Nonlinear Control and State Estimation for the Hand Axes of a Pneumatic Robot

Seyed Houman Mirafzal
Independent Researcher, Germany

Keywords: Mechatronics, Robotics, Predictive Control, Flatness-Based Methods, Pneumatic Muscle.

Abstract: This paper presents a nonlinear control for the hand axes of a robot with three pneumatic muscles. A vector-based approach is employed for the modeling. Due to the structure of the system, a flatness-based control method is chosen and used. A control system is designed in which three types of compensators, including feedback, feedforward, and observer (estimator) are used to improve the trajectory tracking of the main joint angles and the muscle force control. The Kalman Filter is used to estimate the disturbance friction torque in the system. Through a combination of theoretical analysis and experimental validation, the proposed methods demonstrate significant improvements in control accuracy and system stability. As a result, the control system tracks the desired trajectories very well, as various trajectories are implemented to test the tracking behavior of the control system.

1 INTRODUCTION

The integration of automatic control systems has become pivotal in modern industries, enhancing both efficiency and precision across a wide range of applications. While open-loop control systems suffice for the operation of simpler devices, such as household appliances, the demand for increasingly complex feedback control systems has steadily grown. Feedback control systems, which continuously monitor and adjust outputs to align with desired set points, are essential for high-precision tasks, ranging from industrial automation to delicate surgical procedures.

Although control using electric actuators is more common due to its processing flexibility, pneumatic control, which operates based on the pressure and force of compressed air, remains crucial in modern industry (Franklin et al., 2019). Pneumatic actuators are commonly used in automation and robotics due to their lightweight and naturally compliant behavior, which comes from air's compressibility. This compliance, adjustable through pressure control, is essential for safe human-machine interactions and delicate tasks like handling fragile objects. In contrast, hydraulic and electric systems are more rigid and require relatively more complex feedback control to achieve similar flexibility (Daerden et al., 2002). In addition, because pneumatic muscles can produce relatively big forces and make these forces last long without much effort, using them is also advantageous (Schindele et al., 2013).

Pneumatic robots, characterized by their inherent compliance and safety, present unique challenges in control due to their nonlinear behavior and dynamic uncertainties. These robots are particularly advantageous in environments where human-robot interaction is frequent, as their compliant nature reduces the risk of injury. However, the nonlinear dynamics of pneumatic actuators necessitate advanced control strategies to achieve precise and reliable performance.

This experiment addresses the critical need for robust nonlinear control and state estimation techniques for the hand axes of a pneumatic robot, in which the axes of the three pneumatic muscles are regarded as the axes of the robot hand. The experiment aims for optimization of the control process by merging all the compensators (feedback, feedforward, and observer) together, while the model is also considered as similar as possible to the real physical system.

Figure 1 shows two images of the robot, in which its main components are introduced, while the muscles are still not filled with the compressed air. A comprehensive approach is proposed that includes the development of nonlinear controllers designed to

Mirafzal, S.
Nonlinear Control and State Estimation for the Hand Axes of a Pneumatic Robot.
DOI: 10.5220/0013006500003822

In *Proceedings of the 21st International Conference on Informatics in Control, Automation and Robotics (ICINCO 2024) - Volume 2*, pages 236-247
ISBN: 978-989-758-717-7; ISSN: 2184-2809

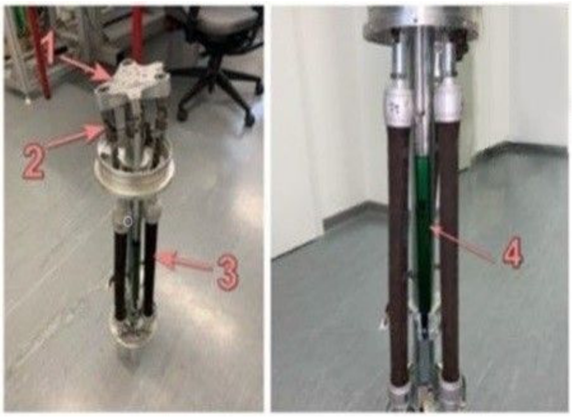

Figure 1: The pneumatic robot. The system consists of four major parts: the joint top plate (1), the joint level (2), the three pneumatic muscles (3) and the two measuring cylinders or potentiometer sensors (4).

enhance trajectory tracking and force control accuracy.

This paper is structured as follows: First, the system is modeled using vectors and various coordinate systems to simplify calculations. While some components, like the chains and fully realistic top plate, are approximated, the model closely represents the real system, resulting in minimal residual errors. Multiple coordinate systems, tailored to the shape, degrees of freedom, and movement directions, make the modeling and logic easier to understand. Transformation matrices facilitate easy variable transfers between coordinate systems. Next, the equations for modeling the pneumatic muscles, including required coefficients, are derived experimentally. New coefficients and muscle characteristics specific to this system are calculated. Furthermore, flatness-based controllers are designed for the desired control response. Initial controllers manage angles that approximate the rotated top plate angles. Secondary controllers manage the pneumatic system, controlling muscle forces based on desired torques and mean force. Moreover, the desired trajectories and feasible variable ranges, considering the real physical system, are explained. Movement, pressure, and force limitations in the pneumatic system are detailed, such as restricted rotation angles of the top plate and pressure limits affecting muscle forces. As the air mass flow rate $\dot{m}$ is considered the manipulated variable of the control system, the initial and the maximum air pressure in the muscles are considered 1 bar and 8 bars, respectively. Additionally, data derived from plotting the results in MATLAB/Simulink indicate that the desired mean force should not fall below 15 newtons to ensure logical system responses. Observers are also used to estimate and compensate for friction as a disturbance. The Kalman Filter is chosen as an optimal method among various observer options. Finally, the entire control system model is evaluated regarding its responses and results, comparing them to expected outcomes. As a result, it is shown that the desired trajectories are being tracked fast with a high accuracy, due to the precise design of the control system, including different compensators. The paper concludes by summarizing the thesis aims and discussing system applications.

2 MULTI-BODY MODELING

As discussed in the introduction, the kinematic modeling of the robot is based on vectors and different coordinate systems, corresponding to the system's parts and their movement directions. The bottom plate, located at the base of the pneumatic muscles, remains stationary, so the initial point (O_1) of the primary coordinate system (system 1) is set at the center of this plate.

