

Article

Comparative Study of Methods for Robot Control with Flexible Joints

Ranko Zotovic-Stanasic ^{1,*} , Rodrigo Perez-Ubeda ²  and Angel Perles ³ 

¹ Institute of Industrial Control Systems and Computing (ai2), Universitat Politècnica de València, 46022 Valencia, Spain

² Mechanical Engineering Department, Universidad de Antofagasta, Antofagasta 1200000, Chile; rodrigo.perez.ubeda@uantof.cl

³ ITACA Institute, Universitat Politècnica de València, 46022 Valencia, Spain; aperles@upv.es

* Correspondence: rzotovic@isa.upv.es

Abstract: Robots with flexible joints are gaining importance in areas such as collaborative robots (cobots), exoskeletons, and prostheses. They are meant to directly interact with humans, and the emphasis in their construction is not on precision but rather on weight reduction and soft interaction with humans. Well-known rigid robot control strategies are not valid in this area, so new control methods have been proposed to deal with the complexity introduced by elasticity. Some of these methods are seldom used and are unknown to most of the academic community. After selecting the methods, we carried out a comprehensive comparative study of algorithms: simple gravity compensation (Sgc), the singular perturbation method (Spm), the passivity-based approach (Pba), backstepping control design (Bcd), and exact gravity cancellation (Egc). We modeled these algorithms using MATLAB and simulated them for different stiffness levels. Furthermore, their practical implementation was analyzed from the perspective of the magnitudes to be measured and the computational costs of their implementation. In conclusion, the Sgc method is a fast and affordable solution if joint stiffness is relatively high. If good performance is necessary, the Pba is the best option.

Keywords: control; robot flexible joints; backstepping; passivity



Citation: Zotovic-Stanasic, R.;

Perez-Ubeda, R.; Perles, A.

Comparative Study of Methods for Robot Control with Flexible Joints.

Actuators **2024**, *13*, 299. <https://doi.org/10.3390/act13080299>

Academic Editor: Giulia Scalet

Received: 3 June 2024

Revised: 18 July 2024

Accepted: 3 August 2024

Published: 6 August 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Robots with flexible joints are becoming increasingly relevant. New types of robots are gaining importance on the market, such as collaborative robots (cobots), exoskeletons, and prostheses. They are meant to directly interact with humans. In this new generation of robots, the emphasis in their construction is not on precision (such as for rigid robot counterparts) but rather on weight reduction (collaborative robots) and/or soft interaction with humans (exoskeletons and prostheses). Thus, these new robots use more elastic mechanical transmissions.

Cobots typically have harmonic drive transmissions instead of classical gears [1] due to their light weight, high reduction ratio, and relatively good back-driveability. Wearable robotics mostly use series elastic actuators (SEAs) [2,3] for transmission. SEAs are added to some cobots to increase the compliance of their harmonic drives [4], such as those produced by the Rethink company.

Before the advent of flexible robots, most robots were rigid to achieve high precision. Controlling rigid manipulators is well covered and included in robotics textbooks [5–7]. In these cases, the best performance is obtained using inverse dynamics control methods, also called computed torque. This involves compensating all nonlinear forces that act on the robot, such as gravity, inertia, and centrifugal and Coriolis forces.

When approaching the problem of controlling flexible robots, the first idea that comes to mind is adapting the well-known inverse dynamics method. However, in a rigid robot, the actuators are directly connected to the links, compensating for external forces. In a robot

with flexible links, the motor acts on an elastic element, causing its torsion, which causes the link to move. Thus, the dynamic between the actuator and the link does not directly compensate for external forces.

Many applications involve a wide range of compliances in their joints. According to [8], stiffnesses may vary from 5 to 10 kNm/rad down to 0.2 to 1 kNm/rad. This wide elasticity range complicates control considerably. In addition, stability analysis is much more difficult.

For example, oscillations may occur, possibly prohibiting many robotics tasks.

To achieve a task, the trajectory of the link (q, \dot{q} , etc.) must be controlled, but it is only possible to act on the motor ($\theta, \dot{\theta}$, etc.).

Another complication in flexible robots vs. rigid ones is the higher order of the system. While the former is second-order, the latter is fourth [9,10]. Thus, it may be necessary to measure and include higher-order derivatives.

Several control strategies have been proposed to deal with this wide range of elasticity. The late 1980s and early 1990s were prolific regarding contributions in this field; researchers aimed to control motor position and velocity to achieve good trajectory tracking with links.

In [11,12], some less conventional control methods, like the *singular perturbation method* (Spm) or *backstepping control design* (Bcd), were proposed. Tomei [13] introduced an extremely simple PD with the *simple gravity compensation* (Sgc) method and demonstrated its stability criteria. The authors of [14] improved the previous method, proposing *exact gravity cancellation* (Egc) while introducing less restrictive criteria with better trajectory tracking. The authors of [15,16] introduced the *passivity-based approach* (Pba) to determine the control action.

For each case, it is difficult to decide which method is appropriate and which constraints to use for its practical application, such as computational costs and expensive sensor requirements. Although some of these methods have been described in previous work [11,12,17,18], this study models and simulates a selection of methods to provide a clearer picture of the performance of each for different stiffness levels.

This study is dedicated to applying “classical” methods to control robots with flexible joints. A few recent strategies have not been included since they have several versions. Their analysis would be very extensive and has been left for future work. However, they are briefly mentioned below.

One approach is model predictive control (MPC) [19–21]. This method includes constraints such as maximum motor torques and velocities in the controller design.

Another strategy is sliding mode control [22–24]. It achieves good and robust trajectory tracking, but it may need a very fast sampling period.

Several authors have dedicated their research to robustly controlling robots with elastic joints [25–27]. This is a wide area, and there are many very different contributions.

This paper is organized as follows: the Section 2 presents the approach used to model the selected control algorithms. It then briefly describes the basis of each control algorithm and, finally, the simulation parameters. Then, Section 3 presents the output of the simulations for different stiffness levels. Next, Section 4 provides an interpretation of the simulation results, the requirements of each method for its practical implementation, and the pros and cons. Finally, Section 5 discusses the advantages and disadvantages of each controller. At the end of the article, Appendix A describes the first and second derivatives of the inertia, gravity, centrifugal, and Coriolis matrices.

2. Materials and Methods

2.1. Approaches for Modeling Robots with Flexible Joints

The dynamic model of the rigid robot is well known and can be found in textbooks [5–7]. It can be represented by the following expression:

$$\tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) \quad (1)$$

where τ is the vector of the motor torque; q , \dot{q} , and \ddot{q} are the vectors of the motor position, velocity, and acceleration, respectively; $M(q)$ is the inertia matrix of the robot; $C(q, \dot{q})$ is the matrix of the centrifugal and Coriolis forces; and $G(q)$ is the vector of gravity torques on the motors.

The following subsections describe the two possible ways to model the dynamics of robots with elastic joints: conventional modeling and the singularly perturbed model.

2.1.1. Conventional Elastic Modeling

The main difference between modeling a rigid robot and a flexible robot is an elastic element between the motor rotor and the link (see Figure 1).

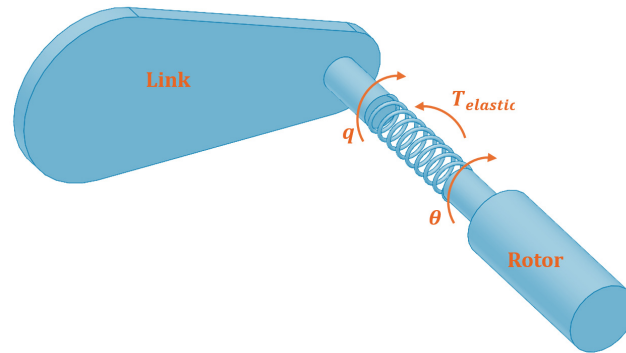


Figure 1. Schema of an elastic joint. θ is the position of the motor rotor, q is the position of the link, and $\tau_{elastic} = K(\theta - q)$ is the elastic torque. K is the stiffness of the joint in this figure.

