

# FP2: Torso Pose Estimation on the HRP4 Humanoid Robot

Michele Cipriano, Godwin K. Peprah, Lorenzo Vianello

---

*Supervisor:* Nicola Scianca

*Professors:* Giuseppe Oriolo, Alessandro De Luca

Autonomous and Mobile Robotics, Robotics 2

Department of Computer, Control and Management Engineering

Sapienza University of Rome

# Introduction

- Aim of the project following methodology in [1]
- Pose of the torso for localization of robot points using encoder readings.
- Estimate used to close feedback loop in the MPC gait generator
- Estimation done with EKF using standard humanoid sensors
- Use of accelerometer and the gyroscope of the IMU and improving results with a filter
- Trilateration algorithm used as last step
- Testing the behaviour of the EKF through regulation tasks
- C++ and HRP4 robot model in a V-REP environment used

# Kinematic Model

Let's use the following kinematic model to describe the evolution of the state  $\mathbf{x}$  through time:

$$\dot{\mathbf{x}} = J(\mathbf{q}_s, \mathbf{o}_s) \dot{\mathbf{q}}_s$$

with  $\mathbf{o}_s$  orientation of  $\mathcal{F}_s$  and  $\dot{\mathbf{q}}_s$  velocities of the support joints acting as control inputs.

The robot is equipped with a RGBD camera and an IMU, used to measure the pose of the torso frame:

$$\mathbf{y} = h(\mathbf{x}, \mathbf{q}_n) = \begin{pmatrix} p_t \\ o_t \end{pmatrix}$$

It is now possible to define a discrete-time stochastic system:

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathbf{x}_k + T J(\mathbf{q}_{S,k}, \mathbf{o}_S) \dot{\mathbf{q}}_{S,k} + \mathbf{v}_k \\ \mathbf{y}_k &= h(\mathbf{x}_k, \mathbf{q}_{n,k}) + \mathbf{w}_k\end{aligned}$$

with  $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{V}_k)$  and  $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{W}_k)$  zero-mean white Gaussian noises and  $\mathbf{V}_k \in \mathbb{R}^{6 \times 6}$ ,  $\mathbf{W}_k \in \mathbb{R}^{6 \times 6}$  their respective covariance matrices.

# Extended Kalman Filter: Prediction

At each timestep  $k$ , a prediction  $\hat{\mathbf{x}}_{k+1|k}$  is generated using  $\hat{\mathbf{x}}_k$ :

$$\hat{\mathbf{x}}_{k+1|k} = \hat{\mathbf{x}}_k + J(\mathbf{q}_{s,k}, \mathbf{o}_s) \Delta \mathbf{q}_{s,k}, \quad \Delta \mathbf{q}_{s,k} = \mathbf{q}_{s,k+1} - \mathbf{q}_{s,k}$$

with  $\Delta \mathbf{q}_{s,k} \approx T \dot{\mathbf{q}}_{s,k}$  obtained using encoder readings.

The covariance prediction matrix is updated accordingly:

$$\mathbf{P}_{k+1|k} = \mathbf{P}_k + \mathbf{V}_k$$

The predicted output associated to  $\hat{\mathbf{x}}_{k+1|k}$  is computed as well:

$$\hat{\mathbf{y}}_{k+1|k} = h(\hat{\mathbf{x}}_{k+1|k}, \mathbf{q}_{n,k+1})$$

## Extended Kalman Filter: Correction

It is possible to determine the corrected state estimate by computing:

$$\hat{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{G}_{k+1}\boldsymbol{\nu}_{k+1}$$

with  $\boldsymbol{\nu}_{k+1} = \mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1|k}$  innovation and  $\mathbf{G}_{k+1}$  Kalman gain matrix:

$$\mathbf{G}_{k+1} = \mathbf{P}_{k+1|k} \mathbf{H}_{k+1}^T (\mathbf{H}_{k+1} \mathbf{P}_{k+1|k} \mathbf{H}_{k+1}^T + \mathbf{W}_{k+1})^{-1}$$
$$\mathbf{H}_{k+1} = \left. \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{k+1|k}}$$

