

# Laboratory Report 4

## Longitudinal state-space control of the balancing robot

Trabucco Giovanni; email: [giovanni.trabucco@studenti.unipd.it](mailto:giovanni.trabucco@studenti.unipd.it)

Group: 2,2

Components: Chioccarello Stefano, Scola Davide,  
Serena Leonardo, Trabucco Giovanni

June 24, 2021

## 1 Introduction

### 1.1 Activity Goal

The goal of this laboratory experience was to design, implement and validate a longitudinal state-space controller for the two-wheeled balancing robot (also referred as “two-wheeled inverted pendulum robot”, or “Segway-like robot”) available in laboratory. The aim of the controller is to be able to stabilize the robot body to its upward vertical position while keeping the robot base in a specific longitudinal position set-point, namely if the robot gets moved or shifted it should be able to return to the initial position. The design of the controller was performed by resorting to a simplified model of the robot dynamics, in which the motion occurs along a straight line, thus ignoring the lateral or heading-angle dynamics.

### 1.2 System and Model

Before presenting the controlling techniques required in the assignment of this laboratory activity, and the solutions adopted to implement such controllers, it is fundamental to give an insight into the physical system and its components, in order to highlight the most important aspects that had to be taken in consideration during the design of the controllers.

As mentioned previously, a simplification of the robot dynamics was used, by assuming that the robot moves only in one direction and no lateral motion occurs (i.e. the heading-angle is constant).

The two-wheeled balancing robot can be represented as a multi-body system involving different parts:

- Left and right wheels.
- Left and right DC gearmotor rotors

- Robot body, which includes the robot chassis (electronic boards and the motor supporting brackets) the DC gearmotor stators (left/right) and the battery.

The mathematical model of the robot, necessary to define a state-space model for the control design activity, was derived by making use of a set of generalized coordinates to uniquely describe the robot configuration with a minimum set of independent variables. These coordinates include: robot position, robot tilt angle  $\vartheta$ , left and right wheel angles (the two wheels angles can be considered always identical thus  $\gamma$  denotes the common value of the two wheels angles). A set of auxiliary coordinates was also introduced to simplify the derivation of the model. In particular, the following variables were used: heading (or yaw) angle, left and right rotor angles.

In order to derive the equations of motion of the robot, the Lagrangian approach was used. For this purpose, the kinematic equations of each body were first computed. After that, for each body part the kinetic and potential energies were calculated. According to the Lagrangian approach, the following step was to compute the Lagrangian function, which finally led to the Equations of Motion (EoM) of the model. The complete EoM for the longitudinal dynamics can be found in the Appendix. Instead, in the following is presented a compact matrix formulation of the dynamical model. If  $\mathbf{q} = [\gamma, \vartheta]^T$  is the vector of the generalized coordinates, then:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{F}_v\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau}, \quad (1)$$

where  $\mathbf{M}(\mathbf{q})$ ,  $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ ,  $\mathbf{F}_v$  are, respectively, the inertia matrix, the matrix of centrifugal and Coriolis-related coefficients and the matrix of viscous friction coefficients, while  $\mathbf{g}(\mathbf{q})$  is the torque contribution due to gravity. The full expression of such matrices is reported in the Appendix. However, the model presented above is not linear. In order to obtain a convenient state-space representation, the dynamical model has to be linearized about the unstable equilibrium configuration with the robot body in steady vertical position.

A compact matrix formulation of this linearized model is

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{F}_v\dot{\mathbf{q}} + \mathbf{G}\mathbf{q} = \boldsymbol{\tau}. \quad (2)$$

Again, the full expression of the matrices appearing in (2) is presented in the Appendix. A possible state-space model, where the state vector is defined as  $\mathbf{x} = [\mathbf{q}, \dot{\mathbf{q}}]^T = [\gamma, \vartheta, \dot{\gamma}, \dot{\vartheta}]^T$  is the following:

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu} \quad (3)$$

with:

$$\mathbf{A} = \begin{bmatrix} \mathbf{0}_{2x2} & \mathbf{I}_{2x2} \\ -\mathbf{M}^{-1}\mathbf{G} & -\mathbf{M}^{-1}\mathbf{F}_v \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{0}_{2x2} \\ -\mathbf{M}^{-1} \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \quad (4)$$

where the matrices  $\mathbf{M}$ ,  $\mathbf{G}$ ,  $\mathbf{F}_v$  are described in the Appendix, and  $u = 2\tau$ .

In addition, in order to consider the electromechanical torque generated by each DC gearmotor, the effect of the armature circuit and the angular displacement of the rotor with respect to the

stator, it is possible to derive the differential equation describing the actuation system which allows to write

$$\tau = \frac{NK_t}{R_a} u_a - \frac{N^2 k_t k_e}{R_a} (\dot{\gamma} - \dot{\vartheta}). \quad (5)$$

Substituting then (5) into (1) one gets the nonlinear model

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{F}'_v \dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau}'$$

while substituting it into (2) one gets the linear model

$$M\ddot{\mathbf{q}} + \mathbf{F}'_v \dot{\mathbf{q}} + \mathbf{G}\mathbf{q} = \boldsymbol{\tau}'. \quad (6)$$

with  $\mathbf{F}'_v$  and  $\boldsymbol{\tau}'$  as defined in the Appendix. The state-space model matrices become:

$$\mathbf{A} = \begin{bmatrix} \mathbf{0}_{2x2} & \mathbf{I}_{2x2} \\ -\mathbf{M}^{-1}\mathbf{G} & -\mathbf{M}^{-1}\mathbf{F}'_v \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{0}_{2x2} \\ -\mathbf{M}^{-1} \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \quad (7)$$

## 2 Tasks, Methodologies and Results

### 2.1 Balance-and-position state-space control using LQR methods

The control systems requested for this laboratory were based on state-space methods involving the LQR approach. Before implementing the balance-and-position controller, it was first crucial to estimate the robot state  $\mathbf{x} = [\gamma, \vartheta, \dot{\gamma}, \dot{\vartheta}]^T$  from the measurements provided by the onboard sensors, i.e. the incremental encoder and the MPU (accelerometer and gyroscope). In order to do that, an observer was implemented in the following way:

- For the estimation of the body tilt angle  $\vartheta$ , a complementary filter approach was adopted. With this method, a pair of filters was used to isolate the accelerometer-based estimate  $\hat{\vartheta}_a$  (more reliable at low frequencies) and the gyroscope-based estimate  $\hat{\vartheta}_g$  (more reliable at high frequencies), which were then combined together allowing to obtain a more accurate estimate on  $\vartheta$ . Since the whole system operated in discrete-time, even the filters had to be discretized. By choosing the *backward Euler* discretization method for a simple first-order filter, the resulting transfer function had the form:

$$H(z) = \frac{C}{1 - ((1 - C)z^{-1})}, \quad \text{with } C = \frac{T}{T_c + T}, \quad (8)$$

with  $T$  denoting the sampling time and  $f_c = \frac{1}{T_c}$  the filter's cut-off frequency, chosen to be 0.35 [Hz].

At this point, in time domain the estimated angle had the expression:

$$\hat{\vartheta}[k] = C\hat{\vartheta}_a[k] + (1 - C)(\hat{\vartheta}[k - 1] + Ty_g[k]) \quad (9)$$

- For the estimation of the wheel angle  $\gamma$  the identity  $\Delta\vartheta_{rot} = N(\gamma - \vartheta)$  was used to obtain the estimate:

$$\hat{\gamma} = \frac{\Delta\vartheta_{rot}}{N} + \hat{\vartheta} \quad (10)$$

with  $\Delta\vartheta_{rot}$  being the rotor angular displacement measured by the encoder and  $N$  the gearbox ratio.

- For the estimation of the angular velocities  $\dot{\vartheta}$  and  $\dot{\gamma}$  from  $\hat{\vartheta}$  and  $\hat{\gamma}$  a "real derivative" filter was used, i.e.:

$$H_\omega(z) = \frac{1 - z^{-3}}{3T} = \frac{z^3 - 1}{3Tz^3}. \quad (11)$$

A possible implementation of the state observer, using two different realizations of the complementary filtering scheme, is reported in the Appendix.

### 2.1.1 Assignment #1: Nominal perfect tracking design

The goal was to design a discrete-time state-space controller in order to stabilize the robot body to its upward vertical position while guaranteeing the nominal perfect tracking of a constant wheel angle position set-point  $\gamma^*$ .

This was done by first discretizing the continuous-time plant model in (3) with matrices  $A$  and  $B$  as in (7) using the exact discretization method with sampling time  $T = 0.01\text{ s}$ .

With this assumption, the resulting discretized model plant had the form

$$\begin{cases} \mathbf{x}[k+1] = \Phi \mathbf{x}[k] + \Gamma u[k] \\ y[k] = \mathbf{H} \mathbf{x}[k] \end{cases}$$

with

$$\Phi \simeq \begin{bmatrix} 1.00 & 0.002 & 0.01 & 0.00 \\ 0.00 & 1.00 & 0.00 & 0.01 \\ 0.00 & 0.43 & 0.96 & 0.03 \\ 0.00 & 0.55 & 0.02 & 0.98 \end{bmatrix}, \quad \Gamma \simeq \begin{bmatrix} 0.00 \\ 0.00 \\ 0.05 \\ -0.03 \end{bmatrix}, \quad \mathbf{H} = [1 \ 0 \ 0 \ 0],$$

where the choice of  $\mathbf{H}$  was dictated by the fact that the model output  $y$  had to be equal to the signal to track, namely the wheel angle position  $\gamma$ .

At this point, the discrete-time state-space control law had the following structure:

$$u[k] = -\mathbf{K} \mathbf{x}[k] + N_r r[k], \quad N_r = (N_u + \mathbf{K} \mathbf{N}_x),$$

with  $r = \gamma^*$  being the wheel angle reference signal, and  $\mathbf{x}$  the state vector estimated with the state observer designed above.

Furthermore, as it was done in activity number 2, the feedforward gains  $\mathbf{N}_x$  and  $N_u$  were determined by solving the following set of linear equations:

$$\begin{bmatrix} \Phi - \mathbf{I} & \Gamma \\ \mathbf{H} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{N}_x \\ N_u \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix},$$

i.e.

$$\begin{bmatrix} 0.00 & 0.002 & 0.01 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.01 & 0.00 \\ 0.00 & 0.43 & -0.04 & 0.03 & 0.05 \\ 0.00 & 0.55 & 0.02 & -0.02 & -0.03 \\ 1.00 & 0.00 & 0.00 & 0.00 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{N}_x \\ N_u \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix},$$

yielding

$$\mathbf{N}_x = [1 \ 0 \ 0 \ 0]^T, \quad N_u = 0.$$

The feedback matrix  $\mathbf{K}$ , on the other hand, was computed using the LQR method, i.e. minimizing the cost function

$$J = \sum_{k=0}^{+\infty} \mathbf{x}^T[k] \mathbf{Q} \mathbf{x}[k] + \rho \cdot r u^2[k] \quad (12)$$

where the cost weights  $\mathbf{Q}$  and  $r$  were chosen using Bryson's rule; in particular, since it was desired, at steady state, to have

$$|\gamma - \gamma^*| < \pi/36 \text{ (5 deg)}, \quad |\theta| < \pi/360 \text{ (0.5 deg)}, \quad |u| < 1 V,$$

they were selected as follows:

$$\bar{\gamma} = \pi/18, \quad \bar{\vartheta} = \pi/360, \quad \bar{u} = 1.$$

This values yielded:

$$\mathbf{Q} = \text{diag} \left\{ \frac{1}{\bar{\gamma}^2}, \frac{1}{\bar{\vartheta}^2}, 0, 0 \right\} \simeq \begin{bmatrix} 33 & 0 & 0 & 0 \\ 0 & 13\,000 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad r = \frac{1}{\bar{u}^2} = 1.$$

In the  $\mathbf{Q}$  matrix, the weights of the two angular velocities  $\dot{\gamma}$  and  $\dot{\vartheta}$  were set to zero, while the extra weight in (12) was used to adjust the relative weighting between the state and the input contributions to the total cost function value; the possible choices for such weight were considered as follows:

$$\rho \in \{500, 5000\}.$$

In particular, the resulting values of  $\mathbf{K}$  were:

$$\begin{aligned} \rho = 500 &\implies \mathbf{K} \simeq [-0.24 \ -53.70 \ -1.15 \ -6.08], \\ \rho = 5000 &\implies \mathbf{K} \simeq [-0.07 \ -51.94 \ -1.08 \ -5.90]. \end{aligned}$$

As always, the tuning of the parameters was performed on the Simulink model first, in particular to select the best-performing value of  $\rho$ .

We may anticipate that, for all the laboratory tests concerning the nominal design, the value  $\rho = 500$  was chosen, while for the tasks involving the robust one, the choice fell on  $\rho = 5000$ . For this reason and for the sake of clarity, in the following only the results obtained with the values of  $\rho$  which have then been used in the laboratory are presented; a deeper analysis and the comparison between the performances obtained in each test depending on the choice,

along with a brief explanation of why one was preferred over the other, is reported in section **A.2** of the Appendix.

As a final remark, it may be useful to describe now how the Figures presented will be structured: most of the plots will have two axes: on the left one, in blue, will be reported the value for  $\gamma$ ; on the right one, in green, the value for  $\vartheta$ .

