

RAM

● ROBOTICS
AND
MECHATRONICS

DESIGN AND DEVELOPMENT OF A MAGNETIC NON-LINEAR ELASTIC ELEMENT AND CONTROL FOR PROGRESSIVE SERIES ACTUATION

B. (Boi) Okken

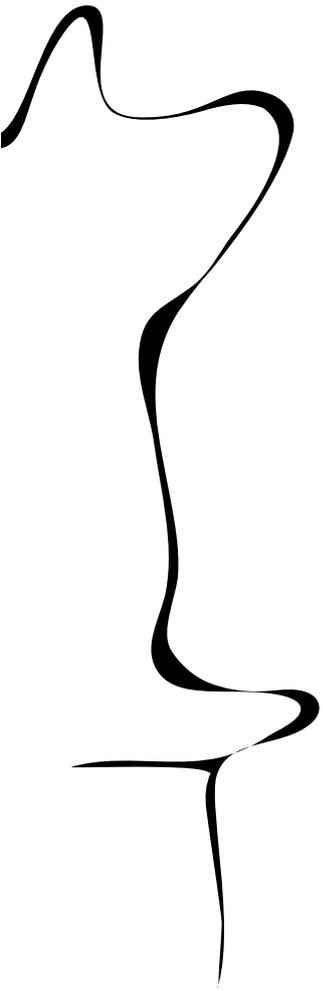
MSC ASSIGNMENT

Committee:

prof. dr. ir. S. Stramigioli
dr. ir. W. Roozing
prof. dr. ir. H. van der Kooij

February, 2021

005RaM2021
Robotics and Mechatronics
EEMCS
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands



This document was designed with double sided printing in mind to save the environment.

For this purpose this page is intentionally left blank

Design and Development of a Magnetic Non-linear
Elastic Element and Control for Progressive Series
Actuation

Faculty EEMCS, dept. RAM

**UNIVERSITY
OF TWENTE.**

B. Okken, Bsc.

February 24, 2021

This document was designed with double sided printing in mind to save the environment.

For this purpose this page is intentionally left blank

Abstract

Traditional robotics use stiff actuators, which are inherently unsafe to work with. Series elastic actuators were developed to deal with this issue, and provided a compromise between performance and safety metrics. By implementing a series elastic element, motor dynamics are decoupled and the system will have a limited bandwidth and thus reduced control, but increased safety. Additional benefits of the series elastic element are convenient and cheap force measurement, and the element can be used as energy storage. Non-linear series elastic elements are investigated to tackle the compromise, offering the safety and torque resolution of low stiffness actuators, with the higher bandwidth of high stiffness actuators.

Most commonly non-linear stiffness is generated by deflecting a linear spring in a non-linear fashion. In this work, a novel way of generating non-linear stiffness was designed with the use of magnets. This involves a parametric investigation and the presentation of a model. Finally, theory is made into reality with a practical proof-of-concept prototype magnet-based series elastic element. This element is characterized and compared against the model to verify performance and model accuracy.

Most control applied to non-linear series elastic actuators is traditional linear control. This means that the inherent advantages of non-linear series elastic actuators are not fully exploited. A gain-scheduled controller has been proposed, designed and simulated to fully exploit the advantages a non-linear stiffness actuator presents without compromising stability or adding additional cost.

Declaration

I hereby declare that the presented thesis is composed solely by me, B. Okken for the purpose of a graduation thesis as a candidate for the title of MSc. Electrical Engineering. The work presented is my own and has not been submitted for any other degree or professional qualification except the afro-mentioned Msc. title. All work is my own work, except if indicated otherwise. Any findings or work that are not my own contain references to the original author in the bibliography.

Parts of this work may be submitted for publication at a future date.

Preface and Acknowledgements

The research presented here is both the end of one journey, and the start of another. On one hand, I have completed by master at the University of Twente, and on the other it will mark the beginning of a different phase of life. Whether that will be in research, the industry or pursuing a PhD, one thing is certain, this thesis is an important milestone in my life.

I was looking forward to combine and apply knowledge learned over the course of my masters in a project, and at the same time push the boundaries of a field that I previously did not know a lot about. It is always both rewarding and humbling to study a relatively new subject and contribute to it. Of course, even after the thesis is done, there is still much to learn, but at least I can say that I have learned much in many different fields and aspects.

What I have accomplished I could have never done without the people who supported, contributed and helped me along the way. My most sincere gratitude goes out to everyone who has helped me with the thesis.

Firstly, I would like to thank dr.ir. W. Roozing for his continued feedback and excellent guidance 'beyond the call of duty' during the thesis process. His experience has helped to guide the research more effectively and his critical thinking always provided useful feedback and helped to provide a different point of view. The weekly meetings have both been useful in the thesis in keeping track of what needs to be done, as well as providing a nice bit of social contact in these times when working from home is the norm.

Next, my thanks go out to the technical staff at the RAM department of the University of Twente, in particular ir. Te Riet O/G Scholten and ir. S. Smits, for their generous feedback and help in practical considerations of the construction and manufacturing. Their patience with my many requests for purchases, 3d prints and laser cut orders is astounding and much appreciated. Additionally my appreciation goes out to prof.dr.ir. L. Abelmann for his insight into the workings of magnets, help with simulations of said magnets and help with figuring out the cause of degradation of actuator performance over time.

As for my fellow students, I would like to thank R.J.H. Freije, Bsc for his time brainstorming bondgraph models, and R.N. Timmer, Msc for his feedback and suggestions with respect to mechanical systems. Due to this I have learned much more on the practical aspects of mechanical systems.

Lastly, of course I would like to thank my family and friends who have encouraged and supported me all the way, regardless of what happened.

This document was designed with double sided printing in mind to save the environment.

For this purpose this page is intentionally left blank

Contents

1	Introduction	7
2	Requirements	9
2.1	Project goals	9
2.2	Non-linear series elastic actuator requirements	9
2.2.1	Physical dimensions	10
2.2.2	Displacement, Torque and stiffness range	10
2.3	Control system requirements	12
3	Paper	13
4	Elastic element concepts and iterations	33
4.1	Concept exploration	33
4.1.1	Pseudo-non-linear modular elastic element	33
4.1.2	Compressible gas based	34
4.1.3	Rubber non-linearity based elastic element	35
4.1.4	Magnet based elastic element	36
4.1.5	Magnet based elastic element with linear spring	36
4.2	Design iterations of the magnet based series elastic element	37
4.2.1	Version 1 and 2	37
4.2.2	Version 3	37
4.2.3	Version 4	38
4.2.4	Version 5	38
4.2.5	Version 6	39
4.2.6	Version 7	40
5	Derivation of model and equations	43
5.1	IPM and Bondgraph	43
5.2	State-space deduction	45
5.3	Magnet parameter deduction	47
5.4	Magnet model derivation	49
5.5	Spring model derivation	50
6	Test setups	53
6.1	Static test setup	53
6.1.1	Requirements	53
6.1.2	Mechanical design	54
6.1.3	Sensors	55
6.1.4	Materials	57
6.2	Design iterations static test setup	59

6.2.1	Version 1	59
6.2.2	Version 2	59
6.2.3	Version 3	60
6.3	Dynamic test setup	62
6.3.1	Requirements	62
6.3.2	Mechanical design	62
6.3.3	Materials	62
6.3.4	Prototype iterations and overview	63
7	Conclusion	65
7.1	Non-linear elastic element	65
7.2	Gain-scheduled control system	65
7.3	Test benches	66
A	Literature analysis	71
A.1	Analysis of the non-linear series elastic actuator field	71
A.2	Non-linear series elastic actuators	71
B	Code	73
B.1	LQR Controller Matlab code	73
B.2	LQR parameter analysis Matlab code	77
B.3	Elastic element model Matlab code	80
B.4	Step response data processing Matlab code	83
B.5	Transparency data processing Matlab code	85
B.6	Torque tracking data processing Matlab code	87
B.7	Result processing Matlab code	90
B.8	Embedded code	92
B.8.1	Main.cpp	92
B.8.2	StrainGauge.h	93
B.8.3	StrainGauge.cpp	93

Chapter 1

Introduction

This thesis is presented as part of the final evaluation for a master degree in electrical engineering, with a specialization in robotics and mechatronics.

Robotic design has always wanted to make the interface between actuator and load as stiff as possible to maximize the position control bandwidth possible. This increase in performance is advantageous for position-control systems, but does not apply to all robotic systems. Sometimes a lower stiffness interface can provide benefits that are inherently not possible to achieve with a high stiffness link. Additionally, for some systems a very stiff interface is hard to achieve. These facts have fuelled research into so called 'Series Elastic Actuators' (SEAs) [2, 3]. By varying the stiffness, a trade-off between position control bandwidth at high-stiffness, and advantage such as increased safety and better force/torque control at lower-stiffness can be made [3–5, 14].

Various concepts have been proposed to overcome the trade-off, and provide both the benefits of a high-stiffness and a low-stiffness link between actuator and load. The first is so called Variable Stiffness Actuators (VSAs) [7–10]. These actively and dynamically change the stiffness of the mechanism with an additional actuator. Another approach are Non-linear series elastic actuators (NSEAs) [11–14]. These actuators are similar to regular series elastic actuators, except they replace the elastic element with a non-linear elastic element. These provide a high torque resolution at lower deflection, whilst providing higher maximum torque capabilities, higher torque control bandwidth and better actuator transparency than a linear series elastic element with similar deflection range.

Most existing NSEA designs use a mechanical way of generating non-linear stiffness [11–16]. This project has two main goals.

- 1) Present a novel way of generating the non-linear stiffness.
- 2) Evaluate the potential benefits of gain-scheduling controllers applied to NSEAs.

To achieve these goals a mathematical model is shown that accurately describes the performance of the presented concept. A review of main design parameters is made, and usage of the parameters in trade-offs and design decisions is evaluated. A physical proof-of-concept prototype for the non-linear stiffness element concept has been designed and manufactured to compare real-world performance against model

predictions. The focus of the thesis is thus on the design and analysis of the elastic element, not the full NSEA.

Series elastic actuators require some form of control to function. Most NSEA designs use 'traditional' linear controllers [11, 13, 15, 16]. This project also presents an investigation into whether gain scheduling controllers provide additional benefits for NSEAs compared to linear controllers. This investigation is done with simulations on the previously mentioned model.

The thesis is organized as follows. Firstly, the general goals and requirements of the project are given in section 2 as to give a concept of the direction of the design. It gives motivation into why certain requirements and design goals are pursued. This is followed by the core content of the research, which is presented in the paper under chapter 3. The chapters following the main paper can be seen as supporting chapters for the main paper. The paper glances over the mechanical design and iterations in chapter 4 & 6. They add more detail on the various designs and iterations of various prototypes and test benches. Chapter 5 presents further derivations of models and equations that are given in the paper. As the main research conclusions have already been presented in the core paper, chapter 7 reflects on the initial design requirements (as stated in chapter 2) and if these requirements have been met. These additional chapters are not directly relevant to the conclusions found in the paper, but can be seen as supporting and are relevant for future work, for the faculty and for evaluation of the thesis. If the reader desires to have more background information, additional details on existing literature are given in appendix A. This includes a more encompassing and visual overview of what previous work has been done. Code that is seen as particularly relevant to the research contributed in the paper (e.g. for calculating models or analysing collected data) is given in appendix B.

Chapter 2

Requirements

2.1 Project goals

As previously mentioned in the introduction, the main goals of the project are to:

- 1) Present a novel method of generating non-linear series elasticity
- 2) Present a gain-scheduling controller for non-linear series elastic actuators, and its potential advantages over linear controllers

The requirements are organized with their alignment to these goals. The MoSCoW approach is used to hierarchically rank the importance of the requirements, and quantifiable requirements are estimated to produce a final list of concrete functional requirements.

2.2 Non-linear series elastic actuator requirements

The first main goal of the thesis is to investigate a unique approach in stiffness generation. This includes a mathematical model and a physically realized prototype to demonstrate the concept working in practice. The design parameters of the model are assessed in how they impact actuator performance. The performance of the prototype will be compared to predicted performance by the model to evaluate model performance.

- Must have:
 - A novel way of producing non-linear elasticity for a non-linear series elastic actuator
 - A model of the proposed non-linear series elastic element
 - A model of a non-linear series elastic actuator using the proposed non-linear elasticity
 - A 3D CAD model of the proposed non-linear series elastic element
 - A comparison of performance between non-linear series elastic actuators and linear series elastic actuators
- Should have:
 - A produced physical prototype of the non-linear series elastic element

- * Should be easy to assemble
- * Should be able to be constructed using rapid prototyping techniques
- * Should have no dead-zone at no deflection
- * Should have a 'reasonable' range of torque and stiffness
- Rough performance measurements of the produces hardware design
- Analysis of key design parameters
- Could have:
 - Performance measurements of the produces elastic element integrated into a non-linear series elastic actuator together with the proposed gain-scheduled controller
 - An optimized controller running on the produced hardware
- Won't have:
 - A comparison of the produced prototype with more traditional (mechanically produced stiffness) non-linear series elastic actuators

Of these requirements, some are quantifiable requirements. These are further elaborated upon in the next sections to help establish a baseline.

2.2.1 Physical dimensions

The physical dimensions of the produced actuator can be a trade-off with the needed torque/stiffness range. Generally speaking a larger actuator can produce larger torques. Larger torques are also associated with higher stiffness. Because the proof-of-concept will be produced with rapid prototyping techniques such as a 3D printer, this puts limits on the maximum size of the elastic element. The specific 3D printer available is the *Markforged Mk II* [31]. This specific unit provides a build volume of 330mmx132mmx154mm (width x depth x height). For a circular design this would entail a maximum diameter of 132mm.

Generally the size of the actuator scales with the produced torque. There is a limit on the amount of torque that the actuator can handle. This thus presents a trade-off between size and torque range, which can be explored with mathematical models and iteration. For the requirements, a maximum thickness of 50mm is selected.

The weight of the actuator should be under 2 kg. Setting aside approximately 60% for the motor, gearbox and electronics leaves 800 gr for the elastic element.

2.2.2 Displacement, Torque and stiffness range

Arguably the maximum angular deflection (displacement), stiffness and torque are the most important parameters of the NSEA design. These three parameters are all are interrelated.

Torque and stiffness should be similar to existing NSEA designs, but considering it is a first proof-of-concept it does not need to exceed them. Various different presented designs of non-linear series elastic actuators are shown in table 2.2.2. These are however, of different physical dimensions. The design presented by Thorson

et.al. [12] is with 9 kg significantly physically larger than the proposed proof-of-concept. Malzahn et al. [11] and Austin et al. [13] both present concepts that are smaller than the maximum dimensions allow. The presented physical prototype will be somewhere between these actuator sizes.

Table 2.1: Comparison of performance other non-linear series elastic actuators.

Reference	Torque range [Nm]	Max. displacement [rad]	Stiff. range [$\frac{\text{Nm}}{\text{rad}}$]
[13]	+ - 2.5	0.7	1.7 to 3.4 (2x)
[11]	+ - 4.2	0.26	?
[12]	+ - 40	1.05	?

Extrapolating using the information that the proposed concept will be larger than the two smaller actuators presented in previous torque range, a torque range of +-10Nm is reasonable. This would be a high enough torque to be usable, but not so large that the higher forces associated with the design would inhibit rapid prototyping techniques.

Most designs do not present any information on the stiffness range. Only the design by Austin et al. [13] presents a stiffness range, with a dynamic range of approximately 2x. However, this is not all that should be considered for the stiffness requirement. The requirements lists 'no-deadzone'. Dead-zone would entail a low minimum stiffness. The low stiffness at low deflection levels would feel as if there is play in the mechanism. This would result in high actuator efforts with low deflection. Thus a minimum stiffness of 25Nm is chosen. A dynamic range of at least 2x is chosen to be aimed for.

A lower deflection is beneficial for higher torques. A trade-off, is that force measurements are of lower resolution. As a general design range, 0.07-0.26 radians of deflection to a single direction is aimed for. The device should be able to deflect equally to both clockwise and counter-clockwise deflection.

This results in the following quantifiable requirements:

Table 2.2: Quantifiable requirements for the series elastic element.

Parameter	Amount
Displacement [rad]	0.08 - 0.27
Torque range [Nm]	+ -10
Minimum stiffness [Nm/rad]	25
Stiffness dynamic range [x]	2
Maximum diameter [mm]	132
Maximum thickness [mm]	50
Maximum weight [gr]	800

2.3 Control system requirements

The second main goal is investigating the properties of a gain-scheduling controller on a non-linear stiffness actuator. This is compared to the performance of a linear controller, to see if gain-scheduling versus the changing plant conditions (stiffness versus deflection) will provide a benefit in operation.

- Must have:
 - A design of a gain-scheduled controller appropriate for non-linear series elastic elements
 - A comparison of simulated results of the proposed gain-scheduled controller with a traditional linear controller
- Should have:
 - Analysis of key design parameters
- Could have:
 - Performance measurements of the produces hardware design with the proposed gain-scheduled controller
 - An optimized controller running on the produced hardware

As there has been no previous research into non-linear gain scheduling controllers for non-linear series elastic actuators, the absolute performance is not as important. The produced controller will be compared a linear controller as to deduce whether the more complex controller offers any advantages over traditional linear controllers.

Chapter 3

Paper

The following pages show the core of the research, presented as a separate paper which will later possibly be sent for publication.

This document was designed with double sided printing in mind to save the environment.

For this purpose this page is intentionally left blank

Design and Development of a Magnetic Non-linear Elastic Element and Control for Progressive Series Actuation

Boi Okken, BSc., *b.okken@student.utwente.nl*, *s1441833*, *Electrical Engineering, dept. RAM*

Abstract—Traditional robotics use stiff actuators, which are inherently unsafe to work with. Series elastic actuators were developed to deal with this issue and show that an increase in stiffness and thus torque control bandwidth results in a reduction in safety metrics such as actuator transparency. Non-linear series elastic elements are investigated to tackle the compromise, offering transparency, safety and torque resolution at low deflection and torque, with the higher torque control bandwidth at high deflection and torque. A novel way of generating non-linear stiffness based on magnets is presented and implemented into a non-linear series elastic element. For this elastic element design parameters and trade-offs have been identified. Additionally a gain-scheduled control system is proposed to investigate potential improvements over traditional linear control and to further exploit the advantages a non-linear stiffness actuator.

Keywords—*Design and development, Gain-scheduling control, LQR-control, Non-linear stiffness actuator.*

I. INTRODUCTION

TRADITIONAL actuator design methods focus on making the interface between motor and load as stiff as possible. This has been the norm for actuator design until the introduction of Series Elastic Actuators (SEAs). Although traditional actuators exhibit a large position control bandwidth, they pose issues concerning safety when operating with humans [1] or when operating in unpredictable and unconstrained environments. Collisions present forces that have large high frequency content which the control system cannot compensate for due to the inherently limited bandwidth. Furthermore, an approximation of an ideal stiff interface can be physically impractical, hard and in some cases even impossible to achieve.

A. Series elasticity

Series elastic actuators [2], [3] try to address these issues. This type of actuator employs an elastic element in series with the actuator, reducing the output impedance and thus increasing safety. The elastic element decouples the reflected motor inertia from the output, and reduces the open-loop system bandwidth. By adding this elastic element, they provide the designer with a trade-off between actuator transparency and torque control bandwidth [3]–[5], [14]. Additionally, they provide a convenient implementation for force control by measuring the elastic deformation. This turns the force control problem into a position control problem with the use of Hooke's law. Lastly, the series elastic element can provide an energy storage reservoir which can increase

the efficiency of periodic movements such as walking gaits [6].

Series elastic actuators present a trade-off between safety and achievable closed-loop bandwidth, and research into overcoming the trade-off has been focussed on changing the stiffness of the elastic element. Having an adjustable stiffness attempts to overcome the compromise by offering both the safety and torque resolution of low-stiffness elements, whilst also providing a high bandwidth like high-stiffness elements. Research has yielded two primary implementations, so called Variable Stiffness Actuators, and Non-linear Series Elastic Actuators.

1) *Variable Stiffness Actuators*: Firstly, the concept of so called Variable Stiffness Actuators, or VSAs [7]–[10], tries to deal with the inherent trade-off by using a secondary actuator to dynamically change the stiffness as required. This comes at the cost of increased mechanical complexity, and the solutions are generally relatively bulky.

2) *Non-linear Series Elastic Actuators*: The second approach makes the elastic element itself passively non-linear. These types of actuators are called Non-linear Series Elastic Actuators (NSEAs) [11]–[14]. By increasing the stiffness with deflection (progressive non-linear stiffness) they can provide a high torque resolution and higher safety with low deflection, whilst offering a higher bandwidth than a traditional series elastic element with similar deflection and torque resolution.

Non-linear Series Elastic Actuators can be divided into two main categories. These are 1) mechanical based and 2) material based. The former deflects either a linear series elastic element through a non-linear cam system [12]–[16] or uses structure controlled stiffness [11] to generate the non-linearity. Conversely, material based NSEAs use the inherent non-linear property of a material to generate a non-linear stiffness. Some research has been done with non-linear materials such as rubber [13], however none have done so with the intent to solely use the inherent material properties to achieve elastic element non-linearity.

For any non-linear stiffness implementation, stiffness profile selection is important. Stiffness profiles can be categorized in one of three categories. 1) Degressive 2) Linear 3) Progressive. For a profile to be non-linear, it would need to be either degenerative or progressive. The former implies that the further the elastic element is deflected, the lower the stiffness becomes. The latter implies the opposite: the further the element deflects, the stiffer the elastic element is. Taking the

previously discussed trade-off between actuator transparency and torque control bandwidth into account, it makes logical sense that a progressive profile provides the best of both worlds. With smaller forces (and thus lower deflections) a higher transparency is usually desired. A progressive stiffness profile is able to deliver the same transparency with an increase in maximum torque the actuator can handle.

B. Controllers in Series Elastic Actuation

Non-linear series elastic actuators require a controller for operation. Most existing literature either specify no specific controller, or use traditional fixed gain linear controllers such as PID for control [11], [15], [16]. Also in the case where more complicated state-observer based controllers have been implemented, the controller itself is generally still linear [13]. If these controllers are used with a non-linear elastic element this could possibly lead to not achieving the expected controller performance over the entire stiffness operating range. Tuning the linear controller for one stiffness would sacrifice bandwidth over the entire torque range. NSEAs can present a substantial dynamic range in terms of stiffness, and thus might not achieve optimal performance achievable if driven with a linear controller.

There have been attempts in improving control systems for non-linear series elastic actuators. Research by Axelsson et. al. [17] approaches the change in stiffness as a plant uncertainty, and uses $H-\infty$ control. The main drawback of the design presented is, that it ensures performance in a particular stiffness band, but not outside of it. It is assumed that the elastic element spends most of its time in this region, whilst outside this region stability is ensured, however, performance requirements are not. This thus limits the flexibility of robotic designs that this methodology can be applied to.

One of the proposed solutions for the control of non-linear systems is gain-scheduling. This approach actively adapts the control system parameters based on one or more scheduling variables that indicate the state of the plant. It can be applied to a variety of different control structures. LQR control provides an intuitive way of tuning an optimal controller in a lot of systems where system states are known or estimated. Gain scheduling has been applied to LQR controllers [18], but has yet to be applied to non-linear series elastic actuators.

C. Scope and contributions

This work is primarily focused on the design and construction of a novel magnetic-based non-linear elastic element for use in non-linear series actuators. This includes the analysis and exploration of different implementation concepts with analytical models for the torque and stiffness profiles. A parametric exploration is given for guidance in the practical design of a magnet-based NSEA. The usage of the guidelines and models are demonstrated in the practical design of a box magnet based prototype. This prototype is characterized and the empirical data is used to validate the concept and compared against the model to check model

validity.

To overcome the limitations of traditional linear control, a gain-scheduled LQR controller which uses stiffness as the scheduling variable is presented and evaluated for performance gains compared to traditional linear controllers.

The main contributions can be summarized as follows:

- The presentation of the novel idea of magnetic-based stiffness for NSEAs, including mathematical model and parameter exploration.
- Successful practical implementation a magnet-based series elastic element with progressive stiffness characteristics.
- Characterization of said magnet-based series elastic element, and verification of the mathematical model.
- Proof that traditional linear controllers have sub-optimal performance when applied to NSEAs.
- The presentation of a solution in the form of gain-scheduling a linear quadratic regulator, which provides consistent and predictable performance over the entire stiffness range.
- Simulated results showing that NSEAs combine the advantage of high transparency and torque resolution at lower torques, and higher torque tracking bandwidth at higher torques.

The paper is organized as follows. Firstly section II will focus on the analysis and design of the magnetic non-linear stiffness element. This is followed by section III which will focus on the practical implementation of such a magnet-based NSEA and provide empirical data of the constructed prototype, which is used to validate model performance. Section IV then proceeds to show the concept and design of the gain scheduling LQR controller. Section V presents the simulated closed-loop performance of the controller and elastic element combination, and compares this against a variety of combinations of linear controllers and linear plants. Finally sections VI, VII, VIII and IX present a discussion on the results, a conclusion, highlight possible areas of future work and give concluding remarks.

II. CONCEPT AND ANALYSIS

A. Principle of operation

For a non-linear series elastic actuator, the elastic element is connected between the load and the power delivery actuator. The non-linear stiffness in the presented device is generated with the use of magnets. This is possible due to the inherent non-linear repellent force in magnets, which is demonstrated in Fig.1. By orienting magnets such that the same poles are facing each other (i.e. north facing north, and south facing south), the force imposed on each magnet increases progressively as distance between the magnets decreases.

Because of the rotational movement of typical electrical motors, the non-linear elastic element functions in the rotational domain as well. This implies a radial orientation of the magnets such that a rotational movement moves them physically closer or further apart as shown in Fig. 2. It shows that the device

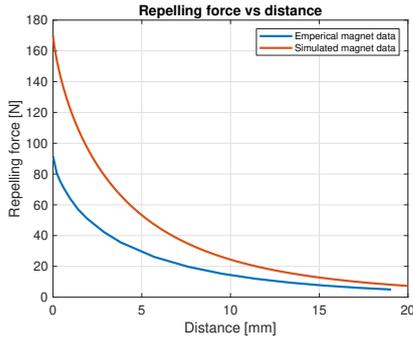


Fig. 1: Magnet Repelling force for a magnet 30x12x12 [mm], N52 grade. Emperical and simulated data.

consists of two separate halves that fit together, an inner hub and an outer hub. α represents the 'free travel' in one direction that the two halves of the actuator can move with respect to each other, β is the space that an individual magnet segment can occupy, and N is the total number of 'arms' that one of the two halves has. It should be noted that $2N$ denotes the number of opposing magnet pairs. The combination of β and N together determines the total volume inside the elastic element occupied by the magnets. This relationship is also shown in Fig.3.

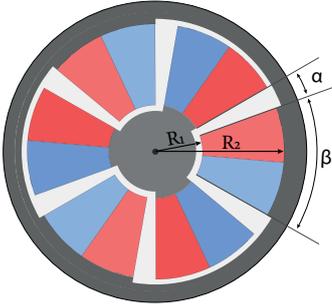


Fig. 2: Top down overview of the series elastic element with arc-segment magnets, in this illustration $N=3$.

B. Parameter exploration

1) *Free travel and magnet area:* An increased volume available for the magnets to occupy increases field strength, and thus torques produced by the elastic element. However, this cannot be done freely as there is a limited amount of volume available within the elastic element for a given outer radius. This forms a trade-off between the volume that can be occupied by the magnets, and the amount of free travel the actuator has. Eq.1 describes how these design parameters influence each other.

$$\beta = \frac{\pi}{N} - \alpha \quad (1)$$

Plotting Eq.1 for various N yields Fig.3, which can be helpful in determining the possible magnet area for a required free travel α . For a selected free travel, a vertical line can be drawn which intersects with the lines of different values of N . As the ideal size of β is dependent on the available magnets, iterating on the choice of N and β is required to find a combination that satisfies the design criteria.

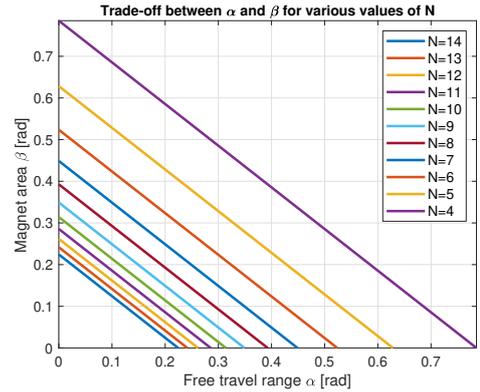


Fig. 3: Trade-off between α and β for various values of N .

It should be noted that adjusting the variables in Eq.1 also changes other parameters of the design. Increasing the amount of magnet pairs whilst keeping α constant has a similar effect if the magnet area β is used fully. If β or N is increased, sacrificing free travel range, an increase of maximum torque of the actuator can be observed. An analytic model of this torque and its associated stiffness is derived under Sec. II-C. This model shows the trade-off relationship between β/N and α , visually demonstrated in Fig.4. It shows that an increase in N (effectively increasing the area of the actuator occupied by magnets) increases both maximum torque, maximum stiffness and base-stiffness at the cost of a reduced free travel.

A higher maximum stiffness for an identical or smaller free travel also results in a higher base stiffness. A higher magnetic field when the magnets are touching also implies a higher magnetic field at rest state (with the assumption that the distance between magnets at rest state does not decrease). When looking at a typical magnet repelling force curve (For example Fig. 1), the higher the absolute repelling forces at a given distance, the steeper the slope, and thus the higher perceived stiffness that would result. This shows that if the magnets are closer together in rest state (i.e. the element has smaller free travel), the stiffness at rest state is also higher. This is also apparent in Fig.4.

