Online Model Predictive Integrated Control and Guidance to Intercept Maneuvering Targets

Mohammad Mahdi Soori,[1] Seyed Hossein Sadati.[2]*

[1] *PhD student, Mechanical Engineering Department, K. N. Toosi University of Technology, Tehran,Iran,*
Email: mmsoorimech@yahoo.com
[2] *Associate Professor, Mechanical Engineering Department, K. N. Toosi University of Technology, Tehran,Iran,*
Email: sadati@kntu.ac.ir
**Corresponding author*

Abstract:

Integrated Guidance and Control (IGC) is a method devised in a framework in which guidance and control are considered integrated within, and unified rather than independent of each other. The advantage of IGCs is their ability to use interactions between guidance and control subsystems. This methodology is employed, intended to increase the performance of the Flying Vehicle by taking advantage of the synergy between the two processes of guidance and control. This article describes the process of designing and simulating the performance of the online model predictive controller, which was devised in order to guide the Flying Vehicle in a three-dimensional scenario to minimize the time to collision as well as the miss distance to the target. As for the controller design, an online predictive model is devised. In general, the controller model can be implemented in two ways: online (implicit) and offline (explicit). In the implementation of the online type, the optimization problem of the control cost function is solved online in each time step, and The solution to this problem will determine the optimal control signal. According to the simulations, it was shown that the use of the proposed controller and the application of the integrated guidance and control model, led to smaller values for the Flying Vehicle-target miss distance and the time to collision as compared to those from the PID and LQR controllers.

Keywords: Flying Vehicle, target, integrated guidance and control, model predictive control, optimization

1. Introduction

Guidance, navigation and control functions are critical to all forms of air and space vehicles, including missiles. In practice, these functions work together in series to maneuver a vehicle. It is now common to develop guidance completely separate from control (autopilot) and vice versa. Almost all textbooks and technical articles on this topic have dealt with it [1].

Some more advanced guidance algorithms not only achieve interception, but also control the interception angle of the missile upon impact. However, all these algorithms are rooted in the collision triangle concept, which minimizes the change of line of sight between the interceptor and the target, and may suffer from instability at the end of the task. In a multi-loop structure, steering is generated using engagement kinematics while the autopilot stabilizes the body dynamics and follows the acceleration provided by the steering.

Unlike the conventional three-loop autopilot structure, Integrated Guidance and Control (IGC) is an integrated framework in which guidance and control are considered to be integrated within rather than independent of each other.

The advantage of IGC is its ability to use interactions between command and control subsystems. IGC intends to increase the performance of the missile by taking advantage of the synergy between the guidance and control processes. Depending on the structure of the IGC, some provide additional feedback paths in the flight control system, while others require less. Putting G&C into a single IGC system improves its optimization potential. Because optimization of parameters can be done directly. Cost functions include key performance parameters such as missile and target relative speed of approach, line-of-sight angle, impact angle, and many parameters not readily available to autopilot are now directly available. In the conventional approach, the guidance law has no knowledge of the amount of spin or acceleration applied to the missile, instead, guidance only knows the relative position and speed of engagement. As the range-target decreases, small changes in geometry result in large acceleration commands that can exceed the performance range of the autopilot. In addition, the autopilot cannot adjust itself based on relative engagement kinematics, as it does not receive this information. As a result, conventional G&C systems rely on making the autopilot time constant as small as possible to improve stability. The autopilot time constant designs the distance from miss to target in conventional G&C systems [2].The integrated guidance and control (IGC) framework combines the objectives of guidance and control systems to leverage their synergy. The primary merits of this approach include reduced time to collision, improved trajectory tracking, and better stability under dynamic conditions. These merits are quantitatively evaluated in terms of minimized miss distance, shorter interception time (t), and smoother control inputs, as detailed in the results section.

One of the main approaches to IGC using SMC was presented in 2010 by Harrell and Balakrishnan using terminal second-order sliding mode control [3]. by Harrell and Balakrishnan using terminal second-order sliding mode (TSM) control. In 2019, Wang et al proposed an integrated guidance and control method with limited impact angle for the missile to achieve omnidirectional attack capability [4]. To improve the ability to damage the target, He et al. [5] designed an integrated guidance and control law with impact angle constraint to deal with the problem of tracking unknown maneuvering targets. To deal with the limitations of stimulus saturation in real systems, Ma et al in [6] investigated an integrated control law using dynamic level control, feedforward control and adaptive neural network. And Michel and Stechel thoroughly investigated the sliding mode control for the integrated plane model [7]. In 2020, Tian et al. presented a unified model to avoid practical problems such as the field of view limitation, and solved the field limitation by converting output to input saturation [8]. In 2021, Sinha et al presented an integrated guidance and control model with limited time. In this research, due to the simplicity of the design, sliding mode control is used, while a non-linear finite time disturbance observer is used to estimate the target maneuver [9]. In 2022, Lee proposed a unified model for hypersonic homing missiles. In this design, high-speed targets are hit with proper accuracy by using the sliding model controller, and by using the Monte Carlo method, the non-hit distance was reduced to the minimum [10]. In 2023 Xiaohui Liang et al investigated the nonlinear integrated missile guidance and control system with external uncertainties and disturbances and proposed a new adaptive neural network (NN) control scheme with the help of estimates obtained by NN and disturbance observer (DOB). In this paper, the weight learning rule NN and DOB are updated according to the tracking and estimation errors. Under the operation of the proposed adaptive NN rules, a good tracking

characteristic and guidance effects can be obtained for the integrated missile guidance and control (IGC) system. Finally, the simulation results of two different scenarios show the correctness of the designs. It is worth mentioning that the missile tracking process shows a smoother trajectory and a shorter distance can be achieved with the proposed NN adaptive control approach [11]. Xiangyu et al. 2024 investigates the integrated guidance and control (IGC) law design problem with impact angle and general field of view (FOV) constraints. First, the IGC model for non-maneuvering moving target tracking is parameterized by state-dependent coefficient matrices. The nominal IGC law for target interception with the desired impact angle is obtained by solving the state-dependent Riccati equation. Second, since the relative degrees of general FOV constraints exceed one according to the IGC model, high-order control barrier functions are constructed. Satisfying the FOV constraints is equivalent to ensuring that the hypersurface sets defined by the barrier functions are constant, which translate into dependent constraints on the control input. The nominal IGC law is modified in a minimally invasive way by quadratic programming. Then, the proposed method is extended to the case of maneuvering target tracking using a relative coordinate framework. Finally, numerical simulations are performed to confirm the effectiveness of the proposed method [12].
Missile guidance and control have been widely studied using various control strategies, including classical proportional navigation, optimal control techniques, and nonlinear robust control approaches. While classical methods such as PID controllers offer simplicity and ease of implementation, they often fail to provide accurate interception in complex and dynamic engagement scenarios. Linear Quadratic Regulator (LQR) controllers offer an improvement by optimizing control inputs based on a quadratic cost function; however, their performance is heavily dependent on linearization, making them less effective in handling nonlinear missile dynamics and rapid target maneuvers. Recent advances in nonlinear control techniques, such as sliding mode control (SMC) and adaptive control, have addressed some of these limitations by enhancing robustness against uncertainties. However, these methods still face challenges in achieving an optimal trade-off between guidance accuracy, computational efficiency, and real-time adaptability. Model Predictive Control (MPC) has emerged as a powerful approach for handling complex dynamic systems by solving an optimization problem at each time step while considering system constraints. Despite its advantages, the application of MPC in real-time missile guidance remains an open challenge due to the computational burden and the need for fast optimization routines. This study aims to bridge these gaps by developing an online MPC-based integrated guidance and control (IGC) framework for missile-target engagements. Unlike conventional approaches that treat guidance and control separately, the proposed method integrates both processes into a unified framework, leveraging the predictive capability of MPC to dynamically adjust control inputs in real-time. The key contributions of this work include:

1. Development of a real-time implementable online MPC framework for missile guidance and control.
2. Comparison with conventional PID and LQR controllers to highlight improvements in miss distance, time to impact, and trajectory optimization.
3. Comprehensive simulation results in a short-range air defense scenario, demonstrating the effectiveness of the proposed approach in minimizing interception time while maintaining optimal control effort.

The remainder of this paper is organized as follows: Section 2 presents the mathematical modeling of the missile-target engagement. Section 3 details the proposed control strategies, including PID, LQR, and MPC-based IGC. Section 4 discusses simulation results and a comparative performance evaluation, followed by conclusions in Section 5.

2. Mathematical modeling

A missile-target engagement scenario involves a missile trying to intercept a target by changing course. All through homing guidance, sensors onboard the missile are used to guide the missile to impact. The initial conditions of this scenario include three main assumptions. (a) Midcourse guidance is successful, (b) The speed of the missile and the target are close to each other in the collision course, (c) In order to intercept and completely destroy the target, the impact angle of the missile and the target is limited. The engagement geometry of this conflict scenario is shown in Fig. 1.

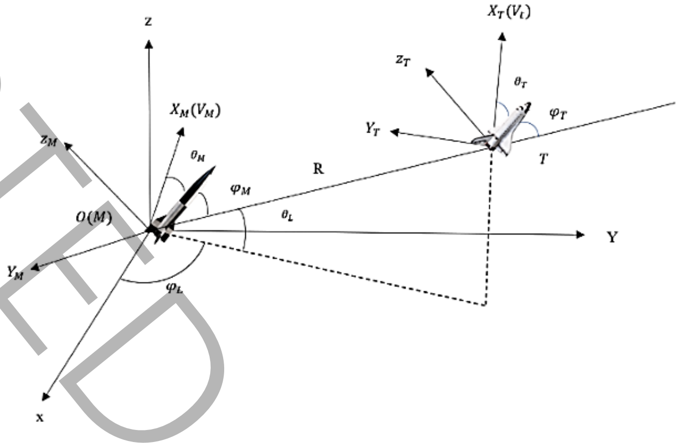

Fig. 1 - Kinematics of conflict

The general purpose of this article will be to design a proper controller for accurate target tracking. Therefore, the problem of missile-target engagement is discussed, which includes all the topics required for accurate modeling, including engagement kinematics, missile dynamics, and integrated guidance and control model. The o-xyz in Fig. 1 represents the inertial coordinate system. The coordinate system in which the missile and target velocities are described is shown in Fig. 1 as $(M - x_M y_M z_M ‘ T - x_T y_T z_T)$, and R is the relative missile/target range. In respective orders, vectors V_M and V_T are the velocities of the missile and target, $(\theta_M\ \varphi_M)$ and $(\theta_T\ \varphi_T)$ are the angles of the missile and the target relative to the velocity coordinates and line of sight coordinate system, $(\theta_L \varphi_L)$ is the elevation angle and the side angle [13]. The kinematics of the engagement is described by the equations

$$\ddot{R} - R\dot{\theta}_L^2 - R\dot{\phi}_L^2 \cos^2 \dot{\theta}_L = a_{tR} - a_{mR} \tag{1}$$

$$R\ddot{\theta}_L + 2\dot{R}\dot{\theta}_L + R\dot{\phi}_L^2 \sin \dot{\theta}_L \cos \theta_L = a_{t\theta} - a_{m\theta} \tag{2}$$

$$-R\ddot{\phi}_L^2 \cos \theta_L - 2\dot{R}\dot{\phi}_L \cos \theta_L + 2R\dot{\theta}_L\dot{\phi}_L \sin \theta_L = a_{t\phi} - a_{m\phi} \tag{3}$$