The dynamics can be separated into two parts: the motor side and the link side. We can directly actuate the former, but we need to control the latter to achieve tasks, for example, as in [17,18]. This fact can be determined by assuming three conditions:

- A1: Joint deflections are small, so flexibility effects are limited to the linear elasticity domain.
- A2: Actuator rotors are modeled as uniform bodies with their centers of mass on the rotation axis.
- A3: Each motor is located in the robot arm before the driven link. This can be generalized to the case of multiple motors simultaneously driving multiple distal links.

In this case, the complete model can be represented by the following expression:

$$\begin{bmatrix} M(q) & S(q) \\ S^T(q) & J_m \end{bmatrix} \begin{bmatrix} \ddot{q} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} c(q, \dot{q}) + c_1(q, \dot{q}, \dot{\theta}) \\ c_2(q, \dot{q}) \end{bmatrix} + \begin{bmatrix} G(q) + K(q - \theta) \\ K(\theta - q) \end{bmatrix} = \begin{bmatrix} 0 \\ \tau \end{bmatrix} \quad (2)$$

where τ is the vector of the motor torque; q , \dot{q} , and \ddot{q} are the vectors of the link position, velocity, and acceleration, respectively; θ , $\dot{\theta}$, and $\ddot{\theta}$ are the vectors of the rotor position, velocity, and acceleration, respectively; $M(q)$ is the inertia matrix of the robot; $c(q, \dot{q})$, $c_1(q, \dot{q}, \dot{\theta})$ and $c_2(q, \dot{q})$ are the matrices of the centrifugal and Coriolis forces; $G(q)$ is the vector of gravity torque on the motors; and $K(\theta - q)$ is the elastic torque. The matrix, S , represents the inertial coupling between the rotors and the links.

S is smaller than the other terms and is neglected by most authors, as are the $c_1(q, \dot{q}, \dot{\theta})$ and $c_2(q, \dot{q})$ components, providing a reduced model:

$$\begin{bmatrix} M(q) & 0 \\ 0 & J_m \end{bmatrix} \begin{bmatrix} \ddot{q} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} c(q, \dot{q}) \\ 0 \end{bmatrix} + \begin{bmatrix} G(q) + K(q - \theta) \\ K(\theta - q) \end{bmatrix} = \begin{bmatrix} 0 \\ \tau \end{bmatrix} \quad (3)$$

This model is used for stability analysis in all of the studies mentioned in this article and, in general, by most authors. It will also be used in this study.

2.1.2. Singularly Perturbed Model

Another approach is using a singular perturbation model, that is, to refer to a situation in which a system exhibits two or more distinct time scales of motion. In these systems, one of the time scales is much slower than the others, separating the fast and slow dynamics. It was used in [6,11–13].

With a flexible joint, the elastic torque is much faster than the link. This separates the fast dynamics (elastic torque) from the slow dynamics (motion of the link). A singularly perturbed model can be obtained using a new coordinate space:

$$\begin{bmatrix} q \\ z \end{bmatrix} = \begin{bmatrix} I & 0 \\ -K & K \end{bmatrix} \begin{bmatrix} q \\ \theta \end{bmatrix} = \begin{bmatrix} q \\ K(q - \theta) \end{bmatrix} \quad (4)$$

where $z = K(q - \theta)$ is the elastic torque.

From Equation (3),

$$\ddot{\theta} = J_m^{-1}(\tau + z) \quad (5)$$

and

$$\ddot{q} = M(q)^{-1}(-c(q, \dot{q}) - G(q) - z) \quad (6)$$

From Equation (4),

$$\ddot{z} = K(\ddot{\theta} - \ddot{q}) = K(J_m^{-1}(\tau + z) - M(q)^{-1}(-c(q, \dot{q}) - G(q) - z)) \quad (7)$$

$$\ddot{z} = K((J_m + M(q))^{-1}z + J_m^{-1}\tau + M(q)^{-1}(c(q, \dot{q}) + G(q))) \quad (8)$$

If we assume that the matrix, K , has large and similar elements, it is possible to extract a large common scale factor, $\hat{K} \gg 1$, from K : $K = \frac{1}{\epsilon^2}\hat{K} = \frac{1}{\epsilon^2}\text{diag}\{\hat{k}_1, \hat{k}_2, \dots, \hat{k}_n\}$, $0 < \epsilon \ll 1$.

Thus, Equation (8) can be rewritten as

$$\epsilon^2\ddot{z} = \hat{K}((J_m + M(q))^{-1}z) + J_m^{-1}\tau + M(q)^{-1}(c(q, \dot{q}) + G(q)) \quad (9)$$

Higher stiffness values mean lower ϵ values.

2.2. Control Strategies

This subsection briefly explains the control methods used in this study.

2.2.1. Singular Perturbation Method

The singular perturbation method [28] control strategy is used for processes that have one part that is much faster than the other. This method treats the slow and the fast parts separately, making control much easier. Two control actions are generated: one for the slow part and another one for the fast one.

The output of the slow loop is used as the input for the fast loop. To obtain the final control action, slow and fast control actions are added.

For the slow part, the control action (torque) can be generated according to the laws of control for rigid robots, which have been known for decades, for example, the inverse dynamic method provided by Equation (4).

The fast control receives the slow control action as a reference value and must ensure that it will be tracked. According to [18], a possible control law is

$$\tau_{fast} = K_{p\tau}(\tau_{slow} - \tau_{elastic}) - \epsilon K_{d\tau}\dot{\tau}_{elastic} \quad (10)$$

This is a PD control law for the elastic torque, and $K_{p\tau}$ and $K_{d\tau}$ are the proportional and derivative constants, respectively.

The final motor torque should be

$$\tau = \tau_{fast} + \tau_{slow} \quad (11)$$

Notably, the stability criteria for this control method are established according to the Tikhonov theorem [28]. This states that if both the slow and fast loops are separately asymptotically stable and ϵ tends toward zero, their respective errors also tend toward zero. However, since $\epsilon = \frac{1}{k^2}$, it is greater than 0. Consequently, the convergence and stability cannot be determined analytically.

Since the convergence criteria assume that ϵ tends toward zero, the singular perturbation control will work better for robots with stiffer joints than robots with elastic ones.

This unclear stability criteria definition limits the singular perturbation method. It cannot be used for applications such as robust or adaptive control.

2.2.2. Backstepping Control Design

Backstepping control design [29] is a control technique that stabilizes systems with nonlinear dynamics. It involves transforming the nonlinear dynamics into a series of intermediate systems with linear or linearizable dynamics and then applying a sequence of feedback controllers to each intermediate system, from top to bottom. The goal is to design a feedback control law that drives the system to its desired trajectory.

This system must be expressed so that each state variable derivative depends on this state, the next, and the previous ones. Only the last state derivative depends on the control action and all the previous states:

$$\begin{aligned}\dot{x}_1 &= f_1(x_1) + g_1(x_1, x_2)x_2 \\ \dot{x}_2 &= f_2(x_1, x_2)x_2 + g_2(x_1, x_2, x_3)x_3 \\ &\vdots \\ \dot{x}_n &= f_n(x_1, x_2, \dots, x_n)x_2 + g_n(x_1, x_2, \dots, x_n)u\end{aligned}\quad (12)$$

x_2 is a virtual input to guarantee the stability of x_1 . Then, x_3 is used as a virtual input to guarantee the stability of x_2 . This is repeated iteratively until the last state, which is stabilized by the control action, u .

This control method was first used to control elastic joints in [11]. As will be demonstrated in simulations, this method works well. Nonetheless, it needs higher derivatives for the link position, and the system must be represented in a chained form, as in Equation (14).

2.2.3. Simple Gravity Compensation

Simple gravity compensation [13] proposes a PD controller with gravity compensation. The control law is

$$\tau = K_p(\theta_d - \theta) - K_d\dot{\theta} + G(q_d) \quad (13)$$

where

$$\theta_d = q_d + K^{-1}G(q_d) \quad (14)$$

q_d is the reference position of the link.

Asymptotic stability is demonstrated for this case if $\lambda_{\min}\left(\begin{bmatrix} K & -K \\ -K & K + K_p \end{bmatrix}\right) > \alpha$, where α is a number that fulfills the following condition for the given robot: $\|G(q_1) - G(q_2)\| \leq \alpha\|q_1 - q_2\|$.