The main joint, which allows two perpendicular angular rotations, is located in the middle of the joint level. These rotation angles are the main controlling angles. The center of this joint is the center point (O_2) of the next coordinate system (system 2), formed by rotating system 1 around the x-axis by angle φ_1. The final coordinate system is created by rotating system 2 around its y-axis (y_2-axis) by angle φ_2. Figure 2 shows the simplified system model, including the three coordinate systems.

The robot hand axes design benefits from symmetry, simplifying calculations and modeling. Additionally, the end joints of the pneumatic muscles are aligned with the main joint's center, further easing the modeling process.

2.1 Modeling the Hand Axes and the Pneumatic Actuators

The primary goal in modeling the robot hand axes is to determine the relationship between the vectors of the model's different parts, the rotation angles, and the changes in the lengths of the two potentiometer sensors, X_{10} and X_{11}, as shown in figure 1. Since X_{10} and X_{11} are the only means of measuring the current positions or lengths of other components, establishing this relationship is crucial. The modeling process is conducted in several steps, as described in the following sections.

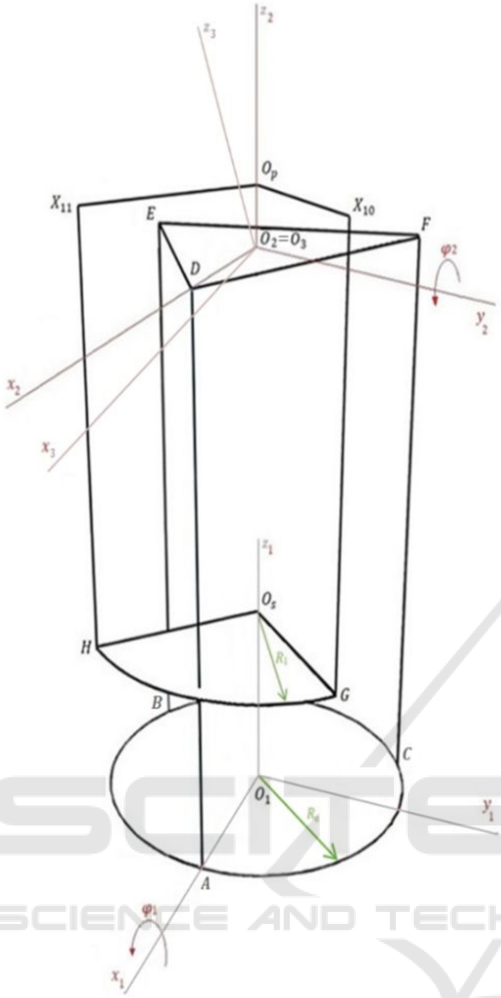

Figure 2: The simplified model, including the bottom surface of the muscles, top triangular surface and the sensors' bottom and top positions.

2.1.1 Selecting Different Coordinate Systems

As mentioned briefly earlier, the modeling involves three coordinate systems. To explain the system's behavior better, the second and third coordinate systems are placed at the main joint just below the top surface. Key points for calculations include the centers of the pneumatic muscle top joints (*D*, *E*, and *F*), as shown in figure 2.

2.1.2 Vector-Based Modeling

In the modeling approach, muscles and their chains together are treated as vectors in three-dimensional space for calculations and coding purposes. As previously mentioned, the goal is to relate these vectors to the rotation angles of the main central joint, O_2, to the outputs from sensors X_{10} and X_{11}. For this purpose, a vector must represent each pneumatic muscle.

The secondary coordinate system, system 2, centered at the main joint with origin O_2, simplifies calculations for vectors such as ${}^{1}\vec{r}_{D\,1}$, which represent distances like O_1 to D. These vectors are transformed into the initial coordinate system using a straightforward multiplication by a transformation matrix explained in section 2.1.3.

2.1.3 Derivation of the Transformation Matrix

In subsection 2.1.2, the concept of representing vectors in different coordinate systems is introduced. In order to calculate each vector component in a new coordinate system, a transformation matrix is required. The transformation matrix to form a new coordinate system 3, after two consequent rotations about x- and y-axes, can be stated as (Woernle, 2016)

$$
{}^{13}T = \begin{bmatrix} \cos\varphi_2 & 0 & \sin\varphi_2 \\ \sin\varphi_1 \sin\varphi_2 & \cos\varphi_1 & -\sin\varphi_1 \cos\varphi_2 \\ -\cos\varphi_1 \sin\varphi_2 & \sin\varphi_1 & \cos\varphi_1 \cos\varphi_2 \end{bmatrix}. \tag{1}
$$

Similarly, this 2 degrees of freedom (DOF) system involves rotations around the main center joint along x- and y-axes. Thus, analogous to the equation (1), separate transformation matrices are developed: ${}^{12}T$, ${}^{23}T$, and ${}^{13}T$, representing movements between coordinate systems. Therefore, each vector can be calculated in the third coordinate system to ease the calculations, by having its components in the first coordinate system as follow

$$
{}^{1}\vec{r} = {}^{13}T\ .\,{}^{3}\vec{r}\ . \tag{2}
$$

2.2 Experimental Identification of Unknown Parameters

Pneumatic systems involve diverse muscle types, each with specific characteristics like air pressure requirements for contraction. A key challenge in pneumatic muscles is hysteresis. This effect in pneumatic muscles arises from the relative motions occurring during muscle inflation or deflation, which increase the system's nonlinearities and complexities (Vo et al., 2010). Internal friction between the aramid fibers and the surrounding elastic material also contributes to hysteresis (Deaconescu et al., 2016). Despite its potential impact, hysteresis is negligible in this experiment and is omitted in the pneumatic control model design.

The complete force characteristic F_{Mi} can be stated as (Schindele et al., 2013)

$$F_{Mi} = F_{Mi,st} + F_{Mi,hys}\ , \tag{3}$$

where $F_{Mi,st}$, and $F_{Mi,hys}$ denote the static muscle force and the hysteresis effect, respectively.

