

Design of Intelligent Missile Control System Using Model Predictive Control

M. Fawzy^(*), M. A. S. Aboelela⁽⁺⁾, O. Abd El Rhman^(*), and H. T. Dorrah⁽⁺⁾

^(*) Egyptian Armed Forces

⁽⁺⁾ Cairo University, Faculty of Engineering,
Electric Power and Machines Dept., Giza, Egypt.

Abstract

The goal of this paper is to control the trajectory of the flight path of six degree of freedom flying body model using Model predictive control (MPC) controller. MPC controller with constraints will be developed and able to compensate for constraints that represent physical limits of actuators in pitch and yaw angles. The design of MPC controller with linear system for the six degree of freedom flying body is described. MPC controllers are computationally intensive because an on-line optimization problem is formed and solved at each control cycle.

Keywords: six degree of freedom missile model, Linearization model, Model predictive control

1. Introduction

In recent years, the requirements for the quality of automatic control increased significantly due to increased complexity of plants and sharper specifications of product. This paper will address the design of optimal variable structure controllers applied to a six degree of freedom missile model. The six degree of freedom missile model is the solution to obtain a detailed accurate data about the missile trajectory. Linear model of the investigated systems will be considered. The linearization will be obtained during two phases that are boost phase and sustain phase so the controllers are designed for two linear time invariant LTI models. The paper objectives are

- To develop a general flexible sophisticated mathematical model of flight trajectory simulation for a hypothetical anti tank missile, which can be used as a base line algorithm contributing for design, analysis, and development of such a system and implement this model using Simulink to facilitate the design of its control system
- Developing control system, by using MPC control techniques

This paper is organized as follows. Section 2 reviews mathematical model of six degree of freedom missile equations and linearization model is represented. Section 3 gives MPC

controller design. Section 4 presents control applications and results. Finally, conclusions are outlined in section 5.

2. Mathematical model of the missile

The model constitutes the six degree of freedom (6-DOF) equations that break down into those describing kinematics, dynamics (aerodynamics, thrust, and gravity), command guidance generation systems, and autopilot (electronics, instruments, and actuators). The input to this model is launch conditions, target motion, and target trajectory characterization, while the outputs are the missile flight data (speed, acceleration, range, etc.) during engagement.

2.1 nonlinear six degree of freedom missile model

The basic frames needed for subsequent analytical developments are the ground, body and velocity coordinate systems. The origins of these coordinate systems are the missile center of gravity (c.g). In the ground coordinate system, the $X_g - Z_g$ lie in the horizontal plane and the Y_g axis completes a standard right-handed system and points up vertically. In the body coordinate system, the positive X_b axis coincides with the missile's center line and it is designated as roll-axis. The positive Z_b axis is to the right of the X_b axis in the horizontal plane and it is designated as the pitch axis. The positive Y_b axis points upward and it is designed as the yaw axis. The body axis system is fixed with respect to the missile and moves with the missile. In the velocity coordinate system, X_v coincides with direction of missile velocity (V_m), which related to the directions of missile flight. The axis Z_v completes a standard right-handed system, [6, 11].

The pitch plane is X-Y plane, the yaw plane is X-Z plane, and the roll plane is Y-Z plane. The ground coordinate system and body coordinate system are related to each other through Euler's angles (Φ, Ψ, Υ). The ground coordinate system and velocity coordinate system are related to each other through the angles (θ, σ). In addition, the velocity coordinate system is related to the body frame through the angle of attack (α) in the pitch plane and angle of attack (β) in the yaw plane (sideslip angle). The angles between different coordinate systems are shown in Fig. 1, [6, 11].

The relation between the body and the velocity coordinate systems can be given as follows

$$\begin{bmatrix} X_b \\ Y_b \\ Z_b \end{bmatrix} = \begin{bmatrix} \cos(\beta) \cos(\alpha) & \cos(\beta) \sin(\alpha) & -\sin(\beta) \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ \sin(\beta) \cos(\alpha) & \sin(\beta) \sin(\alpha) & \cos(\beta) \end{bmatrix} \begin{bmatrix} X_v \\ Y_v \\ Z_v \end{bmatrix} \quad (1)$$

The body and velocity axes system as well as forces, moments and other quantities are shown in Fig. 2

