Trajectory Tracking Controller for a 4-DoF Flexible Joint Robot Arm

Dimitar Ho
TU Darmstadt

Viktor Kisner
TU Darmstadt

Abstract

In this paper, we present a trajectory tracking controller for the flexible joint robot arm BioRob. The controller compensates gravity and stiction torques with a feedforward part and joint and motor angle tracking errors with a linear feedback part. In order to parametrize the controller, system identification methods have been designed to estimate the gravity vector, elastic transmission and stiction model. We show that the proposed trajectory tracking controller outperforms other control laws on the BioRob system in terms of the overall RMS trajectory tracking error.

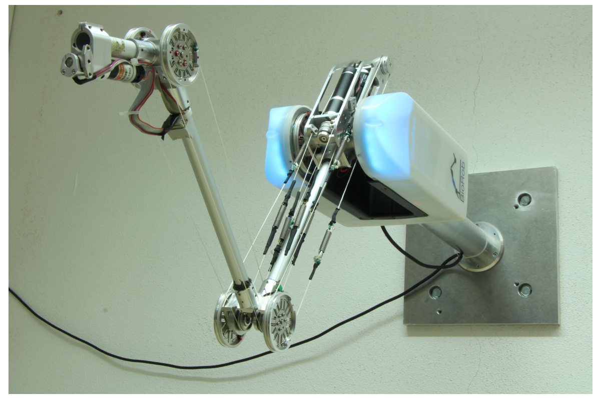

Figure 1: BioRob, a flexbile joint robot with four revolute joint. Its joint flexibility comes from elastic transmission devices consisting of pulleys, cables, and incorporated springs.

1 Introduction

Flexible joint robot arms come with elastic elements incorporated in their transmission devices. A typical example of such robots is the BioRob arm, see Figure 1. It is a lightweight robotic manipulator with four revolute joints coupled with actuators by elastic transmission devices, which consist of cables, pulleys and translational springs.

BioRob's joint flexibility has proven beneficial for safety in close human interaction, as the robot links are inertially decoupled from the actuators. Therefore, the robot links have a lower kinetic energy in case of an accidental collision with a human. Another advantage involves the ablility to store potential energy in the translational springs, allowing for peak velocities in throwing tasks.

On the other hand, joint flexibility introduces great challenges in modeling, identification and control of the arm. From the viewpoint of system theory, the elastic transmission devices double the manipulator's system order. Since they furthermore exhibit a nonlinear behaviour and, particularly, the actuators involve friction, a highly complicated model is required to describe BioRob's dynamics. To identify the parameters of the model only data from rotary encoders on motors and joints and current sensors in motors are available. As the robot links are inertially decoupled from the actuators, BioRob poses an underactuated system, which means that the manipulator is no longer capable to follow arbitrary trajectories in the configuration space. As a result, oscillations of the robot links that occur due to the joint flexibility are very difficult to damp.

In this paper we address the trajectory tracking problem for the BioRob arm. We adapt a control algorithm presented in [1], which combines a modelbased feedforward control signal with a linear full state feedback. To compensate for friction we add an additional feedback path to the control algorithm, which is endowed with a hysteresis to avoid chattering phenomena. The control algorithm is designed using a mathematical model presented in [2] that describes BioRob's dynamics in the so-called joint space. To identify the necessary model parameters we have applied system identification approaches, which we have customized for the BioRob arm.

The remainder of this paper is organized as follows: The BioRob arm and its dynamical equations are presented in Section 2. The control algorithm for trajectory tracking is derived in Section 3. In Section 4 we introduce a special procedure to identify all system parameters used in the control algorithm. The performance of our control algorithm is evaluated in comparision with other control laws in Section 5. Section 6 summarizes and reflects results of our proposed controller. The Appendices A and B show open problems regarding BioRob's hardware and software, as well as other related work that has been done during this project.

2 Dynamical System

We begin with a concise descripton of the BioRob arm. Afterwards we introduce two different sets of generalized variables to facilitate the description of BioRob's dynamics. Finally we present BioRob's equations of motion in the end of this section. For a more detailed analysis the reader is referred to [2].

2.1 System Description

The BioRob arm shown in Figure 1 is a lightweight flexible joint robot. It has four revolute joints coupled with electrical actuators by cable and pulley mechanisms. The cables have built-in translational springs causing joint flexibility. The pulleys are mounted on joints and actuators. The electrical actuators consist of DC-motors and reduction gears. To reduce the inertia of the robot links they are placed near the base of the manipulator. That is, the actuators of the first and second joint are attached to the first link, whereas these of the third and fourth joint are on the second link of the arm. Since the actuator of the fourth joint is located on the second link, an idler pulley is necessary to drive the fourth joint. This introduces a kinematic coupling between the third actuator and joint angles and the fourth joint angle, what is made precise later. On BioRob's fifth link an end effector consisting of a small DC motor with a gripper is mounted, which for example may carry a table tennis racket. However, this degree of freedom is not used in this work. For sensing its position BioRob has rotary encoders on the DC motors and joints with a resolution of 11 and 12 bits. Additionally, the DC motors are equipped with current sensors.

2.2 Actuator and Joint Space

For dynamic analysis, two different sets of generalized variables are introduced, which form the so-called actuator and joint space. The joint space is of particular interest, as it allows for writing BioRob's dynamic equations in a more transparent manner.

The actuator space consists of the coordinates ${}^{e}\theta$ and ${}^{e}q$. The coordinates ${}^{e}\theta$ pose physical quantities, namely the motor angles as reflected through the reduction gears to the actuator side pulleys. Contrarily, the coordinates ${}^{e}q$ are virtual quantities, which can be understood as the joint angles reflected through the elastic transmission devices to the actuator side pulleys.

Similarly, in the joint space ${}^{j}\theta$ and q denote generalized coordinates. The joint angles q are physical quantities, whereas the actuator angles ${}^{j}\theta$ are virtual quantities reflected through the elastic transmission devices to the joint side pulleys.

The transformation matrix that reflects the actuator angle into joint space is given by

$$\mathbf{J}_{\mathrm{t}} = \begin{pmatrix} \frac{r_1}{R_1} & 0 & 0 & 0 \\ 0 & \frac{r_2}{R_2} & 0 & 0 \\ 0 & 0 & \frac{r_3}{R_3} & 0 \\ 0 & 0 & -\frac{r_3}{R_4}\frac{r_{4\mathrm{d}3}}{R_4} & \frac{r_4}{R_4} \end{pmatrix}, \tag{1}$$

where the diagonal elements are the conversion rates of the cable pulley mechanisms. r_i and R_i denote the radii of the first joint's actuator and joint side pulley. The non diagonal form is a result of the kinematic coupling between the third and fourth joint. Thereby, the radius of the idler pulley is denoted by $r_{4\mathrm{d}3}$. The transformation matrix is derived for the case when the springs exhibit their prestretched length and force. An illustrative derivation of the transmission matrix can be found in [2].

In terms of equations the transformation of the actuator angle into joint space is given by

$${}^{j}\theta = \mathbf{J}_{\mathrm{t}}\,{}^{e}\theta \tag{2}$$

and, conversely, the joint angle is reflected into the actuator space by

$${}^{e}q = \mathbf{J}_{\mathrm{t}}^{-1} q\,. \tag{3}$$

Due to BioRob's joint flexibility the actuator angle reflected into joint space ${}^{j}\theta$ does normally not coincide with the joint angle q and, vice versa, the joint angle viewed in the actuator space ${}^{e}q$ is not the same as the actuator angle ${}^{e}\theta$. Nevertheless the definition of the joint and actuator space allows for interpreting the differences of the angles as the deflections of virtual torsional springs, placed at the corresponding joints or actuators.

Additionally, we define the generalized torques in both spaces, which are applied about the axes of the generalized coordinates. In the actuator space they are denoted by ${}^{e}\tau_{\mathrm{m}}$ and τ_{e}. The former act about ${}^{e}\theta$ and can be perceived as the physical actuator torques, as they are the motor torques reflected through the reduction gears. The latter act about ${}^{e}q$ and are measureable as the torques of the elastic transmission devices on the actuators. They can also be visualized as torques exterted on joints by virtual torsional springs, which are reflected through the transmission devices. Therefore, the transformation matrix can be derived with the principle of virtual work [2].

In a similar fashion, the torques for the joint space are defined as ${}^{j}\tau_{\mathrm{m}}$ and ${}^{j}\tau_{\mathrm{e}}$. While ${}^{j}\tau_{\mathrm{m}}$ act about ${}^{j}\theta$ and are viewed as reflected actuator torques, ${}^{j}\tau_{\mathrm{e}}$ act about q and have the meaning of torques between two adjacent links that are exerted from virtual torsional springs directly placed in the corresponding joints.

The relation between the torques of both spaces is described by

$$ {}^{j}\tau_{\mathrm{m}} = \mathbf{J}_{\mathrm{t}}^{-\mathrm{T}}\, {}^{e}\tau_{\mathrm{m}} \tag{4} $$

$$ {}^{j}\tau_{\mathrm{e}} = \mathbf{J}_{\mathrm{t}}^{-\mathrm{T}}\, {}^{e}\tau_{\mathrm{e}} \tag{5} $$

where $\mathbf{J}_{\mathrm{t}}^{-\mathrm{T}}$ denotes the inverted and transposed matrix $\mathbf{J}_{\mathrm{t}}$. The difference between the coordinates of the joint space ${}^{j}\theta - q$ can be interpreted as the deflection of virtual torsional springs with the spring torques ${}^{j}\tau_{\mathrm{e}}$ acting between the links of the corresponding joints. As a result, the joint space allows to model BioRob as a rigid link robot with series elastic actuators placed in the joints.

2.3 Robot Dynamics

Next, we will present the Lagrangian equations of BioRob in the joint space. To do so, three assumptions are made, see [2] and [3]. First, it is assumed that the center of mass of each actuator is on its rotation axis. In this case, the gravitational potential energy is independent of the actuator angles. Second, it is assumed that the kinetic energy of each actuator rotor is due only to its own spinning, implying that gyroscopic effects are neglected. Third, the kinetic energy of the cables and springs is also assumed to be negligable.

The dynamics of the BioRob arm in joint space are given by the following set of equations

$$ {}^{j}\mathbf{I}_{\mathrm{m}}\, {}^{j}\ddot{\theta} + {}^{j}\mathbf{D}_{\mathrm{m}}\, {}^{j}\dot{\theta} + {}^{j}\tau_{\mathrm{e}} + {}^{j}\tau_{\mathrm{s}} = {}^{j}\tau_{\mathrm{m}} \tag{6} $$

$$ \mathbf{M}(q)\,\ddot{q} + \mathbf{C}(q,\dot{q})\,\dot{q} + \mathbf{D}\,\dot{q} + g(q) = {}^{j}\tau_{\mathrm{e}} \tag{7} $$

$$ {}^{j}\tau_{\mathrm{k}}({}^{j}\theta - q) + {}^{j}\tau_{\mathrm{d}}({}^{j}\dot{\theta} - \dot{q}) = {}^{j}\tau_{\mathrm{e}} \tag{8} $$

where Equations (6) and (7) describe the actuator and robot link dynamics. Equation (8) illustrates the elastic actuator torques, which are torques transmitted from the actuator to the joint side. With ${}^{j}\mathbf{I}_{\mathrm{m}}$ and ${}^{j}\mathbf{D}_{\mathrm{m}}$ we denote the actuator rotor inertia and damping matrix. The matrices $\mathbf{M}(q)$, $\mathbf{C}(q,\dot{q})$ and $\mathbf{D}$ denote the link inertia, the Coriolis and centripetal and the damping matrix, while $g(q)$ denotes the gravity vector. By ${}^{j}\tau_{\mathrm{k}}$ and ${}^{j}\tau_{\mathrm{d}}$ we represent the elastic actuator stiffness and damping vectors. ${}^{j}\tau_{\mathrm{s}}$ and ${}^{j}\tau_{\mathrm{m}}$ denote the friction and the motor torque vectors, which are non-conservative generalized forces. Note, that the above mentioned parameters with a superscripted j are reflected to the joint space.

The synthesis of the control algorithm is based on the terms ${}^{j}\tau_{\mathrm{k}}({}^{j}\theta - q)$, $g(q)$ and ${}^{j}\tau_{\mathrm{s}}$ of the robot's dynamic equations, which are spelled out in the following. The elastic actuator stiffness vector can be rewritten as

$$ {}^{j}\tau_{\mathrm{k}}({}^{j}\theta - q) = \mathbf{J}_{\mathrm{t}}^{-\mathrm{T}}\,\mathbf{R}\,F_{\mathrm{k}}(\mathbf{R}\,\mathbf{J}_{\mathrm{t}}^{-1}\,({}^{j}\theta - q)) \tag{9} $$

where F_{k} denotes a vector function with the physical spring characteristics. The matrix $\mathbf{R}$ which consists of the joint side pulley radii is given by

$$ \mathbf{R} = \mathrm{diag}(r_1, r_2, r_3, r_4) \tag{10} $$

with diag standing for diagonal form. The argument of the spring force vector $\mathbf{R}\,\mathbf{J}_{\mathrm{t}}^{-1}\,({}^{j}\theta - q)$ is the elongation of the translational cable springs. $\mathbf{R}\,F_{\mathrm{k}}(\cdot)$ represents the elastic actuator input torques with respect to the elastic actuator space ${}^{e}\tau_{\mathrm{e}}$, when ${}^{j}\tau_{\mathrm{d}}({}^{j}\dot{\theta} - \dot{q})$ is negligable or the robot does not move.

The gravitational torques in joint space are given by

$$ \mathbf{g}(q) = \begin{bmatrix} s_1 & c_1 & s_1 s_2 & s_1 s_{23} & s_1 s_{234} \\ 0 & 0 & c_1 c_2 & c_1 c_{23} & c_1 c_{234} \\ 0 & 0 & 0 & c_1 c_{23} & c_1 c_{234} \\ 0 & 0 & 0 & 0 & c_1 c_{234} \end{bmatrix} \begin{bmatrix} -\beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \end{bmatrix} \tag{11} $$

whereas the letters s and c abbreviate the functions sin and cos. The letters' indexed numbers denote the sum of joint angles used as arguments, i.e. s_{23} stands for $\sin(q_2 + q_3)$. The parameters β_i denote gravitational terms, which will be estimated in Section 4.

