State-Dependent Riccati Equation

The 19 joints of the hand are driven by 38 tendons that have a nonlinear stiffness characteristic. It means that the motor displacement required to adjust the tendon force depends on the current force. As demonstrated previously, the gain scheduling method is effective to assign the poles (of the pointwise linear system) but does not account for the input command magnitude. The optimal control method, that leads to the ARE in the case of a linear system, is able to account for the cost of the state error and the input amplitude. However, its genuine form is limited to linear problems. The optimization problem, that is solved relatively easily in the case of a linear system, is not anymore trivial to solve in the presence of nonlinearities. The exact solution of an optimal control problem is obtained by solving the Hamilton-Jacobi-Bell (HJB) equation given by

$$V(u(t)) = \int_0^T C(\boldsymbol{x}(t), u(t))dt + D(\boldsymbol{x}(T)), \tag{14.1}$$

where $\boldsymbol{x} \in \mathbb{R}^n, n \in \mathbb{N}$ is the state vector. The running state cost and the terminal state cost are denoted $C \in \mathbb{R}$ (resp. $D \in \mathbb{R}$). The functional to be minimized by the choice of the input function $u(t) \in \mathbb{R}, t \in [0, T], T > 0$ is represented by $V(u(t) \in \mathbb{R}$. Direct methods to solve the optimal control problems are reported as early as in 1959, in [87]. It has been applied to solve offline optimization problem such as space shuttle trajectory, ship maneuver or, more recently, throwing problem [88]. A result of optimal control due to Pontryagrin [86] is that in many cases bang-bang control is the solution (saturated maximum/minimum control input). However only a limited number of forms can be solved analytically. One must resort to numerical methods for the other cases, nonetheless their form can give further insights on the most efficient numerical techniques to be employed. Unfortunately, they require forward and backward integrations and, in general, are extremely expensive to compute. Especially, they are generally for real-time or online application.

An intermediate way between the linear optimal control, with the ARE, and the optimal nonlinear control, with the HBJ equation, has been proposed around 1962 by Pearson under the name of State Dependent Riccati Equation (SDRE) [89]. It has been expended by Wernly [90] and popularized by Cloutier [91–95]. The method is an intuitive extension of the ARE, applied to a pointwize linearized system. The existence of a SDRE stabilizing feedback is discussed in [96]. The method offers only limited theoretical results for global stability but proved to be effective in practice. More details can be found in the extensive survey [97].

In a first section the method is presented with a generic example based on [91]. The second section applies the method to two problems: the control

of the tendon force, similar to the gain scheduling example, and the control of a single joint with one motor and a nonlinear spring. The third section evaluates the controller with the help of simulations. Finally, section four discusses the results.

14.1 State Dependant Riccati Equations

Considering a nonlinear multi-variable system,

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{u} \ , \tag{14.2}$$

where the state dimension is $n \in \mathbb{N}$ and $\boldsymbol{x} \in \mathbb{R}^n$ is the state vector. A nonlinear function of the state variables, that is assumed to be sufficiently smooth, is denoted $\boldsymbol{f}(\boldsymbol{x}) \in \mathbb{R}^n$. The control input is $\boldsymbol{u} \in \mathbb{R}^n$. It is possible to write (14.2) in a pseudo-linear form, also referred to as the pointwise linear form, as

$$\dot{\boldsymbol{x}} = \boldsymbol{A}_{\mathrm{k}}\boldsymbol{x} + \boldsymbol{B}_{\mathrm{k}}\boldsymbol{u} \ , \tag{14.3}$$

One pointwise linearized form and the associated input for a given factorization $\Xi_{\mathrm{k}}, \mathrm{k} \in \mathbb{N}$ are denoted $\boldsymbol{A}_{\mathrm{k}} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{B}_{\mathrm{k}} \in \mathbb{R}^{n\times m}$. It should be noted that, excepted the case $n = 1$, there exists an infinite number of factorization Ξ_{k} and its associated matrices $(\boldsymbol{A}_{\mathrm{k}}, \boldsymbol{B}_{\mathrm{k}})$. Once a factorization has been selected, the ARE can be used to select the optimal gains. According to Chapter 13, the state feedback gains are selected as

$$\boldsymbol{K} = \boldsymbol{R}^{-1}\boldsymbol{B}_{\mathrm{k}}^T\boldsymbol{S} \ , \tag{14.4}$$

where $\boldsymbol{R}(t) \in \mathbb{R}^{m\times m}$ is a positive definite cost matrix for the input, $\boldsymbol{B}_{\mathrm{k}}$ is the input matrix and $\boldsymbol{S}(t)$ is one solution of the Riccati equation defined by

$$\boldsymbol{S}\boldsymbol{A}_{\mathrm{k}} + \boldsymbol{A}_{\mathrm{k}}^T\boldsymbol{S} + \boldsymbol{S}\boldsymbol{B}_{\mathrm{k}}\boldsymbol{R}^{-1}\boldsymbol{B}_{\mathrm{k}}^T\boldsymbol{S} - \boldsymbol{Q} = 0 \ , \tag{14.5}$$

where $\boldsymbol{Q} \in \mathbb{R}^{n\times n}$ (resp. $\boldsymbol{R} \in \mathbb{R}^{m\times m}$) is the state error cost (resp. the control input cost) both positive definite. The closed-loop system is

$$\dot{\boldsymbol{x}} = \boldsymbol{A}_{\mathrm{k}}\boldsymbol{x} + \boldsymbol{B}_{\mathrm{k}}^T\boldsymbol{R}^{-1}\boldsymbol{B}_{\mathrm{k}}^T\boldsymbol{S}\boldsymbol{x} \ . \tag{14.6}$$

Under the assumption that all quantities are continuous and continuously differentiable (C^1), and by construction of $\boldsymbol{S}$, the closed-loop system of (14.6) is Hurwitz, therefore locally asymptotically stable.

14.2 Applications

In this section the state-dependent Riccati equation (SDRE) is derived for two particular cases. First, the force regulation of the tendon forces when

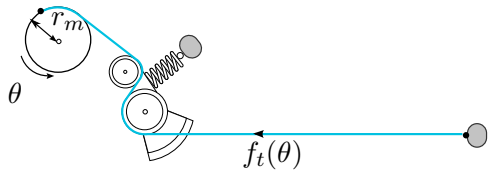

Figure 14.1: Model for the tendon force controller. The link is assumed to be fixed, thus the tendon force only depends on the motor position.

considering the joint fixed is studied. It is the problem that was motivating the gain scheduling method of Chapter 10. Second, a single nonlinear flexible joint model driven by a single motor is proposed. The second problem is a simplification of the real case problem that allows to understand the effect of the control.

14.2.1 Tendon force controller

The model comprises a motor, a spring element and a tendon (cf. Fig. 16.1). The tendon is attached to a fixed reference (grounded). The control objective is to regulate the tendon force ($f_\mathrm{t} \in \mathbb{R}$), measured by the spring lever, by adjusting the position $\theta \in \mathbb{R}$ of the motor with the torque input $u \in \mathbb{R}$. The dynamic equation of the system is

$$B_\theta \ddot{\theta} = -f_\mathrm{t}(\theta) + u \ , \tag{14.7}$$

where $B_\theta \in \mathbb{R}$ is the motor inertia (w. r. t. the motor acceleration), $\theta \in \mathbb{R}$ and $u \in \mathbb{R}$ are classically the motor position and the torque input. The tendon force depending on the motor position is denoted $f_\mathrm{t}(\theta) = \varphi(\theta)$. It is important to note that for the following analysis, the function $\varphi(\theta)$ is required to be at least C^2 w. r. t. θ. To apply the SDRE method it is first necessary to establish the pointwise linear form. One possible solution is given by equation (14.8). The linearization w. r. t. to θ is

$$B_\theta \ddot{\theta} = -f_\mathrm{t}(\theta_0) - \frac{\partial f_\mathrm{t}(\theta)}{\partial \theta}|_{\theta_0}(\theta - \theta_0) + u \ , \tag{14.8}$$

where θ_0 is the linearization point. Adding a feedforward term to the command $u = f_\mathrm{t}(\theta_0) + v$ shifts the equilibrium to the origin. Introducing the error $\xi = \theta - \theta_0$ leads to the matrix form,

$$\dot{\boldsymbol{x}} = \boldsymbol{A}(\boldsymbol{x})\boldsymbol{x} + \boldsymbol{B}v = \begin{bmatrix} 0 & 1 \\ \frac{1}{B_\theta}\frac{\partial f_\mathrm{t}}{\partial \theta}|_{f_0} & 0 \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 0 \\ \frac{1}{B_\theta} \end{bmatrix} v \ , \tag{14.9}$$

where $\boldsymbol{x} = [\xi, \dot{\xi}]$. It is also possible to linearize w. r. t. to the tendon forces.

