Robotics 2 project

Robust tracking control based on bounds on dynamic coefficients

Paradiso Emiliano, 1940454

31 August 2024

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

Introduction

In modern robotics, ensuring accurate and reliable performance in presence of uncertainties is crucial. Robust control offers a powerful solution, particularly when system dynamics are influenced by moderate variations due to uncertainties known to lie within bounded intervals.

The aim of this presentation is to explore a robust control strategy that relies on known bounds for dynamic coefficients variations.

This approach is especially useful for systems where disturbances or parameter variations cannot be ignored, offering enhanced stability and performance in realistic scenarios compared to traditional feedback linearization methods.

Robust control

- Unlike previous classical theories that required complex way to extract bounds for the reference trajectory, the manipulator state vector and some boundness in norm of the estimated inertia matrix from the actual one, our work is based on M. Spong's research. This allows the derivation of a robust control law using only dynamic coefficients, simplifying the calculation of the bounds.

- The designed robust control requires only the linear parameterizability of robot dynamics and the skew-symmetry property:

$$M(q)\ddot{q} + S(q,\dot{q})\dot{q} + g(q) = Y(q,\dot{q},\ddot{q})\theta = u$$

$$\dot{M} - 2S \; skew \; symm$$

- Where $Y(q,\dot{q},\ddot{q})$ is the $N \times p$ regressor matrix and $\theta \in \mathbb{R}^p$ is the minimal set of dynamic coefficients.

Robust control

- Supposing to have uncertainties only on the dynamic coefficients and they are bounded in norm:

$$||\Delta\theta|| = ||\theta - \theta_0|| \leq \rho$$

- θ is the actual parameter vector of the system, it is affected by uncertainty.
- $\theta_0 \in \mathbb{R}^p$ is the nominal parameter vector, it is an estimation of θ.
- $\rho \in \mathbb{R}^+$ is the upper bound on the norm of the uncertainty.

- With this in mind we can design a Robust control law composed by a 'nominal' control vector u_0 and an added continous robust term δ:

$$u_0 = M_0(q)a + S_0(q,\dot{q})v + g_0(q) - Kr = Y(q,\dot{q},v,a)\theta_0 - Kr$$

$$u = u_0 + Y(q,\dot{q},v,a)\delta = Y(q,\dot{q},v,a)(\theta_0 + \delta) - Kr$$

- With the quantites v,a and r, given by:

$$v = \dot{q}^d - \Lambda\bar{q};\;\; a = \dot{v};\;\; r = \dot{\bar{q}} + \Lambda\bar{q};\;\; \bar{q} = q - q^d$$

Robust control

- q^d a twice continously differentiable reference trajectory and K and Λ diagonal gain matrices.
- In this law θ_0 is not updated iteratively as in an adaptive control law, but is is defined in terms of fixed parameters (we avoid the problem of parameter drift), the price we pay is that an *a priori* bound on the parametric uncertainty $\Delta\theta$ is required.

- The added term is designed to achieve robustness to the uncertainty as follows:

$$\delta = \begin{cases} -\rho \dfrac{Y^T r}{||Y^T r||} & if \; ||Y^T r|| > \epsilon \\ -\rho \dfrac{Y^T r}{\epsilon} & if \; || \, Y^T r|| \leq \epsilon \end{cases}$$

- With $\epsilon > 0$, it can be demonstrated that the closed loop equation:

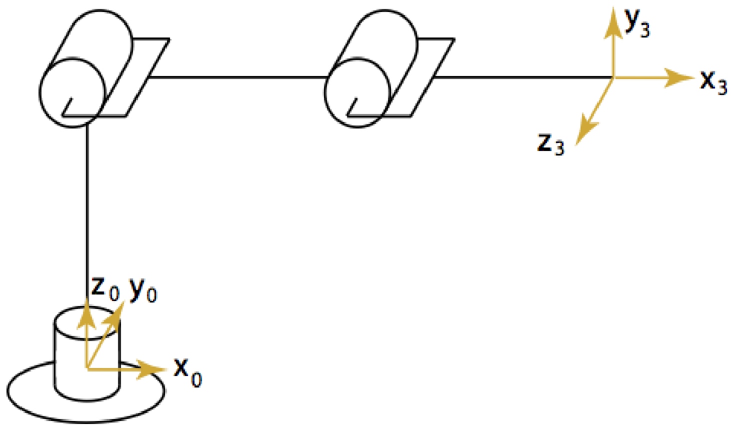
$$M(q)\dot{r} + S(q,\dot{q})r + Kr = Y(q,\dot{q},v,a\,)(\Delta\theta + \delta)$$

- It is uniformly ultimately bounded.

