is also a global minima. Moreover, unlike general nonlinear optimization problems numerical techniques for solving these problems are well developed and guarantee convergence. Some of the convex optimization formulations are quadratic programming, second-order cone programming, geometric programming, semi-definite programming etc. Tillerson et al. have used a convex optimization formulation for the control of spacecraft formation. Controller design by posing them as convex optimization problems is similar to predictive control approach discussed in Ref.5.

Equations of motion governing the full three dimensional translation, rotation and actuator dynamics are derived in Ref.1. The current paper derives a planar version of these equations, however, in closed form and also analytically obtains a linearized version of the same. Moreover, equations of motion are obtained directly in terms of inertial co-ordinates of the missile in contrast to the body-frame velocity components. This facilitates designing an integrated guidance-controller with this model. The linear model is then used in the controller design for which quadratic programming methodology is adopted. This paper is organized as follows: Section II deals with the dynamic model of the moving mass actuated missile, Section III deals with the quadratic programming formulation of the controller, closed loop simulation results are presented in Section IV.

II. Model

The following assumptions are made in modeling the dynamics of the moving mass actuated missile.

- Motion of the missile is restricted to the vertical plane.
- Gravity vector is assumed to act along the inertial Z direction.
- The missile is flying outside the atmosphere therefore no aerodynamic forces act on the vehicle.

Shown in Fig.1 is a diagram of the moving mass actuated missile. External rocket motor thrust acts along the axis of the missile. The position of the mass is manipulated by the controller to control the attitude of the missile. By changing the attitude of the missile the orientation of the rocket-motor thrust is changed in an inertial frame, thereby changing the path of the missile.

Body-frame to inertial frame transformation matrix is given by:-

\[
C_b = \begin{bmatrix}
c_\theta & 0 & -s_\theta \\
0 & 0 & 0 \\
s_\theta & 0 & c_\theta
\end{bmatrix}
\] (1)
Velocity of the body in the body-frame is given by:

\[
\dot{\mathbf{R}}_b = C_b \begin{bmatrix}
X \\
0 \\
\dot{Z}
\end{bmatrix} = \begin{bmatrix}
c_\theta \dot{X} - s_\theta \dot{Z} \\
0 \\
s_\theta \dot{X} + c_\theta \dot{Z}
\end{bmatrix}
\] (2)

Inertial velocity of the mass in the body-frame is given by:

\[
\dot{\mathbf{R}}_m = \begin{bmatrix}
c_\theta \dot{X} - s_\theta \dot{Z} \\
0 \\
s_\theta \dot{X} + c_\theta \dot{Z}
\end{bmatrix} + \begin{bmatrix}
\delta \theta \\
\delta + l \dot{\theta}
\end{bmatrix} = \begin{bmatrix}
c_\theta \dot{X} - s_\theta \dot{Z} + \delta \theta \\
0 \\
s_\theta \dot{X} + c_\theta \dot{Z} + \delta + l \dot{\theta}
\end{bmatrix}
\] (3)

The total kinetic energy of the system can be written as:

\[
T = \frac{1}{2} M_b \dot{X}^2 + \frac{1}{2} M_b \dot{Z}^2 + \frac{1}{2} I \dot{\theta}^2 + \frac{1}{2} m(c_\theta \dot{X} - s_\theta \dot{Z} + \delta \theta)^2 + \frac{1}{2} m(s_\theta \dot{X} + c_\theta \dot{Z} + \delta + l \dot{\theta})^2
\] (4)

\[
T = \frac{1}{2} M_b (\dot{X}^2 + \dot{Z}^2) + \frac{1}{2} (I + m\delta^2 + ml^2) \dot{\theta}^2 + \frac{1}{2} m\dot{\delta}^2 + mc_\theta \dot{X} \delta \theta - ms_\theta \dot{Z} \delta \theta + ms_\theta \dot{X} l \dot{\theta} + mc_\theta \dot{Z} l \dot{\theta} + mc_\theta \dot{Z} l \dot{\theta} + m \delta l \dot{\theta}
\] (5)

