Towards a Concurrent Multidisciplinary Design Optimization of Mechatronic Systems

Fariba Rahimi

Doctoral thesis in Machine Design
KTH Royal Institute of Technology
Stockholm, Sweden, 2021

TRITA-ITM-AVL 2021:11
ISBN 978-91-7873-815-1

Towards a Concurrent Multidisciplinary Design Optimization of Mechatronic Systems
Fariba Rahimi
Doctoral thesis

Academic Dissertation which, with due permission of the KTH Royal Institute of Technology, is submitted for public defence for the Degree of Doctor of Philosophy on Thursday September 2nd 2021, at 1:00pm in a Zoom Webinar.

Abstract

Multidisciplinary design optimization of mechatronic systems is a cumbersome task that considers integration of several engineering domains simultaneously. Hence, a holistic method should treat these engineering domains concurrently in the development phase and result in a solution for the system that is optimum considering several disciplines. Conventional design approaches for multi-disciplinary systems often consider each domain separately and integrate them at the end stage of the design. In these methods, the interactions and couplings between parameters from different disciplines can get lost and if any error arises at later design phase, it might lead to back-tracking and debugging, and hence, be time and cost consuming.

Therefore, a method that includes concurrent design of mechanics, electronics, control, considering the impact of embedded control implementation (on physical design and performance) which results in an integrated system is of noteworthy importance. This dissertation summarizes research by the author regarding ideas and suggestions for an integrated multi-criteria design method. The purpose of this research is to enable an early-phase design that takes into account three domains (*physical design*, *control design*, and *embedded control implementation*) simultaneously. Therefore objectives, specifications and constraints from each domain are taken into account. The efficiency feature is enhanced by the use of an early-phase design method which reduces time and cost consuming debugging, and removes the necessity to have iterative design loops in later design phases.

The method develops two types of components: *physical* and *control*. Physical components are basic mechanical/electrical elements which include three types of models: *physical dimension*, *static properties*, and *dynamic behaviour* models. Control components include *control methods* and *dynamic performance constraints*. The concept of a mechatronic system under design in the supporting software toolbox is configured using the components library. A multi-criteria optimization method is employed in a system level which yields an optimal solution for the system in terms of size, implementation cost, hardware cost and control performance. Using this system level optimization, there is no need to partition the problem or to integrate

several optimization loops in the method.

Four design cases are used to enable some features of the software toolbox and investigate capability of the method to handle multi-DOF nonlinear systems; and to highlight correlation between engineering domains and broaden the coverage of disciplines. The feasibility of the method is evaluated by variations of design tests for the design cases. Accordingly, mechanical and control components are studied, developed and integrated into the IDIOM (Integrated Design Optimization of Mechatronic Systems) software toolbox. Since the model of each component is treated separately in the design and modeling stage, any system configuration that uses the available components can be handled by the method. The contribution of the thesis can be summarized as follows:

- Multidisciplinary design method and investigation of couplings and correlations between engineering domains
- Models and co-design methods to include nonlinear multi-degree of freedom mechatronic systems
- Extended method to cover key aspects in discrete time systems, and key factors in embedded control implementation

The goal of this thesis is to improve system development efficiency by integrating engineering domains in an early design phase. Accordingly, the method in this thesis is a fundamental move in evaluation of mechatronic systems design which assists in better system development and analysis. However, there is no *single 'best'* approach for the design of mechatronic systems; the presented method in this thesis facilitates an *efficient* simultaneous integrated design optimization and has a broader coverage of engineering domains. The results achieved by the method ensure an *optimum system solution* in regards to the different involved engineering domains.

Acknowledgements

I would like to express my gratitude towards Jan Wikander, Malin Selleby and Martin Edin Grimheden for their insightful advice. My sincere thanks also goes to Stefan Björklund, Bengt Eriksson, Martin Törngren and Carl During. I wish to express my appreciation to Daniel Frede for our enjoyable technical and non-technical discussions. I would like to offer my special thanks to all colleagues at the division of mechatronics at machine design department. I would like to thank Björn Finer for providing IT-related support. I would like to pay my regards to Vicki Derbyshire for proofreading of my thesis and providing valuable feedback. I wish to extend special appreciation to the ITM school staff. A heartfelt thanks to all wonderful KTH members and friends.

Finally, I would like to express my sincere and deepest gratitude to my family for their unyielding support, unparalleled love, never-ending care and encouragement. Mum, Dad, Zhale, Mehdi, Sevil, Nima, Pinar and Taha, love you all!

Fariba Rahimi
Stockholm, March 2021

List of appended papers

Paper A

Fariba Rahimi, "Multi-Criteria Co-Design Optimization of Mechatronic Systems," 2020 IEEE International Conference on Mechatronics and Automation (ICMA), October 2020.

This paper is awarded the best paper of the IEEE ICMA 2020 Conference and is indexed by EI Compendex and CFP. An extension of this paper is published in Int. Journal of Mechatronics and Automation (IJMA) (Paper B).

Paper B

Fariba Rahimi, "Holistic Multidisciplinary Method for Optimization of Mechatronic Systems," Int. Journal of Mechatronics and Automation (IJMA), vol.8(2), 2021.

This paper is an extended version of Paper A and overlaps in the case study and some test results. The paper is extended by developing a design method for the belt-drive component and applying a state feedback observer control method on the system, and performing the analysis and comparisons by conducting design tests.

Paper C

Fariba Rahimi, Jan Wikander, "Concurrent and Optimal Structure, Control and Implementation Design," 2020 IEEE 11th International Conference on Mechanical and Intelligent Manufacturing Technologies (ICMIMT), Cape Town, South Africa, January 2020.

This paper is indexed in the Web of Science Core Collection.

Paper D

Fariba Rahimi, Jan Wikander, "Towards a Formal Framework for Integrated Design-Optimization and Control of Mechatronic Systems," Accepted to be published in Science Progress: SAGE Journals.

Paper E

Fariba Rahimi, Lei Feng, Jan Wikander, Daniel Frede, "Early Phase Design-

Optimization of Mechatronic Systems," in Proceedings of the 2017 The 5th International Conference on Control, Mechatronics and Automation, Edmontron, Canada, Pages 42–49, October 2017.

Division of work between authors

Paper A

Fariba Rahimi, "Multi-Criteria Co-Design Optimization of Mechatronic Systems," 2020 IEEE International Conference on Mechatronics and Automation (ICMA), October 2020.

Fariba Rahimi performed the research and wrote the paper.

Paper B

Fariba Rahimi, "Holistic Multidisciplinary Method for Optimization of Mechatronic Systems," Int. Journal of Mechatronics and Automation (IJMA), vol.8(2), 2021.

Fariba Rahimi performed the research and wrote the paper.

Paper C

Fariba Rahimi, Jan Wikander, "Concurrent and Optimal Structure, Control and Implementation Design," 2020 IEEE 11th International Conference on Mechanical and Intelligent Manufacturing Technologies (ICMIMT), Cape Town, South Africa, January 2020.

Jan Wikander provided feedback on the technical aspects and reviewed the paper. Fariba Rahimi extended the method and toolbox and performed the simulations and analysis and wrote the paper.

Paper D

Fariba Rahimi, Jan Wikander, "Towards a Formal Framework for Integrated Design-Optimization and Control of Mechatronic Systems," Accepted to be published in Science Progress (SAGE Journal).

Fariba Rahimi performed the research and wrote the paper. Jan Wikander provided feedback.

Paper E

Fariba Rahimi, Lei Feng, Jan Wikander, Daniel Frede, "Early Phase Design-Optimization of Mechatronic Systems," in Proceedings of the 2017 The 5th

International Conference on Control, Mechatronics and Automation, Edmontron, Canada, Pages 42–49, October 2017.

Lei Feng assisted in editing the paper. Jan Wikander provided technical feedback. Daniel Frede was involved in technical discussions. Fariba Rahimi performed the research and wrote the paper.

Terminology

A few key terms that are used throughout the thesis are presented and defined here to assist the reader. Some of these terms might differ depending on the context they are used in or might not be a common knowledge.

Component– A part of a mechatronic system.

Concurrent– Pseudo parallelism treatment of different stages of the development phase in design.

Curse of dimensionality– Exponential solution space increase when adding extra dimensions to a design space [1].

Co-design– An approach of design that involves two disciplines/technical fields/specifications in the design process to ensure the result meets the requirements.

Discipline– An academic engineering field.

IDIOM– A supporting software corresponding to the tool that realizes the method. IDIOM is an abbreviation for 'Integrated Design Optimization of Mechatronic Systems'.

Interface– A point or a device/program which assists the interaction and communication between two systems, elements, components, behaviour, etc.

Method– A design process with encoded algorithms.

Multidisciplinary– Combining or involving several disciplines or specializations in an approach.

Pareto optimality– A situation where no preference criterion can be better without making at least one preference criterion worse [2].

System concept– Interconnected or configured components.

Nomenclature

δ	Sensor resolution factor $[\frac{rad}{pulse}]$
γ	number of control performance constraints
$\hat{a}$	Actuator component
$\hat{s}$	Sensor Component
ν	Poisson's ratio
σ_y	Yield stress of the material
τ	Cycle time of the profile
τ_m	Maximal stress of the arm material
b_c	Carrier width of the gear $[m]$
b	Total gear width $[m]$
C_m	Motor type constant
C_n	Control component
c_p	Control parameters
C_{mj}	Constant for specific machine type
cpc	Control constraint
d_{bs}	Damping in the screw drive $[kg/s]$
d_{sh}	Shaft damping $[kg/s]$
Dy_k	Dynamic behaviour of component k
E	Young's modulus
Eq	Equation list

G	Shear modulus
ip	Input profile
ISE	Integrated square error
J_{bs}	Inertia of the screw drive $[kgm^2]$
J_l	Load inertia $[kgm^2]$
J_m	Rotor inertia $[kgm^2]$
J_{p1}	Inertia of the driving pulley $[kgm^2]$
J_{p2}	Inertia of the driven pulley $[kgm^2]$
J_{pg}	Planetary gear inertia $[kgm^2]$
$J_{sh,a}$	Area moment of inertia of shaft (torsional constant)
J_{sh}	Shaft inertia $[kgm^2]$
k_{bs}	Stiffness in the screw drive $[N/m]$
k_{sh}	Shaft stiffness $[N/m]$
K_T	DC-motors torque constant
l_m	DC-motor length $[m]$
lps	Load specifications
M_l	Load mass $[kg]$
M_b	Mass of the rotational beam
M_n	Mass of the nut drive
MAE	Mean absolute error
$max(er)$	Maximum error
N_r	Number of requirements
obj	Objective function
op	Output profile
Pd_k	Physical dimension model of component k
q	Sensor model

r_g	Planetary gear radius $[m]$
r_m	DC-motor radius $[m]$
r_{sh}	Shaft radius $[m]$
St_k	Static transformation model of component k
$T_{m,rated}$	Rated torque $[Nm]$
T_{rms}	Root mean square torque $[Nm]$
T_s	Sampling time $[s]$
v_m	Volume of DC-motor $[m^3]$
w_b	Fastest pole of the closed loop system
w_c	Bandwidth of continuous time closed loop system
$w_{m,peak}$	Maximum allowed speed of DC-motor
w_s	Sampling frequency $[\frac{rad}{s}]$

Contents

Chapter 1

Introduction

Mechatronic systems are designed using a wide range of engineering disciplines [3]. Design optimization of these systems is a challenging and up-to-date topic that is usually based on sequential and step by step approaches which integrate the domains at the end stage. This requires backtracking and debugging if errors appear in later design phases, which involves both time and cost consuming iterations [4]. Thus, developing approaches that are based on early-phase integration of domains is of high importance. Several design tools are introduced to support mechatronic system features and are divided in two groups: 'specific purposes' tools and 'generic purposes' tools. The tools focused on specific purposes, such as Ansys [5], lack concurrent integration of physical design, control design and embedded control implementation within an optimization algorithm [6, 7]. On the other hand, generic tools such as Modelica [8], even though they have broad applications, do not cover physical scaling/dimensioning approaches or constraints and embedded control implementation factors, therefore narrowing the use case on a different range of problems (design tools for mechatronic systems are discussed in Sections 2.4.2.7 and 2.4.2.8).

In previous studies, a simplification is adopted that ignores the embedded control implementation impact on the final control performance [7, 9] (follow-up literature studies are discussed in Sections 2.4.2.5 and 2.4.2.6). This yields an unreasonable choice of implementation related parameters which deteriorates the control performance and may even cause instability in the system. Hence embedded implementation decisions have significant influence on the performance of the final system solution and should be considered in the integrated design optimization methods and tools. So far, designing embedded control systems has been considered in two dependent steps: first the controllers are synthesized, and second these controllers are mapped and scheduled on a given platform. This approach often leads to either resource under-utilization or poor control performance, and may even

lead to instability of control applications because of the timing problems that arise due to certain implementation decisions [10, 11].

This research focuses on improving efficiency in design and optimization of mechatronic systems by introducing an early-phase co-design method which assists in reducing time and cost consuming backtracking approaches. The method considers integrating physical design, control design and embedded control implementation aspects, simultaneously. A multidisciplinary approach is employed and models for new components are developed. The models are deliberately kept simple by some assumptions on the dynamics, statics and physical design models which focus on capturing the main characteristics of the components (these assumptions are discussed in detail in Chapter 3). The thesis is built upon a concluded PhD dissertation [9] and considers developing models and extending an introduced IDIOM (Integrated Design Optimization of Mechatronic Systems) supporting software toolbox in [9] to encompass multi-DOF nonlinear mechatronic systems with multidisciplinary features. Moreover, the correlations and the impact of design parameters from different engineering domains on objectives are studied. Three engineering disciplines are considered and the key features are integrated in the supporting software toolbox.

1.1 Motivation

Mechatronic systems are found in many everyday complex products. Mechatronics is a multidisciplinary field of study that synergistically combines mechanics, electronics, computers and automation [12, 13, 14]. An interdependent collaboration among the engineers from these fields of sciences motivates a high potential for innovations and information sharing. Systems engineering uses a concurrent method for preliminary design [15]. Accordingly, mechatronics can be considered as an extension of systems engineering which is supplemented with information systems to assist the design, and is employed not only in the preliminary design phase but at all stages of the design. Mechatronic design methods have synergistic combination features and using these methods can exhibit performance characteristics which are difficult to achieve otherwise. In summary, mechatronics increases functionality and features, and enhances efficiency in the design. The solutions derived by mechatronic design are usually more cost effective and environmental friendly, and they improve design time, safety, flexibility and reliability [16, 17].

1.1.1 Mechatronics design facets

Mechatronics has a wide range of application in daily life and industry such as in aircrafts, automation, telerobotics, intelligent highways, servomechanism, sensing and control systems, vision mechanisms, calibration devices, medical equipment, design mechanisms, home security, home appliance and industrial products. A core advantage of mechatronic systems design is the possibility to relate product features to different involved engineering domains. A typical generic design process for mechatronic systems goes through a few phases as follows:

- Research on potential customers and identification of customer/client needs
- Identification and analysis of the problem
- Requirement specification; i.e. identifying product constraints, criteria and design variables
- Information search and literature review
- Concept generation
- Concept evaluation
- Detailed design of chosen concepts
- Evaluation of final concepts regarding requirements, constraints and specifications
- Design of final solution

In this thesis the focus of the multidisciplinary method is on the detailed design phase. Detailed design of mechatronic systems are classified in two groups: sequential and concurrent. A typical sequential mechatronic design process is discussed by Shetty et al. [18] which consists of three phases: modeling and simulation; prototyping; and deployment. Traditional mechatronic design approaches often aim to add reliability and performance into the mechanical part of the system during the development stage. In these methods, the control is usually designed later to provide performance/reliability and reduce errors in the design. Since the design steps are often performed sequentially, these approaches can be classified as sequential engineering design approaches. If a mechatronic system is effectively integrated with the concurrent engineering management strategy, then it can achieve impressive optimal results [19] and it is called a concurrent design method. This thesis favours an integrated concurrent method.

1.2 Thesis overview

Pursuant to the presented motivation, broad-spectrum of mechatronics application, and a need for an integrated concurrent design method, the goals, design choices, level of abstraction, delimitation and domain coverage of this research are presented in this section.

1.2.1 Research goals

The goal of this research is to:

G1: Deepen the analysis of holistic design to enhance efficiency and performance in the concurrent method.

In order to achieve this, a simultaneous design method for mechatronic systems with its supporting software toolbox is to be advanced; mechanical elements are considered as physical components, sensors and actuators as electrical components, control laws and embedded control implementation factors as control components.

G2: Develop a multidisciplinary multi-objective method to derive a balanced optimum solution for a mechatronic design case.

In order to achieve this, design objectives will be derived from different disciplines; hereupon various constraints from the same disciplines will be imposed to the system. The design solution considering a multidisciplinary method is a Pareto front set, and to derive a final optimum system solution there will be a need to apply a function which prevents deteriorating other objectives by improving just one.

G3: Verify the feasibility of the method using a few design cases with multiple degrees of freedom and non-linearity.

In previous publications [9, 20, 21, 22, 23], the method is mainly limited to linear one-degree-of-freedom systems. This thesis will extend the method to handle multiple degrees of freedom components and configurations to cover more complex systems. To achieve this, the same assumption as for simple systems will hold, namely that the individual degrees of freedom will be modelled as linear multiple-mass oscillators. However, multi-degree-of-freedom (MDOF) systems typically lead to coupled dynamics, and in many cases to nonlinear differential equations. Coupled dynamics as such is easy to handle, but to resolve nonlinear dynamics and time-varying coefficients, fixed operating point(s) (in the relevant state variables) should be defined in the method wherein the equations are linearized around it to achieve a linear local approximation. Once this is accomplished, the linear MDOF multiple mass oscillator assumption will hold.

1.2.2 Design choices

Throughout the development of the method, several design decisions are made which are briefly discussed in the following subsection.

A multidisciplinary optimization approach is employed which, with respect to the implemented models in the software toolbox, is capable of deriving an optimum solution considering three engineering domains: mechanical, control and embedded control implementation. The volume, sampling frequency and sensor resolution are considered as the main objectives of the design from the mentioned domains. Other domains and objectives are feasible to be considered by extending the method to include the corresponding models. In practice, there can be more than three objectives as long as the models are integrated in the method.

Genetic algorithm (GA) is chosen as an optimization algorithm that is implemented in the method since it can handle problems with discrete design variables and is implementable on problems with nonlinear objectives. GA is an approach to solve both constrained and unconstrained problems and can be applied to a diverse range of complex problems. To use other optimization algorithms with the method, the corresponding algorithm should be integrated in the method and implemented to the toolbox. To linearize nonlinear systems, fixed operating points are defined. Each operating point yields a new system and requires new evaluation and should be treated separately in the optimization run. The choice of objective function(s) and constraint(s) are flexible for each design and can be reversed.

Four design cases are considered throughout the study which had enough complexity (regarding degrees of freedom, non-linearity, design variables, and involved domains) and represented a real system design concept. Other design cases with different components and configurations require extension of the method to include corresponding models of the components. Some features and properties relevant to modeling and evaluation are enabled in the method and software toolbox by implementing these design cases. The rationale for choosing these design cases is as follows (briefly presented in Table 1.1):

- Ball-screw drive system: Physical component models (beam and ball-screw) are studied and physical dimension, static properties, dynamic behaviour models of the mentioned components are developed and added to the supporting software toolbox. This system is a nonlinear multi-DOF system, hence the method is improved to handle nonlinear systems by the use of this design case. Multi-DOF system configuration is enabled and linearization approach is added as an extra step to the method.

- 2-DOF robotic arm: 2-DOF arm component is studied, designed and

Table 1.1: Method development using the design cases

Design Case	Design and Modelling	Evaluation and Properties
Ball-screw drive	-Beam -Ball-screw	-Linearization -Multi-DOF configuration
Two-DOF robotic arm	-Two-DOF arm	-Lagrangian mechanics -Numerical comparisons with literature
Servomechanism	-Adopted from [9]	-Discrete-time systems -Multi-domain -Multi-objective -Embedded control implementation -Numerical comparisons with literature
Belt-drive system	-Belt-drive -Sensor	-Domain correlation -Parameter couplings -Sensor resolution/location -Qualitative comparison with literature

modelled. To handle nonlinearities in this system, Lagrangian mechanics is employed, therefore this feature is enabled in the method and its software toolbox. This design case is used to compare numerical results of the optimization achieved in our method with results from [24]. The results of this comparison are discussed in Section 4.2 and appended Paper D.

- Servomechanism: To comprehend the impact of parameters from different domains, a simple system is needed where the main characteristics of the system still remain. A servomechanism design case is used to be able to analyse the system in detail and compare the numerical results with a previously published study [9] which does not cover embedded control implementation features. For the analysis using this design case, the design and models are derived from [9], therefore the design and modelling field is left empty in Table 1.1. A discrete time system is considered and modeled. A multidisciplinary design method is used where three engineering disciplines are integrated concurrently and a multi-objective optimization problem is implemented. Two different load profiles that should be handled by the system one at a time are considered to ensure broad application of the method.

- Belt-drive system: A belt-drive component is studied and modelled (statics, dynamics, physical design). A very simple linear model of the belt-drive system is used in [25] which highlights the ratio between the pulleys in the model, however physical and control design for this system are not integrated within an optimization. This system is used to qualitatively visualize the level of design and discipline coverage of our method in comparison to the method in [25]. Nonlinear models of the belt-drive component are studied and the design case is used to further integrate design, control and optimization. The impacts of sensor resolution and location are analysed, and the couplings between disciplines, domains and parameters are investigated.

It should be noted that there is no specific requirement on the components to be integrated in the method as long as it would be feasible to develop dynamic, static and physical dimension (scaling/dimensioning approaches/physical constraints) models for each component. Individual component models are developed throughout the method which allows configuration of system concepts, therefore any system which uses the available developed components is realizable by the method.

1.2.3 Level of abstraction

The method is focused on an early-phase design optimization of multidisciplinary single and multi-objective problems for mechatronic systems. The models that are developed in this thesis are in component level. The method includes both component and system level evaluations. Two types of components are developed in the method as shown in Fig. 1.1: physical and control. Physical components are basic mechanical/electrical elements and control components include control laws, dynamic constraints and embedded control implementation factors. The models for each physical component comprise physical dimension, static properties and dynamic behaviour models.

The physical dimension model basically captures the load actuation capacity or load carrying capacity of the component as a function of component properties such as size, mass, friction or cost (depending on the target optimization criteria). Subject to the component type, the physical dimension is based on load properties such as RMS (root mean square) and peak values derived from the given load and motion profile as requirement. For efficient model evaluation, Roos et al. [26] proposed a way where the models are essentially based on algebraic relations and scaling approaches and this work were later extended by Malmquist et al. in [22]. Budinger et al. [27] applied a similar scaling approach to efficient early phase concept

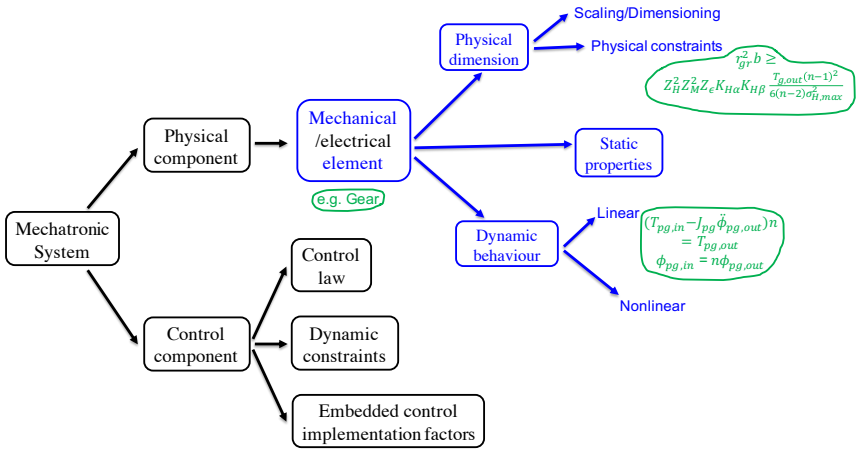

Figure 1.1: Level of abstraction for the design

optimization, but they do not consider control. In this thesis, the physical dimension model includes component level physical constraints (such as speed requirements, torque/power measurements, strength), and dimensioning and scaling factors, or either of these. Based on the required load profile for each component, the physical dimension models are executed and the satisfaction of physical constraints are checked to ensure that non-feasible component dimensions are rejected.

The static properties model captures static transformation of the load from one side of a component to the other. These models are essential for the physical dimension model to initially design the component with respect to transfer of an external system load through the chain of components in the system. As an example, the static properties of a gear is a function of gear ratio and possibly gear friction. The static properties model excludes the effect of damping and stiffness in the system.

The dynamic behaviour model captures the internal dynamics of a component. Since the component models allow for straightforward configuration of more complex systems, the component models are defined to have one functional degree of freedom (for example a rotational degree of freedom for a gear) and two perpendicular non-functional degrees of freedom. In the functional degree of freedom, the mechanical component models are substantially of the same kind, i.e. second order systems consisting of two inertial bodies (J_1, J_2) with physical connection(s) described by linear constitutive equations. Clearly this means that adding just a few components—say a motor, a coupling and a gear—already gives a high order (6^{th}) system.

One way of reducing the model order is to presumably consider stiffer components as infinitely stiff and thereby merge multiple initial bodies into one. In our models, this corresponds to replacing the 2^{nd} order component model with a 1^{st} order component model. These models and example components are discussed further in Chapter 3.

The dynamic constraints as part of control components are imposed to the system in the system level analysis and design. Independent design and modeling of components allows flexibility to configure any type of systems using the available developed components, therefore each component model is treated separately in the design and modeling process. Dynamic constraints are evaluated via passing the system model to an optimization solver, where the physical dimension models are also calculated. Embedded control implementation factors are also incorporated within the optimization loop. The system is designed, controlled and optimized with respect to a specified requirement such as force/torque profile. One optimization loop is utilized which removes a necessity to divide the problem into several problems and a need for several loops, hence it is a time and cost efficient method. The overall structure of the proposed design method in this thesis is presented in Fig. 1.2.

1.2.4 Delimitation

In order for the method to be as holistic as possible, all design parameters are required to be free variables, which would be unrealistic even for a small number of components due to the curse of dimensionality. This is also true for the large number of components. The library of components is not yet rich enough to model and evaluate every type of system; however, the method is capable of handling any system configuration which uses the available developed component models. Concept generation is not the scope of this study. The choice of the used software was made in a previously published dissertation [9] and is not the aim of this thesis. The method is not applicable using other software/tools, since at least one aspect of the design will be lost. The developed software toolbox including the encoded algorithms and models is not yet openly available.

1.2.5 Domain coverage

The contributions of research throughout the thesis covers a few domains which are summarized as follows:

- Multidisciplinary design approach and investigation of couplings and correlations between engineering domains: the overall contribution of the thesis.

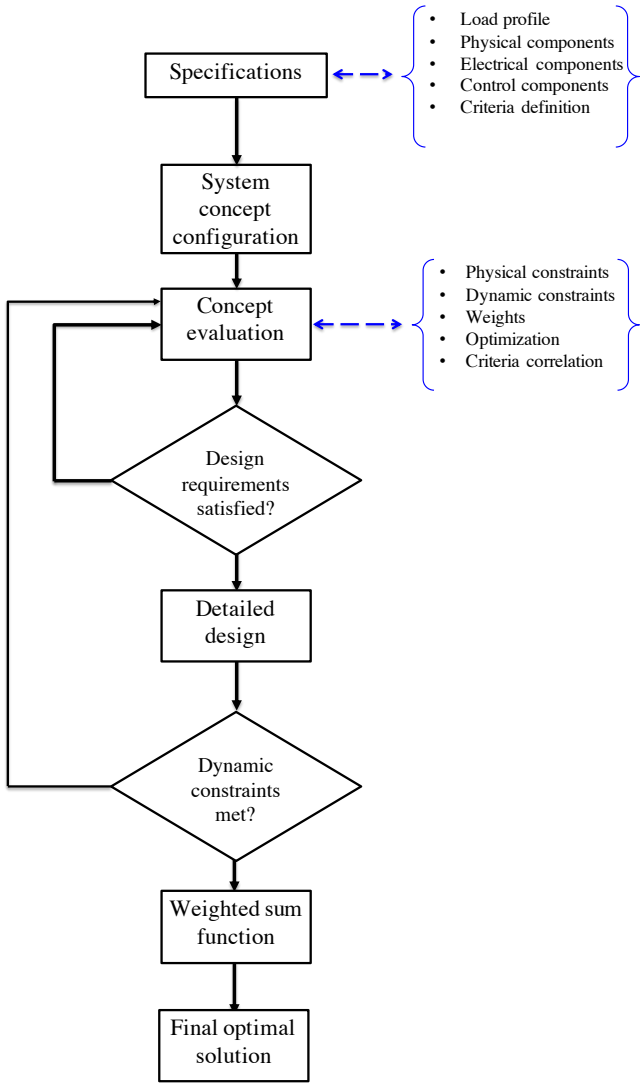

Figure 1.2: Structure of the proposed design method

- Integration of static properties, dynamic behaviour, and physical dimension models for mechatronic components within a multidisciplinary design: the overall contribution of the thesis.

- Models and co-design approaches to include nonlinear multi-DOF

mechatronic systems: provided by Papers D and E. The following detailed aspects are covered by the mentioned papers.

  - Statics, dynamics and physical dimension models for ball-screw and beam components are developed and implemented in IDIOM (Paper E)
  - Two-DOF arm is designed and modeled dynamically and statically (Paper D)
  - Linearization approach is integrated in the method (Paper E)
  - Multi-DOF configuration is enabled (Papers D and E)
  - Lagrangian mechanics are added to the method (Paper D)

- Extended method to cover discussions in discrete time systems, key factors in embedded control system implementation: provided by Papers A, B, and C. The following detailed contributions are covered by the mentioned papers.

  - Multi-domain approach is enabled (Papers A, B, and C)
  - Multi-objective problem is studied (Papers A, B, and C)
  - Pareto optimal set is derived and weighted sum function is integrated in the method (Papers A, B, and C)
  - Embedded control implementation impact is integrated in the model (Papers A, B, and C)
  - Features of discrete-time systems are enabled and the impact of sampling frequency/time on the co-design method is studied (Paper C)
  - Sensor resolution model is developed using a pole placement control (Papers A)
  - Statics and dynamics for the belt-drive component as a new component are developed (Paper A)
  - Design method for the belt-drive system is developed (Paper B)
  - Sensor resolution model is implemented using an observer control (Paper B)
  - Domain correlation and parameters couplings are studied (Papers A, B, and C)

1.2.6 Research questions

A number of research questions, in line with the goals, form the basis of this dissertation.

Q1: Can a balanced optimum solution be realized considering physical (mechanical/electrical), control and embedded control implementation domains simultaneously without increasing the cost and complexity of the evaluations and models?

Q2: Can the design parameters, constraints, objectives, dynamic behaviour, static properties and physical dimension models of mechatronic systems be incorporated within a simultaneous multidisciplinary method to yield a potential design for a mechatronic system concept considering several engineering domains?

Q3: Can multi-degree of freedom and nonlinear behaviour of mechatronic systems be dealt in an integrated manner in an early-phase of design with less computationally demanding models?

1.3 Research design

Several studies have been carried out in the literature [28, 29, 30] on the research methods in the field of systems engineering. Ferris [31] presented a method for engineering design that is based on the project goals and the pragmatic views of the researcher. Creswell [29] presented a framework to characterize the research in terms of philosophical assumptions, strategies of inquiry, and research approaches. Philosophical assumptions refer to "a basic set of beliefs that guide action" [32], i.e. implicit philosophies that the research is based on; strategies of inquiry is the use of quantitative, qualitative, or a mix of both design; and research approach is the approach to collect and analyze data including surveys, literature study, case studies etc.

The research design choice is based on considering these three elements as well as the research goal and problem in the study, and the personal experiences of the researcher [29]. Four different philosophical assumptions are discussed in [29]: postpositivism, constructivism, advocacy/participatory, and pragmatism. Throughout this thesis, in the initial phase a philosophy assumption (namely pragmatism [28]) is used. The pragmatic worldview arises from actions, situations and consequences rather than prior conditions and solutions to the problem. In the pragmatic worldview, the concern is on *what works* and solutions to problems [33]. Therefore the main attention is on the research problems and deriving knowledge about the problem [34, 35]. In pragmatism, it is not essential to use any of the qualitative or quantitative methods, hence a mixed method can as well be used

as long as it best meets the needs and purposes. In a mixed method which is aligned with the goal of this thesis, researchers look to many approaches for collecting and analysing data rather than subscribing to only one way [29].

Strategies of inquiry are types of qualitative, quantitative and a mix of methods designs which provide defined direction for the process in a research design. Quantitative research includes true experiments and non-experimental designs, such as surveys. In the quantitative method there exists an absolute truth that can be observed by empirically analyzing a phenomenon and identifying the cause, since the laws of nature can be observed through carefully designed experiments. The ways to conduct qualitative research are narrative research, phenomenology, grounded theory studies and case studies. This thesis favours either of these research methods or a mix of both. Different procedures are developed for mixed method strategies such as sequential mixed method, concurrent mixed method, and transformative mixed method [29]. Throughout this thesis, concurrent mixed method is employed in which quantitative and qualitative data are integrated to provide comprehensive analysis of the research problem. Research approach involves the forms of data collection, analysis and interpretation. Observing behaviours, conducting interviews and visiting a research site are some ways of collecting data. In some forms of research, both types of quantitative and qualitative data are collected, analysed and interpreted [29].

Given that the entire research design in this thesis is defined by a philosophy assumption (namely pragmatism), a mixed method for the strategies of inquiry, and a concurrent mixed method as a research procedure, the research method that is substantially used throughout the thesis is given in the following subsection.

1.3.1 Design Science research method

Fig. 1.3 presents a framework for Design Science research which covers the relationship between needs and research actions, and existing knowledge and problem solutions. The research method in this thesis is fortified by the design science research method [36], which requires two inputs: the *needs* and *existing knowledge*. The needs should be identified in the initial phase to form the research goals and problems. The needs are defined in an environment comprised of people, organization and technologies. Analysing the aspects of environment makes it easier for the researcher to evaluate how the research results are used in the mentioned environment. Among the many needs that might be identified through research process, the *relevant needs* are considered which, if met, can improve the mentioned environment.

