Nonlinear Control of a Robot Arm

Gia Quoc Bao Tran and Ricardo Wang

Institute of Engineering, Université Grenoble Alpes

Date of presentation: 15 December 2020

Introduction

This lab's objective is to learn to apply stabilization by input/output (I/O) linearization for the nonlinear control of a robot arm system.

Contents

1- System modeling and state-space representation

System modeling

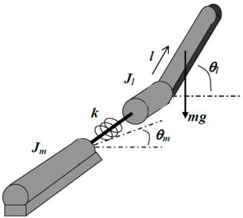

Figure: Robot arm with flexible joint.

Dynamical equations:

$$\begin{cases} J_l\ddot{\theta}_l + B_l\dot{\theta}_l + k\cdot(\theta_l - \theta_m) + mglcos(\theta_l) = 0 \\ J_m\ddot{\theta}_m + B_m\dot{\theta}_m - k\cdot(\theta_l - \theta_m) = u \end{cases} \quad (1)$$

Parameters: $k = 0.8Nm/rad,\ J_m = J_l = 4\cdot 10^{-4}Nms^2/rad,$
$B_m = 0.015Nms/rad,\ B_l = 0.0,\ m = 0.3kg,\ l = 0.3m,\ g = 9.8m/s^2$

Nonlinear state-space representation

Denoting $x_1 = \theta_l$, $x_2 = \dot{\theta}_l$, $x_3 = \theta_m$, and $x_4 = \dot{\theta}_m$, we have the nonlinear state-space representation:

$$\dot{x} = f(x) + g(x) \cdot u \tag{2}$$

where

$$f(x) = \begin{bmatrix} x_2 \\ -\frac{1}{J_l}(mglcos(x_1) + k \cdot (x_1 - x_3) + B_l x_2) \\ x_4 \\ \frac{1}{J_m}(k \cdot (x_1 - x_3) - B_m x_4) \end{bmatrix}, g(x) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{J_m} \end{bmatrix} \tag{3}$$

2- Control approaches

Approximate linearization

If we want to stabilize the system at $\theta_l = 0 = x_{1,0}$, the considered output is θ_l. By letting $\dot{x} = 0$, we obtain $x_{2,0} = 0$, $x_{4,0} = 0$, and:

$$\begin{cases} mgl - k \cdot x_{3,0} = 0 \\ -k \cdot x_{3,0} + u_0 = 0 \end{cases} \rightarrow \begin{cases} x_{3,0} = \frac{mgl}{k} \\ u_0 = mgl \end{cases} \tag{4}$$

Equilibrium point: $x_0 = \begin{pmatrix} 0 & 0 & \frac{mgl}{k} & 0 \end{pmatrix}^T$ and $u_0 = mgl$.

Approximate linearization

Using Jacobian linearization, we obtain the following linear system:

$$\Sigma_l : \begin{cases} \dot{\tilde{x}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k}{J_l} & -\frac{B_l}{J_l} & \frac{k}{J_l} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{k}{J_m} & 0 & -\frac{k}{J_m} & -\frac{B_m}{J_m} \end{bmatrix} \cdot \tilde{x} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{J_m} \end{bmatrix} \cdot \tilde{u} \\ y = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \cdot \tilde{x} \end{cases} \tag{5}$$

where $\tilde{x} = x - x_0$ and $\tilde{u} = u - u_0$.

- Stabilize the system at $\theta_l = 0 \rightarrow$ Drive the output to 0.
- Controllable & observable $\rightarrow$ Possible linear control methods, e.g., state feedback, LQR,...
- For good performance $\rightarrow$ Apply a nonlinear approach called I/O stabilization as the following.

