

A Design Expert System for Guidance and Assessment of the  
Design Evolution of Motion Control Systems

by

Ibrahim M. Gadala

B.A.Sc., The University of British Columbia, 2010

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF  
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

(Mechanical Engineering)

The University of British Columbia  
(Vancouver)

March 2012

© Ibrahim M. Gadala, 2012

# Abstract

The automation of engineering design is of great significance in the development of machinery and products in competitive industries. Using an automated and “optimal” design process to redesign the faulty components and poorly performing regions of an existing engineering system will facilitate the realization of realistic design alternatives, with benefits such as improved quality, reliability, and cost effectiveness. Motivated by such needs, this thesis develops a Design Expert System (DES) for motion control (MC) applications. The developed DES is expected to be integrated into a multi-system Evolutionary Design Framework (EDF) which is being developed in our laboratory. The EDF integrates techniques of condition monitoring, modeling, and evolutionary optimization for autonomous identification, diagnosis, and redesign of poorly performing aspects of an existing machine. Through integration with optimization routines and the use of a comprehensive knowledge base (KB) in the MC domain, the DES developed in this work is able to guide the evolution of optimal design alternatives and assess their feasibility and effectiveness.

Due to the prevalence of electric motors as actuators in many industrial applications, MC design and actuator (motor) selection represent the application domain of the DES developed in the present research. The KB of the DES includes knowledge of typical mechanical structures used in industrial MC systems, common profiles of load speed or position (duty cycles), and the effect of practical issues such as s-curve profiling, geometric trajectory blending, intermittent duty cycles, rms torque, and the thermal response of motors. A systematic methodology for detailed design analysis and subsequent selection of commercially available motors, their drive systems, and

transmission devices (e.g., gears) from an external database is developed. Selections by the DES are compared to those by a human designer for both hypothetical and actual designs, thereby verifying the DES procedure. To facilitate the interaction between different systems in the EDF, a graphical user interface (GUI) is created for the DES in Excel<sup>®</sup>. The DES is synchronized with Matlab<sup>®</sup> to guide optimization routines based on its built-in human expertise and heuristic design knowledge. A guided optimization case study is presented and benefits of the guidance process are discussed.

# Table of Contents

Abstract .....	ii
Table of Contents .....	iv
List of Tables .....	vi
List of Figures .....	vii
List of Abbreviations .....	ix
Acknowledgements .....	x
Dedication .....	xi
<b>Chapter 1 Introduction .....</b>	<b>1</b>
1.1 Engineering Design .....	1
1.2 Motivation .....	3
1.3 Evolutionary Design Framework .....	6
1.3.1 Machine Health Monitoring System (MHMS) .....	8
1.3.2 Modeling & Evolutionary Design Optimization System (MEDOS) .....	9
1.3.3 Design Expert System (DES) .....	10
1.4 Expert Systems .....	14
1.4.1 Fundamentals .....	14
1.4.2 Literature Survey of Applications .....	18
1.5 Research Objectives .....	21
1.6 Organization of the Thesis .....	22
<b>Chapter 2 Application Domain .....</b>	<b>24</b>
2.1 Motion Control Domain .....	24
2.2 Motor Selection .....	26
2.2.1 Importance .....	26
2.2.2 Methodology .....	28
2.3 Inertia of Mechanical Structure .....	32
2.3.1 Generic Applications .....	32
2.3.2 Custom Applications .....	33
2.3.3 Custom Components .....	34
2.3.4 Effect of Gearheads on Inertia .....	35
2.4 Motor and Gear Databases .....	37
<b>Chapter 3 System Implementation .....</b>	<b>40</b>
3.1 Introduction .....	40
3.2 Motion Profile Options .....	41
3.3 Load Speed Profiles .....	43
3.3.1 Generic and Custom Speed Profiles .....	43
3.3.2 S-Curve Profiling .....	46
3.4 Load Position Profiles .....	48

3.4.1	Transition Window Technique .....	49
3.4.2	Generic and Custom Position Profiles .....	51
3.5	Practical Considerations.....	55
3.5.1	Effect of Mechanical Structure .....	55
3.5.2	RMS Torque.....	57
3.5.3	Thermal Response.....	61
3.6	Implementation and Verification .....	63
3.6.1	Program Segments .....	63
3.6.2	Synchronization with Microsoft Excel® .....	66
3.6.3	Verification .....	67
<b>Chapter 4</b>	<b>Optimization of Mechanical Components .....</b>	<b>73</b>
4.1	Introduction.....	73
4.2	Constrained Minimization .....	76
4.2.1	Optimality Conditions.....	77
4.2.2	Sequential Quadratic Programming.....	82
4.3	Case Study and MCDES Integration .....	83
4.4	Problem Formulation and Results.....	86
4.4.1	Hollow shaft under static torque.....	86
4.4.2	Hollow shaft with combined loading.....	94
4.4.3	Belt Pulley Drive.....	98
<b>Chapter 5</b>	<b>Conclusions.....</b>	<b>103</b>
5.1	Synopsis and Contributions .....	103
5.2	Significance.....	105
5.3	Suggestions for Future Work.....	105
	References.....	107
	Appendix A – Inertia Equations .....	113
	Appendix B – Motor and Gear Data .....	120
	Appendix C – MCDES Code.....	122
	Appendix D – Conveyor Specifications .....	123

# List of Tables

Table 2.1: Inertia of GA in motor-gear drive scenario .....	36
Table 2.2: Sample specifications for motors used in MCDES .....	39
Table 2.3: Sample specifications for gears used in MCDES .....	39
Table 3.1: Effect of MS on profile parameters .....	56
Table 3.2: Location of rule groups for generic MS analysis.....	64
Table 3.3: Location of rule groups for custom MS analysis.....	65
Table 3.4: Verification of generic MS and DC.....	68
Table 3.5: Design data for custom conveyor belt MS .....	72
Table 4.1: Sources of required parameters for optimization.....	86
Table 4.2: Parameter values for Figure 4.7 .....	89
Table 4.3: Parameter values for results of Table 4.4 .....	92
Table 4.4: Case 1 optimum solutions vs. shaft material & impact on required $T_{motor}$ ....	92
Table 4.5: Case 1 optimum solutions vs. human judgment and impact on required $T_{motor}$ ..	93
Table 4.6: Case 2 optimum solutions vs. shaft material & impact on required $T_{motor}$ ....	97
Table 4.7: Case 3 geometric and design parameters.....	99
Table 4.8: Parameter values for BPD optimization .....	101
Table 4.9: Case 3 optimum solutions vs. belt specification & impact on $T_{motor}$ .....	102
Table A.1: Equivalent polar moments of inertia of MCDES GA.....	113
Table A.2: Equivalent polar moments of inertia of MCDES custom components.....	117
Table B.1: Full specifications list of motors used in MCDES.....	120
Table B.2: Full specifications list of gears used in MCDES .....	121

# List of Figures

Figure 1.1: Evolutionary Design Framework .....	7
Figure 1.2: Knowledge acquisition and representation in ES development .....	15
Figure 1.3: Developed rule-based ES elements and structure of ES .....	18
Figure 1.4: Design requirement input GUI for cogeneration plant DES .....	21
Figure 2.1: Objective for a fully completed automatic design evolution routine .....	25
Figure 2.2: Variation of motor efficiency with loading.....	27
Figure 2.3: MS and DC characterization procedure in MCDES .....	30
Figure 2.4: Selection procedure of the MCDES .....	31
Figure 2.5: Examples of real designs which can be built by the MCDES custom design feature .....	33
Figure 2.6: Sample catalogue sheets for motors and gears used in MCDES.....	38
Figure 3.1: Motion speed profile options in MCDES.....	42
Figure 3.2: Motion position profile options in MCDES .....	42
Figure 3.3: Triangular load speed profile .....	43
Figure 3.4: Trapezoidal load speed profile .....	44
Figure 3.5: Custom speed profile example and relevant parameters .....	45
Figure 3.6: 100% s-curve profiling of a trapezoidal LSP and effect on torque .....	46
Figure 3.7: Generic s-curve profiled acceleration ramp .....	48
Figure 3.8: One-dimensional path segment transition .....	49
Figure 3.9: Single forward stroke LPP .....	51
Figure 3.10: Double stroke LPP.....	54
Figure 3.11: Operating range of sample motor from MCDES database.....	60

Figure 3.12: Part of the MCDES GUI template in Excel® .....	66
Figure 3.13: Existing conveyor belt setup used for custom verification .....	71
Figure 4.1: Conventional design process .....	74
Figure 4.2: Optimum design process .....	75
Figure 4.3: A feasible region and direction in nonlinear multi-constraint minimization .	80
Figure 4.4: High-level custom hoister design with modification components .....	84
Figure 4.5: Flow of a design process with integrated DOAO and MCDES .....	85
Figure 4.6: 3D plot of the objective function for Case 1 .....	88
Figure 4.7: Objective function contour and constraint plot for Case 1 using 4140 Alloy Steel.....	89
Figure 4.8: Objective function contour and constraint plot for Case 1 using Magnesium A261 Alloy.....	90
Figure 4.9: Sample screenshot of Case 1 numerical optimization.....	93
Figure 4.10: Problem basis of Case 2 optimization .....	94
Figure 4.11: Graphical comparison of 4140 Alloy Steel and Magnesium A261 Alloy in Case 2 optimization.....	96
Figure 4.12: Sample screenshot of case 2 numerical optimization.....	97
Figure 4.13: Geometric and design parameters for flat BPD design and optimization ....	98
Figure D.1: Geometric parameters of custom conveyor setup used for verification .....	123
Figure D.2: Engineering drawing of custom conveyor setup used for verification.....	124

# List of Abbreviations

AI	Artificial Intelligence
BPD	Belt Pulley Drive
BG	Bond Graphs
CFD	Computational Fluid Dynamics
CAD	Computer Aided Design
DES	Design Expert System
DOAO	Design Optimization Add On
DD	Direct Drive
DC	Duty Cycle
EDF	Evolutionary Design Framework
MEDOS	Modeling & Evolutionary Design Optimization System
ES	Expert System
FEA	Finite Element Analysis
GA	Generic Application
GP	Genetic Programming
GUI	Graphical User Interface
KA	Knowledge Acquisition
KB	Knowledge Base
LHS	Left Hand Side
LG	Linear Graphs
LPP	Load Position Profile
LSP	Load Speed Profile
MHMS	Machine Health Monitoring System
MS	Mechanical Structure
MC	Motion Control
MCDES	Motion Control Design Expert System
MP	Motion Profile
MPP	Motion Profile Processing
QP	Quadratic Programming
RMS	Root Mean Squared
SQP	Sequential Quadratic Programming
SS	Stainless Steel
WRT	With Respect To

# Acknowledgements

I wish to sincerely thank my supervisor and mentor, Professor Clarence W. de Silva, for his instrumental role in the completion of this work. During my senior undergraduate years at UBC, I had the opportunity to be taught by him and was always captivated by his vast knowledge and unique style. It has truly been a privilege and honor to work under his supervision and guidance for my Master's degree. I thank him for his endless patience, assistance, and encouragement. In addition, the generous financial support provided by him in the form of a Research Assistantship is greatly appreciated.

Funding for the research presented in this thesis has been provided through grants from the Natural Sciences and Engineering Research Council (NSERC), Tier 1 Canada Research Chair (CRC), Canada Foundation for Innovation (CFI), and British Columbia Knowledge Development Fund (BCKDF).

Special gratitude goes to all my colleagues at the Industrial Automation Laboratory (IAL) for their valuable friendship and positive advice ever since I came to the lab in September 2010, notably Dr. Khan, Dr. Al-Rasheed, Buddhika, Shahram, Tufail, Roland, Edward, and more recently, Hani. I also sincerely thank the IAL student volunteer from the Computer Science department, William Li, for his help in writing and editing codes.

No words can describe how grateful I feel for the love and backing of all my family members. I am greatly indebted to them all, from my mother for her unconditional love and concern, my father, Dr. Mohamed Gadala, for his wise advice and encouragement, to my beautiful sisters Marwa and Mariam for their affection, even when living abroad. I only wish I am as good a son and brother as they are parents and sisters.

*To the Martyrs of the  
Arab Spring Revolutions*

# Chapter 1

## Introduction

### 1.1 Engineering Design

Engineering design is a process of multi-stage decision making. Successful designers are able to efficiently and adequately satisfy various theoretical and practical constraints of a design problem. In many cases, these constraints can be contradictory. Such trade-offs are very often handled using instinctive and intuitive judgements that rely primarily on empirical knowledge and experience. Yet, in more complicated and stringent design scenarios, a well-defined systematic approach is needed for the design process. The conflicting constraints and the risks involved of sacrificing certain aspects or areas of the design are too difficult to handle intuitively. Furthermore, many steps are needed to translate the design requirements and constraints into a functioning engineering system. For every step, a highly iterative process of comprehensive analysis and detailed examination is needed [1]. The steps are also inter-related, so decisions made in one step almost certainly affect the others and vice versa. Thus, an efficient methodical design approach must be used.

The usefulness of a well-defined design or product development process can be summarized in the following manner [2]:

- **Quality Assurance:** When the steps and check-points of a design process are chosen and followed properly, they can significantly contribute to assuring the quality of the resulting product.

- **Coordination:** A properly developed and clearly articulated design process acts as a master plan which defines the steps of the process and the roles of each of the members and subsystems involved.
- **Management and Improvement:** The design process can serve as a benchmark for assessing the performance of the ongoing design effort. By comparing the actual process to the established process and by documenting the design effort, areas with possible problems and opportunities for improvement can be identified.

In view of the mentioned and more, significant work has been done in streamlining the process of design and product development in recent years [3, 4]. Many designers or organizations define and follow precise and detailed processes that are developed for the design purposes, while others may not be able to clearly describe or formulate the steps and processes they use in the design. Although every designer or organization does not employ the same process or steps in design, there remains a common high-level procedure which all seem to follow. The steps of this “generic design process” are [2]: Planning, Concept Development, System-level Design, Detailed Design, Testing and Refinement, and Production.

Even with the aforementioned advantages, the employment of systematic methodologies for engineering design does not eliminate the occurrence of failures due to poor or bad design. The design of “optimal” products that are safe and reliable, for the ever-increasing user needs and requirements is still a very challenging task. For this reason, several tools and techniques have been developed to support and enhance the steps of the design process. One such tool which has had an enormous impact on nearly all stages of the design process is Computer-Aided Design (CAD). The use of computers

in design has greatly improved the ability to carry out numerous steps of the design process ranging from conceptual design (e.g., solid-modeling software) to detailed design, analysis (e.g., FEA and CFD software), and testing.

Despite its complexity, the design of engineering systems can be robustly performed by engineering experts, a fact which suggests the use of artificial intelligence (AI) in the design process [5]. An important branch of AI with tangible and proven realizations is Expert System (ES). ES have been applied to many different domains of engineering design and have been successfully deployed by large corporations in tasks of assistive design and consultation [6, 7]. Nowadays, ES which can function autonomously or with minimal human input in automated design environments are of great interest.

In the present thesis, an ES is developed to operate in an environment of automated design evolution. In this context, the ES contains practical (analytical and heuristic) knowledge of motion control (MC) design and follows a methodological process of analyzing the design proposals for feasibility and optimality. The growth and expansion of the ES in the future has been taken into serious consideration in the developments of the present thesis. Thus, features which allow the easy integration of the ES with health monitoring, design evolution, and design optimization schemes are developed and tested in the present work.

## **1.2 Motivation**

The automation of engineering design and product development can be of great significance in competitive industries. An autonomous design process with competent features can greatly reduce the lead-time and costs of development of a product. Automated systems with a strong knowledge base and rigorous optimization and

inference schemes can even improve the quality and reliability of designed products while offering economic and logistic benefits. The use of automated design systems in the improvement or “evolution” of existing designs, while giving particular attention to “weak” or “faulty” aspects or regions of the design is the focus of this research.

Prior to improving the design performance of an existing system, detection of the weak areas of the design is paramount. An expectation in the implementation of the overall framework in which the present research an integral part, is that a machine health monitoring system (MHMS) which can detect operational faults or malfunctions in the system can also point to possible design weaknesses in the system. Hence, in an automated design evolution system, as will be explained in detail in the next section, the early detection of fault occurrences is a critical starting point of the evolution process. Traditionally, a MHMS is important in avoiding product deterioration, performance degradation, major damage to the machinery itself, and harm to human health or even loss of life [8]. Current approaches in this step succeed in detecting “faulty” or degraded operation of target components or systems, but cannot identify whether the degraded or unacceptable performance is a result of a design weakness (design problem) or simply misuse, wear, improper installation, or any other fault in a component or of the system. A design ES can help in classifying data from the detection step into a fault versus a design weakness. However, this task is not the focus of the ES that is developed in the present thesis. This work assumes that there is an intermediate decision making process between the MHMS and the framework of design evolution that can separate a region (area) or aspect of weak design from a faulty part of the system.

Once an area of design weakness in a system is identified, evolving a new and improved design proposal is needed. First, a representation of the system being evolved is required. This is a “model” of the system. Various types of models may be incorporated for this purpose, ranging from analytical models to experimental models. Within the present framework, analytical modeling is being investigated by our research team. Different modeling tools with ranging capabilities may be used for analytical modeling. Our team is investigating such tools as Linear Graphs (LG) and Bond Graphs (BG) in achieving the goals of the developed system framework. Evolution procedures are then used to grow formations in the models that will fulfil given specifications and constraints with respect to system parameters, a target response or an objective. Weighting functions, performance indices, or cost functions based on such important requirements may be used to “optimize” the design. This will determine the best performing model formations which will lead to “optimized” design alternative proposals for the system.

In order for this process to generate practically useful design alternatives, the evolution procedure must be assisted and guided by human expert(s) who can judge the practicality of intermediate results and the trend towards which the solution is moving. Then, by updating the requirements and constraints or even restarting another evolution with fundamental modifications, the evolution process can be geared towards a feasible direction for the system. The final proposals for design alternatives must also be judged for feasibility by the expert(s). In an automated system, the human expert(s) in this context are replaced with an ES. A successful ES replacement of the human expert(s) will result in the following advantages:

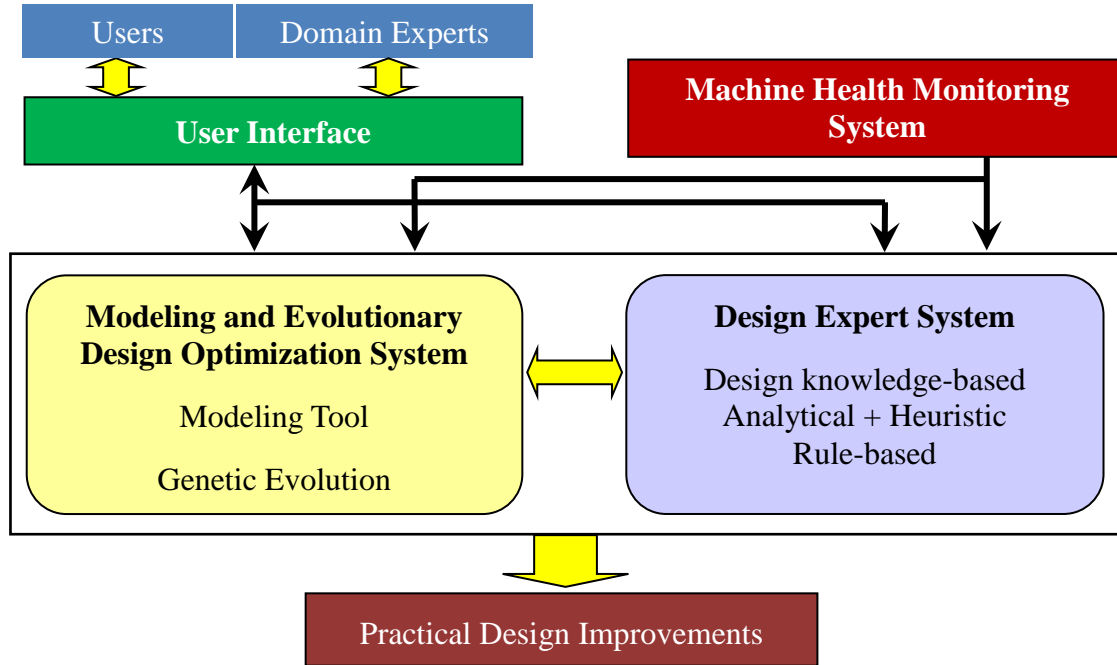
- Significantly faster process time from the detection of design weaknesses to the proposal of feasible design alternatives.
- Considerably cheaper design and operation cost of the overall system, due to the elimination of a dedicated human element from these activities.
- Possibilities for larger number of iterations, larger and regularly updated knowledge databases, and integration of multiple knowledge domains, due to the superior computational power and storage capacity of modern computers.

For these reasons, in the present thesis, an ES is developed to independently assess the feasibility of proposed designs in a specified domain and help guide the design optimization process in more practical directions using built-in expertise as acquired from domain experts.

### **1.3 Evolutionary Design Framework**

The ES that is developed in this research is built to work within an Evolutionary Design Framework (EDF), as proposed by de Silva in 2008 [9]. The EDF uses on-line monitoring of machine health, multi-domain modeling, design expert systems, and evolutionary optimization for the automated design evolution of engineering/mechatronic systems [9, 10]. Figure 1.1 is an illustration of the various components of the proposed EDF and their interactions with one another.

The EDF contains three main sub-systems: a Machine Health Monitoring System (MHMS), a Modeling and Evolutionary Design Optimization System (MEDOS), and a Design Expert System (DES). In the MHMS and MEDOS, numerous technologies have been researched, developed, and deployed by the research team in the Industrial Automation Laboratory to provide fault detection and design evolution, respectively.



**Figure 1.1: Evolutionary Design Framework.**

These developments will be highlighted in the upcoming sections. The test-bed for these technologies is an existing industrial scale mechatronic system called the Iron Butcher (IB). The Iron Butcher is an intelligent fish processing machine which minimizes the wastage of useful meat during the fish head cutting process. The IB was designed and developed by the researchers of the Industrial Automation Lab, in a previous project with B.C. Packers Ltd. The developed machine was transferred to the company. A prototype of the machine is located in our laboratory at the University of British Columbia. During the development of this machine, its design could not be optimized due limitations of the available technologies and resources. In this backdrop, the IB is an ideal industrial system for testing the developments of the EDF.

In the context of the EDF, the research reported in the present thesis is the first dedicated work that addresses the important duties of the DES. The development of a specialized DES and illustration of its utility are the core contributions of the present

work. The accomplishments related to this end objective are presented throughout the thesis.