2) *Actuator radius:* Increase in radius of the elastic element results in both a higher torque and base-stiffness. This is because of two principles, firstly the increase in magnet area

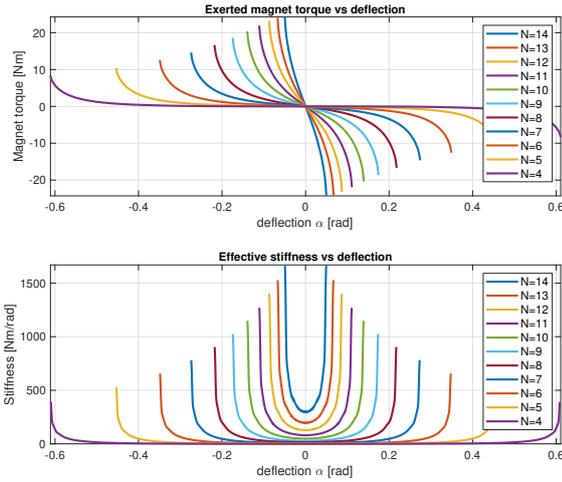


Fig. 4: Effect of N on the free-travel α , torque and stiffness. Magnet size is $30 \times 2.6 \times 12 \text{ mm}$, $\beta = 10^\circ$, $R_2 = 60 \text{ mm}$.

that can be occupied scales progressively with radius and secondly a higher radius results in a higher translated torque from a given repelling force. Therefore changing the radius of the actuator has a comparatively large impact on torque and stiffness effects of the actuator.

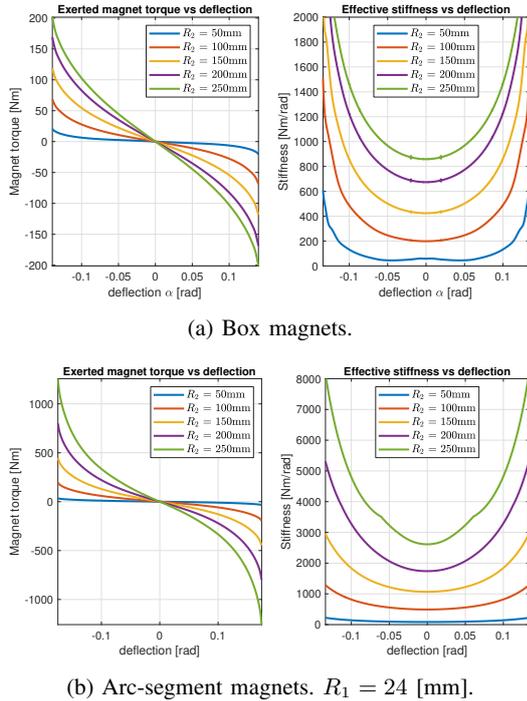


Fig. 5: Effect of R_2 on torque and stiffness. $\beta = 10^\circ$, $N = 10$.

This can be seen in Fig. 5a and 5b. Note that for a

TABLE I: Box magnets and results used in Fig. 5a

Radius [mm]	M. vol. [mm^3] (l x w x d)	Torque [Nm]	Base stiff. [$\frac{\text{Nm}}{\text{rad}}$]	Torque dens. [$\frac{\text{Nm}}{\text{rad}\cdot\text{mm}^3}$]	Stiff. dens. [$\frac{\text{Nm}}{\text{rad}\cdot\text{mm}^3}$]
50	$25 \times 2.18 \times 12$	± 20	58	2.12×10^{-4}	6.16×10^{-4}
100	$50 \times 4.37 \times 12$	± 68	198	1.80×10^{-4}	5.25×10^{-4}
150	$75 \times 6.56 \times 12$	± 118	424	1.39×10^{-4}	5.00×10^{-4}
200	$100 \times 8.75 \times 12$	± 168	674	1.11×10^{-4}	4.47×10^{-4}
250	$125 \times 10.94 \times 12$	± 201	856	8.54×10^{-5}	3.63×10^{-4}

TABLE II: Arc magnets and results used in Fig. 5b

Radius [mm]	M. vol. [mm^3]	Torque [Nm]	Base stiff. [$\frac{\text{Nm}}{\text{rad}}$]	Torque dens. [$\frac{\text{Nm}}{\text{rad}\cdot\text{mm}^3}$]	Stiff. dens. [$\frac{\text{Nm}}{\text{rad}\cdot\text{mm}^3}$]
50	163	± 33	85	1.72×10^{-2}	4.34×10^{-2}
100	818	± 194	491	1.98×10^{-2}	5.00×10^{-2}
150	1.9×10^3	± 446	1.0×10^3	1.95×10^{-2}	4.65×10^{-2}
200	3.4×10^3	± 800	1.7×10^3	1.94×10^{-2}	4.22×10^{-2}
250	5.4×10^3	$\pm 1.2 \times 10^3$	2.6×10^3	1.94×10^{-2}	4.03×10^{-2}

change in radius, different magnets are needed. For Fig. 5a box magnets are used which are calculated for optimum area occupation as described under Sec.II-B3. Fig. 5b uses arc-segment magnets which have a constant inner radius of 24 [mm], these types of magnets make more efficient use of area available. The magnets used per radius and their resulting approximate torque and stiffness range are shown in Tab. I and II for box and arc-segment magnets respectively. Also shown are the overall elastic element torque density and stiffness density with the given radius of the actuator and a 12mm thickness. The larger the radius, the lower the torque and stiffness density when using box magnets. For arc-segment magnets this area is used fully, and thus the torque and stiffness density vary little with change in outer radius.

Conversely, increasing the thickness of the actuator linearly increases the volume of the magnets and thus also the torque generated. This relationship makes it relatively simple to scale a design if a different torque and stiffness are desired. If a certain stiffness and torque curve are achieved up to a scaling factor A , the thickness of the magnets can be scaled by the same factor A to achieve the desired result.

3) *Magnet type:* For implementation 'Circumferentially polarized Arc Segment Magnets' (example Fig. 2) offer an ideal volume usage for a given arc (β) and radius they fit in. They are characterised by a given inner radius and outer radius. Although arc segment magnets offer an optimal space utilization, they are more expensive and are generally not available as off-the-shelf components. Therefore regular 'box' type magnets are also considered for construction, as shown in Fig.6. The device will still exhibit non-linear stiffness behaviour to demonstrate the concept. Exchanging the box type magnets for arc-segment type magnets will yield a higher total output torque range, as the principle of operation remains unchanged. This will be shown in Sec.II-C, where both the box and arc-segment type magnets implemented in similar actuator dimensions are analysed.

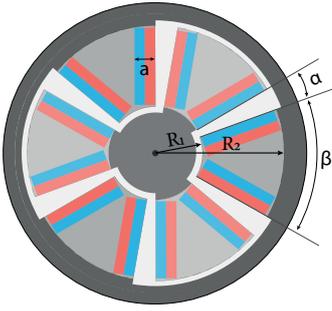


Fig. 6: Top down overview of the series elastic element with box segment magnets.

Because box-magnets do not fully occupy β as arc-segment magnets do, additional parameters are needed for their dimensions and placement. What needs to be considered is how the magnets can optimally be placed in the space of an arc that can be occupied by the magnets. This problem is approached assuming a given maximum radius that can be used. Fig. 7 shows a top down overview of how box magnets are oriented inside a β segment for optimal coverage.

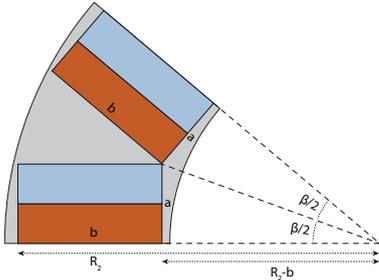


Fig. 7: Illustration showing the orientation of box magnets.

Given a maximum outer radius of R_2 , and a magnet width a , the inner radius for maximum surface utilization of the box magnets is given by Eq.2. Design iteration with available magnet sizes can be used to maximize the surface area $A = ab$ occupied by the magnets, thus maximizing the possible maximum torque for a given maximum radius and β available for magnets to occupy. The theoretical maximum surface area (and thus volume) for a given β and R_2 is at a magnet length of $b = R_2 - R_1 = \frac{R_2}{2}$.

$$R_1 = \frac{a}{\tan \frac{\beta}{2}} \quad (2)$$

When using box sized magnets, it is important to note that the individual box magnets within a β -wide arc section should be of similar magnetic polarity (i.e. both north poles facing the same tangential direction). This ensures that the magnets are

held in place by their attractive forces during assembly, and do not require any adhesive or mechanical fasteners to keep them in place.

C. Elastic element model

Traditionally analysing the repelling force of magnets is analytically impossible and computationally expensive. The magnet strength can be approximated by using empirical data available from magnet manufacturers and resellers (for example Fig.1), or experimentally obtained magnet data. This data can be obtained by measuring the forces between two parallel pairs of magnets with a varying linear distance. Fig. 1 also shows simulated magnet repelling force using the calculator tool in the *MacMMems1.3* software suite [24]. It is chosen to use the empirical data to provide the most accurate result for modelling a real-world elastic element.

Using the repelling force versus distance, the torque curve can be derived. It should be noted that the linear distance between the surfaces of a pair of magnets changes with both the deflection angle α as well as the radius at which the distance is measured. For the same angle between magnets, a point closest to the center 'sees' the opposing magnet closer than a point further away from the radial center. To account for this effect in the model, the magnet is divided into infinitesimally small slices along the radial direction (Fig.8). The surface area of this small slice is calculated and normalized against the total surface area to produce a scaling factor. This compensates for change of surface area over radius for arc segment magnets.

Each magnet segment is then treated as the scaled version of the original magnet in terms of force. The distance between the opposing magnet segments is calculated, resulting in a force (and because the radius is known, torque) for a given angle. The resulting set of torque functions of the segments are summed for the total torque per magnet pair, and multiplied for the amount of opposing magnet pairs in the actuator for the final actuator torque at a deflection angle α .

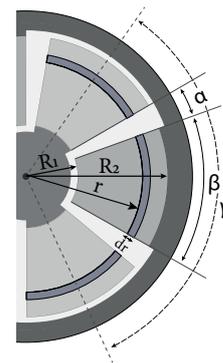


Fig. 8: Associated magnetic elastic element definitions.

For both box and arc-segment type magnets respectively the output torque at a given deflection angle α can be written as:

$$\tau_{box}(\alpha) = \frac{N}{R_2 - R_1} \int_{r=R_1}^{R_2} r F_{lin,box}(2r \sin(\frac{\alpha}{2})) dr \quad (3)$$

$$\tau_{arc}(\alpha) = \frac{4N}{R_2^2 - R_1^2} \int_{r=R_1}^{R_2} r^2 F_{lin,arc}(2r \sin(\frac{\alpha}{2})) dr \quad (4)$$

For both cases the stiffness is computed with the derivative with respect to the deflection angle:

$$k(\alpha) = \frac{d\tau(\alpha)}{d\alpha} \quad (5)$$

Eq. 3 and 4 can be used to estimate the torque produced by an elastic element design, by using empirically derived repelling (linear) force data (as in Sec. II-A). An appropriate stiffness and torque can be found by design iteration with varying parameters and available magnets.

Outer radius and thickness of the actuator can be physical constraints in the requirements, the inner radius is mostly determined by additional structure to connect the two halves. These, combined with the chosen free travel and number of magnet pairs determine the achievable size of the magnets.

For the proof-of-concept prototype, a torque range of ± 10 [Nm] and a minimum stiffness of at least 20 $[\frac{Nm}{rad}]$ are desired. A deflection angle of $\alpha = 10^\circ$ is chosen as a compromise to achieve these objectives and still provide a reasonable deflection for torque measurement. This results in $\beta = 50^\circ$ with $N=3$.

Design iteration with available magnets and Eq.2 yields a chosen box magnet size of 30mm x 12mm x 12mm. Usage of these magnets results in an inner radius of 30mm and an outer radius of 60mm. The magnet grade is the highest common commercially available, N52. Using online available magnet data [21] and scaling this to the quoted maximum repelling strength of the manufacturer yields the magnet-repelling curve shown in Fig.1 and the box-magnet torque and stiffness curves shown in Fig. 9.

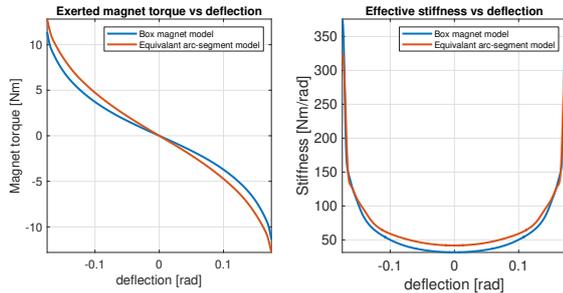


Fig. 9: Model prediction of the torque and stiffness of the elastic element with box magnets of size 30x12x12mm and approximately equivalent arc-segment model with inner radius of 24mm and an outer radius of 40mm.

Fig. 10 shows the torque and stiffness profiles of an arc-segment based elastic element with identical physical parameters. As arc-segment magnet data is not common, the repelling force is estimated by taking the repelling force data of a box magnet of similar volume to said arc-magnets. Some iteration has found that an equivalent arc-segment magnet based elastic element (in terms of performance) would need to be approximately 40mm in diameter, a significant reduction of radius and thus also volume compared to the box-magnet element.

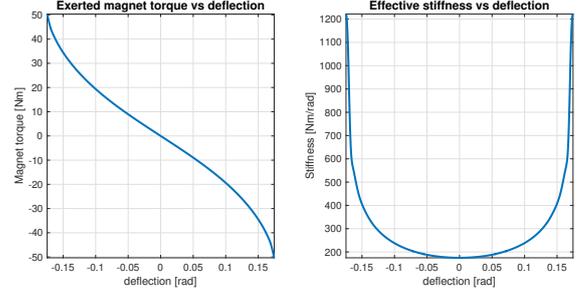


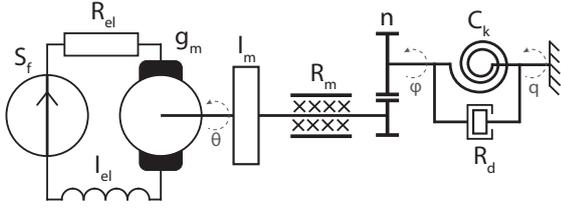
Fig. 10: Model prediction of the torque and stiffness of the elastic element with arc-segment magnets. Note that contrary to Fig.9 the outer radius is similar to the box magnet element at 63mm.

The model indeed predicts a non-linear stiffness for both types of magnets, and that an elastic element with arc-segment magnets exhibit a higher maximum torque than one with box magnets. The torque and stiffness of the arc-segment non-linear elastic unit are higher by a factor of approximately 4.5x and 3x respectively.

D. Complete actuator model

An ideal physical model (IPM) of a non-linear series elastic actuator is shown in figure 11. For testing the actuator, the output of the actuator is considered to be fixed. The motor is driven with the use of a current source S_f . The torque constant of the motor is denoted as g_m . The output of the motor shaft is represented by the inertia I_m , and the motor and gearbox friction are combined into a single friction element R_m . This is coupled through the gearbox with ratio n to the non-linear spring element $C_k(\alpha)$ which also has parasitic friction forces R_d . Due to the magnet-based stiffness these friction forces are considered negligible. Note that the (electrical) motor resistance and inductance, R_{el} and I_{el} respectively, are included in the model for completeness sake, but are not relevant for the current-to-torque transfer function. These elements can be relevant in simulation to determine the minimum voltage required to feed the low-level current controller. They can also be used for power consumption and heating of the motor.

As the goal is to provide output (spring) torque control $\tau_{C_k} = \tau_{C_k}(\phi - q)$ for a given current input introduced at S_f defined as function U , the appropriate torque-to-current transfer function is deduced into a state space system. This can be done with any desired method. For this, a linearised


 Fig. 11: IPM of the actuator with the output fixed ($q = 0$).

model needs to be used with thus a fixed stiffness. The state space system shown in Eq. 7 and 8 results from choosing the state matrix as:

$$\bar{X} = \begin{bmatrix} \tau_{C_k} \\ \frac{d\tau_{C_k}}{dt} \end{bmatrix} \quad (6)$$

The states can be directly derived from the non-linear elastic element deflection and change of this deflection in time. By choosing the states as such, they can be measured directly from the system in a physical prototype facilitating easy implementation of a full-state feedback system.

$$s\bar{X} = \begin{bmatrix} 0 & 1 \\ -\frac{n^2}{I_m C_k} & -\frac{R_m + n^2 R_d}{I_m} \end{bmatrix} \bar{X} + \begin{bmatrix} 0 \\ \frac{g_m n}{I_m C_k} \end{bmatrix} U(s) \quad (7)$$

$$Y(s) = [1 \quad 0] \bar{X} \quad (8)$$

When modelling the actuator in software, the spring stiffness is implemented as a polynomial, which is derived from the torque model (Eq.3 - 4) or from a polynomial fit on empirical data from a constructed actuator (see Sec.III-D). This can be written as $\tau_{C_k} = p(\Delta)\Delta$ where Δ is the deflection experienced by the element. Care should be taken that the polynomial does not only match the torque, but that its derivative also adequately follows the stiffness curve as stiffness is defined as the change of torque over deflection. This can be written in non-linear state space form, choosing deflection and its derivative with respect to time at the states.

$$\bar{Q} = \begin{bmatrix} \Delta \\ \frac{d\Delta}{dt} \end{bmatrix} \quad (9)$$

$$s\bar{Q} = \begin{bmatrix} 0 & 1 \\ -\frac{n^2}{I_m p(\Delta)} & -\frac{R_m + n^2 R_d}{I_m} \end{bmatrix} \bar{Q} + \begin{bmatrix} 0 \\ \frac{g_m}{I_m} \end{bmatrix} U(s) \quad (10)$$

$$Y(s) = [p(\Delta)n \quad 0] \bar{Q} \quad (11)$$

For modelling the non-linear stiffness, with the data in Fig. 9 polynomial $p(\Delta)$ is constructed (Eq. 12) for initial testing and tuning of the controller. This is a 11th degree polynomial, describing a non-linear version of Hooke's law for the spring element in the model.

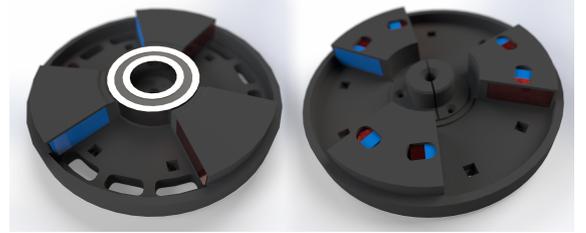
$$\begin{aligned} \tau(\alpha) = p(\Delta)\alpha = & -7.778 \times 10^9 \alpha^{11} + 4.541 \times 10^{-5} \alpha^{10} \\ & + 5.201 \times 10^8 \alpha^9 - 3.068 \times 10^{-6} \alpha^8 \\ & - 1.298 \times 10^7 \alpha^7 + 7.205 \times 10^{-8} \alpha^6 \\ & + 1.320 \times 10^5 \alpha^5 - 6.967 \times 10^{-10} \alpha^4 \\ & - 1.069 \times 10^3 \alpha^3 + 2.603 \times 10^{-12} \alpha^2 \\ & - 3.0963 \times 10^1 \alpha + 0 \end{aligned}$$

III. NON-LINEAR ELASTIC ELEMENT PROTOTYPE

A. CAD and prototype

For quick iterations, rapid prototyping techniques are to be used for the construction of the prototype. The inside of the magnetic elastic element is shown in Fig.12a and 12b, for the CAD model and the realised elastic element prototype respectively.

Figure 13 shows a section view of the elastic element fitting together. For illustration the upper half of the element is coloured in black, the lower in white. This shows how the two elastic element halves fit together, including the position of the bearings. The two halves are held together with a bolt inserted through the top half, which screws into an embedded nut in the lower half.



(a) CAD design.



(b) 3D printed proof-of-concept prototype.

Fig. 12: View of the two separate halves of the actuator. The left image shows the upper half, the right image the lower half.

There are witness holes in the upper half of the actuator, so that operation can be viewed whilst the device is assembled and operating. Above the magnets a cutout is placed to facilitate placing and removing the magnets. Finally in the center is a flush fitting flange bearing. Notice that the central hub on the lower half has a split in the middle, to create a friction fit with the main hub bearing. Inside the central hub there is

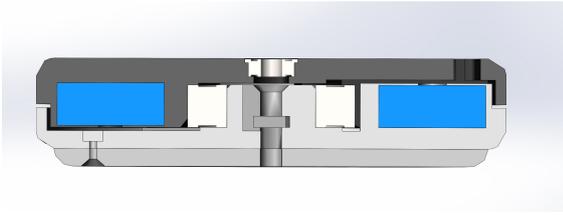


Fig. 13: A section view showing how the halves fit together, and the clearance between halves.

also an embedded nut in which a nut screws to clamp the assembly of the two halves together. Both sides have locations for embedded M3 nuts that are fixed with an epoxy. These holes can be used to attach the load and actuator to the elastic element. There is a rim on both parts for dust protection, additional stiffness and to prevent the halves from warping during the printing process.

B. Practical considerations

The usage of rapid prototyping techniques such as 3D printing and laser-cutting parts was selected to be used in construction, this simplifies construction and improves the ability to iterate quickly.

3D printing allows for the construction of relatively complex parts in place, rendering it ideal for the main elastic element due to its complex structure. It significantly speeds up proof-of-concept design whilst reducing cost. It would be mechanically more complex and costly in both time and money to produce the CAD design in more traditional construction techniques (ex. CNC machining). It should be noted that the presented magnet-based non-linear elastic element is less mechanically complex than traditional NSEA and VSA designs and thus relatively simpler and cheaper to produce.

Using 3D printing also necessitates the selection of a proper material for the designed device. Out of the various materials that are available, carbon fiber filled Nylon was selected to be used for its combined mechanical toughness and being able to resist the peak torques in the design.

C. Electronics

Static testing is done using a load-cell. The *HTC-Sensor TAL220* 10kg straight bar load-cell [25] is selected to be used. It provides sufficient range to measure the peak torque of the actuator.

The *HX711* integrated circuit takes care of the amplifier and analog-to-digital conversion. These are available in convenient prototyping boards for ease of implementation [26]. The interface for the *HX711* is a serial digital interface, which can be read out with a variety of different electronics.

The load-cell measurements are calibrated using an *ATI Industrial Mini-40* force-torque sensor [23] as a ground-truth

reference. For the load-cells selected, calibration is done with a zero-offset, and a calibration curve.

To read out the measurements of the *HX711*, and to transport this information to a personal computer for further analysis, a microcontroller is used. Most common microcontrollers are sufficient for the task, the only requirement being the presence of an interface port that a computer can read (e.g. Serial port or USB) and at least two digital pins to interface with the *HX711*. The *NXP* (Formerly *Freescale*) *Frdm K64F* is selected for the flexible web based IDE. It also has an integrated serial to USB converter so no additional electronics are needed besides the development board.

D. Static characterisation

The deflection-torque characteristics of the realised prototype are found by performing static tests. The deflection-stiffness characteristics can be derived by taking the derivative with respect to angle of this data. These results are subsequently validated against the analytical model derived in Sec. II-D.

The found deflection-torque curve can be implemented into the model, allowing for more accurate control loop tuning. Additionally, this curve can be used to approximate the torque in the spring from a given deflection, which provides a cheap way to practically measure torque in the element for implementation.



Fig. 14: CAD design and the associated manufactured unit of the static test setup.

The test setup fixes the elastic element on one side to a frame (optionally through the *ATI Mini-40* force/torque sensor) and allows the other side to be deflected through a pointer, which can be fixed in 1 degree increments. As discussed previously the main sensor to read out is a load-cell, which is attached to the pointer. With the pointer fixed at one end, the bending moment measured by the load-cell is proportional to the torque produced by the actuator. The designed and manufactured unit can be seen in figure 14.

E. Results

Fig. 15 shows the torque and deduced stiffness results from the produced prototype elastic element, as well as the polynomial derived from it (in red). Also, the torque and stiffness results predicted by the model are shown (in

yellow). This figure makes it apparent how the modelled results compare against the measured results from the proof-of-concept prototype.

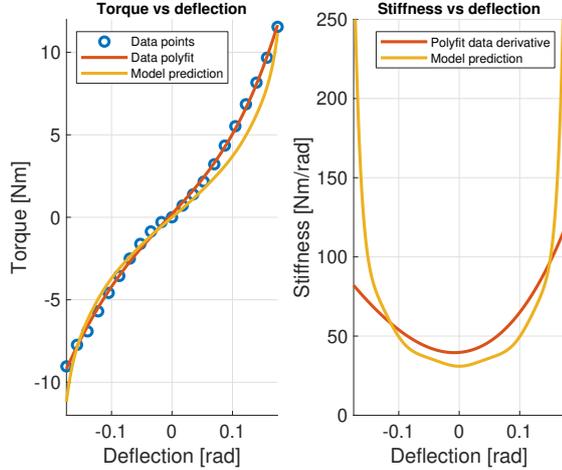


Fig. 15: Results from the static test.

With these results it can be observed that the practical performance adheres relatively closely to model predictions, especially for the positive deflection direction. The desired torque range of ± 10 [Nm] is achieved. The data is however slightly less non-linear than the model, resulting in a stiffness varying from 40 to 120 $\frac{\text{Nm}}{\text{rad}}$.

A peculiarity that should be highlighted is that during repeated measurements it was found that stiffness non-linearity decreased. This is shown in figure 16. This is further discussed in Sec. VI.

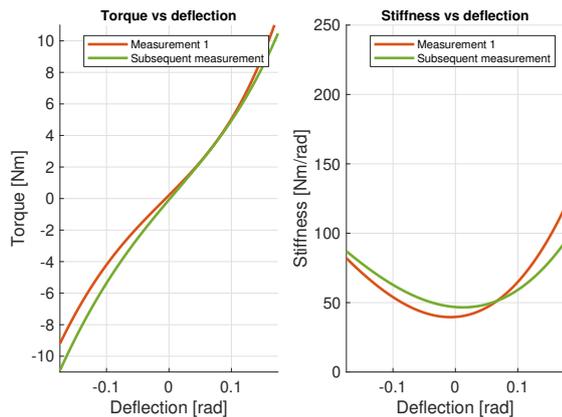


Fig. 16: Data shown in Fig.15 and a subsequent test.

IV. CONTROL

A. Concept

Linear control system design is often done by linearising a possibly very complex system around an operating point. For small variations around this operating point, and if the plant does not have a strong non-linearity, this approximation holds. However, this approach can be insufficient for some systems, and is unable to guarantee control system performance and stability margins over a wider operating range. One of the techniques to tackle this issue is gain-scheduling. Gain-scheduling entails that a linear controllers operating point is parametrized with the use of one or more scheduling variables. The linear control system gains are changed with the use of these scheduling variables which change with plant condition. This increases performance for non-linear plants and enables wider operating regions. It is a wide-spread and effective approach of overcoming the limitations of linear controllers whilst at the same time being able to exploit linear control design methods.

Most plant variations are unknown and/or non-trivial to measure. However, the main source of non-linearity in a NSEA as compared to a regular SEA, is the non-linear stiffness. The stiffness can be deduced without additional sensors if a torque control is desired, due to the presence of a deflection sensor. By choosing the stiffness as the scheduling variable, and dynamically changing the feedback gains of the quadratic regulator with this, the performance can be guaranteed at different operating conditions. In this work we will present a gain-scheduled quadratic regulator with full-state feedback.

The gains in a quadratic regulator can be derived in various ways. Most textbooks provide theoretical detail into how the gains for an LQR can be determined, and various computer programs can be used to quickly derive appropriate gains for a certain system as well. For this project *Mathworks Matlab* has been used to derive the control gains of a linear quadratic regulator. This can be done with the built in '*lqr*' command within the *Control System Toolbox* [22].

B. Linear Controller

A possible implementation of a LQR for a series elastic actuator is shown in figure 17. The controller itself consists of three primary gains. State feedback gains N_1 and N_2 are applied to the deduced states of the system, the torque and change in torque over time. Feed-forward gain N_{FF} is used to compensate any steady state offset. An alternative solution to this is implementing integral feedback [19], [20], which increases robustness to model uncertainties and external disturbances, but it can also introduce stability and analysis issues. This is out of the scope of this research.

This controller layout is a equivalent to a generic PD state feedback controller, and the feedback gains could also be determined with other linear control techniques such as pole placement. What makes LQR distinct from other controllers is how the feedback gains N_1 and N_2 are derived. The

advantage of using LQR, is that tuning the controller is more intuitive for the designer, as choosing a set of poles that satisfy design criteria could result in unacceptably high actuator gains, i.e. finding the correct pole location can be non-intuitive.

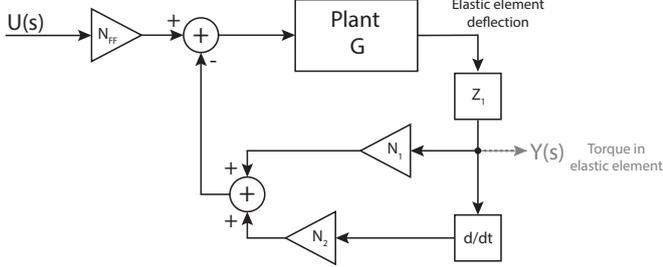


Fig. 17: Overview of a regular linear quadratic regulator.

In LQR costs are assigned to the different states via the state cost matrix $\bar{Q} \in \mathbf{R}^{d \times d}$, and to the effort with the effort cost matrix $\bar{R} \in \mathbf{R}^{e \times e}$. d denotes the amount of states in the system (size of the state vector) and e denotes the size of the input vector. Usually \bar{Q} and \bar{R} are diagonal matrices with values along the diagonal (ex. $Q_1 \dots Q_d$), of which an increase of the value penalizes the associated row of state performance (ex. Q_1 weighs performance of X_1) and input actuator effort respectively.

It is important to understand how changes in the associated gains affect closed loop system response. Because of the chosen states (Fig. 17), Q_1 weighs error in torque compared to the reference signal, Q_2 weighs change in torque error. There is only one input thus $\bar{R} = R$ is a single real value that penalizes actuator effort. The function of R is fairly simple. It effectively decreases the maximum actuator effort that is allowed to be exerted. This can be useful in implementation because actuator effort is in practice not limitless. It can prevent the motor controller and/or power-supply running against current limits, or to keep the motor cool during operation. Q_1 penalizes torque error, and thus directly weighs performance in how quickly this error is reduced. Finally Q_2 penalizes a change torque error, and thus effectively adjust the damping applied. A larger weighing of Q_2 , means more damping is applied.

