

**A CLASSIFICATION AND PERFORMANCE
COMPARISON OF FORCE-GENERATORS FOR
APPLICATION IN EXOSKELETONS AND ROBOTIC
MANIPULATORS**

by

S.J. WAGEMAKER

In partial fulfillment of the requirements of the degree of

Master of Science
in Mechanical Engineering

at the Delft University of Technology.

To be defended publicly on Friday September 13, 2019 at 10:30 AM.

Student number: 4299140

Thesis committee:

dr. ir. D.H. Plettenburg,	TU Delft, Chair
Prof. dr. ir. J.L. Herder,	TU Delft
ir. W.W.P.J. van de Sande,	TU Delft, Supervisor

Independent members:

ir. B.M. Wisse,	Laevo B.V.
-----------------	------------

Copyright © 2019 by S.J. Wagemaker

ISBN 000-00-0000-000-0

An electronic version of this dissertation is available at
<http://repository.tudelft.nl/>.

CONTENTS

Preface	v
1 Introduction	1
2 Paper I: Bottom-up classification method of force generators applied to perfect static balancers with one non-auxiliary revolute joint	5
3 Paper II: Analytical comparison of optimized spring-to-mass balancers incorporating realistic springs	13
4 Discussion	31
5 Conclusion	33
A Appendix A: Functions of an adjustment mechanism in an exoskeleton	41
B Appendix B: Functional requirements and strategies of an adjustment mechanism in an exoskeleton	49
C Appendix C: Design of an adjustment mechanism applied in an upper extremity exoskeleton	51
C.1 Path of spring attachment point	51
C.2 Locking and unlocking the spring attachment point	52
C.3 Energy storage	53
C.3.1 Limitations and further research	55
C.4 Design choices	56
D Appendix D: Time strategies of an adjustment mechanism in an exoskeleton	61
E Appendix E: Properties of springs from database	63
F Appendix F: Circle optimization	67
G Appendix G: Net adjustment energy	69
H Appendix H: MATLAB code of models	73

PREFACE

This thesis concludes my Master BioMechanical Design at the faculty of Mechanical, Maritime and Material Engineering at the Delft University of Technology. I enjoyed learning about all kind of Engineering facets: the classical technical aspects such as dynamics, but also the human ones such as human-machine interfaces. I conducted my internship, literature review and thesis research at the company Laevo. One of the reasons I chose to do this was to experience engineering in the 'real world'. I learned how to deal with the (often contradicting) interests of the company and university. Furthermore, I learned that it is not always about the optimal mechanical solution, but human, financial and commercial aspects need to be taken into account as well.

I would like to express my gratitude to the people that helped me through the last period of my student career. I would like to thank my daily supervisor, Werner, for the countless useful discussions about my work. I would like to thank Just and Dick for the constructive feedback and for motivating me when things did not work out the way I wanted. I would like to thank the people from Laevo, especially Mike and Boudewijn, for their support and for supplying me with everything I needed to conduct my research. I would like to thank the fellow students at Laevo and InteSpring for the feedback during the weekly presentations and the fun we had during the (many) breaks. Finally, I would like to thank my friends and family for their never ending support and interest in my work.

*S.J. Wagemaker
Delft, September 2019*

1

INTRODUCTION

In many engineering applications, it is desirable to eliminate or reduce the effect of gravity on a particular part of a mechanism to reduce the required operating power [4]. The compensation of the effects of gravity is referred to as *gravity balancing* or *static balancing*. The overall goal of this thesis is to supply an overview of different mechanisms that can supply static balance and to investigate the implementation of these mechanisms in applications in which the payload varies, especially in exoskeletons and robotic manipulators.

Two basic options for static balancing can be distinguished: active and passive approaches. The former approach incorporates active actuators, such as motors, and control systems which require energy to operate. The latter approach incorporates non-powered devices, such as springs, to compensate the gravity effects. Generally, active solutions are easier to control compared to the passive solutions, but rely on power sources to operate. The passive solutions are relatively simple and therefore more reliable compared to active solutions. Generally, when both active and passive balancing technologies meet the system's requirements, passive balancing technologies are preferred over active ones [21].

Passive balancers can be categorized into mass-to-mass and spring-to-mass balancers [1, 9]. Mass-to-mass balancers incorporate counterweights to keep the center of mass of the system stationary and thereby balance the mass. Applications include drawbridges, elevators, cranes, rehabilitation devices [5, 32] (Figure 1.1a) and robotic manipulators [28, 34, 33]. A disadvantage of mass-to-mass balancing is the relatively large inertia and mass of the system, which makes it less suitable to be applied in mobile applications.

Spring-to-mass balancers incorporate one or multiple springs to balance the mass. In spring-to-mass balancers, the total potential energy is constant by exchanging energy between the spring and mass. They are applied in robotic manipulators [11, 22, 23], exoskeletons and rehabilitation devices [15, 10, 25] (Figure 1.1b) to reduce the operating power. Spring-to-mass balancers are generally more lightweight than mass-to-mass bal-

ancers [9], making them interesting for mobile applications. However, a transmission is required to transform the potential energy curve of the spring, which makes spring-to-mass balancers generally more complex than mass-to-mass balancers.

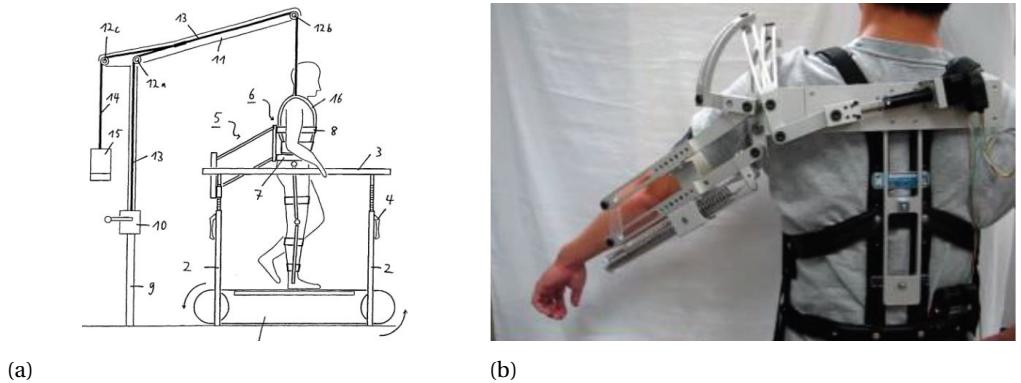


Figure 1.1: Example of (a) a mass-to-mass balancer in a rehabilitation device [5] and (b) a spring-to-mass balancer in an exoskeleton [16]

A classification that identifies new design opportunities can be a valuable tool in the design process [6]. Existing classifications of spring-to-mass balancers (on basis of their structure [15], applications [21], inertia force and performance [20, 31] and type of transmission [26]) are top-down structured. Top-down structured classifications focus on the existing designs, and therefore supply very limited guidance in the design of innovative solutions. Bottom-up structured classifications, however, start with a systematic identification of the complete design space. By doing so, all possible design opportunities become clear, which then can be compared to the existing designs to identify novel design opportunities.

Generally, a balancer is designed to balance a specific payload [9]. When another payload is to be balanced, the balancer needs to be adjusted. This can be the case in balancers applied in exoskeletons and robotic manipulators. The load on the balancer in those application varies (e.g. when an object is picked up) and for optimal support quick adjustment is required at any instance. Besides, the amount of energy involved in the adjustment should be limited to enable the use of lightweight and compact actuators. Various methods to adjust a balancer have been proposed [2], but the amount of energy that is involved in the adjustment has not been discussed. In addition, so called energy-free adjustment methods, in which no external energy is involved, have been introduced [3, 8, 9, 15]. However, in these methods, the balancer can be adjusted in one balancer position only and therefore do not meet the above mentioned requirements. Besides, balancers without a transmission that can supply perfect balance over the complete range of motion incorporate custom springs [2] which increases their complexity.

THESIS OUTLINE

This thesis consists of five chapters. Chapter 2 and 3 each present an independent paper.

In Chapter 2, the focus is on the classification of force generators, which is the family of mechanisms to which balancers belong to [19]. The goal of this chapter is to introduce a bottom-up classification method to identify the complete solutions space of transmissions in force generators based on their transfer function. Using this method, spring-to-mass balancers with one non-auxiliary revolute joint will be classified to identify their working principles and find new design opportunities.

In Chapter 3, the focus is on balancers applied in exoskeletons and robotic manipulators. The goal of this chapter is twofold. The first goal is to introduce the optimal balancer adjustment types for application in exoskeletons and robotic manipulators. The second goal is to compare their performance based on the balancing quality, energy involved in adjustment and dimensions through an analytical MATLAB model. The introduced adjustment types incorporate a commercial available spring with realistic properties and their performance is considered in their complete range of motion.

The results of the complete thesis research are discussed in Chapter 4, and in Chapter 5, the main conclusions are highlighted. This thesis ends with additional information about the performed research, which can be found in Appendices A to H.

2

PAPER I: BOTTOM-UP CLASSIFICATION METHOD OF FORCE GENERATORS APPLIED TO PERFECT STATIC BALANCERS WITH ONE NON-AUXILIARY REVOLUTE JOINT

Bottom-up classification method of force generators applied to perfect static balancers with one non-auxiliary revolute joint

S.J. Wagemaker

Masters BioMechanical Design

Delft University of Technology, The Netherlands

September 2018

s.j.wagemaker@tudelft.nl

Abstract— Since a force generator is not unique, optimal design is difficult. Current classifications of force generators solely classify existing mechanisms. Since the complete solution space is not identified, these classifications provide limited guidance in optimal design.

A classification is proposed in which the complete solution space is systematically identified by exploring all possible combinations of mechanism parameters for which the desired force-displacement behaviour is exhibited by the force generator. Firstly, desired output of the force generator (load function) is determined. Secondly, the motor parameters (determined by the type of motor) are identified which form the motor function. Thirdly, all possible combinations of the motor parameters for which the motor function equals the load function are identified. From these combinations, the working principle of the force generators is deduced. Lastly, the combinations can be compared to existing force generators to find new design opportunities.

The classification is applied to planar spring-to-mass perfect static balancers with one non-auxiliary revolute joint. Five out of 27 possible parameter combinations are found in existing balancers. Considering linear springs only, five out of seven combinations are found in existing mechanisms; the two remaining combinations are considered as new design possibilities. The working principles of the balancers found in literature include the use of:

- ideal springs,
- trigonometric identities,
- cam mechanisms,
- cancellation of (undesired) terms and
- phase shifted trigonometric functions.

NOMENCLATURE

F	Force [N]
i, j, k	Unit vectors in x, y, z direction
M	Moment [Nm]
r	Moment arm [m]
u	Deformation [m]
φ_i	Degree of freedom i
a_q	Motor parameter q
g	Gravity acceleration [m/s^2]
h_k	Height of mass k [m]
k	Stiffness [N/m]
L, l	Length [m]
l_0	Free length [m]
m_k	Mass k [kg]

I. INTRODUCTION

A force generator is a mechanism that exhibits a specific force-deflection behaviour. Generally, a force generator is not unique: different mechanisms with identical behaviour exist, which makes optimal design difficult. The goal of this study is to simplify optimal design of force generators by proposing a systematic classification method, based on combination of actuator parameters by which the desired force-deflection behaviour is obtained.

The classification is applied to force generators that exert a force to statically balance a mass (hereinafter referred to as balancers). The balancers that are considered meet the following criteria. The balancers:

- are perfectly balanced,
- use springs to generate force,
- work in two-dimensional space and
- have one revolute joint.

Herder [1] developed a classification of one-degree-of-freedom balancers based on the adjustability to changing payload in an energy-free manner. Van Drunen [2] used a similar approach to classify adjustable one-degree-of-freedom balancers and also considered energy-costly adjustment. Although the adjustability follows from the mechanism principle, the mechanism principles are not discussed and classified. Rijff [3] classified vertical translating gravity balancers using the number of springs and the number of mechanisms. Rijff identifies distinct mechanisms, but focuses on the number of mechanisms instead of the nature of the mechanisms. Baradat [4] classified both one and two degree of freedom-balancers based on the force-generating element and the transmission between the balancing force and the load. However, Baradat did not investigate why specific transmission ratios are used.

All of the above classifications are top-down structured: starting at the existing balancers, an overview is created. Top-down classifications do not facilitate optimal design, since existing mechanisms are considered only, whereas the optimal design could be yet non-existing. The proposed classification has a bottom-up structure: the complete design space is systematically described and compared to the mechanisms found in literature. Furthermore, the use of specific mechanism principles can be explained by the desired load-displacement behaviour.

II. METHOD

A. Literature search

The balancers that are considered in this study meet the requirements described in the Section I and are published in scientific articles, found on Google Scholar and Scopus. Combinations of the search terms that are listed in Table I are used. The following terms are excluded: active, control, motor and *electr*. The cited references of the found articles are reviewed to find other relevant articles. Relevant patents are reviewed but not considered in the classification because correct functioning of the patented mechanisms cannot be guaranteed.

TABLE I: Keywords used in literature search

AND	AND	AND
static	balancing	device
gravity	counterbalance	mechanism
spring	"constant force"	compensation
	"constant load"	equilibrator
	equipoising	
	support	

B. Classification

The motor-load framework

The motor-load framework is used to classify the force generators. The load can be either prescribed by a physical system (e.g. a mass) or by the designer (e.g. a custom load-displacement curve). The behaviour of the force or moment exerted by the load is described by the load function, which is a function of the degrees of freedom ($\varphi_1, \dots, \varphi_N$):

$$G_{load} \equiv f(\varphi_1, \dots, \varphi_N) \quad (1)$$

The motor is formed by one or multiple mechanical (actuator) elements. The behaviour of the force or moment exerted by the motor is described by the motor function and is dependent on the degrees of freedom and the motor parameters (a_1, \dots, a_q):

$$G_{motor} \equiv g(\varphi_1, \dots, \varphi_N, a_1, \dots, a_q) \quad (2)$$

The motor parameters, which follow from the type of motor, are combined so that the motor function equals the load function:

$$G_{motor} = G_{load} \quad (3)$$

$$g(\varphi_1, \dots, \varphi_N, a_1, \dots, a_q) = f(\varphi_1, \dots, \varphi_N) \quad (4)$$

Force generators can be classified by performing the following steps:

- 1) Determination of the load function
- 2) Identification of the motor parameters
- 3) Identification of the set of possible combinations of the motor parameters to form the load function
- 4) Allocation of existing mechanisms to the combinations of motor parameters

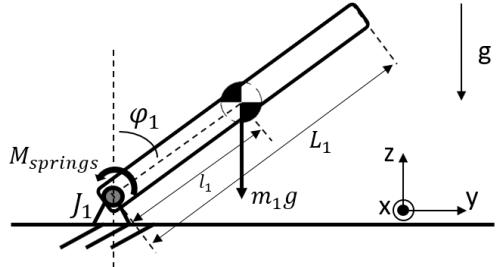


Fig. 1: One degree of freedom-balancer

Combinations of motor parameters that are not found in existing mechanisms are considered as new design opportunities.

C. Classification of one degree of freedom-balancers

Load function

Consider a two-dimensional balancer consisting of one non-auxiliary revolute joint, a massless link and a concentrated mass (Figure 1). The mass describes a circular path about the joint and the load function can be described by the moment of the mass about the revolute joint (J_1):

$$\begin{aligned} G_{load} &= M_{mass} \\ &= -mg\mathbf{k} \times (l_1 \sin(-\varphi_1)\mathbf{j} + l_1 \cos(-\varphi_1)\mathbf{k}) \\ &= -mgl_1 \sin(\varphi_1)\mathbf{i} \end{aligned} \quad (5)$$

in which m , g , l_1 and φ_1 represent the mass, gravity acceleration, distance between the pivot and the mass and the angle between the vertical and the line between the mass and the pivot respectively.

Motor function

Supposing the motor is formed by n springs between the link and the external world, the motor function can be described by the moment of the spring force(s) about the revolute joint and is equal to:

$$G_{motor} = M_{springs} = \sum_{i=1}^n \mathbf{F}_{si} \times \mathbf{r}_i = \sum_{i=1}^n k_i \mathbf{u}_i \times \mathbf{r}_i \quad (6)$$

in which \mathbf{F}_{si} , k_i , \mathbf{u}_i and \mathbf{r}_i represent the force, stiffness, deformation and moment arm of spring i respectively. The latter three are considered as the motor parameters of the system.

In the case of a balancer consisting of one spring, the (cross) product of the motor parameters k_1 , u_1 and r_1 has to equal the load function:

$$k_1 \mathbf{u}_1 \times \mathbf{r}_1 = -mgl_1 \sin(\varphi_1)\mathbf{i} \quad (7)$$

The motor function of a balancer consisting of multiple springs is formed by the summation of the motor functions

corresponding to the individual springs. In the case of a two-spring balancer, this results in:

$$k_1 \mathbf{u}_1 \times \mathbf{r}_1 + k_2 \mathbf{u}_2 \times \mathbf{r}_2 = -mgl_1 \sin(\varphi_1) \mathbf{i} \quad (8)$$

III. RESULTS

Mechanisms that incorporate either one or two springs are found in literature. The combinations of motor parameters that form the load function of one-degree-of-freedom balancers are listed in Table II. In the second last column, the principle that is used to construct the load function by the motor parameters is listed. The principles are further discussed in Section IV. The existing balancers are represented by the name of the author of the article in which they are found.

IV. DISCUSSION

As shown in Table II, the balancers found in literature rely on one or two of the following mechanism principles: the use of an ideal spring, a trigonometric identity, a cam, the cancellation of terms or phase shifted trigonometric functions in the motor function. The mechanism principles are explained by means of discussing the balancers found in literature.

A. One-spring balancers

The set of balancers that incorporate one spring is formed by the balancers of Wisse [5], Hung [10], Shieh [11] and Lee [12].

Ideal springs (Wisse [5])

In the balancer of Wisse (Figure 2), the moment arm of the spring force can be written as:

$$r_1 = \frac{C \sin(\varphi_1)}{L_{\text{spring}}(\varphi_1)} = \frac{C \sin(\varphi_1)}{l_{01} + u_1(\varphi_1)} \quad (9)$$

in which $L_{\text{spring}}(\varphi_1)$, l_{01} and $u_1(\varphi_1)$ represent the length, free-length and deformation of the spring respectively. This balancer, in contrast to the other balancers, uses an ideal spring. An ideal spring is characterized by exerting a force which is proportional to the length of the ideal spring rather than the elongation. Furthermore, the free length is equal to zero in an unloaded configuration [6]. Since the free length of the spring is zero, the deformation of the spring cancels out in the motor function, resulting in the desired load function. Consequently, the balancer only provides perfect balance when an ideal spring is incorporated. In practice, ideal springs are difficult to manufacture and often limit the working range of the mechanism [7]. Therefore, in most designs, the properties of an ideal spring are emulated by wire-pulley arrangements [8] [9].

Trigonometric identities (Hung [10], Shieh [11])

Interestingly, trigonometric identities can be used to obtain the load function. The balancers by Hung (Figure 3) and Shieh (Figure 4) both use the double-angle identity:

$$2 \sin(\varphi) \sin(\varphi) = \sin(2\varphi) \quad (10)$$

A transmission is required to obtain the desired angular transmission ratio of 1 : 2. In the balancers of Hung, this transmission is formed by a combination of a cardan-gear and a planetary gear whereas in the balancer of Shieh an adaption of the Scotch yoke mechanism is used.

Cam mechanisms (Lee [12])

The balancer of Lee (Figure 7) incorporates an interior cam and a compression spring. The deformation of the spring and the moment arm of the spring force are determined by the shape of the cam. In the balancer of Lee, both the spring deformation and the moment arm are a function of $\sqrt{\sin(\varphi)}$, so that multiplication of those parameters in the motor function equals the desired load function. Although the design of the cam profile is limited by various parameters [13] [14] [15], a great variety of profiles can be constructed [16]. Furthermore, Schroeder [17] showed that a combination of multiple cams can be used. Although an additional cam increases the mechanism complexity, the radius of the cams can be small compared to the radius of a single cam.

B. Two-spring balancers

In mechanisms that incorporate two springs, the moments exerted by the two springs about the revolute joint are summed to obtain the motor function. The summation enables the cancellation of undesired terms in the motor function. Consequently, phase shifted trigonometric functions can be combined to obtain the desired motor function. The balancers that incorporate two springs include the balancers of Takesue [18] and te Riele [19].

Cancellation of terms (Takesue [18])

The balancer of Takesue (Figure 5) incorporates a pair of orthogonal springs: a horizontal and a vertical spring. The moment about the joint created by the horizontal spring cancels out a (undesired) part of the moment exerted by the vertical spring, resulting in the desired motor function.

Phase shifted trigonometric functions (Te Riele [19])

The balancer of Te Riele (Figure 6) incorporates a left and a right spring and relies on the cancellation of terms in the motor function and phase shifted trigonometric functions. The right spring cancels out an undesired part of the moment about the joint of the left spring (or vice versa). The two springs are connected to a 45-45-90 triangle, which results in the relation between the angular degree of freedom of the mass (φ) and the angle of left (ψ_1) and the right (ψ_2) spring (measured counterclockwise from the vertical, symbols according to [19]):

$$\psi_1 = \frac{\pi}{4} - \varphi \quad (11)$$

$$\psi_2 = \frac{\pi}{4} + \varphi \quad (12)$$

The angles of the two springs (which are phase shifted with respect to the angle of the mass) are used to obtain the desired motor function according to:

$$\cos\left(\frac{\pi}{4} - \varphi\right) - \cos\left(\frac{\pi}{4} + \varphi\right) = \sqrt{2}\sin(\varphi) \quad (13)$$

C. Spring linearity

Table II lists the possible combinations of parameters with which the desired motor function of one degree of freedom balancers can be obtained. As can be seen in the Table, only five of the 27 parameter combinations are found in literature. All balancers found in literature use linear springs ($k \neq f(\varphi)$). In fact, the general purpose of the force generators found in literature is to convert the linear behavior of the spring to a nonlinear behavior using a transmission. For this reason, the mechanisms with a non-constant stiffness can be excluded from Table II, resulting in Table III. As seen in the latter Table, five out of the seven parameter combinations are found in existing mechanisms. The mechanisms corresponding to the two remaining combinations are not found in literature yet and might be constructed by altering the mechanisms of Wisse or Hung and Shieh respectively. Herder [6] described modification operations that may be used to accomplish this.

Recently, various mechanisms are developed with which nonlinear spring behaviour can be obtained [20]. Moreover, the upcoming field of compliant mechanisms offers a solution. De Jong [16] identified various compliant mechanisms that can exert a nonlinear force. In the case of balancers, constructing a full range sinusoidal force-displacement curve may be not feasible (yet), but it can be approximated for a specific part of the range using compliant mechanisms.

D. Extension of the motor-load framework

The motor-load framework is applied on balancers by using moment equilibrium as balancing condition. For a perfect balanced mechanism, it also holds that the total potential energy of the system is constant [6]. In this case, the load and motor function are equal to the potential energy of the p masses and the potential energy of the n springs respectively:

$$G_{load} = \sum_{j=1}^p m_j g h_j \quad (14)$$

$$G_{motor} = \sum_{i=1}^n 0.5 k_i u_i^2 \quad (15)$$

in which h_j represents the vertical height of mass j . Consequently, instead of adding up to zero (as in Equation 3), the motor and load function need to add up to any constant value.

In this study, the motor-load framework is used to classify balancers with one non-auxiliary revolute joint. However, balancers with any number of non-auxiliary joints can be

classified. It is expected that the number of load and motor functions is equal to the number of degrees of freedom of the system.

Furthermore, the framework can be used to classify mechanisms with any kind of prescribed force-displacement curve. Applications include angular and translational gravity balancing, orthoses and prostheses. Besides, three-dimensional mechanisms can be classified using this framework. In this case, the load and motor function are defined in three-dimensional space.

E. Further development

In further development, the use of trigonometric identities as a basis to design a gravity balancer can be interesting. Besides, the extension of the classification to two or multiple degree of freedom mechanisms will provide insight in the working principles and can therefore decrease the mechanism complexity.

V. CONCLUSION

A method by which force generators can systematically be classified, by identifying all possible combinations of actuator and mechanism parameters that lead to the desired output function, is introduced. Since the complete set of design possibilities is identified, the optimal mechanism design can be selected from this set.

Regarding one degree of freedom planar perfect balancers, 27 design principles are identified of which five are found in literature. Considering springs with a linear stiffness only, five out of seven possible design principles are found in existing mechanisms. The found balancers and especially the distinct working principles are discussed. The working principles include the use of:

- ideal springs,
- trigonometric identities,
- cam mechanisms,
- cancellation of (undesired) terms and
- phase shifted trigonometric functions.

VI. ACKNOWLEDGEMENT

This work is written as part of the Masters graduation thesis of the author. However, this study could not have been performed without the critical and useful feedback of J.L. Herder, D.H. Plettenburg and W.W.P.J. van de Sande. Furthermore, the help of the people from Laevo B.V. and InteSpring B.V. has been really useful; especially the meetings with F.G. Baas, in which all challenges faced while writing this report have been discussed.

REFERENCES

- [1] Herder, J.L., Barents, R., Wisse, B.M., van Drosser, W.D. (2011) *Efficiently variable zero stiffness mechanisms*, In: 4th International Workshop on Human-Friendly Robotics, pp. 2-3.
- [2] Van Drunen, A. (2008) *Balancing = Beautiful*, Master-Thesis, Delft University of Technology, The Netherlands.
- [3] Rijff, B.L. (2009) *An overview of vertically translating gravity balancers*, Literature Research, Delft University of Technology, The Netherlands.

- [4] Baradat, C., Arakelian, V.A., Briot, S., Guegan, S. (2008) *Design and Prototyping of a New Balancing Mechanism for Spatial Parallel Manipulators*, Journal of Mechanical Design, Vol. 108.
- [5] Wisse, B.M., van Drosser, W.D., Barents, R., Herder, J.L. (2007) *Energy-Free Adjustment of Gravity Equilibrators Using the Virtual Spring Concept*, In: Proceedings of the 2007 IEEE 10th International Conference on Rehabilitation Robotics.
- [6] Herder, J.L. (2001) *Energy free systems: Theory, conception and design of statically balanced spring mechanisms*, Ph.D. Thesis, Delft University of Technology, The Netherlands.
- [7] Barents, R., Schenck, M., van Dorsser, W.D., Wisse, B.M. (2011) *Spring-to-spring Balancing as Energy-free Adjustment Method in Gravity Equilibrators*, In: Journal of Mechanism Design, vol. 133.
- [8] Gund, K. (1995) *Krfeausgleichsvorrichtung*, Patent No. DE936966.
- [9] Soethoudt, B. (1998) *A New Ideal Spring Arrangement*, Internal Report, Delft University of Technology.
- [10] Hung, Y.C., Kuo, C.H. (2017) *A Novel One-DoF Gravity Balancer Based on Cardan Gear Mechanism*, In: Wenger, P., Flores, P. (eds) New Trends in Mechanism and Machine Science. Mechanisms and Machine Science, vol 43.
- [11] Shieh, W.B., Chou, B.S. (2015) *A Novel Spring Balancing Device on the Basis of a Scotch Yoke Mechanism*, In: Doroftei, I., Oprisan, C., Pisla, D., Lovasz, E.C. (eds) New Advances in Mechanism and Machine Science. Mechanisms and Machine Science, vol. 57.
- [12] Lee, G., Lee, D., Oh, L. (2018) *One-Piece Gravity Compensation Mechanism Using Cam Mechanism and Compression Spring*, In: International Journal of Precision Engineering and Manufacturing - Green Technology, vol. 5, pp. 425-420.
- [13] Kobayashi, K. (2001) *New Design Method for Spring Balancers*, In: Transactions of the ASME, vol. 123, pp. 494-500.
- [14] Tidwell, P.H., Bandukwala, N., Dhande, S.G., Reinholtz, C.F., Webb, G. (1994) *Synthesis of Wrapping Cams*, In: Transactions of the ASME, vol. 116, pp. 634-638.
- [15] Simionescu, I., Ciupitu, L. (2000) *The static balancing of the industrial robot arms, Part II: Continuous balancing*, In: Mechanism and Machine Theory, vol. 35, pp. 1299-1311.
- [16] De Jong, M.G. (2018) *Synthesis of a Force Generator using Two-Fold Tape Loops*, Master-Thesis, Delft University of Technology, The Netherlands.
- [17] Schroeder, J.S., Perry, J.C. (2017) *Development of a Series Wrapping Cam Mechanism for Energy Transfer in Wearable Arm Support Applications*, In: International Conference on Rehabilitation Robotics (ICORR), pp. 585-590.
- [18] Takesue, N., Ikematsu, T., Murayama, H., Fujimoto, H. (2011) *Design and Prototype of Variable Gravity Compensating Mechanism (VGCM)*, In: Journal of Robotics and Mechatronics, vol. 23, pp. 249-257.
- [19] te Riele, F.L.S., Herder, J.L. (2001) *Perfect Static Balance with Normal Springs*, In: Proceedings of the ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE), pp. 18.
- [20] Vanderborght, B. et al. (2013) *Variable impedance actuators: A review*, In: Robotics and Autonomous Systems, vol. 61, pp. 1601-1614.

