Optimal guidance of a homing missile

Francesco Bernardini
Dept. of Electrical Engineering
University of Houston
Houston, Texas
fbernard@CougarNet.UH.EDU

Hannah Vu
Dept. of Electrical Engineering
University of Houston
Houston, Texas
htvu25@CougarNet.UH.EDU

Manal Taroua
Dept. of Mechanical Engineering
University of Houston
Houston, Texas
mtaroua@CougarNet.UH.EDU

***Abstract*—We present a quantitative approach for the choice of the LQR parameters of an optimal guidance problem. The procedure accounts for boundedness of the control and for the finite size of the pursuer-target system.**

***Index Terms*—Optimal control, Missile guidance, Riccati equation, Lyapunov equation, Bounded controls, LTI systems.**

I. Introduction

In the past decades, intense efforts have been put in place by worldwide governments towards the development of techniques and devices aimed at the detection and neutralization of air or marine threats. Under these efforts, extremely sophisticated systems such as the U.S. SAM MM-104 "Patriot" or the French-Italian SAMP-T complex [1] have been produced. Although the idea of inflicting damage using ballistic projectiles can be dated back to the Greek catapults or the Chinese trebuchets around the 4^{th} century BC, the very first proposal for the endowment of artillery with some form of guidance was presented in 1870 by Werner von Siemens [2]: in his letter to the Prussian ministry of war, he explicitly mentioned "guided torpedoes", whose purpose was the destruction of enemy vessels. Later, this idea evolved in what is now known as **proportional navigation** (PN), and was further refined under the aegis of *optimal control theory*, giving rise to the branch of **optimal guidance** (OG).

Guidance theory literature is extremely vast, and certainly not limited to military applications, however in this paper we will focus on one specific problem, the *homing missile guidance*: in this application, a pursuer has the final goal of colliding with a target at some final time T; additionally, the pursuer can count on some level of knowledge about the trajectory of the target.

Among the techniques that can be used in order to compute the trajectory of a homing missile, *optimal guidance* (OG) makes use of optimal control theory, in particular of the *linear quadratic regulator* (LQR). This method needs parameters to be initialized, and usually those are chosen on the basis of mission and performance requirements. As it is well known from the theory of operational amplifiers, adding external constraints such as the saturation of the controls generally spoils the linearity of the problem [3] [4], therefore it is desirable to devise alternate methods to impose such constraints, in order to preserve linearity and its computational benefits. In this paper we propose a quantitative approach that accounts for the saturation of the control and for physical characteristics of the missile-target system such as the size of the target. This method intrinsically preserves linearity as it postprocesses data which are obtained solving the LQR problem.

This paper is structured as follows: in section II we will introduce the homing problem and define the physical model that describes its dynamics, briefly introducing proportional navigation and formulating the corresponding Optimal Guidance problem. In section III we will present and comment the results of our simulations, while in section IV we will discuss possible future developments.

II. The homing missile problem

By definition [5], a homing missile is a weapon that once detected a target, is able to adjust its trajectory in order to collide with it at some future time. This goal is reached practically by means of sophisticated sensor systems interconnected by suitable feedback loops (the so called **guidance loop** [2]) and complicated hardware and software logic, but this paper focus only on the computational aspects of the problem. Before delving into the details of how the right trajectory for the missile is chosen, we will review the kinematics of the problem.

A. Kinematics of collision

Let us consider two pointlike objects P and T, respectively the pursuer and the target: in a 3-dimensional space they are completely specified by a pair of vectors $(\vec{r}, \vec{v}) \in \mathbb{R}^3 \times \mathbb{R}^3$ specifying their position and velocity, but it is customary in guidance theory to apply the following set of assumptions in order to simplify the problem:

- Ultimately collision between two pointlike objects is a planar (i.e. 2-dimensional) problem, so we will assume that the motion of both the pursuer and the target is entirely contained in a fixed plane.
- We consider the distances between the pursuer and the target to be always small enough to neglect any curvature effect of the Earth.

