

# Requirements Derivation for RoboCup Small Size League Robot <sup>\*</sup>

Gabriel B. Farias<sup>\*</sup> Marcos R. O. A. Máximo<sup>\*\*</sup>  
Rubens J. M. Afonso<sup>\*\*\*,\*</sup>

<sup>\*</sup> *Electronic Engineering Division, Aeronautics Institute of Technology,  
São Paulo, Brazil*

*(e-mails: gabrielbandeirafarias@gmail.com, rubensjm@ita.br)*

<sup>\*\*</sup> *Autonomous Computational Systems Lab (LAB-SCA), Computer  
Science Division, Aeronautics Institute of Technology, São Paulo,  
Brazil*

*(e-mail: mmaximo@ita.br)*

<sup>\*\*\*</sup> *Institute of Flight System Dynamics, Technical University of  
Munich, Bayern, Germany  
(e-mail: rubens.afonso@tum.de)*

**Abstract:** This work develops a method for deriving requirements for the goalkeeper of the robot soccer competition RoboCup Small Size League (SSL) using Monte Carlo simulation. Initially, an overview of the SSL competition is presented and related works are shown. Then, the parameters of interest are selected and the developed method is discussed. Afterwards, different models and control laws are designed to simulate the goal defense performance for different parameter values. Finally, the data generated is analyzed and a set of requirements for the mobile robot is selected. Lastly, the method utility is evaluated and possible extensions of this work are proposed.

*Keywords:* Requirement Derivation; RoboCup Small Size League; Monte Carlo simulation; Mobile robot.

## 1. INTRODUCTION

The RoboCup Small Size League (SSL) is a robot soccer competition that focuses on advancing state-of-the-art research in robotics, including multi-agent cooperation and dynamic systems control (RoboCup, 2020). The robots must have up to 0.18 m of diameter and 0.15 m of height and are allowed to have dribbling and kicking devices. An external computer receives updates from an external camera and selects diverse tasks to the robots to perform, such as shooting the ball, following a trajectory, and intercepting the ball. Finally, to execute these tasks, the computer calculates and sends commands of wheel rotation speed to each robot. Then, the robots use an embedded control law to steer the present rotation speed of each wheel to the respective commanded value. An example of this scheme is shown in Figure 1.

However, the influence of the robot dynamics in the performance of these tasks is not always clear. Thus, a study of this relationship is necessary. Kress-Gazit et al. (2018) study a framework for the synthesis of robot controllers that guarantees to execute complex tasks, and Pinheiro

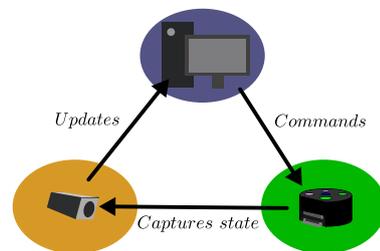


Figure 1. Overview of the interfaces between computer, robot and camera.

et al. (2016) derive requirements for the goalkeeper and attacker robots of the IEEE Very Small Size Competition.

Requirement derivation is a concept of systems engineering that aims at producing engineering specifications that ensure the fulfillment of a set of user requirements. Watson et al. (1967) developed a methodology for requirement derivation focused on space missions. Bonfè et al. (2012) leads a study on automated surgical robots and applies requirement engineering to obtain specifications for the robots.

The present work uses Monte Carlo simulations (Hanson and Beard, 2010) to derive robot requirements based on the performance of the task of defending shots towards the goal by the goalkeeper of an SSL team. Due to the complexity of the SSL competition, a hard requirement of minimum probability of defense cannot be enforced.

<sup>\*</sup> Rubens Afonso acknowledges the support of CAPES (fellowship proc. # 88881.145490/2017-01) and the Federal Ministry for Education and Research of Germany through the Alexander von Humboldt Foundation. Marcos Maximo and Gabriel Farias acknowledge ITAndroids' sponsors: Altium, Cenic, Intel, ITAEx, Mathworks, Metinjo, Micropress, Polimold, Rapid, Solidworks, ST Microelectronics, WildLife, and Virtual Pyxis.

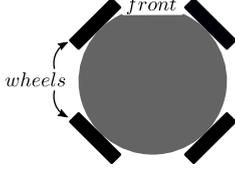


Figure 2. Typical robot for the SSL competition. Robot body (gray) and four wheels (black).

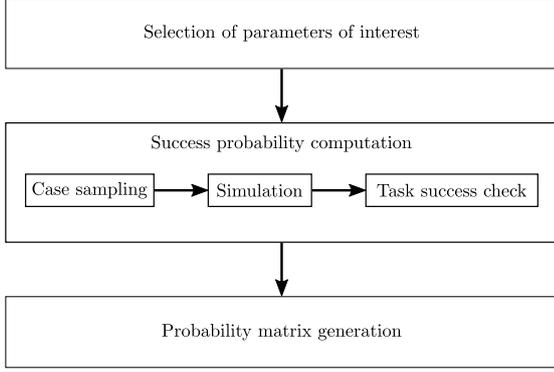


Figure 3. Flowchart illustrating the steps used during analysis.