TABLE II: Results one degree of freedom-balancers

Abbreviations: ideal spring (is), trigonometric identity (tri), cam mechanism (cm), cancellation of terms (cl), phase shifted function (ps)

	k_1	u_1	r_1	k_2	u_2	r_2	Principle	Author
1	$C_1 \sin(\varphi)$	C_2	C_3	-	-	-	is	
2	C_1	$C_2 \sin(\varphi)$	C_3	-	-	-	is	
3	C_1	C_2	$C_3 \sin(\varphi)$	-	-	-	is	Wisse [5]
4	C_1	$C_2 \sin(0.5\varphi)$	$C_3 \cos(0.5\varphi)$	-	-	-	tri	Hung [10], Shieh [11]
5	C_1	$C_2 \cos(0.5\varphi)$	$C_3 \sin(0.5\varphi)$	-	-	-	tri	
6	$C_1 \sin(0.5\varphi)$	C_2	$C_3 \cos(0.5\varphi)$	-	-	-	tri	
7	$C_1 \cos(0.5\varphi)$	C_2	$C_3 \sin(0.5\varphi)$	-	-	-	tri	
8	$C_1 \sin(0.5\varphi)$	$C_2 \cos(0.5\varphi)$	C_3	-	-	-	tri	
9	$C_1 \cos(0.5\varphi)$	$C_2 \sin(0.5\varphi)$	C_3	-	-	-	tri	
10	C_1	$C_2 \sqrt{\sin(\varphi)}$	$C_3 \sqrt{\sin(\varphi)}$	-	-	-	cm	Lee [12]
11	$C_1 \sqrt{\sin(\varphi)}$	C_2	$C_3 \sqrt{\sin(\varphi)}$	-	-	-	cm	
12	$C_1 \sqrt{\sin(\varphi)}$	$C_2 \sqrt{\sin(\varphi)}$	C_3	-	-	-	cm	
13	C_1	$\sin(\varphi)$	$\cos(\varphi)$	C_4	$C_5 \cdot \cos(\varphi)$	$\sin(\varphi)$	cl	Takesue [18]
14	$\cos(\varphi)$	C_2	$\sin(\varphi)$	C_4	$C_5 \cdot \cos(\varphi)$	$\sin(\varphi)$	cl	
15	$\sin(\varphi)$	$\cos(\varphi)$	C_3	C_4	$C_5 \cdot \cos(\varphi)$	$\sin(\varphi)$	cl	
16	C_1	$\sin(\varphi)$	$\cos(\varphi)$	$C_4 \cdot \cos(\varphi)$	C_5	$\sin(\varphi)$	cl	
17	$\cos(\varphi)$	C_2	$\sin(\varphi)$	$C_4 \cdot \cos(\varphi)$	C_5	$\sin(\varphi)$	cl	
18	$\sin(\varphi)$	$\cos(\varphi)$	C_3	$C_4 \cdot \cos(\varphi)$	C_5	$\sin(\varphi)$	cl	
19	C_1	$C_2 + \sin(\pi/4 - \varphi)$	$\cos(\pi/4 - \varphi)$	C_4	$C_5 + \sin(\pi/4 + \varphi)$	$\cos(\pi/4 + \varphi)$	cl, ps	Te Riele [19]
20	$\cos(\pi/4 - \varphi)$	C_2	$C_3 + \sin(\pi/4 - \varphi)$	C_4	$C_5 + \sin(\pi/4 + \varphi)$	$\cos(\pi/4 + \varphi)$	cl, ps	
21	$C_1 + \sin(\pi/4 - \varphi)$	$\cos(\pi/4 - \varphi)$	C_3	C_4	$C_5 + \sin(\pi/4 + \varphi)$	$\cos(\pi/4 + \varphi)$	cl, ps	
22	C_1	$C_2 + \sin(\pi/4 - \varphi)$	$\cos(\pi/4 - \varphi)$	$\cos(\pi/4 + \varphi)$	C_5	$C_6 + \sin(\pi/4 + \varphi)$	cl, ps	
23	$\cos(\pi/4 - \varphi)$	C_2	$C_3 + \sin(\pi/4 - \varphi)$	$\cos(\pi/4 + \varphi)$	C_5	$C_6 + \sin(\pi/4 + \varphi)$	cl, ps	
24	$C_1 + \sin(\pi/4 - \varphi)$	$\cos(\pi/4 - \varphi)$	C_3	$\cos(\pi/4 + \varphi)$	C_5	$C_6 + \sin(\pi/4 + \varphi)$	cl, ps	
25	C_1	$C_2 + \sin(\pi/4 - \varphi)$	$\cos(\pi/4 - \varphi)$	$C_4 + \sin(\pi/4 + \varphi)$	$\cos(\pi/4 + \varphi)$	C_6	cl, ps	
26	$\cos(\pi/4 - \varphi)$	C_2	$C_3 + \sin(\pi/4 - \varphi)$	$C_4 + \sin(\pi/4 + \varphi)$	$\cos(\pi/4 + \varphi)$	C_6	cl, ps	
27	$C_1 + \sin(\pi/4 - \varphi)$	$\cos(\pi/4 - \varphi)$	C_3	$C_4 + \sin(\pi/4 + \varphi)$	$\cos(\pi/4 + \varphi)$	C_6	cl, ps	

TABLE III: Results one degree of freedom-balancers with linear springs

Abbreviations: ideal spring (is), trigonometric identity (tri), cam mechanism (cm), cancellation of terms (cl), phase shifted function (ps)

	k_1	u_1	r_1	k_2	u_2	r_2	Principle	Author
2	C_1	$C_2 \sin(\varphi)$	C_3	-	-	-	is	
3	C_1	C_2	$C_3 \sin(\varphi)$	-	-	-	is	Wisse [5]
4	C_1	$C_2 \sin(0.5\varphi)$	$C_3 \cos(0.5\varphi)$	-	-	-	tri	Hung [10], Shieh [11]
5	C_1	$C_2 \cos(0.5\varphi)$	$C_3 \sin(0.5\varphi)$	-	-	-	tri	
10	C_1	$C_2 \sqrt{\sin(\varphi)}$	$C_3 \sqrt{\sin(\varphi)}$	-	-	-	cm	Lee [12]
13	C_1	$\sin(\varphi)$	$\cos(\varphi)$	C_4	$C_5 \cdot \cos(\varphi)$	$\sin(\varphi)$	cl	Takesue [18]
19	C_1	$C_2 + \sin(\pi/4 - \varphi)$	$\cos(\pi/4 - \varphi)$	C_4	$C_5 + \sin(\pi/4 + \varphi)$	$\cos(\pi/4 + \varphi)$	cl, ps	Te Riele [19]

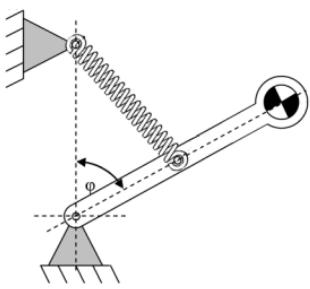


Fig. 2: Adapted from Wisse [5]

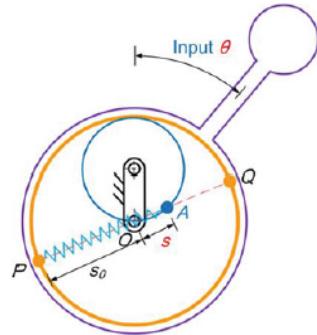


Fig. 3: Hung [10]

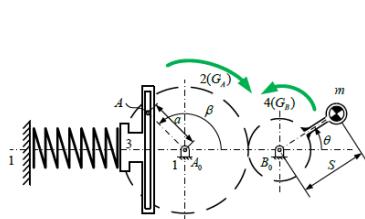


Fig. 4: Shieh [11]

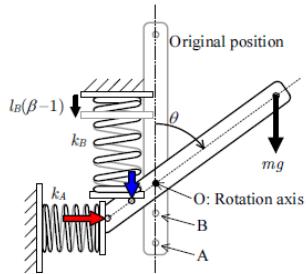


Fig. 5: Takesue [18]

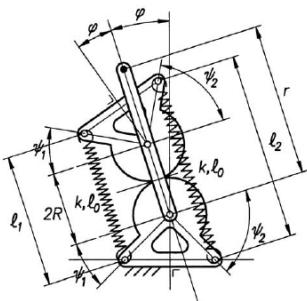


Fig. 6: te Riele [19]

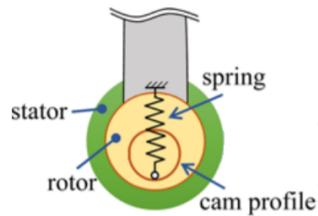


Fig. 7: Lee [12]

3

PAPER II: ANALYTICAL COMPARISON OF OPTIMIZED SPRING-TO-MASS BALANCERS INCORPORATING REALISTIC SPRINGS

Analytical comparison of optimized spring-to-mass balancers incorporating realistic springs

S.J. Wagemaker

Masters BioMechanical Design

Delft University of Technology, The Netherlands

September 2019

s.j.wagemaker@tudelft.nl

Abstract— Exoskeletons and robotic manipulators incorporate static balancers to reduce the required operating power. Since the payload of the balancer varies (e.g. when an object is picked up), quick adjustment at any instance is required for optimal power reduction. It is important to limit the energy involved in the adjustment, to minimize the size and mass of the energy storage and actuators. Multiple methods to adjust these balancers have been introduced in literature, but the amount of energy involved in the adjustment in the complete range of motion has not been investigated yet. Furthermore, existing balancers either incorporate springs with special behavior or cannot balance over their complete range of motion. The goal of this study is twofold. The first goal is to introduce balancer adjustment types that are suitable for application in exoskeletons and robotics and incorporate commercial available springs. The second goal is to compare their performance in the complete range of motion based on the balancing quality, energy involved in adjustment and dimensions through an analytical MATLAB model. The results show clear differences between the performance of the considered adjustment types. Perfect static balance in the complete range of motion cannot be obtained by the adjustment types due to the initial tension, null-length and maximum energy storage of the spring. However, the balancing quality of two adjustment types is close to perfect (within 5% error). The minimum amount of energy involved in adjustment is equal to the difference between the potential energy of the two balanced masses. Two adjustment types approach this minimum (within an error of 10%) and the third adjustment type is the most compact. However, its balancing quality is poor for large differences between the moments of the balanced masses and its adjustment energy is typically forty times larger compared to the other adjustment types.

NOMENCLATURE

φ	Radial position of the mass [rad]
a	Vertical distance between the pivot and the spring attachment point to the fixed world [m]
g	Gravity acceleration [m/s^2]
k	Stiffness [N/m]
L	Length of lever arm [m]
L_0	Spring null-length [m]
m	Mass [kg]
r	Distance between the pivot and the spring attachment point to the lever arm [m]
T	Torque or moment [Nm]
U	Potential energy [J]
u_0	Initial elongation [m]

I. INTRODUCTION

Static balancers are applied in robotic manipulators [1, 2, 3, 4], orthoses and exoskeletons [5, 6, 7, 8] to reduce the required operating power that is supplied by motors or muscles. The balancers support the system's constant payload: the mass of the robotic arm or human limbs. The operating power can be further reduced by balancing the variable payload as well. The variable payload is not constantly part of the system, e.g. the mass of an object that is picked up by the robotic arm or user of the exoskeleton. One or multiple properties of the balancer need to be adjusted to balance the varying payload. The adjustment should be possible in any position in the range of motion to enable picking up and releasing the object at any instance. Furthermore, the closer the balancing quality of the balancer is to perfect balance, the greater the reduction of the operating power.

The total potential energy of a balancer with perfect balancing quality is constant over its range of motion [9] and a transmission is required to transform the potential energy curve of the spring. These transmissions have been classified by Wagemaker [10] on basis of their transfer functions. The simplest method to obtain the desired transfer function is to select the right location of the spring attachment points. Existing balancers based on this method use the ideal spring principle [10]. This principle assumes a spring with a free-length equal to zero (*zero-free-length spring (ZFLS)*) [11]. The manufacturing of such springs is complex and therefore the properties of a ZFLS are often emulated using a string-pulley arrangement. At least three pulleys are required to obtain constant potential energy over the range of motion of the balancer [11], although this can be closely approximated by optimizing the position of a single pulley [12]. Besides perfect ZFLS principle balancers, imperfect balancers based on the normal spring (NS) principle exist. In this principle, the null-length of the spring is taken into account in the design of the balancer. This evades the necessity of a pulley arrangement to emulate a ZFLS. The NS balancers maintain approximately constant total potential energy over a limited part of its range of motion [11, 13]. In the current balancer designs, only the spring stiffness [5, 11, 12, 14, 15, 16] and null-length [11, 13] have been taken into account and important spring properties such as the maximum energy storage and the initial tension have not been considered.

Multiple methods to adjust a ZFLS balancer to accommodate a new payload have been introduced by Herder [11]. Additionally, adjustment methods that do not involve external energy (*energy-free adjustment*) have been developed [5, 14, 15, 16]. Energy-free adjustment is possible in one position only (see Appendix I) and is therefore not considered in this study. However, it is crucial to limit the energy involved in the adjustment, to enable slim and lightweight design and fast adjustment when the payload is altered. This study focuses on the performance of two classes of balancers. The first class incorporates a pulley at point A in Figure 1 to emulate a ZFLS (*pulley balancer*). The second class is based on the NS principle which evades the necessity of a pulley (*non-pulley balancer*). Balancers incorporating an actual ZFLS or three pulleys to emulate a ZFLS are not considered to limit the complexity of the balancer.

The goal of this study is twofold. The first goal is to introduce the optimal balancer adjustment types for application in exoskeletons and robotics. The second goal is to compare their performance based on the balancing quality, energy involved in adjustment and dimensions through an analytical MATLAB model. The analytical approach gives insight in the relations between performance metrics and spring properties. The introduced adjustment types incorporate a commercial available spring with realistic properties and their performance is considered in their complete range of motion. The paper starts with the definition of a reference balancer and the minimum energy involved in adjustment. Next, optimal adjustment types for application in exoskeletons and robotics are determined and their balancing conditions are derived. Subsequently, performance measures are defined and the spring selection method is explained. Then, the model of the selected adjustment types is described. This is followed by the results of the evaluation of the model for the selected adjustment types. In the discussion, general trends in the performance of the types are discussed. The paper concludes with the most important findings of this study. In the appendices, support material can be found.

II. METHOD

The basic balancer

The basic balancer (bb) is described in detail in literature and it will be used as reference to which the performance of the introduced adjustment types is compared. The basic balancer is based on the ideal spring principle [11]. One end of the spring is mounted to the lever arm at distance r from the pivot, the other end is fixed at distance a vertically above the pivot (Figure 1). The radial position of the mass (*balancer position*) is described by the angle φ between the vertical and the lever arm. The balancing condition which needs to be fulfilled to obtain perfect balance is described in equation 1, in which k , m , g and L represent the spring stiffness, balanced mass, gravity acceleration and lever arm length respectively [14]. The latter three parameters are assumed to be prescribed by the application and can therefore not be altered.

$$akr = mgL \quad (1)$$

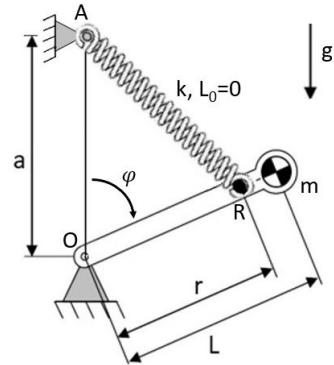


Fig. 1: The basic balancer that balances mass m and incorporates a zero-free-length spring ($L_0 = 0$) between point A and R. Adapted from [12].

The potential energy curve of the spring and mass (Figure 2) have two important characteristics: a slope and a magnitude (or vertical position). The mass-curve is completely defined by the mass, the length of the lever arm and the gravity acceleration. The slope of the spring-curve is determined by the transmission. In a perfectly balanced system, the slope of the spring curve is opposite to the slope of the mass curve to obtain constant total potential energy. The vertical position of the spring curve is determined by the elongation of the spring at $\varphi = 0$ (*initial elongation*). The initial elongation causes the spring-curve to start at a nonzero value and is determined by distances a and r . An initial elongation equal to zero corresponds to the condition $r = a$.

Adjustment energy

By altering the payload on the balancer, i.e. replacing mass m_A (situation A) by a larger mass m_B (situation B), the slope of the mass-curve is scaled. The slope of the spring-curve needs to be scaled as well to maintain constant total potential energy. Theoretically, this means that the transfer function of the transmission is multiplied by a constant factor. In practice, the balancer is adjusted by altering certain balancer properties. If the length of the spring is altered in the adjustment, energy is involved in the adjustment process. Since the spring length is altered, its initial elongation is modified. Consequently, besides scaling the slope of the spring-curve, the adjustment alters the vertical position of the spring curve as well (Figure 3). The energy involved in the adjustment (*adjustment energy*) is equal to the difference between the energy contained by the spring in situation A and B (equation 2, Figure 3).

$$U_{adj}^{AB}(\varphi) = U_{spring}^B(\varphi) - U_{spring}^A(\varphi) \quad (2)$$

$$U_{adj}^{BA}(\varphi) = -U_{adj}^{AB}(\varphi) \quad (3)$$

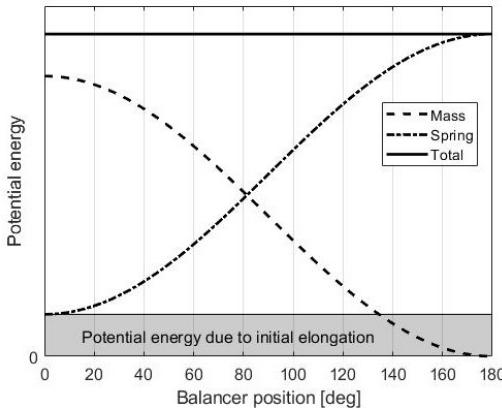


Fig. 2: Potential energy curves of a perfectly balanced system

The adjustment energy is a function of the balancer position and the energy involved in adjusting from situation A to B is equal but opposite to the energy involved in adjusting from situation B to A (equation 3). A positive value of adjustment energy implies that external energy is required in the adjustment, a negative value implies that energy is released by the spring.

Minimum adjustment energy

The minimum adjustment energy of the basic balancer is derived to determine to what extend the proposed balancer adjustment types can approach this optimum. The minimum adjustment energy corresponds to the minimal difference between the two spring curves over the complete range of motion (Figure 3). The amount of energy stored in the spring of the basic balancer can be calculated using equation 4 [14]. In case distance a is altered in the adjustment process, by finding an expression for a from equation 1 and substituting it into equation 4, a modified expression for the energy stored in the spring (equation 5) is obtained. The adjustment energy of the basic balancer (equation 6) can be obtained by deriving the expressions of the energy stored in the spring in situation A and B using equation 5, and substituting these expressions into equation 2.

$$U_{spring}^{bb} = \frac{1}{2}k(a^2 + r^2 - 2ra \cos \varphi) \quad (4)$$

$$U_{spring}^{bb} = \frac{1}{2}k\left(\frac{m^2 g^2 L^2}{k^2 r^2} + r^2 - 2\frac{mgL}{k} \cos \varphi\right) \quad (5)$$

$$U_{adj}^{AB}(\varphi) = \frac{1}{2}\frac{g^2 L^2}{kr^2}(m_B^2 - m_A^2) - gL \cos \varphi(m_B - m_A) \quad (6)$$

Considering equation 6, a constant part and a variable part with respect to the balancer position can be distinguished. The variable part is equal to the difference between the potential energy of the two masses, is completely

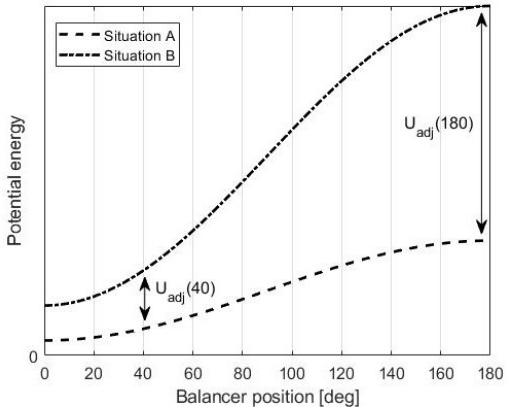


Fig. 3: Graphical representation of the adjustment energy

determined by the prescribed variables and can therefore not be minimized. The constant part can be minimized by selecting a relatively large value of k or r . Assuming that the constant part can be reduced until it approaches zero, the theoretical minimum adjustment energy is equal to the difference between the potential energy of mass m_B and m_A (equation 7, Figure 4). As shown in Figure 4, the adjustment energy can be equal to zero at a certain balancer position. This corresponds to the adjustment in which the spring is consecutively lengthened and shortened by the same amount (or vice versa). Consequently, the net energy involved in the adjustment is equal to zero.

$$U_{adj,min}^{AB}(\varphi) = -gL \cos \varphi(m_B - m_A) \quad (7)$$

Alternatively, in case distance r is adjusted, an expression for r can be derived from equation 1, and can be substituted into equation 4. The adjustment energy is then equal to equation 8. As seen in this equation, the constant part can be minimized by selecting a relatively large value of k or a and the minimum adjustment energy is identical to equation 7. This illustrates the fact that the basic balancer is symmetric with respect to parameters r and a and that these parameters can be interchanged without altering the behavior of the balancer.

$$U_{adj}^{AB}(\varphi) = \frac{1}{2}\frac{g^2 L^2}{ka^2}(m_B^2 - m_A^2) - gL \cos \varphi(m_B - m_A) \quad (8)$$

Adjustment type selection

The pulley and non-pulley balancer can be adjusted by altering distance a (*a-type*), r (*r-type*) or spring stiffness k (*k-type*) [11, 12]. Additionally, the pulley balancer can be adjusted by introducing an additional spring elongation u_0 , without altering the direction of the spring force exerted on the lever arm [12]. In fact, the free-length of the spring is modified and therefore this adjustment type is referred to

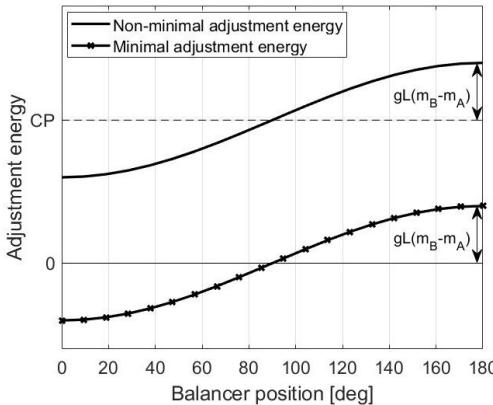


Fig. 4: Minimal and non-minimal adjustment energy of a balancer. The lines are a cosine-function with amplitude $gL(m_B - m_A)$. The baseline of the minimum adjustment energy is equal to zero, the baseline of the non-minimum adjustment energy is equal to the constant part of the adjustment energy (CP). The point at which the minimum adjustment energy is equal to zero corresponds to the balancer position in which the spring length before adjustment is equal to the spring length after adjustment.

as free-length adjustment (*FL-type*). In all the adjustment methods described above, the position of the lever arm is locked before adjustment is performed.

In exoskeletons and robotics, it is important to limit the mass and inertia of the mechanism, especially of moving parts. Locating the adjustment mechanism on the lever arm increases the mass of lever arm, which excludes r-type adjustment. Furthermore, the adjustment should be possible in every balancer position, which excludes k-type adjustment [14]. Consequently, the optimal adjustment types for application in exoskeletons and robotics include:

- 1) Pulley balancer a-type adjustment (*AP-type*, Figure 5a)
- 2) Pulley balancer FL-type adjustment (*FL-type*, Figure 5c)
- 3) Non-pulley balancer a-type adjustment (*ANP-type*, Figure 5c)

In each adjustment type, one variable is altered (*adjusted variable*) to adjust the balancer while all other variables remain constant (*non-adjusted variables*). A design of each adjustment type is defined by a set of three variables (*design variables*). The adjusted, non-adjusted and design variables per adjustment type are shown in Table I.

Balancing conditions of adjustment types

A balancing condition enables simple calculation of the balancer parameters that correspond to the optimal balancing quality. Since the total potential energy of the AP-type, ANP-type and FL-type cannot be constant over the range of motion, perfect balance over the complete range of motion is not possible. Consequently, the balancing conditions of the adjustment types correspond to *optimal* balancing quality, rather than *perfect* balancing quality as is the case in the basic balancer. To obtain optimal balancing quality, two conditions

TABLE I: Adjusted variables, non-adjusted variables and variables that define a balancer design per adjustment type

Adjustment type	Adjusted variable	Non-adjusted variables	Design variables
AP-type	a	r	a_A, a_B, r
ANP-type	a	r	a_A, a_B, r
FL-type	u_0	a, r	a, r, u_0

are found in which the dependency of the total potential energy on the balancer position is minimal: $r \gg a$ and $r \ll a$. Two balancing conditions per adjustment type are derived: one corresponding to each condition. The balancing conditions of the AP-type, ANP-type and FL-type are presented in equations 9, 10 and 11 respectively (complete derivation in Appendix III). The balancing conditions of the FL-type correspond to situation B, since in this situation an additional spring elongation is introduced. The balancing conditions of the FL-type in situation A are identical to the balancing conditions of the AP-type.

$$r \ll a \implies a = \frac{mgL}{kr} - \frac{F_0}{k} \quad (9a)$$

$$r \gg a \implies a = \frac{mgL}{kr + F_0} \quad (9b)$$

$$r \ll a \implies a = \frac{mgL}{kr} - \frac{F_0}{k} + L_0 \quad (10a)$$

$$r \gg a \implies a = \frac{mgL}{kr - kL_0 + F_0} \quad (10b)$$

$$r \ll a \implies u_0 = \frac{m_B g L}{kr} - \frac{F_0}{k} - a \quad (11a)$$

$$r \gg a \implies u_0 = \frac{m_B g L}{ka} - \frac{F_0}{k} - r \quad (11b)$$

Performance metrics

The performance of the proposed adjustment types is compared on (1) balancing quality, (2) adjustment energy and (3) balancer dimensions and is described by three metrics. In these metrics, the performance of the adjustment type is compared to the (optimal) performance of the basic balancer.

Balancing quality

The balancing quality (BQ) is an important property because it determines to the reduction of the operating power. The BQ is defined as the percentage by which the resultant moment (T_{res}) about the pivot is reduced with respect to an unbalanced system (equation 12). The balancing quality is a function of the balancer position. To convert the balancing quality to a single value, the root mean square error (RMSE) of the difference between the actual and perfect balancing quality (*balancing error* (BQe), equation 13) is calculated. A balancing error equal to zero implies perfect balance. Since the balancing quality in both situation A and B is important, the balancing error in situation A (BQe_A) and B (BQe_B) is considered.

