

Autonomous and Mobile Robotics:
Project Overview

External force observers for the NAO

Analysis and C++ Implementation of a Testbed for External Force Observers acting on the NAO robot.

Postolache Emilian, 1649271
Ratini Riccardo, 1656801
Simionato Giada, 1822614

Contents

1	Introduction	2
2	External Force Observers	3
2.1	Luenberger Observer	3
2.2	Kalman Filter Based Observer	4
2.3	Stephens' Observer	7
2.4	Theoretical Comparison	8
3	Implementation Details	10
3.1	Framework Background	10
3.2	Observers Implementation	12
3.3	Deployment	13
4	Experiments and Results	16
4.1	Simulations Results	17
4.1.1	Stand without Balance	17
4.1.2	Stand with Balance	22
4.1.3	Walking	27
4.1.4	Experimental Comparison	31
4.2	Experimental Considerations	34
5	Conclusions	38
References		38
A	Supplementary Material	40

Chapter 1

Introduction

Humanoid robots are increasingly gaining popularity, both in the research community and in the industry, due to their ability in manipulating objects while moving and accomplishing diverse tasks difficult for wheeled robots such as walking on rough terrains.

Among all the problems in the control of humanoid robots, maintaining balance while walking is one of the most relevant [1]. The most common and challenging expression of this problem is when the robot has to deal with unknown external forces acting upon it. The reasons why this may happen are manifold: from the collaboration with humans for performing a task, such as moving an heavy object, or by carrying out a manipulation task with an oscillating weight [4]. Therefore the knowledge of the external force perturbing the state of the robot is fundamental in the design of a controller able to counteract it to maintain a stable gait, especially for robots with small footprints [1, 2]. If the force is unknown, however, precisely estimating it becomes crucial: this estimate usually relies on expensive Force/Torque (F/T) sensors able to directly measure the external force acting on the robot [3, 4]. However, the most widespread type of humanoid robots are those smaller and less expensive, such as the family of the NAO robots by Softbank, that lacks such type of sensors. To overcome this problem a variety of methods for estimating these quantities using the available sensors were developed [2, 4, 5].

With this aim, in this work we have implemented three state-disturbance observers, namely a *Luenberger* observer [2], a *Kalman Filter based* observer [4] and *Stephens'* observer [5] to provide precise estimates of an unknown external force acting on a robot. Moreover, we have inserted the above mentioned implementations in a common wrapper and provided external interfaces to control the experiments. We exhaustively validated this testbed on the NAO robot using DART simulator. We have adapted our work to be easily used and extended by future researchers.

This report is organized as follows. In Chapter 2 there will be a theoretical overview of all the observers implemented in this work, along with their comparison, while in Chapter 3 all the details about the implementation of the testbed will be discussed. In Chapter 4 all the performed experiments along with their results will be explained and commented. In addition, we will discuss the most interesting behaviours that we have encountered. Finally, in Chapter 5 conclusions will be drawn.

Chapter 2

External Force Observers

The aim of this work was to provide a precise estimate of an external force acting on the NAO robot through the implementation of particular observers. To cope with the difficulties in modeling such complex humanoid structure, all the observers relied on the perturbed *Linear Inverted Pendulum* model (LIPM). Denoting by $\mathbf{X}_c = (x_c, y_c, z_c)$ the coordinates of the Center of Mass (CoM), by $\mathbf{p} = (x_z, y_z)$ the non-zero ones of the Zero Moment Point (ZMP) and by $\mathbf{F}_{ext} = (F_x, F_y, F_z)$ the components of the external force, that scaled by the mass of the robot represent a quantity in the more general disturbance $\mathbf{w} = (w_x, w_y, w_z)$ collecting all the the unmodeled dynamics, the continuous version of the LIPM for the sagittal motion is

$$\ddot{x}_c = \eta^2(x_c - x_z) + w_x, \quad (2.1)$$

where $\eta = \sqrt{g/\bar{z}_c}$, that highlights the constant value of the height of the CoM, i.e. one of the basic assumptions of the LIPM. One of the main advantages of this model is its simplicity yet its precision in approximating the dynamics of a humanoid body, as well as providing a linear, identical, decoupled dynamics for the x and y axes, exploited during the construction of the observers.

In the next sections all the observers implemented in this work will be briefly introduced and compared from a theoretical point of view.

2.1 Luenberger Observer

Inspired by the work in [2], the *Luenberger* observer uses the system¹ output to steer its estimate of the state of the system.

Starting from the generic continuous LTI system in Eq. (2.3), the extension of the perturbed LIP model adopted in this work, allowed us to assess the structure of the system matrices for the sagittal motion as follows

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ \eta^2 & 0 & -\eta^2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{C} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}^T \quad (2.2)$$

and to identify the components of the state, input and output vectors as in Eq. (2.4).

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu} \\ \mathbf{y} = \mathbf{Cx} \end{cases} \quad (2.3)$$

¹The system must be observable.

$$\begin{aligned}\mathbf{x} &= (x_c \ x_c \dot{x}_c \ x_z \ w_x \ \dot{w}_x) \\ \mathbf{y} &= (x_c \ x_z) \\ \mathbf{u} &= \dot{x}_z\end{aligned}\tag{2.4}$$

It is worth noticing that the state vector contains the position and the velocity of the CoM, the position of the ZMP, the disturbance and its derivative².

The Luenberger observer was then obtained through the addition of a correction term, determined by the difference between the actual and estimated outputs and then weighted by a *gain* matrix \mathbf{G} , to the original dynamics equation in Eq. (2.3). Formally the observer takes the form:

$$\begin{cases} \dot{\hat{\mathbf{x}}} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}\mathbf{u} + \mathbf{G}(\mathbf{y} - \hat{\mathbf{y}}) \\ \hat{\mathbf{y}} = \mathbf{C}\hat{\mathbf{x}}, \end{cases}\tag{2.5}$$

where $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ denote the vectors containing the estimates of the state and output quantities, respectively.

This observer is *asymptotically stable* if the error $\mathbf{e}(t) = \hat{\mathbf{x}}(t) - \mathbf{x}(t)$ tends to zero when $t \rightarrow \infty$. By observing the error dynamics, i.e. $\dot{\mathbf{e}} = (\mathbf{A} - \mathbf{GC})\mathbf{e}$, it is straightforward to see that this happens when $\mathbf{A} - \mathbf{GC}$ is a stable matrix: to satisfy this condition its eigenvalues must have strictly negative real part and, since \mathbf{A} and \mathbf{C} are fixed and imposed by the model, a convenient way to accomplish this is to choose \mathbf{G} through a fine pole placement (more on this can be found in Section 3.2).

2.2 Kalman Filter Based Observer

Another observer implemented and validated in this work was the *Kalman Filter based observer*, inspired by the work in [4]. Based on the Kalman Filter technique, this observer relies on the *discretized* perturbed LIP model. Each of the three components of the external force \mathbf{F}_{ext} was estimated through a different observer, all derived from the dynamics implied by the chosen model. Fundamental in the Kalman Filter matrices definition was the computation of the ZMP as follows

$$\mathbf{p} = \mathbf{N} \frac{\mathbf{n} \times \boldsymbol{\tau}^o}{f_n^o},\tag{2.6}$$

where \mathbf{N} is a constant matrix aimed at removing the third component and \mathbf{n} is the normal vector to the ground, as in

$$\mathbf{N} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \quad \mathbf{n} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.\tag{2.7}$$

The resulting force \mathbf{f}^o from the inertial, gravity and external forces, projected onto the normal vector, i.e. f_n^o , can be expressed as in Eq. (2.8), making explicit its direct dependency from the z -axis component of the external force³.

$$f_n^o = -M_c g - M_c \ddot{z}_c + F_z\tag{2.8}$$

²It was assumed that the external disturbance was generated by an *exosystem* of type $\dot{\mathbf{w}} = \mathbf{0}$, that means forces with null acceleration or at most piecewise linear. This choice was motivated by the assumption made in the IS-MPC of piecewise linear ZMP. More on the IS-MPC controller can be found in Section 3.1.

³ M_c is the total mass of the robot.

Finally, τ^o denotes the moment of the external force about the origin of the world frame, given as

$$\tau^o = \mathbf{X}_c \times \mathbf{f}^o. \quad (2.9)$$

Putting Eq. (2.7), (2.8) and (2.9) into Eq. (2.6), results in Eq. (2.10), obtaining an explicit dependency also for the other two components of the external force, as Eq. (2.8) did for the z -axis:

$$\mathbf{p} = \begin{pmatrix} x_c + \frac{M_c z_c}{f_n^o} \ddot{x}_c - \frac{z_c}{f_n^o} F_x \\ y_c + \frac{M_c z_c}{f_n^o} \ddot{y}_c - \frac{z_c}{f_n^o} F_y \end{pmatrix}. \quad (2.10)$$

As previously mentioned, each dimension had its own observer and the first quantity to be estimated was F_z . In order to express the problem of the disturbance estimation in a suitable form for the Kalman Filter application, the system in Eq. (2.1) was discretized and the inputs \mathbf{u} , namely the *jerk* \ddot{z}_c of the CoM and the *second derivative of the external force* \dot{F}_z , were treated as part of the Gaussian process noise. The obtained system is reported as follows

$$\begin{cases} \mathbf{\mathcal{X}}_z(k+1) = \mathbf{A}\mathbf{\mathcal{X}}_z(k) + \boldsymbol{\omega}_z(k) \\ \mathbf{Y}_z(k) = \mathbf{C}_z\mathbf{\mathcal{X}}_z(k) + \mathbf{v}_z(k) \end{cases} \quad (2.11)$$

where the state vector⁴ $\mathbf{\mathcal{X}}_z(k) = [z_c(kT) \ \dot{z}_c(kT) \ \ddot{z}_c(kT) \ F_z(kT) \ \dot{F}_z(kT)]^T$ contained the z -coordinates of the CoM position, velocity and acceleration, the external force and its derivative, respectively. The output signal, instead, was composed by the z -coordinates of the CoM position, acceleration and by the ground reaction force plus a fixed term, i.e.

$\mathbf{Y}_z(k) = [z_c(kT) \ \ddot{z}_c(kT) \ f_n^o(kT) + M_cg]^T$. $\boldsymbol{\omega}_z$ and \mathbf{v}_z represented respectively the process and measurement noises, both assumed white Gaussian noises with covariance matrices \mathbf{Q} and \mathbf{R}_z . The latter was computed including the covariances of the inputs, i.e. $\sigma_{\ddot{z}_c}^2$ and $\sigma_{\dot{F}_z}^2$, and the matrix \mathbf{B} , expressing the relation between the input and the state vectors, as follows

$$\mathbf{Q} = \mathbf{B} \begin{pmatrix} \sigma_{\ddot{z}_c}^2 & 0 \\ 0 & \sigma_{\dot{F}_z}^2 \end{pmatrix} \mathbf{B}^T. \quad (2.12)$$

Finally, the matrices involved in Eq. (2.11), both explicitly (\mathbf{A} and \mathbf{C}_z) and implicitly (\mathbf{B}), are defined as follows, where it is worth noticing the structure of \mathbf{C}_z , that conveys the relation in Eq. (2.8), fundamental to estimate F_z :

$$\mathbf{A} = \begin{pmatrix} 1 & T & \frac{T^2}{2} & 0 & 0 \\ 0 & 1 & T & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} \frac{T^3}{6} & 0 \\ \frac{T^2}{2} & 0 \\ T & 0 \\ 0 & \frac{T^2}{2} \\ 0 & T \end{pmatrix}, \quad \mathbf{C}_z = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -M_c & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}^T. \quad (2.13)$$

By applying the Kalman Filter equations to the system in Eq. (2.11), it was possible to obtain an estimate of the state vector, thus containing the z -component of the unknown external force F_{ext} :

$$\hat{\mathbf{\mathcal{X}}}_z(k) = [\hat{z}_c(kT) \ \hat{\dot{z}}_c(kT) \ \hat{\ddot{z}}_c(kT) \ \hat{F}_z(kT) \ \hat{\dot{F}}_z(kT)]^T. \quad (2.14)$$

As regards the others two components of the external force, namely F_x and F_y , it was possible to exploit the behaviour of the system, whose dynamics is identical and decoupled along these two axes: for this reason, only the observer for the former will be presented here⁵. In order to

⁴T is the sampling rate of the discretized system.

⁵The observer for the y component can be easily derived by replacing the x -quantities with the corresponding y ones.

estimate F_x , [4] used the relation in Eq. (2.10), similarly to what was done for F_z with Eq. (2.8), and a state vector whose only difference with respect to the z -observer was the presence of the x coordinate in the state elements, in place of the z one. Also \mathbf{A} , \mathbf{Q} , \mathbf{v} and $\boldsymbol{\omega}$ were defined in the same way, with the exception of a change in the coordinates. However, the output signal, and therefore the matrix \mathbf{C} , were defined in a different way, in order to express the relation in Eq. (2.10), as follows

$$\mathbf{Y}_x(k) = [x_c(kT) \ \ddot{x}_c(kT) \ x_z(kT)]^T,$$

$$\mathbf{C}_x = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & \frac{M_c \dot{z}_c(kT)}{\hat{f}_n^o(kT)} & -\frac{\dot{z}_c(kT)}{\hat{f}_n^o(kT)} & 0 \end{pmatrix}. \quad (2.15)$$

The structure of the \mathbf{C}_x matrix revealed the dependency of the x and y observers from the z one. The quantities \hat{z}_c and \hat{f}_n^o , for each instant kT , are the estimated values obtained through the z observer. Recalling Eq. (2.8), through the substitution of its quantities with their estimated counterparts, including \hat{F}_z , it is possible to obtain an estimate of the ground reaction force, and therefore computing the elements in \mathbf{C}_x . Finally, by applying the Kalman Filter equations it was possible to obtain the full estimate of the state vector for the x (and y) components. How the quantities needed to update the estimates of the states of the observers were retrieved, such as the coordinates of the ZMP, the position and acceleration of the CoM and the ground reaction force, will be disclosed in Section 3.1.

In conclusion, it is worth pointing out that the assumption of applying the external force exclusively to the robot center of mass, does not imply a lack of generality. Applying the disturbance to another point of the robot body, such as to one of its arms, will generate an additional external moment $\boldsymbol{\tau}^{ext}$ to the CoM⁶: by substituting this updated term into Eq. (2.6), and after some computations and regroups, it is possible to obtain Eq. (2.16).

$$\mathbf{p} = \begin{pmatrix} x_c + \frac{M_c z_c}{f_n^o} \ddot{x}_c - \frac{z_c}{f_n^o} (F_x + \frac{\tau_y^{ext}}{z_c}) \\ y_c + \frac{M_c z_c}{f_n^o} \ddot{y}_c - \frac{z_c}{f_n^o} (F_y + \frac{\tau_x^{ext}}{z_c}) \end{pmatrix} \quad (2.16)$$

The previous equation suggests that, without additional sensors, it is not possible to discern between the components of the external force and the additional moments and therefore giving a precise estimate of the unknown disturbance. However it is possible to consider the overall term, i.e. the external force applied to a generic point of the robot plus the associated external moments, as a *virtual force* applied to the center of mass of the robot, as highlighted in [4] and formally stated in Eq. (2.17). For this reason in this work it was followed the decision of [4] to consider only external forces applied to the robot CoM.

$$\tilde{\mathbf{F}}_{ext} = \begin{pmatrix} \tilde{F}_x \\ \tilde{F}_y \\ \tilde{F}_z \end{pmatrix} \doteq \begin{pmatrix} F_x + \frac{\tau_y^{ext}}{z_c} \\ F_y - \frac{\tau_x^{ext}}{z_c} \\ F_z \end{pmatrix} \quad (2.17)$$

The complete structure of this Kalman Filter based observer is reported in Fig. 2.1.

⁶This implies that Eq. (2.9) becomes $\boldsymbol{\tau}^o = \mathbf{X}_c \times \mathbf{f}^o + \boldsymbol{\tau}^{ext}$.