This results in a simple method in tuning LQRs for series elastic actuators employing this control layout. Firstly making Q_2 very small, and using a combination of R and Q_1 to adjust the response as required by the specific application in terms of closed-loop bandwidth and actuator effort. One can be chosen constant, whilst varying the other. After this, use Q_2 to adjust damping until a desired response is achieved, which is usually critically damped.

First considered is the application of a linear quadratic regulator to two linear plants with stiffness values of $k = 30$ and $k = 300$, representing the low and high stiffness ranges of the non-linear elastic element respectively. The other plant

parameters are as described under Sec. V-A. As both R and Q_1 are in direct trade-off, Q_1 was chosen to be constant at 1, whilst varying R . The following response of a low-stiffness and high stiffness system is shown in Fig. 18.

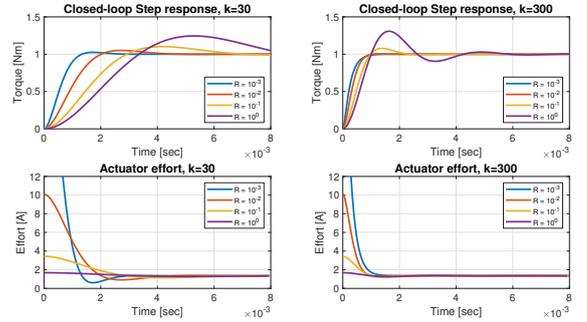


Fig. 18: Closed-loop performance for a linear QR with system parameters described in table III, with a stiffness of $k = 30$ (left), $k = 300$ (right) and $Q_1 = 1$, $Q_2 = 5 \times 10^{-8}$.

This demonstrates two things. Firstly, impact of a change in R on the performance and effort. An increase R (i.e. penalizing actuator effort) decreases actuator effort, but at the cost of reduced system performance in both settling time and overshoot. Taking R as constant and varying Q_1 would yield a similar response, that an increase in Q_1 causes an increase in bandwidth at the cost of increased actuator effort. It is up to the designer to strike a balance between acceptable performance and exerted effort for a particular application. Secondly, it can be observed that for a similarly tuned controller, an increase in stiffness results in a significant change in the damping ratio of the closed loop response, but more notably a significant reduction in rise time. This shows that a linear controller tuned for any stiffness of a non-linear elastic element shows very different behaviour when operating in another stiffness region. This implies that a linear controller for one stiffness is insufficient and thus warrants the consideration of more complicated control structures, as the one proposed.

C. Proposed Gain-scheduling LQR

Shown in Fig.19 is an overview of the proposed gain scheduled quadratic regulator. It is marked by the addition of mappings M_1 and M_2 (and associated mapping Z_2) when compared to the 'regular' LQR as presented in Fig.17. These actively change the feedback gains using information available on the current stiffness state, which is computed using Z_2 . Plant G models the input actuator current to the measurable output state of actuator deflection.

The gain scheduling part of the controller is formed by the previously mentioned deflection to stiffness mapping Z_2 , and the gain-scheduling blocks M_1 and M_2 . Z_1 and Z_2 (Eq. 13) denote the mappings from deflection to torque and deflection to stiffness respectively. These functions can be used directly to infer the torque and stiffness from the measured actuator deflection. Gain scheduling blocks M_1 and M_2 use this

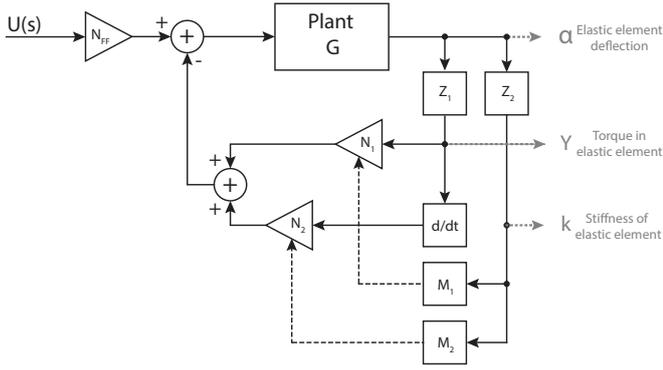


Fig. 19: Overview of the proposed gain-scheduled quadratic regulator.

information to apply changes to the state feedback gains N_1 and N_2 .

$$\tau = Z_1(\alpha), k = Z_2(\alpha) \quad (13)$$

The range of values for N_1 and N_2 can be derived by taking the linear stiffness plant and varying the stiffness parameters over the expected operating range and calculating the linear quadratic regulator gains at each point. A curve can be derived from this information which can then be subsequently used to schedule the gain of the controller. Fig.20 shows the change in QR gains with a varying stiffness. The parameters used are described under sec.V-A.

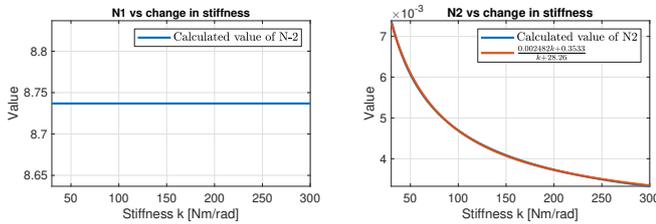


Fig. 20: Variation of LQR gains N_1 and N_2 over stiffness range with parameters of sec.V-A.

As shown in Fig. 20 QR gain N_1 does not vary with change in stiffness. Because N_1 stays constant over deflection/stiffness range, its associated gain-scheduling mapping M_1 is equal to zero. N_2 does vary with change in stiffness. The mapping of associated gain-scheduling mapping M_2 can be implemented in a variety of different ways. It can be done with a pre-calculated lookup table, dynamically calculating the required LQR gain or by fitting a curve to the modelled change in N_2 versus stiffness (as is done here).

The last QR gain is the feed-forward component N_{FF} , which purpose is to achieve zero steady state error. This can

analytically be proven to be equal to Eq.14.

$$N_{FF} = N_1 + \frac{n}{g} \quad (14)$$

It is purely dependent on the feedback gain (known and constant) and the ratio of the torque constant and the gearbox reduction (both known and constant). This means that N_1 is both constant and known when designing the control system, and relatively robust to changes in the plant.

V. SIMULATED RESULTS

The gain-scheduled linear quadratic controller performance is compared against the linear quadratic regulator performance in simulation to evaluate viability of the concept. Also evaluated is the compromise that non-linear series elastic actuators offer in the low and high deflection ranges, with respect to torque tracking and transparency, and how these compare with fixed stiffness series elastic actuators.

A. Parameters

Tab. III describes the physical system parameters. The motor parameters g , n , R_m and I_m are estimated with the help of the datasheet of the motor that will likely be used for later practical control tests. C_k is implemented with the derived torque/stiffness curve model shown under Sec.II-D.

TABLE III: Physical parameters used for simulations

Parameter	Value
Motor torque constant (g) [$\frac{Nm}{A}$]	5.25×10^{-2}
R_m [$\frac{Ns}{m}$]	1.5×10^{-4}
I_m [kgm^2]	4.45×10^{-7}
R_d [$\frac{Ns}{m}$]	1.5×10^{-4}
C_k [$\frac{Nm}{rad}$]	30 - 300 (see Sec.II-D)
Gearbox ratio (n) [x]	7.1111×10^{-2}

TABLE IV: Control system parameters used for the simulations

Parameter	Value
Q_1	1
Q_2	5×10^{-8}
R	1×10^{-2}

The parameters used for tuning the controller are shown in Tab. IV. These values are used for tuning the controllers presented unless noted otherwise. When using the above LQR parameters with a plant at varying stiffnesses, the resulting state-feedback gain N_2 changes as described in Sec. IV-C. Fitting to these values with a fractional fit yields Eq. 15, which is used for gain-scheduling.

$$N_2 = \frac{2.482 \times 10^{-3}k + 3.533 \times 10^{-1}}{k + 2.826 \times 10^1} \quad (15)$$

B. Step response

The most intuitive way of seeing how a gain-scheduling controller would improve upon a linear controller with fixed gains is with a step response. Evaluation of two linear plants with stiffness values at the extrema of what the non-linear series elastic element can attain can be used for this. The low stiffness linear plant has an associated set of 'low stiffness' gains, and the high stiffness linear plant an associated calculated set of 'high-stiffness' gains. The two linear plants can each be then evaluated with the controller tuned for the lower stiffness plant, the controller tuned for the higher stiffness plant, and the gain scheduled controller. The tuning is done with the QR parameters as shown in Tab. IV.

The result is shown in Fig.21. Initial tuning of the control systems is done to achieve as close to critical damping for each of the respective controllers as possible (i.e. low stiffness linear controller reaching critical damping in the low stiffness plant, high stiffness linear controller with high stiffness plant). Comparing the low-stiffness plant results (blue and yellow lines), it is clear that the controller tuned specifically for low-stiffness plants (blue) is outperforming the controller that was initially tuned for high-stiffness plants (yellow). The controller tuned for high-stiffness plant shows under-damped response and excessive actuator effort compared to the low-stiffness controller. Now comparing the controllers applied to the high-stiffness linear plant (red and purple) the opposite can be observed. Here the controller tuned of high-stiffness plants (red) shows faster rise time, whilst the controller for low-stiffness plants (purple) shows over-damped behaviour. This comparison already shows the potential benefits of scheduling the gain of the controller over operating range.

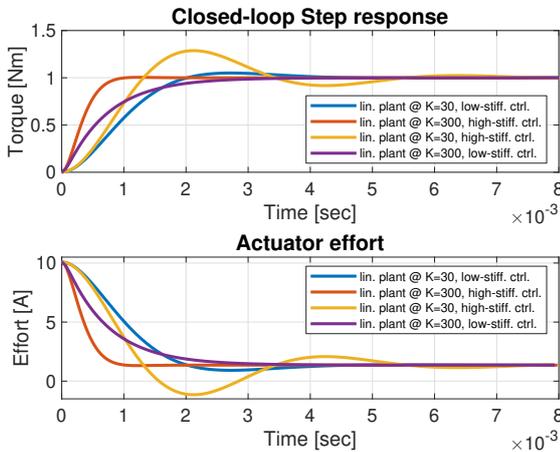


Fig. 21: Step-response of the system at stiffness extremes, with a linear quadratic regulator for feedback.

The controller tuned for high-stiffness applied to a low-stiffness plant seems 'too fast', whilst the controller tuned for low-stiffness applied to the high-stiffness plant seems 'too slow'. This can indeed be seen in Fig.20, where there exists a

larger difference in N_2 for a given change in k for the lower stiffness region. Parameter N_2 is responsible for the damping of the controller.

Applying the controllers tuned for specific stiffnesses and gain-scheduled controllers to the non-linear model provides a similar conclusion. Fig. 22a, 22b and 22c show the response in the low-stiffness region (low torque), middle of the stiffness region (medium torque), and high-stiffness region (high torque) respectively.

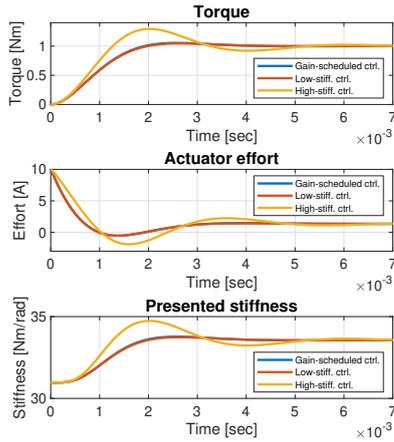
The response in the low stiffness region (Fig. 22a), stepping from 0 to 1 [Nm] torque shows that the gain-scheduled controller (blue) shows an almost identical response to the linear controller tuned for low stiffness (red). This is because the non-linear plant only experiences minimal deflection and is in the low stiffness region of operation. The linear controller tuned for high stiffness (yellow) shows excessive overshoot and oscillation compared to the other two controllers.

In the case of a stiffness in the middle of the stiffness range (Fig. 22b), the gain-scheduled controller offers a compromise between the slightly over damped response of the linear controller tuned for the low-stiffness plant, and the under damped response of the linear controller tuned for the high-stiffness plant. It has a faster settling time than either controller. Comparing the time it takes for the signal to be within a 5% error band of the steady state signal, the gain-scheduled controller is in this case approximately 30% and 43% faster than the controller tuned for the low-stiffness plant and high-stiffness plant respectively.

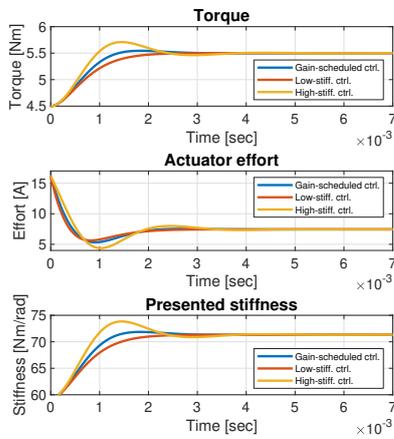
Lastly, the high stiffness region (Fig. 22c) presents a mirrored situation to the low-stiffness region. The gain-scheduled controller acts nearly identical to the linear controller tuned for high stiffness, whilst the linear controller tuned for low stiffness shows over-damped behaviour. One should also note that a higher-stiffness plant clearly shows an increase in bandwidth of the system. Comparing the settling time between plants with the same gain-scheduled controller, there is an approximately 75% reduction in settling time comparing the high-stiffness situation (Fig. 22c) versus the low-stiffness situation (Fig. 22a).

This show that either linear controller tuned for a specific stiffness shows a weakness on the opposite end of the stiffness range. In contrast, the gain scheduled controller is able to handle the wide range of presented stiffnesses, and provide reasonable performance at any stiffness. For stiffnesses between the maximum and minimum, the gain scheduled controller also shows superior behaviour to either controller tuned for a single stiffness extrema.

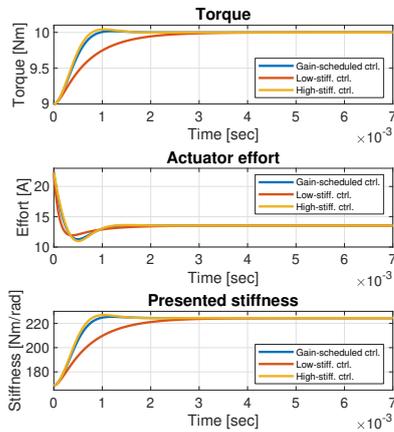
Now, a step through the entire stiffness range can present other challenges due to the system non-linearity. A step from 0 to 10 Nm, and in the opposite direction 10 to 0 Nm are shown in Fig. 23. Stepping from 0 to 10 [Nm] for the controller tuned



(a) Step from 0 to 1 [Nm] torque.



(b) Step from 4.5 to 5.5 [Nm] torque.



(c) Step from 9 to 10 [Nm] torque.

Fig. 22: Step response of the non-linear plant at varying stiffnesses, comparing different controllers.

for the high-stiffness plant shows an under-damped response with excessive actuator effort. The low-stiffness controller shows the best behaviour out of the three, being slightly over-damped. The gain-scheduled controller exhibits a response that is in between the two. Looking at the step from 10 to 0 [Nm] shows that the low-stiffness controller exhibits the highest settling time. The high-stiffness controller has significant undershoot. In this case, the gain-scheduled controller shows a response in between the two controllers tuned for stiffness extrema, which for this particular step response provides the best result when compared to the other two controllers.

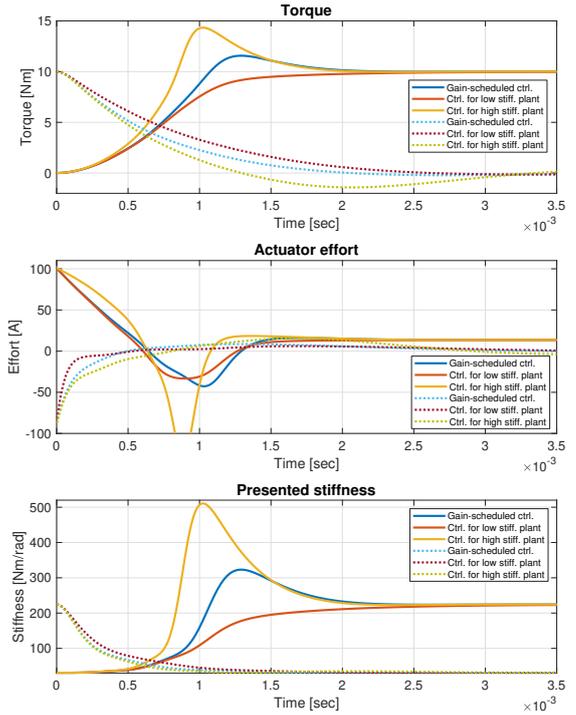


Fig. 23: Step response from 0 to 10 [Nm, and from 10 to 0 [Nm] of the non-linear plant, comparing different controllers.

C. Torque tracking

Torque tracking bandwidth gives an indication as to how well a particular controller and plant combination can track a desired torque over different frequencies. For measuring torque tracking, the load side of the series elastic element is fixed. The controller is commanded to follow a swept sine input with an amplitude of 1 [Nm], with the output being analysed in the frequency domain. The results are shown in Fig. 24a and Fig. 24b.

Similar results to those of the step response are seen here. It should be noted that in the $300 \frac{\text{Nm}}{\text{rad}}$ graphs for both Fig. 24a and Fig. 24b, the gain-scheduled controller response is identical to the high-stiffness response. Observing both figures, the first thing that can be noticed is that for the

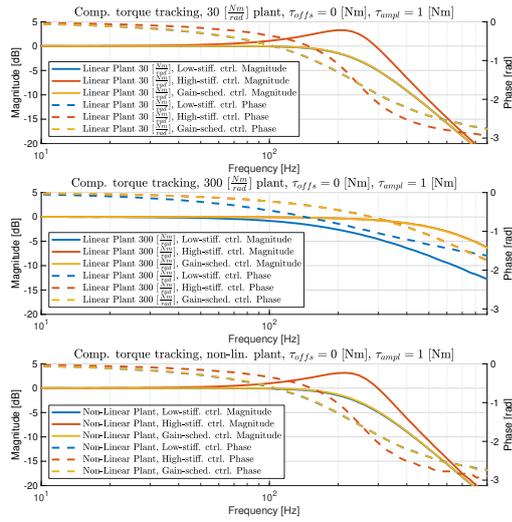
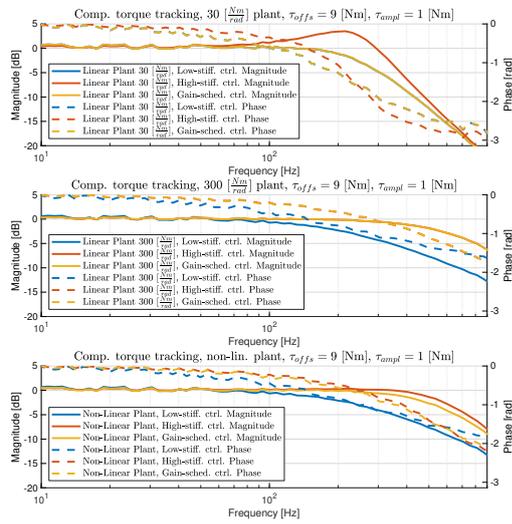

 (a) Rest state, $\tau_{offset} = 0$ [Nm].

 (b) Offset of $\tau_{offset} = 9$ [Nm].

Fig. 24: Torque tracking of the various actuator and controller combinations.

linear plants, the gain-scheduled controller shows nearly identical results to the properly tuned controller associated with that particular stiffness plant. Secondly looking at only Fig. 24a, it is clear that the high-stiffness plant shows superior torque tracking performance to the low-stiffness controller and gain-scheduling controller. This is as expected, as the actuator deflection is minimal, and thus the stiffness range the non-linear actuator is in is nearly identical to the stiffness presented by the low-stiffness linear plant. This however does not show the complete picture.

If we proceed by giving an offset to the swept sine of 9

[Nm], the performance of the non-linear plant becomes similar to that of the linear high-stiffness plant. The torque tracking performance of the linear plants does not change with a change in torque offset. This entails that the non-linear series elastic actuator shows better torque tracking performance than a low-stiffness linear controller under higher loads.

To highlight the specific advantage of a non-linear plant versus low-stiffness plant. Comparing the low-stiffness plant and low-stiffness controller with the non-linear plant with gain-scheduling controller at $\tau_{offset} = 9$ [Nm], there is an improvement in -3dB torque tracking bandwidth of 107%. For the low-stiffness linear plant and non-linear plant at their respective -3dB points the phase shift is -90° and 86° , showing this increase in bandwidth is not at the expense of stability. The phase shift of the high-stiffness plant at its -3dB point is slightly better, at -74° .

D. Transparency

For compliant actuators that are used in interaction standard control system metrics such as closed-loop tracking bandwidth are less valuable and not all encompassing [4]. Because of the employment in interaction with humans, output impedance is of primary interest to evaluate interaction performance. This can be defined as the ability to track commanded torques in the presence of external disturbances.

Transparency is measured by commanding the actuator to track a given torque, whilst injecting a disturbance signal in the form of movement in the load. For an external 'interactor', when setting the torque setpoint at 0 [Nm], a high transparency of the actuator and controller combination would make the actuator seem as if it is not there. For deducing the frequency response of the actuator load motion whilst regulating to desired output torque, a sine wave of increasing frequency is forced upon the velocity of the load.

Fig. 25a and 25b show the residual torques of various controller and plant combinations, at fixed constant 0 [Nm] and 9 [Nm] torque setpoints respectively. The first thing that can be seen, is that the low-stiffness linear plant (and the non-linear plant with low torque set-point) have a better transparency than the linear plant with high-stiffness.

The performance of the linear plants is again practically identical regardless of the desired torque set point. These figures however show that the transparency of the non-linear plant approach those of the high-stiffness plant at higher loads. A practical situation in which this can happen, is with a robot arm holding a load against gravity, whilst being interacted with.

E. Result discussion

The purpose of the gain-scheduled controller was to take advantage of the change in plant dynamics with a change

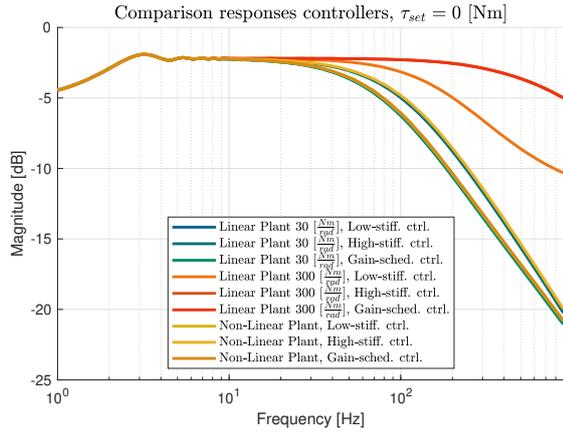
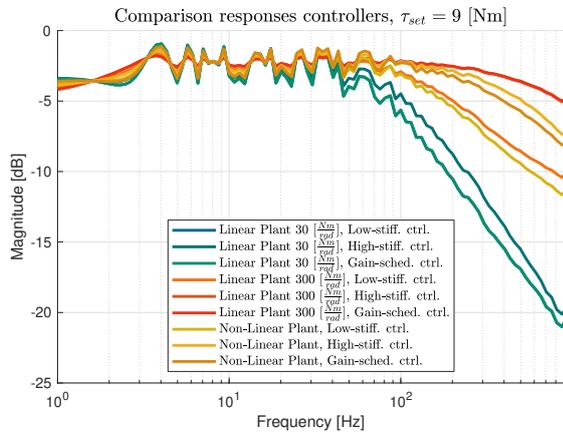

 (a) Torque setpoint of $\tau_{set} = 0$ [Nm].

 (b) Torque setpoint of $\tau_{set} = 9$ [Nm].

Fig. 25: Residual torque response of the various actuator and controller combinations.

in stiffness. Observing the step responses (Sec. V-B) of the gain-scheduled controller on the non-linear system, and comparing those against the linear controllers, it shows that the gain-scheduled controller shows an improved consistency and predictability across different operating conditions, compared to a linear controller tuned for either low or high stiffness, respectively. Also when making a large step through the whole stiffness range the non-linear gain scheduled controller shows a compromise in behaviour which again makes it more predictable and consistent for the design engineer. In some specific situations the gain scheduled control might perform similar or worse than a linear actuator tuned for a specific stiffness, but over a variety of different operating conditions it performs better overall.

Torque tracking results (Sec. V-C) firstly show that the gain-scheduled controller always achieves good results regardless of the stiffness range the non-linear series elastic

element is operating in. Additionally, it shows that although the non-linear plant shows similar torque tracking results to the low-stiffness controller at low load levels, when the load increases the performance markedly improves. This is similar to how human muscles operate, increasing stiffness with load.

The transparency of the non-linear plant (Sec. V-D) is also dependent on the stiffness range, which is often load dependent. This is also similar to the behavior of human muscles. At lower torque levels it exhibits nearly identical performance to the low-stiffness plant. The transparency performance reduces with increase in torque output.

The torque tracking and transparency results can be summarised by observing that the non-linear series elastic actuator offers the transparency advantages of a low-stiffness controller at low load levels, whilst providing the advantages of high torque tracking bandwidth like high-stiffness plants at high loads. It also retains advantages that are not immediately visible in the graphs (safety and higher torque resolution at lower torques) compared to a high-stiffness actuator. Additionally compared to a low-stiffness actuator, it can achieve a larger torque range, given a maximum deflection.

VI. DISCUSSION

When considering the elastic element performance, the base-stiffness of 40 Nm is similar to model predictions. The maximum stiffness however, is significantly lower. This could be due to manufacturing tolerances and the magnets pushing each other away in non-axial directions with high torques/forces. The approximate stiffness range (min to max stiffness) of 3x is well within design criteria and clearly demonstrates the validity of a magnetic based series elasticity concept. It also shows that the presented analytical magnet-based series elastic model in Sec. II-D gives accurate predictions and provides a useful tool in future design of magnetic based non-linear series elastic actuators.

The difference between the positive and negative stiffness direction could be explained by misalignment in the center of rotation of the test-bench, versus that of the elastic element.

Some degradation of performance of the elastic element can be observed over time. The exact cause has not been determined, however, various possible causes have been identified:

- 1) Permanent magnet degradation: Magnet proximity to other strong magnetic fields can cause permanent loss of field strength. The magnets are brought in close proximity of each other at high deflection, and thus this might be the cause. However, measuring the magnets has shown no measurable reduction of field strength, and field strength of the magnets is also too low to cause demagnetization. Additionally, the degradation is only in stiffness non-linearity, and seems to have negligible to no effect on the maximum torque range.
- 2) At high deflection and torque levels, some deformation in the structure is visible. The 3D printed structure is

not infinitely stiff, and layer orientation can exacerbate this problem. It is recommended to change the design to better resist non-ideal axial forces. Using a different material such as aluminium would also make the design more rigid.

- 3) Internal structural damage and/or weakening of the 3D printed prototype due to high internal forces/torques could also explain the phenomena. No apparent damage can be observed on the outer skin of a used actuator that exhibits degradation. One of the used actuators has been cut open, however, it is hard to spot any layer de-bonding or damage caused by deflection. Therefore this cannot be crossed off as a possible cause.

VII. CONCLUSION

The concept of a non-linear series elastic element based on magnets has been proven to be viable. The developed model gives an accurate approximation of an actuator given chosen parameters and magnets, and provides a valuable tool for both simulation and development. The proof-of-concept prototype shows that the concept also holds in practice, and that it has attractive benefits for many practical applications.

The progressive stiffness presented in non-linear series actuators combines the benefits of low and high stiffness series elastic elements, in low and high output torque respectively.

Gain-scheduling the quadratic regulator gives more predictable, consistent and often superior results over the wide operating range compared to traditional linear controllers. The larger the stiffness range presented by the elastic element, the larger the advantage the gain-scheduled controller offers. Gain scheduling can guarantee designed operating behaviour over the entire stiffness range. Implementation of a gain scheduling QR controller over a linear QR controller does not require any additional resources or sensors, which makes the usage doubly attractive to improve actuator performance at no additional cost. Computational complexity is also low, allowing the controller to be parametrized off-line.

VIII. FUTURE WORK

It would be beneficial to explore the usage of arc-segment magnets and verifying the simulated result that they can indeed achieve similar torque and stiffness characteristics at reduced physical size. It should be verified that the presented model also holds for these magnets, and if it thus would be beneficial to use these types of magnets.

In terms of elastic element design, a new design that can better resist non-axial forces would be beneficial, especially for real world applications. For manufacturing stiffer materials such as aluminium or steel could be investigated. By using stiffer and stronger materials the total dimensions can be reduced.

It is yet unclear what causes the reduction in torque non-linearity and thus stiffness after repeated testing. The above

mentioned suggestion of using stiffer materials could rule out internal mechanical failure.

Gain-scheduling a quadratic regulator has proven to be advantageous applied to non-linear series elastic elements. Researching into design procedures that define the operating bounds of an applied controller would be highly beneficial in making gain-scheduling more accessible.

Applying gain-scheduling to different types of controllers and finding design procedures for these controllers is a potentially interesting avenue to explore. Other types of non-linear control in general might also find similar improvements as were found in this work.

Practical experiments with the constructed prototype evaluating the real-world performance of the gain-scheduling QR controller for further exploration of the concept and to verify the simulation results found in this research are currently being worked on.

IX. ACKNOWLEDGEMENTS

Firstly, I would like to thank dr.ir. W. Roozing for his continued feedback and excellent guidance during the thesis process. His experience has helped to guide the research more effectively and his critical thinking always provided useful feedback and helped to provide a different point of view.

