Modeling and simulation of an UR5 Robot Manipulator

Battistini Enrico
Samorì Filippo
Subini Jacopo

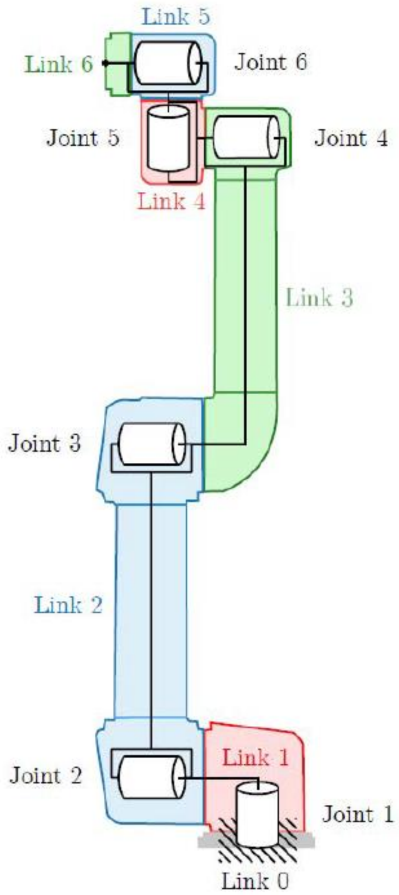
Contents

The UR5 is a robotic arm manipulator with 6-dof.

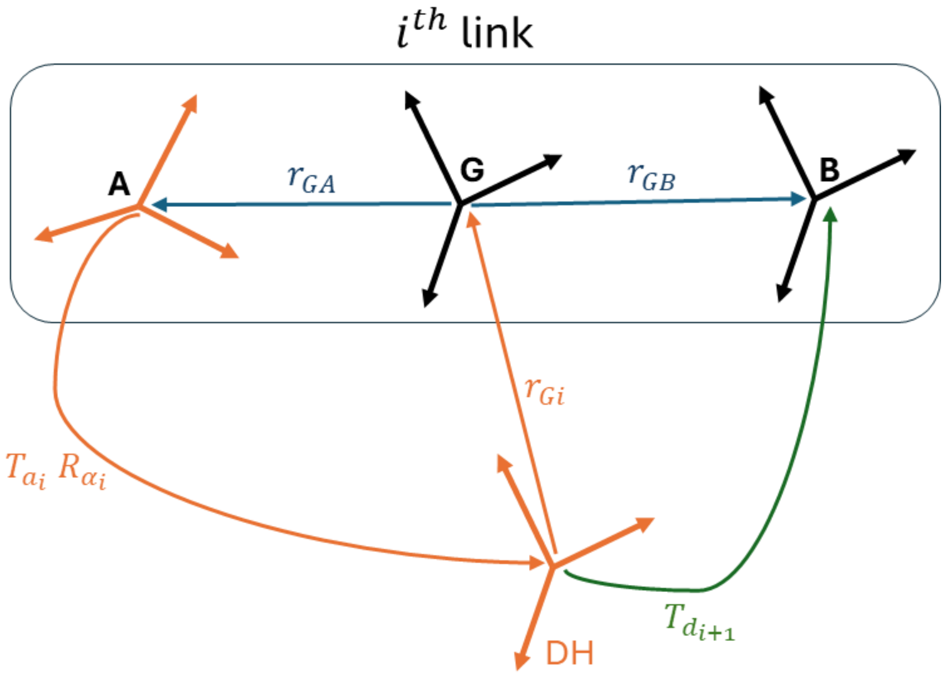
The hardware parts that have been modelled are:

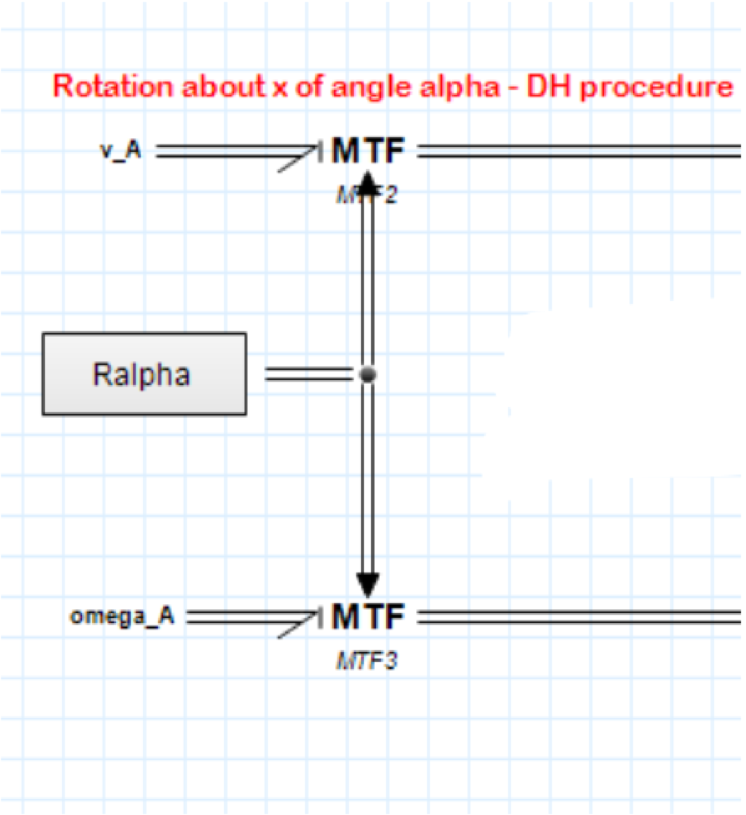
- The HFUS-20 version 2SH for the first three joints
- The HFUS-14 version 2SH for the last three joints
- The manipulator dynamics, composed of 6 links and 6 joints

In addition, a PD + Gravity compensation control has been implemented to evaluate the model's behaviour and its validity.



