Part II

Control design of robot arm process

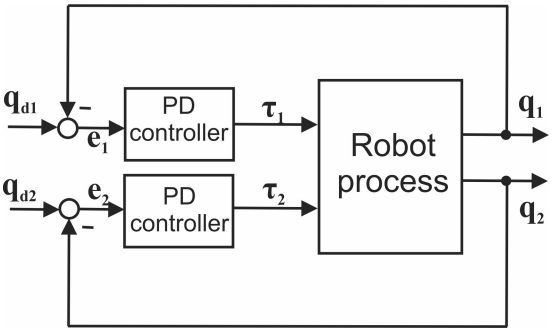

Figure 2: Independent joint control - PD controllers

5. Independent Joint Control

Independent joint control is the classical control approach, where a controller is designed for each individual joint (Fig. 2). The effect of other joints is considered as a disturbance that the controller has to reject.

Let us first rewrite model in terms of each individual joint i ([4]):

$$(J_{mi} + r_i^2 d_{ii})\ddot{q}_i + B_i\dot{q}_i + r_i F_{mi} = (r_i K_{mi}/R_{ai})v_i - r_i^2 w_i, \quad i = 1, 2, \tag{33}$$

where d_{ii} includes only the constant diagonal terms of $D'(q)$, while the disturbance term w_i includes all other i^{th} terms of $D'(q)$, and i^{th} components of $C(q, \dot{q})$,$F(\dot{q})$ and $G(q)$. It important no notice that if the gear ratio r_i is small, and the Coriolis and centripetal terms are not very large, then the process dynamics can be well approximated by *n decopuled linear second-order systems.*

We will further consider that the same motor type is used to drive each joint. Then, by also incorporating the friction term F_m into the disturbance w, we can rewrite 33 as [10]

$$(\underbrace{J_m + r^2 d_{ii}}_{J_{pi}})\ddot{q}_i + B\dot{q}_i = (\underbrace{rK_m/R_a}_{K})v_i - w_{pi}, \quad i = 1, 2, \tag{34}$$

Control problem: Design a linear controller for each joint that ensure tracking (q tracks a reference signal q_d) and disturbance rejection (w_p).

Consider standard PD controllers

$$v_i = K_{di}\dot{e}_i + K_{pi}e_i, \tag{35}$$

where e_i represents the tracking error, defined as $e_i = q_{di} - q_i$, and with $\dot{e}_i = \dot{q}_{di} - \dot{q}_i$.

If we consider the case of set-point tracking, that is $\dot{q}_{di} = 0$, and replace (35) in 34 we obtain:

$$J_{pi}\ddot{q}_i(t) + (B + KK_{di})\dot{q}_i(t) + KK_{pi}q_i(t) = KK_{pi}q_{di}(t) - w_{pi}(t), \quad i = 1, 2, \tag{36}$$

By applying the Laplace transform we get the transfer function relating the outputs (q_i) to the reference and disturbance inputs (q_{di} and w_{pi}):

$$q_i(s) = \frac{KK_{pi}}{J_{pi}s^2 + (B + KK_{di})s + KK_{pi}}q_{di}(s) - \frac{1}{J_{pi}s^2 + (B + KK_{di})s + KK_{pi}}w_{pi}(s), \quad i = 1, 2, \tag{37}$$

The characteristic equations are

$$J_{pi}s^2 + (B + KK_{di})s + KK_{pi} = 0,$$

which can also be written as

$$s^2 + \frac{B + KK_{di}}{J_{pi}}s + \frac{KK_{pi}}{J_{pi}} = 0.$$

[10] B and K are now scalars.

Because the standard second order equation is given by

$$s^2 + 2\zeta_i \omega_{ni} s + \omega_{ni}^2 = 0,$$

the controller parameters can be expressed in terms of damping ratio ζ and natural frequency ω_n:

$$K_{pi} = \frac{J_{pi}\omega_{ni}^2}{K}, \quad K_{di} = \frac{2\zeta_i \omega_{ni} J_{pi} - B}{K}. \tag{38}$$

Usually ζ is set to 1 (critical damping), and ω_n is chosen as high as possible. One possible limitation in the value adopted for ω_n is the control signal v_i saturation. The control strategy proves very efficient in practice. If we further want to force a very small or null steady state error e_{ss}, then we can either adopt PID type controllers

$$v_i = K_{di}\dot{e}_i + K_{pi}e_i + K_{ii}\int_0^t e\, dt, \tag{39}$$

or PD controllers with an additional gravity term [11]

$$v_i = K_{di}\dot{e}_i + K_{pi}e_i + G_i'(q). \tag{40}$$

[11] Note that in this case the control signals are not decoupled anymore.