The corrected covariance matrix is updated as well:

$$\mathbf{P}_{k+1} = \mathbf{P}_{k+1|k} - \mathbf{G}_{k+1} \mathbf{H}_{k+1} \mathbf{P}_{k+1|k}$$

# Accelerometer Integration

Considering a constant acceleration  $\ddot{\mathbf{p}}_{t,k}$  in an interval  $[t_k, t_{k+1})$ :

$$\dot{\mathbf{p}}_{t,k+1} = \dot{\mathbf{p}}_{t,k} + \ddot{\mathbf{p}}_{t,k}T$$

$$\mathbf{p}_{t,k+1} = \mathbf{p}_{t,k} + \dot{\mathbf{p}}_{t,k}T + \frac{1}{2}\ddot{\mathbf{p}}_{t,k}T^2$$

At each timestep  $k$ , the accelerometer returns the linear acceleration  ${}^t\mathbf{a}_{t,k}$  expressed in  $\mathcal{F}_t$ . It must be transformed to  $\mathcal{F}_w$ :

$$\ddot{\mathbf{p}}_{t,k} = {}^wR_t(\hat{\mathbf{x}}_k){}^t\mathbf{a}_{t,k}$$

with:

$${}^wR_t(\hat{\mathbf{x}}_k) = R_z(\gamma)R_y(\beta)R_x(\alpha)$$

where  $\alpha$ ,  $\beta$  and  $\gamma$  respectively roll, pitch and yaw angles of  $\hat{\mathbf{x}}_k$ .

# Gyroscope Integration

Considering a constant angular velocity in an interval  $[t_k, t_{k+1})$ :

$$\mathbf{o}_{t,k+1} = \mathbf{o}_{t,k} + T\dot{\mathbf{o}}_{t,k}$$

At each timestep  $k$ , the gyroscope returns the angular velocity in  ${}^t\boldsymbol{\omega}_{t,k}$  expressed in  $\mathcal{F}_t$ . It must be transformed to  $\mathcal{F}_w$ :

$$\dot{\mathbf{o}}_{t,k} = T(\hat{\mathbf{x}}_k){}^t\boldsymbol{\omega}_{t,k}$$

with:

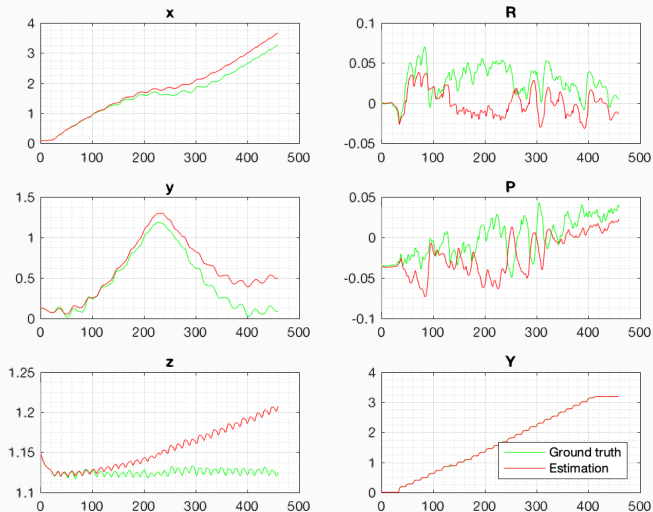
$$T(\hat{\mathbf{x}}_k) = \begin{pmatrix} \cos(\gamma)/\cos(\beta) & \sin(\gamma)/\cos(\beta) & 0 \\ -\sin(\gamma) & \cos(\gamma) & 0 \\ \cos(\gamma)\tan(\beta) & \sin(\gamma)\tan(\beta) & 1 \end{pmatrix}$$

where  $\beta$  and  $\gamma$  respectively pitch and yaw angles of  $\hat{\mathbf{x}}_k$ .