### **1.3.1 Machine Health Monitoring System (MHMS)**

In manufacturing and processing industries, maintenance costs are responsible for approximately half of all operating costs [11]. The use of monitoring based maintenance programs is advantageous for industries due to the lower downtimes, reduced maintenance costs, and improved safety which these programs offer. The MHMS is a signal condition monitoring scheme used to predict, detect, and diagnose faults and malfunctions of a system or a component. Various schemes for the MHMS have been developed and tested on the Iron Butcher, an existing industrial scale mechatronic system.

In the MHMS schemes that have been developed and tested on the Iron Butcher, different sensors have been used for signal acquisition. In the work done by Raman and de Silva [12], vibration and acoustic signals were acquired through accelerometers and microphones strategically placed on the machine. Razavi and de Silva [8] later developed a scheme which monitored the condition of the machine's hydraulic positioning system. They used pressure and position transducers to monitor the crucial performance parameters of the hydraulics.

In the scheme developed by Raman and de Silva, the vibration and acoustic signals were represented as a feature vector using wavelet packet transform and node energies. A genetic algorithm for feature selection was developed and tested on the system using one healthy machine condition and five faulty conditions. Classification of the faults was done through two popular methods, the Support Vector Machines and

Radial Basis Function neural networks. In [8], the machine health monitoring scheme revolves around the use of an Unscented Kalman Filter. The scheme was tested on four deliberately introduced hydraulic faults such as external/internal leakage of the cylinders and dry friction buildup on the positioning table (actuated by the hydraulic cylinder). Another scheme developed for the MHMS was that of Lang and de Silva [13] in 2008. In this scheme of sensor fusion, signal acquisition was done through an accelerometer, microphone and CCD camera. Faults in the Iron Butcher were detected and classified through sensor fusion using an integrated approach of fuzzy logic and neural network.

### **1.3.2 Modeling & Evolutionary Design Optimization System (MEDOS)**

In the EDF, the MEDOS is primarily responsible for the development of new design proposals using search and optimization techniques. Optimization techniques are used to develop configurations that fulfill given constraints such as the parameters for a target response or objective. To date, the main search mechanism used by the MEDOS is Genetic Programming (GP). Inspired by biological evolution, Genetic Programming is a technique which is used to create problem-solving computer programs. GP was developed by Koza [14, 15] in 1992 and is a specialization of Genetic Algorithms (GA), as developed in 1975 by Holland [16]. GP is commonly used for search and optimization mechanisms. In system design, it is often integrated with modeling techniques such as BG to allow for the expansion of the design space into multiple domains. Seo et al. [17] were the first to propose this GP-BG integration, and many other works followed suit, such as Behbahani and de Silva [18] and Samarakoon and de Silva [19].

The modeling stage of the MEDOS is an important prerequisite to smooth optimization and configuration development. Dynamic models, which are used to

represent an engineering system using effective modeling tools, can be very helpful in understanding the physical interactions and mathematical relationships governing each component in the system. The modeling technique commonly used in the MEDOS is LG. LG is a graphical tool for developing and representing lumped-parameter models of dynamic systems [9] even though BG has been considered as well, prior to that. LG is an attractive modeling technique for the MEDOS due to many reasons, such as: a one-to-one correlation with the physical system, providing a unified approach to model multi-domain systems, and the presence of a systematic methodology to obtain a unique state-space model corresponding to an LG model. In addition, there is the promise in the extension of LG techniques to distributed parameter and nonlinear systems [20]. A detailed discussion of modeling techniques such as LG and BG is outside the scope of this thesis, but can be found in books such as [21, 22, or 23].

### **1.3.3 Design Expert System (DES)**

In the context of the EDF, the DES is essential in that it provides a much needed human judgment element to the overall process. The DES harbors domain-specific human expertise in design and general common sense judgment in a system which can operate at a much lower cost and higher speed than manually by a human expert. The duties of the DES in the context of the EDF can be divided into three broad categories. In the order of their occurrence in the overall EDF, each of these categories is explained below:

- **Detection**

When the MHMS monitors the machine of interest and processes the acquired signal data, faulty components or operation can be detected and diagnosed, and in some cases, predicted. However, in order to detect underlying trends in the data

that might correlate to a weakness in the design of either a specific component or a multi-component interaction within a subsystem, the intervention of an expert engineer is required. With time, the expert can distinguish between failures or malfunctions simply due to wear, lack of maintenance, or inappropriate commissioning and human error, and poor performance and/or failure resulting from weakness in the design (e.g., undersized actuators, improperly matched transmissions and other coupling components, excessive vibration due to improper damping, poor material choice, etc.). Using built-in human expertise and judgement principles, the DES should contribute to the automation of this classification procedure.

- **Supervision and Guidance**

In the interaction between the DES and the MEDOS, the DES should supervise the selection, search and optimization process of the MEDOS and guide the process towards more practical solutions. For example, the selection process for modification (crossover, mutation, etc.) in a typical evolutionary optimization process involves a degree of randomness. The feasibility issues are not incorporated in them. The DES should intervene in the process if intermediate solutions read are deemed inadequate or infeasible, and might decide on a less “optimal” yet feasible solution or even reinitiate the optimization process from the beginning using refined search criteria or design constraints.

Supervision is a demanding task for the DES since it requires online monitoring of the optimization process and continuous translation of the solution from the form that is being optimized in the evolutionary scheme into a form that

the DES can handle and make judgements about the results. The fact that LG modeling provides one-to-one correlation between the model and the actual system proves to be very important here. The guidance task is less intensive on the DES since it mainly requires the quick transfer of pertinent design expertise to the MEDOS as needed. This knowledge transfer can: create new design constraints, modify existing criteria, select a less optimal yet feasible solution, or even terminate the optimization routine and restart a new one with fundamental changes in the overall formulation. The effect of this knowledge transfer is the improvement of the feasibility of the design alternative.

In Chapter 4 of this thesis, a pilot scale method for the DES guidance of an optimization routine is proposed and a synchronization of the main DES that is developed in earlier chapters with a gradient-based optimization routine is performed. The DES feeds the optimization routine with domain-specific human knowledge on constraints, data, and practical design parameter ranges to improve the optimization results. The effect this guidance has on the overall proposed design alternative is measured.

- **Feasibility**

Even after performing a supervised and guided optimization routine, a feasible alternative design proposal made by the MEDOS is not guaranteed. Thus, an important DES duty is to assess the proposed MEDOS design alternative for feasibility. This feasibility analysis might require rigorous computations and frequent iterations, therefore, cannot simply be amalgamated with the guidance duty described above. Without doubt, the process of sending criteria and pertinent

constraint knowledge to the MEDOS is much less intensive on the DES than performing detailed analysis on the design alternatives that are proposed by the MEDOS. Furthermore, during supervision, the computation load on the DES is already high and hence should not be further burdened with a demanding feasibility analysis of every possible design alternative. For that reason, the feasibility assessment is only done after a final proposed design alternative is reached through a supervised and guided MEDOS run.

In an engineering design context, feasibility is a measure of whether a design can be accomplished within some reasonable limits. A feasibility analysis can entail a wide range of assessment indices. For example, a proposed design can be assessed for the interrelated indices of cost, size, complexity, and ease of manufacture. Performance in some of these indices can be measured in a simple manner, such as a component count for assessment of a design's complexity, while others may require more rigorous analysis.

In specific, the feasibility analysis performed by the DES targets more open-ended design questions like “can this design be built,” “what are the limitations of the individual components and the requirements of the actuator,” or “are the necessary components commercially available.” To answer such questions, the DES should contain knowledge of, for example, reasonable ranges of many component parameters (shaft diameters, gear ratios, belt thicknesses etc.), procedures for assessing design integrity (structurally, thermally etc.) and actuation (motor selection), and catalogues of purchasable components and

actuators to confirm that the design is buildable without unduly inflating the cost and resource and time requirements.

In this thesis, a DES is developed for the detailed assessment of the MC designs as proposed by the MEDOS. This DES contains knowledge of the practical design ranges of many components used in MC, techniques for assessing the structural integrity of these components, and an extensive procedure for selecting suitable drives for the MC design from real-life motor and gear catalogues and other commercially available databases.

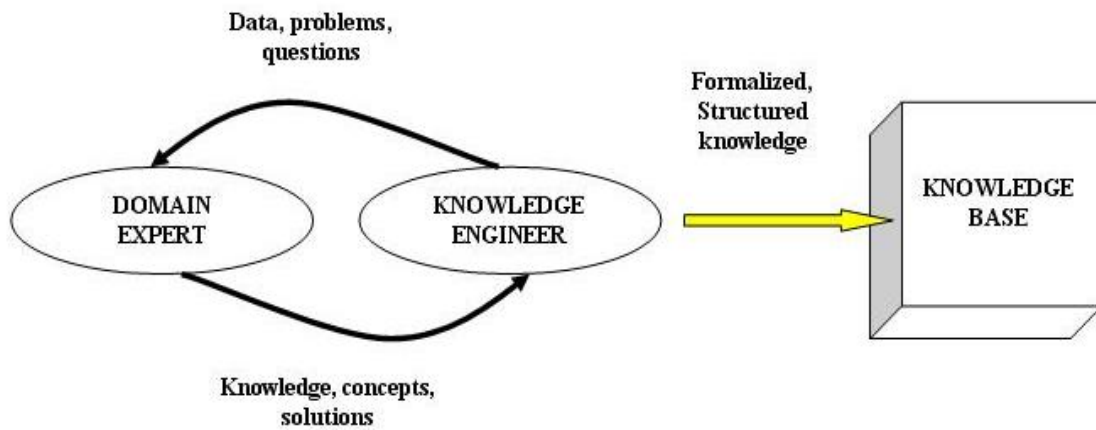
## **1.4 Expert Systems**

Due to the centrality of ES in this work, the basics of the field and a review of its widespread applications are presented in the following sections. For further information, a very useful reference on ES is the book by Giarratano and Riley [24].

### **1.4.1 Fundamentals**

*Introduction:* ES is a branch of AI that makes extensive use of specialized knowledge to solve problems at the level of a human expert [24, 25]. A human expert can solve problems that most people or a computer cannot, or can solve them much more efficiently. An ES mimics the capabilities of one or a team of human experts. It gains its power from the knowledge it contains, which can be valuable expertise acquired from domain experts or simply standardized knowledge obtained through sources such as books, manuals, or even knowledgeable people [26, 27]. The knowledge of an ES cannot possibly encompass all domains, and thus ES applications mostly focus on one or few problem domains, known as the knowledge domain of the ES.

*Knowledge Acquisition:* The process of acquiring domain-specific knowledge from human experts or other sources is known as knowledge acquisition (KA), shown between the domain expert and the knowledge engineer in figure 1.2. There are well established methods for efficient KA, such as interviews, observations, taking experts through case studies, and rule induction performed by machines [27, 28]. KA is seen as a crucial aspect concerning the success of an ES and has always been regarded as the bottleneck in its development [29].



**Figure 1.2: Knowledge acquisition and representation in ES development.**

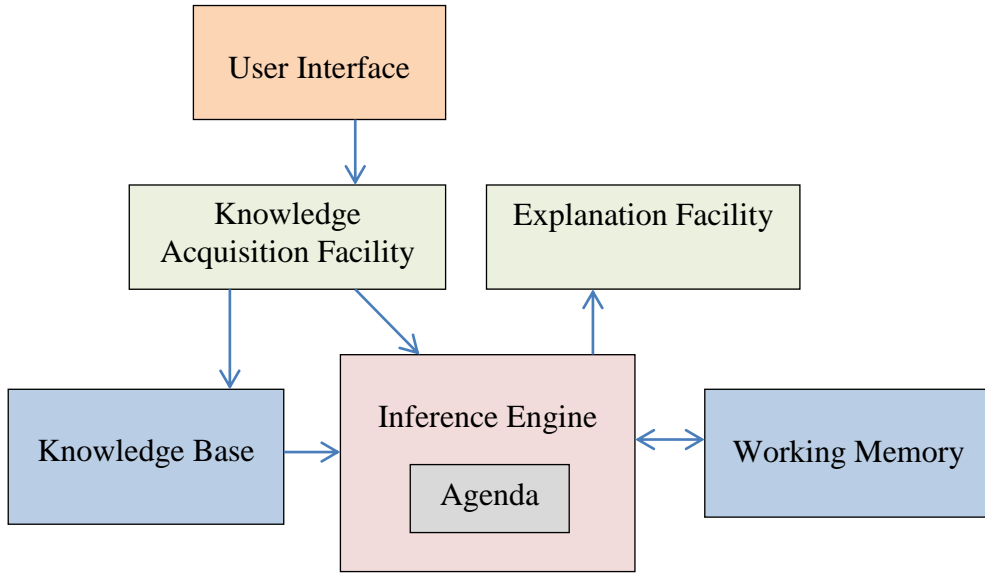
*Knowledge Engineering:* Of the three possible individuals (or groups) who interact with the ES (the end-user, knowledge engineer, and domain expert if knowledge is acquired from a human), the knowledge engineer is the most important one. The knowledge engineer determines the appropriate representation of the knowledge that is acquired from the experts and stored in the KB, defines the inference technique that is required to solve the problem, and is generally responsible for the implementation of all components of the ES in a coherent package. These tasks, which encompass the building, maintaining and development of ES, are known as knowledge engineering [27, 30].

*Knowledge Representation:* The process of encoding the information that is acquired through KA for use by the ES is called knowledge representation. The knowledge of an ES can be represented in a number of ways, including but not limited to: rules (if-then and/or production rules), object attributed value (OAV), semantic networks and frames, and logic statements [27]. The most common method of representing knowledge is in the form of if-then type rules. If a fact exists in the working memory of the ES that matches the “if” statement, the rule is satisfied (or, “fired”) and the action preceding the “then” statement is performed. Although if-then rules are simple when witnessed alone, many significant ES have been built using them as the only representation of knowledge [24]. If the knowledge representation uses fuzzy attributes, then the rules and their “firing mechanism” or “inference machine” will involve fuzzy logic [27]. Furthermore, conflicts can arise in the choice of a rule in decision making. Conflict resolution is also an important aspect of decision making in a knowledge-based system [27]. With thousands of if-then rules available in a KB, complicated problems can be solved. If the KB of an ES is mainly comprised of rules, such as the ES developed in the present work, it is called a rule-based system.

*Elements of ES:* The main elements of a characteristic ES are the user interface, explanation facility, working memory, inference engine, agenda, and KA facility [27, 31]. The user interface is the mechanism by which the user and the ES communicate. The user does not necessarily have to be a human (as in the present application), although this is typically the case. The ES can be built with a GUI that autonomously interacts with other computer systems, without the intervention of a human. The ES in this work is built to do so with a synchronized Excel<sup>®</sup> spreadsheet (sections 3.6.2 and 4.3).

The KA facility is not a necessary element of ES, but can be included as a beneficial extra feature. It provides an automatic way for the user to enter knowledge in the ES knowledge database instead of the knowledge engineer having to explicitly code it in. The ES presented in this work is equipped with such a facility. Databases of motor and gear specifications from catalogues (section 2.4) automatically interact with the ES during actuator selection. Modifications and updates in the motor and gear specifications need not be coded explicitly into the ES and can simply be changed in the database. The KB contains all the domain-specific rules resulting from the KA and the knowledge representation steps. Typically, this database does not change for different runs of the ES. However, sophisticated ES implementations may incorporate “self-learning” features that will enable various components (KB, database, etc.) of the ES to be improved or enhanced continuously. The rules of the KB for the ES developed in this work are derived from principles of the application domain, presented throughout chapters 2 and 3.

The working memory is a continuously changing database of the facts established by direct inputs specific to a certain run of the ES or as a result of previous rules which have been satisfied. These facts are the “if” statements used in future if-then rules of that run. When a rule is satisfied, the “then” statement becomes a fact in the working memory. During a run of the ES, the prioritized list of pertinent rules whose patterns are satisfied by facts in the working memory is the agenda of that run. Accordingly, the inference engine makes inferences or decisions by determining which rules are satisfied by facts in the working memory, prioritizing them, and executing the one with the highest priority. In the ES of this work, the software used for implementation (Visual Studio C++) handles the working memory, creates the agenda and makes the inferences for any specific run.



**Figure 1.3: Developed rule-based ES elements and structure of ES.**

The explanation facility displays the reasoning of the ES to the user through the standard compiler output window. Figure 1.3 shows the elements of a typical rule-based ES with the specific structure representing the ES developed in this thesis.

#### 1.4.2 Literature Survey of Applications

ES provide powerful and flexible means for obtaining solutions to a wide variety of problems. Their use proliferates through many sectors of society and technology, where their applications are proving to be critical in the process of decision support, problem solving, and design [32]. In a brief literature review on ES applications, one cannot possibly encompass the enormous number of fields, problems, or tasks for which ES have been developed. Instead, in the present section, a focused and limited number of ES in the engineering domain will be discussed. Furthermore, to provide a worthwhile background for the ES that is developed in this work, only the literature that can be associated with the DES duties discussed in section 1.3.3 is considered. In doing so, it is useful to divide

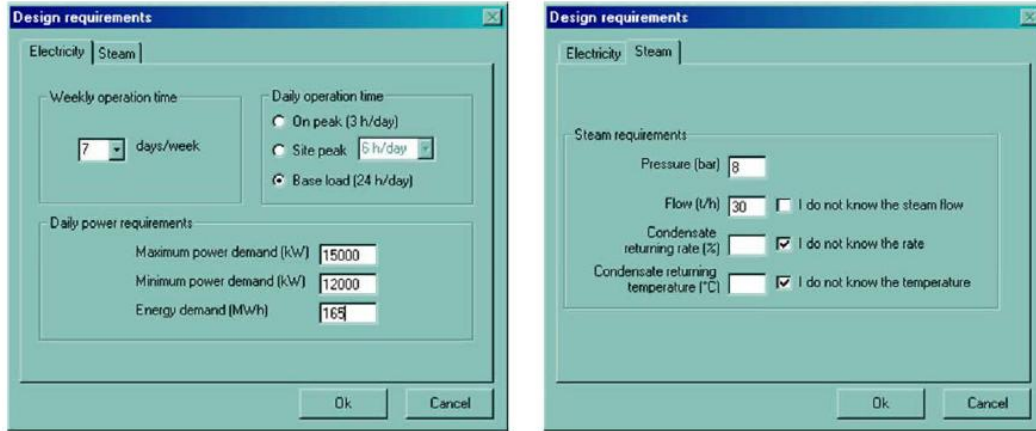
the literature on engineering ES that is discussed in this section into three broad categories: selection, detection and diagnosis, and design. For each of these categories, a handful of example applications from the literature will be listed below and several of them will be discussed in detail.

A selection ES uses its domain-specific knowledge for the selection of design specifics such as equipment, materials, manufacturing processes, geometry, or even individual parts of a proposed product. Examples of material selection ES include the work of Agrawal and Vasudevan [33], Sapuan [34], and Zarandi et al. [35]. Even within the focused category of material selection, the practical domains of these ES differ. For example, [33] only deals with the selection of plastics and [35] focuses only on the sustainability aspects of chosen materials. ES applications in equipment and processes in the engineering domain include the selection of suitable mining equipment by Ganguli and Bandopadhyay [36] and process selection for cast components by Er and Dias [37]. Examples of ES component selection include the selection of speed reduction components for power transmission by Ferguson et al. [38] and the selection of fastening tools for automobile assembly lines by Milani and Hamed [39]. The common drawback of the papers cited above is that they only accept input from a human user and do not perform any processing of the data before starting the selection routine. These drawbacks are not present in the ES presented in this thesis, where data can be received automatically from another computer system or program, and a numerical optimization routine is run on certain data *before* starting the selection process.

Detection or diagnosis ES use domain-specific knowledge for the detection of faults in specific machine parts such as roller bearings [40] or metal frames [41], or the

diagnosis of failures in machinery [42, 43], automobiles [44], or even central energy plants [45]. Detection or diagnosis ES are usually developed to run in real-time, providing continuous monitoring and decision making support. Thus, the drawback associated with the reliance on human input is overcome in these applications. However, there remains the drawback that the detection or diagnosis is only limited to faulty components or machinery, and the ES is unable to infer whether a design weakness is the root cause.

DES use design domain-specific knowledge to provide assistance and consultation in design tasks such as the design of electronic devices [46], machines or machine parts [47], or even full scale power plants such as a natural cogeneration plant [48]. In the literature, design ES can also be developed to contain guiding principles in specific engineering design domains such as guidelines for structural design [49], thermal system design [50], or general design-for-manufacture [51], design-for-machining [52], and design-for-reliability [53]. The work of Matelli et al. in 2009 [48] has set a benchmark for the level of requirements and features a good present-day DES should possess. In their work, an ES prototype to support engineers in the conceptual and preliminary design phases of natural gas cogeneration plants was presented. The KB consisted of rules, OAV, component technical information and procedural rules. Using the convenient GUI shown in figure 1.4, the design requirements are entered into the DES. Then, the complex process of choosing a suitable plant layout, sizing the associated components, and providing explanations for all the decisions is done by the DES. Although the DES presented in this thesis handles a different application domain, many of the same program features have been implemented. Modifications were made to the



**Figure 1.4: Design requirement input GUI for cogeneration plant DES.**

GUI to make it updatable from other systems in the EDF, and autonomous interaction with a pre-selection optimization routine of the design data was developed (chapter 4). In all the literature on DES as reviewed for this research, a fully autonomous DES that performs the duties outlined in section 1.3.3 was not found. This thesis presents a DES with preliminary features which can enable such a realization.

## 1.5 Research Objectives

The overall goal of the research presented in this thesis is to develop a design expert system that can guide the evolution of optimal MC designs and assess the feasibility of the final MC design proposals. The DES will be built to work within an evolutionary design framework consisting of a machine health monitoring system and a modeling and evolutionary design optimization system. Features will be built to ease the integration of the DES with these systems, such as a GUI which can interact with both a developed Matlab® optimization code and the DES itself. Verification of results and evaluation of the performance improvement offered by the integrated system compared to a standalone DES are also associated objectives of the present research.

Within the global objective of developing the DES, there are a number of intermediate objectives which are important. First, a comprehensive understanding of the important technical, practical, and procedural aspects of MC system design and actuator selection or sizing will be attained. This domain-specific design knowledge will be translated to a large set of appropriate if-then rules, in a heuristic and mathematical form. A systematic methodology for navigating through these rules will be developed and implemented in a commercial software development program. The developed DES will automatically obtain specifications of purchasable actuators from an external database through a KA facility, and will interact with a developed Matlab<sup>®</sup> optimization routine of pertinent MC design components to reduce their weights and costs without sacrificing functionality and structural safety. Subsequent chapters of this thesis will present the development of the intermediate KA, implementation, and performance evaluation steps.

## **1.6 Organization of the Thesis**

This thesis consists of five chapters. In the current chapter, the motivation for developing tools for automated optimization and design was discussed. The rationale for using ES in this regard and the duties associated with a DES in the context of the EDF was presented. An introduction to the basic concepts of ES was given and applications in literature relevant to the DES duties were reviewed in detail. In this introductory chapter, the objectives of the present research have been highlighted as well.