I/O linearization: Check conditions

$$y = \frac{\pi}{2} - x_1 \tag{6}$$

$$\dot{y} = -\dot{x}_1 = -x_2 \tag{7}$$

$$\ddot{y} = -\dot{x}_2 = \frac{1}{J_l}(mglcos(x_1) + k \cdot (x_1 - x_3) + B_l x_2) \tag{8}$$

$$\begin{aligned} \dddot{y} &= \frac{1}{J_l}(-mglx_2 sin(x_1) + k \cdot (x_2 - x_4) + \\ &+ B_l(-\frac{1}{J_l}(mglcos(x_1) + k \cdot (x_1 - x_3) + B_l x_2))) \end{aligned} \tag{9}$$

$$\begin{aligned} y^{(4)} = \frac{1}{J_l}(-mgl(-\frac{1}{J_l}(mglcos(x_1) + k \cdot (x_1 - x_3) + B_l x_2))sin(x_1) + \\ - mglx_2^2 cos(x_1) + k \cdot ((-\frac{1}{J_l}(mglcos(x_1) + k \cdot (x_1 - x_3) + B_l x_2)) + \\ -(\frac{1}{J_m}(k \cdot (x_1 - x_3) - B_m x_4) + \frac{1}{J_m}u)) + B_l \cdot (-\frac{1}{J_l}(-mglx_2 sin(x_1) + \\ + k \cdot (x_2 - x_4) + B_l \cdot (-\frac{1}{J_l}(mglcos(x_1) + k \cdot (x_1 - x_3) + B_l x_2))))) \end{aligned}$$

I/O linearization: Two-part control input

The system's order is $n = 4$. Its relative order is $r = 4$: $y^{(k)}(t)$ only depends on u for $k = n$ and not for $0 \leq k \leq n-1$, which is the condition for I/O linearization. Rewriting the obtained result:

$$y^{(4)} = a(x) + b(x) \cdot u \tag{11}$$

Now if we let $u = \frac{1}{b(x)} \cdot (-a(x) + v)$, then $y^{(4)} = v$ where v is the control input that stabilizes the linearized system.
The diffeomorphism $z = T(x)$ changes the system into a linear one, where the states are the output and its time derivatives. We have a linear system defined with the following states:

$$z_1 = y, z_2 = \dot{y}, z_3 = \ddot{y}, z_4 = \dddot{y} \tag{12}$$

There is no zero dynamics, i.e., we completely linearize the original system. Stabilizing this linearized system is therefore equivalent to stabilizing the original one.

I/O linearization

The I/O linearized system's state-space representation is:

$$\Sigma_{I/O}: \begin{cases} \dot{z} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \cdot z + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \cdot v =: A_{I/O} \cdot z + B_{I/O} \cdot v \\ y = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \cdot z =: C_{I/O} \cdot z \end{cases} \tag{13}$$

The steps of finding $a(x)$, $b(x)$, and the diffeomorphism $T(x)$ have been done by MATLAB using the Lie derivative.
This linearized system has been verified to be both controllable and observable, and therefore we can control it to track a certain reference using a linear control method, e.g., state feedback control, to track $r = 0$.

Stabilizing state-feedback controller

The control input is $v = -K \cdot z + G \cdot r$ where K is chosen such that the closed-loop system has four poles at -100 (we want later to simulate the system in a short period of time because the simulation runs quite slowly) and G is found accordingly such that a unit static gain is guaranteed, i.e., $G = (C_{I/O} \cdot (-A_{I/O} + B_{I/O} \cdot K)^{-1} \cdot B_{I/O})^{-1}$. Simulation with the closed-loop linearized system shows that it can track a reference with no overshoot.

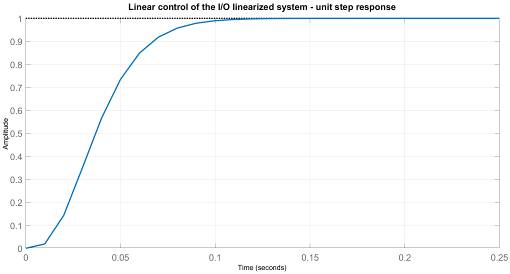

Figure: Simulation of the closed-loop linearized system.

