Nonlinear System Analysis and Control
Lab 2: Robot Arm Control

Date: 8 December 2020

1 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.

2 System modeling

Consider a robot arm made of a drive actuating a shaft through a flexible joint, illustrated as in the following figure:

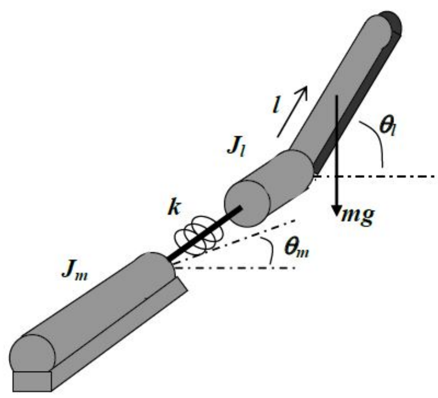

Figure 1: Robot arm with flexible joint.

This system's dynamical equations are:

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

We can write this second-order system as a nonlinear state-space representation. Denoting the states as $x_1 = \theta_l$, $x_2 = \dot{\theta}_l$, $x_3 = \theta_m$, and $x_4 = \dot{\theta}_m$, we have:

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

where

$$f(x) = \begin{bmatrix} x_2 \\ -\frac{1}{J_l}(mglcos(x_1) + k\cdot(x_1 - x_3) + B_lx_2) \\ x_4 \\ \frac{1}{J_m}(k\cdot(x_1 - x_3) - B_mx_4) \end{bmatrix}, g(x) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{J_m} \end{bmatrix} \tag{3}$$

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

3 Stabilization using 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:

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

The equilibrium point is $x_0 = (0 \quad 0 \quad \frac{mgl}{k} \quad 0)^T$ and $u_0 = mgl$.

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 $\tilde{u} = u - u_0$. By stabilizing the system at $\theta_l = 0°$, we drive the output to 0. Possible design methods include using state feedback (this linear system has been verified to be both controllable and observable), with or without integral action to reject the disturbances, and the controller gains can be calculated using the pole placement or linear-quadratic regulator (LQR) approach.
However, in the previous lab, we have seen that applying a linear control method to a nonlinear system may lead to bad performance. Therefore, we would like to apply a nonlinear approach called I/O stabilization as below.

4 Stabilization using I/O linearization

As we want to stabilize the system at $\theta_l = 90°$, the considered output is $y = \frac{\pi}{2} - \theta_l = \frac{\pi}{2} - x_1$. First, we check the system's relative degree, beginning by taking the output's time derivative:

$$\dot{y} = -\dot{x}_1 = -x_2 \tag{6}$$

The control input has not appeared yet, so:

$$\ddot{y} = -\dot{x}_2 = \frac{1}{J_l}(mglcos(x_1) + k \cdot (x_1 - x_3) + B_l x_2) \tag{7}$$

The control input still has not appeared yet, so:

$$\dddot{y} = \frac{1}{J_l}(-mglx_2 sin(x_1) + k \cdot (x_2 - x_4) + B_l(-\frac{1}{J_l}(mglcos(x_1) + k \cdot (x_1 - x_3) + B_l x_2))) \tag{8}$$

Taking the derivative the last time, we obtain:

$$\begin{aligned} y^{(4)} = & \frac{1}{J_l}(-mgl(-\frac{1}{J_l}(mglcos(x_1) + k \cdot (x_1 - x_3) + B_l x_2))sin(x_1) - mglx_2^2 cos(x_1) + \\ & +k \cdot ((-\frac{1}{J_l}(mglcos(x_1) + k \cdot (x_1 - x_3) + B_l x_2)) - (\frac{1}{J_m}(k \cdot (x_1 - x_3) - B_m x_4) + \frac{1}{J_m}u)) + \\ & +B_l \cdot (-\frac{1}{J_l}(-mglx_2 sin(x_1) + k \cdot (x_2 - x_4) + B_l \cdot (-\frac{1}{J_l}(mglcos(x_1) + k \cdot (x_1 - x_3) + B_l x_2))))) \end{aligned} \tag{9}$$