This will be done in order to convey as much visual information as possible. Moreover, reference signals, when present, will be plotted using a yellow dashed line and will be the only ones to appear in the legend.

**Test 1.** The first test was aimed at verifying that the controller could 'return' the robot to its vertical balance position when starting from an initial body tilt angle of  $\vartheta_0 = \pi/36$  [rads] ( $= 5^\circ$ ). For this reason, in the simulations the initial state was set to  $x_0 = [0 \ \pi/36 \ 0 \ 0]^T$ , while the reference input and the load disturbance were both set equal to 0. The test run in the simulation gave the results shown in Figure (1).

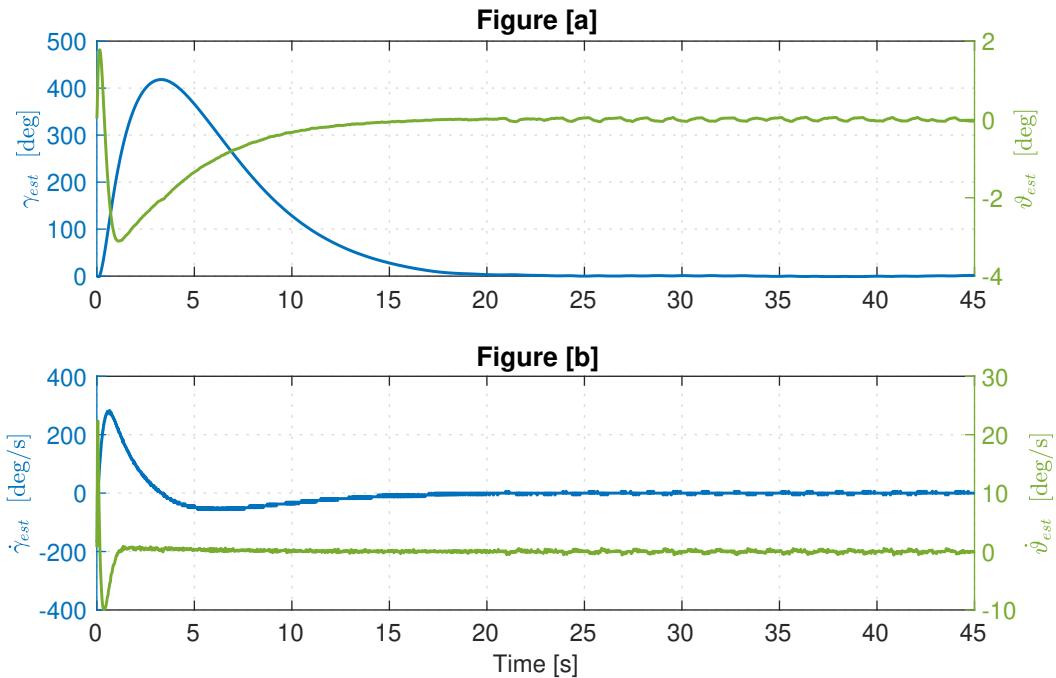


Figure 1: Simulation of Test 1:  $x_0 = [0 \ \pi/36 \ 0 \ 0]^T$  and no input nor load disturbance applied (nominal case).

It is clear that the controller is able to restore balance after that the robot is released from a position off the upward vertical equilibrium. It may also be meaningful to notice that the tilt angle  $\vartheta_{est}$  seems to be 0 at  $t = 0$ ; this could lead to believe that something is not working as it should, given how  $x_0$  has been set before.

A closer look, though, shows that this is indeed to be expected: in fact, what Figure (1a) shows is not the 'actual' tilt angle, but rather the initial transient of the observer. In fact, since the initial state of the observer  $x_{0,obs} = [0 \ 0 \ 0 \ 0]^T$  does not coincide with the plant's one,  $x_{0,robot} = [0 \ \pi/36 \ 0 \ 0]^T$ , it is inevitable to have an initial estimation error, which

will however converge to zero as long as the observer is asymptotically stable.

In particular, for the specific observer implemented in the model, the initial transient is caused by the dynamics of the low-pass filter used to filter the estimation of the tilt angle coming from the output data of the accelerometer. Figure (2) below seems to corroborate this thesis.

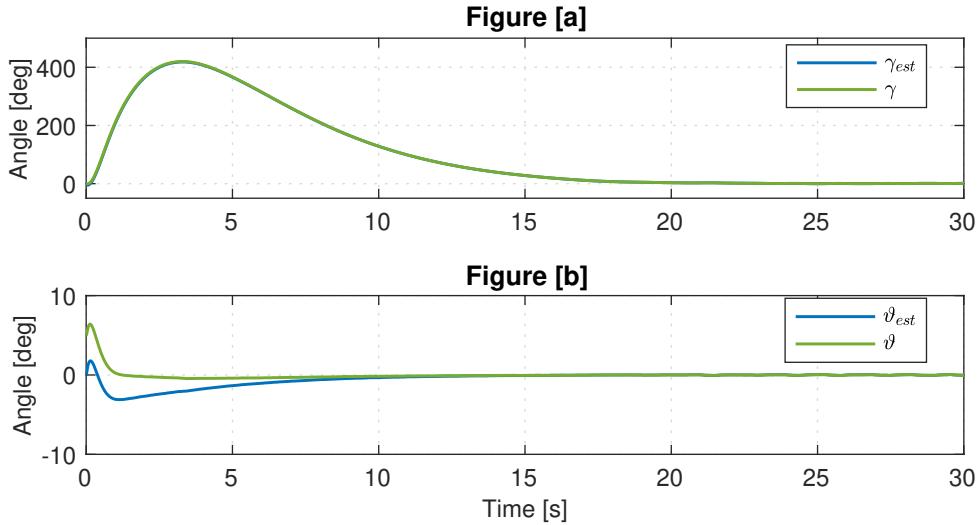


Figure 2: Comparison between  $(\gamma, \vartheta)$  and  $(\gamma_{est}, \vartheta_{est})$  in the simulated nominal controller.

Finally, the test was run on the experimental setup in the laboratory, where as described in the introductory part, the controller was activated through a button. For this reason, and also given the fact that the memory buffer of the Bluetooth receiver was being overwritten every  $\sim 50$  seconds, the tests were physically harder to perform and 'clean' data was tricky to obtain.

In particular, the first two tests were executed in a single run: the robot's vertical body was tilted of about  $5^\circ$  and then released at  $t = 17$  s, making sure to simultaneously press the button to activate the controller. The step input, as designed, sprung into action 10 seconds later, i.e. at  $t = 27$  s (thus leaving 10 seconds to the controller to stabilize the system). From that moment on, the controller had to ensure the tracking of the input step signal.

Thus, for what regards test 1, below in Figure (3) is the data collected in the time interval  $[16, 27]$  of the experiment executed with the procedure described above.

It can be seen from (3a) that the controlled robot does indeed return to its balancing vertical position.

To be more precise, given the fact that the vertical equilibrium is not stable, it is physically impossible for the system to keep the tilt angle equal to zero for longer than a few instants. Rather, the wheels will keep it moving forward and back while the robot body will keep oscillating around its vertical equilibrium.

Figure (3c) is not very informative since no load disturbance was applied for this test and thus the voltage at the input of the voltage driver was the same as the output voltage of the controller.

As expected, the curves were not as 'sharp' as those of the simulation, but this is perfectly

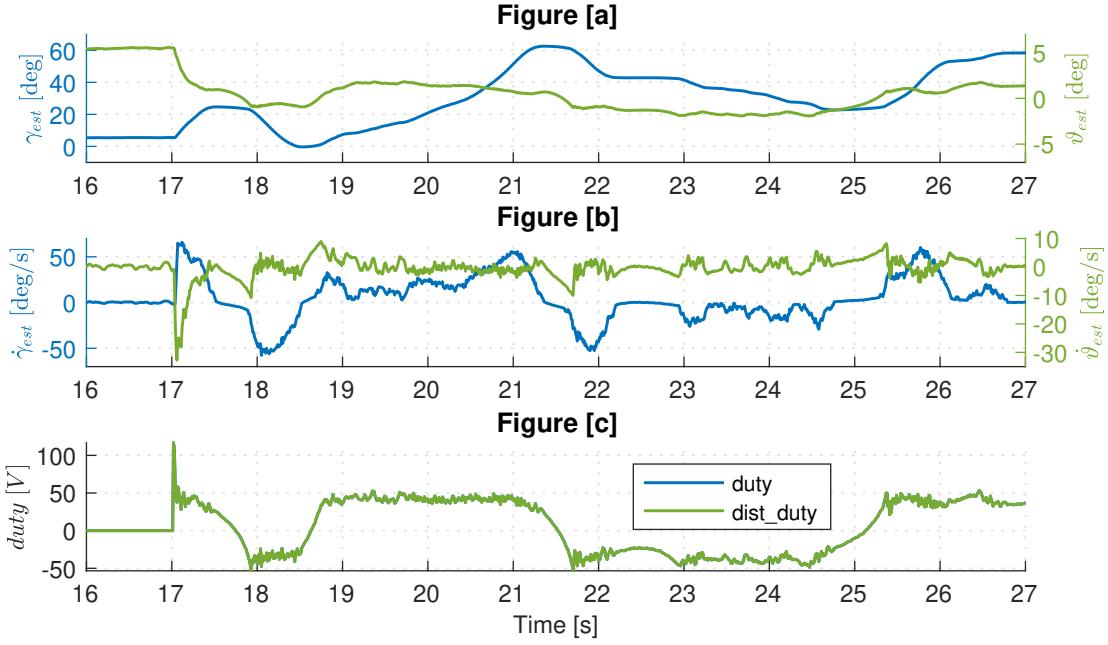


Figure 3: Test 1 run on the experimental setup (nominal controller).

consistent with the fact that the Simulink model is only an approximation of the real system and does not keep into account the whole dynamics of the robot.

**Test 2.** The second request was to validate the design of the controller by testing the response of the system when fed with a step reference of  $\gamma^* = 0.1/r \simeq 2.9$  [rads] ( $\simeq 158^\circ$ ) ( $r$  being the wheel radius) starting from a position of vertical equilibrium. The aim was to ensure that the closed-loop system could achieve steady state tracking of wheel-position set-points (while guaranteeing vertical equilibrium of the body) in absence of load disturbances.

Below in Figure (4) are shown the results on the Simulink model. It is clear from the plots that the controller ensures perfect tracking of the reference. Moreover, the system stabilizes around its vertical equilibrium, as it was expected.

The plot of the velocities, on the other hand, is somewhat odd: it looks like some noise at high frequency is being amplified. The same behaviour will be found in the robust controller as well. For this reason, both cases will be analyzed more in detail in Section (A.3) of the Appendix.

Next, the controller was tested on the experimental setup in the laboratory; in Figure (5) are the plots of the data (obtained as described above, performing both test 1 and test 2 in a single run), starting from  $t = 26$  s, i.e. one second before the step signal was fed as input, up to  $t = 43$  s, when the data collection was terminated.

The results confirmed what was observed in the simulations: the robot was able to achieve step-tracking of wheel-position constant set points, while preserving vertical balance. Also, with respect to the first (and to the third, as the next figures will show), the velocities suffer more of that 'high frequency issue' found in the simulation, even though this does not look to

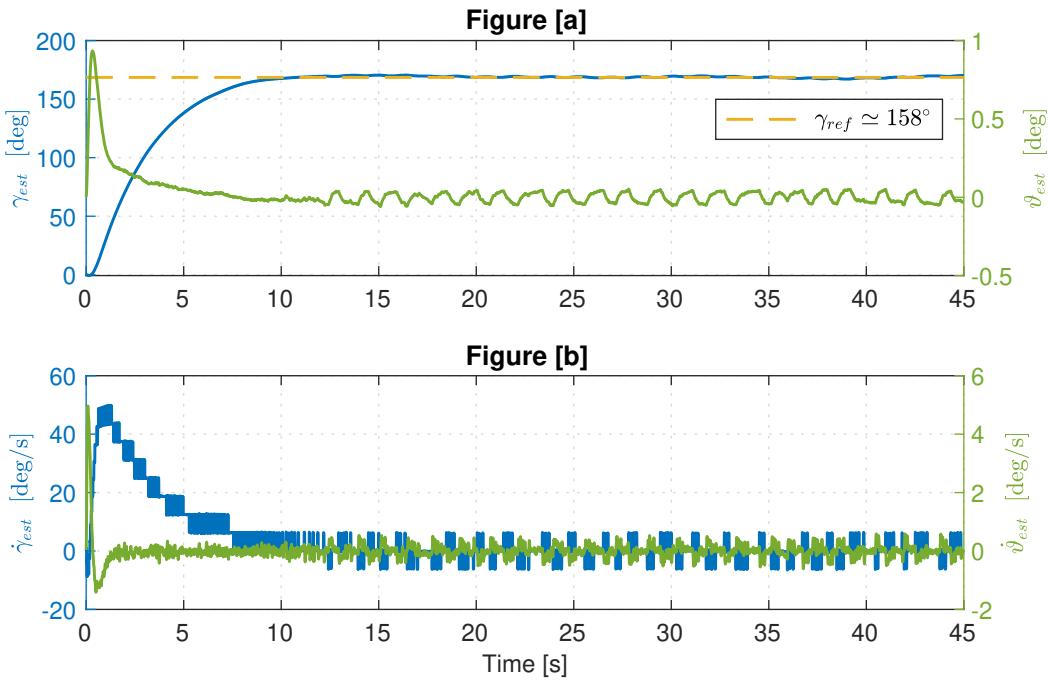


Figure 4: Simulation of Test 2:  $x_0 = [0 \ 0 \ 0 \ 0]^T$ , step reference  $\gamma_{ref} \simeq 158^\circ$  and no load disturbance is being applied (nominal case).