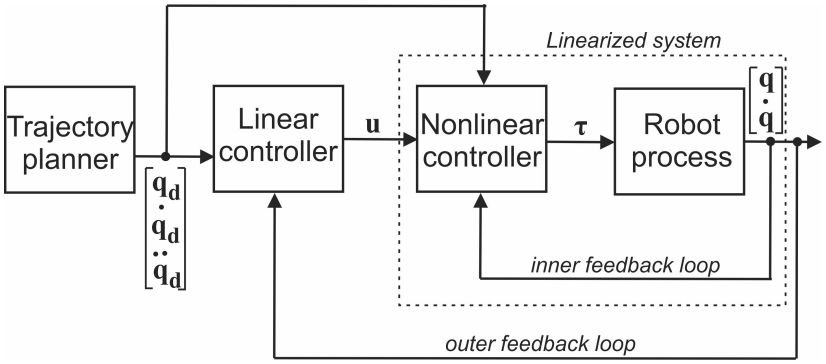

Figure 3: Computed torque control - feedback linearization - principal control structure

6. Computed Torque Control

Computed torque control (in some places called feedback linearization control) is a more sophisticated and modern control strategy that can be used to increase the control performances [12]. The control consists out of an inner feedback loop and an outer feedback loop - Figure 3. Although it is a nonlinear control approach, because the inner feedback loop achieves dynamic linearization, the outer feedback loop resume to a classical linear control design.

Consider the robot arm process model 30, where we neglect the motor dynamics [13]. By the change of notation $V(q,\dot{q}) = C(q,\dot{q})\dot{q} + F(\dot{q}) + G(q)$ we can rewrite the model more compactly as

$$D(q)\ddot{q} + V(q,\dot{q}) = \tau \tag{41}$$

Consider the following control law for the inner control loop:

$$\tau = D(q)(\ddot{q}_d - u) + V(q,\dot{q}) \tag{42}$$

We define the tracking error as

$$e = q_d - q. \tag{43}$$

Then $\dot{e} = \dot{q}_d - \dot{q} \quad and \quad \ddot{e} = \ddot{q}_d - \ddot{q}$.

By replacing the torque from 42 in model 41, and using the definition 43, we obtain:

$$\ddot{e} = u \tag{44}$$

Thus the inner feedback loop achieves dynamic linearziation, in other words the outer loop "sees" a double integrator process. The double intergator model 44 can be written in state space form:

$$\dot{x} = Ax + Bu \tag{45}$$

with

$$x = \begin{bmatrix} e \\ \dot{e} \end{bmatrix}, A = \begin{bmatrix} 0_2 & I_2 \\ 0_2 & 0_2 \end{bmatrix}, B = \begin{bmatrix} 0_2 \\ I_2 \end{bmatrix} \tag{46}$$

Now the outer loop can be designed by any classical linear control technique. We will consider here a state feedback control law with integrator component:

$$u = -Kx + K_i\epsilon \tag{47}$$

where ϵ is the output of the intergator ($\dot{\epsilon} = e$) - see Figure 4. We will first design the state feedback gain K using Ackerman's formula for a 4^{th} order system:

$$K = [0\,0\,0\,1]M_c^{-1}\Delta(A) \tag{48}$$

[12] Independent joint control and Computed Torque control are the most widely ecountered control tehniques in the literature on Robot Control Systems ([4],[1], [5])

[13] i.e. the robot is controlled directly in torque.