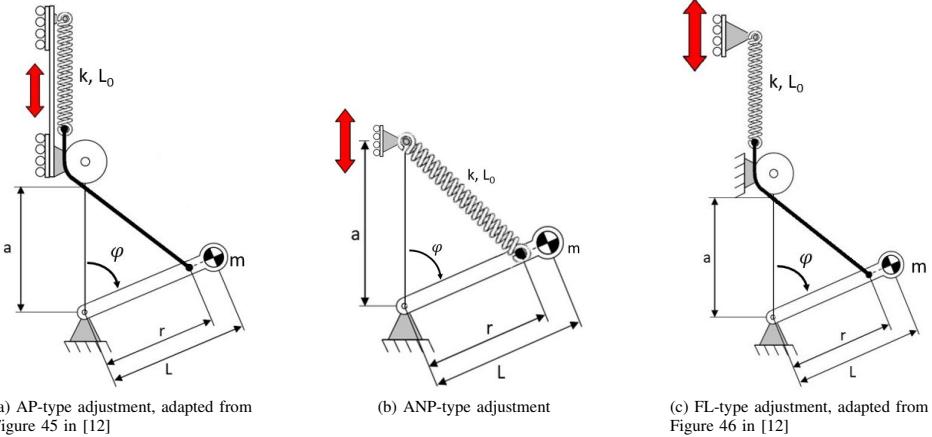


Fig. 5: Selected methods to adjust the balancer

$$BQ(\varphi) = \frac{T_{mass}(\varphi) - T_{res}(\varphi)}{T_{mass}(\varphi)} * 100\% \quad (12)$$

$$BQe = RMSE(BQ(\varphi) - 100\%) \quad (13)$$

Adjustment energy

The lower the adjustment energy, the smaller and more lightweight the required actuators and energy storage can be. The adjustment energy metric (AE_m) is defined as the relative difference between the maximum value of the actual adjustment energy and the minimum adjustment energy (equation 14). The maximum value over the range of motion is considered since this is the maximum amount of energy that could possibly be involved in the adjustment. An AE_m equal to 50% indicates that the maximum value of the actual adjustment energy is 50% larger than the maximum value of the minimum adjustment energy.

$$AE_m = \frac{\max(U_{adj}^{AB}(\varphi)) - (m_B - m_A)gL}{(m_B - m_A)gL} * 100\% \quad (14)$$

Balancer dimensions

The dimensions of the balancer are important since a compact system is favored. The dimension metric (DM_m) is defined by the area of the triangle OAR (S_{OAR}) at $\varphi = 90\text{deg}$ in situation B. Situation B is considered because the balancer dimensions are the largest in this situation, and therefore it determines the amount of available space required to implement the balancer. The dimension metric is equal to relative difference between the area of triangle OAR of the considered adjustment type and the basic balancer (equation 15). The area of triangle OAR of the basic balancer can be described by equation 16. A DM_m equal to 50% indicates that the area of the triangle in the actual balancer is 50% larger than the area of the basic balancer.

$$DM_m = \frac{S_{OAR} - S_{OAR}^{basic}}{S_{OAR}^{basic}} * 100\% \quad (15)$$

$$S_{OAR}^{basic} = \frac{ra}{2} = \frac{r}{2} \frac{m_B g L}{k r} = \frac{m_B g L}{2k} \quad (16)$$

Spring selection

A database consisting of 4280 commercial available springs [17] is used to select a suitable spring. The minimum amount of energy that should be stored in the spring to balance the mass is equal to the difference between the potential energy of the mass in its lowest and highest position. To increase the life time of the spring, the minimum amount of energy that should be stored in the spring is multiplied by a safety factor equal to two. The springs that cannot store this amount of energy are discarded. In practice, this filters out the most lightweight springs, since the energy that can be stored in the spring can be estimated by its energy density times the spring mass. From the set of remaining springs, a spring with desired properties can be selected.

Modelling

The balancing conditions do not take the maximum energy storage of the spring into account, which implies that the optimal balancing quality can correspond to plastic deformation of the spring. Therefore, to compare the performance of the adjustment types, an analytical MATLAB model which includes the maximum energy storage is developed. The variables that correspond to optimal balancing quality that are obtained using the model will be graphically compared to the parameters calculated by the balancing conditions. The input of the model includes the amplitude of the mass moment that is balanced in situation A and B and the spring choice from the database. Since the moment of the mass has a sinusoidal profile, it can be completely defined by its amplitude.

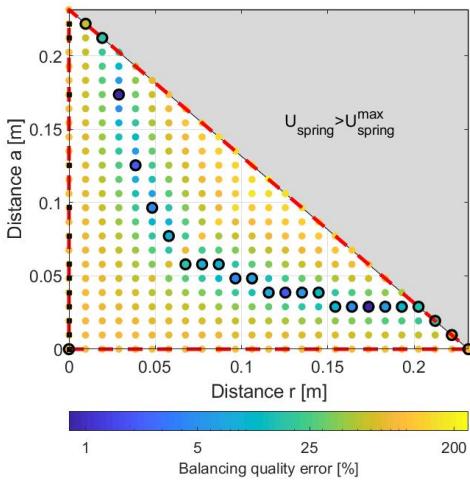


Fig. 6: Example of the balancing quality error in situation A (BQe_A) of the AP-type as function of the variable grid (step 1 to 5 of the model). The grid size is reduced to 25×25 for purposes of clarity and the range of distances a (adjusted variable) and r (non-adjusted variable) is selected as example. Filled circle: grid point, red dashed line: variable grid boundary, black contoured circles: grid points corresponding to minimum BQe per non-adjusted variable, filled black squares on vertical axis: distances a_A corresponding to the minimum adjustment energy per r .

The model consists of seven steps and will be explained based upon an example of the AP-type, which is illustrated in Figure 6.

Step 1. The variable boundaries are determined. The lower boundary of the variable space is equal to zero. The upper boundary is derived by expressing the maximum allowable spring elongation (dL_{spring}^{max}) in terms of the maximum energy storage of the spring (U_{spring}^{max}) and the spring stiffness. The maximum spring elongation is determined by a , r , u_0 and L_0 at $\varphi = 180\text{deg}$ and should be smaller than the maximum allowable spring elongation. The maximum spring elongation and variable boundaries are listed in Table II.

Step 2. The triangular variable grid (base and height consist of 200 grid points) is constructed. Step 3 to 5 are performed for situation A and B.

Step 3. The BQe is evaluated in each grid point.

Step 4. The minimum balancing error per non-adjusted variable is selected. For the AP-type, this means that the minimum balancing error per distance r is selected.

Step 5. The adjusted variable that corresponds to the minimum balancing error per non-adjusted variable is determined. In case of the AP-type, this means that the optimal distances a_A and a_B are determined per distance r . These three variables define a balancer design.

Step 6. To select the optimal balancer designs, the twenty designs with the smallest cumulative balancing error ($BQe_A + BQe_B$) are selected. In case of the AP-type, this means

TABLE II: Maximum spring elongation and variable boundaries for each adjustment type

Adjustment type	Maximum spring deflection	Variable boundaries
AP-type	$a_B + r$	$0 < a_B + r < \sqrt{2U_{spring}^{max}/k}$
ANP-type	$a_B + r - L_0$	$0 < a_B + r < \sqrt{2U_{spring}^{max}/k} + L_0$
FL-type	$a + r + u_0$	$0 < a_B + r + u_0 < \sqrt{2U_{spring}^{max}/k}$

TABLE III: Maximum moment of mass, spring selection criterion and relative difference between mass moments in each comparison case

Case	Situation A	Situation B	Spring	Relative difference
1	2 Nm	4 Nm	Lightest	100%
2	4 Nm	24 Nm	Lightest	500%
3	80 Nm	100 Nm	Lightest	25%
4	80 Nm	140 Nm	Lightest	43%

that twenty sets of variables r , a_A and a_B are selected. The designs with the smallest cumulative balancer error are selected since the balancing quality is considered as the most important balancer characteristic. The main reason for this is that the balancing quality determines to what extend the operating power is reduced, which is the primary goal of the balancer's implementation. An iterative approach showed that by selecting considerably less than twenty designs, limited insight is obtained in the trade offs between the performance metrics. Besides, by selecting considerably more than twenty designs, balancers with a poorer balancing quality are included. This counteracts the balancer's primary goal to optimally reduce the operation power.

Step 7. The AE_m and DM_m are evaluated for the selected designs.

Comparison cases

The performance of the three adjustment types is compared in four use cases (Table III). The four cases are derived from the application in robotics and exoskeletons. The specification of the selected values and reasons why these cases are selected are described in Appendix II.

III. RESULTS

The scores of the three balancers on the four metrics are shown in Figures 7 to 10 and explained in the following paragraphs. The properties of the spring selected in each comparison case are listed in Table IV.

Balancing quality error

The adjustment types cannot obtain perfect balance in their complete range of motion, because, in contrast to the basic balancer, the null-length, initial tension and maximum energy storage of the spring are taken into account. In Figures 11 to 18, the balancing quality of the basic balancer, AP-type, ANP-type and FL-type is shown as function of the variable grid. In Figure 11 to 17, the balancing conditions are

TABLE IV: Properties of the selected spring in each comparison case

Case	Serial number	Mass [kg]	Stiffness [N/m]	Null-length [m]	Initial tension [N]
1	Z-130K-01I	0.0760	275	0.145	6.88
2	Z-18II	0.603	876	0.249	24.5
3	Z-305I	3.12	1010	0.486	45.0
4	Z-336I	4.972	905	0.600	70.0

represented by the dashed lines. As shown in these Figures, the variable grid is bounded to prevent plastic deformation of the spring. The variable grid is bounded by zero and a diagonal line between the maximum value of a and the maximum value of r . The lines that represent the balancing conditions cross this diagonal boundary because they do not take the maximum energy storage in the spring into account. The balancing condition of the FL-type in situation B (Figure 18) is out of the scope of the graph, because the optimal initial elongation calculated by the balancing conditions corresponds to plastic deformation of the spring.

As seen in Figures 14, 16 and 18, generally, in situation B, a small balancing error is possible for a certain set of points in the grid. However, since only one parameter is adjusted in each adjustment type (either a , r , or u_0), this set of grid points is often not within reach. For example, in situation A of the AP-type (Figure 13), a small balancing error can be obtained for $(r; a) = (0.2; 0.025)m$. Since a-type adjustment is used, distance a is altered while distance r remains constant. Since $r = 0.2m$, the smallest balancing error in situation B (Figure 14) cannot be reached and a larger balancing error is present. This illustrates the trade-off between the balancing quality situation A and B: a good balancing quality in one situation can result in a poor balancing quality in the other.

Regarding the AP-type and FL-type, the initial tension of the spring causes a peak in the balancing error around the line $a = r$ (Figure 13, 14, 17 and 18). Regarding the ANP-type, the initial tension and null-length of the spring cause a peak in the balancing error at the same location (Figure 15 and 16). Due to this peak, the best balancing quality corresponds to $r >> a$ and $r << a$. This is in accordance with the two conditions that are found in the derivation of the balancing conditions. The BQE of the FL-type is the smallest in case 3, followed by case 4, 1 and 2. This sequence corresponds with ascending relative difference between the moments of the two masses that are balanced (last column of Table III).

Adjustment energy

The adjustment energy of the FL-type is 400% to 500% larger than the adjustment energy of the AP-type and ANP-type. This is because in the FL-type, only the magnitude of the spring force is adjusted. This proves to be less efficient than adjusting the magnitude and direction of the spring force as is the case in the AP-type and ANP-type.

In case 1, 2 and 4, the adjustment energy of the AP-type and ANP-type are clustered around 10%. In case 3, the

adjustment energy of the AP-type and ANP-type seems to be spread over a wide range. However, actually, the adjustment energy consists of two groups: most of the results (18) are clustered around 10% and a small number of them (2) are clustered around 500%. The first group corresponds to $r >> a$, since the adjustment energy can be minimized by selecting a relatively large distance r . The 500%-group corresponds to a relatively small r ($r << a$).

Dimensions

The dimension metric of the AP-type is typically 10% smaller than the basic balancer, because, in contrast to the basic balancer, the initial tension of the spring is taken into account. The energy storage per unit elongation of a spring with initial tension is larger with respect to a spring without initial tension. Since a smaller spring elongation is required to store the energy, the dimensions of the AP-type are smaller than the dimensions of the basic balancer. This reasoning is supported by the negative sign of the initial tension-term in equation 9a and the initial tension-term in the denominator in equation 9b. The dimensions of the ANP-type are approximately 50% larger than the dimensions of the basic balancer. The reason for this is that, in contrast to the basic balancer, the null-length of the spring cannot be hidden behind a pulley. This can be seen by the positive sign of the null-length in equation 10a and the negative sign of the null-length in the denominator in equation 10b. The dimensions of the FL-type are the smallest, but the variation between the cases is the largest. This is because adjusting the free-length of the spring does not alter the dimensions of the triangle OAR. Therefore, the area is determined by the dimensions corresponding to situation A. These dimensions are smaller than in situation B.

IV. DISCUSSION

The results of this study enable the selection of the optimal adjustment type depending on the application's requirements. In case perfect balance is strictly required, the considered adjustment types cannot be used, since none of them can supply perfect balance over the complete range of motion. In case a small balancing error (<5%) is allowed, the AP-type and ANP-type can be considered. The balancing error of the FL-type is comparable to the AP-type and ANP-type if the relative difference between the moments of the two masses is small (<20%). In case limiting the adjustment energy is important, the AP-type and ANP-type can be considered, since their adjustment energy is typically only 10% larger than the minimum adjustment energy. A practical difference in the adjustment of those types is that in the adjustment of the AP-type, the complete spring and pulley need to be displaced, whereas in the ANP-type only the location of the spring attachment point needs to be altered. The adjustment energy of the FL-type is typically four to six times larger than the minimum adjustment energy and is therefore suitable in case the amount of adjustment energy does not have to be minimized.

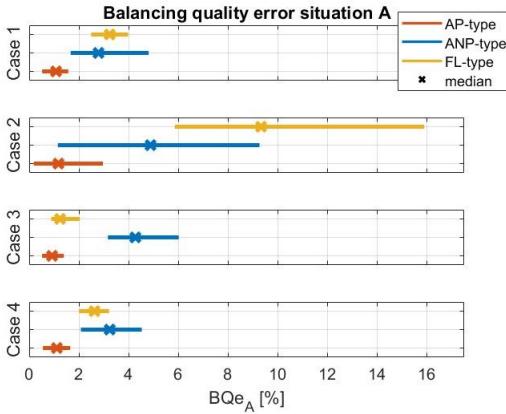


Fig. 7: Results of the BQe_A metric

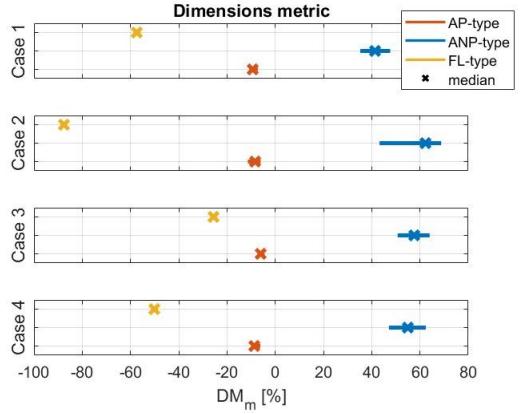


Fig. 10: Results of the DM_m metric

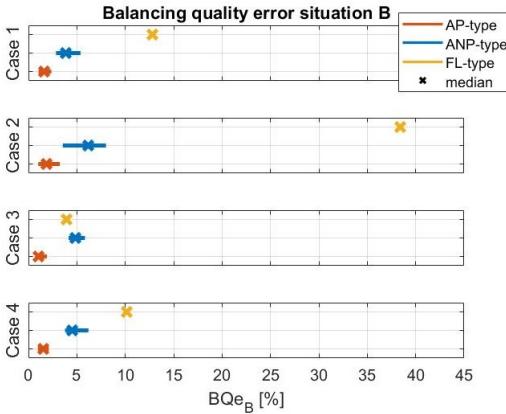


Fig. 8: Results of the BQe_B metric

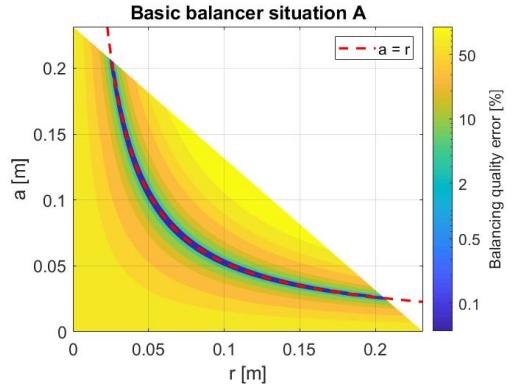


Fig. 11: Balancing error of the basic balancer as function of the variable grid. The dashed line represents the balancing condition. $T_{mass}=6\text{Nm}$; $k=1120\text{N/m}$; $L_0=0.1\text{m}$, $F_0=10\text{N}$, $U_{spring}^{max}=30\text{J}$.

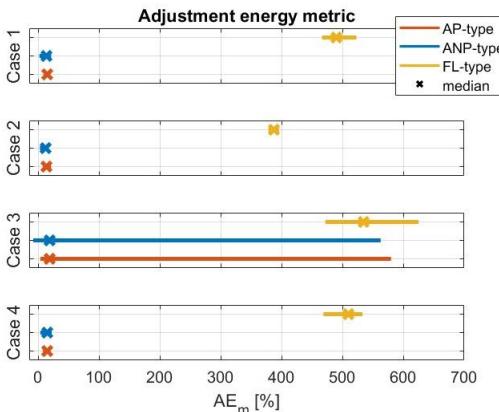


Fig. 9: Results of the AE_m metric

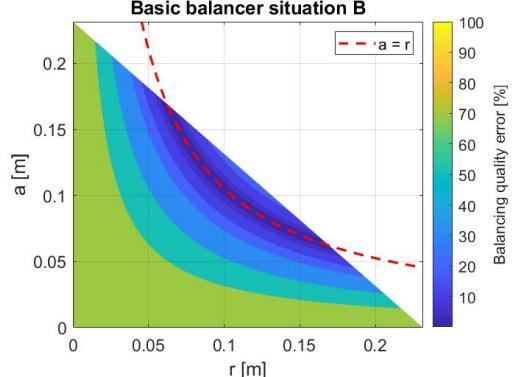


Fig. 12: Balancing error of the basic balancer as function of the variable grid. The dashed line represents the balancing condition. $T_{mass}=12\text{Nm}$; $k=1120\text{N/m}$; $L_0=0.1\text{m}$, $F_0=10\text{N}$, $U_{spring}^{max}=30\text{J}$.

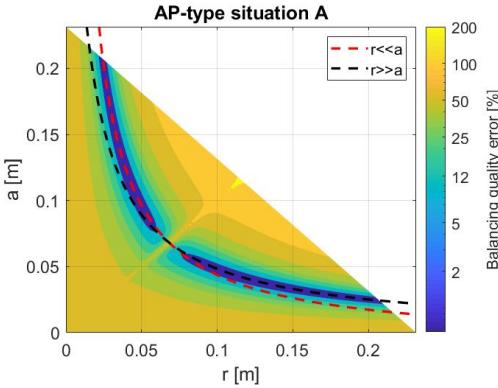


Fig. 13: Balancing error of the AP-type as function of the variable grid. The dashed lines represent the balancing conditions. $T_{mass}=6\text{Nm}$; $k=1120\text{N/m}$; $L_0=0.1\text{m}$, $F_0=10\text{N}$, $U_{spring}^{max}=30\text{J}$.

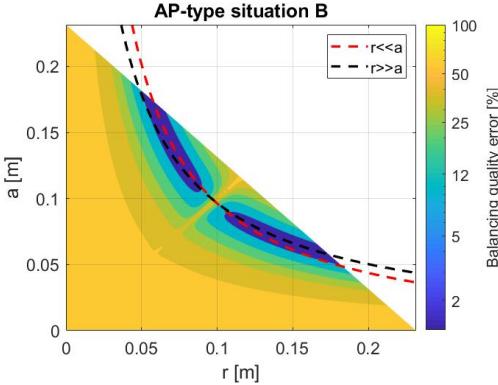


Fig. 14: Balancing error of the AP-type as function of the variable grid. The dashed lines represent the balancing conditions. $T_{mass}=12\text{Nm}$; $k=1120\text{N/m}$; $L_0=0.1\text{m}$, $F_0=10\text{N}$, $U_{spring}^{max}=30\text{J}$.

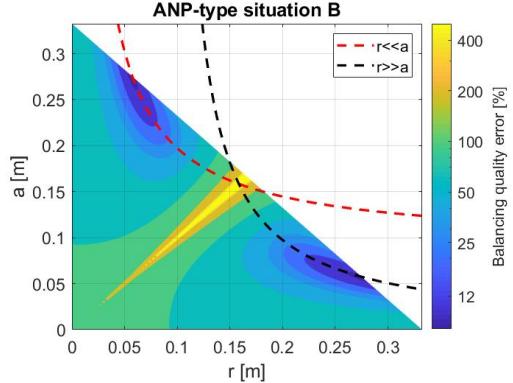


Fig. 16: Balancing error of the ANP-type as function of the variable grid. The dashed lines represent the balancing conditions. $T_{mass}=12\text{Nm}$; $k=1120\text{N/m}$; $L_0=0.1\text{m}$, $F_0=10\text{N}$, $U_{spring}^{max}=30\text{J}$.

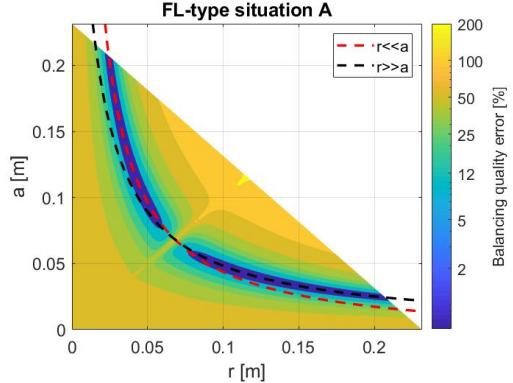


Fig. 17: Balancing error of the FL-type as function of the variable grid. The dashed lines represent the balancing conditions. $T_{mass}=6\text{Nm}$; $k=1120\text{N/m}$; $L_0=0.1\text{m}$, $F_0=10\text{N}$, $U_{spring}^{max}=30\text{J}$.

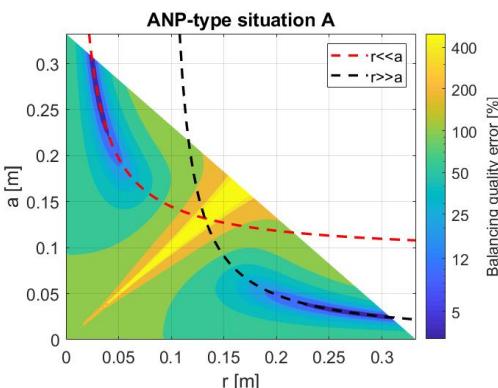


Fig. 15: Balancing error of the ANP-type as function of the variable grid. The dashed lines represent the balancing conditions. $T_{mass}=6\text{Nm}$; $k=1120\text{N/m}$; $L_0=0.1\text{m}$, $F_0=10\text{N}$, $U_{spring}^{max}=30\text{J}$.

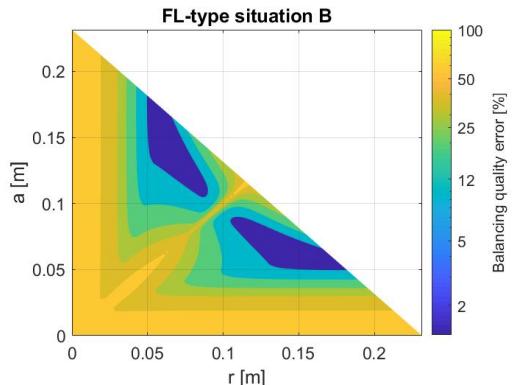


Fig. 18: Balancing error of the FL-type as function of the variable grid. The balancing conditions are located out of the scope of the graph since they correspond to plastic deformation of the spring. $T_{mass}=12\text{Nm}$; $k=1120\text{N/m}$; $L_0=0.1\text{m}$, $F_0=10\text{N}$, $U_{spring}^{max}=30\text{J}$.

In case a compact design is preferred, the FL-type, since its dimensions are determined by the mass balanced in situation A, and the AP-type, since it is typically 10% smaller than the basic balancer, can be considered. The dimensions of the ANP-type are determined by the null-length of the spring and are typically 50% larger than the basic balancer. Therefore, the ANP-type is suitable in case a compact design is not favored or a spring with a small null-length can be used.

The minimum adjustment energy as presented in equations 6 and 8 is a theoretical optimum. To obtain this minimum adjustment energy, the product kr^2 or ka^2 should be equal to infinite, which is practically not possible. Therefore, the typical adjustment energy of the AP-type and ANP-type (10% larger than the minimum adjustment energy) can be considered as close to the practical minimum adjustment energy. The adjustment energy is defined as the net energy involved in the adjustment. Regarding the AP-type and ANP-type, if point A (Figure 1) intersects the horizontal line through point R during the adjustment, the spring is consecutively lengthened and shortened (or vice versa). This is primarily the case for a large distance r and when the adjustment takes place at the beginning of the range of motion. In those cases, energy is consecutively required and released (or vice versa) and to calculate the adjustment energy, these energies are summed. This implies that the released energy is stored and supplied to adjust the balancer at perfect efficiency, which is practically not possible. In further research, this practical effect could be taken into account by multiplying the released energy by an efficiency factor or by assuming complete dissipation. This would lead to another situation in which the adjustment energy is minimum. In case of the FL-type, since only the magnitude of the spring force is altered during the adjustment process, the spring cannot be consecutively lengthened and shortened in one adjustment process and the above described effect do not occur.

The larger the difference between distance r and a , the smaller the effects of the peak in the balancing error. In other words, a maximum initial elongation of the spring is required to obtain the best balancing quality. This means that the energy storage capacity of the spring is completely utilized. This can deteriorate the life time of the spring and justifies the used safety factor in the maximum energy storage of the spring.

The model is based on a grid search to find the value of the variables (r , a and u_0) that correspond to the optimal balancing quality. Since the grid search uses a finite step size, the exact optimal is typically not found. However, the model is based on continuous analytical equations and the solution space is continuous as well, the found optimum is close to the actual optimum. Regarding the ANP-type, an alternative optimization method is considered, which was introduced by Herder [11]. Although this optimization method enables an intuitive and graphical solution, the solutions found by this optimization method are less optimal than the solutions of the grid search. For this reason and to equalize the methods

used in all adjustment types, the grid search is used in the model.

The use of one pulley in the AP-type and FL-type introduces a balancing quality error due to the variable amount of string that is wrapped around the pulley. This error is not taken into account in this study. The wrapping error can be minimized by minimizing the radius of the pulley or by optimization the location of the pulley [12]. Friction in the pivot and pulley is not considered, just as the hysteresis in the spring. The effects of the friction can be minimized by using bearings and the typical hysteresis in spiral springs is respectively small.

The balancing conditions of the basic balancer, AP-type, ANP-type and FL-type do not take the maximum energy storage of the spring into account. Therefore those conditions should be used with caution. Especially regarding the FL-type, the optimal additional elongation often corresponds to a large amount of energy that should be stored in the spring, since this type of adjustment proved to be inefficient in terms of energy. An advantage of the balancing conditions, however, is that the influence of a certain variable on the performance of the balancer can be simply determined. Further research is required to quantitatively validate the balancing conditions, to derive an expression to calculate the balancing error corresponding to the optimal balancing quality and to take the variable boundaries into account in the balancing conditions.

The fact that the best balancing quality corresponds to $r >> a$ or $r << a$ implies that one dimension of the balancer is typically an order of magnitude larger than the others. This effect is not captured in the dimension metric, since the area of the triangle is not affected. In future work, to design more compact solutions, the performance of balancers incorporating an additional transmission can be considered.

The performance of the adjustment types may be improved by optimizing the spring selection. However, selecting the optimal spring is complex, since the spring properties cannot be varied independently of each other. In the comparison cases, the most lightweight spring is selected from the set of suitable springs to minimize the mass of the system. An exploratory evaluation of the performance metrics for springs from the database with a slightly larger mass yielded to results comparable to the results presented in this paper. However, further research is required to investigate the effect of the implementation of springs with very different properties on the performance of the adjustment types to eventually enable optimal spring selection.