# EKF with IMU Integration

Figure 1: Estimates of the pose of the torso using IMU measurements.



# Filtering Linear Velocities

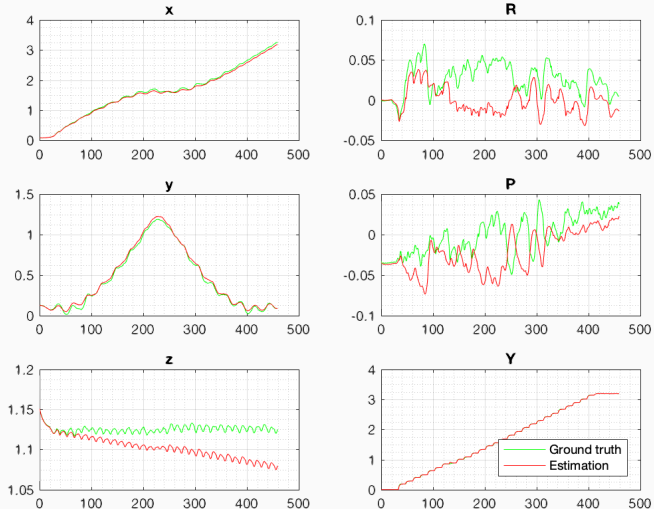
Add to the state an approximated estimation of the velocity in such a way to fix drift accumulation.

Velocity is obtained using position of the robot in 2 successive frames

$$\dot{\mathbf{p}}_{t,k} \approx \hat{\dot{\mathbf{p}}}_{t,k} \approx \hat{\dot{\mathbf{p}}}_{t,k-1} = \frac{\hat{\mathbf{p}}_{t,k} - \hat{\mathbf{p}}_{t,k-1}}{T}$$

# Filtering Linear Velocities

Figure 2: Improved estimate of  $(x, y)^T$  coordinates decreasing the error on  $z$ .



Remove integration and double integration from the calculus of the position.

Required stuff: RGBD camera posed on the head of the robot, identifiable landmarks with known positions.

TODO: QUI FORSE CI STA UNO SCREENSHOOT PER MOSTRARE LA CONFIGUARZINE... E FOTO TIPO TRILATERATION

Obtain the distance of the landmarks from the camera:

$${}^{cam}x = \frac{2p_x - w^{cam}}{w} z \cdot \tan\left(\frac{\phi}{2}\right)$$

$${}^{cam}y = \frac{2p_y - h^{cam}}{h} z \cdot \tan\left(\frac{\phi}{2}\right)$$

$${}^{cam}z = (\lambda_f - \lambda) \cdot p_z + \lambda$$

$$r_i = \left\| \begin{pmatrix} {}^{cam}x_i \\ {}^{cam}y_i \\ {}^{cam}z_i \end{pmatrix} \right\|^2 + \bar{r} \quad (i = 1, 2, \dots, n)$$

FOTO TIPO CAMERA MODEL

# Trilateration

Knowing position of each landmarks in the world frame  $(x_i, y_i, z_i)^T$  and its distance from the camera  $r_i$ , find the position of the camera is equal to resolve an equation for landmark

$$(p_{h,x} - x_i)^2 + (p_{h,y} - y_i)^2 + (p_{h,z} - z_i)^2 = r_i^2 \quad (i = 1, 2, \dots, n)$$

which can be rewritten as linear system  $Ax = b$ :

$$A = \begin{pmatrix} x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_1 & y_3 - y_1 & z_3 - z_1 \\ \vdots & \vdots & \vdots \\ x_n - x_1 & y_n - y_1 & z_n - z_1 \end{pmatrix}, \quad \mathbf{x} = \mathbf{p}_h - \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} b_{21} \\ b_{31} \\ \vdots \\ b_{n1} \end{pmatrix}$$