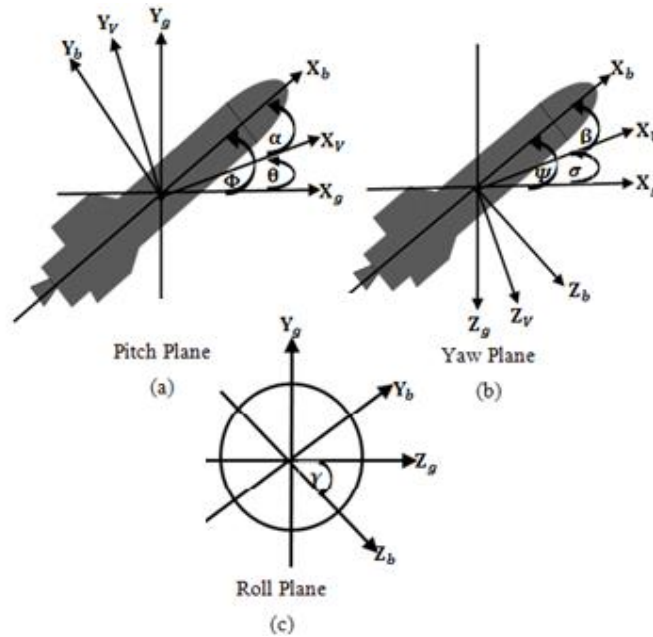


Fig. 1 The angles between different coordinate systems

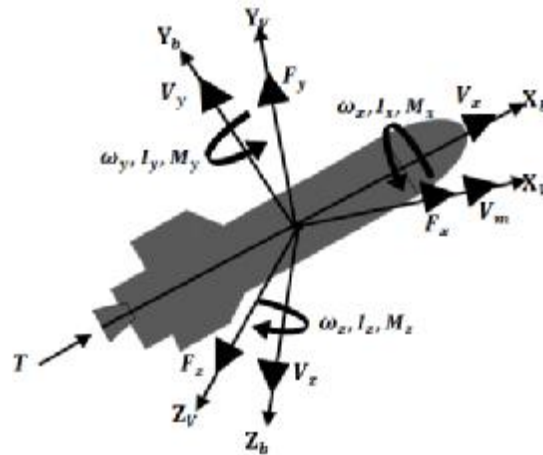


Fig. 2 Forces, moments and other quantities

There are 6 dynamic equations (3 for translational motion and 3 for rotational motion) and 6 kinematic equations (3 for translational motion and 3 for rotational motion) for a missile with six degrees of freedom. The equations are somewhat simpler, if the mass is constant. The missile 6DOF equations in velocity coordinate system are given as following, [6]

$$F_x = m\dot{V}_m \quad (2)$$

$$F_y = mV_m\dot{\theta} \quad (3)$$

$$F_z = -mV_m \cos(\theta) \dot{\sigma} \quad (4)$$

$$\mathbf{M}_x = \mathbf{I}_x \dot{\omega}_x - (\mathbf{I}_y - \mathbf{I}_z) \omega_y \omega_z \quad (5)$$

$$\mathbf{M}_y = \mathbf{I}_y \dot{\omega}_y - (\mathbf{I}_z - \mathbf{I}_x) \omega_z \omega_x \quad (6)$$

$$\mathbf{M}_z = \mathbf{I}_z \dot{\omega}_z - (\mathbf{I}_x - \mathbf{I}_y) \omega_x \omega_y \quad (7)$$

$$\dot{X} = V_m \cos(\theta) \cos(\sigma) \quad (8)$$

$$\dot{Y} = V_m \sin(\theta) \quad (9)$$

$$\dot{Z} = -V_m \cos(\theta) \sin(\sigma) \quad (10)$$

$$\dot{\Psi} = (\omega_y \cos(\Upsilon) - \omega_z \sin(\Upsilon)) / \cos(\Phi) \quad (11)$$

$$\dot{\Phi} = \omega_y \sin(\Upsilon) + \omega_z \cos(\Upsilon) \quad (12)$$

$$\dot{\Upsilon} = \omega_x - \tan(\Phi) (\omega_y \cos(\Upsilon) - \omega_z \sin(\Upsilon)) = \omega_x - \dot{\Psi} \sin(\Phi) \quad (13)$$

In these equations, $\mathbf{F}_x, \mathbf{F}_y, \mathbf{F}_z$ are component of forces acting on missile in velocity coordinate system; $\mathbf{M}_x, \mathbf{M}_y, \mathbf{M}_z$ are moments acting on missile in body coordinate system; $\omega_x, \omega_y, \omega_z$ are angular velocity in body coordinate system; $\mathbf{I}_x, \mathbf{I}_y, \mathbf{I}_z$ are moments of inertia in body coordinate system; X is missile range; Y is missile altitude; Z is horizontal displacement of the missile; and m is missile mass. The forces and the moments acting on missile are due to thrust, gravity and aerodynamic forces and moments are given as following, [4, 6, 11].

$$\mathbf{F}_x = \mathbf{T} \cos(\alpha - \delta_\alpha) \cos(\beta - \delta_\beta) - \mathbf{QS} \left(\mathbf{C}_{x0} + \mathbf{C}_x (\alpha^2 + \beta^2) \right) - \mathbf{mg} \sin(\theta) \quad (14)$$

$$\mathbf{F}_y = \mathbf{T} \sin(\alpha - \delta_\alpha) + \mathbf{QSC}_y \alpha - \mathbf{mg} \cos(\theta) \quad (15)$$

$$\mathbf{F}_z = -\mathbf{T} \cos(\alpha - \delta_\alpha) \sin(\beta - \delta_\beta) - \mathbf{QSC}_z \beta \quad (16)$$

$$\mathbf{M}_x = \mathbf{DQS} \mathbf{m}_{x0} \frac{\omega_x \mathbf{D}}{2V_m} \quad (17)$$

$$\mathbf{M}_y = -\mathbf{T} \cos(\delta_\alpha) \sin(\delta_\beta) \mathbf{X}_{cg} + \mathbf{DQS} \left(\mathbf{m}_{y\beta} \beta + \mathbf{m}_{y0} \frac{\omega_y \mathbf{D}}{V_m} \right) \quad (18)$$

$$\mathbf{M}_z = \mathbf{T} \sin(\delta_\alpha) \mathbf{X}_{cg} + \mathbf{DQS} \left(\mathbf{m}_{z\alpha} \alpha + \mathbf{m}_{z0} \frac{\omega_z \mathbf{D}}{V_m} \right) \quad (19)$$