In equations 1 to 3, $(a_{mR}, a_{m\theta}\ , a_{m\phi})$ and , $(a_{tR}, a_{t\theta}\ , a_{t\phi})$ represent the acceleration components for the missile and target, respectively. Also, θ is the angle of the flight path. Since the acceleration for the missile is usually provided by the aerodynamic force, in particular in the final phase of the guidance, closer attention must be given to the relationship between the acceleration of the missile and the aerodynamic force. The aerodynamic force on the missile is calculated in the inertial coordinate system as equations 4 to 7.

$$a_{mz} = \frac{Z}{m} \tag{4}$$

$$a_{my} = \frac{Y}{m} \tag{5}$$

$$Y = qS\left(c_y^{\alpha}\alpha + c_y^{\beta}\beta + c_y^{\delta_z}\delta_z\right) \tag{6}$$

$$Z = qS\left(c_z^{\alpha}\alpha + c_z^{\beta}\beta + c_z^{\delta_y}\delta_y\right) \tag{7}$$

In equations 4 to 7, (a_{mz}, a_{my}) are the acceleration components of the missile along the inertial coordinate system, m indicates the mass of the missile, ρ is the air density, $(Y ، Z)$ indicate the upward and lateral forces, $(q = \frac{1}{2}\rho V_m^2)$ is the dynamic pressure, S represents the aerodynamic reference area of the missile, α is the angle of attack , β is the lateral slip angle, $(\delta_x \delta_y \delta_z)$ denotes the deflection angles of the missile wings, $(c_z^{\beta}\ c_z^{\alpha}\ c_z^{\delta_y})$ are partial derivatives of the lift force coefficient, and $(c_z^{\beta}\ c_z^{\alpha}\ c_z^{\delta_y})$ partial derivatives are the lateral force coefficients.

By combining equations 2, 3, 6, and 7, the equations for the line of sight angle appear as equations 8 and 9, as follows.

$$\begin{aligned}\ddot{\theta}_L = -\dot{\phi}_L^2 \sin\theta_L \cos\theta_L - \frac{2\dot{R}\dot{\theta}_L}{R} \\ - M_1 Y \cos(\gamma_V) - \frac{\sin\theta_L \sin(\phi_L - \varphi_V)}{mR}(Y\cos(\gamma_V) + Z\sin(\gamma_V)) \\ + M_1(mg\cos\theta + Z\sin(\gamma_V)) + d_{\theta L}\end{aligned} \tag{8}$$

$$\begin{aligned}\ddot{\phi_L} = -\frac{2\dot{R}\dot{\phi}_L}{R} + 2\dot{\theta}_L\dot{\phi}_L \tan\theta_L - M_2 Y\cos(\gamma_V) + \frac{\cos(\phi_L - \varphi_V)}{mR\cos\theta_L}(Y\cos(\gamma_V) + Z\sin(\gamma_V)) \\ + M_2(mg\cos\theta + Z\sin(\gamma_V)) + d_{\phi L}\end{aligned} \tag{9}$$

In equations 8 and 9, parameters M_1 and M_2 are defined as relations 10 and 11.

$$M_1 = \frac{\cos\theta\cos\theta_L + \sin\theta\sin\theta_L\cos(\phi_L - \varphi_V)}{mR} \tag{10}$$

$$M_2 = \frac{\sin\theta\cos(\phi_L - \varphi_V)}{mR\cos\theta_L} \tag{11}$$

In equations 8 to 11, φ_V is the ballistic angle, γ_V the rotation angle, $(d_{\theta L}\ d_{\phi L})$ are the approximate error for the θ_L and ϕ_L. Kinematic equations of flight path angle and ballistic angle are shown as equations 12 and 13, as follows.

$$\dot{\theta} = \frac{Y\cos\gamma_V - Z\sin\gamma_V - mg\cos\theta}{mV_m} \tag{12}$$

$$\dot{\varphi_V} = \frac{-Y\sin\gamma_V - Z\cos\gamma_V}{mV_m\cos\theta} \tag{13}$$

The kinematic and dynamic equations of the missile rotating around the center of mass in three-dimensional space are given as equations 14 through 16 as follows.

$$\dot{\alpha} = -\omega_x \tan\beta\ \cos\alpha + \omega_y\ \tan\beta\sin\alpha + \omega_z - \frac{Y}{mV_m\cos\theta} + \frac{g\ \cos\theta\ \cos\gamma_V}{V_m\cos\theta} \tag{14}$$

$$\dot{\beta} = \omega_x\sin\alpha + \omega_y\cos\alpha + \frac{Z + mg\cos\theta\ \sin\gamma_V}{mV_m} \tag{15}$$

$$\dot{\gamma_V} = \cos\alpha\,\sec\beta\;\omega_x - \sin\alpha\;\sec\beta\;\omega_y + \frac{Y(\tan\theta\;\sin\gamma_V + \tan\beta) + Z\,\tan\theta\,\cos\gamma_V}{mV_m} - \frac{\cos\theta\;\cos\gamma_V\,\tan\beta}{V_m}g \tag{16}$$

In equations 14 to 16, parameters ω_x , ω_y , and ω_z are the angular velocities in the roll, yaw, and pitch channels, respectively.
The dynamic equations of the state of the missile are obtained as equation 17.

$$\begin{cases} \dot{\omega_x} = \dfrac{J_y - J_z}{J_x}\,\omega_z\omega_y + \dfrac{M_x}{J_x} \\ \dot{\omega_y} = \dfrac{J_z - J_x}{J_y}\,\omega_x\omega_z + \dfrac{M_y}{J_y} \\ \dot{\omega_z} = \dfrac{J_x - J_y}{J_z}\,\omega_y\omega_x + \dfrac{M_z}{J_z} \end{cases} \tag{17}$$

In equation 17, $\left(J_x\,,\,J_y\,,\,J_z\right)$ denote the moments of inertia for the roll, yaw, and pitch channels, respectively. Also, the three moments $(M_x\,,\,,M_y\,,\;M_z\,)$ are defined as in equation 18.

$$\begin{cases} M_x = qSL\left(m_x^{\alpha}\,\alpha + m_x^{\beta}\,\beta + m_x^{\delta_x}\delta_x\right) \\ M_y = qSL\left(m_y^{\beta}\,\beta + m_y^{\delta_y}\delta_y\right) \\ M_z = qSL\left(m_z^{\alpha}\,\alpha + m_z^{\delta_z}\delta_z\right) \end{cases} \tag{18}$$

In equation 18, L is the reference length, $(m_x^{\alpha}\,,m_x^{\beta}\;m_x^{\delta_x})$ are the moment coefficient of the partial derivatives related to the roll channel, $(m_y^{\beta}, m_y^{\delta_y})$ are those related to the yaw channel, and $(m_z^{\alpha}\,,\,m_z^{\delta_z})$ are those related to the pitch channel.

By combining relations 8, 9, and 14 with 18, the three-dimensional equations of the integrated missile guidance and control can be written as equation 19 [13].

$$\begin{cases} \dot{x}_0 = x_1 \\ \dot{x}_1 = f_1 + b_1\overline{x_2} + d_1 \\ \dot{x}_2 = f_2 + b_2x_3 + d_2 \\ \dot{x}_3 = f_3 + b_3u + d_3 \end{cases} \tag{19}$$

, where

$$x_0 = \begin{bmatrix} \theta_L - \theta_{Lf} \\ \phi_L - \phi_{Lf} \end{bmatrix},\quad x_1 = \begin{bmatrix} \dot{\theta}_L \\ \dot{\phi}_L \end{bmatrix}\;,\quad \overline{x_2} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}\;,\quad x_2 = \begin{bmatrix} \alpha \\ \beta \\ \gamma_v \end{bmatrix}\text{‘}\quad x_3 = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix},\quad u = \begin{bmatrix} \delta_x \\ \delta_y \\ \delta_z \end{bmatrix}$$

and the $d_i's$, $i = 1,2,3$ show the approximate system errors, which are deviations in elevation, lateral, and range dimensions, respective. Moreover, the matrices b_i and f_i are shown in relations (20) to (25).

$$b_1 = \begin{bmatrix} -M_1 qSc_y^{\alpha} & -\dfrac{qSc_z^{\beta}\, sin\,\theta_L\, sin(\phi_L - \phi_V)}{mR} \\ -M_2 qSc_y^{\alpha} & \dfrac{qSc_z^{\beta}\, cos(\phi_L - \phi_V)}{mR\, cos\,\theta_L} \end{bmatrix} \tag{20}$$

$$b_2 = \begin{bmatrix} -\,tan\,\beta\, cos\,\alpha & tan\,\beta\, sin\,\alpha & 1 \\ sin\,\alpha & cos\,\alpha & 0 \\ cos\,\alpha\, sec\,\beta & -\,sin\,\alpha\, sec\,\beta & 0 \end{bmatrix} \tag{21}$$

$$b_3 = \begin{bmatrix} \dfrac{qSLm_x^{\delta_x}}{J_x} & 0 & 0 \\ 0 & \dfrac{qSLm_y^{\delta_y}}{J_y} & 0 \\ 0 & 0 & \dfrac{qSLm_z^{\delta_z}}{J_z} \end{bmatrix} \tag{22}$$

$$f_1 = \begin{bmatrix} -\dfrac{2\dot{R}}{R}\,\dot{\theta}_L - \dot{\varphi}_L^2\, sin\,\theta_L\, cos\,\theta_L + M_1 mg\, cos\theta \\ -\dfrac{2\dot{R}}{R}\,\dot{\phi}_L + 2\dot{\theta}_L\dot{\phi}_L\, tan\,\theta_L + M_2 mg\, cos\theta \end{bmatrix} \tag{23}$$

$$f_2 = \begin{bmatrix} -\dfrac{qSY_y}{mV\, cos\,\beta} + \dfrac{g}{V\, cos\,\beta}\, cos\,\gamma \\ -\dfrac{qSZ_z}{mV} + \dfrac{g}{V}\, cos\,\theta\, sin\,\gamma \\ -\dfrac{F_{YZ}}{mV} - \dfrac{g}{V}\, cos\,\theta\;\; cos\,\gamma\;\; tan\,\beta \end{bmatrix} \tag{24}$$

$$f_3 = \begin{bmatrix} \dfrac{J_y - J_z}{J_x}\,\omega_z\omega_y + \dfrac{qSL\left(m_x^{\alpha}\,\alpha + m_x^{\beta}\,\beta\right)}{J_x} \\ \dfrac{J_z - J_x}{J_y}\,\omega_x\omega_z + \dfrac{qSLm_x^{\beta}\,\beta}{J_y} \\ \dfrac{J_x - J_y}{J_z}\,\omega_y\omega_x + \dfrac{qSLm_z^{\alpha}\,\alpha}{J_z} \end{bmatrix} \tag{25}$$

3 - Controller design

The missile hits the target when R (the distance between the missile and the target) approaches R_{hit} (the minimum distance between the missile and the target). The goal of the controller design in this article is to the make the angles θ_L and ϕ_L converge to their reference values θ_{Lf} and ϕ_{Lf}, respectively, in addition to making the distance between the missile and the target approach R_{hit}.

