Nonlinear model-predictive integrated missile control and its multi-objective tuning

Vincent Bachtiar* and Chris Manzie†, 

The University of Melbourne, Melbourne, Victoria 3010, Australia

Eric C. Kerrigan‡ 

Imperial College London, London SW7 2AZ, U.K.

I. Introduction

Advanced missile control designs have been founded on recent developments aimed at a more streamlined engineering design approach to improve performance and cost of design. Central to these developments is the attempt for a full integration of subsystems to exploit any potential synergy. In missile control, much effort has been concentrated on integrating the autopilot and guidance subsystems [1, 2]. Such a unification of the control algorithm has direct benefits for the performance of the missile, primarily by removing the previously existing lag between the commanded and tracked acceleration from guidance and autopilot, respectively [3]. Several strategies have been proposed following this integrated control outlook, including sliding mode control [4, 5, 6], state feedback regulators [7, 8], as well as model-based approaches, for instance, a recently proposed linear quadratic regulation with model bias [9]. Model predictive control (MPC) is recognized for its ability in handling nonlinearities and constraints, therefore it is suited for high-performing agile missiles operated near constraints. This is demonstrated, for example, by the fact that the predominant proportional navigation (PN) in missile guidance is based on linear quadratic regulation [10], which is a special case of model-predictive control where constraints are absent and linear prediction model is used. Furthermore, the performance and feasibility of model-predictive missile autopilot has been studied in [11, 12].

Although promising a superior performance, MPC is typically associated with high levels of computational cost. Therefore, prior to the real-time realization of MPC, it is important to know the required computational capacity to implement MPC, especially in applications involving fast sampling rates such as missile control. A comprehensive offline tuning of MPC that balances both performance and cost is therefore needed before the implementation. In light of this, as a natural tuning method for an integrated autopilot and guidance algorithm, a multi-objective design approach that considers both closed-loop performance and required computational capacity allows for the comprehensive consideration of both algorithm and implementation designs of the system.

A coupled approach in control tuning is known to be valuable due to its potential in streamlining the design process and produce system-optimal designs [13]. This allows for a more comprehensive analysis, such as the optimization of multiple underlying tuning/design objectives along with constraints, as introduced and discussed in [14]. This notion

*Ph.D., Department of Mechanical Engineering. 
†Professor, Department of Mechanical Engineering. 
‡Reader, Department of Electrical and Electronic Engineering and Department of Aeronautics.
is echoed in the design of MPC where closed-loop control performance competes with computational capacity due to the computationally expensive nature of model-predictive control architectures, particularly due to the need for online optimization. In particular, consideration of both control algorithm and implementation to optimize both closed-loop performance and cost based on the required computational capacity is an effective design approach, avoiding conservative and unnecessarily over-specified designs as well as extensive design iterations due to implementation feasibility issues when algorithm and implementation are considered separately.

Given its benefits, existing techniques to find optimal solutions in a multi-domain control tuning are in abundance. Due to the complexity of the problem, optimizers are predominantly founded on stochastic/meta-heuristic approaches which are general purpose optimizers that can solve a wide range of problems without much knowledge of the problem nature and characteristics. Examples include particle swarm optimization and genetic algorithms [15]. More recently, combined optimization problems in control have been analyzed with more rigor and depth to reveal certain characteristics and properties of the problem. In these analyses, deterministic optimization algorithms have been used [16, 17].

This paper presents a model-predictive integrated missile control design. The controller is based on MPC with a nonlinear prediction model on the full dynamics of the missile engagement as well as input and state constraints, extending existing works on model-based integrated missile control, e.g. [9]. The study then demonstrates how the proposed nonlinear MPC-based controller can be calibrated (tuned) offline with a multi-objective outlook, considering both closed-loop performance and required computational capacity to implement the controller. This extends recent work of [16] on linear MPC. Two tuning objectives are considered, the first being the miss-distance to measure control performance whilst the second is the required computational load to implement the controller as a measure of the design cost of the controller. The tuning parameters studied are those pertaining to the structure of the MPC, namely the sampling time and number of prediction steps, thus affecting both objectives competitively. The solution method for the resulting multi-objective optimization problem is based on a Lipschitzian optimizer, dubbed Dividing Triangles (DITRI) and developed in [16]. The result of the design approach is the optimal trade-off curve consisting of a set of optimal designs that the practitioner can base the controller design on. The optimal trade-off curve can be used to draw insights for instance on the maximum performance for a range of computational capacity as well as the required cost to be paid in computational capacity for a unit improvement in performance.

Notational conventions and definitions

Unless otherwise stated, $\|v\|_M^2 := v^T M v$. $\|v\| := \sqrt{v_1^2 + \ldots + v_n^2}$. Binary operators such as $=$, $<$, $\leq$, etc. define element-wise relationship. $U[a,b]$ is a random number uniformly distributed between $a$ and $b$. $O_n$ denotes an $n \times n$ matrix with zero elements. $\ell(p) \prec \ell(p^*)$ denotes that an evaluation point $\ell(p^*)$ dominates the point $\ell(p)$, which is true iff $\ell_i(p^*) \leq \ell_i(p)$ for all $i \in \{1, \ldots, n_e\}$ and $\ell_i(p^*) < \ell_i(p)$ for at least one $i$. 

American Institute of Aeronautics and Astronautics
II. Equations of motion

A. Missile dynamics

A cruciform, tail-controlled, and roll-stabilized missile is considered in this study. Focus is given on the end-game phase of the flight, in which the target can be assumed to be close, such that control can be separated into two equivalent planar channels that are perpendicular to each other. Modeling of the missile equations of motion is presented for control in one channel in the context of pitch control.