In these equations, $\mathbf{C}_x, \mathbf{C}_{x0}, \mathbf{C}_y, \mathbf{C}_z$ are aerodynamic force coefficient; $\mathbf{m}_{x0}, \mathbf{m}_{y\beta}, \mathbf{m}_{y0}, \mathbf{m}_{z\alpha}, \mathbf{m}_{z0}$ are aerodynamic moment coefficients; \mathbf{D} is the diameter of maximum cross section area of body; \mathbf{S} is the reference area; \mathbf{Q} is the dynamic pressure; δ_α is the nozzle deflection angle in the pitch plane; δ_β is the nozzle deflection angle in the yaw plane; \mathbf{T} is the thrust force; \mathbf{X}_{cg} is the distance between center of gravity (c.g) and the nozzle; and g is acceleration due to gravity and is taken to be constant 9.81 m/sec².

2.2 Linearization of six degree-of-freedom missile (LTI model)

The linearized model takes the following form

$$\dot{\mathbf{x}} = \mathbf{A} \mathbf{x} + \mathbf{B} \mathbf{u} \quad (20)$$

$$\mathbf{y} = \mathbf{C} \mathbf{x} + \mathbf{D} \mathbf{u} \quad (21)$$

$$\mathbf{x} = [V_m \quad \theta \quad \sigma \quad X \quad Y \quad Z \quad \Psi \quad \Phi \quad \dot{\Psi} \quad \dot{\Phi} \quad \omega_x \quad \omega_y \quad \omega_z]^T$$

$$\mathbf{u} = [\delta_\alpha \quad \delta_\beta \quad t]^T \quad \mathbf{y} = [\Psi \quad \Phi]^T$$

Where matrices A, B, C and D are matrix coefficient of LTI system; x is state vector; u is input vector; y is output vector.

Missile solid propellant thrust will be divided into two phases, first phase is Boost phase that will take about 5.8 sec of total flight time ($0 \leq t < 5.8\text{sec}$) and thrust force $T=T_{\max}$. The second phase is Sustain phase that will start after boost region until the impact with target ($5.8 \leq t < 25\text{sec}$) and thrust force $T=T_{\min}$. Therefore, we will discuss the linearization of missile motion equation at boost and sustain phases.

A linear time-invariant (LTI) model is implemented in a boost phase around the operating point at $t = 0 \text{ sec}$. A linear time-invariant (LTI) model is implemented in sustain phase around the operating point at $t = 5.8 \text{ sec}$. Open loop LTI model of missile motion equation is represented in Fig.3, [2, 9].

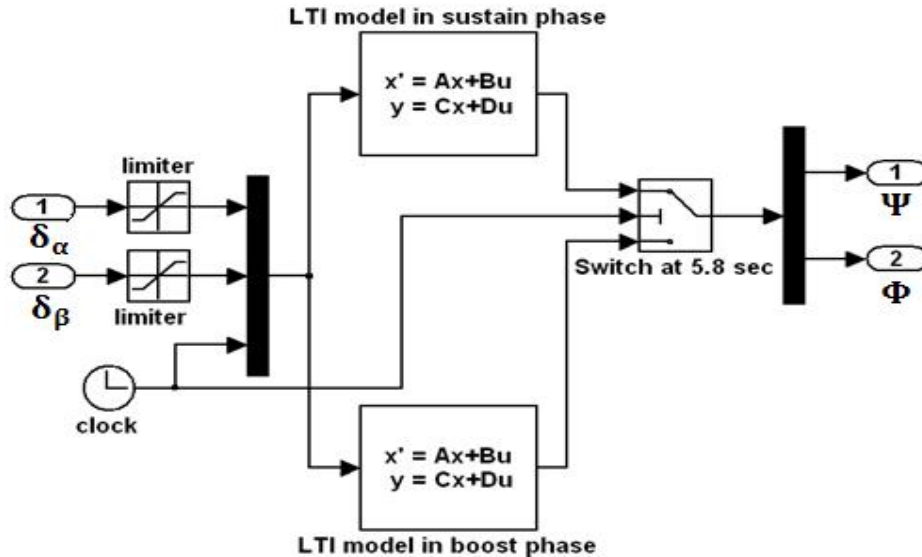


Fig.3 Open loop LTI model of six degree-of-freedom missile equation

3. Model Predictive Control (MPC) design

Model Predictive Control (MPC) is an advanced method of process control that has been in use in the process industries since the 1980s. MPC is a control strategy that is suitable for optimizing the performance of constrained systems. Constraints are present in all control systems due to actuators physical limits, boundaries of safe operation and lower limits for product quality. The MPC uses the same powerful linear dynamic modeling that employ transfer functions, state-space matrices, or a combination of the two. MPC systems rely on the idea of generating values for process inputs as solutions of an on-line (real-time) optimization problem. This problem is based on a process model and process measurements, [5, 10, 12].