Inside the model, in order to change properly the reference frame when we compute the vectors need to be used, we perform the rotations given by the Denavit-Hartenberg convention using Modulated Transformers blocks. In this way we are always in the correct coordinate frame.





Example of frame rotation

Denavit – Hartenberg Table

joint	q_i [°]	d_i [m]	a_i [m]	α_i [°]	$\tilde{q}_i$ [°]
Base	q_1	0.089159	0	90	0
Shoulder	q_2	0	-0.425	0	0
Elbow	q_3	0	-0.39225	0	0
Wrist1	q_4	0.10915	0	90	0
Wrist2	q_5	0.09465	0	-90	0
Wrist3	q_6	0.0823	0	0	0

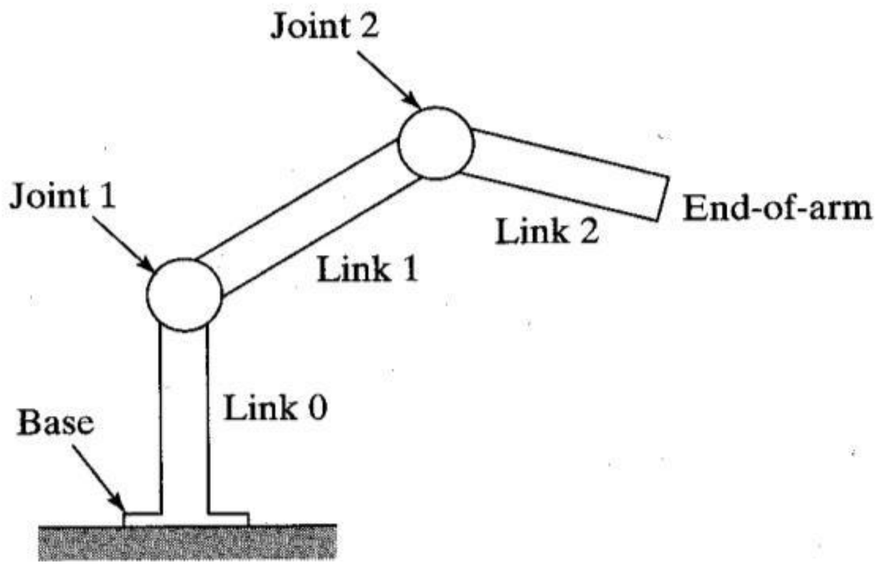
$$^{n-1}T_n = \mathrm{Trans}_{z_{n-1}}(d_n) \cdot \mathrm{Rot}_{z_{n-1}}(\theta_n) \cdot \mathrm{Trans}_{x_n}(r_n) \cdot \mathrm{Rot}_{x_n}(\alpha_n)$$

A robotic arm is a chain of rigid bodies connected by revolute joints. In order to implement a model of the UR5, we need to model each link as a rigid body with two port connections. Each port corresponds to the center of a joint where torques and forces are exchanged.

The rigid body dynamics is described by the Euler Equations:

$$\boxed{\begin{aligned}\dot{p} &= F - \omega \times p \\ \dot{h} &= \tau - \omega \times h\end{aligned}}$$

Where p, h, F and τ are described in the Body reference frame and ω is the rotation vector of the rigid body with respect to the inertial frame.



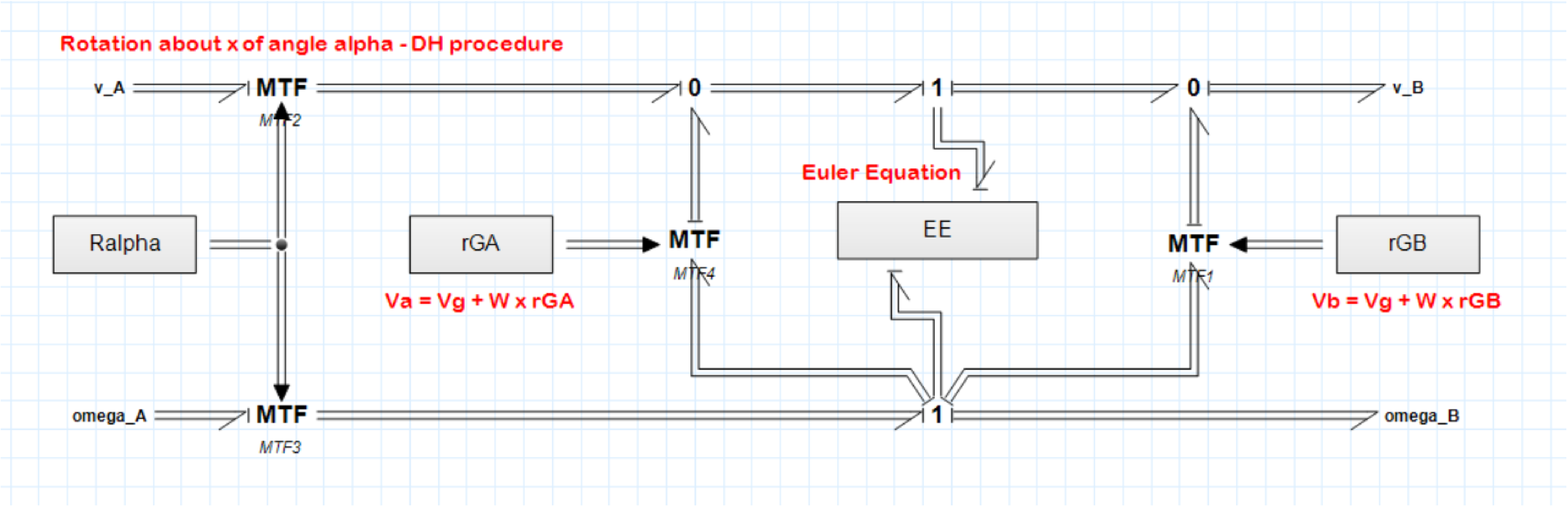
To solve the Euler Equations we need to write the resultant force and torque effects on the rigid body with respect to the barycenter.