The missile dynamics are defined around the body and wind axes. The body axis is fixed to the physical frame of the missile, whereas the wind axis is aligned to the velocity of the missile (see Fig. 1). Both axes are centered at the center of gravity of the missile. The missile translational and rotational dynamics are described with respect to the body axis, defined as

\[ \dot{\alpha} = q - \frac{L(\alpha, \delta)}{(mV)}, \]
\[ \dot{\theta} = q, \]
\[ \dot{q} = \frac{M(\alpha, \delta)}{I_y}. \]

\( \alpha \) is the angle of attack and \( q \) is the pitch rate of the missile. The missile speed \( V = M_a V_s \), where \( V_s \) is the speed of sound, is constant at Mach number \( M_a \). The missile is flying at speed and altitude with dynamic pressure \( P \). The actuation of the fin deflection \( \delta \) is modeled as a second order system,

\[ \dot{\delta} = \dot{\delta}, \]
\[ \ddot{\delta} = \omega_a^2 (\delta_c - \delta - 2\zeta \dot{\delta}), \]
\[ \dot{\delta}_c = u, \]

where the commanded fin deflection \( \delta_c \) rate is the input \( u \) of the system. Aerodynamic lift and moment are modeled to be cubic with respect to the angle of attack and linear with respect to the effective control surface deflection [11, 18]. That is,

\[ L = PSC^{L}, \quad M = PSC^{M}, \]

where \( P \) is the dynamic pressure and \( S \) is the effective surface area. The aerodynamic coefficients are

\[ C^{L} := \sum_{i=1}^{3} c_i^F \alpha_i + c_i^F \delta, \quad C^{M} := \sum_{i=1}^{3} c_i^M \alpha_i + c_i^M \delta. \]

\( c \) is the stability and control derivatives of the missile, describing how forces and moments (denoted by its superscript) change with respect to its parameters (denoted by its subscript).
B. Missile and target kinematics

The kinematics of the missile is defined with respect to a fixed inertial coordinates, e.g. ground/earth-based, as illustrated in Fig. 2. The missile flight-path angle is denoted by $\gamma$. The seeker detects the location of the target relative to the missile, which is defined at a distance $r$ and angle $\chi$. Based on the seeker and pitch angles, $\chi$ and $\theta$, a line-of-sight (LOS) angle $\lambda$ can be derived, which measures the relative location of the target to the missile with respect to the fixed inertial axis.

The evolution of the missile location in Cartesian coordinates $s_M := (s_{M1}, s_{M2})$ with respect to the inertial axis is given by

\[
\dot{s}_{M1} = V \cos \gamma, \quad \dot{s}_{M2} = V \sin \gamma,
\]

where $\gamma = \theta - \alpha$. The target is assumed to have a constant acceleration $a_T$. The target location and velocity is denoted by $s_T$ and $v_T$.

\[
\dot{s}_T = v_T, \\
\dot{v}_T = a_T.
\]

In relative terms, the separation between the missile and target is denoted by $s$ and modeled as

\[
\dot{s}_1 =: v_1 = v_{T1} - V \cos \gamma, \quad \dot{s}_2 =: v_2 = v_{T2} - V \sin \gamma,
\]

and the relative acceleration between the missile and the target is

\[
\dot{v}_1 = a_{T1} - \frac{L(\alpha, \delta)}{m} \sin(\theta - \alpha), \quad \dot{v}_2 = a_{T2} - \frac{L(\alpha, \delta)}{m} \cos(\theta - \alpha).
\]

The LOS angle and its evolution in time can be derived from the relative distances $\lambda = \arctan(s_2/s_1)$ such that the LOS second order rate is given as

\[
\ddot{\lambda} = \frac{(s_1^2 + s_2^2)(s_1 \ddot{s}_2 - s_2 \ddot{s}_1) + 2(s_1 \dot{s}_2 - s_2 \dot{s}_1)(s_1 \dot{s}_1 + s_2 \dot{s}_2)}{(s_1^2 + s_2^2)^2}. \quad (11)
\]
C. Engagement model

Summarizing the kinematics and dynamics is a plant model with states \( x := (\alpha, \theta, q, \dot{\delta}, \delta_c, s, v, \dot{\lambda}) \) and control input \( u := \dot{\delta}_c \). Put concisely,

\[
\dot{x} = f(x, u),
\]

where \( f \) is obtained from (1)–(6) and (9)–(11).

III. Control formulation

A. Control strategy

An intercept control strategy will be considered. An intercept means that the relative position of the missile and the target \( s \) is zero. This is contrasted to, for example, a rendezvous strategy where in addition to zero relative distance, the lateral relative velocity between the missile and the target is required to be zero.

The proposed integrated model-predictive autopilot and guidance (iMAG) scheme is based on a parallel navigation that regulates the rate of the LOS angle \( \dot{\lambda} \) that dictates the relative trajectory between the missile and the target. Parallel navigation relies on the fact that a zero LOS rate during engagement is a sufficient condition for the missile to intercept the target given that the missile is approaching the target (see, for example, [10]). Therefore, the control strategy employed will be to steer the missile such that the LOS rate \( \dot{\lambda} \), whose evolution is as described in (11), is regulated to zero.

Note that as the relative distance approaches intercept at \( s = 0 \), the LOS second order rate (11) approaches infinity,

\[
\text{if } \dot{s} \neq 0 \text{ or } \ddot{s} \neq 0 \text{ or } \dot{s}_1 \neq \dot{s}_2 \text{ or } \ddot{s}_1 \neq \ddot{s}_2,
\]

then

\[
\lim_{s_1 \to 0, s_2 \to 0} \dot{\lambda} = \infty.
\]

As the missile approaches the target, relative velocity and acceleration is nonzero throughout the engagement and its components are not guaranteed to be equal. Therefore, the conditions in (13) are validated and thus it is inevitable that the LOS and its rates deviates significantly at the end of the engagement.