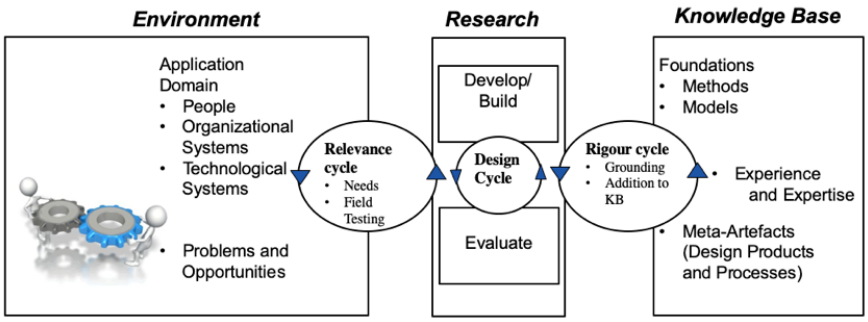

Figure 1.3: Design Science research cycles; adopted from [36]

The existing knowledge includes methods, models, experiences, and meta-artefacts which are developed during the research activities. Existing knowledge includes rigour cycle in its application to guarantee if the produced methods, models and experiences are valid. Research activities are performed using these two inputs, and theories are built which are evaluated by an appropriate method [37]. The outcome of the research in terms of contribution can differ. In the case of theory and experience as the outcomes, it is part of the knowledge base and can be utilized in research activities. If the outcome is an artefact, then it is considered as part of environment that it is used in.

1.3.2 Design Science in this thesis

In this thesis, the process of research (according to Fig. 1.3) is illustrated in Fig. 1.5, considering the Design Science research method and is divided into the five phases (depicted in Fig. 1.4). The needs are identified which correspond to the gaps, and are translated to the research goals and research questions in order to understand the facts and concepts, synthesize existing knowledge and analyze patterns. The needs are derived considering the environment in terms of the application domain. The environment is specified by existing literature and the industrial background. A preliminary state of the art review is carried out in the initial phase (phase 1) to analyse the industrial needs and gaps in the mechatronics design area. Moreover, each of the phases includes relevant literature to ensure that there are unsolved gaps related to the problem, and that the solution by the method is well reasoned. Additional follow-up state of the art is performed in the final phase (phase 5) to certify the previous literature studies and validate the coverage of the method and achieved results.

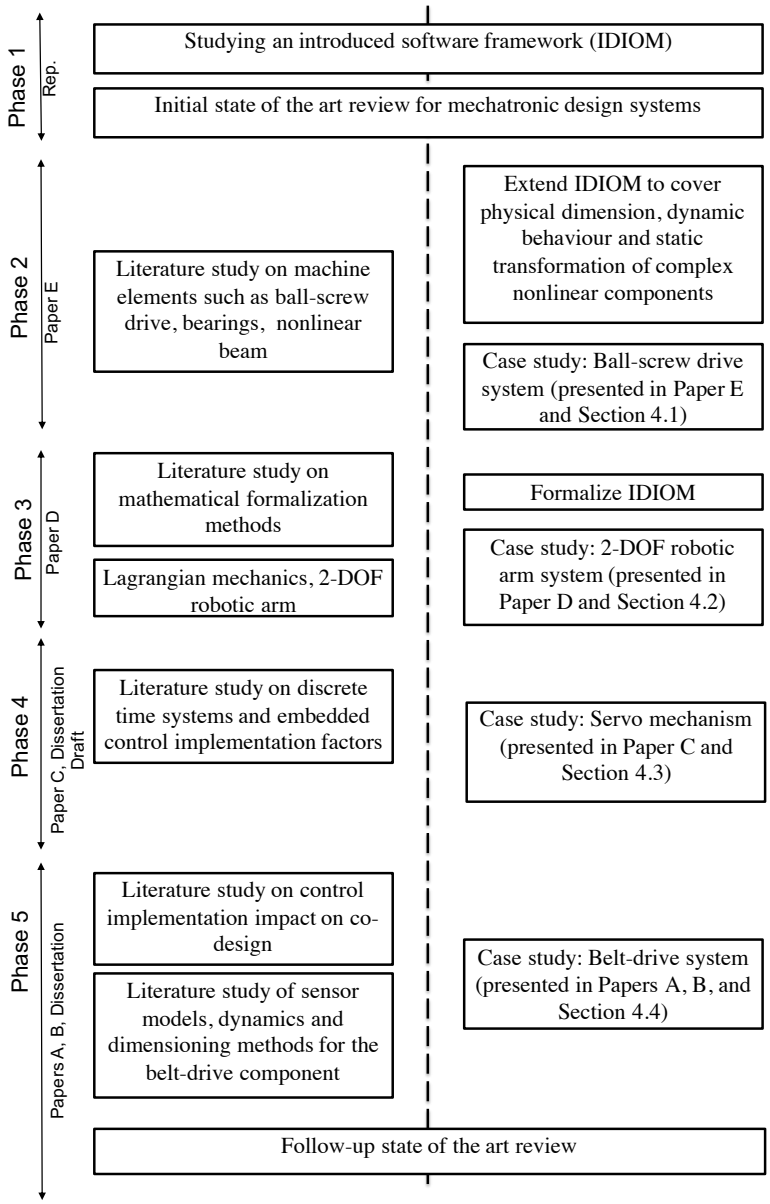

Figure 1.4: Overall outline of the research approach

The research goals in this thesis are formed by the identified gaps, needs and unsolved problems in the literature. For the design method and models that are employed in this thesis which are the basis of research activities, *usability* in the chosen environment is specified as the main criteria. Moreover, when it was feasible, available methods (e.g. developed control laws, common optimization algorithms) with broad applications were applied in the design method.

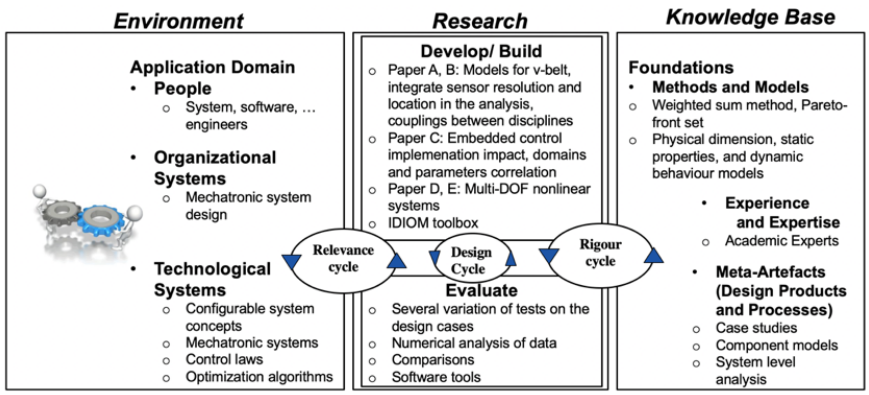

Figure 1.5: Design Science research method in this thesis

The evaluation of methods and research activities is performed using various approaches as shown in Fig. 1.5. In phases 2 and 3, and for the method in appended Papers D and E, the evaluation is performed using case studies and a comparison with previous studies to test if the developed method produces better results. The software toolbox presented in the papers is evaluated through several design cases and the application of it in a realistic industrial context is assessed. Throughout phases 4 and 5, and for the method in the appended Papers A, B and C, the evaluation is carried out in two ways. Firstly, a case study and variation of design tests are used to assess the application of the developed method and models on an industrial conceptual design case, and express the achievements in terms of numeric and graphs. Secondly, the feasibility of the method is evaluated by other tools (Simulink/Simscape is used to further analyse and derive the same type of data, and the obtained optimal system is simulated geometrically using SOLIDWORKS to ensure that it represents a real system). The results are compared numerically with previously published studies and the domain coverage is evaluated qualitatively.

The contribution of appended papers in relation to the Design Science research method is as shown in Fig. 1.5. The contribution of appended

papers are considered to be in relation to the classification of the Design Science research. Each appended paper includes both conceptual problem, applications in the considered environment, and implementation in the developed toolbox. Therefore they can contribute to both knowledge base and environment.

1.4 Thesis outline

The thesis is organized as follows. The first chapter includes an introduction to the field, motivation and application of mechatronics followed by research approach and goals. Chapter 2 presents a brief background research and literature study in the field. Chapter 3 presents a proposed mechatronics system design method consisting of applied multidisciplinary method, developed component models, optimization method, and a brief presentation of the supporting software toolbox. In chapter 4, four design cases which the method is applied on and tested for, are briefly discussed. Chapter 5 reviews the contribution, and gives a summary of the appended papers. The thesis is concluded in chapter 6 with discussions around the research questions.

Chapter 2

Background and related research

It is a significant challenge for industrial companies to design mechatronic systems considering reliability, size, energy consumption, cost and efficiency. Design approaches require involvement of several engineers from different disciplines to brainstorm and take into account the important aspects of each domain. This is an intricate task due to the contradictions originating from interaction and interfaces of different domains. This design method, if not simultaneous, necessitates backtracking in case of errors happening in the system, which is a costly iteration during the process of design, meaning that the final optimal solution might not be a "true" optimum. A lot of research haas been conducted on design optimization of mechatronic systems. This chapter reviews the background, important related work in the area of mechatronic systems design optimization, and multidisciplinary design.

2.1 Design of mechatronic systems

An engineered system is a combination of interacting elements to achieve one or more (n) stated purposes [38]. Fig. 2.1 shows a relationship between system and purposes using UML (Unified Modeling Language) abstract class diagram in graphical form. An engineered system includes a defined concept, requirements and specifications which results in defined objective(s) in terms of stated purpose(s) while satisfying imposed constraints.

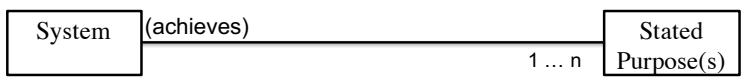

Figure 2.1: Relationship between system and stated purposes [38]

Mechatronics is a combination of mechanical engineering, electronic and

control in the design of products and manufacturing systems. It considers the design of systems and products to achieve an optimal balance between mechanical/electrical structure and its control [13]. A typical mechatronic system includes four different modules [39]:

- A 'base' structure, which is usually a mechanical structure,
- One or more actuators (electrical component) which actuates the machine,
- One or more sensors (control component) to measure and then provide the information on the current state of the machine,
- One or more information processing devices, namely an embedded system, which analyzes the information given by the sensors and control the actuator to yield the desired behaviour.

Tetsura Mori created the name 'mechatronics' in 1969 when he was working for a Japanese company called Yaskawa Electric Corporation where they were building mechanical products. Yaskawa Electric Corporation was using some electronic specifications for manufacturing mechanical systems. Mori introduced a technical term for the new technology by combining the words 'mechanical' and 'electronics' and created the new word "mechatronics" [40]. Mechatronic system design is a design of controlled systems including the integration of main functions from several disciplines. It uses system engineering method (V-model) to realise the combination of different subsystems by the required functions.

Nowadays, V-model [41] is a standard procedure of design for mechatronic system development in industry. The V-model is decomposed into three sections as shown in Fig. 2.2. The first section is the system design with the input requirements, the second section is the domain specific design process in different disciplines, and the third section is the system integration of disciplines which at the end results in a product that is a designed mechatronic system. Most modern methods of mechatronic system design are based on model based design. A model based system design using W-model is presented in [42] which uses SysML (Systems Modeling Language) as a tool for integrated system design. The W-model [43] facilitates the virtual integration of the sub-systems during the system design. The W-model is in a "W" shape and is based on the V-model according to VDI Guideline 2206 [39]. The W-model has the hierarchical structure of the V-model with three levels of system, subsystem and disciplinary-specific components (depicted in Fig. 2.3). Complementary to the V-model, the W-model describes the system design process with five phases: system analysis, specific solutions and dependency analysis, virtual system integration, model analysis and detailed development, and system integration.

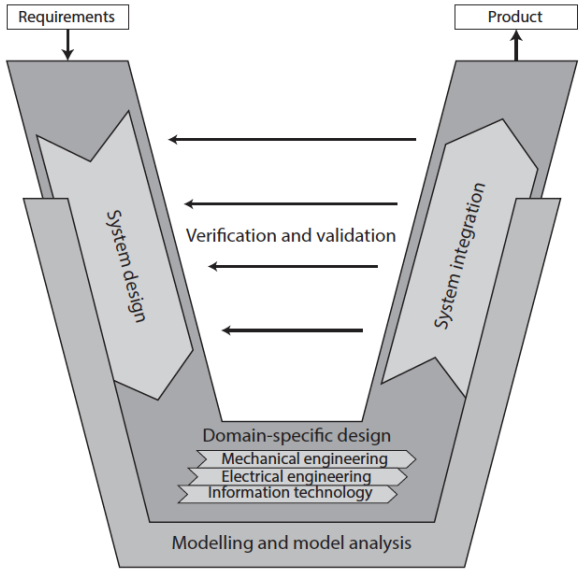

Figure 2.2: V-model structure [41]

An extra phase in the W-model compared to the V-model is the virtual system integration which includes integration of system requirements, system model and functions that allow analyses of the domain specific phase in concept without manual integration of the system. Pahl et al. [45] introduced a process for designing systems in 4 phases:

- definition phase,
- conceptual phase in an abstract level,
- embodiment phase which consists of technological ideas to implement defined concepts,
- detailed design phase.

They considered cognitive psychology to increase the innovation in design and proposed a method for product planning. A few principles of design for recycling were studied and methods for quality assurance were discussed.

Traditional design approaches are based on sequential design methods which deal with various domains separately, and integrate them at the later design phase which, in a case of errors appearing in the product design stage, requires backtracking and debugging that is costly and time consuming [46].

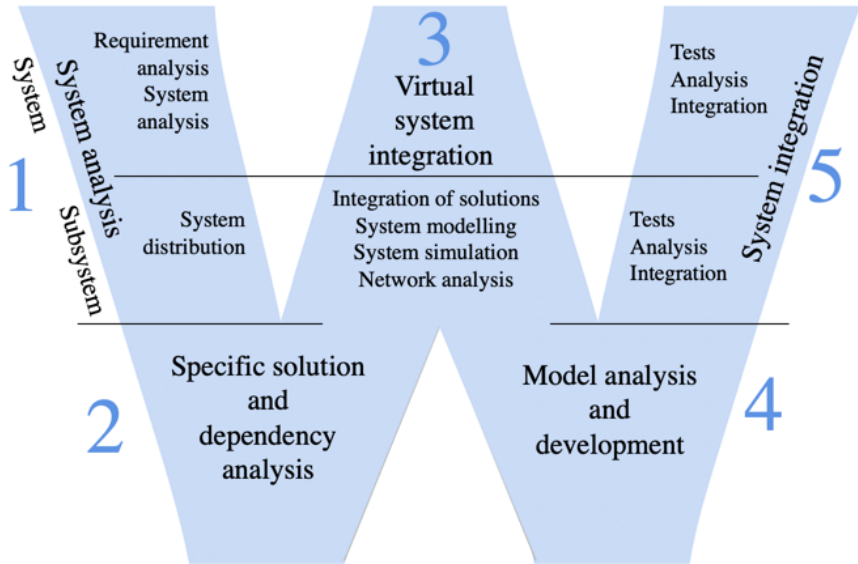

Figure 2.3: W-model structure

2.2 Multidisciplinary design optimization

Multidisciplinary design optimization (MDO) was introduced by Schmit [47] and Haftka [48]. They extended their approach in structural optimization to include several other disciplines. Multidisciplinary design optimization (MDO) refers to a research area which is concerned with developing methods to the design of complex engineering artifacts and systems by interacting physical phenomena [49]. MDO highlights the need for different aspects of the design problem, such as aerodynamics, mechanics and control. MDO was first applied on an aircraft, integrating the three disciplines of aerodynamics, structures and controls [50, 51]. Later, the use cases of MDO continuously extended to other engineering systems such as automotive, mechanical, naval and civil.

The MDO approach is a wide research area which comprises optimization, sensitivity analysis, concept generation and information technology for better decision making in an integrated structure, and it deals with multiple disciplines and interactions of subsystems and models [52]. Development of MDO methods has been an active research study in recent years. The

multidisciplinary development of mechatronic systems attempts to find an optimal balance between the mechanical design, sensor and actuator, implementation impacts, information processing and the control [13, 14] as depicted in Fig. 2.4. Therefore it is of major importance to consider elec-

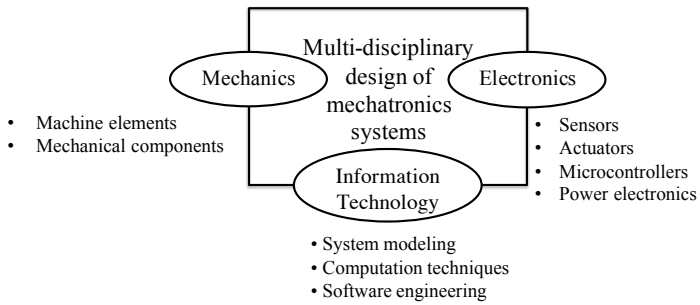

Figure 2.4: Multidisciplinary design of mechatronics systems [13]

tronics, mechanics and embedded control for a holistic design procedure which results in an integrated optimized system. To derive this optimal solution, an optimization algorithm that is computationally and cost efficient should be implemented. In the next section different optimization methods are briefly reviewed.

2.3 Optimization methods

When one or several design concepts and disciplines are selected to optimize a system with respect to a set of initial requirements and specifications, it is called an early-phase (conceptual) integrated design method [53]. The goal of such a design is to search for feasible design solutions and choose a few of them for later design phase analysis [54]. To apply an integrated design optimization requires choosing an optimization algorithm that is implementable to a wide range of problems. Here is a brief review of different optimization algorithms.

2.3.1 Gradient-based optimization

Gradient-based optimizations are iterative algorithms that use the gradient information of the objective function. An example of these methods is the gradient-based quasi-Newton sequential quadratic programming method. The gradient-based optimization employs function derivatives and considers design variables to obtain better optimization solutions. These optimization

algorithms have broad applications, but are unable to solve problems with discrete solution space or discrete design variables [55].

2.3.2 Direct search optimization

Direct search algorithms are used when the objectives are not differentiable or continuous. In contrast to traditional optimization algorithms, direct search methods do not use gradient or derivatives to search for optimal points. Instead they search for an optimal set of points around a current point where the objective function's value is better than the current value. Direct search methods were first proposed in the late 1950s and early 1960s and remained popular because of their simplicity and their use cases cover a broad range of problems [56].

2.3.3 Evolutionary optimization

An evolutionary method known as metaheuristic optimization was proposed some years ago to solve complex engineering problems considering time as the main factor. Metaheuristic optimization is a stochastic search algorithm which finds near-optimal solutions in any problem. Metaheuristic optimization emulates behaviours inspired by mechanisms and processes that exist in nature, such as evolution [57]. Metaheuristic optimization can handle optimization problems with highly nonlinear objective functions and is suitable for optimization problems with discrete design variables. However, it has high computational costs which reduces the use cases of these optimization algorithms. Genetic algorithm (GA) [58], particle swarm optimization (PSO) [59], and ant colony optimization [60] are counted as metaheuristic optimization methods.

2.4 Literature review

2.4.1 Method

To date, many design methods for mechatronic systems have been researched and published. In this section, we have applied a method presented in [61] which suggested three stages for a review: planning the review, conducting the review, reporting the review. The keywords are selected from the research goals and formalized questions in the thesis. During the search, the author has tried to cover the most important and common synonyms of the keywords in the research field. A digitised keyword search that is a fast and efficient method is used which limits the search to electronically available studies (published after 1993). The keywords used in the search are

mechatronic system, *concept development*, *multidisciplinary design*, *multi-objective optimization*, *integrated design optimization*, *control implementation*, and *concurrent design*. Electronic documents in English are considered. A database search is used (not a forward and backward search method [62]). The search strings are used on 3 frequently used digital libraries:

- Science Direct (https://www.sciencedirect.com/)
- Google Scholar (scholar.google.com)
- IEEExplore

Science Direct and *IEEExplore* include up to date research and highly cited publications (both journals and conferences) in the area. *Google Scholar* is a library that does not limit the search to a specific publisher. The articles in other languages than English, and ones with no access to the full text, were excluded. The articles merely focusing on analysing mechanical aspects/control aspects/optimization approaches without integration method were excluded. The studies focusing on firmware routines, hardware protocols, co-simulation of control systems, cyber physical systems, product management, and product life cycle were excluded. 107 publications of more relevant research in the area were compiled (for each manuscript, preliminary relevance was determined by its title and abstract), including 64 articles in scientific journals, 40 from conferences, and 3 book and dissertation chapters.

The search for keywords was conducted as below:

1.
   - "integrat* design optimization" AND "mechatronic*"
   - "concurrent design" AND "mechatronic*"
   - "simultaneous design" AND "mechatronic*"
   - "*design" AND "mechatronic*"
   - "early* design" AND "mechatronic*"
2.
   - "concept develop*" AND "mechatronic*"
   - "system* concept" AND "mechatronic*"
   - "*design concept" AND "mechatronic*"
3.
   - "*disciplinary design" AND "mechatronic*"
   - "collaborative design" OR "parallel*" AND "mechatronic*"
   - "synthesized design" AND "mechatronic*"
   - "*domain" AND "mechatronic*"
4.
   - "*objective optimization" AND "mechatronic*"

- "*criteria" AND "optimization" AND "mechatronic*"
- "optim* solution" AND "mechatronic*" AND "dimension*" AND "control"

5.
- "dimension*" AND "control implementation" AND "mechatronic*"
- "dimension*" AND "embedded control implementation" AND "mechatronic*"
- "sensor resolution optimization" AND "mechatronic*" AND "*objective"

2.4.2 Results

In this Section, with respect to the selected keywords, a few topics in mechatronic systems design are covered as a result of the literature background including 'design management', 'conceptual design', 'design optimization' and 'design tools'. Table 2.1 presents a summary of the topics and selected references covered in this review, divided in two groups of multi-domain and single domain.

2.4.2.1 Design management of mechatronic systems

Adamsson [63] addressed the management of multidisciplinary and complex product development and studied ways to increase operational performance in multidisciplinary engineering organizations. He emphasized the interaction and collaboration facets in discipline integration rather than considering mechatronic as a technical solution. He has highlighted that specific integration features (such as organizational structure, work procedures and methods, training, social systems and computer-aided engineering) are more important than other aspects. Carvalho et al. [64] studied the influence and integration of design thinking, concurrent engineering and agile as the most effective design philosophies in mechatronic systems. They discussed that design thinking is for exploring ideas and requirements in the early phases of design development, whilst concurrent engineering and agile are for coordinating skills of developers in early stages of the design process.

Bretz et al. [65] discussed the challenges in development of interdisciplinary complex systems using model-based system engineering method. They highlighted the need for an integrated meta models, organization, and system model as the basis for an information management tool for the early-phase development of engineering products. Törngren et al. [66] presented an approach which elaborates on integrating tools using models which describe the relationships between them in terms of data exchange, traceability

Table 2.1: Summary of the topics and selected references

Selected Topics *(covered area)*	**Publication Year**	
	Multi-domain	**Single domain**
Design Management *(operational performance, interaction, agile, data exchange, design thinking, collaborative architecture, knowledge management)*	2021(1), 2019(1), 2018(1), 2016(1), 2007(1), 2002(1)	2014(1), 2003(1)
Conceptual Design *(performance, cost, safety, damping time, functionality, trajectory tracking, synergy, system integrity, flexibility, efficiency, intelligence)*	2019(1), 2018(1), 2017(2), 2016(1), 2015(2), 2014(2), 2012(3), 2008(1), 2007(1), 2004(1), 2002(1)	2016(1), 2014(1), 2012(2), 2010(1), 2002(1), 2001(1)
Design Optimization *(performance, topology power/energy consumption, efficiency, safety, vibration effect, uncertainties, path generator, latency, reliability, robustness, structural design, disturbance/noise rejection thermal characteristics manipulability measure)*	2019(4), 2018(2), 2017(3), 2016(4), 2015(3), 2014(3), 2013(7), 2012(2), 2011(1), 2010(2), 2009(3), 2008(2), 2007(4), 2006(2) 2005(3), 2004(2), 2002(3), 2001(2), 1996(1), 1993(1)	2020(1), 2018(2) 2017(2), 2014(1) 2012(1), 2009(1) 2008(1), 2007(1) 2005(1), 2002(1)
Mechatronic Design Tools	2017(1), 2015(1), 2012(2), 2011(1), 2010(1), 2003(1)	2011(1), 2010(2) 2006(1)

and invocation. In their approach, product development is considered from level of people, using dependency models, to relations between tools.

Wang et al. [67] described collaborative design as a design method where a system is designed by collaboration of many designers from different engineering disciplines. An approach based on knowledge management (KM) and graph theory is presented in [68] for design optimization of mechatronic systems. They have discussed the importance of KM in the design cycle of mechatronic products and developed a platform to support it. They have considered product knowledge, collaborative architecture and process knowledge in their platform. Mcharek [69] proposed a KM approach to organize the obtained knowledge during collaboration, and synthesized the design cycle. He introduced a method for integration of system engineering

(SE), disciplinary engineering (DE), and multidisciplinary design optimization (MDO) for an integrated design. A conceptual design method for mechatronic systems that is based on multi-agent approaches is discussed in [70]. In this method, agility, intelligence, cooperation and networking are all incorporated into the design paradigm.

2.4.2.2 Comments/discussion on the design management of mechatronic systems

The studies presented in Section 2.4.2.1 highlight the necessity for integration of concurrent engineering with mechatronic design during the design process. The need for tool integration for structuring more technical co-design facets is discussed. However, they mainly focus on management and organizational aspects of the design in mechatronic sytems, which is not the scope of design in this thesis. In case the method would be developed to integrate design philosophies such as cooperation, agile, information management, data exchange and networking, then these studies would be of great relevance.

2.4.2.3 Conceptual design of mechatronic systems

Scheidl and Winkler [71] studied the relationships and differences between the conceptual and detail design to evaluate the validity of a system concept. They have discussed the mapping from the space of design parameters to the space of functional requirements, and considered the development of a hydraulic variable valve train for combustion engines as a design example. A model-based early design phase method is presented by Safavi et al. [72] which considers the performance, cost and safety features of a system overall, and was examined on an industrial aircraft system. The integration method is performed by surrogate modelling techniques which yields to low computational cost models. A method for modeling and simulation of aircrafts is investigated in [73] which considers the sizing and analyses the performance. This method is discussed for implementation on an aircraft power system.

Hafner et al. [74] presented a mechatronic design method and a software/hardware environment for engine control systems. They studied a dynamic engine test including a Rapid Control Prototyping (RCP) system to increase the speed of implementation of new control algorithms. Awtar et al. [75] presented a design for a ball-on-plate balancing system which was built from an initial design concept. The design objective was to make the ball-on-plate balancing system compact, cheaper and quicker considering constraints such as cost, performance and functionality. Gausemeier et al.

[76] presented a graphical method to express the functional principle solution of a mechatronic system called the semi-formal specification language. They applied this method on a vehicle guidance system.

A method based on VDI 2206 standard is presented in [77] to design, validate and integrate a two-axis sun tracking system. They created an efficient energy collection system through a combination of design mechanisms, instrumentation systems and trajectory tracking control. Their method does not use optimal design techniques. Rooker et al. [78] presented a concept to model flexible mechatronic systems using simulation support and distributed control programs. They developed modular mechatronic components and analysed the behaviour and requirements of each component in the assembly system. Chaaban et al. [79] used AUTOSAR standard rules to design a safety critical mechatronic system and applied their method on an embedded steering-by-wire system. They specified the software and hardware architectures based on safety and functional requirements of the system, and simulated the system both at system and behavioural levels. Komoto and Tomiyama [80] presented a framework for conceptual design of multidisciplinary systems where the system architecture was used to decompose design information within different engineering domains. Their method consisted of a few steps:

- identify requirements and translate them to a system level specifications,
- decompose the specifications to component level specifications,
- define structure of subsystems and interfaces.

A multi-disciplinary design method based on model-based systems engineering (MBSE) is presented in [81], which integrates the interdisciplinary modeling tool SysML. The feasibility of applications in mechanical and plant engineering for this method is presented by using five design cases based on an industrial development environment. The risk assessment is integrated to the mechatronic system design in an early stage of design by Sierla et al. [82]. Their approach is capable of identifying fault propagation paths that has different disciplinary constraints, and investigates the impact of several faults in software-based automation subsystems, electric subsystems, and mechanical subsystems. They used a boiling water nuclear reactor as a design case. A design method based on multi-agent technique is presented by Hammadi et al. [83] to improve the design integration while reducing optimization complexity and computation cost. They applied their method to a design case of an electric vehicle and verified the effectiveness of their method.

A multi-agent method is studied in [84] which considers multi-abstraction modelling issues in mechatronic systems. Using their method, a multi-level design is decomposed into agents where each agent has an abstraction level. The links between these levels were derived by relationship models. They tested their method on a piezoelectric energy harvesting system, and investigated how their method reduces design cycle and modelling errors. Yavuz [85] introduced a method for systematic conceptual design of an intelligent autonomous mobile robot. The core facts in his approach are system integrity, efficiency, performance and synergy. This approach is capable of developing numerous types of robots with different configurations and specifications. The advantages of using smart sensors are highlighted, and the benefits of function oriented analysis in mobile robots to derive integration guidelines are studied. A conceptual data model is presented in [86] to evaluate system properties caused by design changes in the physical architecture of systems. They studied the impact of design decisions on final system performance and applied their approach on a mechatronic design of an electrical assistance for a wheelchair.

For multi-disciplinary conceptual design, a method based on integrated principle solution synthesis (IPSS) is proposed in [87] which uses functional knowledge and structural knowledge. Their method is capable of deriving a solution that satisfies the desired function as well as solving physical conflicts between different selected principle solutions. An approach for the conceptual design that is based on an evaluation model is presented in [88]. The proposed method includes three features: intelligence, flexibility, and complexity of mechatronic systems. Their method includes conceptual decision making in a discrete search space. However, continuous space in the concurrent detailed design is not discussed and their method does not consider interactions between criteria. Hammadi et al. [89] presented a multi-criteria performance indicator as a neural network of radial basis functions for early stage mechatronic systems design. However, this method does not consider modeling the interactions of objectives or sub-domains. Oestersötebier et al. [90] studied a design approach which considered flexible manipulators as design systems. They applied a linear quadratic regulator (LQR) method to evaluate system performance. The objectives of design in their method were damping time, the deflection of link tip and the rise time.

A method for mechatronic system development is presented in [91] which incorporates fuzzy based delphi method (FDM), interpretive structural modeling (FISM), analytical network process (FANP) and quality function deployment (FQFD). FDM identifies the requirements and engineering characteristics, FISM is used to priorotize the important factors, and FQFD and FMEA handle the customer requirements and failure analysis in the design process, respectively. Casner et al. [46] integrated optimization in

the design phase of mechatronic systems where their focus was mainly on the embodiment phase of the design. Their approach assists in deriving a design layout solution from a set of design concepts. In another study [92], they presented a novel multidisciplinary design optimization method for mechatronic systems which extended the V-model with the modelling and optimization process. Their presented approach consisted of four levels:

- the definition step from V-model,
- the modelling step which derives a parametric model for different components, subsystems and systems,
- the optimization step which derives an optimized prototype,
- the validation phase which the design process terminates with.

2.4.2.4 Comments/discussion on the conceptual design of mechatronic systems

The focus in the presented research in Section 2.4.2.3 is on the conceptual design rather than detail design of the systems. These studies consider functionality and performance aspects; moreover, they have used complex and costly modeling approaches and applied their approach on a specific design case or design problem. The models are in system level rather than in component level, which limits the generality aspect of the design. Our method focuses on individual component models rather than system level models, therefore it gives flexibility to configure any system concept (generic feature) which uses the available developed components. A few of these studies, along with their focus area and application, are summarized in Table 2.2. Computation time, cost, safety, efficiency, synergy, performance, intelligence, functionality, flexibility, failure analysis and complexity are the main criteria discussed in the above research. Some of these factors (such as computational cost, efficiency, flexibility and complexity) are already taken into account in our models and method. The rest are good criteria to be considered as an extension of the method in the future. As with much other research in the field, these studies do not cover the integration of physical design, control design and embedded control implementation in the early phases of the design process. Accordingly, a correlation between the domains and parameters is not investigated.

2.4.2.5 Design optimization of mechatronic systems

A single objective optimization problem based on an artificial bee colony algorithm (ABC) is presented in [24]. They have tuned control parameters

Table 2.2: Design methods and their focus area

Focus	Ref.	Application
Detailed vs conceptual design	[71]	Combustion engine
Surrogate modeling	[72]	Industrial aircraft
Functional principle solution	[76]	Vehicle guidance system
Efficient energy collection	[77]	Two-axis sun tracking
Safety	[79]	Steering-by-wire system
Risk assessment	[82]	Boiling water nuclear reactor
Multi-agent technique	[83]	Electric vehicle
Multi-abstraction modeling	[84]	Energy harvesting system

for a two-DOF robotic manipulator by defining three objectives one at a time: mean absolute error (MAE), mean of root of squared error (MRSE), and reference based error with control effort (RBECE). They investigated the coupling between the joint torques applied by the actuators and the position and acceleration of the robot arm. A concurrent design optimization method is presented in [93] which considers quad-copter as a mechanical device and optimizes the length of the arms and the LQR control gains. The control effort is minimized by minimizing the power consumption of the electric motors to increase the battery life of the aircraft, and a few control related objectives such as Integrated Square Error (ISE), Integral of Time Squared Error (ITSE), Integral of Time Multiply Absolute Error (ITAE), and Integral of Absolute Error (IAE) are considered one at a time to be minimized by the genetic algorithm. However, there are no physical and control design constraints integrated in the approach. Additionally, specifying more free physical design parameters could yield a better and more realistic design.

A constrained linear-quadratic optimization method is used in [94] to design a dual-linked dual drive gantry system. The objectives of the design in their method was to minimize the vibration of the end effector, and the tracking errors by finding solutions for flexure stiffness and the control parameters. A concurrent design optimization method is applied on a single-

link flexible manipulator system in [95] where a simple PID control is also integrated in the model. In their method, the efficiency and complexity of the optimization are considered as the main design characteristics.

A controlled random search (CRS) approach is presented in [96] for structural and control design optimization, applied on a single-beam manipulator. They used a finite element method (FEM) to model the flexible link system dynamically and applied a PD control to derive the closed loop system model. A method for structural and control optimization of an active suspension system is presented in [97]. Their method considered the riding and safety performance as the main system criteria. They used a simulated annealing (SA) method to optimize the system parameters. SA is advantageous for problems where finding an approximate global optimum is more important than finding a precise local optimum in a fixed amount of time, and is often used when the search space is discrete. An optimization approach to model and simulate a mechatronic system is proposed in [98]. This approach was implemented in MATLAB [99] and Simulink and was tested on a controlled semi-active suspension. In this method, the optimization optimizes the sub-system parameters while preserving the car ride and performance.

A multi-objective problem is formulated in [100] for robust concurrent design of a 2-DOF collaborative robot. The error of trajectory tracking and sensitivity to uncertainties in the model are defined as optimization objectives. The source of uncertainty in their design is from the human operators (due to interactions). The uncertainties are modeled as time-variant forces imposed on the system. They formulated their robust concurrent design approach as a nonlinear dynamic optimization problem and verified the effectiveness of it by conducting a comparison between a robust and a non-robust method. Carabin et al. [101] presented a point-to-point trajectory optimization to reduce the energy consumption of a linear axis 1-DOF mechatronic system with an electric motor, a transmission and a toothed belt and a load. Yacoub et al. [102] implemented an energy optimization algorithm using predictive control on a two-DC motor-driven wheels mobile robot while crossing a ditch. They simulated and compared the predictive control algorithm with the PID control and the open-loop control.