14.2.2 Flexible joint model

A generic flexible joint model is depicted in Fig. 14.2. As mentioned previously, there exists an infinite number of factorizations but the method used to establish the dynamic equations naturally leads to a factorization by the stiffness of the tendons.

$$\boldsymbol{A}(\boldsymbol{x}) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k(x_1-x_3)}{m} & -\frac{d_q}{m} & \frac{k(x_1-x_3)}{m} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{k(x_1-x_3)}{b} & 0 & -\frac{k(x_1-x_3)}{b} & -\frac{d_\theta}{b} \end{bmatrix}, \tag{14.10}$$

where $n \in \mathbb{N}$ is the state dimension, $\boldsymbol{x} \in \mathbb{R}^n$ is the state vector defined as $\boldsymbol{x} = [q, \dot{q}, \theta, \dot{\theta}]$. The control input is denoted $u \in \mathbb{R}$. The joint stiffness is represented by $k(x_1 - x_3) \in \mathbb{R}$. The viscous frictional torque of the joint (resp. motor) are denoted d_q (resp. d_θ). Finally, the inertias of the link and the motor are $m \in \mathbb{R}$ and $b \in \mathbb{R}$. One pointwise linear form is given by

$$\dot{\boldsymbol{x}} = \boldsymbol{A}(\boldsymbol{x})\boldsymbol{x} + \boldsymbol{B}u \ , \tag{14.11}$$

with

$$\boldsymbol{A}(\boldsymbol{x}) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k(x_1-x_3)}{m} & -\frac{d_\mathrm{q}}{m} & \frac{k(x_1-x_3)}{m} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{k(x_1-x_3)}{b} & 0 & -\frac{k(x_1-x_3)}{b} & -\frac{d_\theta}{b} \end{bmatrix}$$

and

$$\boldsymbol{B}(\boldsymbol{x}) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{b} \end{bmatrix} \ .$$

The control input is a state feedback defined by

$$u = -\boldsymbol{K}^T\boldsymbol{x} \ , \tag{14.12}$$

where the gain vector $\boldsymbol{K} \in \mathbb{R}^m$ is given by the SDRE method, ie. $\boldsymbol{K} = \boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{P}$. The matrix $\boldsymbol{P} \in \mathbb{R}^{4\times4}$ being the solution of the Riccati equation (14.5).

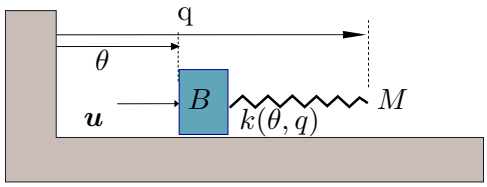

Figure 14.2: Mass spring damper system in the case of a flexible joint model

14.3 Simulation and experiments

The two cases derived above are verified by simulations. First, the tendon force controller (as depicted in Fig. 16.1) is evaluated. The flexible joint model (cf. Fig. 14.2) is verified in a second step.

14.3.1 Application to a tendon force controller

The simulations are performed using the numerical solver ode23t from MATLAB®, with the parameters of Table 14.1.

Table 14.1: Simulation parameters for the tendon controller

Symbol	Value	Units
B_θ	$2e-3$	$kg.m^2$
Q	$\begin{bmatrix} 1 & 0 \\ 0 & 0.0001 \end{bmatrix}$	
R	0.000001	

The simulation results are depicted in Fig. 14.3. First an initial desired force step from 0 N to 40 N is commanded. Then, a smaller adjustment is made to reach 45 N. The differences between the two controllers are hardly visible. The main reason is that the plant equations are not changing as much as one might expect. The change in stiffness of the real mechanism during the experiment only results in a minimal change of the optimal gains.

14.3.2 Application to a joint controller

The parameters of Table 14.2 are used for the joint simulation. The resulting

Table 14.2: Simulation parameters for the joint controller

Symbol	Value	Units
B_θ	$2e-3$	$kg.m^2$
M	$7.2e-7$	$kg.m^2$
Q	$\begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.01 \end{bmatrix}$	$N.A$
R	0.0001	$N.A$

link trajectories are reported in Fig. 14.4. The improvement is not noticeable in the case of a free motion. Indeed, the low link inertia does not create a significant dynamic load, thus the stiffness change is extremely small.

However, when a load is applied externally, the stiffness change is visible, as depicted in Fig. 14.5. In such a case, the SDRE method is able to modify the gains to account for the modified plant equations.

14.4 Discussion

This chapter has presented an extension of the optimal linear control method of the previous chapter. The method, called the SDRE method, has been appreciated in the optimal control research groups because of its good practical results. The first section described the general idea of the method on an abstract example. Since the method is based on the pointwise linear form of a system, the second section transformed the system dynamics into the proper form. The third section applied the method to two different systems and proposed several simulations. It was shown that for the tendon control problem, the method only marginally contributes to improve the behavior mainly because the optimal gains do not change significantly. On the contrary, the improvements were visible in the case of a flexible joint. Nonetheless, the gain designed for the nominal load were also satisfying, especially in case of free motion. Nonetheless, the method is relatively easy to use and the optimal control community is very active in developing the supporting theory. From the implementation point of view it is very similar to the gain scheduling method. Some first analysis and simulations[1] are showing that the ARE gains can be computed at a lower rate than the control loop without significant effect on the resulting behavior. Practically, a rate of about 200Hz is sufficient. The method is theoretically limited to a local analysis. Thus, the following chapters are focusing on using global, nonlinear controller designs. The ARE and SDRE method are reused in the last chapters as a mean to select the best, optimal in a sense, gains for the backstepping controller.

[1] not reported in this thesis

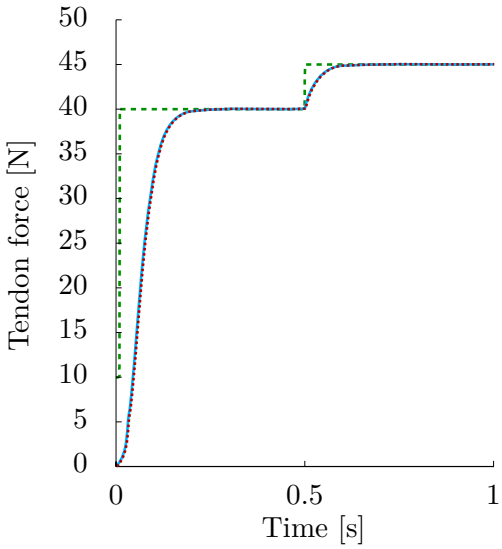

Figure 14.3: Simulation: Comparison between the SDRE controller and the fixed gains controller for a tendon force control problem. The green/dashed line is the desired tendon force. The light blue/solid and red/dotted lines represent the tendon forces.

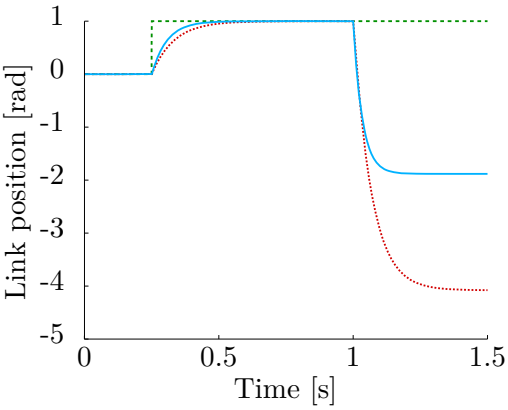

Figure 14.4: Simulation: Comparison between the SDRE controller and the fixed gains controller for a link positioning task. The green/dashed line is the desired link position. The light blue/solid and red/dotted lines depict the link position with the SDRE controller and the fixed gains controller.

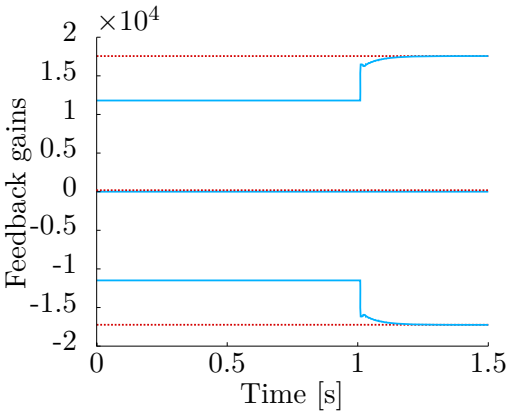

Figure 14.5: Simulation: Comparison between the SDRE controller and the fixed gains controller for a link positioning task. The light blue/solid lines depict the gains with the SDRE controller. The gains of the fixed gains controller are represented in red/dotted lines.