Now we see that the control input u (in red) has appeared. We can conclude that the system's relative order is $r = 4$, while its order is $n = 4$. We have checked that $y^{(k)}(t)$ only depends on u for $k = n$ and not for $0 \leq k \leq n-1$, which is the condition for I/O linearization, for the diffeomorphism $T(x)$, i.e., $z = T(x)$ to be defined. The diffeomorphism changes the system into a linear one, where the states are the output and its time derivatives.
Now we write the last equation as:

$$y^{(4)} = a(x) + b(x) \cdot u \tag{10}$$

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.
Now if we let $u = \frac{1}{b(x)} \cdot (-a(x) + v)$, then $y^{(4)} = a(x) + b(x) \cdot (\frac{1}{b(x)} \cdot (-a(x) + v)) = v$, where v is the control input that stabilizes the linearized system $y^{(4)} = v$. After this variable change using the diffeomorphism, a new, linear system is defined with the following states:

$$z_1 = y, z_2 = \dot{y}, z_3 = \ddot{y}, z_4 = \dddot{y} \tag{11}$$

Then we have $y^{(4)} = v$ and so the I/O linearized system's state-space representation is:

$$\Sigma_{I/O} : \begin{cases} \dot{z} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \cdot z + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \cdot v =: A_{I/O} \cdot z + B_{I/O} \cdot v \\ y = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \cdot z =: C_{I/O} \cdot z \end{cases} \tag{12}$$

In practice, 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 such as state feedback where the gains are determined using pole placement. Here, the value to track is $r = 0$.
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.

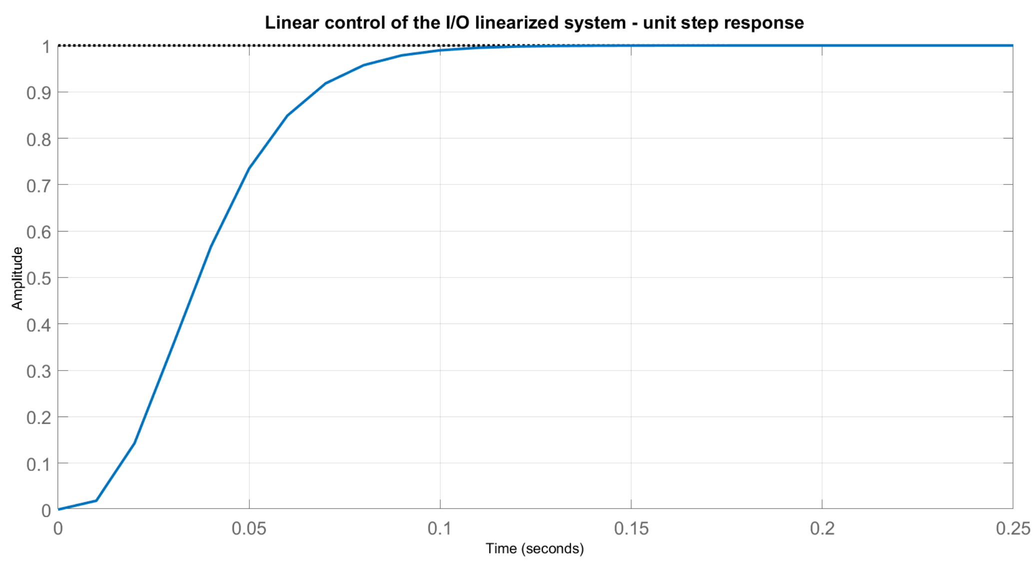

Figure 2: Simulation of the closed-loop linearized system.

5 Validation by simulation

We have constructed a Simulink model, as shown. It consists of two feedback loops. The inner one (the upper loop in the figure below) 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 in the figure below) is for v to stabilize the linearized system.

