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

Intro.

Kinematic Model

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

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

with $\boldsymbol{o}_s$ orientation of $\mathcal{F}_s$ and $\dot{\boldsymbol{q}}_s$ 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:

$$\boldsymbol{y} = h(\boldsymbol{x}, \boldsymbol{q}_n) = \begin{pmatrix} \boldsymbol{p}_t \\ \boldsymbol{o}_t \end{pmatrix}$$

Extended Kalman Filter

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

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

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

Extended Kalman Filter: Prediction

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

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

with $\Delta\boldsymbol{q}_{s,k} \approx T\dot{\boldsymbol{q}}_{s,k}$ obtained using encoder readings.
The covariance prediction matrix is updated accordingly:

$$P_{k+1|k} = P_k + V_k$$

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

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

Extended Kalman Filter: Correction

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

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

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

$$G_{k+1} = P_{k+1|k}H_{k+1}^T \left(H_{k+1}P_{k+1|k}H_{k+1}^T + W_{k+1}\right)^{-1}$$

$$H_{k+1} = \left.\frac{\partial h}{\partial x}\right|_{x=\hat{x}_{k+1|k}}$$

The corrected covariance matrix is updated as well:

$$P_{k+1} = P_{k+1|k} - G_{k+1}H_{k+1}P_{k+1|k}$$

Extended Kalman Filter: Correction

Note that, in our implementation, we defined V and W as:

$$V = \mathrm{diag}\{5, 5, 5, 100, 100, 100\} \cdot 10^{-6}$$
$$W = \mathrm{diag}\{5, 5, 5, 5 \cdot 10^{-2}, 5 \cdot 10^{-2}, 5 \cdot 10^{-2}\} \cdot 10^{-2}$$

Accelerometer Integration

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

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

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

$$\ddot{\boldsymbol{p}}_{t,k} = {}^{w}\boldsymbol{R}_t(\hat{\boldsymbol{x}}_k)\,{}^{t}\boldsymbol{a}_{t,k}$$

with:

$${}^{w}\boldsymbol{R}_t(\hat{\boldsymbol{x}}_k) = \boldsymbol{R}_z(\gamma)\boldsymbol{R}_y(\beta)\boldsymbol{R}_x(\alpha)$$

where α, β and γ respectively roll, pitch and yaw angles of $\hat{\boldsymbol{x}}_k$.

Gyroscope Integration

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

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

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 $\mathcal{F}_w$:

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

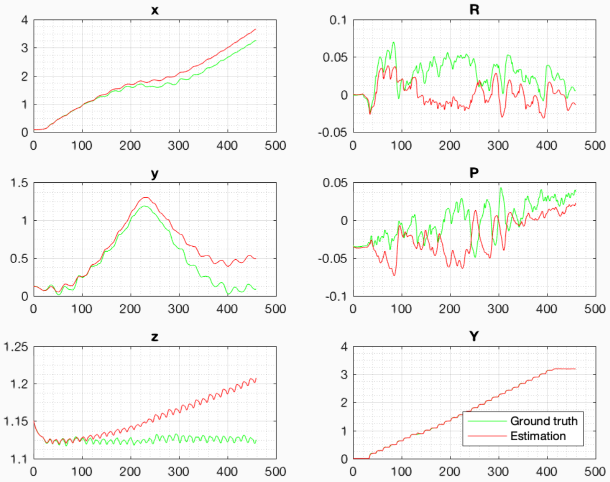
with:

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

where β and γ respectively pitch and yaw angles of $\hat{\boldsymbol{x}}_k$.

EKF with IMU Integration

Figure 1: Experiments (5.1).