3- Validation of approaches

Description of simulation model

We have constructed a Simulink model, as shown. It consists of two feedback loops. The inner one (the upper loop) is for I/O linearization, where u linearizes the original system into a linear one controlled by v. The diffeomorphism changes that into another one in terms of y and its time derivatives. The outer feedback loop (the lower loop) is for v to stabilize the linearized system.

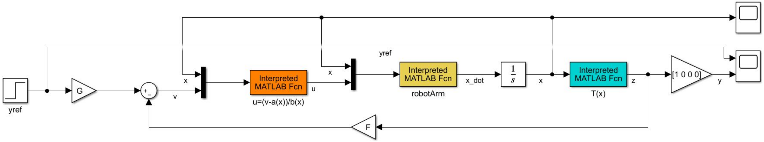

Figure: Simulation of the controlled system.

Validation by simulation

Simulation results are as below. We see that the output can track $r = 0$.

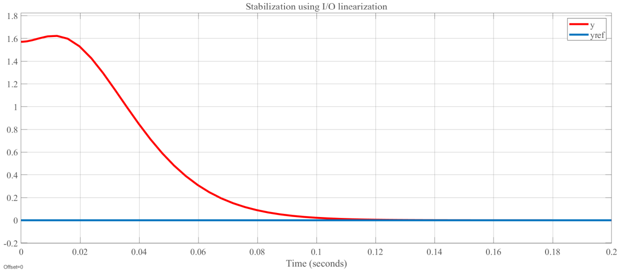

Figure: Stabilization using I/O linearization simulation results.

We observe that the settling of the nonlinear system is the same as the linearized one (around 0.1s), and there is no overshoot.

4- Extensions

Internal stability analysis

If we perform I/O stabilization, there is a possibility that the internal stability of the system is not guaranteed, i.e., (some of) the states can diverge. We have watched the states, and here we can see that they do not diverge.

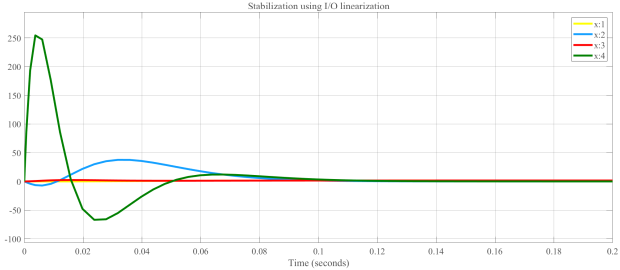

Figure: Stabilization using I/O linearization stability analysis.

The stabilization using the I/O linearization approach works in this case, which is coherent. If internal stability is not guaranteed, we need to apply I/O linearization with input/state (I/S) linearization.

Robustness analysis

The control approach here depends on an inner I/O linearization loop that relies on the knowledge of the system's parameters. Therefore, we would like to examine this controlled system's robustness w.r.t system modeling uncertainty by experimenting with different system modeling errors.

As $J_l = J_m$ are among the parameters that the system is most sensitive to (they are multiplied with the acceleration term), we want to analyze the system's robustness w.r.t an error on these parameters. To do this, in our calculation of the diffeomorphism and the functions $a(x)$ and $b(x)$, we use $J_l = J_m = 5 \cdot 10^{-4} Nms^2/rad$, i.e., an increase by 25% from the nominal values.

It follows that since the linearized system remains unchanged, the control input v is the same as above. However, the diffeomorphism and the functions $a(x)$ and $b(x)$ all have changed, leading to u being different from above.

Robustness analysis

Simulation results are as follows:

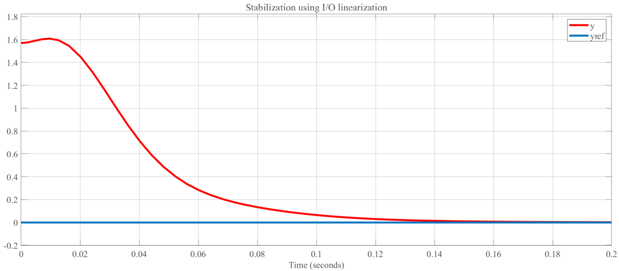

Figure: Stabilization using I/O linearization robustness analysis (output).