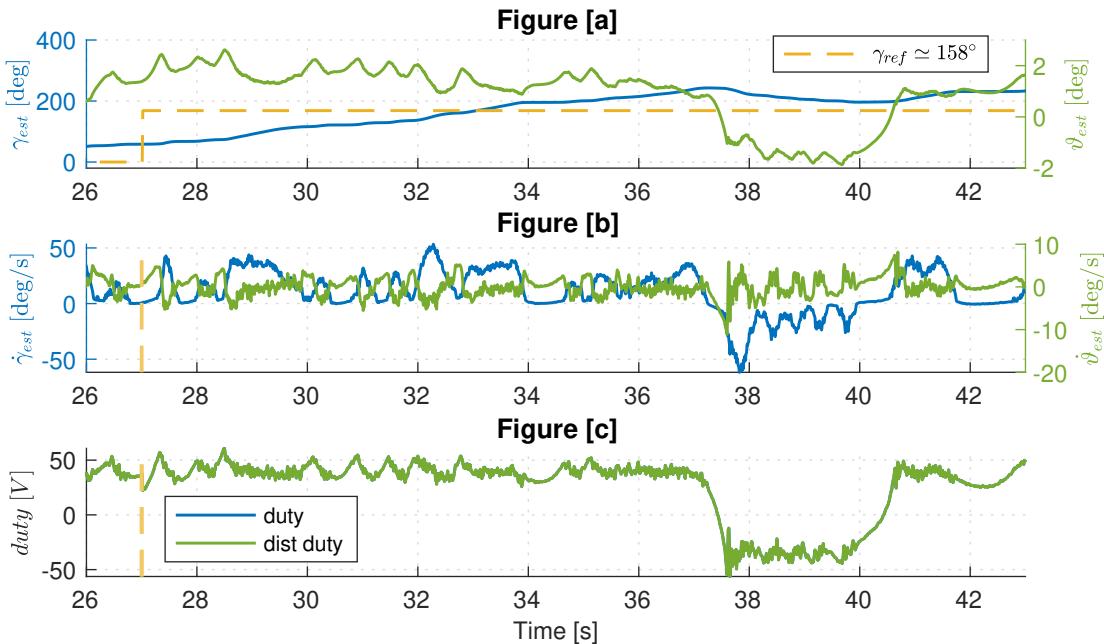


Figure 5: Test 2 run on the experimental setup (nominal controller).

have had any noticeable impact on the transient.

**Test 3.** In the third and last experiment, the aim was to test the capabilities of the controller to ensure perfect step-tracking even in case of disturbances. In particular, the step reference

fed as input was the same as before, but from  $t = 30\text{ s}$  a constant disturbance of 5 [V] was applied to the motor input.

The system was expected to fail in this task, as a nominal controller can, indeed, ensure only *nominal* perfect tracking.

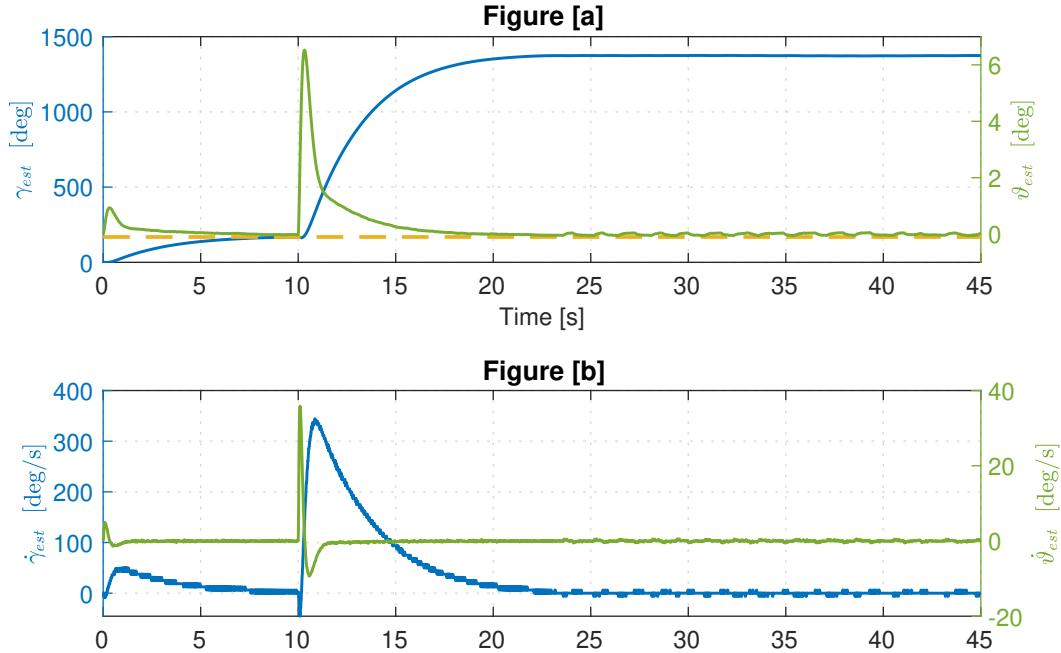


Figure 6: Simulation of Test 3:  $x_0 = [0 \ 0 \ 0 \ 0]^T$ , step reference  $\gamma_{ref} \approx 158^\circ$  and load disturbance  $V_{dist} \approx 115\text{ [V]}$  (nominal case).

As expected, looking at Figure (6), it is evident that the controller is not able to guarantee steady-state tracking in presence of the disturbance.

This limitation, as it was observed in the previous experiences, can be worked around by introducing an integral action in the control law, as it will be done in the following point.

Finally, also the experimental system in the laboratory was tested in this scenario.

Since the controller's perfect steady state tracking capability had already been proven in Test 1, it seemed convenient to run the experiment directly applying the load disturbance, in order to have a wider time window to analyze the behaviour of the system.

For this reason, the robot was simply laid on a surface and the controller button was pushed approximately at  $t = 11$  (no reference signal was fed as input) which resulted in the disturbance signal being fed to the system at  $t \approx 26$ , i.e. 15 seconds later.

Below in Figure (7) is plotted the data that was collected.

Once more the simulation results were confirmed by the experiments on the robot: the nominal controller was *not* able to guarantee perfect steady state tracking in presence of load disturbances. As stated before, some slight variation in the behaviour was to be expected since the Simulink model was only approximating the dynamics of the experimental setup.

Finally, it is worth pointing out that, looking at (7c), the controller output had a strange-looking behaviour. No direct cause was isolated in the laboratory, nor effective solution was found.

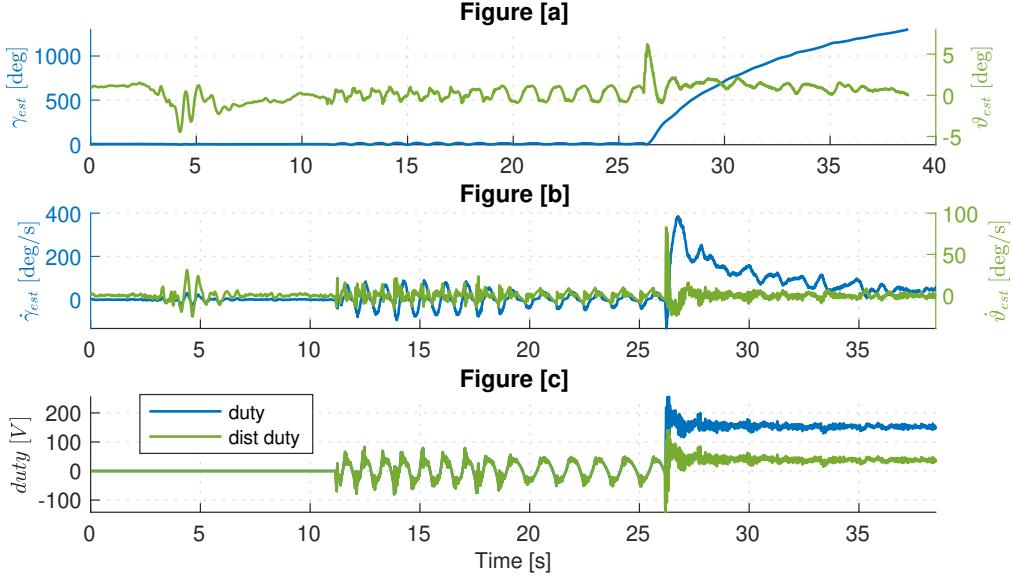


Figure 7: Test 3 run on the experimental setup (nominal controller).

A tentative was made by changing arbitrarily the cut-off frequency of the filters, since it looks from the plot that the system was amplifying some high frequency noise, but this did not help.

### 2.1.2 Assignment #2: Robust perfect tracking design via integral action

The goal of the second assignment was to implement an integral action in the controller in order for it to be able to ensure perfect tracking of step inputs not only in the nominal case, but in the *robust* as well. In particular, it was required to achieve perfect tracking of constant wheel angle position set-points even in presence of external disturbances.

This was done, as in the previous activities, by extending the state of the system by introducing an integrator entry  $x_I$  in the state in order to form the extended state  $\mathbf{x}_e = [\mathbf{x}_I \ \mathbf{q} \ \dot{\mathbf{q}}]^T = [x_I \ \gamma \ \vartheta \ \dot{\gamma} \ \ddot{\vartheta}]^T$  of the augmented system

$$\Sigma_e : \mathbf{x}_e[k+1] = \underbrace{\begin{bmatrix} 1 & H \\ 0 & \Phi \end{bmatrix}}_{\Phi_e} \mathbf{x}_e[k] + \underbrace{\begin{bmatrix} 0 \\ \Gamma \end{bmatrix}}_{\Gamma_e} u[k] - \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

with state matrix  $\Phi_e$  and input matrix  $\Gamma_e$  having numerical values:

$$\Phi_e \simeq \begin{bmatrix} 1.00 & 1.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 1.00 & 0.00 & 0.01 & 0.00 \\ 0.00 & 0.00 & 1.00 & 0.00 & 0.01 \\ 0.00 & 0.00 & 0.43 & 0.96 & 0.03 \\ 0.00 & 0.00 & 0.55 & 0.02 & 0.98 \end{bmatrix}, \quad \Gamma_e \simeq \begin{bmatrix} 0.00 \\ 0.0002 \\ -0.0002 \\ 0.0472 \\ -0.0317 \end{bmatrix}$$

Again, to compute the control matrix  $\mathbf{K}$  the optimal LQR control technique was used, with the index to be minimized being again:

$$J_e = \sum_{k=0}^{+\infty} \mathbf{x}^T[k] \mathbf{Q}_e \mathbf{x}[k] + \rho \cdot r u^2[k], \quad (13)$$

using the extended cost matrix:

$$\mathbf{Q}_e = \text{diag}\{q_I, \text{diag}\{\mathbf{Q}\}\}, \quad (14)$$

with  $\mathbf{Q}$  being the same cost matrix defined for the nominal case and  $q_I = 0.1$  the cost of the integral action.

First, the closed-loop system was simulated on the Simulink model.

As in the nominal case, the model was tested using the two choices

$$\rho \in \{500, 5000\}.$$

In particular, the resulting values of  $\mathbf{K}$  were:

$$\begin{aligned} \rho = 500 &\implies \mathbf{K} \simeq \begin{bmatrix} -0.01 & -1.92 & -65.17 & -1.68 & -7.33 \end{bmatrix}, \\ \rho = 5000 &\implies \mathbf{K} \simeq \begin{bmatrix} -0.00 & -0.93 & -58.62 & -1.38 & -6.63 \end{bmatrix}. \end{aligned}$$

As already anticipated above, for the robust design the value of the relative cost to use on the experimental setup in the laboratory was chosen to be  $\rho = 5000$  (again, refer to section **A.2** of the Appendix for a more exhaustive reasoning on this choice).

The tests run using the robust controller were exactly the same as the ones performed for the nominal case and for this reason, in the following, the structure of each test will be only briefly recalled.

**Test 1.** The system started with an initial body tilt angle  $\vartheta_0 = \pi/36$  [rads] and no inputs nor disturbances were considered. In Figure (8) are the results obtained in the simulation.

Not surprisingly, the controller is again able to restore the balance after that the robot is released from a position off the upward vertical equilibrium.

Comparing these (simulated) results with those (simulated) of Figure (1), it can be observed that the robust controller used guarantees both faster and smoother convergence (i.e. requiring less 'wheel work') towards the vertical equilibrium of the robot's body.

This result is non-trivial, since increasing the value of  $\rho$  does indeed allow (in most cases) for a less-peaked response, but at the price of a slower convergence towards the set-point.