Equations of motion corresponding to the four degrees of freedom \( \mathbf{X} = [X \ Z \ \theta \ \delta] \) can be written in Lagrangian\(^6\) form as shown below:

\[
\frac{d}{dt} \frac{\partial T}{\partial \dot{X}} - \frac{\partial T}{\partial X} = Q_X
\] (6)

\[
\frac{d}{dt} \frac{\partial T}{\partial \dot{Z}} - \frac{\partial T}{\partial Z} = Q_Z
\] (7)

\[
\frac{d}{dt} \frac{\partial T}{\partial \dot{\theta}} - \frac{\partial T}{\partial \theta} = Q_\theta
\] (8)

\[
\frac{d}{dt} \frac{\partial T}{\partial \dot{\delta}} - \frac{\partial T}{\partial \delta} = Q_\delta
\] (9)
A detailed evaluation of the above equations is given in the appendix. The exact form of the system dynamics is finally obtained as:

\[ \mathbf{M}(\mathbf{X}) \ddot{\mathbf{X}} + \mathbf{C}(\mathbf{X}, \dot{\mathbf{X}}) = \mathbf{Q} \]  

(10)

where,

\[
\mathbf{M} = \begin{bmatrix}
M_t & 0 & mc_g \delta + ms_l & ms_l \\
0 & M_t & mc_g - ms_g \delta & mc_g \\
mc_g \delta + ms_l & -ms_g \delta + mc_g & (I + m\delta^2 + ml^2) & ml \\
ms_l & mc_g & ml & m
\end{bmatrix}
\]  

(11)

\[
\mathbf{C} = \begin{bmatrix}
-m_s g \delta \dot{\theta}^2 + 2mc_g \dot{\delta} \dot{\theta} + mc_g \ddot{\theta}^2 \\
-mc_g \delta \dot{\theta}^2 - 2ms_g \dot{\delta} \dot{\theta} - ms_g \dot{\theta} \dot{\theta}^2 \\
2m \ddot{\theta} \\
-m \delta \dot{\theta}^2
\end{bmatrix}
\]  

(12)

\[
\mathbf{Q} = \begin{bmatrix}
F_T c_g \\
-F_T s_g + M_t g \\
-m s_g \delta + mc_c l \\
u + mc_c g
\end{bmatrix}
\]  

(13)

\[
\dot{\mathbf{X}} = \mathbf{M}(\mathbf{X})^{-1} [\mathbf{Q} - \mathbf{C}(\mathbf{X}, \dot{\mathbf{X}})]
\]  

(14)

Inverting the mass matrix using Maple software and simplifying the equations further, the following are obtained:

\[
\ddot{X} = \frac{c_g}{M_t - m} F_T - \frac{c_g I_m}{(M_t I + (M_t - m)m \delta^2)} F_T - \frac{s_g}{M_t - m} u + \frac{ml c_g \delta}{(M_t I + (M_t - m)m \delta^2)} u + \frac{M_t I m c_g \delta^2 + 2M_t I m c_g \delta \dot{\theta} - 2I m^2 c_g \dot{\theta} - I m^2 c_g \dot{\theta}^2}{(M_t - m)(M_t I + (M_t - m)m \delta^2)}
\]  

(15)

\[
\ddot{Z} = g + c_g u - \frac{ml s_g \delta}{M_t I + (M_t - m)m \delta^2} u - \frac{s_g}{M_t - m} F_T - \frac{s_g m I}{(m - M_t)(M_t I + (M_t - m)m \delta^2)} F_T + \frac{m^2 I s_g \dot{\theta}^2 + 2m^2 I s_g \ddot{\theta} - m s_g \dot{\theta} M_t I - 2ms_g \dot{\theta} M_t I}{(M_t - m)(M_t I + (M_t - m)m \delta^2)}
\]  

(16)

\[
\ddot{\delta} = -\frac{m \delta}{M_t I + (M_t - m)m \delta^2} F_T - \frac{M_t I}{(M_t I + (M_t - m)m \delta^2)} u - 2m^2 \dot{\delta} \dot{\theta} - m^2 \dot{\theta} \ddot{\theta}^2 + 2 M_t m \delta \dot{\delta} \dot{\theta} + M_t l m \dot{\theta} \dot{\theta}^2
\]  

(17)

\[
\ddot{\theta} = \frac{ml \ddot{\delta}}{M_t I + (M_t - m)m \delta^2} F_T + \frac{M_t I}{m(M_t - m)} u + \frac{M_t^2 I^2}{(M_t I + (M_t - m)m \delta^2)} u + \frac{2ml \ddot{\delta} \dot{\theta}(M_t - m)}{(M_t I + (M_t - m)m \delta^2)} + \ddot{\theta}^2 + \frac{(M_t - m) ml^2}{(M_t I + (M_t - m)m \delta^2)} \ddot{\theta}^2
\]  