V. CONCLUSION

The goal of this study was twofold. The first goal was to introduce the optimal balancer adjustment types, that incorporate realistic springs, for application in exoskeletons and robotics. This study showed that the optimal adjustment types include the adjustment of a balancer (1) with pulley, adjusted by altering the location of the spring attachment point (AP-type), (2) without pulley, adjusted by altering the location of the spring attachment point (ANP-type) and (3)

with pulley, adjusted by altering the free length of the spring (FL-type).

The second goal was to compare their performance based on the balancing quality; energy involved in adjustment; and overall dimensions, through an analytical MATLAB model. The results show clear differences between the performance of the considered adjustment types. Due to the initial tension and null-length of the spring, perfect balance in the complete range of motion cannot be obtained by the adjustment types. In case the balancing quality is critical, AP-type adjustment is the best choice (error within 3%), closely followed by ANP-type (error within 5%). An advantage of the ANP-type over the AP-type is that no pulley is required, which simplifies the mechanism and increases its robustness. However, the dimensions of the ANP-type are larger than the dimensions of the AP-type. The theoretically minimum amount of energy involved in adjustment (1) is equal to the difference between the potential energy of the two balanced masses, (2) is dependent on the position of the mass and (3) can be practically obtained by introducing a maximum initial spring elongation. The adjustment energy of both the AP-type and ANP-type is approximately 10% larger than the minimum adjustment energy. The FL-type should only be considered when the relative difference between the two masses is smaller than 20%, since the balancing quality deteriorates for larger differences. Furthermore, its adjustment energy is generally four to six times larger than the theoretical minimum adjustment energy. Regarding the implementation of the adjustment types in exoskeletons and robotic manipulators, the FL-type is suitable in case the difference between the mass moments is small, for example when the exoskeleton is adjusted to another user. The AP-type and ANP-type require a considerable amount of installation space, since for optimal balancing quality, either r or a needs to be large with respect to the other parameter. The AP-type and ANP-type can be suitable for implementation in exoskeletons and robotic manipulators if sufficient installation space is available. However, especially in exoskeletons, the available space is often limited and more compact solutions, for example cam-balancers, are preferred.

REFERENCES

- [1] S.K. Agrawal and A. Fattah. "Gravity-balancing of spatial robotic manipulators". In: *Mechanism and Machine Theory* 39 (2004), pp. 1331–1344.
- [2] R. Nathan, W. Harwin, and R. Seliktar. "A simple technique to passively gravity-balance articulated mechanisms". In: *Journal of Mechanical Design* (1995).
- [3] P.Y. Lin, W.B. Shieh, and D.Z. Chen. "Design of a gravity-balanced general spatial serial-type manipulator". In: *Journal of Mechanisms and Robotics* 2 (2010).
- [4] V.A. Arakelian. "Gravity compensation in robotics". In: *Advanced Robotics* 30.2 (2015), pp. 79–96.
- [5] W.D Dorsser van et al. "Energy-free adjustment of gravity equilibrators, with application in a mobile arm support". In: *Proceedings of DETC 2006* (2006).
- [6] S.K. Agrawal, S.K. Banala, and A. Fattah. "A gravity balancing passive exoskeleton for the human leg". In: *University of Delaware* (2006).
- [7] S.K. Agrawal et al. "Assessment of motion of a swing leg and gait rehabilitation with a gravity balancing exoskeleton". In: *IEEE Transactions on neural systems and rehabilitation engineering* 15.3 (2007), pp. 410–420.
- [8] Q. Wu, X. Wang, and F. Du. "Development and analysis of a gravity-balanced exoskeleton for active rehabilitation training of upper limb". In: *Journal of Mechanical Engineering Science* 230.20 (2016), pp. 3777–3790.
- [9] M. Schenk and S.D. Guest. "On zero stiffness". In: *Journal of Mechanical Engineering Science* 228 (2014), pp. 1701–1714.
- [10] S.J. Wagemaker. "Bottom-up classification method of force generators applied to perfect static balancers with one non-auxiliary revolute joint". Literature review, Delft University of Technology.
- [11] J.L. Herder. "Energy-free systems". PhD Thesis, Delft University of Technology.
- [12] R. Barents. "Literature review". Delft University of Technology.
- [13] F.L.S. Riele te and J.L. Herder. "Perfect static balance with normal springs". In: *Proceedings of DETC01* (2001).
- [14] W.D Dorsser van et al. "Energy-free adjustment of gravity equilibrators by adjusting the spring stiffness". In: *Journal of Mechanical Engineering Science* 222 (2008), pp. 1839–1846.
- [15] B.M Wisse et al. "Energy-free adjustment of gravity equilibrators using the virtual spring concept". In: *Proceedings of the 2007 IEEE 10th International Conference on Rehabilitation Robotics* (2007), pp. 742–750.
- [16] R. Barents et al. "Spring-to-spring Blancing as Energy-free Adjustment Method in Gravity Equilibrators". In: *Journal of Mechanism Design* 133 (2001). *Gutekunst Federn*. https://www.federnshop.com/en/products/extension_springs.html, 2019.
- [17] Ottobock Paexo Exoskeleton. <https://www.ottobock.com/en/company/ottobock-industrials/paexo/>, 2019.
- [18] University of Rhode Island. *Introduction to Biomechanics, Anthropometric Data*. <https://www.ele.uri.edu/faculty/vetter/BME207/anthropometric-data.pdf>.
- [19] Laevo Exoskeletons. <http://www.laevo.nl/>, 2019.

APPENDIX I ENERGY-FREE ADJUSTMENT

Two types of adjustment principles in which no energy is involved can be distinguished. In the first type, the spring length is kept constant and therefore no energy is involved in the adjustment. This type includes simultaneous displacement [5], altering the spring stiffness by altering the number of active spring coils [14] and the use of a virtual spring [15]. The second type includes spring-to-spring balancing [16]. In this type, the spring length is altered which involves energy. However, this energy is stored or supplied by a second spring and consequently no external energy is involved.

Energy-free adjustment without loss of energy in the system is possible in one balancer position only. This can be illustrated by a thought experiment¹.

Suppose a balancer is used as an elevator, in which a mass can be lifted from a zero to a larger height and that both heights correspond to positions at which the balancer can be adjusted in an energy-free manner. Starting at zero height, the balancer is adjusted from supporting no payload to supporting the mass, without involving energy since this is an energy-free adjustment position. Secondly, the mass is lifted to the higher level which does neither involve energy (neglecting the acceleration and deceleration) since the mass is perfectly balanced. Thirdly, the mass is removed from the balancer and simultaneously the balancer is adjusted to a zero payload in an energy-free manner since, again, this is an energy-free adjustment position. Lastly, the balancer is brought to the starting position without involving energy since the system is perfectly balanced and the procedure can be performed again.

Considering this thought experiment, an infinite number of masses can be lifted without requiring energy. However, since the height of the masses is increased, the potential energy of the masses is increased. In other words, potential energy is created at the cost of zero energy. This violates the first law of thermodynamics, which states that energy cannot be created nor destroyed. Consequently, energy-free adjustment between two masses can be performed in maximum one balancer position. To perform the adjustment in various balancer positions, energy is involved. The thought experiment supports the fact that the minimum adjustment energy is equal to the potential energy difference of the masses, since then no energy is created. Another perspective is proposed by [5], who suggests that as long as on average the mass that is lifted is equal to the mass that is lowered, the system can operate continuously.

¹As discussed with ir. R. Barents

APPENDIX II COMPARISON CASES

The comparison cases are derived from applications of exoskeletons and illustrate the extremes of each application. During the study, cases in between the extremes are considered as well. The results of these cases are in line with the results of the extremes and therefore not elaborated upon.

Case 1 and 2 are based on an upper extremity exoskeleton (comparable to [18]). It is assumed that the lever arm of the balancer is parallel to the upper arm. Furthermore, it is assumed that one end of the lever arm is located at the center of mass of the upper arm and its length (L) is equal to 0.3m. In the calculations, it is assumed that 50% of the mass of the upper arm is supported.

In case 1, the exoskeleton is adjusted to fit a new user. The mass of the arm of the first user (m_A) is assumed to be equal to 1.4kg (corresponding to the 5th female percentile [19]) of which 0.7kg is supported. The mass of the arm of the second user (m_B) is assumed to be equal to 2.8kg (corresponding to the 95th male percentile [19]), of which 1.4kg is supported. This results in maximum moments equal to $T_{mass}^A = m_A g L = 0.7 * 9.81 * 0.3 \approx 2\text{Nm}$ and $T_{mass}^B = m_B g L = 1.4 * 9.81 * 0.3 \approx 4\text{Nm}$.

In case 2, the exoskeleton is adjusted to pick up an object of 20kg using two hands. Before the adjustment, only the upper arm is supported. Similar to case 1, the mass of the arm of the user (m_A) is assumed to be equal to 2.8kg of which 50% is supported. This results in a maximum moment equal to $T_{mass}^A = m_A g L = 1.4 * 9.81 * 0.3 \approx 4\text{Nm}$. It is assumed that the mass of the carried object is equal to 13kg and that half of this mass is supported per balancer since the object is carried in both hands. The total supported mass (half of the upper arm mass plus half of the object mass (m_B)) is then equal to 8kg. This results in a maximum moment equal to $T_{mass}^B = m_B g L = 8 * 9.81 * 0.3 \approx 24\text{Nm}$.

Case 3 and 4 are based on a back exoskeleton (comparable to [20]). It is assumed that the lever arm of the balancer is parallel to the human back. The length of the lever arm is assumed to be equal to 0.4m and that the exoskeleton supports half of the mass of the human trunk.

In case 3, the exoskeleton is adjusted to fit a new user. The mass of the trunk of the first user (m_A) is assumed to be equal to 40kg (approximately equal to the male 50th percentile [19]), of which 20kg is supported. The mass of the trunk of the second user (m_B) is assumed to be equal to 50kg (approximately equal to the male 95th percentile [19]), of which 25kg is supported. This results in maximum moments equal to $T_{mass}^A = m_A g L = 20 * 9.81 * 0.4 \approx 80\text{Nm}$ and $T_{mass}^B = m_B g L = 25 * 9.81 * 0.4 \approx 100\text{Nm}$.

In case 4, the exoskeleton is adjusted to pick up an object with a mass equal to 15kg using two hands. It is assumed that the complete mass of the object is supported by the balancer. Similar to case 3, the mass of the trunk of the first user (m_A) is assumed to be equal to 40kg, of which 20kg is supported. The mass of the trunk plus half of the mass of the object that is picked up (m_B) is then equal to 35kg. This results in maximum moments equal to $T_{mass}^A = m_A g L = 20 * 9.81 * 0.4 \approx 80\text{Nm}$ and $T_{mass}^B = m_B g L = 35 * 9.81 * 0.4 \approx 140\text{Nm}$.

APPENDIX III ANALYTICAL DERIVATION OF BALANCING CONDITIONS

The fact that the best balancing quality is obtained by selecting $r >> a$ or $r << a$ is analytically proved in this section. The proof is based on the fact that for an optimal balancing quality, the variation of the total potential energy of the system over the range of motion should be as small as possible. Firstly, the total potential energy of the system (U_p) is derived. Secondly, the derivative of the total potential energy with respect to the range of motion is calculated. The derivative of the total potential energy is equal to the resultant moment about the pivot and should be zero to obtain perfect balance. A term that is dependent on the balancer position remains present in the resultant moment, making it impossible to equalize it to zero for every balancer position. Therefore, thirdly, the conditions for which the influence of this dependent term are minimized are derived. Fourthly, using these conditions, the dependent term can be approximately made constant, and the balancing condition can be solved. This results in the location of the spring attachment point that corresponds to the optimal balancing quality. Please note that:

- 1) The location of the spring attachment point corresponds to the optimal balancing quality, not to perfect balance as in the basic balancer.
- 2) The derivation does not take the maximum energy storage of the spring into account, so the optimal spring attachment point can correspond to plastic deformation of the spring.

AP-type

Step 1: total potential energy of the system

$$U_p = U_{mass} + U_{spring} = mgL\cos\varphi + \frac{1}{2}k * dL_{spring}^2 + F_0 * dL_{spring} \quad (17)$$

$$U_p = mgL\cos\varphi + \frac{1}{2}k\left(\sqrt{a^2 + r^2 - 2ar\cos\varphi}\right)^2 + F_0\sqrt{a^2 + r^2 - 2ar\cos\varphi} \quad (18)$$

$$U_p = mgL\cos\varphi + \frac{1}{2}ka^2 + \frac{1}{2}kr^2 - kar\cos\varphi + F_0\sqrt{a^2 + r^2 - 2ar\cos\varphi} \quad (19)$$

Step 2: constant total potential energy

Perfectly balanced: $U_p \neq f(\varphi) = \text{constant} \implies \frac{dU_p}{d\varphi} = T_T = 0$

$$T_T = \frac{dU_p}{d\varphi} = -mgL\sin\varphi + kar\sin\varphi + \frac{F_0r\sin\varphi}{\sqrt{a^2 + r^2 - 2ar\cos\varphi}} \quad (20)$$

$$T_T = -mgL + kar + \frac{F_0ar}{\sqrt{a^2 + r^2 - 2ar\cos\varphi}} = 0 \quad (21)$$

Step 3: conditions to make the potential energy approximately constant

To fulfill equation 21 \implies make it independent of $\varphi \implies \sqrt{a^2 + r^2 - 2ar\cos\varphi} \neq f(\varphi)$.

Cond. 1: $r << a \implies \sqrt{a^2 + r^2 - 2ar\cos\varphi} \approx \sqrt{a^2 + 0^2 - 2a * 0 * \cos\varphi} = \sqrt{a^2} = a$

Cond. 2: $r >> a \implies \sqrt{a^2 + r^2 - 2ar\cos\varphi} \approx \sqrt{0^2 + r^2 - 2r * 0 * \cos\varphi} = \sqrt{r^2} = r$

Step 4: optimal location of spring attachment point

Cond. 1:

$$r << a \implies -mgL + kar + \frac{F_0ar}{a} = 0 \implies a = \frac{mgL}{kr} - \frac{F_0}{k} \quad (22)$$

Cond. 2:

$$r >> a \implies -mgL + kar + \frac{F_0ar}{r} = 0 \implies a = \frac{mgL}{kr + F_0} \quad (23)$$

ANP-type

Step 1: total potential energy of the system

$$U_p = U_{mass} + U_{spring} = mgL\cos\varphi + \frac{1}{2}k * dL_{spring}^2 + F_0 * dL_{spring} \quad (24)$$

$$U_p = mgL\cos\varphi + \frac{1}{2}k\left(\sqrt{a^2 + r^2 - 2ar\cos\varphi} - L_0\right)^2 + F_0\sqrt{a^2 + r^2 - 2ar\cos\varphi} \quad (25)$$

$$U_p = mgL\cos\varphi + \frac{1}{2}k\left(a^2 + r^2 - 2ar\cos\varphi - 2L_0\sqrt{a^2 + r^2 - 2ar\cos\varphi} + L_0^2\right) + F_0\sqrt{a^2 + r^2 - 2ar\cos\varphi} \quad (26)$$

$$U_p = mgL\cos\varphi + \frac{1}{2}ka^2 + \frac{1}{2}kr^2 - kar\cos\varphi - kL_0\sqrt{a^2 + r^2 - 2ar\cos\varphi} + \frac{1}{2}kL_0^2 + F_0\sqrt{a^2 + r^2 - 2ar\cos\varphi} \quad (27)$$

Step 2: constant total potential energy

Perfectly balanced: $U_p \neq f(\varphi) = \text{constant} \implies \frac{dU_p}{d\varphi} = T_T = 0$

$$T_T = \frac{dU_p}{d\varphi} = -mgL\sin\varphi + kar\sin\varphi - \frac{L_0akr\sin\varphi}{\sqrt{a^2 + r^2 - 2ar\cos\varphi}} + \frac{F_0rasin\varphi}{\sqrt{a^2 + r^2 - 2ar\cos\varphi}} \quad (28)$$

$$T_T = -mgL + kar - \frac{ar(kL_0 - F_0)}{\sqrt{a^2 + r^2 - 2ar\cos\varphi}} = 0 \quad (29)$$

Step 3: conditions to make the potential energy approximately constant

To fulfill equation 29 \implies make it independent of $\varphi \implies \sqrt{a^2 + r^2 - 2ar\cos\varphi} \neq f(\varphi)$.

Cond. 1: $r \ll a \implies \sqrt{a^2 + r^2 - 2ar\cos\varphi} \approx \sqrt{a^2 + 0^2 - 2a * 0 * \cos\varphi} = \sqrt{a^2} = a$

Cond. 2: $r >> a \implies \sqrt{a^2 + r^2 - 2ar\cos\varphi} \approx \sqrt{0^2 + r^2 - 2r * 0 * \cos\varphi} = \sqrt{r^2} = r$

Step 4: optimal location of spring attachment point

Cond. 1:

$$r \ll a \implies -mgL + kar - \frac{ar(kL_0 - F_0)}{a} = 0 \implies a = \frac{mgL}{kr} - \frac{F_0}{k} + L_0 \quad (30)$$

Cond. 2:

$$r >> a \implies -mgL + kar - \frac{ar(kL_0 - F_0)}{r} = 0 \implies a = \frac{mgL}{kr - kL_0 + F_0} \quad (31)$$

FL-type

Step 1: total potential energy of the system

$$U_p = U_{mass} + U_{spring} = mgL\cos\varphi + \frac{1}{2}k * dL_{spring}^2 + F_0 * dL_{spring} \quad (32)$$

$$U_p = m_B g L \cos \varphi + \frac{1}{2}k \left(\sqrt{a^2 + r^2 - 2ar \cos \varphi} + u_0 \right)^2 + F_0 \left(\sqrt{a^2 + r^2 - 2ar \cos \varphi} + u_0 \right) \quad (33)$$

$$U_p = m_B g L \cos \varphi + \frac{1}{2}k \left(a^2 + r^2 - 2ar \cos \varphi + 2u_0 \sqrt{a^2 + r^2 - 2ar \cos \varphi} + u_0^2 \right) + F_0 \left(\sqrt{a^2 + r^2 - 2ar \cos \varphi} + u_0 \right) \quad (34)$$

$$U_p = m_B g L \cos \varphi + \frac{1}{2}ka^2 + \frac{1}{2}kr^2 - kar \cos \varphi + ku_0 \sqrt{a^2 + r^2 - 2ar \cos \varphi} + \frac{1}{2}ku_0^2 + F_0 \sqrt{a^2 + r^2 - 2ar \cos \varphi} + F_0 u_0 \quad (35)$$

Step 2: constant total potential energy

Perfectly balanced: $U_p \neq f(\varphi) = \text{constant} \implies \frac{dU_p}{d\varphi} = T_T = 0$

$$T_T = \frac{dU_p}{d\varphi} = -m_B g L \sin \varphi + kar \sin \varphi + \frac{u_0 a k r \sin \varphi}{\sqrt{a^2 + r^2 - 2ar \cos \varphi}} + \frac{F_0 r a \sin \varphi}{\sqrt{a^2 + r^2 - 2ar \cos \varphi}} \quad (36)$$

$$T_T = -m_B g L + kar + \frac{ar(ku_0 + F_0)}{\sqrt{a^2 + r^2 - 2ar \cos \varphi}} = 0 \quad (37)$$

Step 3: conditions to make the potential energy approximately constant

To fulfill equation 37 \implies make it independent of $\varphi \implies \sqrt{a^2 + r^2 - 2ar \cos \varphi} \neq f(\varphi)$.

Cond. 1: $r \ll a \implies \sqrt{a^2 + r^2 - 2ar \cos \varphi} \approx \sqrt{a^2 + 0^2 - 2a * 0 * \cos \varphi} = \sqrt{a^2} = a$

Cond. 2: $r \gg a \implies \sqrt{a^2 + r^2 - 2ar \cos \varphi} \approx \sqrt{0^2 + r^2 - 2r * 0 * \cos \varphi} = \sqrt{r^2} = r$

Step 4: optimal location of spring attachment point

Cond. 1:

$$r \ll a \implies -m_B g L + kar + \frac{ar(ku_0 + F_0)}{a} = 0 \implies u_0 = \frac{m_B g L}{kr} - a - \frac{F_0}{k} = (m_B - m_A) \frac{gL}{kr} - \frac{F_0}{k} \quad (38)$$

Cond. 2:

$$r \gg a \implies -m_B g L + kar + \frac{ar(ku_0 + F_0)}{r} = 0 \implies u_0 = \frac{m_B g L}{ka} - \frac{F_0}{k} - r \quad (39)$$

4

DISCUSSION

The classification method presented in Chapter 2 can be used to classify force generators with other force-displacement curves than the sinusoidal one of the balancer. In further research, the possible methods to adjust a force generator can be deduced by determining the parameters that scale the transfer function independently of the force generator's degrees of freedom. Moreover, the load and motor function can be formed by the energy-displacement curve as well. For the balancer, this would mean that the sine-function is replaced by a cosine-function. In further research, the classification can be used to design innovative mechanisms. The results show that, in case of a balancer incorporating a linear spring, two novel combinations of motor parameters have been found which can be used to design a new type of balancer. Furthermore, the classification method can be used to classify multiple degree of freedom systems, such as robotic manipulators or exoskeletons. One motor function per degree of freedom can be derived. These motor functions should together form the desired load function. The complete set of combinations of possible motor functions that together form the desired behavior and novel design opportunities can be identified.

In the paper presented in Chapter 3, three adjustment types are proposed and compared on basis of their balancing quality, adjustment energy and dimensions by an analytical model. The results of this study enable the selection of the optimal adjustment type depending on the application's requirements. The practical validity of the results of this study can be improved by incorporating an efficiency in the adjustment energy storage and supply (for more information, see Appendix G). The MATLAB code of the grid-optimization of the adjustment types can be found in Appendix H. Regarding the ANP-type, an alternative optimization method has been considered (Appendix F), but the results proved to be less optimal than the results of the currently implemented grid search. Regarding the implementation of the adjustment types in exoskeletons and robotic manipulators, the FL-type could be suitable in case the difference between the mass moments is small, for example when the exoskeleton is adjusted to another user. The AP-type and ANP-type require a considerable amount of installation space, since for opti-

mal balancing quality, either r or a needs to be large with respect to the other parameter. The AP-type and ANP-type could be suitable for implementation in exoskeletons and robotic manipulators if sufficient installation space is available. However, especially in exoskeletons, the available space is often limited and more compact solutions, for example cam-balancers, seem to be more convenient.

In the proposed adjustment types, springs with realistic properties are selected from a database. The properties of all springs from the database are discussed in Appendix E. To optimize the performance of the adjustment types, a better understanding of the relations between the spring properties could be helpful. For example, when the optimal balancing quality is obtained for a spring with a very large spring stiffness but a small null-length, a relation between those properties enables the selection of the optimal spring. An iterative attempt was made to obtain a relation between the maximum energy storage of the spring and the spring stiffness. As seen in Figure E.2, a line ($U_{spring}^{max} = 2.5e6 \cdot 1/k$) can be drawn through the points that correspond to the springs with the largest energy storage. This relation may be used to estimate the maximum energy storage for a given spring stiffness, but a detailed study is required for validation. The points that represent the springs seem to form a part of a conical shape (Figure E.5). Therefore, a relation between the spring stiffness, null-length and initial tension could exist. However, further research is required to obtain and validate these relations.

The balancer is adjusted when its payload is altered, for example by picking up or releasing an object, which alters the payload of the balancer instantaneously. Theoretically, to optimally reduce the operation power of the system, the adjustment should be performed instantaneously as well. Since energy is involved in the adjustment, instantaneous adjustment implies an infinite amount of involved power (since $power = energy/time$), which is practically impossible. For practical implementation, the adjustment can be performed in a short period of time. This period of time should be in the other magnitude of 10^{-1} seconds to not delay the working pace of the exoskeleton user or robotic manipulator. However, the shorter this period, the larger the involved power and the heavier the required actuators. Furthermore, especially in exoskeletons, quick adjustment introduces safety and comfort concerns. In Appendix B, the functional requirements of an adjustment mechanism applied in an exoskeleton can be found. The importance of the human factors in the design is confirmed by the fact that one of the two functional requirements is about user comfort. The functional requirements can be translated to mechanical functions (Appendix A). The adjustment process needs to be initiated by a trigger. After this initiation, different time strategies can be implemented in the adjustment of the exoskeleton to ensure the user comfort (see Appendix D). If the increase of support takes longer than the decrease, the greatest part of the available adjusting time can be used to increase the support. Further research is required to investigate the optimal time strategy and to practically implement an adjustment mechanism in an exoskeleton or robotic manipulator.

5

CONCLUSION

The overall goal of this thesis was to supply an overview of different mechanism that can supply static balance and to investigate the implementation of these mechanisms in applications in which the payload varies, especially in exoskeletons and robotic manipulators.

In Chapter 2, a method to systematically identify the complete solution space of force generators is introduced. The method is based on the transfer function of the transmission in the force generator and gives insight in the working principle and new design opportunities. The method is applied on perfect static balancers with one non-auxiliary revolute joint to obtain an overview of their working principles. Five distinctive working principles are identified: (1) ideal springs, (2) trigonometric identities, (3) cam mechanisms, (4) cancellation of terms and (5) phase shifted functions. A total of 27 possible transfer functions is determined, of which 22 incorporate a non-linear spring and five are found in existing balancers.

In Chapter 3, the focus was on the application of balancers in exoskeletons and robotic manipulators. Three adjustment types suitable for these applications are selected and the theoretically minimum amount of energy involved in the adjustment is derived. Furthermore, their balancing conditions are determined and a method to select a spring with realistic spring properties is introduced. Convenient metrics are proposed with the theoretically perfect basic balancer as reference. The metrics are used to compare the three adjustment types using an analytical MATLAB model on basis of their degree of perfect balance, energy involved in adjustment and dimensions in four cases deduced from the applications. The selected adjustment types include a balancer (1) with pulley adjusted by altering the location of the spring attachment point (AP-type), (2) without pulley adjusted by altering the location of the spring attachment point (ANP-type) and (3) with pulley adjusted by altering the free length of the spring (FL-type). The theoretical minimum amount of energy involved in adjustment is equal to the difference between the potential energy of the two balanced masses and is dependent on the position of the mass. Due to the initial tension and null-length of the spring, the adjustment types cannot obtain perfect balance over the complete range of motion. The results show that

there are clear differences in the performance of the different adjustment types. In case the balancing quality is critical, AP-type adjustment is the best choice (error within 3%), closely followed by ANP-type (error within 5%). An advantage of the ANP-type over the AP-type is that no pulley is required, which simplifies the mechanism and increases its robustness. However, the dimensions of the ANP-type are larger than the dimensions of the AP-type. The adjustment energy of both the AP-type and ANP-type is approximately 10% larger than the minimum adjustment energy. The FL-type should only be considered when the percentage difference between the two masses is smaller than 20%, since the balancing quality deteriorates significantly for larger differences. Furthermore, its adjustment energy is generally four to six times larger than the theoretical minimum adjustment energy. Regarding their implementation in exoskeletons and robotic manipulators, the FL-type is suitable when the difference between the moments of the balanced masses is small. The AP-type and ANP-type are suitable when sufficient installation space, at least in one direction, is available. The amount of required installation space is dependent on the balanced masses. However, more compact solutions could be obtained by incorporating an transmission mechanism in the balancer, but further research is required to investigate their performance on adjustment energy and balancing quality.

5

In Appendices A to H, additional information about the complete research is discussed. Although the mechanical considerations are important in the design of balancers applied in exoskeletons, the human aspects cannot be neglected. Functional requirements and general functions of the adjustment mechanism are identified to ensure the comfort and safety of the user. The functional requirements include ensuring user comfort and altering the energy contained in the system. The general functions comprise of (1) locking and unlocking the spring attachment point, (2) supplying and retrieving/dissipating the adjustment energy and (3) guiding the displacement of the spring attachment point. The first function can optimally be performed by a locking mechanism based on singularity. Regarding the second function, important characteristics of the energy storage device are (1) energy density, which describes how much energy can be stored per unit mass (2) power density, which describes how fast the energy can be retrieved from the storage and (3) energy storage duration, which indicates for what period of time the energy can be stored in the energy storage device. No energy storage device that fulfills all the requirements was found, and therefore a combination of a spring (to quickly supply energy) and a battery (to store a large amount of energy) was proposed. Considering the third function, the path of the spring attachment point determines the behavior of the device during the adjustment. Two aspects need to be taken into account in selecting the path, (1) the mechanical possibilities and (2) the user experience. The performance of the adjustment types could be improved by optimizing the spring selection. Expressions that describe the relations between the spring properties could be help full in this optimizing. The relation between the spring stiffness, null-length and initial tension of the springs from the database was investigated using a graphical representation, which indicated that those relations exist. However, more research is required to obtain the mathematical relations between those properties.