Finally, the experiment was performed on the setup in the laboratory: once more, the first two tests were executed in a single run: the robot's vertical body was titled of about  $5^\circ$  and then released at  $t = 10\text{s}$ , at the same time pressing the button to activate the reference input,

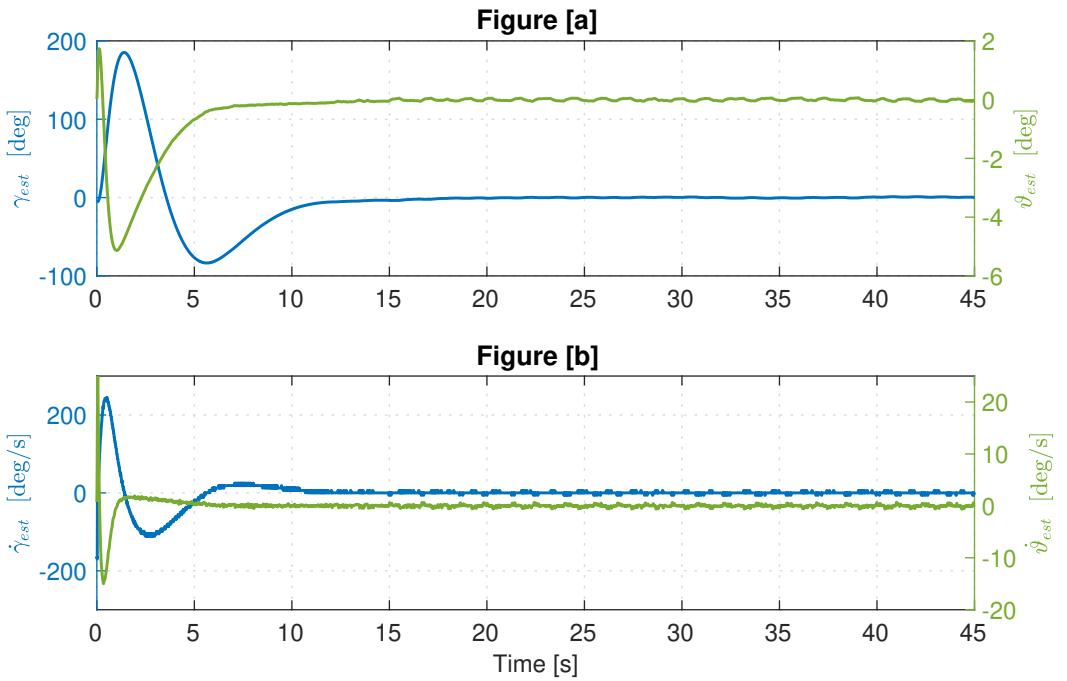


Figure 8: Simulation of Test 1:  $x_0 = [0 \ \pi/36 \ 0 \ 0]^T$  and no input nor load disturbance applied (robust case).

which this time was designed to spring into action 15 seconds later, i.e. at  $t = 25\text{s}$ , thus leaving 15 seconds to the controller to re-balance the system. The results obtained are shown in Figure (9).

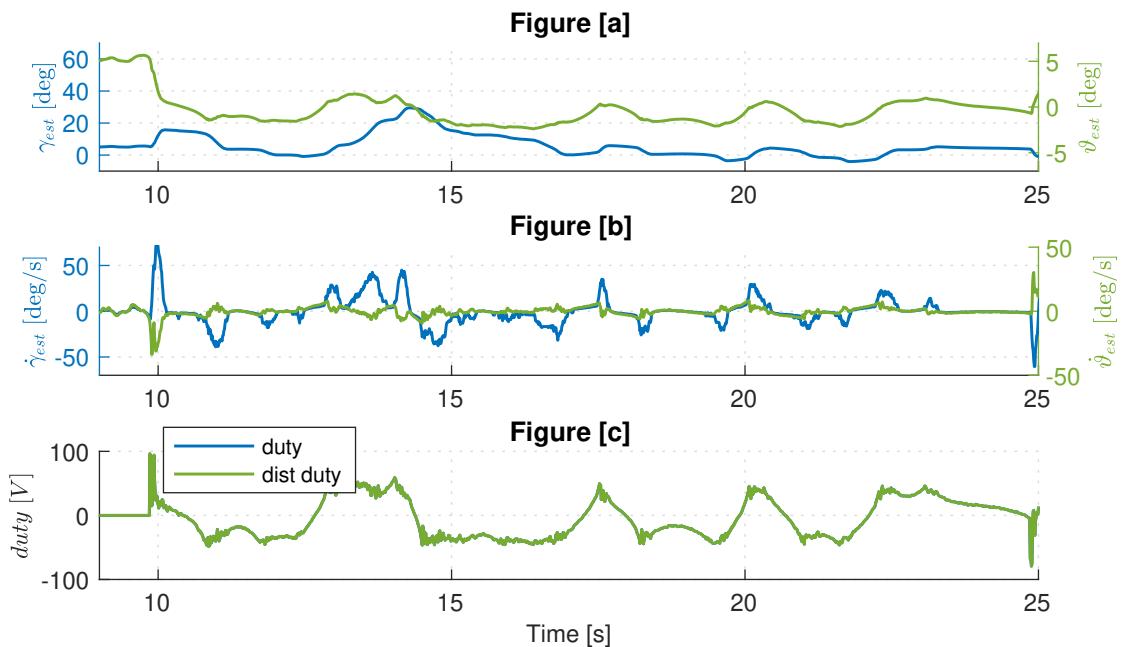


Figure 9: Test 1 run on the experimental setup (robust controller).

The controller behaved as it was expected from the simulations. It would be interesting to compare these results with those of the nominal controller in Figure (3). However, since real systems do not behave as 'smoothly' as their Simulink counterparts, even the differences

are much harder to spot and it is thus difficult to compare quantitatively the two; it can only be said that the stabilizing action is very similar and no relevant difference can be noticed.

**Test 2.** As in the previous section, the aim of the task was to test the response of the system to a step reference  $\gamma^* \simeq 158^\circ$  when starting from a position of vertical equilibrium. The simulation gave the results shown in Figure (10).

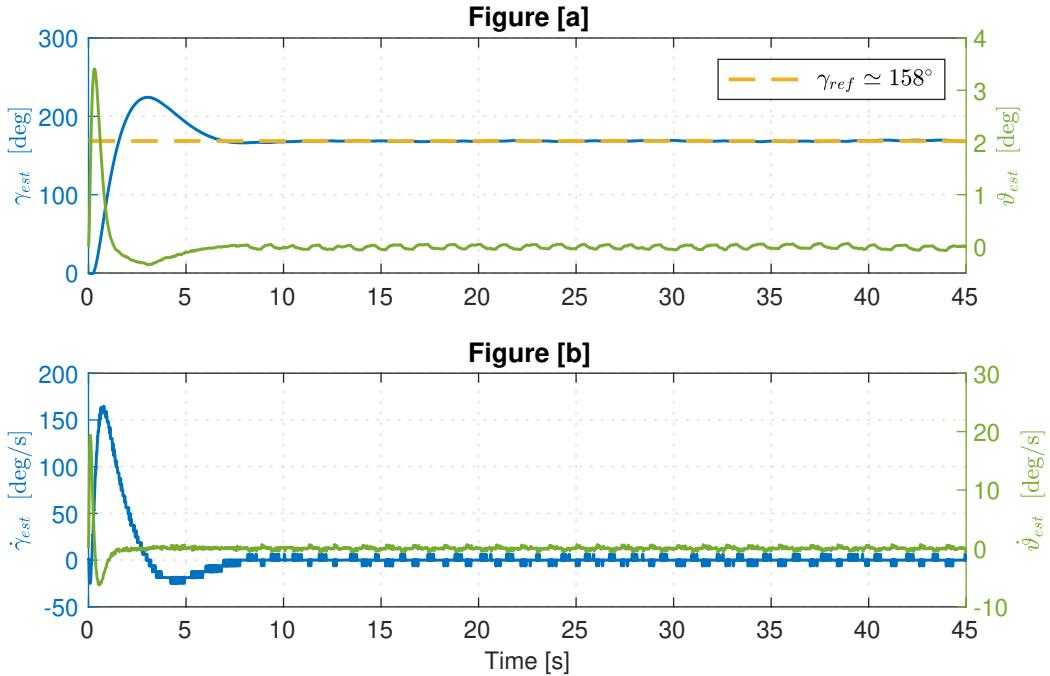


Figure 10: Simulation of Test 2:  $x_0 = [0 \ 0 \ 0 \ 0]^T$ , step reference  $\gamma_{ref} \simeq 158^\circ$  and no load disturbance applied (robust case).

As in the nominal case, the simulation showed that the system reached the desired position with steady state error equal to zero.

The velocities have a similar 'high frequency' behaviour to that of the nominal controller, and again a more detailed discussion is postponed to Section (A.3) of the Appendix.

Finally, the controller was tested on the real system with the procedure described above, i.e. performing both tests 1 and 2 in a single run.

In Figure (11) below are the plots of the data already shown before, now ranging from  $t = 26$  s, one second before the step signal, to  $t = 39$  s, when the run was stopped in order to avoid buffer overwriting. These results confirmed that the controller was indeed able to guarantee step tracking in absence of load disturbances.

And again, on the experimental setup, the velocities do not seem to have influenced in any interesting way the response of the system.

These first two experiments were not very informative as they only proved that the addition of the integral action in the control structure preserved the capabilities of the controller to achieve stabilization of the robot's vertical body and steady state tracking in absence of external disturbances.

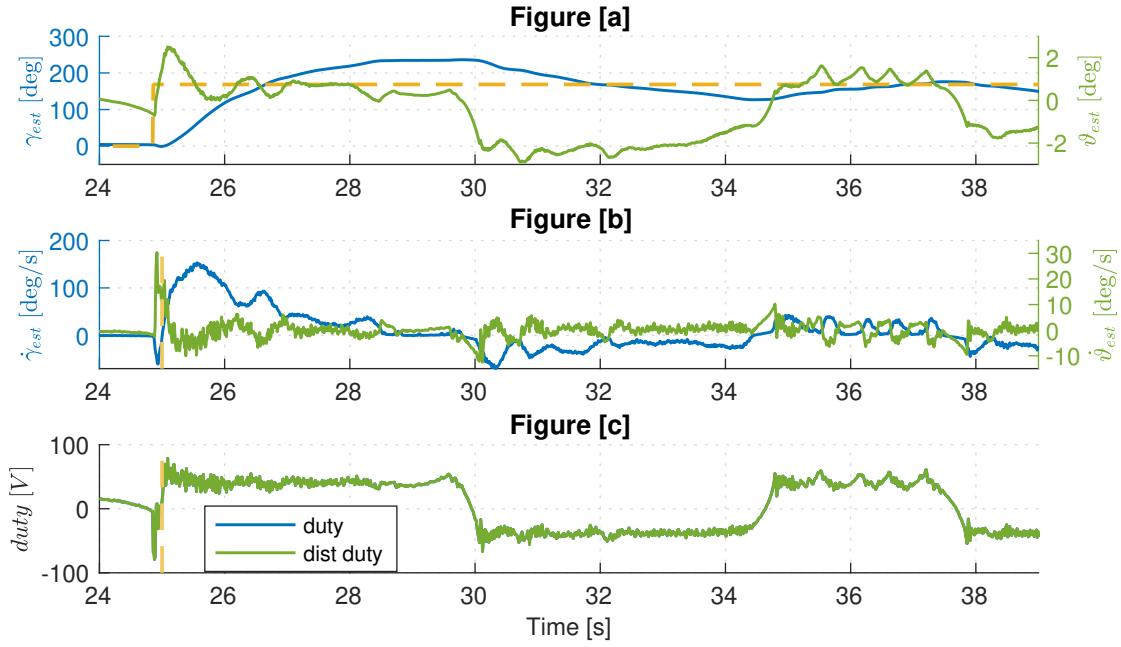


Figure 11: Test 2 run on the experimental setup (robust controller).

The test that was actually relevant to demonstrate the improvement of the robust over the nominal controller was the third one.

**Test 3.** Again, the aim of this task was to inject a constant disturbance signal  $V_{dist} = 5$  [V] into the system while it was tracking a step reference and test if it could reject the noise. In the simulation environment the results shown below in Figure (12) were obtained.