Chapter 2 first discusses the importance of the specific DES application domain, which is MC design. The choice of motor selection as the feasibility assessment of the proposed MC designs is explained. The systematic methodology developed for MC design analysis and subsequent motor-gear selection is outlined. The first stage of this

methodology—calculating the load—is explained in detail with all the various generic or custom designs which the DES can handle. Finally, the development of the motor and gear databases used in the implementation is highlighted.

The first part of chapter 3 explains the second stage of the methodology presented in chapter 2, namely, the duty cycle (DC) calculations. Different DC categories and the corresponding profile options (generic and custom load speed profiles and load position profiles) are described. Practical considerations featured in the developed DES are given sufficient attention in this chapter, with detailed discussions of s-curve profiling, rms torque, geometric trajectory blending, gearhead effects, and thermal response. The second part of this chapter outlines the implementation of all the prior knowledge acquired into the DES. Rule groups in the DES code provided in the Appendix are identified with line numbers, the GUI is presented, and verification results of the DES are highlighted.

Chapter 4 presents the design optimization add-on that is developed for the DES. This is preluded by an explanation of the constrained minimization and sequential quadratic programming techniques used in the optimization. A case study for the custom design of a hoist is presented, where components of the design are optimized under practical design constraints and the resulting design is analyzed for feasibility by DES through selecting suitable actuators. The effect of the design optimization add-on is quantified. This case study presents an example operation of a fully integrated DES in the EDF.

Chapter 5 concludes the work presented in the thesis by highlighting the contributions made and their significance. Also, directions for future research in the area are suggested.

## **Chapter 2**

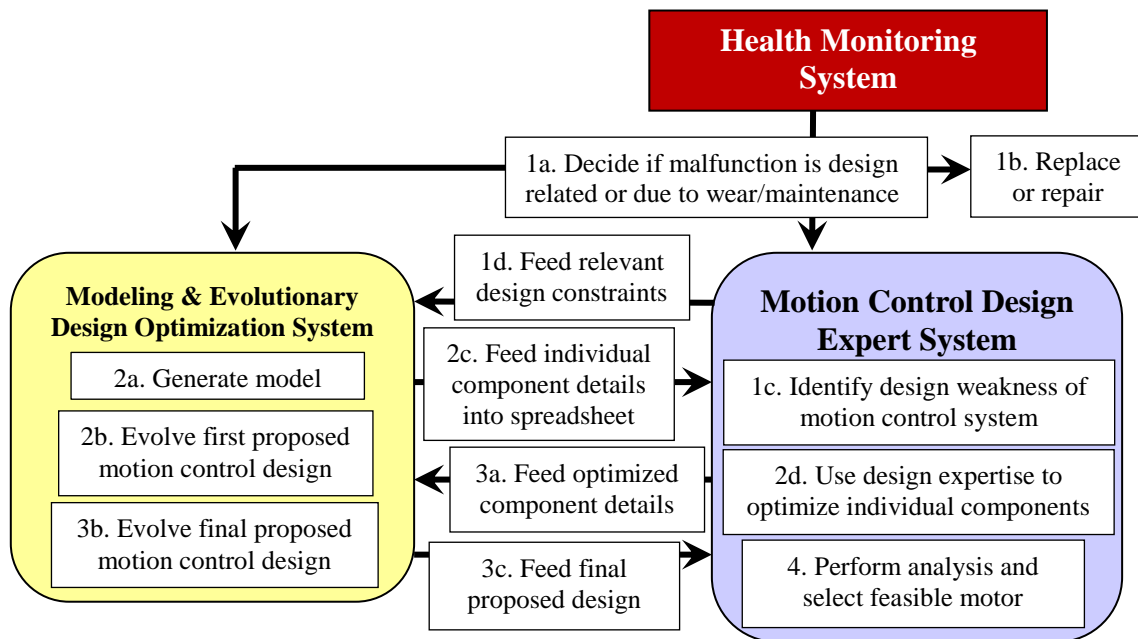
### **Application Domain**

#### **2.1 Motion Control Domain**

The use of advanced MC systems has become an indispensable part of modern-day industrial automation [22]. At the core of any MC system is an electric motor as the actuator which drives the application. In an industrial facility, electric motors are a familiar sight, driving almost everything from blowers and pumps to conveyors, machine tools, mixers, shredders, robots, and more. Due to the prevalence of electric motors in many industrial applications and the importance of proper motor selection (section 2.1.1) in these applications and generally for industrial automation, the domain of MC design and motor selection was chosen for the ES developed in this research. Hence, the ES developed in this research is termed Motion Control Design Expert System (MCDES).

In light of the evolutionary design framework presented in the previous chapter, and since the DES works in close association with the MEDOS in that framework, the MEDOS should evolve designs in the same MC domain as the DES in order for the DES to properly interfere in the process and assess the proposed designs. Currently, the design optimization division of the MEDOS only handles well-defined optimization problems usually comprising core mechanical or electrical elements such as springs, dampers, resistors, inductors, and so on. The ability to optimize a MC system design such as a conveyor belt drive for optimal cylinder sizes or belt materials, for example, is not

present. Thus, in this research, the MCDES is built to take design parameters from a human user and not the MEDOS. That being said, measures have been taken to allow for easier integration with a MEDOS which can supply optimized design parameters and structures for the MC domain, when such a system becomes available. For example, the MCDES code was synchronized with a Microsoft Excel<sup>®</sup> spreadsheet (section 3.6.2) where the MC design can be easily built or modified. When a MEDOS in the same domain as the MCDES becomes available, the outputs and inputs of the MEDOS can be smoothly integrated with the Excel<sup>®</sup> spreadsheet rather than the more complicated MCDES code. A form of this integration is presented in detail in chapter 4, where numerical optimization routines are used to automatically optimize the components of an MC design within structural and heuristic design constraints. Ultimately, with an MEDOS in the MC domain, all these components and features can be integrated into one harmonized system, as illustrated in figure 2.1. This system can go from fault identification to generating a feasible design alternative with no human intervention.



**Figure 2.1: Objective for a fully completed automatic design evolution routine.**

## **2.2 Motor Selection**

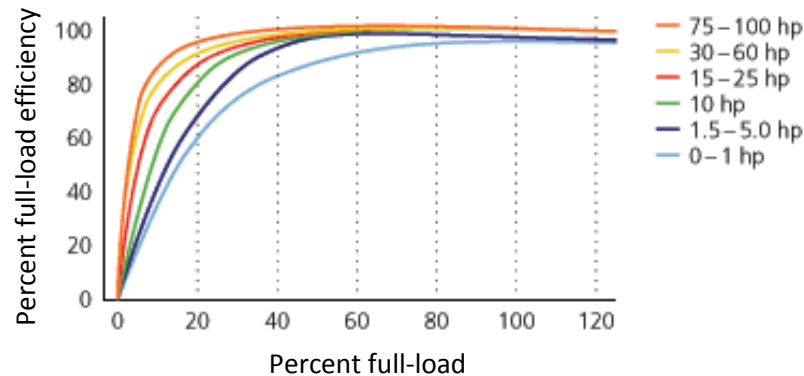
The remainder of this thesis will focus on the details of step 4 in figure 2.1, namely, the MC design analysis and drive selection. Work has also been done on step 2d and intermediate steps between the MCDES and MEDOS (1d, 2c, 3a, and 3c). These are presented in chapter 4. First, the importance of proper MC design analysis and drive selection is presented.

### **2.2.1 Importance**

MC is responsible for about two-thirds of the total energy consumption in a typical industrial facility [54]. The most influential cost factor in MC system design is the motor choice. Not only is the capital acquisition cost higher for a larger motor, its operational cost is also significantly higher [55]. Throughout the lifetime of a motor, the cost due to power consumption during operation is significantly larger than other costs such as service and maintenance, incurred downtime losses, and acquisition capital. In some cases, power consumption can account for up to 96% of the total lifecycle costs of a motor [55]. The cost effectiveness of a properly sized or energy-efficient motor is so large, the U.S. Department of Energy (DOE) predicts that replacements with such alternatives fully payback in less than 2 years [56].

Unfortunately, the importance of motor sizing is routinely underestimated. The U.S. DOE estimates that about 80% of all motors in the United States are oversized [54]. This is mostly due to the designer short-routing the complete selection process. In a proper motor selection process, the system load must be carefully calculated and all the necessary system components must be included. Friction during the interaction and wear

of the components over time is usually accounted for using a design safety factor. Since a motor has its own rotor inertia, for a given load, the overall system torque is different for each individual motor. The calculation of the system torque must be repeated for every motor that is available in the inventory if any one of them is to be chosen, which is a very time-consuming task. Thus, most designers simply select an oversized motor to account for the uncertainties in the load requirements and availability, and to allow for load increases during operation such as those which result from wear. This higher load capacity of the larger motor is often offset by the fact that the efficiency of the motor decreases when it operates at fractions of the full load it is designed for. This behaviour is evident in figure 2.2 [56]. Also, a danger is presented by an oversized motor as it can operate the load at a faster speed and acceleration than what is allowed, with a high torque.



**Figure 2.2: Variation of motor efficiency with loading.**

Automating the procedure of MC design and drive selection can greatly reduce mistakes in analysis and subsequent motor sizing. This will not be at the expense of time, as that will be greatly minimized as well. Such an automated procedure can also greatly increase the choices of selectable drives, since a large number of drives can be included

in the database of the selection program. ES are a viable option for this application, but require time and effort in gathering and formalizing the pertinent expert knowledge for MC design and drive selection. The methodology and pertinent knowledge related to MC design and drive selection which is built into the developed MCDES is presented now.

### 2.2.2 Methodology

Mechanically, motor sizing or “load matched” selection is based on the total polar moment of inertia of the mechanical structure (MS) as reflected at the motor rotor, along with the speeds and accelerations of the MS that are required by the application. For a motor to be suitable for the MS in a specific application, it must be able to comfortably drive the load at the required speeds within the times mandated by the speed or position profiles of the application. This is done by supplying sufficient torque and angular speed to the load as discussed in section 3.5.2. Before finding the required torque and angular velocity however, knowledge of the motion objectives, mechanical components, and the duty cycle (DC) of the application must be established. Overall, the selection process can be divided into the following steps [55]:

1. Establishment of motion objectives
2. Selection of mechanical components
3. Definition of DC
4. Load Calculation
5. Selection of suitable drive

In the first step, a general idea of the overall objectives of the MC system must be established. These overall objectives include whether the application is linear or rotary, horizontal or vertical (if linear), the positioning accuracy, and thermal parameters such as ambient temperature. In the MCDES, the answers to such questions are determined either through direct user input of the required parameter(s) or through inferences made by the

program based on previous inputs.

In the second step, the mechanical components of the MS must be selected and oriented. In the MCDES, the user is given a wide variety of generic applications (GA). The MCDES can determine the required motor torque and speed for these applications once the geometric and inertial/mass parameters are determined and the required DC is known. In addition to GA, the MCDES has a custom MS feature, where a large scale mechanical application can be created by choosing from a wide range of individual components such as shafts, belt pulley drives, chain sprocket drives, lead screws and nuts, roller bearings, gearboxes, and so on. A specialized feature where multiple applications can be driven by one motor is also built in the capabilities of the MCDES. This is made possible with a multi-output gearbox being included as a component in the custom MS design feature.

In the third step of the selection process, the DC of the application must be defined. The processing of the DC by the MCDES results in the speeds and accelerations which the selected motor must be able to supply, as will be explained in chapter 3. The MCDES has two separate DC tracks: load speed profiles (LSP) and load position profiles (LPP). The choice between these two depends on whether the critical parameter of the DC is a speed or a distance. The latter case is mainly intended for servo applications. Note that the MCDES contains both generic and custom profile options for LSP and LPP. In addition, under each of the DC available in the MCDES there are corresponding advanced features such as s-curve profiling and geometric trajectory blending. The flow chart in figure 2.3 graphically represents the MS and DC characterization procedure of the MCDES described above.

The final steps of load calculation and drive selection become straightforward once the MS and DC calculations are performed. In the load calculation step, the required motor torques and speeds are found immediately once the MS and DC are defined.

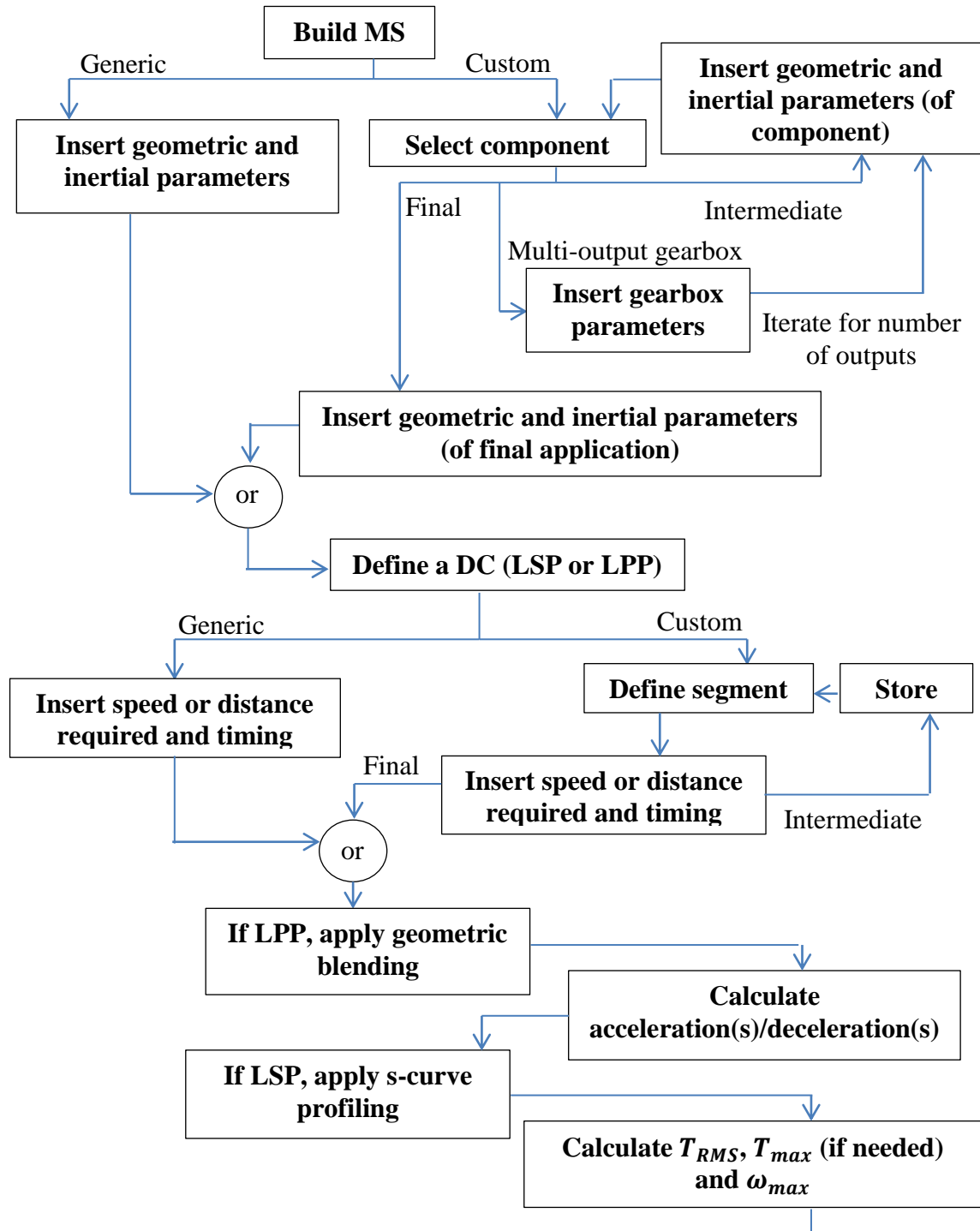
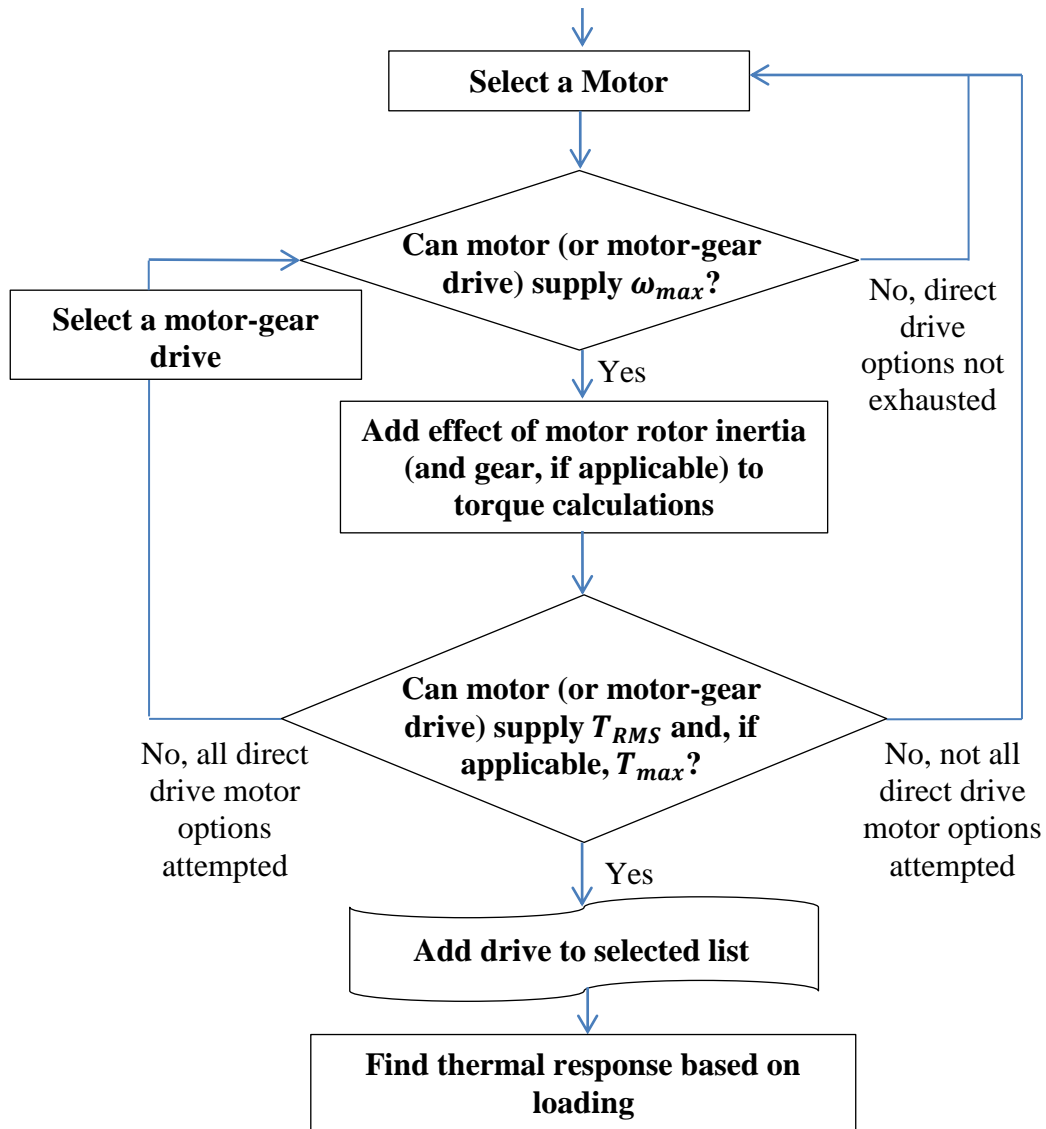


Figure 2.3: MS and DC characterization procedure in MCDES.

Consequently, in the final drive selection step, a comparison is performed between the required torques, speed, and temperature rise and the rated and peak torques, speed, and maximum operating temperature, respectively. Figure 2.4 is the continuation of the flow chart of figure 2.3 and represents the selection procedure of the MCDES.



**Figure 2.4: Selection procedure of the MCDES.**

## 2.3 Inertia of Mechanical Structure

### 2.3.1 Generic Applications

A detailed list of GA has been surveyed [55, 57] and the expression for the polar moment of inertia reflected to the motor has been found for each. Each expression is a function of geometric and inertial/mass parameters related to the application as well as the polar moment of inertia of the motor rotor. Thus, any GA can have  $i$  number of equivalent inertia expressions, where  $i$  is the size of the motor database. Table A.1 in Appendix A is a complete list of the GA supported by the MCDES and the corresponding polar moment of inertia expression for each. Equation 2.1 is an example of the equivalent polar moment of inertia of a belt conveyor GA. Note that these expressions are for the direct-drive (DD) case only. The addition of a gearhead affects the inertia of the GA and the DC parameters. The effect on the inertia is outlined in section 2.3.4 and the effect on the DC is presented in chapter 3. Then, any GA can have  $i \cdot j$  number of equivalent inertia expressions given by equation (2.1), where  $i$  is the size of the motor database and  $j$  is the size of the gear database.

$$J_{eq_i} = J_{m_i} + J_d + J_s + r^2(nW + m_c) \quad (2.1)$$

Here:

$J_m$ : motor rotor inertia [ $\text{kgm}^2$ ]

$J_d$ : driving cylinder inertia ( $J_d = \frac{1}{8}m_d D_d^2$ ) [ $\text{kgm}^2$ ]

$m_d$ : driving cylinder weight [kg]

$D_d$ : driving cylinder diameter [m]

$J_s$ : driven cylinder inertia ( $J_s = \frac{1}{8}m_s D_s^2$ ) [ $\text{kgm}^2$ ]

$m_s$ : driven cylinder weight [kg]

$D_s$ : driven cylinder diameter [m]

Note:  $D_s = D_d = D$

$r$ : driving cylinder radius [m]

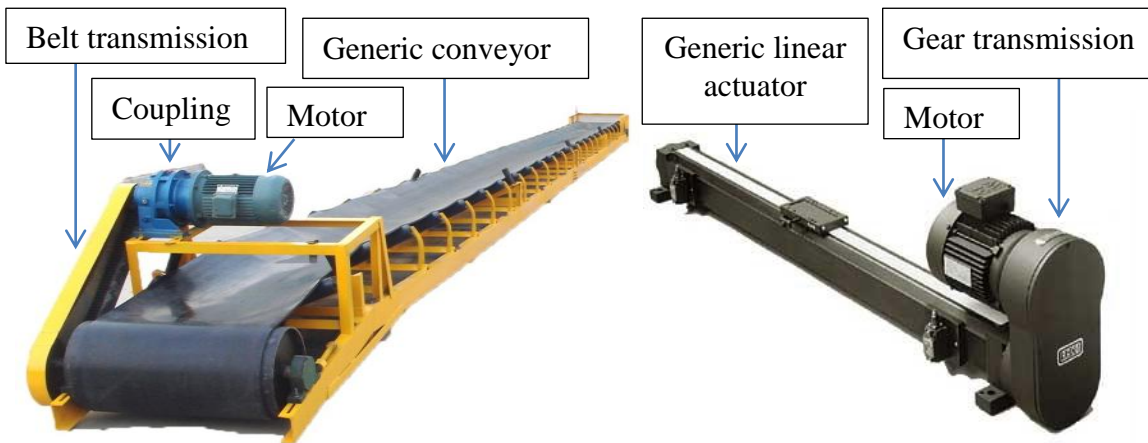
$n$ : number of weights on conveyor

$W$ : mass of load [kg]

$m_c$ : mass of conveyor belt (if considerable)[kg]

### 2.3.2 Custom Applications

The inertia expressions given in table A.1 are applicable only for single GA. In the custom MS building feature of the MCDES, the equivalent inertia of the entire custom system depends on the individual mechanical components used in the structure together with the final GA which directly interacts with the load. In order to compute the correct inertia for the custom MS, the inertia of each mechanical component must be known. Again, they are mainly a function of geometric and mass properties, as listed in table A.2 of Appendix A. Some custom components like special transmissions also affect the speed and acceleration requirements of the motor, which is taken into consideration by the MCDES.