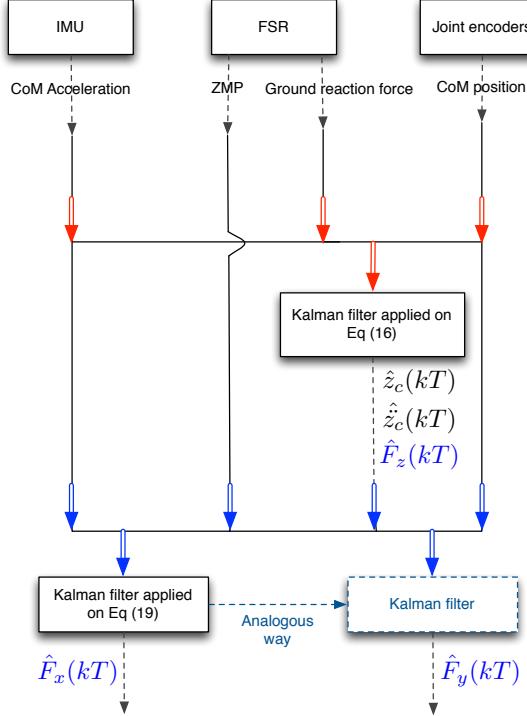


Figure 2.1. Structure of the Kalman Filter based observer for estimating an unknown external force applied to the robot center of mass.

2.3 Stephens' Observer

The *Stephens*⁷ observer was implemented according to [5], that presented a variety of sensing and error models, that could be combined upon different structures of estimators, to provide an estimate of either the magnitude of an external force or an unknown CoM offset.

This observer, as the Kalman Filter based one, relies of the discretized perturbed LIP model, provides estimates through the KF equations and, as the KF observer again, it is possible to augment the system with the process and measurement noises to formulate it as a KF, similarly to what was done in Eq. (2.11). However, differently from the previous observer, the inputs are now explicitly part of the system dynamics:

$$\begin{cases} \mathbf{X}(k+1) = \mathbf{A}\mathbf{X}(k) + \mathbf{B}\mathbf{u}(k) + \boldsymbol{\omega}(k) \\ \mathbf{Y}(k) = \mathbf{C}\mathbf{X}(k) + \mathbf{v}(k). \end{cases} \quad (2.18)$$

It is worth noticing that, differently from the KF based observer, there are no subscripts that indicate to which component the quantities are referring. This is due to the fact that there is no connection among the functioning of the three observers aimed to estimate the state along each dimension: the x , y and z observers are independent from each others.

As above mentioned, the LIP model used for this observer indicated the system matrices as:

$$\mathbf{A} = \begin{pmatrix} 1 & T & 0 & 0 \\ \eta^2 T & 1 & -\eta^2 T & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 0 \\ 0 \\ T \\ 0 \end{pmatrix}, \quad \mathbf{C} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \end{pmatrix}^T, \quad (2.19)$$

⁷We've coined this name by the author of the paper from which it was taken, to distinguish it from the Kalman Filter based observer since both used the KF equations.

where T stands again for the sampling rate.

In this observer, for each coordinate, the state was composed by the CoM position and velocity, the *Center of Pressure* (CoP) position, i.e. (x_p, y_p) , and the disturbance \mathbf{F}_{ext} . The input was the CoP velocity while the CoM and CoP position coordinates were used as measurements. Formally, e.g. for the x component:

$$\begin{aligned}\mathbf{\mathcal{X}}_x &= (x_c \quad \dot{x}_c \quad x_p \quad F_x), \\ \mathbf{Y}_x &= (x_c \quad x_p), \\ u_x &= \dot{x}_p.\end{aligned}\tag{2.20}$$

The noise processes are assumed to be Gaussian, hence $\boldsymbol{\omega} \sim \mathcal{N}(0, \mathbf{Q})$ for the process and $\mathbf{v} \sim \mathcal{N}(0, \mathbf{R})$ for the measurements, with the values of the covariance matrices found empirically and shown in Section 3.2.

How this observer theoretically compares with the previous two ones, will be discussed in the next section.

2.4 Theoretical Comparison

Due to the fact that all the observers implemented in this work aimed at the same purpose, it is possible to make a preliminary comparison from the theoretical point of view.

The first, and most relevant, source of difference among these approaches is the fact that the *Luenberger observer* (LO) deals with *deterministic* systems, while the *Kalman Filter based* (KF) observer and the *Stephens'* one are meant to be used for *stochastic* systems. Therefore due to the deterministic nature of the Luenberger observer, it is expected not to completely fit the real, nor the simulated, environment in which it will be applied, that will certainly contain noise and unmodeled effects. The KF and Stephens' observers, since are based on the Kalman Filter technique, intrinsically contain white Gaussian noises, both for the process and for the measurement parts, meant to take into account such noises or unmodeled effects that could steer the results towards wrong estimates if not considered. Even though LO could be augmented using different techniques to make the observer gaining noise resilience, the vanilla versions of these observers clearly point to the KF ones as the most robust to the unpredictability of such dynamic systems.

As explained in Section 2.1, the Luenberger observer is asymptotically stable if the observed error converges to zero, that happens when the matrix $\mathbf{A} - \mathbf{GC}$ has all the eigenvalues with strictly negative real part. However, without violating this constraint, it is possible to shorten the transient time of the system by placing the poles of the matrix \mathbf{G} in such a way that the ones of $\mathbf{A} - \mathbf{GC}$ result reasonably far from the origin of the complex-real plane. This property is corroborated by the fact that the total response of the system is the sum of its zero-input response and its zero-state response. While the former is affected by the input of the system, the latter it is by the system eigenvalues, and if the system is asymptotically stable, this tends to zero: the far they are to the origin⁸, the sooner this part will tend to zero and therefore the shorter will be the transient. As the best pole placement of the gain matrix can be found empirically in the Luenberger observer to impact the convergence behaviour, it is possible to affect it in a similar way also for the Kalman Filter observers, by acting empirically on the choice of their covariance matrices. In this family of observers, if the system is linear and the noises are Gaussian, then the filter is optimal in mean squared error, but the behavior of the convergence can be changed with a proper setting of the covariances of the process and

⁸This doesn't mean that the poles can be placed arbitrarily far from the origin.

measurement noise processes, i.e. \mathbf{Q} and \mathbf{R} , respectively. Although these matrices are not completely arbitrary, because constrained by the nature of the system, the freedom allowed for tuning these values may also lead to the divergence of the estimating process or towards biased estimates.

All the three observers used in this work estimated at least the center of mass position and velocity of the robot and the external disturbance. Stephens' observer was the one that used the less quantity of data, in terms of both state and measurements vectors. As regards the LO and the KF, they had the same dimension for the state and measurements vectors but while the former explicitly used the inputs in the estimates computation, the latter exploited its covariances as a part of the process noise.

Finally, all the observers have theoretically shown the ability of estimating a 3D external force, i.e. an unknown external force whose components are non-zero along all the three dimensions, acting on the robot CoM, by removing the limitation of a disturbance forced to lie on the sagittal plane, as previously faced in [3]. However, the interesting difference relies on the fact that in the Luenberger and Stephens' observers each one of the three dimensions could be tackled independently, with their proper observers, while the Kalman Filter based one uses the estimates of the observer for the z -axis to compute the matrices necessary to perform the *update* step in the estimation of the x and y coordinates.

As regards the experimental comparison, more will be discussed in Chapter 4.

Chapter 3

Implementation Details

In this chapter we will describe what we have implemented in this work and how we've done this, by focusing on the structure of the testbed and on the collateral elements that we have used in order to build this resource.

The full code can be found at <https://github.com/EmilianPostolache/NaoObservers>.

3.1 Framework Background

Besides the implementation of the observers and their common wrapper, we had to face other decisions to construct the testbed. How to deal with the lack of a real robot to test the observers was the primary issue, followed by which controller to use to generate the gaits and finally how to retrieve the necessary data to feed and validate the observers. In the next paragraphs we'll explain how we tackled all these issues.

DART Environment

To overcome the lack of a real robot, in this work we've used the *Dynamic Animation and Robotics Toolkit* (DART) framework to perform simulations. This open-source library, through a 3D physical engine, allowed us to emulate the NAO robot and its behaviour as closest as possible to the real scenario. It was possible to have access to the robot parameters, such as its mass, as well as the dynamics of the articulated rigid body both when walking and standing still. It also allows the application of external forces of different nature and has a collision detection system. DART allowed us to retrieve all the data necessary to the observers: more on this will be explained later.

Intrinsically Stable MPC Controller

To generate stable gaits, necessary to gain data to validate the observers, we've relied on the *Intrinsically Stable MPC* (IS-MPC) schema as a controller, able to counteract the external disturbances that we've imposed to the simulated NAO. Presented in [2] for the perturbed case, the IS-MPC controller is based on the LIPM model in Eq. (2.1) and uses a causal stability constraint to guarantee the internal stability of the gait. It consists in solving for each iteration k , the *Quadratic Programming* (QP) problem in Eq. (3.1a) over the decision variables $\dot{\mathbf{X}}_z^k = (\dot{x}_z^k \dots \dot{x}_z^{k+C-1})^T$ and $\dot{\mathbf{Y}}_z^k = (\dot{y}_z^k \dots \dot{y}_z^{k+C-1})^T$, that collect the inputs of the system, i.e. the velocities of the ZMP, within the control horizon $C \cdot \delta$, regulated by the width of the

window, i.e. C , and by the duration of the sampling intervals δ .

$$\min_{\dot{\mathbf{X}}_z^k, \dot{\mathbf{Y}}_z^k} \quad \| \dot{\mathbf{X}}_z^k \|^2 + \| \dot{\mathbf{Y}}_z^k \|^2 \quad (3.1a)$$

$$\text{s.t.} \quad \bullet \text{ ZMP constraint,} \quad (3.1b)$$

$$\bullet \text{ stability constraint.} \quad (3.1c)$$

This optimization problem is subjected to two constraints: the first one is related to the ZMP values and it is used to guarantee the dynamic balance by means of a condition involving the rotation matrix \mathbf{R}_j^T associated to the orientation of the j -th footstep, its position (x_f^j, y_f^j) and the approximation of the dimensions of the foot sole, i.e. f_z and f_y as follows¹:

$$\mathbf{R}_j^T \begin{pmatrix} \delta \sum_{l=k}^{k+i-1} \dot{x}_z^l - x_f^j \\ \delta \sum_{l=k}^{k+i-1} \dot{y}_z^l - y_f^j \end{pmatrix} \leq \frac{1}{2} \begin{pmatrix} f_x \\ f_y \end{pmatrix} - \mathbf{R}_j^T \begin{pmatrix} x_z^k \\ y_z^k \end{pmatrix}. \quad (3.2)$$

The ZMP constraint alone is not enough to prevent the system to diverge, in fact since the LIP model can be subdivided into a *stable* x_s and an *unstable* x_u subsystems through a change of coordinates and this latter is responsible for the possible divergence of the model, a constraint must be enforced to control its behaviour. The *stability condition*, reported for the sagittal motion in Eq. (3.4), ties the value of the unstable component to the future values of the system input. From this condition it is possible to derive the stability constraint for the QP problem as follows:

$$\sum_{i=0}^{C-1} e^{-i\eta\delta} \dot{x}_z^{k+i} = - \sum_{i=C}^{\infty} e^{i\eta\delta} \dot{x}_z^{k+i} + \frac{\eta}{1 - e^{-\eta\delta}} (x_u^k - x_z^k). \quad (3.3)$$

It is interesting to note that this formulation uses both the inputs of the system within the control horizon, namely \dot{x}_z^k , and the conjectured ones, i.e. $\dot{\tilde{x}}_z^k$.

$$x_u^k = \eta \int_{t_k}^{\infty} e^{-\eta(\tau-t_k)} x_z(\tau) d\tau \quad (3.4)$$

In summary, putting Eq. (3.2) and (3.3) in the QP problem in Eq. (3.1a), leads, for every instant, to a sequence of optimal inputs, whose first sample is used. The resulting CoM trajectory is then tracked by a standard kinematic controller.

In this work the IS-MPC schema was implemented in the **Controller** class. It takes care of initializing the attributes used by the controller, such as the position of the CoM, its behavior and the duration of the phases of the support foot. In addition it generates the next points for the CoM trajectory using the `generateWalking` and `generateBalance` methods that in turn use the **MPCSolver** class to solve the QP problem in Eq. (3.1a). Finally the **Controller** class also initializes and updates the observers implemented in this work.

Measurements

In a real scenario, the NAO's sensors would provide useful information about the state of the robot. From the values of its joint encoders, via direct kinematics, it would be possible to compute the position of the center of mass and through the IMU, its acceleration. The four *Force Sensing Resistors* (FSRs) under each foot can provide data for computing the ground reaction force and the CoP, using additional geometry information. Recalling that the CoP and the ZMP coincide when the robot is in a stable configuration², also the ZMP can be obtained.

¹This constraint formulation refers to the single support phase.

²This is true in case of flat ground, as assumed through all this work.

In DART we've retrieved such quantities using `getCOM`, `getCOMLinearVelocity` and `getCOMLinearAcceleration` to get the data of the center of mass and `getZMPFromExternalForces` to obtain the CoP/ZMP position and the ground reaction force (even if the simulated NAO approximates the eight sensors with only two values, i.e. one for each foot). There were two other different methods available, besides the one that we've chosen, to compute the ZMP position using diverse kind of data, e.g. `getZMPFromWrench`, `getZMPFromAngularMomentum`: we've surveyed them all to understand which one was the most reliable and performing to use. These information were retrieved for each time-step and used as measurements to feed the observers with.

3.2 Observers Implementation

The observers were implemented using the *Composite* pattern, whose UML class diagram is reported in Fig. 3.1. The common structure of the three observers was defined in the `Observer` Component interface, which declared the state update method and the getters for the name, the state and the uncertainty. Apart from the name getter these methods are pure virtual (abstract) so they are implemented in the subclasses. The Composite and the Leaf interfaces were declared in the `CompositeObserver` and `LeafObserver` classes, respectively. The former is a container class from which particular observers can be updated, while the latter factorizes common properties and functions that all the observers share, such as the initialization methods and attributes for the state (i.e. `xAct`) and the axis.

Luenberger Observer Implementation

The constructor of the Luenberger observer takes as input the matrices \mathbf{A} , \mathbf{B} , \mathbf{C} and \mathbf{G} , whose meaning was defined in Section 2.1. It exposes two `init` functions: one allows to initialize the state estimate with a given initial guess and one that sets for it the default null vector. Additionally, it overrides the `update` method with the implementation of the state dynamics in Eq. (2.5) followed by its Euler integration to obtain the new estimate of the state. The `uncertainty` method returns the identity matrix since this observer doesn't model noise.

As stated in 2.1, the gain matrix \mathbf{G} can be found by pole placement. In order to find the poles configuration that best served the observer, a grid-search over different values of the poles was performed, eventually pointing the set of poles $[-70, -50, -80, -65, -45]$. The corresponding gain matrix, reported in Eq. (3.5), proved to be the one that produced the estimate with lowest peaks and steady-state error.

$$\mathbf{G} = \begin{pmatrix} 240 & 0 \\ 21254.7 & -29.5 \\ 0 & 70 \\ 820250 & 12.36 \\ 11700000 & 236.15 \end{pmatrix} \quad (3.5)$$

Kalman Filter Based Observer Implementation

In order to implement the Kalman Filter based observer we had to subclass the standard `CompositeObserver` with a `KalmanComposite` that manages the coupling between the z -axis observer and the xy -axes ones (all `KalmanFilter` objects). It overrides the `update` method where it first updates the z -axis observer, then creates the matrix from Eq. (2.15) based on its new state. This matrix is finally passed to the `update` method of the xy -axes observers. A `KalmanFilter` object is defined by the matrices \mathbf{A} , \mathbf{B} , \mathbf{C} , \mathbf{R} and it computes \mathbf{Q} in its constructor based on \mathbf{B} and on the covariances of the inputs as in Eq. (2.12). As the Luenberger observer,

it offers two `init` methods to initialize the estimate of the state and of the uncertainty matrix \mathbf{P} to a default or given values. It overrides `update` with the implementation of the Kalman Filter equations and returns the uncertainty matrix via the `uncertainty` method.