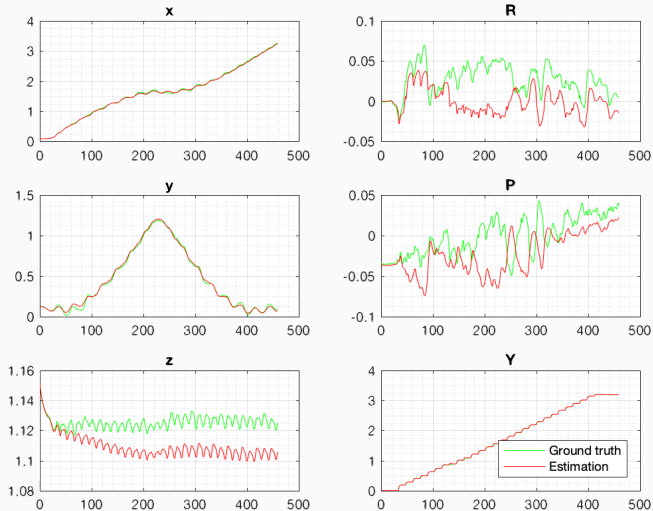
$$b_{k1} = \frac{1}{2} [r_1^2 + r_k^2 + (x_k - x_1)^2 + (y_k - y_1)^2 + (z_k - z_1)^2] \quad (k = 2, 3, \dots, n)$$

Hence, the position of the torso can be determined by:

$$\mathbf{p}_h = \mathbf{x} + \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} \quad \mathbf{p}_t = \mathbf{p}_h - {}^w\mathbf{R}_t(\hat{\mathbf{x}}_k)({}^t\mathbf{p}_h - {}^t\mathbf{p}_t)$$

We need at least 5 landmarks for the computation: 4 for obtain a square matrix, and at least 1 more for execute the pseudoinverse that bring results also in case of noisy measurements.

Figure 3: Further improved estimation for  $x, y, z$ .





## MPC Loop Closure

---

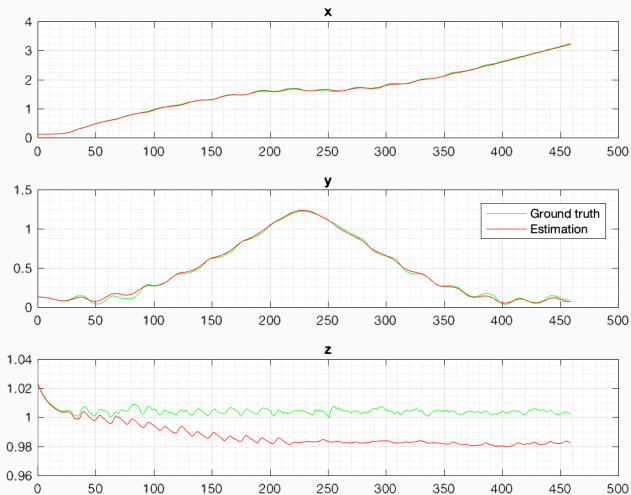
# MPC Loop Closure

The MPC need the position of the CoM with respect to the support foot to close the loop.

We want to estimate the same position but with the support foot reference frame with roll e pitch (wrt world frame) equal to zero.

# MPC Loop Closure

Figure 4: Estimated position of the CoM with respect to the ground truth.



Given the estimate of the pose of the torso  $(\hat{\mathbf{p}}_t, \hat{\mathbf{o}}_t)^T$ , it is possible to obtain the estimate of the position of the support foot  $(\hat{\mathbf{p}}_s, \hat{\mathbf{o}}_s)^T$  and of the CoM  $(\hat{\mathbf{p}}_{CoM}, \hat{\mathbf{o}}_{CoM})^T$  using the kinematics relations.