Therefore, this work contributes with a method that analyses the given task, and proposes a set of parameters considering a tradeoff between performance and demand of the system.

A typical robot of this competition is shown in Figure 2. The front of the robot corresponds to the position of the kicking and dribbling device. These robots contain omnidirectional wheels that enable the movement in all three degrees-of-freedom simultaneously (translation and rotation in the plane). Similar robots are used in warehouse automation (Wurman et al., 2008), hazardous missions (Houshangi and Lippitt, 1999), and other applications (Adascalitei and Doroftei, 2011). Then, this method could be adapted to develop similar analyses for these tasks.

## 2. METHODOLOGY

Initially, the parameters of interest shall be determined based on the analyzed task and previous knowledge about the robot construction. Then, for each set of parameters, different models shall be designed to analyze the influence of each parameter in the desired task. Furthermore, Monte Carlo simulations are performed with different configurations to consider different game strategies and different initial conditions. Finally, the results of the Monte Carlo simulation shall be analyzed, and a set of parameter requirements is chosen. This process is illustrated in Figure 3.

### 2.1 Parameters selection

To avoid a combinatorial explosion of the parameters analyzed, a top-down approach was used by starting from a more simplistic scenario and then adding complexity to it to form a more specific one. In this work, the parameters of the first scenario are linear and angular velocities and

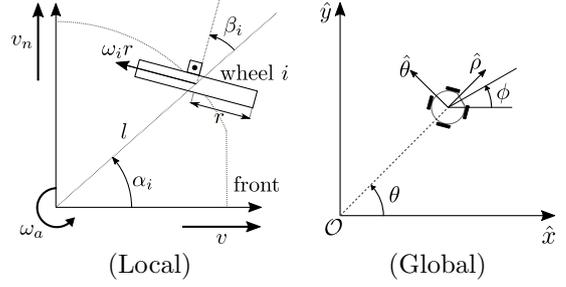


Figure 4. Local and global conventions used models derivation.

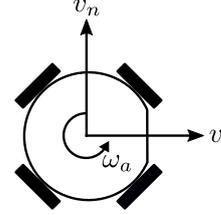


Figure 5. Kinematics of a SSL robot.

accelerations, while the parameters of the second scenario are the phase margin ( $PM$ ) and bandwidth ( $BW$ ) of the robot dynamics, which are common frequency-domain parameters used to characterize control systems. The  $BW$  is the frequency at which the system's gain response falls 3 dB from its value at 0 Hz, while the  $PM$  is the difference between the system's phase lag and  $-180^\circ$  at the cross-frequency (the frequency the gain is 0 dB). The  $BW$  is roughly linked to the speed of the response to changes in the input, whereas the  $PM$  is linked to its damping and to the stability margins regarding phenomena that introduce phase lag, such as delays (Ogata and Yang, 2010).

Initially, the robot dynamics were modeled as double integrators in each wheel with restrictions on velocity and acceleration. Later, the double integrator model was replaced by a underdamped second order model with the restrictions derived by the first scenario.

Assuming that the robot wheels do not slip in the rotation direction, a relationship can be found for the vector of wheel speed  $\boldsymbol{\omega} = [\omega_1 \ \omega_2 \ \omega_3 \ \omega_4]^T$  and the robot's kinematics  $\mathbf{v}_l = [v \ v_n \ \omega_a]^T$ , where  $v$  and  $v_n$  are the robot tangential and normal velocities and  $\omega_a$  is the robot angular velocity. This convention is illustrated in Figure 5.

Then, using the local conventions shown in Figure 4, the condition of no slip is (Pinheiro et al., 2019)

$$\boldsymbol{\omega} = \mathbf{M}\mathbf{v}_l, \quad (1)$$

where

$$\mathbf{M} = \frac{1}{r} \begin{bmatrix} -\sin(\alpha_1 + \beta_1) & \cos(\alpha_1 + \beta_1) & l \cos(\beta_1) \\ -\sin(\alpha_2 + \beta_2) & \cos(\alpha_2 + \beta_2) & l \cos(\beta_2) \\ -\sin(\alpha_3 + \beta_3) & \cos(\alpha_3 + \beta_3) & l \cos(\beta_3) \\ -\sin(\alpha_4 + \beta_4) & \cos(\alpha_4 + \beta_4) & l \cos(\beta_4) \end{bmatrix}, \quad (2)$$

and  $l$  is the robot radius;  $r$  is the radius of the wheels;  $\alpha_i$  is the angle formed by the wheel  $i$ , the robot center, and the robot front;  $\beta_i$  is the inclination angle of the wheel  $i$ . Besides, if  $\det(\mathbf{M}^T \mathbf{M}) \neq 0$ , then

$$\mathbf{v}_l = \mathbf{M}^+ \boldsymbol{\omega}, \quad (3)$$

where  $M^+ = (M^T M)^{-1} M^T$  is the Moore-Penrose pseudo-inverse of  $M$ .