(18)

The above equations are the exact nonlinear equations governing the translational, rotational and actuator dynamics of the missile. These can be used to design guidance and control laws for the missile. Linearizing the above equations about \( \theta = 0, \dot{\theta} = 0, \delta = 0, \dot{\delta} = 0 \) the following are obtained:

\[
\ddot{X} = \frac{1}{M_t} F_T - \frac{\theta}{M_t - m} u + \frac{ml \delta}{M_t I} u
\]  

(19)
\[
\ddot{Z} = g - \frac{u}{M_t - m} - \frac{\theta}{M_t - m} F_T + \frac{\theta m}{M_t(M_t - m)} F_T = g - \frac{u}{M_t - m} - F_T \theta 
\]

\[
\ddot{\theta} = -\frac{m\delta}{M_t I} F_T - \frac{l}{I} u 
\]

\[
\ddot{\delta} = \frac{ml\delta}{M_t I} F_T + \frac{M_t m}{m(M_t - m)} u + \frac{M_t l^2}{I(M_t - m)} u
\]

### III. Controller Design Using Quadratic Programming

Controller design using quadratic programming will be discussed in this section for a general linear system of the form:

\[
\dot{X} = A_c X + B_c u
\]

Discretized version of the above equations obtained using the zero-order-hold are as follows:

\[
X[k+1] = A_d X[k] + B_d u[k]
\]

where \(X\) is a \(n\times1\) vector, \(A_d\) is a \(n\times n\) matrix and \(B_d\) is a \(n\times1\) vector.

The general form of a quadratic programming problem can be written as follows:

\[
\begin{align*}
\min & \quad Z^T S Z \\
\text{subject to} & \quad F Z = g \\
& \quad l l <= Z <= u l
\end{align*}
\]

Define \(Z = [X'[2] \ X'[3] \ldots \ X'[N] \ u[1] \ u[2] \ldots u[N-1]]\) a vector of length \((n+1)(N-1)\) as the optimization state vector. Both the state variables and control variables over a \(N\) steps are treated as optimization variables. The controller objective is posed as the objective function of the optimization problem. The constraint matrix is formed by invoking the discrete-time propagation equation over \(N\) time steps.

\[
\begin{bmatrix}
X[2] \\
X[3] \\
\vdots \\
X[N]
\end{bmatrix} = \begin{bmatrix}
A_d & 0 & \ldots & 0 \\
0 & A_d & \ldots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & \ldots & 0 & A_d
\end{bmatrix} \begin{bmatrix}
X[1] \\
X[2] \\
\vdots \\
X[N-1]
\end{bmatrix} + \begin{bmatrix}
B_d & 0 & \ldots & 0 \\
0 & B_d & \ldots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & \ldots & 0 & B_d
\end{bmatrix} \begin{bmatrix}
u[1] \\
u[2] \\
\vdots \\
u[N-1]
\end{bmatrix}
\]

\[
F Z = g
\]

where \(F\) is a \(n(N - 1) \times (n + 1)(N - 1)\) matrix given by the following equation:

\[
F = [I_{n(N-1)} \times n(N-1) \mid 0_{n(N-1) \times (N-1)}] -
\begin{bmatrix}
0 & 0 & \ldots & 0 & B_d & 0 & \ldots & 0 \\
A_d & 0 & \ldots & 0 & 0 & B_d & \ldots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & \ldots & 0 & A_d & 0 & 0 & \ldots & 0 & B_d
\end{bmatrix}
\]

and \(g\) is a \(n(N - 1) \times 1\) vector:
State and control constraints such as limits can be equivalently imposed as limits on the optimization state vector.

\[ Z_{min} <= Z <= Z_{max} \]  

\[ (30) \]

**IV. Results**

Even though the controller design was done using a discrete-time linear model it was tested on continuous time nonlinear closed loop simulation. Equations 10-14 are used for this purpose. The optimization process is conducted using the ‘quadprog’ function of MOSEK software. The following values have been used for the missile parameters, \( M = 1.7 \text{slugs} \), \( m = 0.1775 \text{slugs} \), \( I = 0.6 \text{slug} \cdot \text{ft}^2 \), \( F_T = 500 \text{lbf} \), \( \delta_{max} = 0.5 \text{ft} \), \( u_{max} = 75 \text{lbf} \). Normalized position and control are used for plotting in the following sub section. They are both normalized by their respective maximum values.

