ROBOTICS
KUKA LWR 4+ - Dynamics analysis

Gonçalo Firme*, Rafael Santos†, Pedro Serradas‡
ISTID: {103319*, 103409†, 100065‡}
Emails: {goncalofirme*, rafael.lima.santos†, pedro.serradas ‡}@tecnico.ulisboa.pt

***Abstract*—This paper presents a dynamic analysis of the KUKA LWR 4+ robotic manipulator, extending a prior study on its kinematic properties. The KUKA LWR 4+ is a lightweight, high-precision robot commonly employed in industrial and research environments. While kinematic analysis offers insights into motion planning, dynamic analysis is crucial for tasks involving external forces, interaction control, and trajectory optimization.**

Using the Newton-Euler Formulation, the equations of motion for the manipulator are derived. Two control strategies, a decentralized PID controller and a centralized inverse dynamics controller, are implemented and evaluated. Extensive MATLAB simulations validate the proposed dynamic model and demonstrate the controllers' effectiveness under varying dynamic conditions. Comparative results highlight the strengths and limitations of each approach, providing valuable insights for future work on advanced control strategies.

***Index Terms*—KUKA LWR 4+, Robot Dynamics, Newton-Euler Formulation, Decentralized PID Control, Centralized Inverse Dynamics, Trajectory Optimization, Nonlinear Systems, MATLAB Simulations**

Dynamic Parameters of the KUKA LWR 4+

The dynamic analysis of the KUKA LWR 4+ requires the identification of key parameters, such as the masses, inertia tensors and distances between links and their centers of mass. These parameters are essential for accurately deriving the dynamic equations of motion.

Mass of the Augmented Links

The mass of each augmented link, which combines the mass of link i and the motor of link $i+1$, is presented below:

Augmented Link i	Mass (kg)
1	2.7782
2	2.7782
3	2.7782
4	2.7782
5	1.8789
6	1.2172
7	0.2373
Total	14.4462

Fig. 1. Mass of the augmented links.

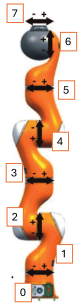

Fig. 2. Robot links' numeration

Inertia Tensors of the Augmented Links

The inertia tensors for each augmented link, represented in units of $\mathrm{kg}\cdot\mathrm{m}^2$, are symmetric and given as follows:

Augmented Link i	Inertia Tensor ($\mathrm{kg}\cdot\mathrm{m}^2$)
$\bar{I}_1$	$\begin{bmatrix} 0.03935326 & 0.00000009 & 0.00000099 \\ 0.00000009 & 0.00879396 & -0.00152896 \\ 0.00000099 & -0.00152896 & 0.03608228 \end{bmatrix}$
$\bar{I}_2$	$\begin{bmatrix} 0.03977837 & 0.00000099 & -0.00000009 \\ 0.00000099 & 0.00879182 & -0.00152896 \\ -0.00000009 & -0.00152896 & 0.03650854 \end{bmatrix}$
$\bar{I}_3$	$\begin{bmatrix} 0.03935426 & 0.00000009 & -0.00000099 \\ 0.00000009 & 0.00879396 & -0.00152896 \\ -0.00000099 & -0.00152896 & 0.03608228 \end{bmatrix}$
$\bar{I}_4$	$\begin{bmatrix} 0.03977837 & 0.00000099 & -0.00000009 \\ 0.00000099 & 0.00879182 & 0.00152896 \\ -0.00000009 & 0.00152896 & 0.03650854 \end{bmatrix}$
$\bar{I}_5$	$\begin{bmatrix} 0.03074898 & 0.00000017 & 0.00000131 \\ 0.00000017 & 0.00514043 & 0.00192026 \\ 0.00000131 & 0.00192026 & 0.02855813 \end{bmatrix}$
$\bar{I}_6$	$\begin{bmatrix} 0.00226285 & 0.00000000 & 0.00000000 \\ 0.00000000 & 0.00266953 & -0.00000036 \\ 0.00000000 & -0.00000036 & 0.00251723 \end{bmatrix}$
$\bar{I}_7$	$\begin{bmatrix} 0.00014719 & 0.00000000 & 0.00000000 \\ 0.00000000 & 0.00014720 & 0.00000000 \\ 0.00000000 & 0.00000000 & 0.00015261 \end{bmatrix}$

TABLE I
Inertia tensors of the augmented links.