Backstepping method is used to solve the problem. The control input (u) will be calculated using the Higher order continuous sliding mode control. In the backstepping approach, a virtual control signal, denoted as x_{2d}, is first designed to regulate the system dynamics. This signal is the desired behavior of the state variable x_2 and its design is such that the state variable x_0 tends to zero. So, in fact, the desired behavior of x_2 is obtained in such a way that the first goal of the control problem is established. In the next step, we design the virtual control signal x_{3d}. This signal is the reference signal for the state variable x_3, and it is designed in such a way that if x_3 follows it, x_2 also follows x_{2d} and as a result x_0 will tend to zero. In the last stage, the real control input u is designed in such a way as to make x_3 converge to x_{3d}, and as a result, the control loop is completed. By doing this, the second goal of the control problem is to force the miss distance assume as small as possible. In effect, the problem here is that of a minimization type, aiming at reducing the distance R to its minimum value, R_{hit}. This distance R is directly related to the errors d_1, d_2, d_3 defined in Eq. (19).

Two virtual control signals are obtained by the backsteping method, but the real control signal, which is the main goal of this article, will be calculated from the Higher order continuous sliding mode control. To calculate virtual signals, we will have relations (26) and (27) [13].

$$S_1 = x_0 + cx_1 \tag{26}$$

$$x_{2d} = -(cb_1)^{-1}(x_1 + cf_1 + K_1 \tanh(S_1))$$

$$S_2 = x_2 - x_{2d} \tag{27}$$

$$x_{3d} = -(b_2)^{-1}(f_2 + K_2 \tanh(S_2))$$

In equations (26) and (27), the coefficients c, K_1 , and K_2 are fixed and will be obtained with the help of optimization methods of PSO along with the variables related to the real controller. Also, the controller block diagram of the article is drawn in Fig. 2

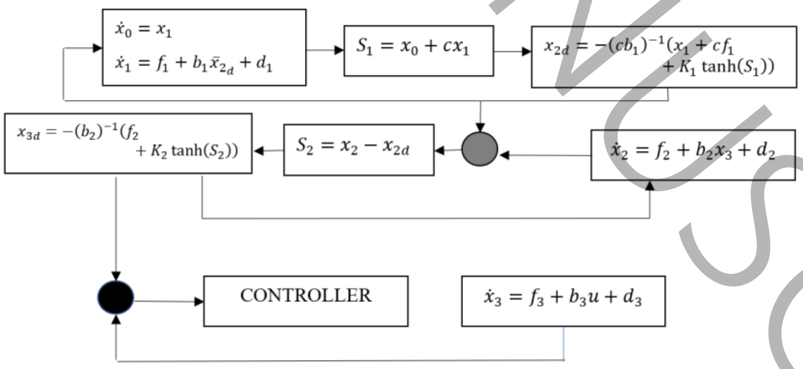
$\dot{x}_0 = x_1$
$\dot{x}_1 = f_1 + b_1\bar{x}_{2_d} + d_1$
$S_1 = x_0 + cx_1$
$x_{2d} = -(cb_1)^{-1}(x_1 + cf_1 + K_1 \tanh(S_1))$
$x_{3d} = -(b_2)^{-1}(f_2 + K_2 \tanh(S_2))$
$S_2 = x_2 - x_{2d}$
$\dot{x}_2 = f_2 + b_2x_3 + d_2$
CONTROLLER
$\dot{x}_3 = f_3 + b_3u + d_3$

Fig. 2 - Controller block diagram

3-1 PID controller design

PID controller is introduced as a standard control structure in classical control theory. The performance of the system is improved by accurately adjusting the values of the three proportional gains K_p, integral gain

T_i , and derivative gain T_d. By adjusting these parameters, the steady state error and output fluctuations can be controlled in response to the step input. In order to evaluate the proposed controller of this article, first a simple PID controller is designed. According to the system equations, the output of the missile system is its angular velocities. These speeds are to tend to zero. As a result, tracking error is defined as equation 28.

$$e = 0 - [\omega_x, \omega_y, \omega_z]^T = -[\omega_x, \omega_y, \omega_z]^T \tag{28}$$

The general form of the PID controller is of the form given in Eq. (29) as:

$$u = K_p e + T_i \int e\, dt + T_d \left(\frac{de}{dt}\right) \tag{29}$$

Ziegler Nichols method is used to obtain the K_p, T_i , and T_d gains in relation (29). For this purpose, the system should be linearized first. By a frequency analysis then, the PID controller gains are obtained from Eq. (30) [14].

$$\begin{aligned} K_p &= 0.6\, K_{cr} \\ T_d &= 0.125\, P_{cr} \\ T_i &= 0.5\, P_{cr} \end{aligned} \tag{30}$$

The values of K_{cr} and P_{cr} are calculated from Eq. (31).

$$K_{cr} = Gm \qquad P_{cr} = \frac{2\pi}{w_{cg}} \tag{31}$$

In relation (31), the parameter Gm is the phase margin and w_{cg} is the frequency where the phase margin is measured and the system phase will be -180.

3-2 LQR controller design

The 2nd-order linear regulator or LQR controller attempts to minimize the objective function (32).

$$J = \int_0^{t_f} \frac{1}{2}[x^T Q(t)x(t) + u^T R(t)u(t)]dt \tag{32}$$

In Eq. (32), the parameter t_f denotes the final time. Also, Q and R are two weight matrices that are obtained through optimization. The equation for the control input associated with this objective function is obtained as given in Eq. (33) [14].

$$u(t) = -R^{-1}B^T K(t)x(t) \tag{33}$$

where B is the matrix multiplier for the control input in the linearized system, and $K(t)$ is obtained from Eq. (34) below.

$$0 = \dot{K}(t) + Q - K(t)BR^{-1}B^T K(t) + K(t)A + A^T K(t) \tag{34}$$

, where A is the matrix multiplier for the state vector in the linearized system. f

3-3 Design of predictive model controller

In most control projects where the use of predictive model control is considered a necessity, the linear type predictive model controller option is preferred. In particular, in industrial projects, due to the need for high speed and reliability along with cost considerations, this type of controller has been almost the only option for designers [15]. In general, the pre-linear model controller can be implemented in two ways: online (implicit) and offline (explicit). In the implementation of the online type, the optimization problem of the control cost function is solved online in each time step, and the solution to this problem will determine the optimal control order. But in the offline method, the optimization problem is solved once for all possible values of the state vector and the optimal solution is calculated as an explicit function of the variables of the state vector and loaded into the controller memory. At each time step then, the controller will determine the value of this function based on the values of the state variables to issue the control commands.
As such, the time required to perform control calculations is reduced, hence making it possible to implement the devised controller on the control hardware with limited processing volume.

3-3-1 Designing an online predictive model controller

We start the design of the predictive model controller by linearizing the open-loop equations of the system. Although it is possible to obtain linearized state equations by using Jacobians of the nonlinear equations, the method of trimming nonlinear equations is more appropriate for problems with high degree of nonlinearity due to the availability of the necessary tools in MATLAB and for better approximation. The difference in the results of these two different linearization methods will be seen in the comparison of the controller performance in the implied linear forecast model and the nonlinear forecast model based on the Matlab f-mincon solver. Assuming that the disturbances (target acceleration) are zero, the standard form of the linearized state equations of the open loop system is expressed as follows [15].

$$\dot{x}(t) = A_c x(t) + B_c u(t) \tag{35}$$

The output equation of the system is expressed as follows.

$$y(t) = C_c x(t) \tag{36}$$

The numerical values of the above matrices are available in the relevant MATLAB file, and their mention has been avoided in order to avoid confusion.
The predictive model controller is usually implemented as discrete time in control systems. Therefore, we express the state equations in the form of discrete-time state space equations using the zeroth-order retainer and the sampling time T_s. Hence, the discrete-time equations appear as:

$$x(k+1) = A_d x(k) + B_d u(k) \tag{37}$$

$$y(k) = C_d x(k) \tag{38}$$

Relations (37) and (38) express the model based on which the predictive model controller is developed. Therefore, this model is called control-oriented model. In order to consider the constraint related to the rate of acceleration changes in the control equations, and also due to the effect of integration on the controller to reduce the steady state error, the control input at the present time step will be a combination of the control input from the previous step and the increase in the input, i.e., $u(k) = u(k-1) + \Delta u(k)$, and the control input $u(k-1)$ is augmented to the state vector as indicated in Eq. (39).

$$x_a(k+1)=A_a x_a(k)+B_a \text{�} u(k)$$
$$\begin{bmatrix}x(k+1)\\u(k)\end{bmatrix}=\begin{bmatrix}A_d & B_d\\0 & I\end{bmatrix}\begin{bmatrix}x(k)\\u(k-1)\end{bmatrix}+\begin{bmatrix}B_d\\I\end{bmatrix}\text{�} u(k) \tag{39}$$

This new control-oriented model is called the augmented model. The output equations of the augmented model can be expressed as follows.

$$y(k)=C_a x_a(k)\ y(k)=\begin{bmatrix}C_d & 0\end{bmatrix}\begin{bmatrix}x(k)\\u(k-1)\end{bmatrix} \tag{40}$$

By defining N_p and N_c as prediction and control horizons respectively, the input and output equations of the prediction model can be calculated using the following equation.

$$Y=F\begin{bmatrix}x(k)\\u(k-1)\end{bmatrix}+\phi\Delta U \tag{41}$$

In the equation of the above forecasting model, the Y vector can be taken to be of the form given in Eq. (42) as:

$$Y=\begin{bmatrix}y(k+1|k)\\y(k+2|k)\\\vdots\\y(k+N_p|k)\end{bmatrix} \tag{42}$$

the ΔU vector taken as shown in Eq. (43)

$$\Delta U=\begin{bmatrix}\Delta u(k|k)\\\Delta u(k+1|k)\\\vdots\\\Delta u(k+N_c-1|k)\end{bmatrix}=\begin{bmatrix}\Delta a_h(k|k)\\\Delta a_h(k+1|k)\\\vdots\\\Delta a_h(k+N_c-1|k)\end{bmatrix} \tag{43}$$

and matrix A taken from Eq. (44) as:

$$F=\begin{bmatrix}C_aA_a\\C_aA_a^2\\\vdots\\C_aA_a^{N_p}\end{bmatrix} \tag{44}$$

and the matrix ϕ is obtained from Eq. (45) as:

$$\phi=\begin{bmatrix}C_aB_a & \cdots & 0\\C_aA_aB_a & \cdots & 0\\\vdots & \ddots & \vdots\\C_aA_a^{N_p}B_a & \cdots & C_aA_a^{N_p-N_c}B_a\end{bmatrix} \tag{45}$$