**A. Body Rate Stabilization**

![Figure 2. Angular Velocity Time History.](image)

The state vector corresponding to attitude dynamics is represented by \( \mathbf{X} = \begin{bmatrix} \dot{\theta} \\ \delta \\ \ddot{\theta} \end{bmatrix} \). Linear dynamics associated with these states can be written as:

\[ \begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \delta \end{bmatrix} = \begin{bmatrix} 0 & -\frac{mF_t}{Mt} & 0 \\ 0 & 0 & 1 \\ 0 & \frac{mF_t}{Mt} & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \delta \\ \dot{\delta} \end{bmatrix} + \begin{bmatrix} \frac{-l}{7} \\ 0 \\ \frac{M}{Mc} + \frac{Ml^2}{7Mc} \end{bmatrix} u \]  

\[ (31) \]

Continuous to discrete(c2d) time conversion utility in Matlab with zero-order-hold is used to numerically compute discretized version of the system matrices \( \mathbf{A}_d \) and \( \mathbf{B}_d \).
The objective of the controller is to regulate the body-rate of the missile. Origin is an equilibrium (unstable) point for the attitude actuator dynamics of the system. Therefore, the objective function for the optimization problem is chosen to minimize the sum of squares of the body-rate as well as the actuator states.

\[ \Sigma_{k=1}^{N} (\dot{\theta}[k]^2 + \delta[k]^2 + \dot{\delta}[k]^2) \] (32)

In other words all entries of the \( S \) matrix corresponding to body-rate, actuator position and actuator speed are set to 1. Entries corresponding to the control terms are set to zero. A receding horizon controller with constant value of \( N = 99 \) is used. The lower and upper limits on the optimization state vector are chosen as follows:

\[
\begin{align*}
-\delta_{\text{max}} & \leq \delta[k] \leq \delta_{\text{max}}, \quad k = 1..N - 1 \\
-u_{\text{max}} & \leq u[k] \leq u_{\text{max}}, \quad k = 1..N - 1
\end{align*}
\] (33) (34)

No limits have been placed on \( \dot{\theta}[k] \) and \( \dot{\delta}[k] \). Initial condition on the angular velocity has been chosen as 10rad/s. The primary objective of regulating the body-rate is achieved as seen in Fig.2 within 2 seconds. The controller avoids both control and state saturation as shown in Fig.3 and Fig.4 respectively. The controller is guaranteed to avoid control saturation in the nonlinear simulation. However, it is not theoretically guaranteed to avoid state saturation in the nonlinear simulation. This is due to the fact that a linear model is used in the optimization process instead of the actual nonlinear model. Closed loop results conducted indicate that
the nonlinear simulation also avoids state saturation. Primary reason for this is the fact that the control is computed in a feedback manner at each instant of time. Therefore, to avoid state saturation it is only required that the linear and nonlinear model predictions be close to each other for only time step which is 0.01 sec in this case.

B. Attitude Command Tracking

Attitude command tracking problem is solved using the optimization approach over a fixed horizon of 3 seconds. The sample time of 0.01 seconds has been retained. The control is computed at each instant of time using the state at the current time as the initial condition. The parameter \( N = \frac{\text{time to go}}{\text{sample time}} \) is also re-computed at each instant of time as a function of the time-to-go where the final time is chosen as 3 seconds. The objective function for the optimization problem is chosen as follows:

\[
\sum_{k=1}^{N} \left( 10(\theta[k] - \theta_c[k])^2 + \dot{\theta}[k]^2 + \dot{\delta}[k]^2 \right) \quad (35)
\]

Limits on the actuator mass position and actuator force are retained from the previous example. Attitude command of \( \theta_c[k] = 10 \) degrees has been chosen for this example. Initial condition on the attitude angle has been set to 45 degrees and the initial condition on body-rate has been set to 5 rad/s. Shown in figures 5, 6, and 7 are the attitude angle time history, normalized mass position and normalized control histories. Clearly, the controller works within the limits of the state and constraints while tracking the attitude command within 3 seconds.

![Figure 5. Attitude Angle Time History.](image)

![Figure 6. Normalized Moving Mass Position.](image)
C. Integrated Guidance and Control with Seeker Constraints

Missile guidance and control have been traditionally addressed in a decoupled fashion. An integrated guidance-control system involving the position states, attitude states and actuator states can be designed using approach discussed in this paper. The above approach can be easily extended to include guidance component as well. Linear model corresponding to the inertial position has been obtained in this paper. Inertial frame terminal position co-ordinate specification can be included in the optimization function. Thus an integrated guidance-controller can be designed using the same optimization approach.