Casner et al. [103] integrated optimization tools to improve the functionalities and performances of mechatronics systems during the development phase. They developed an evolutionary case-based design approach to optimize both the structure and design parameters of a system, and applied their method to optimize an XY table for laser cutting applications. Coelingh et al. [104] presented a method to provide a design solution considering four design factors on the system performance: motion specification, path generator, control system and plant system. They classified plant dynam-

ics and presented different types of open-loop transfer functions and later described how these transfer functions are acquired from different system models and sensor locations. They studied the relationship between design factors and system performance, and applied their method on an electro-mechanical motion system with flexibility.

Jiang et al. [105] introduced a method for design optimization of mechatronic systems by sensitivity analysis of design parameters to reduce the design solution space. They used a neural-network-based optimization design approach to investigate the system performance using the key design parameters. Simon et al. [106] proposed a method for robust control and scheduling co-design. They tuned the control periods according to their impact on the control performance and then regulated an outer scheduling controller in real-time according to the allocated computing power. Subsequently, Simon et al. [107] presented a method to design a robust controller by taking the sampling interval into account as a control variable of the system. They considered plant parameter uncertainties in robustness of the control.

Cervin and Eker [108] presented design of a flexible real-time control system for implementation of feedback control tasks in embedded systems. They studied some features of the model, such as latency and sampling time/jitters, from an embedded control implementation perspective. Kurdi et al. [109] used a temporal spectral element method for discretization of time-dependent differential equation and optimization of a dynamic system by computing monolithic-time responses and constraints. An optimal design of mechatronic systems with configuration-dependent dynamics is considered in [110]. They tested their approach on a pick and place robot with a gripper, and studied the non-convex nature of the optimization problem and challenges in modeling of a serial machine with flexible components and its embedded controller. Subsequently, a method for modelling and simulation of a mechatronic system with configuration dependent dynamics considering its embedded control system is presented in [111]. They have used co-simulation between LMS Virtual.Lab Motion and Simulink. They examined their method on a pick and place machine. Their method is capable of evaluating a system not only in discrete operations but also in a continuous one. The couplings between Finite Element (FEM) analysis and system simulation are considered in [112] to design mechatronic systems. Their method studied the nonlinear effect of the electrical machine, such as saturation and tooth harmonics. Dumlu [113] proposed a fractional-order adaptive integral sliding mode control scheme to perform a trajectory tracking control of six-degrees of freedom robotic manipulator. Their research is one of huge number of other studies that focuses on control of robotic systems where the component design and optimization is neglected.

Affi et al. [114] compared a sequential design optimization method with a simultaneous design method. They tested their method on a four-bar mechatronic system using genetic algorithm optimization. In a sequential method, they considered optimizing geometry of mechanism for a given path, and later solving dynamic problems with a fixed structure. In a simultaneous method, all the parameters of the design are considered concurrently and a multi-objective problem considering path error, current and current fluctuation are specified as objectives of the design. They discussed the advantages of a multi-objective optimization method over a single objective one. A concurrent topology optimization method is presented in [115]. They used level sets to model dynamics with random and interval uncertainties. The material layout was derived for the structure in both macro and micro scales and the dynamic compliance was minimized, and the influences of the uncertainties on the optimized structures were studied.

A bond graph-based system tool was introduced in [116] where a two loop method was developed using genetic algorithm (GA) and genetic programming (GP) for design optimization of mechatronic systems. GA was used to find system solutions in terms of parameters and GP was used to optimize system topology. They implemented their method on an electrohydraulic manipulator of an industrial fish processing machine. Subsequently, a mechatronic design method using a niching genetic algorithm was presented in [117]. The method was performed in two stages: first different feasible configurations were derived using the niching genetic algorithm and then the achieved solutions were compared according to all criteria involved in Mechatronic Design Quotient (MDQ). This method was tested on an industrial fish cutting machine as an electro-mechanical system. A reliability-based framework is introduced in [118] which investigates dynamic interactions between system topology and components. They claim that their method reduces design errors which arise when reliability based topology and components are considered separately.

An approach based on SysML is presented in [119] for designing physical architecture of mechatronic systems. Their method is composed of two stages: gathering all the requirements and information for design; and finding solutions and comparing them to derive a final architecture for the system. A multi-objective optimization method based on a cascaded evolutionary process to design a directional control valve is presented in [120]. In their method, the optimization is divided into an outer and an inner parameter set. The outer loop for the optimization only dealt with hardware without considering control. The tuning of the inner set necessitated the need for parameterized model with lumped parameters. Hence whenever the plant behaviour changed, the controller parameters were updated. An artificial immune system (AIS) method is used in [121] to integrate domain

knowledge incorporation and concurrent mechatronic design. To handle the optimization of both topology and parameters, two optimization loops were used. The inner loop uses an optimization tool for parameter tuning in generated topology by combining AIS. The outer loop synthesizes the number and the structure of the elements using AIS tool. Albers et al. [122] proposed a topology optimization approach to optimize the structure of a controlled mechanical system. In their method, they integrated multi-body system dynamics (MBS), finite element analysis (FEA), control system simulation and topology optimization.

A gradient-based framework is presented by Lee et al. [123] for optimization of an aerodynamic system, and the controller is designed using high-fidelity models. They considered the system's general properties, such as time scales of the model, and used cost function to reduce the computational task of fluid dynamic simulations. They reduced the optimization time by control of the error of gradient which is computed by an optimizer. They provided a few use cases that show their method is applicable for optimizing supersonic vehicle shapes and can handle both sharp and smooth geometry design parameters. An approach for designing an optimal control for 'hybrid' mechanical systems is presented and implemented on a compass gait biped in [124]. In their method, 'hybrid' refers to both continuous dynamics that are in the form of Lagrangian equations and discrete behaviour of different motion phases of the biped. They used Discrete Mechanics and Optimal Control (DMOC) to formulate optimal control as a constrained optimization problem. Samii et al. [125] studied a formulation and solution for control and scheduling co-design problems. They optimized overall performance of control loops for a given set of plants with disturbances and measurement noise.

Hehenberger et al. [126] used hierarchical models to design mechatronic systems, and to analyse and evaluate functional requirements and design parameters of possible solutions. Their method is capable of combining solution principles from different mechatronic domains as well as handling complexity of the integrated design tasks. Villarreal-Cervantes et al. [127] presented an approach which considers robustness in structure-control design in the presence of uncertain parameters. The objectives of the design in their method are the tracking error and the manipulability measure. They tested their method on a five-bar planar parallel robot with a proportional-integral-differential (PID) control. Subsequently, a concurrent design method was applied on a planar five revolut two degrees of freedom (5R 2DOF) parallel robot in [128]. Two optimization approaches were implemented: a nonlinear programming technique and a novel evolutionary approach. A PID control was used and the kinematics and dynamics of the system were treated in only one design step. A decoupled design method

is presented in [129] using decoupled design models. They also conducted a comparison between a decoupled model and a multi-objective optimization problem. Feasibility was evaluated by applying their method on a four-bar mechanism.

Chhabra and Emami [130] presented a method using bond graphs and block diagrams for a concurrent design approach. In their method, a mechatronic system is considered as an energy system and they used the laws of thermodynamics to specify design criteria. The principles of a multidisciplinary system and the flow of energy and information throughout its different constituents were studied. Subsequently, they introduced a fuzzy logic based concurrent design framework and used a 5-degree-of-freedom industrial robot manipulator as a design case. They applied a holistic concurrent design approach to transform a multi-objective constrained optimization to a single-objective unconstrained problem [131]. In another study [132], they have used modular hardware-in-the-loop simulation to discuss a concurrent design method for robot manipulators. Using this method, it is possible to have physical components in the simulation loop which makes the computations efficient. They applied a holistic concurrent design (HCD) approach to convert a multi-objective constrained optimization problem into two single objective unconstrained problems.

Trivedi et al. [133] proposed an optimal design method for OctArm manipulators which involves specifications of air muscle actuators and configuration of sections to maximize dexterity and load capacity for a given actuation pressure. In their method, after generating design rules for the optimization problem, optimal solutions for pneumatic and hydraulically actuated soft robotic manipulators were obtained. Their method addressed trade-off in soft manipulator design between dexterity and strength. Lyshevski [134] studied classification of micro-mechatronic systems (MMS) and developed concepts for structural synthesis and optimization of motion control problem. They used mathematical models for MMS and applied the Lyapunov theory to derive solutions in prospective applications of MMS in aerospace and automotive industry. A method for optimizing a rotary inverted pendulum is presented in [135] which considers the disturbance rejection and domain of attraction as the main objectives of the design. However, their method does not provide a unique optimum solution for a system.

A holistic co-design approach for robot arm structure is presented in [136]. Their approach includes optimization of the design parameters describing arm link geometry, actuator locations and feedback gains considering the settling time of the system as the objective function. The structural parameters are optimized using a non-linear programming technique to obtain an overall optimal performance. A method that combines experimen-

tation with a design process is developed in [137] to integrate design and control of a mechatronic system. The control gains are optimally tuned. Using those optimal values, the mechanical structure is modified until the control performance is improved. Domingues et al. [138] presented a design method to achieve size, efficiency, optimal driving performance and thermal characteristics of electric powertrain components. Their method finds optimal component combinations based on some requirements. The *driving cycle* behaviour and specific operating modes (such as maximum wheel speed) are considered. Their method allows the selection of electrical machine, sizing of the gearbox and the power-electronic converter needed to handle the wheel torque and speed requirements. They used their method to optimize an electric powertrain of a passenger vehicle.

A design optimization method is presented in [139] that is divided into two stages- structure design process and control design. For structural design, a CAD modeling tool (namely Pro/ENGINEER) was employed and 3D models designed, and the AnSys which delivers product modeling solutions was used to design the structure. Later, the control was designed in MATLAB. The method of design in their approach is a sequential method where different stages of design are executed sequentially, therefore the design might not yield a true optimum solution. O'Driscoll [140] used the design for manufacture (DFM) method to address the coupling between product design and manufacturing by considering manufacturing needs during product development. DFM is used to design products by taking manufacturing issues into account. Early consideration of manufacturing related requirements shortens the product development cycle time, minimises the cost for development, and ensures a smooth transition into production.

Broenink et al. [141] presented a method to support design and implementation of control software for mechatronic systems. They considered the process as a step-wise method from physical system models and control laws for efficient computer code. Allison [142] applied plant-limited co-design (PLCD) method on a robotic manipulator. PLCD is usually used on an existing system to modify control or physical design at minimum cost, and attempts to meet new specifications and requirements. The new requirement in their method for this design case was defined to minimize energy consumption for a pick and place task. Subsequently, a formal method to identify plant modifications was introduced in [143]. In this method, a co-design approach was used to minimize the cost of plant modification, as with PLCD. A concurrent multi-objective dynamic optimization approach is presented in [144] which considers energy, tracking error and total weight of the motors to be minimized and attempts to select AC servomotors for industrial manipulators. In their method, the optimization constraints are defined to avoid overheating and overexertion of the powertrain (motor-

reducer-load).

A multidisciplinary method is applied by He and McPhee [145] to design a vehicle with active suspensions. They used the LQR control method with a Kalman filter and integrated design and control and obtained an optimal solution using a genetic algorithm. Allison and Herber [146] presented a multidisciplinary dynamic system design optimization where they used MDO to integrate disciplines such as energy domain, and studied the behaviour of dynamic systems to derive a better system performance that yields efficient solutions. A multidisciplinary interface model is developed in [147] to integrate design discussions from different disciplines. *Interface* is used to refer to logical or physical relationships that integrate components of one mechatronic system (or the components) with the environmental factors. Their method is based on three steps:

- interface classification—the foundation of the multidisciplinary approach,
- interface model—the product model with all the information required for the design,
- interface compatibility—to assure the correct integration of different components in the system.

Alexandrov and Lewis [148] presented a collaborative method for multidisciplinary design optimization purposes. These methods are considered to be in a bi-level structure. In a local level, the system performance is optimized, and in a global level the attempt is to reduce the discrepancies among the disciplines by minimizing interdisciplinary inconsistency in the variables. To support multidisciplinary design optimization, a method is presented in [149] which concentrates on intelligent modeling systems in order to generate a multi-physics simulation automatically. To achieve this, they used a knowledge based engineering (KBE) approach, and integrated the design and simulation knowledge from multiple engineering domains. In their method, the required knowledge from a multi-physics information model (MIM) are extracted to instantiate sub-models, then the sub-models were combined to construct the multi-physics simulation model. Subsequently, a design method that is based on knowledge-based techniques is presented in [150] for mechatronics vehicles. Their method consists of two stages: first multi-physics simulation models are generated within an optimization framework, and then a controller is designed. Their method assists in analysing performance of unstable mechatronic vehicles in early stages of design.

A multi-objective design method is presented in [151] which deals with uncertainties in mechatronic systems. Their method is tested on a mecha-

tronic sewing machine where the motor current and its fluctuations were defined to be the objectives of the optimizer to be minimized. They studied the impact of uncertain design parameters on the performance of the system. Guizani et al. [152] presented a constraint based partitioning method to break down mechatronic design problems to several partitions and then optimize the system. Each partition has a model and optimizer. They used Modelica [8] to handle coupling between partitions and applied their method on an electric vehicle to optimize the electric motor and gearbox with one ratio. Nevertheless, these methods are complex to apply due to the requirement for reformulating the problem [153], hence they can lead to high computational time. A nonlinear multi-objective dynamic optimization method is presented in [154] to design the mechanism, control and servo drive subsystems. In their method, the objectives of the design are considered to be tracking error, manipulability measure and energy. Their method is only discussed to be applicable for robot manipulators.

A multidisciplinary design method is presented in [155] which uses particle swarm optimization algorithm and a fuzzy-based method to integrate the design criteria and objectives from different disciplines into a single performance index. They tested their approach on a vision-guided quad-rotor UAV system. A multidisciplinary method is presented in [156] which considers the aerodynamic and structural disciplines and optimizes a wing design considering minimization of the weight. In this method, a response surface model (RSM) was used to approximate the objective and constraint functions and to obtain the wing geometry model for aerodynamic and structural analysis, and parametric modeling was applied. Later, the structural optimization is employed by NASTRAN, and the aerodynamic flow field was performed using FLUENT.

Ivanov et al. [157] proposed a method based on interdisciplinary knowledge which focuses on environmental mechatronic applications within renewable energy production, smart grids and electric vehicles. Gulec et al. [158] proposed a nonlinear multidisciplinary method with a focus on the magnetic, thermal and structural issues of Axial-Flux Eddy Current Brakes (AF-ECB) with an aim to find the actual brake performance. They studied nonlinear magnetic modelling by 2D reluctance network approach and employed a modified Newton-Raphson method as a nonlinear solver. They considered braking torque, temperature in the disc, safety factor, and deflection of the disc as function of time in their proposed method. A multidisciplinary design approach is presented in [159] for designing a micro gyroscope which focuses on the coupling effects of fabrication process, structural parameters, environments and electronics.

2.4.2.6 Comments/discussion on the design optimization of mechatronic systems

Section 2.4.2.5 presented the most relevant studies in the area, however their focus is mainly on control design optimization, physical design or architecture/concept/topology design optimization for mechatronic systems. They cover different disciplines such as energy, thermal, structural, aerodynamic and control. However, concurrent integration of physical, control and embedded control implementation considering efficiency and performance is not found to be discussed in any of these papers. Moreover, the domain correlation and parameter couplings are not covered in the above research.

The above research integrates different methods of design such as sequential approach, optimization partitioning, the use of several optimization loops as well as complex methods such as finite element analysis (FEA), artificial immune system (AIS), simulated annealing, and fuzzy-based approaches which are computationally costly algorithms. Some detailed design aspects in these studies are considered, such as the influence of uncertainties on the optimized system, which is a good additional research aspect that we have not discussed for our method and can be considered for future research. Our method can also be extended to cover several other engineering domains which are frequently discussed in the literature, such as aerodynamic and energy, to broaden the use cases and application.

The problems formulated in these approaches are single or multi-objective problems. For the latter, the problem is usually converted to several single optimization problems to be solved where the correlation between disciplines, objectives and variables are lost. Furthermore, the models in these studies are often considered in a system level rather than in a component level, which does not allow for consideration of detailed design constraints and objectives, and limits the use cases since the application is merely defined to be on a specific system. Table 2.3 presents some of the research which either covers physical design (Phycl), control design (Cntrl) or embedded control implementation (Embd), where x indicates the inclusion of each discipline. Note that our method integrates these three domains, simultaneously, in a detailed component level design. A large number of studies in the literature are focused only on control or other disciplines which are not included in this table.

2.4.2.7 Mechatronics design software tools

In a multidisciplinary design method, use of modern software design tools with a systematic development approach is required. A frequently suggested solution is the use of tool integration software [160, 161]. This approach is mainly suitable for detailed design of systems and not for concept eval-

Table 2.3: Some recent methods in concurrent and integrated design of mechatronic systems

Method	Ref.	Phycl	Cntrl	Embd	Level of abstraction
Artificial bee colony	[24]	-	x	-	System level
Random search	[96]	x	x	-	System level
Simulated Annealing	[97]	x	x	-	System level
Evolutionary case-based	[103]	x	x	-	Component level
Real-time control	[108] [110]	- x	x -	x x	System level
Temporal spectral element	[109]	-	x	-	System level
Fractional-order adaptive control	[113]	-	x	-	System level
Nonlinear programming	[127]	x	x	-	Component level
Nonlinear programming	[128, 136]	x	x	-	System level
Response surface model	[156]	x	-	-	Component level

uation due to the evaluation time and the modeling effort needed. This becomes crucial when optimization and dynamics are involved in the design process, since a large amount of time-consuming evaluations of different system variations are required. Some commercial system engineering tools such as Modelica [162] and 20-Sim [163, 164] are used for model-driven development within several engineering domains. System models in these tools are generated by combining predefined blocks based on bond graph methods [165] to simulate dynamic behaviour [80]. Some of the papers in the literature which integrated tools in the design are presented as below.

Amerongen [166] discussed the need for tools for early stages of design methods which support simulation and modelling of physical systems including the controllers. They considered a few examples to show how a simulation software such as 20-sim, which is based on physical system models, is helpful in finding proper parameters of mechanical and control components. Subsequently, an energy-based object oriented modelling method

is presented in [25] to model physical systems, which allows reusable and extendable models. They used 20-sim as a modelling and simulation software which gives the possibility to have inputs in the form of iconic diagrams, equations, block diagrams and bond graphs. Their approach was tested on a design example including a flexible transmission and a mobile robot. For planning of a design process for mechatronics systems, distributed development processes were modelled and simulated using object-oriented Petri Nets in [167]. Becerra et al. [168] introduced PSOPT (open source optimal control solver written in C++), which is used for the solution of complex optimal control problems. They used an illustrative example to explain the features and capabilities of this software.

Chami et al. [169] proposed a method for intelligent conceptual design evaluation of mechatronic systems using SysML. Their method captures the interdisciplinary information in system engineering and generates a system design model using SysML. The models were mathematically formulated to satisfy a set of prioritized numerical requirements by non-parametric regression. Subsequently, a SysML based approach for evaluating the design concept of mechatronic systems is presented in [170]. They have integrated model based system engineering and artificial intelligence methods to acquire an efficient design method called SysDICE. A SysML based method for modeling and design of mechatronic packaging machine for Tetra Pak Packaging Solutions SpA is presented in [171]. Their method uses a hierarchy of models to describe the system at different levels of abstraction.

Delbecq et al. [172] presented a Python framework for the design of embedded mechatronic systems, in order to support the designer in satisfying constraints such as energy consumption, influencing the environment, geometrical integration and reliability. The dynamic simulation through 0D-1D models of the system is used to validate the architectural choices. Their optimization approach cannot be applied to the system level even during the early design phases. A UML (Unified Modelling Language) based method is proposed in [173] for conceptual design of mechatronic systems which uses major diagrams to represent the behaviour and structure of systems. This method is applied on a robot design example and the design of a complex mechatronic system which can be considered as an information system design by analysing specifications, functions and design structure.

A method to extend and integrate SysML with simulations in Simulink is proposed in [174] to model the complex hybrid discrete/continuous behaviour. Therefore a meta-model based method is used to integrate the system design and simulation models of mechatronic systems. Stereotypical definitions are formulated to model hybrid dynamic behaviour that is based on SysML, and the required information is used to support analysis of dynamic behaviour with simulations. However, most existing methods

employ SysML to specify the software aspects of mechatronic systems while the use of SysML in modeling mechanical subsystems is still an issue [175]. An object-oriented system integration framework is presented in [176] which was tested on a conceptual design of a plate making system for DNA microarray manufacturing automation. Their method considered cost and design optimization, and focused on system level integration rather than individual components.

2.4.2.8 Comments/discussion on the mechatronics design software tools

A large number of studies in mechatronic system design use 20-sim for their evaluations. However, 20-sim is a modeling and simulation program for only dynamic systems, although it can handle models in the form of equations, block diagrams, bond graphs and physical components. A tool called Petri Net [177] which is a mathematical modeling language is used for description of distributed systems. A number of studies use SysmML [178] which is a graphical modeling language for modeling system requirements, behavior and structure that is used to integrate with other engineering analysis models. SysML is known as an extension of the Unified Modeling Language (UML). UML [179] is usually used to visualize the design of a system.

A number of studies use Modelica to realize their method. Modelica [8] is an object-oriented, multi-domain modeling language for modeling systems which include mechanical, electrical, electronic, hydraulic, thermal, control, electric power or process-oriented components. CAMeL-View [180] supports interactive model based and object oriented design for mechatronic systems. Two types of connection are used in CAMel-View to connect graphical components: physical connect line and signal flow, which are used for hydraulic or mechanical components and electrical components, respectively. CAMeL-View supports modeling of mechatronic systems, however only the mechanical and control components are feasible to model and no embedded control implementation features or physical design constraints for physical component design are included. Ansys [5] is used to develop multiphysics simulation for product design and testing. The application of Ansys is to simulate computer models of structures, electronics or machine components for analyzing strength, toughness, elasticity, temperature distribution, electromagnetism, fluid flow and other attributes. It does not cover dynamics, control and embedded control implementation factors.

Most of these methods and tools are for design thinking or system level design of mechatronic systems and they do not support integration of dimension/scaling approaches/physical constraints for physical design of systems, neither do they include embedded control implementation impacts within

co-design optimization in the concurrent method. MATLAB in combination with SIMULINK [181] is a platform for technical computations, simulation and model based design in both mechanical and electrical domains. MATLAB is often used for algorithm development, data visualization, data analysis and numeric computation. SIMULINK is used for dynamic modeling of continuous time and discrete time systems graphically without any need for code [182]. Therefore, MATLAB is used as a basis for tool development and encoding algorithms in the previously published thesis [9] and as a continuation in this thesis.

2.4.3 Summary

The keywords for the literature study are selected from the research goals and research questions. The search for the keywords tried to cover most common synonyms in the research field. Three digital libraries, *Science Direct*, *Google Scholar*, and *IEEExplore* are considered which are important databases in the research area and include highly cited publications. *Science Direct* and *IEEExplore* are the most frequent libraries used in the review papers (e.g. [183]). Including a non-specific platform would influence the result of study; however, *Google Scholar* is one of the libraries that does not limit the search to a specific publisher. Limiting the studies to the electronically available articles after 1993 which are in English narrows the search; however, according to the research topic and a need for tools/software, up to date research is more appreciated in the field. Research older than 25 years risks being outdated.

As illustrated in the reported results of the literature study, prevalent design methods are usually based on sequential steps, where the physical system is designed first followed by a controller design. The principle in the sequential design process is that each new design task must be started when the previous one has ended. These methods do not allow flexibility for the design and if errors appear in the system, they require costly iterations. Therefore, in order to achieve a concurrent design which results in a true optimal solution, exploiting synergies between physical, control design and embedded control implementation decisions is required. The physical design, control, behavioural aspects and implementation factors are not found to be adequately dealt with (overall, simultaneously) in the discussed research in Section 2.4.2; and the impact of parameters from different engineering domains on the design optimization solution is not considered.

According to the literature so far, the main focus of research in the mechatronics design area is on conceptual design, concept development, topology and architectural design or control design optimization using complex and time consuming algorithms, methods and models. Moreover, when

talking about the design of mechatronic systems, the focus is mainly on dynamic behaviour aspects of the systems rather than static properties, and structural/physical design features (including scaling/dimensioning approaches/imposed physical constraints) and the integration with implementation facets. The concurrent aspect and simultaneous feature of the design is lacking. Dealing with multi-degrees of freedom nonlinear mechatronics systems considering the physical dimension, static properties, dynamic behaviour, and system performance which yields an optimum solution is a cumbersome task that is missing in the literature.

Some of the discussed research in the literature considers design of mechatronic systems with a focus on design management, design thinking, agile, cooperation and networking, information management, intelligence, and functionality in the design. The objectives of the design in these studies are mainly defined to be design philosophies such as safety, reliability, time, integrity, synergy and performance. A large number of studies are in multidisciplinary design fields with a focus on aerodynamic, energy, control, thermal or magnetic domains, however, they are only applicable for a particular mechatronic design case since the models are in a system level. Hence generality of the application is not found to be adequately dealt with in these studies. In our method, the component models are treated independently, which allows for configuring any type of system as long as the models are already developed. Domains coupling, parameters correlation, and early phase integration of mechanical, electrical, control, and embedded control domains as well as a study on the interaction between parameters from the involved disciplines are all missing in the literature. Objectives and constraints from the mentioned engineering domains are not integrated in an early-phase design, therefore the results are usually not balanced/optimal with respect to the employed disciplines; i.e. the true optimum solutions are not obtained.

Designing a mechatronic system which includes both mechanical and electrical properties and checks for dynamic responses involves constraints that need to be satisfied over a time interval where the objective functions should also be evaluated over the mentioned interval. Designing a constrained optimization efficiently is challenging, since the responses are implicitly correlated with the design variables. The main focus of this research is on early-phase integration of physical and control design optimization considering the embedded control implementation impact as well as capturing the correlation between the domains. To achieve this goal, physical dimension models, dynamic behaviour and static properties of the components are all developed. The design cases which represent real system design concepts are studied. The thesis covers modeling, optimization, control and design facets, and the problems that are dealt with are multidisciplinary design

problems.

Chapter 3

Proposed mechatronics system design method

According to the literature background and the gaps in the research area, a method for multidisciplinary co-design optimization of mechatronic systems is presented in this thesis which considers physical structure from mechanical and electrical domains, control, and implementation factors from control and embedded control domains, concurrently. Model based design and optimization allows us to take all domains into account simultaneously, since design variables are determined in parallel by optimizer and not sequentially. A library of re-usable component models allows quick modelling and saves time which assists in evaluating more concepts. Efficient models and straightforward modelling methods allow computational complexity to be kept at a reasonable level. The method results in an efficient optimum solution and the correlations between the domains are studied and highlighted to enable a qualification for a product design without increasing unnecessary iteration loops. Moreover, the impact of core parameters from different domains on the final system solution and performance is investigated.

Among the computer modeling approaches presented in the literature background (Section 2.4), symbolic methods allow building equations of motion in symbolic format, whilst, numerical methods produce the equations of motion as complex numerical procedures. On the other hand, symbolic format has an advantage of efficiency. Therefore in this thesis, for efficiency reasons, a tool called IDIOM (introduced in [9]) which is developed in MATLAB and integrated with Wolfram Mathematica [184] is extended to handle symbolic equations and later use the numerical feature of the mentioned software.

This tool is used to realize the multidisciplinary design optimization method for mechatronic systems. This chapter includes the proposed multidisciplinary method, developed component models, optimization method,

and the extended software toolbox in order to answer the research questions. Details are skipped and the reader is referred to the appended papers.

3.1 Conceptual design method

The method presented in this thesis includes two types of components: physical and control. Fig. 3.1 depicts a typical model of one physical and one control component in the software toolbox where IP_k and OP_k represent input and output sets of physical component k and they carry two types of signals: position and torque (detailed definitions are presented in appended Paper D). Each physical component (C_k) is associated with three main models—*physical dimension*, *static properties* and *dynamic behaviour*. The control component (C_n) represents the control law, dynamic constraints, cpc_γ, related to control performance, and includes embedded control implementation factors.

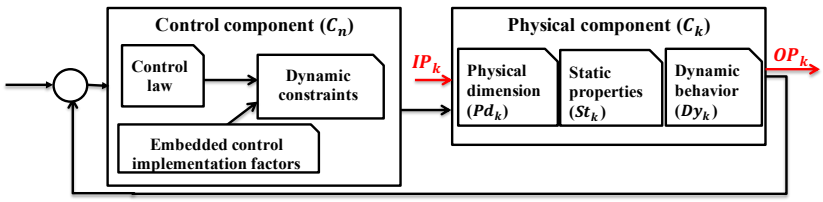

Figure 3.1: The structure of one physical and one control component

The *physical dimension* model (Pd_k) captures the load profile specifications of the component as a function of component properties such as size, mass, friction or cost with respect to the final optimization criteria. Using the physical dimension model, each component is designed based on already defined physical constraints, $cons_k$, (examples of these constraints are presented in Section 3.6). Depending on the component type, the physical dimension is a function of load properties such as RMS (root mean square) and peak values obtained from the given torque and motion profiles. For efficient model evaluation, an approach is adopted from [26] where the models are practically based upon algebraic relations and scaling approaches.

The *static properties* model (St_k) performs static transformation of the load from one side of a component to the other. These models transfer an external system requirement as a load through the chain of components in the system and are required for initial physical dimension models. The static properties model excludes the effect of damping and stiffness in the system. The *dynamic behavior* model (Dy_k) is related to the internal dynamics of a component. The system is designed based on a set of requirements

and specifications. Load specification (lps) is a major requirement on the system and is a position profile connected to an inertia/mass component which imposes constraints on the needed torque/force for the entire system. Fig. 3.2 shows a typical system requirement as a load profile.

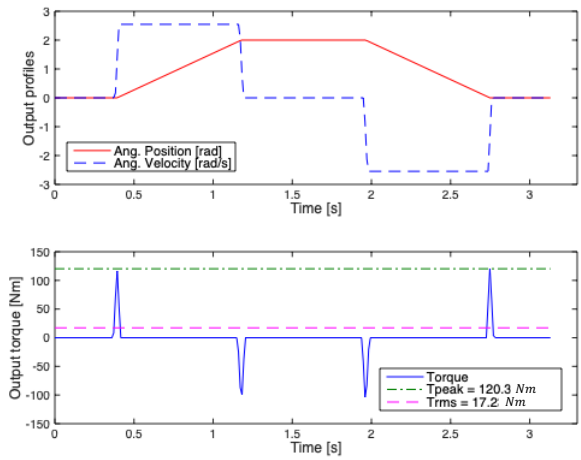

Figure 3.2: Trapezoidal (load) profile

3.1.1 Physical component

Physical components (C_k) refer to basic mechanical and—in some cases—electrical components that are used throughout development of the method. We have different types of physical components regarding the application, such as actuator components, structural components, and load components. Actuator components provide energy/power/torque to the mechanical system and carry the control signal. A structural component carries load and transforms position/torque throughout the system. Examples of structural components are transmissions, mechanical springs and robot arms. A load component is an inertia/mass body and is the last component in the open chain system. It carries the position profile requirement for the system.

3.1.2 Control component

The core capabilities of the method and tool discussed in this thesis are the integration of design and optimization which is performed holistically over a full closed loop system. Therefore consideration of controllers and control

related components is crucial in terms of component modelling. A control component (C_n) is specified by a parameterized control structure and it represents a control method that is designed to minimize the difference between desired output profile and actual system output for a particular implementation.

3.2 Multidisciplinary design

In order to integrate several engineering domains simultaneously, the need for a multidisciplinary method arises. Engineering a system is categorized as holistic and to derive an optimized system considering all design parameters is a challenging task that may be unfeasible even for domain engineers [185]. Thus a multidisciplinary design optimization (MDO) is a fundamental method to explore the solution space for design, and to obtain the final optimum system considering major disciplines concurrently [186]. Throughout this thesis, the system is optimized subject to different engineering disciplines shown in Fig. 3.3 to minimize the size, implementation cost and hardware cost of the systems.

Other disciplines and respective objectives and constraints are feasible to be included in the method as long as the models would be developed and integrated in the method and its toolbox. Dimensioning a system physically relates to a mechanical/electrical domain, control design is from a control domain and implementation aspects are from an embedded control implementation domain. Considering couplings, correlations and interactions between domains increases the complexity of the design. For multi-objective optimization problems, this complexity increases even more.

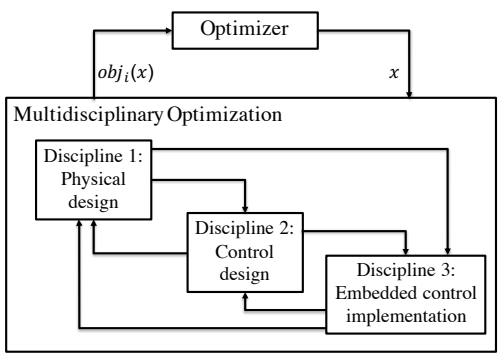

Figure 3.3: A multidisciplinary optimization method

3.3 System model

Given a set of component models, the system concept is configured in the proposed method. The physical dimension and static transfer models are used in component level analysis. However, dynamic behaviour models of each component with already defined interface equations are gathered to form a system model in a conceptual phase for holistic analysis and optimization purposes in a system level. The entire method from component libraries, system concept evaluation, requirements, system configuration through to deriving a final optimal system solution is depicted in Fig. 3.4. Details about evaluation of the static properties model and system model can be found in appended Paper E. An example of a configured system in the system configuration stage, as well as system requirements and the optimization process are visualized separately in Fig. 3.5, Fig. 3.6, and Fig. 3.7, respectively.