Because the hysteresis effect, $F_{Mi,hys}$, is negligible in this experiment, therefore, complete force characteristic F_{Mi} is considered approximately equal to the static muscle force $F_{Mi,st}$. As a result,

$$F_{Mi} \approx F_{Mi,st}\ . \tag{4}$$

The muscle force depends on internal pressure p_{Mi} and contraction length Δl_{Mi} approximated by

$$F_{Mi,st}(p_{Mi}, \Delta l_{Mi}) = \begin{cases} \bar{F}_{Mi}(p_{Mi}, \Delta l_{Mi}) & if\ \bar{F}_{Mi} < 0 \\ 0 & else \end{cases}, \tag{5}$$

$$\bar{F}_{Mi}(p_{Mi}, \Delta l_{Mi}) = \underbrace{\sum_{m=0}^{3} (a_m \Delta l_{Mi}{}^{m})}_{=f_{1i}(\Delta l_{Mi})} p_{Mi} - \underbrace{\sum_{n=0}^{4} (b_n \Delta l_{Mi}{}^{n})}_{=f_{2i}(\Delta l_{Mi})}\ . \tag{6}$$

The amounts of the coefficients a_m and b_n can be defined experimentally. The Recursive Least Squares (RLS) method is a reliable approach for parameter estimation (Xie et al., 2011), requiring extensive data collection—about 20,000 states of p_{Mi} , Δl_{Mi} , and F_{Mi} are practically analyzed in the university laboratory. Figure 3 illustrates results for 10 internal muscle pressures.

By using the RLS method, the quantities for a_m and bn can be calculated.

Consequently, the resulting three-dimensional figure for the identified force characteristic of the pneumatic muscle can be plotted as figure 4.

3 DESIGN OF THE NONLINEAR MIMO AND INVERSE SYSTEM MODEL

The flatness-based control approach is chosen, due to the suitable trajectory tracking structure and inclusion of the feedforward compensator, as it is discussed in more detail in section 3.

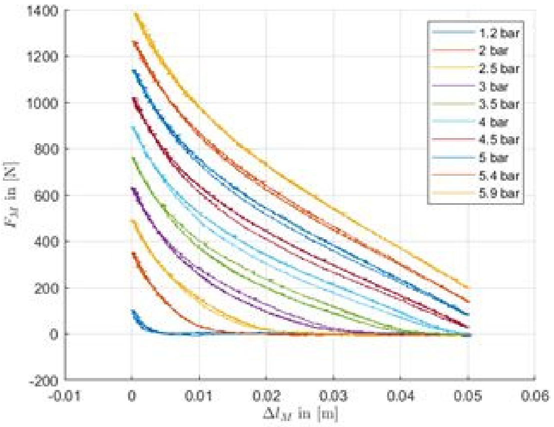

Figure 3: The resulting curves, including the relationship between the created muscle force and the contraction length for 10 different pressures.

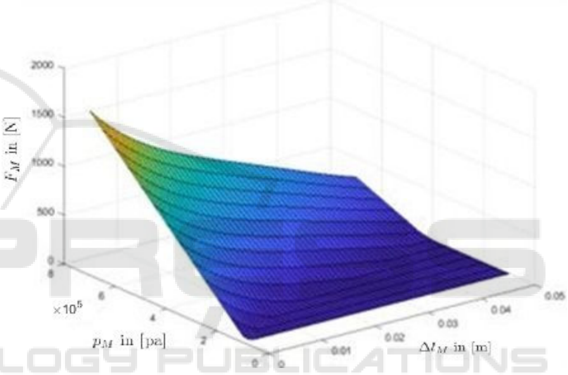

Figure 4: The force characteristic of the pneumatic muscle.

In order to design a complete flatness-based feedback cascade control for a Multiple-Input Multiple-Output (MIMO) system, it is necessary to design not only the MIMO system model, but also the nonlinear inverse system model. The inputs of the MIMO system model are the controlled outputs of the nonlinear inverse model.

The design of the MIMO system and the inverse system models is discussed in this section, however, the design of the flatness-based controllers is discussed in the next section, as mentioned earlier.

3.1 Design of the Nonlinear MIMO System Model

The initial inputs of the MIMO system model are the controlled muscle forces, which must sequentially be transformed into the torques, rotation angles, and sensors' lengths change. Figure 5 shows the schematic block diagrams of the whole MIMO system model.

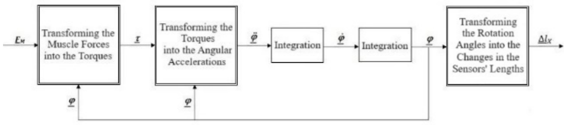
E_M
Transforming the Muscle Forces into the Torques
$\underline{\tau}$
Transforming the Torques into the Angular Accelerations
$\ddot{\underline{\varphi}}$
Integration
$\dot{\underline{\varphi}}$
Integration
$\underline{\varphi}$
Transforming the Rotation Angles into the Changes in the Sensors' Lengths
$\underline{\Delta l_x}$
$\underline{\varphi}$
$\underline{\varphi}$

Figure 5: The block diagrams of the whole MIMO system model.

3.1.1 Transforming the Muscle Forces into the Toques

Based on the physical laws of angular rotation, the corresponding torques can be calculated as follows

$$\underline{\tau} = \underline{r} \times \underline{F} \ , \qquad (7)$$

where $\underline{\tau}$, $\underline{r}$ and $\underline{F}$ represent the resulting torques matrix, the distance vector and the muscles forces matrix, respectively. Figure 6 shows the simplified model of the robot hand axes in which the related vectors to calculate the first muscle vector are shown in blue.

The step-by-step calculations to get the first muscle vector are as follows

$$^{1}\vec{r}_{D3} = {}^{13}T \ . \ {}^{3}\vec{r}_{D3} \ , \qquad (8)$$

$$^{1}\vec{r}_{D1} = {}^{1}\vec{r}_{31} + {}^{1}\vec{r}_{D3} \ , \qquad (9)$$

$$^{1}\vec{r}_{DA} = {}^{1}\vec{r}_{D\ 1} - {}^{1}\vec{r}_{A1} . \qquad (10)$$

If the coordinates of the first muscle force vector $^{1}\vec{r}_{DA}$ along x- , y- and z-axis are considered $r_{DA,x}, r_{DA,y}$, and $r_{DA,z}$, respectively, the whole length of $\vec{r}_{DA}$ can be stated as

$$l_{DA} = \sqrt{{r_{DA,x}}^2 + {r_{DA,y}}^2 + {r_{DA,z}}^2} \ . \qquad (11)$$

Therefore, the unit vector of the first muscle force vector in the first coordinate system can be expressed as

$$^{1}e_{DA} = \frac{1}{l_{DA}} \ . \ {}^{1}\vec{r}_{DA} \ . \qquad (12)$$

Similarly, the unit vector of the second and third muscle vector in the first coordinate system can be determined by the following equations