This method is very simple. The reference position of the motors is necessary to compensate for the gravity torque of the links; it does not need the feedback of the link position. Only the motor position and velocity are used in the loop, helping to assure its stability.

2.2.4. Exact Gravity Cancellation

Exact gravity cancellation was proposed in [14]. Its control action consists of two parts:

$$\tau_m = \tau_g + \tau_0 \quad (15)$$

The first component dynamically compensates for gravity:

$$\tau_g = G(q) + JK^{-1}\ddot{G}(q) \quad (16)$$

The second component is a PD-type law:

$$\tau_0 = K_p(q_d - \theta + K^{-1}G(q)) - K_d(\dot{\theta} - K^{-1}\dot{G}(q)) \quad (17)$$

The global asymptotical stability can be shown via Lyapunov analysis. There are no constraints on the proportional constant.

This method is an improvement over the previous one.

2.2.5. Passivity-Based Approach

The passivity-based approach was proposed by the German Aerospace Center group and the Kuka company [15,16].

The final control law can be expressed as

$$\tau_m = JJ_\theta^{-1}u + (I - JJ_\theta^{-1})\tau_{elastic} \quad (18)$$

$$u = J_\theta\ddot{\theta}_{ref} + K(\theta_{ref} - q_{ref}) - K_\theta\tilde{\theta} - K_{D\theta}\dot{\tilde{\theta}} \quad (19)$$

$$\theta_{ref} = q_{ref} + K^{-1}\left(M(q)\ddot{q}_{ref} + C(q, \dot{q})\dot{q}_{ref} + G(q)\right) \quad (20)$$

J_θ is introduced for inertia shaping of the rotor since control is easier if the rotor and link inertias are similar orders of magnitude. The passivity of the system is thus demonstrated.

However, to obtain θ_{ref} , this method must compensate for the elastic torque and the feedback of the link velocity and acceleration to compute the inertia and centrifugal matrices. Regarding the elastic torque, the authors of [16] proposed a lowpass filter with a cut-off frequency of 250 Hz.

Thus, it is necessary to compute up to the second derivatives of the inertia, centrifugal, and gravity matrices to obtain $\ddot{\theta}_{ref}$, which has a very high computational cost.

2.3. Modeling Robot Dynamics

The described control methods were modeled with MATLAB. The model assumes a two-degrees-of-freedom robot with revolute joints. The MATLAB files needed for the simulations are included in the Supplementary Materials. There are five files, one for each controller. There is also a file called gentryay5 that contains the fifth-order trajectory generator used by the other files.

The dynamics equations were obtained from [17,18]. For simplicity, the S inertia coupling matrix was set to 0 and the gear ratios to 1.

The links were modeled as uniform thin rods, with the following characteristics according to suggestions from experts in the field:

- Their lengths are $L_1 = L_2 = 0.5$ m.
- Their masses are $m_1 = 10$ kg and $m_2 = 0.5$ kg.
- The distances of the centers of gravity from the rotation axes are both $d_1 = d_2 = 0.25$ m.
- Moments of inertia: $I_1 = \frac{m_1 L_1^2}{12}$ kgm² and $I_2 = \frac{m_2 L_2^2}{12}$ kgm².
- Gear ratios: $r_1 = r_2 = 1$.
- Weight of the rotor of the second motor: $m_{r2} = 2$ kg.
- Inertia carried by the second motor: $J_{m2} = (I_2 + m_2 d_2^2) / r_2^2$.
- Inertia carried by the first motor: $J_{m1} = (I_1 + m_1 d_1^2 + m_{r2} L_1^2) / r_2^2$.
- The stiffnesses are set to $K_1 = K_2 = 200$, $K_1 = K_2 = 10^3$, and $K_1 = K_2 = 10^4$ Nm/rad in different simulations.

Some intermediate variables were introduced:

$$\begin{aligned} a_1 &= I_1 + m_1 d_1^2 + (m_{r2} + m_2) L_1^2 + I_2 + m_2 d_2^2 \\ a_2 &= I_2 + m_2 d_2^2 \\ a_3 &= m_2 L_1 d_2 \end{aligned} \quad (21)$$

For the dynamics expressed in Equation (2), the following matrix values were obtained:

$$B(q) = \begin{bmatrix} a_1 + a_3 \cos(q_2) & a_2 + a_3 \cos(q_2) \\ a_2 + a_3 \cos(q_2) & a_2 \end{bmatrix} \quad (22)$$

$$J = \begin{bmatrix} J_{m1} & 0 \\ 0 & J_{m2} \end{bmatrix} \quad (23)$$

$$S = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (24)$$

$$M = \begin{bmatrix} B & S \\ S^T & J \end{bmatrix} \quad (25)$$

$$c(q, \dot{q}) = \begin{bmatrix} -a_3 \sin(q_2) (\dot{q}_1 \dot{q}_2 + \dot{q}_2^2) \\ a_3 \sin(q_2) \dot{q}_1^2 \end{bmatrix} \quad (26)$$

$$c_1(q, \dot{q}, \ddot{q}) = c_2(q, \dot{q}) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (27)$$

$$G(q) = \begin{bmatrix} m_1 g d_1 \cos(q_1) + m_{r2} g L_1 \cos(q_1) + m_2 g (L_1 \cos(q_1) + d_2 \cos(q_1 + q_2)) \\ m_2 g d_2 \cos(q_1 + q_2) \end{bmatrix} \quad (28)$$

2.4. Adjusting the Gains for the Controllers

All the controllers use some sort of feedback, typically proportional–derivative. Their performance will depend on their gains.

To control a single joint [6], there are generally several (or infinite) combinations of proportional and derivative constants that work very well. They are computed based on the desired dynamics of the system, i.e., the natural frequency and damping ratio. To compute the proportional and derivative constants, it is necessary to know the inertia moment and the viscous friction coefficient of the system. Generally, better trajectory tracking is achieved with higher proportional and derivative gains. However, there is a point when increasing the gains practically does not improve the controller.

A multiple-degrees-of-freedom robot is much more complicated. The inertia carried by a motor is variable. Furthermore, centrifugal and Coriolis forces and gravity act on the links.

Most robot controllers (for rigid robots) compensate for the external forces and add a proportional–derivative controller for feedback [5–7]. If all the dynamics (inertia, gravity, centrifugal forces) is compensated, the values of the proportional and derivative gains may be computed for the required natural frequency and damping ratio. However, when, for example, only gravity is compensated, the optimal values of the gains vary as the robot moves.

Usually, authors do not explain how these gains are obtained. One option is to adjust them through trial and error. Another is computing the value of the gains for each motor in real time, as in the case of a single joint, for the desired natural frequency and damping ratio of the system. However, this is time-consuming and not frequently used. Another method [5] is gain scheduling. This involves reading the best gains for the actual robot configuration from a database in every sampling period.

In this study, the trial-and-error method was used. For every controller, many combinations were simulated. The simulations stopped when no more important improvements could be obtained.

3. Results

The simulations were conducted for a fifth-order polynomial trajectory generator. The first joint went from 0 to 2π and the second from 0 to $-\pi$ in five seconds.

The simulation was repeated for stiffnesses of $K_1 = K_2 = 200$, $K_1 = K_2 = 10^3$, and $K_1 = K_2 = 10^4$ Nm/rad for both joints. The first two values are typical for elastic mechanical transmissions like harmonic drives. A value of 200 is almost the most elastic found in the bibliographic research we conducted for this article [30].

Before comparing the different controllers, simulations were conducted, controlling the robot as if it was rigid, i.e., directly compensating for the inertia, gravity, centrifugal, and Coriolis terms. For cases $K_1 = K_2 = 200$ and $K_1 = K_2 = 10^3$, the system became unstable. For case $K_1 = K_2 = 10^4$, it worked acceptably. Of course, this result also depends on the other dynamic parameters of the robot, such as its mass and moments of inertia. Figure 2 shows the results of the simulation for $K_1 = K_2 = 10^4$. The mean quadratic errors for both joints were 0.0006 and 0.0036.