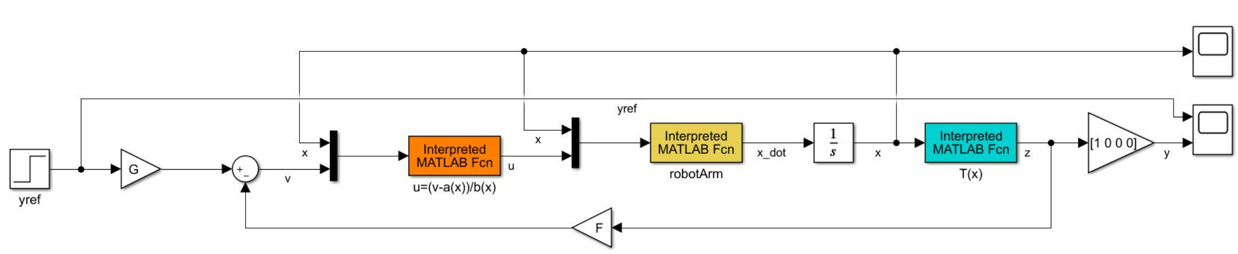

Figure 3: Simulation of the controlled system.

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

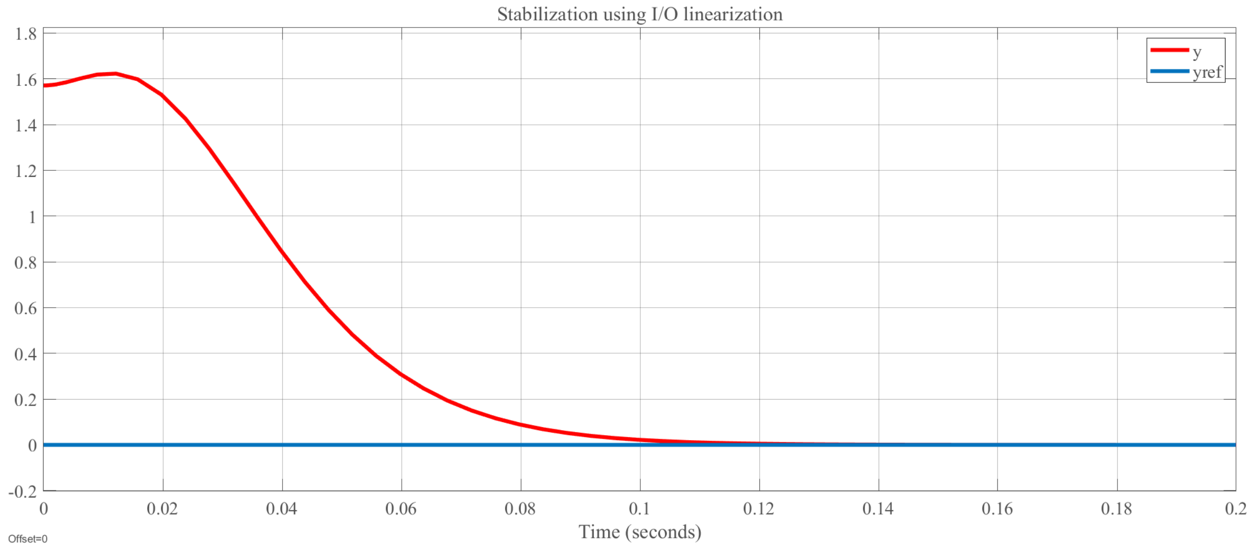

Figure 4: Stabilization using I/O linearization simulation results.

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

6 Extensions

6.1 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. That means the stabilization using the I/O linearization approach works in this case, which is coherent.