In BioRob's equations of motion friction is only considered in the actuator dynamics. This coincides with the observation that the most friction is exerted in the transmission devices of the actuators, whereas the friction in the joints and the joint side pulleys is negligible. The friction with respect to the actuator space is modeled by a piecewise vector function as follows

$$ {}^e\tau_{\mathrm{s}} = \begin{cases} \operatorname{sgn}({}^e\tau_{\mathrm{ext}}) \min(|{}^e\tau_{\mathrm{ext}}|, {}^e\widehat{\tau}_{\mathrm{s}}({}^e\sigma)) & |{}^e\dot{\theta}| = 0 \\ \operatorname{sgn}({}^e\dot{\theta})\, {}^e\widehat{\tau}_{\mathrm{s}}({}^e\sigma) & |{}^e\dot{\theta}| \neq 0 \end{cases} \tag{12} $$

where sgn and min are vector functions with the sign of each component of the argument and the minimum of each argument's component. At rest, the friction acts opposite to the external torque and is given by

$$ {}^e\tau_{\mathrm{ext}} = {}^e\tau_{\mathrm{m}} - {}^e g\,. \tag{13} $$

Furthermore, we model the friction's absolute value ${}^e\widehat{\tau}_{\mathrm{s}}({}^e\sigma) > 0$ to depend on the torsional stress in the actuator joints ${}^e\sigma$, which we model as

$$ {}^e\sigma = {}^e g, \tag{14} $$

since the gravity causes the main torsional joint stress at any resting position of the robot. Notice, that ${}^e g$ is the gravity reflected to the actuator space, i.e. $\mathbf{J}_{\mathrm{t}}^{\mathrm{T}}\, g(q)$. Approximations of the vector function ${}^e\widehat{\tau}_{\mathrm{s}}({}^e\sigma)$ will be derived from static experiments, which will be presented in Section 4. In motion, friction opposes the actuators' angular velocity with the value of the stiction function.

3 Control Algorithm

Given BioRob's dynamic equations we propose a control algorithm for tracking a desired link trajectory q_{d}. It combines a modelbased feedforward and a linear full state feedback similar to [1]. An overview of the control algorithm is illustrated in Figure 2.

In the following derivation we assume that we only know the terms ${}^j\tau_{\mathrm{k}}({}^j\theta - q)$, $g(q)$ and ${}^j\tau_{\mathrm{s}}$ of the robot's dynamic equations. We also assume that the desired link trajectory q_{d} is at least thrice differentiable.

We begin the derivation of the trajectory tracking controller by specifying a desired trajectory for the full state space consisting of q, $\dot{q}$, ${}^j\theta$ and ${}^j\dot{\theta}$. As a desired link trajectory q_{d} is already provided it remains to obtain a desired motor trajectory, which will be denoted by ${}^j\theta_{\mathrm{d}}$. It has to be consistent with q_{d} in terms of the dynamic equations. Thereto it is obtained by rearranging Equation (7) as

$$ \begin{aligned} {}^j\theta_{\mathrm{d}} = {}^j\tau_{\mathrm{k}}{}^{-1}(\mathbf{M}(q_{\mathrm{d}})\,\ddot{q}_{\mathrm{d}} + \mathbf{C}(q_{\mathrm{d}}, \dot{q}_{\mathrm{d}})\,\dot{q}_{\mathrm{d}} + {} \\ \mathbf{D}\,\dot{q}_{\mathrm{d}} + g(q_{\mathrm{d}})) + q_{\mathrm{d}} \end{aligned} \tag{15} $$

where the spring damping ${}^j\tau_{\mathrm{d}}$ is neglected. Since $\mathbf{M}(q_{\mathrm{d}})$, $\mathbf{C}(q_{\mathrm{d}}, \dot{q}_{\mathrm{d}})$ and $\mathbf{D}$ are unknown we approximate ${}^j\theta_{\mathrm{d}}$ with

$$ {}^j\theta_{\mathrm{d}} = q_{\mathrm{d}} + {}^j\tau_{\mathrm{k}}{}^{-1}(g(q_{\mathrm{d}})) \tag{16} $$

what coincides with Equation (15) for constant trajectories.

Next, we design the feedforward control. Its task is to maintain BioRob roughly along the desired trajectory. If all terms of the dynamic equations are known the feedforward control can easily be determined with the inverse dynamics algorithm

$$ {}^j\tau_{\mathrm{m_d}} = {}^j\mathbf{I}_{\mathrm{m}}\, {}^j\ddot{\theta}_{\mathrm{d}} + {}^j\mathbf{D}_{\mathrm{m}}\, {}^j\dot{\theta}_{\mathrm{d}} + {}^j\tau_{\mathrm{k}} \tag{17} $$

which is obtained by inserting q_{d} and ${}^j\theta_{\mathrm{d}}$ in Equation (6). The received ${}^j\tau_{\mathrm{m_d}}$ denotes the computed torque. However, according to the assumptions the parameters ${}^j\mathbf{I}_{\mathrm{m}}$ and ${}^j\mathbf{D}_{\mathrm{m}}$ are unknown. We approximate Equation (17) by

$$ {}^j\tau_{\mathrm{m_d}} = {}^j\tau_{\mathrm{k}}, \tag{18} $$

which can be tansformed into

$$ {}^j\tau_{\mathrm{m_d}} = g(q_{\mathrm{d}}) \tag{19} $$

by inserting the approximated desired motor trajectory from Equation (16). Due to the approximation the feedforward only compensates the gravity load along the desired joint trajectory. It should be noted that the above simplifications are exact in case of constant trajectories.

Due to the above approximations the computed torque is not sufficient to steer BioRob along the desired trajectory. To cope with the trajectory error, i.e. the difference between the desired and the current trajectory, a full state feedback is necessary. Its task is to stabilize the robot along the reference trajectory. In this work a linear full state feedback compensator is applied. Altogether with the computed torque from Equation (19) the control algorithm is given by

$$ \begin{aligned} {}^j\tau_{\mathrm{m}} = \mathbf{K}_{\mathrm{P_q}}\,(q_{\mathrm{d}} - q) + \mathbf{K}_{\mathrm{D_{\dot{q}}}}\,(\dot{q}_{\mathrm{d}} - \dot{q}) + {} \\ \mathbf{K}_{\mathrm{P_\theta}}\,({}^j\theta_{\mathrm{d}} - {}^j\theta) + \mathbf{K}_{\mathrm{D_{\dot{\theta}}}}\,({}^j\dot{\theta}_{\mathrm{d}} - {}^j\dot{\theta}) + g(q_{\mathrm{d}}), \end{aligned} \tag{20} $$

where $\mathbf{K}_{\mathrm{P_q}}$ and $\mathbf{K}_{\mathrm{P_\theta}}$ are proportional and $\mathbf{K}_{\mathrm{D_{\dot{q}}}}$ and $\mathbf{K}_{\mathrm{D_{\dot{\theta}}}}$ derivative diagonal gain matrices. For setpoint tracking a stability proof using a Lyapunov argument and LaSalle's invariance theorem is given by [1].

The transmission devices of BioRob's actuators exert significant friction, which causes a poor tracking performance, when not considered in the control algorithm. Therefore, the control algorithm, Equation (20), is augmented with an additional feedback signal to alleviate friction. Its design is based on the friction model proposed in Equation (12). The friction compensation is determined in actuator space and for each actuator seperately using automata as depicted in Figure 3. The automata consist of four different states s_1,

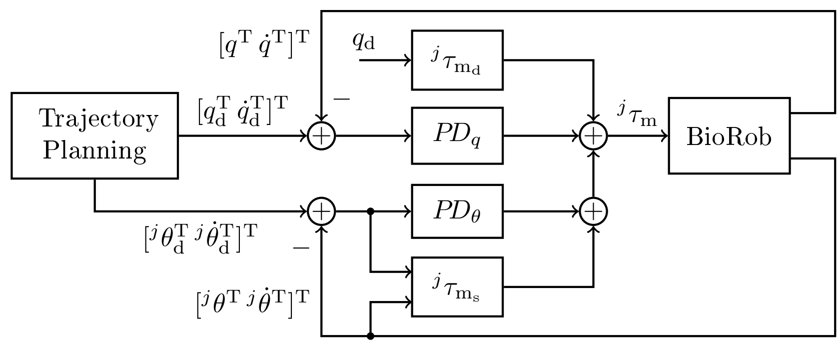

Figure 2: Control algorithm for trajectory tracking. In the beginning of the control task a trajectory for all state variables is planned, i.e., the actuator and joint angles and velocities. Thereafter, a gravity compensation ${}^j\tau_{\mathrm{m_d}}$, a linear full state feedback PD_q and PD_θ, and an additional nonlinear feedback ${}^j\tau_{\mathrm{m_s}}$ to alleviate friction is used to track the desired trajectory.

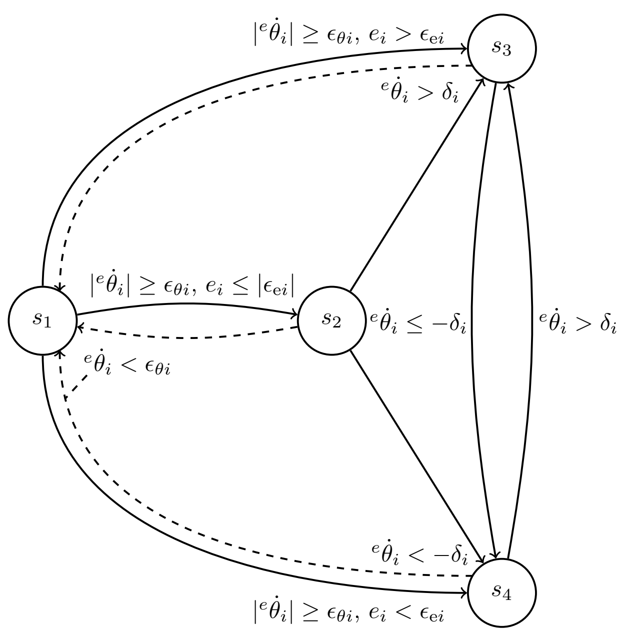

Figure 3: Automaton for friction compensation with hysteresis. The friction compensation torque for each actuator depends on the state of the corresponding automaton. The automaton consists of four states connected by edges with assigned transition conditions. If no transition condition is true, the automaton remains in the current state.

s_2, s_3 and s_4. When the angular velocity of the corresponding actuator ${}^e\dot{\theta}_i$ is below the very small threshold $\epsilon_{\theta i}$, its assigned automaton takes the state s_1. Then the actuator merely moves and the additional torque of the friction compensation

$$ {}^e\tau_{\mathrm{m_s}i} = |{}^e\widehat{\tau}_{\mathrm{s}}({}^e g)|\,\mathrm{sat}_1(e_i/\epsilon_{\mathrm{e}i}) \tag{21} $$

acts into the direction of the error between the desired and the current acutator position

$$ e_i = {}^e\theta_{\mathrm{d}i} - {}^e\theta_i\,, \tag{22} $$

whereas the direction is determined using the saturation function

$$ \mathrm{sat}_1(x) = \begin{cases} \mathrm{sgn}(x) & |x| \geq 1 \\ x & |x| < 1 \end{cases} \tag{23} $$

instead of a sgn function to avoid jumps in the actuator torque. It is also possible to determine the direction of stiction by using the external torque, as it is described in the friction model (12). Since, however, the torque from the PD controller for the actuator angle is very high in relation to the gravity torque, it is negligable. When ${}^e\theta_i$ is above the threshold $\epsilon_{\theta i}$ the automaton changes into state s_2, s_3 or s_4 depending on the error e_i. Contrariwise, if ${}^e\theta_i$ drops below $\epsilon_{\theta i}$ the automaton changes back into s_1. In state s_2 the additional torque is

$$ {}^e\tau_{\mathrm{m_s}i} = |{}^e\widehat{\tau}_{\mathrm{s}}({}^e g)|\,\mathrm{sat}_1({}^e\dot{\theta}_i/\delta_i)\,. \tag{24} $$

The automaton changes into this state when the norm of the error, Equation (22), is below the threshold $\epsilon_{\mathrm{e}i}$. If the error is above or below $\epsilon_{\mathrm{e}i}$ the automaton changes from s_1 into state s_3 or s_4. In s_3 the friction compensation torque is chosen as

$$ {}^e\tau_{\mathrm{m_s}i} = |{}^e\widehat{\tau}_{\mathrm{s}}({}^e g)|\,\mathrm{sat}_2({}^e\dot{\theta}_i/\delta_i) \tag{25} $$

whereby the direction is determined using the actuator velocity with function sat_2 defined as

$$\mathrm{sat}_2(x) = \begin{cases} 1 & x \geq 0 \\ 2x+1 & -1 < x < 0 \\ -1 & x \leq -1 \end{cases} . \tag{26}$$

The friction compensation in state q_4 is nearly the same as in state q_3, however, only sat_3 given by

$$\mathrm{sat}_3(x) = \begin{cases} 1 & x \geq 1 \\ 2x-1 & 0 < x < 1 \\ -1 & x \leq 0 \end{cases} \tag{27}$$

is used instead of sat_2. The automaton changes between s_3 and s_4 depending on the actuator velocity. If it is beneath $-\delta_i$ the automaton changes from s_3 into s_4. Contrariwise, the automaton takes s_4 from s_3 when ${}^e\dot{\theta}_i > \delta_i$ is true. The functions sat_2 and sat_3 form a saturation function with hysteresis to avoid jumps in the friction compensation. When the friction compensation is calculated it has to be transformed into joint space with matrix $\mathbf{J}_\mathrm{t}^{-\mathrm{T}}$ for the control algorithm. The parameters $\epsilon_{\theta i}$, $\epsilon_{\mathrm{e}i}$, and δ_i are tuning parameters, whereas $\epsilon_{\theta i}$ should always be smaller than δ_i.