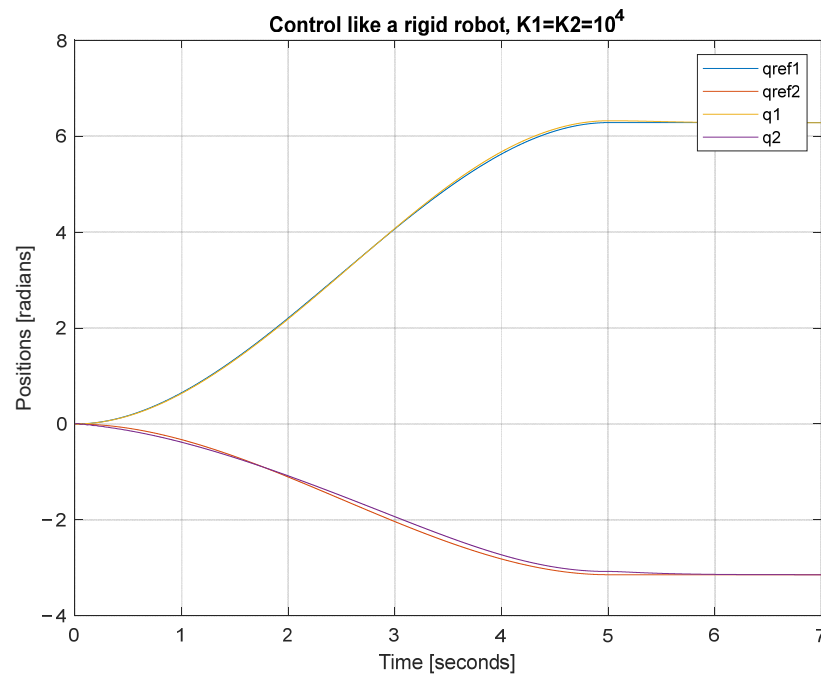


Figure 2. The positions of the links when both stiffnesses are $K_1 = K_2 = 10^4$. The blue (first joint) and red (second joint) lines represent the reference positions (first link in blue and second link in red), while the yellow (first joint) and purple (second joint) lines represent the real positions.

Then, the simulations were run for the different control strategies and stiffnesses. The simulation results for stiffness $K_1 = K_2 = 200$ are shown in Figure 3, and the mean quadratic errors are shown in Table 1.

Table 1. Mean quadratic errors with a stiffness of $K = 200$.

Model	Error J1	Error J2
Simple gravity compensation	0.68959863	0.05578526
Singular perturbation method	0.03245501	0.00112681
Passivity-based approach	0.00005066	0.00093868
Backstepping control design	0.00002342	0.00001048
Exact gravity cancellation	0.05286061	0.02621246

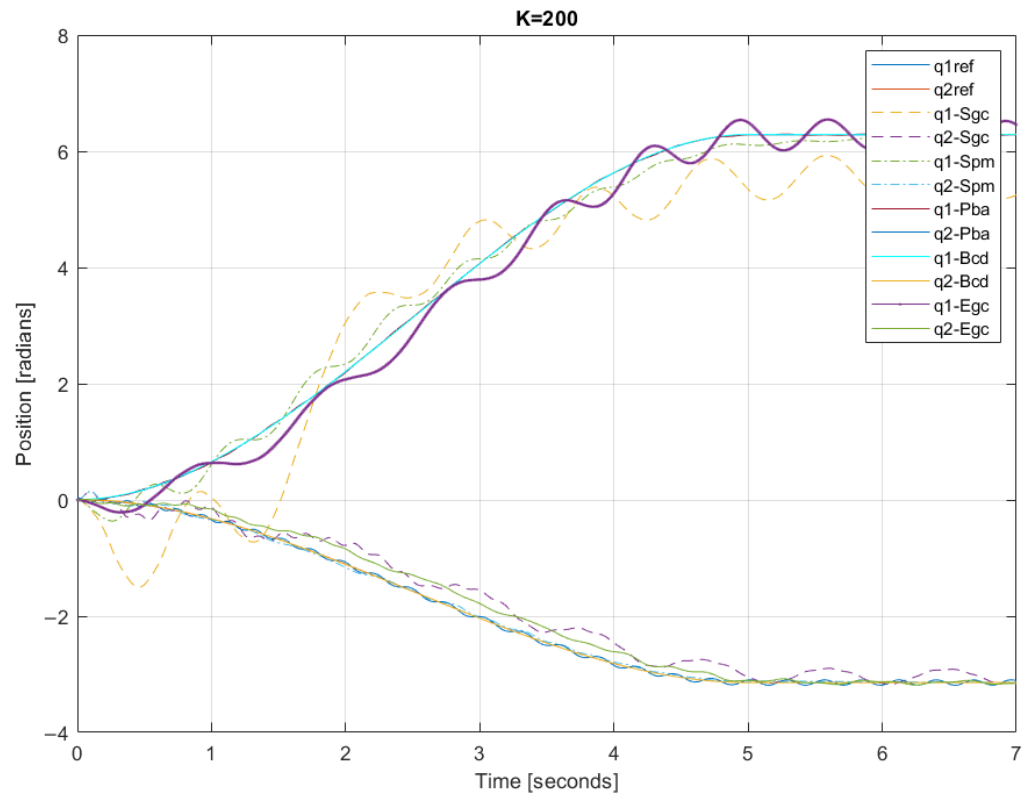


Figure 3. The positions of the links when both stiffnesses are $K_1 = K_2 = 200$. The blue (first joint) and red (second joint) lines represent the reference positions, while the other lines represent the real positions for Sgc, Spm, Pba, Bcd, and Egc.

The simulation results for stiffness $K_1 = K_2 = 10^3$ are shown in Figure 4, and the mean quadratic errors are shown in Table 2.

Table 2. The mean quadratic errors with a stiffness of $K = 1000$.

Model	Error J1	Error J2
Simple gravity compensation	0.030661799	0.004333034
Singular perturbation method	0.001062728	0.000887938
Passivity-based approach	0.000003148	0.000021080
Backstepping control design	0.000023420	0.000010484
Exact gravity cancellation	0.003362791	0.003530729

Finally, the simulation results for stiffness $K_1 = K_2 = 10^4$ are shown in Figure 5, and the mean quadratic errors are shown in Table 3.

Table 3. The mean quadratic errors with a stiffness of $K = 10,000$.

Model	Error J1	Error J2
Simple gravity compensation	0.003610014	0.003408527
Singular perturbation method	0.000083979	0.000830958
Passivity-based approach	0.000000127	0.000000020
Backstepping control design	0.000023419	0.000010488
Exact gravity cancellation	0.003358803	0.003387955

Figures 6 and 7 summarize the mean quadratic errors of the different methods for joint 1 and joint 2 for the three stiffness values.

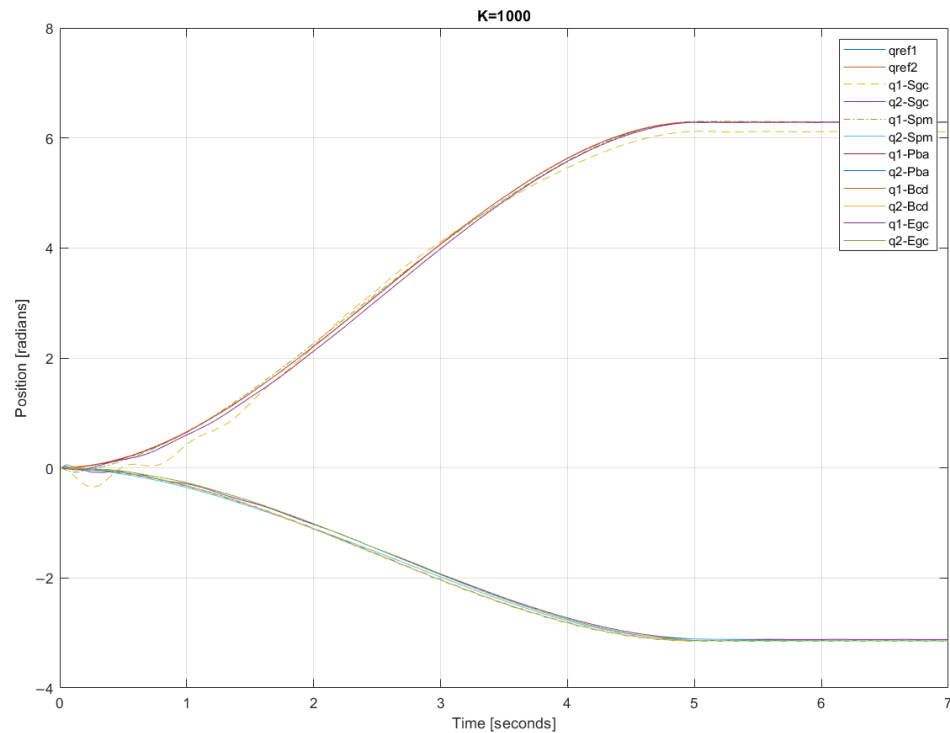


Figure 4. The positions of the links when both stiffnesses are $K_1 = K_2 = 10^3$. The blue (first joint) and red (second joint) lines represent the reference positions (first link in blue and second link in red), while the other lines represent the real positions for Sgc, Spm, Pba, Bcd, and Egc.