3R Spatial robot

- In our work we derived the Robot dynamic model of a 3R spatial robot with no special assumption on the distribution of link masses (CoM location, link inertia matrices)

$$
{}^i r_{ci} = \begin{pmatrix} {}^i r_{cxi} \\ {}^i r_{cyi} \\ {}^i r_{czi} \end{pmatrix}, \quad {}^i I_i = \begin{pmatrix} {}^i I_{xxi} & {}^i I_{xyi} & {}^i I_{xzi} \\ {}^i I_{xyi} & {}^i I_{yyi} & {}^i I_{yzi} \\ {}^i I_{xzi} & {}^i I_{yzi} & {}^i I_{zzi} \end{pmatrix} \quad \text{for } i = 1, \dots, 3
$$



Joint i	α_i	a_i	d_i	q_i
1	$\frac{\pi}{2}$	0	l_1	q_1
2	0	l_2	0	q_2
3	0	l_3	0	q_3

3R Spatial robot

- Lastly we extracted a linear parametrization of the dynamic model in terms of a minimal set of dynamic coefficients:

$$Y(q,\dot{q},\ddot{q})\theta = u; \quad \theta \in \mathbb{R}^{15}$$

- With the parametrization:

$$\theta_1 = m_1(r_{cx1}^2 + r_{cz1}^2) + I_{yy1} + m_2 r_{cz2}^2 + m_2(r_{cx2} + l_2)^2 + I_{yy2} + m_3 r_{cz3}^2 + m_3(r_{cx3} + l_3)^2 + I_{yy3} + m_3 l_2^2$$

$$\theta_2 = -m_2(r_{cx2} + l_2)^2 + m_2 r_{cy2}^2 - I_{yy2} + I_{xx2} - m_3 l_2^2$$

$$\theta_3 = -2m_2 r_{cy2}(l_2 + r_{cx2}) + 2I_{xy2}$$

$$\theta_4 = m_3 r_{cy3}^2 - m_3(r_{cx3} + l_3)^2 + I_{xx3} - I_{yy3}$$

$$\theta_5 = -2m_3 r_{cy3}(l_3 + r_{cx3}) + 2I_{xy3}$$

$$\theta_6 = -m_3 r_{cy3} l_2$$

$$\theta_7 = m_3(l_3 + r_{cx3}) l_2$$

$$\theta_8 = -m_2 r_{cy2} r_{cz2} + I_{yz2}$$

$$\theta_9 = -m_2(l_2 + r_{cx2}) r_{cz2} + I_{xz2} - m_3 r_{cz3} l_2$$

$$\theta_{10} = -m_3 r_{cy3} r_{cz3} + I_{yz3}$$

$$\theta_{11} = -m_3(l_3 + r_{cx3}) r_{cz3} + I_{xz3}$$

$$\theta_{12} = m_2(r_{cy2}^2 + (l_2 + r_{cx2})^2) + I_{zz2} + m_3(l_2^2 + r_{cy3}^2 + (l_3 + r_{cx3})^2) + I_{zz3}$$

$$\theta_{13} = m_3((l_3 + r_{cx3})^2 + r_{cy3}^2) + I_{zz3}$$

$$\theta_{14} = m_2(l_2 + r_{cx2}) + m_3 l_2$$

$$\theta_{15} = -m_2 r_{cy2}$$

3R Spatial robot

- And the $N \times p$ regressor matrix :

$$y_{11} = \ddot{q}_1$$

$$y_{12} = \ddot{q}_1 s_2^2 + 2\dot{q}_1\dot{q}_2 s_2 c_2$$

$$y_{13} = \ddot{q}_1 c_2 s_2 + \dot{q}_1\dot{q}_2(c_2^2 - s_2^2)$$

$$y_{14} = \ddot{q}_1 s_{23}^2 + 2\dot{q}_1(\dot{q}_2 + \dot{q}_3)s_{23}c_{23}$$

$$y_{15} = \ddot{q}_1 c_{23}s_{23} + \dot{q}_1(\dot{q}_2 + \dot{q}_3)(c_{23}^2 - s_{23}^2)$$

$$y_{16} = 2\ddot{q}_1 c_2 s_{23} + 2\dot{q}_1\dot{q}_2(c_2c_{23} - s_2s_{23}) + 2\dot{q}_1\dot{q}_3 c_2c_{23}$$

$$y_{17} = 2\ddot{q}_1 c_2c_{23} - 2\dot{q}_1\dot{q}_2(s_2c_{23} + c_2s_{23}) - 2\dot{q}_1\dot{q}_3c_2s_{23}$$

$$y_{18} = \ddot{q}_2c_2 - \dot{q}_2^2 s_2$$

$$y_{19} = \ddot{q}_2 s_2 + \dot{q}_2^2 c_2$$

$$y_{110} = (\ddot{q}_2 + \ddot{q}_3)c_{23} - \dot{q}_2^2 s_{23} - \dot{q}_3(2\dot{q}_2 + \dot{q}_3)s_{23}$$

$$y_{111} = (\ddot{q}_2 + \ddot{q}_3)s_{23} + \dot{q}_2^2 c_{23} + \dot{q}_3(2\dot{q}_2 + \dot{q}_3)c_{23}$$