A free falling target model is assumed. Vertical translational dynamics of the missile are augmented to the control model. The effect of gravity is eliminated by considering the relative motion dynamics with respect to the target.

\[ \ddot{X}_{\text{target}} = 0, \quad \ddot{Z}_{\text{target}} = g \]  \hspace{1cm} (36)

\[ \Delta X = X_{\text{target}} - X, \quad \Delta Z = Z_{\text{target}} - Z \] \hspace{1cm} (37)

The resulting linear dynamical system is of sixth order. The linear system matrices are:

\[
\begin{bmatrix}
\Delta Z \\
\dot{\theta} \\
\delta \\
\Delta Z \\
\ddot{\theta} \\
\ddot{\delta}
\end{bmatrix}
= \begin{bmatrix}
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & -\frac{F_t}{M_t} & 0 & 0 & 0 & 0 \\
0 & 0 & -\frac{m_F}{M_J} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\Delta Z \\
\dot{\theta} \\
\delta \\
\Delta Z \\
\ddot{\theta} \\
\ddot{\delta}
\end{bmatrix}
+ \begin{bmatrix}
0 \\
0 \\
0 \\
-\frac{1}{M_t-m} \\
-\frac{1}{M_t} \\
\frac{M_t}{M_c} + \frac{M_t I}{T M_c}
\end{bmatrix}
\begin{bmatrix}
u
\end{bmatrix}
\] \hspace{1cm} (38)

An objective function including the relative position states, missile attitude states and the moving mass actuator states is formed as shown in Eq.39. Whereas the attitude and actuator states are penalized over the entire course of interception, the relative position states are penalized only over the terminal phase of interception.

\[ \Sigma_{k=1}^{N} (10 \Delta Z[k]^2 + \Delta \dot{Z}[k]^2) + \Sigma_{k=1}^{N} (\dot{\theta}[k]^2 + 10 \delta[k]^2 + 10 \ddot{\delta}[k]^2) \] \hspace{1cm} (39)

To generate maximum lateral acceleration the missile has a tendency to rotate large angles. Off bore sight angle is the angle between the line of sight vector and the longitudinal axis of the missile. When the off bore sight angle assumes large values the target leaves the field of view of the seeker. Therefore, to always keep the target within the field of view of the seeker it is necessary to impose limits on the attitude of the missile. These limits are not constant unlike the position limits on the moving mass. They are dependent on the relative geometry between the missile and the target. However, in this example the line of sight angle is
so small that limits on off bore sight angle could be approximated as limits on the attitude angle of missile. A 45 degree field of view seeker is chosen in this example. Therefore, the lower and upper limits on the optimization state vector are set to:

\[-45\ \text{degrees} < \theta[k] < 45\ \text{degrees}, \quad k = 1..N - 1\]  
\[-\delta_{\text{max}} < \delta[k] < \delta_{\text{max}}, \quad k = 1..N - 1\]  
\[-u_{\text{max}} < u[k] < u_{\text{max}}, \quad k = 1..N - 1\]

Initial conditions on the missile and target are chosen as follows, \(\mathbf{x}_0 = [0, -100000\ ft, 0, 0, 6000\ ft/s, -10\ ft/s, 0, 0, 6000\ ft/s, 0]\), \(\mathbf{z}_0 = [0, -100000\ ft, -101000\ ft, 5000\ ft/s, 0]\). The value of \(N\) is updated at each instant of time based on the instantaneous range and range rate of the missile-target separation. Time to go is computed as \(\text{time-to-go} = \frac{\text{range}}{\text{range rate}}\) and the value of \(N\) is computed as \(N = \frac{\text{time-to-go}}{\text{sample time}}\). Shown in Fig.8 are the trajectories of the missile and the target. The missile successfully intercepts the target from behind. Attitude angle time history is shown in Fig.9. The controller keeps the attitude of the missile within 45 degrees even in the nonlinear simulation, thereby keeping the target within the field of view of seeker during the entire course of interception.