3.1 Model Predictive Control of SISO Plant

Fig. 4 shows block diagram of single input single output (SISO) plant MPC controller. The main objective is to hold a single unmeasured output y_u at a reference value (or setpoint), r , by adjusting a single manipulated variable (or actuator) u . The SISO plant actually has multiple inputs (manipulated variable input u , measured disturbance v and unmeasured disturbance d). The controller receives the measured disturbance v directly. This allows the controller to compensate for measured disturbance impact on y_u immediately rather than waiting until the effect appears in the measured output y_m . This is called feedforward control. MPC design always provides feedback compensation for unmeasured disturbances and feedforward compensation for any measured disturbance, The MPC design removes the estimated noise z component of the measurement (filtering). [9].

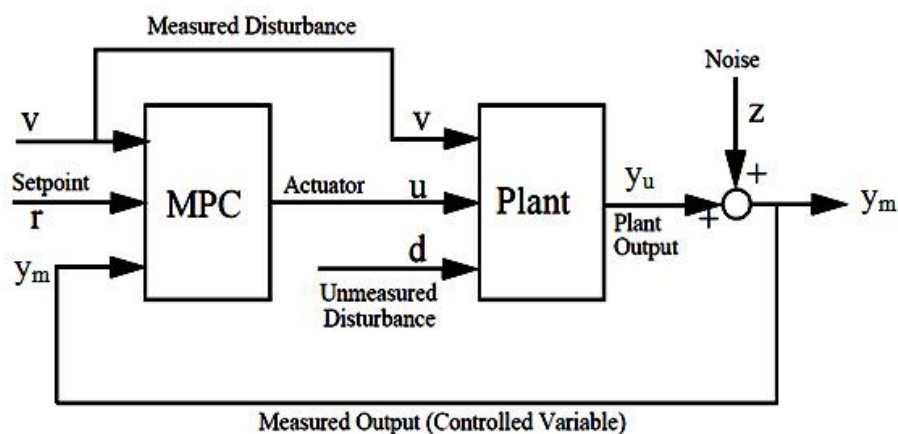


Fig.4 Block diagram of SISO plant MPC controller

3.2 MPC strategy

The methodology of all the controllers belonging to the MPC family is characterized by the following strategy, [5]:

1. The future outputs for a determined prediction horizon P (future sampling periods $P \geq 1$) are predicted at each sampling instant k using the process model. These prediction outputs $\mathbf{y}(k+i|\mathbf{k})$ for $i=1, \dots, P$ depend on the known values up to instant k (past input and output) and on the future control signals $\mathbf{u}(k+i-1|\mathbf{k})$ for $i=1, \dots, M$ (M is control horizon for future moves and $1 \leq M \leq P$) which are those to be sent to the system and calculated. The prediction future output can be explained briefly in [10, 12].
2. The set of future control signals $\mathbf{u}(k+i-1|\mathbf{k})$ is calculated by optimizing a determined criterion to keep the process as close as possible to the reference trajectory $r(k+i)$ that can be the setpoint itself or a close approximation of it. This criterion usually takes the form of a quadratic function of the errors between the predicted output signal and the predicted reference trajectory. The control effort is included in the objective function in most cases. An explicit solution can be obtained if the criterion is quadratic, the model is linear, and there are no constraints; otherwise an iterative optimization method has to be used. Some assumptions about the structure of the future control law are also made in some cases, such as that it will be constant from a given instant.
3. The control signal $\mathbf{u}(k|\mathbf{k})$ is sent to the process whilst the next control signals calculated are rejected, because at the next sampling instant $y(k+1)$ is already known and step 1 is repeated with this new value and all the sequences are brought up to date. Thus the $\mathbf{u}(k+1|\mathbf{k}+1)$ is calculated, which in principle will be different from the $\mathbf{u}(k+1|\mathbf{k})$ because of the new information available, using the receding horizon concept.

Fig. 5 shows the state of a hypothetical SISO MPC system that has been operating for many sampling instants. Integer k represents the current sampling instant. The current measured output, $y(k)$, and previous measurements $y(k-1)$, $y(k-2)$,..... are known and are the filled circles in Fig. 5 (a), [7, 9].

Fig. 5 (b) shows the controller's previous moves, $u(k-4)$, $u(k-3)$, ..., $u(k-1)$ as filled circles. As is usually the case, a zero-order hold receives each move from the controller and holds it until the next sampling instant, causing the step-wise variations shown in Fig. 5 (b).