4 System Identification

As mentioned before in Section 3, we assume the knowledge of ${}^j\tau_\mathrm{k}({}^j\theta - q)$, $g(q)$ and ${}^j\tau_\mathrm{s}$. The goal of this section will be to demonstrate the process and result of the estimation of those terms. The essential system parameters we have to find out are the values of $\mathbf{J}_\mathrm{t}$, $\mathbf{R}$, β_i, and the vector functions ${}^e\widehat{\tau}_\mathrm{s}({}^e\sigma)$ and F_k.

4.1 Geometric System Parameters

For computing the matrices $\mathbf{J}_\mathrm{t}$ and $\mathbf{R}$ shown in Equation (1) and (10), we measured out the radii of the different pulleys in the robotic system. The results of the measurements are noted in the following table:

r_1	0.0079 m
r_2	0.0121 m
r_3	0.0121 m
r_4	0.0090 m
R_1	0.0245 m
R_2	0.0403 m
R_3	0.0306 m
R_4	0.0250 m
r_{4d3}	0.0120 m

Table 1: Geometric system parameters of BioRob.

4.2 Identification of Nonlinear Force-Elongation Curves of Spring System

Calculating the desired motor trajectory ${}^j\theta_\mathrm{d}$ from Equation (16), we need exact knowledge of the vector function F_k and it's inverse. Due to our modeling of the elastic forces between the motor and the joints as (9), F_k takes the decoupled form

$$F_\mathrm{k}(\mathbf{x}) = \begin{bmatrix} f_{\mathrm{k}1}(x_1) \\ f_{\mathrm{k}2}(x_2) \\ f_{\mathrm{k}3}(x_3) \\ f_{\mathrm{k}4}(x_4) \end{bmatrix} \tag{28}$$

where $f_{\mathrm{k}i}(x_i)$ and x_i are the force-elongation functions and the physical elongation of the spring system in joint i respectively. Using the measurement mechanism as shown in Figure 6 we will show in the following, how we can estimate the functions $f_{\mathrm{k}i}$ through measuring a set of discrete points and using curve-fitting techniques to compute function approximations.

The measurement procedure is performed as follows. Firstly, the robot arm is being controlled with a PID-controller

$$\begin{aligned} {}^j\tau_\mathrm{m} &= {}^j\tau_\mathrm{PID} & (29) \\ &= \mathbf{K}_p\,(q_0 - q) - \mathbf{K}_d\,\dot{q} + \mathbf{K}_i \int_0^t (q_0 - q(\tau))d\tau & (30) \end{aligned}$$

to hold the hanging posture $q = q_0$ shown in Figure 6. In this position, the robot is not experiencing any gravitational torque in his joints, thus $g(q_0) = 0$ and the position $q = q_0, \theta = 0$ is an equilibrium point of the system. While the controller is still running, we continuously increase the weight in the plastic bowl depicted in Figure 6 and collect measurements of the deflection $({}^j\theta - q)$ and weight once the PID-controller has controlled the robot to its original gravitation-free posture q_0. Once the bowl has reached the specified maximum weight, we continue the measurements with then decreasing the weights in the bowl. This measurement procedure has been performed in different deflection directions of the robot arm for measuring different springs of the system. Furthermore, the procedure allows to study the eventual hysteresis effect in the springs. From those experiments we yield l pairs of bowl weights m_l and corresponding deflections $({}^j\theta - q_0)_l$ which we can now use to estimate the different $f_{\mathrm{k}i}$-curves.

Controlling the BioRob with PID controller together with the measurement system attached, the equations for the equilibrium of the system can be derived using

Figure 6: This picture shows the four measurement setups we used for performing our measurements to estimate the spring curves and the stiction model.

the Equations (6 - 9) and inserting the mechanics of the measurement system:

$$
\begin{aligned}
{}^{j}\tau_{\mathrm{e}} + {}^{j}\tau_{\mathrm{s}} &= {}^{j}\tau_{\mathrm{PID}} && (31)\\
{}^{j}\tau_{\mathrm{L}1/2} &= {}^{j}\tau_{\mathrm{e}} && (32)\\
{}^{j}\tau_{\mathrm{k}}({}^{j}\theta - q_0) &= {}^{j}\tau_{\mathrm{e}} && (33)\\
\mathbf{R}^{-1}\,\mathbf{J}_{\mathrm{t}}^{\mathrm{T}}\,{}^{j}\tau_{\mathrm{k}}({}^{j}\theta - q_0) &= F_{\mathrm{k}}(\mathbf{R}\,\mathbf{J}_{\mathrm{t}}^{-1}\,({}^{j}\theta - q)). && (34)
\end{aligned}
$$

Notice, that for yielding Equation (32), we used that $g(q_0) = 0$ and included in the original Equations (7) the term ${}^{j}\tau_{\mathrm{L}1/2}$, which describes what torques are being generated on the joints by the loads and attachment position of our measurement mechanism. In the actuator Equations (6), we have substituted the PID-controller to yield (31) and at equilibrium, ${}^{j}\tau_{\mathrm{PID}}$ takes a constant value coming from the I-part of the controller, which compensates the load and also the stiction forces in ${}^{j}\tau_{\mathrm{s}}$, keeping the robot at the desired position q_0. Further we can eliminate Equations (32) and (33) by substituting in the other Equations (33) and (31) and yield

$$
\begin{aligned}
{}^{j}\tau_{\mathrm{L}1/2} + {}^{j}\tau_{\mathrm{s}} &= {}^{j}\tau_{\mathrm{PID}} && (35)\\
\mathbf{R}^{-1}\,\mathbf{J}_{\mathrm{t}}^{\mathrm{T}}\,{}^{j}\tau_{\mathrm{L}1/2} &= F_{\mathrm{k}}(\mathbf{R}\,\mathbf{J}_{\mathrm{t}}^{-1}\,({}^{j}\theta - q)). && (36)
\end{aligned}
$$

The load torque ${}^{j}\tau_{\mathrm{L}1/2}$ which is being generated from the weights, stretches the springs in different joints, which can be seen in the Equation (36). This equation will be the basis for our estimation of F_{k} and the computation of ${}^{j}\tau_{\mathrm{L}1/2}$ will be crucial. Nonetheless, computing ${}^{j}\tau_{\mathrm{L}1/2}$ through the measurable static torque generated from the PID controller would provide too inaccurate results, since as can be seen in (35) the controller not only compensates the load but also compensates the stiction/friction component ${}^{j}\tau_{\mathrm{s}}$, which is known to be fairly big for our robotic system. The load torque is therefore necessary to be computed directly from the weights and the mechanics/geometry of the measurement setup. Depending on which of the experimental setups is chosen in Figure 6, the load torque ${}^{j}\tau_{\mathrm{L}1/2}$ coming from the measurement mechanism takes one of the following forms:

$$
{}^{j}\tau_{\mathrm{L}1} = \begin{bmatrix} 0 \\ l_2 g \\ l_3 g\cos(\alpha) \\ l_4 g\cos(\alpha) \end{bmatrix} m_l \quad \text{or} \quad {}^{j}\tau_{\mathrm{L}2} = \begin{bmatrix} l_1 g \\ 0 \\ 0 \\ 0 \end{bmatrix} m_l \tag{37}
$$

where $l_1 = 0.73681\,\mathrm{m}$, $l_2 = 0.73681\,\mathrm{m}$, $l_3 = 0.42781\,\mathrm{m}$, $l_4 = 0.1172\,\mathrm{m}$ are lengths of different levers with which the load in the bowl acts on the joints, $g = 9.81\mathrm{m/s}^2$ is the gravitational constant and $\alpha = 5.1°$ is the amount of link torsion in degrees which is found on the second link of the robot.

Using the special form of (28), we can further split the vector Equation (36) into its index components and yield the form

$$
\left(\mathbf{R}^{-1}\,\mathbf{J}_{\mathrm{t}}^{\mathrm{T}}\,{}^{j}\tau_{\mathrm{L}1/2}\right)_i = f_{\mathrm{k}i}(\left(\mathbf{R}\,\mathbf{J}_{\mathrm{t}}^{-1}\,({}^{j}\theta - q)\right)_i). \tag{38}
$$

From each pair of bowl weights m_l and corresponding deflections $({}^{j}\theta - q_0)_l$ from our experiments, we can now compute discrete values for spring forces $\hat{f}_{\mathrm{k}il} = \left(\mathbf{R}^{-1}\,\mathbf{J}_{\mathrm{t}}^{\mathrm{T}}\,{}^{j}\tau_{\mathrm{L}1/2}\right)_{il}$ from (37) and their associated spring elongations $x_{il} = \left(\mathbf{R}\,\mathbf{J}_{\mathrm{t}}^{-1}\,({}^{j}\theta - q_0)\right)_{il}$. With this set of $\hat{f}_{\mathrm{k}il}$ and x_{il} values, we can use curve-fitting methods to estimate the functions $f_{\mathrm{k}i}$. We used nonlinear least-squares method to fit polynomial functions to the data and yield the estimation results presented in Figure 4.

Notice that the measurement data displayed in Figure 4 shows, that the springs corresponding to joint 1 and 2 show noticeable nonlinearity. As more detailed described in [2], the nonlinearities in $f_{\mathrm{k}i}$ originate partially from the static slacking effect. Other reasons for the nonlinear characteristic are unmodeled elasticities in the transmission, especially the safety strings incorporated in the springs, which prevent the springs from overstretching. Those strings limit the maximum elongation of the springs and they are considered in our approximations in Figure (4). Furthermore we can see that all springs show hysteresis effects, especially when experiencing higher forces.

4.3 Identification of Stiction/Friction Model

As mentioned in Section 2, the controller is designed to account for the stiction and friction in the robot and therefore it is necessary to estimate the vector function ${}^e\hat{\tau}_{\mathrm{s}}({}^e\sigma)$, which describes the amount of stiction and friction torque that acts on the actuator side with respect to the static joint stress. Since this torque is reflected to the actuator state space, we can further assume, that the friction in a motor joint only depends on the stress experienced in the same motor joint. Therefore the torque ${}^e\hat{\tau}_{\mathrm{s}}({}^e\sigma)$ takes the decoupled form

$$ {}^e\hat{\tau}_{\mathrm{s}}({}^e\sigma) = \begin{bmatrix} {}^e\hat{\tau}_{\mathrm{s1}}({}^e\sigma_1) \\ {}^e\hat{\tau}_{\mathrm{s2}}({}^e\sigma_2) \\ {}^e\hat{\tau}_{\mathrm{s3}}({}^e\sigma_3) \\ {}^e\hat{\tau}_{\mathrm{s4}}({}^e\sigma_4) \end{bmatrix}. \tag{39} $$

To estimate this function, we will use the same measurement procedure and mechanism shown in Figure 6. However, instead of measuring for each weigth m_l the deflection, we will measure the static torque of the PID-controller at equilibrium and compute the load torque ${}^e\tau_{\mathrm{L1/2}}$ originating from the corresponding weights m_l. The idea is, to measure a set of equilibrium torques of the PID-controller, which stabilize the system at the same load torque ${}^e\tau_{\mathrm{L1/2}}$. Computing the maximum difference between the PID-torques and the corresponding load torque, we can approximate the maximum stiction torque for the associated joint stress induced by the load torque. This procedure will be explained now more in detail.

Transforming the static equilibrium equations for the joint side (35) to the actuator side and substituting our model of the stiction force from (12) into the equation, we yield:

$$ {}^e\tau_{\mathrm{L1/2}} + \mathrm{sgn}({}^e\tau_{\mathrm{ext}})\min(|{}^e\tau_{\mathrm{ext}}|, {}^e\hat{\tau}_{\mathrm{s}}({}^e\sigma)) = {}^e\tau_{\mathrm{PID}}. \tag{40} $$

Because our robotic system has the measurement mechanism attached producing an additional load torque ${}^e\tau_{\mathrm{L1/2}}$, the external torque ${}^e\tau_{\mathrm{ext}}$ and the torsional stress ${}^e\sigma$ change for our experiment. The external torque is the difference between the PID-controller and the load torque

$$ {}^e\tau_{\mathrm{ext}} = {}^e\tau_{\mathrm{PID}} - {}^eg(q_0) - {}^e\tau_{\mathrm{L1/2}} = {}^e\tau_{\mathrm{PID}} - {}^e\tau_{\mathrm{L1/2}} \tag{41} $$

and the torsional stress ${}^e\sigma$ consists only of the load torque reflected to the actuator joints

$$ {}^e\sigma = {}^eg(q_0) + {}^e\tau_{\mathrm{L1/2}} = {}^e\tau_{\mathrm{L1/2}} \tag{42} $$

since the gravity ${}^eg(q_0)$ is zero in the equilibrium position $q = q_0$. Plugging (42) and (41) into Equation (40), we finally yield

$$ \begin{aligned} &\mathrm{sgn}({}^e\tau_{\mathrm{PID}} - {}^e\tau_{\mathrm{L1/2}})\min(|{}^e\tau_{\mathrm{PID}} - {}^e\tau_{\mathrm{L1/2}}|, {}^e\hat{\tau}_{\mathrm{s}}({}^e\tau_{\mathrm{L1/2}})) \\ &= {}^e\tau_{\mathrm{PID}} - {}^e\tau_{\mathrm{L1/2}} \end{aligned} \tag{43} $$

and further in index form

$$ \begin{aligned} &\mathrm{sgn}({}^e\tau_{\mathrm{PIDi}} - {}^e\tau_{\mathrm{L1/2i}})\min(|{}^e\tau_{\mathrm{PIDi}} - {}^e\tau_{\mathrm{L1/2i}}|, {}^e\hat{\tau}_{\mathrm{si}}({}^e\tau_{\mathrm{L1/2i}})) \\ &= {}^e\tau_{\mathrm{PIDi}} - {}^e\tau_{\mathrm{L1/2i}} \end{aligned} \tag{44} $$

which describes the equilibrium of the actuator dynamics in closed loop during the experiment. Inspecting the solutions of Equation (44) we get the following possible equilibrium states

$$ {}^e\tau_{\mathrm{PIDi}} - {}^e\tau_{\mathrm{L1/2i}} = s \leq {}^e\hat{\tau}_{\mathrm{si}}({}^e\tau_{\mathrm{L1/2i}}) \tag{45} $$

what matches with the expected behavior of the system at equilibrium. If at rest (${}^e\dot{\theta} = 0$), the external torque is small enough to be compensated by stiction forces, hence $|{}^e\tau_{\mathrm{PIDi}} - {}^e\tau_{\mathrm{L1/2i}}| \leq {}^e\hat{\tau}_{\mathrm{si}}({}^e\tau_{\mathrm{L1/2i}})$, then the external torque is equal to some stiction force $s \leq {}^e\hat{\tau}_{\mathrm{si}}({}^e\tau_{\mathrm{L1/2i}})$ smaller than the maximum stiction force. To the contrary, if the external torque exceeds the maximum stiction force (for example when the I-Part of the controller is increasing due to $q \neq q_0$) the Equation (44) has no solution, because the stiction can not compensate the external torque. The motor starts moving then, leaving the equilibrium state ${}^e\dot{\theta} = 0$. With Equation (45), we can use our collected measurements of ${}^e\tau_{\mathrm{L1/2}}$ and ${}^e\tau_{\mathrm{PID}}$ pairs in the following to provide an estimate for the functions ${}^e\hat{\tau}_{\mathrm{si}}({}^e\sigma_i)$.