![Figure 8. Target Interception.](image_url)

![Figure 9. Attitude Angle Time History.](image_url)
V. Conclusion

Planar equations of motion governing the dynamics of a moving mass actuated missile have been derived in closed form. A linear model was then obtained analytically. Different control problems such as attitude-rate regulation, attitude command tracking and target interception are posed as a quadratic programming problems. Limitations on the moving mass position, actuator force and seeker field of view are explicitly accounted as optimization state constraints. The resulting controller has been effectively tested in nonlinear closed loop simulations.
Appendix

\[ \frac{\partial T}{\partial X} = M_1 \dot{X} + mc_0 \delta \dot{\theta} + ms_0 \dot{\delta} + ms_0 l \dot{\theta}, \quad \frac{\partial T}{\partial X} = 0 \]  \hspace{1cm} (43)

\[ \frac{\partial T}{\partial Z} = M_1 \dot{Z} - ms_0 \dot{\delta} + mc_0 \dot{\delta} + mc_0 \dot{l} \dot{\theta}, \quad \frac{\partial T}{\partial Z} = 0 \]  \hspace{1cm} (44)

\[ \frac{\partial T}{\partial \theta} = (I + m \delta^2 + ml^2) \dot{\theta} + mc_0 \dot{X} \delta - ms_0 \dot{Z} \delta + ms_0 \dot{X} \dot{\theta} + mc_0 \dot{X} \dot{\theta} = 0 \]  \hspace{1cm} (45)

\[ \frac{\partial T}{\partial \delta} = -ms_0 \dot{X} \dot{\delta} - mc_0 \dot{Z} \dot{\delta} + mc_0 \dot{X} \dot{\theta} - ms_0 \dot{Z} \dot{\theta} = 0 \]  \hspace{1cm} (46)

\[ \frac{\partial T}{\partial \dot{\delta}} = m \dot{\delta} + ms_0 \dot{X} + mc_0 \dot{Z} + ml \dot{\theta} \]  \hspace{1cm} (47)

\[ \frac{\partial T}{\partial \dot{\delta}} = m \dot{\delta}^2 + mc_0 X \dot{\theta} - ms_0 Z \dot{\theta} \]  \hspace{1cm} (48)

\[ F_b = \begin{bmatrix} -M_0 s_0 \\ 0 \\ M_0 g \theta \end{bmatrix}, \quad F_1 = \begin{bmatrix} F_T \\ 0 \\ 0 \end{bmatrix}, \quad F_2 = \begin{bmatrix} -F_x \\ 0 \\ -u \end{bmatrix}, \quad F_m = \begin{bmatrix} F_x - m g s_0 \\ 0 \\ u + mg \theta \end{bmatrix} \]  \hspace{1cm} (49)

\[ \dot{R}_b = \begin{bmatrix} c_0 \dot{X} - s_0 \dot{Z} \\ 0 \\ s_0 \dot{X} + c_0 \dot{Z} \end{bmatrix}, \quad \dot{R}_1 = \begin{bmatrix} c_0 \dot{X} - s_0 \dot{Z} \\ 0 \\ \dot{l} \dot{\theta} + s_0 \dot{X} + c_0 \dot{Z} \end{bmatrix}, \quad \dot{R}_2 = \begin{bmatrix} c_0 \dot{X} - s_0 \dot{Z} + \delta \dot{\theta} \\ 0 \\ s_0 \dot{X} + c_0 \dot{Z} + \dot{l} \dot{\theta} \end{bmatrix}, \quad \dot{R}_m = \begin{bmatrix} c_0 \dot{X} - s_0 \dot{Z} + \delta + \delta \dot{\theta} \\ 0 \\ s_0 \dot{X} + c_0 \dot{Z} + \dot{l} \dot{\theta} \end{bmatrix} \]  \hspace{1cm} (50)

\[ Q_X = F_b \frac{\partial \dot{R}_b}{\partial X} + F_m \frac{\partial \dot{R}_m}{\partial X} + F_1 \frac{\partial \dot{R}_1}{\partial X} + F_2 \frac{\partial \dot{R}_2}{\partial X} \]  \hspace{1cm} (51)

\[ Q_Z = F_b \frac{\partial \dot{R}_b}{\partial Z} + F_m \frac{\partial \dot{R}_m}{\partial Z} + F_1 \frac{\partial \dot{R}_1}{\partial Z} + F_2 \frac{\partial \dot{R}_2}{\partial Z} \]  \hspace{1cm} (52)