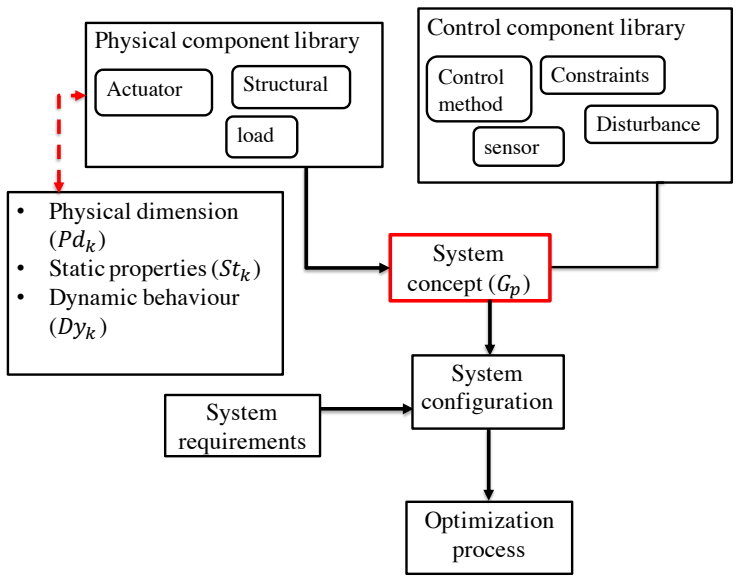

Figure 3.4: System modelling and evaluation

The system concept is a specific configuration of a mechatronic system which is composed using the library of physical and control components. The system model is developed for each concept separately. Different system concepts can be evaluated one at a time. For design analysis and system evaluation, a fixed system concept should be configured. System requirements include load and dynamics specifications, and a definition of

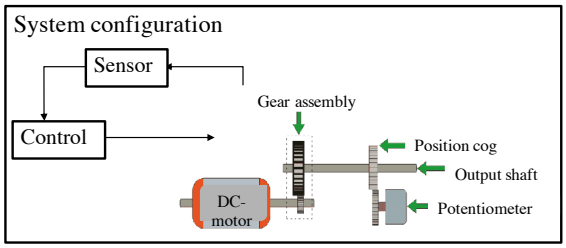

Figure 3.5: Example of a configured system

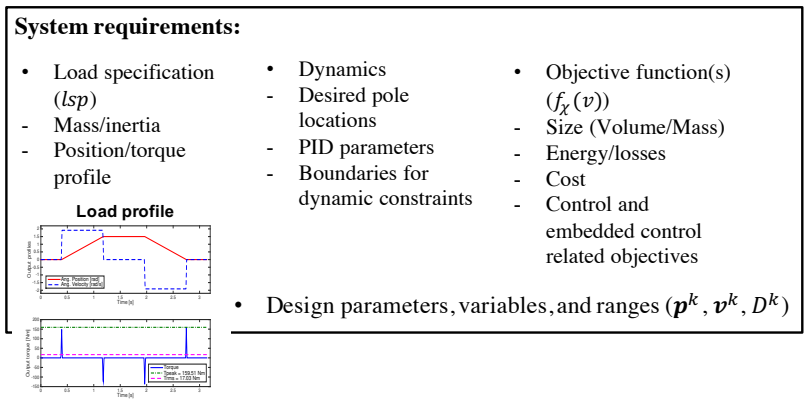

Figure 3.6: System requirements

objectives and design parameters/variables. Physical dimension and static properties models are evaluated within the optimization in component level to evaluate physical design constraints. Dynamic constraints are evaluated in system level to examine satisfaction of the boundary conditions.

The system model is capable of dealing with both linear and nonlinear components and system configurations. For the nonlinear case, the method uses defined fixed operating points (in the relevant state variables) and linearizes the equations around that point to achieve a linear local approximation (this is performed either in component level if the component dynamics are nonlinear, or in the system model evaluation stage if the configuration is nonlinear). Different operating points yield different system models, therefore they should be evaluated separately, e.g. for a system with multiple separated operating points, each operating point results in a new system model and is dealt with independently. The evaluated system is designed and analysed by the optimizer. The optimization is applicable

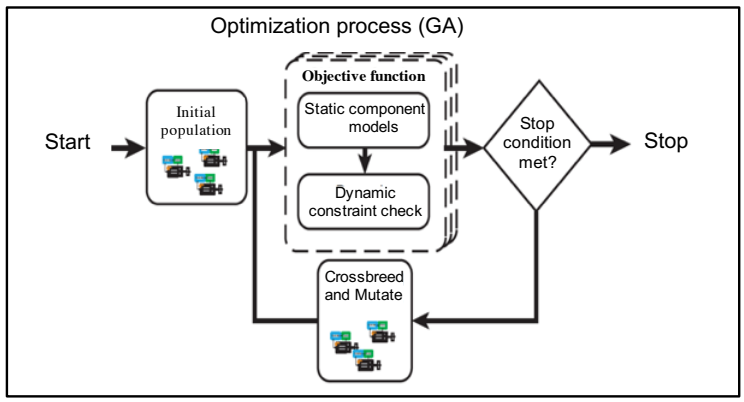

Figure 3.7: Optimization process

on both multi-objective and single objective problems.

3.3.1 Basic assumptions

For simplicity of the implementations and design, and to reduce complexity and computation time, some assumptions are made in the models as follows:

- Dynamic models are intended to be simple and computationally inexpensive, yet accurate enough to capture major component characteristics. The detailed assumptions related to the dynamic behaviour models are presented in models of each component in Section 3.6.
- The developed design method in this work targets early concept phases, hence a reasonable assumption is that it suffices to use controllers with low complexity to have less computationally demanding model evaluations (Section 3.7).
- Volume calculation for all physical components (including complex components such as ball-screws, 2-DOF arm and belt-drive) uses geometrical calculations by simplification of the shapes, e.g. the volume for a DC-motor is approximated as a cylinder.

Detailed assumptions related to the models are discussed in the relevant section.

3.3.2 Encoded algorithms

In order to evaluate each model and later on use them for optimization purposes, three algorithms are employed. The algorithms are presented for

an arbitrary system composition with K component models, and defined objective function(s), $f_\chi(v)$. Related detailed definitions and examples are presented in appended Paper D. The arbitrary system composition is equivalent to the configured system concept in Fig. 3.4. A chain in the algorithms refers to a serial assembly of connected components. Moreover, the algorithms assume a particular 'reading order' for the components and it would terminate when a defined class of component is read.

Algorithm 1 builds dynamic configuration of the component models. Hence linear/nonlinear dynamic differential equations of each physical component with the interface equations are configured to construct the open loop and closed loop models in terms of transfer functions or state space models. The inputs for Algorithm 1 are the system concept and the system attributes. The outputs of Algorithm 1 are the symbolic representation of the open loop and closed loop system models in terms of transfer functions/ state space models. This algorithm is executed in the chain between the two specified physical components as actuator ($\hat{a}$) and sensor ($\hat{s}$) components (the class of these components are defined as inputs, further details are presented in Section 3.8), see the pseudo-code in Algorithm 1.

- In line 1 and 2, when the algorithm starts, the transfer functions/ state space models and the equation list, Eq , are as yet empty sets, $\varnothing$.
- From lines 3 to 10, the dynamic behaviour model, Dy_k, of each component is read and added into a newly formed equations list (Eq).
- The interface equations between physical components are added to Eq if and only if k is not the sensor component; i.e. reading the internal dynamics and interface equations is performed until reaching the sensor component in the chain (the component models are defined as class methods which represent objects in the optimization run, details in Section 3.8).
- The equation list, Eq, is complete when k equals the sensor component.
- Wolfram Mathematica [184] is used in lines 11 and 12 and the Mathematica kernel is started and the symbolic Eq is sent from Matlab to Mathematica together with the control parameters/attributes. Wolfram Mathematica is integrated in order to symbolically manipulate equation systems.
- The open loop transfer functions/state space model is derived in Wolfram Mathematica (for a nonlinear system, the model is linearized in the mentioned software) and is sent back to MATLAB in line 13.

Algorithm 1 Dynamic System Configuration (DSC)

$[TF_{op}/SS_{op}, TF_{cl}, SS_{cl}]$=**DSC**$(G_p, \hat{a}, \hat{s}, \mathbf{c}_p)$

Require: system concept (G_p), actuator component $(\hat{a})$, sensor component $(\hat{s})$, control parameters $(\mathbf{c}_p)$;

Ensure: Symbolic Open Loop Transfer Function (TF_{op})/State Space Model (SS_{op}), Symbolic Closed Loop Transfer Function (TF_{cl})/State Space Model (SS_{cl});

1: $TF_{op}/SS_{op} = \varnothing$
2: $Eq = \varnothing$
3: **if** $k \in \{\hat{a}, ..., \hat{s}\}$ **then**
4: Read Dy_k
5: Write in Eq
6: **if** $k \neq \hat{s}$ **then**
7: Read interface equation $(E_{k,k+1})$
8: Add to Eq
9: **end if**
10: **end if**
11: Send Eq and $\mathbf{c}_p$ to Wolfram Mathematica
12: Obtain TF_{op}/SS_{op} by Wolfram Mathematica
13: Read TF_{op}/SS_{op} by MATLAB
14: Implement (C_n) method on TF_{op}/SS_{op}
15: Construct closed loop system (C_{cl})
16: Output TF_{cl}/SS_{cl}

- Control method, C_n, is applied on open loop system model in line 14.
- Symbolic closed loop system is constructed in line 15.
- The closed loop system model, in a form of transfer function/state space model, is outputted in line 16.

In Algorithm 2 physical dimension (Pd_k) and static properties (St_k) models are used to evaluate the size/energy/cost of each component. The evaluation process for this algorithm is adopted from [9]. The results of analysis in component level are the design answers $(\mathbf{As}^k)$ of each physical component. Since this algorithm is executed in a component level rather than system level, it is not visualized in Fig. 3.4. The design answers are used later to calculate the final objective(s). The inputs for Algorithm 2 are the system concept, the number of requirements (N_r), output profile $(op_K(t))$ of the load component (K) (depicted in Fig. 3.1), design parameters $(\mathbf{p}^k)$, design variables $(\mathbf{v}^k)$ and their ranges (D^k). The output is the

Algorithm 2 Component Level Static Model Evaluation (CLSME)

$\mathbf{As}^k$=**CLSME**(G_p,$op_K(t)$,$\mathbf{p}^k$,D^k,N_r,$\mathbf{v}^k$,lps)

Require: system concept (G_p), output profile ($op_K(t)$), design parameters ($\mathbf{p}^k$), design variables range (D^k), number of requirements (N_r), design variables ($\mathbf{v}^k$), load specifications (lps);

Ensure: design answers ($\mathbf{As}^k$);

Read $op_K(t)$ and lps
Execute St_K on $op_K(t)$
Calculate $ip_K(t)$
for $k = K-1:-1:1$ **do**
  $op_k(t) = ip_{k+1}(t)$
  Read $\mathbf{p}^k$, $\mathbf{v}^k$
  Compute Pd_k
  if $cons_k$ is true **then**
    Calculate As^k
    if $k \neq 1$ **then**
      Execute St_k and Calculate $ip_k(t)$
    end if
  else
    $\mathbf{As}^k = \varnothing$
    $ip_k(t) = \varnothing$
  end if
end for
Output $\mathbf{As}^k, k = 1, ..., K$

design answers ($\mathbf{As}^k$) of each physical component in terms of physical dimension model.

- In lines 1-3, the load profile ($op_K(t)$) is called and the static properties model (St_K) is executed to calculate the input profile ($ip_K(t)$) of the same component.
- In line 4, a for loop starts to go through the components from output to input side in order to execute the physical dimension (Pd_k) model using the output profile ($op_k(t)$).
- The loop starts from component $K-1$ in the open loop chain and ends at the first component which is generally the actuator component (component K is the load component which does not have a physical dimension model and the specification for this component is given).
- In line 5, the profiles on the input side of C_k are defined to be equal to the profiles on the output of C_{k-1} for configuration purposes.

- In lines 6 and 7 in the same loop, physical dimension model, Pd_k, is computed using $\mathbf{p}^k$ and $\mathbf{v}^k$.

- In lines 8-16, component k is designed in the component level considering physical design constraints ($cons_k$) (physical design constraints are explained in detail in Section 3.6, and are defined inside the physical dimension model (Pd_k) of each component).

- If the physical design constraints are satisfied, the design answers ($\mathbf{As}^k$) are calculated in line 9, otherwise they are defined to be empty ($\mathbf{As}^k = \varnothing$) in line 14.

- The static transfer model, St_k, is executed as long as the component is not the first component in the chain ($k \neq 1$): lines 10-12.

- The algorithm is completed when $\mathbf{As}^k$ is evaluated for all of the components in the loop.

Algorithm 3 checks for the satisfaction of the dynamic constraints (cpc_γ) and computes the optimization objectives for the system of physically designed components. Therefore the inputs for this Algorithm are the output of Algorithms 1 and 2.

- In line 2, Algorithm 1 is executed, hence the symbolic transfer functions (TF_{cl}) or state space models (SS_{cl}) are computed.

- In line 3, Algorithm 2 is executed to get the design answers ($\mathbf{As}^k$) of a particular system composition/configuration.

- If the design answers from Algorithm 2 are real values ($\mathbf{As}^k, k = 1, ..., K-1$), the numerical values of TF_{cl} or SS_{cl} in lines 5 and 6 are evaluated by re-using $\mathbf{As}^k$ from Algorithm 2, otherwise there is no optimum solution (the result is invalid).

- In lines 6-15 the dynamic constraints (cpc_γ) are calculated for γ number of constraints and the satisfaction of them are evaluated to be in a defined boundary, $\hat{b}_\gamma$, (these constraints evaluate control performance and are presented in detail in Section 3.7.2).

- If any of the dynamic constraints would not be met, the solution will be rejected (invalid final optimum solution).

- The algorithm is repeated until the optimization generation and population size are achieved: line 20.

Algorithm 3 Optimization Objective(s) Calculation (OOC)

$[\min_{\mathbf{v}^k \in \mathcal{D}^k} \sum_{j=1}^{\chi} f_j(\mathbf{v}^k),\ cpc_\gamma]$ =**OOC**$(TF_{cl}/SS_{cl}, \mathbf{As}^k)$

Require: TF_{cl}/SS_{cl}, $\mathbf{As}^k$;
Ensure: $\min_{\mathbf{v}^k \in \mathcal{D}^k} \sum_{j=1}^{\chi} f_j(\mathbf{v}^k)$;
1: **repeat**
2: $[TF_{op}/SS_{op}, TF_{cl}, SS_{cl}]$=**DSC**$(G, \hat{a}, \hat{s}, \mathbf{c}_p)$
3: $\mathbf{As}^k$=**CLSME**$(G, op_K(t), \mathbf{p}^k, D^k, N_r, \mathbf{v}^k)$
4: **if** $\mathbf{As}^k \in \mathbb{R}^{Q_k}$ **then**
5: Evaluate numerical TF_{cl}/SS_{cl}
6: **for** $j = 1 : \gamma$ **do**
7: Compute cpc_j
8: **end for**
9: **if** $cpc_j \leq \hat{b}_j$ $(\forall j = 1, ...\gamma)$ is true **then**
10: Compute numerical cpc_j
11: Compute $\sum_{n=1}^{\chi} f_n(\mathbf{v}^k)$
12: **else**
13: $cpc_j = \varnothing$
14: $f_n(\mathbf{v}^k) = invalid$ $(\forall n = 1, ..., \chi)$
15: **end if**
16: **else**
17: $cpc_j = \varnothing$
18: $f_n(\mathbf{v}^k) = invalid$ $(\forall n = 1, ..., \chi)$
19: **end if**
20: **until** (generation and population size for the optimizer are finished)
21: Normalize objectives
22: Evaluate $\min_{\mathbf{v}^k \in \mathcal{D}^k} \sum_{n=1}^{\chi} h_n \hat{f}_n(\mathbf{v}^k)$
23: Output $\min_{\mathbf{v}^k \in \mathcal{D}^k} \sum_{n=1}^{\chi} h_n \hat{f}_n(\mathbf{v}^k)$, cpc_γ

- In line 21, the optimization objective(s) are normalized to have even range between 0 and 1 (normalized objectives are denoted with $\hat{f}$), χ is the number of objectives.

- In line 22, a weighted sum function (presented in Section 3.5.3) is applied on the Pareto front [187] set with defined weights as h_n, and the final optimum solution is derived ($\min_{\mathbf{v}^k \in \mathcal{D}^k} \sum_{n=1}^{\chi} h_n \hat{f}_n(\mathbf{v}^k)$).

The algorithm is terminated with an optimal co-design solution for an IDIOM system concept.

3.4 Optimization

Genetic algorithm (GA) [188] is classified as metaheuristic optimization which is applicable on problems with highly nonlinear objectives including discrete design variables. GA is an approach to solve both constrained and unconstrained optimization problems based on a natural selection process that mimics biological evolution. Evolutionary methods such as GA are well suited for mechatronic design optimization due to their ability to obtain solutions for a diverse range of complex problems. Therefore the optimization that is used in this work and implemented in IDIOM is the genetic algorithm (GA).

GA is a method that has its roots in biology and makes use of Darwin's principal of survival of the fittest. In this evolutionary algorithm (EA), individuals that are possible solutions are bred by relying on mutation, crossover and selection steps. The evolution starts with a randomly generated population of individuals and is an iterative process with the population that is called generation inside each iteration. The fittest individuals are selected for reproduction in order to produce offspring for the next generation. The offspring inherit the characteristics of the parents and will be added to the next generation. If parents have good fitness, the offspring will be fitter than their parents and have a better chance at surviving. This process keeps on iterating and at the end, a generation with the fittest individuals will be found [189]. Five phases are considered in a genetic algorithm:

- *Initial population:* a set of individuals that are chosen randomly. Each individual is a solution to the optimization problem that is to be solved.
- *Fitness function:* determines the fitness of each individual where the probability for an individual to be selected for reproduction is based on it.
- *Selection:* two pairs of individuals as parents are selected based on their fitness.
- *Crossover:* a new offspring is generated and added using exchange of the genes between parents until the randomly selected crossover point is reached.
- *Mutation:* in a newly created offspring, some of the genes can be mutated to maintain diversity within the population.

When the population does not produce offspring which are significantly different from the previous generation, the optimization algorithm terminates and provides a set of solutions for the problem.

The method in this thesis treats mechanical, electrical, control and embedded control implementation domains with multiple design objectives as well as constraints in a concurrent manner. Thus the evaluation of different concepts with this method should be simple and quick. Therefore, due to the multidisciplinary nature and consideration of objectives/constraints from different domains, a multi-objective optimization is employed. Such a problem is generally defined as a set of $m > 1$ conflicting objective functions, obj_i, which are maximized [190]. The maximization is to avoid loss of generality and for the case of minimization we have, $min\{obj(x)\} = -max\{-obj(x)\}$. Such a multi-objective problem is defined as given in (3.1).

$$\begin{aligned} \max\ & obj(x) = [obj_1(x), obj_2(x), ..., obj_m(x)] \\ s.t.\ & g_n(x) \leq 0;\ \ n = 1, ..., N \end{aligned} \tag{3.1}$$

where m is the number of objectives, N is the number of inequality constraints and x is a vector of design variables. Multi-objective optimization involves minimizing/maximizing multiple objective functions subject to a set of constraints. In this regard, analyzing design trade-offs, selecting optimal product or processing designs where you need an optimal solution with trade-offs between two or more conflicting objectives need to be considered. This is taken care of by the proposed method.

3.5 Multi-objective design

To name an approach as a multidisciplinary method, it should include specifications, methods and algorithms from each domain separately and integrate them as a whole. An optimization method that is able to integrate physical and control design, together with embedded control implementation aspects is a multidisciplinary method and is likely to lead to better engineering results. A multidisciplinary design method can be formulated as a single or multi-objective problem. In the method presented in this thesis, multiple objectives and constraints from the mentioned disciplines are considered and imposed on the system. The system is optimized for three objectives as volume, sampling frequency and sensor resolution to reduce the size, implementation cost and hardware cost. In practical problems, the method can handle more than three objectives. As an example, the system can be optimized for energy/losses, and other control related objectives, if the corresponding models would be integrated in the method and implemented in the tool.

3.5.1 Physical design objective

For physical design of a system, the objective can be size (mass/volume), energy/losses and cost. Throughout this thesis, the size (volume/mass) of the entire system has been considered as a main mechanical/electrical domain objective. Some examples of volume are as follows:

- *DC-motor volume:* The volume for DC-motor is approximated as a cylinder,
$$v_m = \pi r_m^2 l_m \tag{3.2}$$
where r_m and l_m are the motor radius and length, respectively.

- *Gear volume:* As for the motor, the gearbox volume is approximated as,
$$v_g = \pi r_{\mathrm{g}}^2 b \tag{3.3}$$
where b and r_g are the total width of the gear and outer radius of the gear, respectively.

- *Shaft volume:*
$$v_{sh} = \pi r_{sh}^2 l_{sh} \tag{3.4}$$
where r_{sh} and l_{sh} are the shaft radius and length, respectively.

3.5.2 Control design and implementation objectives

In previous studies [142, 144, 145, 154, 156], a simplification is considered while designing mechatronic systems and that is to neglect the influence of embedded control implementation on the performance of the final controlled system. In control design and embedded control implementation, sampling period choice, communication delays and sensor resolution have significant impact on performance. An improper decision on these implementation related parameters causes the system to be unstable or deteriorates the control performance [7]. Hence, to consider these discussions, sampling frequency and sensor resolution are considered as control objectives of the optimization that require the system to satisfy extra control performance related constraints (appended Paper C). In this method, the control design is applied on a continuous time system and the closed loop system is later discretized using calculated sampling time. Thus, a proper selection of sampling time is a basis for this method. Obviously, a *shorter* sampling time yields a better performance (closer to the corresponding continuous time system). On the other hand, a *longer* sampling time yields lower computational load and may reduce numerical challenges. To address this

trade-off issue, the sampling frequency is defined as one of the optimization objectives and the sampling time is derived using (3.5) [191],

$$T_s = \frac{2\pi}{c_b w_b} \tag{3.5}$$

where w_b is the fastest frequency of the closed loop system and c_b is a frequency coefficient that is defined to be a real valued optimization parameter in the range of $[2, \ldots, 30]$. The defined range for c_b is related to the sampling rates. A factor of 2 is generally a limit posed for signal reconstruction, but it is too small for control purposes. Therefore considering dynamic constraints (Section 3.7.2) would simply take care of this (which is more tangible in the obtained design solutions in appended Papers A, B and C). For economical reasons, sampling rates should be kept as low as possible: a low rate leaves more time for execution of a control algorithm, and it can be carried out on slower computers. On the other hand, if the sampling frequency is too low, the systems may become unstable. The choice of sampling rate depends on the fastest pole of the closed loop system. According to Shannon's law, the sampling rate must be at least two times faster than the fastest closed loop pole [192]. The sampling frequency of the system is derived in (3.6),

$$w_s = \frac{2\pi}{T_s} \tag{3.6}$$

where T_s is variant according to (3.5). To prevent an aliasing effect in sampling signals, it is necessary to set a constraint on the sampling time, T_s. Aliasing is an effect of the sampling that causes signals to become indistinguishable. Due to aliasing, the signal that is reconstructed from samples may be different than the original continuous time signal. This can deteriorate the performance if proper care is not taken. Considering this fact and Nyquist–Shannon sampling theorem, sampling time should be in a defined range in (3.7) [193].

$$5w_c \leq \frac{2\pi}{T_s} \leq 100w_c \tag{3.7}$$

where w_c is the desired bandwidth of the continuous time closed loop system. Although defining the sampling frequency, w_s, as a second objective function reduces computational costs, a too low sampling frequency degrades the control performance. To avoid this problem, control performance constraint(s) are considered. Integrated square error, ISE, and maximum error, $max(er)$, are defined as dynamic constraints on the optimization problem (Section 3.7.2).

Since the implementations are usually performed by digital computers, the quantization and sensor resolution should be considered in embedded control implementation. To the best of our knowledge, this factor has not

been considered in previous integrated design optimization methods for mechatronic systems [21, 22, 136, 137, 145, 187]. To model the quantization, $floor$ function is used which quantizes the value of input for the sensor [194].

$$y_{s,out} = \lfloor \frac{y_{s,in}}{\delta} \rfloor \delta \tag{3.8}$$

where $y_{s,out}$ and $y_{s,in}$ are the output and input signals of the sensor, respectively. Sensor resolution factor, δ, is in $\frac{rad}{pulse}$ which indicates revolution. This sensor in principle acts as a rotary encoder; a sensor with an application of a linear encoder is implementable in the method as long as the model would be developed and integrated). Smaller δ means higher resolution. As δ increases, the system that we are trying to control becomes unstable. As known, resolution is a primary specification in control design. Without enough resolution, the necessary measurement might provide a too large quantization error while an over performing sensor is an increased cost. It is important to note that resolution is different from accuracy. Sensor resolution is meaningful within the context of system bandwidth, the application and the measurement method.

3.5.3 Weighted-sum method

In a multidisciplinary design method, a single point that optimizes all the objectives together does not normally exist. Pareto optimality is the result of multi-objective optimization problems to extract solutions. In the Pareto-optimal set, it is not possible to improve one objective without deteriorating the others. A typical solution of Pareto-optimization is a compromise. Without additional subjective preference, all Pareto optimal solutions are considered equally good. Consequently, in order to find a single solution that satisfies the subjective preference, there is a need to have a proper definition of a function that yields a reasonable single optimal solution for all of the objectives. Hence a weighted sum function is defined as given in (3.9). A weighted sum function method aggregates the objective values to a single quality measure [195, 196].

$$f = \ h_1\ obj_1 + h_2\ obj_2 + ... + h_n\ obj_n \tag{3.9}$$

where h_1, h_2, ..., h_n are the Pareto front weight factors and have a relationship presented in (3.10).

$$h_1 + h_2 + ... + h_n = 1 \tag{3.10}$$

Considering the definition of weight factors, if h_1, h_2, ..., h_n are positive, minimizing f yields a single optimum solution as a Pareto-optimality so-

lution [197]. However, it is still difficult to decide the weights to indicate the relative importance of each objective due to the difference in objectives magnitudes. One proper method to have even weight factors is to normalize [198] the objectives to have identical range between 0 and 1. This is done using (3.11) and (3.12) for minimizing and maximizing objectives, respectively.

$$obj_i^{norm} = \frac{max(obj_i) - obj_i}{max(obj_i) - min(obj_i)} \tag{3.11}$$

$$obj_i^{norm} = 1 - \frac{max(obj_i) - obj_i}{max(obj_i) - min(obj_i)} \tag{3.12}$$

where i is the numeric of total number (n) of objectives, obj.

3.6 Physical components examples

Some of the physical components developed during this PhD project are presented in this section. These models extend the method to cover discussions for non-linear multi-degree of freedom systems and are implemented in the IDIOM software toolbox. The models for DC-motor and gearbox components are adopted from [9, 20] and are used in the design cases. As shown in Fig. 3.4, each physical component consists of three models: *physical dimension*, *static properties*, and *dynamic behaviour* models. The physical dimension model of each component includes internal physical design constraints. In this Section, the models with the constraints for each component are presented. Detailed dynamic models for components are usually complex and costly to evaluate. In many cases, to evaluate performance in early stages of design, it is possible to use reduced order linear models, based on multiple mass oscillators [199]. The models use the indices *in* and *out* to denote the input and output of a component.

Physical design of the system includes component level physical constraints such as speed requirements, torque/power measurements, strength and dimensioning and scaling factors. As an example, there is a constraint on the maximum allowed speed of the DC-motor, $w_{m,peak}$, which should be higher than or equal to the speed specification of the load, $\dot{\phi}_l$, [20].

$$w_{m,peak} \geq \max \mid \dot{\phi}_l \mid n \tag{3.13}$$

3.6.1 Ball-screw drive

Physical design of the ball-screw considers the peak-load force and the allowed stress of the material. During the execution of models and optimization, the peak transferred load force and proposed pitch by optimization are

used to select suitable ball-screw out of given SKF standard tables [200]. For the dynamic behaviour model, the screw is considered as two inertial bodies as $J_{bs,1}$ and $J_{bs,2}$ which are functions of the nut's position ($y_{n,out}$) and the length of screw (l_{bs}) from static specifications. The parameters used in (3.14)-(3.18) for modelling the ball-screw are shown in Fig. 3.8.

$$J_{bs,1}\ddot{\phi}_{bs,in} = T_{bs,in} - k_{bs}(\phi_{bs,in} - \theta_{bs}) - d_{bs}(\dot{\phi}_{bs,in} - \dot{\theta}_{bs}) \tag{3.14}$$

$$J_{bs,2}\ddot{\theta}_{bs} = k_{bs}(\phi_{bs,in} - \theta_{bs}) + d_{bs}(\dot{\phi}_{bs,in} - \dot{\theta}_{bs}) - T_{bs,d} \tag{3.15}$$

$$M_n\ddot{y}_{n,out} = F_{bs,d} - F_{bs,out} \tag{3.16}$$

we have,

$$T_{bs,d} = F_{bs,d}(\frac{p_{bs}}{2\pi\eta}) \tag{3.17}$$

$$\theta_{bs} = y_{n,out}(\frac{2\pi}{p_{bs}}) \tag{3.18}$$

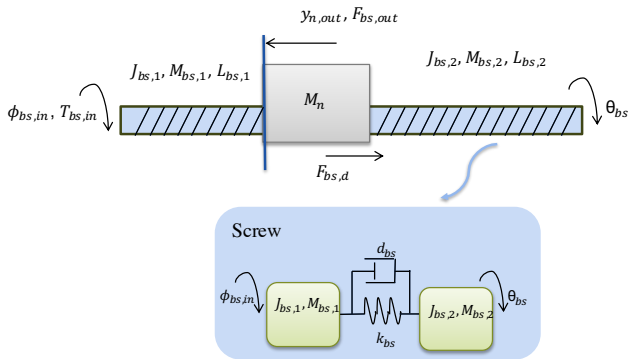

Figure 3.8: Ball-screw drive for dynamic modelling

where k_{bs} is the torsional stiffness of the screw, d_{bs} is the effect of friction damping (material damping is not considered), M_n is the mass of nut and $F_{bs,d}$ is the drive force between the nut and the screw which is proportional to the drive torque ($T_{bs,d}$) by the conversion ratio as given in (3.17). η is the normal efficiency ($0.9 - 0.95$) and p_{bs} is the pitch of screw. It should be noted that the thread friction is neglected in the above model. Indexes bs, n and d are for the ball-screw, the nut and the drive force, respectively.

3.6.2 Rotational rigid beam

A beam is a rotating rigid body with a distributed mass (M_b) and the length of L_b. The dynamics of the beam are derived from a configuration shown in Fig. 3.9. The beam is modelled as it is driven by a DC-motor. DC-motor 2 which drives the beam is along the z-axis (the rotor's rotational angle is γ). The assumption for modelling is that the beam is initially oriented in the y direction. The model is part of a design case presented in Section 4.1 and appended Paper E.

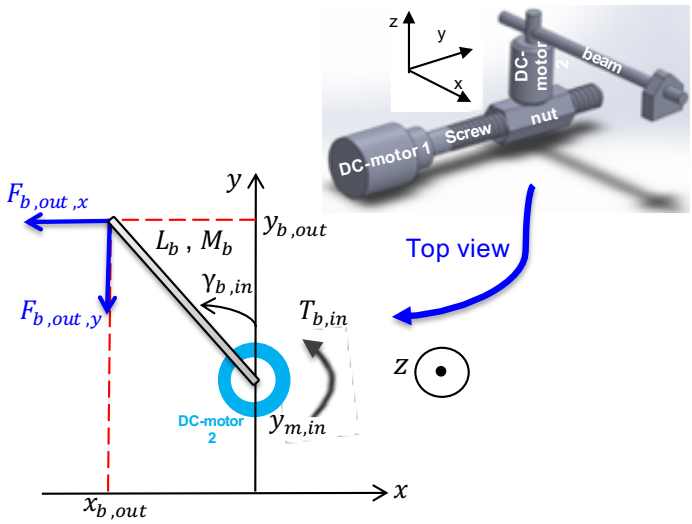

Figure 3.9: Dynamics of a rotational beam component (top view) that is driven by a DC-motor around Z axis.

$$T_{b,in} + L_b((\frac{M_b \ddot{y}_{m,in} \sin(\gamma_{b,in})}{2}) + F_{b,out,y} \sin(\gamma_{b,in}) + F_{b,out,x} \cos(\gamma_{b,in})) = J_b \ddot{\gamma}_{b,in} \tag{3.19}$$

$$F_{b,in,y} - F_{b,out,y} = M_b(\frac{\ddot{y}_{b,out} + \ddot{y}_{m,in}}{2}) \tag{3.20}$$

$$F_{b,in,x} - F_{b,out,x} = M_b(\frac{\ddot{x}_{b,out} + \ddot{x}_{m,in}}{2}) \tag{3.21}$$

$$x_{b,out} = -L_b \sin(\gamma_{b,in}) \tag{3.22}$$

$$y_{b,out} = y_{m,in} + L_b \cos(\gamma_{b,in}) \tag{3.23}$$

Since it is assumed that initial orientation of the beam is in the y direction, the inertial force effect caused by the translational acceleration of the motor is shown by the term $\frac{M_b \ddot{y}_{m,in} \sin(\gamma_{b,in})}{2}$ in (3.19). $T_{b,in}$ is the input torque to the beam by the motor. $F_{b,out,y}$ and $F_{b,out,x}$ are the forces resulting from the load mass by configuring equations at the end of the beam in two directions. (3.19)-(3.21) show that the beam is rotating around the rotor of DC-motor.

3.6.3 2-DOF arm

The physical constraint on the 2-DOF arm for design purposes requires the strength of the two arms to be larger than the required force (for simplicity of the design analysis, bending effect is neglected). The equation is derived from Hamrock et al. [201],

$$max(\frac{F}{A}) \leq \tau_m \tag{3.24}$$

where F is the force orthogonal to the cross sectional area and is derived from the output force signals on the 2-DOF arm, A is the cross sectional area and τ_m is the maximal allowed stress of the arm material.