*Double integrator model* An initial approximation is the use of a double integrator model for each wheel. This model assumes direct control over the wheel acceleration, which although not feasible, provides useful insight of the ideal behavior. This model is described by the following differential equation

$$\dot{\omega}(t) = \mathbf{a}_{ref}^\omega(t), \quad (4)$$

where  $\mathbf{a}_{ref}^\omega(t)$  is the commanded acceleration of the wheels. Then, choosing  $\mathbf{a}_{ref}^\omega(t) = M \mathbf{a}_l^{ref}(t)$ , where  $\mathbf{a}_l^{ref}(t)$  is the desired robot acceleration, and left multiplying (4) by  $M^+$ , the robot dynamics become

$$\dot{v}_l(t) = \mathbf{a}_l^{ref}(t). \quad (5)$$

Including the velocity and acceleration restrictions, the robot model becomes

$$\dot{v}_l(t) = \mathbf{a}_l^{ref}(t), \quad (6a)$$

$$\text{s.t.: } |v(t)|^2 + |v_n(t)|^2 \leq v_{max}^2, \quad (6b)$$

$$|\dot{v}(t)|^2 + |\dot{v}_n(t)|^2 \leq a_{max}^2, \quad (6c)$$

$$|\omega_a(t)| \leq \omega_{max}, \quad (6d)$$

$$|\dot{\omega}_a(t)| \leq a_{max}^\omega, \quad (6e)$$

where  $v_{max}$ ,  $a_{max}$ ,  $\omega_{max}$ , and  $a_{max}^\omega$  are the parameters of interest.

*Underdamped second order model* Now, modeling the robot dynamics, an expression for  $\omega$  is found as (Pinheiro et al., 2019)

$$\dot{\omega} = \mathbf{A}_c^\omega \omega + \mathbf{B}_c^\omega \mathbf{V} \quad (7)$$

where  $\mathbf{A}_c^\omega, \mathbf{B}_c^\omega \in \mathbb{R}^{4 \times 4}$ ,  $\mathbf{V} = [V_1 \ V_2 \ V_3 \ V_4]^T$ , and  $V_i$  is the voltage applied to the motor corresponding to the wheel  $i$ . Then, assuming a PI control law to drive  $\omega$  to a constant reference  $\omega_{ref} = [\omega_1^{ref} \ \omega_2^{ref} \ \omega_3^{ref} \ \omega_4^{ref}]^T$ ,  $\mathbf{V}$  becomes

$$\mathbf{V} = K_p(\omega_{ref} - \omega) + K_i \int (\omega_{ref} - \omega) dt, \quad (8)$$

where  $K_p, K_i \in \mathbb{R}$ . Then, differentiating (8) with respect to time

$$\dot{\mathbf{V}} = -K_p \dot{\omega} + K_i(\omega_{ref} - \omega). \quad (9)$$

Now, differentiating (7) with respect to time and using (9), the robot dynamics becomes

$$\ddot{\omega} = (\mathbf{A}_c^\omega - K_p \mathbf{B}_c^\omega) \dot{\omega} - K_i \mathbf{B}_c^\omega \omega + K_i \mathbf{B}_c^\omega \omega_{ref}. \quad (10)$$

Approximating (10) into four single-input single-output (SISO) systems (i.e. neglecting the coupling between the wheels) (Sherback et al., 2006), it can be written as

$$\ddot{\omega} + 2\xi\omega_n \dot{\omega} + \omega_n^2 \omega = \omega_n^2 \omega_{ref}, \quad (11)$$

where  $\xi$  and  $\omega_n$  are, respectively, the damping ratio and the natural frequency of the system. Finally, left multiplying (11) by  $M^+$  and using (3), the following equation is found

$$\ddot{v}_l + 2\xi\omega_n \dot{v}_l + \omega_n^2 v_l = \omega_n^2 v_l^{ref}, \quad (12)$$

where  $v_l^{ref} = M^+ \omega_{ref}$ , and, by including velocity and acceleration restrictions, the robot position model becomes

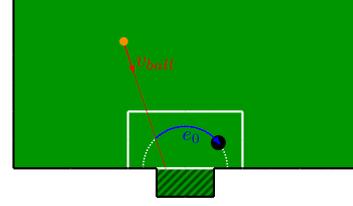


Figure 6. An example of sample case for the defense analysis.