$$^{1}e_{EB} = \frac{1}{l_{EB}} \ . \ {}^{1}\vec{r}_{EB} \ , \qquad (13)$$

$$^{1}e_{FC} = \frac{1}{l_{FC}} \ . \ {}^{1}\vec{r}_{FC} \ . \qquad (14)$$

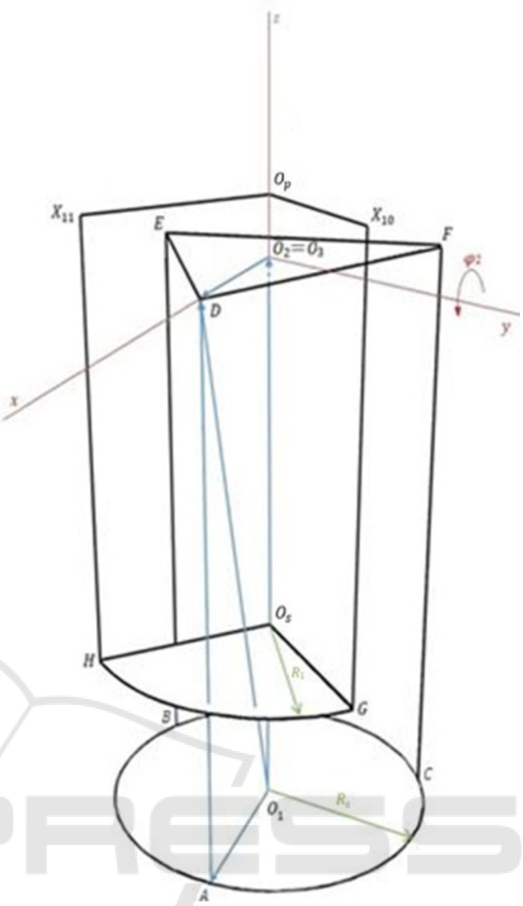

Figure 6: The simplified system model to calculate the first muscle vector, $\vec{r}_{DA}$.

To correctly show the pulling direction of the muscle forces, the unit matrices must be multiplied by -1, resulting in the following vector

$$^{1}\vec{F}_{M1} = - F_{M1} \ . \ {}^{1}e_{DA} \ , \qquad (15)$$

where F_{M1} denotes the scalar value of the produced forces of the first pneumatic muscle.

To simplify the further calculations, the related vectors must be brought into the third coordinate system, therefore

$$^{3}\vec{F}_{M1} = {}^{13}T^{-1} \ . \ {}^{1}\vec{F}_{M1} \ , \qquad (16)$$

Consequently, based on the equation (7) the produced torque of the first muscle can be calculated as follows

$$^{3}\vec{\tau}_1 = {}^{3}\vec{r}_1 \times {}^{3}\vec{F}_{M1} \ , \qquad (17)$$

The corresponding vectors of the second and the third muscles can be calculated similarly. As a result, the aggregate torque can be stated as

$$\sum \underline{\tau} = {}^{3}\vec{\tau}_{1} + {}^{3}\vec{\tau}_{2} + {}^{3}\vec{\tau}_{3} \ . \tag{18}$$

3.1.2 Transforming the Torques into the Angles

According to the physical laws of rotation, the relationship between the angular velocity and the corresponding torque stated as

$$\sum \underline{\tau} = J \cdot \ddot{\underline{\varphi}} \ , \tag{19}$$

where J denotes the moment of inertia of the mass to which the torque has been applied. Therefore, the angular velocity can be stated as

$$\ddot{\underline{\varphi}} = \underline{J}^{-1} . \sum \underline{\tau} \ . \tag{20}$$

In order to calculate J, the top plate is considered a triangular prism. The detailed calculations can be found in (Mirafzal, 2023).

Correspondingly, the matrix of the rotation angles can be calculated easily by two consecutive integrator blocks in Simulink.

3.1.3 Transforming the Angles into the Changes in the Sensors' Lengths

To calculate the sensors' lengths change, each of the corresponding sensor vectors must be calculated separately. Figure 7 shows the simplified model of the robot hand axes in which the related vectors to calculate the X_{11} sensor vector are shown in blue.

Consequently, the X_{11} sensor vector, $\vec{r}_{11H}$, can be calculated by the following equations.

$${}^{1}\vec{r}_{11\,3} = {}^{13}T \ \ . {}^{3}\vec{r}_{11\,3} , \tag{21}$$

$${}^{1}\vec{r}_{11\,1} = {}^{1}\vec{r}_{3\,1} \ + {}^{1}\vec{r}_{11\,3} , \tag{22}$$

$${}^{1}\vec{r}_{11\,S} = {}^{1}\vec{r}_{11\,1} \ - {}^{1}\vec{r}_{S\,1} , \tag{23}$$

$${}^{1}\vec{r}_{11\,H} = {}^{1}\vec{r}_{11\,S} - {}^{1}\vec{r}_{H\,S} \ . \tag{24}$$

As a result, the length of the X_{11} sensor can be stated as

$$l_{X\,11} = \sqrt{r_{11H,x}{}^{2} + r_{11H,y}{}^{2} + r_{11H,z}{}^{2}} \ . \tag{25}$$

Accordingly, if the initial sensor length is considered $l_{s0,}$ the length change of the X_{11} sensor can be stated as

$$\Delta l_{X\,11} = l_{X\ 11} - l_{s\,0} \ . \tag{26}$$

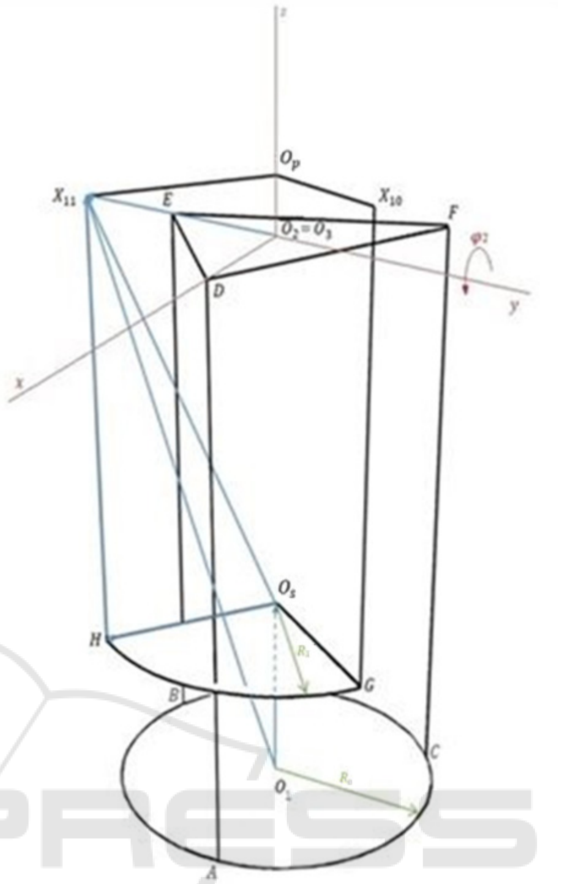

Figure 7: The simplified system model to calculate the X_{11} sensor vector, $\vec{\boldsymbol{r}}_{11\text{H}}$.