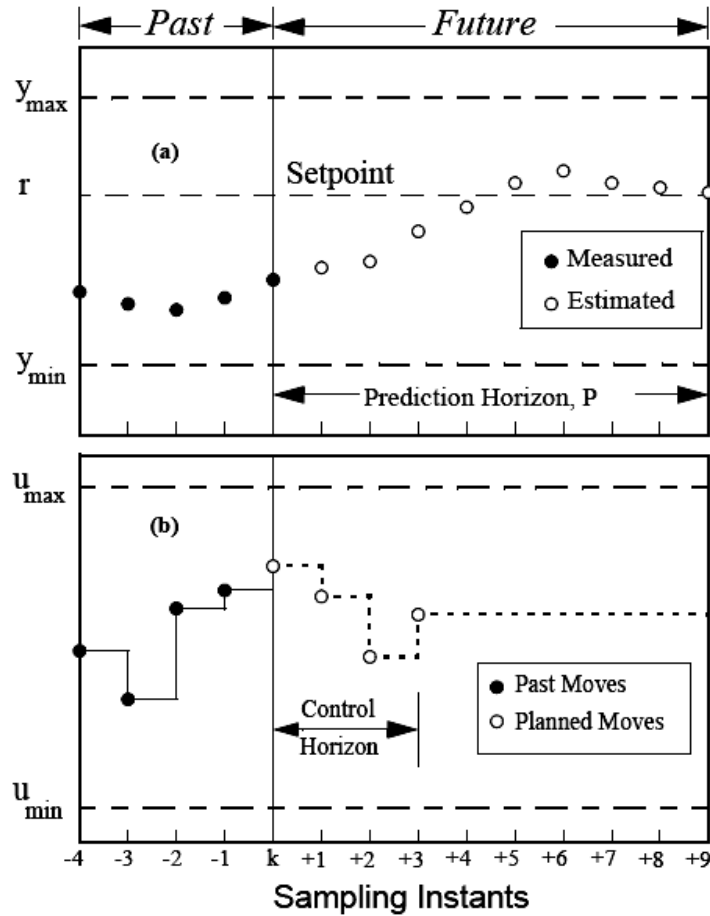


Fig. 5 MPC controller at the k^{th} sampling instant

3.3 Objective function

For the basic formulation of predictive control, we shall assume that the plant model is linear, that the objective function (cost function) is quadratic, and that constraints are in the form of linear inequalities. Furthermore, we shall assume that the cost function does not penalize particular values of the input vector $\mathbf{u}(k)$, but only changes of the input vector, $\Delta \mathbf{u}(k) = \mathbf{u}(k) - \mathbf{u}(k - 1)$. This formulation coincides with that used in the majority of the predictive control literature. To make the useful formulation, we shall not assume the state variables can be measured, but that is obtained an estimate $\hat{\mathbf{x}}(k|k)$ of the state $\mathbf{x}(k)$. Signals $\mathbf{u}(k + i - 1|k)$ will denote a future value of the input \mathbf{u} . Signals $\mathbf{x}(k + i|k)$ and $\mathbf{y}(k + i|k)$ will denote the predictions, made at time k , of the variables \mathbf{x} and \mathbf{y} at time $k+i$, [5].

A cost function J penalizes deviations of the predicted controlled outputs $\mathbf{y}(k + i|k)$ from a reference trajectory $\mathbf{r}(k + i)$. The reference trajectory may depend on measurements made up to time k , in particular, its initial point is the output measurement $\mathbf{y}(k)$. The reference trajectory may also be a fixed set point, or some other predetermined trajectory. The general expression for such an objective function (cost function) will be [5, 10, 12]:

$$J = \sum_{i=1}^P \|y(\mathbf{k} + i | \mathbf{k}) - r(\mathbf{k} + i)\|_{Q(i)}^2 + \sum_{i=1}^M \|\Delta \mathbf{u}(\mathbf{k} + i - 1 | \mathbf{k})\|_{R(i)}^2 \quad (22)$$

$$\text{subject to } \begin{cases} \mathbf{u}_{\min} \leq \mathbf{u}(\mathbf{k} + i - 1 | \mathbf{k}) \leq \mathbf{u}_{\max} & \mathbf{i} = 1, \dots, M \\ \Delta \mathbf{u}_{\min} \leq \Delta \mathbf{u}(\mathbf{k} + i - 1 | \mathbf{k}) \leq \Delta \mathbf{u}_{\max} & \mathbf{i} = 1, \dots, M \\ y_{\min} \leq y(\mathbf{k} + i | \mathbf{k}) \leq y_{\max} & \mathbf{i} = 1, \dots, P \\ \Delta \mathbf{u}(\mathbf{k} + i - 1 | \mathbf{k}) = \mathbf{0} & \mathbf{i} = M + 1, \dots, P \end{cases} \quad (23)$$

Where:

- $Q(i), R(i)$ are nonnegative optimization weight coefficients.
- $\|\mathbf{x}\|_Q^2 = \mathbf{x}^T Q \mathbf{x}$

In some methods the second term, which considers the control effort, is not taken into account, whilst in others the values of the control signal (not its increment) also appear directly.

3.4 Constraints

In practice all process are subject to constraints. The actuators have a limited field of action and a determined slew rate, as is the case of the valves, limited by the positions of totally open or closed and by the response rate. Constructive reasons, safety or environmental ones, or even the sensor scopes themselves can cause limits in the process variables such as levels in tanks, flows in piping, or maximum temperatures and pressures; moreover, the operational conditions are normally defined by the intersection of certain constraints for basically economic reasons, so that the control system will operate close to the boundaries. All of this makes the introduction of constraints in the function to be minimized necessary. Normally, bounds in the amplitude and limits in the output will be considered, [3, 12].