**Figure 2.5: Examples of real designs that can be built by the MCDES custom design feature.**

Using the custom MS building feature, the user is given more flexibility and innovative capacity for their design. Although not exhaustive, the components available for custom MS design are quite sufficient to create medium complexity designs. Two real world examples of designs which can be built in the MCDES using the custom design feature are presented in figure 2.5. The conveyor example to the left can be built using a few shafts, bearings, a motor coupling, a belt transmission, and a final conveyor GA. Similarly, the linear actuator example to the right can be built using shafts, bearings, gears, and a final linear actuator GA.

The custom design of a hoist and the optimization of some of its individual components are presented in chapter 4. The effect of the optimization of the torque requirement is examined as well. The MCDES can also handle designs where a single motor drives multiple different GA. This is done by choosing a multi-output gearbox from the custom component list. Two or three outputs from a single gearbox can be handled. Each output is built separately with custom components and a final GA. After building each line, the DC for the final GA of each must be defined. The MCDES then processes the required inertia, torques, and accelerations for each line and computes the required total power for the entire system. This is used to select a suitable motor. The motor must be able to supply the maximum speed of the multiple GA, adjusted for any transmissions or gearboxes in the design.

### **2.3.3 Custom Components**

Table A.2 in Appendix A gives a list of the components available in the MCDES for building a custom MS and the corresponding polar moment of inertia expressions,

reflected to the motor rotor. Equation 2.2 is an example of the equivalent polar moment of inertia of a chain-sprocket transmission.

$$J = \frac{m_{SM}r_{SM}^2}{2} + \left(\frac{1}{N_r}\right)^2 \frac{m_{SL}r_{SL}^2}{2e} + \frac{W_C}{ge}(r_{SM})^2 + \left(\frac{1}{N_r}\right)^2 \frac{J_L}{e} \quad (2.2)$$

where:

$J_L$ : load inertia [kgm<sup>2</sup>]

$e$ : mechanism efficiency

$N_r$ : transmission ratio  $\left(\frac{r_{SL}}{r_{SM}}\right)$

$r_{SM}$ : motor – side sprocket radius [m]

$r_{SL}$ : load – side sprocket radius [m]

$m_{SM}$ : motor – side sprocket mass [kg]

$m_{SL}$ : load – side sprocket mass [kg]

$W_C$ : chain weight [kg]

### 2.3.4 Effect of Gearheads on Inertia

When all DD options of the MCDES motor database are exhausted, the selection procedure of figure 2.4 switches to motor-gear drive possibilities. Since all the gearheads in the MCDES gear database are reducing type, adding a gearhead to the drive means the motor must supply an angular speed  $p$  times higher than the DD case, where  $p$  is the reduction ratio of the gearhead. Alternatively, the angular speed requirement of the motor can be kept unchanged, but the application will not reach the required speed of the DD case, and the equivalent inertia of the application will be reduced. This reduction affects all terms other than the motor's rotor inertia and is squared since those inertia terms are eventually multiplied by the angular velocity squared, in the expression of kinetic energy. In both cases, the inertia of the gearhead itself must be added to the total inertia of the

system. For a given application, the number of equivalent inertia expressions for a motor-gear drive is  $i \cdot j$ , where  $i$  is the size of the motor database and  $j$  is the size of the compatible gearhead database. Table 2.1 presents the effect an added gearhead has on the equivalent inertia expressions for GA. In the new expressions,  $J_g$  is the gearhead inertia [ $\text{kgm}^2$ ] and  $p$  is the reduction ratio. The remaining parameters are the same as in table A.1. Note that these expressions are applicable for the case where the angular speed requirement of the motor is kept constant.

**Table 2.1: Inertia of GA in the motor-gear drive scenario.**

Generic application	Equivalent polar moment of inertia $J_{eq}$ [ $\text{kgm}^2$ ]
Hoister	$J_{eq_{ij}} = J_{m_i} + J_{g_j} + \frac{J_d}{p_j^2} + \frac{r^2}{p_j^2} (W + m_R)$
Flywheel	$J_{eq_{ij}} = J_{m_i} + J_{g_j} + \frac{J_d}{p_j^2}$
Belt Conveyor	$J_{eq_{ij}} = J_{m_i} + J_{g_j} + \frac{1}{p_j^2} (J_d + J_s) + \frac{r^2}{p_j^2} (nW + m_c)$
Horizontal puller	$J_{eq_{ij}} = J_{m_i} + J_{g_j} + \frac{J_d}{p_j^2} + \frac{r^2}{p_j^2} W(\mu + 1)$
Ball screw drive/ Linear actuator	$J_{eq_{ij}} = J_{m_i} + J_{g_j} + \frac{1}{p_j^2} \left( J_A + \frac{(W_L + W_T)}{g * e} \left( \frac{1}{2\pi * P} \right)^2 \right)$
Nip roll	$J_{eq_{ij}} = J_{m_i} + J_{g_j} + \frac{1}{p_j^2} \left( \frac{m_m r_m^2}{2} + \left( \frac{1}{N_r} \right)^2 \frac{m_l r_l^2}{2e} \right)$

Generic application	Equivalent polar moment of inertia $J_{eq}$ [kgm <sup>2</sup> ]
Rack and pinion	$J_{eq_{ij}} = J_{m_i} + J_{g_j} + \frac{1}{p_j^2} \left( \frac{m_p r_p^2}{2} + \frac{(W_L + W_R)}{g * e} (r_p)^2 \right)$
Rotary table	$J_{eq_{ij}} = J_{m_i} + J_{g_j} + \frac{1}{p_j^2} \left( \left( \frac{1}{N_r} \right)^2 \frac{(m_d + m_l) r_d^2}{2e} + J_G \right)$
Center Driven Winder/ Surface Driven Winder	<p>Center driven:</p> $J_{eq_{ij}} = J_{m_i} + J_{g_j} + \frac{1}{p_j^2} \left( \frac{m_w r_w^2}{2} + 2 \left( \frac{1}{N_r} \right)^2 \frac{(m_s r_s^2)}{e} \right)$ <p>Surface driven:</p> $J_{eq_{ij}} = J_{m_i} + J_{g_j} + \frac{1}{p_j^2} \left( \frac{m_s r_s^2}{2} + \frac{m_s r_s^2}{e} + \left( \frac{1}{N_r} \right)^2 \frac{m_w r_w^2}{e} \right)$
Propeller/Fan	$J_{eq_{ij}} = J_{m_i} + J_{g_j} + \frac{1}{p_j^2} (J_P + J_E)$

## 2.4 Motor and Gear Databases

The values for  $J_m$ ,  $J_g$ , and  $p$  in the inertia expressions as given in the previous section can have a substantial effect on the motor or motor-gear drive unit that is selected. In order for the MCDES to provide practical MC design guidance and assessment, these values must come from real motors and gears that are commercially available. In the MCDES, the motor and gear database is built from over 50 different motors and compatible gears from Maxon Motor Corp. The  $J_m$ ,  $J_g$ , and  $p$  values for each motor and gear are included in this database, in addition to many other relevant parameters such as motor torque capabilities ( $T_{motor\_nominal}$  and  $T_{motor\_max}$ ), speed ( $\omega_{max}$ ), and maximum efficiency

( $\eta$ ). The relevance of these parameters and how they are used in the selection procedure are explained in section 3.5.2.

As time passes, with the availability of new products, the specifications of the motors and gears available in the database may be modified by the manufacturer. Also, the database size may need to be increased for a larger selection pool. If the database was embedded in the MCDES code, performing any modifications or additions would be a time-consuming task which could even risk corrupting part of the code's function. For that reason, the MCDES is synchronized with external Excel<sup>®</sup> spreadsheets containing the motor and gear databases. As will be highlighted in section 3.6.2, the MCDES also reads the MS design and relevant geometric and inertial parameters from a specialized Excel<sup>®</sup> spreadsheet. Figure 2.6 below shows sample Maxon<sup>®</sup> catalogue sheets from which the motor and gear databases for the MCDES are built. The corresponding specifications for several motors and gears used in the MCDES are also shown in tables 2.2 and 2.3. The full motor and gear database used in the MCDES to obtain the results presented in chapters 3 and 4 can be found in Appendix B.

		353294	353295	353296	353297	353298	353299	353300	353301
<b>Motor Data (provisional)</b>									
Values at nominal voltage									
1 Nominal voltage	V	18.0	24.0	36.0	48.0	60.0	70.0	70.0	70.0
2 No load speed	rpm	3350	3880	3770	3480	3490	3260	3020	2550
3 No load current	mA	714	647	407	271	217	170	153	123
4 Nominal speed	rpm	3100	3640	3550	3270	3290	3060	2820	2350
5 Nominal torque (max. continuous torque)	mNm	452	510	654	746	782	819	830	886
6 Nominal current (max. continuous current)	A	10.0	9.62	7.74	6.02	5.04	4.21	3.94	3.54
7 Stall torque	mNm	14700	16800	18600	17100	17200	16000	14500	12900
8 Starting current	A	302	297	208	132	106	78.8	66.3	49.9
9 Max. efficiency	%	81.6	84.3	87.3	88.5	89.1	89.3	89.1	88.9
<b>Characteristics</b>									
10 Terminal resistance	$\Omega$	0.0596	0.0809	0.173	0.363	0.566	0.888	1.06	1.4
11 Terminal inductance	mH	0.252	0.343	0.848	1.79	2.8	4.38	5.11	7.18
12 Torque constant	mNm / A	46.6	56.7	89.1	130	162	203	219	259
13 Speed constant	rpm / V	196	168	107	73.7	58.9	47.2	43.7	36.8
14 Speed / torque gradient	rpm / mNm	0.241	0.24	0.208	0.206	0.206	0.207	0.211	0.199
15 Mechanical time constant	ms	3.49	3.25	3.0	2.9	2.85	2.83	2.83	2.8
16 Rotor inertia	gcm <sup>2</sup>	1380	1290	1380	1340	1320	1310	1280	1340

Special program (on request)							
		110408	110409	110410	110411	110412	110413
<b>Gearhead Data</b>							
1 Reduction		3.7 : 1	14 : 1	25 : 1	51 : 1	93 : 1	308 : 1
2 Reduction absolute		63/17	3969/289	1701/68	250047/4913	107163/1156	19689/64
3 Max. motor shaft diameter	mm	14	14	14	14	14	14
4 Number of stages		1	2	2	3	3	3
5 Max. continuous torque	Nm	20	60	60	120	120	120
6 Intermittently permissible torque at gear output	Nm	30	90	90	180	180	180
7 Max. efficiency	%	80	75	75	70	70	70
8 Weight	g	2300	3000	3000	3700	3700	3700
9 Average backlash no load	"	1.0	1.5	1.5	2.0	2.0	2.0
10 Mass inertia	gcm <sup>2</sup>	165	155	125	88	154	89
11 Gearhead length L1*	mm	84.0	105.7	105.7	127.3	127.3	127.3

\* for EC 60 is L1 + 8 mm

Figure 2.6: Sample catalogue sheets for motors and gears used in MCDES.

**Table 2.2: Sample specifications for motors used in MCDES.**

<b>Motor #</b>	$J_m$	$\omega_{nom}$	$T_{nom}$	$\eta$	$R_T$	$K_T$	$R_{THa}$	$T_{a_{max}}$	$R_{THw}$	$T_{w_{max}}$
<b>353295</b>	1.29E-04	3640	0.51	0.843	0.0809	0.0567	1.3	100	1.85	125
<b>353296</b>	1.38E-04	3550	0.654	0.873	0.173	0.0891	1.3	100	1.85	125
<b>353297</b>	1.34E-04	3270	0.746	0.885	0.363	0.13	1.3	100	1.85	125
<b>353300</b>	1.28E-04	2820	0.83	0.891	1.06	0.219	1.3	100	1.85	125

**Table 2.3: Sample specifications for gears used in MCDES.**

<b>Gear #</b>	$J_g$	$p$	<b>Gear group</b>	$\eta$
<b>110408</b>	1.65E-05	3.7	6	0.8
<b>110409</b>	1.55E-05	14	6	0.75
<b>110410</b>	1.25E-05	25	6	0.75

## Chapter 3

### System Implementation

#### 3.1 Introduction

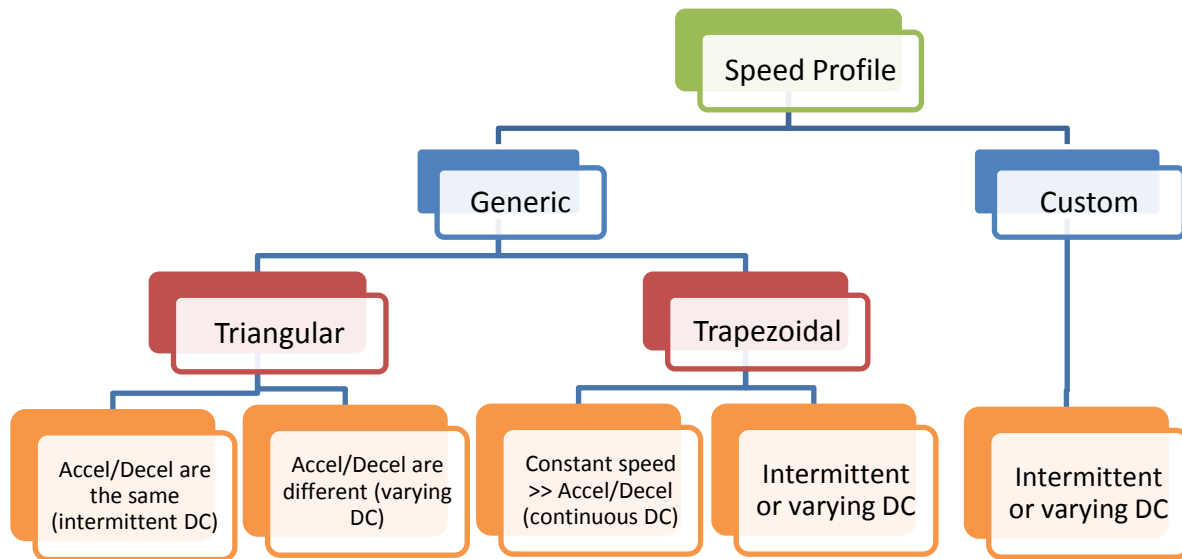
The load which the motor moves is defined as the torque that is required to drive the MS. A discussion of the different generic and custom MS built into our MCDES was given in the previous chapter. However, the motor torque required for any application does not only depend on the inertia reflected to the motor shaft by the MS, but also on the acceleration profile which the shaft must achieve. Even with a constant inertia reflected to the motor (constant weight and geometric parameters), there can be load variations that the motor must be able to handle. These load variations with respect to time, for any given application, describe what is known as the duty cycle (DC) of the motor. DC can be classified into three general categories:

- **Continuous:** DC which demands operation at a virtually constant load for a very long time. The constant load time is substantially longer than the ramp up (acceleration) or ramp down (deceleration time). This is the most common DC and accounts for nearly 90% of motor applications.
- **Intermittent:** DC which demands operation for different intervals of load, no load, and rest. The time durations of each of these intervals are well specified.
- **Varying:** DC which demands operation for different intervals of load, no load, and rest for intervals subject to large variation.

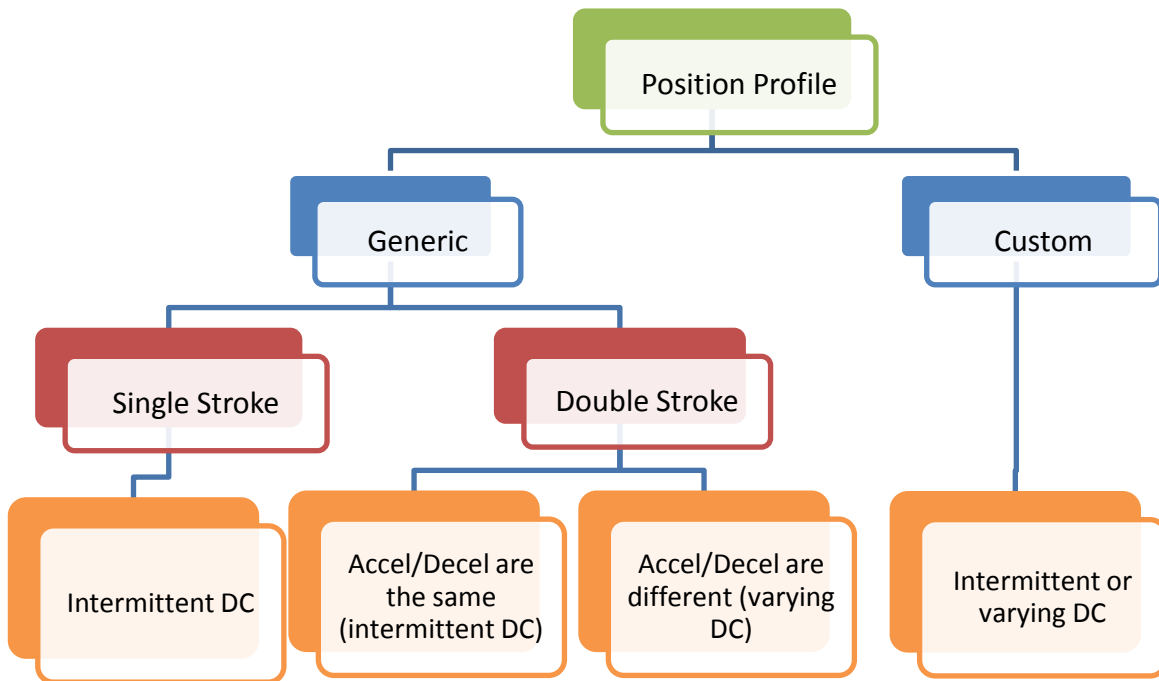
To truly rival the selection process of a human expert, the MCDES must contain the knowledge of the effect that DC have on motor torque requirements and must be able to perform a motor/drive selection with any DC. It must also be able to identify and include practical considerations such as s-curve profiling and thermal effects in the selection routine or as an advisory suggestion to the end user. The first part of this chapter presents the various DC options built into the MCDES along with the necessary computations and practical considerations for each. The second part of this chapter outlines the implementation of the MCDES in a user-friendly program, and verification.

## **3.2 Motion Profile Options**

In our MCDES, the type of DC used in the torque calculations and the motor selection routine is determined based on the choice of motion profile (MP) by the user. Simple profiles which would be used in applications like blowers and conveyor drives do not require extensive processing of the DC. This is because they require only a constant or gradually changing torque over a very long period of time, meaning they are continuous DC. Details of motion profile processing (MPP) for different DC are presented later in this section. For non-continuous DC however, the user is given different MP options to choose from, as outlined in figure 3.1 and 3.2. The MPP of each of these profiles will result in an accurate determination of the rms torque requirement, which is compared to the maximum continuous (nominal) torque of the candidate motors. The candidate motors must be able to supply such torques at the maximum speed of the MS chosen by the user. This required speed is extracted from the MP and is different for each structure, as will be explained later in the chapter. Finally, the peak intermittent torque, which is more easily calculated, is compared to the peak torque specification of the candidate motors.



**Figure 3.1: Motion speed profile options in MCDES.**

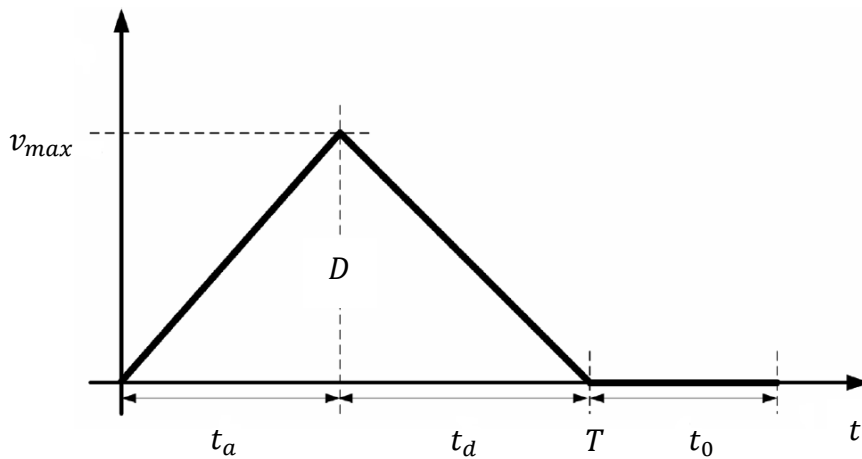


**Figure 3.2: Motion position profile options in MCDES.**

### 3.3 Load Speed Profiles

#### 3.3.1 Generic and Custom Speed Profiles

The user is given the option to have the load follow a generic load speed profile (LSP) or build a custom one on their own. In the generic category, there are triangular and trapezoidal profiles. The triangular LSP is when the motor accelerates to a maximum velocity and then immediately decelerates to zero. The motor may not remain at constant velocity in between the acceleration and deceleration portions.