15 Backstepping

The backstepping design procedure is a design method for nonlinear controllers by Kokotovic in the 90's. It is a recursive method for strict feedback system. In each step, the derivative of the previous error is compensated and well known stabilizing reference is applied to the system. Then, the error introduced is propagated to the next level and the method is applied again. The backstepping procedure has been described in [68, 85, 134] and applied to a large variety of problems. Only little work deals with the practical implementation of the backstepping method and most of the papers are only presenting simulation results. Its main limitations are the need for high order derivatives and the fast growth of the expression, known as the *complexity due explosion of terms* which is a direct consequence of the recursivity of the method. Althought the procedure only requires positive definiteness of the gain matrices, it should also be noted that they do not always have an intuitive interpretation and the manual tuning of the numerous gains can be tedious for complex systems.

Nonetheless, it is a purely nonlinear method that does not require the previous assumptions on the system (eg. cascaded system). Moreover, the designed controller is stable by design as long as the gain matrices are positive definite. This allows a great freedom in the choice of the gains. The main contribution of this chapter is to provide experimental validation of the controllers, derive the backstepping equation in the case of a nonlinear flexible joint and extend the single motor controller to an antagonistic controller.

In the first section, an example of the backstepping method, inspired by [84, p.489] is proposed. The reader familiar with the backstepping method can safely skip the section. The second section applies the backstepping control method to two cases that have a structure similar to the one of the real system. More precisely, a backstepping controller in the case of a constant stiffness (resp. variable stiffness) single flexible joint is derived. Simulations are performed to evaluate the results. The controllers are state controllers and are not suitable for interaction with the grasped object. A *soft* controller is needed to perform stable grasps in the presence of inaccuracies. Therefore, in the third section, the control law for an impedance controller is derived. Because of its importance, simulations and experimental results are presented on a single joint actuated by one motor with a linear spring. It is verified numerically and experimentally that the controller is behaving like an impedance controller. The fourth section extends the single joint controller to the nonlinear case. Unsurprisingly, one of the main condition for the existence of the control law is to have a strictly convex force/displacement characteristic of the spring. The fifth section extends the single joint/single motor controller to an antagonistic joint actuation.

Finally, the backstepping method is applied to the equations of the real system. Simulation and experimental results are reported.

15.1 Concept

This section presents the concept of the integrator backstepping method. In the first section the equation of control are derived on a simple example. In a second part, some simulation results are reported to help the reader to understand the behavior of the controller. It is a very basic introduction to the backstepping concept and can safely be skipped.

15.1.1 Controller design

Consider the dynamic system described by Eq. (15.1), where $(x_1, x_2) \in \mathbb{R}^2$ are the state variables and $u \in \mathbb{R}$ is the control input. It is assumed that all quantities are directly measurable and that all functions are sufficiently smooth.

$$\begin{cases} \dot{x}_1 &= -{x_1}^3 + {x_1}^2 + x_2 \\ \dot{x}_2 &= u \end{cases} \tag{15.1}$$

If $\bar{x}_2 = x_2$ is considered as a virtual input for the (15.1), an exponentially stabilizing control input is

$$\bar{x}_2 = -{x_1}^2 - k_1 x_1 \ , \tag{15.2}$$

where $k_1 \in \mathbb{R}^+$ is a gain used to accelerate the convergence and $\bar{x}_2$ is the reference input. It is proved using the Lyapunov function $V_1(x_1) = \frac{1}{2}x_1^2$. Taking the time derivative of V_1 along the solutions of the first equation of (15.1) one obtains

$$\dot{V}_1(x_1) \ = \ x_1(-{x_1}^3 - k_1 x_1) = -{x_1}^4 - k_1 x_1^2 \ . \tag{15.3}$$

However, it is not possible to track exactly the reference input $\bar{x}_2$. Defining $z_2 = x_2 - \bar{x}_2$, the system (15.1), is transformed in

$$\dot{x}_1 \ = \ -{x_1}^3 - k_1 x_1 - x_1 \tag{15.4}$$

$$\dot{z}_2 \ = \ u - \dot{\bar{x}}_2 \tag{15.5}$$

The second equation is stabilized by $u = \dot{\bar{x}}_2 - k_2 z_2$, where $k_2 \in \mathbb{R}^+$ is a feedback gain used to accelerate the convergence of the system. The global asymptotic stability is demonstrated using the Lyapunov function $V_2(x_1, z_1) = \frac{1}{2}x_1^2 + \frac{1}{2}z_2^2$ along the trajectories. From Equation (15.6), simply replacing the expressions gives

$$\dot{V}_2(x_1, z_2) = x_1\dot{x}_1 + z_2\dot{z}_2 \ , \tag{15.6}$$

simplifying and grouping the terms leads to

$$\dot{V}_2(z_1, z_2) = x_1(-{x_1}^3 - k_1 x_1 - x_1) + z_2(u - \dot{\bar{x}}_2) \ , \tag{15.7}$$

and

$$\dot{V}_2(x_1, z_2) = (-{x_1}^4 - k_1 {x_1}^2 - {x_1}^2) + z_2(-k_2 z_2) \ . \tag{15.8}$$

Finally, one obtains

$$\dot{V}_2(x_1, z_2) = -{x_1}^4 - k_1 {x_1}^2 - {x_1}^2 - k_2 {z_2}^2 \ . \tag{15.9}$$

The final expression of u is obtained by going back to the original coordinates and is reported in (15.10).

$$u = \dot{\bar{x}}_2 - k_2 z_2 = (-2x_1\dot{x_1} - k_1\dot{x_1}) - k_2(x_2 + {x_1}^2 + k_1 x_1) \ , \tag{15.10}$$

where $(k_1, k_2) \in (\mathbb{R}^+ \times \mathbb{R}^+)$ are two feedback gains used to accelerate the convergence of the system.

It should be noted that the presence of the derivative of the reference control signal is the main caracteristic of the backstepping methodology. In each step of the method, the derivative of the reference control is derived once more. This leads to the phenomenon refered to as the *complexity due to the explosion of terms*. Consequently, the backstepping method, although very sound mathematically, can be delicate to apply to high order systems (unless the derivatives of all quantities are available). An interesting property of the backstepping method is that it is not necessary to cancel the *good* nonlinearities (such as $-x_1^2$ in the example). It allows to reduce the control effort w. r. t. the feedback linearization method that systematically cancels the nonlinearities.

15.1.2 Simulations

To analyze the behavior of the controller derived in the previous section several simulations are performed. The system defined by (15.1) together with the control law of (15.10) is simulated using MATLAB®. The feedback gains (k_1, k_2) are modified and the resulting trajectories are reported. In Figure 15.1 the feedback gains are modified and the resulting trajectory for $x_1(t), t \in [1 \dots 10]$ are plotted. As expected, the trajectory are converging to the origin for any combination of $(k_1, k_2) \in (\mathbb{R}^+)^2$. The higher the gains are, the faster the system is converging. Figure 15.2, 15.3 and 15.4 report the phase diagram of x_1 for three different gain combinations and varying initial conditions. All combinations are converging toward the origin, thus confirming that the controller is effective. The convergence trajectory is changing according to the choice of the feedback gains. However, increasing the gains of the outmost layer does not ensure that the convergence will be

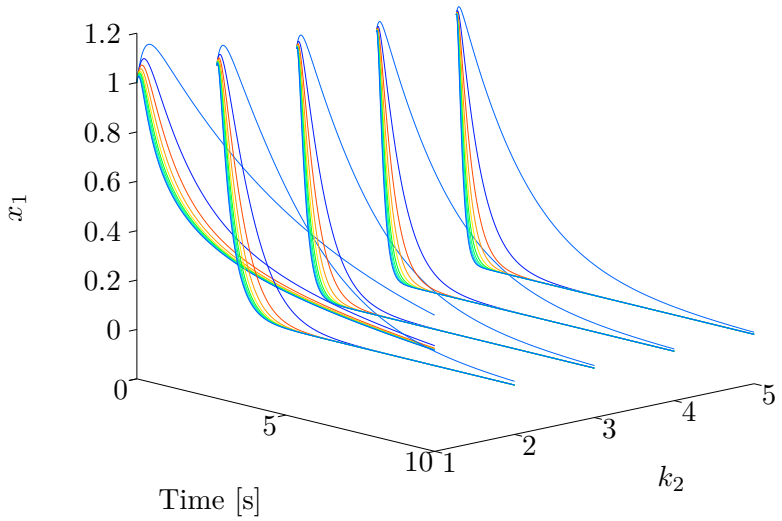

Figure 15.1: Simulation results: x_1 trajectories obtained for different values of k_1 and k_2. Slice are for $k_1 \in [0.01, 2.0, 5.0, 7.5, 10.0]$. Colors are for $k_2 \in [0.5, 1.55, 2.61, 3.67, 4.72, 5.78, 6.83, 7.89, 8.94, 10.0]$. Initial conditions are $x_1 = 1, x_2 = 1$.

faster. Indeed, if x_2 is not regulated to the desired value, no value of x_1 can improve the convergence rate. Moreover, the measurement noise of the low layers $x_2, x_3, \ldots$ is likely to be increasingly large, thought limiting the gains in the real implementation.