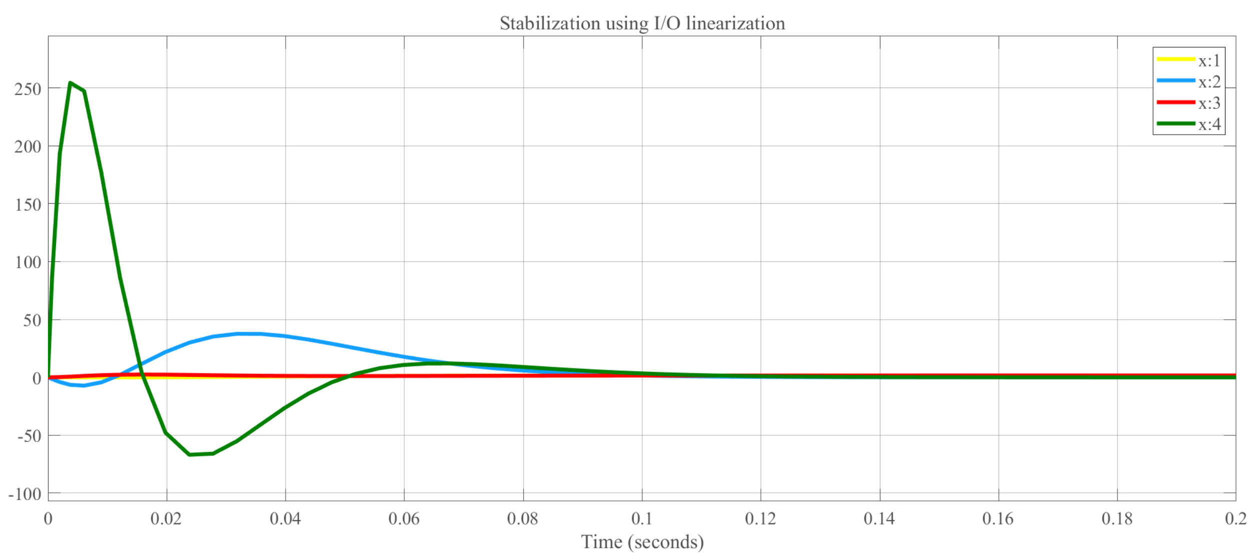

Figure 5: Stabilization using I/O linearization stability analysis.

If we have divergence in the system's states in some cases, it means that internal stability is not guaranteed, and therefore we need to apply I/O linearization with input/state (I/S) linearization.

6.2 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. Simulation results are as follows:

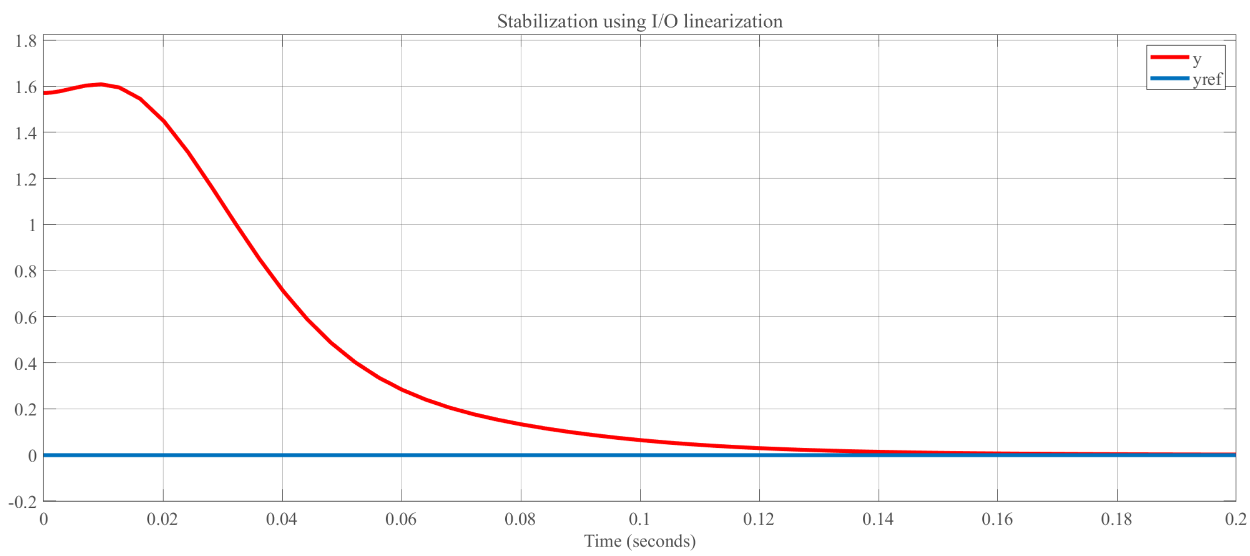

Figure 6: Stabilization using I/O linearization robustness analysis (output).

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.