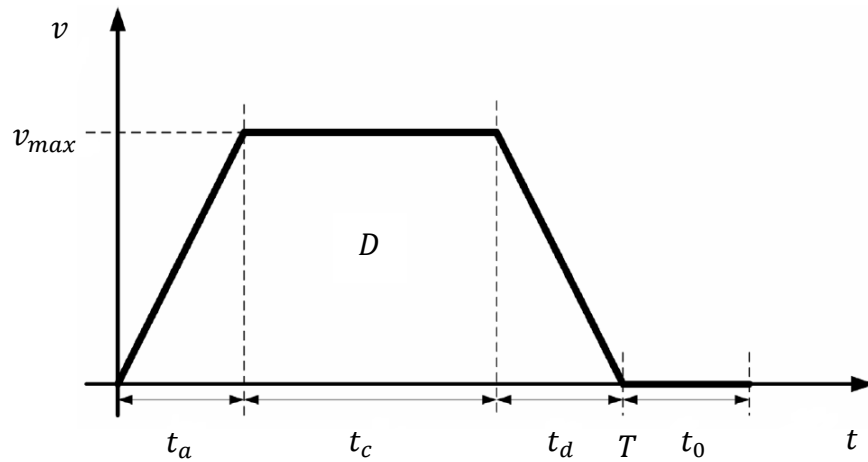


**Figure 3.3: Triangular load speed profile.**

If the acceleration and deceleration times,  $t_a$  and  $t_d$ , are equal, the step distance  $D$  of the load and the total motion time  $T$  are identified and used to find the maximum load velocity  $v_{max}$ . This is used to find the load acceleration and deceleration values,  $a$  and  $d$ . Alternatively,  $v_{max}$  can be identified along with the minimum time  $t_a$  that is needed to accelerate to it, and then  $a$  and  $d$  are found easily, and are equal. If the acceleration and deceleration times are different,  $v_{max}$  and the durations  $t_a$  and  $t_d$  must be specified explicitly, and cannot be solved for using a step distance  $D$ . When specified,  $a$  and  $d$  are found and are unequal. It should be noted that  $a$  and  $d$  are constant values at this stage

only, but become varying when an s-curve profile is added (section 1.3.2). Also, note that the variables with distance elements such as  $v_{max}$ ,  $a$ ,  $d$ , and of course,  $D$ , represent the MS movement, which is most usually rectilinear (e.g., load on a conveyor, table on a ball screw drive). However, the acceleration, deceleration and velocity values that are used to select the motor are rotational. Therefore, when the  $v_{max}$ ,  $a$ , and  $d$  values for any LSP are found after MPP, they must be converted to their rotational versions based on the choice of the MS being analyzed and its geometric dimensions (section 3.5.1).

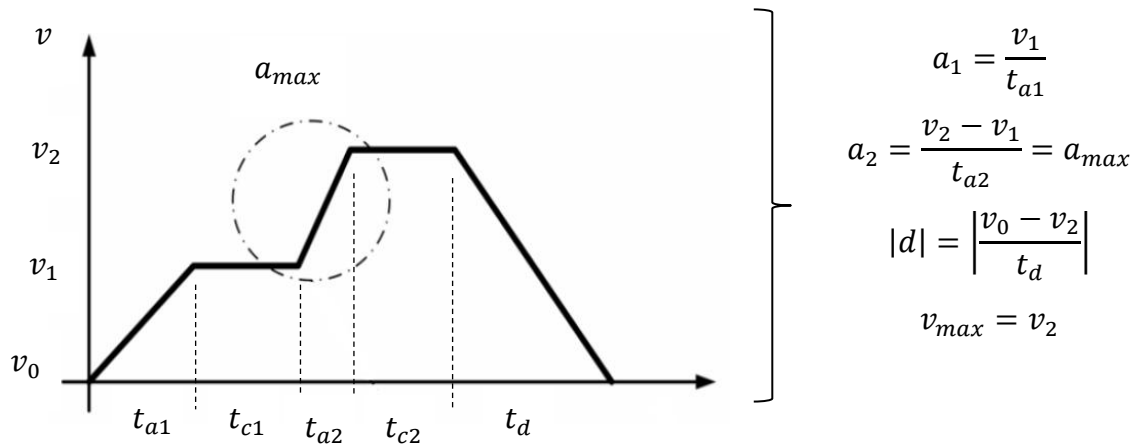
The trapezoidal LSP is present when the motor accelerates to a maximum velocity and then stays constant at that velocity for a prescribed time  $t_c$  before decelerating. Here, the MPP is identical to the triangular LSP except that the constant speed portion adds a torque component to the overall required torque which the motor must supply. This happens due to the need to overcome frictional and gravitational (if applicable) forces to stay at a constant speed, even though the acceleration is nonzero. Thus, in the MPP, the  $v_{max}$ ,  $a$ ,  $d$  values and their associated times  $t_c$ ,  $t_a$ , and  $t_d$ , respectively, should be found. Note that if the constant speed time  $t_c$  is substantially larger than  $t_a$  or  $t_d$ , the constant torque value that is calculated can be considered alone as the required torque which the



**Figure 3.4: Trapezoidal load speed profile.**

motor should supply. This is what was defined earlier as a continuous DC. However, to stay as accurate as possible, all the acceleration, velocity and time components are used in the rms torque calculation. It should also be noted that if there is a nonzero value of  $t_0$  in either the triangular or the trapezoidal LSP, there may be an additional torque component which needs to be included in the overall motor torque. In some MS, gravitational forces can act as a load even without any movement (e.g., hoist). If no brake is included in the design, the motor must supply the holding torque for this load. This is accounted for in the rms torque calculation.

Aside from the generic profiles, the user may want the load to follow a more complicated LSP. This is especially important in servo applications. In our MCDES, this need is catered to by allowing the users to build their own custom profile, such as the one in figure 3.5. Here, the MPP must identify acceleration, deceleration, and constant speed segments and calculate the necessary parameters ( $a$ ,  $d$ , or  $v$ ) for each. The time durations for each of these segments are identified and recorded for use in the rms torque calculation. In addition, the maximum acceleration, deceleration and speed values are found to compute the peak torque and speed requirements of the motor.

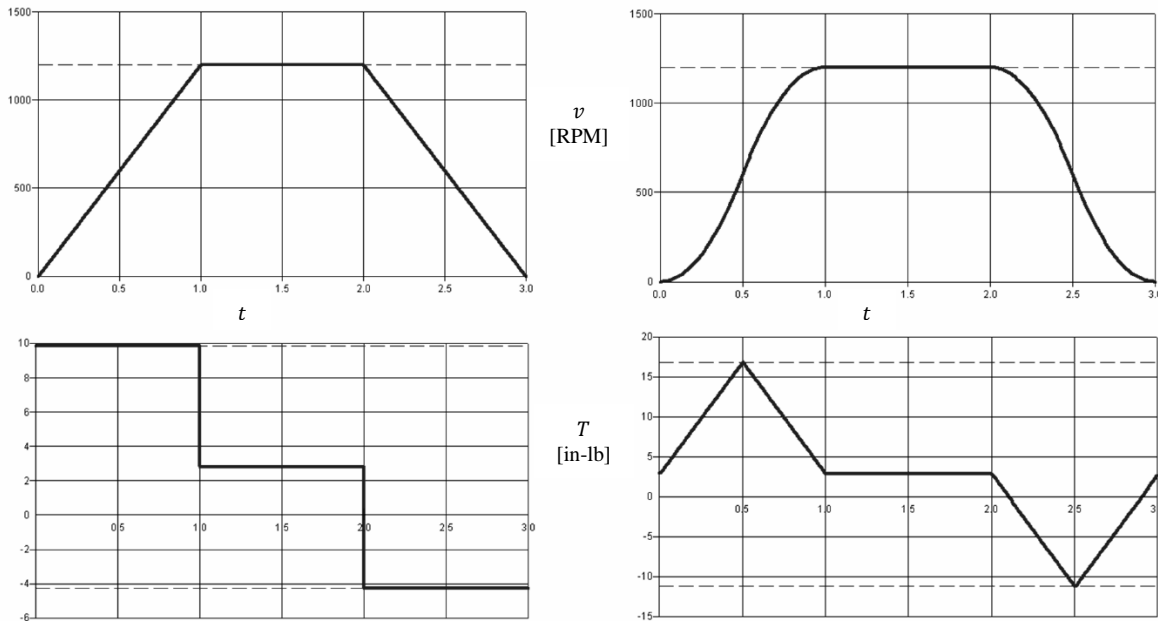


**Figure 3.5: Custom speed profile example and relevant parameters.**

### 3.3.2 S-Curve Profiling

S-curve profiling (SCP) is the process of adjusting the regions of speed changes in an LSP to minimize the jerk (rate of change of acceleration or deceleration). The result is a much smoother profile with a more preferable load response. This improved response can eliminate practical problems like fluid spilling during open container movement. Also, employing SCP will reduce wear on the mechanical components, thus, increasing their life. The amount of SCP done on a profile is specified as a percentage of each deceleration and acceleration portion. The percentage is equally divided between the start and the completion. This means that any SCP specification below 100% will result in some constant acceleration or deceleration segment in the middle of the profiled portion. A zero value represents no profiling, or step acceleration/deceleration. Figure 3.6 is an example of a trapezoidal LSP with and without 100% SCP (left and right, respectively) [55]. The change in the required torque (due to change in acceleration) is evident.

It is seen in figure 3.6 that the peak torques for the SCP case are significantly



**Figure 3.6: 100% S-curve profiling of a trapezoidal LSP and its effect on torque.**

higher than for the non-profiled case. If  $v_{max}$  and the times  $t_a$  and  $t_d$  are to stay the same after profiling, this increase has to occur in order to account for the gradual acceleration change of the profiled case versus the abrupt acceleration change of the non-profiled case. That being said, the non-profiled case will require an enormous jerk to reach  $a_{max}$  almost immediately, which is often not practically feasible.

In our MCDES, the user chooses or builds the LSP and specifies the necessary velocity and time parameters, and is then asked to specify the percentage SCP. The user wants to keep the discrete velocity and time values of their design constant, and is not overly interested in the acceleration values or changes, as long as the motor-drive unit chosen can supply it. Thus, in the MCDES, a numerical factor dependent on the SCP percentage is multiplied by the acceleration or deceleration values extracted from the MPP. The symbolic form of this multiplication factor is found by first finding  $v$  as a function of  $t_a$  or  $t_d$  for a generic SCP acceleration or deceleration ramp. This will be a function of  $a_{max}$  and the profiling percent  $S$ . Setting  $v$  as the  $v_{max}$  specified by the user will allow us to solve for  $a_{max}$  as a function of the user specified  $v_{max}$ ,  $t_a$  or  $t_d$ , and  $S$ .

From figure 3.7 we have

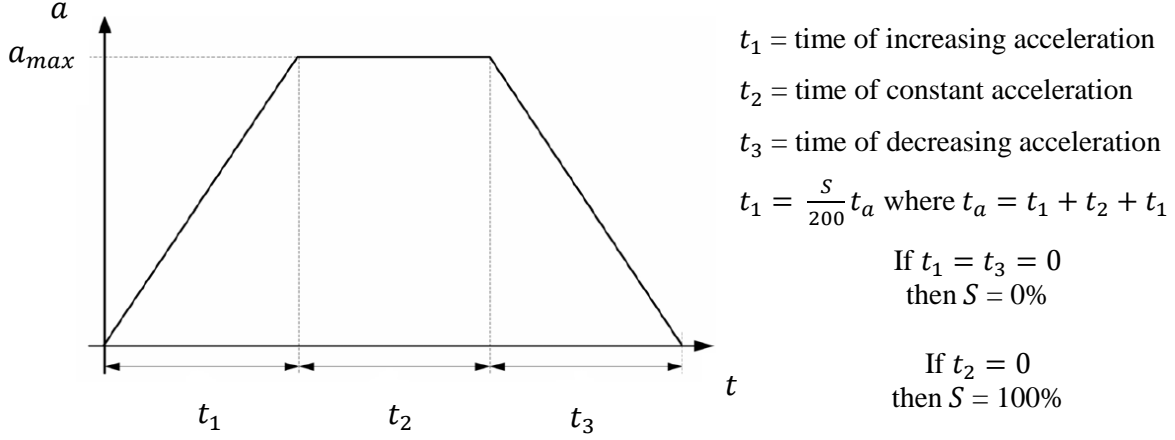
$$v_{max} = \frac{a_{max}}{2} \left( \frac{S}{200} \right) t + \left[ \left( 1 - \frac{S}{200} \right) t - \frac{S}{200} t \right] a_{max} + \left[ t - \left( 1 - \frac{S}{200} \right) t \right] \frac{a_{max}}{2} \quad (3.1)$$

Now, solving for  $a_{max}$  yields a simple multiplication factor  $\beta$  that is used to increase the accelerations or decelerations of the SCP segments based on  $S$ . This result is used in the MCDES to adjust all the maximum acceleration and deceleration values extracted from the MPP of both generic and custom LSP.

$$a_{max_{profiled}} = \beta \frac{v_{max}}{t_a}$$

$$\left| d_{max_{profiled}} \right| = \left| \beta \frac{v_{max}}{t_d} \right| \quad (3.2)$$

$$\beta = \left( \frac{200}{200-S} \right)$$



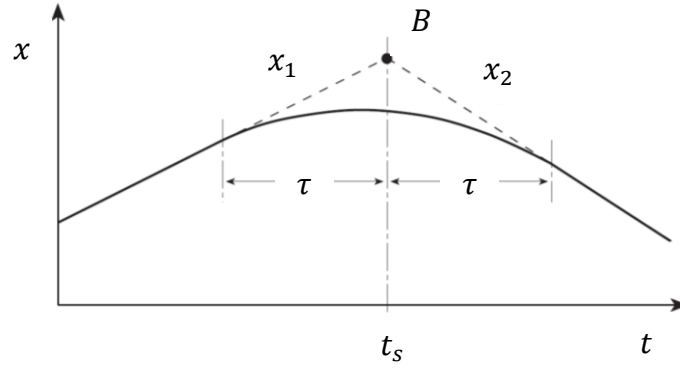
**Figure 3.7: Generic s-curve profiled acceleration ramp.**

### 3.4 Load Position Profiles

When defining the MP of a design, the speed profile of the load might not be available. In many cases, only the desired position of the load as a function of time is known. This is especially common in servo motor applications and robotics. For this reason, the option of defining a load position profile (LPP) is developed and included in the MCDES. The MCDES gives the user the option of single stroke (forward and reverse), double stroke, or custom LPP. The necessary parameters for the motor selection such as acceleration and velocity are extracted from the LPP in a similar fashion to what was done in the MPP of the LSP. Before presenting the development of equations for each LPP option, an overview of the geometric trajectory blending method used for LPP (transition window technique) is given.

### 3.4.1 Transition Window Technique

As discussed in section 3.3.2, limiting the jerk in MP is important for producing smooth speed transitions and reducing wear. This is no different for LPP than it is for LSP. LPP are more common in tracking and servo applications, and limiting the jerk in such applications not only smoothens the loads on the motor [61], but also allows for the trajectory to be tracked faster and more accurately [62]. To limit the jerk on the LPP chosen or built by the user, the MCDES uses the transition window technique developed by Parker and Taylor and modified by Lloyd [63].



**Figure 3.8: One-dimensional path segment transition.**

Suppose that the load follows a path  $x_1(t)$  and then switches to a different path  $x_2(t)$ . Both paths are assumed to have no discontinuities in position or velocity. With no transition applied, a discontinuity will occur at  $B$  ( $t = t_s$ ). To avoid this problem, a smooth transition polynomial spanning  $t \in [t_s - \tau, t_s + \tau]$  is used to connect the two paths. The total transition time  $2\tau$  is estimated as:

$$2\tau \approx \frac{\|\dot{x}_2(t_s + \tau) - \dot{x}_1(t_s - \tau)\|}{a_r} \quad (3.3)$$

where  $a_r$  is a desired reference acceleration. Since the MCDES will solve for accelerations in a wide range of applications, defining a reference application would limit

the range of actuator candidates. Therefore, in the MCDES,  $\tau$  is estimated as a fraction of  $t_s$  instead.

To determine the equation for the connecting polynomial  $x(t)$ , the transition time is first normalized to occur between 0 and 1 by defining a new time coordinate  $s$  as:

$$s = \frac{t - t_s + \tau}{2\tau} \quad (3.4)$$

The transition polynomial  $x(s)$  and the original paths  $x_1(s)$  and  $x_2(s)$  should satisfy the position, velocity and acceleration boundary conditions (BC) to ensure that the transition is compatible and smooth:

$$\begin{aligned} x(0) = x_1(0) = p_1 & & x(1) = x_2(1) = p_2 \\ \dot{x}(0) = \dot{x}_1(0) = v_1 & & \dot{x}(1) = \dot{x}_2(1) = v_2 \\ \ddot{x}(0) = \ddot{x}_1(0) = a_1 & & \ddot{x}(1) = \ddot{x}_2(1) = a_2 \end{aligned} \quad (3.5)$$

These can be fulfilled with a 5<sup>th</sup> degree polynomial with coefficients defined by a vector  $\mathbf{c}$ . If the BC defined by (3.5) are used to construct a vector  $\mathbf{b}$ , then  $\mathbf{c}$  can be found using a Hermite BC matrix  $\mathbf{H}$  [63]:

$$\mathbf{H} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 5 & 4 & 3 & 2 & 1 & 0 \\ 20 & 12 & 6 & 2 & 0 & 0 \end{bmatrix} \quad \begin{aligned} \mathbf{c} &= \mathbf{H}^{-1} \mathbf{b} \\ \mathbf{c} &= (c_5, c_4, c_3, c_2, c_1, c_0)^T \\ \mathbf{b} &= (p_1, v_1, a_1, p_2, v_2, a_2)^T \end{aligned} \quad (3.6)$$

Thus, the components of  $\mathbf{c}$  can be found as a function of the BC in equation 3.5 and the connecting polynomial  $x(s)$  can be evaluated at any point in the transition. Consequently, the motion parameters that are needed in the motor selection such as  $a_{max}$  and  $v_{max}$  can be determined through mathematical manipulation of  $x(s)$ . The two main drawbacks of this method are not of importance in our MCDES application. The first drawback is that

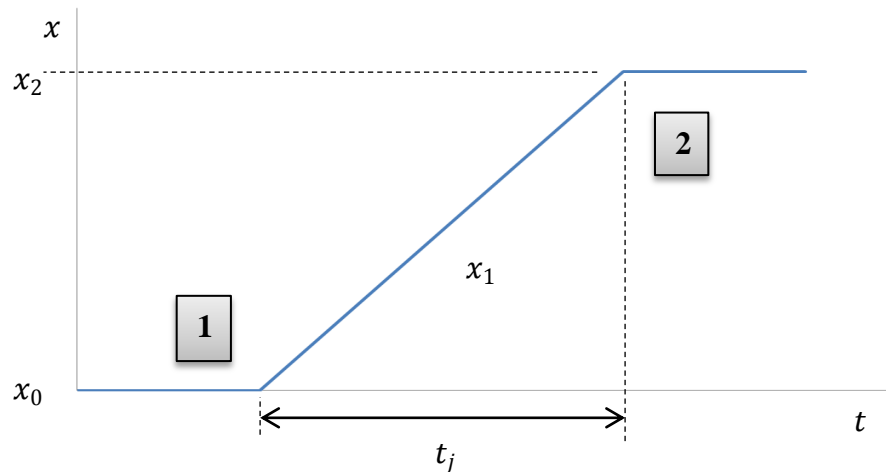
before the transition is created,  $x_2$  must be fully defined. This is true in our case, since the original unsmoothed LPP is fully defined in the MCDES. The second drawback is that there is no control over the shape of the transition. This is again is not a problem in the MCDES, since the user is not interested in the shape of the transition as long as it is compatible with the original LPP.

### 3.4.2 Generic and Custom Position Profiles

In the MCDES, the transition window technique is used to find the necessary motion parameters for motor selection from the user defined LPP. The equations are developed in symbolic form for use with any parameter values in the LPP. To do so, two types of transitions are identified and analyzed hereafter: a transition between a zero and a non-zero velocity path, and a transition between two non-zero velocity paths. The resulting equations from these two transition types can be used to fully define any transition in any generic (single or double stroke) or custom LPP.

*Single Stroke Profiles:*

Single stroke LPP can be used in positioning applications or in intermittent load motions.



**Figure 3.9: Single forward stroke LPP.**

For the user defined single forward stroke LPP in figure 3.9, transitions 1 and 2 are of type 1. Half the transition time is defined in the MCDES as  $\tau = 0.25t_j$ .

At transition location 1, the BC of equation 3.5 (components of  $\mathbf{b}$ ) are zero except:

$$x(1) = x_1(1) = p_1 = \frac{x_2 - x_0}{t_j} \tau + x_0$$

$$\dot{x}(1) = \dot{x}_1(1) = v_1 = \frac{x_2 - x_0}{t_j}$$

Finding the coefficients of the transition polynomial using the Hermite BC matrix  $\mathbf{H}$  yields expressions which are a function of the symbolic components of  $\mathbf{b}$ . However, the LPP is only defined by the user through  $x_0$ ,  $x_2$  and  $t_j$ . Hence, the equivalent expressions of the components of  $\mathbf{b}$  using these parameters are substituted into  $\mathbf{c}$ , which yields:

$$\begin{aligned} c_5 &= \left(1.5 - \frac{3}{t_j}\right)x_2 + \frac{3x_0}{t_j} & c_2 &= 0 \\ c_4 &= \left(-3.75 + \frac{7}{t_j}\right)x_2 - \frac{7x_0}{t_j} & c_1 &= 0 \\ c_3 &= \left(2.5 - \frac{4}{t_j}\right)x_2 + \frac{4x_0}{t_j} & c_0 &= 0 \end{aligned} \quad (3.7)$$

These coefficients are used to construct the transition polynomial equations for position, velocity, and acceleration:

$$\begin{aligned} x(s) &= c_5s^5 + c_4s^4 + c_3s^3 + c_2s^2 + c_1s + c_0 \\ \dot{x}(s) &= (5c_5s^4 + 4c_4s^3 + 3c_3s^2 + 2c_2s + c_1) \left(\frac{ds}{dt}\right) \\ \ddot{x}(s) &= (20c_5s^3 + 12c_4s^2 + 6c_3s + 2c_2) \left(\frac{ds}{dt}\right)^2 \end{aligned} \quad (3.8)$$

From these equations, the MPP can extract the necessary motion parameters for motor selection for this transition,  $v_{max}$  and  $a_{max}$ . The maximum velocity does not occur

during the transition and is already known to be  $v_1$ . The maximum acceleration  $a_{max}$ , however, occurs during the transition and is:

$$a_{max} = (2.5c_5 + 3c_4 + 3c_3 + 2c_2) \left( \frac{ds}{dt} \right)^2$$

If the position  $x_0$  is a zero reference, the expression for  $a_{max}$  that is used in the MCDES selection routine is simplified to (after substituting the coefficients):

$$a_{max} = \frac{1.5x_2}{4t_j\tau^2} \quad (3.9)$$

A similar analysis is done for transition 2 in figure 3.9, where all the BC are zero except:

$$\begin{aligned} x(0) &= x_1(0) = p_1 = 0.75x_2 \\ \dot{x}(0) &= \dot{x}_1(0) = v_1 = \frac{x_2 - x_0}{t_j} \end{aligned} \quad (3.10)$$

$$x(1) = x_2(1) = p_2 = x_2$$

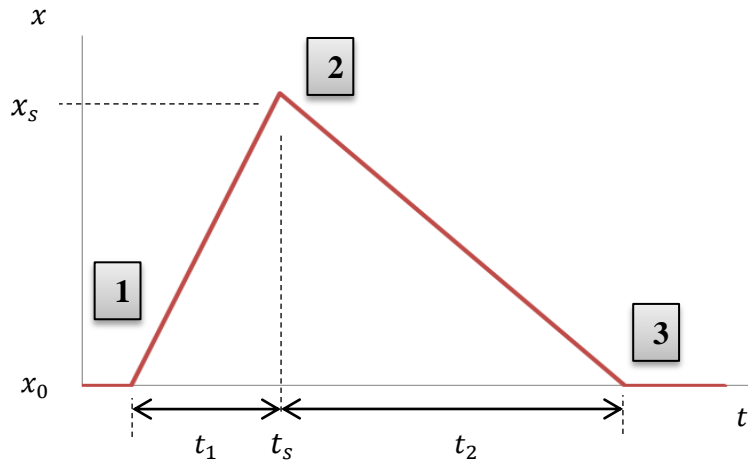
As expected, the  $a_{max}$  is identical to that in equation 3.9 except that it is negative (i.e., a deceleration). Single stroke LPP in reverse are also possible in the MCDES, where  $v_{max}$  and  $a_{max}$  values are the negatives of their forward stroke counterparts.