\[ Q_\theta = F_b \frac{\partial \dot{R}_b}{\partial \theta} + F_m \frac{\partial \dot{R}_m}{\partial \theta} + F_1 \frac{\partial \dot{R}_1}{\partial \theta} + F_2 \frac{\partial \dot{R}_2}{\partial \theta} \]  \hspace{1cm} (53)

\[ Q_\delta = F_b \frac{\partial \dot{R}_b}{\partial \delta} + F_m \frac{\partial \dot{R}_m}{\partial \delta} + F_1 \frac{\partial \dot{R}_1}{\partial \delta} + F_2 \frac{\partial \dot{R}_2}{\partial \delta} \]  \hspace{1cm} (54)

\[ \frac{\partial \dot{R}_b}{\partial X} = \frac{\partial \dot{R}_1}{\partial X} = \frac{\partial \dot{R}_2}{\partial X} = \frac{\partial \dot{R}_m}{\partial X} = \begin{bmatrix} c_0 \\ 0 \\ s_0 \end{bmatrix} \]  \hspace{1cm} (55)

\[ \Rightarrow Q_X = \begin{bmatrix} -M_0 s_0 \\ 0 \\ M_0 g \theta \end{bmatrix}^T + \begin{bmatrix} F_T \\ 0 \\ 0 \end{bmatrix}^T + \begin{bmatrix} -F_x \\ 0 \\ -u \end{bmatrix}^T + \begin{bmatrix} F_x - m g s_0 \\ 0 \\ u + mg \theta \end{bmatrix}^T \begin{bmatrix} c_0 \\ 0 \\ s_0 \end{bmatrix} \]  \hspace{1cm} (56)

\[ \Rightarrow Q_X = F_T c_0 \]  \hspace{1cm} (57)

\[ \frac{\partial \dot{R}_b}{\partial \delta} = \frac{\partial \dot{R}_1}{\partial \delta} = \frac{\partial \dot{R}_2}{\partial \delta} = \frac{\partial \dot{R}_m}{\partial \delta} = \begin{bmatrix} -s_0 \\ 0 \\ c_0 \end{bmatrix} \]  \hspace{1cm} (58)
$$Q_Z = \left\{ \begin{bmatrix} -M_b s_{\theta} \\ 0 \\ M_b c_{\theta} \end{bmatrix}^T + \begin{bmatrix} F_T \\ 0 \\ 0 \end{bmatrix}^T + \begin{bmatrix} -F_x \\ 0 \\ -u \end{bmatrix}^T + \begin{bmatrix} F_x - mgs_{\theta} \\ 0 \\ u + mc_{\theta} \end{bmatrix}^T \right\}^T \begin{bmatrix} -s_{\theta} \\ 0 \\ c_{\theta} \end{bmatrix}$$ (59)

$$Q_Z = -F_T s_{\theta} + M_b g$$ (60)

$$\frac{\partial \hat{\mathbf{R}}_b}{\partial \theta} = 0, \quad \frac{\partial \hat{\mathbf{R}}_1}{\partial \theta} = \begin{bmatrix} 0 \\ 0 \\ l \end{bmatrix}, \quad \frac{\partial \hat{\mathbf{R}}_2}{\partial \theta} = \begin{bmatrix} \delta \\ 0 \\ l \end{bmatrix}, \quad \frac{\partial \hat{\mathbf{R}}_m}{\partial \theta} = \begin{bmatrix} \delta \\ 0 \\ l \end{bmatrix}$$ (61)

$$Q_{\delta} = \left[ \begin{array}{ccc} F_T & 0 & 0 \\ 0 & l & -u \\ 1 & 0 & 0 \end{array} \right] + \left[ \begin{array}{ccc} -F_x \\ 0 \\ u + mc_{\theta} \end{array} \right]$$ (62)

$$Q_{\delta} = -mgs_{\theta} \delta + mgc_{\theta} l$$ (63)

$$\frac{\partial \hat{\mathbf{R}}_b}{\partial \delta} = 0, \quad \frac{\partial \hat{\mathbf{R}}_1}{\partial \delta} = 0, \quad \frac{\partial \hat{\mathbf{R}}_2}{\partial \delta} = 0, \quad \frac{\partial \hat{\mathbf{R}}_m}{\partial \delta} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$ (64)

$$Q_{\delta} = \left[ \begin{array}{ccc} F_x - mgs_{\theta} & 0 & u + mc_{\theta} \end{array} \right]$$ (65)

$$Q_{\delta} = u + mc_{\theta}$$ (66)

References