By extending the IDIOM toolbox library to cover complex component models, some major changes must be performed in the configuration of components. For instance, since a 2-DOF arm component is considered as one whole physical component, a multi-input (multi-port) feature is enabled in the graphical interface. Statics and dynamics of the 2-DOF arm are presented below. The angular positions of the arms are derived from inverse kinematics [202]. For angular position of the second arm, θ_2, we have,

$$\theta_2 = \cos^{-1}(\frac{x_{a,out}^2 + y_{a,out}^2 - l_1^2 - l_2^2}{2 l_1 l_2}) \tag{3.25}$$

For the first arm's angular position, θ_1, we have

$$\theta_1 = \tan^{-1}(\frac{y_{a,out}}{x_{a,out}}) - \tan^{-1}(\frac{l_2 \sin(\theta_2)}{l_1 + l_2 cos(\theta_2)}) \tag{3.26}$$

We employ Lagrangian mechanics to derive the relationship between the input torque and the output position. Let x_i, y_i $(i = 1, 2)$ be the position coordinates of the tips of the two arms as illustrated in Fig. **??**.

$$x_1 = l_1 \cos(\theta_1), x_2 = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) \tag{3.27}$$

$$y_1 = l_1 \sin(\theta_1), y_2 = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) \tag{3.28}$$

To avoid complicated dynamics, we assume that the mass of each arm is concentrated as a point at the tip. The kinetic and potential energies of the two arms are as follows

$$E_{k_i} = \frac{1}{2} m_i (\dot{x}_i^2 + \dot{y}_i^2) \tag{3.29}$$

E_{k_i} $(i = 1, 2)$ are the kinetic energies of the first and second arms, respectively. The total kinetic energy of the 2-DOF arm is obtained from (3.29) and defined as

$$E_k = E_{k_1} + E_{k_2} \tag{3.30}$$

The potential energies are

$$E_{u_i} = m_i g y_i \tag{3.31}$$

The potential energy [203] of the robot arm is the sum of the potential energies of arm1 and arm2 which is obtained as follows

$$E_u = E_{u_1} + E_{u_2} \tag{3.32}$$

The Lagrangian [204, 205] of the robot arm is given in (3.33).

$$L = E_k - E_u \tag{3.33}$$

where for the torques on the joints of the two arms, $T_{\theta 1,in}$ and $T_{\theta 2,in}$, we have

$$\begin{gathered} T_{\theta 1,in} = \frac{\partial}{\partial t}\left(\frac{\partial L}{\partial \dot{\theta}_1}\right) - \left(\frac{\partial L}{\partial \theta_1}\right) = \\ m_1 \ddot{\theta}_1 l_1^2 + m_2 \ddot{\theta}_1 l_1^2 + m_2 \ddot{\theta}_1 l_2^2 + m_2 \ddot{\theta}_2 l_2^2 + 2 m_2 \ddot{\theta}_1 l_1 l_2 \cos(\theta_2) \\ -2 m_2 \dot{\theta}_1 \dot{\theta}_2 l_1 l_2 \sin(\theta 2) + m_2 \ddot{\theta}_2 l_1 l_2 \cos(\theta_2) - m_2 \dot{\theta}_2^2 l_1 l_2 \sin(\theta_2) \\ + m_2 g l_1 \cos(\theta_1) + m_2 g l_2 \cos(\theta_1 + \theta_2) \end{gathered} \tag{3.34}$$

$$\begin{gathered} T_{\theta 2,in} = \frac{\partial}{\partial t}\left(\frac{\partial L}{\partial \dot{\theta}_2}\right) - \left(\frac{\partial L}{\partial \theta_2}\right) = \\ m_2 \ddot{\theta}_1 l_2^2 + m_2 \ddot{\theta}_1 l_2^2 + m_2 \ddot{\theta}_1 l_1 l_2 \cos(\theta_2) - m_2 \dot{\theta}_1 \dot{\theta}_2 l_1 l_2 \sin(\theta_2) + m_2 g l_2 \cos(\theta_1 + \theta_2) \\ + m_2 l_1 l_2 \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) \sin(\theta_2) \end{gathered} \tag{3.35}$$

These torques are equal to the output torques from the DC-motors. Hereupon, in the design phase, the static properties model considers the maximal values of the torque/force throughout the chain of components and uses them to determine the physical dimension of each component.

3.6.4 Timing belt

The physical design model for timing belts considers designing a feasible timing belt structure with a suggested driving pulley's radius as optimization variables. The arc of the belt contact for driving and driven pulleys as shown in Fig. 3.10 are α_1 and α_2, respectively, and are calculated using (3.36) and (3.37).

$$\alpha_1 = \pi - 2sin^{-1}(\tfrac{r_2 - r_1}{C}) \tag{3.36}$$

$$\alpha_2 = \pi + 2sin^{-1}(\tfrac{r_2 - r_1}{C}) \tag{3.37}$$

where r_1 and r_2 are the pitch radii of the driving and driven pulleys. The length of the two arcs corresponding to α_1 and α_2 are given as

$$arc_1 = 2\pi r_1(\tfrac{\alpha_1}{360}) \tag{3.38}$$

$$arc_2 = 2\pi r_2(\tfrac{\alpha_2}{360}) \tag{3.39}$$

For the length of belt, L_b, we have

$$L_b = 2\sqrt{C^2 - (r_2 - r_1)^2} + arc_1 + arc_2 \tag{3.40}$$

where C is the center to center distance between the two pulley [206]. The length of belt segments s_l is l_c (shown in Fig. 3.10), that is used in dynamic behaviour model and is derived using:

$$l_c = \tfrac{L_b - arc_1 - arc_2}{2} \tag{3.41}$$

A selected timing belt should be able to handle the pre-tension force, F_0, that is calculated by the dynamic equations. As depicted in Fig. 3.10, F_1 and F_2 are the belt tensions both on the slack and tight side, and we have

$$F_2 - F_1 = F_{out} \tag{3.42}$$

Dynamics of the timing belt are adopted from [207]

$$J_{p_1}\ddot{\theta}_1 = T_{v,in} + r_1(F_1 - F_2) \tag{3.43}$$

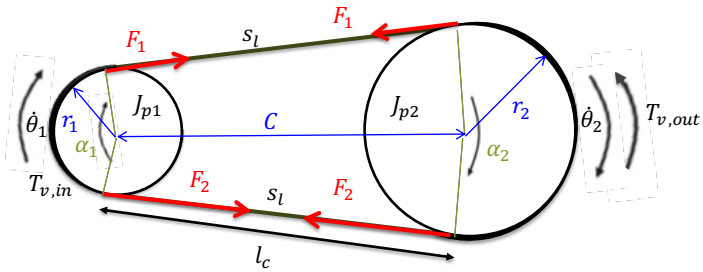

Figure 3.10: Timing belt drive

$$J_{p_2}\ddot{\theta}_2 = T_{v,out} - r_2(F_1 - F_2) \tag{3.44}$$

$$F_1 = F_0 + \frac{EA+F_0}{l_c - r_1\theta_1 + r_2\theta_2}(r_1\theta_1 - r_2\theta_2) \tag{3.45}$$

$$F_2 = F_0 + \frac{EA+F_0}{l_c - r_2\theta_2 + r_1\theta_1}(r_2\theta_2 - r_1\theta_1) \tag{3.46}$$

where J_{p_1}, J_{p_2}, θ_1 and θ_2 are the driving and driven pulley inertias and angular positions, respectively. $T_{v,in}$ and $T_{v,out}$ are the input and output torques to and from the pulleys. A is the cross sectional area of the belt, and E is the Young's modulus of the material. By substituting (3.45) and (3.46) in (3.43) and (3.44), we get

$$J_{p_1}\ddot{\theta}_1 = T_{v,in} + r_1(EA + F_0)\left(\frac{r_2\theta_2 - r_1\theta_1}{l_c - r_2\theta_2 + r_1\theta_1} - \frac{r_1\theta_1 - r_2\theta_2}{l_c - r_1\theta_1 + r_2\theta_2}\right) \tag{3.47}$$

$$J_{p_2}\ddot{\theta}_2 = r_2(EA + F_0)\left(\frac{r_1\theta_1 - r_2\theta_2}{l_c - r_1\theta_1 + r_2\theta_2} - \frac{r_2\theta_2 - r_1\theta_1}{l_c - r_2\theta_2 + r_1\theta_1}\right) + T_{v,out} \tag{3.48}$$

Considering a nonlinear function, $\dot{x} = f(x(t), u(t))$, an equilibrium point, $\hat{x}$, an equilibrium input, $\hat{u}$, and deviation variables, $\delta_x(t) = x(t) - \hat{x}$, $\delta_u(t) = u(t) - \hat{u}$, and by using Jacobian Linearization method (presented in (3.49)), we linearize the nonlinear model in (3.47) and (3.48) at defined operating points equal to $\theta_{1_0} = 0$ and $\theta_{2_0} = 0$, which results in (3.50) and (3.51).

$$\dot{\delta}_x(t) = \left.\frac{\partial f}{\partial x}\right|_{\substack{x=\hat{x}\\u=\hat{u}}} \delta_x(t) + \left.\frac{\partial f}{\partial u}\right|_{\substack{x=\hat{x}\\u=\hat{u}}} \delta_u(t) \tag{3.49}$$

$$J_{p_1}\ddot{\theta}_1 = T_{v,in} - \frac{2r_1^2(EA+F_0)\theta_1}{l_c} + \frac{2r_1r_2(EA+F_0)\theta_2}{l_c} \tag{3.50}$$

$$J_{p_2}\ddot{\theta}_2 = T_{v,out} + \frac{2r_1r_2(EA+F_0)\theta_1}{l_c} - \frac{2r_2^2(EA+F_0)\theta_2}{l_c} \tag{3.51}$$

By substituting (3.45) and (3.46) in (3.42), we derive the following expression for F_{out},

$$F_{out} = (EA + F_0)\frac{2l_c(r_2\theta_2 - r_1\theta_1)}{l_c^2 - (r_1\theta_1 - r_2\theta_2)^2} \tag{3.52}$$

Solving (3.52) for $(r_1\theta_1 - r_2\theta_2)$ we have

$$r_1\theta_1 - r_2\theta_2 = \frac{l_c(EA+F_0) + l_c\sqrt{(EA+F_0)^2 + F_{out}^2}}{F_{out}} \tag{3.53}$$

To get rid of angles in the definition of forces, we substitute (3.53) in (3.45) and (3.46), and we obtain

$$F_1 = \frac{F_0 - F_{out} - EA - \sqrt{(EA+F_0)^2 + F_{out}^2}}{2} \tag{3.54}$$

$$F_2 = \frac{F_0 + F_{out} - EA - \sqrt{(EA+F_0)^2 + F_{out}^2}}{2} \tag{3.55}$$

The required minimum cross section is determined by

$$A_{1,2} = \frac{F_{1,2}}{\sigma_y} \tag{3.56}$$

where σ_y is the allowable stress of the belt, where we have $\sigma_y = \sigma\epsilon$. σ is the yield stress of the belt and ϵ is a safety factor. By solving each equation for the area by Wolfram Mathematica solver, we have

$$A_1 = \frac{4(\sigma_y+E)F_0 - 2F_{out}(2\sigma_y+E) + \sqrt{(-4F_0(\sigma_y+E) + 2F_{out}(2\sigma_y+E))^2 + 32F_0F_{out}\sigma_y(\sigma_y+E)}}{8\sigma_y(E+\sigma_y)} \tag{3.57}$$

$$A_2 = \frac{4(\sigma_y+E)F_0 - 2F_{out}(2\sigma_y+E) + \sqrt{(-4F_0(\sigma_y+E) + 2F_{out}(2\sigma_y+E))^2 - 32F_0F_{out}\sigma_y(\sigma_y+E)}}{8\sigma_y(E+\sigma_y)} \tag{3.58}$$

These two equations are used in design analysis, but since σ_y is very small in comparison with E, for simpler computation, we can consider, $\sigma_y + E \approx E$. The required area of belt is defined to be the maximum of areas derived in (3.57) and (3.58)

$$A_{req} = \max([A_1, A_2]) \tag{3.59}$$

The Young's modulus is used and the belt material chosen is polyester. The actual belt area is adjusted based on a fraction of the cross section area

that the belt's tension member constitutes. Belt-drive dimensions such as volume and stiffness are derived based on the geometry of the belt.

Compared to the nonlinear belt-drive system, a simplified linear model is also considered where the span effect as a modeled spring on the belt stiffness is neglected. Therefore the model for stiffness is assumed to be $F = EA\frac{\Delta l}{L_b}$. Accordingly, simplified dynamics of the belt-drive are as follows

$$F_1 = F_0 + EA\frac{\Delta l_1}{L_b} \tag{3.60}$$

$$F_2 = F_0 + EA\frac{\Delta l_2}{L_b} \tag{3.61}$$

where we have

$$\Delta l_1 = r_1\theta_1 - r_2\theta_2 \tag{3.62}$$

$$\Delta l_2 = r_2\theta_2 - r_1\theta_1 \tag{3.63}$$

By substituting (3.60)-(3.63) in (3.43) and (3.44), we get

$$J_{p_1}\ddot{\theta}_1 = T_{v,in} + 2r_1\frac{EA}{L_b}(r_2\theta_2 - r_1\theta_1) \tag{3.64}$$

$$J_{p_2}\ddot{\theta}_2 = 2r_2\frac{EA}{L_b}(r_1\theta_1 - r_2\theta_2) + T_{v,out} \tag{3.65}$$

This system is already linear and there is no need to apply linearization algorithms on it.

To configure mechatronic system case studies throughout this thesis, some physical components developed in previous studies are used (described as follows).

3.6.5 DC-motor

The DC-motor is an actuator component. The models for DC-motors are adopted from scaling approaches presented by Roos et al. [26] which specify the relationship between the rated torque and the actual root mean square torque as in (3.66). In many machine data sheets the peak torque rating is given as a multiplier of the rated torque. The rated torque is equal to the required torque. In order to avoid overheating, the continuous torque rating of the machine has to be higher than the root mean square (rms) of the machine torque.

$$T_{m,rated} \geq T_{rms} \tag{3.66}$$

where $T_{m,rated}$ and T_{rms} are the rated torque, and the root mean square torque, respectively. The rated torque is derived based on the mechanical, magnetic and thermal effects [20].

$$T_{m,rated} = C_m l_m r_m^{2.5} \tag{3.67}$$

where C_m is a motor type constant for the same cooling conditions (natural convection at an ambient temperature of $25°C$), l_m is the motor's rotor length and r_m is the radius of the stator. The motor's RMS torque is derived as

$$T_{rms} = \sqrt{\frac{1}{\tau}\int_0^{\tau}((C_{mj} l_m r_m^4 + J_m)\ddot{\phi}_{m,out} + T_{m,out})^2 dt} \tag{3.68}$$

where C_{mj} is a constant for a specific machine type and is derived from a reference motor of the same type, τ is the cycle time of one load cycle, i.e. the time period for which the load profile is valid, $\ddot{\phi}_{m,out}$ is the angular acceleration of the rotor. $T_{m,out}$ is the output torque of the motor and J_m is the inertia of motor's shaft and bearings. Combining (3.66)-(3.68) results in (3.69).

$$C_m l_m r_m^{2.5} \geq \sqrt{\tfrac{1}{\tau}\int_0^{\tau}((C_{mj} l_m r_m^4 + J_m)\ddot{\phi}_{m,out} + T_{m,out})^2 dt} \tag{3.69}$$

The dynamic model of a DC-motor can be described as follows

$$K_T\, i = T_{m,out} + J_m \ddot{\phi}_m \tag{3.70}$$

where K_T is the torque constant dependent on the motor type and the reference motor according to the scaling approach presented by Roos [26], i is the current which is considered as actuation/control signal acting on the dynamic input port. $T_{m,out}$ and $\ddot{\phi}_m$ are torque and angular acceleration of the DC-motor and J_m is the total rotor inertia.

3.6.6 Planetary gear

The gearbox design model is adopted from [20]. The main constraints on gears are bending stress in the root of gear teeth and Hertzian pressure on the teeth contact surfaces. If the number of sun gear teeth is small and/or the gear is made from ductile steel, the Hertzian pressure is the constraining factor for the design of the gears. Every tooth surface needs to fulfill (3.71),

$$r_{\mathrm{gr}}^2\, b \geq Z_{\mathrm{H}}^2 Z_{\mathrm{M}}^2 Z_\epsilon K_{\mathrm{H}\alpha} K_{\mathrm{H}\beta} \frac{T_{g,out}(n-1)^2}{6(n-2)\sigma_{\mathrm{H,max}}^2} \tag{3.71}$$

where Z_H, Z_M, Z_ϵ, $K_{H\alpha}$ and $K_{H\beta}$ are parameters with minor impact on the system and therefore can be considered independent of the gear ratio, n. r_g, r_{gr} and b are the outer gear radius, pitch radius of the ring gear and gear total width, respectively, shown in Fig. 3.11. $\sigma_{\mathrm{H,max}}$ is the maximum allowed flank pressure and is given by (3.72).

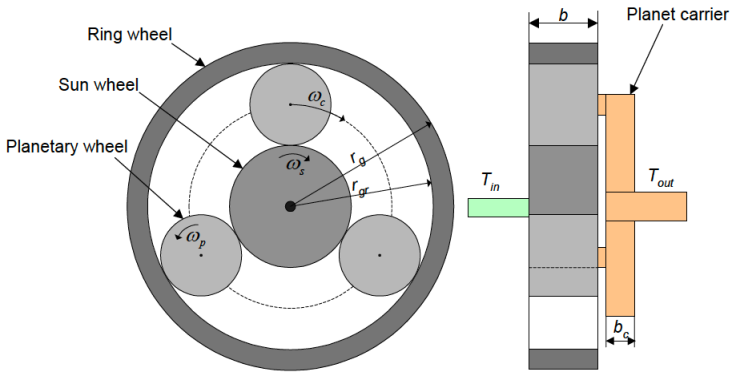

Figure 3.11: Schematic of a three-wheel planetary gear [20]

$$\sigma_{\mathrm{H,max}} = \frac{\sigma_{\mathrm{H,lim}}}{SF} \tag{3.72}$$

where $\sigma_{\mathrm{H,lim}}$ is the maximum allowed Hertzian pressure for the particular gear material and SF is the safety factor. Depending on the shape of the load profile, the safety factor may be set to 1.

There is one more equality constraint that needs to be fulfilled [20]:

$$b_c = 0.5b \tag{3.73}$$

where b_c is the carrier width.

In the dynamic model of the gearbox, the inertia and gear ratio are considered and other factors such as friction and stiffness are neglected.

$$(T_{pg,in} - J_{pg}\ddot{\phi}_{pg,out})n = T_{pg,out} \tag{3.74}$$

$$\phi_{pg,in} = n\phi_{pg,out} \tag{3.75}$$

where $T_{pg,in}$ and $T_{pg,out}$ are the input and output torques of the gearbox. J_{pg} is the combined inertia, from all rotating members in the planetary gear, felt at the output shaft, and $\phi_{pg,out}$ is the angular position.

3.6.7 Flexible shaft

The physical dimension model for the shaft is from classic solid mechanics.

- The shaft is only dimensioned for torsion (bending is neglected)
- The shaft is only dimensioned for yield by shear stresses (fatigue is not included)

The following equations describe shaft properties [23]

$$r_{sh} = \sqrt[3]{\frac{2\hat{T}_{sh,out}}{\tau_{sh,max}\pi}} \tag{3.76}$$

$$J_{sh,a} = \frac{\pi r_{sh}^4}{2} \tag{3.77}$$

$$G = \frac{E}{2(1+\nu)} \tag{3.78}$$

$$k_{sh} = \frac{GJ_{sh}}{l_{sh}} \tag{3.79}$$

where r_{sh} is the shaft radius, $J_{sh,a}$ is the polar second moment of area (torsional constant), $\hat{T}_{sh,out}$ is the peak transferred torque, $\tau_{sh,max}$ is the maximum permissible shear stress, G is the shear modulus, E is the Young's modulus, ν is the Poisson's ratio, and k_{sh} is the shaft stiffness [208]. The shaft dynamics are modeled as all the inertia is at the output side.

$$\begin{gathered} T_{sh,in} = k_{sh}(\phi_{sh,in} - \phi_{sh,out}) - \\ d_{sh}(\dot{\phi}_{sh,in} - \dot{\phi}_{sh,out}) \end{gathered} \tag{3.80}$$

$$\begin{gathered} J_{sh}\ddot{\phi}_{sh,out} = k_{sh}(\phi_{sh,in} - \phi_{sh,out}) + \\ d_{sh}(\dot{\phi}_{sh,in} - \dot{\phi}_{sh,out}) - T_{sh,out} \end{gathered} \tag{3.81}$$

where J_{sh} and d_{sh} are the moment of inertia and damping factors of the shaft. T_{sh} and ϕ_{sh} are the torque and angular position, respectively.

3.6.8 Load component

The geometrical specifications of the load component are defined by the user. Hence it includes no physical dimension model. The dynamics of this component are defined considering the load as rotational or translational. In rotational case we have

$$J_l\ddot{\phi}_{l,in} = T_l \tag{3.82}$$

$$\phi_{l,in} = \phi_{l,out} \tag{3.83}$$

where J_l is the load inertia, $\phi_{l,in}$ and $\phi_{l,out}$ are the input and output load position profiles and T_l is the load torque.

For translational load we have

$$M_l \ddot{x}_{l,in} = F_l \tag{3.84}$$

$$x_{l,in} = x_{l,out} \tag{3.85}$$

where M_l is the load mass, $x_{l,in}$ and $x_{l,out}$ are the input and output load position profiles, and F_l is the force acting on the load.

3.7 Control components examples

This section presents a number of examples that are considered as control components in the IDIOM software toolbox.

3.7.1 Control methods

A few control methods are implemented in the presented IDIOM software toolbox. Here is a brief overview and structure of them.

3.7.1.1 Optimal PID control

PID controllers are widely used in industry to provide optimal and robust performance for stable, unstable and nonlinear processes. In this thesis, PID control is implemented with a low-pass filter to remove high frequency noises caused by the derivative part. The optimization is used to tune and re-tune the PID controller and the low pass filter parameters for a class of time-delayed unstable systems. When applying optimal PID control on a system, the aim is to search the optimal controller parameters by minimising the cost function. The dynamic constraint criterion can be considered as the cost function, which guides the method to get the optimised controller parameters. The transfer function for PID control is given in (3.86).

$$G_c = k_p + \frac{k_d s}{s + r_0} + \frac{k_i}{s} \tag{3.86}$$

where k_p, k_d and k_i are the proportional, derivative and integral gains, respectively. r_0 is a factor of low pass filter.

This control method can be applied as an error feedback or output feedback structure on the plant system (depicted in Fig. 3.12).

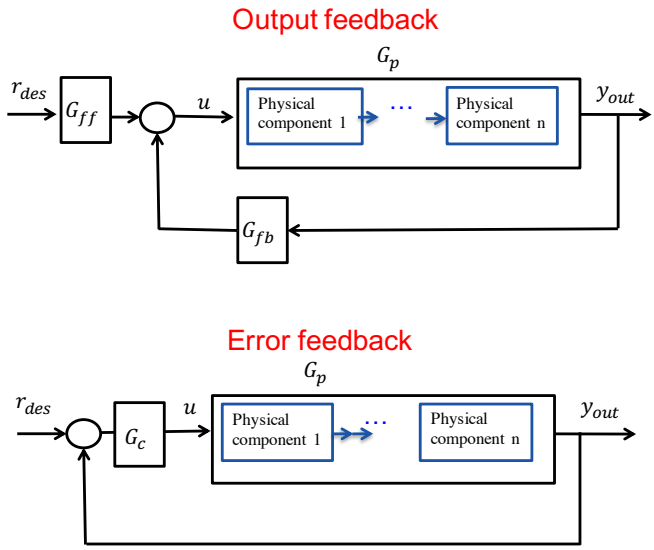

Figure 3.12: Structure of the control method

For output feedback and error feedback, the control signal, u, is derived using (3.87) and (3.88), respectively.

$$u = G_{ff} r_{des} - G_{fb} y_{out} \tag{3.87}$$

$$u = G_c(r_{des} - y_{out}) \tag{3.88}$$

where r_{des} is the desired reference signal and y_{out} is the controlled measured/output signal. If $G_{ff} = G_{fb}$, the two structures are equivalent. In some cases, PID control cannot produce a stable feedback loop due to the presence of poles in the right-hand side of the s-plane, or because the plant does not have sufficient stabilizing coefficients of the characteristic polynomial. Therefore defining the desired pole locations for the controlled system by pole placement control method assures the stability.

3.7.1.2 Pole placement control

The desired pole locations are specified by user to assure the stability of the closed loop system and the pole placement method is applied on the open loop dynamics as shown in Fig. 3.13. The control signal, u, is derived using (3.89).

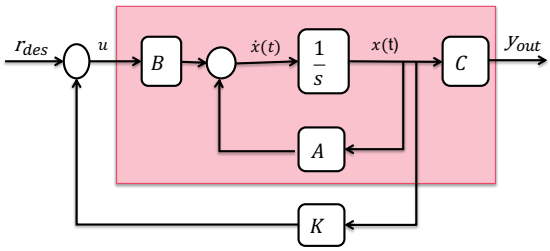

Figure 3.13: Pole placement control

$$u = r_{des} - Kx(t) \tag{3.89}$$

We assume that the physical system is described by a linear time invariant model or is linearized at specified operating points. Hence, if the open loop system state matrices are A, B, C, and D, and the open loop system dynamics are as presented in (3.112),

$$\begin{aligned} \dot{\mathbf{x}}(t) &= A\mathbf{x}(t) + B\mathbf{u}(t) \\ \mathbf{y}_{out}(t) &= C\mathbf{x}(t) + D\mathbf{u}(t) \end{aligned} \tag{3.90}$$

then the closed loop state matrices are $A_{cl} = A - BK$, $B_{cl} = B$, $C_{cl} = C$, $D_{cl} = D$ (for all physicall systems, $D = 0$), and for the states we have

$$\dot{\mathbf{x}}(t) = (A - BK)\mathbf{x}(t) + B\mathbf{r}_{des}(t) \tag{3.91}$$

3.7.1.3 Full state feedback control

The full state feedback control is adopted from an approach presented by Åström and Wittenmark [209] and is derived using a pole placement control. This control has different feedback and feed-forward transfer functions as below

$$G_{fb}(s) = \frac{S(s)}{R(s)} = \frac{S_n s^n + ... + S_0}{s^n + r_{n-1}s^{n-1} + ... + r_0} \tag{3.92}$$

where S_i and r_i are the control parameters and n is the number of plant poles. The closed loop polynomial, A_{cl}, and the desired closed loop polynomials, A_d, are as follows

$$A_{cl}(s) = A(s)\ R(s) + B(s)\ S(s) \tag{3.93}$$

$$A_d(s) = A_m(s) + A_0(s) \tag{3.94}$$

where A and B are the plant's transfer function's denominator and nominator, respectively, and A_m is a desired polynomial that has the same order as A, and A_0 has an order equal to $order(A_{cl}) - order(A_m)$. To derive feedback control parameters, S and R, we solve (3.95)

$$A_{cl}(s) = A_d(s) \tag{3.95}$$

The feed-forward control is

$$G_{ff} = \frac{T(s)}{R(s)} \tag{3.96}$$

where $T(s) = t_0 A_0(s)$, and $t_0 = \frac{A_m(0)}{B(0)}$.

3.7.2 Dynamic constraints

The control parameters, depending on the control method, are either specified by desired close loop pole locations or are defined to be design variables to satisfy control performance constraints such as maximum error $(max(er))$/integrated squared error (ISE) or overshoot and response time.

$$ISE = \sqrt{\int_0^T (\theta_d(t) - \theta(t))^2 dt} \tag{3.97}$$

$$max(er_i) = max(\theta_d(t) - \theta(t)) \tag{3.98}$$

where θ_d is the desired position, and θ is the actual controlled and sensed output position.

3.7.3 Sensor component

A sensor component is defined to sense and measure the output variable of a specific physical component. A sensor component can be defined in the context of resolution and can be a finite or infinite gain, or it can be defined as a transfer function. The resolution can be used as an objective to be optimized. The sensor is usually regarded as a black box where the relationships between its output and input signals matter. This relationship may be a simple linear connection or a nonlinear dependence, (e.g. logarithmic, exponential or power function). In many cases, the relationship is uni-dimensional (i.e. the output versus one input) [210].

$$q(x) = a + bx \tag{3.99}$$

where a is the intercept (i.e. the output signal at zero input signal) and b is the slope, which is called sensitivity. x is the sensor input and $q(x)$

is the sensor output which, depending on the sensor characteristic, can be amplitude, frequency or phase. In some cases, sensor function is not a simple function and is a logarithmic, exponential and power function, as follows

$$q(x) = a + b \ln x \tag{3.100}$$

$$q(x) = ae^{mx} \tag{3.101}$$

$$q(x) = a + bx^m \tag{3.102}$$

where m is a constant.

Modelling the sensor as quantization is another method to define the sensor function. In this regard the resolution is an important factor that is used to quantify sensed input, and therefore the value and specified location for the sensor both have significant impact on the control performance.

$$q(x_i) = \delta_i \left\lfloor \frac{x_i}{\delta_i} \right\rfloor \tag{3.103}$$

where $q(x)$ and x are the output and input signals of the sensor, respectively. δ_i represents the sensor resolution in $\frac{rad}{pulse}$. Lower δ means a high resolution of sensor and is an expensive hardware.

3.7.3.1 State feedback pole placement control

To apply the sensor model in the system using state feedback control, the output of a plant system or the state that is to be quantized should be first sensed by the sensor and then controlled as shown in Fig. 3.14. In this

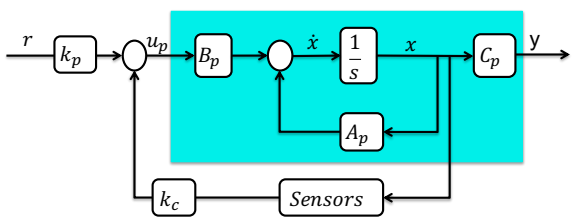

Figure 3.14: Pole placement control

system, the LTI model of the physical system is as follows

$$\begin{aligned} \dot{x} &= A_p x + B_p u_p \\ y &= C_p x \end{aligned} \tag{3.104}$$

The sensor can be located anywhere in the component chain to quantize the output of a defined physical component. It is then fed to the full-state feedback control. The control system is derived as a gain, k_c, and the control output is $y_c = k_c u_c$, where sensor output is $u_c = [q(x_1); x_2; x_3; ..., x_n]$ for the case when sensor quantization is applied on the first state and the control signal is $u_p = k_p r - y_c$. By simplifying further, we derive

$$u_p = k_p r - k_c \begin{bmatrix} q(x_1) \\ x_2 \\ x_3 \\ ... \\ x_n \end{bmatrix} \tag{3.105}$$

Substituting equivalent of u_p in (3.110), the closed loop system is derived as

$$\begin{aligned} \dot{x} &= A_p x + B_p(k_p r - k_c x - k_c \begin{bmatrix} q(x_1) - x_1 \\ 0 \\ 0 \\ ... \\ 0 \end{bmatrix}) \\ y &= c_p x \end{aligned} \tag{3.106}$$

Considering a function $\psi(x)$ as in [211]

$$\begin{aligned} \psi(x) : \mathbb{R}^n &\to \mathbb{R}^n \\ x &\mapsto q(x) - x \end{aligned} \tag{3.107}$$

for every $x \in \mathbb{R}^n$, and $i = 1, 2, ..., n$, we have

$$\mid q(x_i) - x_i \mid \leq \delta_i \tag{3.108}$$

where $q(x_i) - x_i$ is a time variant vector which, at this stage, we do not have information about, so in order to avoid unnecessary simplifications of the method by applying algorithms that reduce the impact of δ, we let Simulink solve the above model. Therefore a Simulink model is integrated inside the constraint function in the optimization algorithm. The only drawback with this is that the system execution time is increased, albeit at a reasonable level to solve a multidisciplinary problem.

The presented above closed loop system is when the sensor quantization is applied on the first state. By applying quantization on the second state, we derive the closed loop system as

$$\dot{x} = A_p x + B_p(k_p r - k_c x - k_c \begin{bmatrix} 0 \\ q(x_2) - x_2 \\ 0 \\ 0 \end{bmatrix}) \qquad (3.109)$$
$$y = C_p x$$

The sensor quantization in estimating states acts as sensor measurement noise and disturbs the tracking performance of the reference, therefore a proportional feed-forward control gain, k_p, that is defined to be an optimization variable, is applied to assist in reducing the rise time and steady state error.

3.7.3.2 Observer-based control

In order to apply an observer on the system to estimate the states, the sensor quantization model should be applied on the output of the system. An observer-based optimal pole placement control is designed by defining desired pole locations for both the observer and state controller as optimization variables. The structure of the controller is depicted in Figure 3.15. The system is continuous-time with a quantized output,

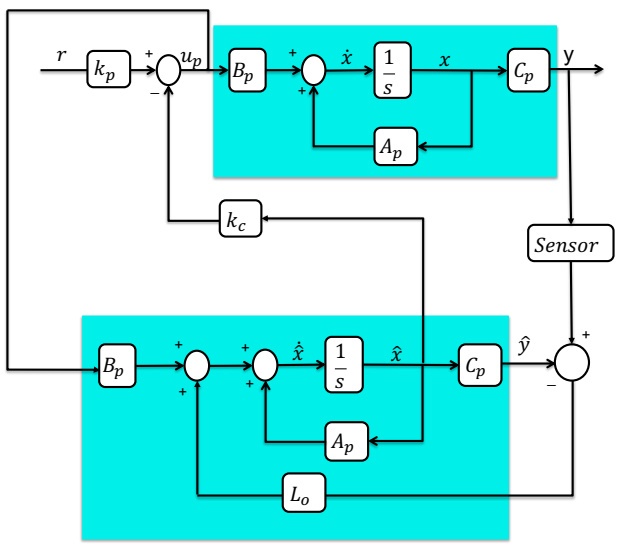

Figure 3.15: Observer-based pole placement control

$$\dot{x} = A_p x + B_p u_p \qquad (3.110)$$
$$y = q(C_p x)$$

The control system is derived as a gain, k_c. We assume that the system given in (3.110) is controlled with a dynamic output feedback control of the form in (3.111).

$$\begin{aligned} \dot{\hat{x}} &= A_p\hat{x} + B_p u + L_o(y - C_p\hat{x}) \\ u_p &= k_p\, r - k_c\, \hat{x} \end{aligned} \tag{3.111}$$

Tracking a non-constant reference is called a servomechanism problem; therefore we added a feed forward gain, k_p, to assist the tracking performance. This system structure is based on a Luenberger observer [212], and the state $\hat{x}$ is considered as an estimate of the state x. Therefore, the closed-loop system can be written as

$$\begin{aligned} \dot{x} &= A_p x + B_p u \\ \dot{\hat{x}} &= (A_p - B_p k_c - L_o C_p)\hat{x} + B_p k_p r + L_o q(C_p x) \end{aligned} \tag{3.112}$$

3.8 IDIOM software toolbox

The presented method in this thesis is supported by a software toolbox which is implemented in MATLAB with symbolic evaluation of the dynamic systems performed by Wolfram Mathematica solver. The algorithms described in Section 3.3.2 are encoded in MATLAB. The software toolbox was developed in a previously published PhD thesis [9], and here in this thesis is extended to cover more complex problems and engineering gaps (such as multi-DOF systems with nonlinear dynamics) and multidisciplinary problems with the impact of embedded control implementation on the co-design. The IDIOM software toolbox is visualized in Fig. 3.16, where the component libraries, load specifications, optimization and system attributes are visualized.