$$y_{22} = -\dot{q}_1^2 s_2c_2$$

$$y_{23} = -\tfrac{1}{2}\dot{q}_1^2(c_2^2 - s_2^2)$$

$$y_{24} = -\dot{q}_1^2 s_{23}c_{23}$$

$$y_{25} = -\tfrac{1}{2}\dot{q}_1^2(c_{23}^2 - s_{23}^2)$$

$$y_{26} = 2\ddot{q}_2s_3 + \ddot{q}_3s_3 - \dot{q}_1^2(c_2c_{23} - s_2s_{23}) + \dot{q}_3(\dot{q}_3 + 2\dot{q}_2)c_3 + \tfrac{g_0}{l_2}s_{23}$$

$$y_{27} = 2\ddot{q}_2c_3 + \ddot{q}_3c_3 + \dot{q}_1^2(s_2c_{23} + c_2s_{23}) - \dot{q}_3(\dot{q}_3 + 2\dot{q}_2)s_3 + \tfrac{g_0}{l_2}c_{23}$$

$$y_{28} = \ddot{q}_1c_2$$

$$y_{29} = \ddot{q}_1 s_2$$

$$y_{210} = \ddot{q}_1c_{23}$$

$$y_{211} = \ddot{q}_1 s_{23}$$

$$y_{212} = \ddot{q}_2$$

$$y_{213} = \ddot{q}_3$$

$$y_{214} = g_0c_2$$

$$y_{215} = g_0s_2$$

$$y_{34} = -\dot{q}_1^2 s_{23}c_{23}$$

$$y_{35} = -\tfrac{1}{2}\dot{q}_1^2(c_{23}^2 - s_{23}^2)$$

$$y_{36} = \ddot{q}_2s_3 - \dot{q}_1^2c_2c_{23} - \dot{q}_2^2c_3 + \tfrac{g_0}{l_2}s_{23}$$

$$y_{37} = \ddot{q}_2c_3 + \dot{q}_1^2c_2s_{23} + \dot{q}_2^2s_3 + \tfrac{g_0}{l_2}c_{23}$$

$$y_{310} = \ddot{q}_1c_{23}$$

$$y_{311} = \ddot{q}_1s_{23}$$

$$y_{313} = \ddot{q}_2 + \ddot{q}_3$$

$$y_{21} = y_{31} = y_{32} = y_{33} = y_{38} = y_{39} = y_{112} = y_{113} =$$
$$y_{114} = y_{115} = y_{312} = y_{314} = y_{315} = 0$$

Simulation

- The derived robust control law can be used to make the 3R Spatial robot execute a trajectory tracking task on periodic joint space trajectory like:

$$q_d(t) = \begin{pmatrix} sin(t) \\ 5cos(t) \\ sin(t) \end{pmatrix} \quad \dot{q}_d(t) = \begin{pmatrix} cos(t) \\ -5sin(t) \\ cos(t) \end{pmatrix} \quad \ddot{q}_d(t) = \begin{pmatrix} -sin(t) \\ -5cos(t) \\ -sin(t) \end{pmatrix}$$

- Providing initial matching conditions: $q_0 = (0,5,0)^T, \dot{q}_0 = (1,0,1)^T$

- We highlight the advantages of our designed controller by comparing it to a classic feedback linearization controller under different operating conditions:

I. In ideal conditions, where no uncertainties or disturbances are present, and the manipulator's behavior perfectly matches the theoretical dynamic model.

II. In the presence of moderate uncertainties that only affect the dynamic coefficients but are within a known, bounded range

Simulation

- From this simulation, we expect both controllers to exhibit similar behavior under ideal conditions. However, significant differences will arise when uncertainties affect the actual dynamic behavior, with the classic feedback linearization controller showing more pronounced deviations due to poor robustness performamces when an incomplete model is provided.

- The Data used for the simulation are:

$Joint_i$	m_i	l_i	r_{ci}	I_i
1	10	1	$\begin{pmatrix} 0.5 \\ -0.5 \\ -0.1 \end{pmatrix}$	$\begin{pmatrix} 0.5 & 0.1 & 0.1 \\ 0.1 & 0.5 & 0.1 \\ 0.1 & 0.1 & 0.5 \end{pmatrix}$
2	10	1	$\begin{pmatrix} -0.5 \\ -0.3 \\ 0.4 \end{pmatrix}$	$\begin{pmatrix} 0.5 & 0.2 & 0.2 \\ 0.2 & 0.5 & 0.2 \\ 0.2 & 0.2 & 0.5 \end{pmatrix}$
3	10	1	$\begin{pmatrix} 0.2 \\ 0.2 \\ 0.2 \end{pmatrix}$	$\begin{pmatrix} 0.5 & 0.2 & 0.2 \\ 0.2 & 0.5 & 0.2 \\ 0.2 & 0.2 & 0.5 \end{pmatrix}$