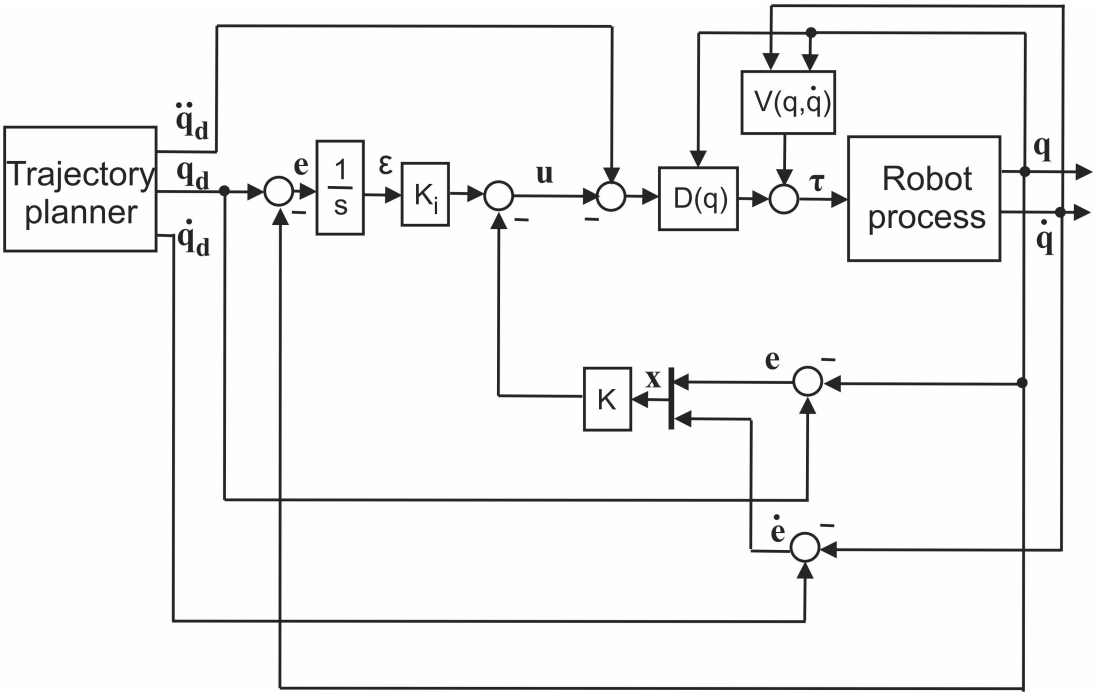

Figure 4: Computed torque control- feedback linearization -detailed control structure

where Δ is the desired characteristic polynomial of the closed loop system

$$\Delta(s) = s^4 + a_3 s^3 + a_2 s^2 + a_1 s^1 + a_0, \tag{49}$$

and M_c is the controllabilty matrix:

$$M_c = [B|AB|A^2B|A^3B]. \tag{50}$$

After the state feedback gain K is designed through pole placement, we will consider that the integrator gain ca be designed through trial and error in simulations, in order the achieve the best tracking performances possible, while tacking into account nonlinear effects like saturation, dead-zones or backlash.

Finally, the detailed control structure is shown in Figure 4.

7. Numerical results

7.1. Robot arm example

As an example of a 2DOF robot with the structure as in Figure 1, consider the robot from Figure 5 [14]. For this particular robot arm, we will neglect the motor dynamics entirely, and consider that we can control the robot process directly in torque [15], that is we will use model 30.

The parameters are [16]:$L_1 = 0.095\ m$, $L_2 = 0.1\ m$, $m_1 = 0.095\ kg$, $m_2 = 0.37\ kg$, $g = 9.81\ m/s^2$, $I_{1x} = 2.27\ 10^{-2}\ kg\ m^2$, $I_{2y} = 2.27\ 10{-}2\ kg\ m^2$, $b_1 = 0.24$, $b_2 = 0.16$, $r = 1$. The torque control signal is limited to the range $[-1.18, 1.18]\ Nm$.

7.2 Independent Joint Control

Here we will design PD controller for independent joint controller

$$\tau_1 = K_{d1}\dot{e}_1 + K_{p1}e_1,$$

[14]This robot was designed and built by Zoltan Nagy - see [2]

[15]This is possible due to the fact that an affine relation between torque and PWM (the actual control signal generated by the hardware/PC on which the controller is implemented) was determined experimentally in [2].

[16]The parameters were either measured or estimated - see [2] for details.