These two assumptions already reduce the problem to a 2-dimensional one. In addition, we also require that

- The **velocity** of the target (i.e. the vector $\vec{V}_T = V_T \hat{V}_T$) is constant, i.e. the target moves along a straight line with constant speed.

- The **speed** of the pursuer (i.e. the magnitude V_P of the vector $\vec{V}_P$) is constant, i.e. we allow a *centripetal acceleration* a_P for the pursuer, always orthogonal to its instant velocity, whose magnitude therefore is unaltered by the maneuver.

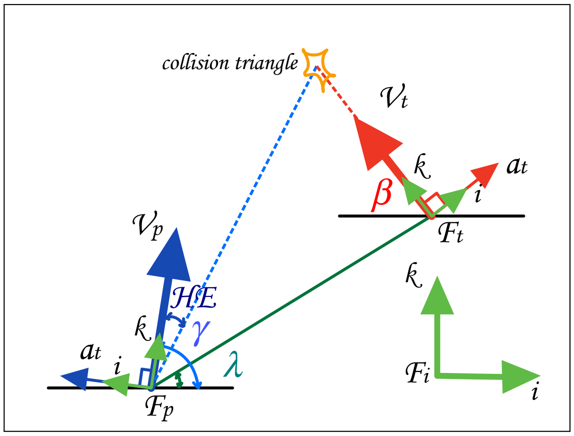

Fig. 1. Kinematic parameters for the pursuer-target system: the accelerations of both the target and the pursuer are assumed to be orthogonal to the instantaneous direction of the velocities, so that their magnitude does not change.

Implicitly, we will always assume that the magnitude V_P of the velocity of the pursuer is large enough to allow a collision to happen. Now that the problem has been constrained to a plane we can define some additional parameters and concepts (see fig. 1):

- The **horizon**: a horizontal line parallel to the ground.
- The **inertial frame** F_i: a coordinate frame at rest with the Earth, where Newton's equations can be considered valid. Quantities expressed in this frame are labeled by an index i. For simplicity only the x and z axes are shown, with unit vectors $\hat{i}$ and $\hat{k}$, respectively. The target and the pursuer positions will be described by a pair of coordinates $P^i(t) = (x_P, z_T)$ and $T^i(t) = (x_T, z_T)$.
- The **target frame** F_t: a coordinate frame at rest with the target. Quantities expressed in this frame are labeled by an index t.
- The **pursuer frame** F_p: a coordinate frame at rest with the pursuer. Quantities expressed in this frame are labeled by an index t.
- The **flight path angle** β **for the target**: the angle formed by the velocity of the target and the horizon.
- The **flight path angle** γ **for the pursuer**: the angle formed by the velocity of the of the pursuer and the horizon.
- The **line of sight** (LOS): the line that connects the pursuer and the target.
- The **LOS angle** λ: the angle between the LOS and the horizon.
- A sufficient condition for a collision to occur is the instatement of the so called **collision triangle**: in its simplest representation both pursuer and target move on a constant line (fig. 2) and the LOS angles $\lambda(t)$ at different instants are all equal. In this case the two trajectories form a continuous set of similar triangles that at any time t have as vertices the two instantaneous positions $P(t)$ and $T(t)$ and the future point of collision.
- The **lead angle** L: the angle that the pursuer should keep from the LOS angle to instate a collision triangle.
- The **heading error** HE: the difference between the actual direction of the pursuer and the lead angle.