3-3-2 Definition of the constrained optimization problem of predictive model control

The method of predictive model control strategy is based on the use of a function of inputs and outputs (or state variables) called the performance function, which should be optimized in the prediction horizon while satisfying the constraints of the problem. The state variables in the future time are calculated using the predictive model and the system conditions in the present time ($x_a(k|k) = x_a(k)$) as the initial conditions. By solving the stated optimization problem, the sequence of control inputs is obtained as $\Delta U = [\Delta a_h(k|k), \dots, \Delta a_h(k + N_{c-1}|k)]^T$. Then, based on the descending horizon control rule, the first input of this sequence is applied to the control system and the same process is repeated again in the next time steps. The cost function in the predictive model control is usually chosen as a linear function or a quadratic function. Although solving the first-order equation requires less computational effort than solving the second-order equation, the existence of the overall minimum value is not guaranteed in this method. Therefore, the cost function is chosen in the form of a quadratic function, so that owing to the convexity of this function, the existence of a local minimum point is equivalent to the existence of a global minimum point (a necessary and sufficient condition for optimality). The cost function for the pursuit problem is defined as follows [15]:

$$J = \sum_{i=1}^{N_p} [y(k+i|k) - r(k+i|k)]^T Q [y(k+i|k) - r(k+i|k)] + \sum_{j=0}^{N_c-1} \Delta u^T(k+j|k) R \Delta u(k+j|k) \tag{46}$$

In the cost function of the investigated problem, Q is a positive definite symmetric matrix and R is a positive scalar. Also, $r(k+i|k) = [x_{r,d}(k+i|k) \quad v_{r,d}(k+i|k) \quad v_{h,d}(k+i|k)]^T$. The reference vector is pertinent to the prediction window and expresses the controller's expectation of the behavior of the reference vector in future times. The values of elements of this vector in reference to the prediction window are considered constant and equal to the values of the reference signal at the sampling moment k. At times it is possible to predict the changes of the controller's reference signals in advance, which is called "foreseeing" the reference signal in the control of the predictive model. Anticipating the reference signal actually means a more accurate definition of control expectations in the future, and it significantly improves system performance, especially in plants with fast dynamics.

The matrix form of the cost function can be expressed as follows [15].

$$\begin{aligned} J = {} & \underbrace{\begin{bmatrix} y(k+1|k) - r(k+1|k) \\ y(k+2|k) - r(k+2|k) \\ \vdots \\ y(k+N_p|k) - r(k+N_p|k) \end{bmatrix}^T}_{Y^T - Ref^T} \underbrace{\begin{bmatrix} Q & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & Q \end{bmatrix}}_{\hat{Q}} \underbrace{\begin{bmatrix} y(k+1|k) - r(k+1) \\ y(k+2|k) - r(k+2) \\ \vdots \\ y(k+N_p|k) - r(k+N_p) \end{bmatrix}}_{Y - Ref} \\ & + \underbrace{\begin{bmatrix} \Delta u(k|k) \\ \Delta u(k+1|k) \\ \vdots \\ \Delta u(k+N_c-1|k) \end{bmatrix}^T}_{\Delta U^T} \underbrace{\begin{bmatrix} R & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & R \end{bmatrix}}_{\hat{R}} \underbrace{\begin{bmatrix} \Delta u(k|k) \\ \Delta u(k+1|k) \\ \vdots \\ \Delta u(k+N_c-1|k) \end{bmatrix}}_{\Delta U} \\ & = (Y - Ref)^T \hat{Q} (Y - Ref) + \Delta U^T \hat{R} \Delta U \end{aligned} \tag{47}$$

$$J=\underbrace{\begin{bmatrix} y\left(k+1|k\right)-r(k+1\,|\,k) \\ y(k+2\,|\,k)-r(k+2\,|\,k) \\ \vdots \\ y\left(k+N_p|k\right)-r(k+N_p\,|\,k) \end{bmatrix}^T}_{Y^T-Ref^T}\underbrace{\begin{bmatrix} Q & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & Q \end{bmatrix}}_{\hat{Q}}\underbrace{\begin{bmatrix} y\left(k+1|k\right)-r\left(k+1\right) \\ y(k+2\,|\,k)-r\left(k+2\right) \\ \vdots \\ y\left(k+N_p|k\right)-r\left(k+N_p\right) \end{bmatrix}}_{Y-Ref}$$

The $\hat{Q}$and $\hat{R}$ are positive definite matrices. Substitution of Eq. (41) in Eq. (47) results in:

$$\begin{aligned} J &= \frac{1}{2}\Delta U^T\underbrace{2[\hat{R}+\phi^T\hat{Q}\phi]}_{\hat{H}}\Delta U+\underbrace{2\left(\begin{bmatrix} x(k) \\ u(k-1) \end{bmatrix}^T F^T-Ref^T\right)\hat{Q}\phi\,\Delta U}_{\widehat{M}} \\ &\quad \underbrace{+\,Ref^T\hat{Q}Ref+\begin{bmatrix} x(k) \\ u(k-1) \end{bmatrix}^T F^T\hat{Q}\left(F\begin{bmatrix} x(k) \\ u(k-1) \end{bmatrix}-2Ref\right)}_{\widehat{N}} \\ &= \frac{1}{2}\Delta U^T\hat{H}\Delta U+\widehat{M}\Delta U+\widehat{N} \end{aligned} \tag{48}$$

The expression $\widehat{H}=2\left[\hat{R}+\phi^T\hat{Q}\phi\right]\in\mathbb{R}^{N_c\times N_c}$ in the above equation is a symmetric positive definite matrix and the column vector $\widehat{M}=2([x^T(k)\quad u^T(k-1)]F^T-Ref^T)\hat{Q}\phi\in\mathbb{R}^{N_c}$, and the scaler $N=Ref^T\hat{Q}Ref+[x^T(k)\quad u^T(k-1)]F^T\hat{Q}(F[x^T(k)\quad u^T(k-1)]^T-2Ref)\in\mathbb{R}$ varies with time.

To implement the problem constraints, the control input vector is rewritten in the form:

$$\begin{aligned} \underbrace{\begin{bmatrix} u(k|k) \\ u(k+1|k) \\ \vdots \\ u(k+N_c-1|k) \end{bmatrix}}_{U} &= \underbrace{\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}}_{G_{1,\Delta U}} u(k-1)+\underbrace{\begin{bmatrix} 1 & 0 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 1 & & \ldots & 1 \end{bmatrix}}_{G_{2,\Delta U}}\underbrace{\begin{bmatrix} \Delta u(k|k) \\ \Delta u(k+1|k) \\ \vdots \\ \Delta u(k+N_c-1|k) \end{bmatrix}}_{\Delta U} \\ &= G_{1,\Delta U}u(k-1)+G_{2,\Delta U}\Delta U \end{aligned} \tag{49}$$

With the help of Eq. (49), the problem constrains for the duration of the prediction window can be obtained as described below.

$$
\begin{aligned}
&U^{min} \leq G_{1,\Delta U}u(k-1) + G_{2,\Delta U}\Delta U \leq U^{max}\\
&\begin{cases} -G_{2,\Delta U}\Delta U \leq G_{1,\Delta U}u(k-1) - U^{min} \\ G_{2,\Delta U}\Delta U \leq -G_{1,\Delta U}u(k-1) + U^{max} \end{cases}
\end{aligned}
\tag{50}
$$

The constraints on the changes in the control input vector is given as:

$$
\begin{aligned}
&\Delta U^{min} = [1 \quad \cdots \quad 1]^T j_{h,min}\\
&\Delta U^{max} = [1 \quad \cdots \quad 1]^T j_{h,max}\\
&\Delta U^{min} \leq \Delta U \leq \Delta U^{max}\\
&\begin{cases} -\Delta U \leq -\Delta U^{min} \\ \Delta U \leq \Delta U^{max} \end{cases}
\end{aligned}
\tag{51}
$$

and the constraints on the output vector are stated as:

$$
\begin{aligned}
&\mathrm{Y}^{min} = \begin{bmatrix} 1 & 1 & 1 \\ \vdots & \vdots & \vdots \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_{r,min} \\ v_{r,min} \\ v_{h,min} \end{bmatrix}\\
&\mathrm{Y}^{max} = \begin{bmatrix} 1 & 1 & 1 \\ \vdots & \vdots & \vdots \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_{r,max} \\ v_{r,max} \\ v_{h,max} \end{bmatrix}\\
&\begin{cases} -\phi\Delta U \leq F\begin{bmatrix} x(k) \\ u(k-1) \end{bmatrix} - \mathrm{Y}^{min} \\ \phi\Delta U \leq -F\begin{bmatrix} x(k) \\ u(k-1) \end{bmatrix} + \mathrm{Y}^{max} \end{cases}
\end{aligned}
\tag{52}
$$

It should be noted that the increase in value of the input signal after the control horizon is considered zero and hence, the control input value remains constant. In summary, the system constraints based on the changes in the control input for the duration of the prediction window can be expressed as follows.

$$
\begin{bmatrix} G_U \\ G_{\Delta U} \\ G_Y \end{bmatrix} \Delta U \leq \begin{bmatrix} W_U \\ W_{\Delta U} \\ W_Y \end{bmatrix}
\tag{53}
$$

In the above inequality, the values are expressed as matrix blocks. Each of the blocks can be calculated according to the following relations.

$$
G_U = \begin{bmatrix} -G_{2,\Delta U} \\ G_{2,\Delta U} \end{bmatrix}
$$

(54)

$$
G_{\Delta U} = \begin{bmatrix} -I \\ I \end{bmatrix}
$$

$$G_Y = \begin{bmatrix} -\phi \\ \phi \end{bmatrix}$$

$$W_U = \begin{bmatrix} G_{1,\Delta U} u\left(k-1\right) - U^{min} \\ -G_{1,\Delta U} u\left(k-1\right) + U^{max} \end{bmatrix}$$

$$W_{\Delta U} = \begin{bmatrix} -\Delta U^{min} \\ \Delta U^{max} \end{bmatrix}$$

$$W_Y = \begin{bmatrix} F\begin{bmatrix} x\left(k\right) \\ u\left(k-1\right) \end{bmatrix} - \mathrm{Y}^{min} \\ -F\begin{bmatrix} x\left(k\right) \\ u\left(k-1\right) \end{bmatrix} + \mathrm{Y}^{max} \end{bmatrix}$$