The position of the CoM in a rotated reference frame of the support foot  $\mathcal{F}_{s'}$  which has the z-axis orthogonal to the floor is:

$${}^{s'}\hat{\mathbf{p}}_{CoM} = \mathbf{R}_z^T(\gamma)(\hat{\mathbf{p}}_{CoM} - \hat{\mathbf{p}}_s)$$

This position can be used as a reference signal in the MPC in order to generate the gait of the humanoid robot.

# Regulation

---

# Kinematic Model of the Unicycle

Let's model the robot as a unicycle:

$$\dot{x} = v \cos(\theta)$$

$$\dot{y} = v \sin(\theta)$$

$$\dot{\theta} = \omega$$

with  $v$  and  $\omega$  respectively linear and angular velocity.

# Proportional Controller

Let's consider the following control law:

$$v = k_1 \|\mathbf{p}_g - \hat{\mathbf{p}}_t\|$$

$$\omega = k_2 e_\theta$$

with  $e_\theta$  angle between the sagittal vector of the unicycle and the vector pointing from the unicycle towards the goal,  $k_1 = 0.18$  and  $k_2 = 0.014$ . Forcing  $v$  and  $\omega$  to zero when  $\|\mathbf{p}_g - \hat{\mathbf{p}}_t\| < 0.25$ . Desired and final configuration:

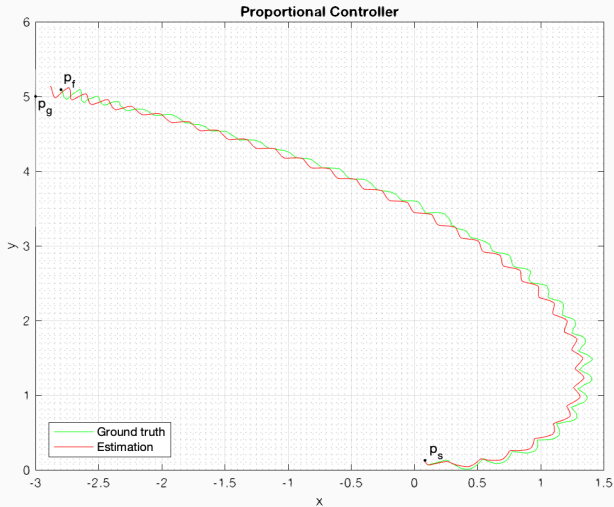
$$\mathbf{q}_g = (-3, 5, \cdot)^T$$

$$\mathbf{q}_f = (-2.798, 5.090, 0.98\pi)^T$$

$$\hat{\mathbf{q}}_f = (-2.885, 5.127, 0.979\pi)^T$$

# Proportional Controller: x-y Plot

Figure 5: Trajectory followed by the robot when using the controller defined above. Setting velocities to zero when  $\|p_g - \hat{p}_t\| < 0.25$ .





Let's express the coordinates of the unicycle in a reference frame  $\mathcal{F}_g$  fixed at a position  $(x_g, y_g)^T$  and rotated by  $\theta_g$  around  $\mathcal{F}_w$ :

$$\begin{pmatrix} {}^g x \\ {}^g y \\ {}^g \theta \end{pmatrix} = R_z^T(\theta_g) \begin{pmatrix} x - x_g \\ y - y_g \\ \theta - \theta_g \end{pmatrix}$$

# Cartesian Regulation

Let's consider the following control law:

$$v = -k_1({}^g x \cos({}^g \theta) + {}^g y \sin({}^g \theta))$$

$$\omega = k_2(\text{Atan2}({}^g y, {}^g x) - {}^g \theta + \pi)$$

with  $k_1 = 0.07$  and  $k_2 = 0.01$ . Forcing  $v$  and  $\omega$  to zero when  $\|\mathbf{p}_g - \hat{\mathbf{p}}_t\| < 0.2$ . Desired and final configuration:

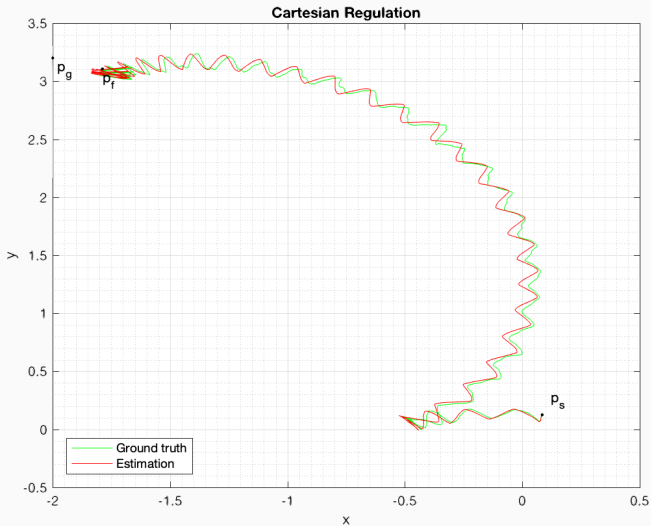
$$\mathbf{q}_g = (-2, 3.2, \cdot)^T$$

$$\mathbf{q}_f = (-1.788, 3.105, 1.562\pi)^T$$

$$\hat{\mathbf{q}}_f = (-1.823, 3.108, 1.562\pi)^T$$

# Cartesian Regulation: x-y Plot

**Figure 6:** Trajectory followed by the robot when using the controller defined above. Setting velocities to zero when  $\|p_g - \hat{p}_t\| < 0.2$ .



# Kinematic Model of the Unicycle in Polar Coordinates

Let's model the robot as a unicycle using polar coordinates:

$$\begin{aligned}\dot{\rho}_r &= -v \cos(\gamma_r) \\ \dot{\gamma}_r &= \frac{\sin(\gamma_r)}{\rho_r} v - \omega \\ \dot{\delta}_r &= \frac{\sin(\gamma_r)}{\rho_r} v\end{aligned}$$

Polar coordinates can be obtained from the generalized coordinates of the unicycle  $(x, y, \theta)^T$  by computing:

$$\begin{aligned}\rho_r &= \sqrt{{}^g x^2 + {}^g y^2} \\ \gamma_r &= \text{Atan2}({}^g y, {}^g x) - {}^g \theta + \pi \\ \delta_r &= \gamma_r + {}^g \theta\end{aligned}$$

# Posture Regulation

Let's consider the following control law:

$$v = k_1 \rho_r \cos(\gamma_r)$$

$$w = k_2 \gamma_r + k_1 \frac{\sin(\gamma_r) \cos(\gamma_r)}{\gamma_r} (\gamma_r + k_3 \delta_r)$$

with  $k_1 = 0.1$ ,  $k_2 = 0.007$ ,  $k_3 = 0.004$ . Forcing  $v$  and  $w$  to zero when  $\rho_r < 0.2$ . Desired and final configuration:

$$\mathbf{q}_g = (-2, 3.2, \pi)^T$$

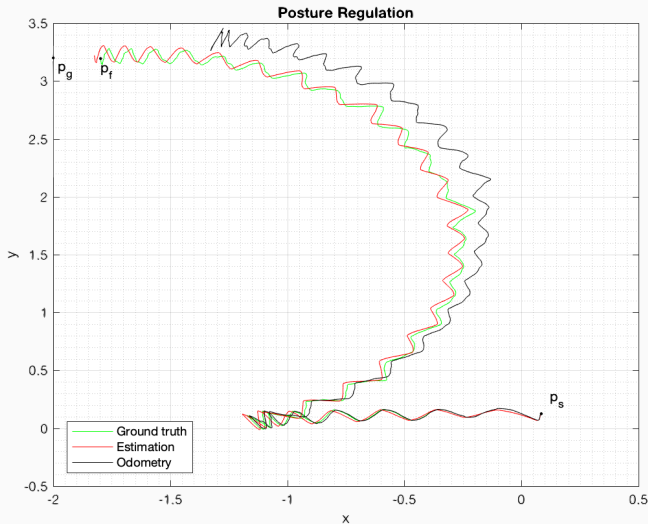
$$\mathbf{q}_f = (-1.797, 3.194, 1.024\pi)^T$$