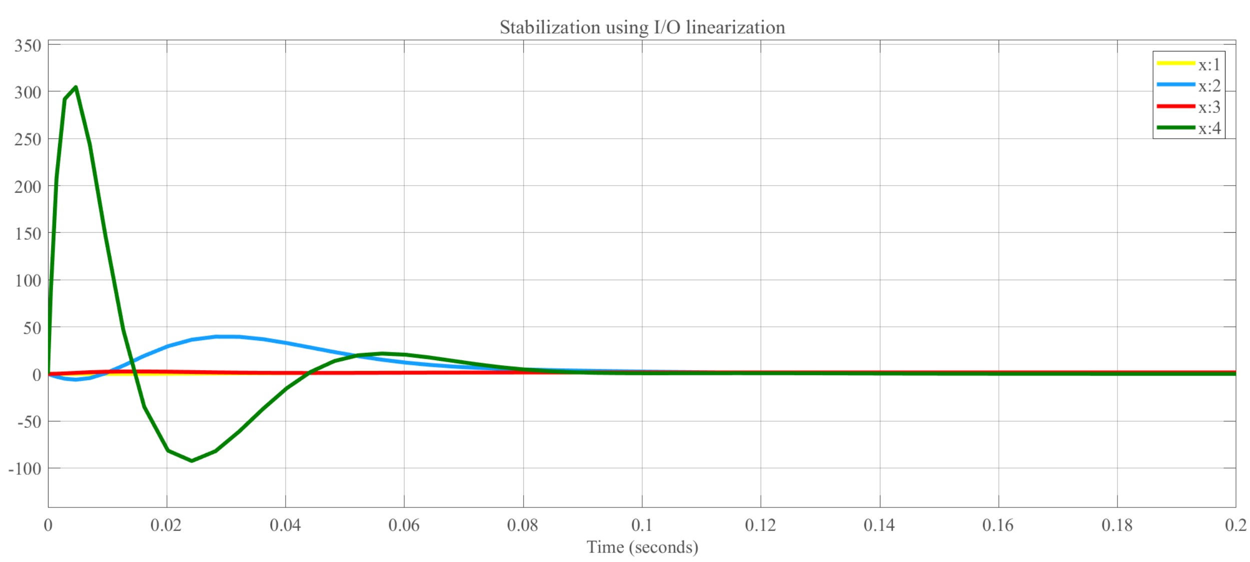

Figure 7: Stabilization using I/O linearization robustness analysis (internal stability).

We also see that internal stability is still guaranteed in this case.

6.3 Disturbance rejection

We can apply integral control which can reject disturbances (and hopefully improve robustness as in the previous lab). Let us denote the integral part:

$$p(t) = \int_0^t (r(\tau) - y(\tau))\, d\tau \tag{13}$$

The extended system is:

$$\Sigma_{I/Oext}: \begin{cases} \begin{bmatrix} \dot{z} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} A_{I/O} & 0 \\ -C_{I/O} & 0 \end{bmatrix} \cdot \begin{bmatrix} z \\ p \end{bmatrix} + \begin{bmatrix} B_{I/O} \\ 0 \end{bmatrix} \cdot v + \begin{bmatrix} 0 \\ I \end{bmatrix} \cdot r + \begin{bmatrix} E \\ 0 \end{bmatrix} \cdot d \\ y = \begin{bmatrix} C_{I/O} & 0 \end{bmatrix} \cdot \begin{bmatrix} z \\ p \end{bmatrix} \end{cases} \tag{14}$$