In conclusion, this study showed that, by classifying static balancers (or force generators in general), a systematic overview of all design opportunities is created. Furthermore, the results indicate that certain adjustment types of static balancers can be implemented in applications in which their payload varies regularly, taking in consideration the balancing quality, energy involved in adjustment and overall dimensions of the balancer. However, since especially in exoskeletons the amount of available space is limited, more compact balancers could be more suitable for this application.

BIBLIOGRAPHY

- [1] V.Arakelian. "Gravity compensation in robotics". In: *Advanced Robotics* 30.2 (2016), pp. 79–96.
- [2] R. Barents. "Literature review". Delft University of Technology.
- [3] R. Barents et al. "Spring-to-spring balancing as energy-free adjustment method in gravity equilibrators". In: *Journal of Mechanical Design* 133 (2011).
- [4] Y.R. Chhetra and R.M Joshi. "A review on passive gravity compensation". In: *International Conference on Electronics* (2017), pp. 184–189.
- [5] G. Colombo, M. Joerg, and P. Hostettler. "Device and method for automating treadmill therapy". U.S. pat. 6821233. 2004.
- [6] ME41095 BioInspired Design. "Creating creativity". Delft Technical University. 2018.
- [7] F. Díaz-González et al. "A review of energy storage technologies for wind power applications". In: *Renewable and sustainable energy reviews* 16 (2012), pp. 2154–2171.
- [8] W.D. van Dorsser et al. "Energy free adjustment simultaneous displacement". In: *ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (2006).
- [9] W.D. van Dorsser et al. "Energy-free adjustment of gravity equilibrators by adjusting the spring stiffness". In: *Proc. IMechE* 222 (2008), pp. 1839–1846.
- [10] A. Fattah and Agrawal S.K. "Gravity balancing of a human leg using an external orthosis". In: *Journal of Medical Devices* 5 (2011).
- [11] A. Fattah and Agrawal S.K. "Gravity-balancing of classes of industrial robots". In: *Proceedings of 2006 IEEE International Conference on Robotics and Automation* (2006), pp. 2872–2877.
- [12] V. Geinitz et al. "Relaxation of helical springs and spring steel wire". In: *56th International Scientific Colloquium* (2011).
- [13] *Gutekunst Federn*. <https://www.federnshop.com/en/products/extensionsprings.html>, 2019.
- [14] J.L. Herder. "Energy-free systems". PhD Thesis, Delft University of Technology.
- [15] J.L. Herder et al. "Principle and design of a mobile arm support device for people with muscular weakness". In: *Journal of Rehabilitation Research Development* 43.5 (2006), pp. 591–604.
- [16] H.C. Hsieh, L. Chien, and C.C Lan. "Mechanical design of a gravity-balancing wearable exoskeleton for the motion enhancement of human upper limb". In: *IEEE International Conference on Robotics and Automation (ICRA)* (2015), pp. 4992–4997.

- 5
- [17] H. Ibrahim, A. Ilinca, and J. Perron. "Energy storage devices - Characteristics and comparisons". In: *Renewable and sustainable energy reviews* 12 (2008), pp. 1221–1250.
 - [18] JM Energy. www.jmenergy.co.jp/en/lithium_ion_capacitor/, 2019.
 - [19] M.G. de Jong. "Synthesis of a force generator using two-fold tape loops". Masters Thesis, Delft University of Technology.
 - [20] K. Kobayashi. "Comparison Between Spring Balancer and Gravity Balancer in Inertia Force and Performance". In: *Journal of Mechanical Design* 123 (2001), pp. 549–555.
 - [21] Q. Lu, C. Ortega, and O. Ma. "Passive Gravity Compensation Mechanisms: Technologies and Applications". In: *Recent papers on Engineering* 5 (2011), pp. 32–44.
 - [22] R.H. Nathan. "A constant force generation mechanism". In: *Transactions of the ASME* 107 (1985), pp. 508–512.
 - [23] Y. Ono and T. Morita. "An underactuated manipulation method using a mechanical gravity canceller". In: *Journal of Robotics and Mechatronics* 16.6 (2004), pp. 563–569.
 - [24] M. Plooij et al. "Review of locking devices used in robotics". In: *Energies* 22.1 (2015).
 - [25] T. Rahman and W. Sample. "Orthosis device". U.S. pat. 6821259. 2004.
 - [26] B.L. Rijff. *An overview of adjustable vertically translating gravity balancers*. Delft University of Technology. 2009.
 - [27] F. Rossi, B. Castellani, and A. Nicolini. "Benefits and challenges of mechanical spring systems for energy storage applications". In: *Energy Procedia* 82 (2015), pp. 805–810.
 - [28] A. Russo, Sinatra R., and Xi F. "Static balancing of parallel robots". In: *Mechanism and Machine Theory* 40 (2005), pp. 191–202.
 - [29] P. Sapin et al. "Thermodynamic losses in a gas spring: comparison of experimental and numerical results". In: *12th International Conference on Heat Transfer, Fluid Mechanics and Thermodynamics* (2016), pp. 172–177.
 - [30] Special Springs. <https://www.specialsprings.com/en/products/gas-springs/>, 2019.
 - [31] A.T. Steinhorsson et al. "Review, categorization and comparison of 1 DOF static balancers". In: *Proceedings of the ASME 2015 International Design Engineering Technical Conferences Computers and Information in Engineering Conference* (2015).
 - [32] A.H.A. Stienen et al. "Freebal: dedicated gravity compensation for the upper extremities". In: *Proceedings of the 2007 IEEE 10th International conference on Rehabilitation Robotics* (2007), pp. 804–808.
 - [33] Laliberté T., Gosselin C.M., and Jean M. "Static balancing of 3-DOF planar parallel mechanisms". In: *IEEE/ASME transactions on mechatronics* 4.4 (1999), pp. 363–377.
 - [34] J. Wang and C.M. Gosselin. "Static balancing of spatial four-degree-of-freedom parallel mechanisms". In: *Mechanism and Machine Theory* 35 (2000), pp. 563–592.

- [35] J. Wang et al. "Overview of compressed air energy storage and technology development". In: *IEEE Robotics and Automation Magazine* 991.10 (2015).

A

APPENDIX A: FUNCTIONS OF AN ADJUSTMENT MECHANISM IN AN EXOSKELETON

In Figure A.1, the general actions involved in adjusting a balancer in an upper extremity exoskeleton are shown. From the general actions, functions of the adjustment mechanism are derived and described in Figure A.2 and illustrated in Figure A.3. The part of the adjustment mechanism that determines when the adjustment is required is considered to be out of scope of this study. This part of the mechanism should recognise the intention of the user and act upon this.

Starting in situation A, the spring attachment point is locked in position A (Figure A.3a). To perform the adjustment, the position of the mass needs to be locked. This can be done by holding the human arm still. Next, if a mechanical stop is implemented to prevent the spring attachment point from moving in a direction other than to point B, the spring attachment point can be unlocked (Figure A.3b1). When no stop is implemented, the actuator force needs to increase until at least static equilibrium is obtained (Figure A.3b2), before it can be unlocked, to prevent movement in opposite direction than point B. The actuator force needs to be increased until at least static equilibrium in position B is obtained (Figure A.3c1 and A.3c2), so that the spring attachment point reaches position B (Figure A.3d). The spring attachment point can now be locked again at position B and the adjustment from situation A to B is finished (Figure A.3e).

To perform the adjustment from situation B to A, the spring attachment point can be unlocked while the actuator force ensures that the resulting force on the spring attachment point is in the direction of B to A (Figure A.3f). The resultant force should be such that position A is reached (figure A.3g) in which the spring attachment point can be locked again (Figure A.3g).

A

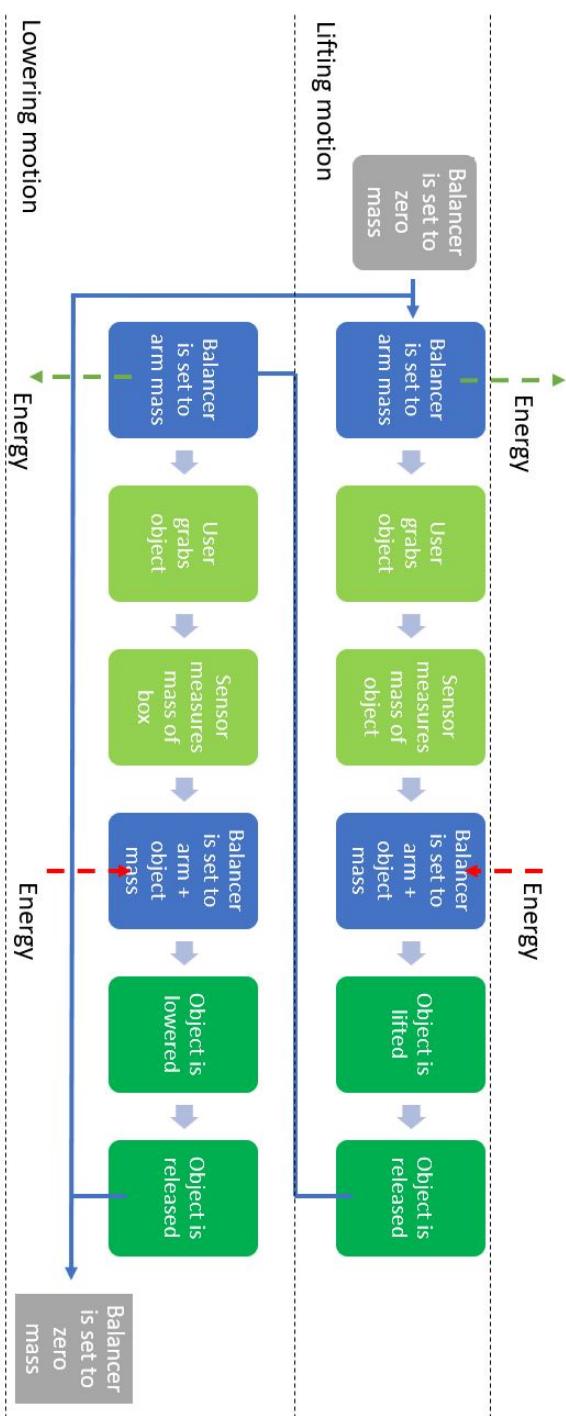


Figure A.1: General scheme of actions in adjusting a balancer in an exoskeleton

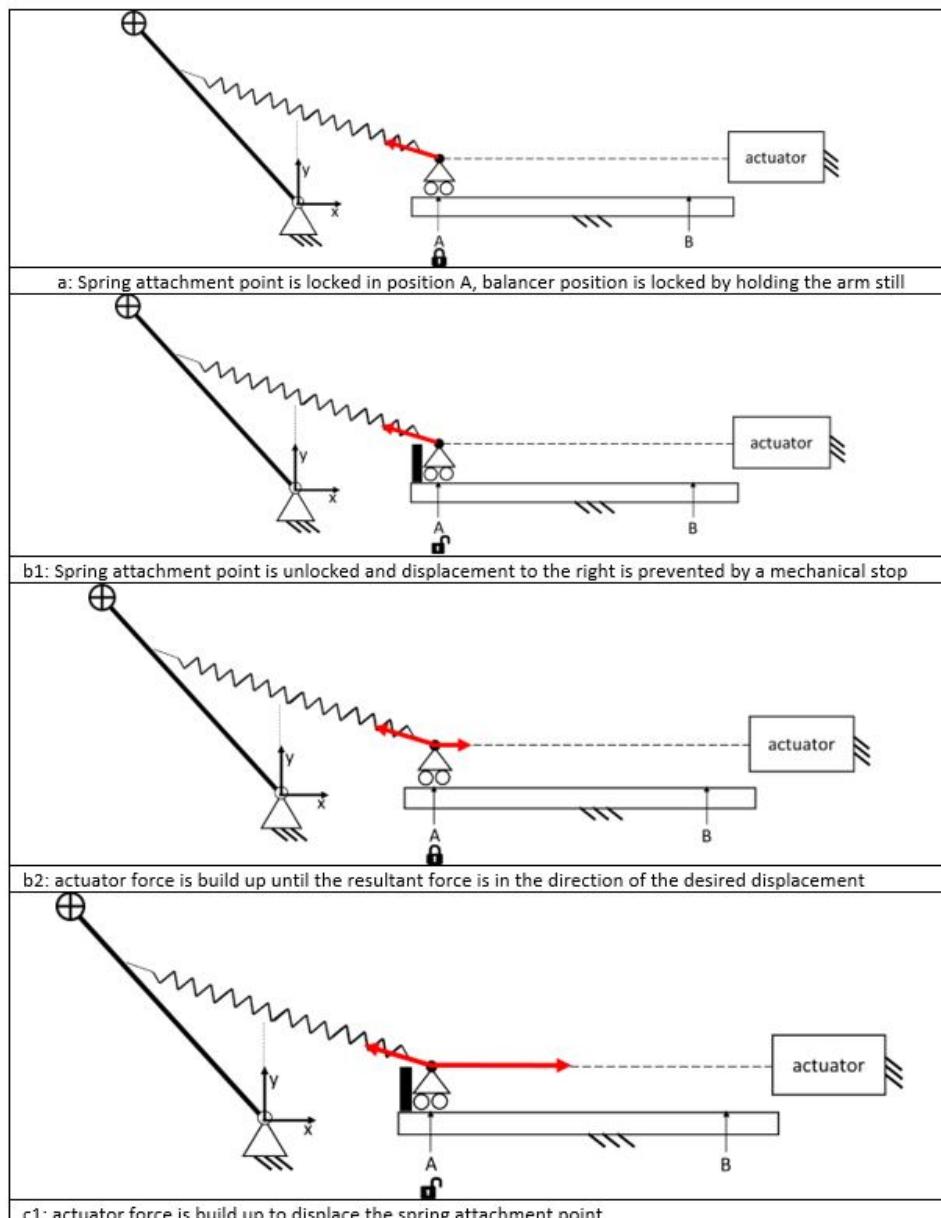
Table 8: functions of the adjustment mechanism

Adjusting from A to B	1	Unlocking spring attachment point in situation A	The spring attachment point needs to be displaced during the adjustment, therefore it needs to be unlocked so that displacement is possible. The unlocking needs to be possible while a force on the spring attachment point is applied.
	2	Prevent displacement of spring attachment point in wrong direction in situation A	When the spring attachment point is unlocked, the balancer spring still exerts a force on it. In situation A, this force moves the point in the wrong direction, which needs to be prevented.
	3	Providing force to displace from A to B	In adjusting from A to B, the balancer spring force needs to be counteracted by the force exerted by the adjustment mechanism to displace it.
	4	Transferring force to spring attachment point	The force that is exerted by the adjustment mechanism needs to be transferred to the spring attachment point.
	5	Guiding displacement from A to B	The spring attachment point needs to be displaced from A to B, but the path is not defined.
	6	Locking spring attachment point in situation B	The balancer spring exerts a force on the spring attachment point. To hinder movement of the spring attachment point, the point should be locked.
	7	Unlocking spring attachment point B	The spring attachment point needs to be displaced during the adjustment, therefore it needs to be unlocked so that displacement is possible. The unlocking needs to be possible
Adjusting from B to A			

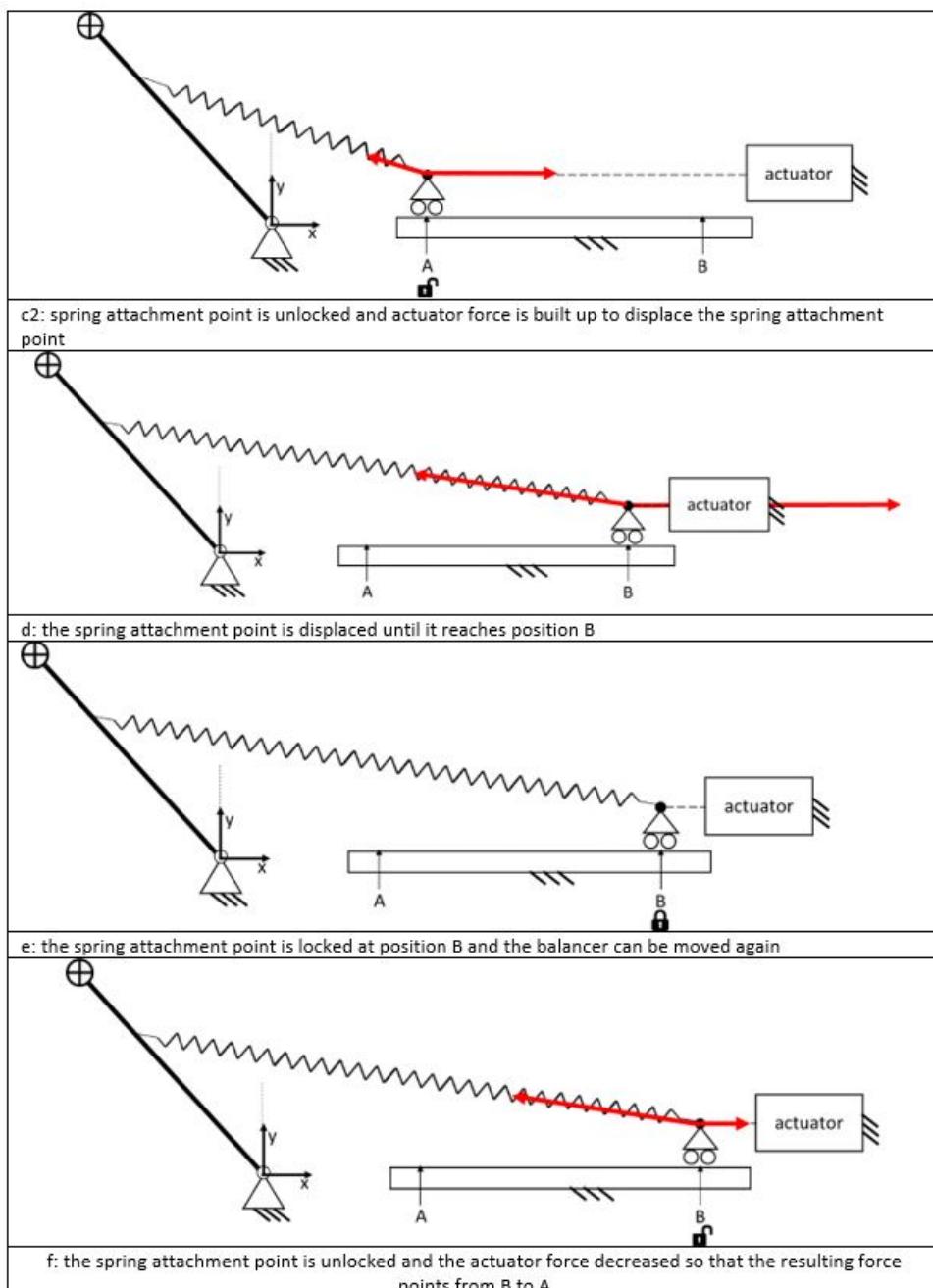
A

		while a force on the spring attachment point is applied.
8	Prevent displacement of spring attachment point in wrong direction in situation B	When the spring attachment point is unlocked, the adjustment mechanism may exert a force on it. In situation B, this force moves the point in the wrong direction, which needs to be prevented.
9	Providing force to displace from B to A	In adjusting from B to A, the resulting force needs to be in the direction of B to A. Therefore, the adjustment mechanism force needs to be less than the balancing spring force.
10	Transferring force to spring attachment point	The force that is exerted by the adjustment mechanism needs to be transferred to the spring attachment point.
11	Guiding displacement from B to A	The spring attachment point needs to be displaced from A to B, but the path is not defined.
12	Locking spring attachment point in situation A	The balancer spring exerts a force on the spring attachment point. To hinder movement of the spring attachment point, the point should be locked.

Figure A.2: Functions of adjustment mechanism applied in an exoskeleton



A



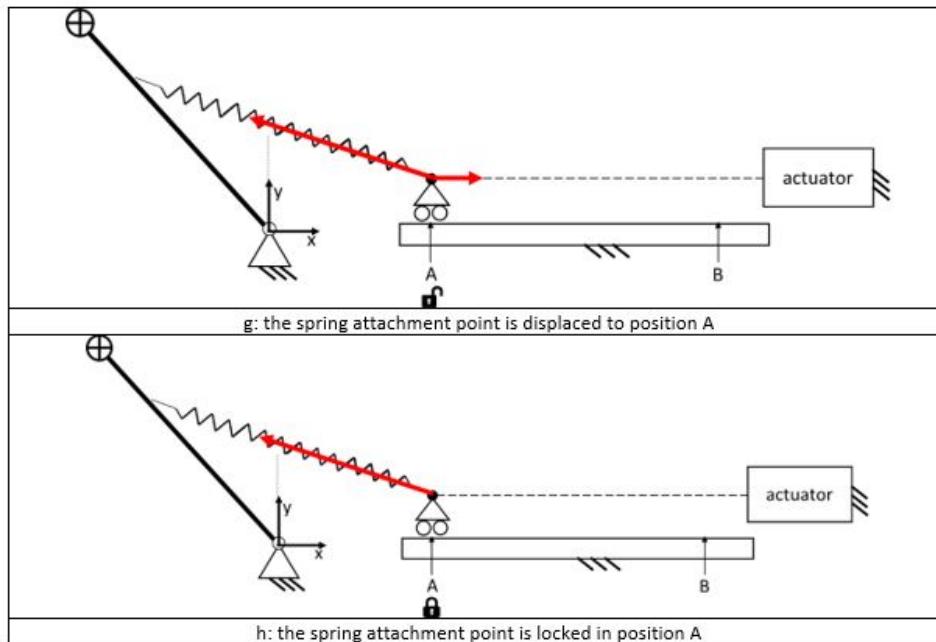


Figure A.3: Illustration of functions of adjustment mechanism applied in an exoskeleton

B

APPENDIX B: FUNCTIONAL REQUIREMENTS AND STRATEGIES OF AN ADJUSTMENT MECHANISM IN AN EXOSKELETON

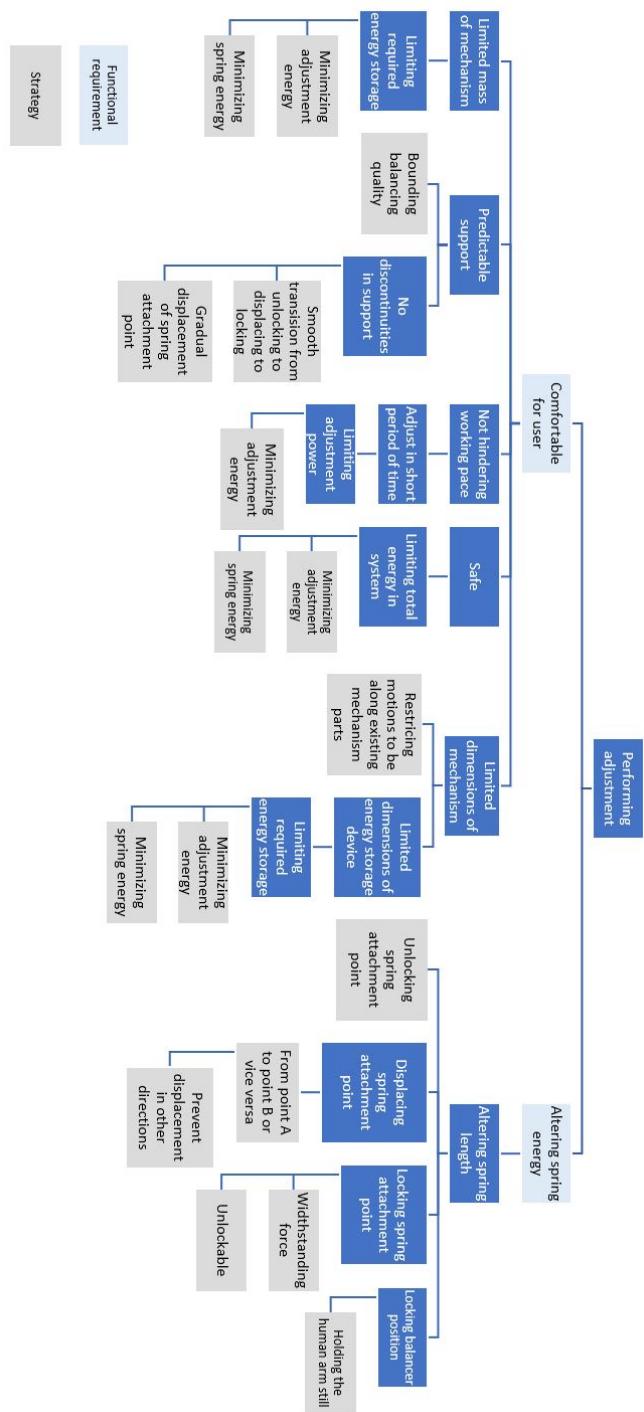


Figure B.1: Functional requirements and strategies regarding an adjustment mechanism in an exoskeleton

C

APPENDIX C: DESIGN OF AN ADJUSTMENT MECHANISM APPLIED IN AN UPPER EXTREMITY EXOSKELETON

In this Section, the mechanical parts of the adjustment mechanism are discussed. The focus is on the adjustment of a balancer applied in an upper extremity exoskeleton. Therefore, the two masses that are supported are the mass of the human arm (situation A) and the mass of the human arm plus (part of) the mass of a carried object (situation B). Furthermore, for comfort reasons, the mechanism should be as lightweight and lean as possible.

C.1. PATH OF SPRING ATTACHMENT POINT

The spring attachment point is displaced to adjust the balancer. The path over which this point is displaced determines the behaviour of the balancer during the adjustment. Two important aspects need to be considered in the determination of the adjustment path.

The first aspect is about the mechanical possibilities. Since the adjustment has to be performed quickly, it is important to consider which path can be travelled quickly. A circular path can be preferable when an rotational motor with lever arm is used. Other options include a straight path using a linear guide or a custom path using for example a four-bar mechanism. The second aspect is about the user experience. Since the balancer is locked by holding still the arm of the user, the balancer exerts a force on the arm during adjustment. If a gradual increase of the support is preferred during adjustment, the optimal adjustment path is a straight line between the spring attachment points corresponding to situation A and B.

C.2. LOCKING AND UNLOCKING THE SPRING ATTACHMENT POINT

During operation of the exoskeleton, the spring attachment point should be fixed at a certain location. During adjustment, however, the spring attachment point is displaced. Therefore, it is required that the spring attachment point can be unlocked. Furthermore, the unlocking should be possible when a force is exerted on the spring attachment point by the spring.

C

A locking device based on singularity is particularly suitable in for this application, since it requires no continuous power consumption, can be unlocked under load and requires a small amount of power to be unlocked [24]. A locking device based on singularity is designed, inspired by the design of ir. M. Hölscher¹. Hölscher developed a locking device for a rotational joint based on singularity (Figure C.1).

A schematic diagram of the locking device can be found in Figure C.2a. The rotational joint that is to be locked consists of two rigid links (L1 and L2) which are connected by a rotational degree of freedom (J1). The rotation of the links is locked by a third link (L3), which is connected to link 1 by a revolute joint (J2). One end of L3 can fit in a slot on L2, which locks the degree of freedom of J1. To unlock, the end of L3 is rotated out of the slot by rotating L3 about J2. In Figure C.2b, the relevant forces working on the locking device are shown. The surface of the slot in L2 is shaped such that the force exerted by L2 on the end of L3 intersects J2. The moment of this force about J2 is zero since its moment arm is zero. Therefore, the moment exerted by L2 on L1 does not influence the moment required to unlock J1 by rotating L3. In fact, the moment required to unlock J1 is only dependent on the friction between the surface of L2 and L3.