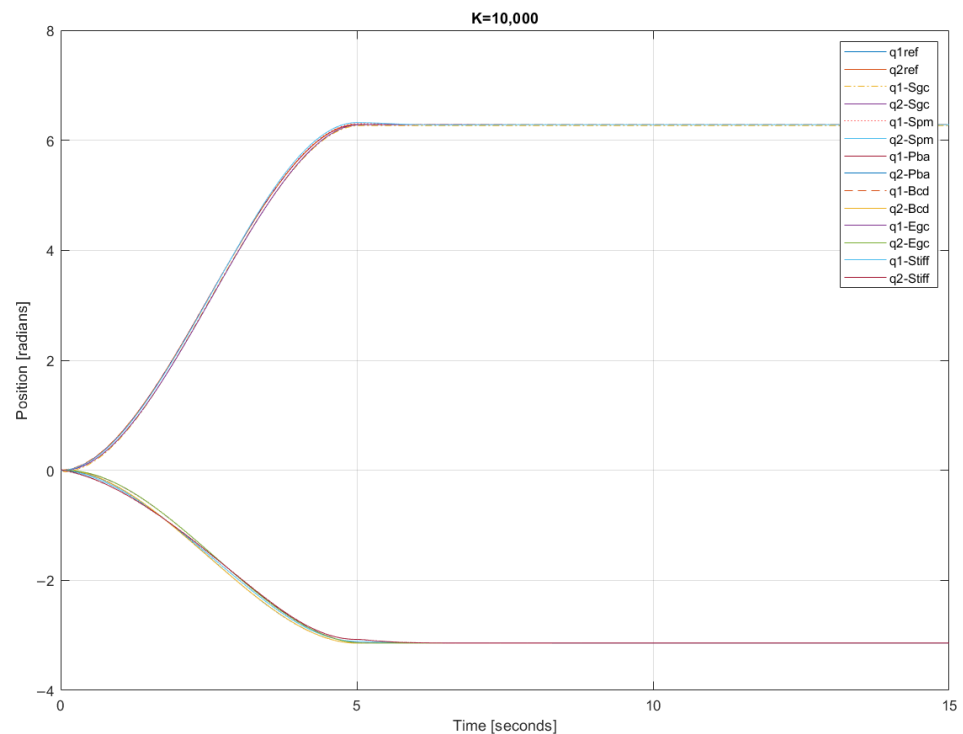


Figure 5. The positions of the links when both stiffnesses are $K_1 = K_2 = 10^4$. The blue (first joint) and red (second joint) lines represent the reference positions (first link in blue and second link in red), while the other lines represent the real positions for Sgc, Spm, Pba, Bcd, and Egc.

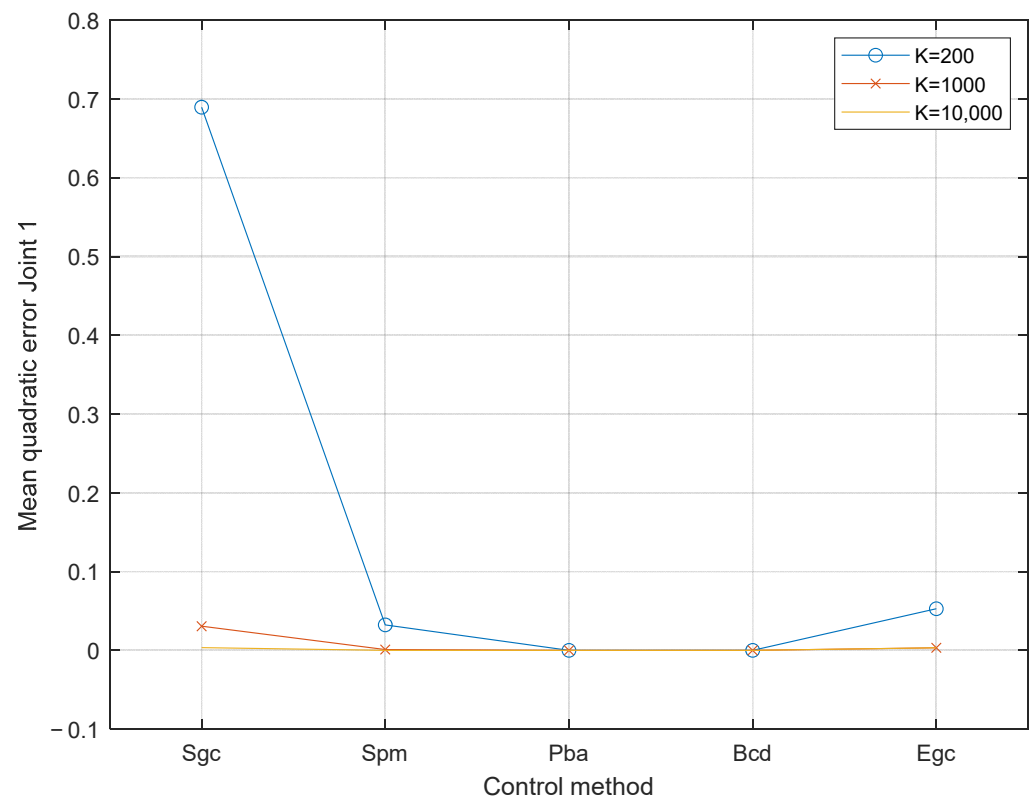


Figure 6. The mean quadratic error of joint 1 when the stiffnesses are 200, 1000, and 10,000.

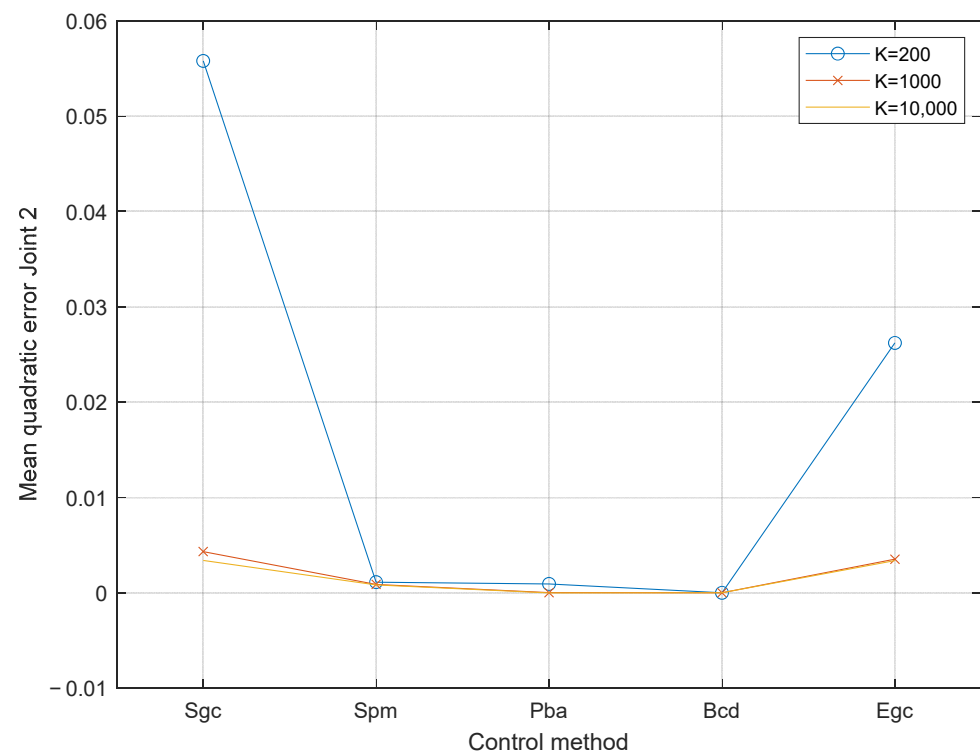


Figure 7. The mean quadratic error of joint 2 when the stiffnesses are 200, 1000, and 10,000.