Filtering Linear Velocities

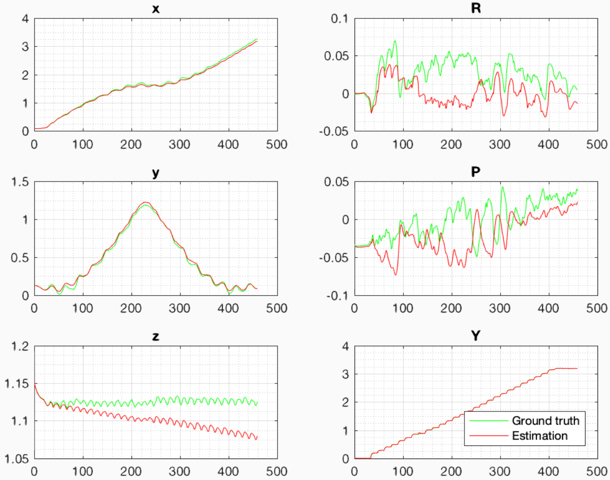
Add to the state an approximated estimation of the velocity in such a way to fix drift accumulation.

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

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

Filtering Linear Velocities

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



Trilateration

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

Trilateration

Obtain the distance of the landmarks from the camera:

$$
\begin{aligned}
{}^{cam}x &= \frac{2p_x - w}{w} {}^{cam}z \cdot tan\left(\frac{\phi}{2}\right) \\
{}^{cam}y &= \frac{2p_y - h}{h} {}^{cam}z \cdot tan\left(\frac{\phi}{2}\right) \\
{}^{cam}z &= (\lambda_f - \lambda) \cdot p_z + \lambda
\end{aligned}
$$

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

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

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

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

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

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

Trilateration

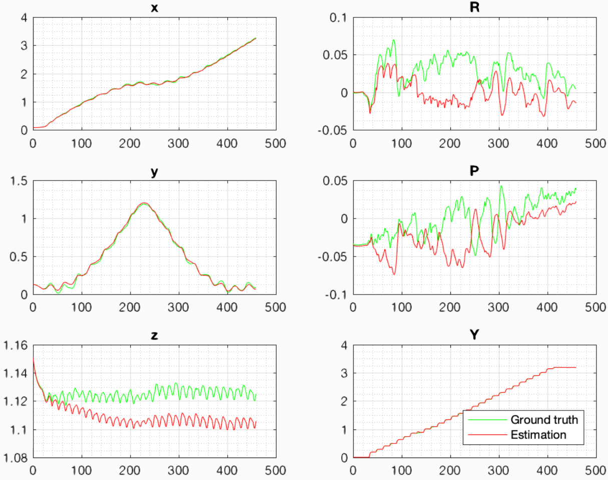
Hence, the position of the torso can be determined by:

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

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.

Trilateration

Figure 3: Improved estimation for x, y, z



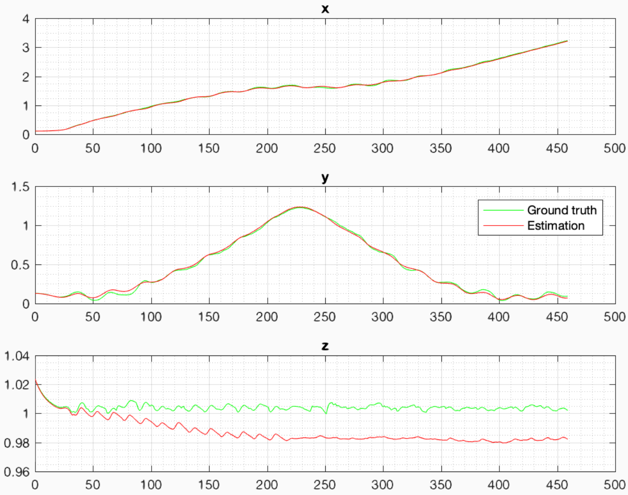
MPC Loop Closure

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.

Figure 4: Todo.



MPC Loop CLosure

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

Brief description:

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

with v and ω respectively linear and angular velocity.