Next, the technical staff at the RAM department of the University of Twente, in particular ir. Te Riet O/G Scholten and ir. S. Smits, for their generous feedback and help in practical considerations of the construction and manufacturing. Also from the University of Twente, prof.dr.ir. Abelmann for his insight as an expert on magnetism and his help with magnet simulations.

REFERENCES

- [1] Zinn, M., Khatib, O., Roth, B. *A New Actuation Approach for Human Friendly Robot Design*. IEEE International Conference on Robotics and Automation, ICRA '04. 2004. doi:10.1109/robot.2004.1307159
- [2] Pratt, G.a., and M.m. Williamson. *Series Elastic Actuators*. Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots, doi:10.1109/iros.1995.525827.
- [3] Pratt, Gill A., et al. *Stiffness Isn't Everything*. Experimental Robotics IV Lecture Notes in Control and Information Sciences, 30 June 1995, pp. 253–262.. doi:10.1007/bfb0035216.
- [4] Roozing, W., Malzahn, J., Kashiri, N., Caldwell, D. G., Tsagarakis, N. G. (2017). *On the Stiffness Selection for Torque-Controlled Series-Elastic Actuators*. IEEE Robotics and Automation Letters, 2(4), 2255–2262. doi:10.1109/lra.2017.2726141
- [5] Robinson, D., Pratt, J., Paluska, D., Pratt, G. (1999). *Series Elastic Actuator Development for a Biomimetic Walking Robot*. 1999 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (Cat. No.99TH8399). doi:10.1109/aim.1999.803231
- [6] Zhang, Juanjuan, and Steven H. Collins. *The Passive Series Stiffness That Optimizes Torque Tracking for a Lower-Limb Exoskeleton in Human Walking*. Frontiers in Neurorobotics, vol. 11, 2017, doi:10.3389/fnbot.2017.00068.
- [7] Morita, T., Sugano, S. (1995). *Design and development of a new robot joint using a mechanical impedance adjuster*. Proceedings of 1995 IEEE International Conference on Robotics and Automation. doi:10.1109/robot.1995.525630

- [8] Choi, J., Park, S., Lee, W., Kang, S. (2008). *Design of a robot joint with variable stiffness*. 2008 IEEE International Conference on Robotics and Automation. doi:10.1109/robot.2008.4543455
- [9] Choi, J., Hong, S., Lee, W., Kang, S. (2009). *A variable stiffness joint using leaf springs for robot manipulators*. 2009 IEEE International Conference on Robotics and Automation. doi:10.1109/robot.2009.5152716
- [10] Migliore, S., Brown, E., Deweerth, S. (n.d.). *Biologically Inspired Joint Stiffness Control*. Proceedings of the 2005 IEEE International Conference on Robotics and Automation. doi:10.1109/robot.2005.1570814
- [11] Malzahn, J., Barrett, E., Tsagarakis, N. (2019). *A Rolling Flexure Mechanism for Progressive Stiffness Actuators*. 2019 International Conference on Robotics and Automation (ICRA). doi:10.1109/icra.2019.8794004
- [12] Thorson, I., Caldwell, D. (2011). *A nonlinear series elastic actuator for highly dynamic motions*. 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems. doi:10.1109/iros.2011.6094913
- [13] Austin, J., Schepelmann, A., Geyer, H. (2015). *Control and evaluation of series elastic actuators with nonlinear rubber springs*. 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). doi:10.1109/iros.2015.7354315
- [14] Schepelmann, A., Geberth, K. A., Geyer, H. (2014). *Compact nonlinear springs with user defined torque-deflection profiles for series elastic actuators*. 2014 IEEE International Conference on Robotics and Automation (ICRA). doi:10.1109/icra.2014.6907350
- [15] Schmit, N., Okada, M. (2012). *Design and Realization of a Non-Circular Cable Spool to Synthesize a Nonlinear Rotational Spring*. Advanced Robotics, 26(3-4), 234-251. doi:10.1163/156855311x614545
- [16] Realmuto, J., Klute, G., Devasia, S. (2015). *Nonlinear Passive Cam-Based Springs for Powered Ankle Prostheses*. Journal of Medical Devices, 9(1). doi:10.1115/1.4028653
- [17] Axelsson, P., Pipeleers, G., Helmersson, A., Norrlöf, M. (2014). *H_∞ Synthesis Method for Control of Non-linear Flexible Joint Models*. IFAC Proceedings Volumes, 47(3), 8372-8377. doi:10.3182/20140824-6-za-1003.00142
- [18] Montanaro, M., Wang, Z., Fallah, S., Sorniotti, A., Lenzo, B. (2018). *H_∞ A gain scheduled robust linear quadratic regulator for vehicle direct yaw moment Control*. Control. Mechatronics, 51, 31-45. doi:10.1016/j.mechatronics.2018.01.013
- [19] Kedjar, B., Al-Haddad, K. (2006). *LQR with Integral Action for Phase Current Control of Constant Switching Frequency Vienna Rectifier*. 2006 IEEE International Symposium on Industrial Electronics, Montreal, QC, Canada, 2006, pp. 1461-1466, doi: 10.1109/ISIE.2006.295687.
- [20] Rocha, R., Silvino J.L., Resende, P. (2012). *LQR with integral action for the control of variable speed induction generator connected to AC grid*. 1061-1066. 10.1109/IECON.2012.6388573.
- [21] K&J. Magnetics Inc., *Magnet repelling force data for box-type magnets*, <https://www.kjmagnetics.com/calculator.repel.asp?calcType=block>, Dec. 2020.
- [22] Mathworks, *Matlab - Linear-Quadratic Regulator (LQR) design*, <https://uk.mathworks.com/help/control/ref/lqr.html>, Jan. 2021.
- [23] ATI Industrial Automation, *F/T Sensor: Mini40*, https://www.atia.com/products/ft/ft_models.aspx?id=Mini40, Jan. 2021.
- [24] G2Elab, *MacMMems : Macro Modeler for Magnetic Mems*, <https://g2elab.grenoble-inp.fr/fr/recherche/macmmems-macro-modeler-for-magnetic-mems>, Jan. 2021.
- [25] Sparkfun Electronics, *Load Cell - 10kg, Straight Bar (TAL220)*, <https://www.sparkfun.com/products/13329>, Jan. 2021.
- [26] Sparkfun Electronics, *SparkFun Load Cell Amplifier - HX711*, <https://www.sparkfun.com/products/13879>, Jan. 2021.

This document was designed with double sided printing in mind to save the environment.

For this purpose this page is intentionally left blank

Chapter 4

Elastic element concepts and iterations

In this chapter, the mechanical design iterations of the elastic element are discussed. First, the concept exploration phase of the project is shown, including the considered candidates for an elastic element. This is followed by the various elastic element iterations that have been designed and constructed, including a motivation as to why these changes were made.

4.1 Concept exploration

Initial design concept exploration did not focus on specifically magnetic based series elastic actuators, but rather at novel elastic element ideas in general. This resulted in a few viable concepts which eventually spawned the magnetic based series elastic actuator as presented in this thesis. The concepts will be shortly presented and discussed below.

4.1.1 Pseudo-non-linear modular elastic element

The first concept describes the use of multiple linear springs of increasing stiffness, which are connected together in parallel with increasing deflection through a cam system. Although the springs individually are linear, when connected in this way they can form a pseudo-non-linear curve. This is similar how logarithmic potentiometers generate their linearity.

The advantages of such an idea is a high degree of flexibility and 'hot swappable' units to tune the stiffness curves. The user could relatively cheaply and easily use readily available springs to make an approximation of a progressive stiffness profile desired. The elastic units themselves could be manufactured relatively cheaply and easily.

An initial CAD concept (fig.4.1) illustrates the concept. In this example, the device consists of 3 individual elastic units. The actuator is connected to the central hub which connects all units together. The outer rings are coupled together through a cam system. The outer ring of the weakest spring (elastic unit #1, which will always be engaged) is connected to the output link. The further the output link is deflected, the more elastic units (each of which have an increase in stiffness) are coupled in parallel, thus increasing the perceived total stiffness of the element

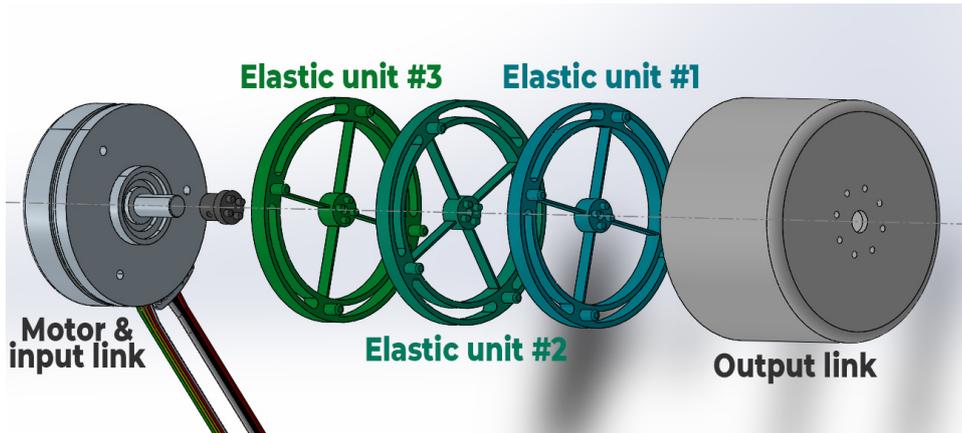


Figure 4.1: CAD of a possible implementation of the pseudo-non-linear concept.

progressively. An example of how this concept generates a non-linear stiffness is shown in Fig. 4.2.

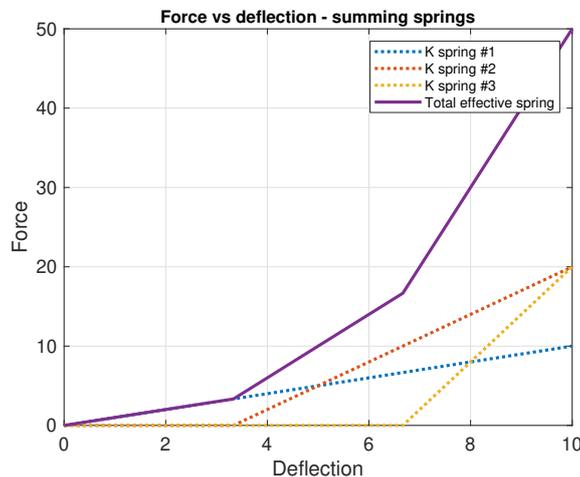


Figure 4.2: Shown is an example of the summing of linear stiffnesses to create a pseudo-non-linear piecewise curve.

Although promising, this concept was less interesting to investigate than other ways of generating non-linearity.

4.1.2 Compressible gas based

Compressed gas within an enclosed chamber exerts a pressure on the walls proportional to the surface area of the wall and the pressure of this gas. Linear gas based springs are common in the industry and available in a variation of sizes and shapes for a large variety of different applications. These types of springs work by compressing the gas they contain. This in turn stores energy because of the increase in pressure, which causes an increase in force acting opposite to the force that causes the compression through the piston. This is a linearly increasing force on the gas system versus deflection.

This concept introduces the idea of changing the surface area that is presented by the gas piston to the compressed gas, depending on the deflection. By regulating

the surface area for a given deflection a spring profile can be attained.

This concept was not evaluated further due to the difficulty of with working with compressed gasses. Additionally, rapid prototyping gas-sealed designs is extremely difficult to achieve. It might still be an interesting venue to explore in the future.

4.1.3 Rubber non-linearity based elastic element

Rubber materials had already been used in previous research done by J. Austin et.al. [13], however in this particular research the rubber is not used for its non-linear properties. Rather, it is deformed through a regular mechanical cam system in its (mostly) linear range to produce non-linear stiffness.

The paper did bring up the idea if rubber could be used in its strongly non-linear region, as it goes through various different regions of stiffness. The first region is the so called 'elastic region', which is used in the previously mentioned research. The stiffness presented here acts nearly linear versus deflection. The end of the elastic region is defined as the yield point of the rubber, after which the rubber acts increasingly non-linear. This region is called the 'non-elastic region'. At the end of the non-elastic region is the failure point and exceeding this point causes instant catastrophic and permanent structural failure.

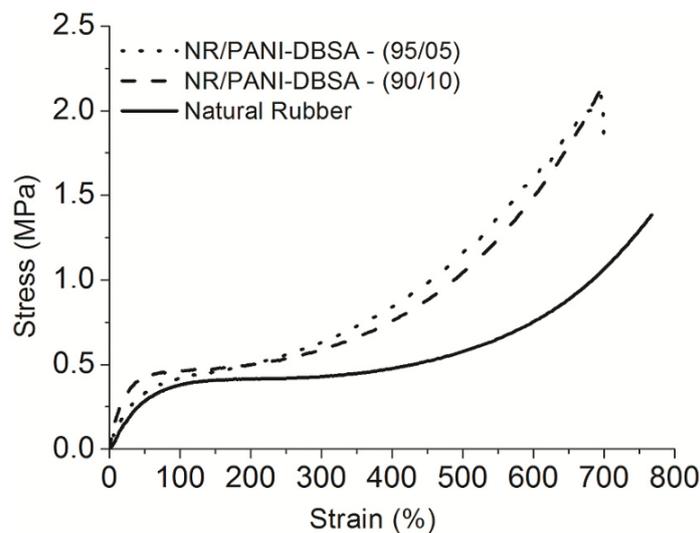


Figure 4.3: Figure showing the stress versus strain characteristics of rubber, source: [21].

Although promising initially, rubber used in the non-elastic region does suffer performance degradation over use because of the tearing of polymer chains. Other non-linear behaviour of rubber (such as the hysteresis exhibited in the pull versus return stroke) can be minimized but not avoided. This makes the use of rubber less attractive for real-world applications and a less promising candidate for a practical and useful novel non-linear elastic element.

4.1.4 Magnet based elastic element

Previous work in variable stiffness actuators had used magnets in the past [8]. However, it has not been used to exploit the non-linear property for non-linear stiffness series elastic actuators. Magnets possibly provide substantial benefits versus existing mechanical systems that are used in the generation of non-linear elasticity.

- Reduce and/or eliminate unwanted non-linearities in mechanical systems such as backlash, stiction and friction.
- Reduce the need of complex mechanical cam mechanisms, thus decreasing weight, size and complexity whilst increasing reliability.
- Reduction in mechanical wear on parts, reducing maintenance and maintenance costs, and increasing reliability further.
- Due to the lack of cam mechanisms, further degrees of freedom are opened in the design of these actuators in various shapes and sizes.

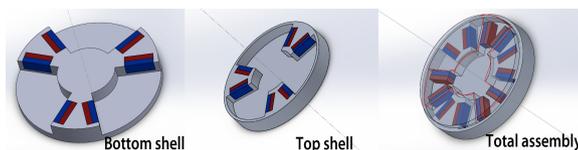


Figure 4.4: Initial magnet-based elastic element sketch.

The inherent magnet-repelling force is progressive and non-linear in nature. This is exploited to generate a non-linear stiffness with Hooke's law. The initial sketch presented is conceptually identical to the developed prototype. It consists of two halves that are deflected with respect to each other. As the deflection with respect to each other brings the magnets in both halves closer to each other, it increases the repelling force and thus torque generated by the magnets. Because the repelling force is inherently non-linear it presents a non-linear progressive stiffness. Note that this concept could also work with a linear elastic element instead of a rotational one.

4.1.5 Magnet based elastic element with linear spring

As during the initial concept exploration phase of the project it was yet unclear whether the magnet based elastic element would produce sufficient stiffness with low levels of deflection. This concept proposed to add a linear spring in parallel with the magnet based elastic element. This would most likely consist of a torsion spring integrated inside the elastic unit. The linear spring would provide a reasonable base stiffness, whilst at higher deflection the stiffness would be dominated by the much larger stiffness produced by the magnetic-based elastic element.

This design would enable the viability of the magnet-based actuator even if the initial magnet torque would be insufficient, and provide more flexibility for the elastic element designer by somewhat decoupling the magnet parameters from the base stiffness. This comes at the cost of decreased reliability due to the extra components, and increased mechanical complexity.

4.2 Design iterations of the magnet based series elastic element

After selection of the magnetic based concept, a design was made and iterated upon. Each subsequent design has slight tweaks and improvements, which are highlighted under this section, including motivation.

4.2.1 Version 1 and 2

Version 1 and 2 were mostly used to rough out the concept and get a concept of the dimensions. They are essentially the same design, version 1 with box magnets, version 2 with arc-magnets. They are both shown in figure 4.5.

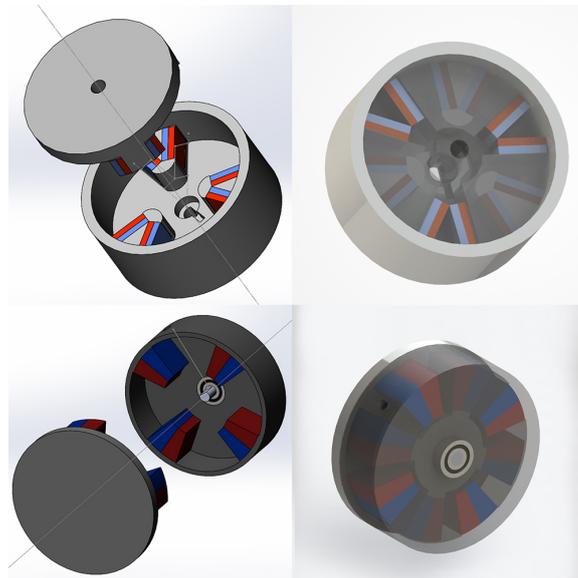


Figure 4.5: (Version 1 and 2 - with box magnets (top) and with arc magnets (bottom)).

4.2.2 Version 3

After the initial sketch versions, it was chosen to further flesh out a design with box-magnets. This was chosen due to widespread availability, lower lead-time and lower cost compared to custom arc-segment magnets. Maximum dimensions of the available magnets and the print space of the 3D printer dominate the choice of dimensions. Version 3 was mostly aimed to create a first 'buildable' prototype, which was to be reviewed with technical staff to refine before actual construction. Note that the two halves in all versions so far, consist of a metal shaft which is fixed to the actuator halves with grub screws.

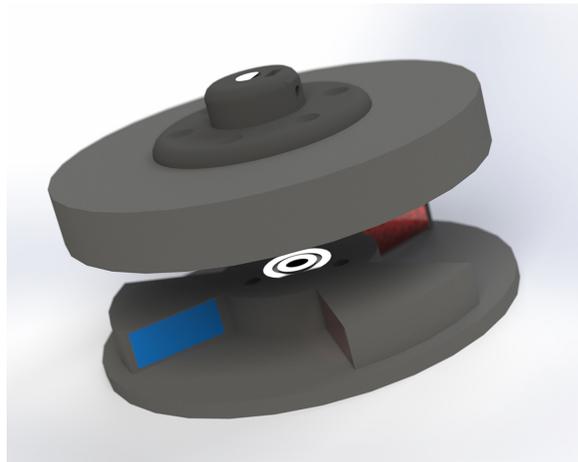


Figure 4.6: Version 3 - note the steel axle to attach the two halves together.

4.2.3 Version 4

One of the main weaknesses of the version 3 design is the weak attachment of the two halves. The axle has to constrain movement in 5 degrees of freedom. This was changed in the subsequent version, which keeps spacing between the two halves with the use of two thrust bearings. A 'locking plate' squeezes the two halves together with a bolt that threads into the lower half. The usage of thrust bearings also means this elastic element design is robust against non-axial forces. A ball bearing ensures axial alignment of the two halves. For this prototype it was also deemed unnecessary to have a linear spring in parallel with the elastic element to increase base stiffness due to initial results of the mathematical model.

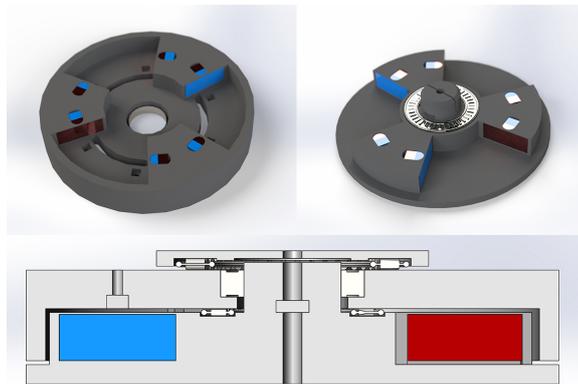


Figure 4.7: Version 4 - Renders of both halves (top) And a sideways cutaway view (bottom).

4.2.4 Version 5

Discussion with technical staff led to the decision to replace the thrust bearings with a ball-bearing combination to keep the two halves aligned. This reduces mechanical complexity. Changing the design to incorporate this results in what is shown in figure 4.8. This is the first printed prototype. Printing this was mostly to determine

adjustments for a version that could be used for testing.

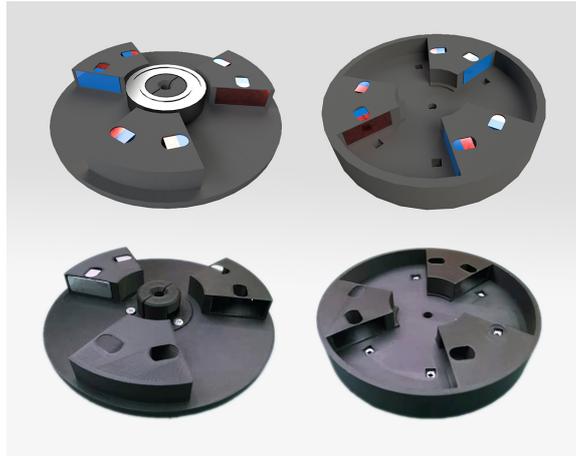


Figure 4.8: Version 5 - the first printed prototype, CAD design and fabricated prototype.

4.2.5 Version 6

After printing it was apparent that the bottom half exhibited a lot of flexing. The 'wall' that previously existed on only the upper half is split in version 6, adding half the wall to the lower part. This ensures the following:

- It stiffens the structure of the lower half, preventing structural deformation during operation
- It stops the print from exhibiting thermal deformation from the printing process
- It makes it easier to detach the printed unit from the build plate of the 3d printer

Next to these changes, other smaller changes were also implemented. These mostly came about with feedback from the technical staff on the printing process, and generally usability. These changes are:

- Adjusting the mounting holes of external devices
- Addition of a 'sensor plate' to which the sensor mounts. This enables the sensor to be added and removed without opening the device
- Tightening tolerances on the main ball bearing
- Tweaks to the magnet positions to maximize torque
- This version also reintroduces witness holes, which enable the user to view inside the device during operation
- Chamfering and rounding of edges to reduce stress points and make the device easier to remove from the 3d printer build plate



Figure 4.9: Version 6 - CAD and produced prototype.

4.2.6 Version 7

The last iteration of the design, version 7, came out of necessity after noticing significant deflection in the previous design over time. A 'skewing' of the top half with respect to the lower half, and bowing with extreme deflection angles could be caused by insufficient stiffness of the design. This resulted in the following refinements to the design:

- Increasing the 'floor' thickness
- Increasing the wall thickness
- Optimizing magnet position
- Increasing magnet 'roof' thickness
- A new design for what supports the bearing
- Minor tweaks to tolerances for fitments of the bearings

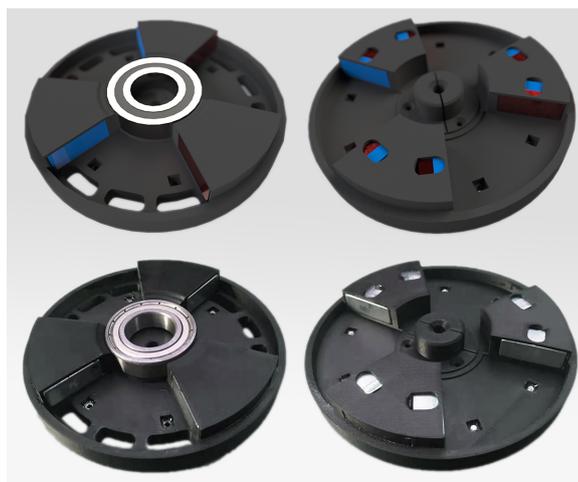


Figure 4.10: Version 7 - The final designed prototype. CAD and realization shown.

These changes do seem to have resolved the skewing and bowing issues. If these changes would not have provided the expected result, three possible directions were to be considered:

- Adding carbon fiber inlays in the floors of the two halves to significantly increase stiffness of the halves
- Redesigning the prototype such that it can be fabricated out of inherently stiffer materials such as aluminium.
- Redesigning the prototype to make use of thrust bearings to resist non-axial forces better.

This document was designed with double sided printing in mind to save the environment.

For this purpose this page is intentionally left blank

Chapter 5

Derivation of model and equations

This chapter will provide a deeper analysis of the presented models, and show the full derivation of the equations presented in the the paper.

5.1 IPM and Bondgraph

Firstly the non-linear series elastic element model is presented. This is done by first presenting a linear model, and deducing the state-space equations associated with this model. These are valuable in designing the controller at various different stiffnesses by varying this parameter. The non-linear model is found by using the linear bondgraph model, and replacing the linear C element by a non-linear element implemented as a non-linear Hooke's law with a polynomial equation as shown in the paper.

The initial ideal physical model of a non-linear series elastic actuator with load fixed is shown in figure 5.1. It can also be represented as a bondgraph, this allows easy cross-domain analysis, including the derivation of transfer function. This bondgraph is shown in figure 5.2.

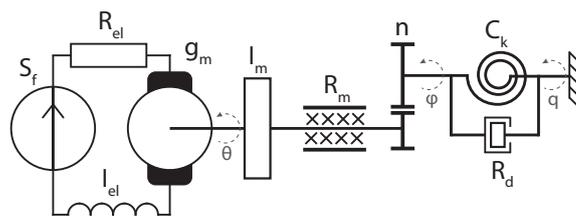


Figure 5.1: IPM of the actuator.

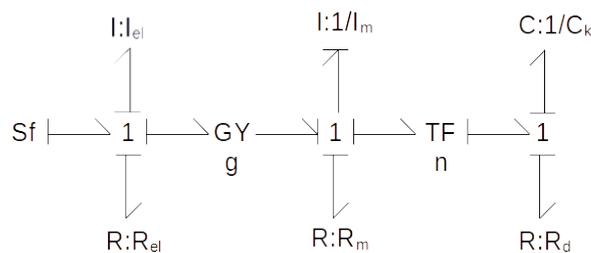


Figure 5.2: Bondgraph of the actuator with fixed load.

This bondgraph can then be simplified further with the traditional bondgraph simplification rules. This aids in making the state space derivation somewhat easier. See figure 5.3 for the simplified bondgraph.

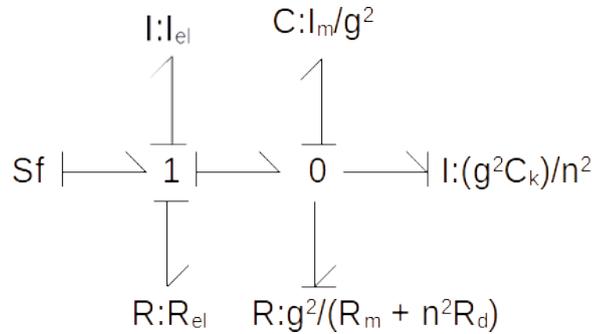


Figure 5.3: Simplified bondgraph.

Note that the electrical parameters (I_{el} and R_{el}) are not relevant for the derivation of the transfer function, however these do make it possible to deduce the voltage required by the motor during modelling. Additionally this information can be used to estimate power consumption and heating of the motor and motor-controller. Another thing to note is that the force in the spring should be scaled by a factor $\frac{g}{n}$ due to the simplification rules applied to simplify the bondgraph.

$$F_{spring} = \frac{g}{n} I_{flow} \quad (5.1)$$

For modelling where a non-fixed output is required, the load is represented as an inertia I_L with a combined bearing and airdrag resistance modelled by R_L . This bondgraph variant is shown in fig. 5.5. It is derived from the associated modified IPM shown in Fig. 5.4.

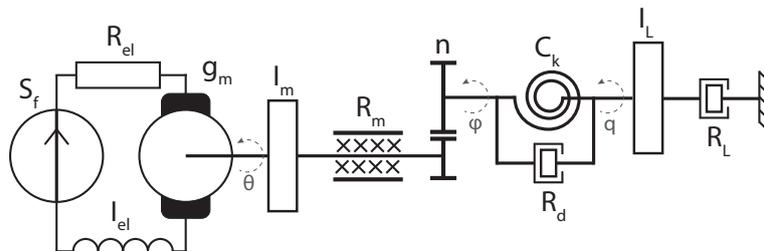


Figure 5.4: IPM of the actuator with a load.

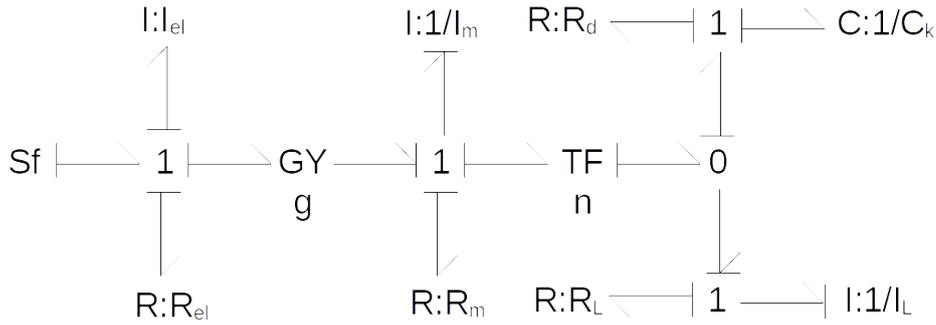


Figure 5.5: Bondgraph of the actuator with load.

5.2 State-space deduction

The transfer from the flow source (implemented in practice as a current controlled motor controller) to the force in the spring is valuable to deduce for controller development and modelling. As the LQR controller is a state based controller, having the transfer function in a state space system is useful for practical implementation. Especially if said states are directly measurable or otherwise deducible from the physical prototype. Otherwise, a state observer could be needed to deduce the required states from measurable states of the system. By deducing the state space system directly from the bondgraph, this can be ensured.