From Equation (45), we know in our experiment that for the same fix load torque $\tau_{\mathrm{L1/2}}$, the resulting set of equilibrium torque differences $({}^e\tau_{\mathrm{L1/2}} - {}^e\tau_{\mathrm{PID}})$ will always be smaller then the corresponding maximum stiction ${}^e\hat{\tau}_{\mathrm{si}}({}^e\sigma_i)$. Starting the robotic system from different enough starting conditions, the PID-controller will produce varying final equilibrium torque differences $({}^e\tau_{\mathrm{L1/2}} - {}^e\tau_{\mathrm{PID}})$. By collecting a big enough set of those measurements, we can explore the span of possible equilibrium torque differences to a certain joint stress ${}^e\sigma_i$ and approximate the maximum stiction force ${}^{ee}\sigma_i = \hat{\tau}_{\mathrm{si}}({}^e\tau_{\mathrm{L1/2i}})$ by the maximum equilibrium torque difference in the measurement set. Therefore, having k measurements of ${}^e\tau_{\mathrm{PID}}$ for a weight m_l, we can approximate ${}^e\widehat{\tau}_{\mathrm{si}}({}^e\sigma_i = {}^e\tau_{\mathrm{L1/2i}}(m_l))$ through

$$ {}^e\widehat{\tau}_{\mathrm{si}}({}^e\tau_{\mathrm{L1/2i}}(m_l)) \approx \max_{\mathrm{k}} |{}^e\tau_{\mathrm{PIDik}} - {}^e\tau_{\mathrm{L1/2i}}(m_l)| \quad (46) $$

Computing this approximation for various weights m_l, we yield a set of l estimated points $({}^e\widehat{\underline{\tau}}_{\mathrm{sil}}, {}^e\sigma_{il})$ which we can again use to construct an approximate function for ${}^e\widehat{\tau}_{\mathrm{si}}({}^e\sigma_i)$ by curve-fitting. As a curve-fitting method we again use nonlinear polynomial least squares and the resulting approximation together with the used measurement data is depicted in Figure 5. Our experiments show that the stiction forces in each joint increases with the acting joint stress. From this approximation of ${}^e\widehat{\tau}_{\mathrm{s}}({}^e\sigma)$, we can compute ${}^j\widehat{\tau}_{\mathrm{s}}({}^e\sigma)$ through transformation and use the function for the control design discussed in Section (3) to compensate for the stiction and friction torques in the robot.

4.4 Identification of Gravity Vector $g(q)$

For estimating $g(q)$, as seen in Equation 11, we need to identify the parameters β_i. We will estimate them through controlling the robot to various positions with a PID-controller and then compensating the actuator's torque through external load torque induced by weights and lever mechanisms attached to the robot. So, we are compensating the gravity torque with external and known load torques and thereby we can estimate the gravity vector at the specified positions. Using the knowledge of the special structure of $g(q)$, we can use those measurements to further estimate the constants β_i. In the following we will explain this procedure in more detail.

First, we specify postures ${}^k\hat{q}$, for which the gravity vector takes a form which will make it convenient to estimate the constants β_i and to attach the compensation weights to the robot. The different postures can be seen in Figure(7) and Figure(8), with the corresponding weigth load attachment. Writen as vector in joint space, those postures ${}^k\hat{q}$ are

$$ {}^0\hat{q} = [-\pi/2, 0, 0, 0]^{\mathrm{T}} \quad (47) $$
$$ {}^1\hat{q} = [0, -\pi/2, 0, 0]^{\mathrm{T}} \quad (48) $$
$$ {}^2\hat{q} = [0, 0, 0, 0]^{\mathrm{T}} \quad (49) $$
$$ {}^3\hat{q} = [0, -\pi/2, \pi/2, 0]^{\mathrm{T}} \quad (50) $$
$$ {}^4\hat{q} = [0, -\pi/2, 0, \pi/2]^{\mathrm{T}} \quad (51) $$

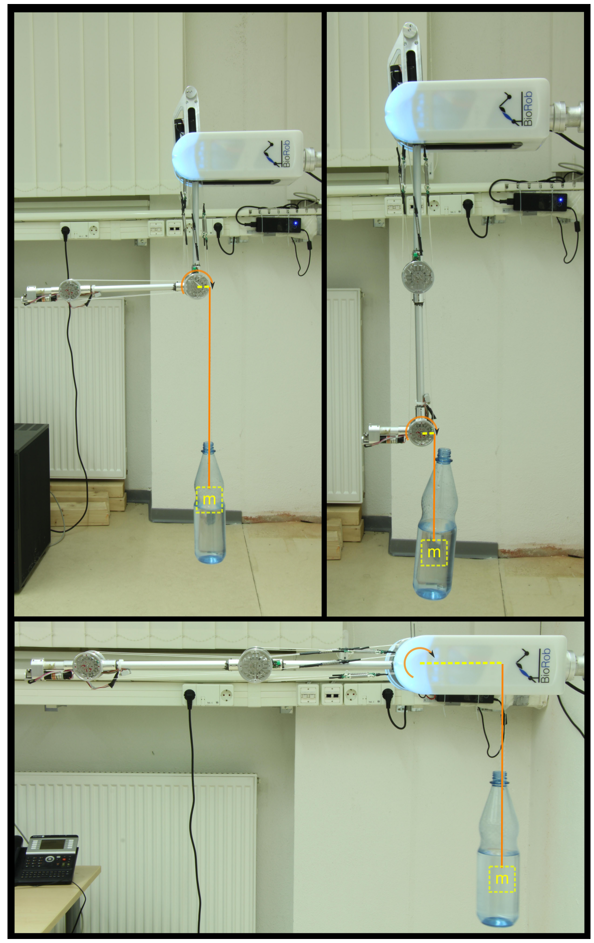

Figure 7: Postures ${}^2\hat{q}$ (lower), ${}^3\hat{q}$ (upper left) and ${}^4\hat{q}$ (upper right) at which we apply external torques at q_2, q_3 and q_4 respectively, through weights and corresponding lever mechanisms. The water bottle has been used in order to be able to change the weight continuously.

which give the associated gravity vectors

$$ g({}^0\hat{q}) = [\beta_1, 0, 0, 0]^{\mathrm{T}} \quad (52) $$
$$ g({}^1\hat{q}) = [\beta_2, 0, 0, 0]^{\mathrm{T}} \quad (53) $$
$$ g({}^2\hat{q}) = [\beta_2, \beta_3 + \beta_4 + \beta_5, \beta_4 + \beta_5, \beta_5]^{\mathrm{T}} \quad (54) $$
$$ g({}^3\hat{q}) = [\beta_2, \beta_4 + \beta_5, \beta_4 + \beta_5, \beta_5]^{\mathrm{T}} \quad (55) $$
$$ g({}^4\hat{q}) = [\beta_2, \beta_5, \beta_5, \beta_5]^{\mathrm{T}} \quad (56) $$

To estimate the constants β_i now, we try to find the correct weight, which through our lever-mechanism shown in Figure 7 and Figure 8 is compensating the actuator torque in a specified joint. If we are managing to compensate the actuator torque, which is meant to keep the robot in the desired position through our chosen weight, the load torque which we are applying to this joint is compensating the

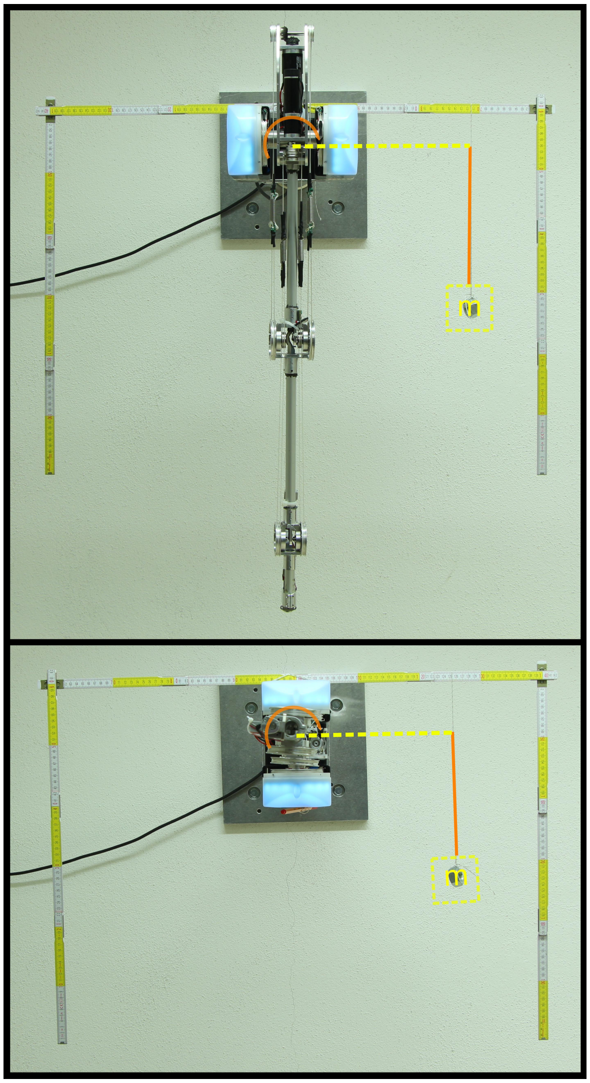

Figure 8: Postures ${}^0\hat{q}$ (lower) and ${}^1\hat{q}$ (upper) at which we apply external torques in q_1 through the weight and corresponding lever mechanisms. The continuous adjustment of the external torque can be achieved by adjusting the acting lever length.

gravity torque acting on this joint. With this method we can measure, the gravity torque acting on a particular joint at a particular posture. To observe when we reached the gravity compensating weight, we check if the elongation of the translational springs of the corresponding actuator is at neutral position. This is the most accurate way to determine if we are compensating the actuator torque, since we are circumventing the influence of stiction, we would have if we simply look when the controller torque turns zero.

In Figure 7 and Figure 8, we show which postures and which joints we chose to estimate the gravity induced joint torque. From the experiments in Figure 8, we could estimate the gravity acting in q_1 in the posture ${}^0\hat{q}$ and ${}^1\hat{q}$ and thereby collected the measurements $z_1 = \beta_1$ and $z_2 = \beta_2$ respectively. From the other experiment setups shown in Figure 7, we could estimate the gravity torques acting in joint q_2, q_3 and q_4 for the postures ${}^2\hat{q}$, ${}^3\hat{q}$ and ${}^4\hat{q}$ respectively. From those experiments we got the measurements $z_3 = \beta_3 + \beta_4 + \beta_5$, $z_4 = \beta_4 + \beta_5$ and $z_5 = \beta_5$. Writing the relation between our measurements z_i and β_i, we get the following linear set of equations:

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \end{bmatrix}. \tag{57}$$

and solving it provides the estimations for β_i as presented in Table 2.

β_1	β_2	β_3	β_4	β_5
0.1992	0.09152	0.80249	0.7541	0.1393

Table 2: Measurement Results for β_i

5 Experimental Results

In this section we will demonstrate the performance of our controller from Section 3 on the BioRob robot. To test the performance and efficacy of our controller, we will try to track the same trajectory with different controllers to compare how our controller manages trajectory tracking regarding the performances of other control laws. The benchmark trajectory is generated using minimum jerk algorithm for a set of points and times of arrival and is plotted in the Figures 9- 12 in blue. The desired trajectory has been designed to cover a large domain of the state space with fairly fast motions. As shown in Figures 9 - 12, the trajectories span more than π radians in each joint and cover different exposure to gravitational torque during execution. The presented trajectory tracking results have been shown to be consistent over many iterations with the same controller configuration. The performance evaluation of each controller will be based on E_{RMS},

$$E_{\mathrm{RMS}} = \sum_{i=1}^{4} \left(e_{qi} + e_{\theta i} \right) \tag{58}$$

where e_{qi} and $e_{\theta i}$ are the RMS trajectory errors of the corresponding state:

$$e_{qi} = \sqrt{\sum_{k=1}^{N} (q_{ik} - q_{idk})^2} \tag{59}$$

$$e_{\theta i} = \sqrt{\sum_{k=1}^{N} (\theta_{ik} - \theta_{idk})^2}. \tag{60}$$

In the following, we will show the experimental results of tracking the trajectory with different controllers and will compare the different controller performances.