Vectors from Link Origins to Adjacent Links and Centers of Mass

The vectors from the origin of link $i-1$ to the origin of link i ($\mathbf{r}_{i-1,i}$) and from the origin of link $i-1$ to the center of mass of link i ($\mathbf{r}_{i,C_i}$) are given in meters in Table II (next page):

Moment of Inertia of the Rotors

The moment of inertia of the rotor for each joint is calculated using the following formula:

$$I_m = \frac{1}{2} m_m r^2 = \frac{1}{2} \times 0.0106 \times 0.05^2 = 1.325 \times 10^{-7}\ \mathrm{kg}\cdot\mathrm{m}^2$$

Where m_m is the mass of the motor and r is its radius.

TABLE II
VECTORS FROM LINK ORIGINS TO ADJACENT LINKS AND CENTERS OF MASS.

Augmented Link i	$\mathbf{r}_{i-1,i}$ (m)	$\mathbf{r}_{i,C_i}$ (m)
1	$\begin{bmatrix}0\\0\\0\end{bmatrix}$	$\begin{bmatrix}0\\-0.07653715\\-0.02714915\end{bmatrix}$
2	$\begin{bmatrix}0\\-0.4\\0\end{bmatrix}$	$\begin{bmatrix}0\\-0.02714915\\0.07730044\end{bmatrix}$
3	$\begin{bmatrix}0\\0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0.07653715\\-0.02714915\end{bmatrix}$
4	$\begin{bmatrix}0\\0.39\\0\end{bmatrix}$	$\begin{bmatrix}0\\0.02714915\\0.07730044\end{bmatrix}$
5	$\begin{bmatrix}0\\0\\0\end{bmatrix}$	$\begin{bmatrix}1.16\times10^{-6}\\-0.10239196\\-0.004375432\end{bmatrix}$
6	$\begin{bmatrix}0\\0\\0.078\end{bmatrix}$	$\begin{bmatrix}0\\-0.00815658\\-0.00437543\end{bmatrix}$
7	$\begin{bmatrix}0\\0\\-0.01476472\end{bmatrix}$	–

I. DYNAMIC MODEL

A dynamic model of the KUKA LWR 4+ robotic manipulator is essential for simulating the robot's motion, designing controllers, and aiding in prototype development. Two common methods for deriving the equations of motion are the Lagrange and Newton-Euler formulations. The Newton-Euler formulation, known for its computational efficiency, was selected for this task. This approach involves balancing forces acting on individual links through forward and backward recursions to determine joint torques. The following sections detail the steps involved in applying the Newton-Euler method.

The Newton-Euler method proceeds in two phases: forward recursion, which computes velocities and accelerations, and backward recursion, which calculates forces, moments, and joint torques.

Forward Recursion: Velocity and Acceleration Calculation

In the forward recursion, the angular velocity ($\boldsymbol{\omega}_i^i$) and angular acceleration ($\dot{\boldsymbol{\omega}}i^i$) of each link are computed, followed by the linear acceleration of the link origins ($\boldsymbol{p}i^i$) and the linear acceleration of their centers of mass ($\boldsymbol{p}{C_i}^i$). Additionally, the rotor's angular acceleration ($\boldsymbol{\omega}_{m_i}^{i-1}$) is obtained using the following equations:

$$
\begin{aligned}
\boldsymbol{\omega}_i^i &= \mathbf{R}_i^{i-1\,T}\left(\boldsymbol{\omega}_{i-1}^{i-1} + \dot{q}_i\mathbf{z}_0\right)\\
\dot{\boldsymbol{\omega}}_i^i &= \mathbf{R}_i^{i-1\,T}\left(\dot{\boldsymbol{\omega}}_{i-1}^{i-1} + \ddot{q}_i\mathbf{z}_0 + \dot{q}_i\boldsymbol{\omega}_{i-1}^{i-1}\times\mathbf{z}_0\right)\\
\ddot{\boldsymbol{p}}_i^i &= \mathbf{R}_i^{i-1\,T}\ddot{\boldsymbol{p}}_{i-1}^{i-1} + \dot{\boldsymbol{\omega}}_i^i\times\mathbf{r}_{i-1,i}^i + \boldsymbol{\omega}_i^i\times\left(\boldsymbol{\omega}_i^i\times\mathbf{r}_{i-1,i}^i\right)\\
\ddot{\boldsymbol{p}}_{C_i}^i &= \ddot{\boldsymbol{p}}_i^i + \dot{\boldsymbol{\omega}}_i^i\times\mathbf{r}_{i,C_i}^i + \boldsymbol{\omega}_i^i\times\left(\boldsymbol{\omega}_i^i\times\mathbf{r}_{i,C_i}^i\right)\\
\dot{\boldsymbol{\omega}}_{m_i}^{i-1} &= \dot{\boldsymbol{\omega}}_{i-1}^{i-1} + k_{r_i}\ddot{q}_i\mathbf{z}_{m_i}^{i-1} + k_{r_i}\dot{q}_i\boldsymbol{\omega}_{i-1}^{i-1}\times\mathbf{z}_{m_i}^{i-1}
\end{aligned}
$$

Where:

- $\mathbf{z}_0 = \begin{bmatrix}0 & 0 & 1\end{bmatrix}^T$ is the unit vector along the z-axis.
- k_{r_i} is the gear ratio of the motor (≈ 110).
- $\mathbf{R}_i^{i-1,T}$ is the rotation matrix from link $i-1$ to link i.
- $\dot{q}_i$ and $\ddot{q}_i$ are the joint velocity and acceleration, respectively.
- $\mathbf{r}_{i-1,i}$ is the vector from the origin of frame $i-1$ to the origin of frame i.
- $\mathbf{r}_{i,C_i}$ is the vector from the origin of frame i to the center of mass of link i.

Backward Recursion: Force and Torque Calculation

In the backward recursion, the forces and moments are propagated from the end-effector (last link) back to the base of the manipulator. The purpose is to compute the joint torques required to sustain the motion and compensate for gravitational and inertial forces.

The forces ($\mathbf{f}_i^i$) and moments ($\boldsymbol{\mu}_i^i$) acting on each link are computed using the following equations:

$$
\begin{aligned}
\mathbf{f}_i^i &= \mathbf{R}_{i+1}^i\mathbf{f}_{i+1}^{i+1} + m_i\ddot{\boldsymbol{p}}_{C_i}^i\\
\mu_i^i &= -\boldsymbol{f}_i^i\times\left(\boldsymbol{r}_{i-1,i}^i + \boldsymbol{r}_{i,C_i}^i\right) + R_{i+1}^i\mu_{i+1}^{i+1}\\
&\quad + R_{i+1}^i\boldsymbol{f}_{i+1}^{i+1}\times\boldsymbol{r}_{i,C_i}^i + \bar{I}_i^i\boldsymbol{w}_i^i\\
&\quad + \boldsymbol{w}_i^i\times\left(\bar{I}_i^i\boldsymbol{w}_i^i\right) + k_{r,i+1}\ddot{q}_{i+1}I_{m_{i+1}}\boldsymbol{z}_{m_{i+1}}^i\\
&\quad + k_{r,i+1}\dot{q}_{i+1}I_{m_{i+1}}\boldsymbol{w}_i^i\times\boldsymbol{z}_{m_{i+1}}^i
\end{aligned}
$$

The joint torque (τ_i) is obtained by projecting the total moment onto the joint axis and adding the rotor inertia contribution:

$$
\tau_i = \boldsymbol{\mu}_i^i\mathbf{R}_i^{i-1\,T}\mathbf{z}_0 + k_{r_i}\mathbf{I}_{m_i}\dot{\boldsymbol{\omega}}_{m_i}^{i-1\,T}\mathbf{z}_{m_i}^{i-1}
$$

Where:

- $\mathbf{I}_i^i$ and $\mathbf{I}_{m_i}$ are the inertia tensors of the link and rotor, respectively.
- m_i is the mass of link i.
- $\mathbf{R}_{i+1}^i$ is the rotation matrix from link $i+1$ to link i.