The IDIOM software toolbox uses an object oriented programming paradigm [9]. The component models are defined as class methods which represent objects in optimization run. When the optimization begins, the software toolbox checks for errors and missing points of the configured system (sanity check). After this, dynamic and static preparation functions of the components are executed to determine the plant model and load the necessary data into the memory. In this stage, parallel pool and workers are employed to acquire best use of a multi-core computer. The multi-objective function is evaluated by the algorithms mentioned in this chapter, and physical dimension, static properties and dynamic behaviour models are executed in

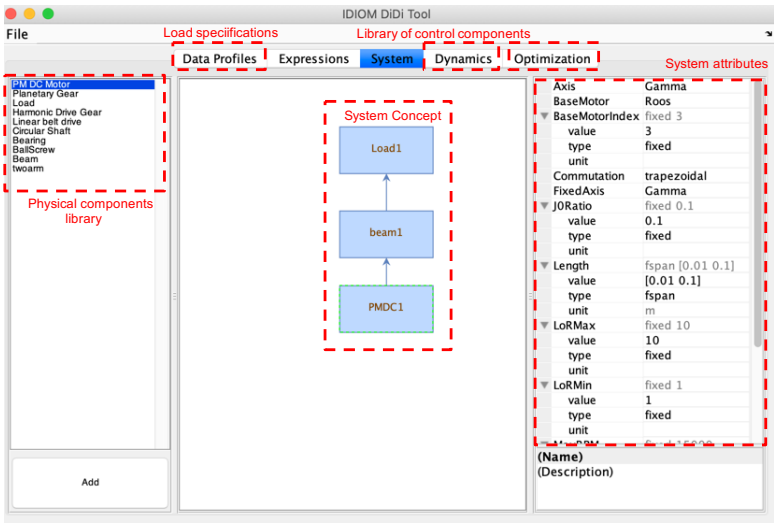

Figure 3.16: IDIOM software toolbox

order to obtain a final optimum solution for a conceptual system. The tool has a graphical user interface, GUI, (depicted in Fig. 3.16), which allows the user to define the motion profiles used as load specifications. The procedure of system configuration in IDIOM is as follows:

1. Earlier mentioned physical and control components required for system concept configuration are dragged, dropped and interconnected from the library of components.

2. Components are connected manually to define the interconnections and interface equations for static transformation and dynamic configuration.

3. System specifications and attributes, such as requirement on the system in the form of a load profile, are defined.

4. Component parameters such as material and dimensions are specified. The design parameters can be either fixed or defined as optimization variables. Design variables can be either a floating point span, integer span or an enumerated set of possible values.

5. Dynamic constraints and their boundaries are specified.

6. The internal optimization parameters (such as tolerance function,

number of generations, the stall generation limit) and the objectives are specified.

Once the system is configured, the optimization begins to derive solutions for a defined multidisciplinary problem. During the optimization loop, the optimizer generates a number of individuals as proposed solutions, and compares them with each other. The objective function(s) is used to evaluate each individual solution. This goes through a few steps [9]:

- Instantiate a system with its components and defined design variables and parameters.
- Evaluate the static transfer model's fitness according to the process described in Algorithm 2.
- Physical dimension models of the components are evaluated in component level, and if the system does not give valid static design answer values (presented in Algorithm 2), then it is discarded.
- The implemented dynamic programming approach prevents unnecessary model evaluations, i.e. the model for a specific component is generated and stored, and if the same component is used more than once, the same information does not need to be reevaluated.
- Dynamic model of the entire system is evaluated symbolically in a system level (Algorithm 1).
- Dynamic constraints (Algorithm 3) are evaluated by replacing symbolic transfer function constants with numerical values for the plant model as well as for the controller.
- If dynamic constraints are not satisfied, the solution is discarded.
- Otherwise, the objectives are derived.
- Final optimum system solution is evaluated.

The method presented in this chapter, unless the models are integrated in the tools, is not implementable using other tools/software, since they do not support dimension/scaling approaches for physical design of the systems, nor do they include embedded control implementation impacts in the co-design method. The core feature of this method and tool is to integrate static and dynamic modelling, optimization, physical design, control design and embedded control implementation factors all together in an early phase of development, concurrently.

Chapter 4

Design cases

The method presented in the previous chapter can be applied on any system configuration which uses already developed models (see Sections 3.6 and 3.7) in the method and toolbox. This chapter briefly presents the design cases that are used to develop the design method and extend the software toolbox features, and covers important gaps in the studies on integrated multidisciplinary design optimization of mechatronic systems. The design cases are modelled and implemented in the toolbox and are verified dynamically in Simulink/Simscape and geometrically in Solidworks. The designs are complex problems since the systems consist of several components with multi-DOF configurations, with several design variables, and include nonlinear dynamics, thus representing real system configurations.

4.1 Ball-screw drive system

In order to apply the method on this case study, the library of components should be advanced to cover physical components: ball-screw drive with a nut and nonlinear beam. The ball-screw is a high-efficiency mechanism that converts rotary motion to translational motion. The ball screw assemblies are made of a ball screw shaft and a nut with ball bearings and end supports. Circulating balls move inside the thread of the screw and nut, and balls are recirculated through return mechanisms. These balls transmit the force between the threads. Without a return mechanism, the balls fall out of the end of the ball nut when they reach the end of the nut. For this reason several different recirculation methods have been developed. Fig. 4.1 depicts a ball-screw assembly. The ball-screw, compared to a conventional sliding screw, requires driving torque of one-third or less, making it ideal for saving drive motor power [214]. The application of ball-screw assemblies in industry is to move heavy objects, such as an axis within a CNC machine,

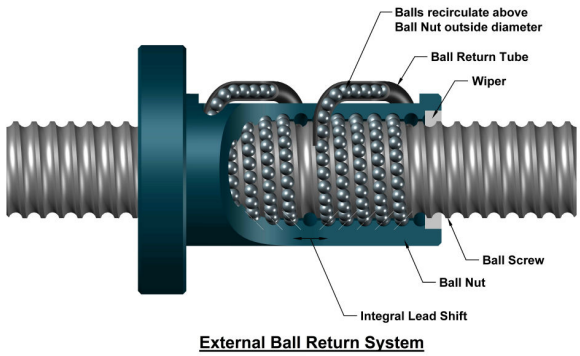

Figure 4.1: Ball-screw drive assembly [213]

or to position an object, e.g. moving an object as a pick and place system. Selecting a ball screw mechanism which meets the user's exact needs and requirements has major importance in the design. With a wrong selection, there might be a need to redesign the component or even the system, or it may result in production losses due to poor machine performance. In our design analysis, to reduce the computational time and complexity of the system, the ball and thread mechanisms are not considered.

A beam is a machine element which resists the load that is applied to the beam's axis laterally. Heavy loads can cause deflection of a beam and bend it. The forces on the beam generate shear forces and bending moments, which produce the deflection, internal stresses and strains. There are many applications for beams in industry, for example constructing floor systems for ships, buildings and bridges, submerged floating tunnels, buried pipelines, railroad tracks, automotive automobile frames, aircraft components, machine frames, etc. [215].

The system configuration for the pick and place application used in this thesis is a multi-DOF mechatronic system with nonlinear dynamics consisting of two DC-motors, a ball-screw-drive and a beam, as depicted in Fig. 4.2. The case study demonstrates the ability of the method and supporting toolbox in modelling and optimizing multi-degree of freedom systems with nonlinearities. The configuration of the system components is described as follows. A DC-motor and a ball-screw are along the y-axis, a second DC-motor is mounted on the nut section of screw system and the rotor of the second motor is along the z-axis. The dynamic force from the beam to the nut is dependent on the angular position of the beam. The features of nonlinear systems with multiple degrees of freedom are enabled

in the supported software toolbox in the presented method. Linearization of the system at several given operating points is performed. This system includes a 2-DOF moving platform on top of which a rotational beam is mounted. The first DOF is implemented by a ball-screw transmission and the second DOF is the rotational movement of the beam. The movement and force on the beam have a nonlinear effect on the ball-screw transmission (appended Paper E). The load specification for this system is here assumed to be given as a 2-DOF profile shown in Fig. 4.3 that is decomposed into two profiles that are used in each degree of freedom, as depicted in Fig. 4.4.

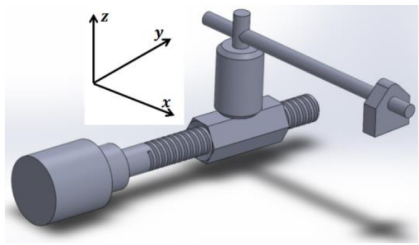

Figure 4.2: System configuration with ball-screw drive

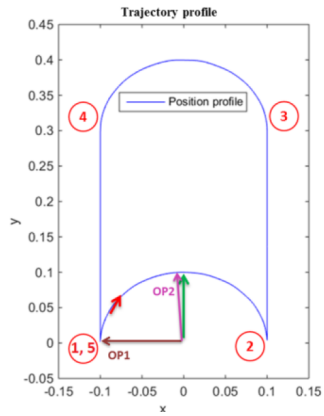

Figure 4.3: Load profile

This system is linearized at specified operating points (OP_1 and OP_2 in Fig. 4.3) as inputs, and the models are verified in Simulink. The nonlinear system is compared to the linear system both in Simulink and SimMechanics. The couplings between the two-degrees of freedom are investigated and modelled dynamically and presented in appended Paper E. The system is optimized for volume of the entire system and the impact of dynamic constraints on the final optimal solution is studied. The results are compared when the system is linearized at Op_2 for two scenarios: including and neglecting dynamic constraints. The optimized variables are shown in Table 4.1.

Comparing the optimized parameters and objective, we clearly see that including the dynamic constraints changes the optimal solution significantly, and it results in a larger volume of the entire system. It is obvious that dynamic constraints have substantial effect on the result of the optimization for the parameters and objectives. Consequently, enabling dynamic constraints makes it harder for the optimizer to find an optimized solution for

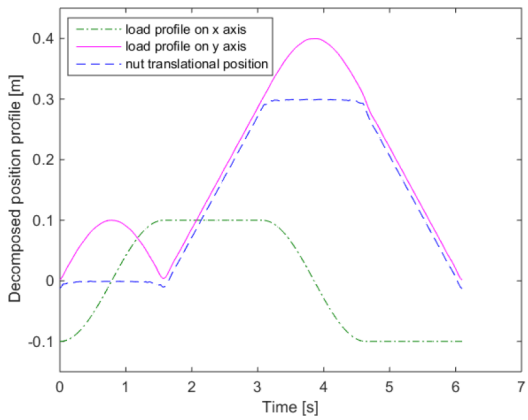

Figure 4.4: Decomposed load profile into two trajectories and the nut position profile

the system. The linearized and nonlinear systems are evaluated and com-

Table 4.1: Design optimization results

	With Dynamic Constraints	**Without Dynamic Constraints**	**Property**
$V_{tot}[m^3]$	$6.9e-04$	$4.8e-04$	Total Volume
$r_{m1}[m]$	0.0122	0.0133	Radius of DC-motor 1
$pbs[m]$	0.0099	0.0068	Pitch of screw
$r_{m2}[m]$	0.0187	0.0157	Radius of DC-motor 2
ISE	$0.097 \leq 0.5$	–	Integrated Square Error
$max(er)$	$0.17 \leq 0.5$	–	Maximum error

pared for both nut and beam movements to investigate the fidelity of the models. The nonlinear model is verified both by Matlab SimMechanics (second generation) and Simulink. SimMechanics is a physical modeling tool with a machine elements component library. There is no need to write equations to evaluate the dynamics, since the equations are provided by Matlab and can be configured by the user. Using these two different modeling tools, the system behaviours are compared with the Simulation result of the linearized state space model obtained by the IDIOM toolbox. Two simulation scenarios are considered for effective comparison of the three distinct models of the design case. First, apply an input of sine wave with maximum am-

plitude of $1Nm$ only to the first DC-motor rotating the ball-screw. Second, the same input is applied only to the second DC-motor. In each scenario, translational and rotational motion profile of the nut transmission of the ball-screw and the beam are analyzed. Fig. 4.5-4.8 depict the simulation results of the three models.

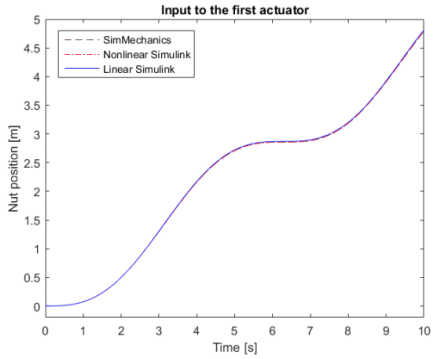

Figure 4.5: Nut movement for the three models

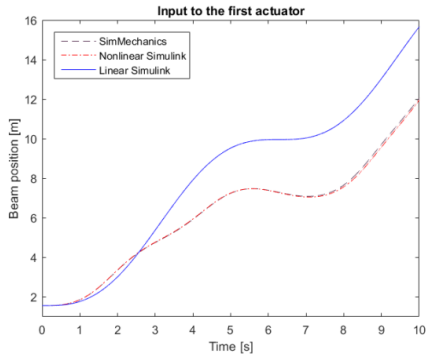

Figure 4.6: Beam movement for the three models

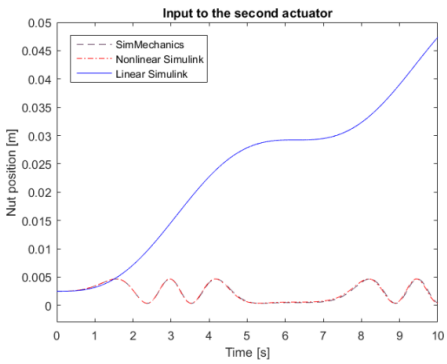

Figure 4.7: Nut movement for the three models

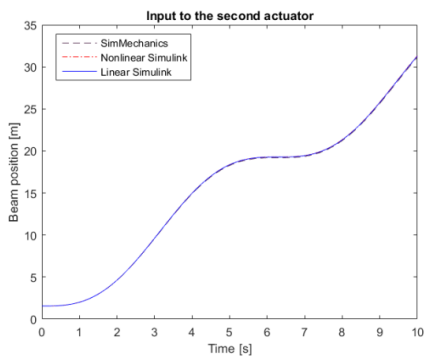

Figure 4.8: Beam movement for the three models

As shown in these figures, the beam in the SimMechanics and nonlinear Simulink models oscillates when there is an input torque to the first actuator; however, it does not fluctuate in the linear model, and it continuously moves in one direction (Fig. 4.6). Similar oscillation behaviour is perceived in the nut's motion in the two nonlinear models when there is an input

torque to the second motor; however, the nut continuously moves in one direction in the linear model (Fig. 4.7). This is because linearization removes the trigonometric functions and hence the cyclic movement becomes unobservable. However, the models should be analyzed in the vicinity of the operating point where all the models have the same magnitude. This is done by analysing the root mean square (RMS) error values in 2 seconds for the beam and nut positions in two different models in the mentioned two scenarios (details in appended Paper E). The tracking performance of the output position profile for the nut when implementing a full-state feedback control for the linear approximation at a specified point is derived (shown in Fig. 4.9) which satisfies the boundary definition for the control performance. Using this design case, the capability of the method and supported software to handle nonlinear multi-DOF systems is evaluated and enhanced (details are referred to appended Paper E).

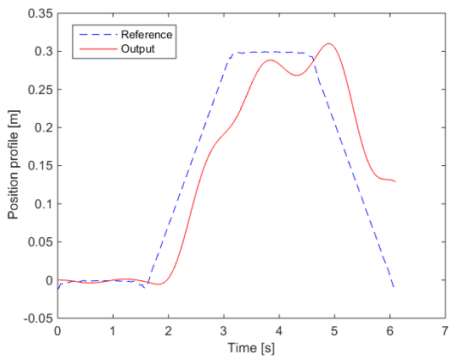

Figure 4.9: Dynamic performance of the linearized system at OP_1

4.2 2-DOF robotic manipulator

Nowadays, the demand for robots is increasing rapidly and they have wide applications across almost every area. The robotic arm design with control optimization has significant uses in a wide range of industries, from manufacturing, bio-mechanics, welding and automotive to monitoring pipelines, space exploration, the military and even online trading [216, 217, 218]. Robotic arms are capable of working in unpredictable and dangerous environments and circumstances that are not secure and safe for humans. The robotic arm has gained increasing interest in health services as well. They are used to distribute and administrate drugs to disabled and aged

people where high accuracy is required for efficient applications [219].

As a design case in this thesis, a two degree of freedom robotic arm representing a nonlinear mechatronic component is implemented in the toolbox as one whole physical component. The system is configured using two DC-motors where each motor drives one arm (depicted in Fig. 4.10). To consider the interactions between the arms and the relationship between the inputs and outputs, Lagrangian mechanics is employed. For this design case, the load to be handled by the system is a 2-DOF load, shown in Fig. 4.11 and Fig. 4.12, that is used in inverse kinematics given in (3.25) and (3.26) to evaluate the angle of rotation of each arm, and DC-motors for dimensioning and scaling purposes. The required torques/forces and speeds are derived using the position profiles.

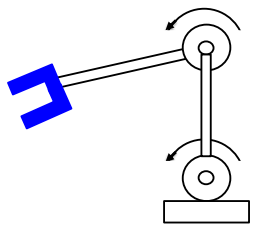

Figure 4.10: 2-DOF robotic arm

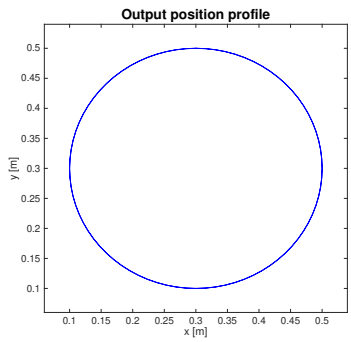

Figure 4.11: Defined load profile (motion profile) for the system

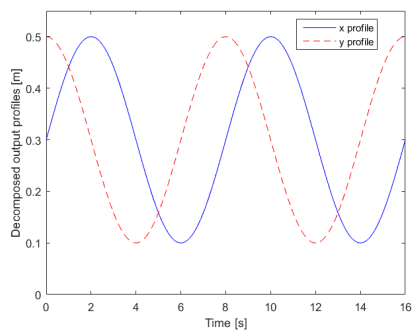

Figure 4.12: Decomposed load trajectory for the system

The trajectory tracking performance of the two arms and the load are evaluated and presented in Fig. 4.13 and Fig. 4.14. Fig. 4.15 shows the error of the trajectory tracking by the 2-DOF arm. Fig. 4.16 depicts the

end-effector trajectory tracking of the 2-DOF arm as a result of controlling the angles which satisfy defined boundaries for dynamic constraints. The

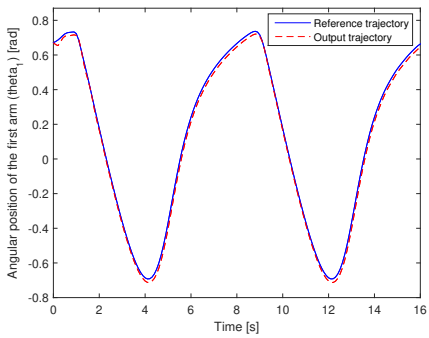

Figure 4.13: Trajectory tracking of the first arm

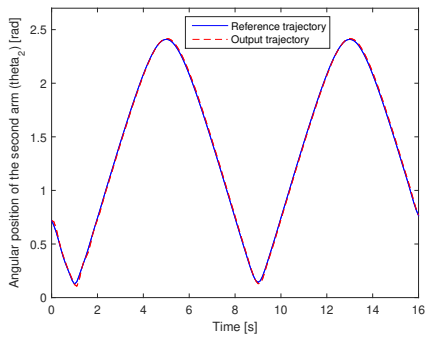

Figure 4.14: Trajectory tracking of the second arm

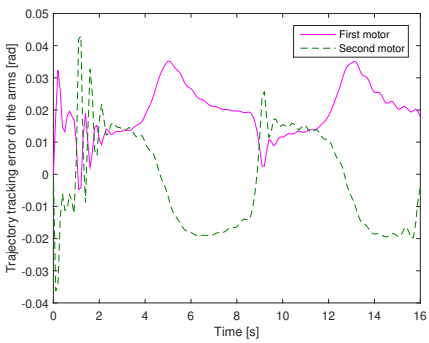

Figure 4.15: Tracking error

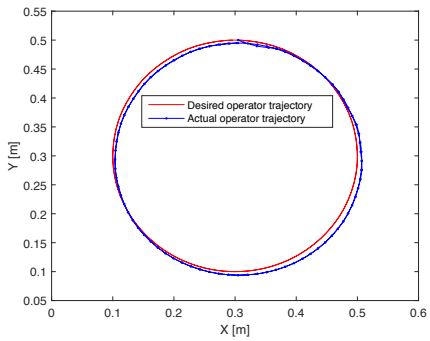

Figure 4.16: Trajectory tracking by the load

system is compared to a previously published method [24] which ignores physical design in terms of dimensioning/scaling and optimizes the system for the control performance using a bee colony method (presented in Table 4.2). l_{m1}, l_{m2}, l_1, and l_2 are the length for DC-motor 1, DC-motor 2, arm 1, and arm 2, respectively. k_{P1}, k_{I1}, k_{D1}, and k_{P2}, k_{I2}, k_{D2}, are the proportional, derivative and integrated gains for the two PID controls applied on the two-DOF of the system, separately. v is the volume of the entire system and MAE is the mean absolute error that is derived using (4.1).

Table 4.2: Comparison results

Optimization Parameter	**Method in this Thesis**	**Method in [24]**
l_{m1}	$0.0121m$	-
l_{m2}	$0.0894m$	-
l_1	$0.328m$	$1m$
l_2	$0.349m$	$1m$
k_{P1}	178	200
k_{I1}	29	50
k_{D1}	3	6.9577
k_{P2}	188	189.496
k_{I2}	12	47.2071
k_{D2}	7	7.2797
v	$0.0083m^3$	-
MAE	0.0427	0.04636

$$MAE = \frac{1}{T}\int_0^T (\theta_{d1}(t) - \theta_1(t)) + (\theta_{d2}(t) - \theta_2(t))dt \tag{4.1}$$

where θ_{d1} and θ_{d2} are defined as desired positions and θ_1 and θ_2 are the actual controlled outputs. As depicted in Table 4.2, the presented method using this design case is complete in the sense of covering physical dimension, dynamic evaluation and static properties of the system.

This method shows that the results achieved by including dynamic constraints are competitive even though considering physical design and dynamic constraints imposes limitations in the final design solution. Using this design case, it is shown that the method can replicate the results presented in [9] in an efficient manner considering time for multi-DOF complex system configurations with nonlinearity. Furthermore, the design case is used as an illustrative example to facilitate comprehension of computation rules, and concepts which are expressed in a form of definitions and theorems in appended Paper D (the details of which are skipped in this thesis to avoid distraction from the main goals of the research).

4.3 Servomechanism

Synchros and servomechanisms are usually used in combination. A synchro consists of a stator and a rotor and they are used to transmit torque [220]. A servomechanism, on the other hand, senses the error between the actual

signal and the reference signal and corrects it [221]. Nowadays, servomechanisms are used in automatic machine tools, satellite-tracking antennas, telescopes, automatic navigation systems and anti-aircraft-gun control systems. To explore the impact of embedded control implementation on the integrated design method, a system with less complexity of the model and fewer design variables is needed, hence a servo system configuration is used (shown in Fig. 4.17). Even though the models of this system are simple, the main characteristics of the system remain, therefore it facilitates comprehension of the impact of parameters from different domains. Using this system, some features of discrete-time systems such as evaluating sampling period/rate are considered in the method. The closed loop system is discretized, modelled and optimized to reduce computational cost and size of the system.

To decide the sampling time, it is necessary to take some major impacts into account. As well-known, small enough sampling time assists in reproducing the open-loop time response precisely, i.e. small enough sampling time mimics a continuous design closer. Moreover, it causes the system to react readily enough to the affecting disturbances. On the other hand, large enough sampling time reduces numerical issues and aids in avoiding fast and expensive control implementation. Hence the method allows for a reasonable choice of the sampling period to minimize the data loss and at the same time leaves enough time for the microcontroller to process the data within the two sampling instants.

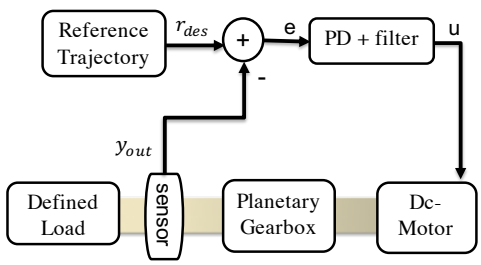

Figure 4.17: Servomechanism with error feedback, PD control

The requirement on this system is defined as two different load profiles that should be handled by the system one at a time to ensure the broad application of the method: a *sinusoidal* load that has different frequencies and amplitudes during time and a *trapezoidal* profile as shown in Fig. 4.18 and Fig. 4.19, respectively. The system is designed and optimized for smallest volume and slowest sampling frequency by varying the boundaries of the dynamic constraint, $max(er)$. Analysing the results clarifies that the

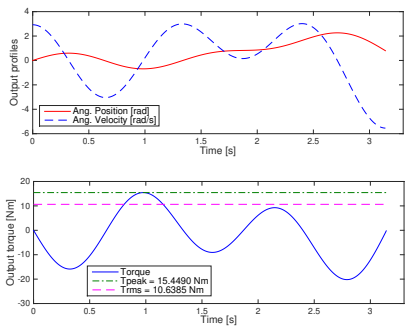

Figure 4.18: Sinusoidal (load) profile

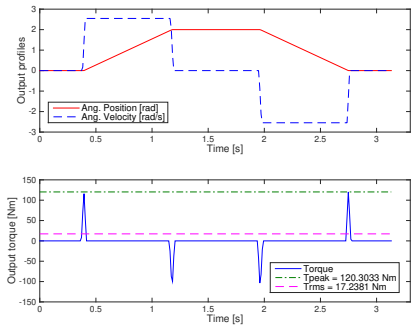

Figure 4.19: Trapezoidal (load) profile

relationship between the dynamic constraint, $max(er)$, sampling frequency, w_s, and bandwidth, w_c, follow a pattern as expected (control theory). The changes for volume value are not considerable (extra investigation is needed, as presented in Section 4.3.1). Derived patterns that are intuitively correct (from control theory) confirm the couplings and correlation between the domains (shown in Fig. 4.20). A few test cases for this very servomechanism

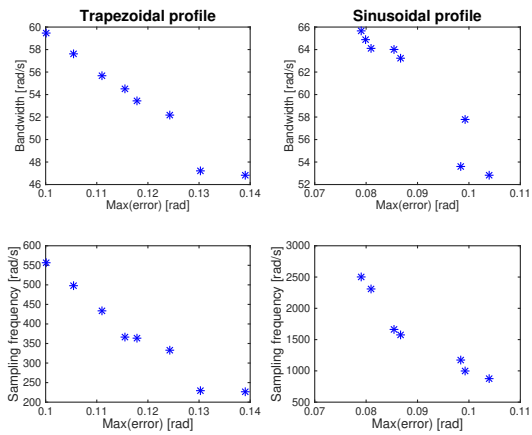

Figure 4.20: The pattern for bandwidth and sampling frequency over $max(er)$

design case are examined by varying the boundary for dynamic constraints,

and the tracking performance of the reference signals are derived (details in appended Paper C). The final system solution for one of the tests is verified dynamically using Simulink (shown in Fig. 4.21 and Fig. 4.22) and geometrically (physically) using SOLIDWORKS (shown in Fig. 4.23). The

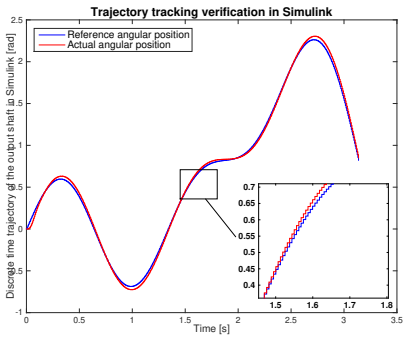

Figure 4.21: Verification of the discrete time trajectory tracking for the output shaft in Simulink

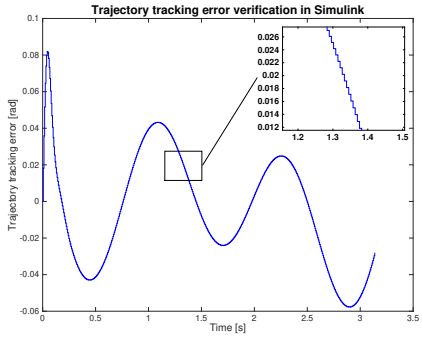

Figure 4.22: Verification of the discrete time trajectory tracking error in Simulink

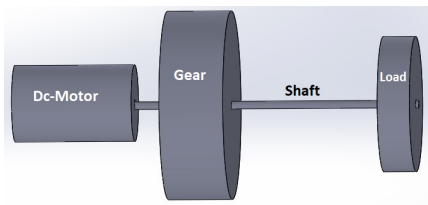

Figure 4.23: CAD model using optimal parameters

sensitivity of the solution with respect to the weight factors (presented in Section 3.5.3) are discussed (extra analysis are presented in Section 4.3.1). Planetary gear is a core component in this configuration. The effect of gear ratio as an important parameter of the planetary gear (in physical and control optimization of mechatronic systems) for the optimization and control performance is investigated. The system is optimized for a few defined fixed values of gear ratio. Fig. 4.24 depicts the result before implementing the weighted sum function on the Pareto optimal set for these tests.

The results depict that the control performance improves with gear ratio, which is in line with Roos [20], i.e. it is possible to decrease dynamic constraint boundaries by increasing the gear ratio. This is due to an increase

of the torque margin with gear ratio. The decrease in $max(er)$ requires an increase in bandwidth of the closed loop system. Hence the sampling frequency of the system is increased. These results show that larger values of gear ratio, n, result in an optimized system without an unnecessarily small sampling time causing computational cost. Moreover, larger values of n result in better control performance from a dynamic constraints perspective. In further studies it should be considered to have n as a free design variable during the optimization.

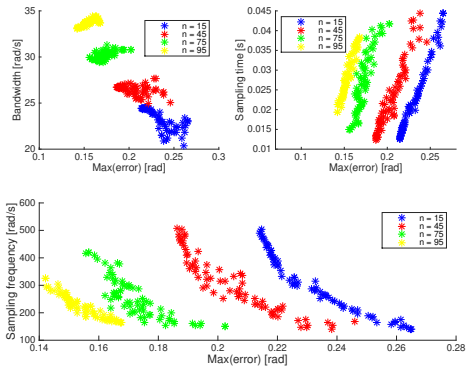

Figure 4.24: Different gear ratio analysis

4.3.1 Complementary analysis

In order to track the impact of co-design parameters in detail, we have extended the presented results and analysis in appended Paper C. In the presented analysis, due to the uneven range of the two objectives (volume and sampling frequency), the impact of sampling frequency on the final optimum solution is highlighted and—accordingly—the impact of volume is not considered. Applying the definition of weight factors in (3.9), and normalization expressions in (3.11) and (3.12), results in a range between zero and one for all of the objectives. Hence the selection of weight factors based on the importance of relevant objectives becomes more realistic. A few tests are performed, presented in Table 4.3. The gear ratio, n, for these tests is selected to be fixed as 40 and the shaft damping, d_{sh}, is defined to be 8. The parameters of PD control, the low-pass filter and sampling frequency factor, c_b, are defined as free variables, but are excluded in these Tables to avoid confusion. The same patterns (as shown in Fig. 4.20) hold for the results given in Table 4.3, i.e. a larger boundary for the dynamic

Table 4.3: Optimization results for $k_1 = 0.5, k_2 = 0.5$

Test	**1**	**2**	**3**	**4**
$l_m[m]$	0.081	0.0864	0.0931	0.091
$r_g[m]$	0.0544	0.0503	0.0424	0.0391
$l_{sh}[m]$	0.0714	0.0452	0.0593	0.0894
$max(er)$	0.11$\leq$ 0.11	0.1196$\leq$ 0.12	0.13$\leq$ 0.13	0.138$\leq$ 0.14
$v[cm^3]$	930.8	844	691.5	625.1
$w_s[rad/s]$	298.8	244.2	228	215.9
w_b	129.1	121.2	102.6	84.4

constraint, $max(er)$, allows for smaller bandwidth of the closed loop system, and—accordingly—smaller sampling frequency, w_s.

For comparison purposes and in-depth study of the impact of weight factors on design parameters and objectives, we have defined $k_1 = 0.9$ and $k_2 = 0.1$ and $k_1 = 0.1$ and $k_2 = 0.9$ in Table 4.4 and Table 4.5, respectively. These choices are made based on the importance of the objectives after normalization. Checking the impact of weight factors k_1 and k_2 on the

Table 4.4: Optimization results for $k_1 = 0.1, k_2 = 0.9$

Test	**1**	**2**	**3**	**4**
$l_m[m]$	0.0826	0.0865	0.0932	0.0912
$r_g[m]$	0.0544	0.0503	0.0424	0.0391
$l_{sh}[m]$	0.0715	0.0453	0.0594	0.0896
$max(er)$	0.1094$\leq$ 0.11	0.1196$\leq$ 0.12	0.13$\leq$ 0.13	0.138$\leq$ 0.14
$v[cm^3]$	935.5	845	691.9	626
$w_s[rad/s]$	295.9	243.2	227.9	215.7
w_b	129	121	102.5	84.2

objectives individually, we see in Table 4.4 for corresponding tests, the final optimized result for w_s is better (minimized) than volume (minimizing w_s is prior to minimizing v). Therefore, in each test in Table 4.4, volume is larger than the equivalent test in Table 4.3, but w_s is smaller. As a result, the dominant parameter in volume calculation, i.e. the length of motor, becomes larger in the presented tests in Table 4.4 than Table 4.3.

Changes in sampling frequency affect physical design parameters indirectly, i.e. smaller w_s results in smaller natural frequency and smaller eigenvalues in the system, w_b, (related to the physical design parameters), which yields a larger length of the shaft, l_{sh}, (even though the differences are very small, we can see this by comparing each test in Table 4.3 and Table 4.4). The opposite explanations hold when comparing Tables 4.3 and 4.5. As it

Table 4.5: Optimization results for $k_1 = 0.9, k_2 = 0.1$

Test	**1**	**2**	**3**	**4**
$l_m[m]$	0.0752	0.0863	0.0931	0.0792
$r_g[m]$	0.0544	0.0503	0.0424	0.0392
$l_{sh}[m]$	0.0655	0.0446	0.0592	0.058
$max(er)$	0.11$\leq$ 0.11	0.1188$\leq$ 0.12	0.13$\leq$ 0.13	0.138$\leq$ 0.14
$v[cm^3]$	914.1	843.8	691.4	589.9
$w_s[rad/s]$	324.2	245.3	228.2	279.9
w_b	131.1	122	102.8	84.7

is obvious from the tables, a larger boundary for the dynamic constraint, $max(er)$, allows larger solution space for both of the objectives (minimized values for v and w_s).

4.4 Belt-drive system

Nowadays, timing belts are widely used in innovative drive solutions in mechatronic systems which combine mechanics, sensors, control systems and servo technology with freely programmable and distributed drive solutions [207]. A timing belt is a type of mechanical belt that transfers rotary motion from one shaft to another. Timing belts have similar accuracy to gears and are cheaper. They have teeth to form based engagement with the pulleys. The name *timing* is due to their performance where they transfer motion at exact preset timing. In addition to an endless variety of industrial applications of timing belts as mechanical elements, there are different every-day machines that employ timing belts such as treadmills and sewing machines.