Remark 1. Note that the LOS can also be derived from the measurable angles \( \lambda = \theta - \chi \) such that

\[
\dot{\lambda} = q - \dot{\chi},
\]

\[
\ddot{\lambda} = \ddot{q} - \ddot{\chi}.
\]

\[
\text{Figure 2. Frame for the kinematics of the missile and target.}
\]
In simulation, the angles \( q, \chi \), and their derivatives have to be calculated based on the relative distances, so there is no advantage in using (14). However, in practice, the seeker angle and its rate are readily available from the inertial measurement unit (IMU) and seeker of the missile, whereas the distances \( s \) might be more difficult to obtain directly. Therefore, (14) might be a preferable alternative to (11) in the actual implementation of the controller.

B. Predictive model

A predictive model is used internally by the controller to effectively steer the missile to intercept the target. The predictive model states are \( x := (\alpha, \theta, q, \dot{\delta}, \delta_c, s, v, \lambda) \) with input \( u := \dot{\delta}_c \). The first six predictive model states are the missile dynamics modeled as (1)–(6). The last three states in the predictive model is required for the modeling of the LOS angle \( \lambda \). The inclusion of the LOS angle in the predictive model used by the controller is required by the control strategy based on the regulation of LOS rate.

Overall, the predictive model is denoted as

\[
\dot{x} = f(x, u),
\]

where \( f \) is obtained from (1)–(6) and (9)–(11). Notice the distinction in notation for the real-time plant model \( \dot{x} = f(x, u) \) and \( \dot{x} = f(x, u) \) used internally for the controller. This is due to the fact that the predicted dynamics does not necessarily always match the actual evolution of the states. Furthermore, some applications might choose a different representation for the predictive model, for example a linearized or reduced version of the plant model.

There are several constraints to make sure that the missile has good performance. A constraint on the angle of attack is required to keep the flight conditions within an envelope where aerodynamic forces can be accurately represented by the models (8). Fin deflection \( \delta \) and its rate \( \dot{\delta} \) are mechanically limited by the actuator. Finally, the angle \( \chi \) may be constrained to naturally reflect the effective scope of the seeker. The constraints can be defined as

\[
x \in \mathcal{X}, \quad u \in \mathcal{U}.
\]

In this study, the sets are polytopic, \( \mathcal{X} = \{ x \mid |x| \leq \pi \} \) and \( \mathcal{U} = \{ u \mid |u| \leq \pi \} \) where \( \pi \) and \( \bar{\pi} \) are the state and input bounds, within which the predictive model states and inputs are allowed in.

C. Model predictive missile autopilot and guidance

The missile is controlled by commanding the fin deflection rate \( u := \dot{\delta}_c \) to steer the missile. Control is commanded in a sampled-data fashion at sampling instants \( t_i, i \in \mathbb{N}_0 \), with sampling interval \( h \), and is restricted to a zero-order-hold (ZOH). Let the control law be denoted by \( \kappa(o_i, p) \), calculated based on the controlled states \( o_i \) available at sampling instant \( t_i \), as well as control design parameters \( p \).

The control states \( o_i \) are measured from the real-time states \( x_i \) that are relevant to the predictive model, obtained via sensors. In real-world, these are affected by disturbance from sensor errors, and will be modeled by

\[
o_i := x_i + d.
\]
Note the studied sensor error is distinct to plant disturbance which enters when simulating the plant dynamics, e.g. \( f(\cdot) + d \).

A model-predictive controller is formulated following the outlined dynamics and kinematics of the missile. Since the input is constrained with ZOH, the optimal control problem (OCP) can be recast in discrete form. The prediction horizon \( T \) is divided into \( N \) prediction steps, each with an interval length of the sampling time \( h \). The predicted states and input are presented in discrete-time with the vectors

\[
\mathbf{x} := (x_0, \ldots, x_N), \quad \mathbf{u} := (u_0, \ldots, u_{N-1}),
\]

where \( x_k := x(kh) \), \( u_k := u(kh) \).

Subsequently, the nonlinear MPC control law is obtained by solving a discrete-time optimal control problem

\[
(x^*(\cdot), u^*(\cdot)) := \arg \min_{(x, u)} J(x, u, p)
\]

s.t. \( x_0 = o_i := x_i + d \) \quad (18a)

\[
x_{k+1} = \mathcal{F}(x_k, u_k) := x_k + \int_{t_i}^{t_i+h} f(x(\tau), u_k) d\tau \quad \forall k \in \{0, \ldots, N_e - 1\}
\]

\[
x_k \in [-\pi, \pi], \quad u_k \in [-\pi, \pi] \quad \forall k \in \{0, \ldots, N_e - 1\}
\]

To keep the notation succinct, the dependence of \( x^* \) and \( u^* \) on \( o_i \) and \( p \) is omitted. Consequently, the control law is given by

\[
u(t) = u_i \quad \forall t \in [ih, ih + h), i \in \mathbb{N}_{\geq 0},
\]

\[
u_i = \kappa(o_i, p) := u^*_0.
\]

The control law constitutes the model-predictive integrated autopilot and guidance proposed in this study.

The OCP optimizes the cost function \( J \) that represents the performance of the plant. The optimization is subject to the predictive model (18c) representing the dynamics of the plant and its constraints (18d) as given in (15) and (16). The prediction model is nonlinear, hence the control law is a nonlinear MPC. The model is initialized at \( o_i \) as given in (18b), where \( o_i \) consists of the relevant states available at sampling instant \( t_i \) from measurements.

The cost function (18a) is chosen as a quadratic

\[
J(x, u, p) := \sum_{k=0}^{N_e-1} \|x_k\|_Q^2 + \|u_k\|_R^2 + \|x_{N_e}\|_{Q_t}^2,
\]

that penalizes the state/input deviations from zero. The function is composed of a stage cost weighted by \( Q \geq 0 \) and \( R > 0 \), and a terminal cost weighted by \( Q_t \geq 0 \), where \( > \) and \( \geq \) denote positive and semi-positive definiteness, respectively, in this case. The number of (effective) prediction steps \( N_e \) defines the prediction horizon length of the OCP.