$$\ddot{v}_l + 2\xi\omega_n \dot{v}_l + \omega_n^2 v_l = \omega_n^2 v_l^{ref}, \quad (13a)$$

$$\text{s.t.: } |v(t)|^2 + |v_n(t)|^2 \leq v_{max}^2, \quad (13b)$$

$$|\dot{v}(t)|^2 + |\dot{v}_n(t)|^2 \leq a_{max}^2, \quad (13c)$$

$$|\omega_a(t)| \leq \omega_{max}, \quad (13d)$$

$$|\dot{\omega}_a(t)| \leq a_{max}^\omega, \quad (13e)$$

where  $v_{max}$ ,  $a_{max}$ ,  $\omega_{max}$ , and  $a_{max}^\omega$  will be selected from the analysis of the double integrator model. Finally, the parameters of interest of this scenario are the  $PM$  the  $BW$ , which can be calculated post-analysis using the formulas (Ogata and Yang, 2010)

$$BW = \omega_n \sqrt{1 - 2\xi^2 + \sqrt{1 + (1 - 2\xi^2)^2}}, \quad (14a)$$

$$PM = \arctan \left( \frac{2\xi \sqrt{2(1 - 2\xi^2)}}{1 - 4\xi^2} \right). \quad (14b)$$

## 2.2 Case Sampling

The case of interest in this work consists of the interception of the ball after it has been shot by an adversary robot. The defending robot is considered to move in a trajectory of constant radius  $r_d$  in relation to the center of the goal. Two cases are considered to select  $r_d$ : (a) the first one considers  $r_d = 1 \text{ m}$ , which is the maximum value such that the robot remains in the defense area; (b) the other one considers  $r_d = 0.5 \text{ m}$ , which is the minimum value such that the robot covers the entire goal.

Then, a sample case ( $\mathcal{S}$ ) of the problem consists of an initial position error of the robot in relation to the interception point in the trajectory ( $e_0$ ), and the arriving time of the ball ( $t_{ball}$ ). An example of a sample case is shown in Figure 6. In this figure,  $v_{ball}$  represents the linear velocity of the ball, and  $e_0$  represents the position error of the robot right after the ball has been shot. In this work, the initial position of the ball is sampled using the random variable defined by

$$\mathbf{B} = [l \cos(\theta_{ball}) \ l \sin(\theta_{ball})]^T, \quad (15)$$

where  $l$  is uniformly distributed between  $r_{B1} = 2 \text{ m}$  and  $r_{B2} = 3 \text{ m}$  and  $\theta_{ball}$  is uniformly distributed between 0 and  $\pi$ . Then, the target position  $\mathbf{T}$  is sampled by using the random variable

$$\mathbf{T} = \mathbf{G}_1 + \alpha(\mathbf{G}_2 - \mathbf{G}_1), \quad (16)$$

where  $\mathbf{G}_1$  and  $\mathbf{G}_2$  are the corners of the goal and  $\alpha$  is a scalar uniformly distributed between 0 and 1. The ball velocity  $v_{ball} = 8 \text{ m/s}$  is considered constant in all cases and the friction loss is ignored.

The initial position of the robot when the shot occurs is given by

$$\mathbf{R} = [r_d \cos(\theta_0) \ r_d \sin(\theta_0)]^T, \quad (17)$$

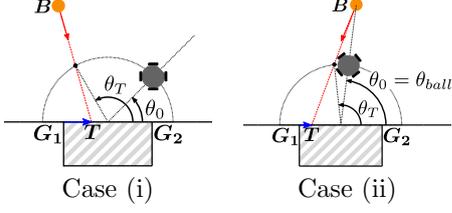


Figure 7. Examples of sampling used during analysis.

where  $\theta_0 = \theta(0)$ , and  $\theta(t)$  is the angle formed by the robot in relation to the origin of the goal ( $\mathcal{O}$ ) at instant  $t$ .

Two approaches are used to select the initial angle  $\theta_0$ : (i) the initial position of the robot is independent of the initial position of the ball and  $\theta_0$  is a uniformly distributed variable between  $0$  and  $\pi$ ; (ii) the robot is aligned with the ball right before the shot, then  $\theta_0 = \theta_{ball}$ . Case (i) does not assume any particular relationship between the position of the robot and the ball, and the results are expected to be more conservative in terms of the requirements. On the other hand, case (ii) assumes that the robot follows the ball and successfully positions itself to mitigate the chance of a goal, as a consequence, less demanding requirements are expected to be derived. The use of these two opposed “pessimistic” and “optimistic” cases aims at finding a compromise to define the requirements. Both cases are illustrated in Figure 7.

### 2.3 Simulation

Initially,  $\theta_T$  is defined as the angle formed the intersection point between the ball trajectory and the robot trajectory, the goal origin, and  $\mathbf{G}_2$ . Then, choosing one of the models previously proposed and a test case  $\mathcal{S}$ , the position error  $e(t) = r_d(\theta(t) - \theta_T)$  can be determined by including the conditions

$$\mathbf{p}(t) = r_d \hat{\mathbf{p}}(t), \quad (18a)$$

$$\dot{\mathbf{p}}(0) = \dot{\hat{\mathbf{p}}}(0) = \mathbf{0}, \quad (18b)$$

where  $\mathbf{p}(t) \in \mathbb{R}^2$  is the robot position,  $\hat{\mathbf{p}}$  and  $\hat{\boldsymbol{\theta}}$  are the polar coordinates, according to the global convention shown in Figure 4. Equation (18a) guarantees that the robot stays in the circular trajectory of radius  $r_d$  in relation to the goal origin  $\mathcal{O}$ , and (18b) is the resting initial condition (IC) of the robot. Then, differentiating (18a) with respect to time