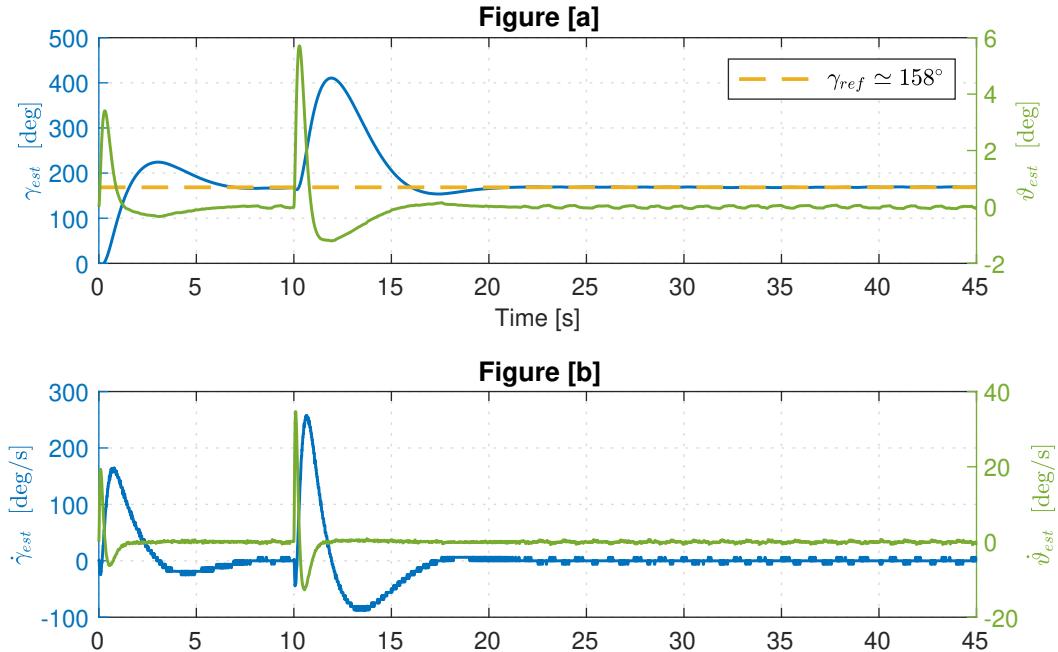


Figure 12: Simulation Test 3:  $x_0 = [0 \ 0 \ 0 \ 0]^T$ , step reference  $\gamma_{ref} \simeq 158^\circ$  and load disturbance  $V_{dist} \simeq 115$  [V] (robust case).

It is clear how the implementation of the integral action in the controller ensures tracking of constant wheels-position set points even in presence of disturbances.

In fact, it can be seen from the plots that the noise entering the plant at  $t = 10\text{ s}$  gets rejected and the wheels return to track the set position.

Lastly, the robust controller was tested on the experimental setup in the laboratory. As it was the case for the nominal one, it seemed convenient to run the experiment directly applying the load disturbance, i.e. starting from any tilt angle  $\vartheta$  and enabling the controller at  $t = 15\text{ s}$  without previously feeding the system with any step-input, in order to have a wider time window to analyze the behaviour of the balancing robot.

The data that was collected is plotted in Figure (13) below.

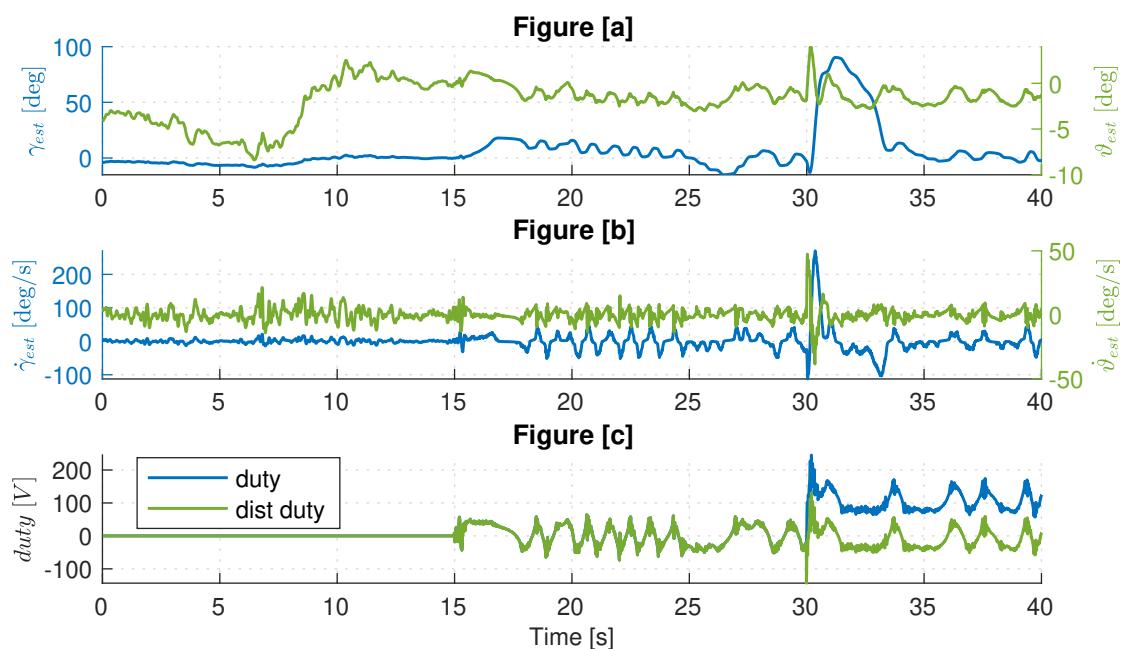


Figure 13: Test 3 run on the experimental setup (robust controller).

The results confirm what was observed in the simulations (which was actually expected from a robust controller), even though there is at least one element of difference with the model's behaviour: in the simulation, the value of  $\gamma_{est}$  reached  $\sim 400^\circ$ , while the wheels of the robot in the lab moved only of approximately  $100^\circ$ . This means that in the real experiment the system traveled a shorter distance with respect with what it was expected to.

Apart from the usual motivation that real systems never behave exactly like their models, a possible explanation for this behaviour could be that the oscillations in the position of the wheels (clearly visible from the first plot) may have reduced the impact of the disturbance (and also, that movement may have deteriorated the estimation of the tilt angle).

### 2.1.3 Optional assignment: Yaw angle control

By observing the balancing robot's movement during the previous assignments, it was noticed how it tended to pivot around one of the two wheels or to drift laterally while performing balance

and position control. This was the case even if the control action fed to the two motors is the same. This phenomenon is due to the fact that the two motor units are not identical and that their parameters differ slightly. Furthermore, the two gearboxes introduce marginally different friction and backlash. The combined effect of the various dissimilarities results in the two motors moving by different angles, even when driven by the same voltage.

To mitigate this behavior a simple PI controller for the robot yaw angle  $\psi$  was introduced. Its control output was combined with the longitudinal state-space controller's one as follows:

$$u_r = u_\Sigma + u_\Delta, \quad u_l = u_\Sigma - u_\Delta \quad (15)$$

where  $u_r$  and  $u_l$  are the right and left motor control inputs respectively,  $u_\Sigma$  is the common control action generated by the longitudinal controller and  $u_\Delta$  is the differential control action generated by the yaw angle controller. The PI controller's parameters were set to

$$K_P = 3.3, \quad K_I = 0.7 \quad (16)$$

as suggested in the Handout. Its reference input was set to  $\psi_{ref} = 0^\circ$  in order to contrast the drift and keep the robot facing the same direction during operation.

In order to have an estimate of the yaw angle  $\hat{\psi}$  used for the computation of the error by the PI controller, the state observer was expanded. A block that, given the two motor-side encoder pulse measurements, converted them into load-side degree measurements  $\vartheta_r$ ,  $\vartheta_l$  and then performed the following calculation:

$$\hat{\psi} = \frac{r}{w} (\vartheta_r - \vartheta_l) \quad (17)$$

(where  $r$  is the wheel radius and  $w = 2 |y_w^b|$  is the distance between the two wheel centers of mass), was implemented.

The yaw control implementation was validated on both the nominal and robust perfect tracking designs by carrying out the same tests of position reference tracking and load disturbance rejection described previously. The response of the balancing robot to a step reference input of  $\gamma_{ref} = 0.1/r \text{ rad} = 168.5^\circ$  corresponding to a longitudinal position displacement of 10 cm is plotted in Figure (14). The performance of the nominal state-space controller, combined with the yaw angle PI controller, can be observed in the upper graph. It is easy to verify how during the pure balancing phase (first 15 seconds) the yaw angle controller was able to maintain the robot's orientation within  $\psi \leq \pm 2^\circ$ . Once the step input reference was applied, larger oscillations in the yaw angle could be seen, however there was no drift towards a particular direction and the value stayed within  $\psi \leq \pm 4^\circ$ . In the lower graph of Figure (14) the performance of the robust state-space controller with integral action is plotted. In terms of the yaw angle  $\psi$ , the robot's behavior was analogous to that described previously during both the pure balancing phase and the longitudinal displacement step response phase. Overall the

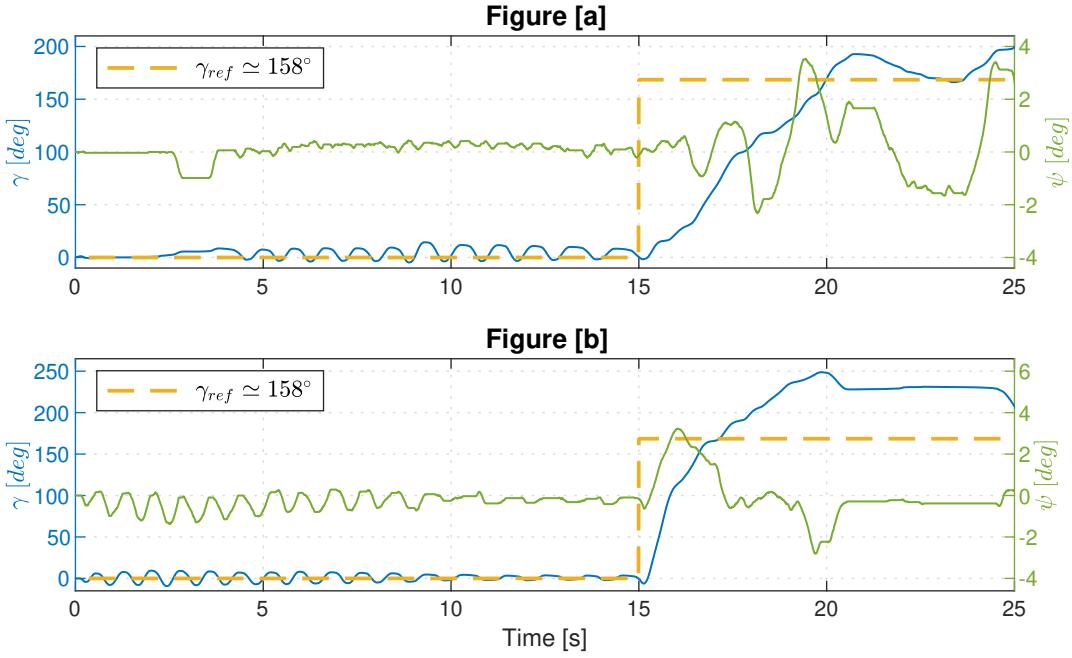


Figure 14: Wheel and yaw angles of the balancing robot with combined longitudinal and heading-angle control in response to a step reference input of  $\gamma_{ref} \simeq 168^\circ$ . Nominal design (a) and robust design (b).

implementation of the PI-based yaw control can be considered satisfactory in scenarios where the robot receives a longitudinal displacement input reference. The response of the system to a load disturbance is plotted in Figure (15).

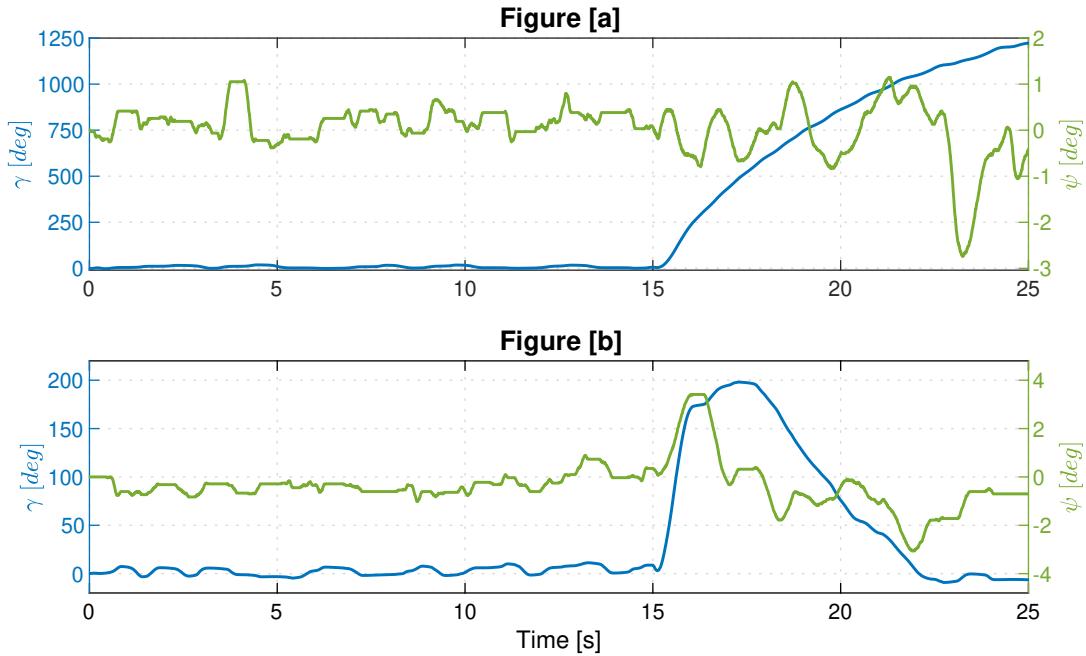


Figure 15: Wheel and yaw angles of the balancing robot with combined longitudinal and heading-angle control in response to a voltage disturbance  $V_{dist} = 5V$  applied at time  $t = 15\text{ s}$ . Nominal design (a) and robust design (b).