By adding constraints to the objective function, the minimization becomes more complex, so that the solution cannot be obtained explicitly as in the unconstrained case. Constraints may be either hard or soft. A hard constraint must not be violated. Unfortunately, under some conditions a constraint violation might be unavoidable (e.g., an unexpected, large disturbance), and a realistic controller must allow for this. The MPC does so by softening each constraint, making a violation mathematically acceptable, though discouraged. The designer may specify the degree of softness in each case, making selected constraints less likely to be violated than others. Briefly, you specify a tolerance band for each constraint. If the tolerance band is zero, the constraint is hard (no violation allowed). Increasing the tolerance band softens the constraint. The tolerance band is not a limit on the constraint

violation, however. (If it were, you would still have a hard constraint.) You need to view it relative to other constraints, [9].

4. Control application and results

In this section, the autonomous flight of six degree of freedom flying body is simulated. The goal is to control the trajectory of the flight path of six degree of freedom flying body model using MPC controller. The design of MPC controller with LTI system for the six degree of freedom flying body is described. This design has been implemented in a simulation environment under Matlab's toolbox Simulink.

4.1 LTI Model description

A linear time-invariant (LTI) model is implemented around the operating point at $t = 0$ sec. The state and input at this point are described as following

$$\mathbf{x}_0 = [8.6 \quad 0.6108 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0.6108 \quad 0 \quad 0 \quad 0 \quad 0]^T$$

$$\mathbf{u}_0 = [0 \quad 0 \quad 0]^T$$

The nozzle deflection angle in pitch plane (δ_α) and yaw plane (δ_β) are limited with $\pm 28.5^\circ$ (± 0.5 rad).

4.2 Building demand generator (Reference trajectory)

The pitch demand programmer is an exponential command and is described as

$$\Phi_p = \Phi_{p0} - (\Phi_A - \Phi_B e^{-t/\tau_p}) \quad (24)$$

where Φ_{p0} is the missile-launching angle with respect to the horizon; Φ_A, Φ_B are vertical position angles depending on target position. For our simulation $\Phi_{p0} = 35^\circ$; $\Phi_A = \Phi_B = 30^\circ$; $\tau_p = 2.1788$ sec .

The yaw demand programmer is an exponential command and is described as

$$\Psi_p = \Psi_s (1 - e^{-t/\tau_\psi}) \quad (25)$$

where Ψ_s is horizontal position angle depending on target position. For our simulation $\Psi_s = 5^\circ$; $\tau_\psi = 0.2$ sec

4.3MPC Controller design

To design an MPC controller the MPC toolbox in Matlab's toolbox Simulink has been used. The controller design requires a LTI model of the plant that is to be controlled. Multi input multi output MPC controller is designed where one MPC controller is used during boost and sustain phases.

4.3.1 Controller parameters

The rate at which MPC operates is $\frac{1}{NT_S}$, where T_S is control interval (sampling period), N is the number of controls in the control history that are applied to the plant. $N=1$ is chosen since this is the value N typically takes. The value of T_S is important since it is the length of each prediction step and the duration each control input is held constant. The method for choosing T_S for this problem is based on tracking performance. After further tests $T_S = 0.01 \text{ sec}$ is chosen since this value of T_S gives the best tracking performance to a sequence of pitch and yaw demand generator programs. [1]

Selecting the prediction horizon P was also affected by the controller. To keep the controller simple, the prediction horizon P and control horizon M were set equal to each other. After further tests a value of $P=M=1$ was chosen since these value of P, M give the best tracking performance to a sequence of pitch and yaw demand generator program. [1]

Optimization parameters (Q, R) start with identity matrices; the values were changed through trial and error to improve the tracking performance to a sequence of pitch and yaw demand generator program [1]. After further tests a values of Q, R was chosen as following

$$Q = \begin{bmatrix} 3.1899 & 0 \\ 0 & 3.1899 \end{bmatrix}, \quad R = 0 \quad (26)$$

4.4 PID controller design

The PID controller has three unknown parameter k_p, k_i and k_d that are required to be designed. Hence, the present problem of controller tuning can be solved by an application of the particle swarm optimization (PSO) algorithm for optimization on a three-dimensional solution space, each particle having a three-dimensional position and velocity vector. The initial positions of the i th particles of the swarm can be represented by a three dimensional vector, and then the initial values are randomly generated based on the extreme values. PSO algorithm and its parameters chosen can be explained briefly in [8, 13].

The position vector of the best particle gives the optimized parameter of integer PID controller as following, [8]

- The PID controller parameters for pitch angle during boost phase are $k_p = 20.432$, $k_i = 4.1353$, $k_d = 1.241$.
- The PID controller parameters for pitch angle during sustain phase are $k_p = 30.2494$, $k_i = 20.6635$, $k_d = 5.4857$.
- The PID controller parameters for yaw angle during boost phase are $k_p = -20.432$, $k_i = -4.1353$, $k_d = -1.241$.
- The PID controller parameters for yaw angle during sustain phase are $k_p = -30.2494$, $k_i = -20.6635$, $k_d = -5.4857$.