A natural choice for the weight \( Q \) is to penalize first and foremost the LOS rate. Secondly, the pitch rate \( q \) will
also be penalized, with relatively less weight than LOS rate, to avoid chattering. Terminal cost weight \( Q_t \) will be set in a similar fashion to the state stage cost weight \( Q \) for the same reasons. For the input, the fin deflection rate, a small nonzero \( R \) is desired to avoid chattering and ensuring uniqueness of the OCP solution.

Finally, the length of the prediction horizon is chosen to be

\[
N_e := \min\{ N, \lfloor t_{go}/h \rfloor \}
\]

(20)

where \( t_{go} := \|s\|/\|v\| \), to ensure that the prediction does not exceed the point beyond intercept. This is motivated by the fact that the LOS second order rate \( \ddot{\lambda} \) is very sensitive around the intercept as shown in (13), and would easily cause numerical instability.

### IV. Multi-objective offline tuning framework

In this section, the multi-objective tuning of MPC (MOD-MPC) problem will be formulated for the calibration of the proposed controller introduced earlier. This is to be solved offline in practice. This can be regarded as an (outer) optimization of the OCP optimization problem. The two tuning objectives are the control performance and required computational capacity of the MPC. These two measures define the pertaining objectives for the control algorithm and implementation designs that are to be optimized via tuning of the structural design parameters \( p \) of the OCP algorithm.

#### A. Tuning objectives

The first and foremost tuning objective to be optimized in a missile control system is the closed-loop performance of the controller based on the ability of the missile to intercept the target and how well the controller guide the missile in doing so, particularly under disturbance. Following the multi-objective approach, the algorithm and implementation design of the control system will be designed simultaneously. In this context, another fundamental objective is the implementation cost dictated by the required computational capacity to implement the control system in real-time. The two objectives, which are often competing with each other, will be optimized in a multi-objective fashion to reveal the design trade-offs.

**Control performance**

In this study, performance is measured based on the miss-distance, that is how close the missile intercepts the target \[19\]. In addition to the miss-distance, the evolution of the states can also be qualitatively observed to see how well does the controller guides the missile in intercepting the target.

An engagement scenario can be numerically simulated by integrating the engagement model (12) controlled by the MPC law (19) with the OCP (18) under disturbance in the measured states (17). A set of simulations will be performed to obtain the distribution of the performance. For one simulation, the miss-distance as a metric for control performance can be calculated based on the relative trajectory of the missile and the target \( s \) for a time period \( T_s \), given
an initial state $x_0$:

$$\Delta(x_0, p) := f_{\Delta}(s(\tau))$$

for $\tau \in [0, T_s]$,  
s.t. dynamics (12) and control law (19)  
for a given $x(0) = x_0$.

Here, $f_{\Delta}()$ calculates the metric for control performance of the miss-distance, defined as

$$f_{\Delta} := s_{\text{sgn}} \left( \min_{\tau \in [0, T_s]} \|s(\tau)\| \right).$$

The simulation time $T_s$ should be large enough to allow for the missile to reach the target. An indicative lower bound of this time would be $\|s(0)\|/\|v(0)\|$. The multiplier $s_{\text{sgn}} := \text{sgn}(s_{M_2}(T_s) - s_{T_2}(T_s))$ indicates the relative position of the missile when it reaches the target, differentiating between the missile passing ‘above’ or ‘below’ the target. The average value of the distributed miss-distance is defined as

$$\overline{\Delta}(x_0, p) := \text{mean}(\|\Delta(x_0, p)\|)$$

with a standard deviation denoted as $\sigma_{\Delta}(x_0, p)$.

**Required computational capacity**

Model-predictive controllers typically has a high level of computational load, an issue accentuated by the fast sampling rate required applications with fast dynamics such as missile control. In cases where the implementation design is not known a priori, the upper-bound on required computational capacity can only be assumed, potentially leading to either a sub-optimal design that is too conservative or a design that could not meet both performance and computational requirements. Consequently, the required computational capacity for the implementation hardware – that predominantly determines the implementation cost – is an important consideration alongside closed-loop performance in the design of MPC systems.

The required computational capacity can be measured based on the time taken by a processor to solve the OCP (18). Since the performance of the controller is simulated, the obtained measurements are reflective of the hardware used for the simulation, namely the *simulation hardware*. Let $\Gamma(p)$ be the upper-bound of this time for the simulation hardware, whose value is governed by the design parameter $p$. The simulation hardware is to be differentiated from the *implementation hardware*, on which the controller will be implemented on board of the missile.

The solution time depends on the solver algorithm that is used to solve the OCP. In this study, the nonlinear OCP (18) is solved using a sequential quadratic program (SQP) algorithm [20, 21]. The routine performs a number of quadratic programming (QP) optimization to iteratively approach the nonlinear OCP solution. QP optimization is known to be a P problem, given that the cost function is positive definite, so that is solvable in polynomial time as its size increases [22]. The size of the QP subproblem is directly associated with the number of unknown variables in the optimization, which is directly proportional to the number of prediction steps $N$. Consequently, the solution time $\Gamma_{\text{QP}}$
of the QP subproblem as a quadratic program can be upper-bounded by a polynomial with respect to $N$ (as validated later in Fig. 5).

**Assumption 1 (QP solution time upper-bound).** The upper-bound on solution time for the QP subproblem is monotonically increasing with the number of prediction steps $N$, modeled by a polynomial of degree $n$,

$$\Gamma_{QP}(p) := \sum_{i=0}^{n} a_i N^i$$  \hspace{1cm} (21)

for some constants $a_i$, $i \in \{0, \ldots, n\}$.