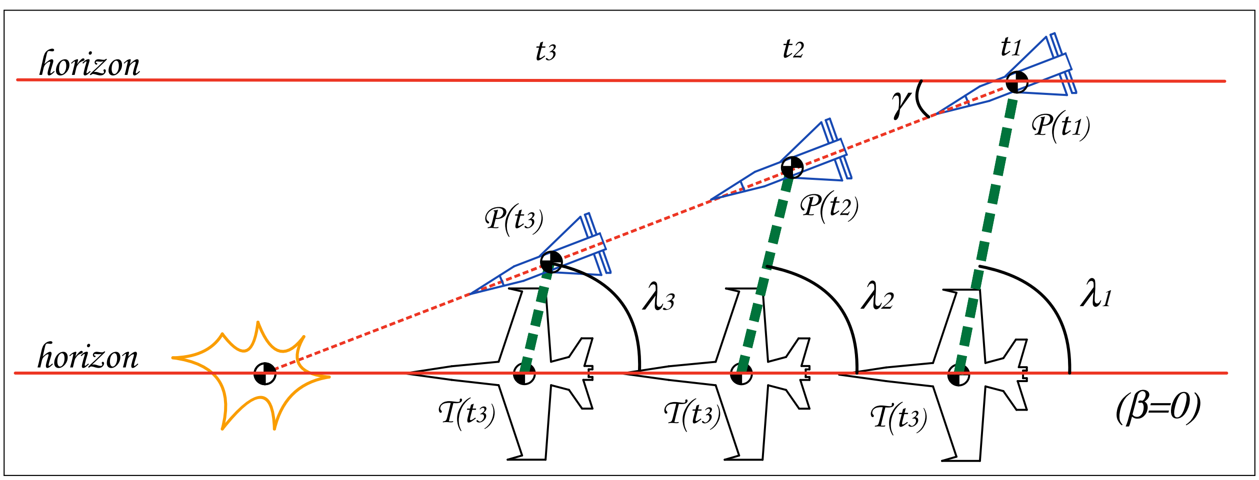

Fig. 2. The collision triangle: if along the trajectory the LOS angle $\lambda(t)$ remains constant (i.e., all the LOS remain parallel to one another), then a collision will necessarily occur. In this figure, the flight path angle β is constantly zero, as the target moves along the horizon.

B. Proportional navigation

The collision triangle is at the very basis of the definition of the proportional navigation equation: if the pursuer and the target trajectories are not forming a collision triangle, it means that the LOS angle λ is changing with time at some rate $\dot{\lambda}$. The defining principle of proportional navigation is that in order to reinstate a collision triangle, the rate of change of the flight path angle γ of the pursuer must be proportional to $\dot{\lambda}$:

$$\dot{\gamma} = N\dot{\gamma} \tag{1}$$

As we can control the direction of movement of the pursuer, we can control the rate of change of γ, therefore (1) can be translated in a sequence of commands that we can impart on the fins or canards of a missile in order to steer it of the amount required to track down its target. It can be shown that $N = 1$ does not give rise to a collision as the collision triangle is very slowly (asymptotically) instated, therefore we talk about a *pursuit* problem. The most common values of N that give rise to collision are between 2 and 5, with $N = 3$ representing an optimal value in terms of amount of control imparted. Higher values of N might still bring to the instatement of a collision triangle, but impart much larger steering angles to the pursuer in the initial moments, and this might be undesirable.

Proportional navigation can also be expressed in terms of the accelerations of the target and of the pursuer, giving rise to the following system of ODEs:

$$\begin{cases} \dot{\beta} = \frac{a_T^t}{V_T^t} \\ \dot{R}_{Tx}^i = -V_T^t \cos\beta \\ \dot{R}_{Tz}^i = V_T^t \sin\beta \\ \dot{V}_{Tx}^i = a_T^t \cos\beta \\ \dot{V}_{Tz}^i = a_T^t \sin\beta \end{cases} ; \begin{cases} \dot{R}_{Px}^i = V_{Px}^i \\ \dot{R}_{Pz}^i = V_{Pz}^i \\ \dot{V}_{Px}^i = -a_P^p \sin(\lambda + L + HE) \\ \dot{V}_{Pz}^i = a_P^p \cos(\lambda + L + HE) \end{cases} \tag{2}$$