We can write a kinematic relation between velocities at ports (A and B) and the velocity at the barycenter G :

$$\boxed{\begin{gathered} v_A = v_G + \omega \times r_{GA} \\ v_B = v_G + \omega \times r_{GB} \end{gathered}}$$

These relations are used to compute the rigid body dynamics from the forces and torques exchanged at the ports.

Link Submodel

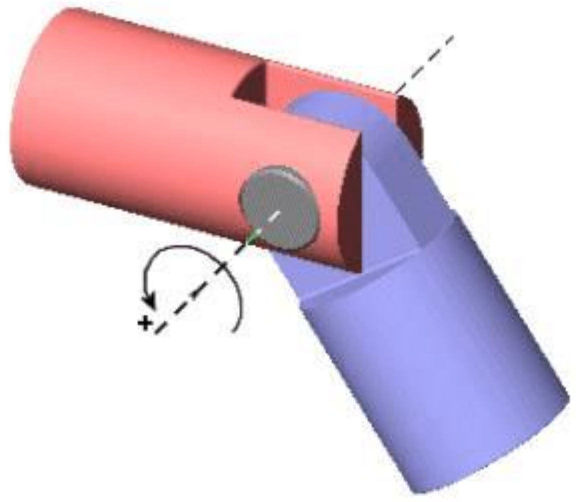


The revolute joint is modelled as a constraint on the relative motion between links.

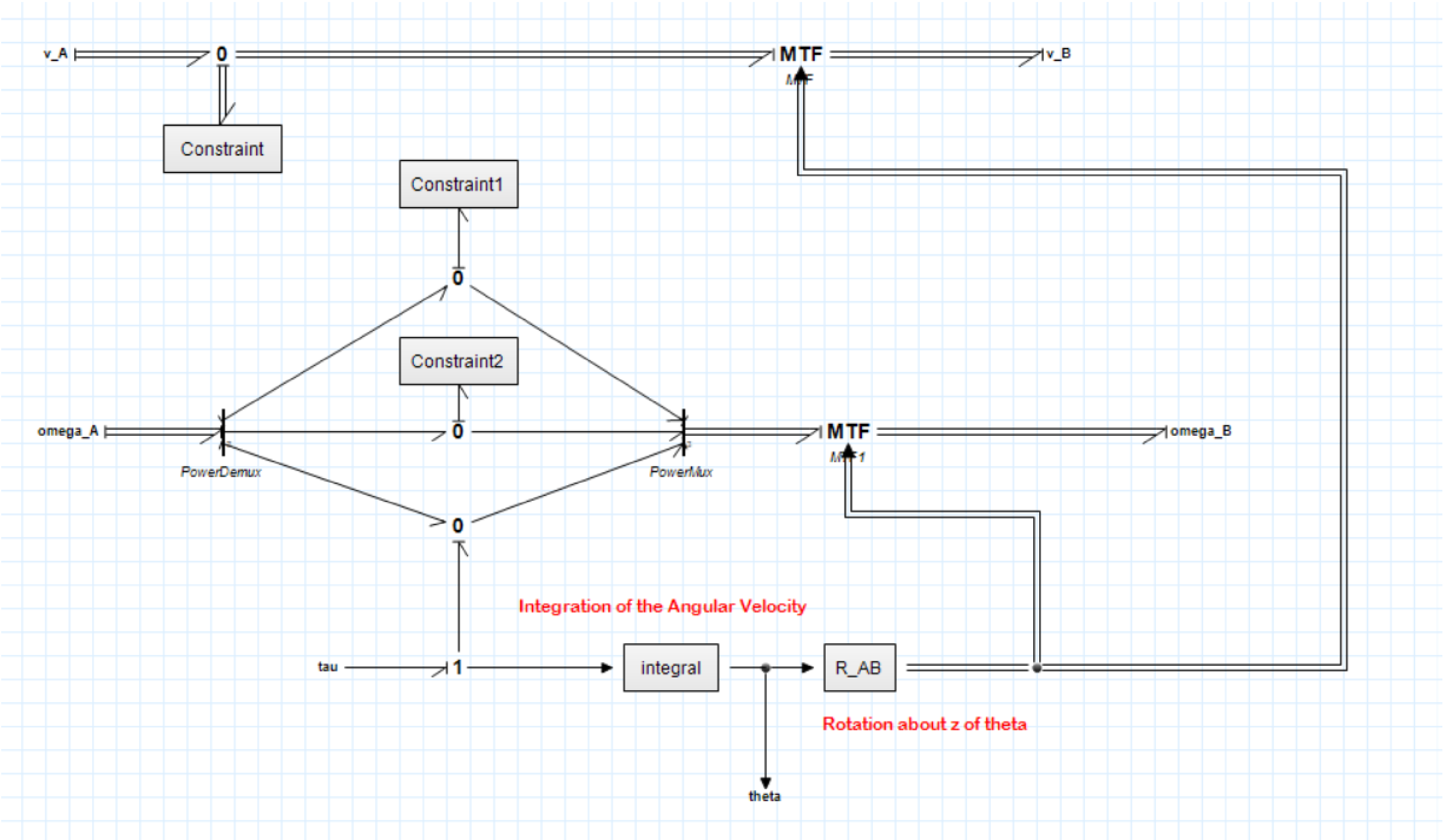
The joint has 1 degree of freedom, the rotation about z, while the other motions are constrained to have a zero relative velocity. The input to the joint is the torque applied to the z axis.

The angle Theta is computed as the integral of the angular velocity.