The SQP algorithm to solve the nonlinear OCP (18) performs a number of QP optimization iterations, and the number of QP iterations performed is upper-bounded by a specified iteration limit $\bar{t}_{SQP}$. Therefore, the solution time $\Gamma_{SQP}$ associated with the problem can be upper-bounded by a polynomial of $N$, that is an integer multiple of the upper-bound (21) on the QP iteration solution time.

**Proposition 1 (SQP solution time upper-bound).** From Assumption 1, the solution time of a QP can be upper-bounded by a polynomial of degree $n$ with respect to $N$, $\Gamma_{QP}(p) := \sum_{i=0}^{n} a_i N^i$ for some constants $a_i$, $i \in \{0, \ldots, n\}$. The SQP algorithm for the OCP (18) performs a number of QP iterations, and the iteration count is limited by $\bar{t}_{SQP}$. Therefore, the upper-bound on the SQP solution time can be modeled by

$$\Gamma_{SQP}(p) := \bar{t}_{SQP} \sum_{i=0}^{n} a_i N^i$$

**Remark 2.** Since the number of QP iteration is limited by $\bar{t}_{SQP}$, the solution obtained might possibly be only sub-/locally optimal solution. The use of the sub-/locally optimal solutions, obtained with even as little as one QP iteration, whilst guaranteeing real-time numerical stability – so that the error between the obtained and true theoretical solution of the OCP remains bounded – has been proven [23, 24, 25]. Specifically, the QP iteration limit $\bar{t}_{SQP}$ can be adjusted to arbitrarily set the error tolerance that the obtained solution is guaranteed convergence into. For example, the SQP algorithm has been shown to exhibit linear convergence with increasing number of QP iterations [20]. This is subject to the initial guess being close enough to the true solution, which can be guaranteed by hot-starting the algorithm [21] that involves the use of the solution at the previous sampling instant as the initial guess at the current instant, taking advantage of the receding horizon principle and the fact that two successive OCPs are numerically similar.

The non-dimensional Capacity Number [16] can be used to indicate the required resources/computational power of the implementation hardware in multiples relative to the simulation hardware. This metric is given by

$$\eta(p) := \Gamma_{SQP}(p)/h.$$  \hspace{1cm} (22)

The Capacity Number is a tuning objective that will be minimized. Based on the choice of $p$, minimizing the required computational capacity is associated with the reduction of the cost required for the control hardware implemented onboard the missile.
B. Tuning parameters

This study focuses on the parameters of the control system algorithm that also dictates the control implementation design, considered as structural MPC parameters. Consequently, these parameters affect both tuning objectives of control performance and required computational capacity. Amongst these parameters, the sampling time $h$ and number of prediction steps $N$ are the underlying and interrelated parameters that are to be tuned for the control system.

The sampling time dictates how often the state of the missile is sampled/measured and a new control action is commanded. The sampling time therefore has a significant impact on control performance; a shorter sampling time (increased sampling rate) generally improves performance as the states of the plant are measured more frequently. However, the relationship between sampling and performance is not completely straightforward and is nominally non-monotonic [26]. Not only does the sampling time determine the frequency of the control action, it also also sets an upper-bound on the time taken to solve the OCP to produce the new control command. As a result, the required computational capacity fundamentally depends on the sampling time. The smaller the sampling time, the more capable the hardware has to be for the control system to be able to produce control commands in time. Furthermore, sampling time affects the numerical conditioning of the OCP when it is discretized in order to be solved numerically. However, common solvers such as interior-point methods can be observed to be insensitive to sampling time of the OCP [16]. This is such that the time taken $\Gamma$ to produce the solution is independent on the sampling time. Nonetheless there are some OCP numerical solvers, for example first-order methods, that are sensitive to the conditioning of the problem [27].

The number of prediction steps $N$ governs how long into the future the plant is modeled when calculating the control command. Along with the sampling time, $N$ dictates the length of the prediction horizon $T := Nh$, within which constraints can be applied to the predictive model to help ensure that the plant is operated inside a desired envelope. Generally, a longer prediction horizon leads to better control performance, even though the exact sensitivity of performance to prediction steps is not straightforward [26]. Like the sampling time, the number of prediction steps affects the required computational capacity for the implementation of MPC. A larger prediction horizon equates to a larger OCP optimization associated with more unknown variables, making it more complex and longer to solve.

C. The multi-objective optimization problem

Sampling time and prediction horizon length are the two parameters that will to be tuned in this study,

$$\mathbf{p} := (h, N) \in \mathcal{P} := \mathbb{R} \times \mathbb{N}_{>0}.$$ 

This leads to the competitive pairing between the two tuning objectives of control performance and required computational capacity such that, within some competing design set, optimization of one objective via a design change in the parameter space comes at the compromise of the other.

**Proposition 2.** The objective functions of the MPC tuning problem $\Delta(x_0, \cdot)$ and $\eta(\cdot)$ are competing in some competing set $\mathcal{P}_c \subseteq \mathcal{P}$ so that $\mathbf{p} \mapsto \Delta(x_0, \mathbf{p})$ is monotonically increasing and $\mathbf{p} \mapsto \eta(\mathbf{p})$ is monotonically decreasing, or vice versa, for all $\mathbf{p} \in \mathcal{P}_c$. 