Due to the presence of the trigonometric functions, this system is nonlinear and it cannot be solved in closed form, therefore it needs to be integrated numerically in order to determine a solution for the acceleration a_P to impart the pursuer. However, in order to acquire more insight about the problem and to enable the application of other methods such as state space controls, we seek to linearize this system. It can be shown that under the assumption of

- *small angles*, i.e. $\beta, \gamma, \lambda < 10°$;
- Velocities of target $\vec{V}_T$ and of pursuer $\vec{V}_P$ practically horizontal, antiparallel and constant;

and defining the following quantities:

- $z(t) = z_T(t) - z_P(t)$: the difference in quote between the target and the pursuer;
- $x(t) = x_T(t) - x_P(t)$: the horizontal distance between the target and the pursuer;
- $t_f = \frac{x(0)}{V_T + V_P}$: the **final time**, at which the target and the pursuer would collide if they were flying head on;
- $t_{go} = t_f - t$: the **time-to-go**, i.e. the difference between the final time and the time elapsed since the homing procedure was initiated;
- $z(t = t_f)$: the **miss distance**;
- A state vector $\vec{z} = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} z \\ \dot{z} \end{pmatrix} =$

then the dynamic equation of the system pursuer-target simplify as:

$$\begin{cases} \dot{z}_1 = z_2 \\ \dot{z}_2 = a_T - a_P \end{cases} \tag{3}$$

As it is, (3) contains a control $u = a_P$ and a term a_T that accounts for a possible acceleration of the target, and therefore is not controllable.

Considering the latter term would imply to reabsorb it in a third state variable z_3, and analyze a third order system. In order to keep the problem simpler, we assume that the target is not maneuvering, therefore $a_T = 0$ and the system assumes exactly a state-space form:

$$\begin{cases} \dot{z}_1 = z_2 \\ \dot{z}_2 = -a_P \end{cases} \tag{4}$$

from which it follows that

$$A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \ B = \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \ u = a_p \tag{5}$$

$$\Rightarrow \dot{\vec{z}} = A\vec{z} + Bu \tag{6}$$

We have thus reduced the problem to a linear time-invariant (LTI) system, that we can analyze by means of the LQR method.

C. Optimal guidance

Proportional Navigation provided us with a well defined equation for the acceleration to impart to the missile, which allows for a redefinition of the problem in terms of optimal control theory, where the initial system of ODE is replaced by a performance index built according to some performance criterion or mission requirement. Let $\vec{z}(t) \in \mathbb{R}^n$ be the state of our system, $\vec{u}(t) \in \mathbb{R}^m$ the control we apply, and $t \in [0, t_f]$ the time interval of our interest: then, the performance index can be written in general as a functional of this form

$$J = \Phi(\vec{z}(t_f), t_f) + \int_0^{t_f} L(\vec{z}(t), \vec{u}(t); t)dt \tag{7}$$

where $\Phi(\vec{z}(t_f), t_f)$ is the so called *penalty function*, that accounts for the final state of the system, and $L(\vec{z}(t), u(t); t)$ is the *loss function*, which weights the behavior of the state and of the control. The system dynamics is assumed to be described by an equation of the form

$$\dot{\vec{z}} = f(\vec{z}, u; t) \tag{8}$$

which in general is nonlinear. In the case of LTI systems and on penalty/loss functions which are quadratic forms in x and u, the problem takes the name of **linear quadratic regulator** (LQR) and has the following explicit expression:

$$J = \frac{1}{2}\vec{z}^T(t_f)S_f\vec{z}(t_f) + \int_0^{t_f} \left[\frac{1}{2}x^T(t)Qx(t) + \frac{1}{2}u^T(t)Ru(t)\right] dt \tag{9}$$

where $S, Q \in \mathbb{R}^{n\times n}$ are positive semidefinite matrices and $R \in \mathbb{R}^{m\times m}$ is a positive definite matrix. The dynamics of the system is

$$\dot{\vec{z}} = A\vec{z} + B\vec{u} \tag{10}$$

where $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ are the usual state-space representation matrices of the system. Under the assumption of a stabilizable system, the optimal control u^* that minimizes (9) while complying with the constraint of (10) is provided by