A system with a belt-drive is considered in this study (depicted in Fig. 4.25). The design case includes a DC-motor, a belt-drive and a rotational load component with an optimal full-state feedback pole placement control and an optimal proportional feed-forward control (presented in Section 3.7.3.1), and a sensor quantization model is considered. This system is used to highlight the impact of sensor resolution, sensor location and couplings on the achieved design solution. It is common knowledge that resolution is an important specification in control design and embedded control implementation. Without enough resolution, the necessary measurement might not be reliable, and an over performing sensor is an increased cost. Using this design case, the resolution factor, δ, is defined as an optimization objective that is maximized to reduce hardware cost (maximization of δ translates to reduced resolution). On the other hand, control perfor-

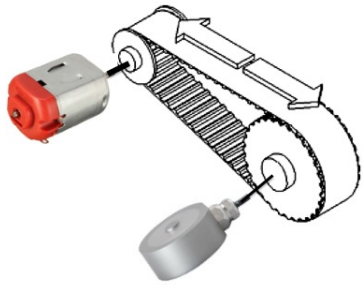

Figure 4.25: Case study representing the concept of a belt-drive system

mance constraint requires small enough δ to satisfy the boundary condition. Therefore, considering these two primary facts as design criteria, a reasonable resolution is derived.

Dynamics of the belt-drive are studied and a design model is developed (presented in Section 3.6.4, and Papers A and B). The system is configured and linearized at specified operating points. A multidisciplinary optimization is executed to obtain a final optimum solution, and the sensitivity of the physical and control design and embedded control implementation parameters are investigated. A few design tests are executed by varying the boundary for dynamic constraint and the sensitivity of the solution is investigated as shown in Fig. 4.26.

The system is optimized and the results are compared for two scenarios, where the sensor quantizes output of the motor ($sensor1$ is applied) and the load component ($sensor2$ is applied), one at a time. Control performance in both scenarios is checked over output of the load component and evaluated by tracking of a reference signal. Considering the results, it is clear that a lower (more optimal) sampling frequency is possible with a more loose definition of dynamic constraint boundaries, which is in line with the results presented in appended Paper C (true for both scenarios of sensor locations, $sensor1$ and $sensor2$). As the boundary for $max(er)$ increases, the derived optimal value for δ increases, which is intuitively correct (control theory). A larger ratio between the pulleys, $n = \frac{r_2}{r_1}$, yields less torque requirement on the motor, consequently smaller length of the motor, l_m, which is in line with the results achieved in [20]. Comparing the two scenarios, we see that when $sensor2$ is applied we have a better resolution (performance-wise) for the system than when $sensor1$ is implemented; i.e. quantizing the position of first pulley degrades resolution in the output side (due to the ratio between the pulleys, i.e. physical design parameter impact on co-design). Moreover, when $sensor2$ is applied, better control performance, $max(er)$, is achieved

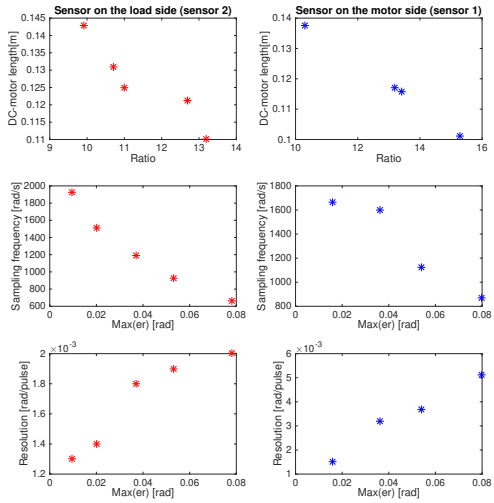

Figure 4.26: Patterns generated from system behaviour

than when $sensor1$ is applied.

The control performance of the optimal pole placement with feed forward proportional gain is verified by the position tracking of the output position (Fig. 4.27). The control performance of the optimal control method satisfies the constraint by adequately tracking the reference input. To evaluate the final solution of the system, different weight factors considering $k_1+k_2+k_3 = 1$ are applied on normalized objectives (shown in Table 4.6). Test *main* is when the weights are equal, i.e. the objectives have the same importance. As stated, Test 1 has higher weight on the volume of system, v_{tot}, therefore volume is more optimal (minimized) than the two other objectives compared to Test *main*. Test 2 gives higher importance to the sampling frequency, w_s, hence the results prioritize minimizing sampling frequency rather than the two other objectives compared to Test *main*. In Test 3, resolution factor, δ, has higher weight and the method results in a better (maximized) solution for δ.

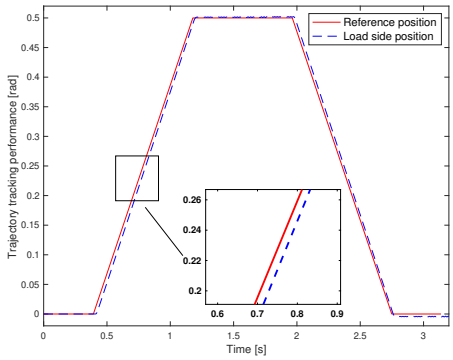

Figure 4.27: Trajectory tracking performance

Table 4.6: Results of the co-design for different weight factors

Test	k_1	k_2	k_3	$v_{tot}[cm^3]$	$w_s[\frac{rad}{s}]$	$\delta[\frac{rad}{pulse}]$
main	$1/3$	$1/3$	$1/3$	$1.17e^4$	658	0.002
1	0.8	0.1	0.1	$9.87e^3$	887	0.0019
2	0.1	0.8	0.1	$1.43e^4$	571	0.002
3	0.1	0.1	0.8	$1.2e^4$	693	0.0022

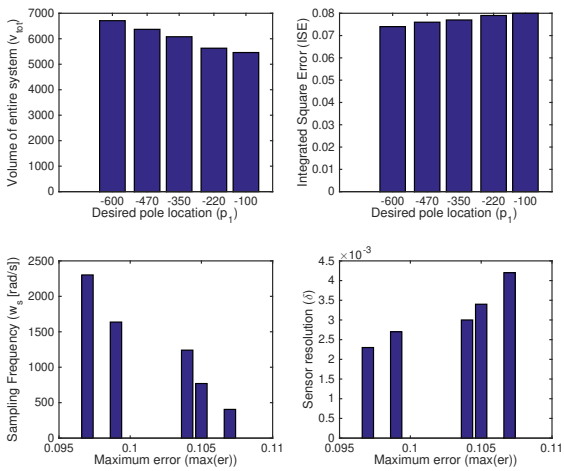

Figure 4.28: System behaviour analysis

Comparing these results verifies the significant importance of weighted sum factors. In order to derive a final solution that is optimum without deteriorating any objective or giving high priority to just one, these factors should have exactly the same weights, i.e. equal numeric. To track the impact of poles on the physical and control design, an observer-based control is implemented and a sensor quantization model is integrated in the model (presented in Section 3.7.3.2). A number of optimizations with different fixed values for the poles, and fixed boundaries for $max(er)$ and ISE are executed. The change for physical properties of the system (volume) with respect to the dynamic specifications are investigated. As depicted in Fig. 4.28, faster poles put higher demand on actuator effort (in line with presented results in [222]).

The sampling frequency of the system and sensor resolution have similar patterns as presented in Paper C, i.e. as the boundary for control constraints $max(er)$ and ISE increase, the optimization allows larger solution space for both of the objectives (decreased w_s and increased δ). Using this design case, the importance of integrating different engineering disciplines concurrently in the design process is highlighted and the correlation between the domains are derived (details in appended Papers A and B).

The problems presented in this chapter are all multidisciplinary design problems, however they are formulated either as single objective or multi-objective optimization problems.

Chapter 5

Summary of appended papers

This dissertation provides a brief explanation of the research presented in the appended papers. The method can be divided into a few sections: approach, component models and case studies. In all papers, besides extending a method to cover some new aspects of integrated design optimization for mechatronic systems, we have presented new component models and extended the supporting software toolbox to handle the detailed features of the models.

Paper A:

Multi-Criteria Co-Design Optimization of Mechatronic Systems

Paper A studies sensor resolution and sensor location impact as an embedded control implementation parameter on the co-design optimization approach. For the physical component, a belt drive is studied and the corresponding dimension and dynamic behaviour models are considered to enhance the modelling capability of the toolbox. The system is highly nonlinear; therefore the system is linearized at specified operating points. A quantization sensor model is applied on the optimal pole placement control method and the system is optimized for three defined objectives—as volume of the entire system, sampling frequency, and sensor resolution—to reduce the implementation and hardware costs, besides keeping the best performance of the system. The method is applied on a case study including a DC-motor, a belt-drive and a load component to evaluate the feasibility of the solution.

Paper B:

Holistic Multidisciplinary Method for Optimization of Mechatronic Systems

Paper B presents a method based on holistic approaches. This paper extends the method published in paper A to investigate the control performance impact on the co-design objectives and parameters, and overlaps with Paper A in the used design case and some test results. A detailed design model for the belt-drive is developed which aided in reducing the complexity of analysis by decreasing the number of free design variables. An observer-based optimal pole placement control design is employed. The system is optimized for three objectives from three engineering domains to reduce size, implementation cost and hardware costs. A weighted sum function is applied on the result of optimization as a Pareto optimal set to provide even weights for the objectives. The sensitivity of the objectives to the weight factors are evaluated. The optimum system is derived, control performance is evaluated, and the correlations between parameters from different domains are studied.

Paper C:

Concurrent and Optimal Structure, Control and Implementation Design

Paper C extends the method to study the impact of parameters such as sampling frequency and bandwidth from the embedded control implementation domain on the co-design. Three engineering domains are considered: mechanical, control and embedded control implementation. The approach is developed to consider a multi-objective optimization where each objective is from the mentioned engineering domains, and corresponding constraints from these domains are included. The couplings and inter-connections of these domains are studied, and the method is applied to a design case consisting of a DC-motor, a planetary gear, a flexible shaft and a rotational load component. The system is optimized for two given trapezoidal and sinusoidal profiles including several amplitudes and frequencies. The impact of physical and control design parameters and weighting factors are presented and the system is verified dynamically via Simulink and geometrically using SOLIDWORKS.

Paper D:

Towards a Formal Framework for Simultaneous Optimization of Dimension and Control of Mechatronic Systems

Paper D explains the system concept, and computational rules in IDIOM toolbox. The modelling features of the toolbox are advanced to include a nonlinear 2-DOF robotic arm. To facilitate comprehension of the concepts, a mechatronic case study including two DC-motors to actuate each arm of a robotic system, 2-DOF arm and a load component is considered. The Lagrangian approach is employed to derive dynamic behaviour model of the configured system and the system is linearized for optimization purposes. Inverse-kinematics is used to determine joint references which results in a specified end-effector position profile. The evaluation of the system model together with the pseudo code for the algorithms is presented.

Paper E:

Early-phase Design Optimization of Mechatronic Systems

Paper E extends the method and IDIOM supporting software toolbox to cover multi-DOF mechatronic systems and to handle nonlinear dynamics of the model. Some physical components such as ball-screw and a rotational beam were studied. Physical dimension, static properties and dynamic behaviour models for a design case including a DC-motor, ball-screw with a nut, DC-motor, rotational beam and a load were developed and added to the toolbox. The system has complex physical dimension and dynamic behaviour models in comparison to previously published design cases in this area. The nonlinear system is linearized at a few specified operating points given a two-DOF load profile, and is optimized for the size of the entire system. The control performance of the system is evaluated. The final optimal solution for the system is verified via both Simulink and Simscape.

Chapter 6

Discussion and Conclusion

The aim of this thesis is to employ a concurrent design optimization method on mechatronic systems that integrates different engineering domains in the early design phase in order to enhance efficiency and reduce cost. Furthermore, the aim is to advance the performance in the system development. The aim is defined in the form of research questions as presented in Section 1.2.6 and are briefly answered in this chapter.

6.1 Discussion on research questions

Q1: Can a balanced optimum solution be realized considering physical (mechanical/electrical), control and embedded control implementation domains simultaneously without increasing the cost and complexity of the evaluations and models?

A1: Different detailed design objectives and constraints from corresponding engineering domains can be considered, simultaneously, by the use of less complex models.

A library of physical and control components allows for treating each component model individually, which assists in the detailed design of systems and inclusion of constraints and objectives from different domains. Design of embedded control systems which is based on two dependent phases—incorporating controllers and mapping them to a given platform—yields resource under-utilization, poor performance or instability of the application, since timing problems may arise due to implementation decisions [10]. The method presented in this thesis considers important implementation factors such as bandwidth, sampling time/frequency and sensor resolution

in the design process. Hence a proper control system is designed with optimized configuration by taking into account design constraints (presented in Chapter 3) from physical, control and implementation domains. The method has a broader coverage of engineering disciplines in comparison with the methods and tools presented in [6, 223, 224]. The mentioned domains are incorporated seamlessly in the co-design optimization method, which resulted in a study on parameters correlation as well as couplings between disciplines (results presented in Chapter 4).

The main advantage of the design method in this thesis is the possibility to relate system properties and parameters to different involved engineering disciplines. With respect to the involved domains, three design objectives are defined—the volume/mass, sampling frequency/time and sensor resolution—in order to minimize size, implementation cost and hardware cost. Other frequently used domains and objectives in the literature such as magnetic, thermal, energy and aerodynamic can be integrated easily in the system analysis, if the corresponding models are added to the method.

The optimizer determines design variables from different domains, concurrently. Cost and efficiency are taken into account in the optimization process by keeping the system complexity at a reasonable level and including simple enough models while capturing the main characteristics of the components. Simple models are based on the detailed assumptions presented in Chapter 3 for the dynamic, static and physical dimension models of each component. A Pareto optimality set is the result of such a multidisciplinary multi-objective problem. In this set, it is not possible to optimize one objective without deteriorating the others. Without additional preference, all Pareto optimal solutions are considered equally good. Hereupon, in order to find a single solution that satisfies the preference, there is a need to have a definition of a function which results in a reasonable single optimal solution for all of the objectives. A weighted sum function is integrated in the method which aggregates the objective values to a single quality measure and yields a balanced optimum solution between the involved domains.

Contradictory to the studies presented in [116, 120], the method consists of one optimization loop which significantly reduces evaluation time. Moreover, the design problem is considered as a holistic problem rather than partitioning the problem and implementing several optimizers to solve the models in partitions separately as in [152]. The method is capable of solving multi-objective constrained problems without the need for approaches which decompose the problem into several single-objective unconstrained problems as presented in [131, 132]. The presented method allows configuration of conceptual mechatronic systems for simultaneous integration of optimal structure, control and implementation design.

Q2: Can the design parameters, constraints, objectives, dynamic behaviour, static properties and physical dimension models of mechatronic systems be incorporated within a simultaneous multidisciplinary method to yield a potential design for a mechatronic system concept considering several engineering domains?

A2: Considering models for dynamics, statics and physical dimension in component level as well as a multidisciplinary method which integrates free design variables, fixed design parameters, design attributes, constraints and objectives can realize a simultaneous design which yields an optimum solution for mechatronic system concepts.

Unlike the studies presented in [139, 150] which are based on sequential design approaches, the method presented in this thesis is a simultaneous design method where the design parameters, constraints, objectives from different disciplines, dynamics, statics and physical dimension models are all considered and evaluated simultaneously. Since the method in this thesis is developed based on individual components rather than considering a model in a system level, it allows more flexibility for design and configuration of systems. Thus any system concept which uses the available developed component models in the method can be configured and designed. The method includes two types of components: physical and control. Physical components are basic electrical/mechanical elements which include three types of models: physical dimension, static properties and dynamic behaviour. The physical dimension models are in a component level and either use scaling of existing components or are based on defined physical constraints to be satisfied. Static properties models are also executed in a component level to transform a defined requirement, such as a load profile, throughout the component chain for physical design purposes. These models are executed in each optimization run based on the presented method and algorithms in Section 3.3.

In order to evaluate a system, the dynamic behaviour models of each component in the form of ordinary differential equations (ODE) are gathered to be configured using algebraic equations (AE) as interface equations. Symbolic representation of the system in the form of a state space model or transfer function model is derived and a controller is implemented on it. The numerical system is evaluated using physical system parameters generated by the optimizer. Both physical and dynamic constraints are imposed to the method to be satisfied by the designed system. A large number of studies [90, 94, 95, 96, 97, 100, 104, 106, 107, 108, 109, 150] in the literature only consider dynamics in the design of mechatronic systems with no integration of physical/structural design. Hence no dimensioning/scaling or static models are considered in their methods.

Q3: Can multi-degree of freedom and nonlinear behaviour of mechatronic systems be dealt within an integrated manner in an early-phase of design with less computationally demanding models?

A3: The main design features for multi-DOF components as well as configurations and nonlinear components/systems can be enabled in an early-phase design method by treating components and models individually and integrating them in a system evaluation step.

Early-phase design refers to the stage where the system configuration is defined. Integrating several design facets in this stage yields an early-phase design method. Design optimization of mechatronic systems in an early phase is currently one of the most challenging and active research topics. This motivated a study on developing a method as well as a tool for early phase design optimization of mechatronic systems to avoid costly iterations in case of errors appearing in later design phases.

Unlike the methods presented in [72, 76, 83, 84, 97, 127, 136, 113] which use time and cost wasting complex models and methods, this thesis focuses on improving efficiency in the design optimization of mechatronic systems by taking into account reasonable design and model assumptions (presented in Chapter 3) and design choices (given in Section 1.2.2) which lead to simple enough models while capturing the main features and characteristics of the components.

The method allows for multi-degree of freedom system evaluation by using either multi-degree of freedom components or multi-degree of configurations (with rotational and translational movements). A library of physical components which handle multi-DOF compositions facilitates these system concept configurations. A linearization method is used either in the component level (if component dynamics is nonlinear) or in the system evaluation stage (if the configuration is nonlinear) by the aid of Wolfram Mathematica solver. Lagrangian mechanics is studied and incorporated in the dynamic behaviour model of components. In the system evaluation stage, differential equations representing the system dynamics are gathered symbolically and sent to the mentioned solver to be linearized at specified operating points (presented in the Algorithms in Section 3.3).

6.2 Discussion

The method presented in this thesis is an early-phase integrated design which considers three engineering domains simultaneously. A non-synchronous method that disregards the impact of control and embedded control im-

plementation in optimization of mechatronic systems might cause an over-dimensioning in one or more of the involved engineering domains. Hence the presented simultaneous method facilitates an integrated design optimization method to cover some major key concepts in the closed loop system design. In this method, contradictory to a sequential design process, the design tasks are all performed concurrently. Sequential design limits the design space by locking the design aspects in each phase of development and adds additional constraints to the next sequence, hence the final system solution is not truly optimized. One other major disadvantage with sequential design is time wasting since each sequence can begin only when the previous one is ended. A sequential design method requires backtracking and debugging if any errors appear in the later phases of design or the later design tasks, which is time and cost consuming and therefore not considered efficient.

Most design methods in the literature [73, 75, 85, 93, 94, 95, 97, 100, 101, 102, 113, 120, 133, 134, 136, 138, 142, 145, 150, 154, 158, 159] are just applicable on a defined set of problems or case studies and they lack the generality feature of the design. This limits the focus of design properties and domain coverage, since the problem is formulated to be implemented on a particular system with respect to the specified characteristics. However, the method presented in this thesis is applicable to any type of system configuration (linear or nonlinear), as long as the models and the constituent components of the system are already developed and integrated in the method. There is no specific requirement on the components to be integrated in the method as long as it would be feasible to develop dynamic, static and physical dimension (scaling/dimensioning approaches/physical constraints) models for each component.

To consider the method as holistic, all design parameters should be defined as free variables, which would be unrealistic even for a small number of components due to the curse of dimensionality. This is also true for systems with a large number of components, since extra free design variables will be required. Despite this, the method is still considered as an efficient design process because, even for larger systems, the method will still focus on simultaneous early-phase analysis of the design.

The method is flexible in the sense that it can solve both single and multi-objective problems, and allows flexibility in the definition of objectives and constraints; i.e. the constraints can be defined as objectives and vice versa. It is feasible to add more objectives from various engineering domains to the design as long as the models are integrated in the method. For system evaluation and analysis, the system concept is defined as fixed, i.e. the number and configuration of components for the system should be specified prior to the evaluation. The method gives the possibility to define different operating points for linearizing components and systems; however, each of

the operating points yields a new system and should be evaluated separately.

According to the main goal and focus of the thesis which is on developing models and extending a tool for integrated design optimization, three engineering domains—physical, control and embedded control implementation—are taken into account. The size of system in terms of mass/volume, sampling frequency/time and sensor resolution are considered as the main objectives of the design from the mentioned domains. However, other domains and objectives are feasible to be considered by extending the method to include the corresponding models. In practice, there can be more than three domains and objectives as long as the models are integrated in the method. The couplings between disciplines and interactions between the design parameters are analysed which is essential in product development and assists in achieving a qualification for a design with reduced unnecessary iteration loops.

GA is implemented in the method and toolbox as an optimization algorithm since it is capable of solving problems that include discrete design variables as well as highly nonlinear objective functions. This algorithm can also handle constrained and unconstrained optimization problems, hence there is no need to divide constraint multi-objective problems to several unconstrained single objective problems. GA is a good fit for mechatronic design optimization problems due to its capability in deriving solutions to a diverse range of complex problems. However, other optimization algorithms with respect to a defined purpose and goal can be studied to incorporate the models in the method and implement them in the toolbox.

For system analysis, a library of components is developed which includes two types of components: physical (mechanical/electrical) and control. Each physical component consists of three models: *physical dimension* model for designing/scaling the elements; *static properties* model to translate the load profile (position/torque) throughout the component chain for physical dimensioning purposes; and *dynamic behaviour* model to evaluate dynamics. The models are used later in the optimization phase as objectives and constraints. This library allows for treating each component separately, but it is not yet rich enough to model and evaluate every type of system; however, the method is capable of handling any system configuration which uses the available developed component models.

Four design cases which represented real system design concepts and had enough complexity regarding degrees of freedom, non-linearity, modeling, evaluation, design variables and involved domains were studied. To evaluate other design cases with different components and configurations, the method should be extended to cover corresponding models of the components. Additionally, there is no specific requirement on the components to be integrated in the method and toolbox, as long as is a dynamic behaviour,

static properties and physical dimension model for the corresponding component.

System specifications and requirements are to be defined both in the system level and component level. Mechatronic systems features are considered both design-wise and dynamic-wise. The method is extended to cover multi-degree of freedom mechatronic components and configurations with nonlinear dynamics. Approaches to handle these features are studied and developed in the method and its supporting software toolbox.

6.2.1 Limitations, threats to validity

This subsection assesses the threats to validity and discusses limitations on the research approach (touched upon in section 1.3).

Validation of quantitative nature research is generally challenging because, unlike qualitative research which handles nonnumerical information and their phenomenological interpretation, and tie in with human senses and subjectivity, quantitative research deals primarily with numerical data and their interpretations under a reductionist, logical and strictly objective paradigm. The results achieved by the presented method in this thesis are reliable since the method and the results are replicable and in accordance with the control, embedded control implementation logic and theories, and are also in line and comparable with previously published research in the area. The results derived from assessing the design cases can be generalised to wider groups of problems and variations of design cases, which is another common way of validating quantitative research. Though the method presented in this thesis has broad application and is verified via design cases, it cannot claim to be the only way to solve the same problem in a similar context. The key words for the claims made through this work are *sufficiency* and *ideally*, not *the only solution*. A threat lies in the lack of opinions from industrial experts and partners. Even though the opinions of various academic experts are used to shape the thesis, missing an industrial viewpoint could induce a bias in the findings. To address this threat, other experts than the authors of the papers were selected to review the validity and assess the findings.

6.3 Conclusion

The method presented in this thesis (along with its supporting software toolbox) is capable of integrating several engineering domains simultaneously. A successful system design can be achieved if the interactions between components and disciplines are thoroughly considered. The overall goal of this thesis is to integrate engineering domains in an early design phase to improve

product development efficiency via a better system analysis. Therefore this work can be considered as a key move in evaluation of mechatronic systems design which assists in better system development and analysis.

The thesis covers a few technological fields such as modelling, optimization, physical design, control and embedded control implementation. The optimization is advanced to solve multidisciplinary problems where engineering features of systems from different engineering domains are considered. The method allows the simultaneous integration of mechanical, electrical, control and embedded control implementation domains.

For the multidisciplinary design, some properties of the systems are studied, such as volume/mass, bandwidth, sampling frequency/time, and sensor resolution. Different constraints with respect to the objectives and involved domains are added to the method. A Pareto front optimal set is derived and the final optimum system is obtained using normalized objectives and a weighted sum function. The couplings between domains and correlation between parameters are studied and the impact of parameters on objectives is investigated.

Design cases are used to evaluate the feasibility of the method and its supporting software toolbox. The method is compared qualitatively and quantitatively with previous research in the field, and the advantages of the method are highlighted. The dynamic models and control performance of the systems are verified using Simulink (Simscape/SimMechanics) and the geometry of the physical designs is verified using SOLIDWORKS.

6.4 Future work

Regarding the multidisciplinary design method, different properties of systems such as energy losses, aerodynamics, cost, sustainability features and other embedded control implementation aspects like communication delays can be considered. This requires extension of the method to include the models and constraints related to each property and discipline. To improve the dynamic behaviour of the method, disturbance and noise models, the impact of uncertainties, and different models of sensors with a variety of applications to measure (e.g. temperature, proximity, acceleration and pressure in a form of complex functions/transfer functions) can be studied.

Another extension to the method can be the integration of management and organizational aspects to the design. Concept generation can be considered as another stage of the design to be incorporated in the method. The research presented in this thesis includes several different theoretical design cases where no physical prototype is built, and no physical experiments are performed. Therefore a good future step would be to apply the presented

method on a designed prototype and examine the feasibility of the solution in the real world with realistic limitations.

Bibliography

[1] R. Bellman, "Dynamic Programming," New Jersey: Princeton University Press, 1957.

[2] Proximedia, "Pareto Front," www.cenaero.be, October 8, 2018.

[3] K. Chen, J. Bankston, J. Panchal, D. Schaefer, "A Framework for Integrated Design of Mechatronic Systems," In: Wang L., Nee A. (eds) Collaborative Design and Planning for Digital Manufacturing. Springer, London, 2009.

[4] D. Malmquist, D. Frede, J. Wikander, "Optimization of mechatronic systems for supporting early design decisions," The 13th Mechatronics Forum International Conference, pp.17-19, Linz, Austria, 2012.

[5] https://www.ansys.com/

[6] M. Mohaqeqi, M. R. Mousavi, and W. Taha, "Conformance testing of cyber-physical systems: A comparative study," in The 14th International Workshop on Automated Verification of Critical Systems, University of Twente, Enschede, Netherlands, 2014.

[7] A. Aminifar, P. Eles, Z. Peng, A. Cervin, K. E. Årzén, "Control-Quality Driven Design of Embedded Control Systems with Stability Guarantees," IEEE Design and Test vol. 35, pp.38-46, 2017.

[8] P. Fritzson, "Principles of Object-Oriented Modeling and Simulation with Modelica 3.3: A Cyber-Physical Approach, 2nd Edition," Hoboken, New Jersey: John Wiley Sons Inc., 2015.

[9] D. Malmquist, "A tool for holistic optimization of mechatronic design concepts," PhD Thesis, KTH, 2014.

[10] B. Wittenmark, J. Nilsson, and M. Törngren, "Timing Problems in Real-Time Control Systems," In: Proceedings of the American Control Conference, pp.2000-2004, 1995.

[11] K. E. Årzen, A. Cervin, J. Eker, and L. Sha, "An Introduction to Control and Scheduling Co-Design," In: Proceedings of the 39th IEEE Conference on Decision and Control., Sydney, Australia, pp.4865-4870, 2000.

[12] R. Isermann, "Mechatronic systems—Innovative products with embedded control," Control Engineering Practice, vol. 16(1), pp.14-29, 2008.

[13] R. Isermann, "Mechatronic Systems Fundamentals," Springer; 2003rd Edition, 2005.

[14] A. Mohebbi, S. Achiche, L. Baron, L. Birglen, "Trends in Concurrent, Multi-Criteria And Optimal Design Of Mechatronic Systems: A Review," 2014 International Conference on Innovative Design and Manufacturing, Montreal, Quebec, Canada, 2014.

[15] R. Curran, M. van Tooren, L. van Dijk, "Systems Engineering Methodology for Concurrent Engineering Education," In: Chou SY., Trappey A., Pokojski J., Smith S. (eds) Global Perspective for Competitive Enterprise, Economy and Ecology. Advanced Concurrent Engineering. Springer, London, 2009.

[16] E. Arai, T. Arai, "Mechatronics for Safety, Security and Dependability in a New Era," Elsevier Science, 1st Edition, 2007.

[17] C. W. de Silva, S. Behbahani, "A design paradigm for mechatronic systems," Mechatronics, vol. 23(8), pp.960-966, 2013.

[18] D. Shetty, R. Kolk, "Mechatronic System Design," Engineering Publications, Toronto, Canada, 1998.

[19] D. Shetty, L. Manzione, A. Ali, "Survey of Mechatronic Techniques in Modern Machine Design," Hindawi Publishing Corporation, Journal of Robotics, 2012.

[20] F. Roos, "Towards a methodology for integrated design of mechatronic servo systems," PhD Thesis, KTH, 2007.

[21] D. Frede, D. Malmquist, J. Wikander, "Holistic Design Optimization in Mechatronics," IFAC Proceedings Volumes, Hangzhou, China, vol. 46(5), pp.655-662, 2013.

[22] D. Malmquist, J. Wikander, D. Frede D, "Optimal design of harmonic drive servo," IEEE/ASME International Conference on Advanced Intelligent Mechatronics: Mechatronics for Human Wellbeing, Wollongong, Australia, pp.1579-1584, 2013.

[23] D. Malmquist, J. Wikander, "Holistically integrated design of a haptic steering wheel," The 14th Mechatronics Forum International Conference, Karlstad, Sweden, 2014.

[24] N. A. Elkateeb, R. I. Badr, "Novel PID Tracking Controller for 2DOF Robotic Manipulator System Based on Artificial Bee Colony Algorithm," Electrical, Control and Communication Engineering, 13(1), 2017.

[25] J. van Amerongen, P. Breedveld, "Modelling of physical systems for the design and control of mechatronic systems," Annual Reviews in Control, vol. 27(1), pp.87-117, 2003.

[26] F. Roos, H. Johansson, and J. Wikander, "Optimal design of motor and gear ratio in mechatronic systems," in Proceedings of the 3rd IFAC symposium on mechatronic systems, 2004.

[27] M. Budinger, J. Liscouët, F. Hospital, J-C. Maré, "Estimation models for the preliminary design of electromechanical actuators," Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering. vol. 226(3), pp.243–259, 2011.

[28] G. Muller, "Systems engineering research methods," Procedia Computer Science, vol. 16, pp.1092-1101, 2013.

[29] J. W. Creswell, "Research Design: Qualitative, Quantitative, and Mixed Methods Approaches," SAGE, 4 edition, 2014.

[30] R. Valerdi and H. L. Davidz, "Empirical research in systems engineering: challenges and opportunities of a new frontier," Systems Engineering, vol. 12(2), pp.169-181, 2009.

[31] T. L. Ferris, "Engineering Design as Research," in Research Methodologies, Innovations and Philosophies in Software Systems Engineering and Information Systems, IGI Global, pp.389-402, 2012.

[32] E. G. Guba, "The alternative paradigm dialog," In E. G. Guba (Ed.), The paradigm dialog, Newbury Park, CA: Sage, pp.17-30, 1990.

[33] M. Q. Patton, "Qualitative evaluation and research methods," Newbury Park. CA: Sage, 2nd ed., 1990.

[34] A. Tashakkori, "Mixed methodology: Combining qualitative and quantitative approaches," Thousand Oaks, CA: Sage, 1998.

[35] D. Morgan, "Paradigms lost and pragmatism regained: Methodological implications of combining qualitative and quantitative methods," Journal of Mixed Methods Research, pp.48-76, 2007.

[36] A. R. Hevner, S. T. March, J. Park and S. Ram, "Design Science in Information Systems Research," MIS Q., vol. 28(1), pp.75-105, 2004.

[37] K. Peffers, T. Tuunanen, M. A. Rothenberger, S. Chatterjee, "A design science research methodology for information systems research," Journal of management information systems, vol. 24(3), pp.45-77, 2007.

[38] C. E. Dickerson, D. N. Mavris, "Architecture and Principles of Systems Engineering," Complex and Enterprise Systems Engineering Series, 2009.

[39] Verein Deutsche Ingenieure, "Design methodology for mechatronic systems," vol. 2206 of VDI-Richtlinien, Beuth Verlag, Berlin, 2004.

[40] U. Shanker Dixit, M Hazarika, J. Paulo Davim, "A Brief History of Mechanical Engineering," 2017.

[41] VDI - Association of German Engineers, "VDI 2206 - Design methodology for mechatronic systems," Beuth-Verlag, Berlin, 2004.

[42] G. Barbieri, C. Fantuzzi, R. Borsari, "A model-based design methodology for the development of mechatronic systems," Mechatronics, 24(7), 2014.

[43] R. Nattermann, R. Anderl, "The W-Model Using Systems Engineering for Adaptronics," Procedia Computer Science, vol. 16, pp.937-946, 2013.

[44] R. Nattermann, R. Anderl, "Approach for a Data-Management-System and a Proceeding-Model for the Development of Adaptronic Systems," International mechanical engineering congress and exposition (ASME), Vancouver, 2010.

[45] G. Pahl, W. Beitz, J. Feldhusen, K-H. Grote, "Engineering Design: A systematic approach," Springer, 2007.

[46] D. Casner, R. Houssin, J. Renaud, D. Knittel, "An optimization-Based Embodiment Design Approach for Mechatronic Product Development," The open Automation and Control Systems Journal, vol. 9(Suppl-1, M4), pp.27-47, 2017.

[47] L. A. Schmit, "Structural Design by Systematic Synthesis," 2nd Conference on Electronic Computation, American Society of Civil Engineers, New York, pp.105-132, 1960.

[48] R. T. Haftka, "Automated Procedure for Design of Wing Structures to Satisfy Strength and Flutter Requirements," NASA Langley Research Center TN-D-7264, Hampton, VA, 1973.

[49] N. Alexandrov, "multidisciplinary design optimization," optimization, Optimization and Engineering, vol. 6(1), pp.5-7, 2005.

[50] J. A. Green, "Aeroelastic Tailoring of Aft-Swept High-Aspect-Ratio Composite Wings," Journal of Aircraft, vol. 24(11), pp.812-819, 1987.

[51] E. Livne, L. Schmit, P. Friedmann, "Towards Integrated Multidisciplinary Synthesis of Actively Controlled Fiber Composite Wings," Journal of Aircraft, vol. 27(12), pp.979-992, 1990.