15.1.3 Conclusion

This section explained how the backstepping method works on a two degrees of freedom example. The stability of the close loop system was numerically demonstrated through a combination of numerical simulations (different initial conditions and different gains).

15.2 Single flexible joint: position controller

In this section, the backstepping methodology is applied to a single joint driven by one motor with a linear spring (i. e. the spring elongation has no influence on its stiffness). The spring stiffness is given by $K = \frac{\partial f(x)}{\partial x}$, where $f \in \mathbb{R}$ and $x \in \mathbb{R}$ are the spring force and the spring elongation (w. r. t. its default length). For a linear spring, the stiffness is constant i. e. $\frac{\partial K}{\partial x} = 0$.

15.2.1 Model

The mechanical model of the flexible joint is depicted in Figure 15.5 and the corresponding differential equations are reported in (15.11) and (15.12).

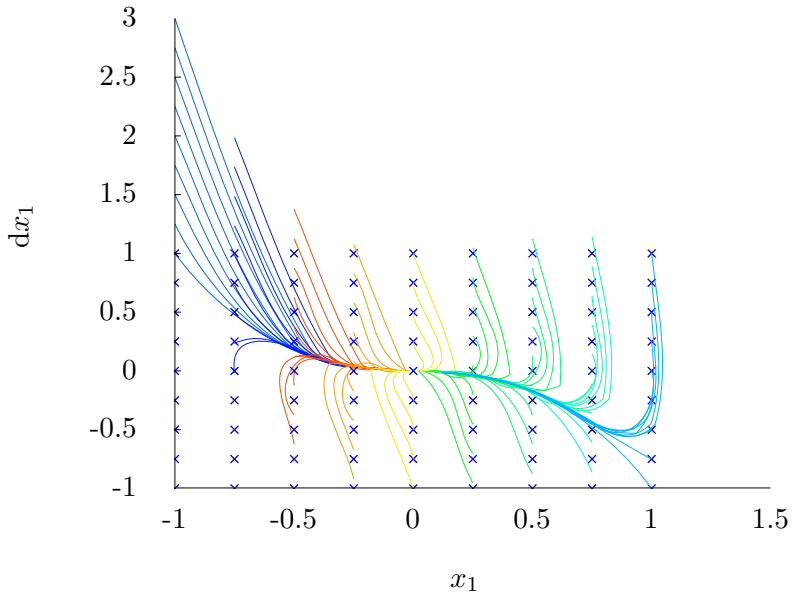

Figure 15.2: Simulation results: solution trajectories for different initial conditions represented in a phase diagram of $\dot{x}_1(x_1)$. Feedback gains are $k_1 = 0.1, k_2 = 5$. Initial conditions are marked by a cross symbol.

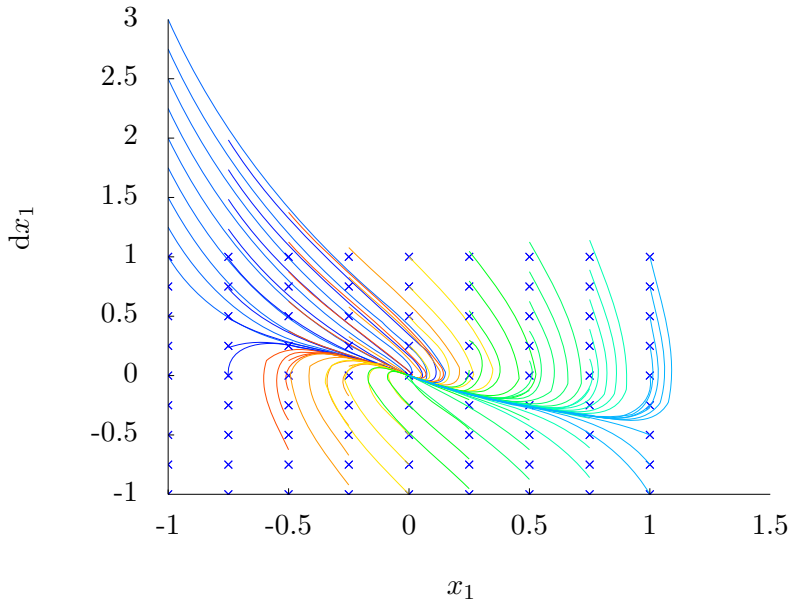

Figure 15.3: Simulation results: solution trajectories for different initial conditions represented in a phase diagram of $\dot{x}_1(x_1)$. Feedback gains are $k_1 = 1, k_2 = 1$. Initial conditions are marked by a cross symbol.

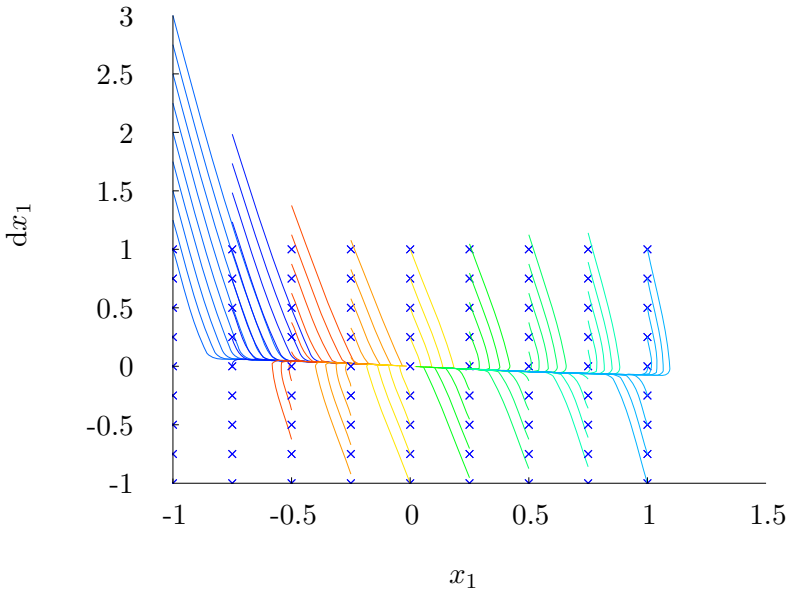

Figure 15.4: Simulation results: solution trajectories for different initial conditions represented in a phase diagram of $\dot{x}_1(x_1)$. Feedback gains are $k_1 = 5, k_2 = 0.1$. Initial conditions are marked by a cross symbol.

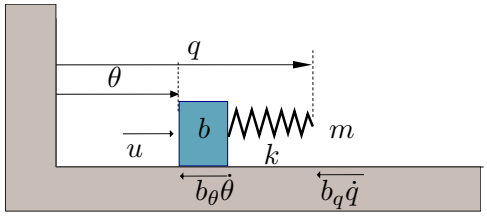

Figure 15.5: Double spring mass damper system in the case of a flexible joint model.

$$m\ddot{q} + b_q\dot{q} = -f(\theta, q) \tag{15.11}$$

$$b\ddot{\theta} + b_\theta\dot{\theta} = f(\theta, q) + u \tag{15.12}$$

$\theta \in \mathbb{R}, q \in \mathbb{R}$ are the motor position and link position. The link mass and the motor mass are denoted $m(q) \in \mathbb{R}, b \in \mathbb{R}$. The force generated by the elastic element is represented by $f(\theta, q) \in \mathbb{R}$. The input vector, that is, the motor force, is denoted $u \in \mathbb{R}$. Finally, b_q (resp. b_θ) is the friction force vector associated to the link (resp. motor). Neglecting the frictional terms to simplify the expression, the system described by (15.11) and (15.12) is written in a vector form as