$$u^*(t) = -K(t)\vec{z}(t) \tag{11}$$

where K, the **Kalman gain** is given by

$$K(t) = R^{-1}B^T S(t) \tag{12}$$

where $S(t)$ is the time-dependent **kernel matrix** which is solution of the Riccati differential equation:

$$-\dot{S} = SA + A^T S - SBR^{-1}B^T S + Q \tag{13}$$

with the initial condition $S(t_f) = S_f$

This equation, quadratic in S, can be solved numerically in order to determine the kernel matrices $S(t)$ that allow to compute the optimal control.

D. From Riccati to Lyapunov equation

Equation (6) has the same form as (10), therefore we can apply the methods presented so far. However, as we are not interested in regulating the trajectory of the missile before the collision, there is no Q matrix in our performance index:

$$J = \frac{1}{2}\vec{z}^T(t_f)S_f\vec{z}(t_f) + \int_0^{t_f} \frac{1}{2}Ru^2(t)dt \tag{14}$$

or in other words, Q can be chosen as the zero matrix, and this allows us to simplify further the problem. First of all we assume that $S(t)$ is invertible for every t in the interval of interest, and we multiply the left and right sides of (13) by $S^{-1}(t)$, the inverse of $S(t)$:

$$-S^{-1}\dot{S}S^{-1} = S^{-1}SAS^{-1} + S^{-1}A^T SS^{-1} - \tag{15}$$

$$-S^{-1}SBR^{-1}B^T SS^{-1} + S^{-1}QS^{-1}$$

Then, we use the fact that $S^{-1}S = \mathbb{I}\ \forall t$ therefore

$$\frac{d}{dt}\left(S^{-1}S\right) = \dot{S}^{-1}S + S^{-1}\dot{S} = \frac{d}{dt}\mathbb{I} = 0 \tag{16}$$

which means

$$\dot{S}^{-1}S = -S^{-1}\dot{S} \tag{17}$$

Plugging this back into the equation

$$-\left(S^{-1}\dot{S}S^{-1}\right) = -\left(\dot{S}^{-1}SS^{-1}\right) = AS^{-1} + S^{-1}A^T - \tag{18}$$

$$-BR^{-1}B^T + S^{-1}QS^{-1}$$

The advantage of this procedure is that we have attached the quadratic term in S to the Q factor, which being zero allows us to rewrite:

$$\dot{S}^{-1} = AS^{-1} + S^{-1}A^T - BR^{-1}B^T \tag{19}$$

which is a simpler *Lyapunov equation* for S^{-1}, with initial condition $S^{-1}(t_f) = S_f^{-1}$. Once $S^{-1}(t)$ is found, it can be inverted in order to obtain finally $S(t)$.

III. Simulation and analysis of the results

The simulation we performed consisted in several repetitions of the LQR method, collecting a selection of data relevant for the subsequent analysis. The data which we considered as the objective of our analysis were:

- The maximum control applied during a trajectory for a given choice of the LQR parameters:

$$|u|_{max} \equiv \max_{t\in[0,t_f]} |u(t)| \tag{20}$$

- The final distance z_f obtained for a given choice of the LQR parameters:

$$z = z(t = t_f) \tag{21}$$

- The integral of the control $u(t)$ along the trajectory:

$$U = \frac{1}{2}\int_0^{t_f} |u|(t)dt \tag{22}$$

As no information about the structural limitations of the missile is built in the LQR procedure, the resulting trajectories that minimize the performance index may in general produce controls which are larger than some u_{max}. Such controls are not applicable in practice, and whenever a trajectory requires them, it should be discarded. Similarly, both the missile and the pursuer have a finite volume, and we can tolerate a nonzero final distance $z(t_f)$ as long as it is within some kind of effective volume V_{eff} which we approximate as a circle of radius z_{max} around the target. As long as a trajectory delivers a finite distance $z(t_f) < z_{max}$, it can be considered as valid.