$$\dot{\mathbf{p}}(t) = r_d \dot{\hat{\boldsymbol{\theta}}}(t) \hat{\boldsymbol{\theta}} \quad (19)$$

is a necessary condition for the robot to remain in the trajectory. To reduce the problem to one degree of freedom, we assumed that the robot front faces the normal direction of the trajectory. Then, defining  $\phi$  as the angle that the front of the robot forms with the  $\hat{\mathbf{x}}$  axis, and assuming  $\phi = \theta$ . Finally, using (19),  $\mathbf{v}_l$  becomes

$$\mathbf{v}_l = [v \ v_n \ \omega_a]^T = [0 \ r_d \dot{\theta} \ \ddot{\theta}]^T. \quad (20)$$

*Double integrator model* Using (5), the derivative of (20) with respect to time becomes

$$\dot{\mathbf{v}}_l = \mathbf{a}_l^{ref} = [0 \ r_d \ddot{\theta} \ \ddot{\ddot{\theta}}]^T. \quad (21)$$

Then, substituting (20) and (21) in (6)

$$r_d \ddot{\theta} = a_{ref}, \quad (22a)$$

$$\text{s.t.: } |r_d \dot{\theta}| \leq v_{max}, \quad |r_d \ddot{\theta}| \leq a_{max}, \quad (22b)$$

$$|\dot{\theta}| \leq \omega_{max}, \quad |\ddot{\theta}| \leq a_{max}^\omega, \quad (22c)$$

where  $a_{ref} \in \mathbb{R}$ . Since (22b) and (22c) differ only by a  $r_d$  term, the latter can be omitted and the constraints determined by

$$\omega_{max} = \frac{v_{max}}{r_d}, \quad (23a)$$

$$a_{max}^\omega = \frac{a_{max}}{r_d}, \quad (23b)$$

which reduce the number of analyzed parameters from 4 to 2. Finally, including the IC and rewriting the equations in terms of  $e(t)$

$$\ddot{e}(t) = a_{ref}(t), \quad (24a)$$

$$\text{s.t. : } e(0) = e_0, \quad \dot{e}(0) = 0, \quad (24b)$$

$$|\dot{e}(t)| \leq v_{max}, \quad (24c)$$

$$|a_{ref}(t)| \leq a_{max}. \quad (24d)$$

This model is intended to be used in an optimal defense reference when compared to the second order model scenarios, so the designed controller shall minimize the final position error. Therefore, an adequate control law that minimizes  $e(t_{ball})$  is described by the following relationship

$$a_{ref}(t) = \begin{cases} a_{max}, & \dot{e}(t) < v_{max} \text{ and} \\ & e(t) < -\Delta e_p(t), \\ -a_{max}, & \dot{e}(t) > -v_{max} \text{ and} \\ & e(t) > -\Delta e_p(t), \\ 0, & \text{otherwise,} \end{cases} \quad (25)$$

where  $\Delta e_p(t) = \dot{e}(t)(t_{ball} - t)$  is the expected error variation if the robot remains with the same velocity.

*Underdamped second order model* Similar to the double integrator model, using (20) in (13), the second order model becomes

$$r_d \ddot{\theta} + 2\xi\omega_n r_d \dot{\theta} + \omega_n^2 r_d \theta = \omega_n^2 v_{ref}, \quad (26a)$$

$$\text{s.t.: } |r_d \dot{\theta}| \leq v_{max}, \quad |r_d \ddot{\theta}| \leq a_{max}, \quad (26b)$$

where  $v_{ref} \in \mathbb{R}$ , and  $v_{max}$  and  $a_{max}$  are previously determined using the double integrator model. Then, including the IC and using  $e(t) = r_d(\theta(t) - \theta_T)$

$$\ddot{e}(t) + 2\xi\omega_n \dot{e}(t) + \omega_n^2 e(t) = \omega_n^2 v_{ref}(t), \quad (27a)$$

$$\text{s.t. : } e(0) = e_0, \quad \dot{e}(0) = 0, \quad \ddot{e}(0) = 0, \quad (27b)$$

$$|\dot{e}(t)| \leq v_{max}, \quad |\ddot{e}(t)| \leq a_{max}, \quad (27c)$$

$$|v_{ref}(t)| \leq v_{max}, \quad (27d)$$

where (27d) is included so that the controller does not command velocities higher than what is achievable by the robot.

Finally, an adequate feedback controller shall be used to steer the position error to zero. In this work, two controllers were developed for this task. The first controller was implemented using a minimum time control law (Kirk, 2004) and the second one was implemented using a proportional control law (Ogata and Yang, 2010).