Initialization

The forward and backward recursions are initialized with the following conditions:

$$
\boldsymbol{\omega}_0^0 = \dot{\boldsymbol{\omega}}_0^0 = \begin{bmatrix}0 & 0 & 0\end{bmatrix}^T, \quad \ddot{\boldsymbol{p}}_0^0 - \mathbf{g}_0^0 = \begin{bmatrix}0 & 0 & g\end{bmatrix}^T
$$

$$
\mathbf{f}_8^8 = \boldsymbol{\mu}_8^8 = \begin{bmatrix}0 & 0 & 0\end{bmatrix}^T
$$

Where g represents the gravitational acceleration. The forces and moments at the last link are assumed to be zero, as no external loads are applied on the end-effector.

SIMULINK MODEL

The dynamic model was generated using the same approach as the kinematic model, utilizing a C-code generator and symbolic expressions. In this case, the symbolic expressions were for the joints' velocities $\dot{q}_i$ and accelerations $\ddot{q}_i$. The torque values were obtained using code that considered the constants, equations, and initial and final conditions presented earlier.

The equation of motion for a robotic manipulator describes the relationship between the joint torques ($\boldsymbol{\tau}$) and the joint positions ($\mathbf{q}$), velocities ($\dot{\mathbf{q}}$), and accelerations ($\ddot{\mathbf{q}}$). This equation is expressed as:

$$\boldsymbol{\tau} = \mathbf{B}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q})$$

Where:

- $\mathbf{B}(\mathbf{q})$ is the 7×7 mass matrix, representing the inertia of the system,
- $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ is the 7×1 Coriolis and centrifugal vector,
- $\mathbf{g}(\mathbf{q})$ is the 7×1 gravity vector.

To obtain $\mathbf{g}(\mathbf{q})$, set the joint velocities and accelerations to zero:

$$\boldsymbol{\tau}(\dot{\mathbf{q}} = 0, \ddot{\mathbf{q}} = 0) = \mathbf{g}(\mathbf{q})$$

To obtain the mass matrix $\mathbf{B}(\mathbf{q})$, first consider null velocities:

$$\boldsymbol{\tau}(\dot{\mathbf{q}} = 0) = \mathbf{B}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{g}(\mathbf{q})$$

Then, the mass matrix $\mathbf{B}(\mathbf{q})$ is generated through partial differentiation:

$$B_{ij}(\mathbf{q}) = \frac{d\left(\tau(\dot{\mathbf{q}} = 0) - \mathbf{g}(\mathbf{q})\right)_i}{d\ddot{q}_j}$$

The third required output is the Coriolis and centrifugal vector $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$, which is obtained using null accelerations:

$$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})(\dot{q} = 0) = \boldsymbol{\tau}(\ddot{\mathbf{q}} = 0) - \mathbf{g}(\mathbf{q})$$

The final Simulink dynamic block, shown in Figure 3 has joints' positions and velocities as inputs, and $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{g}$ as outputs:

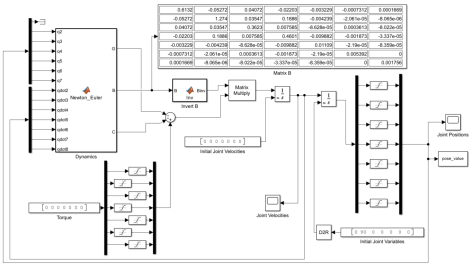

Fig. 3. Simulink dynamic model.