A similar approach can be followed to calculate the length change of the X_{10} sensor, $\Delta l_{X\,10}$.

3.2 Design of the Nonlinear Inverse System Model

As shortly discussed previously, it is necessary to design an inverse system model to complete the flatness-based control block diagram. As the first inverse block, the output values for the changes in the sensors' lengths, $\Delta l_{X\,11}$ and $\Delta l_{X\,10}$, must be converted into the corresponding rotation angle values, φ_1 and φ_2. These output rotation angles as well as the desired rotation angles are the inputs of the first flatness-based angle controller, which is explained in more detail in section 4. Figure 8 shows the schematic block diagrams of the whole inverse system model.

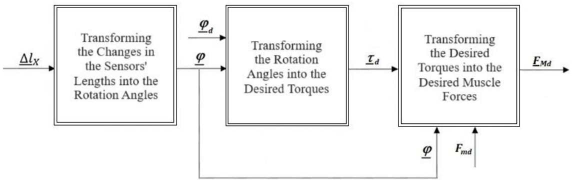
Δl_X
Transforming the Changes in the Sensors' Lengths into the Rotation Angles
φ_d
φ
Transforming the Rotation Angles into the Desired Torques
τ_d
Transforming the Desired Torques into the Desired Muscle Forces
F_{Md}
φ
F_{md}

Figure 8: The block diagrams of the whole inverse system model, in which φ_d, τ_d, F_{md}, and F_{Md} represent the desired rotation angle, torque, mean force, and muscle force, respectively.

3.2.1 Transforming the Changes in the Sensors' Lengths into the Rotation Angles

Although the changes in the sensors' lengths can be analytically calculated from the rotation angles in the MIMO system model, the rotation angles cannot be calculated analytically by the MATLAB "solve" function, due to the complexity of the nonlinear inverse model to calculate the angles. Therefore, a lookup table is employed in Simulink to output the rotation angles from the inputs, which are the changes in the sensors' lengths, numerically.

3.2.2 Transforming the Rotation Angles into the Desired Torques

Given the rotation angles, that result from the lookup table, the desired torques can be outputted. To do this, the designed angle controller, which is described in the equation (32) in section 4, must take the rotation angles from the lookup table, as well as the desired angles, that can be chosen by considering the system limitations. Next, the controller outputs the second derivative of the rotation angles, as the controlled values. Then, the desired torques can be calculated, having the moment of inertia, as described in the equation (33) in section 4.

3.2.3 Transforming the Desired Torques into the Desired Muscle Forces

The conversion of the desired torques to the desired muscle forces could be challenging, due to the complex nonlinear structure of the equations. Nevertheless, by using an innovative approach, the problem is solved completely analytically.

Due to the fact that the coefficients of the desired muscle forces, F_{M1}, F_{M2} and F_{M3} , can be totally separated from the muscle forces in the matrix of the desired torques, the desired torques matrix can be rewritten as follows

$$A_f \cdot \underline{F}_{Md} = \begin{bmatrix} \tau_{1d} \\ \tau_{2d} \\ F_{md} \end{bmatrix} , \tag{27}$$

where A_f and $\underline{F}_{Md}$ can be written as

$$A_f = \begin{bmatrix} a_f & b_f & c_f \\ d_f & e_f & f_f \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix} , \tag{28}$$

$$\underline{F}_{Md} = \begin{bmatrix} F_{M1d} \\ F_{M2d} \\ F_{M3,d} \end{bmatrix} , \tag{29}$$

and F_{md} denotes the desired mean muscle force.

As a result, the desired muscle force matrix can be calculated by

$$\underline{F}_{Md} = A_f^{-1} \cdot \begin{bmatrix} \tau_{1d} \\ \tau_{2d} \\ F_{md} \end{bmatrix} . \tag{30}$$

The desired muscle forces are used as the reference values for the flatness-based force controller, which is discussed in section 4.

4 SYSTEM ANALYSIS AND DESIGN OF A NONLINEAR TRAJECTORY TRACKING CONTROL

Flatness-based control is a method that offers a cost-effective and straightforward approach for nonlinear models, enabling easy tracking of desired inputs with the inclusion of feedforward as a compensator.

The general formula to design a flatness-based controller can be stated as (Lévine, 2009)

$$v = {y_{id}}^{(n)} + \sum_{j=0}^{n-1} a_j \left({y_{id}}^{(j)} - {y_i}^{(j)} \right) , \tag{31}$$

where v denotes the control input to the system, y_{id} represents the desired flat output or the reference trajectory, a_j refers to the feedback coefficient ensuring stability (typically chosen to form a Hurwitz polynomial), and n is the relative degree of the system.

This approach effectively controls angles and muscle forces (or pressures) and is implemented in the control system, detailed in this section.

4.1 Design of the Flatness-based Angle Controller

The angles φ_1 and φ_2 are determined using sensor feedback and the relationship between its length change to the other components of the robot by means of the vector-based modeling. Using numerical methods, including lookup tables in MATLAB/Simulink, sensor data can be converted into controlled outputs.

For systems suitable for flatness-based control, like this system, the design is straightforward. As the relative degree of the angels in the system is 2, according to equation (31), the control equations for the main joint angles can be stated as

$$\ddot{\varphi}_i = \ddot{\varphi}_{id} + a_0 \, . \, (\varphi_{id} - \varphi_i) + a_1 \, . \, (\dot{\varphi}_{id} - \dot{\varphi}_i) \, . \quad (32)$$

Here, i={1, 2} for each of the angles and $\ddot{\varphi}_{id}$ is the desired angular acceleration, also acting as the feedforward compensator, while a_0 and a_1 are positive constants ensuring system stability.