Proportional Controller

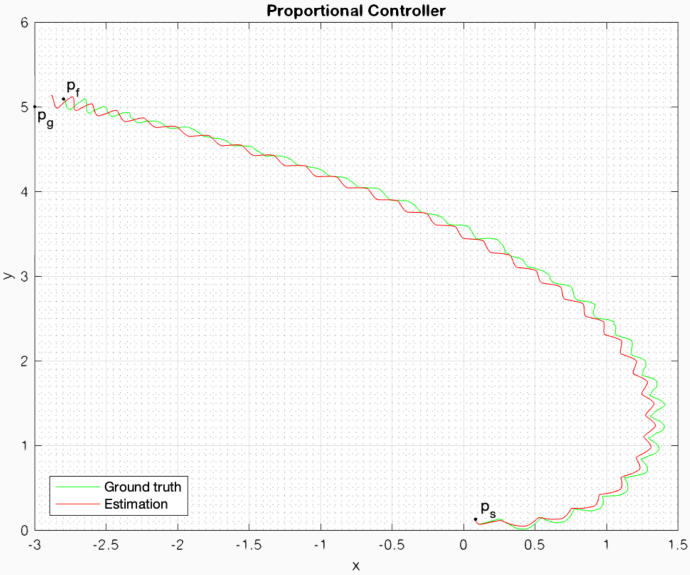
Control law:

$$v = k_1 \left\| \boldsymbol{p}_g - \hat{\boldsymbol{p}}_t \right\|$$
$$\omega = k_2 e_\theta$$

with e_θ 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 ω to zero when $\|\boldsymbol{p}_g - \hat{\boldsymbol{p}}_t\| < 0.25$. Desired and final configuration:

$$\boldsymbol{q}_g = (-3, 5, \cdot)^T$$
$$\boldsymbol{q}_f = (-2.798, 5.090, 0.98\pi)^T$$
$$\hat{\boldsymbol{q}}_f = (-2.885, 5.127, 0.979\pi)^T$$

Proportional Controller: x-y Plot



Cartesian Regulation

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 θ_g around $\mathcal{F}_w$:

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

Cartesian Regulation

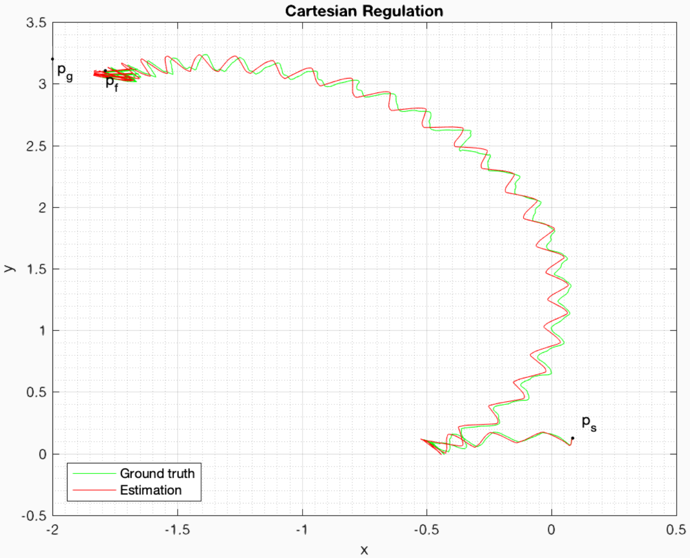
Control law:

$$v = -k_1({}^g x \cos({}^g\theta) + {}^g y \sin({}^g\theta))$$
$$\omega = k_2(\text{Atan2}({}^g y, {}^g x) - {}^g\theta + \pi)$$

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

$$\boldsymbol{q}_g = (-2, 3.2, \cdot)^T$$
$$\boldsymbol{q}_f = (-1.788, 3.105, 1.562\pi)^T$$
$$\hat{\boldsymbol{q}}_f = (-1.823, 3.108, 1.562\pi)^T$$

Cartesian Regulation: x-y Plot



Kinematic Model of the Unicycle in Polar Coordinates

Brief description:

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

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

$$\rho_r = \sqrt{{}^g x^2 + {}^g y^2}$$
$$\gamma_r = \mathrm{Atan2}({}^g y, {}^g x) - {}^g\theta + \pi$$
$$\delta_r = \gamma_r + {}^g\theta$$