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(u) \ , \tag{15.13}$$

where the state vector $\boldsymbol{x} \in \mathbb{R}^4$ is defined as

$$\boldsymbol{x} = \begin{bmatrix} q \\ \dot{q} \\ \theta \\ \dot{\theta} \end{bmatrix} \quad . \tag{15.14}$$

The vector-valued functions $\boldsymbol{f} : \mathbb{R}^4 \mapsto \mathbb{R}^4$ and $\boldsymbol{g} : \mathbb{R} \mapsto \mathbb{R}^4$ are

$$\boldsymbol{f} = \begin{bmatrix} x_2 \\ -\dfrac{f(x_1 - x_3)}{m} \\ x_4 \\ \dfrac{f(x_1 - x_3)}{m} \end{bmatrix} \quad \text{and} \quad \boldsymbol{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \dfrac{1}{b} \end{bmatrix} \quad . \tag{15.15}$$

15.2.2 Strict Feedback Form

In order to apply the integrator backstepping methodology it is required to transform the system into a strict feedback form. That is, the i^{th} differential equation (corresponding to the i^{th} state variable) is only allowed to depend on the variables up to $i-1$. Indeed, one variable must disappear after each backstepping step otherwise the method would not converge to an expression for u. Graphically, the arguments of state function $\boldsymbol{f}$ must be located in a triangle with a line above the diagonal, as depicted in Fig. 15.6. Similarily, the arguments of the input function $\boldsymbol{g}$ should be non zero on the last line.

$$\begin{matrix} \neq 0 & 0 & 0 \\ & \neq 0 & 0 \\ & & \neq 0 \\ & & \end{matrix} \quad + \quad \begin{matrix} \\ \\ \\ \neq 0 \end{matrix} \quad u$$

Figure 15.6: Graphical representation of the state transition matrix of a system in strict feedback form.

This section constructs a new coordinate system in which the equations are in strict feedback form. In the case of a constant stiffness spring the spring torque is simply

$$\tau(\theta, q) = k(\theta - q), \tag{15.16}$$

where $K \in \mathbb{R}^{*+}$ is the spring stiffness. Therefore, it is possible to remove one variable (q or θ). Using θ from (15.12) and replacing it in (15.11) yields a fourth order differential equation on q

$$\frac{bm}{k} q^{(4)}(t) + (b+m) q^{(2)}(t) = u(t) \ , \tag{15.17}$$

where all quantities are defined as previously done. The system can even be written in the linear form $\dot{\boldsymbol{X}} = \boldsymbol{A}(\boldsymbol{X})\boldsymbol{X} + \boldsymbol{B}u$ where the state vector $\boldsymbol{X} \in \mathbb{R}^4$ is defined as

$$\boldsymbol{X} = \begin{bmatrix} q \\ \dot{q} \\ \ddot{q} \\ q^{(3)} \end{bmatrix}. \tag{15.18}$$

The state transition matrix $\boldsymbol{A} \in \mathbb{R}^{4\times 4}$ and the input vector are

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\dfrac{(b+m)k}{bm} & 0 \end{bmatrix} \quad \text{and} \quad \boldsymbol{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \dfrac{k}{bm} \end{bmatrix}. \tag{15.19}$$

The system is written in a strict feedback form and is ready for the application of the backstepping method.

15.2.3 Controller design

According to (15.19), defining the state vector $\boldsymbol{x} \in \mathbb{R}^4$ as $[x_1, x_2, x_3, x_4] = [q, \dot{q}, \ddot{q}, \dddot{q}]$ allows to write the system in the strict feedback form

$$\begin{cases} \dot{x}_1 &= & f_1(x_1) + g_1(x_1)x_2 \\ \dot{x}_2 &= & f_2(x_1, x_2) + g_2(x_1, x_2)x_3 \\ \dot{x}_3 &= & f_3(x_1, x_2, x_3) + g_3(x_1, x_2, x_3)x_4 \\ \dot{x}_4 &= & f_4(x_1, x_2, x_3, x_4) + g_4(x_1, x_2, x_3, x_4)u \end{cases}, \tag{15.20}$$

with

$$\begin{cases} f_1(x_1) &= & 0 \\ g_1(x_1) &= & 1 \\ f_2(x_1, x_2) &= & 0 \\ g_2(x_1, x_2) &= & 1 \\ f_3(x_1, x_2, x_3) &= & 0 \\ g_3(x_1, x_2, x_3) &= & 1 \\ f_4(x_1, x_2, x_3, x_4) &= & -\dfrac{k(b+m)}{bm}x_3 \\ g_4(x_1, x_2, x_3, x_4) &= & \dfrac{k}{bm} \end{cases}. \tag{15.21}$$

Remark : The choice of f_4 and g_4 could be changed to $f_4 = 0$ and $g_4 = 1$ by feedback linearization in a strict integrator form by $u = \dfrac{bm}{k}(\dfrac{k(b+m)}{bm}\ddot{q} + v)$. The new system could be $\dot{\boldsymbol{X}}\boldsymbol{A}_v\boldsymbol{X} + \boldsymbol{B}_v v$, with the state transition matrix

$\boldsymbol{A_v} \in \mathbb{R}^{4\times4}$ and the input vector defined as

$$\boldsymbol{A_v} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad \boldsymbol{B_v} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} . \tag{15.22}$$

However, the backstepping procedure naturally includes the feedback cancellation of f_4 and the scaling g_4 so it is not needed to perform the feedback linearization before designing the controller. The system is of order four and consequently four steps are needed to complete the integrator backstepping procedure. The following sections report the steps along with the stability proofs which helps understanding the procedure.

First equation The arguments of the functions are removed for clarity. According to the state matrix defined (15.19), the system is given by

$$\dot{x}_1 = x_2 \tag{15.23}$$
$$\dot{x}_2 = x_3 \tag{15.24}$$
$$\dot{x}_3 = x_4 \tag{15.25}$$
$$\dot{x}_4 = f_4 + g_4 u \tag{15.26}$$

Considering only (15.23) and taking $\bar{x}_2 = x_2$ as a virtual input, the scalar system is stabilized by

$$\bar{x}_2 = -k_1 x_1, \tag{15.27}$$

where $k_1 \in \mathbb{R}^{*+}$. The stability is proved using the Lyapunov function $V(x_1) = \frac{1}{2}x_1^2$. The time derivative of V_1 along the solution is,

$$\dot{V}(x_1) = x_1\dot{x}_1 = -k_1 x_1^2 \tag{15.28}$$

which, after invoking the LaSalle theorem, concludes the proof.

Second equation The ideal control input of the first equation cannot be exactly tracked because the system has internal dynamics (the input goes through several integrators). Therefore, a tracking error z_2 is defined as $z_2 = x_2 - \bar{x}_2$ and propagated in the system. Eliminating x_2 in the original system leads to

$$\begin{cases} \dot{x}_1 = (\bar{x}_2 + z_2) \\ \dot{z}_2 = x_3 - \dot{\bar{x}}_2 \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = f_4 + g_4 u \end{cases} . \tag{15.29}$$

Replacing $\bar{x}_2$ and $\dot{\bar{x}}_2$ by their expressions gives

$$\begin{cases} \dot{x}_1 = -k_1 x_1 + z_2 \\ \dot{z}_2 = x_3 - \dot{\bar{x}}_2 \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = f_4 + g_4 u \end{cases} . \tag{15.30}$$

Considering only the two first equations of Eq. (15.30) and taking $\bar{x}_3 = x_3$ as a virtual input, it can be stabilized by,

$$\bar{x}_3 = -x_1 + \dot{\bar{x}}_2 - k_2 z_2 \ , \tag{15.31}$$

where $k_2 \in \mathbb{R}^{*+}$. The stability is proved using the Lyapunov function $V(x_1, z_2) = \frac{1}{2}(x_1^2 + z_2^2)$. The time derivative of V_1 along the solution is,

$$\dot{V}(x_1, z_2) = x_1\dot{x}_1 + z_2\dot{z}_2 = x_1(-k_1 x_1 + z_2) + z_2(x_3 - \dot{\bar{x}}_2) \ . \tag{15.32}$$

After simplification, it results in

$$\dot{V}(x_1, z_2) = -k_1 x_1^2 + z_2(x_1 + x_3 - \dot{\bar{x}}_2) \ . \tag{15.33}$$

Replacing the expression of x_3 gives

$$\dot{V}(x_1, z_2) = -k_1 x_1^2 - k_2 z_2^2 \ , \tag{15.34}$$

which concludes the proof.