Desired angle trajectories, e.g. varying sine functions, can be freely chosen within physical constraints, detailed further in section 5.

Next, torques are computed from the angular accelerations, which are used to calculate total torques about x- and y-axes

$$\begin{bmatrix}\tau_{1d}\\ \tau_{2d}\end{bmatrix} = \begin{bmatrix}J_{11} & J_{12}\\ J_{21} & J_{22}\end{bmatrix} \cdot \begin{bmatrix}\ddot{\varphi}_1\\ \ddot{\varphi}_2\end{bmatrix}, \quad (33)$$

where J denotes the moment of inertia of the mass at the top of the robot hands to which the torques have been applied, that has been approximated by a triangular prism. This process is implemented in the angle controller.

4.2 Design of the Flatness-based Force Controller of the Pneumatic Muscles

In addition to rotation angles, it is crucial to control the pneumatic muscle forces through a separate tracking controller.

The muscle force F_{Mi} comprises the static force $F_{Mi,st}$ and the hysteresis force $F_{Mi,hys}$, with the latter being negligible, as written in equation (4) in section 2.

As mentioned earlier, the static force is calculated using coefficients a_m and b_n determined via the Recursive Least Squares (RLS) method, as described in section 2, equations (5) and (6). The derivative of muscle pressure $\dot{p}_{Mi}$ is expressed as (Schindele et al., 2013)

$$\dot{p}_{Mi} = \frac{n}{V_{Mi} + n\frac{\partial V_{Mi}}{\partial p_{Mi}}} \left[R_L T_{Mi} \dot{m}_{Mi} - \frac{\partial V_{Mi}}{\partial \Delta l_{Mi}} \frac{\mathrm{d}\Delta l_{Mi}}{\mathrm{d}z_i} \dot{z}_i \, p_{Mi}\right]$$

$$= k_{ui}\,(\Delta l_{Mi}, p_{Mi})\,\dot{m}_{Mi} - \; k_{pi}\left(\Delta l_{Mi}, \Delta\dot{l}_{Mi}, p_{Mi}\right) p_{Mi} \, , \quad (34)$$

where i = {1, 2, 3}, V_{Mi} denotes the volume of the pneumatic muscles, $R_L = 287$ the air gas constant, $n = 1.26$ the identified polytropic exponent, $T_{Mi} = 294\ K$ the internal temperature, and Δl_{Mi} the contraction length.

The volume of the pneumatic muscles V_{Mi} is defined as

$$V_{Mi}\,(\Delta l_{Mi}, p_{Mi}) \; = \sum_{k=0}^{3}\left(a_k \Delta {l_{Mi}}^k\right) p_{Mi} + \sum_{l=0}^{3} b_l \Delta {l_{Mi}}^l \, , \quad (35)$$

where the coefficients a_k and b_l can be identified experimentally, just like the polynomial function of the muscle forces.

The mass flow rate $\dot{m}_{Mi}$ is given by:

$$\dot{m}_{Mi} = \frac{f_{1i}\, k_{pi}\, p_{Mi} - \dot{f}_{1i}\, p_{Mi} + \dot{f}_{2i} + v_i}{f_{1i}\, k_{ui}} \, , \quad (36)$$

where f_{1i} and f_{2i} are derived from the equation (6). By considering $v_i = \dot{F}_{Mi}$, similar to designing the angle controller based on equation (31), and considering that the relative degree of the muscle forces in the system is 1, the flatness-based force controller can be formulated as

$$\dot{F}_{Mi} = \dot{F}_{Mid} + a_0 \, . \, (F_{Mid} - \; F_{Mi}) \, . \quad (37)$$

Here, i={1, 2, 3} for each of the muscles, and $\dot{F}_{Mid}$ act as the feedforward compensator, while a_0 must be chosen positive to ensure the system stability.

5 DETERMINATION OF THE DESIRED TRAJECTORIES

The robot pneumatic system has restrictions such as the rotation angles, φ_1 and φ_2, of the main joint and the maximum pressure from the pneumatic control unit (PCU), due to its physical characteristics. These limitations must be incorporated into the model for realism. Typically, a target polynomial is used for the desired angles to improve control response and derivatives (Mirafzal, 2023). This section details these system restrictions, methods, and resulting trajectories.

5.1 Minimum Force/Pressure Requirements

The pulling force of the pneumatic muscles is limited by the compressed air pressure. The initial pressure is approximately 1 bar, while the laboratory maximum is 8 bars due to the pneumatic control unit's constraints. Additionally, the muscle forces have minimum and maximum limits. Tests showed that the desired mean force, F_m, should not fall below 15 newtons, as lower values result in unrealistic negative forces. Consequently, the maximum forces and torques are functions of the compressed air pressure, which is capped at 8 bars, limiting the corresponding forces and torques.

5.2 Feasible Trajectories for the end Effector Rotation Angles

The primary constraints of the pneumatic robot system include the feasible rotation angles due to physical limitations. Rotations about the z-axis are not allowed, and rotations about the *x*- and *y*-axes are limited and cannot approach e.g. 90 degrees. Not all combinations of the main angles, φ_1 and φ_2, are practically achievable. Practical trajectory plots for the changes in the sensors' length are created in the laboratory to determine the possible angles from 1377 different states, as shown in figure 9.

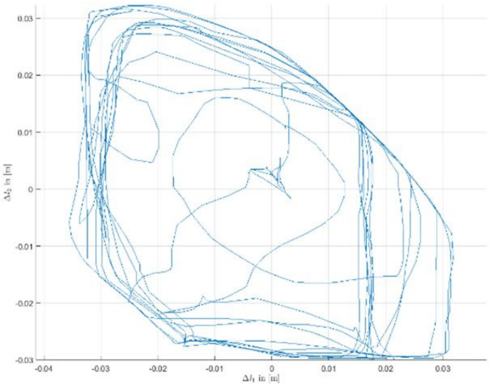

Figure 9: The possible trajectories for the changes in the X_{10} and X_{11} sensors' lengths that are measured practically in the laboratory for 1377 reachable points.

As a result, the corresponding φ_1 and φ_2 angles can be plotted using MATLAB, as shown in figure 10.