5.1 PD-Controller in q

In Figure 9, we show the trajectory tracking using a PD-controller which only uses the joint sensor information q. Thus, the control law is formulated as

$$^{j}\tau_{\mathrm{m}} = \mathbf{K}_{\mathrm{P_q}} (q_{\mathrm{d}} - q) + \mathbf{K}_{\mathrm{D_{\dot{q}}}} (\dot{q}_{\mathrm{d}} - \dot{q}) \tag{61}$$

where q_d describes the desired trajectory we are trying to follow. To achieve the results in Figure 9, the controller was parametrized with $\mathbf{K}_{\mathrm{P_q}} = \mathrm{diag}(20, 20, 15, 10)$, $\mathbf{K}_{\mathrm{D_{\dot{q}}}} = \mathrm{diag}(0.2, 0.2, 0.1, 0.1)$. Evaluating the performance of the controller we get the RMS error values in Table (3).

e_{q1}	0.1016	$e_{\theta 1}$	0.1102
e_{q2}	0.2613	$e_{\theta 2}$	0.2360
e_{q3}	0.2217	$e_{\theta 3}$	0.1993
e_{q4}	0.2565	$e_{\theta 4}$	0.2521
Σ	0.8412	Σ	0.7976
$E_{\mathrm{RMS}} = 1.6388$			

Table 3: RMS errors calculated from trajectory tracking experiment with controller (61)

5.2 PD-Controller in q with gravity compensation

Next, Figure 10 show the trajectory tracking using a PD-controller using only the joint sensor information q and compensating the gravity influence. The controller is the same as in (61), but includes the term $g(q_{\mathrm{d}})$, which compensates the gravity along the desired trajectory of the robot q_d.

$$^{j}\tau_{\mathrm{m}} = \mathbf{K}_{\mathrm{P_q}} (q_{\mathrm{d}} - q) + \mathbf{K}_{\mathrm{D_{\dot{q}}}} (\dot{q}_{\mathrm{d}} - \dot{q}) + g(q_{\mathrm{d}}) \tag{62}$$

To achieve the results in Figure 10, the controller was parametrized with $\mathbf{K}_{\mathrm{P_q}} = \mathrm{diag}(20, 20, 15, 10)$, $\mathbf{K}_{\mathrm{D_{\dot{q}}}} = \mathrm{diag}(0.2, 0.2, 0.1, 0.1)$. Evaluating the performance of the controller we get the RMS error values in Table (4).

e_{q1}	0.0981	$e_{\theta 1}$	0.1079
e_{q2}	0.2377	$e_{\theta 2}$	0.2245
e_{q3}	0.2040	$e_{\theta 3}$	0.1974
e_{q4}	0.2413	$e_{\theta 4}$	0.2358
Σ	0.7810	Σ	0.7655
$E_{\mathrm{RMS}} = 1.5466$			

Table 4: RMS errors calculated from trajectory tracking experiment with controller (62)

5.3 PD-Controller in q and $^{j}\theta$ with gravity compensation

The trajectory tracking performance of a PD-controller using both q and θ with gravity compensation $g(q_{\mathrm{d}})$ is shown in the following Figure 11. The control takes the form of our controller from Section 3:

$$\begin{aligned} ^{j}\tau_{\mathrm{m}} = {} & \mathbf{K}_{\mathrm{P_q}} (q_{\mathrm{d}} - q) + \mathbf{K}_{\mathrm{D_{\dot{q}}}} (\dot{q}_{\mathrm{d}} - \dot{q}) + \\ & \mathbf{K}_{\mathrm{P}_\theta} (^{j}\theta_{\mathrm{d}} - {}^{j}\theta) + \mathbf{K}_{\mathrm{D}_{\dot{\theta}}} (^{j}\dot{\theta}_{\mathrm{d}} - {}^{j}\dot{\theta}) + g(q_{\mathrm{d}}) \end{aligned} \tag{63}$$

To achieve the results in Figure 11, the controller was parametrized with $\mathbf{K}_{\mathrm{P_q}} = \mathrm{diag}(1, 2, 6.5, 9)$, $\mathbf{K}_{\mathrm{D_{\dot{q}}}} = \mathrm{diag}(0, 0, 0, 0)$, $\mathbf{K}_{\mathrm{P}_\theta} = \mathrm{diag}(45, 80, 80, 40)$, $\mathbf{K}_{\mathrm{D}_{\dot{\theta}}} = \mathrm{diag}(0, 0, 0, 0)$. Evaluating the performance of the controller we get the RMS error values in Table (5).

e_{q1}	0.0529	$e_{\theta 1}$	0.0440
e_{q2}	0.0813	$e_{\theta 2}$	0.0635
e_{q3}	0.0630	$e_{\theta 3}$	0.0661
e_{q4}	0.2026	$e_{\theta 4}$	0.1922
Σ	0.3998	Σ	0.3658
$E_{\mathrm{RMS}} = 0.7657$			

Table 5: RMS errors calculated from trajectory tracking experiment with controller (63)

5.4 PD-Controller in q and $^{j}\theta$ with gravity and friction compensation

In this subsection we present the results of trajectory tracking using our full control algorithm of Section 3 in Figure 12. In total, the controller consists of a PD-controller q and θ-wise, a gravity compensation $g(q_{\mathrm{d}})$ and a stiction compensation $^{j}\hat{\tau}_s$ which we described in Section 3:

$$\begin{aligned} ^{j}\tau_{\mathrm{m}} = {} & \mathbf{K}_{\mathrm{P_q}} (q_{\mathrm{d}} - q) + \mathbf{K}_{\mathrm{D_{\dot{q}}}} (\dot{q}_{\mathrm{d}} - \dot{q}) + \\ & \mathbf{K}_{\mathrm{P}_\theta} (^{j}\theta_{\mathrm{d}} - {}^{j}\theta) + \mathbf{K}_{\mathrm{D}_{\dot{\theta}}} (^{j}\dot{\theta}_{\mathrm{d}} - {}^{j}\dot{\theta}) + g(q_{\mathrm{d}}) + {}^{j}\hat{\tau}_s \end{aligned} \tag{64}$$

To achieve the results in Figure 12, the controller was parametrized with $\mathbf{K}_{\mathrm{P_q}} = \mathrm{diag}(1, 2, 6.5, 9)$, $\mathbf{K}_{\mathrm{D_{\dot{q}}}} = \mathrm{diag}(0, 0, 0, 0)$, $\mathbf{K}_{\mathrm{P}_\theta} = \mathrm{diag}(45, 80, 80, 40)$, $\mathbf{K}_{\mathrm{D}_{\dot{\theta}}} = \mathrm{diag}(0, 0, 0, 0)$. The tuning parameters of the stiction compensation were chosen as $\epsilon_e = [0.001, 0.001, 0.001, 0.001]$ and $\delta = [0.1, 0.1, 0.1, 0.5]$. Evaluating the performance of the controller we get the RMS error values in Table (6).

e_{q1}	0.0500	$e_{\theta 1}$	0.0362
e_{q2}	0.0692	$e_{\theta 2}$	0.0497
e_{q3}	0.0648	$e_{\theta 3}$	0.0638
e_{q4}	0.2008	$e_{\theta 4}$	0.1933
Σ	0.3848	Σ	0.3429
$E_{\mathrm{RMS}} = 0.7277$			

Table 6: RMS errors calculated from trajectory tracking experiment with controller (64)

5.5 PD-Controller in ${}^j\theta$ with gravity and friction compensation

Since the PD-gains $\mathbf{K}_{\mathrm{P_q}}$ and $\mathbf{K}_{\mathrm{D_{\dot{q}}}}$ of our final controller are relatively small, we are trying to determine their influence on the overall control performance in this subsection. Therefore in Figure 13, we present the results of trajectory tracking using the controller from Section 5.4 but with $\mathbf{K}_{\mathrm{P_q}}$ and $\mathbf{K}_{\mathrm{D_{\dot{q}}}}$ set to zero, which yields the following control law:

$$ {}^j\tau_{\mathrm{m}} = \mathbf{K}_{\mathrm{P}_\theta}\,({}^j\theta_{\mathrm{d}} - {}^j\theta) + \mathbf{K}_{\mathrm{D}_{\dot{\theta}}}\,({}^j\dot{\theta}_{\mathrm{d}} - {}^j\dot{\theta}) + g(q_{\mathrm{d}}) + {}^j\hat{\tau}_s \tag{65} $$

To achieve the results in Figure 13, the remaining parameters of the controller were chosen as in Subsection 5.4. Evaluating the performance of the controller we get the RMS error values in Table (7).

e_{q1}	0.0493	$e_{\theta 1}$	0.0391
e_{q2}	0.0767	$e_{\theta 2}$	0.0519
e_{q3}	0.0663	$e_{\theta 3}$	0.0636
e_{q4}	0.2154	$e_{\theta 4}$	0.2030
Σ	0.4078	Σ	0.3575
$E_{\mathrm{RMS}} = 0.7653$			

Table 7: RMS errors calculated from trajectory tracking experiment with controller (65)

5.6 Comparison

Comparing the trajectory tracking of all controllers previously presented, we can see how our controller improves the performance with respect to the simpler control algorithms presented in Sections 5.1 - 5.3. In Table (8), we can see the summary of controller tracking errors during the experiments and Figure 14 shows a direct comparison of all controller performances when trying to track a trajectory in q_3.

	Σ_q	Σ_θ	E_{RMS}
PD in q	0.8412	0.7976	1.6388
PD in q + g.c.	0.7810	0.7655	1.5466
PD in q, ${}^j\theta$ + g.c.	0.3998	0.3658	0.7657
PD in ${}^j\theta$ + g.c.+ f.c.	0.4078	0.3575	0.7653
PD in q, ${}^j\theta$ + g.c. + f.c.	0.3848	0.3429	0.7277

Table 8: Summary of trajectory tracking errors for different control laws. (g.c. = gravity compensation, f.c. = friction compensation)

In Figure 9, we see that the controller tracks the trajectory with a big error, because it is not accounting for the gravity torque acting on the joints. Furthermore the closed loop system shows severe undamped oscillations. Although those oscillations still remain, using the PD-controller with gravity compensation in Figure 10 improves the controller performance, when comparing the E_{RMS} values, as seen in Table (3) and (4) of both controllers. In Figure 11 we notice that the oscillations and the fairly big tracking error we observed in the previous controllers are being tremendously reduced by adding a PD-controller in ${}^j\theta$ in Section 5.3. We can also observe a strong decline in the RMS error values (Table 5) of this experiment and see that the PD-controller in q and ${}^j\theta$ with gravity compensation tracks the trajectories with a better accuracy. Furthermore we also notice, that due to stiction forces, the motor angles are slacking and are having diffculties tracking the desired trajectory. This effect is visible the most, when looking at the graphs of ${}^j\theta_2$ and ${}^j\theta_3$ in Figure 11. The big stiction forces acting on the actuator side of the BioRob robot prevent the robot from moving, until the tracking error is big enough, such that the resulting controller torque exceeds the maximum stiction force. This can be improved by including the stiction/friction compensation to our controller in Section (5.4). Comparing Figure 11 with 12 and comparing the RMS-error (Table 5) and (Table 6) of both experiments we see that our stiction/friction compensation reduces the motor angle slacking of our controller and thereby further reduces the tracking error to the trajectories. Comparing our final tuning parameters of our full control law in Figure 12, we see that the PD-gains in q are significantly smaller than the ones of the PD-gains in ${}^j\theta$. By setting the PD-gains in q to zero and running another experiment, we can see as in Figure 13, that aside from attenuating oscillations in q_2, they have little effect on the overall closed loop performance of the system since the RMS error values (Table 7) barely change from (Table 6). We achieved the increase in controller performance through our control design mostly by calculating the desired motor position ${}^j\theta_d$ correctly and implementing a high gain PD-part in ${}^j\theta$.

Overall, the controller performs with a strongly improved tracking behavior, but still shows some significant oscillations in q_2. The remaining tracking error and oscillations in the closed loop system are coming mostly from the lack of knowledge of the dynamical parameters like $\mathbf{M}(q)$, $\mathbf{C}(q, \dot{q})$, etc. and therefore the inability to calculate the desired trajectory of the actuator ${}^{j}\theta_d$ considering not only static but dynamic effects. Furthermore, notice also in all Figure 13 to 9, the fairly big tracking errors in q_1 between 10 and 15 and in q_4 between 13 and 17. Those originate most likely from problems of the electro-mechanical system of the robot as discussed later in Section (A).

6 Conclusion

In this work, we designed a trajectory tracking controller for the flexible joint robot arm BioRob. The controller design is based on [1] and consists of a PD-controller in both joint and actuator state, a gravitational compensation along the desired trajectory and an actuator stiction compensation. For the design of the controller we used knowledge of the gravity vector, the force-elongation characteristics of the springs in the transmission and the stiction torques acting on the actuator side of the robot. All those parameters has been estimated by designing and performing BioRob-tailored system identification methods. Comparing the resulting controller with previous control laws, we noticed that our controller could drastically improve the performance of the closed loop system.

Future work should be invested in improving the controller performance even more, since despite the performance improvement, significant oscillations and tracking errors are still present. In order to improve the performance of the system, more work has to be invested in system identification of the dynamic parameters of the system. Furthermore, adressing the mechanical and software problems we have encountered during our work would reduce the system inherent limits of the BioRob and allow for better control design. Particular focus should be put on trying to learn or estimate the dynamic model of the friction and stiction effects occurring in the BioRob system. Further improvement of the controller performance can also be achieved by computing the desired actuator angle based on the inverse dynamics computed joint torque.

References

[1] A. Albu-Schäffer and G. Hirzinger. State feedback controller for flexible joint robots: a globally stable approach implemented on DLR's light-weight robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2:1087-1093, 2000.

[2] T. Lens, and O. von Stryk. Design and Dynamics Model of a Lightweight Series Elastic Tendon-Driven Robot Arm. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4512-4518, 2013.

[3] A. De Luca and W. Book. Robots with Flexible Elements. In *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer Berlin Heidelberg, pp. 287-319, 2008.