In order to increase the performances of this observer, similarly to what was done in the Luenberger case, we have empirically searched the values for the covariance values of the inputs and for the ones of the measurement noise \mathbf{R} that best fitted our system, eventually we set:

$$\mathbf{R}_z = \begin{pmatrix} 0.01 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{R}_x = \mathbf{R}_y = \begin{pmatrix} 0.01 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.01 \end{pmatrix}, \quad (3.6)$$

$$\sigma_{\dot{x}_c}^2 = \sigma_{\dot{y}_c}^2 = \sigma_{\dot{z}_c}^2 = 10^3, \quad \sigma_{\ddot{F}_x}^2 = \sigma_{\ddot{F}_y}^2 = \sigma_{\ddot{F}_z}^2 = 10^3. \quad (3.7)$$

Stephens' Observer Implementation

Similarly to what has been done for the other observers, the `StephensObserver` class takes as input the matrices \mathbf{A} , \mathbf{B} , \mathbf{C} and the noise covariance matrices \mathbf{Q} and \mathbf{R} . The `update` function works in a similar fashion to the one of the Kalman-based observer, since also this observer relies on the Kalman Filter equations, as described in Section 2.3, but using directly the inputs in the prediction stage and without any connection with the observers of the same type of the other dimensions. As the previous two observers, we offered two ways to initialize the objects of this class through the polymorphism of the function `init`. The covariance matrices for the process and measurement noises were found empirically, and took the form as follows:

$$\mathbf{Q} = \begin{pmatrix} 10^{-8} & 0 & 0 & 0 \\ 0 & 10^{-4} & 0 & 0 \\ 0 & 0 & 10^{-4} & 0 \\ 0 & 0 & 0 & 10^{-1} \end{pmatrix}, \quad \mathbf{R} = \begin{pmatrix} 10^{-8} & 0 \\ 0 & 10^{-4} \end{pmatrix}. \quad (3.8)$$

However, the values that we have decided to include in the search were driven by the idea of assuming lower uncertainty in correspondence to the values directly retrieved by DART, such as the position of the CoM, and give higher uncertainty to the values that were captured in a more difficult way. During this experimental tuning we have observed that increasing the force uncertainty value resulted in a faster convergence for the robot standing still, however stressing the oscillating behaviour in the walking setting.

3.3 Deployment

The observers have been instantiated in the `Controller` class following the object hierarchy in Fig. 3.2. The composite hierarchy is updated by calling `update` on the `observers` object at each step of the simulation along with all the necessary inputs, i.e. a vector containing the ZMP velocity obtained from the IS-MPC solver, and measurements. Each particular observer uses only the information it requires.

We have also extended the standard widget of the simulator in order to allow an external specification of all the parameters of the experiments in a more user-friendly way. We provided a checkbox in order to easily select which observer³ to include in the experiment, to avoid wasting time in computations that are not required.

In addition we have also provided another menu in the same widget to control with ease the external force \mathbf{F}_{ext} acting on the CoM of the robot. With these functionalities it is possible to

³More than one could be selected.

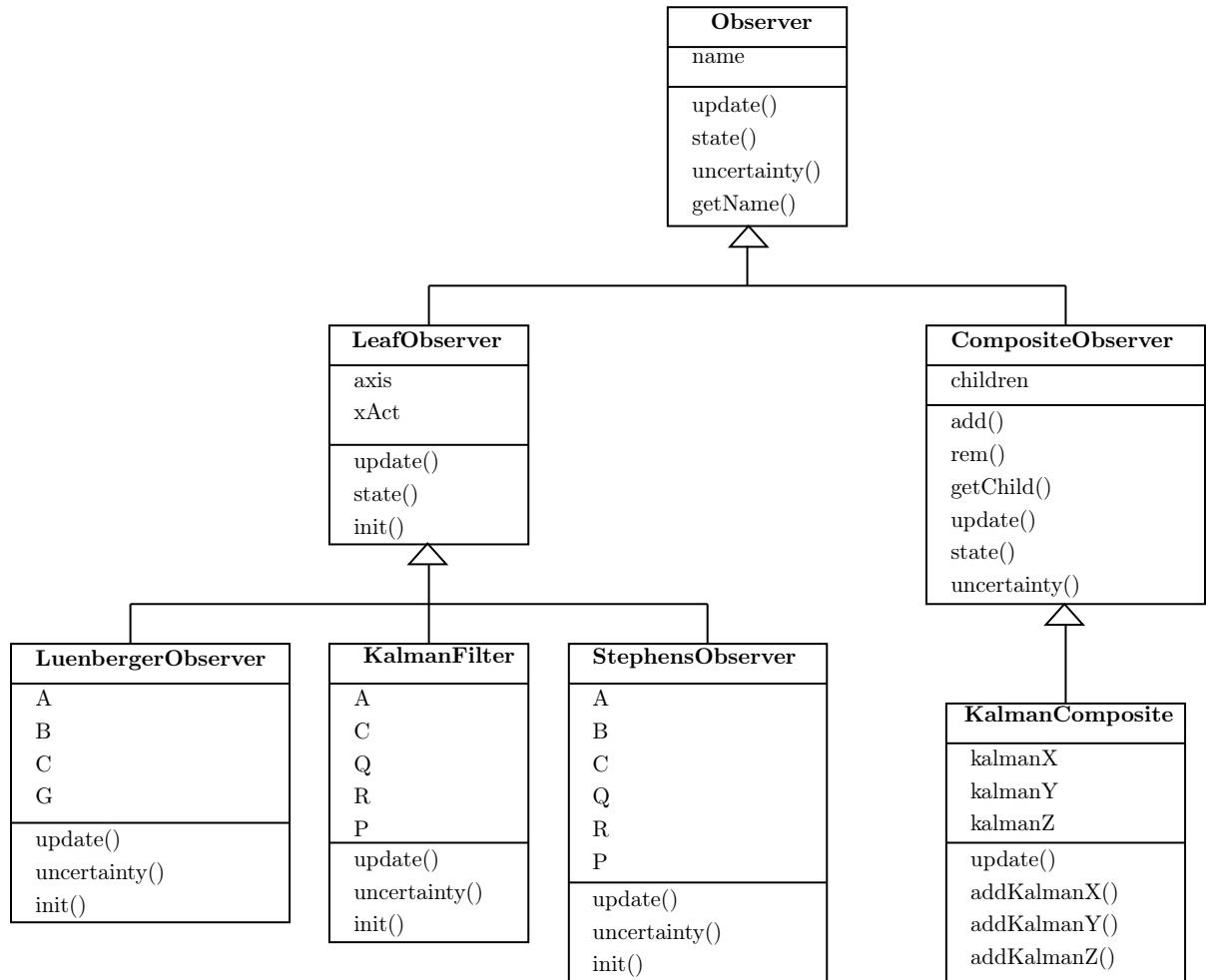


Figure 3.1. UML class diagram of the composite pattern that we have used to model the observers. The arrows represent inheritance. Some minor methods and attributes have been excluded from this diagram.

control the nature of the force, by choosing between a constant or a sinusoidal force. For the former it is also possible to select, through separate sliders, the magnitude of the components of the force along all the three dimensions. For the latter, instead, it is possible to specify, again using sliders, the amplitudes, the frequencies and the phases of the disturbances for each coordinate of the external disturbance. Finally it is possible to change the initial point in time of application of the force by indicating the corresponding frame.

In order to easily retrieve the estimates of the observers and therefore to capture their performances, we have logged them, along with the ground truth data, implementing the `storeData` function of `Controller`.

All these enhancements are indeed available for future researchers too.

In the next chapter we will show the results of the experiments that we've run to test these implementations.

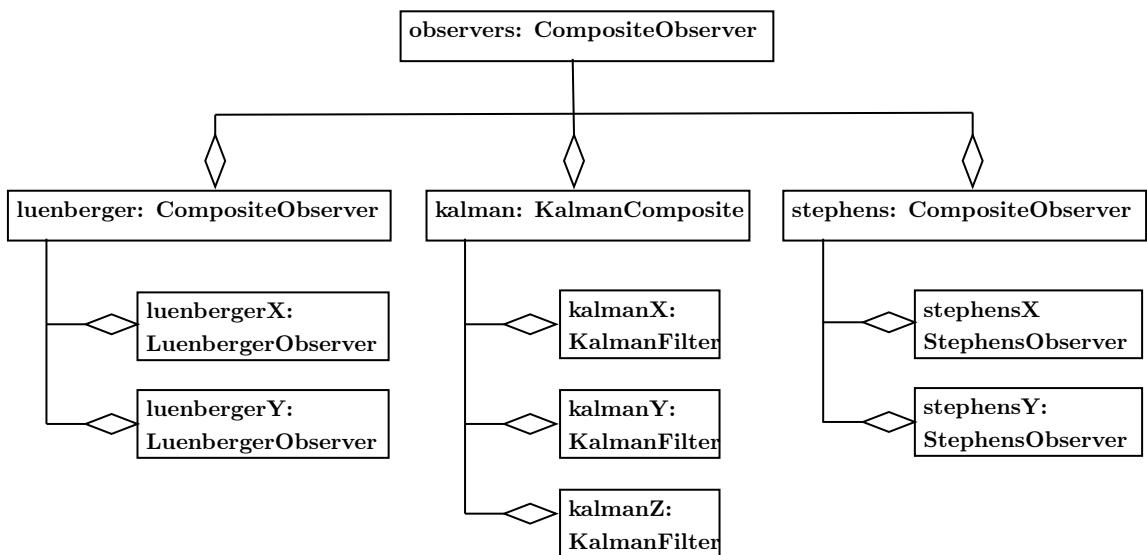


Figure 3.2. UML object diagram of the full `observers` composite object. Each box represents an object whose class (see Fig. 3.1) is given after the comma. The diamond arrow models the `has-a` relationships

Chapter 4

Experiments and Results

After having implemented the testbed as described in Chapter 3, we run several experiments to collect information about the performances of the observers and to validate them to offer a robust benchmark. We performed three main flows of experiments by changing the behaviour of the simulated NAO: the first one, named *Stand without Balance*, as the name suggests, is a setup in which the robot stands still without any control about the inputs that could balance it, by counteracting the presence of the external disturbance, i.e. without using the kinematic controller. The reason why we decided to use this behaviour is that we wanted to obtain the pure potential of the observers without any additional alterations. The second set of experiments, i.e. *Stand with Balance*, was performed with the robot standing still but using the kinematic controller: this was taken into account to verify the actual performances of the observers with a robot that is not walking. The last flow of experiments regard the most common application of a NAO robot: *Walking*. This was done to see how the observers handle a walking robot but also to analyse how walking with respect to standing still degrades the performances.

For all the three behaviours we run experiments with two kinds of external forces on the robot center of mass:

- *Constant force*: We applied a constant force starting from 2.5s from the beginning of the simulation, with different magnitudes for each dimension. In the next section we will report the results of the application of $\mathbf{F}_{ext} = (4.1, 4.7, 0)$ [N] in all the experiments for allowing comparisons;
- *Periodic force*: We applied a slow-variant sinusoidal force again from 2.5s by the beginning of the simulation. The external force used in the experiments has the form $\mathbf{F}_{ext} = (A_x \sin(2\pi f_x t + \phi_x), A_y \sin(2\pi f_y t + \phi_y), 0)$ [N] with $A_x = 4.1$ N, $A_y = 4.7$ N, $f_x = f_y = 0.1$ Hz, $\phi_x = 0$ rad and $\phi_y = \pi/2$ rad.

We've designed an observer for each dimension only for the Kalman Filter based one, while for the Luenberger and the Stephens' observers the axes taken into consideration where only x and y , but for all the experiments we've estimated the full state, not limiting to the external force only.

In order to provide also a quantitative measure of the performances of the observers, we have implemented a script that automatically computes determined metrics. The first indicator, usually used to evaluate the performance of an estimator, was the *Root Mean Square Error* (RMSE), that is the root of the mean of the square of the difference between the estimated samples and the corresponding real values of the parameter to estimate. Since this particular metric is very sensitive to outliers, and as we will discuss in Section 4.2 some of the quantities possessed them, we have also considered the *Mean Absolute Error* (MAE), i.e. the arithmetic average of the absolute error. Both these quantities were computed for all the observers for

all the performed experiments on the whole estimated and true signals. Due to the initial oscillating behaviour that characterizes the transient of some of the experiments, a unique total value for describing the overall performance was not enough to give a full description of the behaviour of the observers. For this reason, we have empirically¹ detected a point in the trend of the estimations that could ideally define the ending of the transient and the beginning of the steady-state. We have chosen this point to identify a more stable portion of the estimated signal into which computing the RMSE that in this way acted as an approximation of the steady-state error. The last metric that we have employed was the *Time To Convergence* (TTC): this complementary measure to the partial RMSE gives an approximation of the transient time, since it is the time elapsed from the beginning of the simulation to the detected point of convergence. For each experiment we will provide a table for a comparison among all the observers with their corresponding metrics, differentiating the total RMSE value and the partial one by denoting the former as RMSE (tot.).

4.1 Simulations Results

In this section all the results of the experiments will be analysed and commented.

4.1.1 Stand without Balance

The first sets of experiments was performed on the NAO robot standing still without balancing. We have applied both a constant and a sinusoidal force as described in the beginning of this chapter.

Constant Force

In Fig. 4.1 it is possible to see the results, for both axes, of the estimates made by the Luenberger observer following the application of a constant force, while in Fig. 4.2 and 4.3 it can be seen the ones of the Kalman Filter based observer for all the three axes and the behaviour of the total norm of the estimates that gives an overall indication of its performances. Finally, in Fig. 4.4 there are the x and y results of the Stephens' observer.

As it is possible to notice, for each observer the y component is estimated with more ease, i.e. with less presence of oscillations, with respect to its counterpart along the x axis. In addition, except for the KF, for the former component the trends seem getting closer to the real value than what happens for the latter.

In Table 4.1 the metrics corresponding to this experiments are reported. As expected the Luenberger observer is the most penalized in the overall RMSE due to its initial peaks (more on this can be found in Section 4.2), however it is also the quickest to converge. The KF and Stephens' observers produced quite close results, with an inclination for the former, however the TTC reveals that the best observer is also the slowest.

¹We have waited at least 2 seconds from the beginning of the simulation.

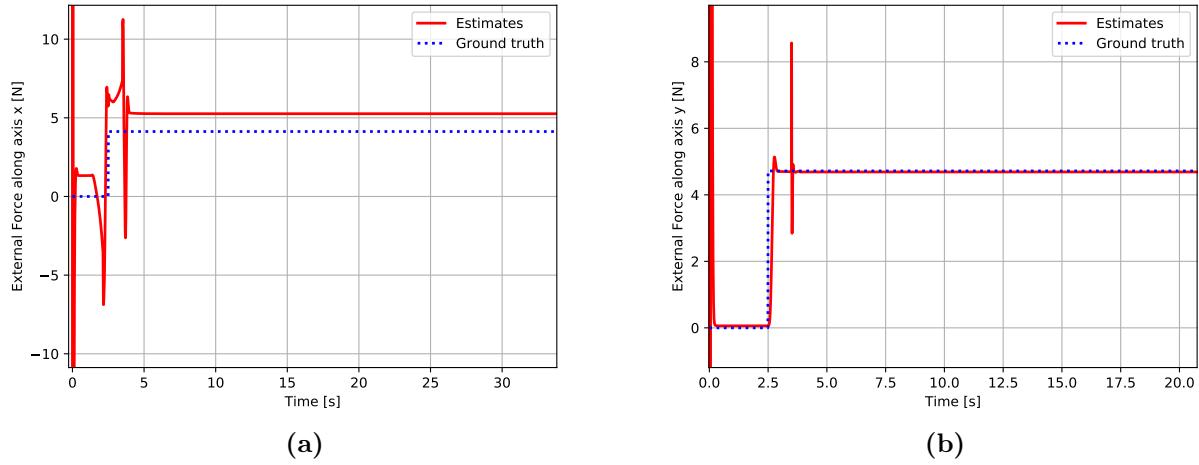


Figure 4.1. Observers behaviour after the application of the constant force to the NAO standing still without balancing: (a) Luenberger observer x axis; (b) Luenberger observer y axis.

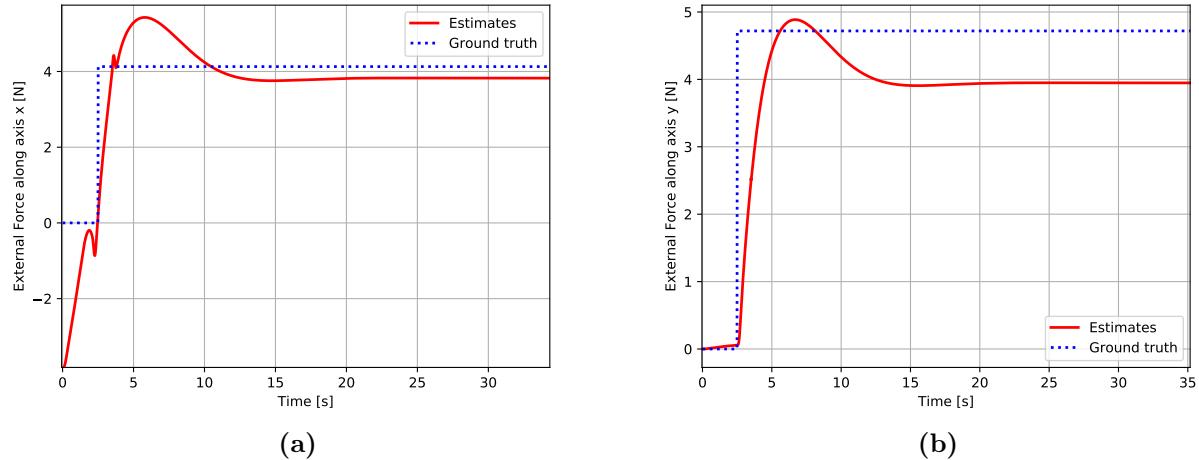


Figure 4.2. Observers behaviour after the application of the constant force to the NAO standing still without balancing: (a) Kalman-based observer x axis; (b) Kalman-based observer y axis.