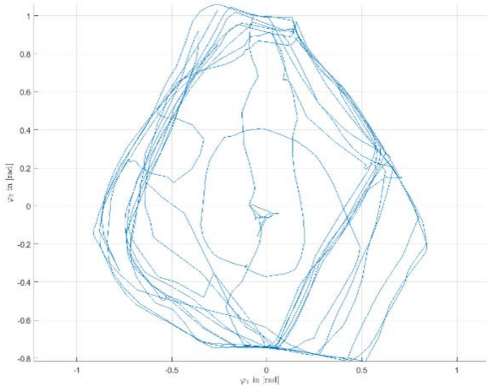

Figure 10: The corresponding trajectories for the main angles for all of the 1377 measured points resulting from the sensors data.

Some maximum points in Figure 10 are achieved manually, not through full compressed air pressure, indicating some unreachable points. In addition, the plot should be symmetric, due to the symmetric structure of the system, revealing missing points.

6 ESTIMATION OF THE STATE AND DISTURBANCE

In advanced control systems, feedback alone is often insufficient for achieving quick and accurate responses due to errors, noise, and disturbances. Feedback compares current outputs with set points to identify errors but compensates only after errors occur, leading to potential delays. In addition to feedback, two main compensators, including feedforward and observer (estimator), are typically employed to address the mentioned issues. Feedforward, as explained briefly in the flatness-based control design, anticipates errors using prior knowledge of disturbances, compensating before errors are detected by feedback. Furthermore, the observer, or estimator, is used when employing sensors is impractical or costly, as it estimates system states and reduces noise and cost, making it essential for complex systems where direct measurement is impossible. These strategies work together to enhance error correction and improve system performance.

The system model described earlier is idealized and does not account for real-world disturbances, particularly friction, which significantly affects mechanical systems. The nonlinear friction model for such systems can be expressed as (Mirafzal, 2023)

$$\tau_u = \tau_R \,.\, tanh\left(\frac{\dot{\varphi}}{\varepsilon}\right), \tag{38}$$

where τ_u denotes the nonlinear friction, τ_R and ε are constants, with ε typically set to a small value, such as 0.01, based on experience. Damping, especially speed-proportional damping, also affects the system and is represented as $b.\dot{\varphi}$, where b is a constant relating angular velocity to friction.

Thus, the total friction torque can be stated as

$$\tau_f = b.\dot{\varphi} + \tau_R \,.\, tanh\left(\frac{\dot{\varphi}}{\varepsilon}\right). \tag{39}$$

In this experiment, the constants are considered $b = 0.1, \tau_R = 0.0015$, and $\varepsilon = 0.01$ The state space representation for the torque about each rotation axis is given by

$$\begin{bmatrix} \dot{\varphi} \\ \ddot{\varphi} \end{bmatrix} = \begin{bmatrix} \dot{\varphi} \\ \frac{\tau}{J} - \frac{b}{J}\,\dot{\varphi} - \frac{1}{J}\,\tau_R \,.\, tanh\left(\frac{\dot{\varphi}}{\varepsilon}\right) \end{bmatrix}, \tag{40}$$

where

$$\dot{\underline{y}}_m = f(\underline{y}_m, \tau, \tau_u) \quad , \tag{41}$$

and

$$\underline{y}_m = \underline{x} = [\varphi \ \dot{\varphi}]^T \quad . \tag{42}$$

Here, $\underline{y}_m$ refers to the state vector. Equation (40) can be applied separately to each rotation angle, φ_1 and φ_2, with τ representing the torque about the *x*- and *y*-axes, respectively.

Therefore, this friction will be estimated and used as a compensator in the control system model.

In this system model, various states, such as the main angles of the central joint, can be estimated. However, compensating friction disturbances has higher priority. Given that friction influences the aggregate torques and the model uses a linear approach for these calculations, the Kalman Filter can estimate friction in this system (Welch et al., 2006). Consequently, it is employed to estimate the disturbance torque attributed to friction. The detailed design of the Kalman Filter can be found in (Mirafzal, 2023).

The next section will compare the estimation results with the noisy model disturbance torque and analyze the overall model control response.

7 SIMULATIVE INVESTIGATION AND EVALUATION

To evaluate the control system's response, adjustments are made to simulate real-world conditions with disturbances, noise, and initial conditions. Potentiometers and pressure sensors received noise with specified variances. System limits and delays are also set. Friction is estimated and compensated using a Kalman Filter.

The system's response is evaluated against desired trajectories for both angles and muscle forces. The next step is to tune the controllers to receive a satisfactory response of the system. Effective tuning of flatness-based controllers is achieved by adjusting coefficients iteratively. To adjust the coefficients of the controllers, the eigenvalues should be chosen for each controller. Through multiple rounds of trial and error and examining how well the outputs can track the desired trajectories in Simulink, the eigenvalues of -225 and -1100 are chosen for the angle controller and the force controller, respectively. As a result, the characteristic polynomial for the angle controller can be stated as

$$\begin{aligned} P(S) &= (S + 225)^2 \\ &= S^2 + 450\,S + 50625\,. \end{aligned} \tag{43}$$

Therefore, the coefficients of the flatness-based angle controller in equation (32) can be chosen as a_0 = 50625 and a_1 = 450.

Similarly, the characteristic polynomial for the force controller can be stated as

$$P(S) = S + 1100\,. \tag{44}$$

As a result, the coefficient of the flatness-based force controller in equation (37) can be chosen as a_0 = 1100.

After implementation of the selected control coefficients, the tracking behavior of the system shows high accuracy and fast response to the desired trajectories.

Figures 11 and 12 show the tracking behavior of the system as a response to the desired sine wave trajectories. Figure 13 also illustrates how well the Kalman Filter estimates the friction torques.

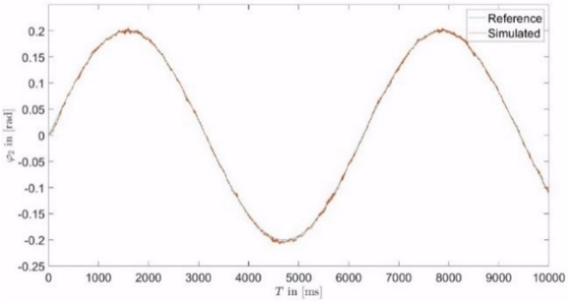

Figure 11: The output of the second angle, φ_2, compared to sample reference sine wave trajectories.

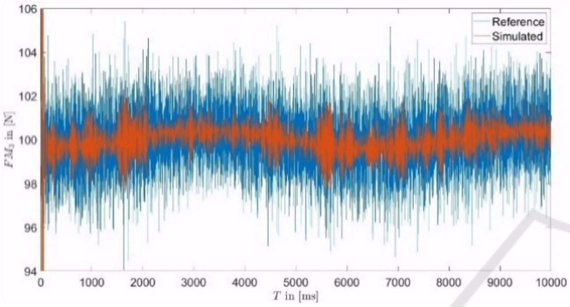

Figure 12: The output of the third muscle force, F_{M3}, compared to the reference values.