#### *Double Stroke Profiles:*

The main transition in double stroke LPP is between two non-zero velocity paths, as shown at point 2 in figure 3.10. The transitions at 1 and 3 are of type 1 and therefore use the same results as developed in the previous section. To develop generic equations for the transition at 2 that will work for any  $x_s$  and time input from the user,  $t_1$  is treated as not equal to  $t_2$ . Since  $\tau = 0.25t_j$ , this also means that  $a_{max}$  will not occur at  $s = 0.5$ . The value of  $s$  where  $a_{max}$  occurs has to be found first and then substituted into the

polynomial that describes the acceleration during the transition. For this case, the BC of equation 3.5 are zero except:

$$\left. \begin{aligned} x(0) = x_1(0) = p_1 &= 0.75x_s \\ \dot{x}(0) = \dot{x}_1(0) = v_1 &= \frac{x_s - x_0}{t_s} \\ x(1) = x_2(1) = p_2 &= x_s - v_2\tau = x_s - \frac{x_s - x_0}{t_2}\tau \\ \dot{x}(1) = \dot{x}_2(1) = v_2 &= -\left(\frac{x_s - x_0}{t_2}\right) \end{aligned} \right\} \quad (3.11)$$



**Figure 3.10: Double stroke LPP.**

The transition polynomial equations for position, velocity, and acceleration are the same as in equation 3.8, except the coefficients are:

$$\begin{aligned} c_5 &= 6 \left( \left(1 - \frac{\tau}{t_2}\right) x_s + \frac{\tau x_0}{t_2} - 0.75x_s \right) - 3 \left( -\left(\frac{x_s - x_0}{t_2}\right) + \frac{x_s - x_0}{t_s} \right) \\ c_4 &= 15 \left( 0.75x_s - x_s + \left(\frac{x_s - x_0}{t_2}\right) \tau \right) + 8 \left( \frac{x_s - x_0}{t_s} \right) - 7 \left( \frac{x_s - x_0}{t_2} \right) \\ c_3 &= 10 \left( \left(1 - \frac{\tau}{t_2}\right) x_s + \frac{\tau x_0}{t_2} - 0.75x_s \right) - 6 \left( \frac{x_s - x_0}{t_s} \right) - 4 \left( \frac{x_s - x_0}{t_2} \right) \\ c_2 &= 0 \\ c_1 &= \frac{x_s - x_0}{t_s} \\ c_0 &= 0.75x_s \end{aligned} \quad (3.12)$$

From this result, the MPP can extract the necessary motion parameters. Again,  $v_{max}$  occurs outside the transition ( $\max\left(\frac{x_s-x_0}{t_s}, \frac{x_s-x_0}{t_2}\right)$ ) while  $a_{max}$  occurs during the transition at a location dependent on  $x_s$  and time inputs by the user. The  $s$  at which  $a_{max}$  occurs in the transition is found using:

$$s = \frac{-0.2c_4}{c_5} \pm \frac{\sqrt{576c_4^2 - 1440c_5c_3}}{120c_5} \quad (3.13)$$

Since  $0 \leq s \leq 1$ , the correct sign of the second term is determined based on the coefficients, and  $a_{max}$  can consequently be found through the acceleration equation in (3.8). Note that  $\frac{ds}{dt}$  is a function of  $\tau$  only and is therefore a scalar value.

## 3.5 Practical Considerations

### 3.5.1 Effect of Mechanical Structure

The rectilinear  $a, d, a_{max}$ , and  $v_{max}$  values extracted from either the LSP or the LPP using MPP cannot be immediately used in the motor torque calculation. First they have to be converted into the angular equivalent reflected to the motor rotor. Then, this can be used with the overall equivalent inertia of the MS (inertia of the structure reflected to the motor rotor plus the inertia of the motor rotor) to find the required motor torque. The conversion from  $a, d, a_{max}$ , and  $v_{max}$  to  $\alpha, \alpha_d, \alpha_{max}$ , and  $\omega_{max}$  is dependent on the MS being motorized. The MCDES identifies the structure and, using the necessary geometric parameters, performs the conversion. Table 3.1 outlines the different conversions for the different structures.

**Table 3.1: Effect of MS on profile parameters.**

Generic MS	Drive Mechanism	
	Direct Drive	with Gears
<ul style="list-style-type: none"> <li>• Hoisting</li> <li>• Belt conveyor</li> <li>• Horizontal pulling</li> <li>• Nip roll</li> <li>• Center or surface driven winder</li> </ul>	$\alpha = \frac{a}{r}$ $\omega_{max} = \frac{60v_{max}}{2\pi r}$ <p>where <math>r</math> [m]:</p> <ul style="list-style-type: none"> <li>• Pulley radius (hoisting or horizontal pulling)</li> <li>• Driving cylinder radius (belt conveyor)</li> <li>• Driver, winding, or support roller radius (Nip roll, center, or surface driven roller, respectively)</li> </ul>	$\alpha = \frac{pa}{r}$ $\omega_{max} = \frac{60v_{max}p}{2\pi r}$ <p>where <math>p</math>:</p> <ul style="list-style-type: none"> <li>• Gearbox reduction ratio</li> </ul> <p>and <math>r</math> is the same as for direct drive</p>
<ul style="list-style-type: none"> <li>• Rack and pinion</li> </ul>	$\alpha = \frac{2\pi a}{P * n}$ $\omega_{max} = \frac{60v_{max}}{P * n}$ <p>where <math>P</math> [m]:</p> <ul style="list-style-type: none"> <li>• Pitch of the rack's teeth</li> </ul> <p>and <math>n</math>:</p> <ul style="list-style-type: none"> <li>• # of teeth on pinion</li> </ul>	$\alpha = \frac{2\pi ap}{P * n}$ $\omega_{max} = \frac{60v_{max}p}{P * n}$ <p>where <math>p</math>:</p> <ul style="list-style-type: none"> <li>• Gearbox reduction ratio</li> </ul> <p>and <math>P, n</math> are the same as for direct drive</p>
<ul style="list-style-type: none"> <li>• Ball screw drive</li> </ul>	$\alpha = \frac{2\pi a}{P}$ $\omega_{max} = \frac{60v_{max}}{P}$ <p>where <math>P</math> [m]:</p> <ul style="list-style-type: none"> <li>• Lead of the feed screw</li> </ul>	$\alpha = \frac{p2\pi a}{P}$ $\omega_{max} = \frac{60v_{max}p}{P}$ <p>where <math>p</math>:</p> <ul style="list-style-type: none"> <li>• Gearbox reduction ratio</li> </ul>

Generic MS	Drive Mechanism	
	Direct Drive	with Gears
<ul style="list-style-type: none"> <li>Rotary table</li> </ul>	$\omega_{max} = \frac{\theta}{6t}$ $\alpha_{max} = \frac{2\pi\omega_{max}}{60t}$ <p>where <math>\theta</math> [deg]:</p> <ul style="list-style-type: none"> <li>Angular step displacement</li> </ul> <p>and <math>t</math> [sec]:</p> <ul style="list-style-type: none"> <li>Step time</li> </ul> <p>Note: the MPP identifies <math>\theta</math> from the LPP in the Rotary Table scenario (LSP is not practical here)</p>	$\omega_{max} = \frac{p\theta}{6t}$ $\alpha_{max} = \frac{2\pi p\omega_{max}}{60t}$ <p>where <math>p</math>:</p> <ul style="list-style-type: none"> <li>Gearbox reduction ratio</li> </ul>
<ul style="list-style-type: none"> <li>Flywheel</li> <li>Propeller</li> </ul>	<p>There is <u>no conversion process</u> since the original motion parameters are already angular. Furthermore, servo-style LSP or LPP are not practical here. In the case that these MS are chosen, the user is prompted for the maximum angular velocity <math>\omega_{max}</math> and the ramp up time <math>t</math>. Then <math>\alpha</math> is simply <math>\frac{\omega_{max}}{t}</math></p>	$\alpha_{max} = \frac{p\omega_{max}}{t}$ <p>where <math>p</math>:</p> <ul style="list-style-type: none"> <li>Gearbox reduction ratio</li> </ul>
Custom mechanical components	Both direct drive and gear drive scenarios	
<ul style="list-style-type: none"> <li>Belt pulley transmission</li> <li>Chain sprocket transmission</li> <li>Gearbox (somewhere in the custom design instead of a motor gearhead)</li> </ul>	<p>Depending on the generic MS chosen, <math>\alpha, \alpha_d, \alpha_{max}</math>, and <math>\omega_{max}</math> from above are multiplied by the transmission ratio <math>N_r</math> which was found from the geometry of the transmission (table A.2) or the known reduction ratio of the gearbox (section 2.4)</p>	

### 3.5.2 RMS Torque

The converted motion parameters that are found from the MP are used with the equivalent inertia of the MS to find the required motor torque. In the most conservative approach, the maximum continuous torque the motor can supply (nominal torque) is compared with the maximum torque of the application  $T_{max}$ :

$$T_{max_i} = J_{eq_i} \alpha_{max} \quad i = \text{motor candidate} \quad (3.14)$$

Even for a single MS and DC, there are several  $T_{max}$  values. This is because for every motor candidate there is a different rotor inertia  $J_m$  and thus a different  $J_{eq}$  and hence the torque requirement changes from rotor to rotor. Equation 3.14 must be evaluated for every motor and motor-gear combination in the MCDES database and each  $T_{max}$  result must be compared with its corresponding motor nominal torque. However, using equation 3.14 consistently results in an oversized motor, since the nature of the load throughout the entire DC is not taken into consideration. In the MCDES, this approach is modified to be dependent on the DC type discussed in section 3.2. The MCDES finds the root mean squared (rms) torque of the application (for every motor and motor-gear candidate), as given by equation 3.15, and compares it to the nominal torques of the corresponding candidate motor. The efficiencies  $\eta$  of the candidate motors are also taken into consideration.

$$T_{RMS_i} = \sqrt{\frac{(T_{a_i} + T_{c_i})^2 t_a + T_{c_i}^2 t_c + (T_{d_i} + T_{c_i})^2 t_d + T_h^2 t_h}{t_a + t_c + t_d + t_h}} \quad (3.15)$$

where

$T_a$ : Acceleration torque [Nm]	$T_h$ : Holding torque [Nm]	$t_c$ : Constant speed time [s]
$T_d$ : Deceleration torque [Nm]	$t_a$ : Acceleration time [s]	$t_h$ : Holding time [s]
$T_c$ : Constant torque [Nm]	$t_d$ : Deceleration time [s]	

The torques  $T_a$  and  $T_d$  are found easily using  $T_{a_i} = J_{eq_i} \alpha$  and  $T_{d_i} = J_{eq_i} \alpha_d$ , respectively.  $T_c$  is the torque required to run the application at constant speed, overcoming friction and other resistance. The expressions for  $T_c$  vary depending on the MS and speed. Finally,  $T_h$  is appropriate only in applications with a holding load during sections of no movement (e.g., stationary load in a hoist). With these individual torque values available, and since the times associated with each section of any MP are recorded during MPP,  $T_{RMS}$  can be found for every candidate.

If the DC is continuous, the rms torque is almost equal to the constant torque for any application. The motor selection is thus based on two criteria:

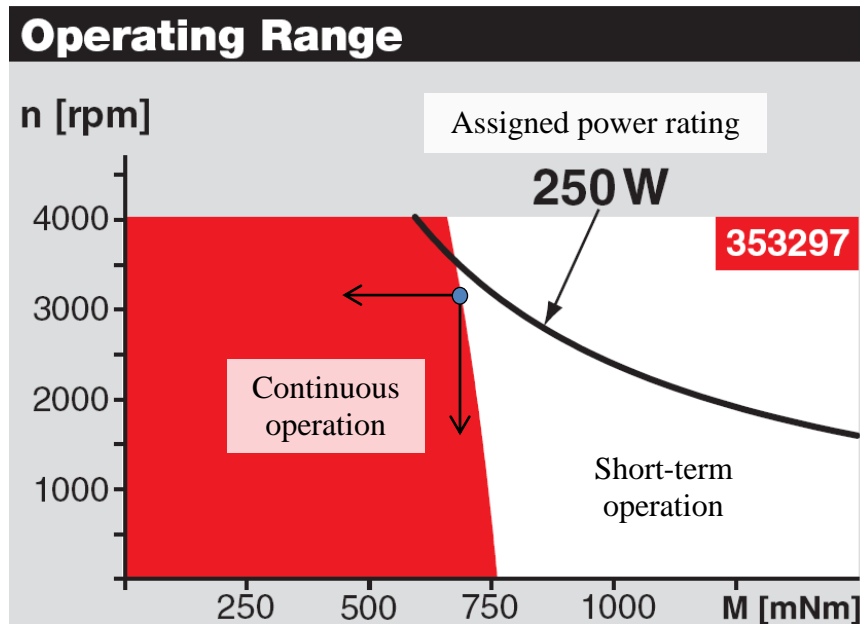
$$T_{c_i} \leq \eta_i T_{motor\_nominal_i} \qquad \omega_{max} \leq \omega_{motor\_nominal_i} \qquad (3.16)$$

Even though the required torque will be larger than  $T_c$  during acceleration and deceleration, it is safe to use the above torque criterion for continuous applications. This is because every motor has the capability of being overdriven for short periods of time. This is not the case for intermittent or varying DC, as the overdriving time becomes almost equal to or sometimes more than the constant speed times.

For intermittent or varying DC, the MCDES uses equation 3.15 to find  $T_{RMS}$  for every drive candidate, and compares  $T_{RMS}$  to  $T_{motor\_nominal}$ . It should be noted that for these DC, there may be many acceleration, deceleration, and constant speed segments. Therefore, equation 3.15 includes all the segment times and torques for the entire DC to compute the correct  $T_{RMS}$ . Also,  $T_{max}$  is computed for every drive candidate using equation 3.14. This is now needed since the overdrive of the motor might not be for short periods of time anymore. In addition to equation 3.16, the motor selection criteria now include:

$$T_{max_i} \leq \eta_i T_{motor\_max_i} \quad (3.17)$$

In the MCDES, the specifications of over thirty Maxon<sup>®</sup> motors and over 15 compatible gearheads are used in building the motor and gear database. As explained in section 2.4, this database is built in Excel<sup>®</sup> and is expandable to many more motors and gears. Also, it can easily be modified without altering the MCDES code. The values for  $T_{motor\_max}$ ,  $T_{motor\_nominal}$ ,  $\omega_{motor\_nominal}$ , and  $\eta$  for each motor are taken directly from this database. With the selection criteria of equations 3.16 and 3.17, it is ensured that the selected motor will operate in the safe region for continuous operation, as seen in the example operating range graph for one of the motors, in figure 3.11.



**Figure 3.11: Operating range of a sample motor from the MCDES database.**

The blue point represents the coordinate of  $\omega_{motor\_nominal}$  and  $T_{motor\_nominal}$  of the motor. If this motor is chosen by the MCDES, the application will require less speed and torque than this point, in the directions of the arrows shown in the figure. Thus, the

operation remains in the safe region for continuous operation. Nevertheless, overdrive towards the right and top can be accommodated for short periods of time.

### 3.5.3 Thermal Response

The operating efficiency and performance of a motor is significantly affected by the operating temperature. Temperature considerations thus affect the power consumption, cost of operation, and even the life of a motor. The thermal performance of a motor is predominantly determined in the design stage of the motor, through factors such as the insulating material, housing type, and heat transfer paths for convection and conduction. The thermal performance of a motor does not only depend on its design and materials, it is also greatly affected by the operating conditions of the motor [64]. Testing the thermal performance (most importantly, the temperature rise) of a motor can be done both experimentally and analytically. Experimentally, the temperature rise can be measured with the motor running in an environment that mimics the actual operational environment. However, a simpler and sufficiently correct alternative is to use theoretical calculations. In order to carry out theoretical temperature rise calculations, the following motor parameters are needed:

- Torque Constant  $K_T$  [Nm/A]
- Terminal Resistance  $R_T$  [ $\Omega$ ]
- Maximum running temperature  $Temp_{max}$  [ $^{\circ}\text{C}$ ]
- Thermal Resistance  $R_{TH}$  [ $^{\circ}\text{C}/\text{W}$ ] – the thermal resistance of a motor indicates how well a motor can dissipate the generated heat during operation

The MCDES motor database includes these parameters for each of its motors. In the MCDES, the thermal response of the motor based on the application is not a part of the selection process. However, a warning will be issued to the user if the temperature rise causes the motor's running temperature to exceed the motor's maximum temperature  $Temp_{max}$ . This temperature rise depends on the torque requirement and is calculated using the following steps:

$$I_{RMS} = \frac{T_{RMS}}{K_T}$$

Here  $T_{RMS}$  is the rms torque of the load [Nm] and  $I_{RMS}$  is the rms current in the motor windings [A].

$$P_D = I_{RMS}^2 R_T$$

Here  $P_D$  is the power dissipation [W] of the motor.

$$\Delta Temp = P_D R_{TH}$$

Here  $\Delta Temp$  is the temperature rise of the motor [ $^{\circ}\text{C}$ ].

$$Temp_A + \Delta Temp = Temp_{motor}$$

The actual temperature of the motor,  $Temp_{motor}$ , is the ambient temperature  $Temp_A$  plus the temperature rise. For safe operation and for optimal efficiency and performance, the motor must run below the maximum operating temperature prescribed by the manufacturer:

$$Temp_{motor} \leq Temp_{max}$$

## 3.6 Implementation and Verification

The MCDES with the features outlined in chapters 2 and 3 is implemented in the Microsoft<sup>®</sup> Visual C++ integrated development environment. ES shell software, such as CLIPS<sup>®</sup> or Jess<sup>®</sup>, were not used since they were difficult to synchronize with an external database and had poor mathematical features (limited operators).

### 3.6.1 Program Segments

The core code of the MCDES is written as a large list of sequential if-then type clauses, or rules. Based on the user input in the MCDES GUI (shown in the next section), the routine enters through the applicable rules in a prescribed order. The core code of the MCDES can be found in Appendix C. In this section, the order of the rules is explained and the location of the main groups of rules in the code is identified with line numbers.

In the simplest case input into the GUI, the MS and DC are generic. This is the first set of rules in the MCDES. If the MS is generic, the routine enters the first group of rules which process the DC for accelerations, velocities, times, and other related parameters. The DC can be a generic or custom LSP or LPP. Then, based on which generic MS is input, the routine enters the corresponding group of rules for that structure. These rules process the inertia and all related aspects of each structure. If the DC was generic, the processed parameters from the DC and the mechanical structure are sent to a corresponding rule group for the MS which will determine the torques, perform the selection based on the motors and gears in the database, and perform thermal analysis. If the DC was custom, these tasks are all done in the same group of rules that processed the MS. The locations of the rules described above, in the code of Appendix C, are given in table 3.2.

**Table 3.2: Location of rule groups for generic MS analysis.**

Rule group				Line number	
DC processing for generic MS				1971 – 2143	
Generic MS processing (and motor selection for custom DC)				2155 – 3258	
Hoisting	2155		Rack and pinion	2726	
Flywheel	2291		Nip roll	2866	
Conveyor-Belt drive	2312		Rotary table	3003	
Horizontal pulling	2458		Center/Surface winder	3031	
Ball-Screw drive	2592		Propeller	3214	
Motor selection functions for generic MS and DC				55 – 970	
Hoisting	55		Rack and pinion	511	
Flywheel	148		Nip roll	604	
Conveyor-Belt drive	232		Rotary table	697	
Horizontal pulling	325		Center/Surface winder	781	
Ball-Screw drive	418		Propeller	888	

If the user builds a custom MS in the GUI, the routine enters groups of rules which first build the line directly connected to the motor. The custom components in this line are chosen and the line inertia is processed. If there are transmission components, their corresponding parameters are found and recorded for use in MPP, as given in table 3.1. If there are no multi-output gearboxes in this first line, the routine enters into the rules which process the DC for the line. At the end of the line there must be a generic MS which performs the task. So the routine enters into the rules which process the inertia and all related aspects of this generic MS. Once again, if the DC was custom, the motor selection would be done immediately. Otherwise, all necessary parameters are sent to the corresponding function for the MS which will perform the selection and thermal analysis.

If there is a multi-output gearbox in the line connected to the motor, each output line from the gearbox is built separately. Each line comprises of custom components leading up to a final generic MS, the final MS itself, and the DC of that MS. Each line is

analyzed using its corresponding rule group, and the necessary parameters from the DC and the MS are found. The motor selection is performed after the load requirements of all lines are joined and reflected to the motor through the line before the gearbox. Table 3.3 identifies the locations of the rule groups for custom MS analysis in the code given in Appendix C.