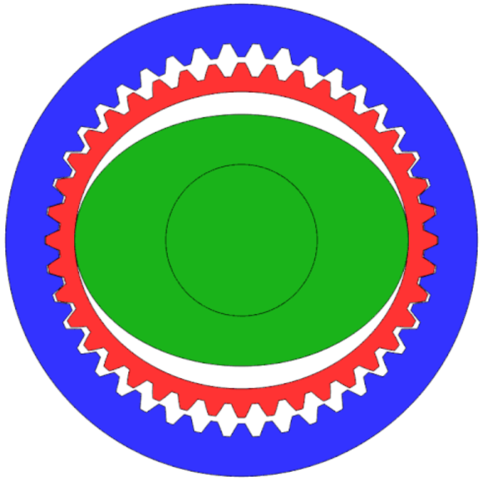
Then the velocities in the port B of the joint are transformed in a new reference frame rotated about z by the angle Theta, to follow the Denavit Hartenberg convention.



Joint Submodel



The harmonic drive is a mechanical gear realized with a compliance mechanism.



It is made of three main components:
- The flex spline (red)
- The wave generator (green)
- The Circular Spline (blue)

The flex spline has two teeth less than the Circular spline. In this way, we have a relative motion between the parts and the corresponding reduction ratio:

$$R = \frac{flex\ spline\ teeth\ - circular\ spline\ teeth}{flex\ spline\ teeth} = -\frac{1}{100}$$

The dumping on the wave generator is described by the following equations:

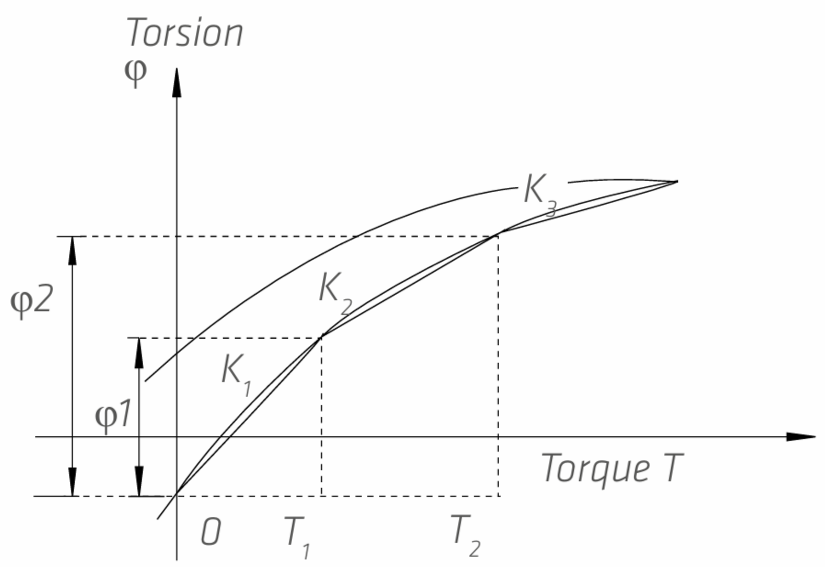
$$\tau = B_{mw} v_{w} + sign(v_{w})(B_{mw}^{+}, B_{mw}^{-})$$

There is a viscous dumping contribution and a static friction that depends on the motion direction.

For the flex spline, the damping equation is the same, while the stiffness has a non linear constitutive equation:

$$\tau = K(\theta)\theta$$

Where $K(\theta)$ can assume three possible constant values (K_1 , K_2 , K_3) and the transition between different stiffness values depends by the torque.



K_1, K_2, K_3 = Torsional stiffness, w = Output angle
φ1 = Torsion angle, with output torque T_1
φ2 = Torsion angle, with output torque T_2

Harmonic drive submodel

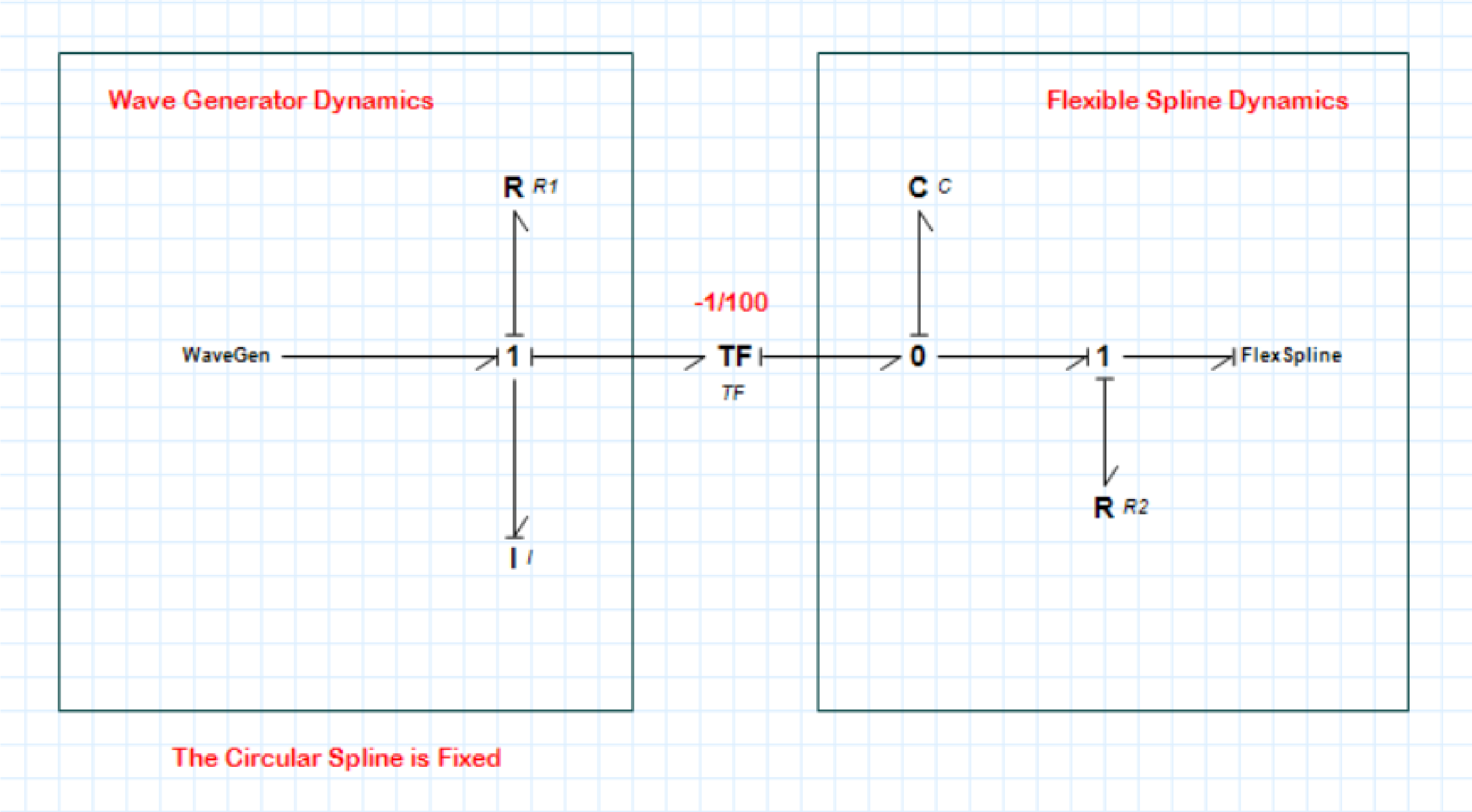
HFUS-20 version 2SH parameters:

$$K_1 = 16e^3$$
$$K_2 = 25e^3$$
$$K_3 = 29e^3$$
$$T_1 = 7$$
$$T_2 = 25$$
$$J_w = 0.404e^{-4}$$
$$B_{mw} = 6.4e^{-4}$$
$$B_{mw}^+ = 0.008$$
$$B_{mw}^- = 0.007$$
$$B_f = 6.4e^{-4}$$
$$B_f^+ = 0.4$$
$$B_f^- = 0.4$$

HFUS-14 version 2SH parameters:

$$K_1 = 4.7e^3$$
$$K_2 = 6.1e^3$$
$$K_3 = 7.1e^3$$
$$T_1 = 2$$
$$T_2 = 6.9$$
$$J_w = 0.091\mathrm{e}^{-4}$$
$$B_{mw} = 6.4e^{-4}$$
$$B_{mw}^+ = 0.008$$
$$B_{mw}^- = 0.007$$
$$B_f = 6.4e^{-4}$$
$$B_f^+ = 0.4$$
$$B_f^- = 0.4$$

Harmonic drive submodel



The control policy we chose is a PD + Gravity compensation controller. This controller has a Lyapunov Stability proof and it is easily implementable.

The Gravity is compensated by a feedforward term, while the PD controllers act as a decentralized controllers for each link.

The compensation of the gravity guarantees that at the steady state, the error is null. This can be seen from the dynamic model of the robot:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + D\dot{q} + G(q) = K_P e + K_D \dot{e} + G(q)$$

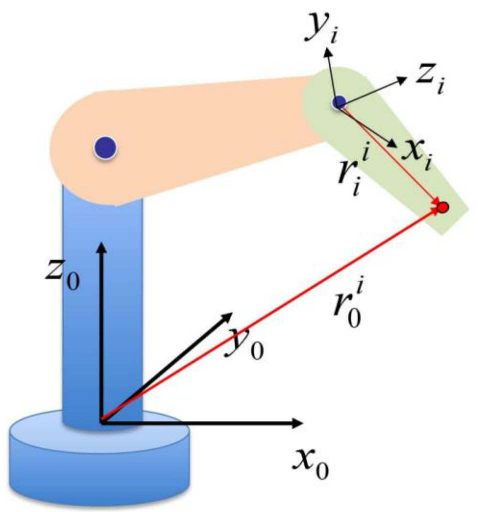
That, since it can be proven the controller is stable and $\dot{e} \rightarrow 0$, at the steady state we obtain:

$$K_P e = 0$$

Potential energy of the robot arm: $$P = \sum_{i=1}^{n} P_i = \sum_{i=1}^{n} \left[-m_i\, \boldsymbol{g}^T\left(r_i^0\right)\right] = \sum_{i=1}^{n} \left[-m_i\, \boldsymbol{g}^T\left(\boldsymbol{T}_i^0\, r_i^i\right)\right]$$

The symbolic computation of the gravity terms is implemented in the *g_symbolic_computation.py* file by using *SymPy*:

1. Define the Denavit-Hartenberg parameters, the position of the barycenters with respect to the corresponding DH-frames $r_i^i = [x_i, y_i, z_i, 1]^T$ and the gravity vector $\boldsymbol{g} = [0, 0, g_z, 0]^T$.
2. Derive the Denavit-Hartenberg matrices $\boldsymbol{H}_i^{i-1}$ for $i = 1, \dots, 6$ (the terms equal to zero, $\sin \alpha_i$ and $\cos \alpha_i$ are simplified).
3. Derive $\boldsymbol{H}_i^0$ for $i = 1, \dots, 6$.
4. Derive the gravity terms for each link $i = 1, \dots, 6$:



$$g_i(\theta) = \frac{\partial P}{\partial \theta_i} = \sum_{j=i}^{n} \left[m_j\, \boldsymbol{g}^T\left(\boldsymbol{U}_{ji}\, r_j^j\right)\right]$$

Effects of motion of joint *i* on the points of link *j*: $\boldsymbol{U}_{ji} = \dfrac{\partial \boldsymbol{H}_j^0}{\partial \theta_i} \quad j = 1, \dots, 6$

PD tuning

The tuning has been performed considering a simplified linear system, keeping into account a linearization of the Harmonic Drive dynamics and a simplified model for the robotic arm.