Applying these two ideas to the simulation we have been able to produce two independent landscapes, both function of the choices of the LQR parameters. Their intersection, if nonzero, represents the only choices of the LQR parameters which are admissible for the mission. If we add also the requirement to minimize the integral of the control along the trajectory, the choice of the parameter narrows down even further. This is the main result of our analysis.

A. Breakdown of the code

The Matlab code provided is so structured:

1) The main function `Final_project.m` collects the user-defined parameters of the simulation, such as the state-space parameters of the system, the range of the LQR parameters to explore N_{steps}, and the initial conditions $[z(0), \dot{z}(0)]$.
2) `ComputeControl` computes the solution of a single LQR problem, in the form of five objects:
   - `MaxControl`: the maximum absolute value of the control imparted.
   - `ControlIntegral`: the integral of the control effort.
   - `zFinal`: the final vertical distance.
   - Two boolean values `zFlag` and `uFlag` that inform if that choice of parameter respects the system constraints.

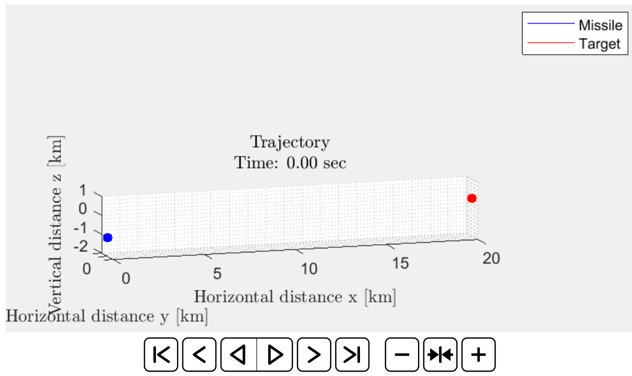

Fig. 3. Animation of the trajectories of the pursuer (blue) and the target (red), corresponding to the choice $(s_{zz}, s_{\dot{z}\dot{z}}) = (1, 1)$.

- An array `xzplot` that contains the coordinates of the target and the missile, for the plot of their trajectories. The number of rows of the array is not fixed as it depends on the number of timesteps that the function `ode45` takes to integrate the problem. However, the number of columns is fixed to 5: x_m, z_m, x_t, z_t, t.

3) The function `ComputeAllControls` calls the function `ComputeControl` $N_{steps} \times N_{steps}$ times, storing the results in two multi-dimensional arrays `uResults` of dimension $N_{steps} \times N_{steps} \times 5$ and `xzplots`, a collection of variable-size `xzplot` items.
4) The function `ComputeSurface` expresses the results of the parameter sweep as a 3D surface $\sigma(s_{zz}, s_{\dot{z}\dot{z}})$: this allows to analyze the trends by simple visual inspection.
5) The function `ComputeSurfaceTresholds` computes the same surfaces, but uses a binary palette: a surface element is red if its values do not respect the system constraints, and green if they do.
6) The function `PlotAllTrajectories` plots all the $N_{steps} \times N_{steps}$ trajectories obtained from the parameters sweep.
7) The function `CreateAnimatedGif` selects one of the $N_{steps} \times N_{steps}$ trajectories (in our code, the first) and creates an animated gif representing the motion of the missile and the target.

B. Simulations

In the simulations we considered a non maneuvering target and the following set of initial conditions:

$$\begin{cases} x_P(0) = 0 \\ z_P(0) = 11000m \\ x_P(0) = 20000m \\ z_P(0) = 12000m \\ V_T = 500m/s \\ V_P = 500m/s \end{cases} \Rightarrow \begin{cases} z_1(0) = 1000m \\ \dot{z}_1(0) = 0 \end{cases} \tag{23}$$

These conditions imply a time to go $t_{go} = \frac{20000m}{1000m/s} = 20s$.