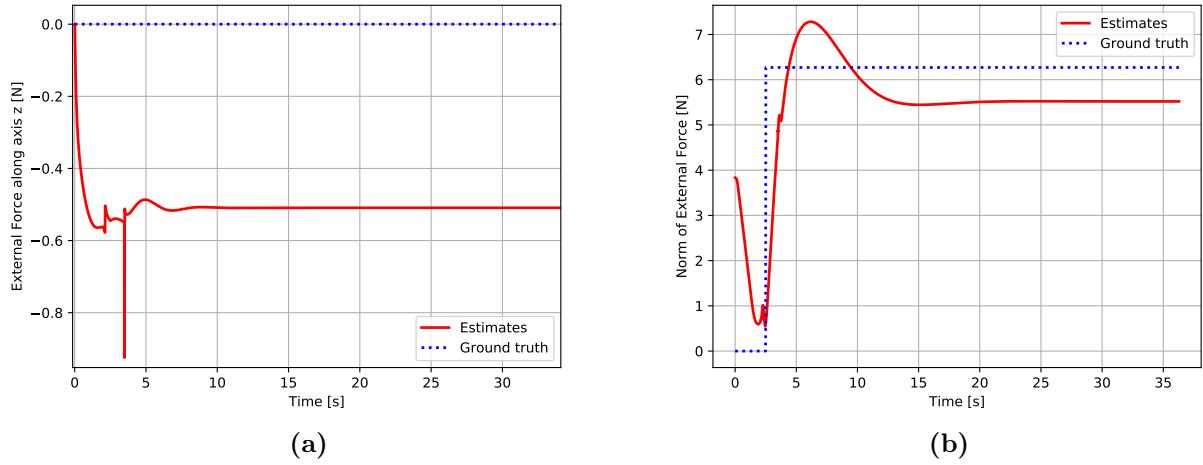


Figure 4.3. Observers behaviour after the application of the constant force to the NAO standing still without balancing: (a) Kalman-based observer z axis; (b) Kalman-based observer norm of \mathbf{F}_{ext} .

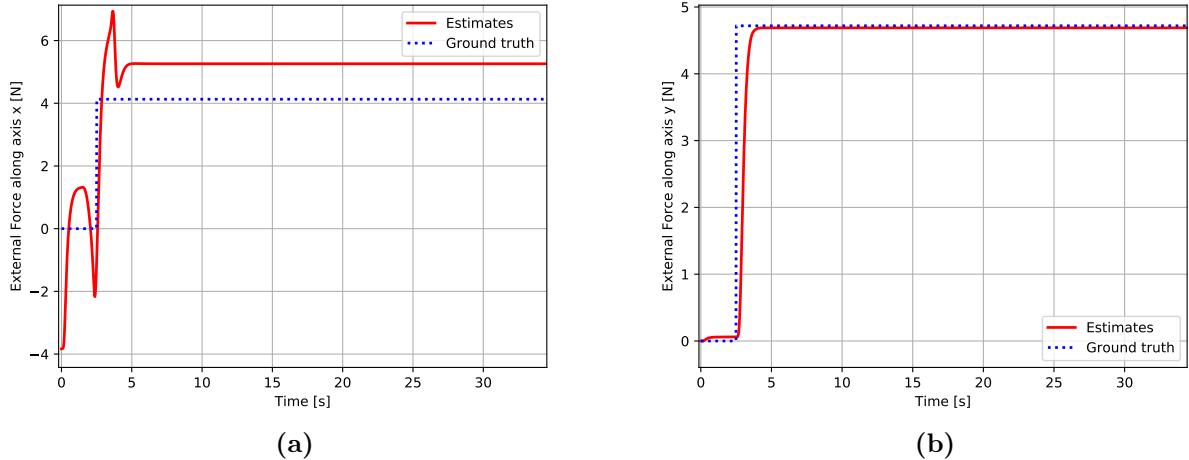


Figure 4.4. Observers behaviour after the application of the constant force to the NAO standing still without balancing: (a) Stephens' observer x axis; (b) Stephens' observer y axis.

Observer	Metric			
	RMSE (tot.) [N]	MAE (tot.) [N]	RMSE [N]	TTC [s]
Luenberger	50.25	2.89	1.13	4.5
KF based	1.21	0.99	0.83	23.0
Stephens	1.32	1.21	1.13	5.0

Table 4.1. Error metrics for the standing without balancing setting with a constant force acting on the robot.

Periodic Force

As the constant force case, the observers estimated more correctly the y axis, especially the Luenberger, that seems perfectly reconstructing the original signal as it is possible to see in Fig. 4.5, and the Stephens' one in Fig. 4.8. The Kalman Filter based observer in Fig. 4.6 and 4.7, instead, is the one that suffers most of the delay in the estimates while, however, replicating the shape of the sinusoidal force with less oscillations. As in the constant case, it underestimates the z component of the force by a small constant error. It is worth noticing that, with the exception of offsets in the estimates, the Luenberger and the Stephens' observers replicate the sinusoid with almost the same amplitude of the original signal, while the KF tends to scale it down. An explanation of this phenomenon could be related to the above-mentioned delay in reconstructing the real shape. A quantitative comparison is reported in Table 4.2, where the Stephens shows the best overall results, while the Luenberger reveals to be the most accurate at steady-state and the quickest to converge.

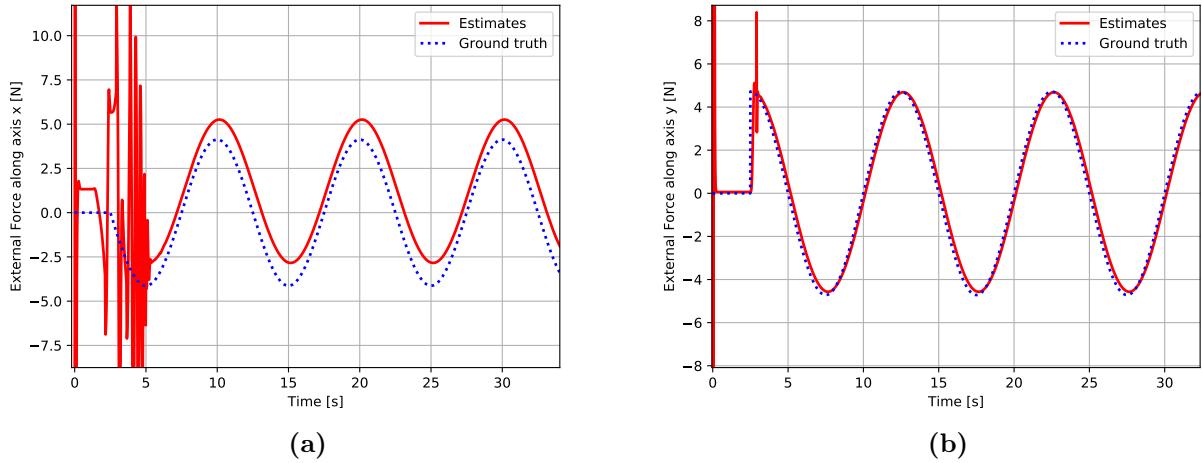
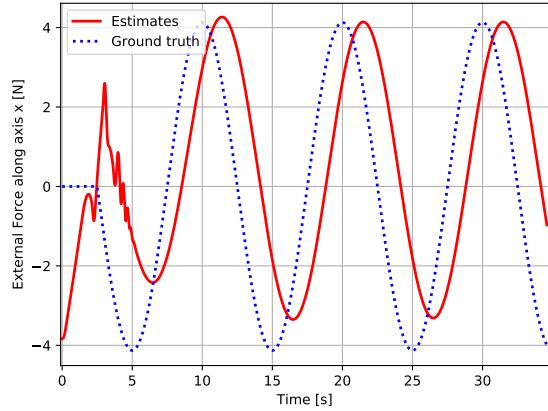
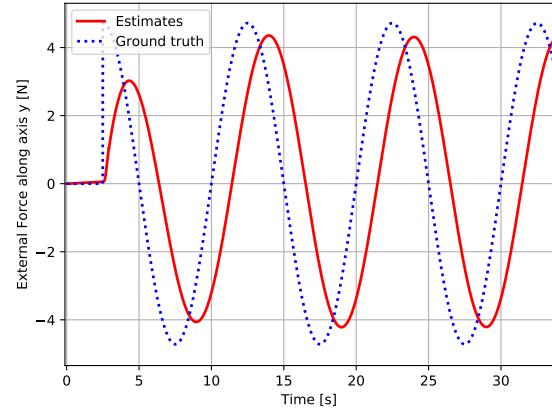


Figure 4.5. Observers behaviour after the application of the periodic force to the NAO standing still without balancing: (a) Luenberger observer x axis; (b) Luenberger observer y axis.

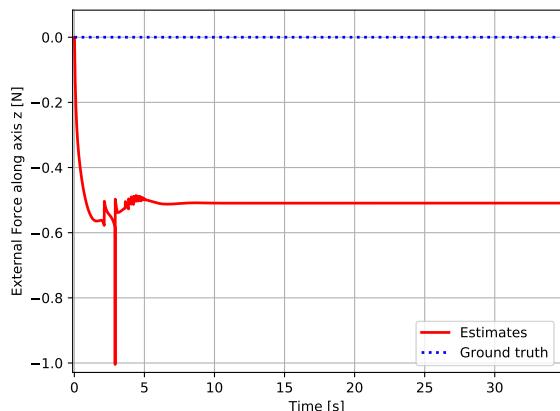


(a)

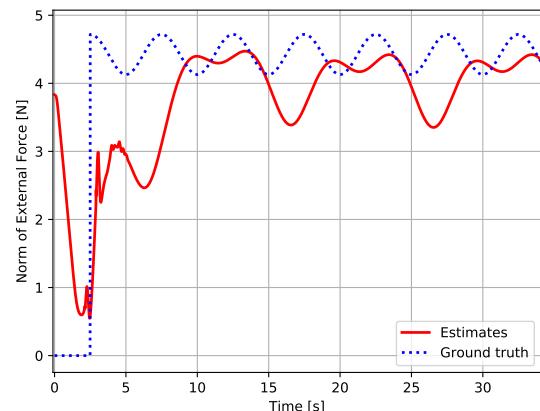


(b)

Figure 4.6. Observers behaviour after the application of the periodic force to the NAO standing still without balancing: (a) Kalman-based observer x axis; (b) Kalman-based observer y axis.



(a)



(b)

Figure 4.7. Observers behaviour after the application of the periodic force to the NAO standing still without balancing: (a) Kalman-based observer z axis; (b) Kalman-based observer norm of \mathbf{F}_{ext} .

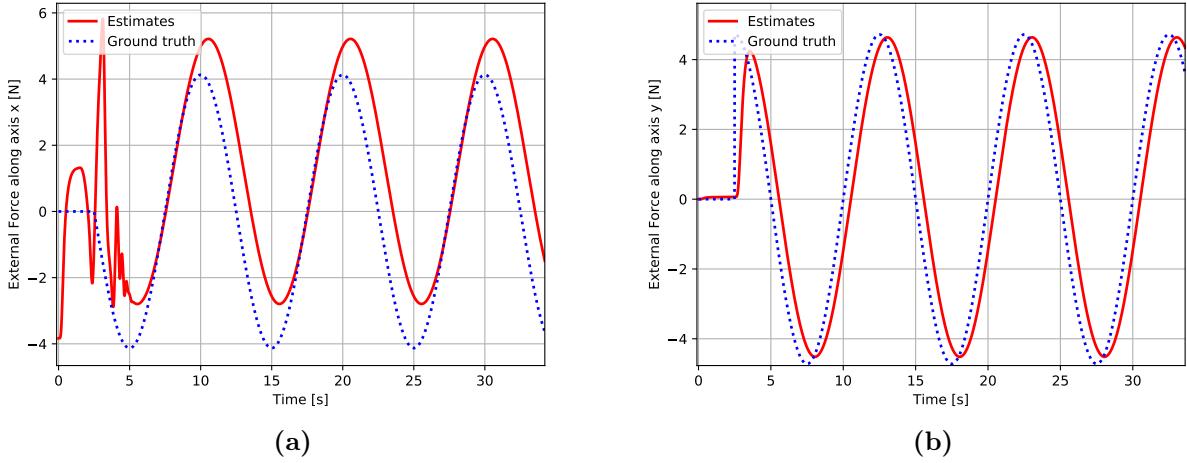


Figure 4.8. Observers behaviour after the application of the periodic force to the NAO standing still without balancing: (a) Stephens' observer x axis; (b) Stephens' observer y axis.

Observer	Metric			
	RMSE (tot.) [N]	MAE (tot.) [N]	RMSE [N]	TTC [s]
Luenberger	51.50	3.35	1.28	5.5
KF based	3.68	3.59	3.75	5.5
Stephens	2.09	1.83	1.91	5.6

Table 4.2. Error metrics for the standing without balancing setting with a periodic force acting on the robot.

4.1.2 Stand with Balance

The second behaviour that we have tested was the Stand with Balance. As the previous setting, the experiments were performed with both a constant and a sinusoidal force.

Constant Force

In case of the application of a constant force on the NAO standing still, the x component is now the one that seems to be estimated better than its y counterpart, while remaining the one most affected by peaks and oscillations, as can be seen in Fig. 4.9 for the Luenberger, Fig. 4.10 and 4.11 for the KF and 4.12 for the Stephens. The Stephens' observer appears to be the one that provided the best estimates for all dimensions, as supported by the metrics in Table 4.3, but it is not as quick as the Luenberger. These results corroborates the characteristics shown by the observers in the previous experiment. More specifically, the Luenberger observer proved to be the fastest to converge but again shows peaks and oscillations that raise its total RMSE. The Kalman filter based one has again very long transient time, while the curve lacks of extreme peaks: this is confirmed by the average and total RMSE values shown in Table 4.3.

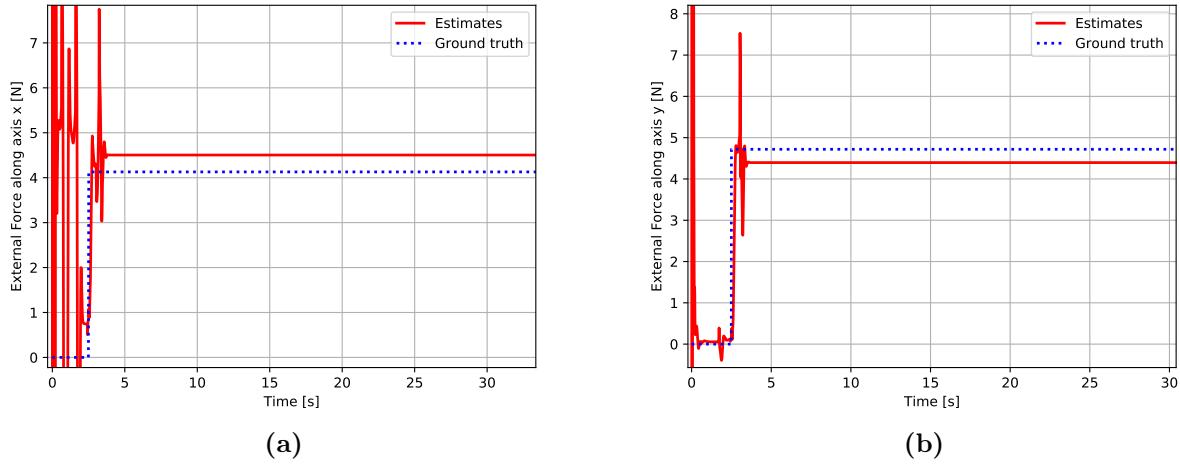


Figure 4.9. Observers behaviour after the application of the constant force to the NAO standing still with balancing: (a) Luenberger observer x axis; (b) Luenberger observer y axis.