The results show that the simple gravity control method presents the highest position errors with many oscillations when the rigidity is 200. A specific analysis was carried out for this control method: First, the position error was evaluated for various levels of rigidity with values between 200 and 1000. In turn, the proportionality and derivative gains of

this controller were modified to observe their influence. Figures 8 and 9 show the position errors of each joint for different stiffness and controller gain levels.

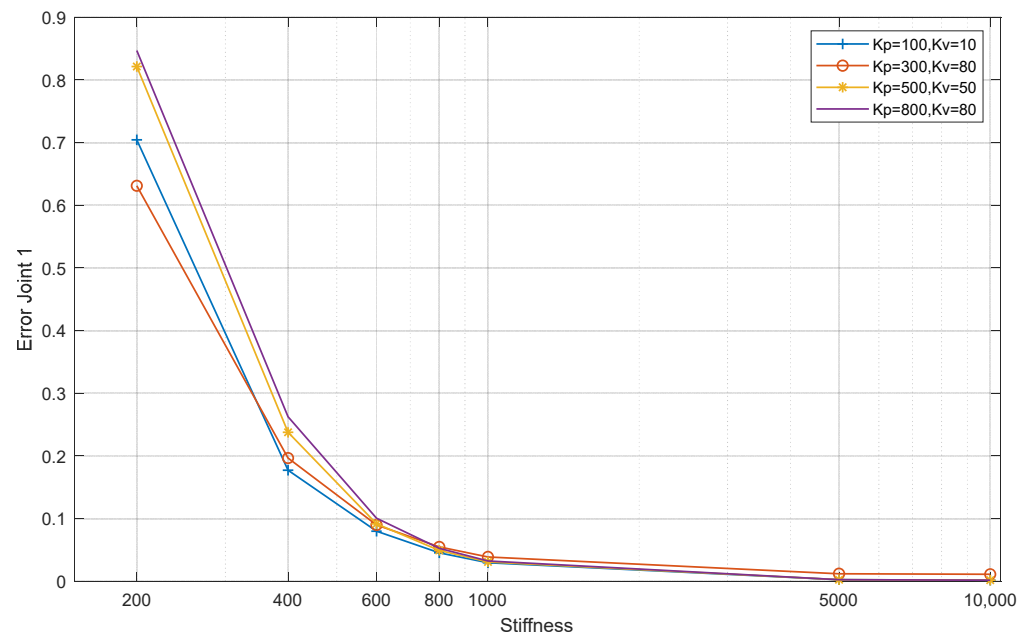


Figure 8. The mean quadratic error of joint 1 with simple gravity compensation with various stiffness values and control gains.

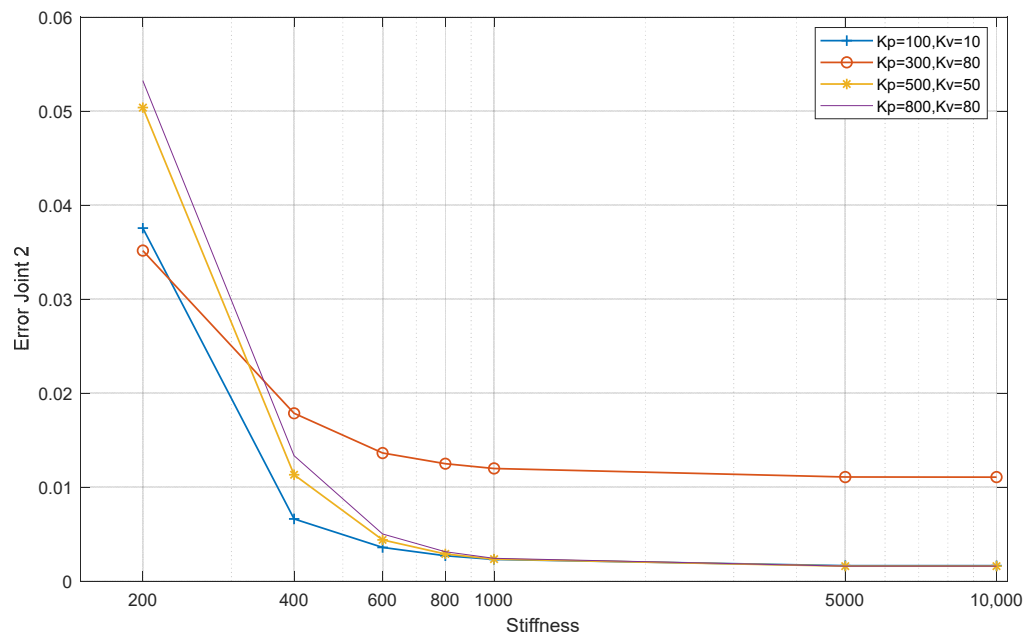


Figure 9. The mean quadratic error of joint 2 with simple gravity compensation with various stiffness values and control gains.

In joint 1, the decreased error is more significant when stiffness increases. The controller gains may provide a minor error, but this is insignificant. For all controllers with low stiffnesses, the system oscillates. When the stiffness value reaches 600–800, the oscillations begin to disappear.

In joint 2, like joint 1, the error decreases as the stiffness increases; however, when low proportional gain and high derivative gain are used, a steady state error occurs with stiffnesses greater than 600.

4. Discussion

As expected, the simulations show that the errors are higher when the joint stiffness is lower. In addition, oscillations appear with low stiffness values for the Sgc, Spm, and Egc controllers.

Regarding mean quadratic errors, the Pba exhibits better results than the other controllers independently of the stiffness value.

The error in the first joint is one order of magnitude higher than that in the second joint, probably because the higher load carried by the first motor increases the nonlinearities. The error is much higher for $K = 200$ than for the other cases. The worst results are obtained with Sgc and then Egc. Spb is comparable to the Pba and Bcd for the second joint but not for the first one.

Although some controllers work very well in simulations, they need to measure or estimate certain magnitudes, such as the high derivatives of the position or torque. For example, position and velocity may be fed back using low-cost sensors and computer interfaces. However, many authors are reluctant to feed back the acceleration because of the significant effect of the noise. Few have used the first and second derivatives of acceleration (jerk and snap, respectively). Thus, the feasibility of these controllers in the real world is doubtful. The same problem occurs with torque feedback because torque measurements are noisy, and its derivative may be impossible to determine.

To summarize the requirements of each method, Table 4 enumerates the necessary sensors and dynamic parameters.

Table 4. Magnitudes necessary to be measured and parameters to be known for each controller.

Method	Magnitudes to Be Measured	Dynamic Parameters to Be Known
Sgc	Joint positions and velocities.	Stiffnesses, masses, and positions of the c.o.g.
Spm	Elastic torques and their first derivatives, positions, and velocities of the links.	All the inertia moments (links and motors), masses, and centers of gravity of the links.
Pba	Position and velocity of the rotor, up to the third derivative of the link position.	All
Bcd	Up to the fourth derivative of the position.	All
Egc	Position and velocity of the rotor, up to the second derivative of the link.	Stiffness, rotor inertia, mass, and position of the c.o.g.

More than one factor influences the complexity of the controller. One aspect is the necessary amount of computation. Another is the set of dynamic parameters that must be known. Some of these factors are not easy to identify, like moments of inertia. The dynamical parameters may vary from one robot to another even if they are the same model.

Another point to be considered is the necessary sampling period. A too-short sampling time may cause problems with real-time calculus and necessitate a more powerful computer. The necessary sampling time depends on the rate of change in the measured magnitude. Thus, typically, methods (e.g., Pba) that need to control the elastic torque need faster sampling than those that require only the positions and their derivatives.