Similarly to what was described previously, the PI yaw angle controller was able to maintain the robot's heading angle within  $\psi \leq \pm 2^\circ$  during the pure balancing phase. During the response to the load disturbance, for both nominal (top) and robust (bottom) longitudinal control cases the yaw angle kept a value of  $\psi \leq \pm 4^\circ$ .

In conclusion, it can be said that the introduction of the simple PI controller was effective and allowed the achievement of the desired design goal of mitigating the drift and pivoting behaviors.

# A Appendix

## A.1 Complements of the mathematical model

In the following we explicit the mathematical details of the dynamical model of the balancing robot presented in the first section.

### A.1.1 Equations of motion (EoM for the longitudinal dynamics)

The equations-of-motion derived with the Lagrangian approach are the following:

$$\begin{aligned} & [2I_{w,yy} + 2N^2 I_{rot,yy} + (m_b + 2m_w + 2m_{rot})r^2]\ddot{\gamma} + 2(B + B_w)\dot{\gamma} + \dots \\ & \dots + [2N(1 - N)I_{rot,yy} + (m_b l + 2m_{rot}z_{rot}^b)r \cos \vartheta]\ddot{\vartheta} - 2B\dot{\vartheta} - \dots \\ & \dots - (m_b l + 2m_{rot}z_{rot}^b)r \sin \vartheta \dot{\vartheta}^2 - 2\tau = 0 \end{aligned} \quad (18)$$

$$\begin{aligned} & [2N(1 - N)I_{rot,yy} + (m_b l + 2m_{rot}z_{rot}^b)r \cos \vartheta]\ddot{\gamma} - 2B\dot{\gamma} + \dots \\ & \dots + [I_{b,yy} + 2(1 - N^2)I_{rot,yy} + m_b l^2 + 2m_{rot}(z_{rot}^b)^2]\ddot{\vartheta} + 2B\dot{\vartheta} - \dots \\ & \dots - (m_b l + 2m_{rot}z_{rot}^b)g \sin \vartheta \dot{\vartheta}^2 + 2\tau = 0 \end{aligned} \quad (19)$$

### A.1.2 Matrix formulation of the nonlinear dynamical model

The matrices of the compact form presented in (1) are the following three:

$$M(\mathbf{q}) = \begin{bmatrix} M_{11}(\mathbf{q}) & M_{12}(\mathbf{q}) \\ M_{21}(\mathbf{q}) & M_{22}(\mathbf{q}) \end{bmatrix}, \quad C(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} C_{11}(\mathbf{q}, \dot{\mathbf{q}}) & C_{12}(\mathbf{q}, \dot{\mathbf{q}}) \\ C_{21}(\mathbf{q}, \dot{\mathbf{q}}) & C_{22}(\mathbf{q}, \dot{\mathbf{q}}) \end{bmatrix}, \quad F_v = \begin{bmatrix} F_{v,11} & F_{v,12} \\ F_{v,21} & F_{v,22} \end{bmatrix}$$

with:

$$\begin{aligned} M_{11}(\mathbf{q}) &= 2I_{w,yy} + 2N^2 I_{rot,yy} + (m_b + 2m_w + 2m_{rot})r^2 \\ M_{12}(\mathbf{q}) &= M_{21}(\mathbf{q}) = 2N(1 - N)I_{rot,yy} + (m_b l + 2m_{rot}z_{rot}^b)r \cos \vartheta \\ M_{22}(\mathbf{q}) &= I_{b,yy} + 2(1 - N^2)I_{rot,yy} + m_b l^2 + 2m_{rot}(z_{rot}^b)^2 \end{aligned}$$

$$\begin{aligned} C_{11}(\mathbf{q}, \dot{\mathbf{q}}) &= C_{21}(\mathbf{q}, \dot{\mathbf{q}}) = C_{22}(\mathbf{q}, \dot{\mathbf{q}}) = 0 \\ C_{12}(\mathbf{q}, \dot{\mathbf{q}}) &= - (m_b l + 2m_{rot}z_{rot}^b) r \sin(\vartheta) \dot{\vartheta} \end{aligned}$$

$$\begin{aligned} F_{v,11} &= 2(B + B_w) \\ F_{v,12} &= F_{v,21} = -2B \\ F_{v,22} &= 2B \end{aligned}$$

plus the torque contribution due to gravity:

$$\mathbf{g}(\mathbf{q}) = \begin{bmatrix} 0 & -\left(m_b l + 2m_{rot} z_{rot}^b\right) g \sin(\vartheta) \end{bmatrix}^T.$$

### A.1.3 Matrix formulation of the linear dynamical model

The matrices of the compact form presented in (2) are the following two:

$$\mathbf{M} = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix}$$

with

$$\begin{aligned} M_{11} &= 2I_{w,yy} + 2N^2 I_{rot,yy} + (m_b + 2m_w + 2m_{rot}) r^2 \\ M_{12} &= m_{21} = 2N(1 - N)I_{rot,yy} + (m_b l + 2m_{rot} z_{rot}^b) r \\ M_{22} &= I_{b,yy} + 2(1 - N)^2 I_{rot,yy} + m_b l^2 + 2m_{rot}(z_{rot}^b)^2 \end{aligned}$$

$$\begin{aligned} G_{11} &= G_{12} = G_{21} = 0 \\ G_{22} &= -\left(m_b l + 2m_{rot} z_{rot}^b\right) g. \end{aligned}$$

### A.1.4 State-space model taking into account the actuation system

The matrices appearing in (1.2) and (6) are:

$$\mathbf{F}'_v = \mathbf{F}_v + \frac{2N^2 k_t k_e}{R_a} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad \text{and} \quad \boldsymbol{\tau}' = \frac{2N k_t}{r_a} \begin{bmatrix} 1 \\ -1 \end{bmatrix} u_a.$$

## A.2 Some details on the performances of the model and of the real system based on the choice of $\rho$

As anticipated in the previous sections, in the following will be presented a comparison, both for the nominal and the robust controller, between the behaviours of the system's model (i.e. in the simulation environment) when choosing the value of  $\rho$  among the two suggested in the Handout.

Before going on, we must state now that no crucial improvement on the performances was obtained by using one over the other: the definitive choice was made 'on the spot' in the laboratory by observing which of the two led the real system to behave better.

It would have been interesting to compare the different results obtained directly on the robot, but unfortunately the data-collection process was not carried out with this in mind. We will thus make due with the information we can extract from the simulations.

### A.2.1 Nominal case

For the nominal controller, the results produced on the Simulink model are presented in Figure (16) below. On the left column are the comparisons between the values of  $\gamma$  ( $\gamma_{est}$ ) using  $\rho = 500$  and  $\rho = 5000$ , for all of the three tasks that were assigned. On the right, the values of  $\vartheta$  ( $\vartheta_{est}$ ) were compared, again for all the tests. The results comparing the wheel angles  $\gamma$

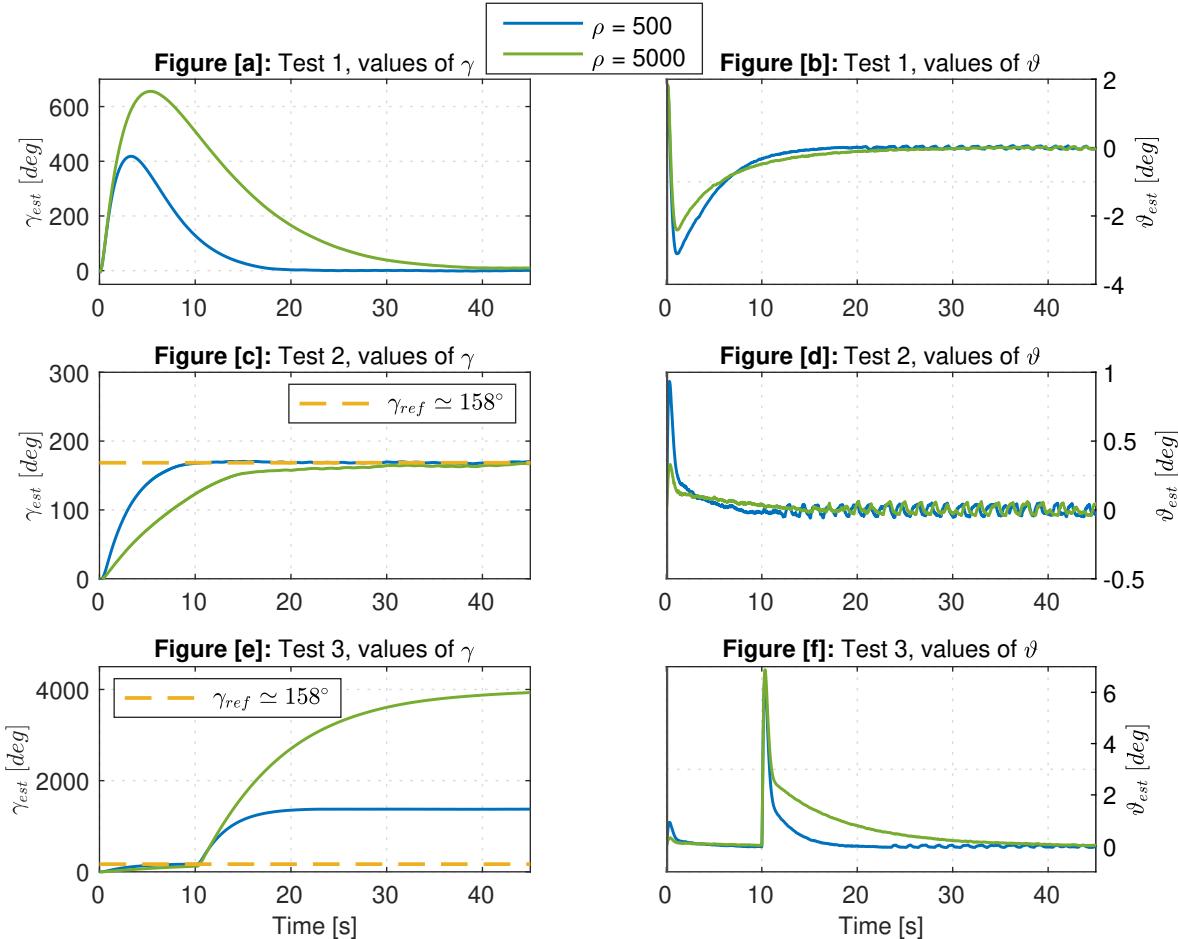


Figure 16: Comparison of the responses for the three tests when using  $\rho = 500$  vs  $\rho = 5000$  (nominal controller).

show that for  $\rho = 5000$  the system reacts slower and takes more time to settle around the reference point.

Since, as already discussed,  $\rho$  controls the ratio between the cost on the input and the one on the state, this behaviour has the intuitive explanation that an higher value of  $\rho$  puts an higher weight on the input and thus the controller will act by limiting the control effort employed as much as possible.

The plots of the tilt angle  $\vartheta$ , on the other hand, are not very informative since the results look similar or even almost identical. This is due to the fact that the stabilization around the vertical equilibrium of the robot's vertical body does not require much 'wheel work' and is thus only slightly dependent on the input effort.

As already mentioned several times, the real system was tested with both values of  $\rho$  and for the nominal controller the choice fell on the lower of the two because it performed better

(faster) and with no significant trade-off on the 'sharpness' of the response, as it happened in the robust case which will be developed next.

### A.2.2 Robust case

The results obtained in the simulations are shown in Figure (17).

No important difference can be spotted with the nominal controller: an higher value for  $\rho$  proved again to penalize the promptness of the system in following the step reference (even in the presence of noise). Once more, no relevant information can be extracted from the plots comparing the tilt angle. On the real setup, the value  $\rho = 5000$  was preferred since for

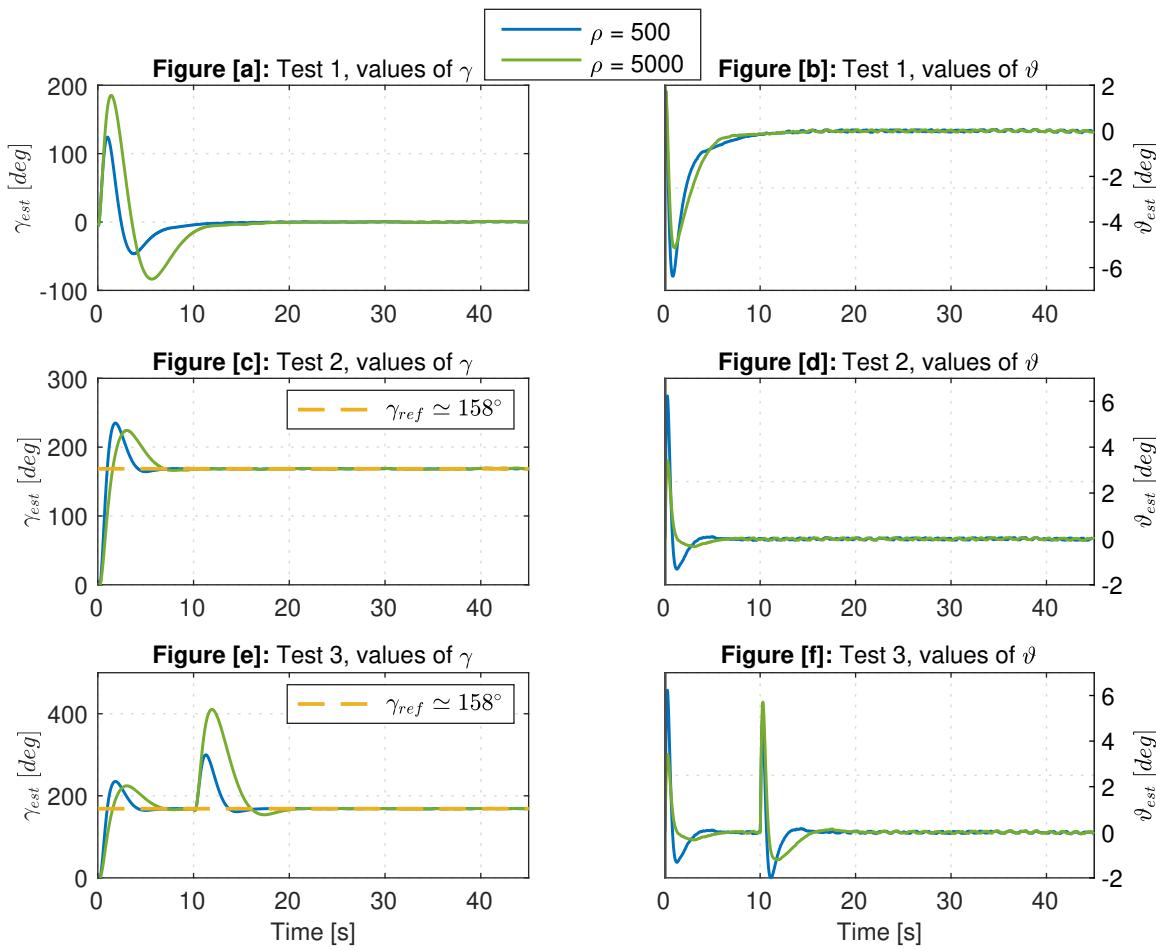


Figure 17: Comparison of the responses for the three tests when using  $\rho = 500$  vs  $\rho = 5000$  (robust controller).