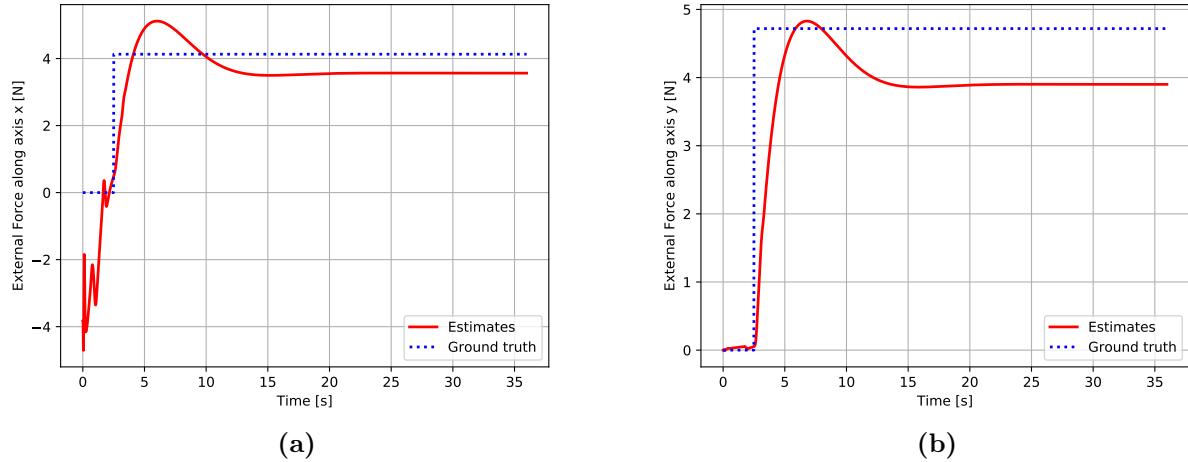


Figure 4.10. Observers behaviour after the application of the constant force to the NAO standing still with balancing: (a) Kalman-based observer x axis; (b) Kalman-based observer y axis.

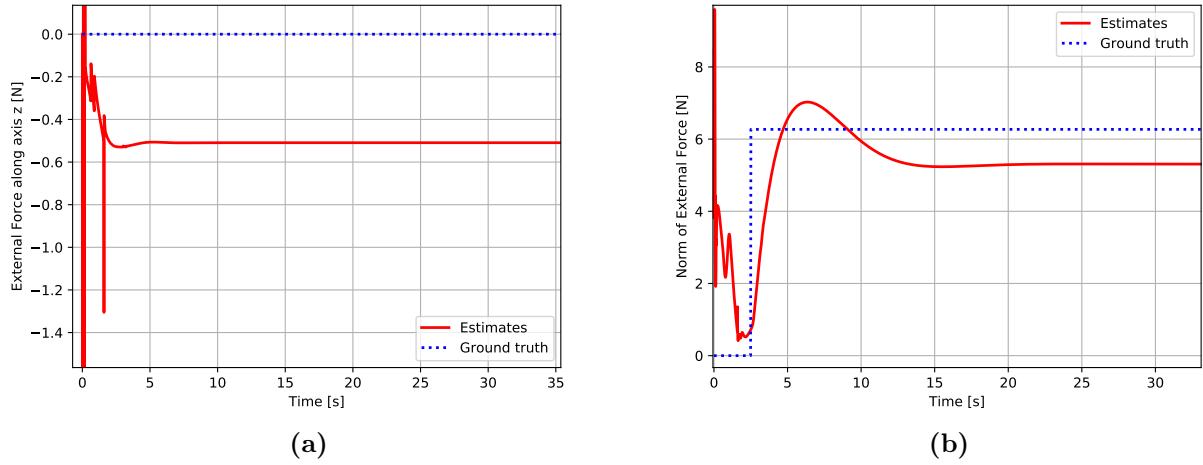


Figure 4.11. Observers behaviour after the application of the constant force to the NAO standing still with balancing: (a) Kalman-based observer z axis; (b) Kalman-based observer norm of \mathbf{F}_{ext} .

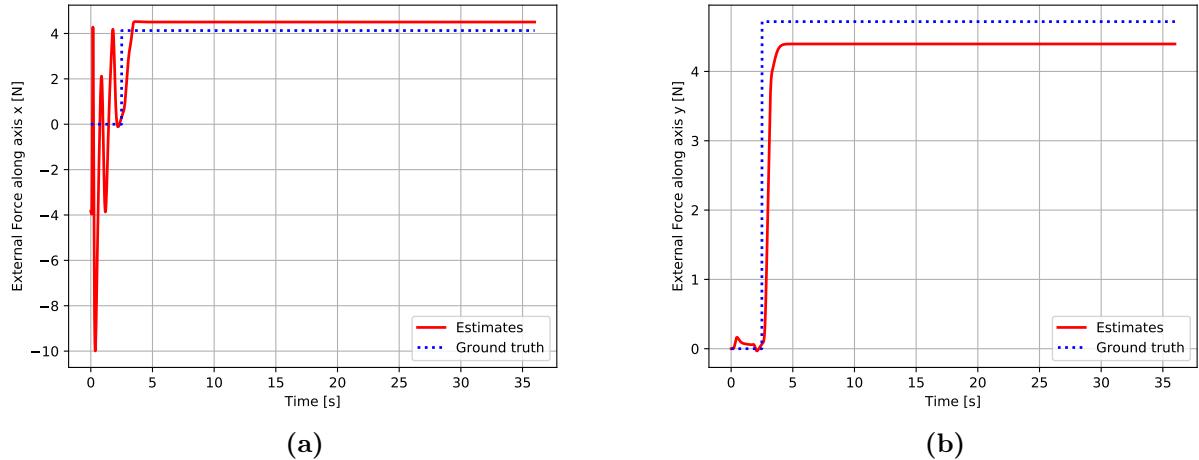


Figure 4.12. Observers behaviour after the application of the constant force to the NAO standing still with balancing: (a) Stephens' observer x axis; (b) Stephens' observer y axis.

Observer	Metric			
	RMSE (tot.) [N]	MAE (tot.) [N]	RMSE [N]	TTC [s]
Luenberger	50.82	2.85	0.50	4.00
KF based	1.33	1.10	0.99	24.0
Stephens	1.24	0.72	0.50	4.6

Table 4.3. Error metrics for the standing with balancing setting with a constant force acting on the robot.

Periodic Force

Applying a sinusoidal force to the NAO CoM when standing still and balancing itself, results in similar behaviors between the x and y components of the external force: with the exception for the presence of peaks and oscillations that, as seen in the previous experiments, affect the former coordinate. The Luenberger and the Stephens' observers in Fig. 4.13 and 4.16 respectively, seem the ones that provided better estimates than the KF, that is affected by a greater delay than its adversaries, that also reflects in the RMSE values, as can be seen in Table 4.4. The KF, whose results are reported in Fig. 4.14 and 4.15, shows a similar behaviour to the one of the periodic force in the standing without balancing setting, i.e. the attenuation of the amplitude of the original signal and an additional slight degradation of the estimate of the z component. As it is possible to note from the corresponding figures and table, the Luenberger is the best at estimating both quantities when excluding the transient.

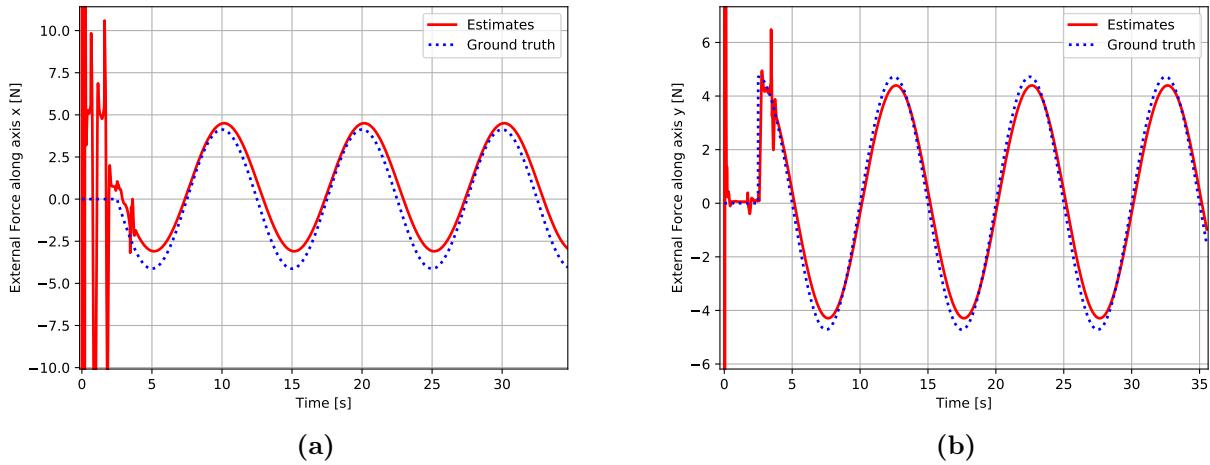


Figure 4.13. Observers behaviour after the application of the periodic force to the NAO standing still with balancing: (a) Luenberger observer x axis; (b) Luenberger observer y axis.

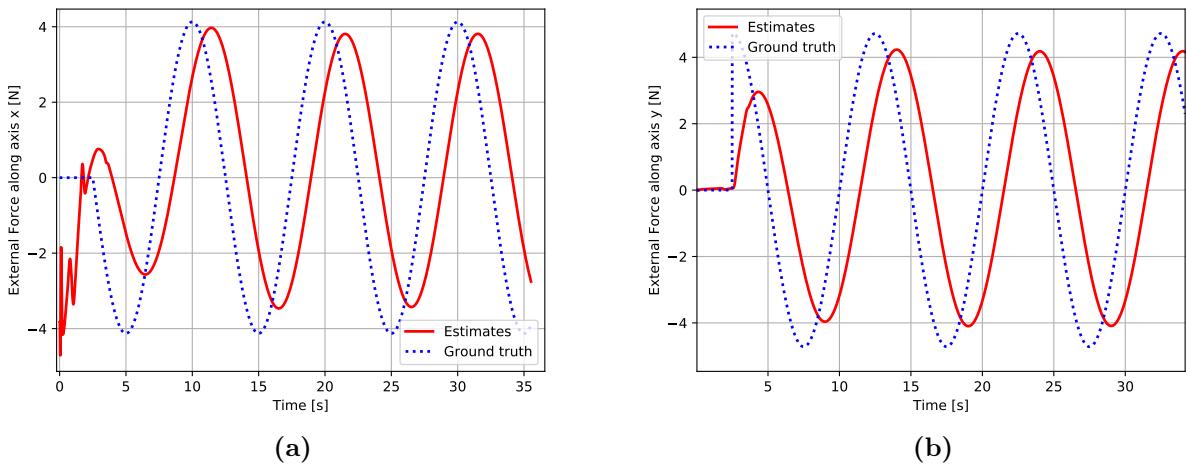


Figure 4.14. Observers behaviour after the application of the periodic force to the NAO standing still with balancing: (a) Kalman-based observer x axis; (b) Kalman-based observer y axis.

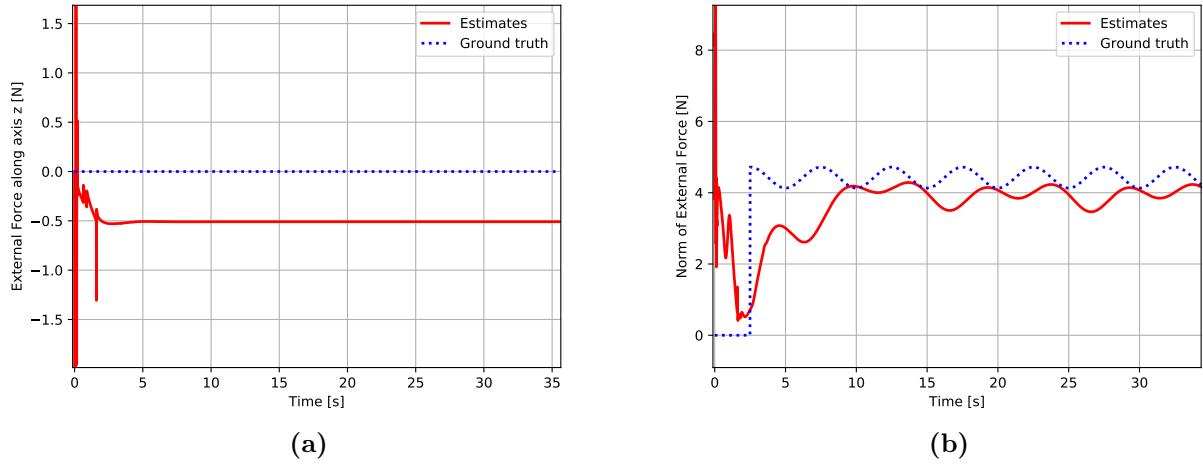


Figure 4.15. Observers behaviour after the application of the periodic force to the NAO standing still with balancing: (a) Kalman-based observer z axis; (b) Kalman-based observer norm of \mathbf{F}_{ext} .

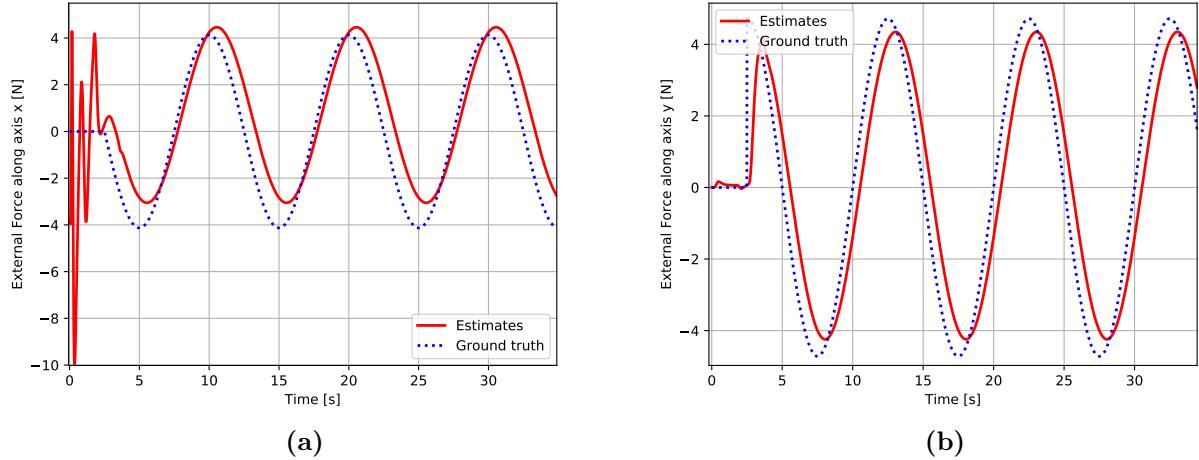


Figure 4.16. Observers behaviour after the application of the periodic force to the NAO standing still with balancing: (a) Stephens' observer x axis; (b) Stephens' observer y axis.

Observer	Metric			
	RMSE (tot.) [N]	MAE (tot.) [N]	RMSE [N]	TTC [s]
Luenberger	51.15	3.20	0.90	4.0
KF based	3.69	3.62	3.78	5.0
Stephens	1.94	1.72	1.69	4.0

Table 4.4. Error metrics for the standing with balancing setting with a periodic force acting on the robot.

4.1.3 Walking

The last set of experiments regards the most challenging behaviour tackled in this work: walking. As usual we have applied the constant and the sinusoidal force.

Constant Force

The walking behaviour represents a challenge due to the introduction of noise and unmodeled effects that partially reflects onto the measurements used to feed the observers. In addition the walking gait genuinely modify quantities that in the previous settings were constant or slowly-varying, by making them more exciting.

The resulting estimation appears severely degraded in the Luenberger case, as can be seen in Fig. 4.17 whose peaking phenomenon reaches its maximum affliction. Also the Kalman Filter based, Fig. 4.18 and 4.19 and the Stephens' observers, Fig. 4.20, that usually resist more to the oscillations, start showing this perturbation.

The presence of such large discrepancies in the case of the Luenberger observer can be explained by its deterministic structure. In fact, the other two observers use white Gaussian noises, which are used to take into account the above mentioned noises or unmodeled effects that could steer the results towards wrong estimates if not considered.

However, although the results are disturbed, it is clear that the observers have detected the shape of the original force, therefore to have a major feeling of the trend of the observers we have designed a low-pass filter, whose details are reported in Section 4.2, devoted to smooth the estimated behaviour as to reveal the underlying intention. This filtered version is reported in the figures through a solid black line. Once having filtered such results the behaviours do not differ as strongly, from the ones of the previous setups, as it seemed with the non-filtered version. By looking at the Table 4.5 we can however point to the Stephens' observer as the one with the best results and to the Luenberger as the quickest.