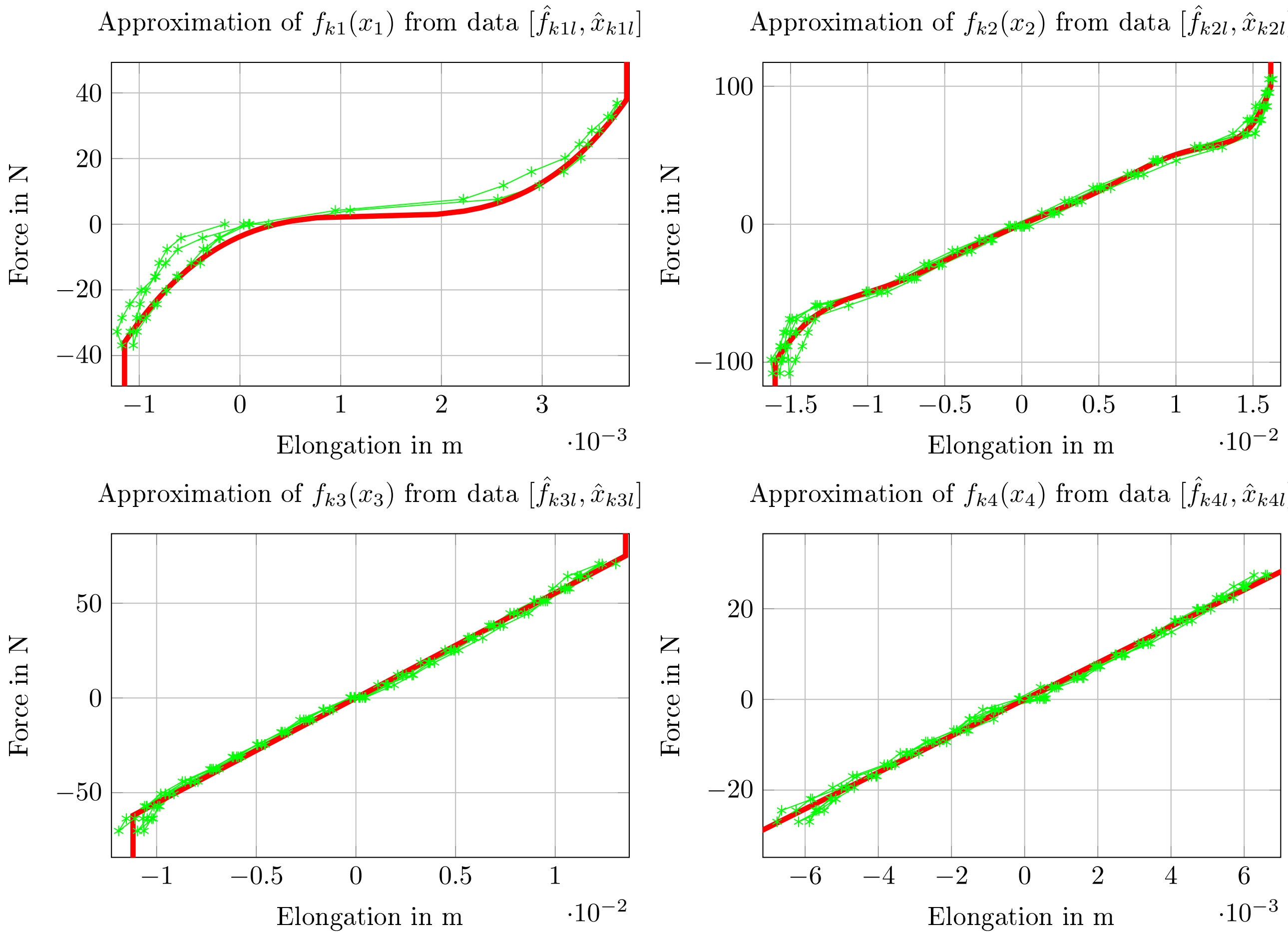

Figure 4: Spring force $\hat{f}_{\mathrm{k}il}$ and the corresponding x_{il} from measurements (green) and the derivative approximation $f_{\mathrm{k}i}(x_i)$ generated from nonlinear least squares curve-fitting (red).

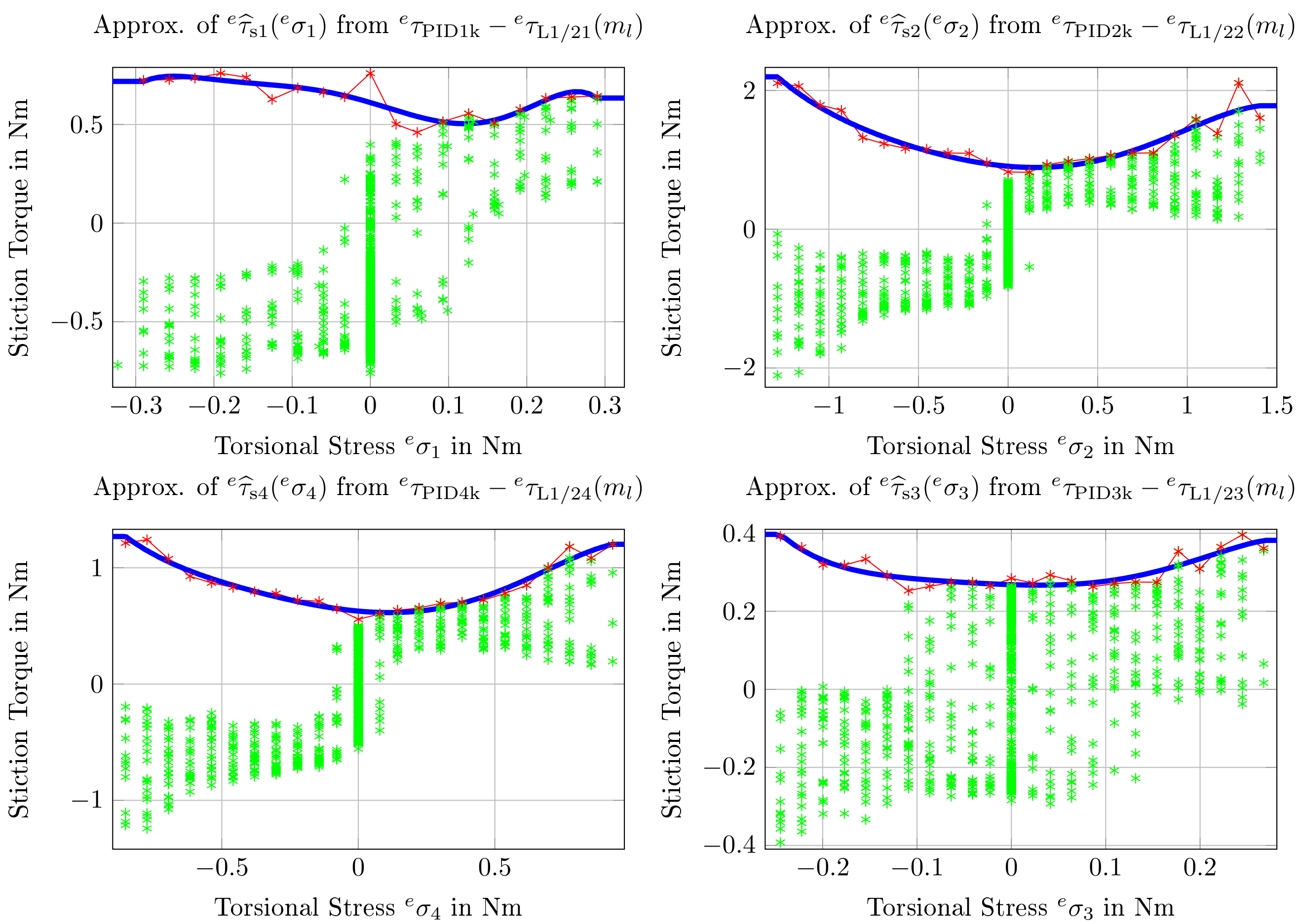

Figure 5: The measured k torque differences $^e\tau_{\mathrm{PIDik}} - {}^e\tau_{\mathrm{L1/2i}}(m_l)$ to the corresponding joint stress $^e\sigma_{\mathrm{i}} = {}^e\tau_{\mathrm{L1/2i}}(m_l)$ (green points), maximum absolute difference $\max_{\mathrm{k}} |{}^e\tau_{\mathrm{PIDik}} - {}^e\tau_{\mathrm{L1/2i}}(m_l)|$ (red) and the derivative approximation for the function $^e\widehat{\tau}_{\mathrm{si}}(^e\tau_{\mathrm{L1/2i}}(m_l))$ generated from nonlinear least-squares (blue).

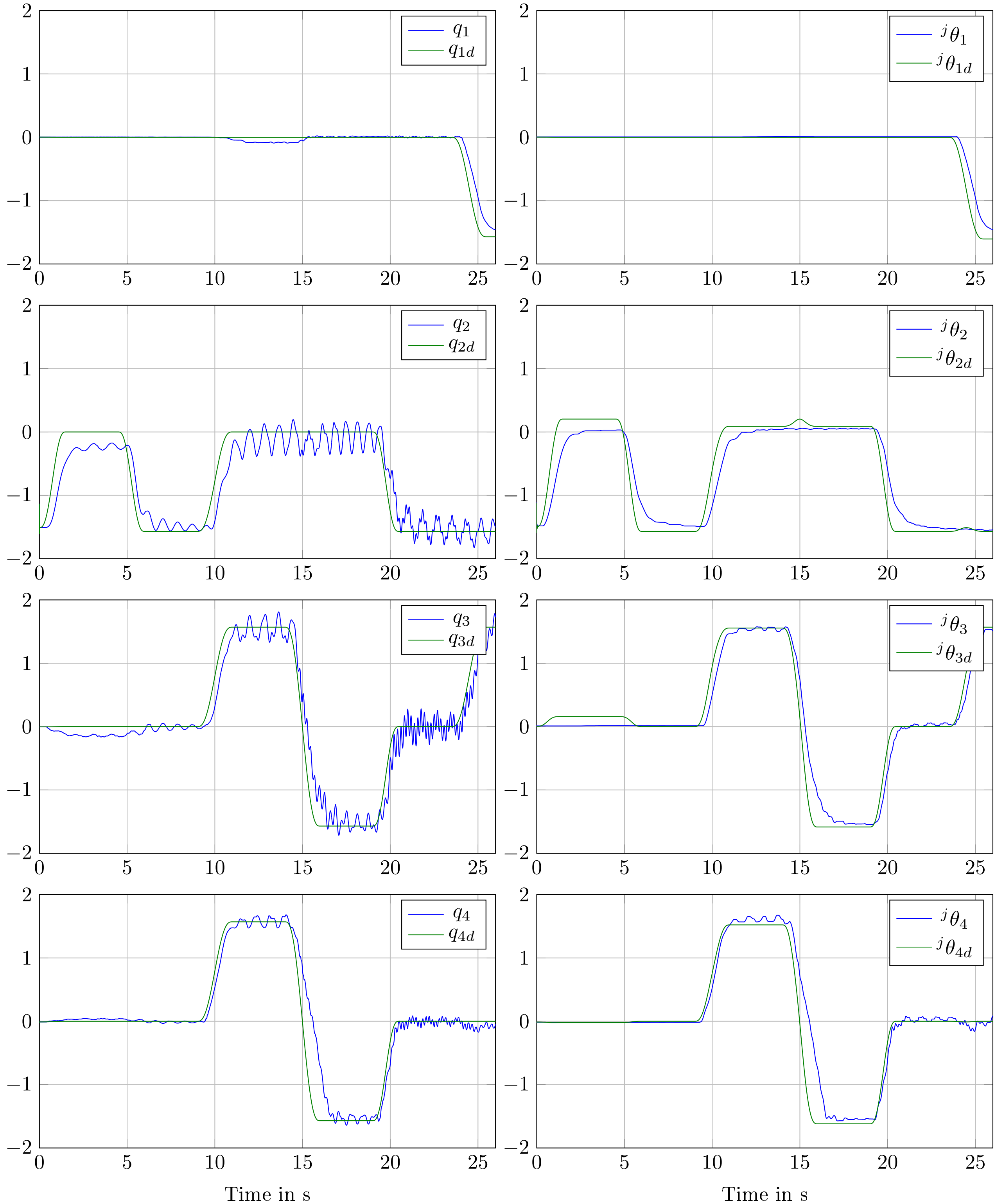

Figure 9: Trajectory tracking with a PD controller in q.

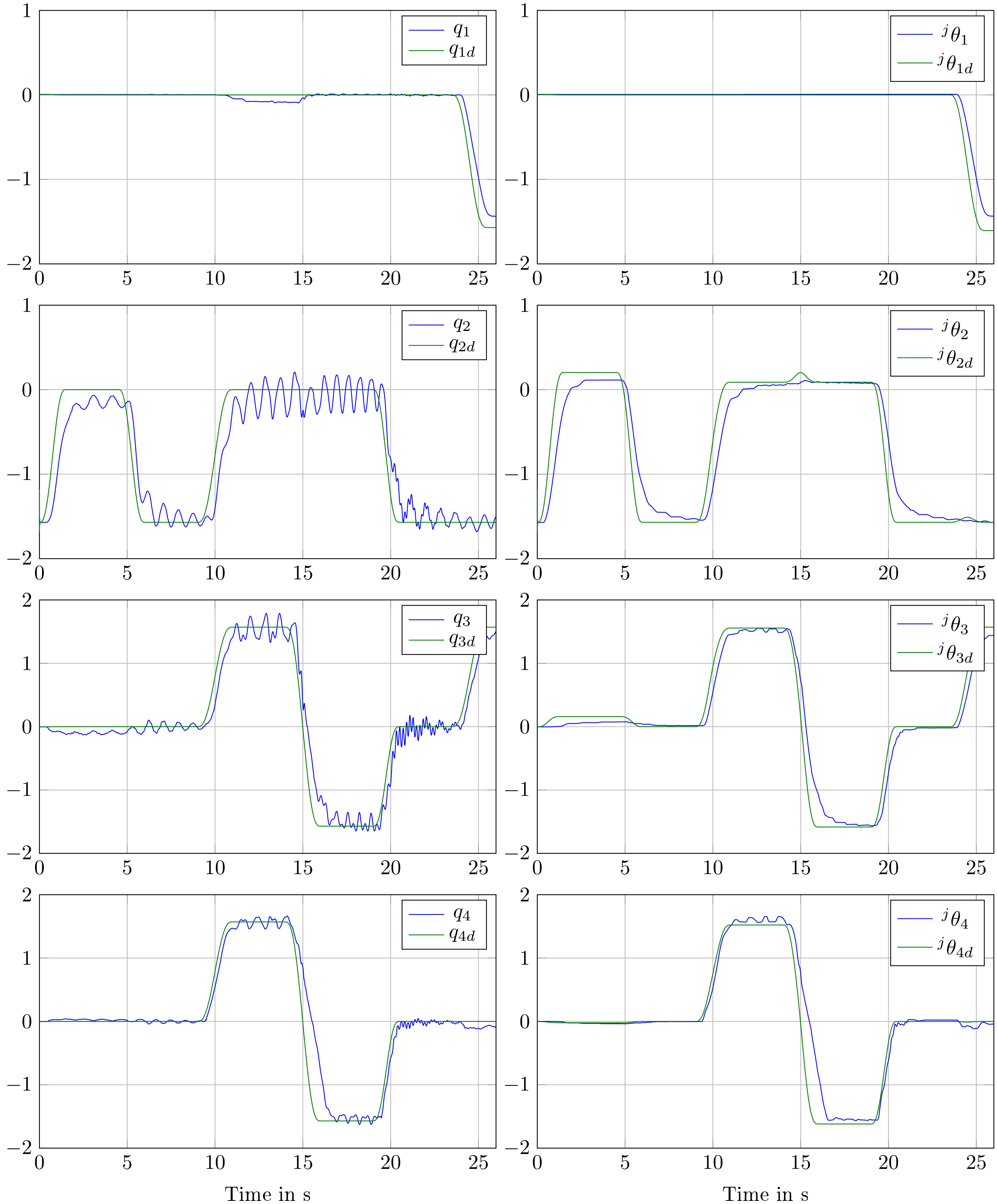

Figure 10: Trajectory tracking with a PD controller in q and gravity compensation $g(q_{\mathrm{d}})$.