**Table 3.3: Location of rule groups for custom MS analysis.**

Rule group				Line number	
Building first line of custom components connected to the motor				3277 – 3374	
DC processing for the final MS of a single line custom design				3379 – 3551	
Final MS processing in the line (and motor selection for custom DC)				3556 – 4750	
Hoisting	3556		Rack and pinion	4178	
Flywheel	3701		Nip roll	4327	
Conveyor-Belt drive	3725		Rotary table	4473	
Horizontal pulling	3884		Center/Surface winder	4507	
Ball-Screw drive	4030		Propeller	4703	
Multi-output gearbox details				4760 – 4777	
*Building lines of custom components connected to the gearbox				4777 – 4903	
*DC processing of final MS for lines connected to the gearbox				4908 – 5080	
*Final MS processing in the line (and motor selection for custom DC)				5082 – 6261	
Hoisting	5082		Rack and pinion	5699	
Flywheel	5226		Nip roll	5847	
Conveyor-Belt drive	5248		Rotary table	5991	
Horizontal pulling	5407		Center/Surface winder	6021	
Ball-Screw drive	5552		Propeller	6216	
Motor selection functions for custom MS and DC				980 – 1942	
Hoisting	980		Rack and pinion	1463	
Flywheel	1078		Nip roll	1559	
Conveyor-Belt drive	1170		Rotary table	1655	
Horizontal pulling	1268		Center/Surface winder	1742	
Ball-Screw drive	1365		Propeller	1852	

\*: iterated for number of outputs of the gearbox

### 3.6.2 Synchronization with Microsoft Excel®

To facilitate easier input of the geometric and mass properties of the components of a design, the MCDES code was synchronized with Microsoft Excel®. Ultimately, this feature has been added to allow for easier integration of the MCDES with the MEDOS of the same domain, as discussed in section 2.1. The MCDES code can interpret the type of design that is created by the user (generic or custom MS and DC) and can read the required properties from an Excel® spreadsheet. The motor selection is performed based on the values. The spreadsheet used by the MCDES follows the structure of a template (figure 3.12) which clearly outlines the procedure to input any design that the MCDES can handle.

This method for interacting with the MCDES proves to be much more practical than using the compiler's GUI. Modifications in the design can be made much faster and the user builds the whole design at once rather than each stage separately. As will be

WELCOME to Motion Control System Design Expert --> MCSD-Expert v1		
Upload your available motor data to updatedMotorData.csv and upload your available gear data to updatedGearData.csv		
Follow the steps carefully. Fill in all the applicable cells based on your choices. When you are done, save as a .csv file and run the MotorExpert VisualStudio application		
<b>SINGLE READY MADE APPLICATION</b>		
1. Hoisting	3. Conveyor-belt	5. Ball-Screw Drive
2. Flywheel	4. Horizontal Pull	6. Rack and Pinion
For a single ready-made application, enter you application choice from above and the corresponding details (if custom, choose custom (11) in application and move to the custom section below):		
Application	a.) Continuous load (e.g. aggregate) or b.) divisible load (e.g. discrete parts)? (a/b)	Total weight of load [kg]
3	a	0
<b>For a single ready-made application, enter you application choice from above and the corresponding details (if custom, choose custom (11) in application and move to the custom section below):</b>		
1. Triangular LSP	3. CUSTOM LSP	5. Single stroke reverse LPP
2. Trapezoidal LSP	4. Single stroke forward LPP	6. Forward-Reverse LPP
Enter the duty cycle (load-speed profile or load-position profile) here (do not enter any values if you chose Flywheel, Rotary table, or Propeller as an application). If you choose a custom LSP, enter 3 in A16 th		
Duty Cycle	Acceleration (or constant speed), deceleration, or end (back to 0) segment? (a/d/e)	Duration of final deceleration segment [s]
3	a	3
0	a	3
0	e	6
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
<b>CUSTOM APPLICATION (if applicable)</b>		
11. Solid connecting shaft	13. Coupling	15. Belt-pulley transmission
12. Hollow connecting shaft	14. Chain-sprocket transmission	16. Roller Bearing
Build the order of components connected to the motor's rotor from the components above (limited to five). Then, choose an applicable final ready-made application which will drive/move the load and the co		
If you choose a multi-output gearbox as one of your components (can only choose once), enter the details and then move on to the multi-output gear box section below.		
Component	\	\
0	0	0
Component	\	\
0	0	0
Component	\	\
0	0	0
Component	\	\
0	0	0
Component	\	\
0	0	0

Figure 3.12: Part of the MCDES GUI template in Excel®.

explained in chapter 4, having the design in an Excel<sup>®</sup> spreadsheet also allows for advanced processing of the design data, such as optimization of custom components like shafts and transmissions. The Excel<sup>®</sup> spreadsheet can be synchronized with optimization routines in Matlab<sup>®</sup> which read the proposed design values, optimize them based on imbedded expertise, and then overwrite the proposed values with the optimized values.

### 3.6.3 Verification

The MCDES was verified with the use of pre-evaluated test scenarios. Due to the very large number of design possibilities using custom MS and DC, the verified scenarios were mainly generic MS and DC. That being said, some custom MS and DC have been tested. A custom MS from an existing machine will be presented in this section. For the generic MS and DC verification, the MCDES results were compared to motors chosen manually by a human from the same motor-gear list of the MCDES database. Many different scenarios were run. As a sample, the results for three generic MS are presented in table 3.4. During all verification procedures, there were evident differences between using the MCDES and the manual selection of the motor or motor gear drive:

- The ES takes a fraction of the time (especially in custom MS and DC, or selections that require a gearhead)
- Inexperienced individuals have trouble deriving and finding the necessary inertia and DC formulas
- Advanced features such as SCP are often discarded (or the effects exaggerated) in manual selection due to the difficulty of exact calculation and time consumption
- The MCDES does not mistakenly miss selections or lose competence during extended selection procedures

**Table 3.4: Verification of generic MS and DC.**

Mechanical structure	Design details	MCDES selection (sample details for first design of each MS)
<b>Hoisting</b>	$W = 20 \text{ kg}$ ; $m_R = 0.5 \text{ kg}$ ; $D = 0.15 \text{ m}$ ; $m_p = 2 \text{ kg}$ Triangular LSP: $v = 5 \text{ m/s}$ ; $t_a = 3 \text{ sec}$ ; $t_d = 6 \text{ sec}$ ; SCP = 75%	<p><b>1.) Maxon Motor model 136210 with gearhead model 203115</b>  <math>(T_a = 0.230 \text{ Nm}</math>; <math>T_d = 0.115 \text{ Nm}</math>; <math>J_m = 2.09\text{E-}5 \text{ kgm}^2</math>; <math>J_d = 0.0056 \text{ kgm}^2</math>; <math>J_g = 1.5\text{E-}6 \text{ kgm}^2</math>;  <math>T_{required} = 0.163 \text{ Nm}</math>; <math>T_{motor-nom} = 0.285 \text{ Nm}</math>; <math>\omega_{max}</math> (with gearhead) = 7640 rpm; <math>\omega_{max-motor} = 8380 \text{ rpm}</math>; <math>p = 12</math>; <math>T_{winding} = 32 \text{ }^\circ\text{C}</math>; <math>T_{max} = 100 \text{ }^\circ\text{C})</math></p> <p><b>2.) Maxon Motor model 353297 with gearhead model 110408</b>  <math>(T_a = 0.739 \text{ Nm}</math>; <math>T_d = 0.369 \text{ Nm}</math>; <math>J_m = 1.34\text{E-}4 \text{ kgm}^2</math>; <math>J_d = 0.0056 \text{ kgm}^2</math>; <math>J_g = 1.65\text{E-}5 \text{ kgm}^2</math>;  <math>T_{required} = 0.522 \text{ Nm}</math>; <math>T_{motor-nom} = 0.746 \text{ Nm}</math>; <math>\omega_{max}</math> (with gearhead) = 2356 rpm; <math>\omega_{max-motor} = 3270 \text{ rpm}</math>; <math>p = 3.7</math>; <math>T_{winding} = 36 \text{ }^\circ\text{C}</math>; <math>T_{max} = 100 \text{ }^\circ\text{C})</math></p>
	$W = 11 \text{ kg}$ ; $m_R = 0.5 \text{ kg}$ ; $D = 0.15 \text{ m}$ ; $m_p = 2 \text{ kg}$ Triangular LSP: $v = 0.75 \text{ m/s}$ ; $T = 2 \text{ sec}$ ; SCP = 50%	<p><b>1.) Maxon motor model 353297</b>  <b>2.) Maxon motor model 353300</b>  <b>3.) Maxon motor model 118638 with Maxon gearhead model 110135</b></p>
	$W = 100 \text{ kg}$ ; $m_R = 0.5 \text{ kg}$ ; $D = 0.15 \text{ m}$ ; $m_p = 10 \text{ kg}$ Single Stroke LPP (smoothened): $d = 0.15 \text{ m}$ ; $T = 2 \text{ sec}$ ;	<p><b>1.) Maxon motor model 118628 with Maxon gearhead model 110135</b>  <b>2.) Maxon motor model 118630 with Maxon gearhead model 110135</b>  <b>3.) Maxon motor model 118638 with Maxon gearhead model 110135</b></p>

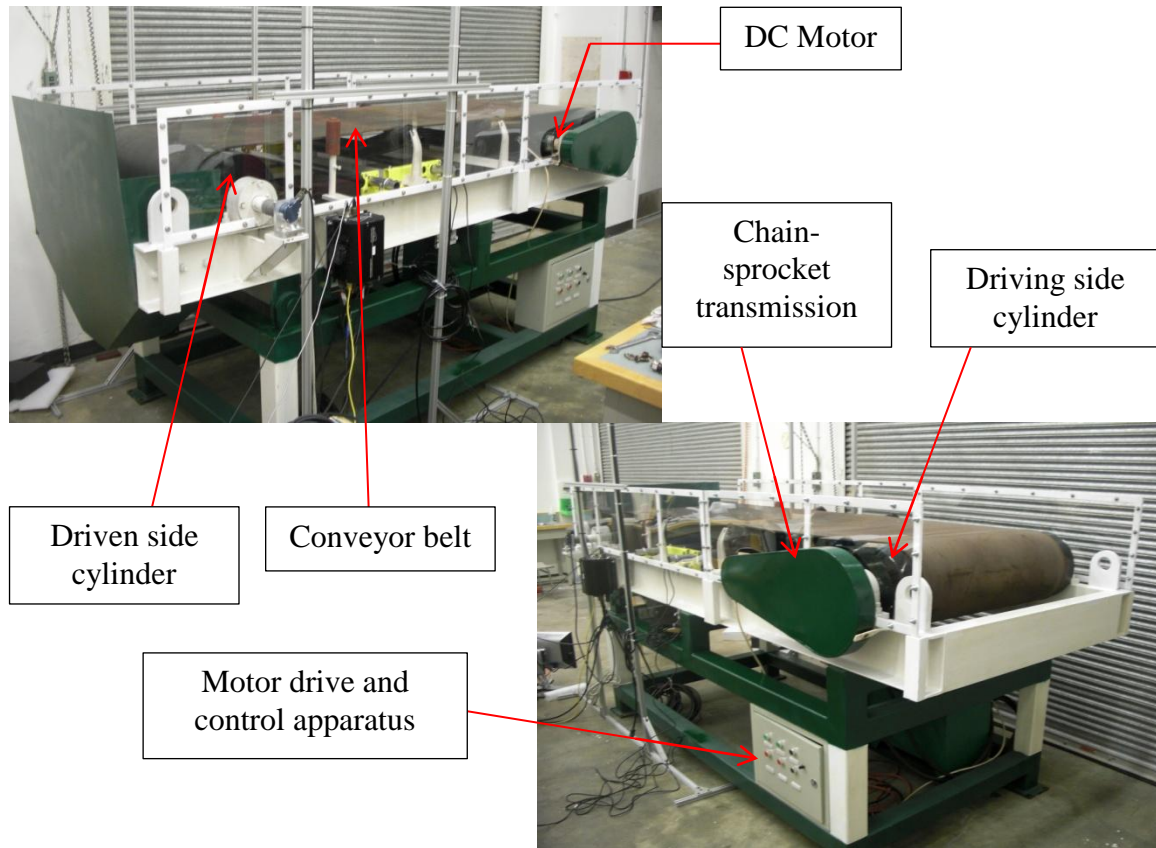
Mechanical structure	Design details	MCDES selection (sample details for first design of each MS)
<b>Belt conveyor</b>	<p><math>W = 12.8</math> kg; <math>m_c = 1</math> kg; <math>n = 3</math>; <math>D_s</math> &amp; <math>D_d = 0.15</math> m; <math>m_d</math> &amp; <math>m_s = 2</math> kg;</p> <p>Single stroke LPP (smoothened):  <math>d = 0.225</math> m; <math>T = 2</math> sec</p>	<p><b>1.) Maxon Motor model 311535 with gearhead model 134163</b>  <math>(T_a = T_d = 0.0267</math> Nm; <math>J_m = 8.91\text{E-}7</math> kgm<sup>2</sup>; <math>J_d = 0.0056</math> kgm<sup>2</sup>; <math>J_g = 4\text{E-}8</math> kgm<sup>2</sup>; <math>T_{required} = 0.0267</math> Nm; <math>T_{motor-nom} = 0.059</math> Nm; <math>\omega_{max}</math> (with gearhead) = 1518 rpm; <math>\omega_{max-motor} = 15600</math> rpm; <math>p = 53</math>; <math>T_{winding} = 26.4</math> °C; <math>T_{max} = 100</math> °C)</p> <p><b>2.) Maxon Motor model 309755 with gearhead model 166938</b>  <math>(T_a = T_d = 0.0427</math> Nm; <math>J_m = 1.83\text{E-}6</math> kgm<sup>2</sup>; <math>J_d = 0.0056</math> kgm<sup>2</sup>; <math>J_g = 8\text{E-}8</math> kgm<sup>2</sup>; <math>T_{required} = 0.0427</math> Nm; <math>T_{motor-nom} = 0.0697</math> Nm; <math>\omega_{max}</math> (with gearhead) = 945 rpm; <math>\omega_{max-motor} = 16800</math> rpm; <math>p = 33</math>; <math>T_{winding} = 27.2</math> °C; <math>T_{max} = 100</math> °C)</p>
	<p><math>W = 5.38</math> kg; <math>m_c = 0.5</math> kg; <math>n = 9</math>; <math>D_s</math> &amp; <math>D_d = 0.1</math> m; <math>m_d</math> &amp; <math>m_s = 1</math> kg;</p> <p>Single stroke LPP (smoothened):  <math>d = 0.25</math> m; <math>T = 1</math> sec</p>	<p><b>1.) Maxon motor model 136196 with Maxon gearhead model 203119</b></p> <p><b>2.) Maxon motor model 136210 with Maxon gearhead model 203119</b></p> <p><b>3.) Maxon motor model 136209 with Maxon gearhead model 203115</b></p>
	<p><math>W = 6.6</math> kg; <math>m_c = 1</math> kg; <math>n = 5</math>; <math>D_s</math> &amp; <math>D_d = 0.15</math> m; <math>m_d</math> &amp; <math>m_s = 5</math> kg;</p> <p>Triangular LSP:  <math>v = 0.9375</math> m/s; <math>T = 2.5</math> sec; SCP = 100%</p>	<p><b>1.) Maxon motor model 353295 with Maxon gearhead model 110410</b></p>

Mechanical structure	Design details	MCDES selection (sample details for first design of each MS)
Ball screw drive	<p><math>W</math> (load and table) = 25 kg; <math>P = 0.049</math> m; <math>J_{feedscrew} = 5E-5 \text{ kgm}^2</math>; <math>\eta = 85\%</math>;</p> <p>Single Stroke LPP (smoothened): <math>d = 1</math> m; <math>T = 10</math> sec</p>	<p><b>1.) Maxon Motor model 226815</b>  <math>(T_a = T_d = 0.0235 \text{ Nm}; J_m = 1.25E-6 \text{ kgm}^2; J_{total} = 0.00286 \text{ kgm}^2; T_{required} = 0.0235 \text{ Nm}; T_{motor-nom} = 0.0299 \text{ Nm}; \omega_{max} = 245 \text{ rpm}; \omega_{max-motor} = 5050 \text{ rpm}; T_{winding} = 33 \text{ }^\circ\text{C}; T_{max} = 85 \text{ }^\circ\text{C})</math></p> <p><b>2.) Maxon Motor model 236644</b>  <math>(T_a = T_d = 0.0235 \text{ Nm}; J_m = 3.9E-6 \text{ kgm}^2; J_{total} = 0.00286 \text{ kgm}^2; T_{required} = 0.0235 \text{ Nm}; T_{motor-nom} = 0.0331 \text{ Nm}; \omega_{max} = 245 \text{ rpm}; \omega_{max-motor} = 2980 \text{ rpm}; T_{winding} = 32 \text{ }^\circ\text{C}; T_{max} = 85 \text{ }^\circ\text{C})</math></p>
	<p><math>W = 10.5</math> kg; <math>P = 0.038</math> m; <math>J_{feedscrew} = 6.2E-5 \text{ kgm}^2</math>; <math>\eta = 90\%</math>;</p> <p>Triangular LSP: <math>v = 3</math> m/s; <math>t_a = 5</math> sec; <math>t_d = 10</math> sec; SCP = 100%</p>	<p><b>1.) Maxon motor model 136204</b>  <b>2.) Maxon motor model 136210</b>  <b>3.) Maxon motor model 136209</b></p>
	<p><math>W = 50</math> kg; <math>P = 0.02</math> m; <math>J_{feedscrew} = 6E-5 \text{ kgm}^2</math>; <math>\eta = 70\%</math>;</p> <p>Triangular LSP: <math>v = 10</math> m/s; <math>t_a = 3</math> sec; <math>t_d = 4</math> sec; SCP = 100%</p>	<p>No motor only or motor-gear drive found suitable for the design. The design <b>infeasible</b> with the available database.</p>

Verification of the custom MS and DC feature of the MCDES is also performed.

In addition to comparing the results of the MCDES to manual analysis, the MCDES was

used to select a drive for an existing industrial scale setup. The following real life design data for a conveyor belt used for transporting mining aggregate was input into the MCDES. All pictures and design data (Appendix D) are courtesy of Motion Metric International Corporation.



**Figure 3.13: Existing conveyor belt setup used for custom verification.**

The following geometric dimensions, mass parameters and LPP were used in the MCDES. These dimensions and parameters are the real data obtained from the engineering drawings of the setup in figure 3.13. The specifications of the motor that was selected by the MCDES was compared to the motor being used in the existing setup, which had a  $T_{motor-nom}$  of 2.9 Nm and a  $\omega_{max-motor}$  of 2500 rpm.

**Table 3.5: Design data for custom conveyor belt MS (see Appendix D).**

Parameter	Value
Maximum load mass	30 kg
Belt width, thickness, and mass	0.762 m, 0.008 m, 5.7 kg
Conveyor length (between cylinder centers)	2.33 m
Cylinder diameter (driving = driven)	0.3048 m (1 foot)
Cylinder mass	10 kg
Large sprocket diameter, mass	0.3 m, 4.54 kg
Small sprocket diameter, mass	0.06, 0.075, or 0.09 m (use largest), 0.5 kg
Chain weight	~ 1 kg
Chain sprocket drive efficiency	95%

To simulate the largest torque requirement for the motor, the small sprocket diameter used was the largest of the three available choices. Also, the DC input was a single stroke LPP which moved the load through the complete length of the conveyor within 7 seconds. This DC requirement ensures that the load moves at the maximum speed specified by the manufacturer (65 feet per minute). Since the DC input is a single stroke LPP, the load starts and ends with zero velocity. The large accelerations and decelerations associated with this profile are needed to select a motor that can fulfil the highest load and speed requirements. With the design data outlined above, the  $T_{required}$  and  $\omega_{max}$  were 3.18 Nm and 1044 rpm, respectively. Hence, the MCDES selection is within a 10% range of the actual motor for the setup. This discrepancy is mainly attributed to the use of “worst-case scenario” design data to simulate the highest requirements that the motor will encounter.

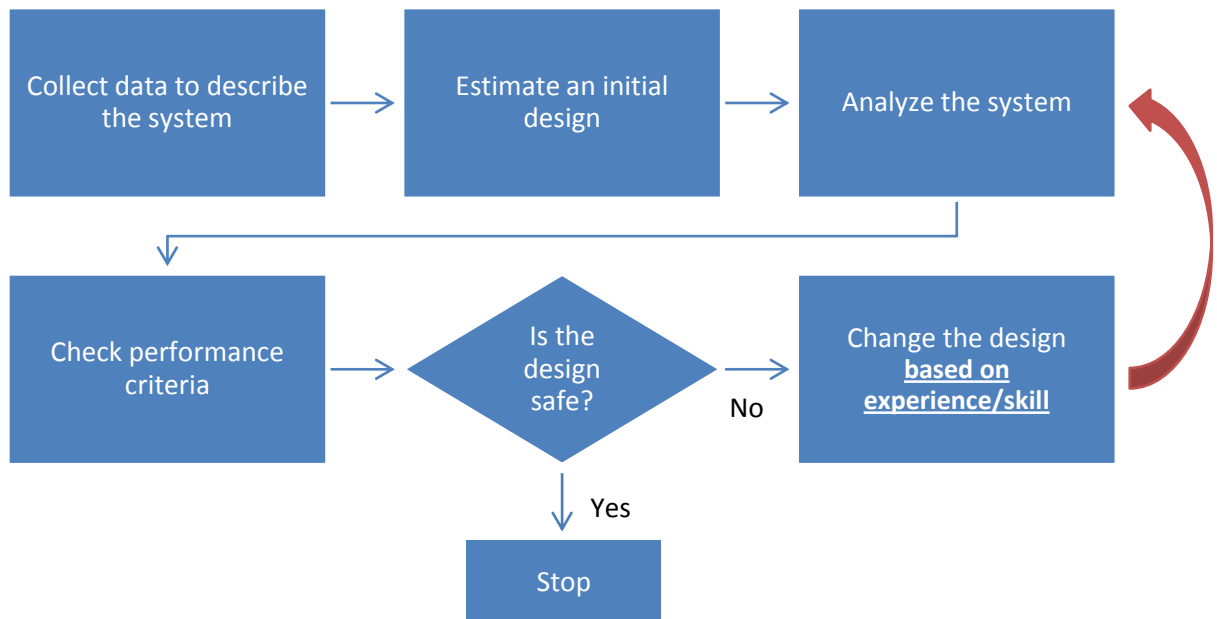
## **Chapter 4**

# **Optimization of Mechanical Components**

### **4.1 Introduction**

Shortage of resources and the need for efficiency in today's competitive world are forcing engineers to design better systems with greater economic benefits and higher efficiencies. A great challenge in engineering design is to create efficient and cost-effective systems without compromising their reliability and integrity. In a conventional design process [65], shown in Figure 4.1, information is gathered from one or more trial designs and prototypes and coupled together with the designer's experience and insight to create a more efficient alternative. The process is highly reliant on the designer's skill and experience. This reliance on a human element requires a highly skilled operator and can sometimes result in unsafe and flawed results, especially in the synthesis of complicated systems.