The statespace system of the actuator can be found in various different ways. The method highlighted here is the usage of Mason's rule to determine the transfer function of the fixed output bondgraph (Fig.5.2). Mason's rule is given by equation 5.2. Here the transfer function G is constructed from all N forward paths in the bondgraph, with G_k denoting the k-th forward path gain, and Δ_k the associated co-factor of the k-th forward path. Loops touching the forward path have been removed there. Δ denotes the entire graph determinant.

$$G = \sum_{k=1}^N \frac{G_k \Delta_k}{\Delta} \quad (5.2)$$

There are 2 loops in the bondgraph shown in figure 5.3 , these are both shown in equation 5.3.

$$L_1 = -\frac{1}{IKs^2}, L_2 = -\frac{1}{KR_s} \text{ where } K = \frac{I_m}{g^2}, I = \frac{g^2 C_k}{n^2}, R = \frac{g^2}{R_m + n^2 R_d} \quad (5.3)$$

The forward path can be found by following the causality of the bondgraph from the desired input to the output. If this is done for the input flow to the output flow of the spring element, Eq.5.4 is found. The forward path touches all loops, and thus the co-factor is unity.

$$P_1 = \frac{1}{KIs^2}, \Delta_1 = 1 \quad (5.4)$$

With this information the system determinant and forward path can be found. This is shown in Eq. 5.5.

$$\Delta = 1 - L_1 - L_2 = \frac{Ks^2 + \frac{1}{R}s + \frac{1}{I}}{Ks^2} \quad (5.5)$$

Now Mason's rule can be applied, which gives the transfer function from input current to torque in the elastic element.

$$H = \frac{\text{output}}{\text{input}} = \frac{P_1\Delta_1}{\Delta} = \frac{1}{KIs^2 + \frac{I}{R}s + 1} \quad (5.6)$$

This is a second order transfer function, which is expected with the given system because of the two energy storage elements. This equation captures the dynamics, but care has to be taken that the gain is compensated for as described by Eq.5.1.

To convert the transfer function to statespace, it can be written as in Eq.5.7, where the states are chosen as $X_1 = Y(s), X_2 = sY(s) = sX_1$.

$$G = \frac{Y(s)}{U(s)} = \frac{A}{Bs^2 + Cs + D}, A = \frac{g}{IKn}, B = 1, C = \frac{1}{RK}, D = \frac{1}{IK} \quad (5.7)$$

$$X_2s = X_1 \frac{-D}{B} + X_2 \frac{-C}{B} + U(s) \frac{A}{B} \quad (5.8)$$

This can be written into the direct state space equation form of

$$\begin{aligned} s\bar{X} &= \mathbf{A}\bar{X} + \bar{B}U(s), Y(s) = \bar{C}\bar{X} + DU(s) \\ \bar{X} &= \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}, \mathbf{A} \in \mathbb{R}^{2 \times 2}, \bar{B} \in \mathbb{R}^2, \bar{C}^T \in \mathbb{R}^2, D \in \mathbb{R} \end{aligned} \quad (5.9)$$

Substituting all the values yields the full state space model in equation 5.10.

$$s\bar{X} = \begin{bmatrix} 0 & 1 \\ -\frac{n^2}{I_m C_k} & -\frac{R_m + n^2 R_d}{I_m} \end{bmatrix} \bar{X} + \begin{bmatrix} 0 \\ \frac{g_m n}{I_m C_k} \end{bmatrix} U(s) \quad (5.10)$$

$$Y(s) = [1 \quad 0] \bar{X} \quad (5.11)$$

It is important to note that the states X_1 and X_2 are output torque and change in output torque over time respectively. These can both be deduced from the physical model. The spring deflection can be measured directly, and using Hooke's law the torque can be approximated. This means that no additional torque sensors are required for practical implementation.

The state space system is a linear model of the plant. It can be used for determining the LQR gains at a linearised point. By then varying the operating condition (spring stiffness) over the operating range the LQR gains can be determined at each operating point. This mapping is then used for implementation of the gain scheduling algorithm.

As for the non-linear statespace system, we can lift the terms over the transformer in Fig. 5.2, denote the non-linear spring as polynomial $\tau_{C_k} = p(\Delta)\Delta$, simplify the

motor to just a torque source $\tau_m = gU$, and proceed to write out the torque equation, with Δ denoting the spring deflection (for rotational domain, $\Delta = \alpha$):

$$I_m \frac{d^2 \Delta}{dt^2} = \tau_m - p(\Delta)n^2 \Delta - (R_m + n^2 R_d) \frac{d\Delta}{dt} \quad (5.12)$$

Choosing the states as:

$$\bar{Q} = \begin{bmatrix} \Delta \\ \frac{d\Delta}{dt} \end{bmatrix} \quad (5.13)$$

Allows us to rewrite the equation into a non-linear statespace system (Eq. 5.14 and Eq. 5.15). Note that the output should be compensated for transformer n.

$$s\bar{Q} = \begin{bmatrix} 0 & 1 \\ -\frac{n^2}{I_m p(\Delta)} & -\frac{R_m + n^2 R_d}{I_m} \end{bmatrix} \bar{Q} + \begin{bmatrix} 0 \\ \frac{g_m}{I_m} \end{bmatrix} U(s) \quad (5.14)$$

$$Y(s) = [p(\Delta)n \quad 0] \bar{Q} \quad (5.15)$$

5.3 Magnet parameter deduction

There is a trade-off between the area the magnets are allowed to occupy (β), and the free travel that the actuator can move (α). The magnet area occupied is determined by two factors, firstly the size per segment, secondly the amount of segments (magnet-pairs). The amount of magnet-pairs determines the size of the magnet segments and free travel. This is illustrated in figure 5.6.

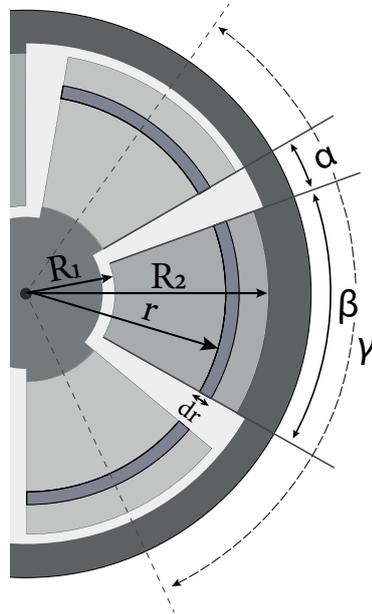


Figure 5.6: Top view diagram of the actuator.

The relationship between free-travel (α), magnet area (β) and number of segments (N) is described in Eq.5.16.

$$\Upsilon = 2\alpha + 2\beta = \frac{2\pi}{N} \quad (5.16)$$

Or rewritten this produces Eq. 5.17 which clearly shows the trade-off of magnet area (and thus actuator torque/stiffness) versus free travel.

$$\alpha = \frac{\pi}{N} - \beta \quad (5.17)$$

Although the previous analysis holds regardless of magnet type selected, care should be taken to make optimal use of the available area for the magnets. The magnets should be placed as far away from the center as possible, to optimally translate the repelling force to an actuator torque. This gives the first boundary condition. Also important is to consider that the volume (and thus for constant thickness, area) scales with magnet strength, which has an optimum. However, a magnet that is less long but more wide is located further away from the center as well, so this might counteract the reduction in strength from the lower volume of the magnet.

From the diagram in figure 5.7 we can deduce equations useful for finding the optimal area of the magnets.

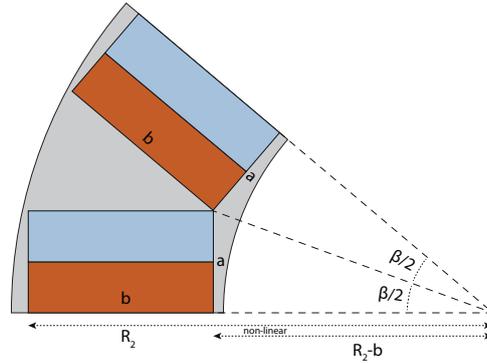


Figure 5.7: Schematic diagram of the placement of box magnets of size $a*b$ in a given segment.

$$a = (R_2 - b)\tan\left(\frac{\beta}{2}\right) \quad (5.18)$$

By multiplying the sides of the magnets, we can have an expression for the surface area of the magnet (eq. 5.19). This describes a curve from $b = 0$ to $b = R_2$. This has an optimum which can be found by equating its derivative shown in 5.20 to zero. This is only true for all β iff $b = \frac{R_2}{2}$, implying that the optimal magnet surface area occupation is always achieved if the length of the magnet is half the radius.

$$A = \tan\left(\frac{\beta}{2}\right)(bR_2 - b^2) \quad (5.19)$$

$$\frac{\delta A}{\delta b} = \tan\left(\frac{\beta}{2}\right)(R_2 - 2b) \quad (5.20)$$

As discussed before, this gives the optimal surface area, but not necessarily the optimal torque.

5.4 Magnet model derivation

Modelling magnet repelling force is analytically impossible. It is however possible to attain a magnet model with data acquired experimentally or through simulations. For this paper both simulated and empirical data is shown. For further development of the model it was chosen to use empirical data, as this was deemed to result in the most accurate predictions for a practical elastic element.

The magnet repelling force could be found with own experimental data, or with data publicly available online at various places. In particular "The K&J Repelling Magnet Calculator" [25] is used. Specific magnet size depends on the chosen physical actuator dimensions, torque/stiffness requirements as well as local availability of magnet sizes. Quoted repelling forces of local manufacturers might differ from the maximum repelling force given by the data from K&J Calculator. To compensate for this, the data is scaled proportionally to the maximum repelling force quoted by the manufacturer. Fig.5.8 shows the magnet repelling data of the chosen 30mm x 12mm x 12mm N52 box magnets, compensated for the maximum quoted repelling force. Additionally it shows simulated repelling force for the same size magnets.

The difference between the simulated data and empirical data in Fig.5.8 can be explained by the strength of the magnets as quoted from the manufacturer is lower than predicted by the K&J Repelling Magnet Calculator. As the torque results of the model do match up with measured torque results, it seems that the empirical magnet curve in Fig.5.8 is accurate. The manufacturer quoted magnet strength is approximately two times lower than the result from both simulation and K&J Repelling Magnet Calculator. As the exact cause of this difference is not of importance to the paper and its conclusions, this was not investigated further.

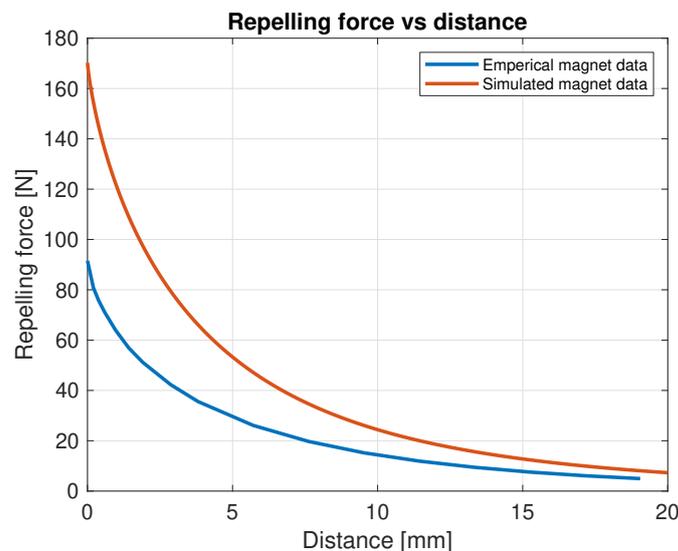


Figure 5.8: Estimated box magnet repelling data of a 30mm x 12mm x 12mm N52 box magnet.

To reiterate, this is the linear repelling force of a single pair of magnets. Because they are oriented radially, and more than one pair is present, further calculations

are needed to estimate the presented torque of the elastic element.

As the magnet data is generally up to a maximum distance, a polynomial fit can be used to extrapolate beyond the furthest measured distance.

5.5 Spring model derivation

To derive the mathematical model for converting the magnet data into expected actuator torque and stiffness, the magnet is split up into small magnet subsections of width δr (see Fig. 5.6 for a visual representation). The torque of one segment is scaled relative to the total size of the magnet, by using a ratio between the segment area, and the total magnet area. The torque is then found for a segment with equation 5.21.

$$\tau_{segment} = \frac{A_{seg}}{A_{total}} r F_{magnet}(\alpha, r) \quad (5.21)$$

Now these segments can be summed over the total length of the magnet. The length is defined as the distance from R_1 to R_2 . The factor N is used for the amount of magnet pairs in the actuator.

$$\tau_{magnet}(\alpha) = N \int_{r=R_1}^{R_2} \frac{A_{seg}}{A_{total}} r F_{magnet}(\alpha, r) dr \quad (5.22)$$

The implementation of A_{seg} , A_{total} and $F_{magnet}(\alpha, r)$ depend on the particular magnet and magnet geometry chosen.

For a box magnet, equation 5.23 holds. Note that w is the width of the box magnet.

$$A_{seg,box} = w \Delta r, A_{total,box} = w(R_2 - R_1) \quad (5.23)$$

For the alternative of arc magnets, these equations are slightly more complicated.

$$A_{seg,arc}(l) = \frac{\beta}{2}(r_{outer}^2 - r_{inner}^2) = \frac{\beta}{2}(4l\Delta r), A_{total,arc} = \frac{\beta}{2}(R_2^2 - R_1^2) \quad (5.24)$$

Finally the repelling force experienced by the magnet can be deduced by translating the linear force found at the beginning of this section.

$$F_{magnet}(\alpha, r) = F_{lin}(2r \sin(\frac{\alpha}{2})) \quad (5.25)$$

The above equations can then be substituted to produce the final torque modelling equations for both box (eq. 5.26) and arc segment (eq.5.27) magnets.

$$\tau_{box}(\alpha) = \frac{N}{R_2 - R_1} \int_{r=R_1}^{R_2} r F_{lin,box}(2r \sin(\frac{\alpha}{2})) dr \quad (5.26)$$

$$\tau_{arc}(\alpha) = \frac{4N}{R_2^2 - R_1^2} \int_{r=R_1}^{R_2} r^2 F_{lin,arc}(2r \sin(\frac{\alpha}{2})) dr \quad (5.27)$$

Equation 5.26 has been plotted using the parameters of the physical model. The magnet data in section 5.4 is used. Using section 5.3 the parameters for the

geometric layout of the actuator are determined. With iteration and optimization the final values are used in the final model of the actuator:

- $N = 3$
- $R_1 = 0.03$ [m]
- $R_2 = 0.06$ [m]
- Magnet volume = $0.03 \times 0.012 \times 0.012$ (length x width x height) [m^3]
- $\alpha = 10$ [°]

This results in the graph shown in figure 5.9. The stiffness is found by taking the derivative of the torque model.

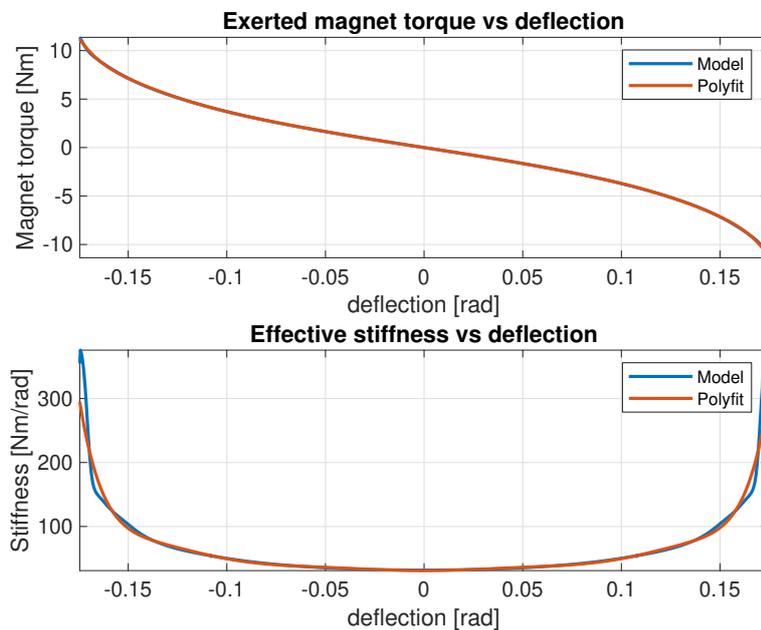


Figure 5.9: Torque and stiffness of the actuator as predicted by the model.

To implement the spring model into the modelling software (20-sim), it was chosen to make a polynomial fit to the model data. Care has to be taken that the polynomial does not only closely match the torque curve, but that the derivative of the polynomial matches the stiffness curve as well. For the particular magnet and parameters, a 12th degree polynomial fit was deemed sufficient. The polynomial curves are shown in figure 5.9 as well. The coefficients from highest to lowest degree are:

-7777726148.81284
 $1.88090695304036 * 10^{-5}$
 520096631.667956
 $-1.52547756579681 * 10^{-6}$
 -12978699.0718667
 $4.21088871109877 * 10^{-8}$
 132015.020444607
 $-4.54847697819680 * 10^{-10}$

-1068.66936346445

$1.68684818995270 * 10^{-12}$

-30.9627709660632

$-9.64177956698911 * 10^{-16}$

In 20-sim the integral of flow and effort are used as the states of models (integral causality). For a 'C' element this implies that the output state is dependent on the integral of the flow, which is the deflection. This makes it trivial to implement the above polynomial in a C element.

The derivative of this polynomial can be found by hand, this maps the deflection of the spring to stiffness. This is useful for actual implementation of the LQR controller. In the case of this non-linear series elastic actuator the gain scheduling is done on the changing stiffness.

Chapter 6

Test setups

Under this chapter, a variety of measurement setups are talked through from concept to design. Design decisions are motivated as to give an understanding how the result came to be. These test benches are intended to be used to evaluate the produced prototype and concept control system. The chapter is split to discuss first the 'static test setup', which is used to evaluate the static characteristics of the non-linear elastic element such as torque versus deflection and stiffness versus deflection curves. This is then followed by an overview of the dynamic test setup, which is intended to be used in complete non-linear series elastic actuator performance. This would integrate the produced magnet-based elastic element, the gain-scheduled controller, electronics and a motor with gearbox. The motor and gearbox suffered delays in shipping, making it impossible to do practical evaluation of the entire NSEA design.

6.1 Static test setup

For measuring the designed prototype elastic element, a test bench has been designed and developed. This is to characterize how the element behaves, and does not include measurements with the control system or other actuator components (such as the motor).

6.1.1 Requirements

Proper working of the test bench is crucial in evaluating actuator performance. Most of the requirements are non-quantifiable, and thus are summed up here.

- Modular, allow various sizes of actuators with various sensors
- Construction with rapid prototyping techniques and materials
- Make use of low-cost sensors
- Be relatively portable
- Able to measure a deflection range of ± 20 degrees
- Allow for quick, easy and repeatable measurements

Some of the requirements arise from enabling rapid prototyping and testing cycles, and reducing/eliminating unnecessary delays when prototypes or dimensions change. Others come from the complications of global COVID-19 lockdown, meaning that most testing has to be done at multiple locations outside the lab and at home.

6.1.2 Mechanical design

Various mechanical designs for the test bench has been thought of. In general, the mechanical design falls into one of two categories:

1. Applying a known torque, and measuring the deflection
2. Deflecting the element, and measuring the produced torque

Which of these is more advantageous is dependent on the design and the design requirements. The difference is purely which sensors are needed, and how the element is excited.

The advantage of the first method is that measuring the deflection can be done quite trivially and cheaply with a scale. It does however require either calibrated weights or for the user to measure the weight applied every time a new measurement takes place. An example of how this could be executed is shown in figure 6.1. It shows the elastic element under test (DUT) with an arm attached of length L_{tot} . A known mass M is used to generate a force F_g , which is translated through a string and pulley to generate a known force F_{arm} onto the arm. This in turn gives a known torque applied to the device of $\tau = F_g L_{tot}$. Note that for this measurement setup, care has to be taken to put make sure the string is perpendicular to the arm. The amount the arm/elastic unit has to be rotated is then equal to the deflection to generate this torque.

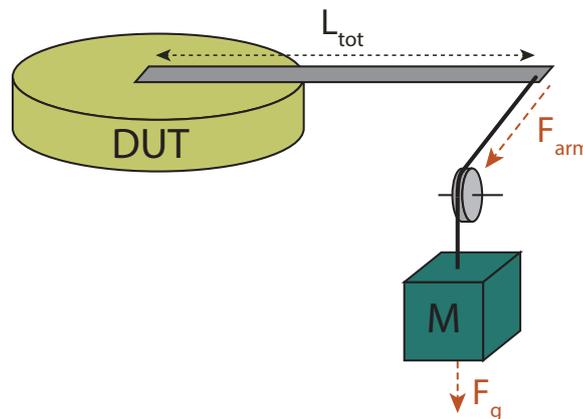


Figure 6.1: Diagram illustrating an example of executing the first method.

The main advantage of the second method is that deflecting the element a known distance is easy. The disadvantage is that torque measurement is more complicated than deflection measurement. This way of doing it does not require any additional

external masses/weights. Figure 6.2 shows an example of how this can be implemented. It features a DUT, and an arm attached to this DUT. One side of the DUT is fixed, whilst the part attached to the arm is deflected by fixing the other end of the arm at varying distances along a slot. The slot is axially centered around the same point as the DUT. The torque can be measured with a load-cell attached in the middle of the arm, which measures the bending moment. This bending moment relates to the torque the actuator produces, and thus the torque at a certain known deflection can be measured.

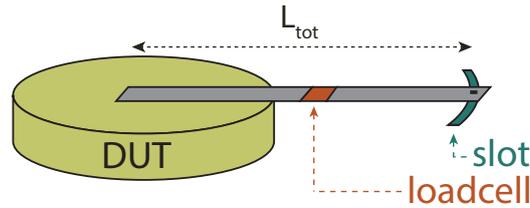


Figure 6.2: Diagram illustrating an example of executing the second method.

An overview of the methods and how they would affect various requirements is shown in table 6.1.2. As quick and repeatable measurements, and not needing expensive/additional equipment are important requirements, method 2 was chosen to develop further into a prototype.

Table 6.1: Requirement comparison of static test bench methods

Requirement	Method 1	Method 2
<i>Modularity</i>	0	0
<i>Material choice</i>	0	0
<i>Measurement range</i>	0	0
<i>Portability</i>	-	+
<i>Sensor cost</i>	+	-
<i>Ease of doing measurements</i>	-	++
Total	-1	2

Figure 6.3 shows an IPM of the second measurement method. It shows a beam with an applied torque τ , and a load-cell indicated by the orange block, at location L_k from the center of rotation. The derived bending moment diagram is shown below the IPM. This can be relevant for deriving the torque with a load-cell. Equation 6.1 can be used to derive the applied torque τ from a bending moment M_k measured at position L_k from the applied torque.

$$\tau = \frac{L_{tot}}{L_k} M_k \quad (6.1)$$

6.1.3 Sensors

As the element is deflected a known amount, and torque should be measured, care should be taken as to select an appropriate sensor. This holds for both verifying

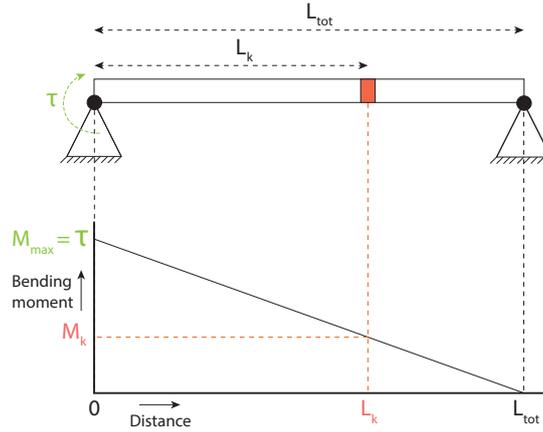


Figure 6.3: IPM diagram of the second method, including associated bending moment over the beam for an applied torque.

the deflection is accurate, as well as the measurement of the torque induced by the elastic element.

Various different measurement methods can be used to estimate/measure the deflection of the element. They vary greatly with complexity, cost and accuracy. Table 6.2 shows a comparison between the most common methods of indicating/sensing deflection.

Table 6.2: Comparison of various deflection sensing methods

Requirement	Scale	Accelerometer	Potentiometer	Rot. enc.
<i>Mechanical complexity</i>	++	-	—	-
<i>Cost</i>	++	+	+	+
<i>Measurement accuracy</i>	-	+	++	++
<i>Flexibility</i>	+	-	+	+
Total	4	0	1	2

Because accuracy for this measurement is not very important (because the deflection is fixed by the test unit), and cost and complexity are more important, it is chosen to integrate a scale into the test bench unit. This makes it possible to quickly and easily read out the current deflection, and offers enough precision to know the current deflection.

There are only two primary ways of sensing torque, which is an integrated force/torque sensor made available by the university (The *ATI Mini40 F/T sensor* [27]) and a load-cell measuring the torque through an arm. Table 6.1.3 shows a comparison between both devices.

The F/T sensor seems ideal, easy to implement, accurate measurements and no additional hardware that needs to be developed. However, it has two primary drawbacks. Firstly, the sensor unit is very expensive. This makes it not possible to do testing in other locations than in the lab, and also requires the sensor to be reserved before usage. This severely constricts the ability to do testing. Additionally, the sensor has a very limited acceptable torque range. The expected torque range of the

Table 6.3: Comparison of various torque sensors

Requirement	Load-cell	F/T sensor
<i>Mechanical complexity</i>	0	++
<i>Cost</i>	+	—
<i>Measurement accuracy</i>	++	+++
<i>Measurement range</i>	+	-
<i>Flexibility</i>	+	+++
Total	5	4

developed prototype exceeded the limitations of the sensor, thus requiring another solution.

The load-cell offers a nice solution for a low-cost, reliable sensor. They are available in a large variety of different shapes and sizes, for any range of torques. The main drawback is that speciality electronics need to be developed to read out the values, and that the load-cell needs to be calibrated before use.

It was decided to allow both sensor to be simultaneously be used. The F/T sensor provides a 'ground truth' measurement which can subsequently be used to calibrate the load-cell. After calibration the load-cell can be used to measure the full torque range of the actuator. It can be verified that the load-cell has sufficient accuracy for the measurements, and testing can be done outside of the lab without F/T sensor if need be.

To read out the sensor electronics, as previously discussed in the paper, a combination of the *HX711* load-cell IC and *Frdm-K64F* have been used. The micro-controller runs a custom firmware to convert load-cell voltages to torques with a calibration curve, and to subsequently log these values to a PC over a USB serial link. The data is saved in a comma-separated value format.

The calibration curve is made by doing 4 separate measurements and finding a calibration curve for each of these. This curve is then averaged for a calibration curve. Between curves the maximum error at +-10Nm is approximately 3-10%. This is deemed to be sufficient, and it is expected that in measurements other factors will dominate the error from run to run.

Figure 6.4 shows that indeed the F/T torque value scales linearly with the load-cell voltage read out. Note that the F/T sensor and the load-cell voltage both requires a zero point calibration. This can be done at all times by feeling that the elastic element is in the zero position and not producing any torque, and taking that measurement as the zero point.

6.1.4 Materials

Material selection is important for the design of the test bench. As stated in the requirements, rapid-prototyping is desired as to ensure sufficient iterations on the design can be done.

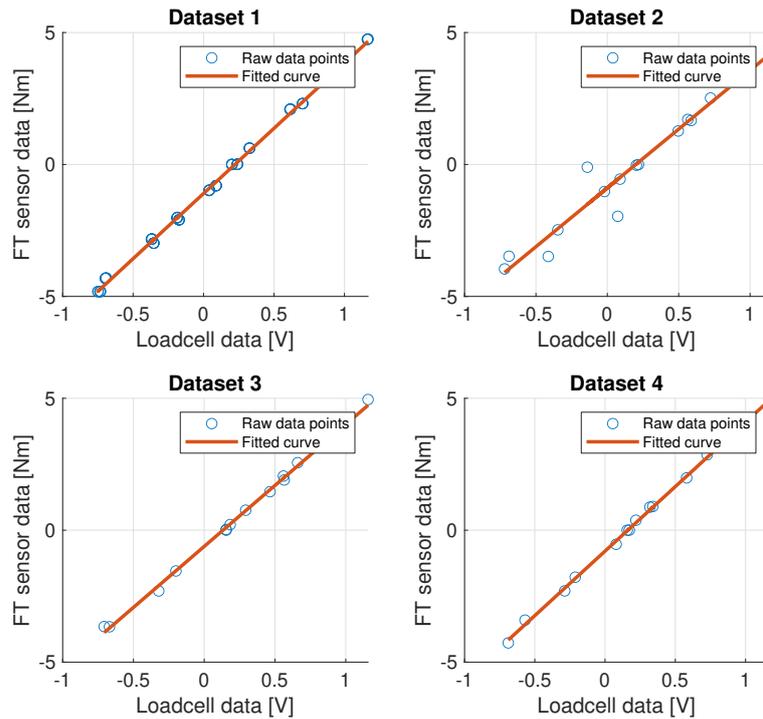


Figure 6.4: F/T torque measurement versus load-cell voltage.

To aid in rapid prototyping the RAM lab has various rapid-prototyping methods available. The most attractive are lasercutting and 3D printing. Due to the size of the test bench, 3D printing is the less attractive of the two options. It is relatively costly and takes much more time than constructing out of laser cut materials.

The preferred construction technique is to stack laser cut plates. This ensures a very rigid construction whilst being very quick and cheap to produce. By using a plate structure, additional plates of varying thicknesses can easily be added to accommodate sensor + actuator combinations of varying heights without making any new test bench. This significantly increases the flexibility of the designed testbench, and makes it possible to test combinations of essentially infinite height. The only constraint is the width of the actuator, but this is already constrained by the 3D printer printing dimensions.