From the cost perspective, position sensors are cheap, and they are necessary for all the control methods described in this article. However, adding torque sensors greatly increases the cost of the system.

The Spm works well for the stiffest case; however, it worsens as elasticity increases. For a stiffness of 200 Nm/rad, oscillations appear. This is logical since the initial supposition of this strategy is that the fast part is much faster than the slow one. On the other hand, the stability of this technique is determined by Tikhonov's theorem [28], which does not provide exact criteria for stability. This affects the robustness of the controller. In addition,

the singular perturbation method requires a torque sensor and the first derivative of the elastic torque. Generally, this method is the third best regarding trajectory tracking.

The Bcd method has the second-best performance regarding trajectory tracking in simulations. However, it is hard to implement it in real applications since it requires feeding back higher-order derivatives.

The Sgc method is extremely simple and cheap (only sensors for the rotor position are required). It has the worst trajectory following, and its performance worsens as joint elasticity increases. For stiffnesses of 200 Nm/rad and 1000 Nm/rad, oscillations appear.

The Egc method is the second worst regarding trajectory tracking. It requires measuring the acceleration of the link to fully compensate the gravity. For a stiffness of 200 Nm/rad, oscillations appear.

The Pba has good performance regardless of joint stiffness. However, it requires an expensive torque sensor for each joint. The sampling period must be faster.

Since adjusting gains is an important part of controller design, a few words will be dedicated to this topic.

The Sgc and Egc methods have no feedback for the link position end velocity—just the motor side. Thus, the link works in an open loop. Varying the gains on the motor side cannot control the link side well for robots with relatively high elasticity.

The Spm has two sets of proportional–derivative gains: one for the fast part and another for the slow part. The fast part is very sensitive, and system stability can be easily lost with small variations in gains.

Regarding the Pba and Bcd methods, all relevant variables are fed back. These controllers are not approximative but exact methods. For these reasons, good trajectory tracking may be achieved with several gain combinations.

5. Conclusions

All the methods performed well for joints with small elasticity; however, oscillations appeared in Sgc for medium and high elasticity and Egc and the Spm for low stiffness.

Considering all the drawbacks and the advantages of the Spm, it is not the most advisable.

Egc is an improvement over simple gravity compensation. However, its small trajectory tracking upgrade does not justify the high derivative requirement or the increased computational cost.

The backstepping method has very good performance in simulations. However, its implementation in the real world is problematic.

In conclusion, the Sgc method is a fast and affordable solution if joint stiffness is relatively high. If good performance is necessary, the Pba is the best option.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/act13080299/s1>, MATLAB files; SingularPerturbationMethod.m, SimpleGravity-Compensation.m, BacksteppingControlDesign.m, PassivityBasedApproach.m, ExactGravityCancellation.m and gentry5.m.

Author Contributions: Conceptualization, R.Z.-S.; methodology, R.Z.-S. and A.P.; software, R.Z.-S. and R.P.-U.; validation, R.Z.-S., R.P.-U., and A.P.; formal analysis, R.Z.-S. and R.P.-U.; investigation, R.Z.-S.; resources, R.Z.-S., R.P.-U. and A.P.; data curation, R.Z.-S. and R.P.-U.; writing—original draft preparation, R.Z.-S.; writing—review and editing, R.Z.-S., R.P.-U. and A.P.; visualization, R.Z.-S., R.P.-U. and A.P.; supervision, R.Z.-S., R.P.-U. and A.P.; project administration, R.Z.-S.; funding acquisition, A.P. All authors have read and agreed to the published version of the manuscript.

Funding: Grant PID2020-117713RB-I00 funded by MCIN/AEI/10.13039/501100011033.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Appendix A.1. Determining the Derivatives of the Dynamic Model

According to Equation (2), this term can be expressed as

$$\tau_{inertia} = \begin{bmatrix} B(q) & S(q) \\ S^T(q) & J_m \end{bmatrix} \begin{bmatrix} \ddot{q} \\ \ddot{\theta} \end{bmatrix} \quad (A1)$$

Since J and S are constant, their derivatives are zero.

Deriving Equation (A1) gives

$$\dot{\tau}_{inertia} = \begin{bmatrix} \dot{B}(q) & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \ddot{q} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} B(q) & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dddot{q} \\ \dddot{\theta} \end{bmatrix} \quad (A2)$$

Deriving it again gives

$$\ddot{\tau}_{inertia} = \begin{bmatrix} \ddot{B}(q) & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \ddot{q} \\ \ddot{\theta} \end{bmatrix} + 2 \begin{bmatrix} \dot{B}(q) & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \ddot{q} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} B(q) & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} q^{(4)} \\ \theta^{(4)} \end{bmatrix} \quad (A3)$$

The first and second derivatives of the matrix can be obtained for Equation (22):

$$\dot{B}(q) = \begin{bmatrix} -2a_3 \sin(q_2) \dot{q}_2 & -a_3 \sin(q_2) \dot{q}_2 \\ -a_3 \sin(q_2) \dot{q}_2 & 0 \end{bmatrix} \quad (A4)$$

$$\ddot{B}(q) = \begin{bmatrix} -2a_3 (\cos(q_2) \dot{q}_2 + \sin(q_2) \ddot{q}_2) & -a_3 (\cos(q_2) \dot{q}_2 + \sin(q_2) \ddot{q}_2) \\ -a_3 (\cos(q_2) \dot{q}_2 + \sin(q_2) \ddot{q}_2) & 0 \end{bmatrix} \quad (A5)$$

In summary, the first derivative of the inertia matrix depends on the positions and velocities of the joints. Its second derivative also depends on acceleration. The total inertia torques depend on up to the fourth derivative of the position.

Appendix A.2. The Centrifugal and Coriolis Terms

Given Equations (21) and (22), for a two-degrees-of-freedom robot, the torque related to centrifugal and Coriolis forces can be expressed as

$$\tau_C = c(q, \dot{q}) \dot{q} \quad (A6)$$

By deriving, we obtain

$$\dot{\tau}_C = \dot{c}(q, \dot{q}, \ddot{q}) \dot{q} + c(q, \dot{q}) \ddot{q} \quad (A7)$$

The first derivative of the matrix c is

$$\dot{c} = \begin{bmatrix} -a_3 (\cos(q_2) \dot{q}_2^2 + \sin(q_2) \ddot{q}_2) & -a_3 (\cos(q_2) \dot{q}_2^2 + \sin(q_2) \ddot{q}_2) \\ a_3 (\cos(q_2) \dot{q}_1 \dot{q}_2 + \sin(q_2) \ddot{q}_1) & 0 \end{bmatrix} \quad (A8)$$

We then introduce the following:

$$aux1 = \left(-\sin(q_2) \dot{q}_2^3 + 3\cos(q_2) \dot{q}_2 \ddot{q}_2 + \sin(q_2) \ddot{q}_2^2 \right) \quad (A9)$$

$$aux2 = \left(\sin(q_2) \dot{q}_1 \dot{q}_2^2 - \cos(q_2) (\dot{q}_1 \ddot{q}_2 + \dot{q}_2 \ddot{q}_1) - \cos(q_2) \dot{q}_2 \ddot{q}_1 + \sin(q_2) \ddot{q}_1^2 \right) \quad (A10)$$

$$aux2 = \left(\sin(q_2) \dot{q}_1 \dot{q}_2^2 - \cos(q_2) (\dot{q}_1 \ddot{q}_2 + 2\dot{q}_2 \ddot{q}_1) + \sin(q_2) \ddot{q}_1^2 \right) \quad (A11)$$

By deriving (A8) and introducing (A9) into (A11), we obtain

$$\ddot{c} = -a3 * \begin{bmatrix} aux1 & aux1 \\ aux2 & 0 \end{bmatrix} \quad (A12)$$

The first derivative of the centrifugal and Coriolis terms depends on the positions, velocities, and accelerations of the joints. Its second derivative depends on the jerks.