By defining $G = [G_U^T \quad G_{\Delta U}^T \quad G_Y^T]^T$ and $W = [W_U^T \quad W_{\Delta U}^T \quad W_Y^T]^T$, the optimization problem of the online predictive model control for adaptive speed control system is finally expressed as shown in Eq. (55) below.

$$\min_{\Delta U}\left(J = \frac{1}{2}\Delta U^T \hat{H} \Delta U + \hat{M}\Delta U + \hat{N} \right) \tag{55}$$

$$s.t. \;\; G\Delta U \leq W$$

3-3-3 Solving the constrained optimization problem

Due to the fact that the cost function in problem (55) is quadratic convex and the constraints of the problem are all affine, the optimization problem is quadratic programming. Considering that the term N ˆhas no role in the calculation of the gradient of the cost function, we can ignore this term in further calculations. The Lagrangian of the cost function and the constraints of the optimization problem are defined as follows [15].

$$\mathcal{L}\left(\Delta U, \lambda\right) = \frac{1}{2}\Delta U^T \hat{H}\Delta U + \hat{M}\Delta U + \lambda^T\left(G\Delta U - W\right) \tag{56}$$

In the above relation, λ is the vector of Lagrangian coefficients. The dual of the original quadratic programming problem is defined as follows.

$$d\left(\lambda\right) = \min_{\Delta U} \mathcal{L}\left(\Delta U, \lambda\right) = \min_{\Delta U}\left\{ \frac{1}{2}\Delta U^T \hat{H}\Delta U + \hat{M}\Delta U + \lambda^T\left(G\Delta U - W\right) \right\} \tag{57}$$

Therefore, the dual problem is expressed as follows.

$$\max_{\substack{\lambda \\ \lambda_i \geq 0}}\left(d\left(\lambda\right)\right) = \max_{\substack{\lambda \\ \lambda_i \geq 0}}\left\{ \min_{\Delta U}\left\{ \frac{1}{2}\Delta U^T \hat{H}\Delta U + \hat{M}\Delta U + \lambda^T\left(G\Delta U - W\right) \right\} \right\} \tag{58}$$

For any arbitrary λ, the above Lagrangian function is convex. Therefore, the necessary and sufficient condition for optimality is that the value of the gradient of the Lagrangian function at the minimum point is zero.

$$\nabla_{\Delta U}\mathcal{L}\left(\Delta \mathrm{U},\lambda\right)=\hat{H}^{T}\Delta \mathrm{U}+\hat{M}^{T}+G^{T}\lambda=0 \tag{59}$$

Considering that $\widehat{H}=\widehat{H}^{T}\succ 0$, the matrix $\widehat{H}=\widehat{H}^{T}\succ 0$ is invertible. Therefore, we can write:

$$\Delta \mathrm{U}=-\hat{H}^{-1}\left(\hat{M}^{T}+G^{T}\lambda\right) \tag{60}$$

By placing this value in the dual problem, we have:

$$\max_{\substack{\lambda \\ \lambda_i\geq 0}}\left(d\left(\lambda\right)\right)=\max_{\substack{\lambda \\ \lambda_i\geq 0}}\left\{-\frac{1}{2}\lambda^{T}\left(G\hat{H}^{-1}G^{T}\right)\lambda-\lambda^{T}\left(W+G\hat{H}^{-1}\hat{M}^{T}\right)-\frac{1}{2}\hat{M}\hat{H}^{-1}\hat{M}^{T}\right\} \tag{61}$$

To calculate the vector of optimal Lagrange coefficients, we calculate the gradient of the above relationship and obtain its roots.

$$\nabla d\left(\lambda\right)=G\hat{H}^{-1}G^{T}\lambda+G\hat{H}^{-1}M^{T}+W=0 \tag{62}$$

If we don't have any redundant constraints in the main problem, the matrix G will be of full order row, and considering that the matrix $\widehat{H}$ is of full order, we can calculate the vector of optimal Lagrange coefficients as follows.

$$\lambda^{*}=-\left(G\hat{H}^{-1}G^{T}\right)_{left}^{-1}\left(G\hat{H}^{-1}M^{T}+W\right) \tag{63}$$

By inserting the value of λ^* in (61), the vector of the string of optimal control changes during the control window is obtained as:

$$\Delta \mathrm{U}^{*}=-\hat{H}^{-1}\left[\hat{M}^{T}-G^{T}\left(G\hat{H}^{-1}G^{T}\right)_{left}^{-1}\left(G\hat{H}^{-1}M^{T}+W\right)\right] \tag{64}$$

It should be noted that in the definition of the matrix M, the information of the generalized state vector $x_a(t)$ and the reference signal vector are included. The W matrix also depends on the generalized state vector. The values of these matrices are changed in each time step according to the changes of state vectors and reference signals, and create a new vector of optimal changes of control commands during the control horizon. In this formulation, it is assumed that the control commands will remain unchanged in the interval between the control horizon and the prediction horizon, and therefore the value of the control input is considered constant in this time interval.

In the descent horizon control strategy, the controller only applies the first array of the optimal control command vector in the control horizon to the system at each sampling time. Therefore, the control input for sampling time k will be equal to:

$$u^{*}\left(k\right)=\Delta u^{*}\left(k|k\right)+u\left(k-1\right)=\begin{bmatrix}1 & 0 & \cdots & 0\end{bmatrix}\Delta U^{*}+u\left(k-1\right) \tag{65}$$

3-3-3-1 KKT conditions for optimality

Considering that the optimization problem is of the second-order programming type, the first-order KKT conditions determine the necessary and sufficient conditions for the existence of the overall optimal solution. These conditions are used to solve the problem numerically.

Necessary condition for optimality:

$$\nabla\mathcal{L}\left(\Delta U^{*},\lambda^{*}\right)=\hat{H}^{T}\Delta \mathrm{U}^{*}+\hat{M}^{T}+G^{T}\lambda^{*}=0 \tag{66}$$

Complementary redundant condition:

$$\lambda_{i}\left(G_{i}\Delta U-W_{i}\right)=0 \tag{67}$$

Feasibility condition of the dual problem:

$$\lambda_{i}\geq 0 \tag{68}$$

Feasibility condition of the primal problem:

$$G\Delta U-W\leq 0 \tag{69}$$

3-3-3-2 Numerical solution of the problem of quadratic programming of linear predictive model control

Algorithms for solving optimal problems of the quadratic programming type usually calculate the optimal point of the objective function by checking the optimality conditions of KKT and minimizing the difference between the value of the initial problem and the dual problem through repetition. In this section, the method of this calculation is briefly explained.

Obviously, each vector $\Delta\bar{U}$ in the set of possible solutions is an upper bound for the optimal value of the cost function $\mathrm{J}(\Delta\mathrm{U})$ and this implies $\mathrm{J}(\Delta\bar{U})\geq \mathrm{J}^{*}=J(\Delta U^{*})$. Now, we have to find a lower bound for this problem. Considering that the Lagrangian of the primal problem assumes that all Lagrangian coefficients are non-negative and the constraints of the problem are expressed in the form of an inequality, the inequality $\lambda^{T}(G\Delta U-W)\leq 0$ is always maintained and therefore we have:

$$\mathcal{L}\left(\Delta U,\lambda\right)\leq \mathrm{J}(\Delta\mathrm{U}) \tag{70}$$

By minimizing the sides, we will have this inequality:

$$\begin{gathered}\min_{\Delta\mathrm{U}}\mathcal{L}\left(\Delta U,\lambda\right)\leq\min_{\Delta\mathrm{U}}\mathrm{J}\left(\Delta\mathrm{U}\right)\\ s.t.\quad G\Delta U-W\leq 0\end{gathered} \tag{71}$$

The above relation gives $\min\limits_{\Delta\mathrm{U}}\mathcal{L}(\Delta U,\lambda)$ as a lower bound for the primal problem. The best lower bound is obtained by maximizing the profit function $\min\limits_{\Delta\mathrm{U}}\mathcal{L}(\Delta U,\lambda)$ through the variables of the dual problem.

$$\begin{gathered}\max_{\substack{\lambda\\ \lambda_{i}\geq 0}}\left\{\min_{\Delta\mathrm{U}}\mathcal{L}\left(\Delta U,\lambda\right)\right\}\leq\min_{\Delta\mathrm{U}}\mathrm{J}\left(\Delta\mathrm{U}\right)\\ s.t.\quad G\Delta U-W\leq 0\end{gathered} \tag{72}$$

By defining the new cost function as $\mathrm{d}(\lambda)=\min\limits_{\Delta\mathrm{U}}\mathcal{L}(\Delta U,\lambda)$, the dual problem is defined as follows.

$$\max_{\substack{\lambda \\ \lambda_i \ge 0}} \; d(\lambda) \tag{73}$$

The latter problem is unconstrained and its optimal point is λ^*. Solving the dual problem always provides the largest lower bound of the primal problem, which is called the weak duality property. Let $f^*$and d^* be the optimal values of primal and dual problem, respectively. As stated, it is always noted that $d^* \le f^*$, and hence we call $\varepsilon = d^* - f^*$ as the Duality gap. If the Duality gap is equal to zero, it is said that there is a Strong duality.

Each possible solution, such as $\Delta\widetilde{U}$ within the set of possible solutions, provides new information about the upper bound of the primal problem, and more precisely, $f^* \le f(\Delta\widetilde{U})$. If we can find a feasible solution such as $\tilde{\lambda}$ for the dual problem $d(\tilde{\lambda}) \le J^*$, we will be able to form an interval for the location of the optimal value of the primal problem where the inequalities $d(\tilde{\lambda}) \le J^* \le J(\Delta\widetilde{U})$ hold. Therefore, without knowing the exact value of J^* and only knowing the value of the duality gap, it is possible to see how far the point obtained is suboptimal.

Algorithms normally solve the primal and dual problems in an iterative manner and form a string of possible solutions such as $\{\Delta\widetilde{U}_k, \tilde{\lambda}_k\}_{k=0,1,\ldots}$ for the primal and dual problems to the point where the stopping condition $\varepsilon < \varepsilon_{desired}$ is established. In quadratic convex programming problems, the feasibility of the solution indicates a strong duality. This means that the desired duality gap can be chosen arbitrarily small.

3-3-4 Softening constraints

All the constraints that have been defined in the problem so far are hard constraints, which means that the controller should not violate them under any circumstances. Hard constraints that are defined on state variables or system output sometimes cause the problem of infeasibility of the solution. For example, unmodeled perturbations may drive outputs outside the range of feasible solutions. Therefore, there will be no defined control command to bring the outputs back within the allowed range at the next time step. For this reason, the constraints imposed on the output (and state variables) are usually relaxed. By defining the variable ε and the vector M, the constraints of the output vector can be softened as follows.

$$y_{\min} - M\varepsilon \le y \le y_{\max} + M\varepsilon \tag{74}$$

In the recent relation, $M_i \ge 0$ is the value related to the degree of softening of the i^{th} adverb of the output. Also, in order to reduce the amount of deviation of the outputs from their allowed values, the term $\rho\varepsilon^2$ is added to the objective function to penalize the violation of the output constraints (ρ is an arbitrary large enough positive number). Therefore, ε will play the role of a new independent variable in the optimization problem [16].