4.5 Results

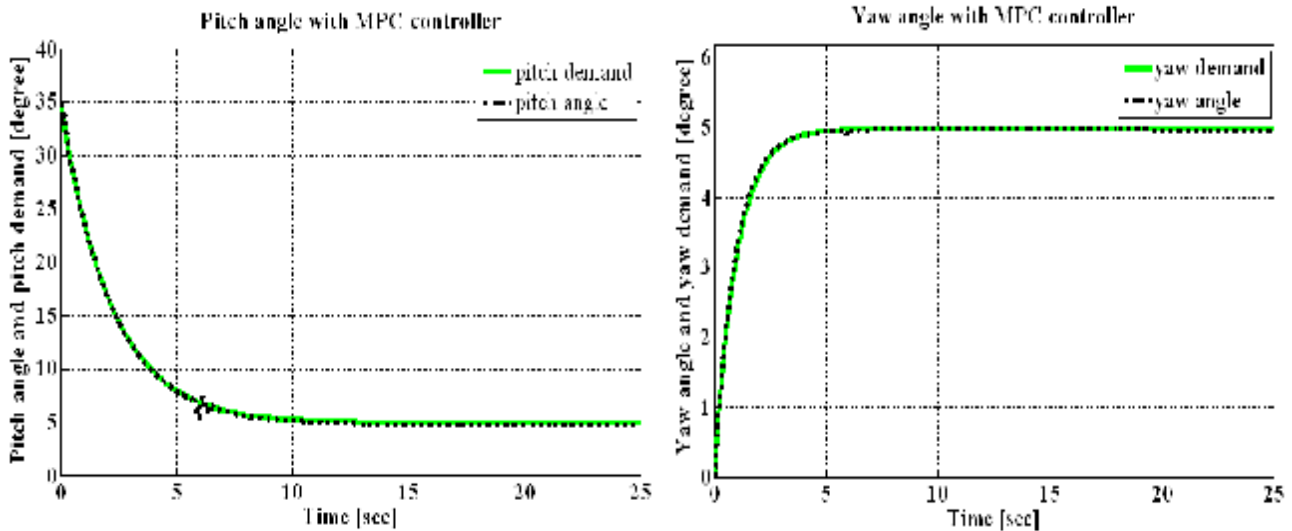


Fig. 6 Pitch and yaw angles with MPC controller vs. time

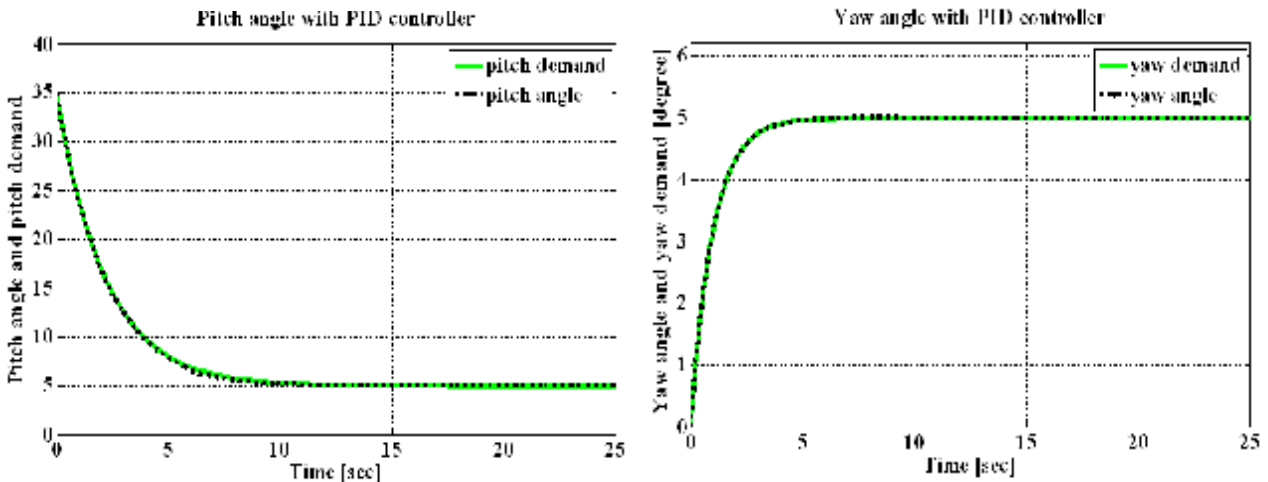


Fig. 7 Pitch and yaw angles with PID controller vs. time

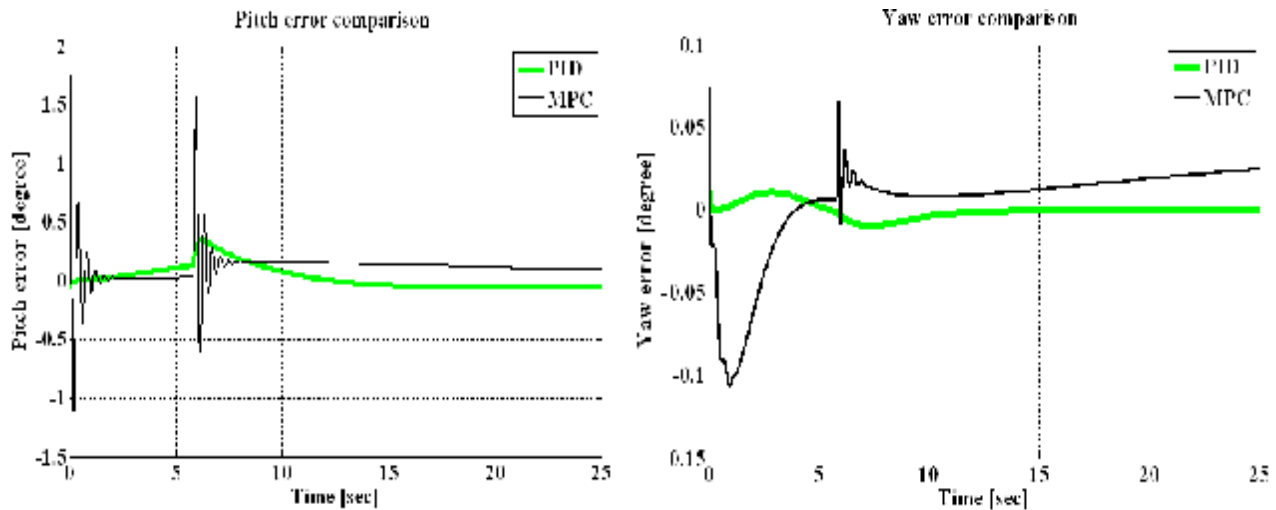


Fig.8 Pitch error and yaw error comparisons with PID and MPC

The pitch error is the difference between pitch demand program (pitch reference trajectory) and pitch angle response. Fig.8 represents the pitch error comparison with PID and MPC. The pitch error with MPC has high overshoot and high oscillation at starts of boost and sustain phases (at $t=0$ and $t=5.8\text{sec}$) and also it has small steady state error during sustain phase. However, for pitch error with PID controller has small overshoot and there is no oscillation.

The yaw error is the difference between yaw demand program (yaw reference trajectory) and yaw angle response. Fig.8 represents the yaw error with PID and MPC. The yaw error of nonlinear system with MPC has high overshoot at start of boost (at $t=0$) and it has high oscillation at start of sustain phase ($t=5.8\text{sec}$), also the steady state error increases during sustain phase. However, for yaw error with PID controller has small overshoot.

5. Conclusion

The design of PID controllers gave the best response for pitch and yaw angles where there is no oscillation (chattering) and has small overshoot. The parameters optimization of PID controllers based on PSO method was highly effective. According to optimization target, the PSO method could search the best global solution for PID controllers' parameters and guarantee the objective solution space in defined search space.

The design of MPC gave response less quality than that was given from PID controller but acceptable responses. However, MPC controller can be used to control a great variety of processes (one MPC controller was used instead of four PID controllers). MPC controller is a limited knowledge of control because its tuning is relatively easy. MPC controller is simple to

the treatment of constraints and relies on the idea of generating values for process inputs as solutions of an on-line (real-time) optimization problem.