For what concerns the LQR parameters:

- R was kept constant and equal to 1;
- N_{steps} was chosen as $N_{steps} = 20$, therefore $(s_{zz}, s_{\dot{z}\dot{z}})$ swept the interval $[1, 20] \times [1, 20] \in \mathbb{Z}^2$. We did not explore values that are smaller than R because we assumed that the mission requirements **will always privilege trajectory over energy consumption**.

A total of 400 trajectories have been computed and analyzed, and an animation of that corresponding to $(s_{zz}, s_{\dot{z}\dot{z}}) = (1, 1)$ is provided in fig. 3.

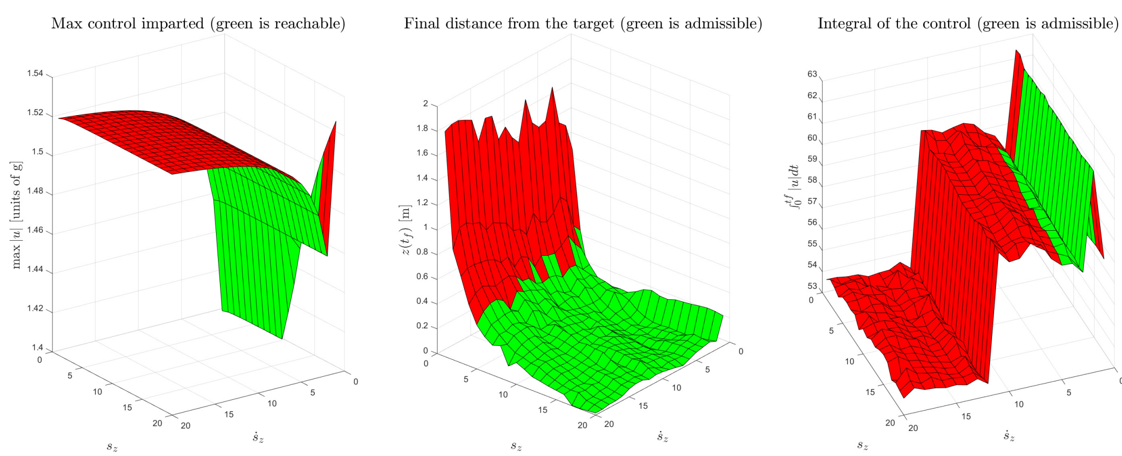

Fig. 4. Surfaces generated by the simulation spanning $(s_{zz}, s_{\dot{z}\dot{z}}) \in [1, 20] \in \mathbb{Z}^2$. The green regions represent the admissible set for what concerns the maximum acceleration (left), the maximum size of the target (center) and the intersection of the two applied to the integral of the control (right).

For what concerns the constraints over the control and over the final distance, we set two fictitious thresholds at:

$$\begin{cases} u_{max} = 1.5g \;\; (g = 9.81m/s^2) \\ z_{max} = 0.5m \end{cases} \tag{24}$$

As forecasted, not all trajectories satisfied the constraints: fig. 4 presents the results of the analysis in terms of choices of parameters that satisfy or not the thresholds. The results showed that for this choice of initial conditions and thresholds, there existed a nonzero intersection between the admissible sets of the two constraints: applying the last criterion of minimization of the integral of the control, we arrive finally to the optimal choice of parameters which was $(s_{zz}, s_{\dot{z}\dot{z}}) = (19, 3)$ (fig. 5).