Third equation Similarly to the the second step, the control input of the second equation cannot be exactly tracked and therefore z_3 is defined as $z_3 = x_3 - \bar{x}_3$. The system is

$$\begin{cases} \dot{x}_1 &= & -k_1 x_1 + z_2 \\ \dot{z}_2 &= & -k_2 z_2 + z_3 - x_1 \\ \dot{x}_3 &= & x_4 \\ \dot{x}_4 &= & f_4 + g_4 u \end{cases} . \tag{15.35}$$

Eliminating x_3 leads to

$$\begin{cases} \dot{x}_1 &= & -k_1 x_1 + z_2 \\ \dot{z}_2 &= & -k_2 z_2 + z_3 - x_1 \\ \dot{z}_3 &= & x_4 - \dot{\bar{x}}_3 \\ \dot{x}_4 &= & f_4 + g_4 u \end{cases} . \tag{15.36}$$

Using x_4 as a virtual input, the system is be stabilized by

$$\bar{x}_4 = -z_2 + \dot{\bar{x}}_3 - k_3 z_3 \ , \tag{15.37}$$

where $k_3 \in \mathbb{R}^{*+}$. The stability is proved using the Lyapunov function $V(x_1, z_2, z_3) = \frac{1}{2}(x_1^2 + z_2^2 + z_3^2)$. The time derivative of V_1 along the solution is,

$$\begin{array}{rcl} \dot{V}(x_1, z_2, z_3) &= & x_1\dot{x}_1 + z_2\dot{z}_2 + z_3\dot{z}_3 \\ &= & x_1(-k_1 x_1 + z_2) + z_2(-k_2 z_2 + z_3 - x_1) + z_3(x_4 - \dot{\bar{x}}_3) \end{array} . \tag{15.38}$$

After simplification

$$\dot{V}(x_1, z_2, z_3) = -k_1 x_1^2 - k_2 z_2^2 + z_3(x_4 + z_2 - \dot{\bar{x}}_3) \ . \tag{15.39}$$

Replacing the expression of x_4 gives

$$\dot{V}(x_1, z_2, z_3) = -k_1 x_1^2 - k_2 z_2^2 - k_3 z_3^2 \ , \tag{15.40}$$

which concludes the proof.

Fourth equation The control input of the third equation cannot be exactly tracked and therefore z_4 is defined as $z_4 = x_4 - \bar{x}_4$ The system is now

$$\begin{cases} \dot{x}_1 &= -k_1 x_1 + z_2 \\ \dot{z}_2 &= -k_2 z_2 + z_3 - x_1 \\ \dot{z}_3 &= -k_3 z_3 + z_4 - z_2 \\ \dot{x}_4 &= f_4 + g_4 u \end{cases} . \tag{15.41}$$

Eliminating x_4 leads to

$$\begin{cases} \dot{x}_1 &= -k_1 x_1 + z_2 \\ \dot{z}_2 &= -k_2 z_2 + z_3 - x_1 \\ \dot{z}_3 &= -k_3 z_3 + z_4 - z_2 \\ \dot{z}_4 &= f_4 + g_4 u - \dot{\bar{x}}_4 \end{cases} . \tag{15.42}$$

The **real**, as opposed to virtual, control input u is selected as

$$u = \frac{1}{g_4}(-f_4 - z_3 + \dot{\bar{x}}_4 - k_4 z_4) , \tag{15.43}$$

where $k_4 \in \mathbb{R}^{*+}$. The stability is proved using the Lyapunov function $V_4(x_1, z_2, z_3, z_4) = \frac{1}{2}(x_1^2 + z_2^2 + z_3^2 + z_4^2)$. The time derivative of V_4 along the solution is,

$$\begin{aligned} \dot{V}_4(x_1, z_2, z_3, z_4) &= x_1\dot{x}_1 + z_2\dot{z}_2 + z_3\dot{z}_3 + z_4\dot{z}_4 \\ &= x_1(-k_1 x_1 + z_2) + z_2(-k_2 z_2 + z_3 - x_1) \\ &+ z_3(-k_3 z_3 + z_4 - z_2) + z_4(x_4 - \dot{\bar{x}}_4). \end{aligned} . \tag{15.44}$$

After simplification

$$\dot{V}_4(x_1, z_2, z_3, z_4) = -k_1 x_1^2 - k_2 z_2^2 - k_3 z_3^2 + z_4(z_3 + f_4 + g_4 u - \dot{\bar{x}}_4) . \tag{15.45}$$

Replacing the expression of u gives

$$\dot{V}_4(x_1, z_2, z_3, z_4) = -k_1 x_1^2 - k_2 z_2^2 - k_3 z_3^2 - k_4 z_4^2 , \tag{15.46}$$

which concludes the proof.

Input equation The input signal is obtained by recursively replacing the expression in terms of x_1, x_2, x_3 and x_4.

$$g_4 u = (-f_4 - z_3 + \dot{\bar{x}}_4 - k_4 z_4) \tag{15.47}$$

Starting with z_4, the input expression is

$$g_4 u = -f_4 - z_3 + \dot{\bar{x}}_4 - k_4(x_4 - \bar{x}_4) . \tag{15.48}$$

Then the $\bar{x}_4$ virtual input is expanded.

$$
\begin{aligned}
g_4 u &= -f_4 - z_3 + \frac{d}{dt}(-z_2 + \dot{\bar{x}}_3 - k_3 z_3) - k_4(x_4 - (-z_2 + \dot{\bar{x}}_3 - k_3 z_3))) \\
&= -f_4 - z_3 + \frac{d}{dt}(-z_2 + \dot{\bar{x}}_3 - k_3 z_3) - k_4 x_4 + k_4(-z_2 + \dot{\bar{x}}_3 - k_3 z_3)) \\
&= -f_4 - z_3 - k_4 x_4 + \frac{d}{dt}(-z_2 + \dot{\bar{x}}_3 - k_3 z_3) + k_4(-z_2 + \dot{\bar{x}}_3 - k_3 z_3)
\end{aligned}
\tag{15.49}
$$

The procedure is continued by removing $z_3 = x_3 - \bar{x}_3$ and results in the input expression

$$
\begin{aligned}
g_4 u &= -f_4 - (x_3 - \bar{x}_3) - k_4 x_4 + \frac{d}{dt}(-z_2 + \dot{\bar{x}}_3 - k_3(x_3 - \bar{x}_3)) + k_4(-z_2 + \dot{\bar{x}}_3 - k_3(x_3 - \bar{x}_3)) \\
&= -f_4 - x_3 + \bar{x}_3 - k_4 x_4 + \frac{d}{dt}(-z_2 + \dot{\bar{x}}_3 - k_3 x_3 + k_3 \bar{x}_3) + k_4(-z_2 + \dot{\bar{x}}_3 - k_3 x_3 + k_3 \bar{x}_3) \\
&= -f_4 - x_3 - k_4 x_4 - k_3 x_4 - k_4 k_3 x_3 + \bar{x}_3 + \frac{d}{dt}(-z_2 + \dot{\bar{x}}_3 + k_3 \bar{x}_3) + k_4(-z_2 + \dot{\bar{x}}_3 + k_3 \bar{x}_3)
\end{aligned}
\tag{15.50}
$$

15.2.4 Simulations

Although the theory guarantees that the control law results in an asymptotically stable system, the analysis does not include errors such as noise, unmodeled dynamics, unmodeled nonlinearities, saturations, or delays. This section presents several numerical simulations that evaluate the backstepping controller under the presence of such errors. The controller will eventually be implemented on a real-time system where sampling, communication and computation delays are unavoidably introduced. Similarly, the maximal motor torque is limited by nature and creates a saturation of the command. The feedback gains are influencing the convergence rate and the simulations can be used to get an order of magnitude of some practical values. The following simulations are performed in order to qualitatively evaluate the different effects.

- several controller gains K_1 and K_2.
- several saturation values for the motor input.
- several time delays in the control loop.

It is important to keep in mind that the simulations must be carefully designed to avoid issues related to the numerical inaccuracies or numerical solvers. For example, using a variable step solver with a continuous derivative block and a continuous integration block creates a convergence issue. Solutions for this issue are:

- use a fixed step solver.

symbol	description	value	units
$gearratio$	gear ratio	100	N.A
m	link side inertia	7.2×10^{-7}	$[kgm^2]$
b	motor inertia	2×10^{-4}	$[kgm^2]$
b_θ	motor damping	10^{-2}	$[Nm/(rad/s)]$
b_q	link damping	10^{-3}	$[Nm/(rad/s)]$
K	joint stiffness	20	$[Nm/rad]$
K_3	controller gain 3	100	N.A
K_4	controller gain 4	100	N.A

Table 15.1: Simulation parameters for a single joint and single motor with linear stiffness

- use a discrete integration or derivative.
- compute symbolically the derivatives (preferred solution).