*Minimum time control law* The first controller was chosen so that the best performance was achieved. Then, a control law that minimizes the time to arrive the final

state is derived by solving the optimization problem (Kirk, 2004)

$$v_{ref}(\cdot) = \arg \min t_f, \quad (28a)$$

s.t. : (27),

$$e(t_f) = 0, \dot{e}(t_f) = 0, \ddot{e}(t_f) = 0. \quad (28b)$$

Equation (28) requires a solution such that the final velocity and acceleration are zero. Although not required, this constraint is important to ensure that the robot reaches the position to intercept the ball in minimal time and remains there. Otherwise, the minimal time trajectory might yield a result that quickly achieves zero error, but at a high speed, which would make the robot overshoot and not be able to defend the shot. Then, (28) was solved by using the optimal control tool Falcon.m (Rieck et al., 2020) for MATLAB.

*Proportional control law* The second controller was chosen such that it is similar to the ones actually implemented in the robotic system. Then, modeling the robot as in (13), for each  $\xi$  and  $\omega_n$ , the proportional control law is

$$v_{ref}(t) = -k_p(\xi, \omega_n)e(t), \quad (29)$$

where  $k_p(\xi, \omega_n)$  is the solution of the optimization problem

$$k_p(\xi, \omega_n) = \arg \min E[e(t_{ball})^2], \quad (30)$$

where  $E[e(t_{ball})^2]$  is the expected value of  $e(t_{ball})^2$ , considering  $e_0$  and  $t_{ball}$  distributed as modeled in subsection 2.2, and  $e(t)$  modeled as in (27).  $E[e(t_{ball})^2]$  was computed using the following approximation

$$E[e(t_{ball})^2] = \sum_{\bar{e}_0 \in \mathcal{E}_0} \sum_{\bar{t}_{ball} \in \mathcal{T}_{ball}} e(\bar{t}_{ball})^2 p(\bar{e}_0, \bar{t}_{ball}), \quad (31)$$

where  $\mathcal{E}_0$  and  $\mathcal{T}_{ball}$  are representative sets of the continuous variables  $e_0$  and  $t_{ball}$ , and  $p(\bar{e}_0, \bar{t}_{ball})$  is the joint probability of the set  $\{\bar{e}_0, \bar{t}_{ball}\} \in \{\mathcal{E}_0, \mathcal{T}_{ball}\}$ .

Note that (27a) leads to the transfer function

$$G(s) = \frac{\omega_n^2}{s^3 + 2\xi\omega_n s^2 + \omega_n^2 s}, \quad (32)$$

which in turn leads to the closed-loop transfer function

$$T(s) = \frac{k_p \omega_n^2}{s^3 + 2\xi\omega_n s^2 + \omega_n^2 s + k_p \omega_n^2}, \quad (33)$$

that requires, from the Routh-Hurwitz criterion (Ogata and Yang, 2010), that  $k_p < 2\xi\omega_n$  to be stable.

#### 2.4 Task success check

To determine if the task was successful, the defense success function  $f : \mathcal{S} \mapsto \{Success, Failure\}$  is defined

$$f(\mathcal{S}) = \begin{cases} Success, & |e(t_{ball})| \leq tol, \\ Failure, & otherwise, \end{cases} \quad (34)$$

where  $tol = 0.04 m$  is a tolerance parameter selected by considering the maximum diameter of the robot (0.18 m).

### 3. RESULTS AND DISCUSSION

The first step of the analysis is to propose limits to the velocity and acceleration of the robot. These limits were produced by using the double integrator model to find regions such that the probability of defense reaches saturation. Figure 8 shows the probability of defense for

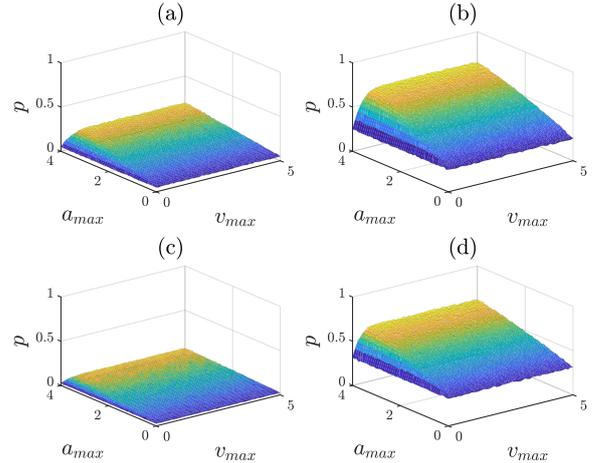


Figure 8. Probability  $p$  of defense for the double integrator model varying  $a_{max}$  and  $v_{max}$ .

the double integrator for the cases: (a)  $r_d = 0.5 m$  and  $\theta_0$  uniformly distributed between 0 and  $\pi$ ; (b)  $r_d = 0.5 m$  and  $\theta_0 = \theta_{ball}$ ; (c)  $r_d = 1 m$  and  $\theta_0$  uniformly distributed between 0 and  $\pi$ ; (d)  $r_d = 1 m$  and  $\theta_0 = \theta_{ball}$ .