As for the plate material, this is relatively unimportant. It should be stiff enough not to deform whilst testing, but not so stiff to crack when handled roughly. The three main choices are (ply) wood, acrylic and POM. Wood is not dimensionally stable under different temperature and humidity conditions. Acrylic is available only in a small variety of thicknesses, and cracks relatively easily when stressed by for example tightening bolts. POM lasercuts cleanly, is readily available in 1, 3, 5mm sheets and is strong. Additionally, unlike other plastics it can be shaped and modified easily using hand tools and does not melt from friction as easily.

6.2 Design iterations static test setup

The test bench went through various iterations, which are presented here, including reasoning as to why it was modified or changed.

Throughout the development, the core design remained mostly the same. The static test setup consists of a base plate, which attaches the sensor/elastic element. The walls surrounding the elastic unit/sensor are build up using laser cut plates. The top plate has a scale etched into it for indication of deflection, as well as a track through which a pointer can be fixed. The pointer is fixed to the 'output' of the elastic unit, whilst the other end of the pointer can be fixed at the various positions in the top plate. This is done with a nut and bolt.

6.2.1 Version 1

Figure 6.5 shows the first iteration of the test bench. At this point it was not yet decided to use a load-cell for torque measurements. The cut-away view gives a clear illustration of how the F/T sensor and elastic element fit inside the unit.

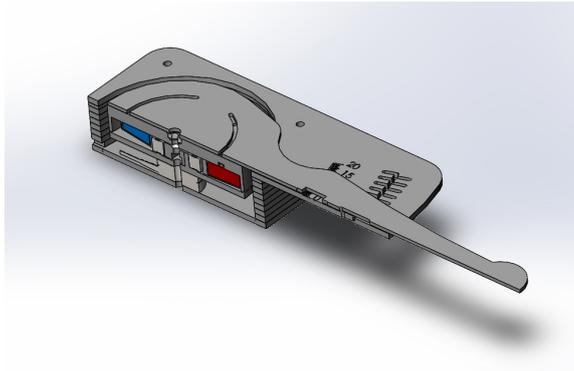


Figure 6.5: First iteration of the static test bench, cut-away view.

Figure 6.6 shows the setup being used during measurements.



Figure 6.6: First iteration of the static test bench, during measurements.

6.2.2 Version 2

Version 2 of the test bench features the inclusion of a load-cell to extent the torque range that can be handled during measurements.



Figure 6.7: Second static test bench iteration CAD model.



Figure 6.8: Second static test bench iteration during testing.

Up until this version, the pointer is fixed to the top plate in a cascading set of indents. What was found during measurements is that these effectively change the length of the arm, thus affecting the measurements.

6.2.3 Version 3

The last iteration of static test benches, version 3, solved two issues with the measurements:

- The cascading changing the effective arm length, causing 'stair stepping' in the measured data
- Tightening the bolt to fix the pointer causes torque to be induced into the arm, affecting measurements

The first problem is solved by using a linear set of indices, instead of cascading indentations. The second one was solved with the addition of a 3D printed bearing holder. This holder effectively decouples the pointer from the bolt used to tighten the pointer to the top plate. The parts making up the bearing holder are printed using PLA.

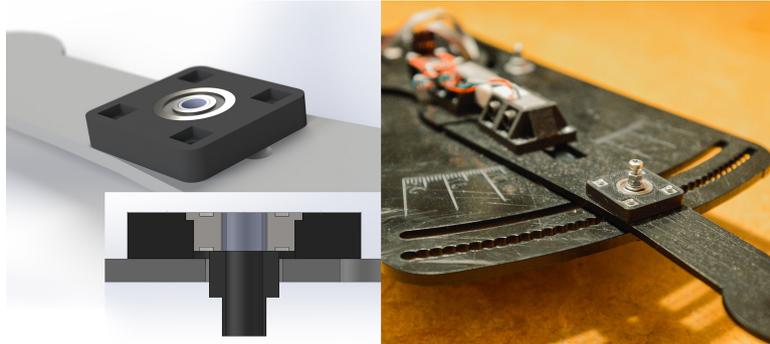


Figure 6.9: Bearing unit. Note the the build in spacer to ensure sufficient distance between the pointer and the top plate to avoid contact.

It should be remarked that there can still be non-idealities in the setup. Most significant is misalignment between the axis of rotation of the test bench, and the elastic unit. When the pointer is fixed whilst these are not aligned, it can cause other forces and torques in the pointer and thus load-cell. This is likely the cause of asymmetrical data of the measured elastic element.

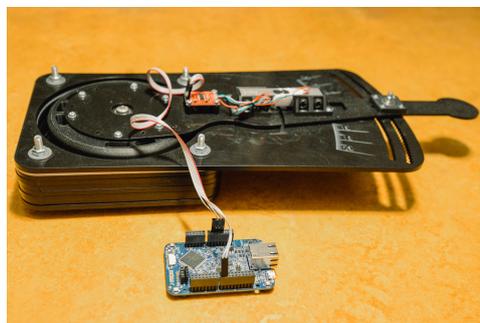


Figure 6.10: Version 3, the final version of the static test bench.

6.3 Dynamic test setup

Due to time constraints and delays in manufacturing and delivery of certain components, applying the control loop to the manufactured proof-of-concept prototype for dynamic testing could not be performed. However, a multifunctional test bench has been developed and constructed for future testing.

6.3.1 Requirements

Like the static test unit, the dynamic test bench also has requirements. These are in a large part similar.

- Modular, allow various sizes of actuators with various sensors
- Construction with rapid prototyping techniques and materials
- Have room for off-the-shelf electronics for implementation of the control loop
- Be relatively portable
- Sturdy to resist forces/torques applied
- Allow for quick, easy and repeatable measurements
- Allow for the actuator to be tested both fixed output, as well as with a load
- Allow for the attachment of various sensors

6.3.2 Mechanical design

The test bench is essentially a frame to which a motor is attached with driver electronics. The device-under-test (DUT) can then be attached to said motor, and to a sensor on the other side to be fixed. It can alternatively also be attached to a load.

6.3.3 Materials

Identical to the test bench for static tests, the dynamic test bed is relatively large, and thus similar constraints result in a similar material choice. The bulk of the device is formed by laser cut POM sheets. These are connected together with 3D printed nylon fixtures and bolts. This ensures a very rigid and robust design, whilst remaining flexible, easy to build and expand. Additionally, acrylic sheets are used as blast shields to protect the environment from sudden unexpected deconstruction during testing.

Another alternative would be to use aluminium extrusions which are widely available. However, this would have made the design significantly heavier. It also would have increased the cost of the design.

6.3.4 Prototype iterations and overview

An overview of the designed dynamic test bench is shown in figure 6.12. It can be seen to consist of 3 distinct units. The first is an electronics and motor housing unit, the second a unit for the device under test, and the third for an additional load. Each of the units is separated by a wall. This isolation between different units ensures that if failure is contained and damage of said failure is limited. There is a hole available in the walls for the pass-through of cables for measurement or power. A large hole in the wall is meant for attaching different kinds of inserts depending on what needs to be achieved. This makes it flexible to attachment of various sensors as well.

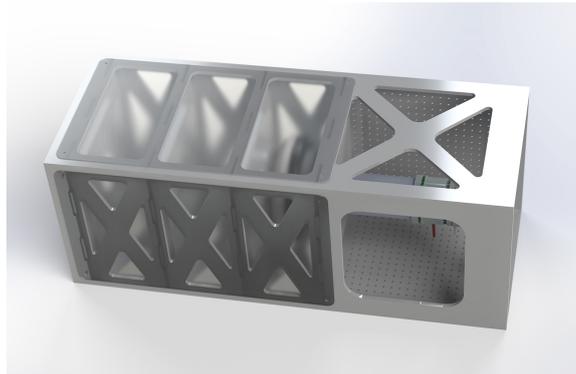


Figure 6.11: First iteration of the dynamic test bench.

Unit 1 houses mostly electronics, and has a grid of 3mm holes which can be used, in combination with stand-offs to mount any electronics necessary. There's a specific layout available to mount a RAMStix board. The sides of this unit are unconstricted and not shielded by acrylic on purpose, to allow proper airflow and cooling of the electronics and motor.

Unit 2 is meant for containing the device under test. It is attached to the motor on one side through a special link. The other side of the device can either be fixed to the wall (optionally through a sensor) or attached to a bearing. The bearing can pass-through the element output to the last unit, or be used to suspend the output of the device under test with as little resistance as possible. This construction makes it possible for the device under test to be set up in a large variety of scenarios for most possible tests.

The last unit is where an optional load can be attached. No tests had been planned, so this was mostly kept for possible future use. Alternatively, for a large device under test it can be used to expand the space that houses the device that needs to be tested.

Figure 6.11 shows the first iteration of the test bench. It was designed with finger joints and to be glued together.

The main changes to the second version, is shortening the overall chassis length. This is done by removing the load-unit, which was deemed unlikely to be used in the first tests, and would save material whilst iterating on other versions. Also, instead of glueing the plates together with epoxy, it was chosen to attach them together

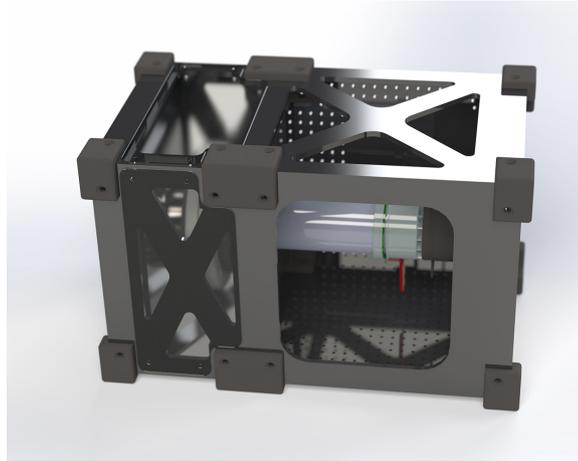


Figure 6.12: Second iteration of the dynamic test bench.

with 3D printed brackets and bolts.

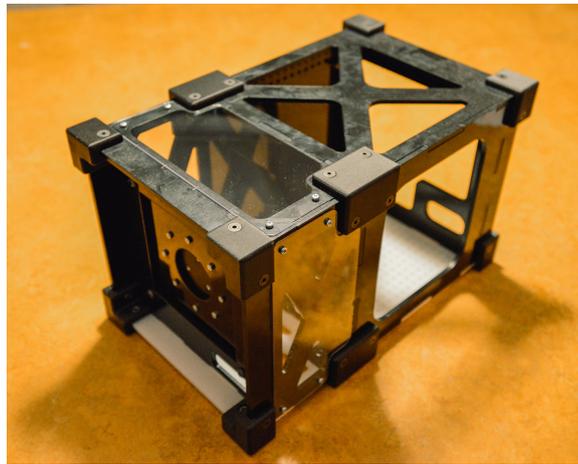


Figure 6.13: Build prototype of the second iteration.

This test bench was manufactured (as shown in figure 6.12 and 6.13) as to be used for dynamic testing. Several delays in manufacturing of the ordered motor combination meant that this setup could not be used for actual measurement. Additional measurements are planned to be done with the unit.

Chapter 7

Conclusion

The main conclusions of the research have already been given in the core paper (Chapter 3), therefore this section will focus on the conclusions with respect to the requirements stated under chapter 2, and chapter 6.

7.1 Non-linear elastic element

Firstly the MoSCoW requirements of the non-linear series elastic element have all been satisfied, with the exception of the "won't haves" and performance measurements with motor and controller. This last requirement could not be met due to delays in motor shipment.

As for the physical dimensions of the produced actuator, they are shown in table 7.1 together with initial design specifications. It is clear that all initial design parameters of the elastic element have been achieved.

Table 7.1: Overview of initial quantifiable design goals for the elastic element and produced element results

Parameter	Desired	Proof-of-concept prototype
Displacement [rad]	0.08 - 0.27	0.1745
Torque range [Nm]	+ -10	+ -10
Minimum stiffness [$\frac{\text{Nm}}{\text{rad}}$]	25	40
Stiffness dynamic range [x]	2	3 (40-120 [$\frac{\text{Nm}}{\text{rad}}$])
Maximum diameter [mm]	132	132
Maximum thickness [mm]	50	39.5
Maximum weight [gr]	800	565

7.2 Gain-scheduled control system

As for the requirements for the control system, all MoSCoW requirements are satisfied except for the ones that need actual hardware (which are the "could haves". All results have thus far been done in simulations. Because the research was focussed on finding whether gain-scheduling has advantages compared to traditional linear control, no initial quantifiable requirements had been set.

7.3 Test benches

For both the static and the dynamic test-bench, the requirements stated have been met. The static test-bench has shown to be able to deflect the elastic element to its stiffness extremes, and be able to measure the whole torque range. The dynamic test bench could not be used for testing yet, but initial inspections show that the design is very rigid and modular.

Bibliography

- [1] Zinn, M., Khatib, O., Roth, B. *A New Actuation Approach for Human Friendly Robot Design*. IEEE International Conference on Robotics and Automation, ICRA '04. 2004. doi:10.1109/robot.2004.1307159
- [2] Pratt, G.a., and M.m. Williamson. *Series Elastic Actuators*. Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots, doi:10.1109/iros.1995.525827.
- [3] Pratt, Gill A., et al. *Stiffness Isn't Everything*. Experimental Robotics IV Lecture Notes in Control and Information Sciences, 30 June 1995, pp. 253–262., doi:10.1007/bfb0035216.
- [4] Roozing, W., Malzahn, J., Kashiri, N., Caldwell, D. G., Tsagarakis, N. G. (2017). *On the Stiffness Selection for Torque-Controlled Series-Elastic Actuators*. IEEE Robotics and Automation Letters, 2(4), 2255-2262. doi:10.1109/lra.2017.2726141
- [5] Robinson, D., Pratt, J., Paluska, D., Pratt, G. (1999). *Series Elastic Actuator Development for a Biomimetic Walking Robot*. 1999 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (Cat. No.99TH8399). doi:10.1109/aim.1999.803231
- [6] Zhang, Juanjuan, and Steven H. Collins. *The Passive Series Stiffness That Optimizes Torque Tracking for a Lower-Limb Exoskeleton in Human Walking*. Frontiers in Neurorobotics, vol. 11, 2017, doi:10.3389/fnbot.2017.00068.
- [7] Morita, T., Sugano, S. (1995). *Design and development of a new robot joint using a mechanical impedance adjuster*. Proceedings of 1995 IEEE International Conference on Robotics and Automation. doi:10.1109/robot.1995.525630
- [8] Choi, J., Park, S., Lee, W., Kang, S. (2008). *Design of a robot joint with variable stiffness*. 2008 IEEE International Conference on Robotics and Automation. doi:10.1109/robot.2008.4543455
- [9] Choi, J., Hong, S., Lee, W., Kang, S. (2009). *A variable stiffness joint using leaf springs for robot manipulators*. 2009 IEEE International Conference on Robotics and Automation. doi:10.1109/robot.2009.5152716
- [10] Migliore, S., Brown, E., Deweerth, S. (n.d.). *Biologically Inspired Joint Stiffness Control*. Proceedings of the 2005 IEEE International Conference on Robotics and Automation. doi:10.1109/robot.2005.1570814
- [11] Malzahn, J., Barrett, E., Tsagarakis, N. (2019). *A Rolling Flexure Mechanism for Progressive Stiffness Actuators*. 2019 International Conference on Robotics and Automation (ICRA). doi:10.1109/icra.2019.8794004

- [12] Thorson, I., Caldwell, D. (2011). *A nonlinear series elastic actuator for highly dynamic motions*. 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems. doi:10.1109/iros.2011.6094913
- [13] Austin, J., Schepelmann, A., Geyer, H. (2015). *Control and evaluation of series elastic actuators with nonlinear rubber springs*. 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). doi:10.1109/iros.2015.7354315
- [14] Schepelmann, A., Geberth, K. A., Geyer, H. (2014). *Compact nonlinear springs with user defined torque-deflection profiles for series elastic actuators*. 2014 IEEE International Conference on Robotics and Automation (ICRA). doi:10.1109/icra.2014.6907350
- [15] Schmit, N., Okada, M. (2012). *Design and Realization of a Non-Circular Cable Spool to Synthesize a Nonlinear Rotational Spring*. *Advanced Robotics*, 26(3-4), 234-251. doi:10.1163/156855311x614545
- [16] Realmuto, J., Klute, G., Devasia, S. (2015). *Nonlinear Passive Cam-Based Springs for Powered Ankle Prostheses*. *Journal of Medical Devices*, 9(1). doi:10.1115/1.4028653
- [17] Axelsson, P., Pipeleers, G., Helmersson, A., Norrlöf, M. (2014). *H_∞ Synthesis Method for Control of Non-linear Flexible Joint Models*. *IFAC Proceedings Volumes*, 47(3), 8372-8377. doi:10.3182/20140824-6-za-1003.00142
- [18] Montanaro, M., Wang, Z., Fallah, S., Sorniotti, A., Lenzo, B. (2018). *H_∞ A gain scheduled robust linear quadratic regulator for vehicle direct yaw moment Control*. *Control. Mechatronics*, 51, 31-45. doi:10.1016/j.mechatronics.2018.01.013
- [19] Kedjar, B., Al-Haddad, K. (2006). *LQR with Integral Action for Phase Current Control of Constant Switching Frequency Vienna Rectifier*. 2006 IEEE International Symposium on Industrial Electronics, Montreal, QC, Canada, 2006, pp. 1461-1466, doi: 10.1109/ISIE.2006.295687.
- [20] Rocha, R., Silvino J.L., Resende, P. (2012). *LQR with integral action for the control of variable speed induction generator connected to AC grid*. 1061-1066. 10.1109/IECON.2012.6388573.
- [21] Silva, M. J., Sanches, A. O., Malmonge, L. F., Malmonge, J. A. (2014). *Electrical, mechanical, and thermal analysis of natural rubber/polyaniline-DBSA composite*. *Materials Research*, 17(Suppl 1), 59-63. doi:10.1590/s1516-14392014005000065
- [22] Bicchi, A., Rizzini, S., Tonietti, G. (2001). *Compliant design for intrinsic safety: General issues and preliminary design*. *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*. (Cat. No.01CH37180). doi:10.1109/iros.2001.976345
- [23] English, C., Russell, D. (1999). *Implementation of variable joint stiffness through antagonistic actuation using rolamite springs*. *Mechanism and Machine Theory*, 34(1), 27-40. doi:10.1016/s0094-114x(97)00103-1

- [24] Schuy, J., Beckerle, P., Wojtusich, J., Rinderknecht, S., Von Stryk, O. (2012). *Conception and evaluation of a novel Variable Torsion stiffness for biomechanical applications*. 2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob). doi:10.1109/biorob.2012.6290778
- [25] K&J. Magnetics Inc., *Magnet repelling force data for box-type magnets*, <https://www.kjmagnetics.com/calculator.repel.asp?calcType=block>, Dec. 2020.
- [26] Mathworks, *Matlab - Linear-Quadratic Regulator (LQR) design*, <https://uk.mathworks.com/help/control/ref/lqr.html>, Jan. 2021.
- [27] ATI Industrial Automation, *F/T Sensor: Mini40*, https://www.atia.com/products/ft/ft_models.aspx?id=Mini40, Jan. 2021.
- [28] G2Elab, *MacMMems : Macro Modeler for Magnetic Memes*, <https://g2elab.grenoble-inp.fr/fr/recherche/macmmems-macro-modeler-for-magnetic-mems>, Jan. 2021.
- [29] Sparkfun Electronics, *Load Cell - 10kg, Straight Bar (TAL220)*, <https://www.sparkfun.com/products/13329>, Jan. 2021.
- [30] Sparkfun Electronics, *SparkFun Load Cell Amplifier - HX711*, <https://www.sparkfun.com/products/13879>, Jan. 2021.
- [31] Markforged, *Markforged Mk Two - Product page*, <https://markforged.com/3d-printers/mark-two>, Jan. 2021.

This document was designed with double sided printing in mind to save the environment.

For this purpose this page is intentionally left blank

Appendix A

Literature analysis

A.1 Analysis of the non-linear series elastic actuator field

This section will go into more detail about the existing literature of both non-linear series elastic actuators, as well as control in these actuators which has not been mentioned in the paper.

A.2 Non-linear series elastic actuators

Figure A.1 is helpful in visually interpreting what the landscape of low stiffness actuators looks like, where NSEAs fit in this landscape, and specifically this research fits. This is by no means an exhaustive list, but gives an idea of the design space and general research direction.

Most papers involving design and construction of actuators can be categorized in this way. They are shown as end-nodes in Fig.A.1, and listed below. The list below also has references to various research for these classes of actuators.

Although not being the same, Variable Stiffness Actuators still exhibit much of the same design challenges as NSEAs. These devices can adjust their stiffness with a secondary actuator. They can be categorized in the following three ways of generating and adjusting non-linear stiffness:

- Antagonized-based VSAs - [10, 22, 23]
- Mechanical impedance adjustment based VSAs - [7, 9, 24]
- Material based VSAs - [8]

As for research into non-linear stiffness elastic actuators, these too can be divided in three main ways of generating non-linearity in their stiffness:

- Mechanical based, with linear spring and cam - [12–16]
- Structure controlled stiffness based - [11]
- Material based NSEAs - No previous research

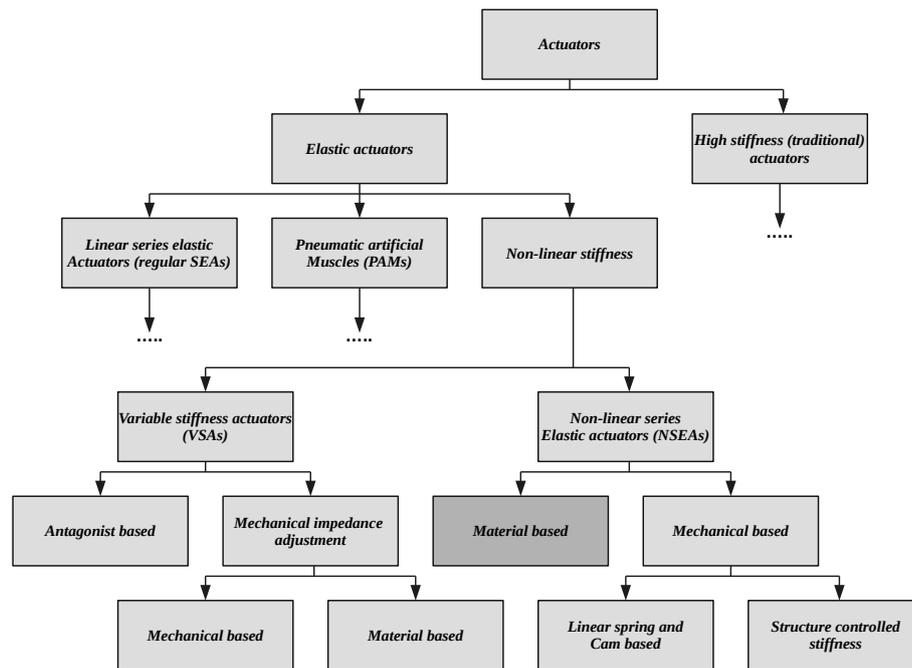


Figure A.1: Overview of various kinds of actuators. The topic of this research is highlighted in dark grey.

It is clear from this overview that the majority of work for NSEAs has been done with linear springs which are non-linearly deflected with a cam mechanism of some sorts. No NSEA designs based on inherent non-linear material properties have been presented so far. Indeed, most papers focus on the application of such actuators, and not novel design methods. This highlights one of the main benefits of a material based NSEA unit, namely the lack of a complex mechanical cam system. This reduces complexity thus increasing reliability, reducing cost and increasing design flexibility.

Appendix B

Code

B.1 LQR Controller Matlab code

The purpose of this code is to analyse the (linear) state-space system, and how different stiffnesses affect LQR parameters. It also calculates the appropriate LQR gains which are used for gain-scheduling.

```
close all;
clear;

%% Parameters and user config
Rel      = 3.44;          % Electrical resistance of motor, only useful for power consumption and efficiency

g        = 0.0525;       % Motor torque constant
n        = 16/225;       % Gearbox transfer ratio

Rmotor   = 0.00015;     % Damping of motorbearings
Routp    = 0.00015;     % Damping of output node

Imotor   = 0.000000445; % Motor inertia

Klow_1   = 30;          % Minimum actuator stiffness
Khigh_1  = 300;         % Max actuator stiffness

% LQR Trade-off matrice
Q = [1 0
     0 0.00000005];    % Penalize error in force
                        % Penalize rate of change in force
R = 0.01;              % Penalize actuator effort 1e-2

% Inverse because of bondgraph
Klow = 1/Klow_1;
Khigh = 1/Khigh_1;

% Final time of step response
tfinal = 0.008;

%% Calculations

% Generate statespace system and LQR commands
% Low stiffness
Alow = [0,1;-(n*n)/(Imotor * Klow)],-((Rmotor + n*n*Routp)/(Imotor))];
Blow = [0;((g*n)/(Imotor * Klow))];
Clow = [1,0];
Dlow = 0;
% High stiffness
Ahigh = [0,1;-(n*n)/(Imotor * Khigh)],-((Rmotor + n*n*Routp)/(Imotor))];
Bhigh = [0;((g*n)/(Imotor * Khigh))];
Chigh = [1,0];
Dhigh = 0;

% Calculate LQR and create feedback systems
Nlow = lqr(Alow,Blow,Q,R)
Nhigh = lqr(Ahigh,Bhigh,Q,R)
% Calculate and display the feedforward terms
'Feed forward term low stiffness:'
ffNlow = (n/g + Nlow(1))
'Feed forward term high stiffness:'
ffNhigh = (n/g + Nhigh(1))

sysClow = ss((Alow - Blow*Nlow),Blow,Clow,Dlow);
sysChigh = ss((Ahigh - Bhigh*Nhigh),Bhigh,Chigh,Dhigh);

sysCllowWhighN = ss((Alow - Blow*Nhigh),Blow,Clow,Dlow);
sysClhighWlowN = ss((Ahigh - Bhigh*Nlow),Bhigh,Chigh,Dhigh);

%% Openloop graphs
sysOLlow = ss(Alow,Blow,Clow,Dlow);
sysOLhigh = ss(Ahigh,Bhigh,Chigh,Dhigh);

figure(1);
step(sysOLlow);
```

Design and Development of a Magnetic Non-linear Elastic Element and Control for Progressive Series Actuation

```

hold on;
step(sys0Lhigh);
grid on;
legend('Low stiffness','High stiffness');
title('Open-loop step response')

%% Show variation of N changes over stiffness
cnt=1;
for q = Klow_1:1:Khigh_1
    k=1/q;
    % Recalculate matrices dependent on k
    A = [0,1;-(n*n)/(Imotor * k)],-(Rmotor + n*n*Routp)/(Imotor)];
    B = [0;(g*n)/(Imotor * k)];
    % Calculate gains and store
    N3(cnt,:) = lqr(A,B,Q,R);
    cnt = cnt+1;
end
tmline = linspace(Klow_1,Khigh_1,cnt -1); % Generate X-axis

%% Stiffness vs values of N
% Plot the stiffness vs N1
figure(2);
subplot(1,2,1);
plot(tmline, N3(:,1),'LineWidth',2);
xlim([Klow_1 Khigh_1]);
ylim([(min(min(N3(:,1)))-0.2) max(max(N3(:,1)))+0.1]);
lgd = legend('Calculated value of N-2');
grid on;
title('N1 vs change in stiffness');
xlabel('Stiffness k [Nm/rad]');
ylabel('Value');
lgd.FontSize = 11;
set(lgd, 'Interpreter','latex')
handle = gca;
handle.XAxis.FontSize = 11;
handle.YAxis.FontSize = 11;
% Do fitting to the N2 variable since it is the only one that changes with
% stiffness
yp = polyfit(tmline, N3(:,2), 4);
yt = polyval(yp, tmline);

% use the fraction fitted for an N2 fit
yfraction = (0.002482 * tmline(1,:) + 0.3533) ./ (tmline(1,:) + 28.26);

% Plot the stiffness vs N2
subplot(1,2,2);
plot(tmline, N3(:,2),'LineWidth',2);
hold on;
%plot(tmline, yt, 'LineWidth', 2);
plot(tmline, yfraction,'LineWidth',2);
grid on;
xlim([Klow_1 Khigh_1]);
ylim([(min(min(N3(:,2)))) max(max(N3(:,2)))]);
title('N2 vs change in stiffness', 'FontSize', 11);
xlabel('Stiffness k [Nm/rad]');
ylabel('Value');
lgd = legend('Calculated value of N2', '$\frac{0.002482 k + 0.3533}{k + 28.26}$');
lgd.FontSize = 11;
set(lgd, 'Interpreter','latex')
handle = gca;
handle.XAxis.FontSize = 11;
handle.YAxis.FontSize = 11;

set(gcf, 'units','points', 'position', [200,200,700,200]);

export_fig changeStiffnessVersusN.pdf -transparent

% Derive variables for curve fitting
x_fit = tmline;
y_fit = N3(:,2);

% Figure showing just N2
figure(6);
plot(tmline, N3(:,2), 'LineWidth', 2);
hold on;
%plot(tmline, yt, 'LineWidth', 2);
plot(tmline, yfraction,'LineWidth',2);
title('Value of N2 versus deflection', 'FontSize', 11);
lgd = legend('calculated value of N2', '$\frac{0.002482 k + 0.3533}{k + 28.26}$');
lgd.FontSize = 11;
set(lgd, 'Interpreter','latex')
xlabel('Stiffness k');
ylabel('N2');
handle = gca;
handle.XAxis.FontSize = 11;
handle.YAxis.FontSize = 11;
grid on;

%% Closed loop graphs

% Calculate inverse of the plant (see V4 documentation.odt) (Currently
% unused)
%InverPlant = s^2 + (Rmotor + Routp*n^2)/Imotor * s + (n^2)/(K * Imotor);
sysCllow = sysCllow * ffNlow;
sysClhigh = sysClhigh * ffNhigh;