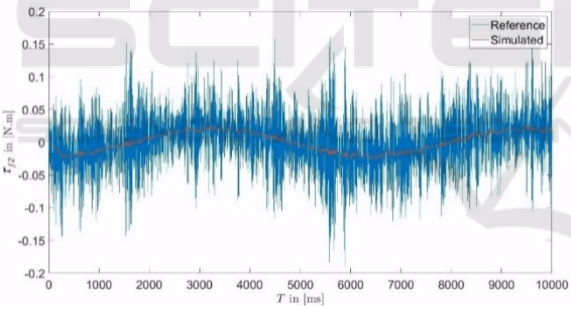

Figure 13: The estimated friction torque about y-axis compared to the reference values.

Similarly, figures 14 and 15 illustrate how well the system follows more complex desired trajectories for the first rotation angle and the third muscle force, respectively, demonstrating effective control despite assumed disturbances. The Kalman Filter also effectively estimates friction torques, as shown in Figures 16, aligning closely with real values under various conditions.

Overall, the simulation results validate the proper determination of the physical conditions, as well as the suitability of the controller designs and the coefficient selections, ensuring the system performs well under different operating conditions, in which factors such as friction torque play important roles in the system's response.

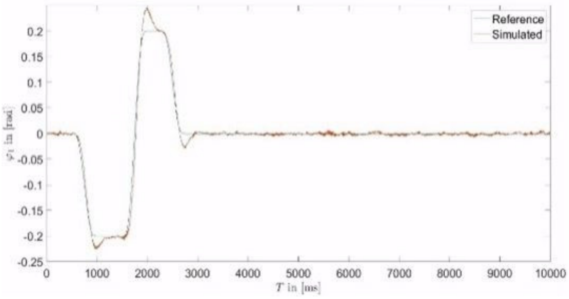

Figure 14: The output of the first angle, φ_1, compared to sample reference trajectories.

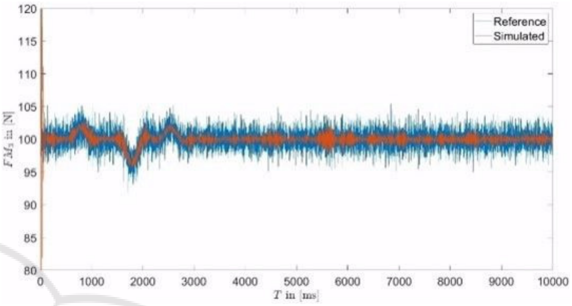

Figure 15: The output of the third muscle force, F_{M3}, compared to the reference values.

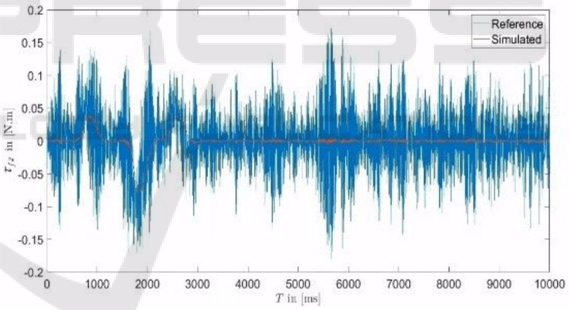

Figure 16: The estimated friction torque about y-axis compared to the reference values.

8 CONCLUSIONS

This experiment focuses on modeling and controlling a pneumatic robot hand axes system, with a key emphasis on central joint rotation angles and pneumatic muscle forces. Using transformation matrices simplifies kinematic relationships, while Recursive Least Squares helps characterize pneumatic muscle behavior.

A flatness-based control approach is chosen for its ability to effectively track desired trajectories despite system non-linearities.

To enhance realism, disturbances like friction torques and sensor noise are introduced and mitigated using a Kalman Filter observer. Tuning controller coefficients through iterative testing optimizes system response, ensuring outputs closely matches desired values.

In conclusion, this research demonstrates the practicality and efficiency of pneumatic actuators in generating substantial forces, supported by robust flatness-based controllers. Future work could explore alternative actuators and control strategies like the Backstepping or the Sliding-Mode-Control for further system enhancement.

REFERENCES

Daerden, F., and Lefeber, D. (2002). *Pneumatic artificial muscles: Actuators for robotics and automation.* European Journal of Mechanical and Environmental Engineering, Vol. 47, No. 1, pages 11–22.

Deaconescu, T., and Deaconescu, A. (2016). *Study Concerning the Hysteresis of Pneumatic Muscles, Applied Mechanics and Materials*, Trans Tech Publications, Switzerland. Vol. 841, pages 209-214.

Franklin, Powell, and Emami-Naeini. (2019). *Feedback Control of Dynamic Systems,* Pearson. 8th edition.

Lévine, J. (2009). *Analysis and Control of Nonlinear Systems: A Flatness-based Approach,* Springer.

Mirafzal, S. H. (2023). *Nonlinear Control and State Estimation for the Hand Axes of a Pneumatic Robot.* Master's thesis at the Faculty of Mechanical Engineering and Ship Technology, University of Rostock.

Schindele, D., and Aschemann, H. (2013). *Comparison of Cascaded Backstepping Control Approaches with Hysteresis Compensation for a Linear Axis with Pneumatic Muscles.* IFAC, Vol. 46, No. 23, pages 773-778.

Vo, T. M., Kamers, Ramon, H., and Van Brussel, H. (2010). *Characterization of Hysteresis in a Pneumatic Muscle Manipulator with Accounting for the Creep Effect,* IFAC, Vol. 43, No. 21, pages 296-302.

Welch, G., and Bishop, G. (2006). *An Introduction to the Kalman Filter.* Department of Computer Science, University of North Carolina at Chapel Hill.

Woernle, C. (2016). *Mehrkörpersysteme. Eine Einführung in die Kinematik und Dynamik von Systemen starrer Körper*, Springer-Vieweg. Wiesbaden. 2nd edition.

Xie, L., Yang, H., and Ding, F. (2011). *Recursive least squares parameter estimation for non-uniformly sampled systems based on the data filtering.* Mathematical and Computer Modelling No. 54, pages 315–324.