Posture Regulation

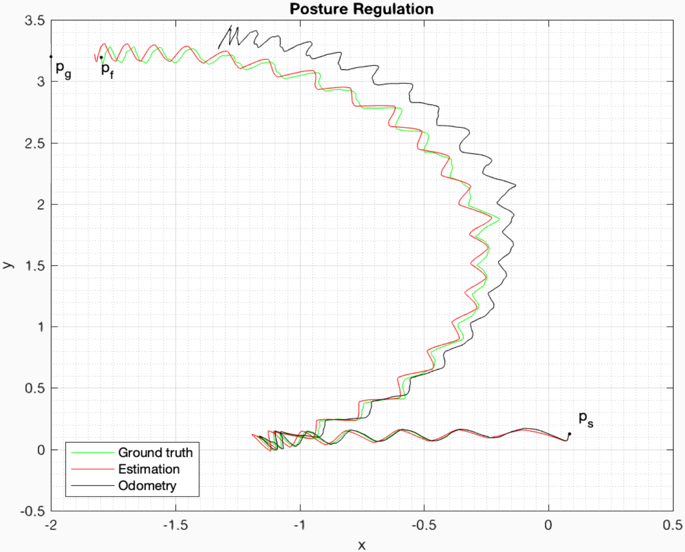
Control law:

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

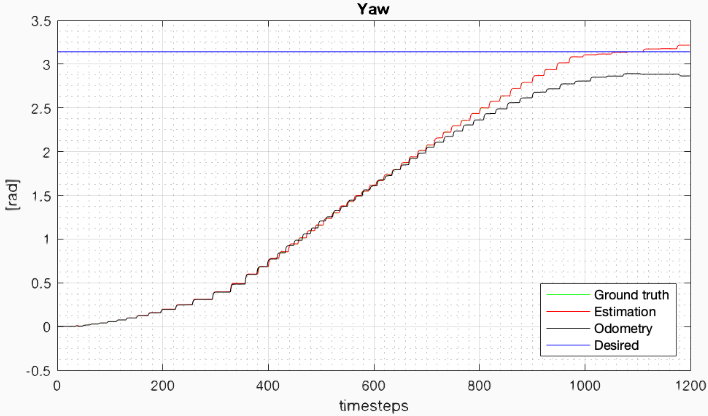
with $k_1 = 0.1$, $k_2 = 0.007$ $k_3 = 0.004$. Forcing v and ω to zero when $\rho_r < 0.2$. Desired and final configuration:

$$\boldsymbol{q}_g = (-2, 3.2, \pi)^T$$
$$\boldsymbol{q}_f = (-1.797, 3.194, 1.024\pi)^T$$
$$\hat{\boldsymbol{q}}_f = (-1.824, 3.222, 1.024\pi)^T$$
$$\bar{\boldsymbol{q}}_f = (-1.282, 3.424, 0.912\pi)^T$$

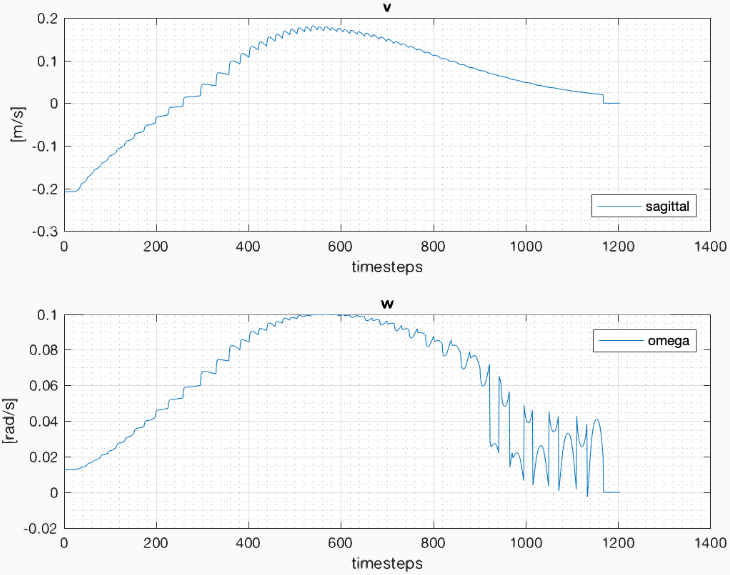
Posture Regulation: x-y Plot



Posture Regulation: Yaw Plot



Posture Regulation: Velocity Profile Plots



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.