```

```

sysCLlowWhighN = sysCLlowWhighN * ffNlow;
sysCLhighWlowN = sysCLhighWlowN * ffNhigh;
% Calculate step responses
[step1, t1, x1] = step(sysCLlow,tfinal);
[step2, t2, x2] = step(sysClhigh,tfinal);

[step3, t3, x3] = step(sysCLlowWhighN,tfinal);
[step4, t4, x4] = step(sysCLhighWlowN ,tfinal);
% Multiply with static gain of k + n/g
%step1 = step1*ffNlow;
%step2 = step2*ffNhigh;
%step3 = step3*ffNlow;
%step4 = step4*ffNhigh;
% Derive the associated state variables
statevar1 = [step1(1:(size(step1)-1)),diff(step1)];
statevar2 = [step2(1:(size(step2)-1)),diff(step2)];
statevar3 = [step3(1:(size(step3)-1)),diff(step3)];
statevar4 = [step4(1:(size(step4)-1)),diff(step4)];
% % Multiply statevariables with the N matrix to get the actuator effort
% FeedbackEffort1 = Nlow.*statevar1;
% FeedbackEffort2 = Nlow.*statevar2;
% FeedbackEffort3 = Nlow.*statevar3;
% FeedbackEffort4 = Nlow.*statevar4;
% Calculate the associated actuator efforts
stepsize = 1;
eff1 = stepsize*ffNlow - Nlow*transpose(statevar1);
eff2 = stepsize*ffNhigh - Nhigh*transpose(statevar2);
eff3 = stepsize*ffNhigh - Nhigh*transpose(statevar3);
eff4 = stepsize*ffNlow - Nlow*transpose(statevar4);

%sysCLlowWhighN
%sysCLhighWlowN

% Step response
figure(3);
subplot(2,1,1);
plot(t1,step1,'LineWidth',2);
hold on;
plot(t2,step2,'LineWidth',2);
grid on;
plot(t3,step3,'LineWidth',2);
grid on;
plot(t4,step4,'LineWidth',2);
grid on;
title('\fontsize{14}Closed-loop Step response');
lgd = legend('lin. plant @ K=30, low-stiff. ctrl.','lin. plant @ K=300, high-stiff. ctrl.','lin. plant @ K=30, high-stiff. ctrl.','lin. plant @ K=300, low-stiff. ctrl.','Location','southeast');
xlabel('Time [sec]');
ylabel('Torque [Nm]');
xlim([0 tfinal]);
handle = gca;
    handle.XAxis.FontSize = 12;
    handle.YAxis.FontSize = 12;

subplot(2,1,2);
plot(t1(1:size(eff1,2)), eff1,'LineWidth',2);
hold on;
plot(t2(1:size(eff2,2)), eff2,'LineWidth',2);
hold on;
plot(t3(1:size(eff3,2)), eff3,'LineWidth',2);
hold on;
plot(t4(1:size(eff4,2)), eff4,'LineWidth',2);
title('\fontsize{14}Actuator effort');
legend('lin. plant @ K=30, low-stiff. ctrl.','lin. plant @ K=300, high-stiff. ctrl.','lin. plant @ K=30, high-stiff. ctrl.','lin. plant @ K=300, low-stiff. ctrl.');
```

```

xlabel('Time [sec]');
ylabel('Effort [A]');
xlim([0 tfinal]);
grid on;
handle = gca;
    handle.XAxis.FontSize = 12;
    handle.YAxis.FontSize = 12;

% Pole zero plot
figure(4);
pzmap(sysCllow);
hold on;
pzmap(sysClhigh);
grid on;
title('Closed-loop Pole-zero diagram');
legend('low stiffness', 'high stiffness');
```

```

% Bode diagram
figure(5);
options = bodeoptions;
options.FreqUnits = 'Hz';
options.Title.FontSize = 14;
options.XLabel.FontSize = 12;
options.YLabel.FontSize = 12;
options.TickLabel.FontSize = 10;

bode(sysCllow, options);
hold on;
bode(sysClhigh, options);
grid on;
title('Closed-loop Bode diagram');
legend('low stiffness', 'high stiffness');
```

```

figure(3);

```

```
subplot(2,1,1);  
ylim([0 1.5]);  
subplot(2,1,2);  
ylim([-1.5 10.5]);  
export_fig steprespdemon.pdf -transparent
```

B.2 LQR parameter analysis Matlab code

The following code is used to analyze the change in performance of a linear plant at the two stiffness extrema of the non-linear element, with change in LQR parameters.

```

close all;
clear;

%% Parameters and user config
Rel      = 3.44;          % Electrical resistance of motor, only useful for power consumption and efficiency

g        = 0.0525;       % Motor torque constant
n        = 16/225;       % Gearbox transfer ratio

Rmotor  = 0.00015;      % Damping of motorbearings
Routp   = 0.00015;      % Damping of output node

Imotor  = 0.000000445;  % Motor inertia

Klow_1  = 30;           % Minimum actuator stiffness
Khigh_1 = 300;          % Max actuator stiffness

% LQR Trade-off matrices
Q = [1 0 % Penalize error in force
     0 5e-8]; % Penalize rate of change in force
R = 1e-2; % Penalize actuator effort

% Inverse because of bondgraph
Klow = 1/Klow_1;
Khigh = 1/Khigh_1;

% Final time of step response
tfinal = 0.008;

%% Vary R with constant Q
% Generate a non-linearly spaced (logarithmic) vector for R
numberofmeasurements = 4;
Rvect = [1e-3, 1e-2, 1e-1, 1e0];
for i = 1:numberofmeasurements
    %% Low stiffness system
    R = Rvect(i); % Retrieve the proper R
    % Generate statespace system and LQR commands
    % Low stiffness
    Alow = [0,1;-((n*n)/(Imotor * Klow)),-((Rmotor + n*n*Routp)/(Imotor))];
    Blow = [0;((g*n)/(Imotor * Klow))];
    Clow = [1,0];
    Dlow = 0;
    % LQR commands
    N = lqr(Alow,Blow,Q,R);
    ffN = (n/g + N(1));
    % Apply LQR
    sysCl = ss((Alow - Blow*N),Blow,Clow,Dlow);
    sysCl = sysCl * ffN; % Static gain compensation

    % Calculate step responses
    [step1, t1, x1] = step(sysCl,tfinal);
    % Derive the associated state variables
    statevar1 = [step1(1:(size(step1)-1)),diff(step1)];
    % Multiply statevariables with the N matrix to get the actuator effort
    FeedbackEffort1 = N.*statevar1;
    % Calculate the associated actuator efforts
    stepsize = 1;
    eff1 = stepsize*ffN - N*transpose(statevar1);

    % Plot
    % Generate a legend first
    legendtext = sprintf('R = 10^{-%i}',(i-4));
    % Step response
    figure(1);
    subplot(2,2,1);
    plot(t1,step1,'LineWidth',2,'DisplayName',legendtext);
    hold on;
    title('\fontsize{14}Closed-loop Step response, k=30');
    xlabel('Time [sec]');
    ylabel('Torque [Nm]');
    xlim([0 tfinal]);
    handle = gca;
    grid on;
    legend show
    handle.XAxis.FontSize = 12;
    handle.YAxis.FontSize = 12;
    % Actuator effort
    subplot(2,2,3);
    plot(t1(1:size(eff1,2)), eff1,'LineWidth',2,'DisplayName',legendtext);
    hold on;
    title('\fontsize{14}Actuator effort, k=30');
    xlabel('Time [sec]');
    ylabel('Effort [A]');
    xlim([0 tfinal]);
    ylim([0 12]);
    grid on;
    handle = gca;
    legend show
    handle.XAxis.FontSize = 12;
    handle.YAxis.FontSize = 12;

    %% high stiffness system

```

```

R = Rvect(i);% Retrieve the proper R
% Generate statespace system and LQR commands
% Low stiffness
Ahigh = [0,1;-((n*n)/(Imotor * Khigh)),-((Rmotor + n*n*Routp)/(Imotor))];
Bhigh = [0;((g*n)/(Imotor * Khigh))];
Chigh = [1,0];
Dhigh = 0;
% LQR commands
Nh = lqr(Ahigh,Bhigh,Q,R);
ffNh = (n/g + Nh(1));
% Apply LQR
sysClh = ss((Ahigh - Bhigh*Nh),Bhigh,Chigh,Dhigh);
sysClh = sysClh * ffNh; % Static gain compensation

% Calculate step responses
[step1, t1, x1] = step(sysClh,tfinal);
% Derive the associated state variables
statevar1 = [step1(1:(size(step1)-1)),diff(step1)];
% Multiply statevariables with the N matrix to get the actuator effort
FeedbackEffort1 = N.*statevar1;
% Calculate the associated actuator efforts
stepsize = 1;
eff1 = stepsize*ffN - N*transpose(statevar1);

% Plot
% Generate a legend first
legendtext = sprintf('R = 10^{%i}',(i-4));
% Step response
figure(1);
subplot(2,2,2);
plot(t1,step1,'LineWidth',2,'DisplayName',legendtext);
hold on;
title('\fontsize{14}Closed-loop Step response, k=300');
xlabel('Time [sec]');
ylabel('Torque [Nm]');
xlim([0 tfinal]);
handle = gca;
grid on;
legend show
handle.XAxis.FontSize = 12;
handle.YAxis.FontSize = 12;
% Actuator effort
subplot(2,2,4);
plot(t1(1:size(eff1,2)), eff1,'LineWidth',2,'DisplayName',legendtext);
hold on;
title('\fontsize{14}Actuator effort, k=300');
xlabel('Time [sec]');
ylabel('Effort [A]');
xlim([0 tfinal]);
ylim([0 12]);
grid on;
handle = gca;
legend show
handle.XAxis.FontSize = 12;
handle.YAxis.FontSize = 12;

%% TODO Temporary bode plots
legendtext
'low'
damp(sysCl)
'high'
damp(sysClh)
end

figure(1);
set(gcf,'units','points','position',[100,100,800,400]);
subplot(2,2,1);
legend('Location','southeast');
subplot(2,2,3);
legend('Location','northeast');
subplot(2,2,2);
legend('Location','southeast');
subplot(2,2,4);
legend('Location','northeast');
export_fig LQRDiffR.pdf -transparent

%% Vary Q1 with constant R
% Generate a non-linearly spaced (logarithmic) vector for R
numberofmeasurements = 5;
Qvect = [1e-2, 1e-1, 1e-0, 1e1, 1e2];
for i = 1:numberofmeasurements
    Q(1,1) = Qvect(i);% Retrieve the proper Q
    R = 1e-2;
    % Generate statespace system and LQR commands
    % Low stiffness
    Alow = [0,1;-((n*n)/(Imotor * Klow)),-((Rmotor + n*n*Routp)/(Imotor))];
    Blow = [0;((g*n)/(Imotor * Klow))];
    Clow = [1,0];
    Dlow = 0;
    % LQR commands
    N = lqr(Alow,Blow,Q,R)
    ffN = (n/g + N(1));
    % Apply LQR
    sysCl = ss((Alow - Blow*N),Blow,Clow,Dlow);
    sysCl = sysCl * ffN; % Static gain compensation

    % Calculate step responses

```

```
[step1, t1, x1] = step(sysCl,tfinal);
% Derive the associated state variables
statevar1 = [step1(1:(size(step1)-1)),diff(step1)];
% Multiply statevariables with the N matrix to get the actuator effort
FeedbackEffort1 = N.*statevar1;
% Calculate the associated actuator efforts
stepsize = 1;
eff1 = stepsize*ffN - N*transpose(statevar1);

% Plot
% Generate a legend first
legendtext = sprintf('Q_1 = 10^{%i}',(i-3));
% Step response
figure(3);
subplot(2,1,1);
plot(t1,step1,'LineWidth',2,'DisplayName',legendtext);
hold on;
title('\fontsize{14}Closed-loop Step response');
xlabel('Time [sec]');
ylabel('Torque [Nm]');
xlim([0 tfinal]);
handle = gca;
grid on;
legend show
handle.XAxis.FontSize = 12;
handle.YAxis.FontSize = 12;
% Actuator effort
subplot(2,1,2);
plot(t1(1:size(eff1,2)), eff1,'LineWidth',2,'DisplayName',legendtext);
hold on;
title('\fontsize{14}Actuator effort');
xlabel('Time [sec]');
ylabel('Effort [A]');
xlim([0 tfinal]);
ylim([0 12]);
grid on;
handle = gca;
legend show
handle.XAxis.FontSize = 12;
handle.YAxis.FontSize = 12;
end

export_fig LQRDiffQ.pdf -transparent
```

B.3 Elastic element model Matlab code

This code generates an elastic element model of torque and stiffness versus deflection, given a linear repelling force versus distance curve of a set of magnets and other elastic element parameters.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Calculations.m %%%
%%% This file calculates and generates the %%%
%%% plots for a magnet by dividing it up %%%
%%% %%% %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% How does the calculation work?
% Given: A magnet force profile, magnet radius from center, magnet length
% Split up the magnet profile into 1000 subsections, each subsection has a profile with force/1000
% For each subsection, calculate the force it experiences at every angle (at radius calculated by the average
% magnet radius and the 1000 pieces)
% Sum the results of all 1000 subsections for every angle

close all;
clear;

% Quickly calculate the ideal thickness for a given length
ActuatorRadius = 60000; % In um
MagnetAngleA = deg2rad(20); % magnet design

% These following variables hold the force in lbs force and distance in inch
% respectively

% w = 12, l = 30, t = 12, M=N52
% https://www.kjmagnetics.com/largergraph.asp?CI=4&D=0.842893639202757&T=0.47244094488189&L=1.18110236220472&W
% =0.47244094488189&OD=&ID=&PF=41.16&PFMAX=41.1645498083425&C=3.566&XSCALE=0.75&YSCALE=45&X=0.000001&
% calcType=block&GRADE=52
% https://www.supermagnete.nl/blokmagneten-neodymium/blokmagneet-30mm-12mm-12mm_Q-30-12-12-Z
PFA = [41.16 36.35 34.09 32.04 28.77 25.58 23.01 19.01 16 11.73 8.87 6.84 5.36 4.25 3.4 2.74 2.23];
PFA = PFA/2; % Compensate for difference in supermagneten statistics.
PdistA = [0 0.008 0.015 0.023 0.038 0.056 0.075 0.113 0.15 0.225 0.3 0.375 0.45 0.525 0.6 0.675 0.75];
MaxDeflectA = MagnetAngleA/2; % Max deflection to 1 side

% Calculate the result for each of the magnet types
polyfitvals = main(PFA, PdistA,MaxDeflectA, 0);

% Anotate the resulting figure
figure(1);
title('Exerted magnet torque vs deflection, Magnet Design A');
figure(2);
title('Effective stiffness vs deflection, Magnet Design A');

function a = main(PF, Pdist,Deflection, fignum)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% USER CONFIG DATA
%% Magnet configuration details for the user to tweak
MagnetLength = 30000; %Magnet Length in um
MagnetPosition = 30000; % Magnet placement away from the radius in um. Note that this is not the magnet center,
% but the point closest to the center

% Number of pole pairs in the actuator
PolePairs = 3;

%% Other configuration options (mostly for resolution of results)
% Conversion values
% These are needed to convert from the found magnet data to SI data
LbToNewton = 4.4482216282509; % 1 lb = 4.4482216282509 newton
InchToUM = 25400; % 1 inch = 25.4mm = 25400 um

% Resolution settings
numberOfSections = 1000; % Details into how many parts the magnet will be split
AngleSteps = 1001; % Number of steps that spans the total angle (Deflection * 2 is total angle)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Non-user modifiable variables/constants
% The underlying variables do not need to be modified by the user, are
% calculated through other input data.

% Do SI conversion on the input data for magnet force
ForceInNewton = PF*LbToNewton;
DistanceInUM = Pdist * InchToUM;
% use the above info to precalculate the angles needed. From -Deflection to
% Deflection spaced at Anglesteps
AngleVector = linspace(0,2*Deflection,AngleSteps); % This maps 0 to 2*Defl
AngleVector2 = linspace(-Deflection,Deflection,AngleSteps);% This maps -Defl to Defl, mostly used for
% displaying the results
maxlindeflect = round(2*(MagnetPosition + MagnetLength)*sin(0.5*Deflection))*2; % This is the maximum
% deflection in um thats taken into account for the
ForcePerUmDistance = magnetParamToForceVector(DistanceInUM,ForceInNewton,maxlindeflect); % This array returns
% the force experienced between two same magnets at a distance of the given amount of um. It is used as a
% lookup table

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% MAIN CODE
%Split the magnet into 1000 equal pieces
ForceperUmDistancePerSegment = ForcePerUmDistance / numberOfSections;

% Calculate the torque at every distance r for every section
```

Design and Development of a Magnetic Non-linear Elastic Element and Control for Progressive Series Actuation

```
torqueSect = zeros(numberOfSections,AngleSteps); % Preallocate for speed
currentR = MagnetPosition; % Start with the section closest to the center, work our way outwards
for currentMagnetSegment = 1:numberOfSections
    torqueSect(currentMagnetSegment,:) = calculateTorqueAtDistance(currentR, ForceperUmDistancePerSegment,
        AngleVector); % Calculate the torque generated by the current segment
    currentR = currentR + (MagnetLength/numberOfSections); % Calculate the R for the next part of the cycle
end

% Summing the segments to produce the total torque of one magnet pair in
% Num
TotalTorqueOneMagnetNum = sum(torqueSect,1);
% Calculate it in Nm
TotalTorqueOneMagnetNm = TotalTorqueOneMagnetNum / 1000000;

% Here we extrapolate the result of a single magnet, to a pair of magnets
% Sum two opposite facing magnet pairs
TotalTorqueSinglePoleNm = TotalTorqueOneMagnetNm - flip(TotalTorqueOneMagnetNm);
% Calculate the total torque of the entire actuator
TotalTorqueActuatorNm = TotalTorqueSinglePoleNm * PolePairs;

% Now finally do numerical differentiation to achieve the stiffness that
% results for the actuator
TotalStiffnessActuator = zeros(1,AngleSteps); % Presize for speed
anglePerStep = 2*Deflection / AngleSteps; % Step size in rad per little step taken
for i = 1:AngleSteps-1
    TotalStiffnessActuator(i) = ((TotalTorqueActuatorNm(i+1)-TotalTorqueActuatorNm(i))/(anglePerStep));
end
TotalStiffnessActuator(AngleSteps) = TotalStiffnessActuator(AngleSteps-1); % Cant calculate this one, copy it
    from the previous entry to prevent jumps in data
TotalStiffnessActuator = abs(TotalStiffnessActuator); % Stiffness is an absolute value

% Do a polyfit on the torque
polpoints = doPolyFit(AngleVector2, TotalTorqueActuatorNm);
a = polpoints;
y1 = polyval(polpoints, AngleVector2);
% Derive the stiffness from this by numerical differentiation
y2 = zeros(1,AngleSteps); % Presize for speed
anglePerStep = 2*Deflection / AngleSteps; % Step size in rad per little step taken
for i = 1:AngleSteps-1
    y2(i) = ((y1(i+1)-y1(i))/(anglePerStep));
end
y2(AngleSteps) = y2(AngleSteps-1); % Cant calculate this one, copy it from the previous entry to prevent jumps
    in data
y2 = abs(y2); % Stiffness is an absolute value

%% Plotting

% Plot Torque
figure(fignum+1);
plot(AngleVector2, TotalTorqueActuatorNm);
xlabel('deflection [rad]');
ylabel('Magnet torque [Nm]');
title('Exerted magnet torque vs deflection');
axis([-Deflection Deflection -inf inf]);
grid on;
hold on;

% Plot stiffness
figure(fignum+2);
plot(AngleVector2, TotalStiffnessActuator);
xlabel('deflection [rad]');
ylabel('Stiffness [Nm/rad]');
title('Effective stiffness vs deflection');
axis([-Deflection Deflection -inf inf]);
grid on;
hold on;

% Add the polyfit to the Plots
figure(fignum+1);
plot(AngleVector2, y1);
xlabel('deflection [rad]');
ylabel('Magnet torque [Nm]');
title('Exerted magnet torque vs deflection');
axis([-Deflection Deflection -inf inf]);
legend('Calculated data', 'Polyfit'); % TODO add the polyfit values
grid on;
hold on;
figure(fignum+2);
plot(AngleVector2, y2);
xlabel('deflection [rad]');
ylabel('Stiffness [Nm/rad]');
title('Effective stiffness vs deflection');
axis([-Deflection Deflection -inf inf]);
legend('Calculated data', 'Polyfit'); % TODO add the polyfit values
grid on;
hold on;

% Put them together in 1 figure for exporting
figure(3);
subplot(2,1,1);
plot(AngleVector2, TotalTorqueActuatorNm, 'LineWidth', 1.5);
hold on;
plot(AngleVector2, y1, 'LineWidth', 1.5);
xlabel('deflection [rad]', 'FontSize', 11);
ylabel('Magnet torque [Nm]', 'FontSize', 11);
title('Exerted magnet torque vs deflection', 'FontSize', 11);
legend('Model', 'Polyfit');
axis([-Deflection Deflection -inf inf]);
handle = gca;
handle.XAxis.FontSize = 10;
```

Design and Development of a Magnetic Non-linear Elastic Element and Control for Progressive Series Actuation

```
handle.YAxis.FontSize = 10;
grid on;
hold on;
subplot(2,1,2);
plot(AngleVector2, TotalStiffnessActuator, 'LineWidth', 1.5);
hold on;
plot(AngleVector2, y2, 'LineWidth', 1.5);
xlabel('deflection [rad]', 'FontSize', 11);
ylabel('Stiffness [Nm/rad]', 'FontSize', 11);
title('Effective stiffness vs deflection', 'FontSize', 11);
legend('Model', 'Polyfit');
axis([-Deflection Deflection -inf inf]);
handle = gca;
handle.XAxis.FontSize = 10;
handle.YAxis.FontSize = 10;
grid on;
hold on;

export_fig springmodeltorqueandstiffness.pdf -transparent

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Helper Functions

% Given a radius and a forcemapping vector, calculate the torque vector
% from -Deflection to +Deflection (these are mapped on index 1 to AngleSteps+1)
% Note that radius must be given in um
function torqueVector = calculateTorqueAtDistance(radius, forcemappingVector, AngVect)
    lindist = round(2*radius*sin(0.5* AngVect)); % Calculate the linear distance from radius and the angle
    vector
    lindist(lindist<1) = 1;% Ensure that any distance is larger than 1 for the lookup table
    linforce = forcemappingVector(lindist);% use the linear distance to calculate the force at every point
    torqueVector = linforce*radius; % Convert the linear force into a torque
end

% Generates a force vector with a given amount of force for any distance
% from 0 deflection to Deflection um away. Does steps of ium.
% It does so by making a cubic fit to the given magnet data, and
% extrapolating this.
function forceVectorGenerated = magnetParamToForceVector(DistInUM, FInNewton, Deflc)
    % Extrapolate input data with linear spline interpolation
    s = csaps(DistInUM, FInNewton,1); % Generate cubic smoothing spline
    extrapolatedSpline = fnxtr(s,2); % Extrapolate

    % preallocate for speed
    forceVectorGenerated = zeros(1,Deflc+1);
    % Now generate a vector from 0 to Deflection um with steps of ium
    for curdist = 1:Deflc
        forceVectorGenerated(curdist) = fnval(extrapolatedSpline,curdist); % Evaluate at every um
    end

    % Forces under zero are set to 0
    forceVectorGenerated(forceVectorGenerated<0) = 0;
end

% This is a small utility function to return the maximum allowable
% thickness of a magnet given a length, max actuator radius and allowed angle
function maxthickness = calculateMagnetThickness(radius, angle, magnetlength)
    maxthickness = (radius-magnetlength)*tan(angle/2);
end

% This function performs a polyfit on the given angle and torque vector,
% returning the coefficients of the polynomial found
function polynomialfit = doPolyFit(anglevector, torquevector)
    polynomialfit = polyfit(anglevector, torquevector, 11);
    polynomialfit
end
```

B.4 Step response data processing Matlab code

Given the data from simulations with the non-linear series elastic element, this block processes the results for the step response.

```
clear all;
close all;

%% Load in data
nonlintable = readtable('Non-linContr.csv');
linLStable = readtable('LinContrLowS.csv');
linHStable = readtable('LinContrHighS.csv');

nonlin = table2array(nonlintable);
linLS = table2array(linLStable);
linHS = table2array(linHStable);

% Global setting for max time
xlimglobal = 0.007;

% Remove the parts that initialize
nonlin = nonlin(100003:end,:);
linLS = linLS(100003:end,:);
linHS = linHS(100003:end,:);
% Decrease time for it too
nonlin(:,1) = nonlin(:,1)-0.01;
linLS(:,1) = linLS(:,1)-0.01;
linHS(:,1) = linHS(:,1)-0.01;

%% Plotting
figure(1);
subplot(3,1,1); % First plot torque
plot(nonlin(:,1),nonlin(:,2),'LineWidth', 2, 'DisplayName', 'Gain-scheduled ctrl. ');
hold on;
plot(linLS(:,1),linLS(:,2),'LineWidth', 2, 'DisplayName', 'Low-stiff. ctrl. ');
hold on;
plot(linHS(:,1),linHS(:,2),'LineWidth', 2, 'DisplayName', 'High-stiff. ctrl. ');
hold on;
grid on;

title('\fontsize{14}Torque');
xlabel('Time [sec]');
ylabel('Torque [Nm]');
handle = gca;
legend('show','Location','southeast');
handle.XAxis.FontSize = 12;
handle.YAxis.FontSize = 12;
xlim([0 xlimglobal]);
ylim([0 1.4]);

subplot(3,1,2); % Actuator effort
plot(nonlin(:,1),nonlin(:,4),'LineWidth', 2, 'DisplayName', 'Gain-scheduled ctrl. ');
hold on;
plot(linLS(:,1),linLS(:,4),'LineWidth', 2, 'DisplayName', 'Low-stiff. ctrl. ');
hold on;
plot(linHS(:,1),linHS(:,4),'LineWidth', 2, 'DisplayName', 'High-stiff. ctrl. ');
hold on;
grid on;

title('\fontsize{14}Actuator effort');
xlabel('Time [sec]');
ylabel('Effort [A]');
handle = gca;
legend('show','Location','northeast');
handle.XAxis.FontSize = 12;
handle.YAxis.FontSize = 12;
xlim([0 xlimglobal]);
ylim([-3 10]);

subplot(3,1,3); % Stiffness
plot(nonlin(:,1),nonlin(:,3),'LineWidth', 2, 'DisplayName', 'Gain-scheduled ctrl. ');
hold on;
plot(linLS(:,1),linLS(:,3),'LineWidth', 2, 'DisplayName', 'Low-stiff. ctrl. ');
hold on;
plot(linHS(:,1),linHS(:,3),'LineWidth', 2, 'DisplayName', 'High-stiff. ctrl. ');
hold on;

title('\fontsize{14}Presented stiffness');
xlabel('Time [sec]');
ylabel('Stiffness [Nm/rad]');
ylim([30 35]);
xlim([0 xlimglobal]);
grid on;
handle = gca;
legend('show','Location','southeast');
handle.XAxis.FontSize = 12;
handle.YAxis.FontSize = 12;
xlim([0 0.007]);

set(gcf,'Position',[100 100 500 560])

export_fig Comparison1Nm.pdf -transparent

ST = 0.05; % Percentage before the step is considered settled
```

Design and Development of a Magnetic Non-linear Elastic Element and Control for Progressive Series Actuation

```
S1 = stepinfo(nonlin(:,3),nonlin(:,1),'SettlingTimeThreshold',ST)
S2 = stepinfo(linLS(:,3),linLS(:,1),'SettlingTimeThreshold',ST);
S3 = stepinfo(linHS(:,3),linHS(:,1),'SettlingTimeThreshold',ST);

(S1.SettlingTime - S2.SettlingTime) / (S2.SettlingTime) * 100;

(S1.SettlingTime - S3.SettlingTime) / (S3.SettlingTime) * 100;
```

B.5 Transparency data processing Matlab code

Given the data from simulations with the non-linear series elastic element, this block processes the results for the actuator transparency.