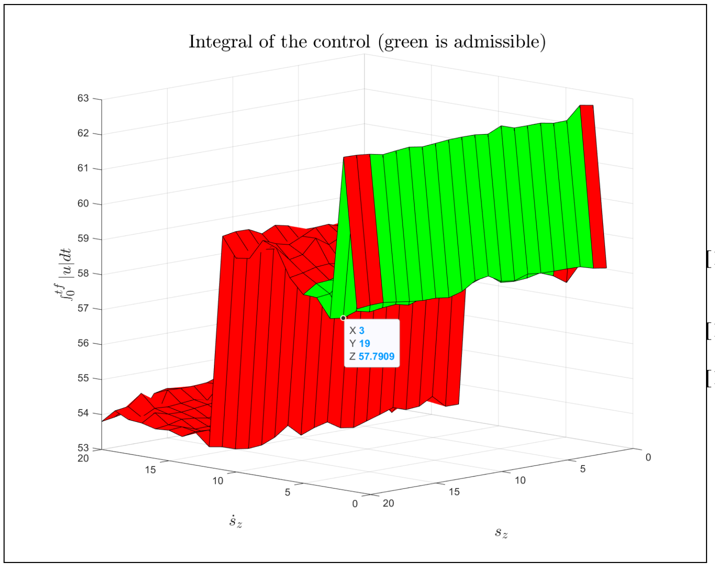

Fig. 5. Optimal choice of LQR parameters resulting from the application of the procedure.

IV. Conclusions and future work

In this paper we presented a procedure for the choice of the LQR parameters on the basis of structural characteristics of the missile-target system, and on mission-specific requirements. All the results of the computation can be stored in suitable lookup tables, saving the time needed for the explorative search during an actual mission.

Due to time constraints, this analysis was limited to the simplest case of non-maneuvering target (2-dimensional state), but it may be extended in the following directions:

- Including the transverse accelerations of the target;
- Optimizing the code and the algorithm in order to work on an embedded computer and to perform the choice of parameters automatically;
- Considering a broader range of initial conditions, or specific values of constraints, proper of actual models of missiles and targets.
- Comparing the results of this approach with more refined (and possibly non-linear) methods that include the constraints as part of the formalism, and in case of appreciable agreement, compare the respective computation speeds.

References

[1] "Top 5 best anti-aircraft missile systems in the world", https://www.youtube.com/watch?v=L47-RHdWVh0

[2] Shneydor, N. A. (1998). *Missile guidance and pursuit: kinematics, dynamics and control*. Elsevier.

[3] Hu, T., Lin, Z. (2001). Control systems with actuator saturation: analysis and design. Springer Science & Business Media.

[4] Lewis, F. L., Vrabie, D., Syrmos, V. L. (2012). Optimal control. John Wiley & Sons.

[5] Palumbo, N. F., Blauwkamp, R. A., Lloyd, J. M. (2010). Modern homing missile guidance theory and techniques. Johns Hopkins APL technical digest, 29(1), 42-59.

[6] Siouris, G. M. (2004). *Missile guidance and control systems*. Springer Science & Business Media.

[7] He, S., Lee, C. H., Shin, H. S., Tsourdos, A. (2020). *Optimal Guidance and Its Applications in Missiles and UAVs*. Berlin/Heidelberg, Germany: Springer.

[8] Liang, H., Jianying, W. A. N. G., Yonghai, W. A. N. G., Linlin, W. A. N. G., Peng, L. I. U. (2020). *Optimal guidance against active defense ballistic missiles via differential game strategies*. Chinese Journal of Aeronautics, 33(3), 978-989.

[9] Qilong, S. U. N., Naiming, Q. I., Zheyao, X. U., Yanfang, L. I. U., Zhang, Y. (2017). *An optimal one-way cooperative strategy for two defenders against an attacking missile*. Chinese Journal of Aeronautics, 30(4), 1506-1518.

[10] Liang, H., Li, Z., Wu, J., Zheng, Y., Chu, H., Wang, J. (2022). *Optimal Guidance Laws for a Hypersonic Multiplayer Pursuit-Evasion Game Based on a Differential Game Strategy*. Aerospace, 9(2), 97.

[11] "MBDA SAMP/T anti-air system with Aster 30 missiles" https://www.youtube.com/watch?v=ciqS3bor2u4

[12] Dickinson, B. T. – *Optimal guidance* https://www.youtube.com/playlist?list=PLcmbTy9X3gXsh-o1W60E7rEA35igNj__q