This method can be used to soften control input constraints or control input changes. In the problems of tracking the reference vector by the controller with the property of integration, it is usually recommended that the output constraints are selected as soft as possible and the control input constraints or control input changes are also softened.

3-3-5 Setting the controller parameters of the forecast model

One of the characteristics of the predictive model controller compared to other control approaches is the multiplicity of adjustable parameters in this type of controller. Although this feature gives the designer more freedom to better adjust the behavior of the controller, sometimes it becomes a big challenge for him to properly set the controller parameters. Because at the time of adjusting the controller parameters, the compromise between the conflicting goals of stability, agility, tracking accuracy, and controller speed is inevitable, taking into account the limitation of computing volume and controller memory. It is important to emphasize that the optimality of the control command provided by the predictive model controller is completely dependent on the control parameters setting, and if these parameters are not set properly, the controller will not provide optimal performance.

During many years of development and application of the predictive model controller in industrial applications, general rules have been developed to set the parameters of this type of controller. For example, in the articles related to the adjustment of the controlling parameters of the predictive model, it is recommended that the prediction horizon be chosen to be large enough to include the effective part of the process dynamics. Specifically, for a stable open-loop system, it has been said that 80-90% of the request time of the system's open-loop response to the step input can be a reasonable value for the length of the prediction window. Regarding the control horizon, 10-20% of the length of the prediction window is recommended in, but choosing a control horizon between 3 and 5 is an empirical rule recommended in the control literature [17].

In the predictive control implementation, the following parameters were used:
Prediction Horizon (P): 10 steps, capturing the relevant system dynamics for the engagement scenario.
Control Horizon (C): 3 steps, ensuring computational efficiency while maintaining control accuracy.
Runtime: The MATLAB implementation required approximately 0.2 seconds per control cycle.
These parameters were selected to optimize the trade-off between computational demands and control performance.

3-3-6 Implementation of the algorithm of the online forecasting model

In this paper, the predictive control toolbox of MATLAB has been used to implement the online predictive model controller. This toolbox, while providing the facilities needed to design the implicit predictive model controller with reference signal prediction capability, eliminates the need to build a predictive model from the control-oriented model. For this purpose, it is sufficient to create an object of the predictive control class using the open loop dynamic model of the system and change the methods and features presented in this class based on the needs of the problem.

3-3-8 Controller stability analysis and solution feasibility condition

In general, the stability of the predictive model controller is not guaranteed in advance. Furthermore, the controller may direct the state variables to a part of the state space where no optimal response satisfying the constraints of the problem can be computed in finite time. Therefore, the controller of the predictive model can be implemented when the stability and the condition of the existence of the solution in the entire problem space are examined.

The stability of the predictive model controller feedback loop has been investigated by several researchers. Most of the approaches to prove the stability of the predictive model control are essentially dependent on the arguments of Kirtshi and Gilbert, which show that under some conditions, the optimal objective function is actually a Lyapunov function. In these types of controllers, stability is generally a complex function of various adjustable parameters such as N_p ،N_c ،P ، , and R. If a short control horizon is selected, the controller can easily become unstable. To avoid this situation, the prediction horizon can be considered very large (and ideally infinite). It is clear that such a choice will lead to growing increase in the processing volume of the controller.

Another way to ensure the stability of the controller for an arbitrary prediction horizon is to apply the Terminal constraint on the last state vector in the prediction horizon . In this way, we will ensure that the state vector will converge to a certain vector at the end of the prediction horizon. The drawback of the mentioned method is that the equality condition of the Terminal constraint may cause inefficiency of the controller operation [18]. In addition, in order for such an approach to be possible, the open loop system must be achievable in addition to sustainability [19]. As it was shown earlier, the system studied in this research is not fully controllable and therefore all the state space vectors are not accessible in this problem and this method cannot be used to ensure the stability of the controller.

However, it has been shown that the stability of the controller with a limited prediction horizon is also possible in the absence of stability guarantees. Specifically, it is proved in that a closed-loop control system with predictive model controller is asymptotically globally stable if and only if the associated optimization problem is feasible. Therefore, by showing that the constrained optimization problem is possible in any situation, we can implicitly prove the stability of the controller.

4- Simulations and results

After completing the design of the controllers used, the performance of the designed controllers is investigated in this section. The parameters of the missile and the values of the initial parameters of the missile and target engagement, and the initial conditions in all these simulations are the same and according to Table 1.

Table 1 - Parameters of the homing missile [13]

variable	value	Variable	value
S	0. 42 m^2	J_x	100 kg. m^2
L	0. 68 m	J_y	5700 kg. m^2
m	1200 kg	J_z	5600 kg. m^2
ρ	1. 1558 kg/m^3	m_y^β	-27. 31
m_z^α	-28. 16	$m_y^{\delta_y}$	-26. 57
$m_z^{\delta_z}$	-27. 92	m_x^α	0. 46
c_y^α	57. 16	m_x^β	-0. 37
c_y^β	0. 08	$m_x^{\delta_x}$	2. 12
$c_y^{\delta_z}$	5. 74	c_z^α	-56. 31
c_z^β	-5. 62	$c_z^{\delta_y}$	0. 09

Table 2 - Initial parameters of missile and target engagement [13]

variable	value	Variable	value
$\theta(0)$	45 (deg)	V_m	600 m/s
$\Phi_c(0)$	0 rad	V_t	600 m/s
ω_x	0. 1 rad/s	$x_t(0)$	1136 m
ω_y	0. 1 rad/s	$y_t(0)$	8603 m
ω_z	0. 2 rad/s	$z_t(0)$	5192. 8 m
$x_m(0)$	0 m	θ_{Lf}	30 (deg)
$y_m(0)$	0 m	ϕ_{Lf}	-30 (deg)
$z_m(0)$	0 m	a_T	19. 6*cos(t)

4-1 Linear equations of missile motion and interception in three-dimensional space

For a nonlinear system, consider the general equation (75):

$$\dot{x} = f\left(x,u\right) \tag{75}$$

To obtain the linear system $\dot{x} = Ax + Bu$ corresponding to the nonlinear system (75), we use relations (76):

$$A = \frac{\partial f\left(x,u\right)}{\partial x}\left|x = x_0, \quad u = u_0\right.$$

$$B = \frac{\partial f\left(x,u\right)}{\partial u}\left|x = x_0, \quad u = u_0\right. \tag{76}$$

In relation (76), x_0 and u_0 are the operating points of linearization. According to this relationship, to find the two matrices A and B, we must find the partial derivatives of f with respect to x and u at the operating point (i.e., the Jacobian matrix), respectively.

We now define a new linear system as Eq. (77):

$$\dot{x} = Ax + Bu \tag{77}$$

In Eq. (77), the state and control vectors are defined as $x = \left[x_0^T, x_1^T, x_2^T, x_3^T, \theta, \phi_v, R, \dot{R}\right]^T$ and $u = \left[\delta_x, \delta_y, \delta_z\right]^T$, respectively. In fact, we define a linear system in such a way that it includes all the state variables defined in the previous sections. According to the above equations, we will have the numerical values given in Table 2, and finally using the equation (76), The following matrices are obtained for A and B.

$$A=\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -6.67 & 7.23 & 0 & 0 & -6.70 & -204.6 & 0 & 0 & 0 & 0 & -5.22 & -1.04 & -8.09 & 0 \\ 2.41 & 2.41 & 0 & 0 & -3.46 & 0.682 & 0 & 0 & 0 & 0 & 4.8 & -2.41 & -4.17 & 0 \\ 0 & 0 & 0 & 0 & -0.0116 & 0.0003 & -0.0003 & -0.0175 & 0.0003 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.0114 & 0.0011 & 0.0142 & 0.0175 & 1 & 0 & -0.0001 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.0063 & -0.0137 & 0 & 1 & -0.0175 & 0 & 0.0004 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.45 & -0.36 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.47 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.49 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.0118 & 0 & 0.0002 & 0 & 0 & 0 & 0.0082 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.0112 & 0.0013 & -0.0002 & 0 & 0 & 0 & 0.0001 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -0.1672 & -0.0609 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B=\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 2.0994 & 0 & 0 \\ 0 & -0.4616 & 0 \\ 0 & 0 & -0.4937 \\ 0 & -3.17e^{-7} & 0.0012 \\ 0 & -2.10e^{-5} & -2.33e^{-5} \\ 0 & 0 & 0 \\ 0 & 9.6252 & -567.3836 \end{bmatrix}$$

In obtaining the numerical values of these matrices, the operating points are considered as follows (actually, all operating points should have been considered zero, but since the denominator of some ratios became zero and undefined, small near values are considered for some operating points instead).

$$x_0^0=[0,0]^T\ ,\ x_1^0=[0,0]^T\ ,\ x_2^0=\left[\frac{\pi}{180},\frac{\pi}{180},\frac{\pi}{180}\right]^T,\ x_3^0=[0,0,0]^T\ ,\ \theta^0=\frac{\pi}{6}\ ,\ \phi_v^0=0\ ,\ R^0=1\ ,dR^0=0$$

4-2 PID controller simulation performance

In this section, the performance of the PID controller, whose parameters are adjusted by the Ziegler-Nichols method according to Table 3, is examined.

Table 3 – Parameters of the PID controller

gain type	gain amount
K_p	0.732
K_i	0.417
K_d	1.669

The simulation results for PID controller performance are shown in Fig. 3 to 6.