$\rho = 500$  the controller output kept saturating for unknown reasons; this led the robot to have a 'shaky' behaviour, even though it could still track the step reference and re-position itself vertically.

This degradation of the performances could have been caused by an inappropriate choice on the cut-off frequency (0.35 [Hz]), since the instability observed looked like an high-frequency limit cycle, or maybe a mismatch between the model used for the control design and the real plant dynamics, or even by some other implementation errors which were scrupulously 'hunted' but not found.

However, since a cause-effect could not be established during the tests, the decision was to just go with the value  $\rho = 5000$ .

### A.3 Comments on the velocities obtained in Test 2

In Figures (4) and (10) it could be seen that the plots of the *estimated* velocities seemed to have a strange behaviour in the high frequencies; this could be, for example, due to the filters in the observer amplifying some high-frequency signal, for example a noise.

To verify this hypothesis, a tentative was made to lower the pass-band of the filters: the block diagram of Figure (25a) was modified by introducing a 0.5 **gain** block between both the **Speed filter** blocks and the output; higher values for the gain resulted in the loss of stability ( $\gamma_{est}$  diverging).

The results of this procedure are shown in Figure (18) and (19). In both cases, the problem

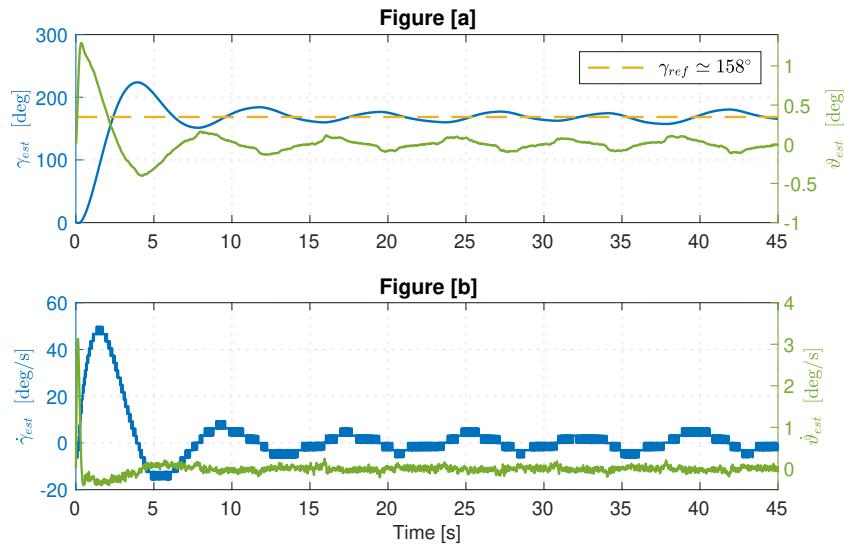


Figure 18: Simulation of Test 2 using modified filters (nominal case)

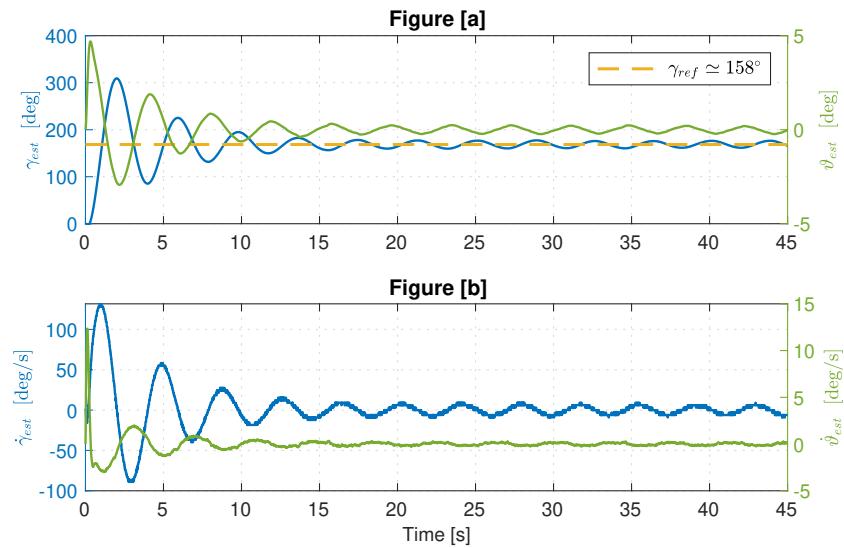


Figure 19: Simulation of Test 2 using modified filters (robust case)

did not disappear, but at least it was slightly mitigated. The system is still able to track the reference signal, even though the price to pay seems to be in the fact that the tilt-angle-adjusting capability of the controller gets (very mildly) deteriorated.

## A.4 Simulink model

Below are the Simulink blocks used to implement the model of the system.

First, the overall model will be shown, and then each block will be reported starting from the voltage driver to the state-space controller.

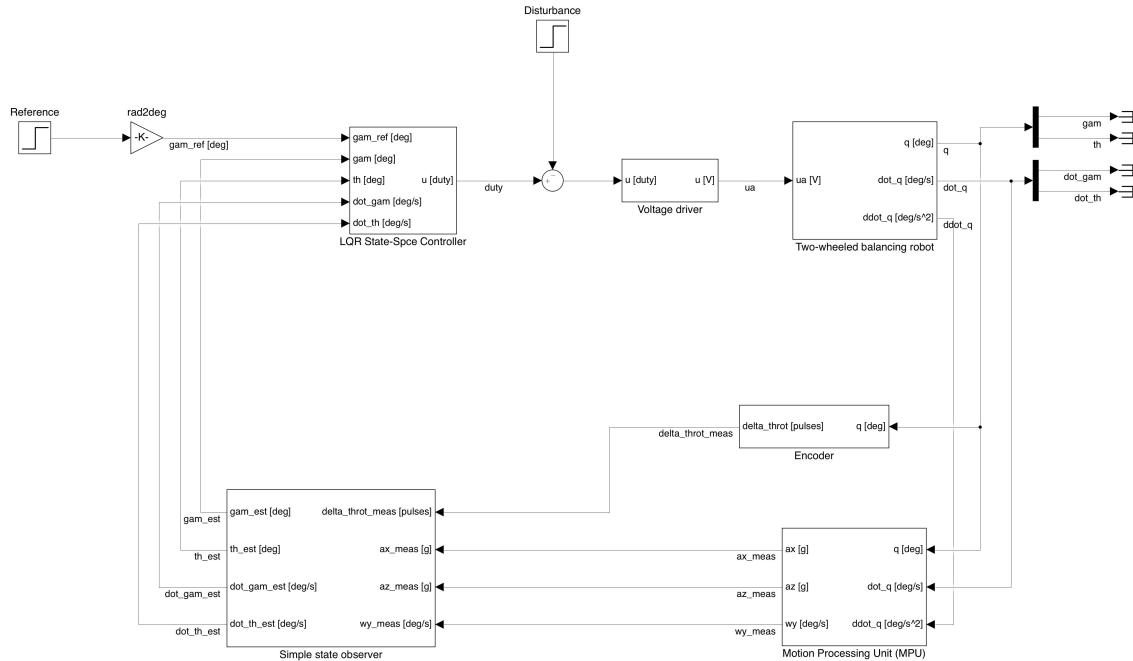


Figure 20: Balance and position state-space control system.

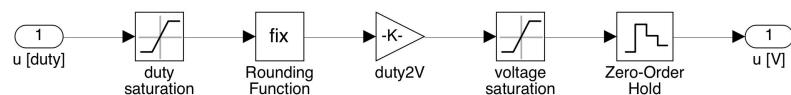


Figure 21: Voltage driver.

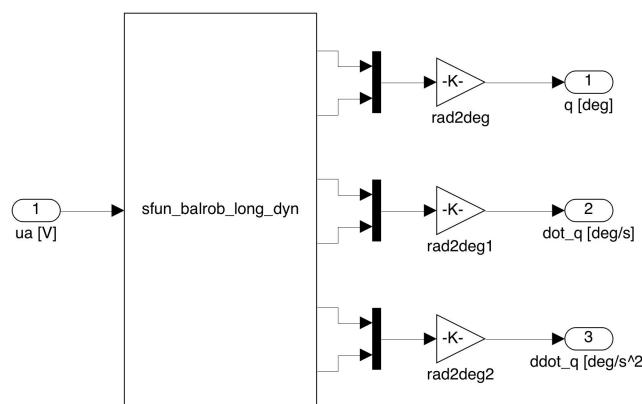


Figure 22: Two wheeled balancing robot.

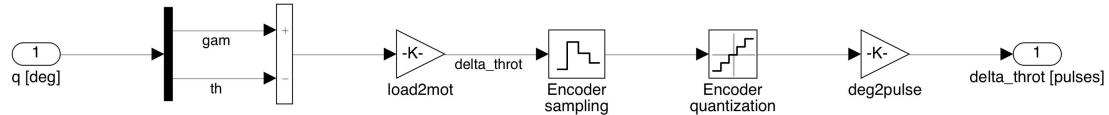
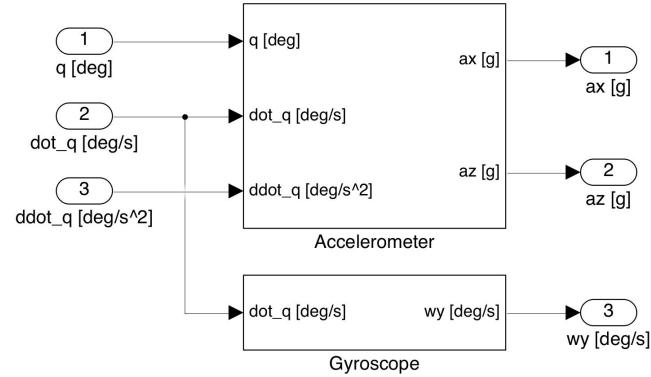
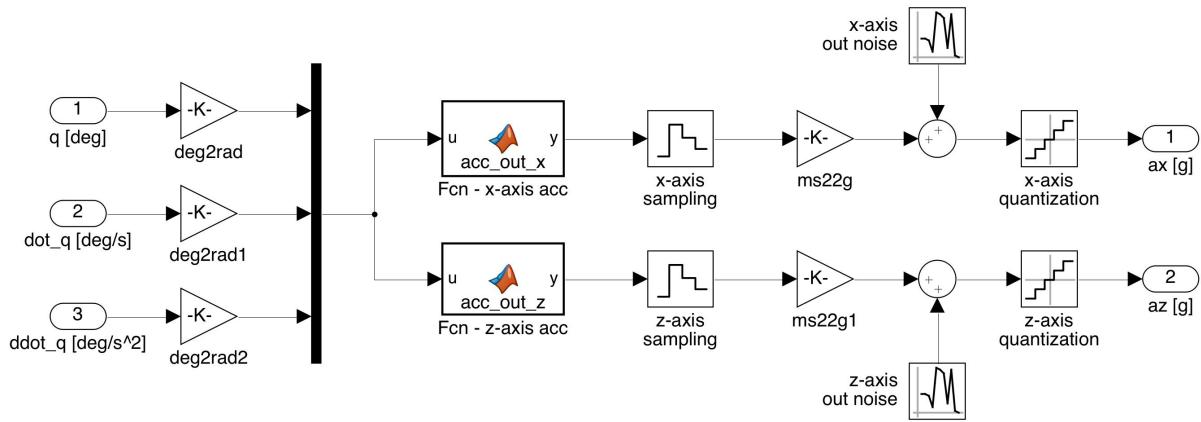


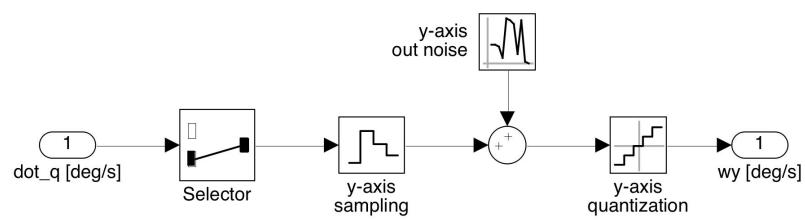
Figure 23: Encoder.