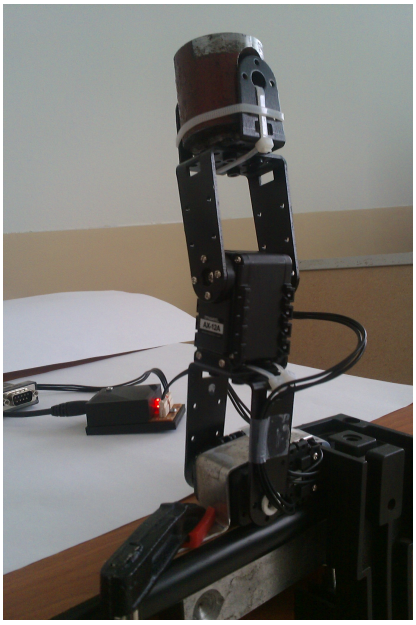

Figure 5: Real robot arm process - from [2]

$$\tau_2 = K_{d2}\dot{e}_2 + K_{p2}e_2,$$

,for the robot process 30, with the parameters given in previous section. We will use the formula 38, with $B = 0, J_{p1} = 0.0263, J_{p2} = 0.0236$, $\zeta = 1$, $K = 1$. We started from a value for $\omega_n = \omega_{n1} = \omega_{n2}$ of 0.1, an increased it until the step respose of the closed loop system is fast enough, and the control torques reach saturation for a small time interval. In the end, a value of 12 provided good enough results.

The implementation of the controllers and process was done in Matlab/Simulink. Figure 6 shows simulations results of the closed loop system with step reference signals, while Figure 7 shows the control torques. The results show that a small steady state is present. This can be eliminated by adding a gravity term as in 40. The results with PD+gravity control are shown in Figure 8. It can be noticed that now the steady state error is zero.[17] A more interesting and demanding tracking scenario is that when the reference signals are sinusoidal ($q_d(t) = sin(t)$), which is shown Figure 9 for the PD+gravity controllers. Although the steady state error is now exactly zero, because the reference continuously changes, the error is kept into very small limits.

7.3 Computed Torque Control

Consider the inner feedback loop with control law 42, and the outer feedback loop with control law 47 (Figure 3).

We will design the state feedback gain K for the double integrator process 44, through pole placement. Consider the following closed loop pole configuration [-4 -4 -9 -9] (desired eigenvalues for $A - BK$). The poles where chosen such that we get an overdamped and fast enough response, while avoiding saturation as much as possible [18]. We obtained the gain values

$$K = \begin{bmatrix} 36 & 0 & 13 & 0 \\ 0 & 36 & 0 & 13 \end{bmatrix},$$

using the *place* function in Matlab.

The integrator gain K_i is found through trial and error in simulations: we increase the gain starting from an initial value of 0.1 - until the maximum steady state error starts to increase. We keep the previous iterated value. Thus, we finally arrive at the value 0.6 (that is $K_i = [0.6 \quad 0.6]$).

For a better comparison between the performances of the PD+gravity control approach versus the computed torque control, we reconsider the sinusoidal reference input from the previous section, but now

[17] Note that in the case of PD+gravity controller, the controller is no longer joint independent, due to the gravity terms that contain expressions in both joint variables q_1 and q_2.

[18] As the poles are moved farther to the left in the complex plane, the response becomes faster, but the control effort increases.