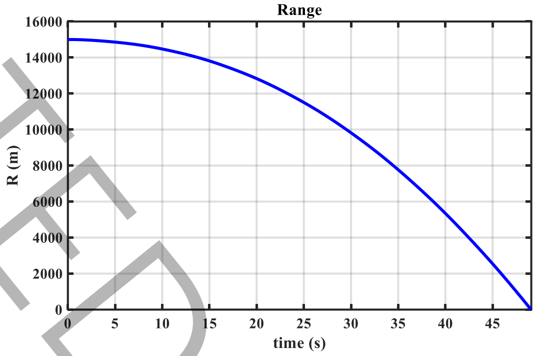

Fig. 3- The missile-target relative distance using the PID controller

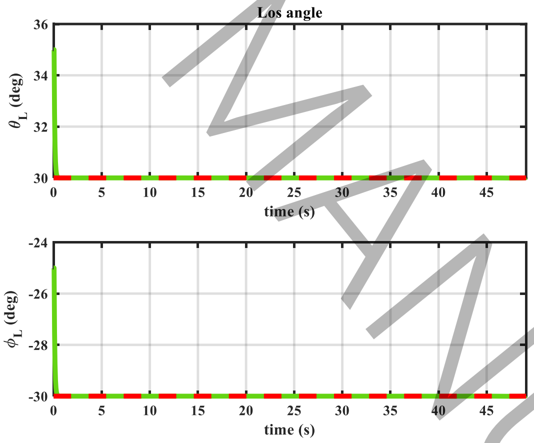

Fig. 4 - Changes of elevation and side angles with time using the PID controller

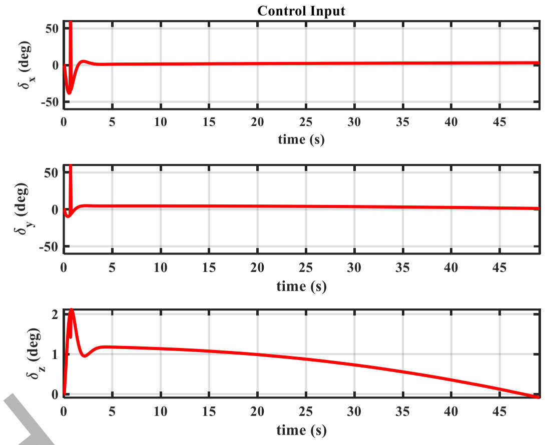

Fig. 5 - Missile control input with the PID controller

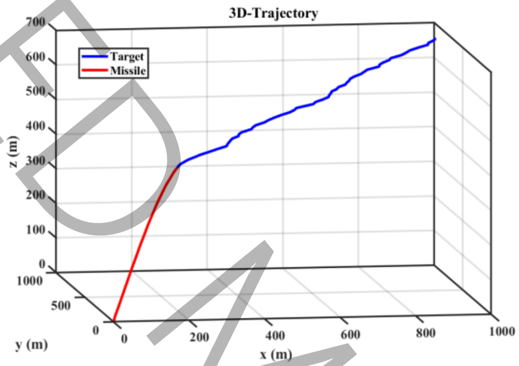

Fig. 6 – Missile-target trajectories using the PID controller

The graphs in Fig. 3 and 4 show that in about 50 seconds, the relative distance between the missile and the target reaches zero. Also, the altitude and side angles reach their target values at the beginning of the flight. The control input in the diagram of Fig. 5 forced the missile to move towards the target according to the trajectories shown in Fig. 6 and hit the target at a height of 310 meters. In general, it can be said that the performance of the PID controller is poorly evaluated, since the flight time in this controller is rather too long for a short-range scenario, and the collision between the missile and the target occurred at a too low altitude, intuitively not considered an appropriate height for collision in air defenses. In addition, in the PID controller, the angles reach their reference values at the beginning of the missile's flight and do not converge during the flight, and this makes the target recognize the missile's path, hence lures the target to try to evade. The PID controller coefficients are obtained using Ziegler-Nicols method. The output of this method is normally fine and usually works better than that from optimization methods. Use of optimization tools in adjusting the PID controller gains are time-consuming and the desired result may not be quite reasonable due to spending too long a time. Moreover, the linearization required for the PID controller will make PID responsive only in limited operating points. According to the dynamics of the missile, linearization is not guaranteed for all modes, and even if the PID controller gains were designed using the neural and fuzzy network method, because the linearization was done, it is not valid in all domains. In addition, by changing the coefficients of the PID controller, its results are not improved in comparison with those from other controllers. Even if it is possible to achieve such a state, its control effort will be greatly increased and the system will be in a saturated state. As a result, PID is not considered as an appropriate controller for this system. In addition, the missile-target collision time with the PID controller is so high that PID is not a proper controller for the missile to hit the target in a reasonable time span. In air defense systems, particularly in short range scenarios, time plays a very key role, even if some control effort is increased. In the end, it can be said that although the PID controller is simpler from an implementation perspective, it does not respond properly to complex systems. As a result, a controller that is suitable for complex nonlinear systems will be more reliable.

4-3 Simulation of LQR controller performance

In order to simulate the LQR controller, first, the appropriate values for the Q and R matrices for target-missile engagement in this research were extracted and determined using the particle swarm optimization algorithm. The results of this optimization process for these two diagonal matrices are obtained as follows.

$Q = diag[0.0988,5.8532,0.0893,4.8708,9.3566,0.8765,1.2646,3.1068,9.8594,4.4510,9.8985,3.0687,0.1200,0.0380]$

$R = diag[9.1104,1.7283,1.3463]$

Using these two matrices in the cost function of relation (32), the changes in the values of the cost function were obtained and depicted in Fig 7. It is observed that the time to reach the end of the simulation is 1154 seconds for the PSO algorithm.

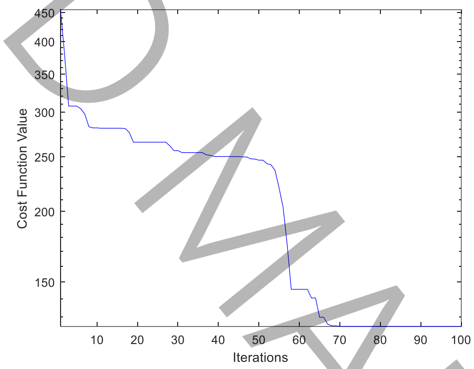

Fig. 7 - Changes in the cost function in optimization with time using the LQR controller and genetic algorithm

In the following, using the linearized state matrices and Riccati equation, the control input and gain matrix are calculated and the results from the simulation can be seen in Fig. 8 to 11.

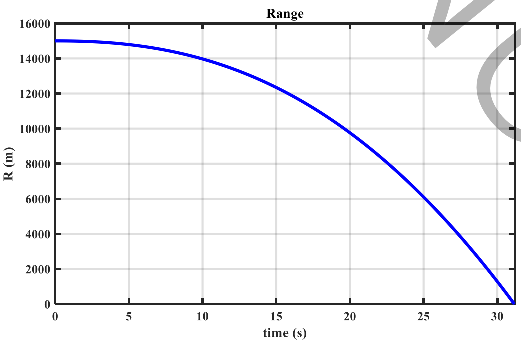

Fig. 8 - The missile-target relative distance using the LQR controller

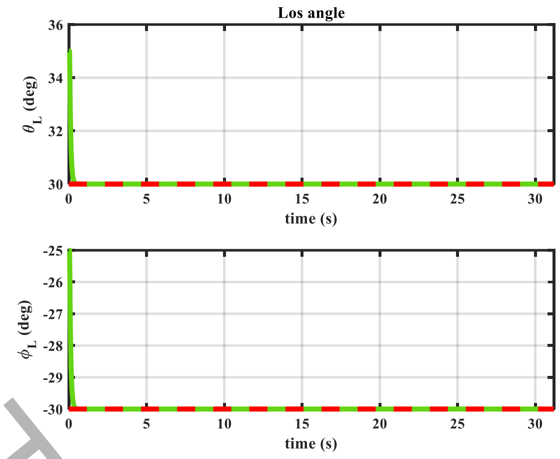

Fig. 9 - Changes of elevation and side angles with time using the LQR controller

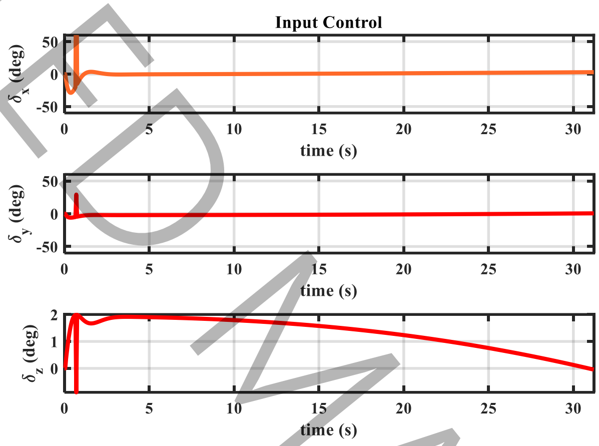

Fig. 10 Missile control input with the LQR controller

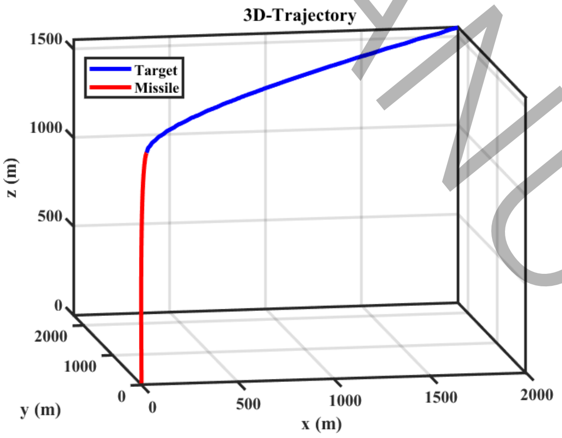

Fig. 11 – Missile-target trajectories using the LQR controller

The graphs in Fig. 8 show that in about 32.3 seconds the relative distance between the missile and the target reaches zero, resulting in collision. Also, the altitude and side angles reach their target values at the beginning of the flight, as shown in Fig. 9. The control input in the diagram of Fig. 10 forced the missile to move towards the target according to the trajectory drawn in Fig. 11, and hit the target at an altitude of 823 meters. As mentioned earlier, in the design of this LQR controller, the particle swarm optimization method was employed to calculate the weight matrices, rendering this controller outperform the PID controller.

Although a better result was obtained, it can be stated that the performance of the LQR controller is yet evaluated as fair at best, because the flight time in this controller is still rather large for a short-range surface-to-air engagement, plus the fact that the height at which the missile-target collision occurred is still rather low, making this controller still inappropriate for air defense purposes. As stated earlier, in cases of short-range air defense, it is essential that the missile and the target collide at a rather higher altitude, and the target should not get the chance to come close to the defense site positions. In addition, in the LQR controller, the angles reach their reference values at the beginning of the missile's flight rather than converge throughout the flight, which offers the target chances to recognize or foresee the missile's path and try to evade the missile and avoid collision. The weight matrices of this LQR controller are obtained using the particle swarm optimization method. Also, for the LQR controller, linearization must be done, with the linearized model effective only in limited operating points. Due to missile dynamics, linearization is not guaranteed for all modes. The time to collision using this LQR controller is rather high, which is not desirable, since time plays a key and decisive role in a missile-target conflict.

4-4 Simulating the performance of the implicit predictive model controller

In this section, performance of the implicit predictive model controller is examined.