Figure C.1: Locking device of Hölscher

¹Contact: mike@laevo.nl

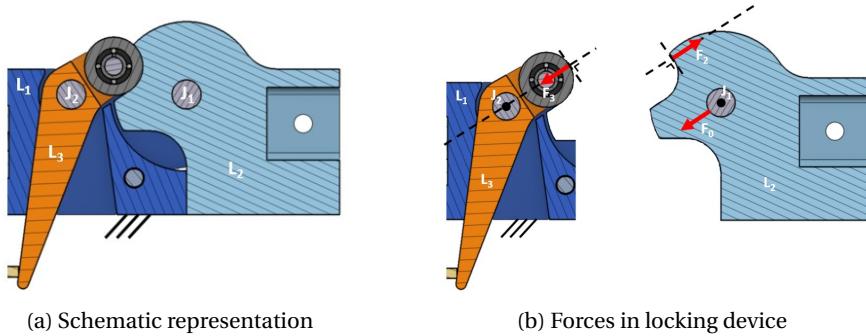


Figure C.2: Schematic representation of locking device of Hölscher

C

To enable the use of a linear guide to define the path of the spring attachment point, a locking device is designed for a carriage on a linear guide (Figure C.3). The principle is identical to that of the locking device of Hölscher: the force working on the lever arm intersects its revolute joint. Therefore, this force does not exert a moment on the lever arm and only the friction between in the ball bearing and needs to be overcome to unlock the device.

The front of the carriage is shaped such that when it reaches the lever arm, the lever arm is lifted upwards and falls into the slot. In future design, a spring can be incorporated to push the lever arm on the carriage to enable more secure locking.

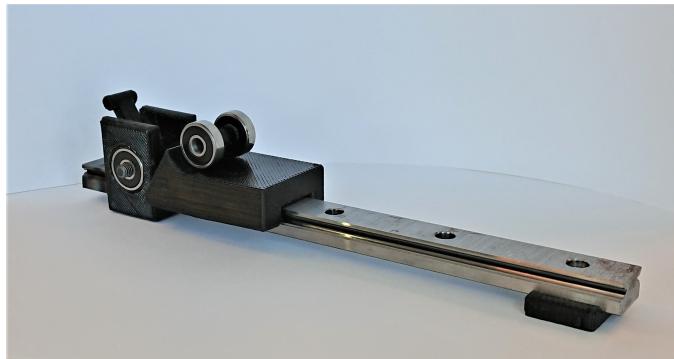


Figure C.3: Locking device based on singularity

C.3. ENERGY STORAGE

As explained in Chapter 3, the energy involved in adjusting from situation A to B in a specific balancer position is equal but opposite to the energy involved in adjusting from situation B to A in that same balancer position. In other words, if the energy can be retrieved and stored at perfect efficiency and on average the number of adjustments from situation A to B is equal to the number of adjustment from situation B to A in one bal-

cancer position, no external energy is involved in the adjustment. However, practically, storing and retrieving energy at perfect efficiency is not possible. Furthermore, it is unlikely that the number and positions of adjustments from situation A to B and vice versa is equal. Therefore, external energy needs to be stored in the exoskeleton before using it. This energy can then be used in the adjustment².

The energy that is involved in the adjustment needs to be stored in an energy storage device (ESD). Three important characteristics of the energy storage device are considered:

1. Energy density: how much energy [J] can be stored per unit mass [kg] of the ESD?
2. Power density: how much power [W] can be stored per unit mass [kg] of the ESD?
3. Storage duration: for what period of time can the energy and power be stored in the ESD?

The three characteristics of different storage devices are listed in Table C.1. The difference between the energy density and power density is that the latter characteristic tells something about the speed with which the energy can be retrieved from the ESD.

The required mass of the different ESDs is calculated on basis of comparison case 2 (see Chapter 3). In this case, the (worst case) adjustment energy is equal to 20J and it is assumed that the adjustment takes place in 0.1 seconds. The results are shown in Table C.2. As seen in this Table, for the electrical ESDs and the flywheel, the required mass is determined by the amount of power that needs to be stored. The steel spring and compressed air ESDs can release their energy (almost) instantaneously and therefore the power requirement has no effect on the required mass. Please note that in the two Tables, only one adjustment action is considered. In reality, the adjustment is performed multiple times, which increases the mass of the ESD.

Table C.1: Characteristics of different energy storage devices

Device	Lithium ion battery	Sodium sulfur battery	Lead acid battery	Redox flow battery	Super capacitor	Spring (steel coiled)	Spring (CNT yarn)	Spring (CNT twisted)	Gas spring ³	Flywheel
Energy density [kJ/kg]	540 [7]	540 [7]	140 [7]	90 [7]	10 [7]	0.14 [27]	4.2 [27]	8.3 [27]	0.8 ⁴	150 [27]
Power density [W/kg]	400 [18]	400 [18]	400 [18]	400 [18]	3000 [7]	Inf ⁵	- ⁶	- ⁶	Inf ⁵	12000 [27]
Storage duration	Min-Day [35, 17]	Min-Day [35]	Min-Day [35, 17]	Min-Day [35, 17]	Sec-Hour [35, 17]	Hour-Day ⁷	- ⁶	- ⁶	Hour-Month ⁸	Sec-Min [35, 17]

²Please note that this does not imply that storing the energy that is released during adjustment is not necessary. The more energy that can be retrieved, the less energy needs to be stored in advance in the exoskeleton.

³The characteristics of compressed air are more favourable, but give a distorted view because the mass of the housing and sealing is not taken into account

⁴The energy density is calculated from a gas spring data sheet [30]

Table C.2: Required mass per energy storage device based on energy and power density

Device	Lithium ion battery	Sodium sulfur battery	Lead acid battery	Redox flow battery	Super capacitor	Spring (steel coiled)	Spring (CNT yarn)	Spring (CNT twisted)	Gas spring	Flywheel
Energy mass [g]	0.037	0.037	0.14	0.22	2.0	143	4.8	2.4	25	0.13
Power mass [g]	500	500	500	500	67	0	-	-	0	17

C

The energy needs to be stored for at least a day, to avoid charging during a work shift. This excludes super capacitors and the flywheel. None of the ESDs can store a sufficiently large amount of energy and power in a reasonable mass. For example, to store sufficient energy to perform fifty adjustments⁵, a 7kg steel spring is required. Similarly, to store sufficient power to preform fifty adjustments⁶, a 25kg battery is required. In addition, in the latter case, an actuator that transforms the electrical energy into mechanical energy is required.

A combination of different ESDs is considered to solve this issue. One ESD can be used as an energy reservoir to store the energy required for multiple adjustments. A second ESD can be used to quickly supply the energy during the adjustment. Lithium Ion batteries or Sodium Sulfur batteries are suitable as energy reservoir because of their large energy density. A steel spring or gas spring can deliver energy in a short period of time. The energy stored in the battery can than be used to 'charge' the spring in between adjustments. To store the energy required for fifty adjustments, a battery of a few grams is required. However, the power required to charge the spring determines the mass of the battery. Assuming that there is a two second time window between two adjustments, the maximum required power to charge the spring is equal to 10W. To perform fifty adjustments, a 1.25kg battery is required. The spring only needs to store the power required for one adjustment and a 150g steel spring or a 25g gas spring suffices.

C.3.1. LIMITATIONS AND FURTHER RESEARCH

A disadvantage of combining a battery and a spring is that an additional actuator is required that can transform the electrical energy from the battery into mechanical energy in the spring. A DC-motor seems to be a feasible option. However, the mass of the DC-motor will most likely be a order of magnitude larger than the mass of the ESDs. The DC-motor should be placed in series with the spring, so that the motor can 'charge' the spring in between adjustments. The larger the time between the adjustments, the smaller the power that is required to charge the spring.

In the above calculations, the efficiency of the ESDs and the durability is not taken into account. Furthermore, to extend the life time of a battery, complete discharge should be avoided. Therefore, a ESD with a larger energy and power density than presented in the previous section is required.

⁵The energy can be released (almost) instantaneously, which implies an infinite power density

⁶The power density and storage duration of Carbon NanoTube springs could not be found in literature

⁷After 100 hours, 10% of the force in the spring is lost due to stress relaxation [12]

⁸The losses are minimal, although thermodynamic losses can affect the energy storage duration [29]

⁹Assumed that no energy is retrieved during adjustment and that all the energy in the ESD can be used.

Although the mass of the battery in the solution that combines a battery and spring (1.25kg) is significantly smaller than the mass of a completely electric system (25kg), it is still large in an exoskeleton perspective. By retrieving (part of) the released energy, the required energy storage capacity of the battery can be decreased. A option to make this possible is to use the DC-motor as a generator as well. However, an additional electrical circuit is required to enable storing the retrieved energy in the battery.

A second option could be to incorporate an additional spring that stores the released energy. If the energy is not retrieved, it should be dissipated safely and some form of damping should be incorporated in the adjustment mechanism.

Another option would be to duplicate the combined system, and alternately use each system to perform the adjustment. This doubles the time that is available to charge the spring, and therefore cuts the required power and number of adjustments per battery in half (5W and 25 adjustments respectively). In this case, two 0.31kg batteries are required, which add up to mass of 0.63kg. Also, two springs are required that can to store the power required for one adjustment, which results in a mass of 0.30kg. The total combined mass of the ESDs in the original system (one battery and spring) adds up to 1.4kg. The total mass of the ESDs in the split system (two batteries and two springs) is equal to 0.93kg. However, in the split systems, a transmission that switches between the two springs or two DC-motors are required.

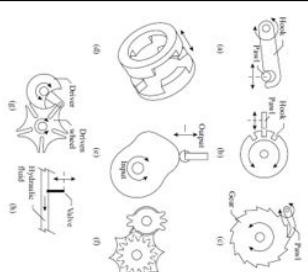
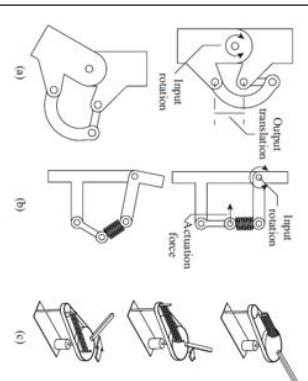
C.4. DESIGN CHOICES

A morphological chart (Figure C.4) is set up including all possible solutions to fulfill the functions of the adjustment mechanism (see Appendix A).

Table 9: morphological chart

	Solution → Function ↓	1	2	3	4	5	6	7
1	Unlocking spring attachment point A	Undo reversible lock	Decreasing counter force	None				
2	Prevent displacement in wrong direction	Unlocking when counter-force is large enough	Mechanical stop	Spindle	Worm gear	None		
3	Providing force to displace from A to B (actuators)	Extension spring	Compression spring	DC-motor	AC-motor	Stepper motor	Magnetic	Hydraulic, pneumatic

Adjusting from A to B

4	Transferring force to spring attachment point	Direct	Pulley + string	Spindle	Worm gear	
5	Guiding displacement from A to B	Square rail	Round rail	Spindle	Worm gear	
6	Locking spring attachment point B (separable and immobile)	Mechanical	Friction	Singularity		
						

7	Unlocking spring attachment point B	Separate shape fit	Decreasing force	None			
8	Prevent displacement in wrong direction	Unlocking when counter-force is large enough	Mechanical stop	Spindle	Worm gear	None	
9	Providing force to displace from B to A	Extension spring	Compression spring	DC-motor	AC-motor	Stepper motor	Magnetic
10	Transferring force to spring attachment point	Direct	Pulley + string	Spindle	Worm gear		
Adjusting from B to A							

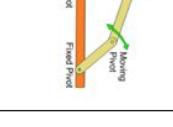
	11 Guiding displacement from B to A	Square rail	Round rail	Spindle	Worm gear	Four-bar mechanism	
12 Locking spring attachment point A (separable and immobile)	Mechanical						

Figure C.4: Morphological chart of adjustment mechanism

D

APPENDIX D: TIME STRATEGIES OF AN ADJUSTMENT MECHANISM IN AN EXOSKELETON

In one displacement motion, the exoskeleton should be adjusted at two instances: from a low to high level of support when grabbing the object and from a high to low level when releasing the object. It is assumed that after grabbing the object, the box held for a period of time (box in hands) after which it is released. Sequentially, the hands are moved towards the next box (returning) and the cycle starts again (see [D.1](#)). The returning period can be used to prepare the actuator to adjust from the low level to the high level of support.

Three basic strategies regarding the distribution of the adjustment actions in the available period of time are distinguished ([Figure D.2](#)). In all strategies, the support starts at the low level of support (just before the object is grabbed) and is at the low level again just after the object is released. In the first strategy (denoted as 1 in [Figure D.2](#)), the support is quickly build up to the high level and then gradually decreased until the low level is reached when the box is released. In the second strategy (2 in [Figure D.2](#)), the level of support is gradually increased until the high level of support is reached and quickly decreased to the low level of support when the box is released. In the third strategy (3 in [Figure D.2](#)), the support is increased and afterwards decreased at a similar rate.

Since the support is gradually decreased in strategy 1 and 3, the moment in time at which the object is released needs to be estimated so that the support is decreased at a sufficiently large rate to prevent overbalancing after the object is released and the support is not decreased too quick while the user still carries the object. In strategy 2, the support is decreased quickly, which decreases the time in which overbalancing is possibly present and in which the user has to support the object. Furthermore, adjusting from a low level to a high level requires energy and to limit the required power, a large

D. APPENDIX D: TIME STRATEGIES OF AN ADJUSTMENT MECHANISM IN AN EXOSKELETON

time window in which the adjustment can be performed is advantageous. Adjusting from a high to a low level releases energy, which can be dissipated or stored more quickly.

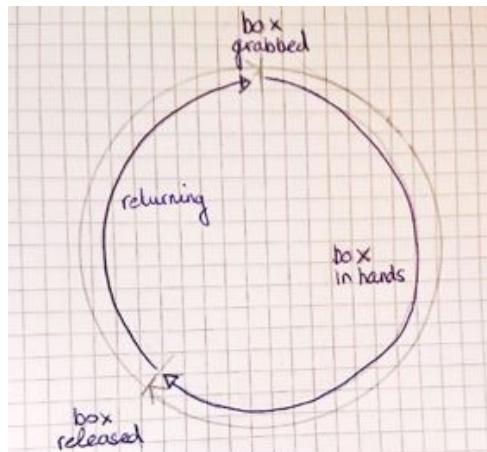


Figure D.1: Time cycle

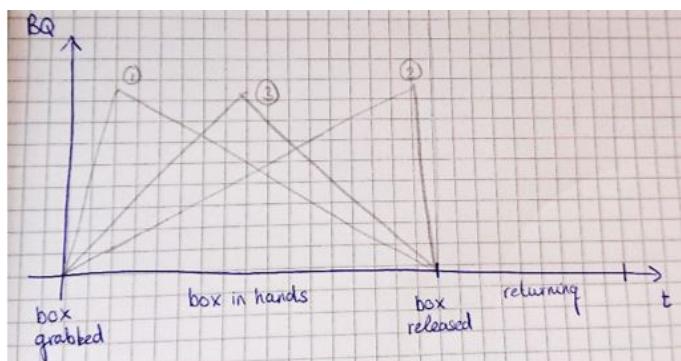


Figure D.2: Time strategies

E

APPENDIX E: PROPERTIES OF SPRINGS FROM DATABASE

A database consisting of commercial available springs [13] is used to select a spring in the design of the adjustment types. Since all spring properties are known, interesting relations between them can be found.

In Figure E.1, it can be seen that the maximum energy storage in the spring is correlated with the spring mass. The maximum energy storage can be calculated by multiplying the spring mass by the energy density of mechanical springs. The energy density can be derived from the data by fitting a linear line and determining its slope and is equal to 111 J/kg for this data set. In Figure E.2 the maximum energy storage (U_{spring}^{max}) is plotted against the spring stiffness (k). It can clearly be seen that the largest amount of energy can be stored in springs with a small stiffness. The maximum energy storage for a given spring stiffness has a clear outer boundary (red line). The red line equals the equation $U_{spring}^{max} = 2.5e6 * 1/k$. Using this equation, the maximum energy storage for a given spring stiffness can be estimated. In Figure E.3 the maximum energy storage is plotted against the null-length of the spring. It can clearly be seen that the largest amount of energy can be stored in a spring with a large null-length. In Figure E.4 the maximum energy storage is plotted against the initial tension in the spring. Two observations stand out. Firstly, there is no clear relation between the maximum energy storage and the initial tension. Secondly, the springs are clustered around certain values for the initial tension. This could show the strategy that is used to design the springs: first the initial tension is determined after which the other spring properties are obtained.

The above described spring properties are combined in Figure E.5. In this Figure, it can be seen that a conical shape is formed by the point corresponding to the individual springs.

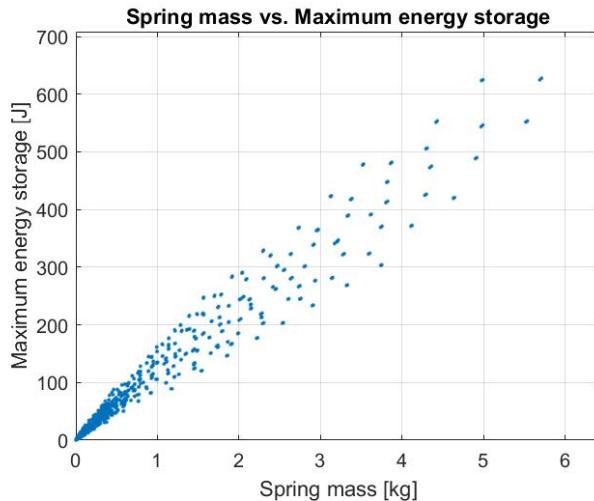


Figure E.1: Maximum energy storage in the spring plotted against the mass of the spring. Data retrieved from [13].

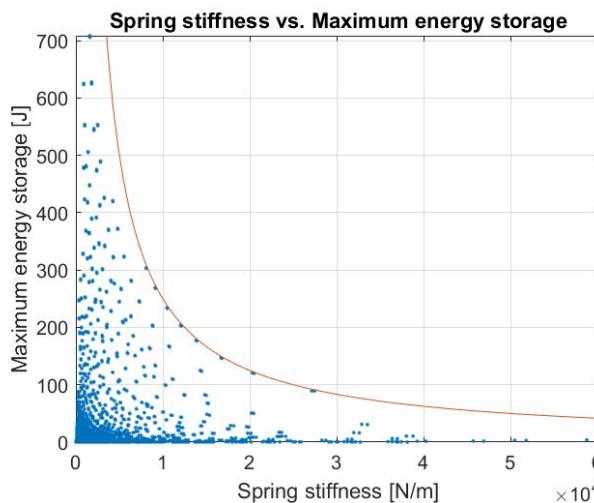


Figure E.2: Maximum energy storage in the spring plotted against the stiffness of the spring. Data retrieved from [13].

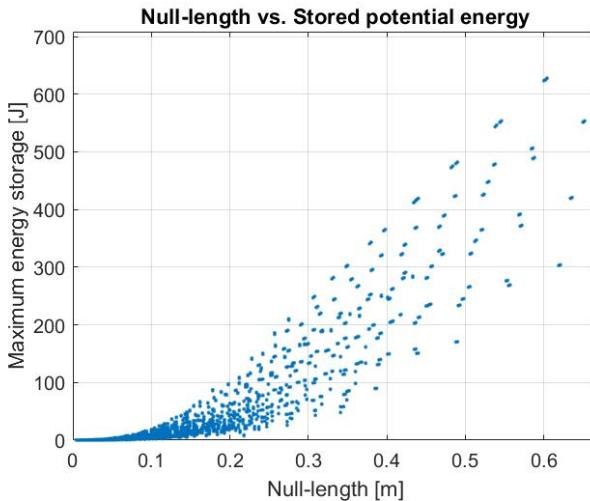


Figure E.3: Maximum energy storage in the spring plotted against the null-length of the spring. Data retrieved from [13].

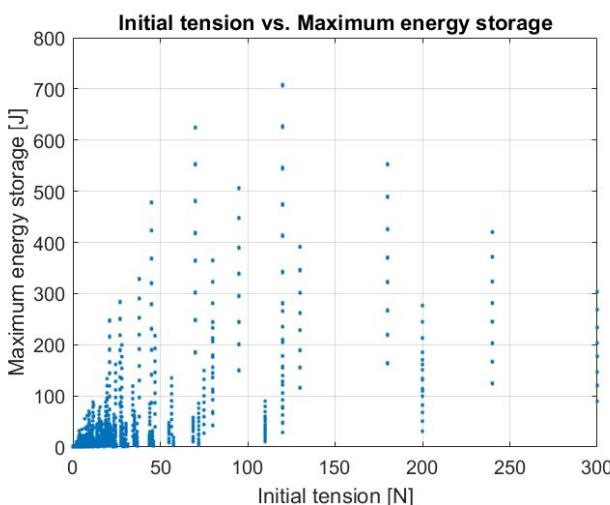


Figure E.4: Maximum energy storage in the spring plotted against the initial tension of the spring. Data retrieved from [13].

E

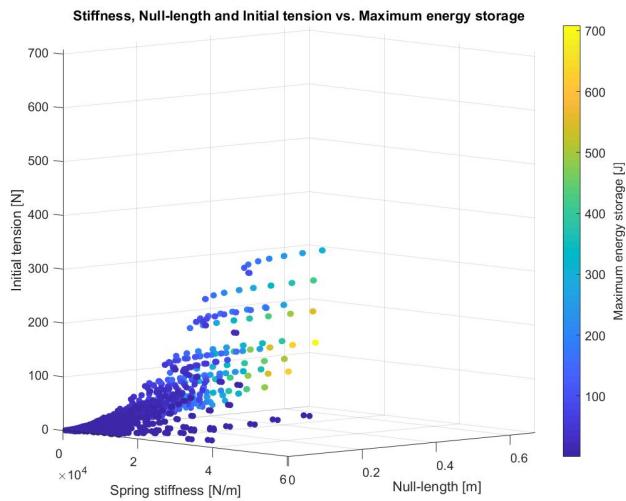


Figure E.5: Null-length, spring stiffness, initial tension and maximum energy storage. Data retrieved from [13].

F

APPENDIX F: CIRCLE OPTIMIZATION

The circle optimization method is introduced by [14] as the *Field Fitting method*. In the optimization, the location of the attachment point of a regular spring to the fixed world is optimized. This evades the necessity of one or multiple pulleys to emulate an ideal spring. The optimization objective is to obtain an optimal balancing quality in the complete range of motion of the balancer. The optimization is based on the fact that in a perfectly balanced system, the total potential energy is constant (equation F1).

$$U_{spring}(\varphi) + U_{mass}(\varphi) = C_B \quad (\text{F1})$$

The energy contained by the spring and mass are given by equation F2 and F3 respectively, in which $dL(\varphi)$ represents the spring elongation.

$$U_{spring}(\varphi) = \frac{1}{2}k * dL(\varphi)^2 \quad (\text{F2})$$

$$U_{mass}(\varphi) = mgL\cos\varphi \quad (\text{F3})$$

Substituting equation F2 and F3 into F1 and rewriting leads to the expression of the required spring elongation (equation F4).

$$dL(\varphi) = \sqrt{\frac{2(C_B - mgL\cos\varphi)}{k}} \quad (\text{F4})$$

The total spring length is then equal to the sum of the null-length (L_0) and the required elongation (equations F5 and F6).

$$L_{spring}(\varphi) = L_0 + dL(\varphi) \quad (\text{F5})$$

$$L_{spring}(\varphi) = L_0 + \sqrt{\frac{2(C_B - mgL\cos\varphi)}{k}} \quad (\text{F6})$$

Using equation E6, at every balancer position, the required spring length to store the energy to perfectly balance the mass can be calculated. A circle with its radius equal to the total spring length (L_{spring}) and midpoint located at the attachment point of the spring at the lever arm can be plotted at each balancer position (see Figure F1). The radius of the circles is increased by increasing the constant C_B until the circle perimeters intersect in one point. In case no point exists at which all circle perimeters intersect, the point with the smallest distance to all circle parameters is selected as optimal point (red dot in Figure F1). The (optimal) intersection point represents the location of point the spring attachment point for which the balancing quality is optimal.

The adjustment principle is obtained by performing the above procedure for two situations, in which mass m_A and m_B are supported respectively. Consequently, two optimal intersection points corresponding to mass m_A and m_B are found. By displacing the spring attachment point from one of those locations to the other, the balancer is adjusted.

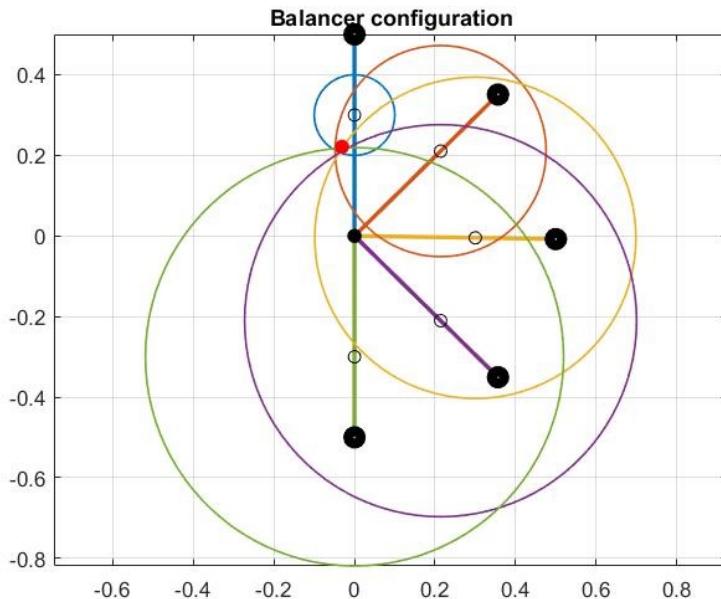


Figure F1: Visualisation of the circle optimization

G

APPENDIX G: NET ADJUSTMENT ENERGY

The adjustment energy as presented in chapter 3 is defined as the net energy involved during adjustment. This means that, when both energy is required (positive sign) and released (negative sign) by the spring during the adjustment, the net adjustment energy is equal to equation G.1. Regarding the AP-type and ANP-type, this is the case if point A intersects the horizontal line through point R (Figure 1 of chapter 3) during the adjustment. Consequently, the spring is consecutively lengthened and shortened (or vice versa) in which energy is required and released. This is primarily the case for designs with a relatively large distance r and when the adjustment takes place at the beginning of the range of motion. The net adjustment energy can thus be divided in required and released energy, as shown in Figure G.1. As seen in this Figure, in part of the range of motion (40-70deg), both energy is required and released during the adjustment.

$$U_{adj}(\varphi) = U_{adj}^{required}(\varphi) + U_{adj}^{released}(\varphi) \quad (\text{G.1})$$

To obtain a situation in which the actual energy that is involved in the adjustment is equal to the net adjustment energy, the released energy should be stored and then supplied at perfect efficiency. In practice, the energy will be stored and supplied with an imperfect efficiency. Therefore, an efficiency factor (ϵ) can be implemented in the calculation of the net adjustment energy (equation G.2). In Figure G.2, the net adjustment energy is shown calculated with an efficiency of 50% ($\epsilon=0.5$). As shown in this Figure, the curve of the net adjustment energy is shifted towards the curve of the required adjustment energy. Furthermore, the energy involved in the adjustment from situation A to B is not longer equal to the energy involved in adjustment from situation B to A (Figure G.3) and the angle at which the net adjustment energy is equal to zero is smaller. If the released energy is completely dissipated ($\epsilon=0$), the net adjustment energy is equal to the required adjustment energy. The minimum adjustment energy in this case is shown in Figure G.4. In this case, a method should be found to safely dissipate the energy.

$$U_{adj}(\varphi) = U_{adj}^{required}(\varphi) + \epsilon U_{adj}^{released}(\varphi) \quad (G.2)$$

In case of the FL-type, since only the magnitude of the spring force is altered during the adjustment process, the spring cannot be consecutively lengthened and shortened in one adjustment process. The energy that is released during an adjustment in which the spring is shortened can be stored to be supplied in an next adjustment in which the spring is lengthened. Thus, the efficiency of the energy storage and supply plays a similar role as described for the AP-type and ANP-type, however, the energy should be stored until the next adjustment.