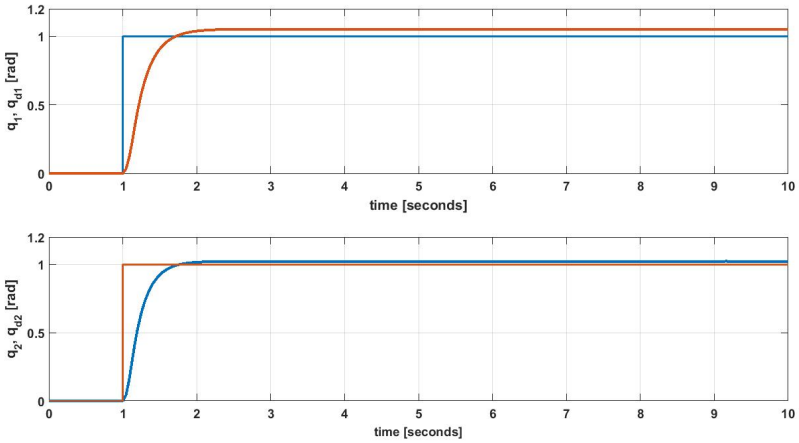

Figure 6: Simulations with PD independent joint control - step response

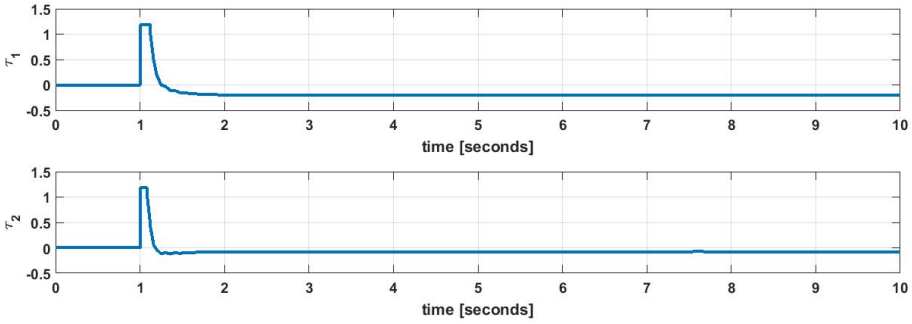

Figure 7: Control torques for PD independent joint control

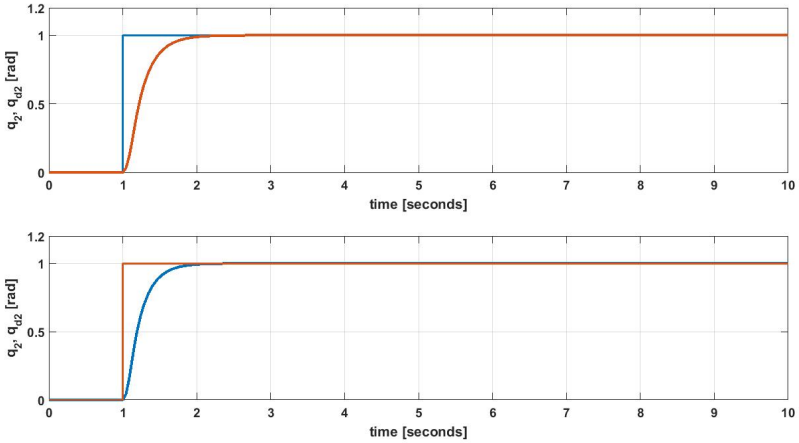

Figure 8: Simulations with PD+Gravity joint control - step response

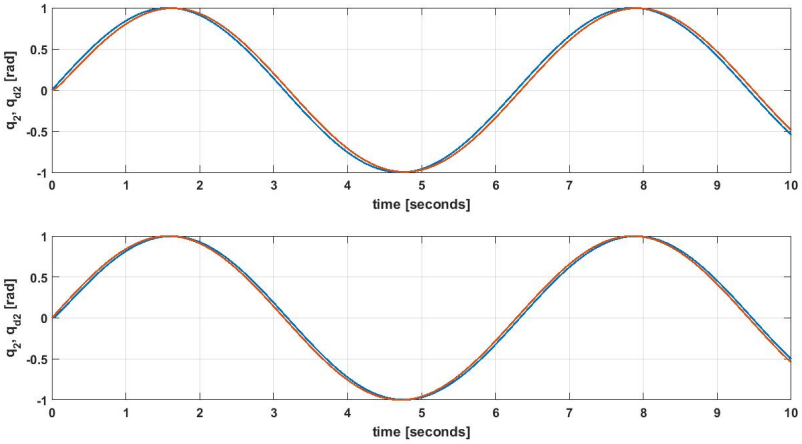

Figure 9: Simulations with PD+Gravity joint control - sinusoidal response

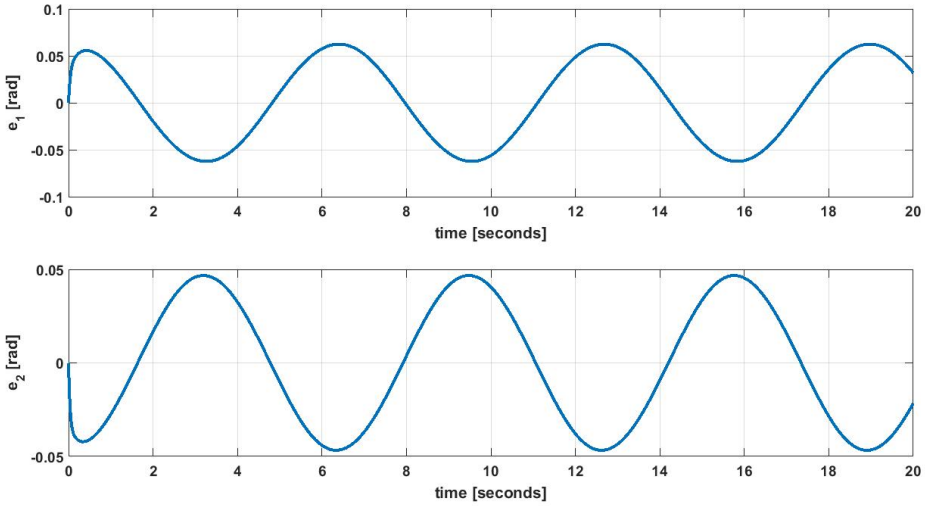

Figure 10: Tracking error for PD+grav control

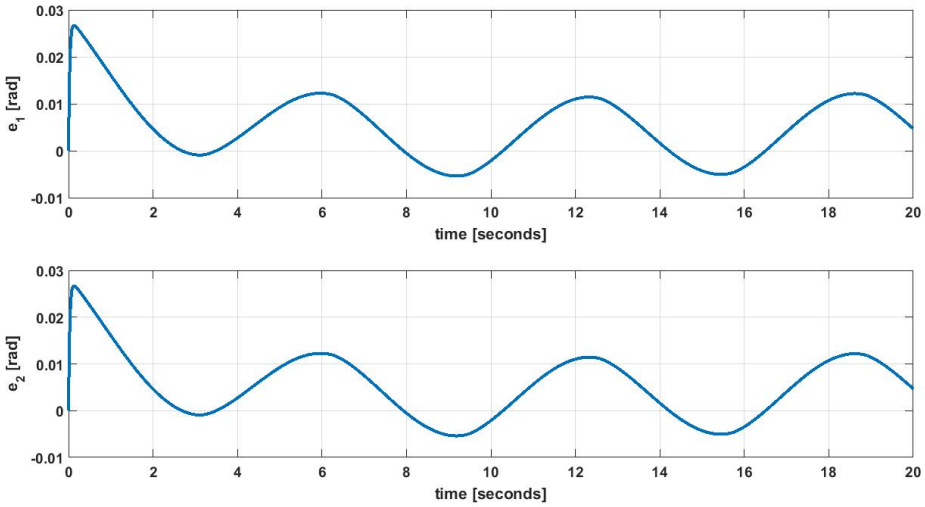

Figure 11: Tracking error for Computer Torque Control

we will look directly at the tracking errors defined as $e_1 = q_{d1} - q_1$, $e_2 = q_{d2} - q_2$. Figure 10 shows the tracking errors for the two joint positions in the case of PD+gravity control. The maximum error is about 0.05 rad (2.8 deg). For the Computed Torque control the results are shown in Figure 11. Notice that the maximum error is now much smaller 0.01 rad (0.5 deg). Of course that the cost is an increase complexity for the controller - the implementation requires a considerable increase in computational power.

Bibliography

[1] Mark W. Spong, Seth Hutchinson, M. Vidyasagar, *Robot Modeling and Control*, Wiley, 2006.

[2] Zoltán Nagy, *Hardware and controller design for mechanical systems - Inverted pendulum and robot arm*, Master Thesis, Technical University of Cluj-Napoca, 2017.

[3] Daniel Liberzon, *Calculus of Variations and Optimal Control Theory: A Concise Introduction*, Princeton University Press, 2012.

[4] Frank L. Lewis, Darren M. Dawson, Chaouki T. Abdallah, *Robot Manipulator Control: Theory and Practice*, CRC Press, 2003.

[5] Jean-Jacques Slotine , Weiping Li , *Applied Nonlinear Control*, Pearson, 1991.