We see that even though the uncertainty makes the output converge to 0 more slowly (around 0.14s instead of 0.1s), it still converges. Our controller is robust enough w.r.t a 25% uncertainty in J_l and J_m.

Disturbance rejection: Formulation

With the use of the inner linearization loop, we can expect to apply different linear control approaches, including integral control which can reject disturbances (and hopefully improve robustness as in the previous lab). Let us denote the integral part:

$$p(t) = \int_0^t (r(\tau) - y(\tau))\, d\tau \tag{14}$$

The extended system is:

$$\Sigma_{I/Oext} : \begin{cases} \begin{bmatrix} \dot{z} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} A_{I/O} & 0 \\ -C_{I/O} & 0 \end{bmatrix} \cdot \begin{bmatrix} z \\ p \end{bmatrix} + \begin{bmatrix} B_{I/O} \\ 0 \end{bmatrix} \cdot v + \begin{bmatrix} 0 \\ I \end{bmatrix} \cdot r + \begin{bmatrix} E \\ 0 \end{bmatrix} \cdot d \\ y = \begin{bmatrix} C_{I/O} & 0 \end{bmatrix} \cdot \begin{bmatrix} z \\ p \end{bmatrix} \end{cases} \tag{15}$$

where E is the gain of the disturbance d. The pole placement method is used to obtain the gains of the controllers, where the poles are chosen similar to above. A faster pole (further from the imaginary axis) is given to the integral part so as not to interfere with the system's dynamics.

Disturbance rejection: Results

The simulation results are as follows, where we have a step input disturbance on u at 0.2s:

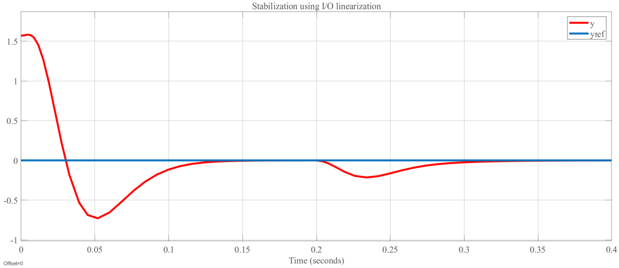

Figure: Integral control using I/O linearization.

We see that the disturbance is rejected thanks to integral action. Besides, we have verified that internal stability is still guaranteed.

Disturbance rejection: Robustness

We also remark that this integral controller is more robust w.r.t system modeling uncertainty, as we have tried with a different value of J_l and J_m like in the last part:

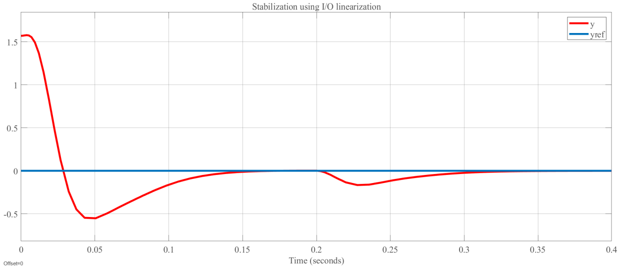

Figure: Integral control using I/O linearization robustness analysis (output).

In this case the settling time remains almost the same (around 0.15s) with the same modeling error (25%), which means that the system's robustness has been improved. And indeed, internal stability is still guaranteed.

Conclusion

After this lab, the students understand better the use of I/O linearization for a nonlinear control problem. We have also analyzed the controlled system in terms of internal stability and robustness. We see that when the nonlinear system can be linearized thanks to the I/O approach, we can apply linear control methods, such as state-feedback or integral control.

Questions/discussions/suggestions

Thank you for your attention!