11 of 21

American Institute of Aeronautics and Astronautics
Proof. A shorter sampling time in or longer prediction horizon can potentially improve closed-loop control performance (decrease $\Delta$) as the control receives more frequent feedback and capture more future dynamics of the plant, respectively. However, the faster sampling rate and/or longer prediction horizon required more computational load (increase $\eta$) as per (22) with Assumption 1. Let $\mathcal{P}_+ := \{(h, N) | \Delta(h, N) < \Delta(h, N), \eta(h, N) > \eta(h, N)\}$ or $\Delta(h, N) < \Delta(h, N), \eta(h, N) > \eta(h, N)\}$ for all $h_- < h$ and $N_+ > N$, i.e. the set for which shortening sampling time or increasing prediction horizon result in a decrease $\Delta$ and increasing $\eta$. The reverse is also true, a delayed sampling time and shorter prediction horizon worsens performance and reduces computational burden, associated with $\mathcal{P}_-$. There thus exist some set $\mathcal{P}_c := \mathcal{P}_+ \cup \mathcal{P}_-$ so that the two objective functions are competing. □

The MOD-MPC problem can be formulated as the following

$$\mathcal{P}_*(\mathcal{P}_s) := \arg\min_{\mathcal{P}_s} \ell(p)$$

s.t. $p \in \mathcal{P}_s$. (23a)

(23b)

m-min is used to denote a multi-objective optimization problem that is distinguished from a single-objective problem (e.g. (18)). The two objectives are contained in the objective vector $\ell := (\Delta, \eta)$. $\mathcal{P}_s$ defines the search space and $\mathcal{P}_*$ defines the set of the design parameters that have an optimal trade-off, that is, the Pareto optimal solution. It is assumed that the search space intersects the competing set in Proposition 2.

Assumption 2. $\mathcal{P}_* = \mathcal{P}_s \cap \mathcal{P}_c \neq \emptyset$.

The Pareto front $\mathcal{L} := \{\ell(p)|p \in \mathcal{P}_*\}$ gives the optimal trade-off relationship between the two objectives. This set contains Pareto optimal points that are non-dominated. That is, a point $\ell(p_*)$ where $p_* \in \mathcal{P}$ is a Pareto optimal in $\mathcal{P}$ if it is non-dominated so that there does not exist another design choice $p \in \mathcal{P}$ such that $\ell(p) < \ell(p_*)$.

The Pareto front $\mathcal{L}$ establishes the optimal trade-off relationship between the objectives of the design problem. From the Pareto solution, a number of insights can be drawn to assist with an effective tuning of MPC. First and foremost, the curve defines the best control performance that can be achieved for a range of computational capacity. Subsequently, a practitioner can select a particular design on the curve to achieve a specified performance, with the information on the minimum computational capacity of the hardware for the control implementation. By extension, the minimum cost for the hardware is can also be obtained. Other conclusions that can be made based on the Pareto front include the gradient of the curve which tells how much needs to be paid in terms of computational capacity for a unit increase in performance.

To evaluate an objective point $\ell(p)$ given the design parameter $p$ a set of closed-loop simulations of the controlled plant needs to be performed. Multiple simulations are done to obtain a more representative measure of performance, for example an average value under disturbance. This motivates the use of an optimizer\(^a\) that is both accurate and fast in finding the optimal solution to reduce the time required to obtain the Pareto front of interest.

---

\(^a\)To establish clarity and consistency, the term optimizer is associated with the multi-objective design optimization whilst solver is used for the OCP optimization throughout the thesis.

---

12 of 21

American Institute of Aeronautics and Astronautics
V. Results

A. Preliminaries

Missile specifications

The missile used in this study is a tail-controlled axisymmetric ogive-nosed missile that is flying at a cruising altitude of 6,100 m (20,000 ft). Fig. 3 shows the shape of the missile along with the dimensions proportional to the diameter $D$ of the missile. The physical, actuation and flight specifications of the missile are given in Table 1.

The models for aerodynamic lift and moment (7) of the missile is obtained from DATCOM [28]. Lift and moment coefficients in (8) are modeled as polynomials,

$$C_L = 9.0126\alpha + 46.4585\alpha^3 + 3.2390\delta,$$

$$C_M = -20.1395\alpha - 51.7438\alpha^3 + 843.4588\alpha^5 - 24.1742\delta.$$

The missile geometry and aerodynamic model is adapted from [11, 18].

Engagement simulation: Initial conditions, constraints and disturbance

The engagement considered is an illustrative missile defence scenario where the target is a hypersonic cruise missile. Note that the specified scenario is a particularly challenging engagement chosen to represent the high requirements in advanced missile control. The initial conditions represent the final stages of the engagement where the autopilot is employed. The missile is initially positioned at the origin, traveling horizontally at Mach 2.5. The target is positioned at 5000 m crossrange and 300 m downrange traveling at Mach 5, representing the speed of a typical cruise missile, on the crossrange towards the missile. At the given speeds, the missile has to accelerate and adjust its direction in only

![Figure 3. Missile dimensions.](image)

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D$</td>
<td>diameter</td>
<td>0.2286</td>
<td>m</td>
</tr>
<tr>
<td>$S$</td>
<td>surface area</td>
<td>0.04088</td>
<td>m$^2$</td>
</tr>
<tr>
<td>$m$</td>
<td>mass</td>
<td>204</td>
<td>kg</td>
</tr>
<tr>
<td>$I$</td>
<td>moment of inertia</td>
<td>247.4</td>
<td>kg m$^2$</td>
</tr>
<tr>
<td>$\xi$</td>
<td>Actuator damping ratio</td>
<td>0.7</td>
<td>-</td>
</tr>
<tr>
<td>$\omega_a$</td>
<td>Undamped natural frequency</td>
<td>150</td>
<td>rad/s</td>
</tr>
<tr>
<td>$V_s$</td>
<td>speed of sound</td>
<td>315.9</td>
<td>m/s</td>
</tr>
<tr>
<td>$V$</td>
<td>missile speed</td>
<td>2.5$V_s$</td>
<td>m/s</td>
</tr>
<tr>
<td>$P$</td>
<td>dynamic pressure</td>
<td>2100000</td>
<td>Pa</td>
</tr>
</tbody>
</table>

Table 1. Physical, actuation and flight specifications of the missile.
about 2 seconds before it passes the target. The acceleration of the target is at $5g$ towards the initial missile location and all dynamic states are initialized at zero. This is illustrated in Fig. 4.