The cases in which the robot is aligned with the ball ((b) and (d)) show higher defense probability than otherwise ((a) and (c)), as would be expected, since  $e_0$  tends to be smaller in absolute value. To a lesser extent, a smaller  $r_d$  ((a) and (b)) suggests a better performance when compared to the larger  $r_d$  cases ((c) and (d)), due to a higher  $t_{ball}$  and, usually, smaller  $e_0$ . Moreover, the defense probability seems to saturate for  $v_{max} > 1 m/s$ , while still increasing for  $a_{max}$ . However, (Purwin and D'Andrea, 2005) shows for a typical SSL robot that  $a_{max} > 4 m/s^2$  may not be achievable due to the limited friction between the wheel and the surface. Then, the velocity and acceleration requirements chosen are  $a_{max} = 4 m/s^2$  and  $v_{max} = 5 m/s$ , where the velocity requirement was selected as a value high enough to saturate but still under typical specifications (Maranhão et al., 2019). Then, the defense probability for each case of the double integrator model with chosen requirements is shown in Table 1. Furthermore,  $p \neq 0$  is observed even if  $v_{max}$  or  $a_{max}$  equals 0. This phenomenon also occurs in the other scenarios and happens because the robot may intercept the ball even if it stays still.

Then, using  $a_{max} = 4 m/s^2$  and  $v_{max} = 5 m/s$  in the second order model (27) with minimum time control law, Figure 9 shows a comparison of the defense probability for different values of natural frequency  $\omega_n$  and damping ratio  $\xi$ . Table 1 shows the maximum defense probability achieved in each case of this comparison. Since the minimum time control law is optimal, it manipulates the system such that every feasible acceleration can be achieved instantly by the robot. Then, the performance is virtually equal for every natural frequency and damping ratio. Besides, lower defense probabilities are observed for some cases when compared to the double integrator analysis. This is likely due to the constraint of zero velocity and acceleration at the final instant imposed by (28b). Nevertheless, the probability decrease is below 5% for all cases, suggesting

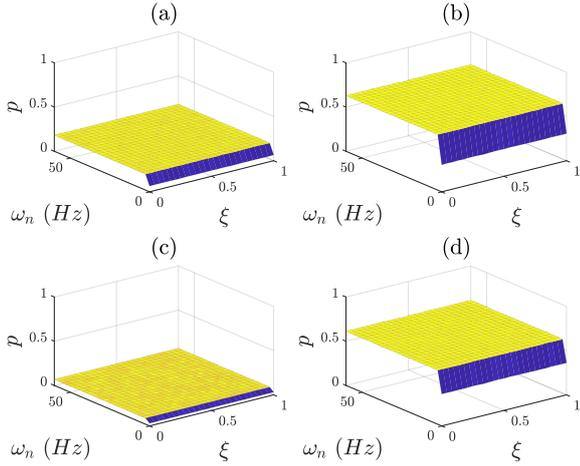


Figure 9. Probability  $p$  of defense for the underdamped second order model using the minimum time control law.

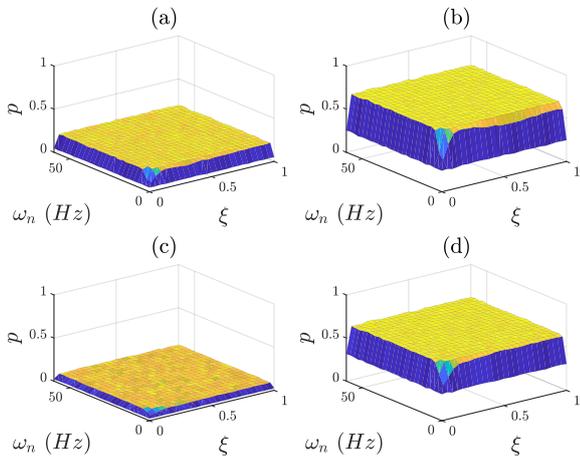


Figure 10. Probability  $p$  of defense for the underdamped second order model using the proportional control law.

Table 1. Probability defense of each scenario: (i) double integrator; (ii) minimum time control; (iii) proportional control.

Case	(i)	(ii)	(iii)
(a)	0.21	0.19	0.21
(b)	0.67	0.63	0.67
(c)	0.08	0.07	0.08
(d)	0.63	0.62	0.64

that this requirement does not excessively over-constrain the analyzed task.

Although the minimum time control law provides a promising result, it is usually not implementable, due to the high computational cost required to solve the optimization in real-time. A more practical control law is proportional control. Figure 10 shows the comparison for this control law, and Table 1 shows the maximum defense probability achieved in each case of the comparison. In particular, a higher defense probability was observed for the case (d) when compared to the double integrator. This

Table 2. Minimum robot requirements.