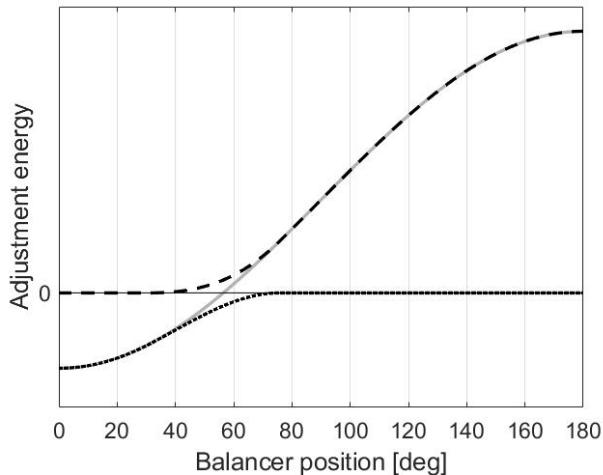


Figure G.1: Example of the AP-type adjustment from situation A to B. Net adjustment energy (continuous line), required adjustment energy (dashed line) and released adjustment energy (dotted line) as function of the range of motion. The net adjustment energy is equal to the sum of the required and released adjustment energy. $T_{mass}^A=6\text{Nm}$, $T_{mass}^B=20\text{Nm}$, $k=500\text{N/m}$, $L_0=0.1\text{m}$, $F_0=10\text{N}$, $r=0.22\text{m}$, $a_A=0.054\text{m}$, $a_B=0.19\text{m}$.

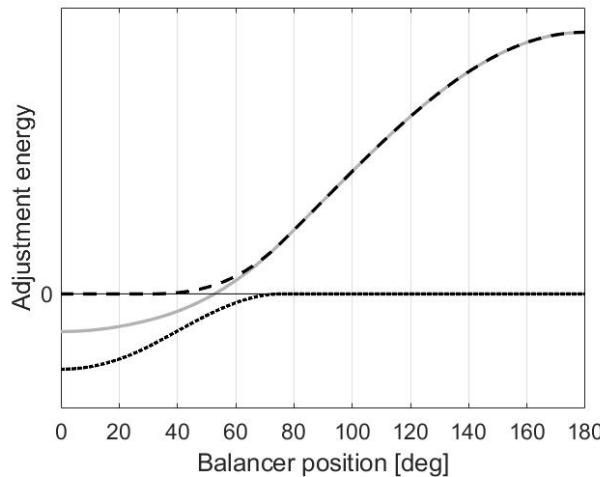


Figure G.2: Example of the AP-type adjustment from situation A to B. Net adjustment energy with an energy storage and supply efficiency of 50% ($\epsilon=0.5$) (continuous line), required adjustment energy (dashed line) and released adjustment energy (dotted line) as function of the range of motion. $T_{mass}^A=6\text{Nm}$, $T_{mass}^B=20\text{Nm}$, $k=500\text{N/m}$, $L_0=0.1\text{m}$, $F_0=10\text{N}$, $r=0.22\text{m}$, $a_A=0.054\text{m}$, $a_B=0.19\text{m}$.

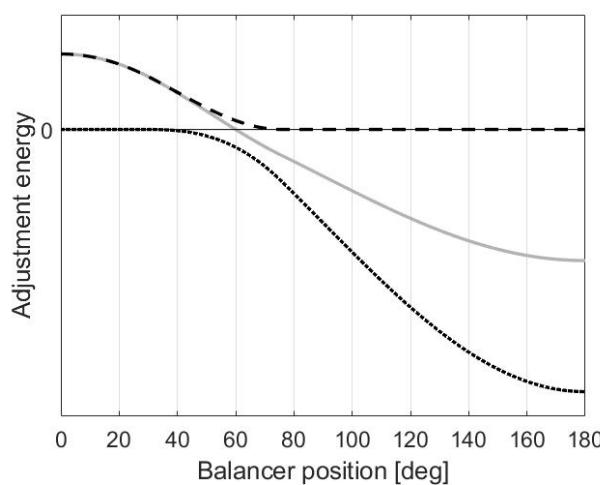


Figure G.3: Example of the AP-type adjustment from situation B to A. Net adjustment energy with an energy storage and supply efficiency of 50% ($\epsilon=0.5$) (continuous line), required adjustment energy (dashed line) and released adjustment energy (dotted line) as function of the range of motion. $T_{mass}^A=6\text{Nm}$, $T_{mass}^B=20\text{Nm}$, $k=500\text{N/m}$, $L_0=0.1\text{m}$, $F_0=10\text{N}$, $r=0.22\text{m}$, $a_A=0.054\text{m}$, $a_B=0.19\text{m}$.

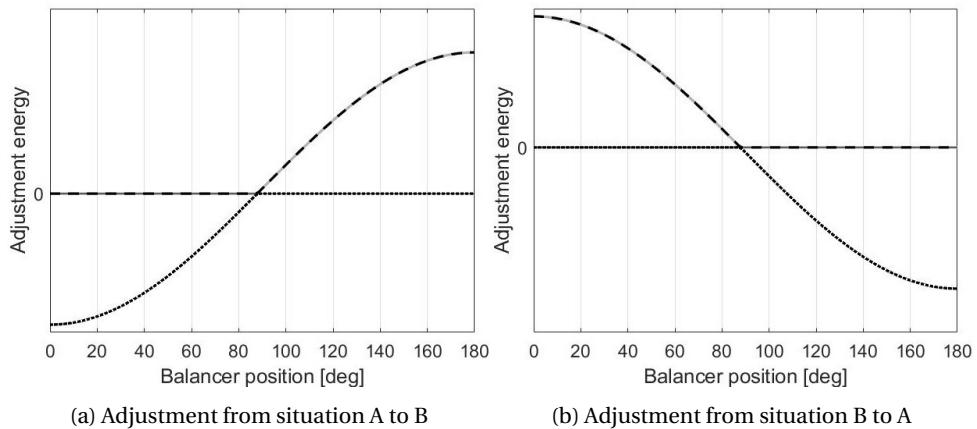


Figure G.4: Example of the AP-type adjustment. Minimum net adjustment energy with an energy storage and supply efficiency of 0% ($\epsilon=0$) (continuous line), required adjustment energy (dashed line) and released adjustment energy (dotted line) as function of the range of motion. $T_{mass}^A=6\text{Nm}$, $T_{mass}^B=20\text{Nm}$, $k=500\text{N/m}$, $L_0=0.1\text{m}$, $F_0=10\text{N}$, $r=0.22\text{m}$, $a_A=0.054\text{m}$, $a_B=0.19\text{m}$.

H

APPENDIX H: MATLAB CODE OF MODELS

MAIN SCRIPT TO COMPARE DIFFERENT ADJUSTMENT TYPES

```
%% Bas Wagemaker, September 2019
% Part of Master Thesis for Mechanical Engineering.
% This file calculates the performance of three
% adjustment types of a
% balancer (AP, ANP, FL-type). The performance regarding
% the degree of
% perfect balance, energy involved in adjustment and
% dimensions is
% compared.
% A spring is selected from a spring database (data from
% www.gutekunst.nl).

clc
clear all
close all

% Looping for all four comparison cases
for comparisonCase=1:4
    display(['Case: ' num2str(comparisonCase)])

    %% Selecting right properties for each case
    switch comparisonCase
        case 1
            TspringAmax=2; % [Nm] peak moment in
                            situation A
```

```

    TspringBmax=4; % [Nm] peak moment in
                     situation B
case 2
    TspringAmax=4; % [Nm] peak moment in
                     situation A
    TspringBmax=24; % [Nm] peak moment in
                     situation B
case 3
    TspringAmax=80; % [Nm] peak moment in
                     situation A
    TspringBmax=100; % [Nm] peak moment in
                     situation B
case 4
    TspringAmax=80; % [Nm] peak moment in
                     situation A
    TspringBmax=140; % [Nm] peak moment in
                     situation B
end

%% Properties of balancer
L=0.2; % [m] lever arm length
g=9.81; % [m/s^2] gravity acceleration
mA=TspringAmax/(g*L); % [kg] balanced mass situation
A
mB=TspringBmax/(g*L); % [kg] balanced mass situation
A
Phi=linspace(0,pi,100); % [rad] range of motion
SF=2; % [-] safety factor for energy in spring

%% Grid steps
Na=200; % number of a's
Nr=200; % number of r's
Nu0=200; % number of u0's

%% Loading spring data
T = readtable('spring_data.csv');
T = sortrows(T,19);
% Spring properties
mSpring=T{:,19}/1000; % spring mass
mSpring(2224)=0.0789/1000; % filtering out a mistake
                             in the data
kSpring=T{:,18}*1000; % spring stiffness
L0Spring=T{:,14}/1000; % spring null-length
dLmaxSpring=T{:,16}/1000; % maximum elongation spring
SerialNumber=T{:,1}; % serial number spring

```

```

diameterSpring=T{:,9}/1000; % diameter spring
UmaxSpring=1/2*kSpring.*dLmaxSpring.^2; % maximum
    storables energy spring
F0Spring=T{:,10}; % initial tensions spring

%% Determining the minimum required energy storage of
    the spring
UmassMax=2*mB*g*L; % maximum potential energy of mass
UspringLim=SF*UmassMax; % minimum required potential
    energy storage of spring

%% Filtering springs
% Filtering out the springs that cannot store
    sufficient energy
indices=find(UmaxSpring>UspringLim); % indices of
    suitable springs

mSpringFiltered=mSpring(indices); % spring mass
kSpringFiltered=kSpring(indices); % spring stiffness
L0SpringFiltered=L0Spring(indices); % spring null-
    length
dLmaxSpringFiltered=dLmaxSpring(indices); % maximum
    elongation of spring
SerialNumberFiltered=SerialNumber{indices}; % serial
    number spring
UmaxSpringFiltered=UmaxSpring(indices); % maximum
    storables energy spring
diameterSpringFiltered=diameterSpring(indices); %
    diameter spring
F0SpringFiltered=F0Spring(indices); % initial tension
    spring
% F0SpringFiltered=zeros(size(mSpringFiltered)); %
    initial tension spring

if F0SpringFiltered(1,1)==0
    display('The initial tension is set to zero')
elseif F0SpringFiltered(1,1)~=0
    display('The initial tension is taken into
        account')
end

%% Selecting a spring
[mspringMin, indexSpring]=min(mSpringFiltered); %
    selecting spring with smallest mass

```

```

% [L0spring, indexSpring]=min(L0SpringFiltered); %
    selecting spring with smallest null-length
% [mspring, indexSpring]=max(mSpringFiltered); %
    selecting spring with largest mass
% [k, indexSpring]=min(kSpringFiltered); % selecting
    spring with smallest stiffness
% [k, indexSpring]=max(kSpringFiltered); % selecting
    spring with largest stiffness
% [UspringMax,indexSpring]=max(UmaxSpringFiltered); %
    selecting spring with largest energy storage
%
indexSpring=indexSpring;

% Properties of selected spring
mspring=mSpringFiltered(indexSpring); % spring mass
k=kSpringFiltered(indexSpring); % spring stiffness
L0spring=L0SpringFiltered(indexSpring); % spring null
    length
UspringMax=UmaxSpringFiltered(indexSpring); % maximum
    storabile energy spring
F0spring=F0SpringFiltered(indexSpring); % initial
    tension spring

% Safety factor to assure that the spring is not
    loaded to its maximum
UspringMax=UspringMax/SF;

%% Calculating performance of balancer types

% AP-type
[performanceAPtype]=      APtype(mA, mB, g, L, Phi, k,
    L0spring, F0spring, UspringLim, Na, Nr);
% performance = [rVec, aAVec, aBVec, BQAAError,
    BQBABError, AEmetric, BQerrormetric, Compactmetric
    ]

% FL-type
[performanceFLtype]=      FLtype(mA, mB, g, L, Phi, k,
    L0spring, F0spring, UspringLim, Na, Nr, Nu0);
% performance = [rVec, aAVec, u0Vec, BQAAError,
    BQBABError, AEmetric, BQerrormetric, Compactmetric
    ]

% ANP-type
[performanceANPtype]=      ANPtype(mA, mB, g, L, Phi, k
    , L0spring, F0spring, UspringLim, Na, Nr);

```

```

% performance = [rVec, aAVec, aBVec, BQAAError,
    BQBABerror, AEmetric, BQerrormetric, Compactmetric
]

%% Calculating plot information
% Number of non-adjusted variables that are
    considered in the plots
Nplot=20;

% Balancing quality in situation A
BQeAmedian(1,comparisonCase)=median(performanceAPtype
    (1:Nplot, 4)); % AP
BQeAmedian(2,comparisonCase)=median(
    performanceANPtype(1:Nplot, 4)); % ANP
BQeAmedian(3,comparisonCase)=median(performanceFLtype
    (1:Nplot, 4)); % FL

BQeAmin(1,comparisonCase)=min(performanceAPtype(1:
    Nplot, 4)); % AP
BQeAmin(2,comparisonCase)=min(performanceANPtype(1:
    Nplot, 4)); % ANP
BQeAmin(3,comparisonCase)=min(performanceFLtype(1:
    Nplot, 4)); % FL

BQeAmax(1,comparisonCase)=max(performanceAPtype(1:
    Nplot, 4)); % AP
BQeAmax(2,comparisonCase)=max(performanceANPtype(1:
    Nplot, 4)); % ANP
BQeAmax(3,comparisonCase)=max(performanceFLtype(1:
    Nplot, 4)); % FL

% Balancing quality in situation B
BQeBmedian(1,comparisonCase)=median(performanceAPtype
    (1:Nplot, 5)); % AP
BQeBmedian(2,comparisonCase)=median(
    performanceANPtype(1:Nplot, 5)); % ANP
BQeBmedian(3,comparisonCase)=median(performanceFLtype
    (1:Nplot, 5)); % FL

BQeBmin(1,comparisonCase)=min(performanceAPtype(1:
    Nplot, 5)); % AP
BQeBmin(2,comparisonCase)=min(performanceANPtype(1:
    Nplot, 5)); % ANP
BQeBmin(3,comparisonCase)=min(performanceFLtype(1:
    Nplot, 5)); % FL

```

```
BQeBmax(1,comparisonCase)=max(performanceAPtype(1:  
    Nplot, 5)); % AP  
BQeBmax(2,comparisonCase)=max(performanceANPtype(1:  
    Nplot, 5)); % ANP  
BQeBmax(3,comparisonCase)=max(performanceFLtype(1:  
    Nplot, 5)); % FL  
  
BQeBmin(1,comparisonCase)=min(performanceAPtype(1:  
    Nplot, 5)); % AP  
BQeBmin(2,comparisonCase)=min(performanceANPtype(1:  
    Nplot, 5)); % ANP  
BQeBmin(3,comparisonCase)=min(performanceFLtype(1:  
    Nplot, 5)); % FL  
  
% Adjustment energy  
AEmedian(1,comparisonCase)=median(performanceAPtype  
    (1:Nplot, 6)); % AP  
AEmedian(2,comparisonCase)=median(performanceANPtype  
    (1:Nplot, 6)); % ANP  
AEmedian(3,comparisonCase)=median(performanceFLtype  
    (1:Nplot, 6)); % FL  
  
AEmin(1,comparisonCase)=min(performanceAPtype(1:Nplot  
    , 6)); % AP  
AEmin(2,comparisonCase)=min(performanceANPtype(1:  
    Nplot, 6)); % ANP  
AEmin(3,comparisonCase)=min(performanceFLtype(1:Nplot  
    , 6)); % FL  
  
AEmax(1,comparisonCase)=max(performanceAPtype(1:Nplot  
    , 6)); % AP  
AEmax(2,comparisonCase)=max(performanceANPtype(1:  
    Nplot, 6)); % ANP  
AEmax(3,comparisonCase)=max(performanceFLtype(1:Nplot  
    , 6)); % FL  
  
% Dimensions  
DMmedian(1,comparisonCase)=median(performanceAPtype  
    (1:Nplot, 8)); % AP  
DMmedian(2,comparisonCase)=median(performanceANPtype  
    (1:Nplot, 8)); % ANP  
DMmedian(3, comparisonCase)=median(performanceFLtype  
    (1:Nplot, 8)); % FL
```

```

DMmin(1,comparisonCase)=min(performanceAPtype(1:Nplot
    , 8)); % AP
DMmin(2,comparisonCase)=min(performanceANPtype(1:
    Nplot , 8)); % ANP
DMmin(3,comparisonCase)=min(performanceFLtype(1:Nplot
    , 8)); % FL

DMmax(1,comparisonCase)=max(performanceAPtype(1:Nplot
    , 8)); % AP
DMmax(2,comparisonCase)=max(performanceANPtype(1:
    Nplot , 8)); % ANP
DMmax(3,comparisonCase)=max(performanceFLtype(1:Nplot
    , 8)); % FL

display('Modelling finished')
end

%% Colors for plotting
blue=[0 0.4470 0.7410];
red=[0.8500 , 0.3250 , 0.0980];
orange=[0.9290 , 0.6940 , 0.1250];

%% Plotting BQeA for all cases
dataavg=BQeAmedian;
datamin=BQeAmin;
datamax=BQeAmax;
figure
for caseDummy=1:4
    % AP
    subplot(4,1,caseDummy)
    h(1)=plot(dataavg(1,caseDummy) , 1, 'x' , 'color' , red ,
        'Markersize' , 10 , 'Linewidth' , 3);
    hold on
    plot([datamin(1, caseDummy) , datamax(1, caseDummy)] ,
        [1 1] , 'color' , red , 'Linewidth' , 3)
    % ANP
    h(2)=plot(dataavg(2, caseDummy) , 2, 'x' , 'color' ,
        blue , 'Markersize' , 10 , 'Linewidth' , 3);
    plot([datamin(2, caseDummy) , datamax(2, caseDummy)] ,
        [2 2] , 'color' , blue , 'Linewidth' , 3)
    % FL
    h(3)=plot(dataavg(3, caseDummy) , 3, 'x' , 'color' ,
        orange , 'Markersize' , 10 , 'Linewidth' , 3);

```

```

plot([datamin(3, caseDummy), datamax(3, caseDummy)],
      [3 3], 'color', orange, 'Linewidth', 3);
% Plotting dummies for legend
d(1)=plot(NaN, NaN, '-','color', red,'Linewidth', 2);
d(2)=plot(NaN, NaN, '-','color', blue,'Linewidth',
           2);
d(3)=plot(NaN, NaN, '-','color', orange, 'Linewidth',
           2);
d(4)=plot(NaN, NaN, 'kx', 'Linewidth',2);
% Properties of plot
set(gca, 'YTickLabel', [])
ylabel(['Case ' num2str(caseDummy)])
xlim([0 17.5])
ylim([0.5 3.5])
grid on
set(gca, 'FontSize',12)

switch caseDummy
    case 1
        title({'Balancing quality error situation A'
               })
        legend(d, 'AP-type', 'ANP-type', 'FL-type', ' '
               median')
        set(gca, 'XTickLabel', [])
    case 2
        set(gca, 'XTickLabel', [])
    case 3
        set(gca, 'XTickLabel', [])
    case 4
        xlabel('BQe_A [%]')
end

end

%% Plotting BQeB for all cases
dataavg=BQeBmedian;
datamin=BQeBmin;
datamax=BQeBmax;
figure
for caseDummy=1:4
    % AP
    subplot(4,1,caseDummy)
    h(1)=plot(dataavg(1,caseDummy), 1, 'x', 'color', red,
               'Markersize', 10, 'Linewidth', 3);
    hold on

```

```

plot([datamin(1, caseDummy), datamax(1, caseDummy)],
      [1 1], 'color', red, 'Linewidth', 3)
% ANP
h(2)=plot(dataavg(2, caseDummy), 2, 'x', 'color',
           blue, 'Markersize', 10, 'Linewidth', 3);
plot([datamin(2, caseDummy), datamax(2, caseDummy)],
      [2 2], 'color', blue, 'Linewidth', 3);
% FL
h(3)=plot(dataavg(3, caseDummy), 3, 'x', 'color',
           orange, 'Markersize', 10, 'Linewidth', 3);
% Plotting dummies for legend
d(1)=plot(NaN, NaN, '--', 'color', red, 'Linewidth', 2);
d(2)=plot(NaN, NaN, '--', 'color', blue, 'Linewidth',
           2);
d(3)=plot(NaN, NaN, '--', 'color', orange, 'Linewidth',
           2);
d(4)=plot(NaN, NaN, 'kx', 'Linewidth', 2);
% Properties of plot
set(gca, 'YTickLabel', [])
ylabel(['Case ' num2str(caseDummy)])
xlim([0 45])
ylim([0.5 3.5])
grid on
set(gca, 'FontSize', 12)

switch caseDummy
    case 1
        title({'Balancing quality error situation B'
               })
        legend(d, 'AP-type', 'ANP-type', 'FL-type',
               'median')
        set(gca, 'XTickLabel', [])
    case 2
        set(gca, 'XTickLabel', [])
    case 3
        set(gca, 'XTickLabel', [])
    case 4
        xlabel('BQe_B [%]')
    end

end

%% Plotting AEm for all cases
dataavg=AEmedian;
datamin=AEmin;

```

```

datamax=AEmax;
figure
for caseDummy=1:4
    % AP
    subplot(4,1,caseDummy)
    h(1)=plot(dataavg(1, caseDummy), 1, 'x', 'color', red,
               'Markersize', 10, 'Linewidth', 3);
    hold on
    plot([datamin(1, caseDummy), datamax(1, caseDummy)],
          [1 1], 'color', red, 'Linewidth', 3)
    % ANP
    h(2)=plot(dataavg(2, caseDummy), 2, 'x', 'color',
               blue, 'Markersize', 10, 'Linewidth', 3);
    plot([datamin(2, caseDummy), datamax(2, caseDummy)],
          [2 2], 'color', blue, 'Linewidth', 3)
    % FL
    h(3)=plot(dataavg(3, caseDummy), 3, 'x', 'color',
               orange, 'Markersize', 10, 'Linewidth', 3);
    plot([datamin(3, caseDummy), datamax(3, caseDummy)],
          [3 3], 'color', orange, 'Linewidth', 3)
    % Plotting dummies for legend
    d(1)=plot(NaN, NaN, '--', 'color', red, 'Linewidth', 2);
    d(2)=plot(NaN, NaN, '--', 'color', blue, 'Linewidth',
               2);
    d(3)=plot(NaN, NaN, '--', 'color', orange, 'Linewidth',
               2);
    d(4)=plot(NaN, NaN, 'kx', 'Linewidth', 2);
    % Properties of plot
    set(gca, 'YTickLabel', [])
    ylabel(['Case ' num2str(caseDummy)])
    xlim([-15 700])
    ylim([0.5 3.5])
    grid on
    set(gca, 'FontSize', 12)

    switch caseDummy
        case 1
            title({'Adjustment energy metric'})
            legend(d, 'AP-type', 'ANP-type', 'FL-type',
                   'median')
            set(gca, 'XTickLabel', [])
        case 2
            set(gca, 'XTickLabel', [])
        case 3
            set(gca, 'XTickLabel', [])

```

```

    case 4
        xlabel('AE_m [%]')
    end

end

%% Plotting DMm for all cases
dataavg=DMmedian;
datamin=DMmin;
datamax=DMmax;
figure
for caseDummy=1:4
    % AP
    subplot(4,1,caseDummy)
    h(1)=plot(dataavg(1,caseDummy), 1, 'x', 'color', red,
               'Markersize', 10, 'Linewidth', 3);
    hold on
    plot([datamin(1, caseDummy), datamax(1, caseDummy)],
         [1 1], 'color', red, 'Linewidth', 3)
    % ANP
    h(2)=plot(dataavg(2, caseDummy), 2, 'x', 'color',
               blue, 'Markersize', 10, 'Linewidth', 3);
    plot([datamin(2, caseDummy), datamax(2, caseDummy)],
         [2 2], 'color', blue, 'Linewidth', 3)
    % FL
    h(3)=plot(dataavg(3, caseDummy), 3, 'x', 'color',
               orange, 'Markersize', 10, 'Linewidth', 3);
    plot([datamin(3, caseDummy), datamax(3, caseDummy)],
         [3 3], 'color', orange, 'Linewidth', 3)
    % Plotting dummies for legend
    d(1)=plot(NaN, NaN, '--', 'color', red,'Linewidth', 2);
    d(2)=plot(NaN, NaN, '--', 'color', blue,'Linewidth',
               2);
    d(3)=plot(NaN, NaN, '--', 'color', orange, 'Linewidth',
               2);
    d(4)=plot(NaN, NaN, 'kx', 'Linewidth',2);
    % Properties of plot
    set(gca, 'YTickLabel', [])
    ylabel(['Case ' num2str(caseDummy)])
    xlim([-100 80])
    ylim([0.5 3.5])
    grid on
    set(gca,'FontSize',12)

switch caseDummy

```

```

    case 1
        title({'Dimensions metric'})
        legend(d, 'AP-type', 'ANP-type', 'FL-type', 'median')
        set(gca, 'XTickLabel', [])
    case 2
        set(gca, 'XTickLabel', [])
    case 3
        set(gca, 'XTickLabel', [])
    case 4
        xlabel('DM_m [%]')
    end

end

```

MAIN SCRIPT TO DETERMINE SENSITIVITY OF ADJUSTMENT TYPES TO DIFFERENT SPRINGS

```

%% Bas Wagemaker, September 2019
% Part of Master Thesis for Mechanical Engineering.
% This file calculates the performance of three
% adjustment types of a
% balancer (AP, ANP, FL-type). This is done for different
% springs with
% different properties, to see whether a performance
% improvement can be
% obtained by selecting a different spring.
% A spring is selected from a spring database (data from
% www.gutekunst.nl).

% clc
clear all
close all

%% Inputs
% Comparison case that is considered
comparisonCase=3;
NdifferentSprings=10; % number of springs that are
compared

%% Selecting right properties for each case
switch comparisonCase
    case 1
        TspringAmax=2; % [Nm] peak moment in situation A
        TspringBmax=4; % [Nm] peak moment in situation B

```

```

case 2
    TspringAmax=4; % [Nm] peak moment in situation A
    TspringBmax=24; % [Nm] peak moment in situation B
case 3
    TspringAmax=80; % [Nm] peak moment in situation A
    TspringBmax=100; % [Nm] peak moment in situation
        B
case 4
    TspringAmax=80; % [Nm] peak moment in situation A
    TspringBmax=140; % [Nm] peak moment in situation
        B
end

%% Properties of balancer
L=0.2; % [m] lever arm length
g=9.81; % [m/s^2] gravity acceleration
mA=TspringAmax/(g*L); % [kg] balanced mass situation A
mB=TspringBmax/(g*L); % [kg] balanced mass situation A
Phi=linspace(0,pi,100); % [rad] range of motion
SF=2; % [-] safety factor for energy in spring

%% Grid steps
Na=200; % number of a's
Nr=200; % number of r's
Nu0=200; % number of u0's

%% Loading spring data
T = readtable('spring_data.csv');
T = sortrows(T,19);
% Spring properties
mSpring=T{:,19}/1000; % spring mass
mSpring(2224)=0.0789/1000; % filtering out a mistake in
    the data
kSpring=T{:,18}*1000; % spring stiffness
L0Spring=T{:,14}/1000; % spring null-length
dLmaxSpring=T{:,16}/1000; % maximum elongation spring
SerialNumber=T{:,1}; % serial number spring
diameterSpring=T{:,9}/1000; % diameter spring
UmaxSpring=1/2*kSpring.*dLmaxSpring.^2; % maximum
    storables energy spring
F0Spring=T{:,10}; % initial tensions spring

%% Determining the minimum required energy storage of the
    spring
UmassMax=2*mB*g*L; % maximum potential energy of mass