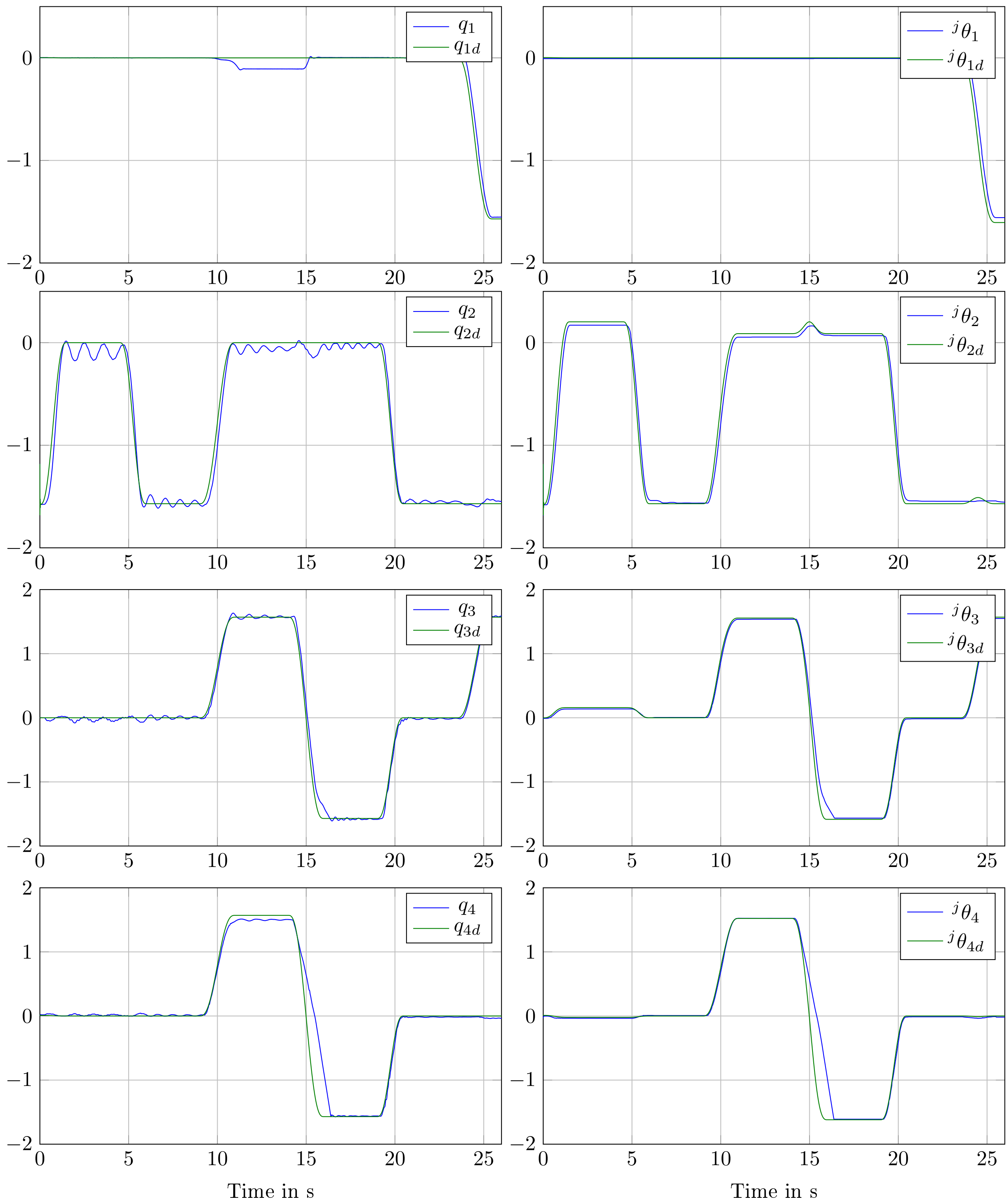

Figure 11: Trajectory tracking with a PD controller in q and θ and gravity compensation $g(q_\mathrm{d})$.

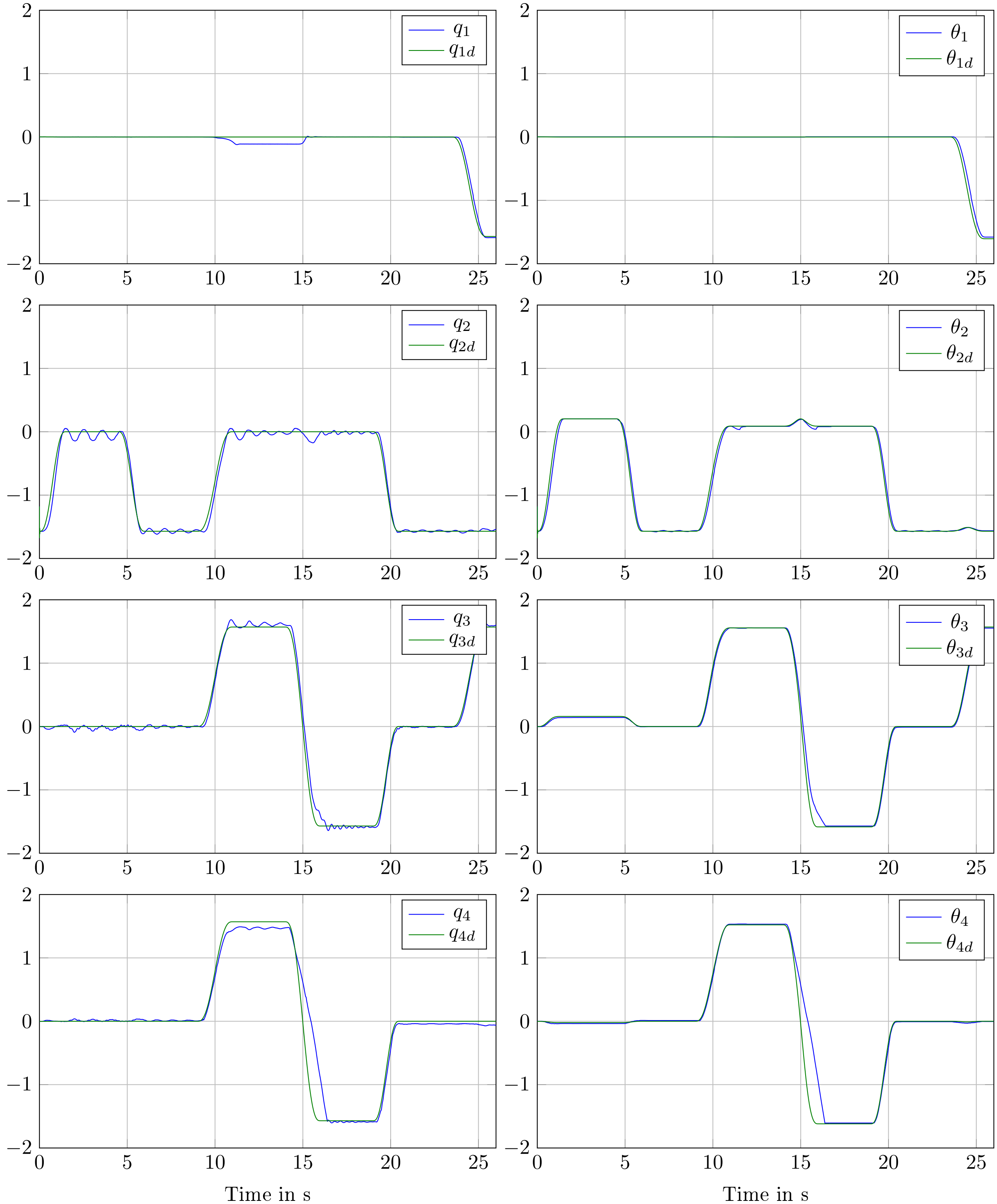

Figure 12: Trajectory tracking with full controller: PD controller in q and θ with gravity compensation $g(q_\mathrm{d})$ and friction compensation ${}^j\hat{\tau}_s$.

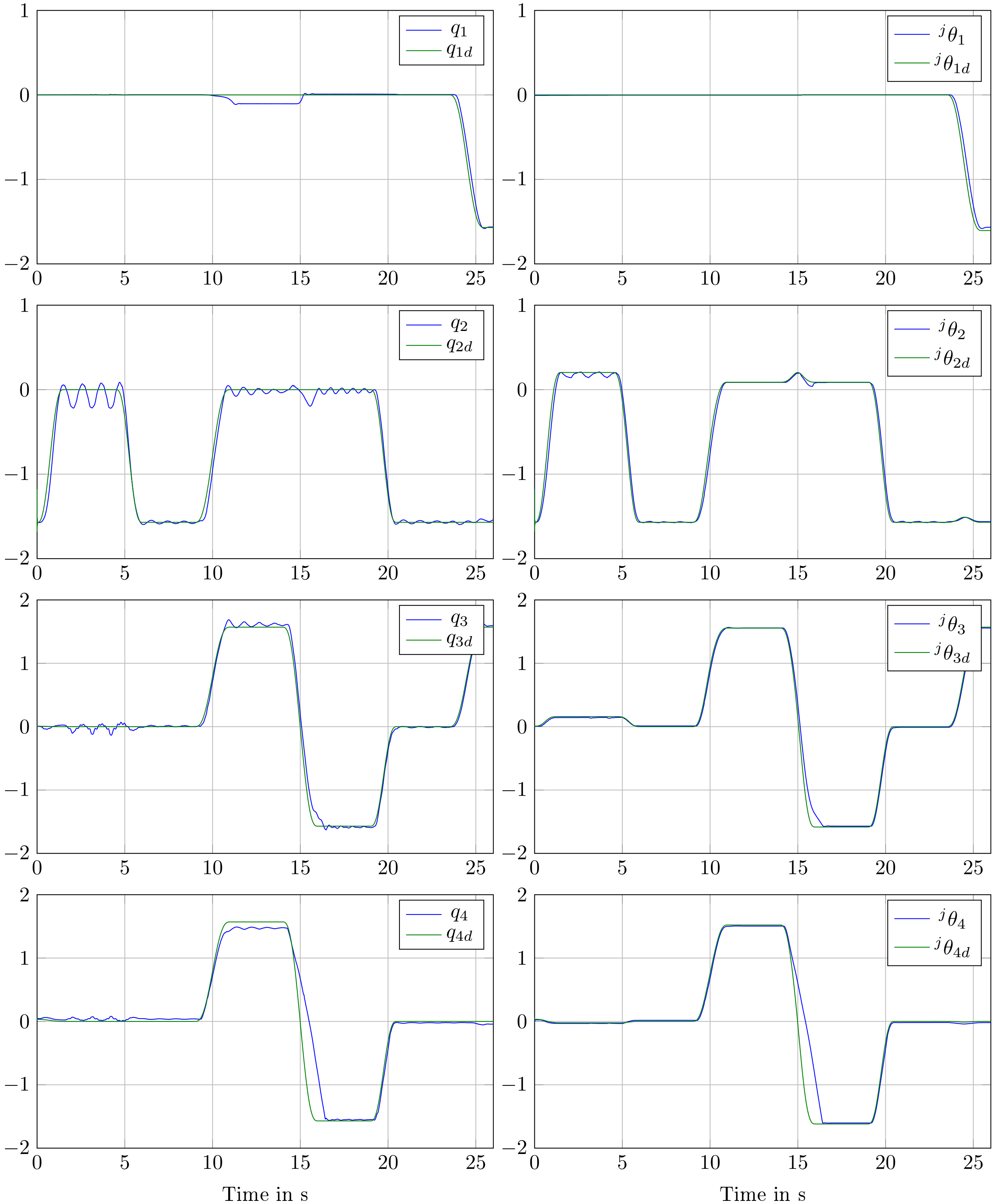

Figure 13: PD controller in θ with gravity compensation $g(q_\mathrm{d})$ and friction compensation $^j\hat{\tau}_s$.