Simulation

- The ranges in which the uncertainties lies are taken arbitrarily as follows:

$$m_i + \Delta m_i \quad \text{with} \quad 0 \le \Delta m_i \le 10 \quad \text{for } i = 1, ..., 3$$

$$r_{cij} + \Delta r_{cij} \quad \text{with} \quad 0 \le \Delta r_{cij} \le 1 \quad \text{for } i, j = 1, ..., 3$$

$$I_{ij} + \Delta I_{ij} \quad \text{with} \quad 0 \le \Delta I_{ij} \le 1 \quad \text{for } i = 1, ..., 6, \text{ and } j = 1, ..., 3$$

- Provide the following values for the dynamic coefficients not affected by uncertainty:

θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	θ_9	θ_{10}	θ_{11}	θ_{12}	θ_{13}	θ_{14}	θ_{15}
33	-11.6	3.4	-14	-4.4	-2	12	1.4	-3.8	-0.2	-2.2	29.2	15.3	15	3

- And for the nominal parameter vector, taken as composed by the mean values of possible θ_i varying in the range of uncertainty:

θ_{01}	θ_{02}	θ_{03}	θ_{04}	θ_{05}	θ_{06}	θ_{07}	θ_{08}	θ_{09}	θ_{010}	θ_{011}	θ_{012}	θ_{013}	θ_{014}	θ_{015}
164.25	-33.9	-19.1	-41.5	-54.8	-13	28	-9	-34.8	-14.4	-27.4	116.3	71.2	32.5	-5.5

- From this we can compute numerically the upper bound on the norm of the uncertainty, necessary for our controller:

$$||\theta - \theta_0|| \le \rho = 186.8129$$

Simulation

- The simulation lasts $T = 10\ s$, during which we evaluate the tracking trajectory performance.

```
% Integration function for ode15s
function dydt = robot_dynamics_ode(t, y, q_d, dq_d, ddq_d)
    global T
    q = y(1:3);
    dq = y(4:6);

    % Interpolation of the desired values at time t
    q_d_t = interp1(linspace(0, T, length(q_d)), q_d', t, 'linear', 'extrap')';
    dq_d_t = interp1(linspace(0, T, length(dq_d)), dq_d', t, 'linear', 'extrap')';
    ddq_d_t = interp1(linspace(0, T, length(ddq_d)), ddq_d', t, 'linear', 'extrap')';


    %SIMULATION WITH ROBUST CONTROL
    u = robust_control(q, dq, q_d_t, dq_d_t, ddq_d_t);

    %SIMULATION WITH FEEDBACK LINEARIZATION
    %u = feedbacklinearization(q, dq, q_d_t, dq_d_t, ddq_d_t);

    ddq = robot_dynamics(q, dq, u);
    dydt = [dq; ddq];
end
```

- The simulation steps are:

I. Firstly, the commanded torque, given by the control law, is computed based on the current $q, \dot{q}$ and the desired values $q_d, \dot{q}_d, \ddot{q}_d$.

II. Then, the dynamic behavior of the robot is simulated by the function robot_dynamics(q, dq, u) that use the dynamic model and return $\ddot{q}$

III. The integration routine ode15s is called to integrate numerically the differential equations represented by the state vector:

$$\dot{x} = \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} x_2 \\ -M^{-1}(x_1)[c(x_1, x_2) + g(x_1)] \end{pmatrix} + \begin{pmatrix} 0 \\ M^{-1}(x_1) \end{pmatrix} u$$