As Figure 3 suggests, the matrix $\mathbf{B}$ is symmetric, as expected. To understand and verify the behavior of the manipulator, it was decided to run a solution with null torque and initial joint velocities. The arm aligned with $\mathbf{z}_0 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$. The integrators provide the actual position and velocity of the joints, which serve as inputs to the dynamic block itself. Additionally, angular position limits and maximum allowable torques were inserted, as shown in Table III:

TABLE III
MAXIMUM AND MINIMUM ANGULAR POSITIONS AND MAXIMUM ALLOWABLE TORQUES.

Link	Range of Motion (°)	Maximum Torque (Nm)
1	±170	176
2	−30/ + 210	176
3	±170	100
4	±120	100
5	±170	100
6	±120	38
7	±170	38

WORST-CASE INERTIA CONFIGURATION

The worst-case inertia configuration is used to project controllers' gains. The coefficients of the matrix $B_{ij}(q)$ represent the moment of inertia at Link$_i$ axis, in a specific arm's configuration, when the rest of the joints are blocked. By the maximization of the matrix $B_{ij}(q)$, it is possible to obtain the desired values. For each link, the inertia value to extract from the matrix is the coefficient b_{ii}. Through an intuitive and iterative process, it was possible to understand the joints' configurations that most hinder the movement for each joint being studied, as presented in the following table:

TABLE IV
WORST-CASE INERTIA CONFIGURATION

Link$_i$	q_1	q_2	q_3	q_4	q_5	q_6	q_7	$b_{\max,ii}$(kg m^2)
1	–	0	0	–	0	–	–	2.924
2	–	±90	0	0	–	0	–	2.908
3	–	–	0	0	–	±90	–	0.4903
4	–	–	±90	0	–	±90	–	0.4742
5	–	–	–	±90	–	–	–	0.01127
6	–	–	–	–	±90	–	–	0.005392
7	–	–	–	–	–	–	–	0.001756

TASK TRAJECTORY

To achieve our personal goal of performing a trajectory, it was decided to design a path that traces the letters "IST", representing the name of the institution. This trajectory, to be done in 14 s, is composed of straight lines and semicircles, ensuring a balanced and smooth motion across all joints.

The Cartesian coordinates of the points to follow are:

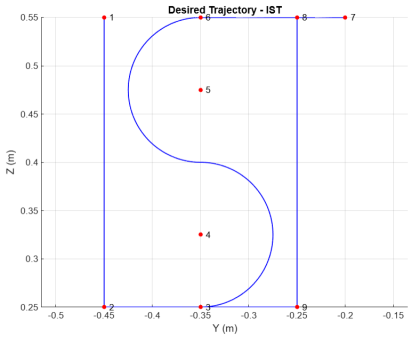

Fig. 4. Trajectory to be implemented

TABLE V
COORDINATES OF TRAJECTORY POINTS

Point	X (m)	Y (m)	Z (m)
1	-0.05	-0.45	0.55
2	-0.05	-0.45	0.25
3	-0.05	-0.35	0.25
4	-0.05	-0.35	0.325
5	-0.05	-0.35	0.475
6	-0.05	-0.35	0.55
7	-0.05	-0.20	0.55
8	-0.05	-0.25	0.55
9	-0.05	-0.25	0.25

DECENTRALIZED PID TYPE JOINT CONTROLLER

A decentralized control strategy was chosen due to its simplicity and ease of implementation. In this approach, the manipulator is treated as i independent systems, where i is the number of joints. Each joint is controlled independently as a SISO (Single Input Single Output) mechanism. This approach reduces the complexity of the control design, making it easier to tune and implement controllers for manipulators like the KUKA LWR 4+. However, it does not account for dynamic coupling between joints, which may lead to performance limitations during coordinated movements.

A proportional-derivative (PD) action is implemented, which has the capacity to predict the future error of the system's response, improving performance. Considering a gravity compensation factor, the control law is expressed as:

$$\tau(q) = B(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = g(q) + K_P\tilde{q} - K_D\dot{q}$$

Where:

- K_P is proportional gain;
- K_D is the derivative gain;
- $\tilde{q} = (q_d - q)$ is the error between the desired and actual position.

The term $g(q)$ represents gravity compensation, ensuring that the controller only needs to regulate the tracking error and velocity. This simplifies the control problem and improves performance during complex trajectories.