```
clear all;
close all;

% Preformat figure 2 for displaying properly
figure(2);
ax = gca;
RGBvec = [7 103 143; 0 114 119; 7 143 109;
          240 116 17; 217 83 25; 240 51 17;
          214 179 19; 236 177 32; 214 139 19]; % Colors for plotting, RGB values, Go from blue shades, to red
          shades, to yellow shades
ax.ColorOrder = RGBvec/255;
%ax.ColorOrder = [0 0.447 0.741; 0.85 0.325 0.098; 0.929 0.694 0.125]; % The three default colors only (after
the last starts with first again)
%ax.LineStyleOrder = {'-', '--', '-.'}; % character vector deciding the line style, doesnt complete until a lap
of color order is completed

%% Read in the data
% Linear plant, 30 Nm/rad stiffness, figure 1
readCSVAndPlot('linplant30_lowstiffcontrol.csv',1,3,1,'Linear Plant 30 [ $\frac{\text{Nm}}{\text{rad}}$ ]', Low-stiff. ctrl. ');
readCSVAndPlot('linplant30_highstiffcontrol.csv',1,3,1,'Linear Plant 30 [ $\frac{\text{Nm}}{\text{rad}}$ ]', High-stiff. ctrl. ');
;
readCSVAndPlot('linplant30_nonlincontrol.csv',1,3,1,'Linear Plant 30 [ $\frac{\text{Nm}}{\text{rad}}$ ]', Gain-sched. ctrl. ');

% Linear plant, 300 Nm/rad stiffness, figure 2
readCSVAndPlot('linplant300_lowstiffcontrol.csv',1,3,2,'Linear Plant 300 [ $\frac{\text{Nm}}{\text{rad}}$ ]', Low-stiff. ctrl. ');
;
readCSVAndPlot('linplant300_highstiffcontrol.csv',1,3,2,'Linear Plant 300 [ $\frac{\text{Nm}}{\text{rad}}$ ]', High-stiff. ctrl. ');
;
readCSVAndPlot('linplant300_nonlincontrol.csv',1,3,2,'Linear Plant 300 [ $\frac{\text{Nm}}{\text{rad}}$ ]', Gain-sched. ctrl. ');

% Non-linear plant, figure 3
readCSVAndPlot('nonlinplant_lowstiffcontrol.csv',1,3,3,'Non-Linear Plant, Low-stiff. ctrl. ');
readCSVAndPlot('nonlinplant_highstiffcontrol.csv',1,3,3,'Non-Linear Plant, High-stiff. ctrl. ');
readCSVAndPlot('nonlinplant_nonlincontrol.csv',1,3,3,'Non-Linear Plant, Gain-sched. ctrl. ');

figure(1);
subplot(3,1,1);
ylim([-20 0]);
grid on;
lgd = legend('Linear - low stiffness','Linear - high stiffness','Non-linear', 'Location', 'southeast');
t = title('Linear plant - 30 [ $\frac{\text{Nm}}{\text{rad}}$ ] stiffness','FontSiz',13);
set(t, 'Interpreter','latex');
handle = gca;
handle.XAxis.FontSize = 10;
handle.YAxis.FontSize = 10;
ylabel('Magnitude [dB]');
xlabel('Frequency [Hz]');
set(gca, 'XScale', 'log');
subplot(3,1,2);
ylim([-20 0]);
grid on;
lgd = legend('Linear - low stiffness','Linear - high stiffness','Non-linear', 'Location', 'northeast');
t = title('Linear plant - 300 [ $\frac{\text{Nm}}{\text{rad}}$ ] stiffness','FontSiz',13);
set(t, 'Interpreter','latex');
handle = gca;
handle.XAxis.FontSize = 10;
handle.YAxis.FontSize = 10;
ylabel('Magnitude [dB]');
xlabel('Frequency [Hz]');
set(gca, 'XScale', 'log');
subplot(3,1,3);
ylim([-20 0]);
grid on;
lgd = legend('Linear - low stiffness','Linear - high stiffness','Non-linear', 'Location', 'southeast');
t = title('Non-linear plant','FontSiz',13);
set(t, 'Interpreter','latex');
handle = gca;
handle.XAxis.FontSize = 10;
handle.YAxis.FontSize = 10;
ylabel('Magnitude [dB]');
xlabel('Frequency [Hz]');
set(gca, 'XScale', 'log');

filename = ['actTransparency_Comp_FFT_ALL.pdf'];
set(gcf, 'Position', [500 100 500 750]);
export_fig(filename, '-transparent')

figure(2);
t = title('Comparison responses controllers,  $\tau_{\text{set}} = 0$  [Nm]','FontSiz',13);
set(t, 'Interpreter','latex');
set(gca, 'XScale', 'log');
legend('Location','south');
grid on;
handle = gca;
handle.XAxis.FontSize = 10;
handle.YAxis.FontSize = 10;
legend('show');
ylabel('Magnitude [dB]');
xlabel('Frequency [Hz]');

filename = ['actTransparency2_Comp_FFT_ALL.pdf'];
set(gcf, 'Position', [100 100 600 400])
```

Design and Development of a Magnetic Non-linear Elastic Element and Control for Progressive Series Actuation

```
export_fig(filename, '-transparent')

function a = readCSVAndPlot(filename, figurenumber, subplottotal, subplotnumber, labeltext)

    %% Read in the CSV datafile
    Raw_data_t = readtable(filename);
    % Delete unnecessary columns
    Raw_data_t.OutputPosition_input = [];
    Raw_data_t.PresentedStiffness0L_input = [];
    Raw_data_t.PresentedStiffness_input = [];
    % Delete the first two rows of BS readings
    Raw_data_t([1,2],:) = [];
    % Delete last row of BS readings
    Raw_data_t([height(Raw_data_t)-1,height(Raw_data_t)],:) = [];
    % Convert to array for further processing, and ensure only unique rows
    Raw_data = unique(table2array(Raw_data_t),'rows');

    %% Generate a timescale and extrapolate the data to this timescale
    % (10khz for 100s)
    totaltime = 100;
    fs = 5000;
    timestep = 1/fs;
    timescale = 0:timestep:totaltime;
    % Extrapolate to the new timescale
    translatedData(:,1) = timescale;
    translatedData(:,2) = interp1(Raw_data(:,1),Raw_data(:,2),timescale);
    translatedData(:,3) = interp1(Raw_data(:,1),Raw_data(:,3),timescale);
    % Trim away the last two values because those are NaN
    transData(:, :) = translatedData(1:(fs*totaltime - 2),:);

    d = 3;
    %% Do fourier transform
    Y = fft(transData(:,2:3));
    L = length(transData(:,1));
    n = 2^nextpow2(L);
    % Generate frequency scale
    f = fs*(0:(n/2))/n;
    P1 = abs(Y(:,1)/n).^2;
    P2 = abs(Y(:,2)/n).^2;

    maxdb = 2.3778e-06; %max([max(P1) max(P2)])
    P1db = 10*log10(P1/maxdb); % No controller
    P2db = 10*log10(P2/maxdb); % Controlled

    % figure(5)
    % hold off;
    % plot(translatedData(:,1),translatedData(:,2));
    % hold on;
    % plot(translatedData(:,1),translatedData(:,3));
    % figure(6);
    % hold off;
    % plot(f, P1db(1:n/2+1));
    % hold on;
    % plot(f, P2db(1:n/2+1));

    difference = P2db - P1db; % No controller - controller

    % Try resampling before plotting to smooth out?
    P2db = smooth(P2db);

    fnew = logspace(0,3,100);
    difnew = interp1(f, difference(1:n/2+1),fnew);
    difnew = smooth(difnew);
    P2new = interp1(f, P2db(1:n/2+1),fnew);

    %% Plotting the result
    % Select plot stuff
    %fn = figure(figurenumber);
    %close(fn);
    figure(figurenumber);
    subplot(subplottotal,1,subplotnumber);
    hold on;
    %semilogx(f, difference(1:n/2+1), 'LineWidth', 2);
    semilogx(fnew, difnew, 'LineWidth', 2);
    %semilogx(f, P1db(1:n/2+1), 'LineWidth', 2);
    %hold on;
    %semilogx(f, P2db(1:n/2+1), 'LineWidth', 2);
    xlim([1 1000])

    figure(2);
    hold on;
    %semilogx(f, P1db(1:n/2+1), 'LineWidth', 2, 'DisplayName', 'Open L');
    %semilogx(f, P2db(1:n/2+1), 'LineWidth', 2, 'DisplayName', labeltext);
    semilogx(fnew, P2new, 'LineWidth', 2, 'DisplayName', labeltext);
    xlim([1 900]);
    ylim([-25 0]);

    legend('Interpreter','latex');
end
```

B.6 Torque tracking data processing Matlab code

Given the data from simulations with the non-linear series elastic element, this block processes the results for the actuator transparency.

```
clear all;
close all;

% Preformat figure 2 for displaying properly
figure(1);
ax = gca;
RGBvec = [7 103 143; 0 114 119; 7 143 109;
          240 116 17; 217 83 25; 240 51 17;
          214 179 19; 236 177 32; 214 139 19]; % Colors for plotting, RGB values, Go from blue shades, to red
          shades, to yellow shades
ax.ColorOrder = RGBvec/255;
%ax.ColorOrder = [0 0.447 0.741; 0.85 0.325 0.098; 0.929 0.694 0.125]; % The three default colors only (after
the last starts with first again)
%ax.LineStyleOrder = {'-', '--', '-.'}; % character vector deciding the line style, doesnt complete until a lap
of color order is completed

%% Read in the data
% Linear plant, 30 Nm/rad stiffness, figure 1
readCSVAndPlot('TT_30-lin-plant_Low-stiff-ctrl.csv', [0,0.447,0.741], 3,1, 'Linear Plant 30 [ $\frac{\text{Nm}}{\text{rad}}$ ],
Low-stiff. ctrl. ');
readCSVAndPlot('TT_30-lin-plant_High-stiff-ctrl.csv', [0.85,0.325,0.098], 3,1, 'Linear Plant 30 [ $\frac{\text{Nm}}{\text{rad}}$ ]
, High-stiff. ctrl. ');
readCSVAndPlot('TT_30-lin-plant_Non-lin-ctrl.csv', [0.929,0.694,0.125], 3,1, 'Linear Plant 30 [ $\frac{\text{Nm}}{\text{rad}}$ ],
Gain-sched. ctrl. ');

% Linear plant, 300 Nm/rad stiffness, figure 2
readCSVAndPlot('TT_300-lin-plant_Low-stiff-ctrl.csv', [0,0.447,0.741], 3,2, 'Linear Plant 300 [ $\frac{\text{Nm}}{\text{rad}}$ ],
Low-stiff. ctrl. ');
readCSVAndPlot('TT_300-lin-plant_High-stiff-ctrl.csv', [0.85,0.325,0.098], 3,2, 'Linear Plant 300 [ $\frac{\text{Nm}}{\text{rad}}$ ]
, High-stiff. ctrl. ');
readCSVAndPlot('TT_300-lin-plant_Non-lin-ctrl.csv', [0.929,0.694,0.125], 3,2, 'Linear Plant 300 [ $\frac{\text{Nm}}{\text{rad}}$ ]
, Gain-sched. ctrl. ');

% Non-linear plant, figure 3
readCSVAndPlot('TT_Non-lin-plant_Low-stiff-ctrl.csv', [0,0.447,0.741], 3,3, 'Non-Linear Plant, Low-stiff. ctrl. ');
readCSVAndPlot('TT_Non-lin-plant_High-stiff-ctrl.csv', [0.85,0.325,0.098], 3,3, 'Non-Linear Plant, High-stiff.
ctrl. ');
readCSVAndPlot('TT_Non-lin-plant_Non-lin-ctrl.csv', [0.929,0.694,0.125], 3,3, 'Non-Linear Plant, Gain-sched. ctrl.
');

figure(1);
subplot(3,1,1);
t = title('Comp. torque tracking, 30 [ $\frac{\text{Nm}}{\text{rad}}$ ] plant,  $\tau_{\text{offs}} = 9\text{ [Nm]}$ ,  $\tau_{\text{ampl}} = 1\text{ [Nm]}$ ',
'FontSize', 13);
set(t, 'Interpreter', 'latex');
set(gca, 'XScale', 'log');
lgd = legend('Location', 'southwest');
lgd.FontSize = 10;
grid on;
handle = gca;
yyaxis left;
handle.XAxis.FontSize = 10;
handle.YAxis(1).FontSize = 10;
handle.YAxis(1).Color = 'black';
legend('show');
ylabel('Magnitude [dB]');
xlabel('Frequency [Hz]');
handle = gca;
yyaxis right;
handle.XAxis.FontSize = 10;
handle.YAxis(2).FontSize = 10;
handle.YAxis(2).Color = 'black';
legend('show');
ylabel('Phase [rad]');
xlabel('Frequency [Hz]');

subplot(3,1,2);
t = title('Comp. torque tracking, 300 [ $\frac{\text{Nm}}{\text{rad}}$ ] plant,  $\tau_{\text{offs}} = 9\text{ [Nm]}$ ,  $\tau_{\text{ampl}} = 1\text{ [Nm]}$ ',
'FontSize', 13);
set(t, 'Interpreter', 'latex');
set(gca, 'XScale', 'log');
lgd = legend('Location', 'southwest');
lgd.FontSize = 10;
grid on;
handle = gca;
yyaxis left;
handle.XAxis.FontSize = 10;
handle.YAxis(1).FontSize = 10;
handle.YAxis(1).Color = 'black';
legend('show');
ylabel('Magnitude [dB]');
xlabel('Frequency [Hz]');
handle = gca;
yyaxis right;
handle.XAxis.FontSize = 10;
handle.YAxis(2).FontSize = 10;
handle.YAxis(2).Color = 'black';
legend('show');
ylabel('Phase [rad]');
xlabel('Frequency [Hz]');
```

Design and Development of a Magnetic Non-linear Elastic Element and Control for Progressive Series Actuation

```
subplot(3,1,3);
t = title('Comp. torque tracking, non-lin. plant,  $\tau_{offs} = 9\%$  [Nm],  $\tau_{ampl} = 1\%$  [Nm]', 'FontSize', 13);
set(t, 'Interpreter', 'latex');
set(gca, 'XScale', 'log');
lgd = legend('Location', 'southwest');
lgd.FontSize = 10;
grid on;
handle = gca;
yyaxis left;
handle.XAxis.FontSize = 10;
handle.YAxis(1).FontSize = 10;
handle.YAxis(1).Color = 'black';
legend('show');
ylabel('Magnitude [dB]');
xlabel('Frequency [Hz]');
handle = gca;
yyaxis right;
handle.XAxis.FontSize = 10;
handle.YAxis(2).FontSize = 10;
handle.YAxis(2).Color = 'black';
legend('show');
ylabel('Phase [rad]');
xlabel('Frequency [Hz]');

filename = ['actTorqueTracking9Nm_Comp_FFT_ALL.pdf'];
set(gcf, 'Position', [100 100 800 800]);
export_fig(filename, '-transparent')

function a = readCSVAndPlot(filename, color, subplottotal, subplotnumber, labeltext)

    %% Read in the CSV datafile
    Raw_data_t = readtable(filename);
    % Delete unnecessary columns
    %Raw_data_t.ActuatorEffort_input = [];
    %Raw_data_t.PresentedStiffness_input = [];
    % Delete the first two rows of BS readings
    Raw_data_t([1,2], :) = [];
    % Delete last row of BS readings
    Raw_data_t([height(Raw_data_t)-1, height(Raw_data_t)], :) = [];
    % Convert to array for further processing, and ensure only unique rows
    Raw_data = unique(table2array(Raw_data_t), 'rows');

    %% Generate a timescale and extrapolate the data to this timescale
    % (10khz for 100s)
    totaltime = 100;
    fs = 5000;
    timestep = 1/fs;
    timescale = 0:timestep:totaltime;
    % Extrapolate to the new timescale
    translatedData(:,1) = timescale;
    translatedData(:,2) = interp1(Raw_data(:,1), Raw_data(:,2), timescale);
    translatedData(:,3) = interp1(Raw_data(:,1), Raw_data(:,3), timescale);
    % Trim away the last two values because those are NaN
    transData(:, :) = translatedData(1:(fs*totaltime - 2), :);

    d = 3;
    %% Do fourier transform
    Yout = fft(transData(:,2));
    Yin = fft(transData(:,3));
    L = length(transData(:,1));
    n = 2^nextpow2(L);
    % Generate frequency scale
    f = fs*(0:(n/2))/n;

    Y = Yout./Yin;

    % Magnitude
    P1 = abs(Y(:,1))/n).^2;
    % Phase
    % Mask out very small values in phase since they give erroneous results
    % with atan2
    Ycopy = Y;
    threshold = max(abs(Ycopy))/10000;
    Ycopy(abs(Ycopy)<threshold) = 0; % Set all the values where too small to 0
    % Use atan2 for wrapping
    P1_phase = atan2(imag(Ycopy(:,1)), real(Ycopy(:,1)));
    P1_phase = angle(Y);

    maxdb = 2.3778e-06; %max([max(P1) max(P2)])
    P1db = 10*log10(P1/maxdb);

    % Smooth out the result
    P1db = smooth(P1db);

    fnew = logspace(0,3,100);
    P1new = interp1(f, P1db(1:n/2+1), fnew);

    P1phasenew = interp1(f, P1_phase(1:n/2+1), fnew);

    % Normalize to 0dB
    P1new(:) = P1new(:) - P1new(1);

    %% Plotting the result
    % Select plot stuff
    figure(1);
```

```
subplot(3,1,subplotnumber);
yyaxis left
hold on;
a = semilogx(fnew, P1new, '-.', 'LineWidth', 2, 'DisplayName', [labeltext ' Magnitude']);
a.Color = color;
legend('Interpreter','latex');
hold on;
xlim([10 900])
ylim([-20 5]);
yyaxis right
hold on;
a = semilogx(fnew, P1phasenew, '--', 'LineWidth', 2, 'DisplayName', [labeltext ' Phase']);
a.Color = color;
legend('Interpreter','latex');
hold on;
grid on;
xlim([10 900])
ylim([-pi 0]);
end

%% % Linear plant, 30 Nm/rad stiffness
% LP30_LS = readtable('linplant30_lowstiffcontrol.csv');
% LP30_HS = readtable('linplant30_highstiffcontrol.csv');
% LP30_NL = readtable('linplant30_nonlincontrol.csv');
%
%% % Linear plant, 300 Nm/rad stiffness
% LP300_LS = readtable('linplant300_lowstiffcontrol.csv');
% LP300_HS = readtable('linplant300_highstiffcontrol.csv');
% LP300_NL = readtable('linplant300_nonlincontrol.csv');
%
%% % Non-linear plant
% NP_LS = readtable('nonlinplant_lowstiffcontrol.csv');
% NP_HS = readtable('nonlinplant_highstiffcontrol.csv');
% NP_NL = readtable('nonlinplant_nonlincontrol.csv');
```

B.7 Result processing Matlab code

The below code is used to process measured data from the elastic element. It also provides a comparison with the model.

```
clear all;
close all;

% Read in the measured torque data
data1 = readLoadcellTableDoubledir('Data.csv');
% Plot the data
plotTorqueAndStiffness(data1,1);

% Generate a polyrange of the modelled torque
anglerange = -10:0.01:10;
load('polyfittorque.mat');
polyvals = -polyval(polyfitvals, deg2rad(anglerange));
% Shift 0 point
%polyvals = -polyval(polyfitvals, deg2rad(anglerange)) - 1.4;
% Add the data to the plot
figure(1);
subplot(1,2,1);
hold on;
plot(anglerange, polyvals);
ylim([-12 12]);

polyfitvalsder = polyder(polyfitvals);
polyvalsder = abs(polyval(polyfitvalsder, deg2rad(anglerange)));
subplot(1,2,2);
hold on;
plot(anglerange, polyvalsder);

export_fig torqandstiff_1_2021-01-18.pdf -transparent

% figure(2);
% subplot(1,2,1);
% hold on;
% plot(anglerange, polyvals);
% ylim([-12 12]);
%
% polyfitvalsder = polyder(polyfitvals);
% polyvalsder = abs(polyval(polyfitvalsder, deg2rad(anglerange)));
% subplot(1,2,2);
% hold on;
% plot(anglerange, polyvalsder);

%export_fig torqandstiff_2_2021-01-18.pdf -transparent

function data = readLoadcellTableSingledir(filename)
% Read in the table
table = readtable(filename);
% Convert to variable
table_data = table2array(table);

% Format data for output
data(:,1) = 10:-1:-10; % Generate angles for angle column
data(:,2) = table_data(:,4); % Torque column
end

function data = readLoadcellTableDoubledir(filename)
% Read in the table
table = readtable(filename);
% Convert to variable
table_data = table2array(table);

% Format data for output
data(1:11,1) = 0:1:10; % Generate angles for angle column
data(12:22,1) = 0:-1:-10;
data(:,2) = table_data(:,4); % Torque column of table

% Remove the first row because of duplicates
data(1,:) = [];
end

function t = plotTorqueAndStiffness(data, figurenumber)
% Apply a polyfit to the data
temppoly = polyfit(data(:,1),data(:,2),11);
rng = -10:0.01:10;
temppolyvals = polyval(temppoly,rng);
temppolyder = polyder(temppoly);
temppolydervals = polyval(temppolyder,rng);
temppolydervals = 1/deg2rad(1)*temppolydervals;

torquepoly0121 = temppoly;
save('torquepoly0121','torquepoly0121')

% Select appropriate figure to edit
figure(figurenumber);
% plot the torque curve
subplot(1,2,1);
hold on;
scatter(data(:,1),data(:,2)); % Scatter the torque measurements
hold on;
plot(rng, temppolyvals);% Plot the polyfit
title('Torque vs deflection');
```

```
xlabel('Deflection [degrees]');
ylabel('Torque [Nm]');
grid on;
handle = gca;
handle.XAxis.FontSize = 12;
handle.YAxis.FontSize = 12;
ylim([-11 11]);

% TODO calculate the stiffness
stiffnesscalculated = calculateStiffness(data);
% Plot the stiffness curve
subplot(1,2,2);
plot(stiffnesscalculated(:,1),stiffnesscalculated(:,2));
hold on;
plot(rng, temppolydervals)% Add the polyfit
title('Stiffness vs deflection');
xlabel('Deflection [degrees]');
ylabel('Stiffness [Nm/rad]');
grid on;
handle = gca;
handle.XAxis.FontSize = 12;
handle.YAxis.FontSize = 12;

end

function stiffnesscurve = calculateStiffness(data)

% Generate a spline from the data over the angle range
steps = 100;
anglerange = 10:-1/(steps/2):-10;
% Interpolate from the given data so that there's a higher resolution
% for numerical differentiation
torinterp = interp1(data(:,1),data(:,2),anglerange,'Spline');

% Numerical differentiation
stiffnesscurve(:,1) = anglerange;
stiffnesscurve(:,2) = zeros((10*steps)+1,1);
for i = 1:10*steps
    stiffnesscurve(i,2) = (torinterp(i+1)-torinterp(i))/(1/steps); %dx/dy
end
stiffnesscurve(:,2) = abs(1/deg2rad(1)*stiffnesscurve(:,2));

% Delete the last row because it does not have proper stiffness
stiffnesscurve(steps*10 + 1, :) = [];

end
```

B.8 Embedded code

This code is used for reading out the strange gauge sensor (HX711), doing formatting and sending this information back to the computer for further processing.

B.8.1 Main.cpp

```
/* mbed Microcontroller Library
 * Copyright (c) 2019 ARM Limited
 * SPDX-License-Identifier: Apache-2.0
 */

#include "mbed.h"
// #include "MODSERIAL.h" // for more efficient serial comm
#include "platform/mbed_thread.h"

#include "StrainGauge.h"

// Design constants
#define ARMRADIUS 0.15 // In meters
#define MVTGRAM 0.00001 // TODO The scaling factor of measured 1 mv to gram

#define AVERAGINGLOOPS 10 // How many samples per measurement time, NOTICE the maximum 24 bit 2comp value is
                          // 8388608, max 31 bit value is 2147483648, so at most 256 averaging loops to guarantee no overflow

// MODSERIAL pc(USBTX, USBRX); // Serial communication with host PC
static Serial pc(USBTX, USBRX);

DigitalIn btn(PTC6);
DigitalOut ledb(LED_BLUE);
DigitalOut ledr(LED1);

// Dat (data output to MCU from hx711)      PTC4 D9
// Clk (Serial input clock)                 PTC12 D8
DigitalIn ADDO(PTC4);
DigitalOut ADSK(PTC12);

// Calibration values for the loadcell
const double LC_Cal_slope = 4.7263;
const double LC_Cal_offset = -0.8488;

// This function converts a given unsigned long number (24 bits) to a signed two's complement (32 bit)
signed long toTwosComplement(unsigned long number);

// These functions do the unit conversions
float bitsToMV(signed long number);
float mVToTorque(float mv);

int main()
{
    pc.printf("\r\n\r\nLogging starts, table format:\r\n\r\nraw value, signed value, Voltage [mV], torque[Nm]\r\n\r\n");
    ;

    while (true) {
        if (!btn.read()) {
            ledb = true;

            // Read the sensor X times, averaging and sending over serial
            signed long result_conv_t = 0; //
            unsigned long result = 0;
            for (int i = 0; i < AVERAGINGLOOPS; i++) {
                result = ReadStrainGauge(); // Read a value
                result_conv_t += toTwosComplement(result); // Convert 24 bit value to two's complement and add
                to the total
            }
            wait_ms(100); // A wait time between samples, with 10 samples and 100ms, this is a 1s
                measurement time which seems a nice compromise
            signed long result_conv = result_conv_t / AVERAGINGLOOPS; // Devide for the average

            // Do the conversion for all the other desired measurements
            float millivolt = bitsToMV(result_conv);
            float torque = mVToTorque(millivolt);

            pc.printf("%lu , %ld, %.4f, %.3f \r\n", result, result_conv, millivolt, torque); // Print the
                result to the user
            // NOTE: The printed 'raw' value is the last measured value, NOT AVERAGED. Because of twos
                complement this cannot easily be averaged without artefacts around the 0 point

            // Change color of LED based on last measurement, just for fun
            if (result_conv < 0) {
                ledr = true;
            } else {
                ledr = false;
            }

            while (!btn.read()) {} // Pause until the button is released
            ledb = false;
        }
        // wait_ms(500);
    }
}
```

```

    }
}

// This function converts a given unsigned long number (24 bits) to a signed two's complement (32 bit)
signed long toTwosComplement(unsigned long number){
    signed long result;
    if( (number&0x00800000) != 0){ // Bit mask the 24th bit (indicates positive or negative number
        // Negative number
        result = (~number) & 0x007FFFFFFF;
        result = -result - 1;
    }else{
        // Positive number
        result = number & 0x007FFFFFFF; // Mask out the 23 first bits as the result
    }
}

return result;
}

// These functions do the unit conversions
float bitsToMV(signed long number){
    return (number*0.00238); // For 40mv full swing with 24 bit = 2.38nV per bit 40/16777216(2^24)
}

float mVToTorque(float mv){
    mv = mv / 1000; // Convert to volts;
    return (LC_Cal_slope*mv + LC_Cal_offset); // Torque = Force * arm = Weight*9.8 * arm = weight_gr
        *0.0098066500286389*arm
}

```

B.8.2 StrainGauge.h

```

/* HX711 MODULE PIN CONNECTIONS
// The HX711 has 5 connections to the MCU
// Vdd (should be the typical IO voltage)      3.3V
// Vcc (power to the chip)                    5V
// Dat (data output to MCU from hx711)
// Clk (Serial input clock)
// Gnd (Power and signal ground)
//
// The other side of the HX711 module has 5 connections, these should match up the colors of the strain gauge
// Red = excitation voltage (5V)
// white = Output positive
// green = Output negative
// black = ground
// Yellow = Shield (ground)
*/

#ifndef _STRAIN_GAUGE_H_
#define _STRAIN_GAUGE_H_

#include <stdbool.h>
#include <stdint.h>

// This function reads a HX711 based strain gauge sensor
// It will pause until the sensor can be read, and return the result
unsigned long ReadStrainGauge(void);

// Pulses the ADSK output, has an optional delay
void pulseADSK();

// These are implementation dependent functions, which make it easier to port to different platforms
// FOR PORTING CHANGE THESE FUNCTIONS
void setADSK(bool on);
bool readADDO();

#endif//_STRAIN_GAUGE_H_

```

B.8.3 StrainGauge.cpp

```

#include "StrainGauge.h"

/* HX711 MODULE PIN CONNECTIONS
// The HX711 has 5 connections to the MCU
// Vdd (should be the typical IO voltage)      3.3V
// Vcc (power to the chip)                    5V
// Dat (data output to MCU from hx711)        PTC4 D9
// Clk (Serial input clock)                  PTC12 D8
// Gnd (Power and signal ground)
//
// The other side of the HX711 module has 5 connections, these should match up the colors of the strain gauge
// Red = excitation voltage (5V)
// white = Output positive
// green = Output negative
// black = ground
// Yellow = Shield (ground)
//
// Datasheet: https://cdn.sparkfun.com/assets/b/f/5/a/e/hx711F\_EN.pdf
// Pin PD_SCK and DOUT are used for data retrieval, input selection, gain selection and power down
// controls. When output data is not ready for retrieval, digital output pin DOUT is high.

```

Design and Development of a Magnetic Non-linear Elastic Element and Control for Progressive Series Actuation

```
// Serial clock input PD_SCK should be low. When DOUT goes to low, it indicates data is ready for
retrieval. By applying 25~27 positive clock pulses at the PD_SCK pin, data is shifted
out from the DOUT output pin.
// Each PD_SCK pulse shifts out one bit, starting with the MSB bit first, until all 24 bits are shifted out.
The 25th pulse at PD_SCK input will pull DOUT pin back to high (Fig.2).
// Input and gain selection are controlled by the number of the input PD_SCK pulses (Table 3).
PD_SCK clock pulses should not be less than 25 or more than 27 within one conversion period
, to avoid causing serial communication error.
*/

#include "mbed.h"

extern DigitalOut ADSK;
extern DigitalIn ADD0;

// This function reads a HX711 based strain gauge sensor
// It will pause until the sensor can be read, and return the result
unsigned long ReadStrainGauge(void){

    //uint8_t SAMPLES[24] = {1,0,0,1,1,0,0,0,1,1,0,1,1,0,1,1,0,0,0,1,0,1,1,1}; // 10017556 inv 2677529 //
    DEBUG
    //int k = 0; // DEBUG

    unsigned long result = 0; // 32 bit result
    //ADD0=1;
    setADSK(false); // Ensure serial clock line is low at the beginning

    while(readADD0()){ // Wait until data is ready for retrieval, signified by a low bit on D0

    for(int8_t i = 23; i>-1; i--){
        pulseADSK(); // Give a pulse (Immediately pull SK low again to avoid going into sleep mode!!)

        //if(SAMPLES[k] == 1){ // DEBUG
        if(readADD0()){
            result |= (1<<i); // Results are clocked in MSB first
        }
        //k++; // DEBUG
    }

    // Last pulses for settings
    //settingsselect=1 => inp A, gain 128
    //settingsselect=2 => inp B, gain 32
    //settingsselect=3 => inp A, gain 64
    const uint8_t settingsselect = 1;
    for(uint8_t k = 0; k<settingsselect ; k++){
        pulseADSK(); // Pulse as many as required
    }

    return result;
}

// Pulses the ADSK output, has an optional delay
void pulseADSK(){
    // Note that SK has to be pulsed between 0.2 and 50us (normal = 1)
    setADSK(true);
    wait_us(2);
    setADSK(false);
}

// These are implementation dependent functions, which make it easier to port to different platforms
// FOR PORTING CHANGE THESE FUNCTIONS
void setADSK( bool on){
    ADSK = on;
}
bool readADD0(){
    return ADD0.read();
}
```