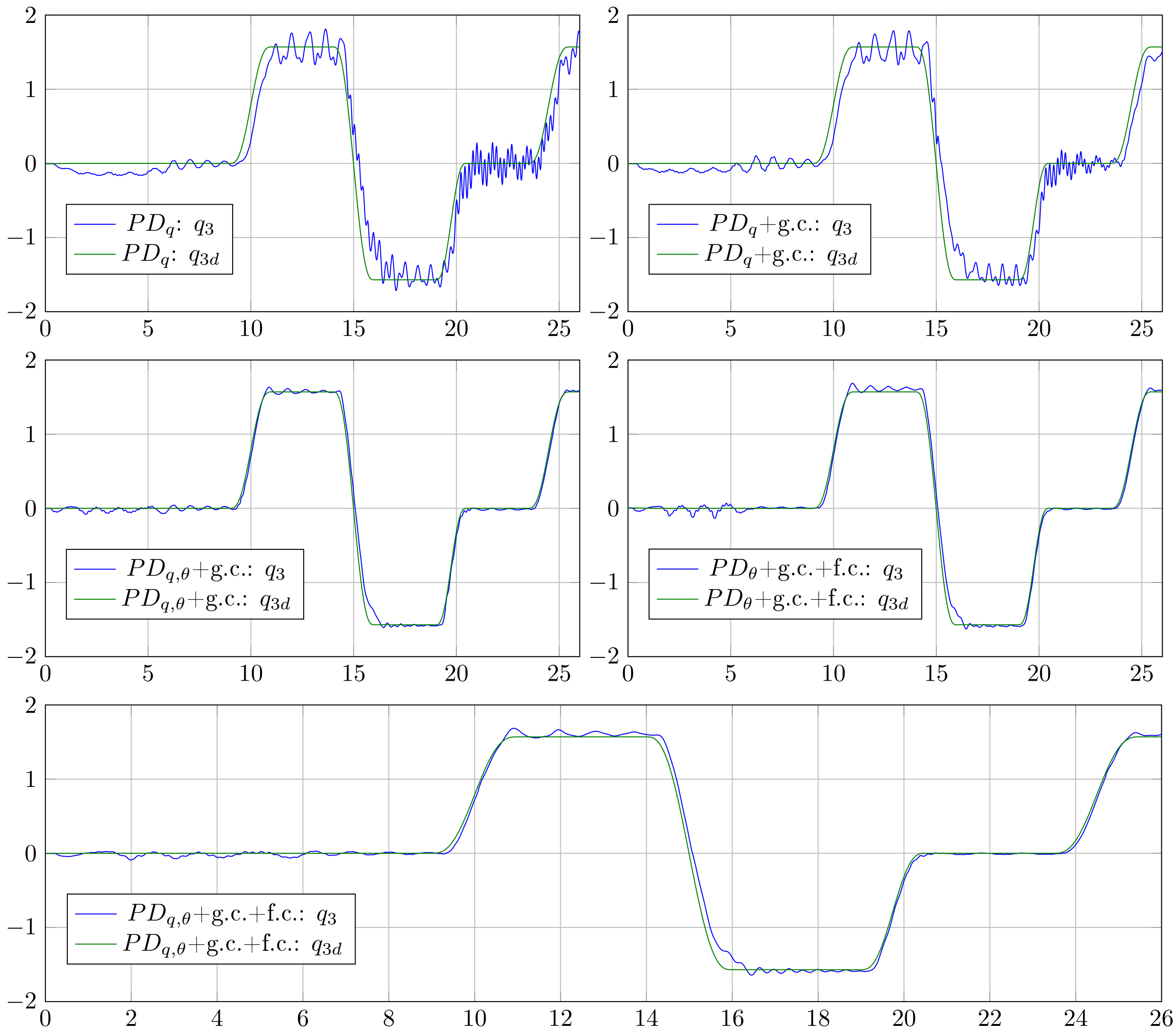

Figure 14: Comparisson of performances of all controllers regarding trajectory tracking in joint q_3.

A Bugs and other Problems

In this appendix, we would like to point out some present and unsolved problems we noticed, which should be adressed in future work with the BioRob.

A.1 Mechanical Problems

Taking a look at the second link of the BioRob, one can see that it is not aligned with the the rest of the robot body. It is torsionally twisted by 5.1° with respect to its longitudinal rotational axis. This should be fixed, since mostly the inverse kinematics currently is not considering this effect. Furthermore, the dynamical parameters like inertia, coriolis and centrifugal matrices change due to this misalignment.

When trying to track fast trajectories, the ropes attached to the springs come into slacking and slip off the pulleys. This even leads to the ropes ripping and damaging the robot. The springs of the system have to be either prestressed or become more stiff in order to prevent slacking of the springs, which sofar frequently occurs.

The mechanical elastic transmission system of the first joint shows some problems, which limit the control performance of the robot. Firstly, the mechanical design of the transmission in the first joint has a significant deadzone of approximately $10° - 20°$. One could solve this problem by adjusting the sliding bearing, which is attached to the spring and rope. Another possibility would be to make the whole joint completely stiff, as the region in which the transmission is elastic, is anyway really limited. From the control theory point of view, the deadzone induces mechanical impacts and uncontrollability when directions are changed during a trajectory. One could theoretically remove the problems originating from the deadzone, with a high-gain controller which produces high accelerations on the motorside, but this solution is not realistic and not practical to aim for. The deadzone is also the cause for the sudden increase in tracking error in q_1 between the 10 and 15 second in Figures 13 to 9. During that time frame, the third link moves and due to the torsional misalignment of the second link, the thereby induced and gradually increasing gravity torque in q_1 compensates the gravity torque of link q_1 and makes the first joint rotate to the other border of the deadzone, which causes the visible tracking error.

The second link of the robot (between joint q_2 and q_3) is not completely rigidly attached to the metal plates it is mounted on. During fast motions in q_2, the inertial forces make the link jump around $5° - 10°$ which requires manual readjustment afterwards.

Another problem has been noted when actuating joint q_4 in the negative direction. Although it is possible to achieve good tracking behavior in q_4 for positive angles, enormous torques are needed to perform motions in the opposite directions. This phenomena is also the reason for the big tracking errors in q_4 and θ_4 between the 13 and 17 second in Figures 13 to 9, since the controller is faced more resistance when trying to move the forth link into the negative direction. Possible explanations for this behavior might be problems with the gears or the motor of the forth joint, but the exact reason behind this has to be further explored.

A.2 Software Problems

A phenomenon has been noticed regarding the joint sensor of q_1. While trying to measure out the exact amount of the deadzone, which is mechanically constant, experiments with the joint sensor show discrepant results. Depending on, in which position and order you try to measure out the deadzone, the sensor values show strongly differing deadzones. Discussion with the manufacturers of the robot have raised suspicion, that the software reading out and transforming the sensor values might have a problem.

Further research on the electrical and software system has shown, that the joint sensors show nonlinear transmission when reading out the joint angle. This phenomenon obviously corrupts the information of the joint angle and also causes sudden jumps in the angle derivatives in SL, since backwards differentiation is used to estimate the joint angle velocity. Reasons for this originates from the Hall-sensors built in the joints, which have to be calibrated in order to gain a linear characteristic.

The simulation algorithm used in SL might not be suitable for stiff systems and its stability region is small. The both options in SL to simulate the mechanical closed loop system are symplectic euler and a forth order Runge-Kutta scheme. In most cases, the simulation gives good results but some important cases have been encountered in which the simulation is unstable. Firstly, simulating the rigid version of the BioRob with a LQR-controller or a PD-controller with high gains results in an unstable simulation for all initial conditions, which from a control theory point of view should not happen. Secondly, experiments have casted doubts if the current simulation schemes would be able to simulate the BioRob system including the elastic transmission system. Extending the computer model of the BioRob by the elastic actuators, consisting of the motor and spring dynamics, would yield a stiff differential equation, which is inherently more difficult to simulate. We build a simulation in Matlab of a two link robot arm with torsional elasticity between motor and joints and encountered that it is not possible to simulate the system stable with symplectic euler or a Runge-Kutta scheme. Only stiff solvers could provide stable and accurate simulation results and therefore it is questionable if the current simulation model in SL could be extended to the full model of the BioRob including elastic actuators with springs.

B Related Work

In this appendix, we will provide an overview of other related work which has been done during the project of improving the trajectory tracking controller of BioRob. The following section consists of early attempts, supporting tasks and improvement/fix of the software system of the BioRob.

B.1 Early Attempt: Linear System Identification and LQR-Controller for the BioRob

In the early works of our project, we researched about adaptive control techniques for robots and its application to BioRob. Furthermore, we aimed for identifying linear models of the BioRob and designing a LQR-controller with gravity compensation to track trajectories.

We started off by trying out this control approach on rigid manipulators with the same kinematic structure as BioRob in the SL simulation environment. We performed closed loop system identification methods to estimate a local linear model of the rigid robot arm. Exciting the simulation model with sweep sines and PRBS-signals in closed loop, we could find a linear model of the system through Least-Squares system identification. The open loop system could be calculated by transforming the linearized equations of the closed loop system. Using the open loop equations we designed LQR-controller and DLQR-controller. They could stabilize the robot around the setpoint if the resulting gains were not too high, otherwise the simulation turned unstable due to the non-adaptive simulation schemes.

Pursuing this approach further has been stopped because of problems simulating the closed loop system and doubts considering the applicability of the approach to the physical system. In order to simulate the approach, the SL-model had to be extended, such that the model includes the elastic actuator system which is a crucial part of the real BioRob robot. Research and experiments have casted doubt, that the simulation schemes implemented in SL would be capable of simulating the resulting stiff dynamical system stable. Furthermore, the physical system of the BioRob has strong stiction and friction forces acting in the actuator, which pose a considerable obstacle during system identification. To estimate a linear model around an equilibrium, the dynamical system has to be continuously differentiable at that point. Regarding the huge stiction forces on the actuator side of the BioRob which pose a significant discontinuity in the differential equation, most system identification techniques to estimate a linear model through excitation are prone to fail. In addition, deriving a LQR-controller from a linear model would make a good controller for stabilization at a fixed point, but the same controller would not be provably suitable for trajectories. One would have to estimate numerous linear models around the trajectories to make a time-varying LQR-controller, but this does not seem practical for the BioRob system.

B.2 Testing Controllers on a Flexible Joint Robot Arm in Matlab

Since simulating stiff differential equations did not seem possible in SL, we programmed a simulation environment for a 2-DoF flexible joint robot arm in Matlab. Thereby, we implemented the dynamic equations of the robot and a graphical output. We used our simulation environment to test our concern about symplectic euler and non-adapative explicit schemes not being able to simulate the stiff system stable. Further, we implemented different controllers and compared their performance regarding trajectory tracking. We tested out controllers like LQR, time-varying LQR and our final control approach to validate which control law is most promising.

B.3 Verification of Sensor Value Processing

During our work on the robot, we encountered problems with the angle sensors and decided to verify if the motor angles are being correctly transformed and given out in SL. In order to do that, we decided to artificially make the springs stiff by the mechanism shown in Figures 15 and perform measurements with the quasi rigid robot to eventually re-estimate the transformation matrix between the motor and joint sensors. Therefore we designed and built mechanical constructions to stiffen the springs. We noticed that the motor angles are being transformed into joint space by the BioRob API, which we had no access to. Our method

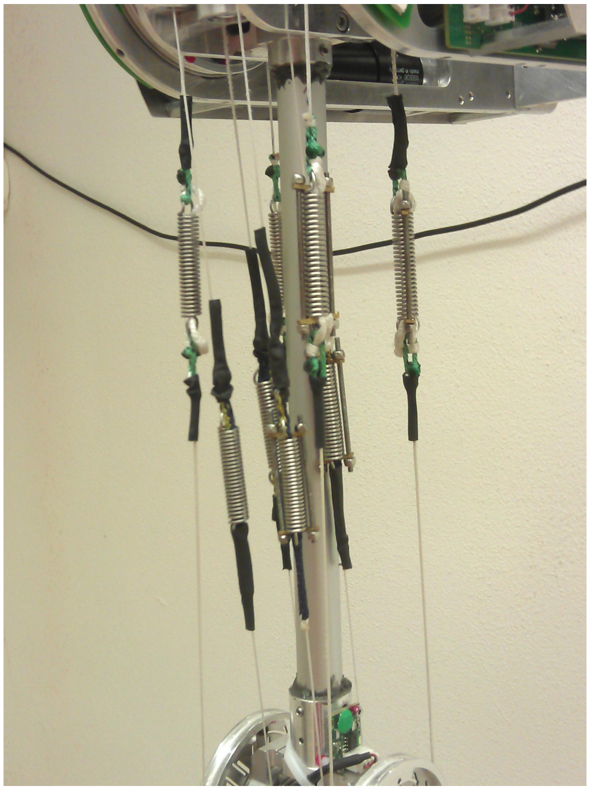

Figure 15: Mechanism to artificially stiffen the springs.

revealed that the transformation of the motor angles was not performed completely since the kinematic coupling through the idler pulley was not considered. We corrected this in the SL file *biorob_servo_unix.cpp*, which establishes the communication between SL and the ROS.

B.4 Fixing BioRob Software Bugs

While working on the BioRob in SL, we encountered two bugs in the software, which we were able to fix. First, BioRob's sensor calibration for the rotary encoders on the motors was performed around a setpoint, which was not suitable for an offset free calibration. The task commanded the robot arm with a PID-contoller to stretch out horizontally. In this position gravity torques had to be compensated with the actuators and therefore the springs were elongated. For calibration, however, the springs must have their prestretched length. We have changed the setpoint into the hanging position of the robot arm. In this position the PID-controller ensures that the elongation of the springs equals the prestretched lenght and, thereby, it compensates even friction/stiction torques in the actuators.

Second, the motor voltages for BioRob were not calculated correctly from the joint side motor torques. To calculate them, the actuator torques in the joint side have to be reflected to motor side and transformed into voltages with the composition of the transformations $\mathbf{J}_{\mathrm{t}}^{\mathrm{T}}$, $\mathbf{J}_{\mathrm{g}}$ and $\mathbf{J}_{\mathrm{u}}$. The first transformation is already explained with Equation (4). It reflects the torques on the actuator side. With the transformation $\mathbf{J}_{\mathrm{g}} = \mathrm{diag}(1/23, 1/23, 1/18, 1/19)$ the resulting torques are reflected through the reduction gears to the motor side. The diagonal entries stand for inverses of the gear ratios n_{g_i}. The final transformation $\mathbf{J}_{\mathrm{u}} = \mathrm{diag}(23.59, 23.59, 23.59, 93.59))$ leads to the motor voltages, whereas the diagonal entries are the ratios of the terminal resistances to the torque constants of the DC-motors.