Simulation

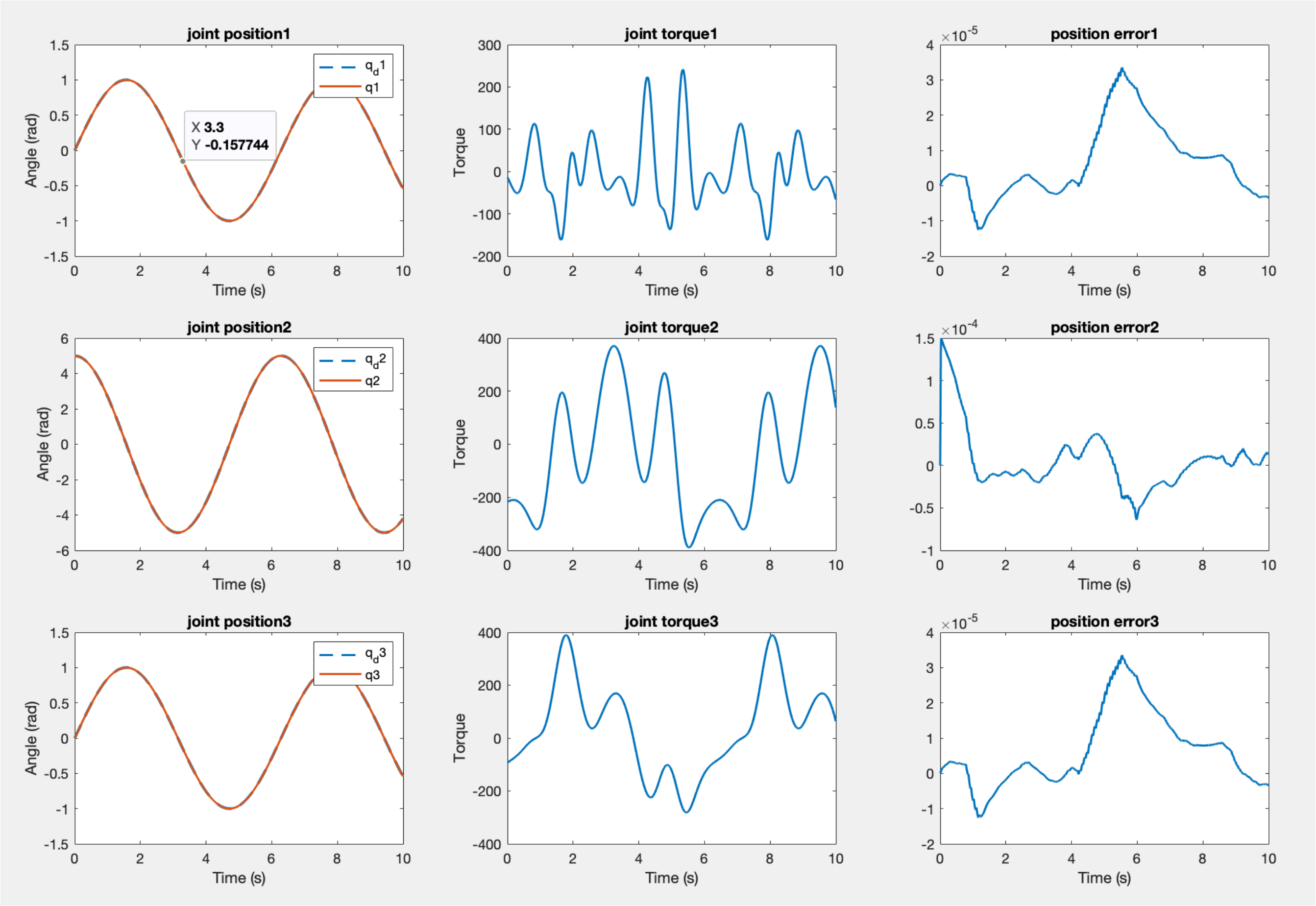
- We used the ode15s integration routine instead of ode45 because, during the simulation —especially when working with robust control— we encountered stiff ordinary differential equations.

- In stiff systems, some variables change much more rapidly than others. This often occurs due to the robust controller, which is designed to handle parameter uncertainties and must react quickly and suddenly to external variations, causing ode45 to drastically reduce the integration step size to maintain stability and accuracy. This leads to excessively long simulation times.

- In contrast, ode15s is specifically designed to handle stiff systems. It uses a more stable approach to manage rapid variations in the variables, preventing the integrator from becoming excessively slow.