Parameter	Minimum value	Units
$v_{max}$	5	$m/s$
$\omega_{max}$	5	$rad/s$
$a_{max}$	4	$m/s^2$
$a_{max}^{\omega}$	4	$rad/s^2$
$PM$	33	$^{\circ}$
$BW$	22	$Hz$

effect is likely due to numeric imprecision but requires more investigation. Since the defense probability is mostly uniform, an achievable set of requirements were selected to compensate for unmodeled effects, such as camera time delay, controller discretization, and measurement noise of the sensors. Then,  $\omega_n > 15 Hz$  and  $\xi > 0.2$  were selected, and using  $r_d = 1 m$  to minimize the angular velocity and acceleration requirements, the gathered results are summarized in Table 2.

#### 4. CONCLUSION AND FUTURE WORK

The method developed in this work provides the robot designer with relevant information to the robot synthesis. Due to the competitive nature of the game, we avoided considering a minimum desired defense probability for the requirement derivation. Instead, the requirements herein derived propose to the design engineer a compromise between maximizing the defense probability and restricting the demand of the system to viable constructions.

Two possibilities to improve the calculated defense probability are: (i) to design a more complex test case generation since the current assumptions are too generic; (ii) to derive other requirements that are assumed ideal in the current analysis, such as time delay of the camera, control discretization, and measurement noise.

Finally, this method can be extended to different tasks of the SSL competition, such as trajectory tracking and shoot to the goal. Moreover, emerging areas can use this method for open problems, such as automated parking in autonomous driving systems, and the development of urban air mobility vehicles.

#### REFERENCES

- Adascalitei, F. and Doroftei, I. (2011). Practical applications for mobile robots based on mecanum wheels - a systematic survey. *Romanian Review Precision Mechanics, Optics and Mechatronics*, 21–29.
- Bonfè, M., Boriero, F., Dodi, R., Fiorini, P., Morandi, A., Muradore, R., Pasquale, L., Sanna, A., and Secchi, C. (2012). Towards automated surgical robotics: A requirements engineering approach. In *2012 4th IEEE RAS EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, 56–61.
- Hanson, J. and Beard, B. (2010). Applying monte carlo simulation to launch vehicle design and requirements analysis.
- Houshangi, N. and Lippitt, T. (1999). Omnibot mobile base for hazardous environment. In *Engineering Solutions for the Next Millennium. 1999 IEEE Canadian Conference on Electrical and Computer Engineering (Cat. No.99TH8411)*, volume 3, 1357–1361 vol.3.
- Kirk, D.E. (2004). *Optimal control theory: an introduction*. Courier Corporation.

- Kress-Gazit, H., Lahijanian, M., and Raman, V. (2018). Synthesis for robots: Guarantees and feedback for robot behavior. *Annual Review of Control, Robotics, and Autonomous Systems*, 1, 211–236.
- Maranhão, A., Schuch, A., Azevedo, A., Rodrigues, A., Lima, E., Sarmiento, J., Santos, M., Máximo, M., Sales, M., Dias, M., and Lima, R. (2019). Itandroids small size league team report 2019. URL <http://www.itandroids.com.br/en/publications/>.
- Ogata, K. and Yang, Y. (2010). *Modern control engineering*, volume 5. Prentice hall Upper Saddle River, NJ.
- Pinheiro, F.C.R., Máximo, M.R.O.A., and Yoneyama, T. (2019). Model predictive control for omnidirectional small size robot with motor and non-slipping constraints. In *2019 Latin American Robotics Symposium (LARS)*, 13–18.
- Pinheiro, F.C.R., Pinto, S.C., Okuyama, I.F., Máximo, M.R.O.A., and Viana, N.L. (2016). Performance requirements derivation for ieee very small size competition. In *2016 XIII Latin American Robotics Symposium*, 109–114.
- Purwin, O. and D’Andrea, R. (2005). Trajectory generation for four wheeled omnidirectional vehicles. In *Proceedings of the 2005, American Control Conference, 2005.*, 4979–4984 vol. 7.
- Rieck, M., Bittner, M., Grüter, B., Diepolder, J., and Piprek, P. (2020). FALCON.m User Guide. URL [www.falcon-m.com](http://www.falcon-m.com).
- RoboCup (2020). Rules of the RoboCup Small Size League, viewed 7 June 2020. URL <https://robocup-ssl.github.io/ssl-rules/sslrules.html>.
- Sherback, M., Purwin, O., and D’Andrea, R. (2006). Real-time motion planning and control in the 2005 cornell robocup system. In K. Kozłowski (ed.), *Robot Motion and Control*, 245–263. Springer London, London.
- Watson, H.S., Mulvihill, R.J., Reed, M.D., Klein, L.V., Graham, W.C., Dye, H.M., and Liberman, D.S. (1967). Introduction to the derivation of mission requirements profiles for system elements. Technical Report NASA-SP-6503, NASA, United States.
- Wurman, P.R., D’Andrea, R., and Mountz, M. (2008). Coordinating hundreds of cooperative, autonomous vehicles in warehouses. *AI magazine*, 29(1), 9–9.