The Harmonic Drive has been linearized for small velocities and displacements, neglecting the static friction. The resultant linear dynamic model has four states and is described by the equations below:

$$\tau_m = J_w \ddot{q}_w + B_{mw} \dot{q_w} + \frac{1}{100}\left(k_1\left(\frac{q_w}{100} - q_f\right)\right)$$

$$0 = J_f \ddot{q}_f + B_f \dot{q}_f + \left(k_1\left(-\frac{q_w}{100} + q_f\right)\right)$$

- J_w: Wave generator Inertia
- J_f: Flex spline Inertia, that correspond to the i-th element in the $\bar{M}$ diagonal matrix
- B_{mw}: Wave generator dumping term
- B_f: Flex spline dumping term
- k_1: Flex spline stiffness
- q_f: Flex spline position, that correspond to the i-th joint position
- q_w: Wave generator position

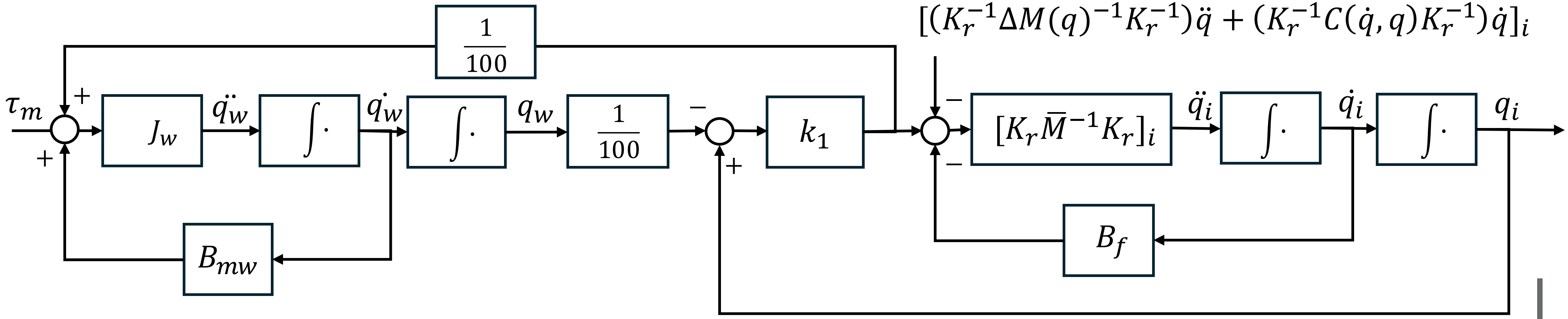
PD tuning

Considering only the robot dynamics, thanks to the reduction ratio, low accelerations, and low velocities, the coupling terms and the non-linearities are reduced and can be considered as disturbances.

Removing the non-linear terms, each link is uncoupled and the mass matrix becomes diagonal and constant.

Adding the dynamics of the Harmonic Drive, we obtain the following scheme for each link. Notice that the flex spline is rigidly attached to the link, so the flex spline inertia corresponds to the link inertia.

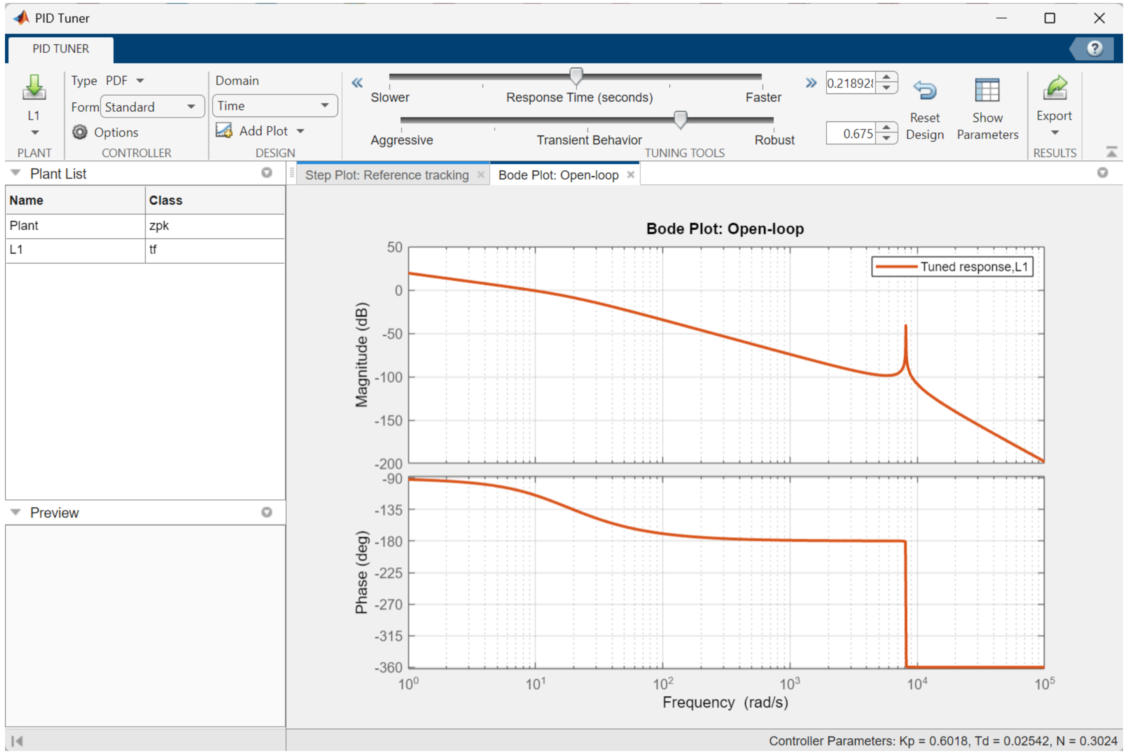
- $\bar{M}$ is a diagonal constant matrix ($M(q) = \bar{M} + \Delta M(q)$)
- K_r is the matrix with the reduction ratio in the diagonal



PD tuning

The tuning is performed in the frequency domain, the transfer function for each link is computed with Matlab starting from the state-space representation. The input is the torque τ_m applied to the wave generator, and the output is the position of the flex spline q_f, which is also the position of the joint.