Manipulating the dynamic system's equation using the Laplace transform:

$$B(q)\ddot{q} + \big(C(q,\dot{q}) + K_D\big)\dot{q} = K_P(q_d - q) \iff$$
$$\iff \big(Bs^2 + (C + K_D)s + K_P\big)q(s) = K_P q_d(s)$$

$$G(s) = \frac{q(s)}{q_d(s)} = \frac{K_P}{Bs^2 + (C + K_D)s + K_P}$$

The closed-loop transfer function obtained is of second order:

$$G(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}$$

Neglecting the nonlinear term C for this effect:

$$K_D = 2\xi\omega_n \cdot B,$$
$$K_P = \omega_n^2 \cdot B.$$

The values of inertia used for each joint are calculated based on the worst-case inertia configuration to ensure correct robot performance. Gains are calculated by tuning the parameters:

- $\xi = 1$ –Critical damping is chosen to return the system to steady-state as quickly as possible while avoiding oscillations;
- $\omega_n = 50\,\text{rad/s}$ –Determined through trial and error to balance a fast response time with acceptable position error, joint velocities, and torque limits.

The proportional and derivative gains for each joint, considering the worst-case inertia values, are shown in Table VI:

TABLE VI
PROPORTIONAL AND DERIVATIVE GAINS FOR EACH JOINT.

Joint	K_P	K_D
1	7310	292.4
2	7270	290.8
3	1225.7	49.03
4	1185.5	47.42
5	28.2	1.127
6	13.5	4.4
7	0.5392	0.1756

The complete Simulink-designed model is displayed Figure 5.

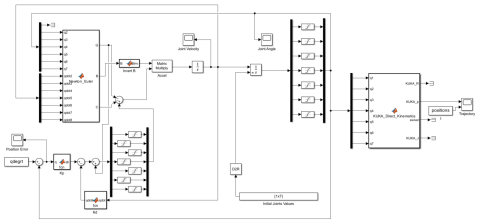

Fig. 5. Simulink model of the Decentralized control strategy.

For the same trajectory, the maximum, minimum, and average position errors for each joint were calculated (Table VII):

TABLE VII
POSITION ERROR FOR DECENTRALIZED CONTROL (MM).

Joint	Avg. Pos. Error (mm)	Max. Pos. Error (mm)	Min. Pos. Error (mm)
1	4.6	11.4	-10.7
2	6.1	17.0	-20.4
3	1.8	8.8	-7.3
4	11.4	30.1	-33.8
5	1.4	5.7	-5.4
6	5.5	13.4	-13.5
7	5.0	12.5	-14.9

Analyzing the graphical representation (Figure 6), joints 2 and 4 exhibited the highest maximum errors, reaching values of 0.017 m and 0.0301 m, respectively. This can be attributed to their significant involvement in the semi-circular movements of the trajectory, which require precise coordination in multiple Cartesian directions. To validate the proposed path, we verified that both torque and velocity limits (Tab. III were respected (due to space constraints, detailed results are not presented).

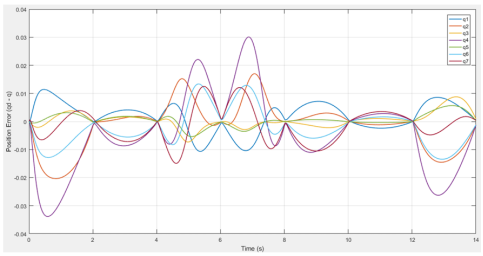

Fig. 6. Position error for decentralized control.

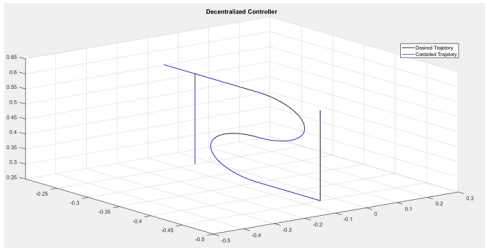

Fig. 7. Trajectory obtained with Decentralized Controller

CENTRALIZED INVERSE DYNAMICS CONTROLLER

A centralized control strategy was chosen to improve the robot's performance by compensating for the nonlinear dynamics of the manipulator. Unlike decentralized control, which treats each joint independently, the centralized controller accounts for the dynamic coupling between joints, ensuring better coordination and precision during high-speed movements. This approach uses an inverse dynamics model to linearize the system globally, enabling the application of well-known linear control techniques.