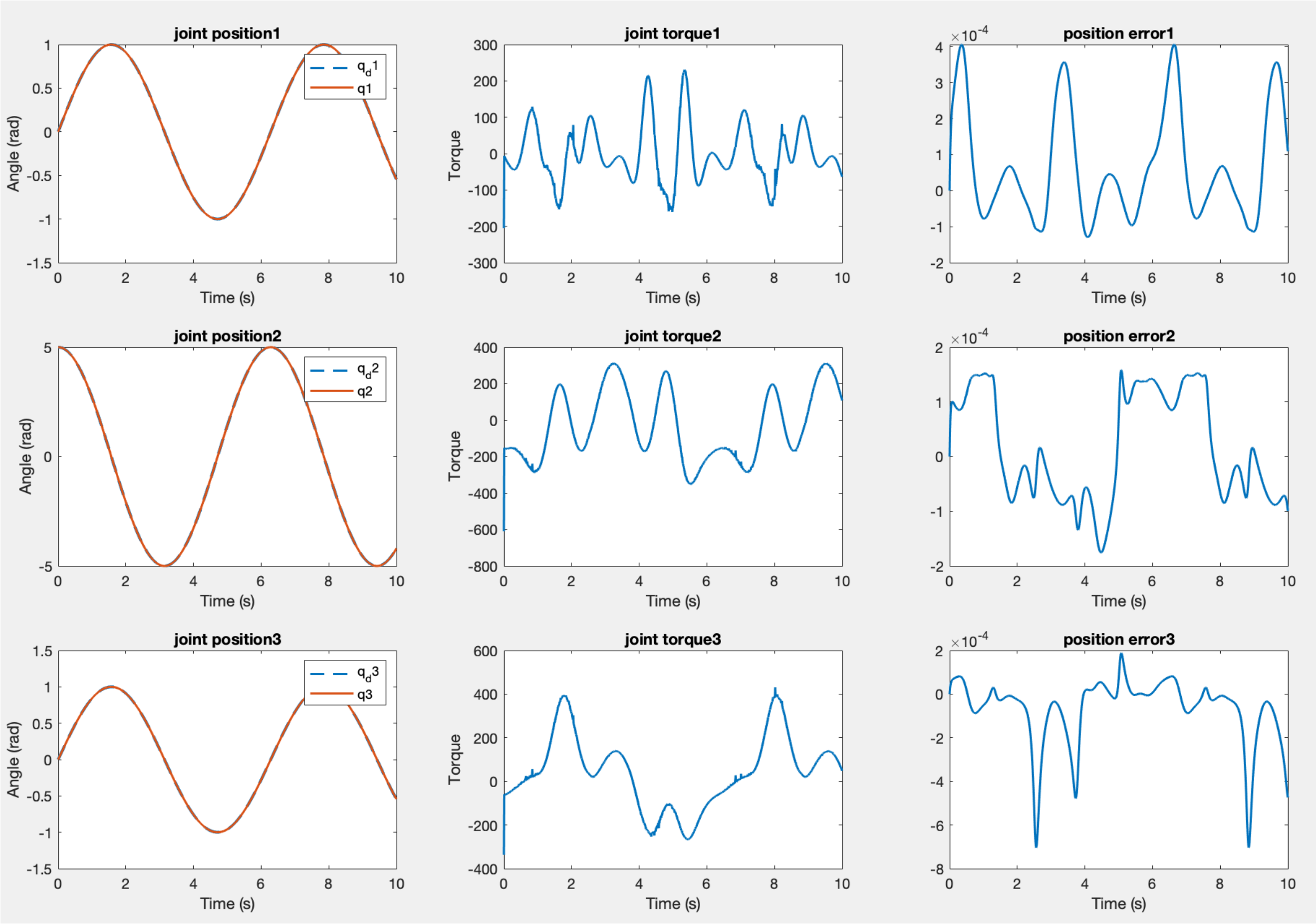
Simulation

- The simulaton under ideal conditions using the feedback lianearization control law is executed with the gains: $K_p = diag(100{,}100{,}100,)\ K_d = diag(100{,}100{,}100)$:



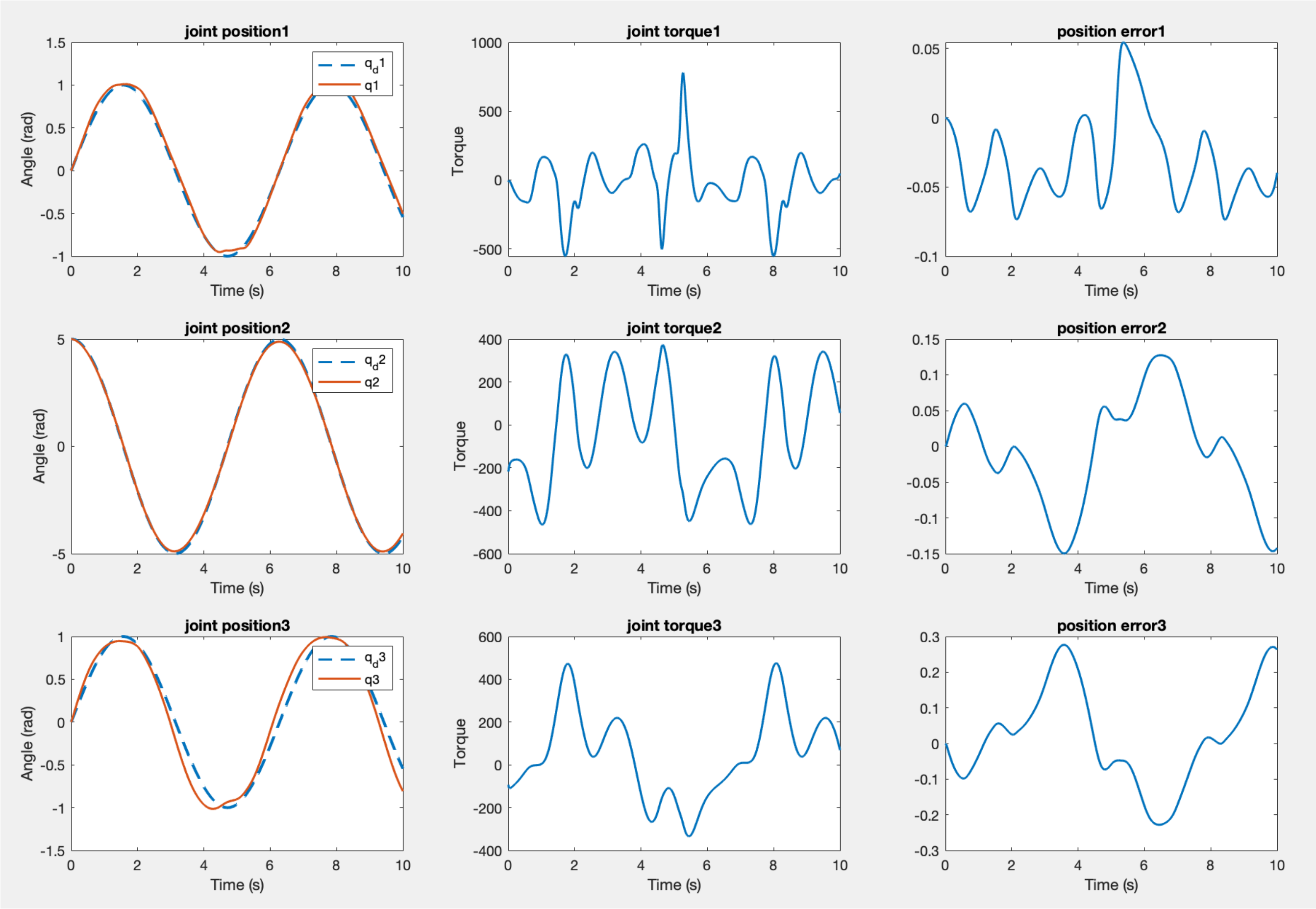
Simulation

- The simulaton under ideal conditions using the Robust control law is executed with the gains: $\rho = 186.8129, K = diag(100,100,100),\ \Lambda = diag(50,50,50),\ \epsilon = 0.5$:



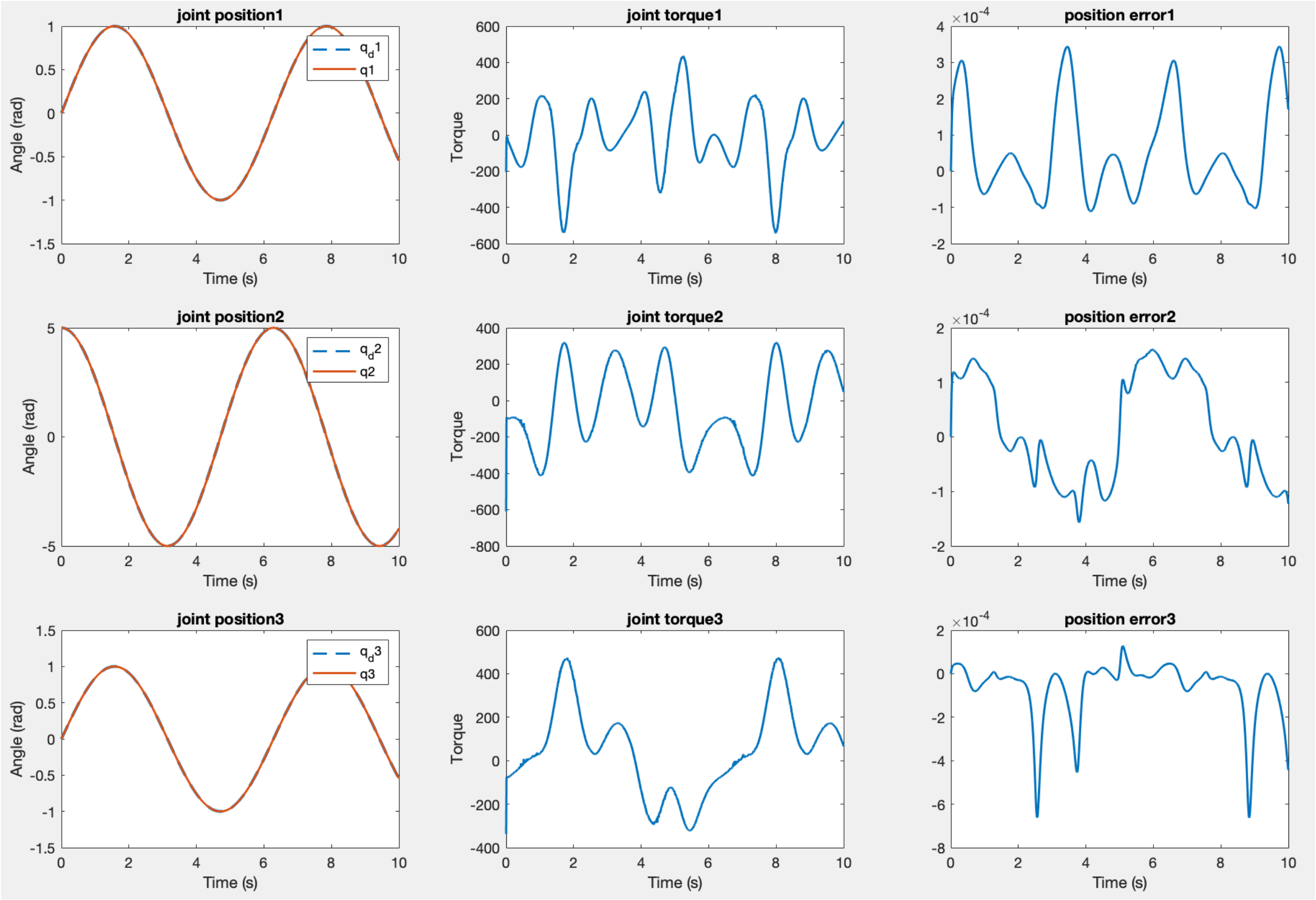
Simulation

- The simulaton under perturbed conditions using the feedback lianearization control law is executed with the gains: $K_p = diag(100{,}100{,}100,)\ \ K_d = diag(100{,}100{,}100)$:



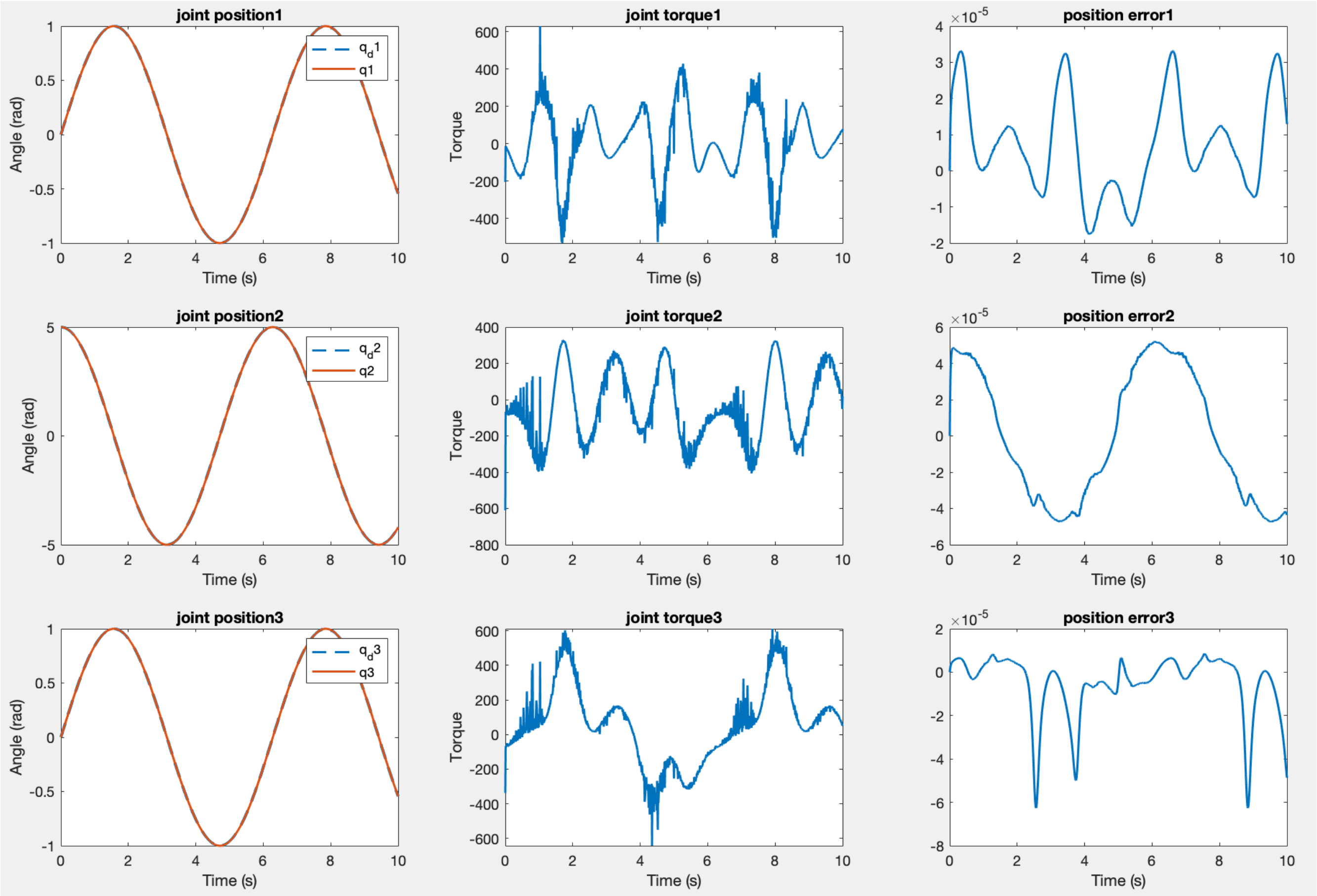
Simulation

- The simulaton under perturbed conditions using the Robust control law is executed with the gains: $\rho = 186.8129, K = diag(100{,}100{,}100),\ \Lambda = diag(50{,}50{,}50),\ \epsilon = 0.5$:



Simulation

- $\epsilon = 0.05$, pay attention to chattering!



Conclusion

- **Robust Control** offers superior performances in more realistic scenarios, particularly when dynamic coefficients are affected by uncertainty, exceeding in situations where a classic feedback linearization controller struggles.

- **Feedback Linearization Control**, on the other hand, is more suited to ideal scenarios or cases where the dynamic model is well understood and accurately defined.

- **Robust Control** is an effective strategy when moderate and predictable uncertainties are present. It provides reliable performance and strong disturbance rejection, making it a solid choice in real-world applications where variability in system dynamics is expected.

- The control law we designed offers a good balance between **performance and complexity**. The price we pay is that we must have prior knowledge on bounds of these uncertainties to ensure effective performance.

Conclusion

- However, this control law provides **ultimately uniformly stability (u.u.s.)**, ensuring the system remains stable over time even when exposed to disturbances, making it a valuable solution for robust and reliable robotic control.