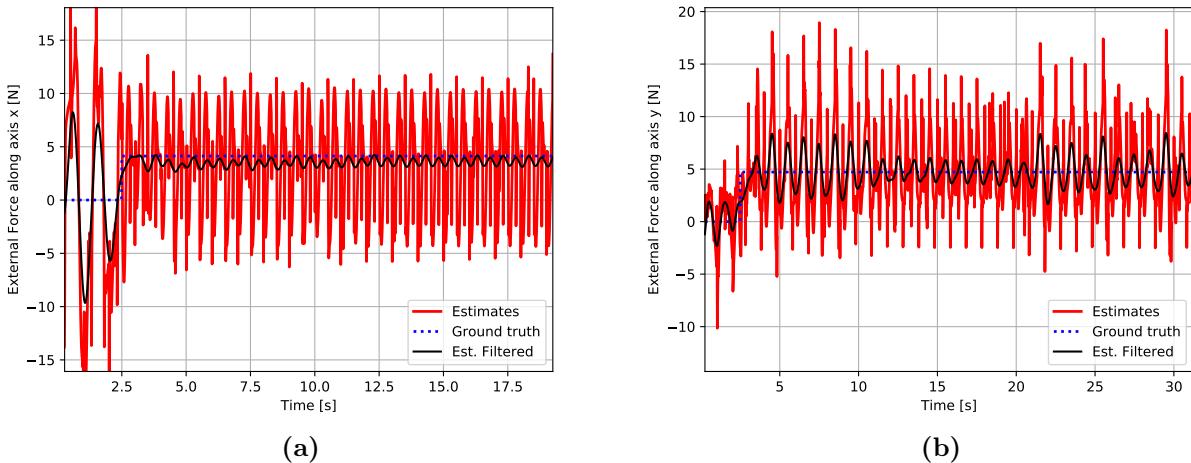


Figure 4.17. Observers behaviour after the application of the constant force to the NAO walking: (a) Luenberger observer x axis; (b) Luenberger observer y axis.

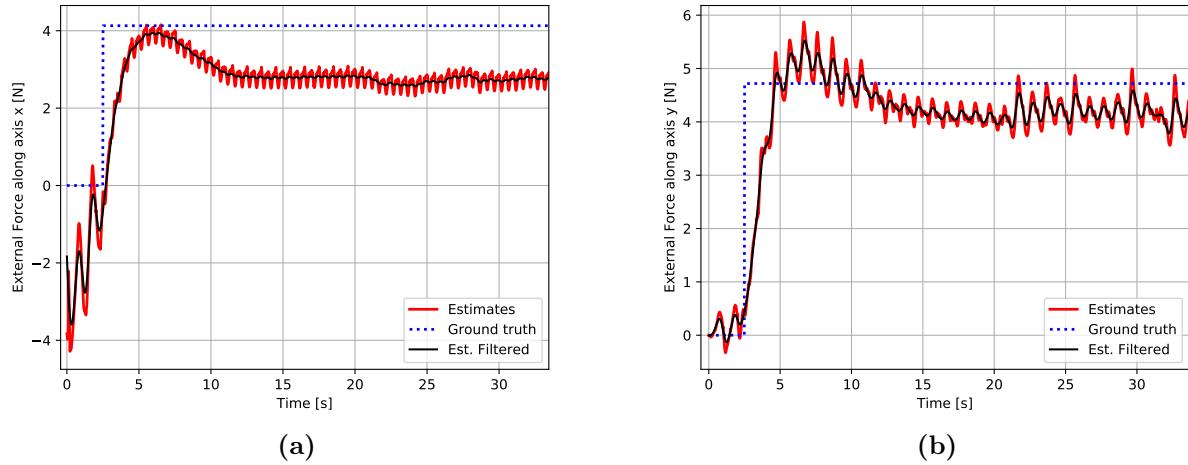


Figure 4.18. Observers behaviour after the application of the constant force to the NAO walking: (a) Kalman-based observer x axis; (b) Kalman-based observer y axis.

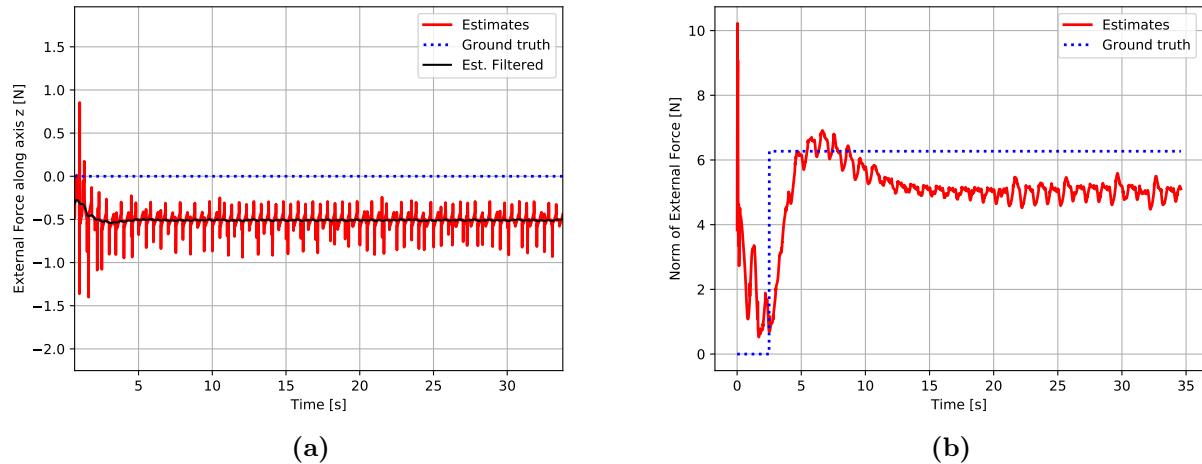


Figure 4.19. Observers behaviour after the application of the constant force to the NAO walking: (a) Kalman-based observer z axis; (b) Kalman-based observer norm of \mathbf{F}_{ext} .

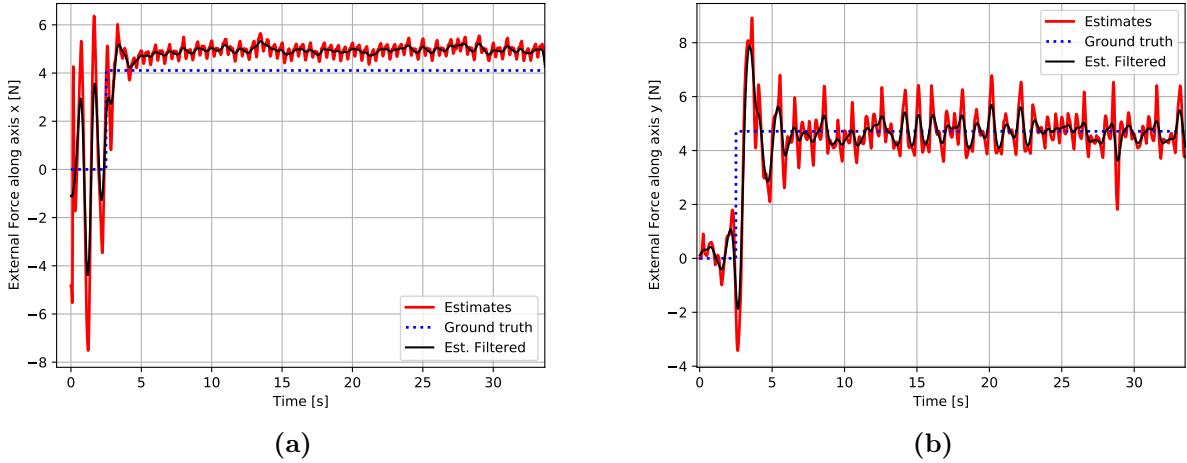


Figure 4.20. Observers behaviour after the application of the constant force to the NAO walking: (a) Stephens' observer x axis; (b) Stephens' observer y axis.

Observer	Metric			
	RMSE (tot.) [N]	MAE (tot.) [N]	RMSE [N]	TTC [s]
Luenberger	52.10	7.33	5.85	4.5
KF based	1.68	1.49	1.54	20.0
Stephens	1.76	1.35	1.09	6.0

Table 4.5. Error metrics for the walking setting with a constant force acting on the robot.

Periodic Force

The application of a periodic force to the walking NAO was the hardest experiment performed in this work, due to the combination of the disturbances introduced by the walking gait and the varying of the external force. We noticed that the Luenberger observer in Fig. 4.21, follows the oscillation of the force with no delay, as suggested by the low pass filter, but induces in frequent tall spikes that degrade the performance. The Kalman Filter based observer in Fig. 4.22 and 4.23, provides results that are fairly regularized, evolving relatively smoothly even when not filtered. The biggest issue affecting this observer is the high delay, almost 1.5s, with which it follows the ground truth signal. The Stephens' observer, in Fig. 4.24, embodies a good trade-off by combining the quality of the previous observer, without introducing unpleasant delay or noisy behaviour (although some spikes can be noticed at peaks and valleys of the plot in the y axis, but they can be filtered with success). The corresponding metrics can be found in Table 4.6.

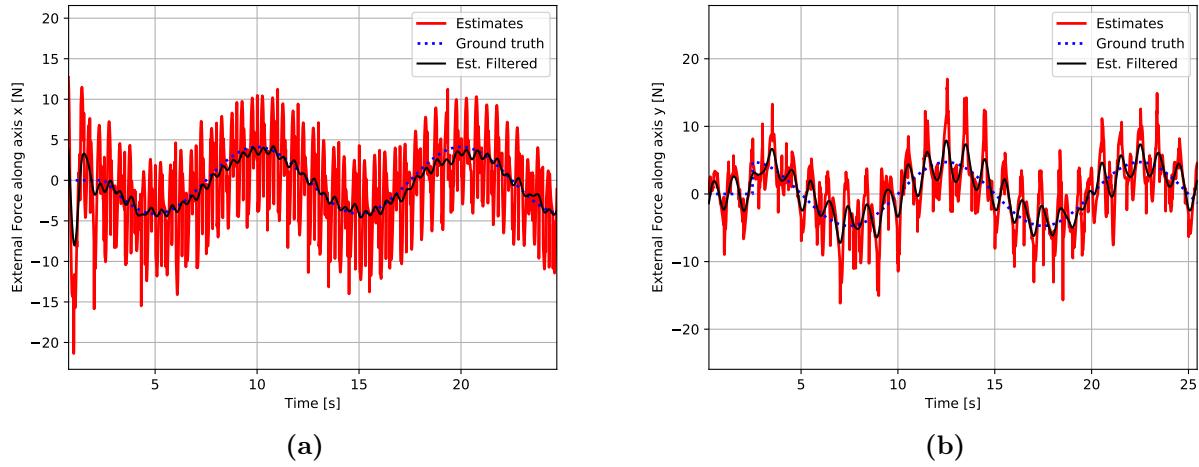


Figure 4.21. Observers behaviour after the application of the periodic force to the NAO walking: (a) Luenberger observer x axis; (b) Luenberger observer y axis.

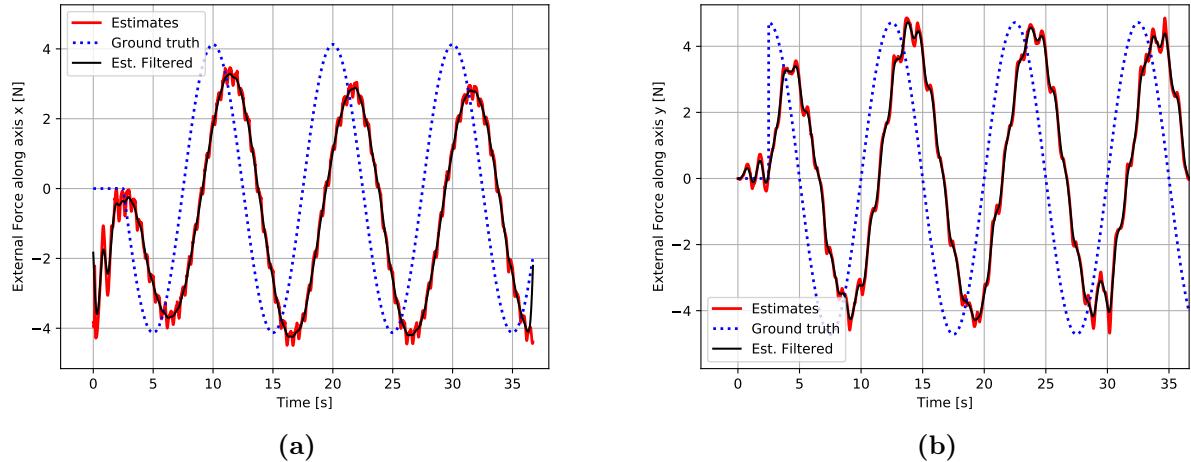


Figure 4.22. Observers behaviour after the application of the periodic force to the NAO walking: (a) Kalman-based observer x axis; (b) Kalman-based observer y axis.

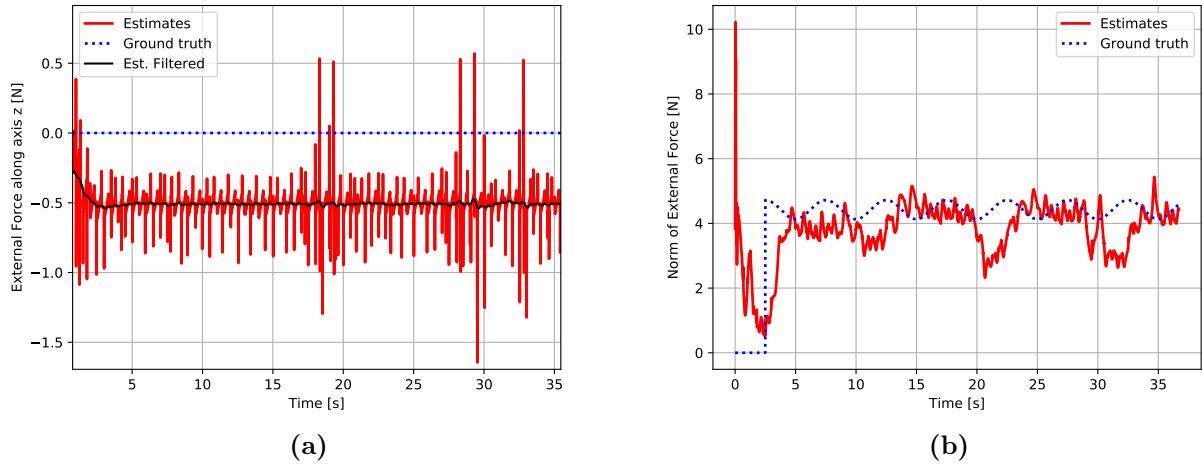


Figure 4.23. Observers behaviour after the application of the periodic force to the NAO walking: (a) Kalman-based observer z axis; (b) Kalman-based observer norm of \mathbf{F}_{ext} .

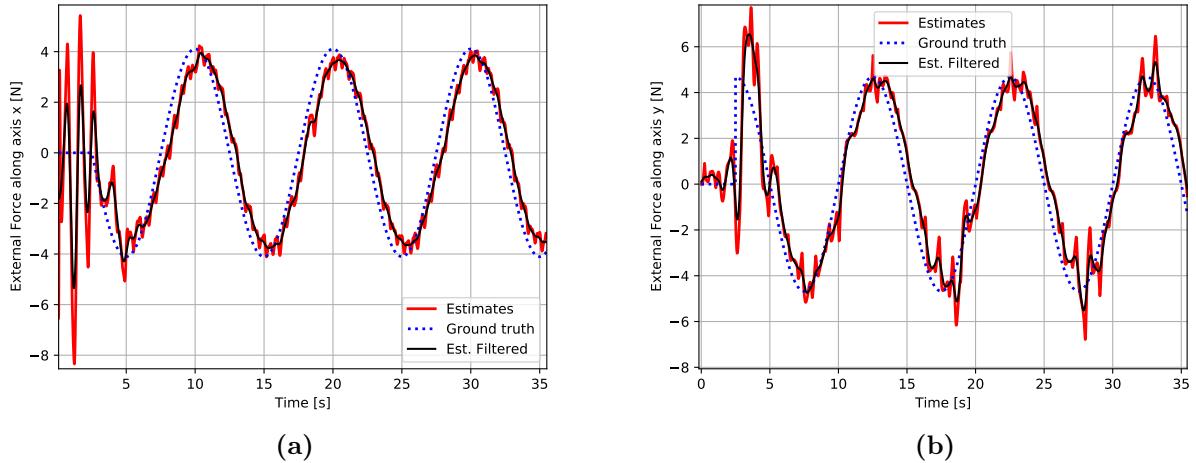


Figure 4.24. Observers behaviour after the application of the periodic force to the NAO walking: (a) Stephens' observer x axis; (b) Stephens' observer y axis.