It is a good practice to slightly modify the sampling time or the error tolerance and check that the results of the simulation are not changed significantly. It is advisable to verifiy the results in the case where the results are very sensitive to the solver parameters. Either by performing some experiments or running some reference simulation. Table 15.1 reports the important simulation parameters and their values. The simulations are performed using a variable step solver and use symbollic derivatives (*ode23t* of Matlab).

In the Figures 15.7 and 15.8, the influence of the two first feedback gains is investigated. In Fig. 15.7, the first gain is increased and consequently, the stiffness of the link is increased. Oscillations are appearing if the value is increased too much. Fig. 15.8 shows that the second feedback gain behaves mainly as a damping coefficient. Increasing the value of K_2 slows down the response of the link.

An electrical motor has a limited torque capability. This limitation is either due to the maximum torque the structure can support or the maximum current that can flow through the coils. In practice, to avoid any damages, the motor maximum desired torque is limited by firmware or software. In the case of the motors of the hand arm system, a first limitation is implemented in the system driver and a hard limit is implemented in the motor controller FPGA. The saturation introduces a nonlinearity that can destabilize the system. Although some theories (such as the sliding mode control [135]) are able to explicitly deal with saturation effects, this remains an open field of research. In this work, the influence of the saturation is evaluated by simulation. The diagram corresponding to the simulation is reported in Fig. 15.9. In a first step the simulation is performed without saturation (cf. Fig. 15.11 red/solid). A second simulation with the same

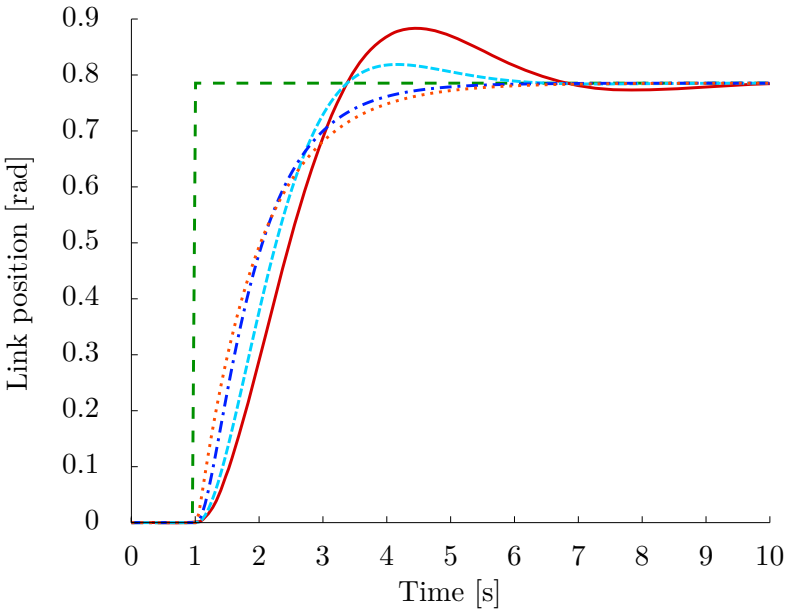

Figure 15.7: Simulations, influence of K_1: link position after a commanded step of 0.8 rad. The red/solid, light blue/dashed, blue/dot dashed and orange/dotted lines depict the responses obtained for a gain K_1 of 0.2, 1, 5 and 50 (the K_2 coefficient being set to $K_2 = 1$). The coefficient K_1 has a strong influence on the stiffness of the link.

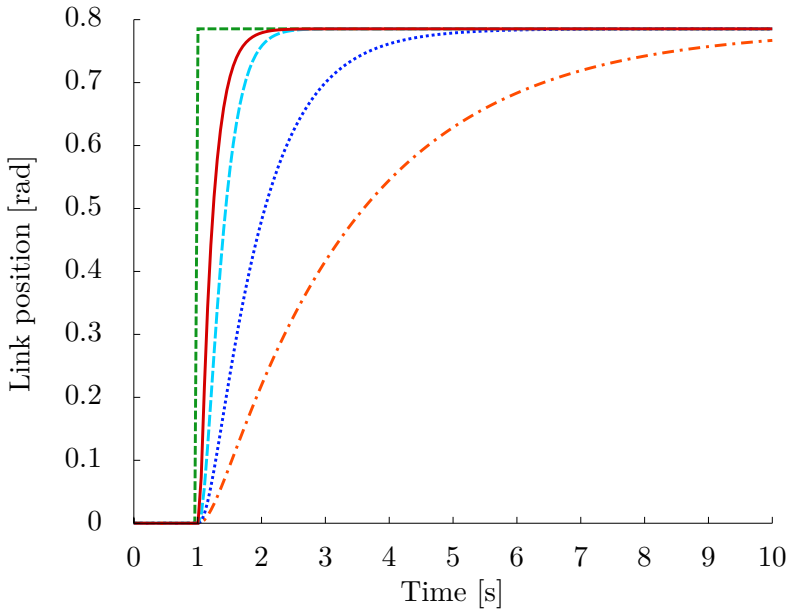

Figure 15.8: Simulations, influence of K_2: link position after a commanded step of 0.8 rad. The red/solid, light blue/dashed, blue/dot dashed and orange/dotted lines are representing the link position obtained for a K_2 coefficient of 0.2, 1, 5 and 50 (the K_1 coefficient being set to $K_1 = 5$). The coefficient K_2 has a strong influence on the damping of the link.

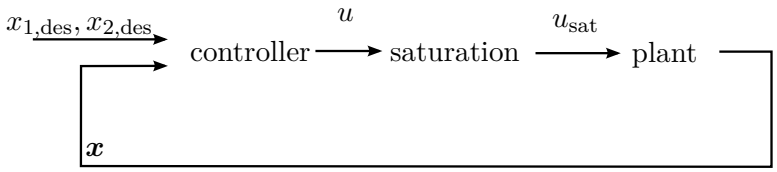

Figure 15.9: Diagram of the simulation used for the evaluation of the influence of input saturation. A saturation block is placed between the controller output and the plant.

ïĄą

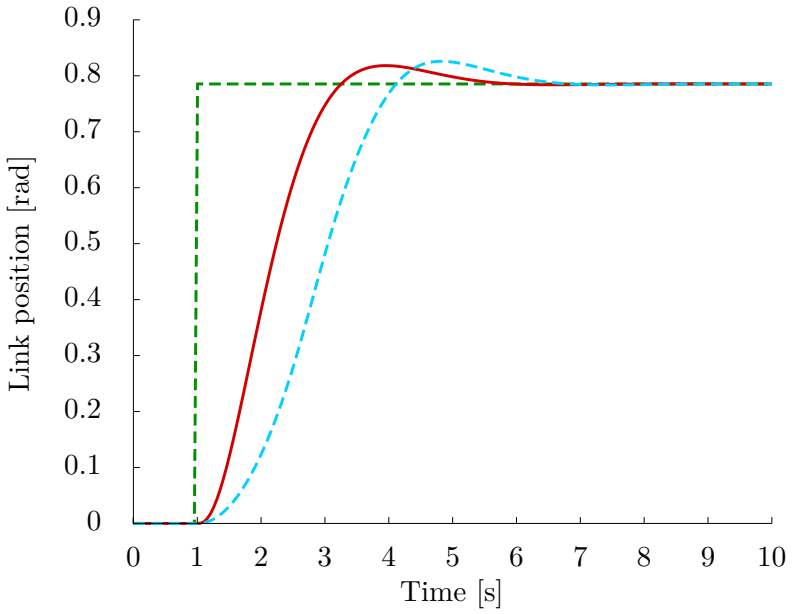

Figure 15.10: Simulations, influence of a saturation of the control input u: link position after a commanded step of 0.8 rad. The red/solid and light blue/dashed curves are the responses obtained without and with a saturation of $|u| < 0.0005$ (the coefficients are set to $K_1 = 1$, $K_2 = 1$, $K_3 = 100$ and $K_4 = 100$).

parameters and the same initial conditions is conducted with the saturation (cf. Fig.15.11 light blue/dashed). The plots are showing that the controller remains stable in the two cases, despite the strong saturation visible in the command (cf. Fig. 15.11).