For each link, we have used the PID Tuner Matlab App to tune the PD controllers and achieve fast enough responses.

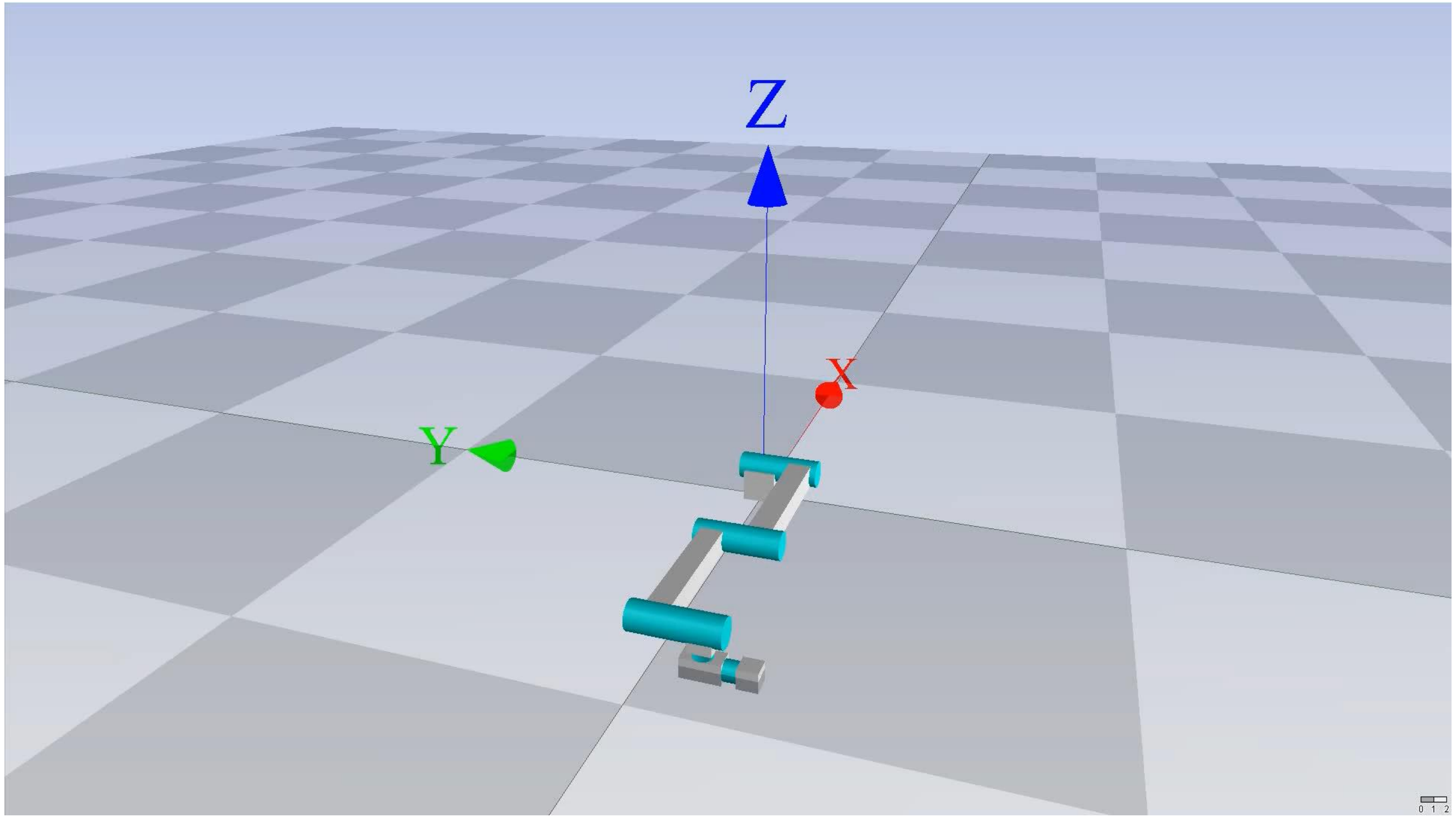


$$\frac{1.635\,e^{10}}{s^4 + 18.48s^3 + 6.608e^7 s^2 + 1.046e^9 s - 0.0008531}$$

Example: first joint transfer function and bode diagram

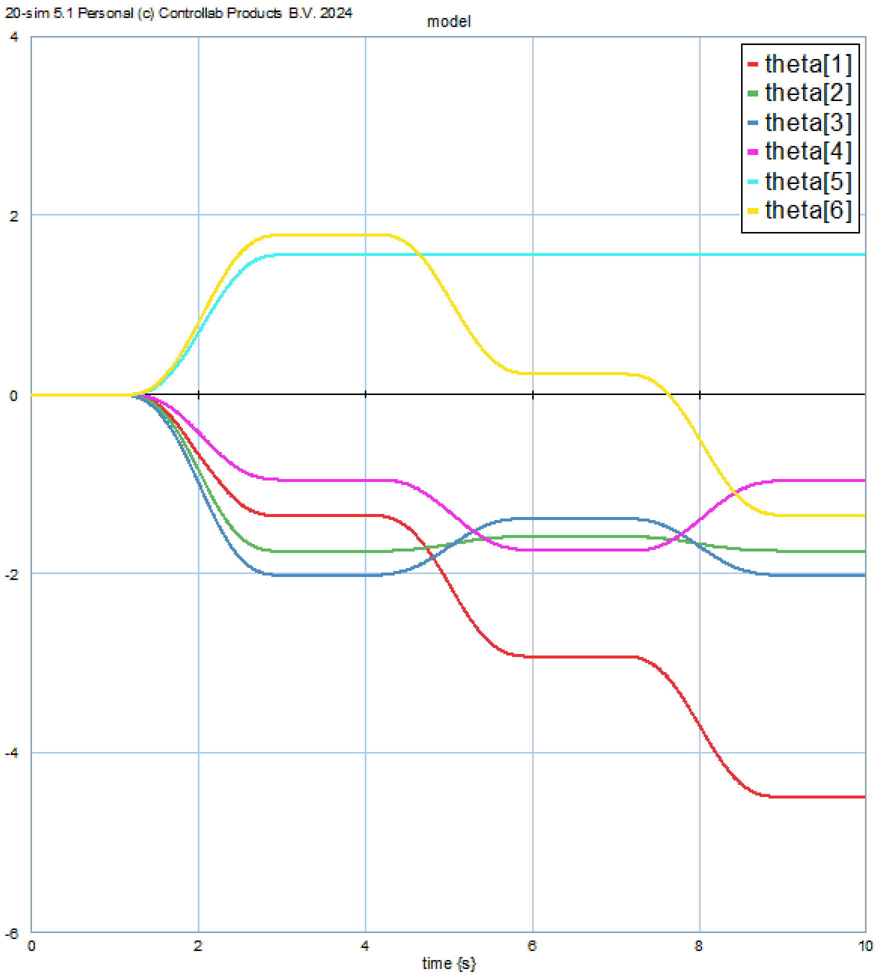
For simulation purpose, we have desiged a three points trajectory for a pick and place task.

The transition between the points follows cycloidal trajectory and the points are interpolated in the joint space.

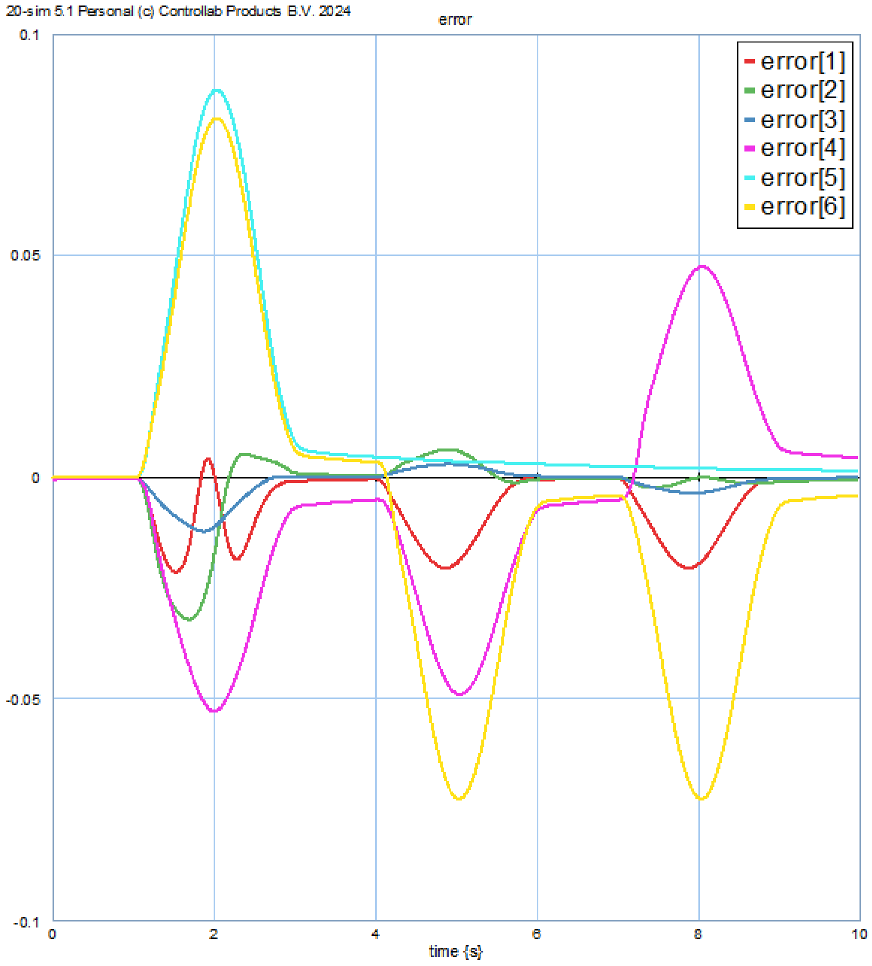


Simulation Results

Joint positions:



Error:



Conclusions

From the simulation results, we can conclude that the robot model behaves consistently with the real physical system. Gravity compensation is accurately achieved, and the PD controller demonstrates performance close to the expected behavior.

For more comprehensive model validation, it would be beneficial to compare these simulation results with experimental data obtained from a real UR5 robot.

Overall, this model provides a reliable and effective approximation for control design and simulation purposes, laying a solid foundation for further development and testing

Bibliography

- Macchelli, A. (n.d.). *Modelling and Simulation of Mechatronic Systems M* [Slide]. Department of Electrical, Electronic, and Information Engineering (DEI), University of Bologna. Disponibile su: http://www.unibo.it, Email: alessandro.macchelli@unibo.it

- Melchiorri, C. (n.d.). *Industrial Robotics* [Slide]. Dipartimento di Ingegneria dell'Energia Elettrica e dell'Informazione (DEI), Università di Bologna. Disponibile su: http://www.unibo.it, Email: claudio.melchiorri@unibo.it