$$\hat{\mathbf{q}}_f = (-1.824, 3.222, 1.024\pi)^T$$

$$\bar{\mathbf{q}}_f = (-1.282, 3.424, 0.912\pi)^T$$

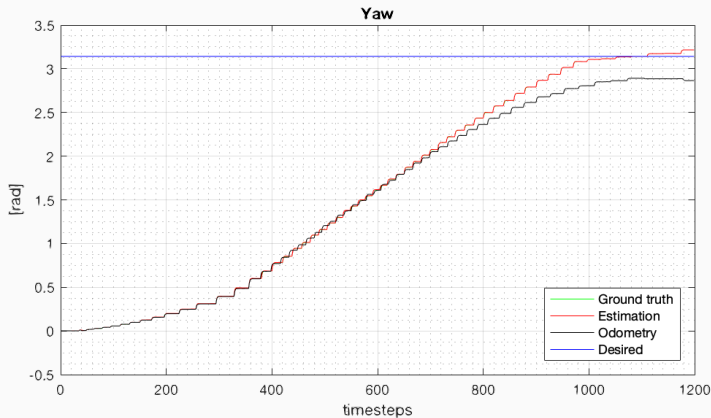
# Posture Regulation: x-y Plot

Figure 7: Trajectory followed by the robot when using the controller defined above. Setting velocities to zero when  $\rho_r < 0.2$ .



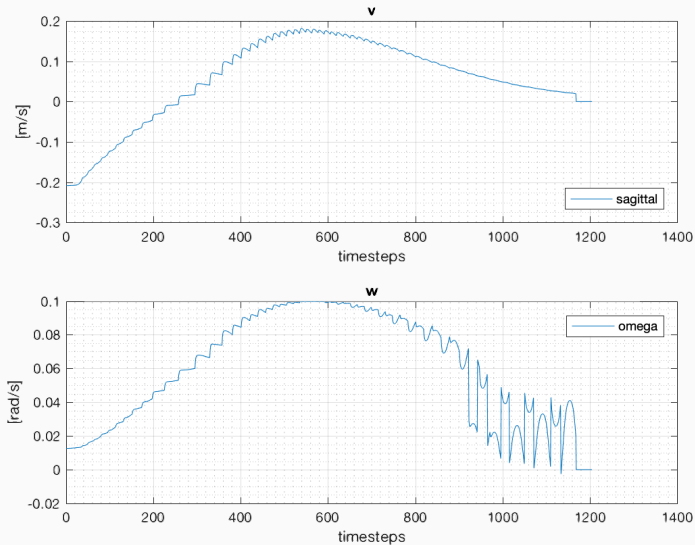
# Posture Regulation: Yaw Plot

Figure 8: Evolution of the yaw while following the trajectory of the previous slide.



# Posture Regulation: Velocity Profile Plots

Figure 9: Profiles of the linear velocity  $v$  and the angular velocity  $w$  used to control the robot.








# Conclusion

Conclusion.

Q&A

# References

-  G. Oriolo, A. Paolillo, L. Rosa, and M. Vendittelli, “Humanoid odometric localization integrating kinematic, inertial and visual information,” *Auton. Robots*, vol. 40, no. 5, pp. 867–879, 2016.
-  W. Hereman and J. William S. Murphy, “Determination of a Position in Three Dimensions Using Trilateration and Approximate Distances,” 1995.
-  B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Springer Publishing Company, Incorporated, 1st ed., 2008.