[52] A. H. Gandomi, X. Yang, S. Talatahari, A. H. Alavi, "Metaheuristic Applications in Structures and Infrastructures," pp.103-127, 2013.

[53] S. A. Brandt, R. J. Stiles, J. J. Bertin, R. Whitford, "Introduction to Aeronautics: A Design Perspective," AIAA Educational Series, Third Edition, 2015.

[54] G. Pahl, W. Beitz, J. Feldhusen, "Engineering design: a systematic approach," Springer-Verlag, London, 2007.

[55] P. Mohammadzadeh, M. Shirazi, "Multi-disciplinary design Optimization based on metamodel," pp.103-127, 2013.

[56] R. M. Lewis, V. Torczon, M. W. Trosset, "Direct search methods: then and now," Journal of Computational and Applied Mathematics, vol. 124(1–2), pp.191-207, 2000.

[57] D. Oliva, M. A. Elaziz, S. Hinojosa, "Metaheuristic Algorithms for Image Segmentation: Theory and Applications," Studies in Computational Intelligence (SCI), 825, 2019.

[58] A. Konak, D. W. Coit, A. E. Smith, "Multi-objective optimization using genetic algorithms: A tutorial," Reliability Engineering System Safety, vol. 91(9), pp.992-1007, 2006.

[59] M. Settles, "An Introduction to Particle Swarm Optimization," GECCO, pp.185-192, 2005.

[60] M. Dorigo, M. Birattari, T. Stutzle, "Ant Colony Optimization," IEEE Computational Intelligence Magazine, vol. 1(4), pp.28-39, 2006.

[61] B. A. Kitchenham, S. M. Charters, "Guidelines for performing Systematic Literature Reviews in Software Engineering," EBSE 2007-001, Keele University and Durham University Joint Report, 2007.

[62] Y. Xiao, M. Watson, "Guidance on Conducting a Systematic Literature Reviews," Journal of Planning Education and Research, vol. 39(1) pp.93-112, 2019.

[63] N. Adamsson, "Interdisciplinary integration in complex product development," PhD Thesis, KTH, 2007.

[64] R. A. de Carvalho, H. da Hora, R. Fernandes, "A process for designing innovative mechatronic products," International Journal of Production Economics, vol. 231, 107887, 2021.

[65] L. Bretz, C. Tschirner, R. Dumitrescu "A concept for managing information in early stages of product engineering by integrating MBSE and workflow management systems," 2016 IEEE International Symposium on Systems Engineering (ISSE), Edinburgh, UK, 2016.

[66] M. Törngren, A. Qamar, M. Biehl, F. Loiret, J. El-khoury, "Integrating viewpoints in the development of mechatronic products," Mechatronics, vol. 24(7), pp.745-762, 2014.

[67] L. Wang, W. Shen, H. Xie, J. Neelamkavil, A. Pardasani, "Collaborative conceptual design—state of the art and future trends," Computer Aided Design, vol. 34(13), pp.981-996, 2002.

[68] M. Mcharek, T. Azib, C. Larouci, M. Hammadi, "Collaboration and multidisciplinary design optimization for mechatronic systems," IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society, Lisbon, Portugal, Portugal, 2019.

[69] M. Mcharek, "Knowledge management for collaborative design and multi-physical optimization of mechatronic systems," Université Paris-Saclay, PhD thesis, 2018.

[70] G. Rzevski, "On conceptual design of intelligent mechatronic systems," Mechatronics, vol. 13(10), pp.1029-1044, 2003.

[71] R. Scheidl, B. Winkler "Model relations between conceptual and detail design," Mechatronics, vol. 20(8), pp.842-849, 2010.

[72] E. Safavi, M. Tarkian, H. Gavel, J. Ölvander, "Collaborative multidisciplinary design optimization: A framework applied on aircraft conceptual system design," Concurrent Engineering, vol. 23(3), pp.236-249, 2015.

[73] S. Liscouet-Hanke, "A Model-Based Methodology for Integrated Preliminary Sizing and Analysis of Aircraft Power System Architectures," Doctoral Thesis, 2008.

[74] M. Hafner, O. Jost, R. Isermann, "Mechatronic design approach for engine management systems," Mechatronics, vol. 12(8), pp.1035-1046, 2002.

[75] S. Awtar, C. Bernard, N. Boklund, A. Master, D. Ueda, K. Craig, "Mechatronic design of a ball-on-plate balancing system," Mechatronics, vol. 12(2), pp.217-228, 2002.

[76] J. Gausemeier, M. Flath, S. Möhringer, "Modeling of Functions of Mechatronic Systems, Exemplified by Tire Pressure Control in Automotive Systems," International Journal of Vehicle Design, Special Issue, 2001.

[77] D. A. Flores-Hernandez, S. Palomino-Resendiz, N. Lozada-Castillo, A. Luviano-Juarez, "Mechatronic design and implementation of a two axes sun tracking photovoltaic system driven by a robotic sensor," Mechatronics, vol. 47, pp.148-159, 2017.

[78] M. N. Rooker, T. Strasser, G. Ebenhofer, M. Hofmann, R. V. Osuna "Modeling Flexible Mechatronical based Assembly Systems through Simulation Support," 2008 IEEE International Conference on Emerging Technologies and Factory Automation, pp.452-455, 2008.

[79] K. Chaaban, N. Rizoug, B. Barbedette, S. Saudrais, "Model-based Development of an Embedded Steering-by-wire System," 2012 8th International Symposium on Mechatronics and its Applications, Sharjah, United Arab Emirates, 2012.

[80] H. Komoto, T. Tomiyama, "A framework for computer-aided conceptual design and its application to system architecting of mechatronics products," Computer-Aided Design, vol. 44, pp.931-946, 2012.

[81] H. Li, M. Zou, G. Hogrefe, D. Ryashentseva, M. Sollfrank, G. Koltun, B. Vogel-Heuser, "Application of a multi-disciplinary design approach in a mechatronic engineering toolchain Application of a multidisciplinary design approach in a mechatronic engineering tool chain," Automation Technology, vol. 67(3), pp.246-269, 2019.

[82] S. Sierla, I. Tumer, N. Papakonstantinou, K. Koskinen, D. Jensen, "Early integration of safety to the mechatronic system design process by the functional failure identification and propagation framework," Mechatronics, vol. 22(2), pp.137-151, 2012.

[83] M. Hammadi, A. Kellner, J-Y. Choley, P. Hehenberger, "Mechatronic Design Optimization using Multi-agent Approach," The 14th Mechatronics Forum International Conference, Karlstad-SWEDEN, 2014.

[84] M. Hammadi, J-Y. Choley, F. Mhenni, "A multi-agent methodology for multi-level modeling of mechatronic systems," Advanced Engineering Informatics, vol. 28(3), pp.208-217, 2014.

[85] H. Yavuz, "An integrated approach to the conceptual design and development of an intelligent autonomous mobile robot," Robotics and Autonomous Systems, vol. 55(6), pp.498-512, 2007.

[86] P. Couturier, M. Lô, A. Imoussaten, V. Chapurlat, J. Montmain, "Tracking the consequences of design decisions in mechatronic systems engineering," Mechatronics, vol. 24(7), pp.763-774, 2014.

[87] J. Qi, J. Hu,Y-H. Peng, "An integrated principle solution synthesis method in multi-disciplinary mechatronic product conceptual design," Concurrent Engineering, vol. 26(4) pp.341-354, 2018.

[88] V. C. Moulianitis, N. A. Aspragathos, A. J. Dentsoras, "A model for concept evaluation in design—an application to mechatronics design of robot grippers," Mechatronics, vol. 14(6), pp.599-622, 2004.

[89] M. Hammadi, J-Y. Choley, O. Penas, A. Rivière, J. Louati, M. Haddar, "A new multicriteria indicator for mechatronic system performance evaluation in preliminary design level," 2012 9th France-Japan and 7th Europe-Asia Congress on Mechatronics (MECATRONICS) / 13th Int'l Workshop on Research and Education in Mechatronics (REM), Paris, France, 2012.

[90] F. Oestersötebier, F. Abrishamchian, C. Lankeit, V. Just, A. Trächtler, "Approach for an Integrated Model-Based Design of Intelligent Dynamic Systems Using Solution and System Knowledge," Procedia Technology, vol. 26, pp.436-446, 2016.

[91] R. Parameshwaran, C. Baskar, T. Karthik, "An integrated framework for mechatronics based product development in a fuzzy environment," Applied Soft Computing, vol. 27, pp.376-390, 2015.

[92] D. Casner, J. Renaud, R. Houssin, D. Knittel, "A novel design approach for mechatronic systems based on multidisciplinary design optimization," ICAM 2012 : International Conference on Automation and Mechatronics, Oslo, Norway. pp.2010-376X/2010-3778, 2012.

[93] G. H. Tuta Navajas, J. A. Parra Raad, S. Roa, Prada, "Concurrent Design Optimization of a Custom Designed Quadcopter," 2015 16th International Conference on Research and Education in Mechatronics (REM), Bochum, Germany, 2015.

[94] J. Ma, S. Chen, N. Kamaldin, C. S. Teo, A. Tay, A. A. Mamun, K. K. Tan, "Integrated Mechatronic Design in the Flexure-Linked Dual-Drive Gantry by Constrained Linear-Quadratic Optimization," IEEE Transactions on Industrial Electronics, vol. 65(3), 2018.

[95] X. Zhiquan, "Concurrent Design of Flexible Manipulator System," Proceedings of the 26th Chinese Control Conference, Zhangjiajie, Hunan, China, 2007.

[96] Y. Lou, W. Gong, J. Shi, G. Liu, Z. Li, "An Integrated Structure/Control Design of Mechatronic Systems," Proceedings of the 7th World Congress on Intelligent Control and Automation, Chongqing, China, 2008.

[97] Y. Zhang, Z. Fang, G. Wu, "Integrated Structure and Control Parameters Optimization for an Automotive Active Suspension System via LMIs," 2009 International Conference on Measuring Technology and Mechatronics Automation, Zhangjiajie, Hunan, China, 2009.

[98] P. Duysinx, O. Bruls, J-F. Collard, P. Fisette, C. Lauwerys, J. Swevers, "Optimization of mechatronic systems: application to a modern car equipped with a semi-active suspension," 6th World Congresses of Structural and Multidisciplinary Optimization, Rio de Janeiro, Brazil, 2005.

[99] https://se.mathworks.com/products/matlab.html

[100] O. Mendoza-Trejo, C. A. Cruz-Villar, "Robust Concurrent Design of a 2-DOF Collaborative Robot (Cobot)," IEEE/ASME Transactions on Mechatronics, 2020.

[101] G. Carabin, R. Vidoni, E. Wehrle, "Energy Saving in Mechatronic Systems Through Optimal Point-to-Point Trajectory Generation via Standard Primitives," The International Conference of IFToMM ITALY, Italy, vol. 68, Springer, 2019.

[102] M. I. Yacoub, D. D. Necsulescu, J. Z. Sasiadek, "Energy Consumption Optimization for Mobile Robots Motion Using Predictive Control," 9th Asian Control Conference (ASCC), Istanbul, Turkey, 2013.

[103] D. Casner, R. Houssin, J. Renaud, D. Knittel, "Optimization as an innovative design approach to improve the performances and the functionalities of mechatronic devices," Procedia Engineering, vol. 131, pp.941-950, 2015.

[104] E. Coelingh, Th. J. A. de Vries and R. Koster, "Assessment of Mechatronic System Performance at an Early Design Stage," IEEE/ASME Transactions on Mechatronics, vol. 7(3), 2002.

[105] J. Jiang, G. Ding, J. Zhang, Y. Zou, Sh. Qin, "A Systematic Optimization Design Method for Complex Mechatronic Products Design and Development," 7th European Systems Engineering Conference EuSEC, Stockholm, Sweden, 2010.

[106] D. Simon, D. Robert, O. Sename, "Control and Real-time Scheduling Co-design Application to Robust Robot Control," 11th IEEE Real Time and Embedded Technology and Applications Symposium, San Francisco, CA. USA, 2005.

[107] D. Simon, A. Seuret, O. Sename, "Real-time control systems : feedback, scheduling and robustness," International Journal of Systems Science, vol. 48(11), pp.2368-2378, 2017.

[108] A. Cervin, J. Eker, "Control-Scheduling Codesign of Real-Time Systems: The Control Server Approach," Journal of Embedded Computing, 2004.

[109] M. Kurdi, P. Beran, "Optimization of Dynamic Response using Temporal Spectral Element Method," In the Proceedings of the 46th AIAA Aerospace Sciences Meeting and Exhibit, No. AIAA-2008-0903, AIAA, Reno, Nevada, 2008.

[110] M. M. da Silva, O. Brüls, W. Desmet, H. Van Brussela, "Integrated structure and control design for mechatronic systems with configuration-dependent dynamics," Mechatronics, vol. 19(6), pp.1016-1025, 2009.

[111] M. M. da Silva, W. Desmet, H Van Brussel, "Design of mechatronic systems with configuration-dependent dynamics: simulation and optimization," 2007 IEEE/ASME international conference on advanced intelligent mechatronics, Zurich, Switzerland, 2007.

[112] U. Killat, P. J. van Duijsen, P. Bauer, "Integrated Simulation of Mechanics, Electronics and Control," IFAC Proceedings Volumes, vol. 39(16), pp.223-228, 2006.

[113] A. Dumlu, "Design of a fractional-order adaptive integral sliding mode controller for the trajectory tracking control of robot manipulators," Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, vol. 232(9), pp.1212-1229, 2018.

[114] Z. Affi, B. El-Kribi, L. Romdhane, "Advanced mechatronic design using a multi-objective genetic algorithm optimization of a motor-driven four-bar system," Mechatronics, vol. 17(9), pp.489-500, 2007.

[115] J. Zheng, Z. Luo, C. Jiang, J. Gao, "Robust topology optimization for concurrent design of dynamic structures under hybrid uncertainties," Mechanical Systems and Signal Processing, vol. 120, pp.540-559, 2019.

[116] S. Behbahani, C. W. de Silva, "Mechatronic Design Evolution Using Bond Graphs and Hybrid Genetic Algorithm With Genetic Programming," IEEE/ASME Transactionc on Mechatronics, vol. 18(1), 2013.

[117] S. Behbahani, C. W. de Silva, "A New Multi-Criteria Mechatronic Design Methodology Using Niching Genetic Algorithm," 2006 IEEE International Conference on Evolutionary Computation,Vancouver, BC, Canada, 2006.

[118] H. Wang, R. Li, J. Miao, X. Zhang, Z. Wang, "Reliability-Based Design Integrating System Topology and Structural Components," 2012 International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering, Chengdu, China, 2012.

[119] F. Mhenni, J-Y. Choley, O. Pena, R. Plateaux, M. Hammadi, "A SysMl-based methodology for mechatronic systems architectural design," Advanced Engineering Informatics, vol. 28(3), pp.218-231, 2014.

[120] A. Makarow, J. Braun, C. Rösmann, G. Schoppel, I. Glowatzky, T. Bertram, "Cascaded Evolutionary Multi-Objective System Optimization for a Proportional Directional Control Valve," Proceedings of the 2018 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Auckland, New Zealand, 2018.

[121] S. M. Karim Tabatabaei, S. Behbahani, C. W. de Silva, "Design evolution and synthesis of multi-domain engineering systems using artificial immune system," Mechatronics (AIM), vol. 39, pp.103-112, 2016.

[122] A. Albers, J. Otttnad, J. Minx, P. Häußler, "Topology Optimization In Mechatronic Systems," Proceedings of ICED 2007, the 16th International Conference on Engineering Design, Paris, France, 2007.

[123] K. W. Lee, W. Moase, A. Ooi, Ch. Manzie, E. C. Kerrigan, "Optimization Framework for Codesign of Controlled Aerodynamic Systems," AIAA Journal., vol. 54(10), pp.3149-3159, 2016.

[124] D. Pekarek, A. D. Ames, J. E. Marsden, "Discrete Mechanics and Optimal Control Applied to the Compass Gait Biped," 46th IEEE Conference on Decision and Control, New Orleans, LA, USA, 2007.

[125] S. Samii, A. Cervin, P. Eles, Z. Peng, "Integrated Scheduling and Synthesis of Control Applications on Distributed Embedded Systems," in Proc. Design, Automation and Test in Europe Conference and Exhibition, Nice, France, 2009.

[126] P. Hehenberger, F. Poltschak, K. Zeman, W. Amrhein, "Hierarchical design models in the mechatronic product development process of synchronous machines," Mechatronics, vol. 20(8), pp.864-875, 2010.

[127] E. A. Padilla-García, A. Rodrguez-Ángeles, J. R. Resendiz, C. A. Cruz Villar, "Concurrent Optimization for Selection and Control of AC Servomotors on the Powertrain of Industrial Robots," IEEE Access, vol. 6, pp.27923-27938, 2018.

[128] M. G. Villarreal-Cervantes, C. A. Cruz-Villar, J. Alvarez-Gallegos, "Structure-Control Mechatronic Design of the planar 5R 2DOF Parallel Robot," 2009 IEEE International Conference on Mechatronics, Malaga, Spain, 2009.

[129] Z. Sun, D. He, B. Zhang, X. Liu, W. J. Zhang, "On a Decoupled Integrated Design Approach to Robotic Mechanisms," 2012 7th IEEE Conference on Industrial Electronics and Applications (ICIEA), Singapore, Singapore, 2012.

[130] R. Chhabra, R. Emami, P. Eles, Z. Peng, "Holistic system modeling in mechatronics," Mechatronics, vol. 21(1), pp.166-175, 2011.

[131] R. Chhabra, R. Emami, P. Eles, Z. Peng, "A holistic approach to concurrent engineering and its application to robotics," Concurrent Engineering, vol. 22(1), pp.48-61, 2013.

[132] R. Chhabra, R. Emami, "A holistic concurrent design approach to robotics using hardware-in-the-loop simulation," Mechatronics, vol. 23(3), pp.335–345, 2013.

[133] D. Trivedi, D. Dienno, C. D. Rahn, "Optimal, Model-Based Design of Soft Robotic Manipulators," ASME. J. Mech. Des., vol. 130(9):091402, 2008.

[134] S. E. Lyshevski, "Optimal structural synthesis, modeling, and control of micro-mechatronic systems," Mechatronics, vol. 11(7), pp.837-851, 2001.

[135] B. Potsaid, J. T. Wen, "Optimal Mechanical Design of a Rotary Inverted Pendulum," IEEE/RSJ International Conference on Intelligent Robots and Systems, Lausanne, Switzerland, Switzerland, 2002.

[136] J. Park and H. Asada, "Concurrent Design Optimization of Mechanical Structure and Control for High Speed Robots," Proc. IEEE Robotics and Automation, San Francisco, CA, USA, 1993.

[137] A. C. Pil and H. Asada (2013), "Integrated structure/control design of mechatronic systems using a recursive experimental optimization method," Mechatronics IEEE/ASME Transactions on Mechatronics, vol. 1(3), pp.191-203, 1996.

[138] G. Domingues, F. J. Márquez-Fernández, P. Fyhr, A. Reinap, M. Andersson, M. Alaküla, "Scalable performance, efficiency and thermal models for electric drive components used in powertrain simulation and optimization," IEEE Transportation Electrification Conference and Expo (ITEC), pp.644-649, Chicago, IL, USA, 2017.

[139] C. Y. Chen, C. C. Cheng, "Integrated Design for a Mechatronic Feed Drive System of Machine Tools," Proceedings, 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Monterey, CA, USA, 2005.

[140] M. O'Driscoll, "Design for Manufacture," Journal of Materials Processing Technology, vol. 122(2-3), 2002.

[141] J. F. Broenink, G. H. Hilderink, J. Eker, L. Sha, "A structured approach to embedded control systems implementation," IEEE International Conference on Control Applications, M. W. Spong, D. Repperger, and J. M. I. Zannatha, Eds. México City, México: IEEE, pp.761-766, 2001.

[142] J. T. Allison, "Plant-Limited Co-Design of an Energy-Efficient Counterbalanced Robotic Manipulator," ASME. J. Mech. Des., vol. 135(10):101003, 2012.

[143] J. T. Allison, "Engineering System Co-Design With Limited Plant Redesign," Engineering Optimization, vol. 46(2), pp.200-217, 2013.

[144] E. A. Padilla-Garcia, A. Rodriguez-Angeles, J. R. Resendiz, C. A. Cruz-Villar, "Engineering System Co-Design With Limited Plant Redesign," Engineering Optimization, vol. 46(2), pp.200-217, 2013.

[145] Y. He, J. McPhee, "Multidisciplinary design optimization of mechatronic vehicles with active suspensions," Journal of Sound and Vibration, vol. 283(1–2), pp.217-241, 2005.

[146] J. T. Allison, D. R. Herber, "Special Section on Multidisciplinary Design Optimization: Multidisciplinary Design Optimization of Dynamic Engineering Systems," AIAA Journal, vol. 52(4), 2014.

[147] Ch. Zheng, J. L. Duigou, M. Bricogne, B.Eynard, "Multidisciplinary interface model for design of mechatronic systems," Computers in Industry, vol. 76, pp.24-37, 2016.

[148] N. Alexandrov, R. Lewis, "Analytical and computational aspects of collaborative optimization for multidisciplinary design," AIAA Journal vol. 40(2), pp.301-309, 2002.

[149] F. Tian, M. Voskuijl, "Automated generation of multiphysics simulation models to support multidisciplinary design optimization," Advanced Engineering Informatics, vol. 29(4), pp.1110-1125, 2015.

[150] F. Tian, M. Voskuijl, "Mechatronic Design and Optimization Using Knowledge-Based Engineering Applied to an Inherently Unstable and Unmanned Aerial Vehicle," IEEE/ASME Transactions on Mechatronics, vol. 21(1) pp.542-554, 2016.

[151] N. Bilel, N. Mohamed, A. Zouhaier, R. Lotfi, "Multi-objective robust design optimization of a mechatronic system with uncertain parameters, using a polynomial chaos expansion method," Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, vol. 231(9), pp.729-739, 2017.

[152] A. Guizani, M. Hammadi, J-Y. Choley, Th. Soriano, "Multidisciplinary approach for optimizing mechatronic systems: Application to the optimal design of an electric vehicle," 2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Besacon, France, 2014.

[153] R. Perez, H. Liu, K. Behdinan, "Evaluation of multidisciplinary optimization approaches for aircraft conceptual design," In: AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, NY., 2004.

[154] R. Citalan-Lara, C. A. Cruz-Villar, "Multidisciplinary Optimization of Servodrives for Robot Manipulators," 2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Besancon, France, 2014.

[155] A. Mohebbi, S. Achiche, L. Baron, "Integrated and concurrent detailed design of a mechatronic quadrotor system using a fuzzy-based particle swarm optimization," Engineering Applications of Artificial Intelligence, vol. 82, pp.192-206, 2019.

[156] H. Liang, M. Zhu, H. Liang, X Guo, "Multidisciplinary design optimization of composite wing by parametric modeling," Proceedings 2013

International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC), Shengyang, China, 2013.

[157] V. Ivanov, M. Ruderman, J. Schiffer, M. Horn, "Interdisciplinary Design Methodology for Systems of Mechatronic Systems," IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society, Beijing, China, 2017.

[158] M. Gulec, M. Aydin, J. Nerg, P. Lindh, J. J. Pyrhönen, "Nonlinear Multidisciplinary Design Approach for Axial-Flux Eddy Current Brakes," IEEE Transactions on Energy Conversion, vol. 34(4), pp.1917-1927, 2019.

[159] H. Chang, W. Li, W. Yuan, B. Ma, "Multidisciplinary Design and Optimization of Integrated Microsystems," 2004 International Conference on Intelligent Mechatronics and Automation, 2004. Proceedings, Chengdu, China, China, 2004.

[160] A. Qamar, J. Wikander, and C. During, "A mechatronic design infrastructure integrating heterogeneous models," IEEE International Conference on Mechatronics, pp.212-217, Istanbul, Turkey, 2011.

[161] G. Barbieri, C. Fantuzzi, and R. Borsari, "Tools for the development of a design methodology for mechatronic systems," IEEE 18th Conference on Emerging Technologies Factory Automation (ETFA), Cagliari, Italy, 2013.

[162] P. Fritzson, V. Engelson, "Modelica - A Unified Object-Oriented Language for System Modeling and Simulation," ECOOP'98 (the 12th European Conference on Object-Oriented Programming), Brussels, Belgium, 1998.

[163] Ir. C. Kleijn, Ir. M. A. Groothuis, "20-sim 4.4 Reference Manual," Control lab Products B.V., 2013.

[164] J. F. Broenink, "20-sim software for hierarchical bond-graph/block-diagram models," Simulation Practice and Theory. vol. 7(5–6), pp.481-492.

[165] J. U. Thoma, "Introduction to Bond Graphs and their Applications," Elsevier, 2016.

[166] J. van Amerongen, "Mechatronic design," Mechatronics, vol. 13(10), pp.1045-1066, 2003.

[167] F.-L. Krause, C. Biantoro, "Modelling and Simulating Approach of Distributed Development Processes in Mechatronics," IFAC Proceedings Volumes, vol. 39(16), pp.336-341, 2006.

[168] V. M. Becerra, P. Beran, "Solving Complex Optimal Control Problems at no Cost with PSOPT," In the Proceedings of the 2010 IEEE International Symposium on Computer-Aided Control System Design, pp.1391-1396, Yokohama, Japan, 2010.

[169] M. Chami, H. B. Ammar, H. Voos, K. Tuyls, G. Weiss, "A Nonparametric Evaluation of SysML-based Mechatronic Conceptual Design," Conference on artificial intelligence, Maastricht, Netherlands, 2012.

[170] M. Chami, J.-M. Bruel, "Towards an Integrated Conceptual Design Evaluation of Mechatronic Systems: The SysDICE Approach," Procedia Computer Science, vol. 51, pp.650-659, 2015.

[171] L. Bassi, C. Secchi, M. Bonfe, C. Fantuzzi, "A SysML-Based Methodology for Manufacturing Machinery Modeling and Design," IEEE/ASME Transactions on Mechatronics, vol. 16(6), 2010.

[172] S. Delbecq, M. Budinger, I. Hazyuk, F. Sanchez, J. Piaton, "A framework for sizing embedded mechatronic systems during preliminary design," IFAC-PapersOnLines, vol. 50(1), pp.4354-4359, 2017.

[173] T. Shi-Xiang, W. Sheng-Ze, "The conceptual design and simulation of mechatronic system base on UML," 2010 2nd International Conference on Computer Engineering and Technology, Chengdu, China, 2010.

[174] Y. Cao, Y. Liu, C. J.J Paredis, "System-level model integration of design and simulation for mechatronic systems based on SysML," Mechatronics, vol. 21(6), pp.1063-1075, 2011.

[175] M. Hochwallner, M. Hörl, S. Dierneder, R. Scheidl, "Some aspects of SysML application in the reverse engineering of mechatronic systems," International Conference on Computer Aided Systems Theory, Springer, vol. 6928, Berlin, Heidelberg, Germany, 2011

[176] M. Zhang, W. Fisher, P. Webb, T-J. Tarn, "Functional Model Based Object-Oriented Development Framework for Mechatronic Systems," Proceedings of the 2003 IEEE International Conference on Robotics and Automation, Taipel, Taiwan, 2003.

[177] J. Desel, G. Juhás, ""What Is a Petri Net?" Informal Answers for the Informed Reader," In: Ehrig H., Padberg J., Juhás G., Rozenberg G. (eds) Unifying Petri Nets. Lecture Notes in Computer Science, vol. 2128. Springer, Berlin, 2001.

[178] J. Holt, S. Perry, "SysML for Systems Engineering," The Institution of Engineering and Technology, 2008.

[179] J. Holt, "UML for Systems Engineering: Watching the Wheels," IET, 2004.

[180] https://ix.ixtronics.com/de/.

[181] The MathWorks web. http://www.mathworks.com, 2008.

[182] A. B. Fotso, A. Rettberg, "State of the art for mechatronic design concepts," Proceedings of 2012 IEEE/ASME 8th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications, Suzhou, China, 2012.

[183] C. V. Lozano, K. K. Vijayan, "Literature review on Cyber Physical Systems Design," Procedia Manufacturing, vol. 45, pp.295-300, 2020.

[184] Wolfram Research, "Inc. (www.wolfram.com), Mathematica Online," Champaign, IL, 2020.

[185] Y. T. Ko, "Optimizing product architecture for complex design," Concurrent Engineering 21, pp.87-102, 2013.

[186] Z. Bi, L. Wang, G. Yang, D. Zhang, "Multidisciplinary Design Optimization in Engineering," Mathematical Problems in Engineering, 2013.

[187] T. Keßler, F. Logist, M. Mangold, "Use of predictor corrector methods for multi-objective optimization of dynamic systems," Computer Aided Chemical Engineering, vol. 38, pp.313-318, 2016.

[188] C. G. Papanicolaou, I. C. Papantoniou, "Optimum design of textile-reinforced concrete as integrated formwork in slabs," Textile Fibre Composites in Civil Engineering, pp.245-274, 2016.

[189] W. L. Hosch, "Genetic algorithm," Encyclopædia Britannica, inc., 2017.

[190] F. Hoffmeister, T. Bäck, "Genetic Algorithm and Evaluation Startegies: Similarities and Differences," Technical Report SYS-1/92, FB Informatik, University of Dortmund: Dortmund, Germany, 1992.

[191] L. Chaparro, "Signals and Systems using MATLAB," Academic Press, 2011.

[192] M. Ahrens, L. Kucera, "Cross Feedback Control of a Magnetic Bearing System: Controller Design Considering Gyroscopic Effects," 1996.

[193] A. Bemporad, "Automatic Control 2: Sampling," University of Trento, Academic year 2010-2011.

[194] Y. Zeng, Ch. Rose, W. Taha, A. Duracz, K. Atkinson, R. Philippsen, R. Cartwright, M. O'Malley, "Modeling Electromechanical Aspects of Cyber-Physical Systems," Journal of Software Engineering for Robotics, vol. 7(1), pp.100-119, 2016.

[195] W. Jakob, C. Blume, "Pareto Optimization or Cascaded Weighted Sum: A Comparison of Concepts," Algorithms, vol. 7(1), pp.166-185, 2014.

[196] R. T. Marler, J. S. Arora, "The weighted sum method for multi-objective optimization: new insights," Structural and Multidisciplinary Optimization, vol. 41, pp.853-862, 2010.

[197] E. Coelingh, Th. J. A. de Vries and R. Koster, "Assessment of Mechatronic System Performance at an Early Design Stage," IEEE/ASME Transactions on Mechatronics, vol. 7(3), 2002.

[198] O. Grodzevich1, O. Romanko, "Normalization and Other Topics in MultiObjective Optimization," Proceedings of the Fields–MITACS Industrial Problems Workshop, 2006.

[199] K. Janschek, "Mechatronic systems design," Berlin Heidelberg: Springer-Verlag, 2012.

[200] "Ball screws," SKF, Publication 6971, 2009.

[201] B. J. Hamrock, S. R. Schmid, B. O. Jacobson, "Fundamentals of machine elements," 2004.

[202] D. Saini and P. Gaur, "Control of 2-DOF robotic manipulator using brushless DC motor to track the motion of object in a plane," IEEE 5th India International Conference on Power Electronics (IICPE), 1–4, Delhi, India, 2012.

[203] A. Okubanjo, O. Oluwadamilola, O. Martins, "Modeling of 2-DOF Robot Arm and Control," Federal University of Technology Owerri, Journal Series (FUTOJNLS), vol. 3(2), pp.80-32, 2017.

[204] M. Ö. Efe, "Fractional Fuzzy Adaptive Sliding-Mode Control of a 2-DOF Direct-Drive Robot Arm," IEEE Transactions on Systems, Man, and Cybernetics, Part B, vol. 38(6), pp.1561-1570, 2008.

[205] S. M. Mahil, A. Al-Durra, "Modeling analysis and simulation of 2-DOF robotic manipulator," IEEE 59th International Midwest Symposium on Circuits and Systems (MWSCAS), March 2017, 1–4, Abu Dhabi, United Arab, 2016.

[206] J. Carvill, "Mechanical Engineer's Data Handbook," Elsevier, 2012.

[207] R. Perneder, I. Osborne, "Handbook Timing Belts", Springer-Verlag Berlin Heidelberg, 2012.

[208] J. Awrejcewicz, Z. Koruba, "Classical Mechanics: Applied Mechanics and Mechatronics," Springer-Verlag, Berlin, 250 pp. (ISBN 978-1-4614-3977-6), 2012.

[209] K. Åström and B. Wittenmark, "Computer-controlled systems: theory and design, Edition, 3," Upper Saddle River, NJ: Prentice Hall, 1997.

[210] J. Fraden, "Sensor Characteristics. In: Handbook of Modern Sensors," Springer, Cham, 2016.

[211] F. Ferrante, "On quantization and sporadic measurements in control systems: stability, stabilization, and observer design," PhD thesis, Université de Toulouse, 2015.

[212] G. Ellis, "Control System Design Guide," Butterworth-Heinemann, pp.407-429, 2012.

[213] "Barnes new ball screws, new acme screws reconditioning services", Barnes Industries, Inc.

[214] "Ball screw", THK General Catalog.

[215] S. N. Akour, "Dynamics of Nonlinear Beam on Elastic Foundation," Proceedings of the World Congress on Engineering, vol. 2, London, U.K, 2010.

[216] F. A. Salem, "Modeling, simulation and control issues for a robot ARM," International Journal of Intelligent Systems and Applications, vol. 6(4), pp.26-39, 2014.

[217] A. A. Mohammed, M. Sunar, "Kinematics modeling of a 4-DOF robotic arm," 2015 International Conference on Control, Automation and Robotics, Singapore, 2015.

[218] I. Virgala, M. Kelemen, M. Varga, K. Piotr, "Analyzing, Modeling and Simulation of Humanoid Robot Hand Motion," Procedia Engineering, vol. 96, pp.489-499, 2014.

[219] V. Patidar, R. Tiwari, "Survey of robotic arm and parameters," 2016 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 2016.

[220] A. R. Upson and J. H. Batchelor, "Synchro Engineering Handbook," Beckenham: Muirhead Vactric Components, 1978.

[221] A. G. Corbet and W. Burger, "Ship Stabilizers: A Handbook for Merchant Navy Officers," pp.55-79, 1966.

[222] D. Malmquist, D. Frede, J. Wikander, "Holistic design methodology for mechatronic systems," Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, vol. 228(10), pp.741-757, 2014.

[223] D. Pekarek, A. D. Ames, J. E. Marsden, "Discrete Mechanics and Optimal Control Applied to the Compass Gait Biped," 46th IEEE Conference on Decision and Control, New Orleans, LA, USA, 2007.

[224] T. Luukkonen, "Modelling and control of quadcopter," Independent research project in applied mathematics, Aalto University, Espoo, 2011.

Appended papers