The inverse dynamics of the manipulator is calculated and compensated. For higher operational speeds, the nonlinear term has a significant influence on the behavior of the system. It is now considered a nonlinear MIMO plant:

$$B(q)\ddot{q} + n(q,\dot{q}) = \tau$$

Where:

- $B(q)$ is the mass matrix, describing the manipulator's inertia;
- $n(q,\dot{q}) = C(q,\dot{q}) + g(q)$ is the Vector of nonlinear forces, including Coriolis, centrifugal, and gravitational forces;
- τ is vector of joint torques.

A new input vector y is defined so that it converts the system to a linear MIMO, resulting in a global linearization by using a nonlinear state feedback:

$$\tau = B(q)y + n(q,\dot{q})$$

Where:

- $\ddot{q} = y$ is possible since $B(q)$ is invertible.

The control law that stabilizes the system is given by:

$$y = -K_P - K_D\dot{q} + r$$

By choosing K_P and K_D as diagonal matrices, it is possible to obtain the decoupled system:

$$K_P = \text{diag}\{\omega_{n1}^2, \omega_{n2}^2, \omega_{n3}^2, \omega_{n4}^2, \omega_{n5}^2, \omega_{n6}^2, \omega_{n7}^2\}$$
$$K_D = \text{diag}\{2\xi_1\omega_{n1}, 2\xi_2\omega_{n2}, 2\xi_3\omega_{n3}, 2\xi_4\omega_{n4}, 2\xi_5\omega_{n5}, 2\xi_6\omega_{n6}, 2\xi_7\omega_{n7}\}$$

Having q_d as the trajectory to track, the trajectory law is:

$$r = \ddot{q}_d + K_D\dot{q}_d + K_P q_d$$

It was decided to define $\ddot{q}_d = 0$, since it is not desired to control acceleration. Finally, the error is described by:

$$\ddot{\tilde{q}} + K_D\dot{\tilde{q}} + K_P\tilde{q} = 0$$

For the project's effect, it was decided to use the same values of natural frequency ω_n and damping ratio ξ:

$$K_P = \text{diag}\{2500, 2500, 2500, 2500, 2500, 2500, 2500\}$$
$$K_D = \text{diag}\{100, 100, 100, 100, 100, 100, 100\}$$

The complete Simulink model can be seen in Figure 8.

In terms of results, both position and velocity errors were obtained. Table VIII presents the average, maximum, and minimum position and velocity errors for each joint.

As expected, joints 2 and 4 exhibited higher position errors. Regarding velocity, these two joints, along with joint 3, showed the highest errors for the proposed trajectory.

Graphically, this is confirmed in Figure 9, where the peak velocity error occurs at the beginning of the process, marking

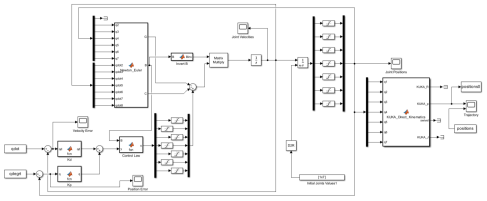

Fig. 8. Simulink model of the centralized control strategy.

TABLE VIII
POSITION AND VELOCITY ERRORS FOR DECENTRALIZED CONTROL (MM AND MM/S).