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

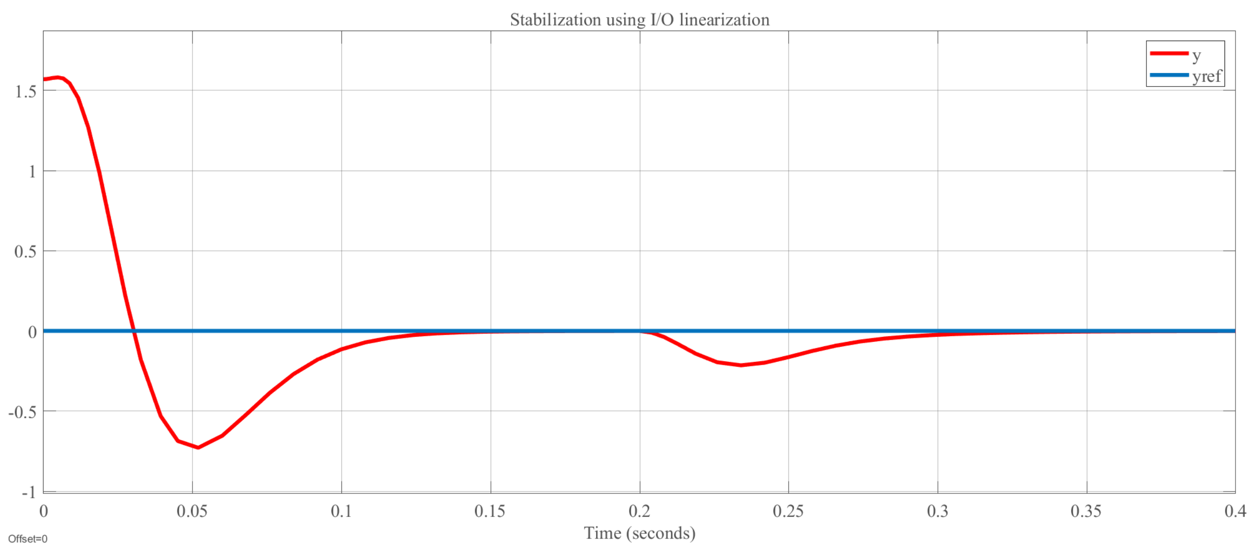

Figure 8: Integral control using I/O linearization.

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

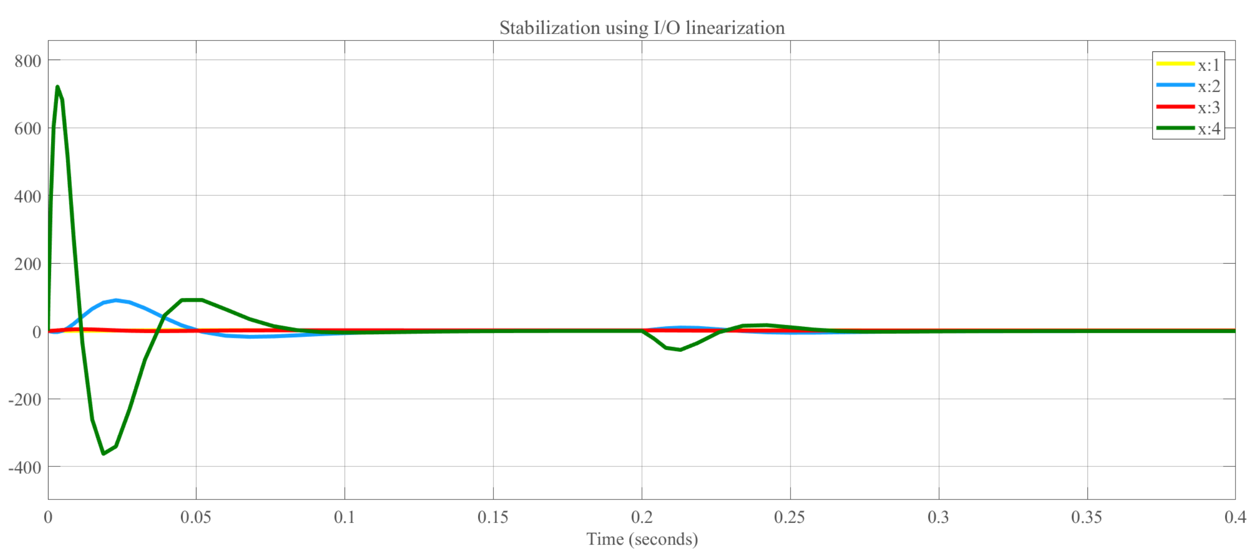

Figure 9: Integral control using I/O linearization (internal stability analysis).