Observer	Metric			
	RMSE (tot.) [N]	MAE (tot.) [N]	RMSE [N]	TTC [s]
Luenberger	50.60	7.44	6.09	2.5
KF based	3.84	3.74	3.99	4.5
Stephens	1.85	1.50	1.31	6.0

Table 4.6. Error metrics for the walking setting with a periodic force acting on the robot.

4.1.4 Experimental Comparison

After having analysed qualitatively all the results that our experiments provided, we have computed also a common quantitative indicator of the performances of the observers starting

from the values of the metrics gathered from the simulations. We have averaged in Table 4.7 all the corresponding values to have an overview of the potentialities of the implementations in our work. This distribution confirmed what was observed during the survey of the results, i.e. that the Stephens' observer was the most performing state-disturbance observer. The strengths of this observer come from the fact that it was the one that provided the lowest errors, therefore highest performances, according to the employed metrics, but also that was one of the fastest to converge in every experiment. Even if the Kalman Filter based observer shows total results very close to those of the Stephens' observer, it cannot withstand the competition with it either under the time to convergence measure or in term of delay of replication of the waveform. Another point in which the Stephen's is better than the KF, despite both rely on the Kalman Filter equations, is that its implementation is easier than the KF based one and does not make use of the information produced by one of its sub-observers acting on one coordinate to compute the estimates for the others components, thus making possible to speed up the computations required, e.g. by using a multi-thread solution. Finally, the Luenberger observer has shown however to be the quickest to converge and quite often the one with the lowest steady-state error, however the peaking and oscillating behaviour that occurs in almost all the simulations, not only degrades the metrics results but also makes it unreliable to be used in certain settings, e.g. when walking, that is the most important one in practice.

Observer	Metric			
	RMSE (tot.) [N]	MAE (tot.) [N]	RMSE [N]	TTC [s]
Luenberger	51.07	4.51	2.63	4.17
KF based	2.57	2.42	2.48	13.67
Stephens	1.7	1.39	1.27	5.2

Table 4.7. Error metrics computed by averaging the results of each experiment.

To further validate this claim, we have performed a final simulation in which we have added an additional perturbation to some measurements, i.e. CoM position, CoM acceleration and CoP position along both x and y axes, to show the robustness of the Stephens' observer in estimating the state of the system using even noisier data. We have added a Gaussian noise process of zero mean and standard deviation of 0.002 to these quantities. In Fig. 4.25 are reported the results of the estimation of the external force made in the situation of a constant force applied to the walking NAO, while in Fig. 4.26 are shown also the estimates of the other quantities in the system state. As it is possible to see in this latter figure, the estimation of the state quantities does not excessively suffers from the addition of the noise, especially the CoP position due to the fact that was filtered, thus reducing the impact of also this extra disturbance. Although, the CoM velocities seem the most degraded ones. Looking at the estimates of the external force, as corroborated from the metrics values in Table 4.8, it is possible to see that its performance is still valid and comparable with the results in absence of these additional perturbations described in Section 4.1.3.

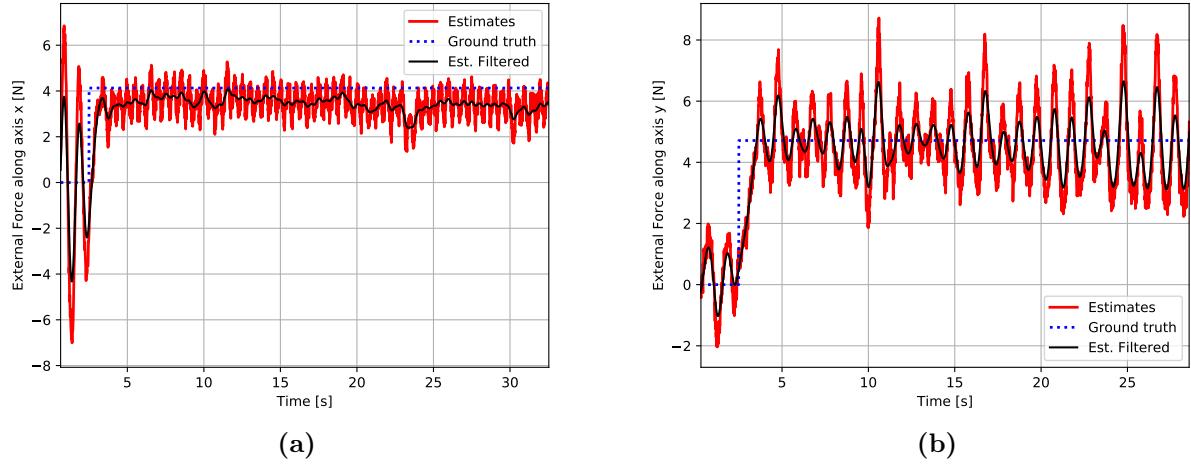


Figure 4.25. Results of the estimation of the external force for the application of a constant force while the NAO is walking, produced by the Stephens' observer with additionally degraded measurements: (a) External force along x axis; (b) External force along x axis.

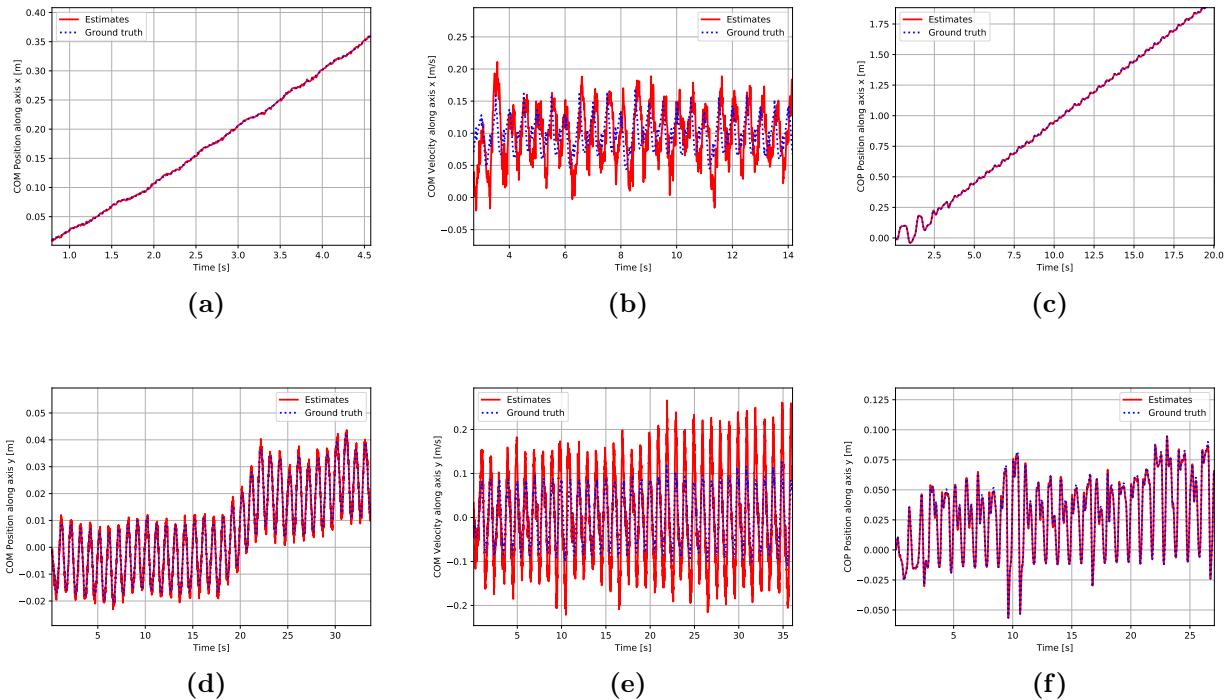


Figure 4.26. Results of the estimation of the full state for the application of a constant force while the NAO is walking, produced by the Stephens' observer with additionally degraded measurements: (a) CoM position along x axis; (b) CoM velocity along x axis; (c) CoP position along x axis; (d) CoM position along y axis; (e) CoM velocity along y axis; (f) CoP position along y axis.

Metric				
Observer	RMSE (tot.) [N]	MAE (tot.) [N]	RMSE [N]	TTC [s]
Stephens	2.03	1.66	1.63	4.5

Table 4.8. Error metrics for the walking setting with a constant force, computed by the Stephens' observers with perturbed measurements (CoM position, CoM acceleration and CoP position affected by white Gaussian noise of zero mean and st.dev. of 0.002).

4.2 Experimental Considerations

During the experimental phase we've encountered a series of interesting behaviours and insights that, once tackled, allowed us to improve the testbed, both from the structural point of view and of the performances of the observers. Next, we will report the most relevant.

Once having generated the first set of results, we surveyed the behaviour of the quantities used as measurements to understand whether the poor estimates in certain zones were due to wrong or too noisy references. We've discovered that DART reported periodically a set of zero-valued measurements for the position of the ZMP, due to a bug in the simulator that prevented the foot sole of the robot to contact correctly with the ground, therefore the ground reaction forces computed at the feet were cancelled out resulting in a wrong zero ZMP position measure. To avoid this, we've automatically isolated and eliminated these outliers while guaranteeing a continuity in the shape of the quantity. However, in order to further reduce the noise in the ZMP position reference, we've also designed a low pass filter able to smooth the x and y components of the ZMP as shown in Fig. 4.27(a) and (b) respectively.

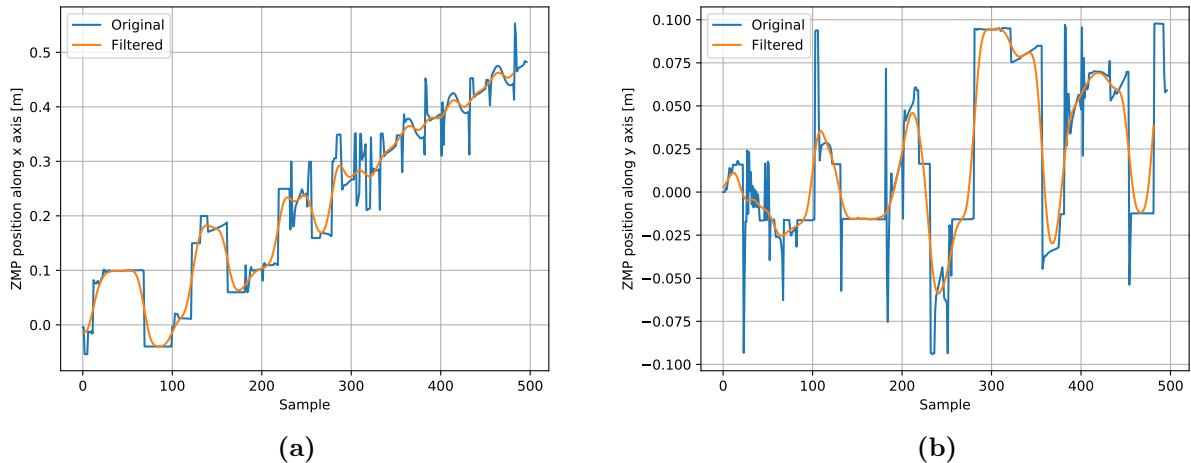


Figure 4.27. Results of the low pass filtering procedure on the ZMP position about the (a) x axis and (b) y axis. Note that the filtered behaviour is actually shifted by half of the order of the filter, but here it is reported overlapped to better understand the smoothing of the original signal.

The implemented low pass filter was obtained by analyzing the Fourier Transform of the ZMP signal produced by the robot when walking while being subjected to a constant force of $\mathbf{F}_{ext} = (5.2, 5.2, 0)$ [N], to detect the cutoff frequency, i.e. a normalized frequency of 0.04. Then

we have applied a finite impulse response (FIR) filter of order 31 obtained with an Hamming window.

This enhancement also increased the performances about other quantities that were estimated starting from the ZMP data, as it is possible to see in Fig. 4.28 for the external force along the y axis

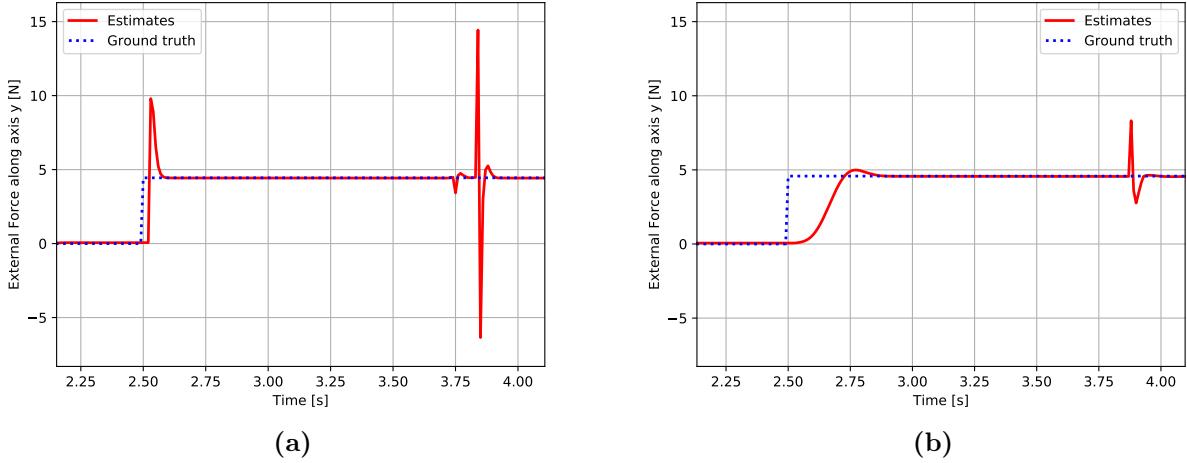


Figure 4.28. Impact of the ZMP filtering on the estimation of a constant external force $\mathbf{F}_{ext} = (5.2, 5.2, 0)$ [N] by a Luenberger observer on the walking NAO: (a) before the application of the filter and (b) after.

An interesting behaviour was discovered by looking at the estimates of the external force along the x axis: they were always affected by a relevant systematic error, *only* along the x axis, independently from the behaviour of the robot or from the observer used to estimate the force. In order to compute this *offset*, we've analysed each experimental setup by isolating the steady-state of the trends and computing the mean of the error. After having collected the distribution of the offsets, we have retrieved its mean and its standard deviation, i.e. 3.84 N and 0.86 N, respectively. Finally we have removed this systematic error from the estimates of the force in all the experiments. A glimpse of this phenomenon and its solution can be seen in Fig. 4.29. We promote the hypothesis of the incorrect assumption, made by DART, about the x CoM coordinate and the real point of application of the external force being coincident. Since we believe that they were not in fact overlapped, this offset could be explained as the term induced by the additional torque produced by the displacement of these two points, as introduced in Section 2.2.

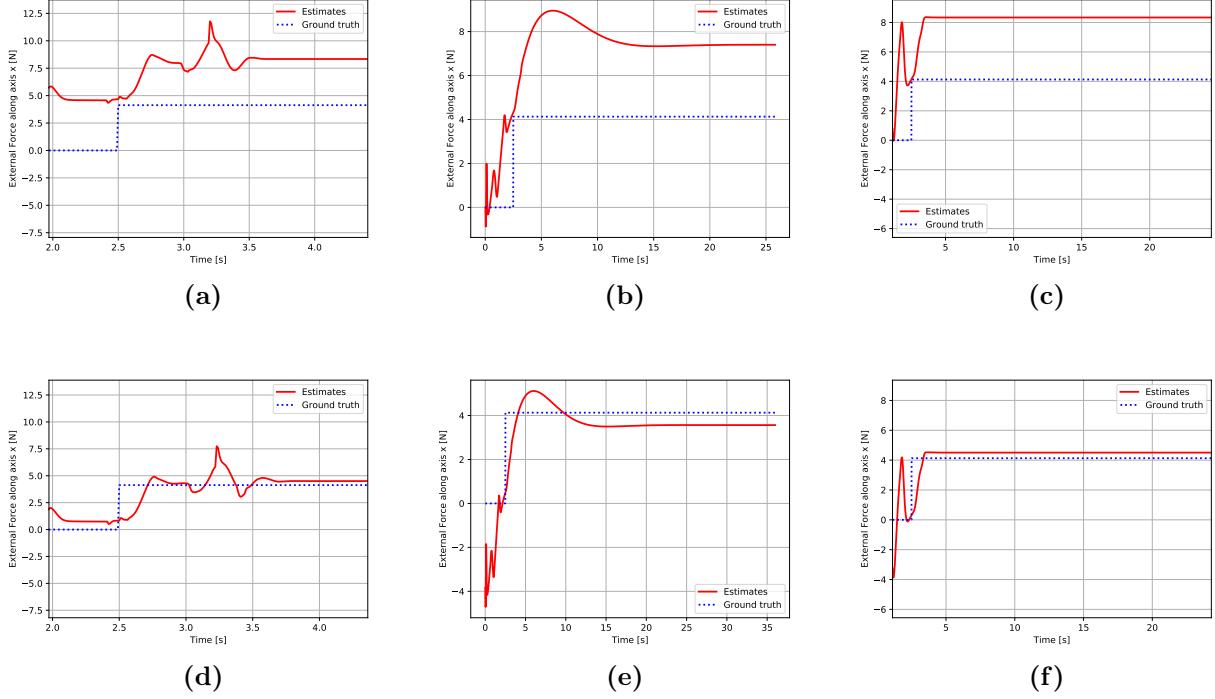


Figure 4.29. Before and after the removal of the x axis offset for the estimation of a constant external force $\mathbf{F}_{ext} = (4.1, 4.1, 0)$ on the robot standing and balancing: (a) Luenberger observer before; (b) Kalman-based observer before; (c) Stephens' observer before; (d) Luenberger observer after; (e) Kalman-based observer after; (f) Stephens' observer after.