```

```

UspringLim=SF*UmassMax; % minimum required potential
    energy storage of spring

%% Filtering springs
% Filtering out the springs that cannot store sufficient
    energy
indices=find(UmaxSpring>UspringLim); % indices of
    suitable springs

mSpringFiltered=mSpring(indices); % spring mass
kSpringFiltered=kSpring(indices); % spring stiffness
L0SpringFiltered=L0Spring(indices); % spring null-length
dLmaxSpringFiltered=dLmaxSpring(indices); % maximum
    elongation of spring
SerialNumberFiltered=SerialNumber(indices); % serial
    number spring
UmaxSpringFiltered=UmaxSpring(indices); % maximum
    storables energy spring
diameterSpringFiltered=diameterSpring(indices); %
    diameter spring
F0SpringFiltered=F0Spring(indices); % initial tension
    spring
% F0SpringFiltered=zeros(size(mSpringFiltered)); %
    initial tension spring

if F0SpringFiltered(1,1)==0
    display('The initial tension is set to zero')
elseif F0SpringFiltered(1,1)~=0
    display('The initial tension is taken into account')
end

%% Selecting a spring
[mspring , indexSpring]=min(mSpringFiltered); % selecting
    spring with smallest mass
% [L0spring , indexSpring]=min(L0SpringFiltered); %
    selecting spring with smallest null-length
% [mspring , indexSpring]=max(mSpringFiltered); %
    selecting spring with largest mass
% [k , indexSpring]=min(kSpringFiltered); % selecting
    spring with smallest stiffness
% [k , indexSpring]=max(kSpringFiltered); % selecting
    spring with largest stiffness
% [UspringMax , indexSpring]=max(UmaxSpringFiltered); %
    selecting spring with largest energy storage
% indexSpring=indexSpring;

```

```

Nsprings=size(mSpringFiltered,1); % number of possible
    springs
isprings=round(linspace(1,NdifferentSprings,
    NdifferentSprings)); % indices of springs that are
    compared

for icase=1:NdifferentSprings
    indexSpring=isprings(icase);
    display(['Spring ' num2str(icase) ' out of ' num2str(
        NdifferentSprings)])

    % Properties of selected spring
    mspring=mSpringFiltered(indexSpring); % spring mass
    k=kSpringFiltered(indexSpring); % spring stiffness
    L0spring=L0SpringFiltered(indexSpring); % spring null
        length
    UspringMax=UmaxSpringFiltered(indexSpring); % maximum
        storable energy spring
    F0spring=F0SpringFiltered(indexSpring); % initial
        tension spring
    SerialNumber=SerialNumberFiltered(indexSpring); %
        serial number spring

    % Safety factor to assure that the spring is not
        loaded to its maximum
    UspringMax=UspringMax/SF;

    springSelection(icase,:)=[mspring k L0spring F0spring
        SerialNumber];

%% Calculating performance of balancer types

%% AP-type
[performanceAPtype]= APtype(mA, mB, g, L, Phi, k,
    L0spring, F0spring, UspringLim, Na, Nr);
% [rVec, aAVec, aBVec, BQAAError, BQBABerror,
    AEmetric, BQerrormetric, Compactmetric]

%% FL-type
[performanceFLtype]= FLtype(mA, mB, g, L, Phi, k,
    L0spring, F0spring, UspringLim, Na, Nr, Nu0);
% [rVec, aAVec, aBVec, BQAAError, BQBABerror,
    AEmetric, BQerrormetric, Compactmetric]

```

```

%% ANP-type
[performanceANPtype]= ANPtype(mA, mB, g, L, Phi, k
    , L0spring, F0spring, UspringLim, Na, Nr);
% [rVec, aAVec, aBVec, BQAABerror, BQBABerror,
AEmetric, BQerrormetric, Compactmetric]

%% Comparing different springs
Ncompare=1; % number of the best scoring balancer
configurations on the cumulative BQe that are
averaged in the next step

%% Metrics of Ncompare best performers
% Balancing quality error in situation A
BQeAavg(icase, 1)=sum(performanceAPtype(1:Ncompare,4)
)/Ncompare;
BQeAavg(icase, 2)=sum(performanceANPtype(1:Ncompare
,4))/Ncompare;
BQeAavg(icase, 3)=sum(performanceFLtype(1:Ncompare,4)
)/Ncompare;
% Balancing quality error in situation B
BQeBavg(icase, 1)=sum(performanceAPtype(1:Ncompare,5)
)/Ncompare;
BQeBavg(icase, 3)=sum(performanceFLtype(1:Ncompare,5)
)/Ncompare;
BQeBavg(icase, 2)=sum(performanceANPtype(1:Ncompare
,5))/Ncompare;
% Adjustment energy metric
AEavg(icase, 1)=sum(performanceAPtype(1:Ncompare,6))/(
Ncompare;
AEavg(icase, 3)=sum(performanceFLtype(1:Ncompare,6))/(
Ncompare;
AEavg(icase, 2)=sum(performanceANPtype(1:Ncompare,6))/(
Ncompare;
% Compact metric
CTavg(icase, 1)=sum(performanceAPtype(1:Ncompare,8))/(
Ncompare;
CTavg(icase, 3)=sum(performanceFLtype(1:Ncompare,8))/(
Ncompare;
CTavg(icase, 2)=sum(performanceANPtype(1:Ncompare,8))/(
Ncompare;

end

%% Releative difference

```

```

BQeArel(:,1)=(BQeAavg(:,1)-BQeAavg(1,1))./BQeAavg(1,1)
    *100;
BQeArel(:,2)=(BQeAavg(:,2)-BQeAavg(1,2))./BQeAavg(1,2)
    *100;
BQeArel(:,3)=(BQeAavg(:,3)-BQeAavg(1,3))./BQeAavg(1,3)
    *100;
BQeAmin=min(BQeArel,[],1);

BQeBrel(:,1)=(BQeBavg(:,1)-BQeBavg(1,1))./BQeBavg(1,1)
    *100;
BQeBrel(:,2)=(BQeBavg(:,2)-BQeBavg(1,2))./BQeBavg(1,2)
    *100;
BQeBrel(:,3)=(BQeBavg(:,3)-BQeBavg(1,3))./BQeBavg(1,3)
    *100;
BQeBmin=min(BQeBrel,[],1);

AErel(:,1)=(AEavg(:,1)-AEavg(1,1))./AEavg(1,1)*100;
AErel(:,2)=(AEavg(:,2)-AEavg(1,2))./AEavg(1,2)*100;
AErel(:,3)=(AEavg(:,3)-AEavg(1,3))./AEavg(1,3)*100;
AEmin=min(AErel,[],1);

CTrel(:,1)=(CTavg(:,1)-CTavg(1,1))./CTavg(1,1)*100;
CTrel(:,2)=(CTavg(:,2)-CTavg(1,2))./CTavg(1,2)*100;
CTrel(:,3)=(CTavg(:,3)-CTavg(1,3))./CTavg(1,3)*100;
CTmin=min(CTrel,[],1);

%% Evaluation of metrics
% Standard deviation
stdBQeAavg=nanstd(BQeAavg);
stdBQeBavg=nanstd(BQeBavg);
stdAEavg=nanstd(AEavg);
stdCTavg=nanstd(CTavg);

% Mean absolute deviation
meanBQeAavg=nanmean(BQeAavg);
meanBQeBavg=nanmean(BQeBavg);
meanAEavg=nanmean(AEavg);
meanCTavg=nanmean(CTavg);

% Mean absolute deviation
madBQeAavg=mad(BQeAavg);
madBQeBavg=mad(BQeBavg);
madAEavg=mad(AEavg);
madCTavg=mad(CTavg);

```

```

% Coefficient of variation
covBQeA=stdBQeAavg./meanBQeAavg*100;
covBQeB=stdBQeBavg./meanBQeBavg*100;
covAE=stdAEavg./meanAEavg*100;
covCT=stdCTavg./meanCTavg*100;

% Coefficient of variation
covModBQeA=madBQeAavg./meanBQeAavg*100;
covModBQeB=madBQeBavg./meanBQeBavg*100;
covModAE=madAEavg./meanAEavg*100;
covModCT=madCTavg./meanCTavg*100;

display('Modelling finished')

%% Tables
% types={'AP', 'ANP', 'FL'};
% varNames={'Type', 'BQeA', 'BQeB', 'AE', 'CT'};
% Tstd=table(types', stdBQeAavg', stdBQeBavg', stdAEavg',
%             stdCTavg', 'VariableNames', varNames)
%
% types={'AP', 'ANP', 'FL'};
% varNames={'Type', 'BQeA', 'BQeB', 'AE', 'CT'};
% Tcov=table(types', covBQeA', covBQeB', covAE', covCT',
%             'VariableNames', varNames)
%
% types={'AP', 'ANP', 'FL'};
% varNames={'Type', 'BQeA', 'BQeB', 'AE', 'CT'};
% TcovMod=table(types', covModBQeA', covModBQeB',
%                 covModAE', covModCT', 'VariableNames', varNames)
%
% types={'AP', 'ANP', 'FL'};
% varNames={'Type', 'BQeAmean', 'BQeAstd', 'BQeBmean',
%             'BQeBstd', 'AEmean', 'AEstd', 'CTmean', 'CTstd'};
% TcovMod=table(types', meanBQeAavg', stdBQeAavg',
%                 meanBQeBavg', stdBQeBavg', meanAEavg', stdAEavg',
%                 meanCTavg', stdCTavg', 'VariableNames', varNames)

types={'AP', 'ANP', 'FL'};
varNames={'Type', 'BQeA', 'BQeB', 'AE', 'DM'};
Treldif=table(types', BQeAmin', BQeBmin', AEmin', CTmin',
              'VariableNames', varNames)

%% Plotting
figure
plot(BQeAavg, '.', 'Markersize', 20)

```

```

grid on
title({'BQeA'; ['Case ' num2str(comparisonCase) ', 
    Torque_A=' num2str(TspringAmax) 'Nm, Torque_B=' 
    num2str(TspringBmax) 'Nm']})
legend('AP-type', 'ANP-type', 'FL-type')

figure
plot(BQeBavg, '.', 'Markersize', 20)
grid on
title({'BQeB'; ['Case ' num2str(comparisonCase) ', 
    Torque_A=' num2str(TspringAmax) 'Nm, Torque_B=' 
    num2str(TspringBmax) 'Nm']})
legend('AP-type', 'ANP-type', 'FL-type')

figure
plot(AEavg, '.', 'Markersize', 20)
grid on
title({'AE'; ['Case ' num2str(comparisonCase) ', Torque_A 
    =' num2str(TspringAmax) 'Nm, Torque_B=' num2str( 
    TspringBmax) 'Nm']})
legend('AP-type', 'ANP-type', 'FL-type')

figure
plot(CTavg, '.', 'Markersize', 20)
grid on
title({'DM'; ['Case ' num2str(comparisonCase) ', Torque_A 
    =' num2str(TspringAmax) 'Nm, Torque_B=' num2str( 
    TspringBmax) 'Nm']})
legend('AP-type', 'ANP-type', 'FL-type')

```

H

MODEL OF EACH ADJUSTMENT TYPE

AP-TYPE

```

function [solABsorted]=APtype(mA, mB, g, L, Phi, k,
    L0spring, F0spring, UspringMax, Na, Nr)

%% Potential energy and torque of mass
UMassA=mA*g*L*cos(Phi); % potential energy mass A
UMassB=mB*g*L*cos(Phi); % potential energy mass B
TMassA=gradient(UMassA, Phi); % torque mass A
TMassB=gradient(UMassB, Phi); % torque mass B

%% Maximum value of minimum adjustment energy
AEmin=(mB-mA)*g*L;

```

```

%% Minimum value of area between r, a and spring
Areamin=1/2*(mB*g*L)/k;

%% Constructing ar-grid
% ar grid is bounded by the fact that a spring can store
% a finite amount of
% energy, depending on its stiffness and null-length
arbound=sqrt(2*UspringMax/k); % boundary of ar grid !!!
aVec=linspace(0,arbound,Na);
rVec=linspace(0,arbound,Nr);

%% Calculating balancing quality error
% For each pair or (a,r) the BQerror is calculated
BQAerror=zeros(Nr, Na);
BQBerror=zeros(Nr, Na);
for jr=1:Nr
    r=rVec(jr);
    for ja=1:Na
        a=aVec(ja);
        distanceAR=sqrt(a^2+r^2-2*a*r*cos(Phi));
        dLSpring=distanceAR;
        USpring=1/2*k*(dLSpring).^2+F0spring*(dLSpring);
        TSpring=gradient(USpring,Phi);
        BQA=-TSpring./TMassA*100;
        BQB=-TSpring./TMassB*100;
        BQAerror(jr,ja)=sqrt(sum(abs(BQA-100).^2)/length(
            Phi));
        BQBerror(jr,ja)=sqrt(sum(abs(BQB-100).^2)/length(
            Phi));
    end
end

%% Filtering solutions that lay outside the boundaries of
the ar grid
[R,A]=meshgrid(rVec,aVec); % meshgrid of a and r
% The maximum elongation of the spring is equal to a+r,
which cannot be
% larger than the boundary calculated earlier
z=R+A; % maximum spring elongation
BQAerror(z>arbound)=NaN; % setting all values outside the
ar grid equal to NaN
BQBerror(z>arbound)=NaN; % setting all values outside the
ar grid equal to NaN

```

```

%% Finding the a for each r for which the BQA error is
    minimal in situation A
% For one r, multiple a's can exist for which the BQ
    requirement is met,
% therefore, the a with the smallest BQAerror is selected
[BQAerrorMinRow, iaA]=min(BQAerror, [],2);
solA=[rVec' aVec(iaA)' BQAerrorMinRow];
solA=sortrows(solA,1); % sort the solutions in order of
    increasing r
% solA=solA(rVec./aVec(iaA)>1,:); % delete symmetric
    solutions
NsolA=size(solA,1); % number of solutions for situation A
aAVec=solA(:,2); % a's in situation A
BQAerrorOptA=solA(:,3); % BQAerror in situation A

%% Finding the a for each r for which the BQA error is
    minimal in situation B
% For one r, multiple a's can exist for which the BQ
    requirement is met,
% therefore, the a with the smallest BQAerror is selected
[BQBerrorMinRow, iab]=min(BQBerror, [],2);
solB=[rVec' aVec(iab)' BQBerrorMinRow];
solB=sortrows(solB,1); % sort the solutions in order of
    increasing r
% solB=solB(rVec./aVec(iab)>1,:); % delete symmetric
    solutions
NsolB=size(solB,1); % number of solutions for situation A
aBVec=solB(:,2); % a's in situation A
BQBerrorOptB=solB(:,3); % BQAerror in situation A

%% Adjustment energy
UAdjust=zeros(length(Phi), Nr);
UAdjustmax=zeros(Nr,1);
for jsolAB=1:Nr
    r=rVec(jsolAB);
    aA=aAVec(jsolAB);
    aB=aBVec(jsolAB);
    distanceARA=sqrt(aA^2+r^2-2*aA*r*cos(Phi));
    distanceARB=sqrt(aB^2+r^2-2*aB*r*cos(Phi));
    dLSpringA=distanceARA;
    dLSpringB=distanceARB;
    USpringA=1/2*k*dLSpringA.^2+F0spring*(dLSpringA);
    USpringB=1/2*k*dLSpringB.^2+F0spring*(dLSpringB);
    UAdjust(:,jsolAB)=USpringB-USpringA;

```

```

UAdjustmax(jsolAB,1)=max(UAdjust(:,jsolAB)); %
    maximum adjustment energy over the range of motion
end

%% Area between r, a and spring
Area=1/2.*rVec' .* aBVec;

%% Evaluation of metrics
AEmetric=(UAdjustmax-AEmin)./AEmin*100; % ratio of
    minimum maximum adjustment energy and actual maximum
    adjustment energy
Compactmetric=(Area-Areamin)./Areamin*100;
BQerrormetric=BQAerrorOptA+BQBerrorOptB;

solAB=[rVec' aAVec aBVec BQAerrorOptA BQBerrorOptB
    AEmetric BQerrormetric Compactmetric];
solABsorted=sortrows(solAB,7,'ascend');
NsolAB=size(solAB,1); % number of solutions for situation
    A

%% Filtering out BQerrors larger than the mass difference
massDifference=abs((mA-mB)/mB*100);
solABsorted(solABsorted(:,4)>massDifference/2,:)=[];
solABsorted(solABsorted(:,5)>massDifference/2,:)=[];

end

```

H

ANP-TYPE

```

function [solABsorted]=ANPtype(mA, mB, g, L, Phi, k,
    L0spring, F0spring, UspringMax, Na, Nr)

%% Potential energy and torque of mass
UMassA=mA*g*L*cos(Phi); % potential energy mass A
UMassB=mB*g*L*cos(Phi); % potential energy mass B
TMassA=gradient(UMassA, Phi); % torque mass A
TMassB=gradient(UMassB, Phi); % torque mass B

%% Maximum value of minimum adjustment energy
AEmin=(mB-mA)*g*L;

%% Minimum value of area between r, a and spring
Areamin=1/2*(mB*g*L)/k;

%% Constructing ar-grid

```

```
% ar grid is bounded by the fact that a spring can store
    a finite amount of
% energy, depending on its stiffness and null-length
arbound=sqrt(2*UspringMax/k)+L0spring; % boundary of ar
    grid
aVec=linspace(0.01,arbound,Na);
rVec=linspace(0.01,arbound,Nr);

%% Calculating balancing quality error
% For each pair or (a,r) the BQerror is calculated
BQAerror=zeros(Nr, Na);
BQBerror=zeros(Nr, Na);
for jr=1:Nr
    r=rVec(jr);
    for ja=1:Na
        a=aVec(ja);
        distanceAR=sqrt(a^2+r^2-2*a*r*cos(Phi));
        dLSpring=distanceAR-L0spring;
        USpring=1/2*k*(dLSpring).^2+F0spring*(dLSpring);
        TSpring=gradient(USpring,Phi);
        BQA=-TSpring./TMassA*100;
        BQB=-TSpring./TMassB*100;
        BQAerror(jr,ja)=sqrt(sum(abs(BQA-100).^2)/length(
            Phi));
        BQBerror(jr,ja)=sqrt(sum(abs(BQB-100).^2)/length(
            Phi));
    end
end

%% Filtering solutions that lay outside the boundaries of
    the ar grid
[R,A]=meshgrid(rVec,aVec); % meshgrid of a and r
% The maximum elongation of the spring is equal to a+r,
    which cannot be
% larger than the boundary calculated earlier
z=R+A; % maximum spring elongation
BQAerror(z>arbound)=NaN; % setting all values outside the
    ar grid equal to NaN
BQBerror(z>arbound)=NaN; % setting all values outside the
    ar grid equal to NaN

%% Finding the a for each r for which the BQA error is
    minimal in situation A
% For one r, multiple a's can exist for which the BQ
    requirement is met,
```

```

% therefore, the a with the smallest BQAerror is selected
[BQAerrorMinRow, iaA]=min(BQAerror,[],2);
solA=[rVec' aVec(iaA)' BQAerrorMinRow];
solA=sortrows(solA,1); % sort the solutions in order of
    increasing r
% solA=solA(rVec./aVec(iaA)>1,:); % delete symmetric
    solutions
NsolA=size(solA,1); % number of solutions for situation A
aAVec=solA(:,2); % a's in situation A
BQAerrorOptA=solA(:,3); % BQAerror in situation A

%% Finding the a for each r for which the BQA error is
    minimal in situation B
% For one r, multiple a's can exist for which the BQ
    requirement is met,
% therefore, the a with the smallest BQAerror is selected
[BQBerrorMinRow, iaB]=min(BQBerror,[],2);
solB=[rVec' aVec(iaB)' BQBerrorMinRow];
solB=sortrows(solB,1); % sort the solutions in order of
    increasing r
NsolB=size(solB,1); % number of solutions for situation A
aBVec=solB(:,2); % a's in situation A
BQBerrorOptB=solB(:,3); % BQAerror in situation A

%% Adjustment energy
UAdjust=zeros(length(Phi), Nr);
UAdjustmax=zeros(Nr,1);
for jsolAB=1:Nr
    r=rVec(jsolAB);
    aA=aAVec(jsolAB);
    aB=aBVec(jsolAB);
    distanceARA=sqrt(aA^2+r^2-2*aA*r*cos(Phi));
    distanceARB=sqrt(aB^2+r^2-2*aB*r*cos(Phi));
    dLSpringA=distanceARA-L0spring;
    dLSpringB=distanceARB-L0spring;
    USpringA=1/2*k*dLSpringA.^2+F0spring*(dLSpringA);
    USpringB=1/2*k*dLSpringB.^2+F0spring*(dLSpringB);
    UAdjust(:,jsolAB)=USpringB-USpringA;
    UAdjustmax(jsolAB,1)=max(UAdjust(:,jsolAB)); %
        maximum adjustment energy over the range of motion
end

%% Area between r, a and spring
Area=1/2.*rVec' .* aBVec;

```

```

%% Evaluation of metrics
AEmetric=(UAdjustmax-AEmin)./AEmin*100; % ratio of
    minimum maximum adjustment energy and actual maximum
    adjustment energy
Compactmetric=(Area-Areamin)./Areamin*100;
BQerrormetric=BQAerrorOptA+BQBerrorOptB;

solAB=[rVec' aAVec aBVec BQAerrorOptA BQBerrorOptB
    AEmetric BQerrormetric Compactmetric];
solABsorted=sortrows(solAB,7,'ascend');
NsolAB=size(solABsorted,1); % number of solutions for
    situation A

%% Filtering out BQerrors larger than the mass difference
massDifference=abs((mA-mB)/mB*100);
solABsorted(solABsorted(:,4)>massDifference/2,:)=[];
solABsorted(solABsorted(:,5)>massDifference/2,:)=[];

end

```

FL-TYPE

```

function [solABsorted]=FLtype(mA, mB, g, L, Phi, k,
    LOspring, FOspring, UspringMax, Na, Nr, Nu0)

%% Potential energy and torque of mass
UMassA=mA*g*L*cos(Phi); % potential energy mass A
UMassB=mB*g*L*cos(Phi); % potential energy mass B
TMassA=gradient(UMassA, Phi); % torque mass A
TMassB=gradient(UMassB, Phi); % torque mass B

%% Maximum value of minimum adjustment energy
AEmin=(mB-mA)*g*L;

%% Minimum value of area between r, a and spring
Areamin=1/2*(mB*g*L)/k;

%% SITUATION A

%% Constructing boundary for ar-grid in situation A
% ar grid is bounded by the fact that a spring can store
    a finite amount of
% energy, depending on its stiffness and null-length
arbound=sqrt(2*UspringMax/k); % boundary of ar grid
aVec=linspace(0,arbound,Na);

```

H

```

rVec=linspace(0,arbound,Nr);

%% Compute the (a,r)'s that fullfill the BQ requirement
% in situation A
BQAerror=zeros(Nr, Na);
for jr=1:Nr
    r=rVec(jr);
    for ja=1:Na
        a=aVec(ja);
        distanceAR=sqrt(a^2+r^2-2*a*r*cos(Phi));
        dLSpring=distanceAR;
        USpring=1/2*k*(dLSpring).^2+F0spring*dLSpring;
        TSpring=gradient(USpring,Phi);
        BQA=-TSpring./TMassA*100;
        BQAerror(jr,ja)=sqrt(sum(abs(BQA-100).^2)/length(
            Phi));
    end
end

%% Filtering solutions that lay outside the boundaries of
% the ar grid
[R,A]=meshgrid(rVec,aVec); % meshgrid of a and r
% The maximum elongation of the spring is equal to a+r,
% which cannot be
% larger than the boundary calculated earlier
z=R+A; % maximum spring elongation
BQAerror(z>arbound)=NaN; % setting all values outside the
% ar grid equal to NaN

%% Filtering the soltuions that do not meet the desired
% balancing quality
% For one r, multiple a's can exist for which the BQ
% requirement is met,
% therefore, the a with the smallest BQAerror is selected
[BQAerrorMinRow, iaA]=min(BQAerror, [],2);
solA=[rVec' aVec(iaA)' BQAerrorMinRow];
solA=sortrows(solA,1); % sort the solutions in order of
% increasing r
% solA=solA(rVec./aVec(iaA)>=1,:); % delete symmetric
% solutions
NsolA=size(solA,1); % number of solutions for situation A
rAVec=solA(:,1);
aAVec=solA(:,2); % a's in situation A
BQAerrorOptA=solA(:,3); % BQAerror in situation A

```

```

%% SITUATION B

%% Calculating the maximum allowable u0 for each (a,r)
u0maxVec=sqrt(2*UspringMax/k)-aAVec-rAVec;

%% Checking whether the total energy stored in the spring
%% does not exceed the maximum spring storage
UspringLimcheck=1/2*k*(rAVec+aAVec+u0maxVec).^2;
margin=0.001;

if UspringLimcheck>UspringMax+margin
    error('The energy stored in the spring exceeds the
          maximum storable energy')
end

%% Calculating BQBerror for each (a,r) pair and each u0
%% within the u0max limit
u0Vec=zeros(NsolA, Nu0);
BQBerror=zeros(NsolA, Nu0);
for jsolA=1:NsolA
    r=rAVec(jsolA);
    a=aAVec(jsolA);
    u0max=u0maxVec(jsolA);
    u0Vec(jsolA,:)=linspace(0, u0max, Nu0);
    for ju0=1:Nu0
        distanceAR=sqrt(a.^2+r.^2-2*a.*r.*cos(Phi));
        dLSpring=distanceAR+u0Vec(jsolA, ju0);
        USpring=1/2*k*dLSpring.^2+F0spring*dLSpring;
        TSpring=gradient(USpring,Phi);
        BQB=-TSpring./TMassB*100;
        BQBerror(jsolA,ju0)=sqrt(sum(abs(BQB-100).^2)/
            length(Phi));
    end
end

%% Filtering the solutions that do not meet the desired
%% balancing quality
[BQBerrorMinRow, iu0B]=min(BQBerror,[],2);
% BQBerrorRowMin: minimum BQBerror per row
% iu0B: index of column of BQBerror in which minimum
% value is located
iu0BLinear=sub2ind(size(u0Vec),[1:NsolA]', iu0B); %
% linear index of u0Vec for which the minimum BQBerror
% is below the limit

```

```

u0BVec=u0Vec(iu0BLinear); % u0's for which BQAerror and
    % BQB error are below the set limit
solB=[rAVec u0BVec BQBerrorMinRow];
NsolB=size(solB,1); % number of solutions for situation A
rBVec=solB(:,1);
u0BVec=solB(:,2); % a's in situation A
BQBerrorOptB=solB(:,3); % BQAerror in situation A

%% Adjustment energy
UAdjust=zeros(length(Phi), NsolA);
UAdjustmax=zeros(NsolA,1);
for jSolAB=1:NsolA
    r=rAVec(jSolAB);
    a=aAVec(jSolAB);
    u0=u0BVec(jSolAB);
    distanceARA=sqrt(a^2+r^2-2*a*r*cos(Phi));
    distanceARB=sqrt(a^2+r^2-2*a*r*cos(Phi));
    dLSpringA=distanceARA;
    dLSpringB=distanceARB+u0;
    USpringA=1/2*k*(dLSpringA).^2+F0spring*dLSpringA;
    USpringB=1/2*k*(dLSpringB).^2+F0spring*dLSpringB;
    UAdjust(:,jSolAB)=USpringB-USpringA;
    UAdjustmax(jSolAB,1)=max(UAdjust(:,jSolAB)); %
        maximum adjustment energy over the range of motion
end

%% Area between r, a and spring
Area=1/2.*rVec' .* aAVec;

%% Evaluation of metrics
AEmetric=(UAdjustmax-AEmin)./AEmin*100; % ratio of
    minimum maximum adjustment energy and actual maximum
    adjustment energy
Compactmetric=(Area-Areamin)./Areamin*100;
BQerrormetric=BQAerrorOptA+BQBerrorOptB;

solAB=[rAVec aAVec u0BVec BQAerrorOptA BQBerrorOptB
    AEmetric BQerrormetric Compactmetric]; % Pairs of (r,a)
    % for which the BQAerror is smaller than the set limit
solABsorted=sortrows(solAB,7,'ascend');

%% Filtering out BQerrors larger than the mass difference
massDifference=abs((mA-mB)/mB*100);
solABsorted(solABsorted(:,4)>massDifference/2,:)=[];
solABsorted(solABsorted(:,5)>massDifference/2,:)=[];

```

end

H