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, and the results are almost exactly the same:

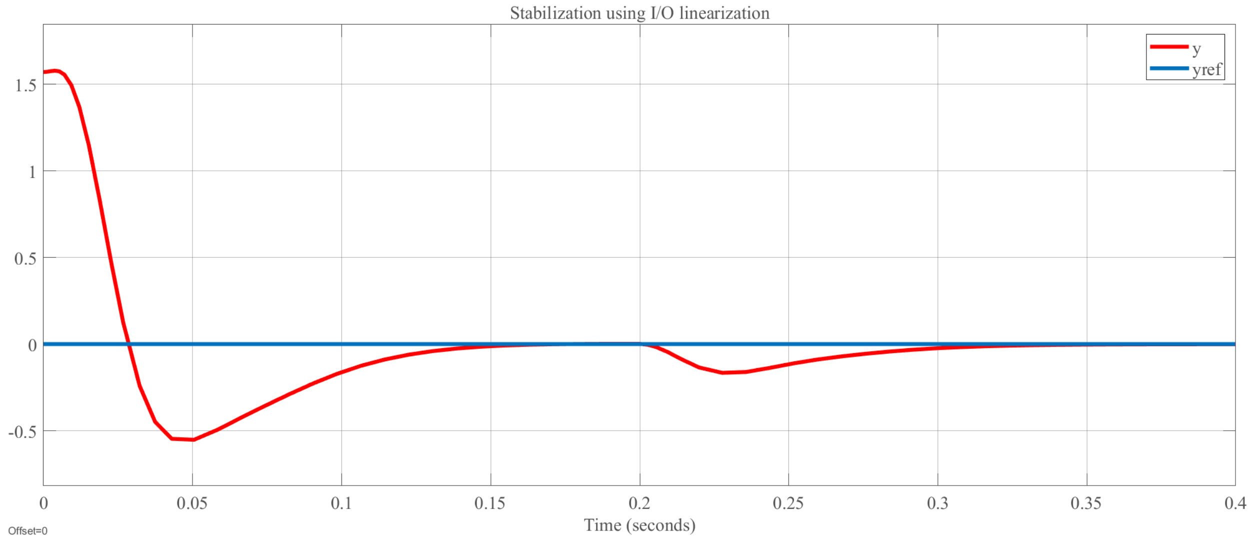

Figure 10: Integral control using I/O linearization robustness analysis (output).

We see that unlike state-feedback control, 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.

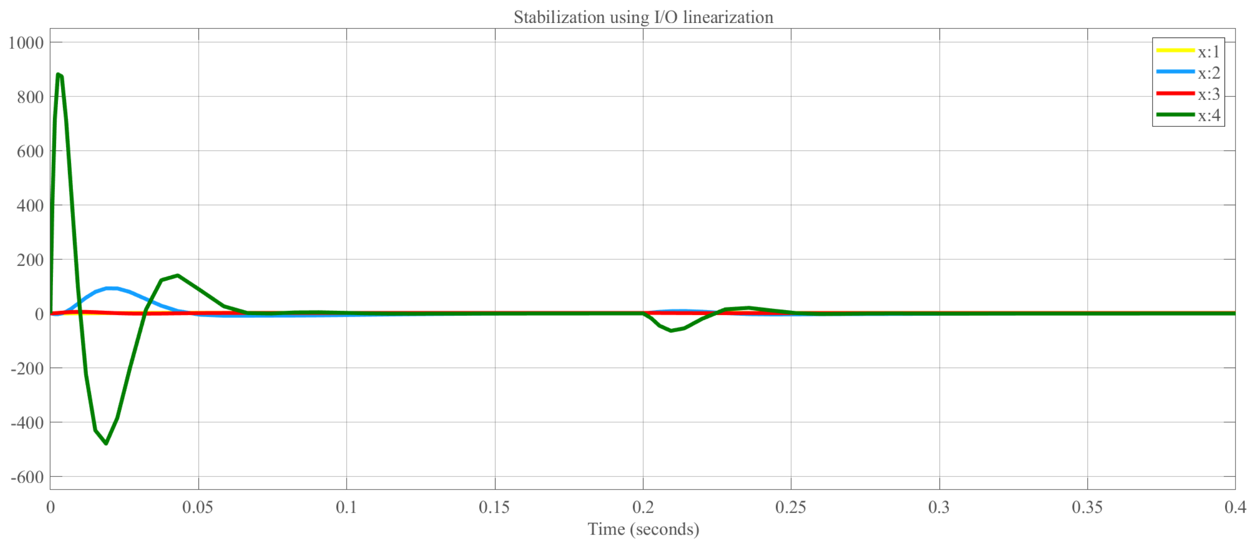

Figure 11: Integral control using I/O linearization robustness analysis (internal stability).

7 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.