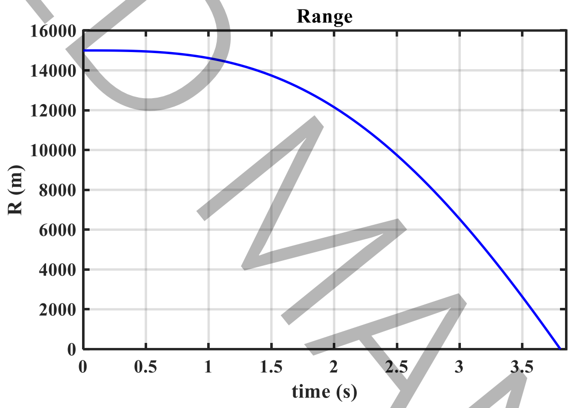

Fig. 12 - The missile-target relative distance using the implicit predictive model controller

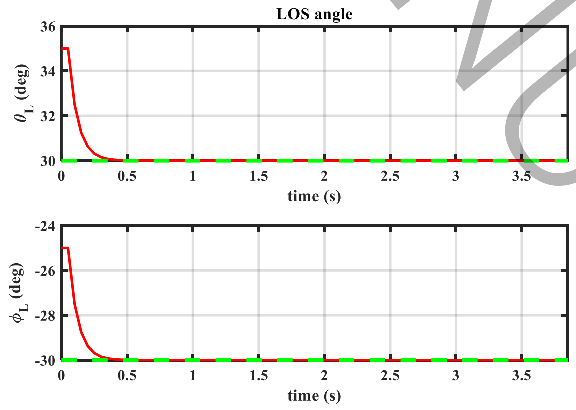

Fig. 13 - Changes of elevation and side angles with time using the implicit predictive model controller

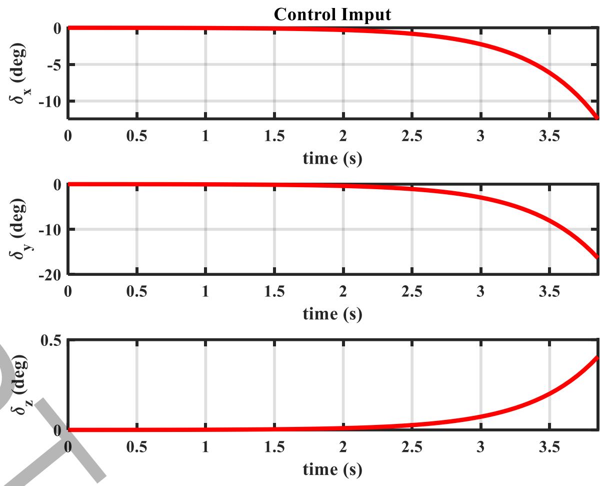

Fig. 14 - Missile control input with the implicit predictive model controller

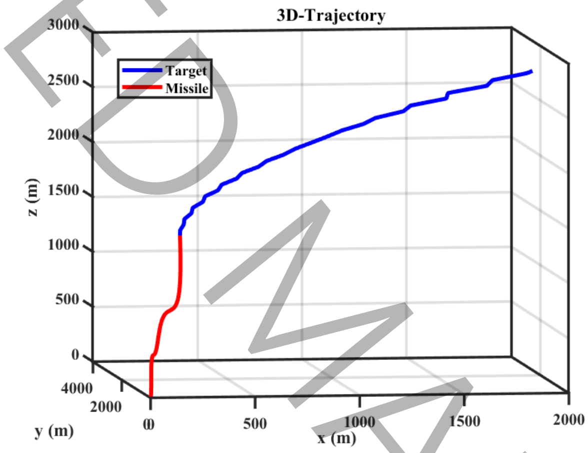

Fig. 15 Missile-target trajectories using the implicit predictive model controller

Fig. 12 shows the missile-target relative distance, which distance approaches close to zero in about 3.72 sec. by which collision takes place. This is a reasonable time in air defenses for short range scenarios, because after firing, the missile quickly moves towards the target in this short time, and the missile does not leave a chance for the target to evade the missile by a maneuver to avoid collision with the missile. Also, Fig. 13 shows that the flight angles converge in reasonable time. According to Fig. 14, the control input, appearing quite reasonable, forced the missile to move towards the target, as verified by the missile and target trajectories depicted in Fig. 15, leading to collision at a height of approximately 1260 meters. Online linearization method was used in MPC controller design and according to the simulation results, the performance of this controller was quite superior compared to those of PID or LQR controllers. As the simulation results indicate, it can be inferred that the performance of the MPC controller is evaluated as superior due to the fact that the flight time with this controller is reasonable for a short-range surface-to-air engagement, and the height at which the missile-target collision occurs is more of a reasonable one in cases of short-range defenses. As stated earlier, in a short-range air defense, it is normally best for the missile and the target to collide at a higher altitude so that the target would not come too close to the missile defense site.

To ensure a comprehensive evaluation of the controllers, not only were the miss distance and time to impact considered, but also the control effort required by each approach was examined. The control effort was quantified using the Integral of Squared Control Effort (ISCE) metric, defined as:

$$J_u = \int u^2 dt \tag{70}$$

, where u represents the control input. This metric provides insight into the total energy expenditure of the control system, ensuring that performance improvements do not come at the cost of excessive actuator effort.

The results, presented in Table 4, indicate that the MPC-based controller achieves the best overall performance, minimizing both the miss distance and time to impact, while also requiring significantly lower control effort compared to PID and LQR controllers. The PID controller exhibits the highest control effort, which can be attributed to its reliance on direct error correction without predictive optimization. The LQR controller demonstrates a better balance between performance and effort; however, it lacks the constraint-handling capabilities of MPC, which allows for more efficient actuator usage.

The MPC controller, benefiting from its predictive capability and optimal constraint handling, results in the lowest control effort, ensuring that the missile maneuvers efficiently without excessive actuator activation. This highlights the advantage of using an online predictive control strategy for integrated guidance and control systems, particularly in scenarios where energy efficiency and actuator longevity are critical factors. Finally, a quantitative comparison between the results of the controllers studied in this article is presented in Table 4.

Table 4 – Quantitative performance comparison of the three controllers examined

	PID	LQR	implicit MPC
missile flight time	49.66 (sec)	32.30 (sec)	3.72 (sec)
Missile-target collision height	310 (m)	823 (m)	1260 (m)
Control Effort	245.7	190.2	85.4

5- Conclusion

In this paper, the guidance and control of an air defense Flying Vehicle is proposed using online model predictive control for integrated 3D Flying Vehicle-target model. The integrated guidance and control equations of the Flying Vehicle and the target were fully derived , followed by the design of the controllers. In order to evaluate the proposed controllers, a PID controller was considered as well. The results of this controller were not evaluated favorably due to long flight time and inappropriate control law, leading to low flight height. Next, the LQR controller was designed and the results of this controller were evaluated according to the simulations. Then the proposed controller of this article, namely the predictive controller of the online model, was designed. According to the simulations, it can be said that the total time to missile-target collision is about 3.72 seconds, which is a quite reasonable time for short-range air defense missiles, and the control law is fully applied within the available range without saturation. This control law causes the missile to maneuver and the missile hits the target at an altitude of 1260 meters. Such an acceptable height is quite important. In addition, the time to collision was not too high in short-range situations, so that the hostile target did not have the chance to position itself and perform unpredictable maneuvers to evade the missile and avoid collision. Overall, the performance of the predictive controller of the online model was evaluated favorably.

5- References

[1] P. Zarchan, Tactical and strategic missile guidance. American Institute of Aeronautics and Astronautics, Inc. , 2012.

[2] Neil F. Palumbo, Ross A. Blauwkamp, and Justin M. Lloyd. Basic principles of homing guidance. Johns Hopkins APL Technical Digest (Applied Physics Laboratory), 2010.

[3] Nathan Harl and S. N. Balakrishnan. Reentry terminal guidance through sliding mode control. Journal of guidance, control, and dynamics, 33(1): 186—199, 2010. doi. org/10. 2514/1. 42654

[4] Wang, X. H. , Tan, C. P. , & Cheng, L. P. (2020). Impact time and angle constrained integrated guidance and control with application to salvo attack. Asian Journal of Control, 22(3), 1211-1220

[5] He, S. , Song, T. , & Lin, D. (2017). Impact angle constrained integrated guidance and control for maneuvering target interception. Journal of Guidance, Control, and Dynamics, 40(10), 2653-2661.

[6] Ma, J. , Guo, H. , Li, P. , & Geng, L. (2013). Adaptive integrated guidance and control design for a missile with input constraints. IFAC Proceedings Volumes, 46(20), 206-211

[7] Michael A. Cross. Missile interceptor integrated guidance and control: single loop higher order sliding mode approach, PhD thesis , The University of Alabama in Huntsville, 2020.

[8] Jiayi Tian, Neng Xiong, Shifeng Zhang, Huabo Yang, Zhenyu Jiang, Integrated guidance and control for missile with narrow field-of-view strapdown seeker, ISA Transactions, vol. 106,2020, Pages 124-137. https: //doi. org/10. 1016/j. isatra. 2020. 06. 012

[9] Sinha, Abhinav & Kumar, Shashi & Mukherjee, Dwaipayan, Integrated Guidance and Control For Dual Control Interceptors Under Impact Time Constraint, AIAA scitech Forum Conference, January 2021. doi. org/10. 2514/6. 2021-1463

[10] Li Z. , Dong Q. , Zhang X. , Zhang H. , Zhang F. Field-to-View Constrained Integrated Guidance and Control for Hypersonic Homing Missiles Intercepting Supersonic Maneuvering Targets, Aerospace. 2022; 9(11): 640. https: //doi. org/10. 3390/aerospace9110640

[11] Xiaohui Liang, Bin Xu, Kunhao Xinglan LiuAdaptive NN control of integrated guidance and control systems based on disturbance observer, Journal of the Franklin Institute, Volume 360, Issue 1, 2023

[12] Xiangyu Tang, Jianglong Yu, Xiwang Dong, Zhang Ren, Integrated guidance and control with impact angle and general field-of-view constraints, Aerospace Science and Technology, Volume 144, 2024. DOI: 1 10. 1016/j. ast. 2023. 108809

[13] S. Ma, A. Li و Z. Wang, “Integrated Guidance and Control for Homing Missiles with Terminal Angular Constraint in Three Dimension Space", IEEE International Conference on Artificial Intelligence and Information Systems, Dalian, China, 2020. DOI: 10.1109/ICAIIS49377.2020.9194808

[14] Modern control Engineering 5th Edition, by Katsuhiko Ogata,2009.

[15] W. Prestl, T. Sauer, J. Steinle and O. Tschernoster, "The BMW active cruise control

ACC," SAE Technical Paper Series, SAE 200010344, 2000, p. 1–7.

[16] A. Bemporad, "Recent advances in embedded model predictive control," Michigan Engineering, 14 July 2016. [Online]. Available: https://www.youtube.com/watch?v=ugeCx1sytNU. [Accessed 1 June 2019].

[17] J. L. Garriga and M. Soroush, "Model predictive control tuning methods: A review," *Industrial & Engineering Chemistry Research*, vol. 49, no. 8, pp. 3505-3515, 2010.

[18] R. Rajamani, "Vehicle dynamics and control," in *Mechanical Engineering Series*, Springer, 2012, pp. 87-109 & 241-162.

[19] J. H. Lee, "Model predictive control: review of the three decades of development," *International Journal of Control, Automation and Systems*, vol. 9, p. 415–424, 2011.