The constraint sets $\mathcal{X}$ and $\mathcal{U}$ in (16) that bound the predictive states and inputs are set with

$$\pi = (20^\circ, 90^\circ, 500^\circ/s, 40^\circ, 500^\circ/s, 40^\circ, \infty, \infty, \infty, \infty, \infty),$$

$$\pi = 500^\circ/s.$$  

The disturbance $d$ in (17) is modeled to arise from imperfections of the LOS angle $\lambda$ sensor readings. That is,

$$d_\lambda = U[-0.02, 0.02]. \quad (24)$$

Where the subscript $\lambda$ indicates that the disturbance acts only on the LOS rate. The disturbance is time-varying and changes at each sampling instant (regardless of the sampling period length).

**Engagement simulation: OCP solver**

In this study, simulations are done in MATLAB on a desktop computer with specifications as outlined in Table 2. The interior point (barrier) method from the Gurobi solver [29] in MATLAB is used in this study to solve each QP iteration in the SQP algorithm. A representative result for the relationship between the QP solution time and number of prediction steps $N$ for a range of sampling time $h$ is shown in Fig. 5. The results are obtained for 10 trials. It is shown that the average solution time for each QP solved in each of the 10 trials $\Gamma_{QP}$ is generally increasing with $N$ with mostly a linear relationship. This is consistent with the polynomial solvability of a QP given in [22] and supports Assumption 1.

The time to solve the OCP depends on the number of QP iterations performed by the SQP algorithm, which can be upper-bounded. In the simulations of this study, the QP iteration limit is set at $\bar{i}_{SQP} = 15$. From the graph, the model

<table>
<thead>
<tr>
<th>Intel® Core™ i7-3770 Processor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cores (available for simulation)</td>
</tr>
<tr>
<td>Cycle frequency/core</td>
</tr>
<tr>
<td>Operations/cycle</td>
</tr>
<tr>
<td>FLOP/s (available for simulation)</td>
</tr>
</tbody>
</table>

Table 2. Simulation hardware specifications [30].
for QP solution time is chosen as \( \Gamma_{QP} = (0.8N + 4) \times 10^{-3} \) as shown on the solid line in fig. 5 such that

\[
\Gamma_{SQP}(p) = 15(0.8N + 4) \times 10^{-3}
\]

as per Proposition 1.

**Controller tuning: MOD-MPC optimizer**

The numerical optimizer used for optimizing the multi-objective tuning problem (23) is the Dividing Triangles (DITRI) algorithm, an approach based on Lipschitzian optimization that is global, discrete and relies on continuity of the tuning objective functions. The full details of the algorithm are presented in [16]. The algorithm is shown to be effective and efficient in solving the MOD-MPC problem that performs a focused search and is guaranteed to converge to the solution for a given search space [16]. This is important as each evaluation of an objective point involves several closed-loop simulations of the plant and can be very time consuming.

**B. Offline controller tuning**

The formulated MOD-MPC problem with the solution of an optimal design set is solved to demonstrate a multi-objective tuning of the proposed missile control system. The numerical algorithm used is DITRI, motivated by its effectiveness and efficiency, with a chosen bounded search space of \( P_s = \{(h, N)| h \in [0.005, 0.03], N \in [6, 15]\} \).

Fig. 6 shows the solution of the MOD-MPC problem with the disturbance (24). In the figure, evaluation points are plotted as dots with larger, bold dots indicating Pareto optimality. Subfigure (a) shows the resulting solution in the objective space, making up the trade-off curve between performance, \( \overline{\Delta} \) (black) or \( \overline{\Delta} + \sigma_{\Delta} \) and required resource \( \eta \). Subfigures (b-c) show the associated solution in parameter \( p \) space for both the mean (b) and deviated (c) performance metrics. Subfigures (d-e) show the accompanying plot showing evaluation points in normalized space and triangle divisions used internally in DITRI. Total number of evaluations is 20. A trade-off curve is obtained after 20 evaluation points in DITRI. There are 7 different designs in the resulting optimal design set, from which the practitioner can choose from. For example, the practitioner can choose Design 14 with a sampling time of around 25 ms and 12
prediction steps. Based on the result, the average miss distance would be around 1.5 m. From the value of Capacity Number $\eta$, this particular design choice would require around 8 times the speed achieved by the simulation hardware.

The overall required computational capacity for the obtained design choices is considerably large, ranging from a Capacity Number of around $\eta = 5.5$ to 9. This highlights the high computational cost associated with MPC, particularly as a nonlinear MPC is considered in this study. To implement the controller in real-time, a number of techniques that can be used to address the high computational cost by increasing implementation hardware capacity, including the use of a compiled programming language, parallel-processing, an increased clock-frequency, as well as pipelining the implementation architecture.

A plot (shaded) showing the miss-distance $1\sigma$ away from the mean is given to accompany the main result. This can be used to explore Pareto optimal designs when smaller distributions of miss-distance (performance robustness under disturbance) are prioritized more relative to nominal miss distance. As expected, the associated trade-off curve closely resembles that of the mean, but shifted up in miss-distance (Fig. 6(a)). To competitively minimize this metric (instead of the mean) smaller values of prediction steps $N$ are now Pareto optimal, whilst optimal sampling period values $h$
remain similar as before (Fig. 6(c)). This concludes that a reduced prediction horizon length is more competitive in optimizing the required computational load than control performance, when performance is measured by $\overline{\Delta} + \sigma_\Delta$.

Finally, it is observed that the resulting trade-off curve flattens at just over 1 m average miss-distance after around Capacity Number $\eta = 6$. Performance peaks at around this point, after which an increase in computational capacity does not significantly improve control performance. Such an insight provides a guidance in the necessary cost to implement the controller.