6. References

- [1] Alaniz, A., "Model Predictive Control with Application to Real Time Hardware and a Guided Parafoil", Master degree of Science in Aeronautics and Astronautics, Massachusetts Institute of Technology, June 2004
- [2] Ashish T., "Modern Control Design with MATLAB and SIMULINK" Indian Institute of Technology, Kanpur, India, John Wiley & Sons, 2002.
- [3] Bemporad, A., Morari, M., Dua, V. and Pistikopoulos, E. N., "The Explicit Solution of Model Predictive Control via Multi parametric Quadratic Programming", American Control Conference, Chicago, Illinois, June 2000.
- [4] Blakelock, J. H., "Automatic Control of Aircraft and Missiles", John Wiley & Sons, Inc., USA, 2nd edition, 1991.
- [5] Camacho, E. F. and Bordons, C., "Model Predictive Control", Springer-Verlag, London, 2nd edition, 2004.
- [6] Garnell, P., "Guided weapon control systems", Pergamon Press, Oxford, New York, 2nd edition, 1980.
- [7] Lazar, M., "Model Predictive Control of Hybrid Systems: Stability and Robustness", Ph.D. degree of Science in Control system, Faculty of Electrical Engineering, Eindhoven University of Technology, September 2006.
- [8] Maiti, D., Acharya, A., Chakraborty, M. and Konar, A., "Tuning PID and $PI^{\lambda}D^{\delta}$ Controllers using the Integral Time Absolute Error Criterion", IEEE, 2008.
- [9] MATLAB® 9.0 - User's Guide, the Math Works Inc., Natick, MA, USA, 2010
- [10] Orukpe, P. E., "Basics of Model Predictive Control", Master degree of Science in Control Engineering, Imperial College, London, April 2005
- [11] Siouris, G. M., "Missile Guidance and Control Systems", Springer-Verlag, New York, USA, 1st edition, March 2004.
- [12] Tyagunov, A. A., "High-Performance Model Predictive Control for Process Industry", Ph.D. degree of Science in Control system, Faculty of Electrical Engineering, Eindhoven University of Technology, June 2004.
- [13] Yi Cao, J. and Gang Cao, B., "Design of Fractional Order Controller Based on Particle Swarm Optimization", International Journal of Control, Automation and Systems, Vol. 4, No. 6, pp. 775-781, December 2006.