(a) Motion processing unit (MPU).

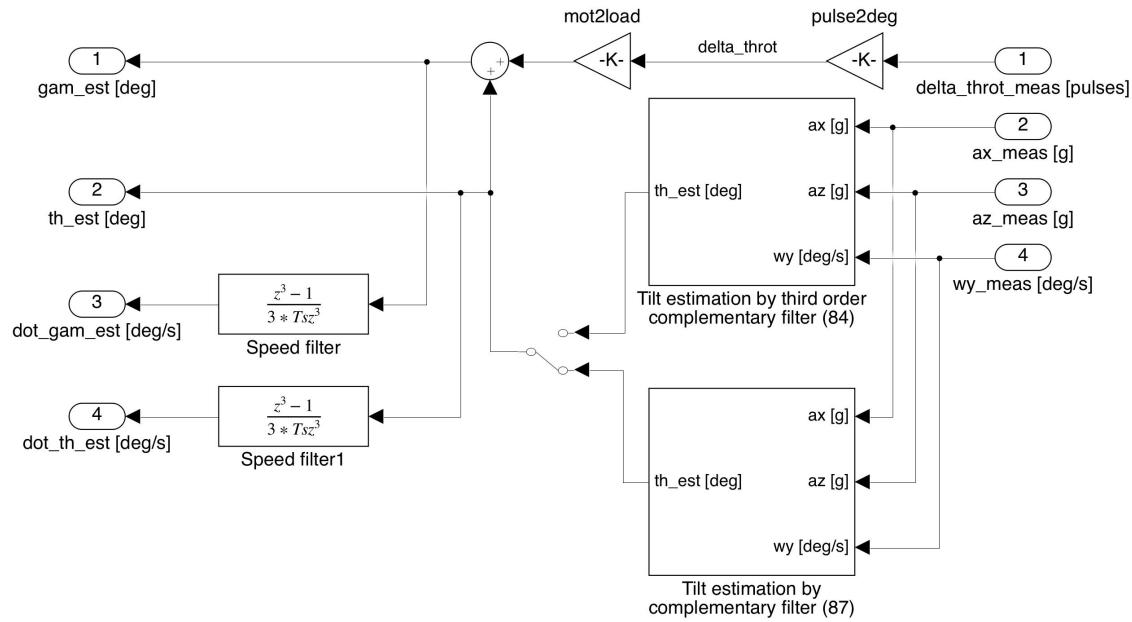


(b) Accelerometer.

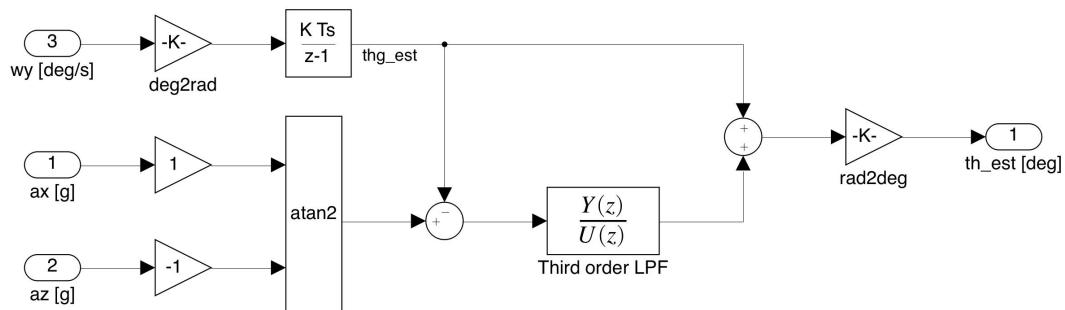


(c) Gyroscope.

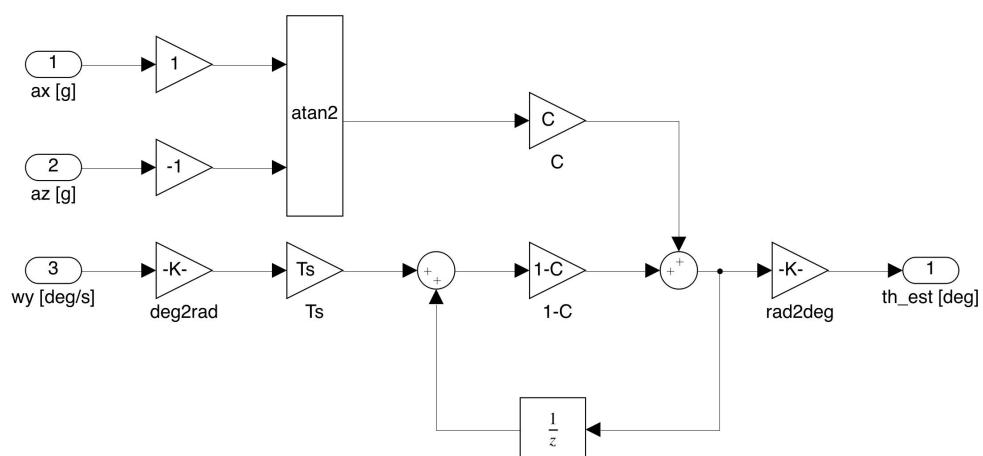
Figure 24: Simulink model implementation details: MPU.



(a) Simple state observer.

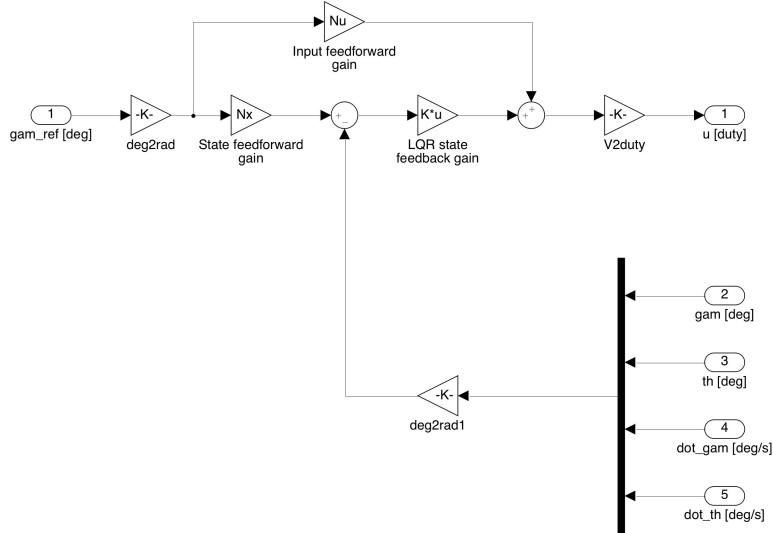


(b) Tilt estimation based on complementary filtering (implemented based on (84)).

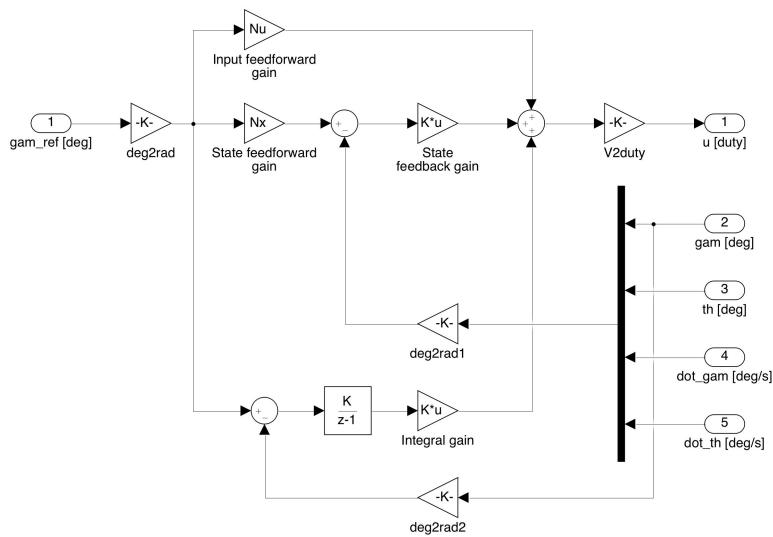


(c) Tilt estimation based on complementary filtering (implemented based on (87)).

Figure 25: Simulink implementation of the state observer and possible design of the complementary filter (the one in (c) was actually used in this activity).



(a) Nominal state-space controller.



(b) Robust state space controller.

Figure 26: Simulink implementation of the nominal (a) and robust (b) state-space controller.

## A.5 Data sheet

In the following is presented the data sheet of the physical system. Not all of the quantities present in the tables have been explicitly cited in this report, but they all appear in the derivation of the model.

<b>• Robot body</b>				
Center-of-Mass coords wrt body frame $\{b\}$	$x_b^b, y_b^b, z_b^b$	0, 0, 46.05	[mm]	
Mass	$m_b$	1.06	[kg]	
Principal Moments-of-Inertia	$I_{b,xx}, I_{b,yy}, I_{b,zz}$	4.22, 2.20, 2.65	[gm <sup>2</sup> ]	
<b>↳ Robot chassis</b>				
Dimensions (width, height, depth)	$w_c, h_c, d_c$	160, 119, 80	[mm]	
Center-of-Mass coords wrt body frame $\{b\}$	$x_c^b, y_c^b, z_c^b$	0, 0, 80	[mm]	
Mass	$m_c$	456	[g]	
Principal Moments-of-Inertia	$I_{c,xx}, I_{c,yy}, I_{c,zz}$	1.5, 0.78, 1.2	[gm <sup>2</sup> ]	
<b>↳ Battery</b>				
Dimensions (width, height, depth)	$w_{batt}, h_{batt}, d_{batt}$	136, 26, 44	[mm]	
Center-of-Mass coords wrt body frame $\{b\}$	$x_{batt}^b, y_{batt}^b, z_{batt}^b$	0, 0, 44	[mm]	
Mass	$m_{batt}$	320	[g]	
Principal Moments-of-Inertia	$I_{batt,xx}, I_{batt,yy}, I_{batt,zz}$	0.51, 0.07, 0.06	[gm <sup>2</sup> ]	
<b>↳ DC gearmotor stator</b>				
Dimensions (height, radius)	$h_{stat}, r_{stat}$	68.1, 17	[mm]	
Center-of-Mass coords wrt body frame $\{b\}$	$x_{stat}^b, y_{stat}^b, z_{stat}^b$	0, ±52.1, -7	[mm]	
Mass	$m_{stat}$	139.75	[g]	
Principal Moments-of-Inertia	$I_{stat,xx} = I_{stat,zz}, I_{stat,yy}$	0.064, 0.02	[gm <sup>2</sup> ]	
<b>• DC gearmotor rotor</b>				
Dimensions (height, radius)	$h_{rot}, r_{rot}$	30.7, 15.3	[mm]	
Center-of-Mass coords wrt body frame $\{b\}$	$x_{rot}^b, y_{rot}^b, z_{rot}^b$	0, ±42.7, -7	[mm]	
Mass	$m_{rot}$	75.25	[g]	
Principal Moments-of-Inertia	$I_{rot,xx} = I_{rot,zz}, I_{rot,yy}$	0.01, 0.009	[gm <sup>2</sup> ]	
<b>• Wheels</b>				
Dimensions (height, radius)	$h_w, r_w$	26, 34	[mm]	
Center-of-Mass coords wrt body frame $\{b\}$	$x_w^b, y_w^b, z_w^b$	0, ±100, 0	[mm]	
Mass	$m_w$	50	[g]	
Principal Moments-of-Inertia	$I_{w,xx} = I_{w,zz}, I_{w,yy}$	0.017, 0.029	[gm <sup>2</sup> ]	

Table 1: Geometrical and inertial nominal parameters.

Gearbox viscous friction coefficient (at output shaft)	$B$	$25 \times 10^{-3}$ Nm/(rad/s)
Wheel viscous friction coefficient	$B_w$	$1.5 \times 10^{-3}$ Nm/(rad/s)

Table 2: Viscous friction nominal parameters.

Armature resistance	$R_a$	$2.4 \Omega$
Armature inductance	$L_a$	n.a. (neglected)
Electric (BEMF) constant	$k_e$	$10.3 \times 10^{-3}$ Vs/rad
Torque constant	$k_m$	$5.2 \times 10^{-3}$ Nm/A
Gearbox ratio	$N$	30

Table 3: DC gearmotor nominal parameters.

Sensor center wrt body frame $\{b\}$	$x_{mpu}^b, y_{mpu}^b, z_{mpu}^b$	0, 0, 13.5	[mm]
--------------------------------------	-----------------------------------	------------	------

Table 4: Motion processing unit (MPU) geometrical parameters.