Another unusual behaviour was produced by the Luenberger observer. We have found that in almost all the quantities estimated by this observer for almost all experiments, there was an extreme peak during the first instants of the simulations. The peak reached extreme and unreliable values, but then run out as quick as it appeared, as corroborated by the plots in Fig. 4.30 that shows several examples of this phenomenon affecting diverse quantities. However, the overall behaviour, did not seem to be strongly penalized by this phenomenon, therefore we haven't investigated it further.

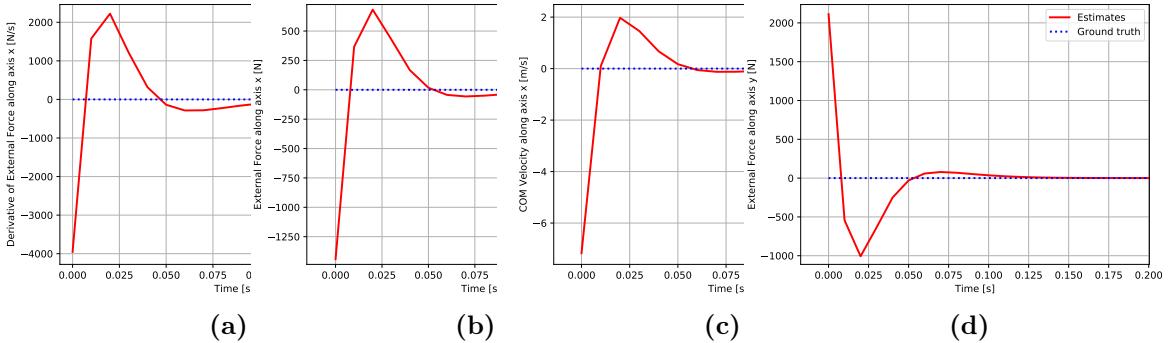


Figure 4.30. Closeup of the peak phenomenon of the Luenberger observer in the application of a constant force to the NAO standing without balancing for different quantities: (a) derivative of the external force in the x axis; (b) external force in the x axis; (c) CoM velocity in the x axis; (d) external force in the y axis. Note the run out speed of the peak.

As seen in Section 4.1.3, the estimates of the external force provided by all the observers, suffer

from high-frequency oscillations, independently from the nature of the applied force. Even if the KF-based observers seem dealing better with this phenomenon, all the observers exhibit a degradation of the performances due to the more complex behavior. However, the corresponding plots show also that the observers have clearly detected the nature of the forces and, in case of the sinusoidal force, also the frequency. From these evidences we came up with the idea of filtering the estimates to have a more direct feeling of their trends. To filter these data we have designed a low pass FIR filter of order 100 obtained with an Hamming window and its results can be seen in the figures of Section 4.1.3. To conveniently smooth the signal, due to the low frequency of the sinusoidal force, we had to increase the order of the filter therefore introducing more delay, that considering the sampling frequency of DART, could prevent this approach to be used in real-time.

As mentioned at the beginning of this chapter the observers that we have implemented in this work haven't limited to provide an estimate of the external force, but also they have generated estimates of the full state. Therefore it was possible to obtain also estimates of other quantities not directly obtainable from neither the simulator nor the real robot, such as the derivative of the external force and the CoM velocity. Several examples of these quantities can be seen in the figures of the Appendix A.

Finally it is worth mentioning that we extensively validated the observers by running several experiments under different setups: from changing the internal parameters of the observers, to the application of the external forces of different nature and magnitude and by collecting the measurements in different ways. All these, and others, arrangements have contributed to progressively increase the performances as to offer a valuable testbed for future researchers.

Chapter 5

Conclusions

In this work we have addressed the task of estimating an unknown external force acting on the center of mass of the NAO robot without relying on the specific sensors aimed at detecting such quantity.

We have implemented a Luenberger, a Kalman Filter based and Stephens' observers to reach this objective. In addition we have included these individual observers in a common framework by building around a structure able to handle these implementations automatically, endowing it with functionalities that may allow its extension and an external control to perform experiments with ease. Then, we have validated this testbed by performing several simulations using the DART platform and analysed the results and intriguing behaviors that raised during the experiments. We have taken the opportunity to investigate further some of these insights to transform them into sources of enhancement.

Through our experiments we have found that the Stephens' observer represents the best candidate for a state-observer devoted to the estimation of an unknown external force under lack of sensors. It is the best trade-off between higher performances and time needed to reach the steady-state. However, even if our results seem promising, especially under certain settings, there is always room for improvements. For example a more accurate and extensive tuning of the hyperparameters of the observers could take place, such as the covariance matrices of the errors for the KF family or a finer pole placement procedure for the Luenberger observer. Another interesting idea is to implement an online filtering of the estimates, especially for very noisy behaviours, such as walking, idea that we have deployed only offline in this work. In addition, even if we have used one of the most accurate open-source simulator available, it would have been interesting to apply our testbed to a real NAO robot to see how the observers would perform.

In summary, through the implementation and validation of this testbed we have contributed to the achievement of this task while laying the foundations for a resource that could be helpful for future researchers.

References

- [1] N. Scianca, M. Cognetti, D. De Simone, L. Lanari, G. Oriolo. *Intrinsically stable MPC for humanoid gait generation.* 2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids), Cancun, 2016, pp. 601-606, doi: 10.1109/HUMANOIDS.2016.7803336.
- [2] F. M. Smaldone, N. Scianca, V. Modugno, L. Lanari, G. Oriolo. *Gait Generation using Intrinsically Stable MPC in the Presence of Persistent Disturbances.* 2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids), Toronto, ON, Canada, 2019, pp. 651-656, doi: 10.1109/Humanoids43949.2019.9035068.
- [3] L. Hawley, W. Suleiman. *External force observer for medium-sized humanoid robots.* 2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids), Cancun, 2016, pp. 366-371, doi: 10.1109/HUMANOIDS.2016.7803302.
- [4] L. Hawley, R. Rahem, W. Suleiman. *Kalman Filter Based Observer for an External Force Applied to Medium-sized Humanoid Robots.* 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, 2018, pp. 1204-1211, doi: 10.1109/IROS.2018.8593610.
- [5] B. J. Stephens. *State estimation for force-controlled humanoid balance using simple models in the presence of modeling error.* 2011 IEEE International Conference on Robotics and Automation, Shanghai, 2011, pp. 3994-3999, doi: 10.1109/ICRA.2011.5980358.

Appendix A

Supplementary Material

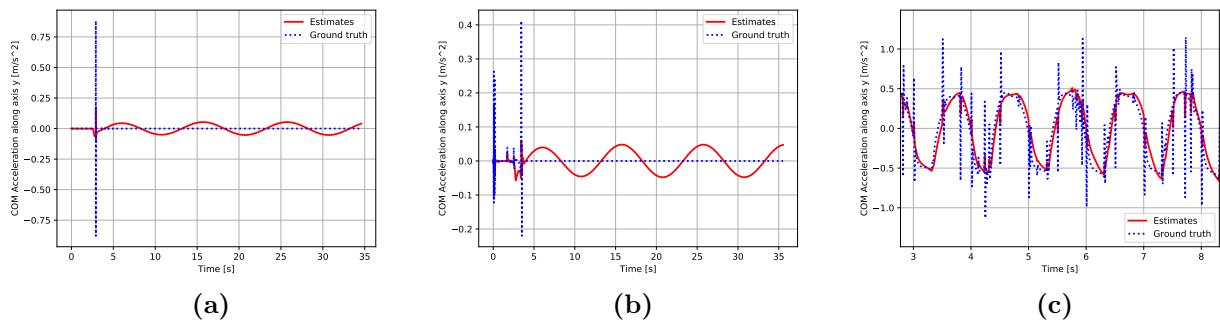


Figure A.1. Estimates of the CoM Acceleration made by the Kalman Filter in case of periodic force: (a) Stand without balance; (b) Stand with balance; (c) Walking.

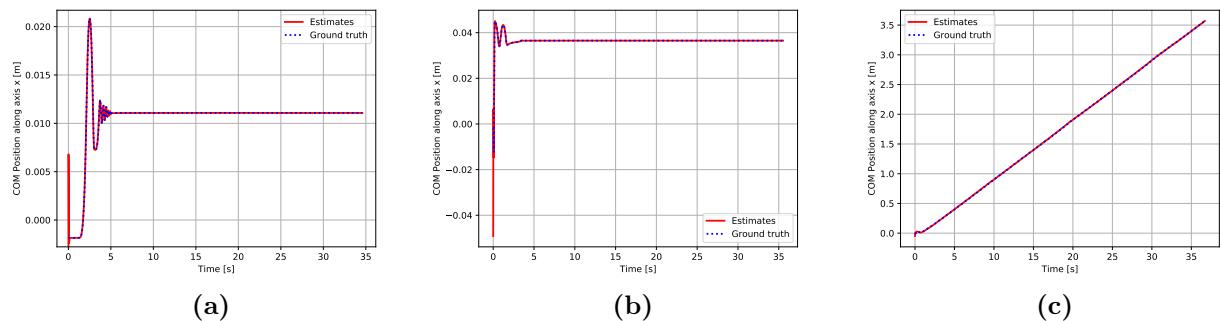


Figure A.2. Estimates of the CoM position made by the Luenberger observer in case of periodic force: (a) Stand without balance; (b) Stand with balance; (c) Walking.

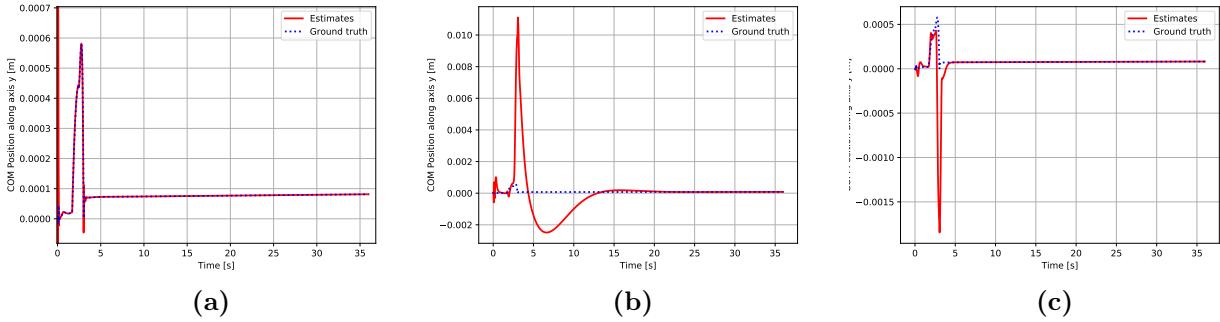


Figure A.3. Estimates of the CoM Position in case of constant force, *Stand with Balance* behaviour: (a) Luenberger; (b) Kalman; (c) Stephens.

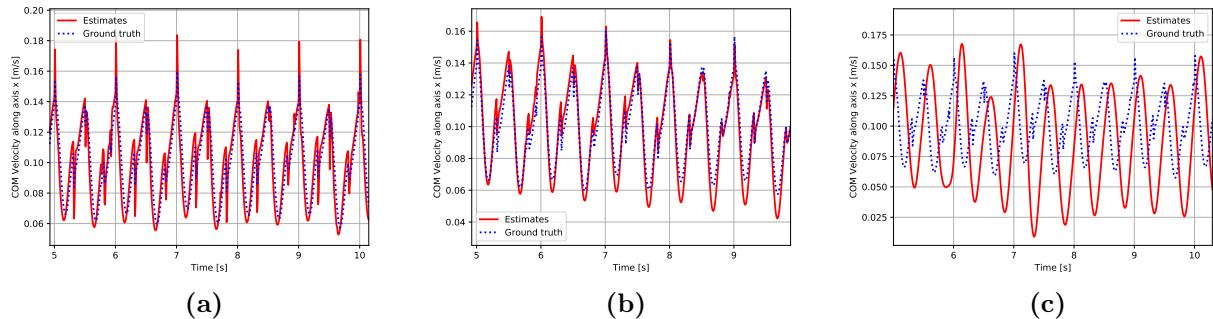


Figure A.4. Estimates of the CoM Velocity in case of periodic force, *Walking* behaviour: (a) Luenberger; (b) Kalman; (c) Stephens.

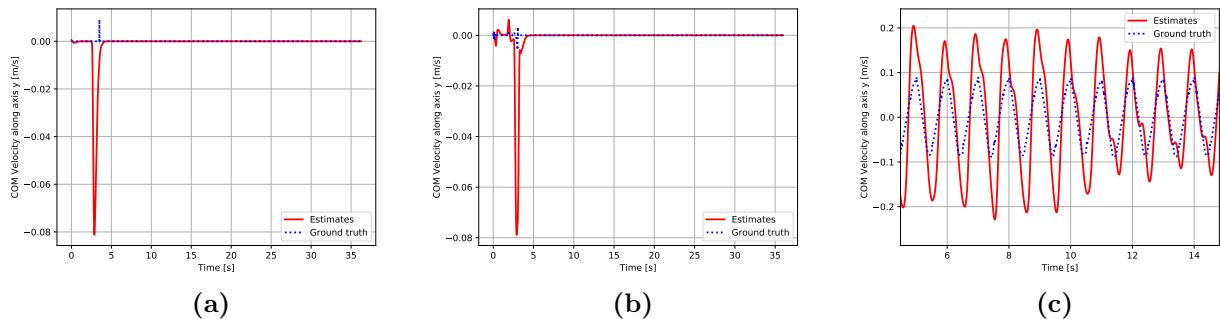


Figure A.5. Estimates of the CoM Velocity in case of constant force, *Walking* behaviour: (a) Without balance; (b) With balance; (c) Walking.

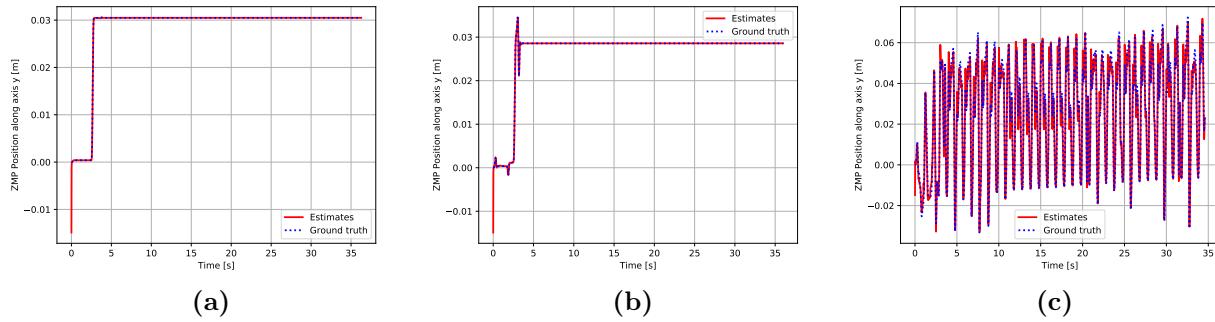


Figure A.6. Estimates of the ZMP position made by the Luenberger observer in case of constant force: (a) Without balance; (b) With balance; (c) Walking.