C. Simulation result

A simulation is carried out to demonstrate how the proposed model-predictive integrated autopilot and guidance (iMAG) steers the missile to engage a target. For the simulation, Design 14 from Fig. 6 is chosen, with sampling time of $h = 25$ ms and the horizon length at $N = 12$ prediction steps. The cost weights are chosen as

\[ Q = P = \text{diag} \left( 0_{10}, 5 \times 10^3 \right), \quad R = 1 \times 10^{-6}. \]

![Figure 7. Evolution of kinematic and dynamic states. $h = 0.025$, $N = 12$.](image-url)
First, a case of zero disturbance is considered. Fig. 7 shows the evolution of the dynamic and kinematic states of the missile during the engagement. In the figure, the black lines are associated with the proposed iMAG, and the gray lines with an augmented proportional navigation (APN) guidance and an ideal autopilot\(^{b}\). Within the first second, the system is mostly transient, with angle-of-attack \(\alpha\) at the constraint of \(20^\circ\). Fin deflection \(\delta\) is quite actively controlled as the missile accelerates upwards (positive \(\ddot{s}_{M2}\)) to regulate LOS rate \(\dot{\lambda}\) quickly. After around 1 second, the missile is at a steady state before intercepting the target.

The proposed control strategy has a fixed prediction horizon (20) that is shortened at the final stages of the engagement so that the prediction does not exceed the intercept point. As a result, at the point of shortening, in this case just before 2 seconds, there is a slight transient as the prediction horizon length is shortened to not predict beyond the instant of intercept. Intercept happens at around \(t = 2.1\) s, as shown by the path of the missile and target. A large deviation in LOS rate \(\dot{\lambda}\) is observed at the final stages of the engagement. This is due to the fact that as the missile and target gets closer, the LOS second order rate \(\ddot{\lambda}\) deviates significantly as discussed earlier and shown in (13).

For the considered simulation, it is observed that the missile intercepts the target with near zero miss-distance under no disturbance. When subject to disturbance, the miss-distance value is distributed over a set of simulations. Fig. 8 plots the histogram of miss-distance values for 100 simulations for the disturbance (24) as formulated above. In the figure, the darker bars are associated with the proposed iMAG, and the lighter bars with an APN and ideal autopilot. Note that nonzero miss distance can still result in a successful engagement as anti-missiles in missile defense typically carry a warhead, which allows for the missile to incapacitate the target for a sufficiently small nonzero miss-distance [31].

The main advantages of a model-predictive integrated missile control over a separated guidance-autopilot system are demonstrated by the LOS rate and acceleration plots in the second bottom row of Fig. 7, as well as Fig. 8. The plots are accompanied by the result obtained using APN with an assumed ideal autopilot, which can track commanded accelerations perfectly. These advantages are

1. \textit{Look ahead.} Using the prediction from the model-based iMAG, the missile corrects its path as soon as the en-

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{histogram.png}
\caption{Histogram of miss-distances over a set of 100 simulations.}
\end{figure}

\(^{b}\)With an ideal autopilot, any commanded acceleration is assumed instantaneously realized, hence there is no evolution of dynamic states simulated.
gagement starts and undergoes large accelerations (Fig. 7). During the second half of the engagement period, the missile is mainly in constant velocity with minimal acceleration. At this stage, the missile is in a ‘neutral’ state and ready for any change in target acceleration obtained from the seeker. In conventional separated guidance-autopilot systems (e.g. with APN guidance) the missile has a nonzero acceleration throughout the engagement period, even under ideal conditions of no autopilot lag, so that it is at no time fully ready for changes in target maneuver. This is consistent with results in [10].

2. **Zero acceleration lag.** In conventional guidance systems (e.g. APN), discrepancies between the guidance command and lagged actual acceleration achieved by the autopilot in separated systems can cause an undesirable growth in commanded acceleration as intercept is approached [32]. Furthermore, in these systems, commanded acceleration is directly proportional to LOS rate which grows unbounded as intercept is approached (see (13)). Consequently, the presence of acceleration lag in tracking would cause a deterioration in missile accuracy. Fig. 8 shows that APN, even with ideal autopilot, generally fails to achieve zero miss-distance. The missile hits consistently over the target as the guidance fails to accelerate the missile early and over-steers/over-compensates at the end of the engagement. The clearly observed bias is also a result of the severity of the scenario tested (the need for ~200 m vertical ascent in about ~2 s) – where a non-zero miss distance is observed even with zero applied disturbance. Over less demanding test scenarios, the miss distance bias for the APN will be reduced.

**VI. Conclusions**

This paper first proposed a model-predictive integrated missile control. The model-predictive control system is aimed to improve control performance by commanding optimal accelerations. It was shown that by being able to predict future kinematics and dynamics of the engagement, the control pushes the missile to be more responsive than when a conventional separated guidance-autopilot system was used. Furthermore, the integrated design circumvents control lag present when the guidance-autopilot subsystems are implemented separately, improving intercept accuracy under disturbance. Secondly, multi-objective tuning of the proposed control system was demonstrated by using a design approach that considers not only control performance but also implementation cost, based on required computational capacity, which is an important design consideration in model-predictive control. Tuning of a nonlinear MPC was considered, thus extending existing results in [16] for when a linear MPC was used.

The result indicates that implementation of the proposed controller requires resources that are considerably more capable than that used for the simulation. This was not unexpected knowing that MPC is computationally heavy. Retrospectively, the paper presented a design approach to assist a practitioner in designing the model-based missile controller with both performance and required computational capacity in mind, and was not intended to proof practical feasibility of the proposed controller. The obtained results indicated what computational capacity is required to be able to implement the proposed controller, which is useful for the practitioner when designing the MPC for real-time implementation. The results also showed an insight around the trade-off between the two objectives upon which the practitioner can base his/her decisions on, for instance the best performance for a given computational capacity and the sensitivity between the two competing objectives of performance and cost.
References