Appendix A.3. The Gravity Term

The first derivative is obtained by deriving the gravity term using Equation (28):

$$\dot{G} = g \begin{bmatrix} -m_1 d_1 \sin(q_1) \dot{q}_1 - m_{r2} L_1 \sin(q_1) \dot{q}_1 - m_2 (L_1 \sin(q_1) \dot{q}_1 - d_2 \sin(q_1 + q_2) (\dot{q}_1 + \dot{q}_2)) \\ -m_2 d_2 \sin(q_1 + q_2) (\dot{q}_1 + \dot{q}_2) \\ 0 \\ 0 \end{bmatrix} \quad (A13)$$

We then introduce the intermediate variables:

$$\begin{aligned} g_{11} &= (m_1 d_1 + m_{r2} L_1) (\sin(q_1) \ddot{q}_1 + \cos(q_1) \dot{q}_1^2) \\ g_{12} &= m_2 L_1 (\sin(q_1) \ddot{q}_1 + \cos(q_1) \dot{q}_1^2) \\ g_{13} &= m_2 d_2 (\sin(q_1 + q_2) (\ddot{q}_1 + \ddot{q}_2) + \cos(q_1 + q_2) (\dot{q}_1^2 + \dot{q}_2^2 + 2\dot{q}_1 \dot{q}_2)) \end{aligned} \quad (A14)$$

The second derivative of the gravity torque is obtained by deriving Equation (A13) and substituting with Equation (A14):

$$\ddot{G} = g \begin{bmatrix} -g_{11} - g_{12} - g_{13} \\ -g_{13} \\ 0 \\ 0 \end{bmatrix} \quad (A15)$$

The first derivative of the gravity term depends on the positions and the velocities of the joints. Its second derivative depends on acceleration.

References

1. Harmonic Drive SE. Robotics, Handling & Automation, Brochure. Available online: <https://harmonicdrive.de/en/applications/robotics-handling-automation> (accessed on 2 June 2024).
2. Pratt, G.A.; Williamson, M.M. Series elastic actuators. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems, Pittsburgh, PA, USA, 5–9 August 1995; Volume 1, pp. 399–406.
3. Pratt, J.E.; Krupp, B.T. Series elastic actuators for legged robots. *SPIE* **2004**, *5422*, 135–144. [[CrossRef](#)]
4. Guizzo, F.; Ackerman, E. How Rethink Robotics Built Its New Baxter Robot Worker. *IEEE Spectrum*. Available online: <https://spectrum.ieee.org/rethink-robotics-baxter-robot-factory-worker> (accessed on 2 June 2024).
5. Barrientos, A.; Peñín, L.F.; Balaguer, C.; Aracil, R. *Fundamentos de Robótica*, 2nd ed.; McGraw-Hill: Madrid, Spain, 1997.
6. Spong, M.W.; Hutchinson, S.; Vidyasagar, M. *Robot Modeling and Control*, 1st ed.; John Wiley & Sons: Hoboken, NJ, USA, 2006.
7. Siciliano, B.; Sciavicco, L.; Villani, L.; Oriolo, G. *Robotics, Modelling, Planning and Control*, 1st ed.; Springer: London, UK, 2009.
8. De Luca, A. A review on the control of flexible joint manipulators. In *IROS: Workshop on Soft Robotic Modeling and Control, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Madrid, Spain, 1–5 October 2018*; Sapienza Università di Roma: Roma, Italy, 2018.
9. Vallery, H.; Veneman, J.; Van Asseldonk, E.; Ekkelenkamp, R.; Buss, M.; Van Der Kooij, H. Compliant Actuation of Rehabilitation Robots. *IEEE Robot. Autom. Mag.* **2008**, *15*, 60–69. [[CrossRef](#)]
10. Hyun, D.J.; Lim, H.; Park, S.; Nam, S. Singular Wire-Driven Series Elastic Actuation with Force Control for a Waist Assistive Exoskeleton, H-WEXv2. *IEEE/ASME Trans. Mechatron.* **2020**, *25*, 1026–1035. [[CrossRef](#)]
11. Spong, M.W. Modeling and Control of Elastic Joint Robots. *ASME J. Dyn. Syst. Meas. Control* **1987**, *109*, 310–319. [[CrossRef](#)]
12. Spong, M.W. *Control of Robots with Flexible Joints: A Survey*; Coordinated Science Laboratory Report no. UILU-ENG-90-2203, DC-116; University of Illinois: Urbana, IL, USA, 1990.
13. Tomei, P. A simple PD controller for robots with elastic joints. *IEEE Trans. Autom. Control* **1991**, *36*, 1208–1213. [[CrossRef](#)]
14. De Luca, A.; Flacco, F. A PD-type regulator with exact gravity cancellation for robots with flexible joints. In Proceedings of the IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011.

15. Albu-Schaffer, A.; Ott, C.; Hirzinger, G. A unified passivity-based control framework for position, torque and impedance control of flexible joint robots. *Int. J. Robot. Res.* **2007**, *24*, 23–39. [[CrossRef](#)]
16. Ott, C. *Cartesian Impedance Control of Redundant and Flexible-Joint Robots*; Springer: Berlin, Germany, 2008.
17. De Luca, A. Elastic joints. In *Theory of Robot Control*; de Witt, C.C., Siciliano, B., Bastin, G., Eds.; Springer: London, UK, 1996; pp. 179–218.
18. De Luca, A. Elastic joints. In *Springer Handbook of Robotics*; Siciliano, B., Khatib, O., Eds.; Springer: Berlin, Germany, 2016; pp. 243–263.
19. Ghahrmani, N.O.; Towhidkhal, F. Constrained incremental predictive controller design for a flexible joint robot. *ISA Trans.* **2009**, *48*, 321–326. [[CrossRef](#)]
20. Iskandar, M.; Van Ommeren, C.; Wu, X.; Albu-Schäffer, A.; Dietrich, A. Model Predictive Control Applied to Different Time-Scale Dynamics of Flexible Joint Robots. *IEEE Robot. Autom. Lett.* **2023**, *8*, 672–679. [[CrossRef](#)]
21. Ott, C.; Beck, F.; Keppler, M. An Experimental Study on MPC based Joint Torque Control for Flexible Joint Robots. In Proceedings of the 13th IFAC Symposium on Robot Control (SYROCO), Matsumoto, Japan, 17–20 October 2022. [[CrossRef](#)]
22. Zaare, S.; Soltanpour, M.R.; Moattari, M. Voltage based sliding mode control of flexible joint robot manipulators in presence of uncertainties. *Robot. Auton. Syst.* **2019**, *118*, 204–219. [[CrossRef](#)]
23. Rsetam, K.; Cao, Z.; Man, Z. Cascaded-extended-state-observer-based sliding-mode control for underactuated flexible joint robot. *IEEE Trans. Ind. Electron.* **2020**, *67*, 10822–10832. [[CrossRef](#)]
24. Tuan, H.M.; Sanfilippo, F.; Hao, N.V. A novel adaptive sliding mode controller for a 2-DOF elastic robotic arm. *Robotics* **2022**, *11*, 47. [[CrossRef](#)]
25. Fateh, M.M. Robust control of flexible-joint robots using voltage control strategy. *Nonlinear Dyn.* **2012**, *67*, 1525–1537. [[CrossRef](#)]
26. Izadbakhsh, A. Robust control design for rigid-link flexible-joint electrically driven robot subjected to constraint: Theory and experimental verification. *Nonlinear Dyn.* **2016**, *85*, 751–765. [[CrossRef](#)]
27. Ullah, H.; Malik, F.M.; Raza, A.; Mazhar, N.; Khan, R.; Saeed, A.; Ahmad, I. Robust Output Feedback Control of Single-Link Flexible-Joint Robot Manipulator with Matched Disturbances Using High Gain Observer. *Sensors* **2021**, *21*, 3252. [[CrossRef](#)] [[PubMed](#)]
28. Kokotovic, P.; Khalil, H.K.; O'Reilly, J. *Singular Perturbation Methods in Control: Analysis and Design*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 1999; Reprint edition.
29. Vaidyanathan, S.; Azar, A.T. An introduction to backstepping control. In *Backstepping Control of Nonlinear Dynamical Systems*; Academic Press: Cambridge, MA, USA, 2021; pp. 1–32. [[CrossRef](#)]
30. De Luca, A.; Siciliano, B.; Zollo, L. PD control with on-line gravity compensation for robots with elastic joints: Theory and experiments. *Automatica* **2005**, *41*, 809–1819. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.