A last simulation on the single joint driven by a single motor and with a linear stiffness is performed to analyze the influence of delays. Based on the experience of the previous robots developed in the institute, it is known that delays can have very deleterious effects on the stability. The analysis of such system delays, together with nonlinear dynamics, is still a research topic and is out of the scope of the present work. The interested reader can consult [136–138] for work, mainly oriented towards the issues of time varying delays in telemanipulation scenario, on the modeling and the control of system with delays. In this work, the influence of time delay in the control loop is evaluated by adding a constant time delay between the command and the plant as well as between the measurements and the controller. The diagram corresponding to the simulation is reported in Fig. 15.12. Increasing the delay from 0ms to 1ms confirms that they have a strong influence on the control performance. As described in the modeling part, the delay for a complete round trip of the signals, i. e. from measure to actuation, is 333μs. Therefore, according to the simulations, the system

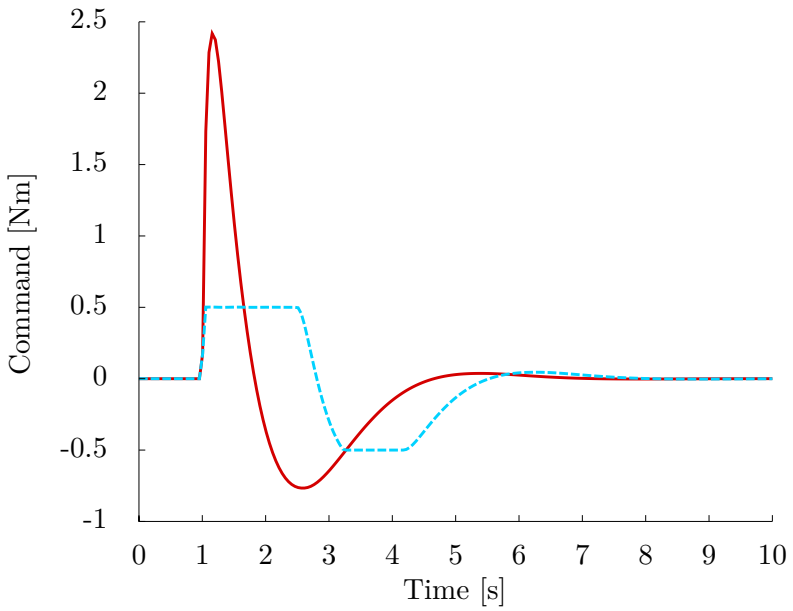

Figure 15.11: Simulations, influence of a saturation of the control input u: input command after a commanded step of 0.8 rad. The light blue(solid) and blue (dashed) lines are the responses obtained without and with a saturation of $|u| < 0.0005$ (the coefficients are set to $K_1 = 1$, $K_2 = 1$, $K_3 = 100$ and $K_4 = 100$).

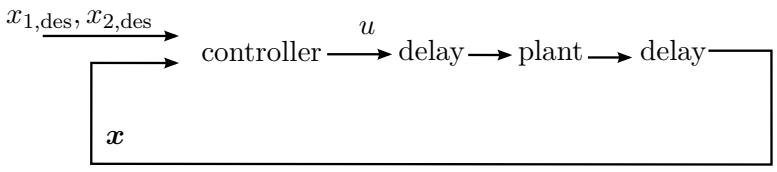

Figure 15.12: Diagram of the simulation used for the evaluation of the influence of time delays. A fixed delay is placed between the command and the actuator as well as between the measurements and the controller.

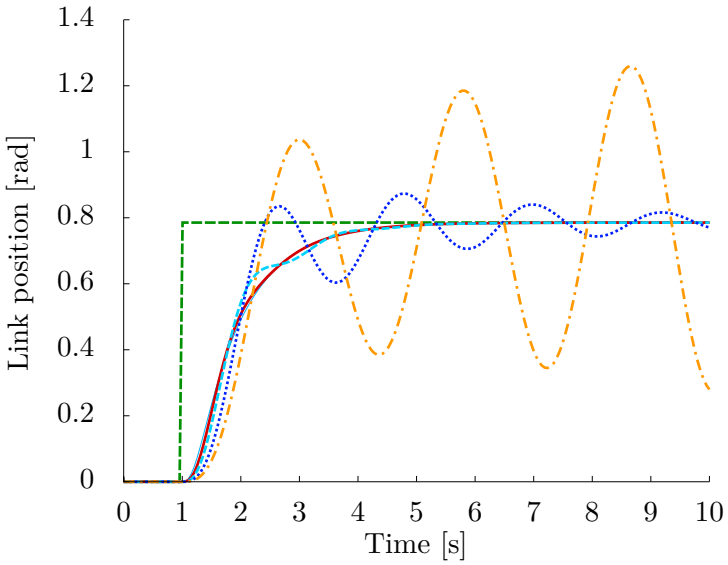

Figure 15.13: Simulations, influence of a delay in the control input u: link position after a commanded step of 0.8 rad. The red/solid (resp. light blue/dashed, blue/dotted, orange/dot dashed) line is the response obtained with a 0ms delay (resp. 0.1, 0.2, 0.5 and 1ms) (the coefficients are set to $K_1 = 5$, $K_2 = 1$, $K_3 = 100$ and $K_4 = 100$).

should be non oscillating even with the large gains that were selected for this simulation.

15.2.5 Experiments

The controller derived and simulated in the previous sections is implemented on a test setup described in Fig. 15.14. A motor, similar to the one of the modeling section, is connected to a low inertia link with two elastic tendons. The stiffness of the tendons is linear (i. e. the force is proportional to the elongation). An internal pretension is required in order to avoid slack in the tendons during motion. However, because of the linearity of the springs it is not influencing the dynamic equations (as long as slackening or breaking is not happening). Table 15.2 reports the values used for the controller and the parameters corresponding to the physical setup. The stiffness of the springs has been obtained by direct measurement and the inertia of the link has been estimated from the CAD data. Figure 15.15 depicts the measured link position obtained after a commanded step in the case of the backstepping controller and a PD controller (for reference). It is clearly visible that the backstepping controller manages to control the link without generating oscillations. The motor trajectory denoted by $\boldsymbol{A}$ in Fig. 15.15 is a characteristic of the flexible joint systems.

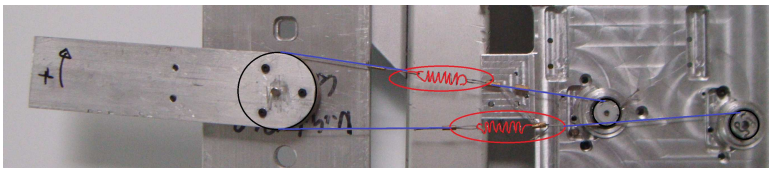

Figure 15.14: Experimental setup used for the verification of the backstepping controller

symbol	description	value	units
$gearratio$	gear ratio	100	N.A
M	link side inertia	7.2×10^{-7}	$[kgm^2]$
B	motor inertia (output inertia)	2×10^{-4}	$[kgm^2]$
b_θ	motor damping	10^{-2}	$[Nm/(rad/s)]$
b_q	link damping	10^{-3}	$[Nm/(rad/s)]$
K	joint stiffness	20	$[Nm/rad]$
K_3	controller gain 3	100	N.A
K_4	controller gain 4	100	N.A

Table 15.2: Experimental parameters and controller parameters for a single joint and single motor with linear stiffness

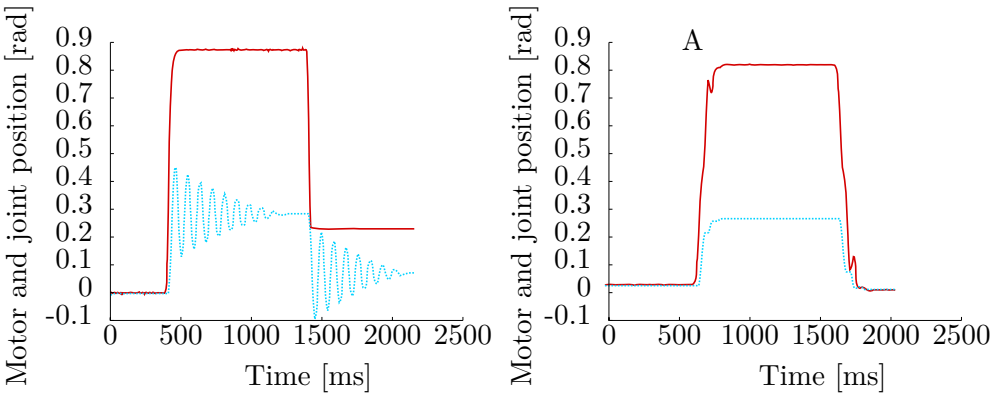

Figure 15.15: Experiment: measured motor position red/solid and link position light blue/dotted after a commanded position step. The pulley ratio between the motor and the link is about 3. Left: a PD controller on the motor position is used. Right: the backstepping controller is used.