Joint	Avg. Pos. Error (mm)	Max. Pos. Error (mm)	Min. Pos. Error (mm)	Avg. Vel. Error (mm/s)	Max. Vel. Error (mm/s)	Min. Vel. Error (mm/s)
1	0.4	0.3	-1.8	1.8	36.2	-10.0
2	9.6	10.5	0.0	5.1	-25.6	2.8
3	1.9	6.5	-3.2	148.3	-7.9	3.2
4	11.4	15.0	0.0	118.9	-21.0	5.7
5	5.2	0.0	-7.0	214.7	-14.7	2.0
6	1.5	3.2	-0.3	1.7	-100.4	1.8
7	0.2	0.5	-0.3	2.0	8.1	-6.5

initialization and a more abrupt movement of the robot. As expected, the mechanical limits were respected, and the trajectory tracking showed good results, as can be seen in the plots presented in Figure 10.

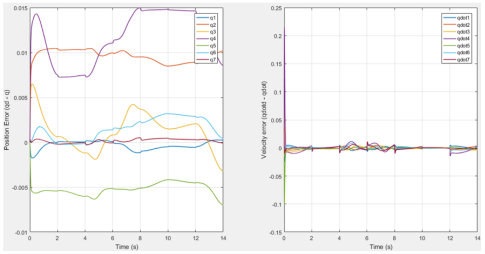

Fig. 9. Position and velocity errors for centralized control.

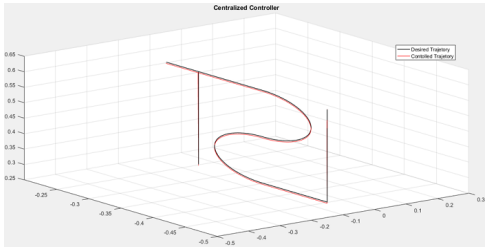

Fig. 10. Trajectory obtained with Centralized Controller

Conclusion

This work has thoroughly addressed the dynamic modeling and control of the KUKA LWR 4+ manipulator. A detailed Newton-Euler-based model was derived by incorporating accurate physical parameters such as link masses, inertia tensors, and center-of-mass offsets. The model was validated in simulation, demonstrating consistency with expected robot behaviors under various joint configurations and motions.

Building on this model, two distinct control strategies were implemented and evaluated: a Decentralized PID controller and a Centralized Inverse Dynamics controller. The Decentralized PID approach relies on treating each joint as an independent system, simplifying gain tuning and controller design. This scheme is advantageous for tasks with moderate accuracy or lower-speed requirements. However, as it neglects the coupling effects between joints, it can result in larger tracking errors, particularly when multiple joints move simultaneously at higher speeds or when the trajectory involves complex, multi-axis motions.

On the other hand, the Centralized Inverse Dynamics controller explicitly compensates for the manipulator's nonlinear inter-joint coupling and gravitational terms, ensuring higher tracking accuracy and smoother motion, even during rapid or coordinated joint movements. While more computationally demanding and requiring a precise dynamic model, its simulation results consistently showed reduced position and velocity errors compared to the decentralized scheme. Moreover, both controllers respected the robot's physical limitations, maintaining joint torques and velocities within allowable limits.

In summary, for applications where simplicity of implementation and moderate performance suffice—such as routine pick-and-place or low-speed operations—the Decentralized PID control proves both practical and reliable. However, when tighter precision, higher speeds, or more complex trajectories are required—such as in advanced research or delicate assembly tasks—the Centralized Inverse Dynamics controller is preferable. Future enhancements could incorporate friction compensation, robust or adaptive control methods, and comprehensive disturbance rejection strategies. These additions would further improve the accuracy, robustness, and overall performance of the KUKA LWR 4+ in dynamic and uncertain environments. disturbances.

References

[0] – Zahi, A., Kuka LWR 4+ Arm [1] – Siciliano, B., Sciavicco, L., Oriolo, G. Robotics – Modelling, Planning and Control [2] - Leal, A., Marreiros, B. KUKA LWR 4+ – Kinematic analysis [3] – KUKA Roboter GmbH, Lightweight Robot 4+ - Specification [4] – McGuire, A. Control of Kinematically Redundant Robotic Manipulators for Orthopedic Surgery:

Work do by:

Dynamic Model - All members; Simulink Dynamic Model - Pedro Serradas;
Simulink Decentralized PID type joint controller - Gonçalo Firme;
Simulink Centralized Inverse Dynamics Controller - Rafael Lima Santos.