The key advantage of the conventional design process however, is that the designer's experience and intuition can tackle more global and conceptual problems. For example, in the MC design discussed in chapters 2 and 3, the designer can determine which GA is more suitable for which task, and can make a change based on their better understanding of the overall task of the system. They can also add, remove, and replace custom components in a custom design scenario, such as transmissions, shafts, and bearings, based on their knowledge of various aspects of the overall design like size constraints and motion objectives. With increased intelligence of the DES, some of these

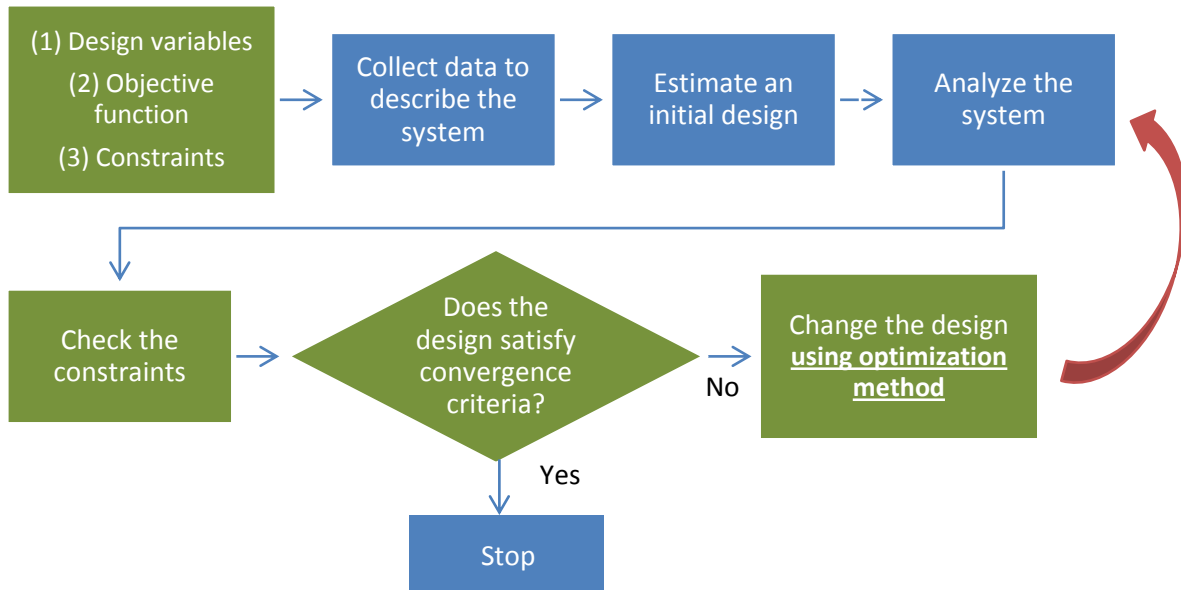


**Figure 4.1: Conventional Design Process.**

decisions can be performed without human intervention. Yet, some core inputs that must be provided by the designer will remain. The main step of the conventional design process is the detailed design. Difficulties arise when particular sections of the design need to be analyzed in detail. This analysis might include the handling of complex constraints such as limits on stress, deflection, or natural frequencies, as well as complex inputs such as multiple loading conditions. In such cases, it is difficult for the designer to rely on experience and intuition in choosing the size and properties of a particular section of the design to satisfy the constraints. Even if the constraints are satisfied, the design might actually be inefficient and uneconomical, which would defeat the purpose of the design change itself. For such design changes, the conventional design process is modified to an optimal design process, as shown in figure 4.2 (*Note: Modifications are in green*).

The optimal design process will need an explicit set of design variables, a cost function or objective function to be minimized, and the constraint equations for the

system being analyzed. It is seen that the conventional design process is less formal, since an objective function measuring the performance of the system is not present, and design changes are made based on human judgment and not calculated criteria. Nonetheless, incorporating a designer's expertise in the optimal design process would add undeniable value. The best approach would be to have an optimal design process (optimization routine) that is assisted and supported through the interaction of a human expert (high-level system design), and vice-versa.



**Figure 4.2: Optimum Design Process.**

This chapter introduces an interactive Design Optimization Add-On (DOAO) to the MCDES discussed in chapters 2 and 3. Using its practical domain-specific expert knowledge, the MCDES guides the DOAO in more feasible directions of search and optimization. The proposed interaction between the two systems, the developed program, and the results serve as a foundation for the full supervision and guidance of the MEDOS in the EDF discussed in chapter 1. First, an overview of constrained minimization and sequential quadratic programming is presented. Then the integration of the optimization

routine with the MCDES is discussed with a case study, giving graphical and numerical results.

## 4.2 Constrained Minimization

In this section, important concepts related to constrained minimization are briefly outlined. Extensive mathematical derivations and proofs are not included, but can be found in the listed references [65-67]. In the DOAO presented in this chapter, custom components from the MCDES MS are optimized for weight and cost, while satisfying the specified constraints. Constrained minimization problems may be expressed in the following standard form:

$$\begin{aligned}
 & \text{minimize } f(\mathbf{x}) \\
 & \text{subject to } g_i(\mathbf{x}) \leq 0 \quad i = 1, \dots, m \\
 & \text{and } h_j(\mathbf{x}) = 0 \quad j = 1, \dots, l
 \end{aligned} \tag{4.1}$$

where  $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$  is a column vector of  $n$  real-valued design variables,  $f$  is the objective function or cost function,  $g$ 's are inequality constraints, and  $h$ 's are equality constraints. Constraints can be linear and/or nonlinear functions of the design variables. The notation  $\mathbf{x}_0$  for starting point,  $\mathbf{x}^*$  for optimum point, and  $\mathbf{x}_k$  for the current point at  $k$ th iteration will be used. A point  $\mathbf{x}$  that satisfies the inequality and equality constraints is called feasible. Furthermore, an inequality constraint  $g_i(\mathbf{x}) \leq 0$  is said to be active at a feasible point  $\mathbf{x}$  if  $g_i(\mathbf{x}) = 0$  and inactive if  $g_i(\mathbf{x}) < 0$ . Logically, an equality constraint  $h_j(\mathbf{x}) = 0$  is active at any feasible point. Two main approaches are used to solve the standard constrained optimization problem of equation 4.1: the indirect or optimality criteria methods and the direct or search methods. In the indirect methods, such as

Karush-Kuhn-Tucker (KKT), a set of optimality criteria must be satisfied to get the minimum or optimum point. The direct or search techniques are based on starting with an estimate of the optimal design and improving that estimate iteratively until all optimality criteria are satisfied. A variety of direct or search methods are available in the literature, such as Rosen's Gradient Projection (linear constraints), Zoutendijk's Method of Feasible Directions, and the Generalized Reduced Gradient method. If the design space is constituted of only two design variables, graphical methods can effectively be used. For the purposes of this research, the Sequential Quadratic Programming (SQP) method has been chosen for the advantages mentioned in its respective section below.

#### 4.2.1 Optimality Conditions

For unconstrained minimization problems, the local minimum of the objective function at the point  $\mathbf{x}^*$  must satisfy the optimality criteria  $\nabla f(\mathbf{x}^*) = 0$ . This condition gives us the gradient vector of the objective function and is necessary for optimality. For constrained minimization however, optimality conditions must be extended to include the effects of the constraints. For problems involving equality constraint, the method of Lagrange multipliers is used [66].

##### *Method of Lagrange Multipliers:*

For an objective function  $f$  to have a stationary value, whether a minimum, maximum, or saddle point, it is required that:

$$df = \frac{\partial f}{\partial x_1} dx_1 + \cdots + \frac{\partial f}{\partial x_n} dx_n = 0 \quad (4.2)$$

with the variation of the equality constraint equation  $h(\mathbf{x}) = 0$  being:

$$dh = \frac{\partial h}{\partial x_1} dx_1 + \cdots + \frac{\partial h}{\partial x_n} dx_n = 0 \quad (4.3)$$

If there were no constraint, each  $dx_i$  would be independent and would yield the optimality condition  $\frac{\partial f}{\partial x_i} = 0$ . Due to equation 4.3, this is not the case. Equation 4.3 can be used though to eliminate one of the design variable increments (e.g.,  $dx_n$ ) by writing it in terms of the other  $dx_i$  and then substituting back into equation 4.2. However, this method is not always feasible or practical, and Lagrange developed a more general approach to reach the optimality conditions. Equation 4.3 is multiplied by an undetermined factor  $\lambda$  called the Lagrange multiplier, and added to equation 4.2, yielding:

$$\sum_{i=1}^n \left( \frac{\partial f}{\partial x_i} + \lambda \frac{\partial h}{\partial x_i} \right) dx_i = 0 \quad (4.4)$$

Then, rather than eliminating  $dx_n$ ,  $\lambda$  is chosen so that the factor multiplying  $dx_n$  vanishes and the sum in the above equation is reduced to  $n - 1$  terms. This process accounts for the restriction imposed by the constraint, making  $dx_i$  independent, thus eliminating each of the coefficients in the equation, to obtain:

$$\frac{\partial f}{\partial x_i} + \lambda \frac{\partial h}{\partial x_i} = 0, \quad i = 1, \dots, n - 1 \quad (4.5)$$

This method is generalized for a minimization subject to  $l$  equality constraints. The objective function is modified to include the constraints, resulting in the Lagrangian:

$$L = f + \sum_{j=1}^l \lambda_j h_j \quad (4.6)$$

If  $\mathbf{x}^*$  is a stationary point, then the following optimality conditions must be satisfied:

$$\frac{\partial L}{\partial x_i} = \frac{\partial f}{\partial x_i} + \sum_{j=1}^l \lambda_j \frac{\partial h_j}{\partial x_i} = 0, \quad i = 1, \dots, n \quad (4.7a)$$

$$h_j(\mathbf{x}) = 0, \quad j = 1, \dots, l \quad (4.7b)$$

### ***Inequality constraints:***

In practical optimization problems, many of the constraints are inequalities. Thus, optimality conditions for minimization subject to constraints  $g(\mathbf{x}) \leq 0$  must be discussed. A method that is used to handle inequality constraint is to change it to an equality constraint by adding a slack variable. This expands the design space due to the addition of the slack variables. Then, optimality criteria are developed as discussed before. However, it is more common to use direct search techniques for standard optimization problems with equality and inequality constraints. In these techniques, we find two vectors in the design space: a *descent direction vector* and a *feasible vector*. In any design variable space  $(x_1, x_2, \dots, x_n)$ , a vector  $\mathbf{d}$  is a descent direction at iteration point  $\mathbf{x}_k$  if:

$$(\nabla f)^T \mathbf{d} < 0 \quad (4.8)$$

where the gradient  $\nabla f$  is evaluated at  $\mathbf{x}_k$ . This condition ensures that for a finite small  $\alpha > 0$ , the inequality

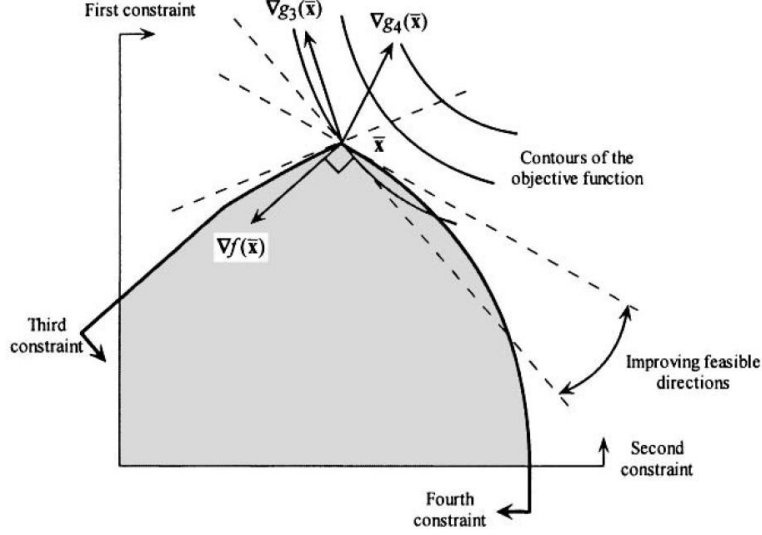
$$f(\mathbf{x}_k + \alpha \mathbf{d}) < f(\mathbf{x}_k) \quad (4.9)$$

is valid. Similarly, a vector  $\mathbf{d}$  is a feasible direction if:

$$(\nabla g_i)^T \mathbf{d} < 0 \quad \text{for each } i \in I \quad (4.10)$$

$$I = \{i: g_i(\mathbf{x}_k) = 0, \quad i = 1, \dots, m\}$$

This ensures a relationship for feasibility that is similar to equation 4.9 and defines, for each  $i \in I$ , a half-space or feasible region. The intersection of all the half-spaces from multiple inequality constraints forms the feasible cone or domain as shown in figure 4.3 [67], where the vector  $\mathbf{d}$  should lie.



**Figure 4.3: A feasible region and direction in nonlinear multi-constraint minimization.**

According to this discussion, if a direction  $\mathbf{d}$  is found that is both descent and feasible, then travelling in that direction would reduce  $f$  and satisfy the constraints. This means that  $\mathbf{x}_k$  at such an iteration is not a minimum. Therefore, for a point  $\mathbf{x}^*$  to be an optimal solution to the objective function, there should not exist any vector  $\mathbf{d}$  at the point which is both descent and feasible. Graphically, the intersection of the descent-feasible cone should be empty. This statement can be expressed using Farkas Lemma, which yields (derivation not included):

$$-(\nabla f) = \sum_{i \in I} \mu_i (\nabla g_i), \quad \mu_i \geq 0 \quad (4.11)$$

where  $\mu_i$  are Lagrange multipliers and  $g_i$  are the active inequality constraints of the problem. Equation 4.11 means that a vector  $\mathbf{d}$  cannot be both descent and feasible if  $-\nabla f$  lies in the convex cone spanned by the vectors  $\nabla g_i, i \in I$ . Inactive constraints can be included by assigning a zero Lagrange multiplier to each of them. Now, the optimality conditions taking into consideration the inequality constraints can be formulated. A point  $\mathbf{x}^*$  is a local minimum of  $f$  subject to  $g_i \leq 0, i = 1, \dots, m$  only if:

$$\left. \begin{aligned}
\frac{\partial L}{\partial x_p} &= \frac{\partial f}{\partial x_p} + \sum_{i=1}^m \mu_i \frac{\partial g_i}{\partial x_p} = 0, & p &= 1, \dots, n \\
\mu_i &\geq 0 & i &= 1, \dots, m \\
\mu_i g_i &= 0 & i &= 1, \dots, m \\
g_i(\mathbf{x}) &\leq 0 & i &= 1, \dots, m
\end{aligned} \right\} \quad (4.12)$$

***KKT Optimality conditions and sufficiency conditions:***

In light of equations 4.7 and 4.12, the optimality conditions (KKT) for equation 4.1 can be given as:

*Optimality:*  $\frac{\partial L}{\partial x_p} = \frac{\partial f}{\partial x_p} + \sum_{i=1}^m \mu_i \frac{\partial g_i}{\partial x_p} + \sum_{j=1}^l \lambda_j \frac{\partial h_j}{\partial x_p} = 0, \quad p = 1, \dots, n$

*Non-negativity:*  $\mu_i \geq 0, \quad i = 1, \dots, m$

*Complementarity:*  $\mu_i g_i = 0, \quad i = 1, \dots, m$

*Feasibility:*  $g_i \leq 0, \quad i = 1, \dots, m \quad \text{and} \quad h_j = 0, \quad j = 1, \dots, l$

If a point  $\mathbf{x}^*$  satisfies these equations, it can be a local minimum or a saddle-point.

There remains a sufficiency condition which ensures that the point is a local minimum and not a saddle point. Let

$$\nabla^2 L(\mathbf{x}^*) = \nabla^2 f(\mathbf{x}^*) + \sum_{i=1}^m \mu_i \nabla^2 g_i(\mathbf{x}^*) + \sum_{j=1}^l \lambda_j \nabla^2 h_j(\mathbf{x}^*) \quad (4.14)$$

The LHS of equation 4.14 is known as the Hessian matrix. The sufficiency condition requires that the Hessian matrix is positive definite on a subspace of  $R^n$ . In practical optimization problems, the Hessian matrix may be quite difficult to calculate and many approximate methods are used for the assessment of equation 4.2 and the approximation of the Hessian matrix. Further discussion of this condition is beyond the scope of this thesis.

### 4.2.2 Sequential Quadratic Programming

Sequential quadratic programming (SQP) methods are a class of gradient-based methods of optimization which offer a superior rate of convergence over other methods [66, 67]. In the case study examples presented later in this chapter, SQP was the optimization method of choice. In addition to a better convergence rate, SQP methods have several other advantages:

- Infeasible starting points may be used
- Method can be *proved* to converge under certain conditions
- Equality and inequality constraints can be handled
- Gradients of only active constraints are needed

SQP methods may be interpreted as Newton's methods as applied to the solution of the KKT conditions which were discussed previously. For the equality constrained portion of equation 4.1, subject to equation 4.7 optimality conditions, the problem is simplified into:

$$\begin{bmatrix} \mathbf{W} & \mathbf{A}^T \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \Delta \mathbf{x} \\ \Delta \lambda \end{Bmatrix} = \begin{Bmatrix} -\nabla f - \mathbf{A}^T \lambda \\ -h \end{Bmatrix} \quad (4.15)$$

where:

$$\mathbf{W} = \frac{\partial^2 f}{\partial \mathbf{x}^2} + \lambda^T \frac{\partial^2 h}{\partial \mathbf{x}^2} \quad \text{and} \quad \mathbf{A} = \frac{\partial h}{\partial \mathbf{x}} \quad (4.16)$$

The direction vector at a given iteration point  $\mathbf{x}_k$  is determined by solving the Quadratic Programming (QP) sub-problem (with both inequality and equality constraints):

$$\begin{aligned} &\text{minimize} && \frac{1}{2} \mathbf{d}^T \mathbf{d} + \nabla f^T \mathbf{d} \\ &\text{subject to} && (\nabla g_i)^T \mathbf{d} + g_i^k \leq 0 \\ &&& (\nabla h_i)^T \mathbf{d} + h^k = 0 \end{aligned} \quad (4.17)$$

$$\mathbf{x}^L \leq \mathbf{d} + \mathbf{x}_k \leq \mathbf{x}^U$$

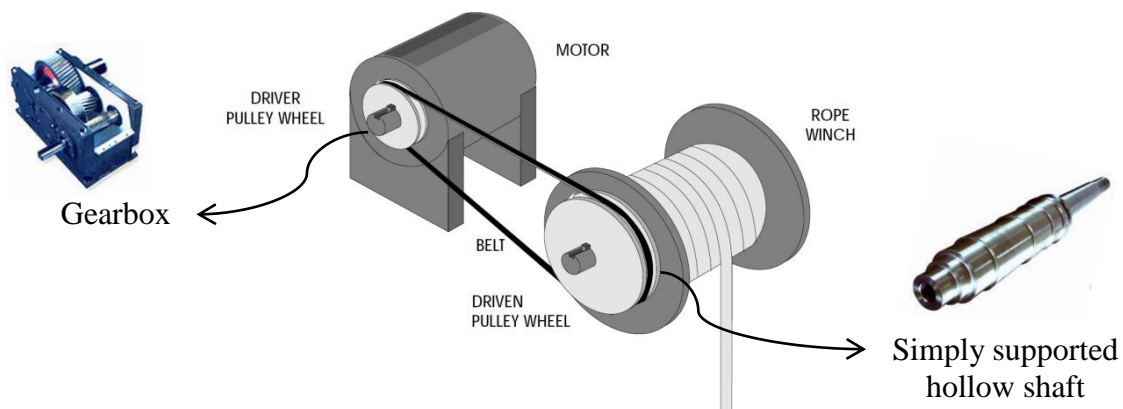
where  $d_i$ ,  $i = 1, \dots, n$  are increments of the design variables in this QP, the solution is denoted as  $\mathbf{d}_k$ , and the gradient vectors are all evaluated at  $\mathbf{x}_k$ . If  $\mathbf{d}_k = \mathbf{0}$ , it can be shown that  $\mathbf{x}_k$  is a KKT point. Also, the iterative process is terminated if there is no change in  $f$  for three consecutive iterations. Constraint violations must be checked since SQP iterates through infeasible regions. The solution algorithm used in the case study presented later in this chapter follows the following four steps:

1. A starting point  $\mathbf{x}_0$  is chosen to satisfy the bounds  $\mathbf{x}^L \leq \mathbf{x}_0 \leq \mathbf{x}^U$ . Choose scalars  $\delta$ ,  $R$ , and  $\gamma$  (typically, 0.05, 100, and 0.1, respectively).
2. Determine the active set, evaluate gradients of *active* constraints, and obtain  $\mathbf{d}_k$  by solving the QP of equation 4.17.
3. Check for convergence criteria.
4. If convergence criteria not met, determine step size  $\alpha_k$  and update design point to  $\mathbf{x}_{k+1} = \alpha_k \mathbf{d}_k$ . Set  $\mathbf{x}_{k+1} = \mathbf{x}_k$  and return to step 2.

### 4.3 Case Study and MCDES Integration

To show the function of the interactive design optimization add-on (DOAO) as developed for our MCDES, a design scenario is set up and selected custom components are optimized. The first GA of our MCDES is a vertical hoisting mechanism application, which can be attached to the motor rotor directly (the pulley of the hoist is fitted onto the motor rotor). However, the MCDES allows the user to build a custom design comprising multiple mechanical components before the hoist pulley. The high-level structure of the custom design is determined by the human user; however, the detailed design analysis of

the individual components is done by the DOAO. For this case study, we assume that the human user builds the high-level custom design shown in figure 4.4. Aside from the belt pulley transmission, the user wishes to add a hollow shaft simply supported by two bearing housings between the driven pulley wheel and the rope winch (to reach loads farther away from the belt-pulley plane). A gearbox between the motor and the driver pulley wheel is also needed for further speed reduction. The role of the DOAO in this case study is to minimize the weight, complexity, and consequently, the cost of each of the modification components (or the original design, the belt-pulley transmission in this case) without sacrificing the structural or functional integrity of the system. The required parameters for this optimization are taken from the MCDES GUI, and the optimized design parameters are returned to the MCDES automatically for motor selection. In other words, when the user builds the high-level design in the MCDES GUI, the DOAO will run an optimization routine which will give the user the option of safely choosing more optimized components (reduced size, weight, cost, etc.), then, the MCDES will use the optimized components in the motor selection routine.



**Figure 4.4: High-level custom hoister design with modification components.**



**Table 4.1: Sources of required parameters of optimization.**

<b>Hollow shaft in hoisting application</b> Design variables: inner radius $r_i$ and outer radius $r_o$	
<b>From MCDES</b>	<b>Coded into DOAO</b>
Mass of load	Objective function (w.r.t. design variables)
Material (with corresponding tensile strength, shear strength, Young's modulus and shear Modulus, and Poisson's ratio)	Stress level, deflection, and twist constraint equations (w.r.t. design variables)
Length of shaft	Practical numerical ranges of design variables
Winch pulley radius	Calculation of maximum static torque
Starting design variable values (non-optimized and may be infeasible since SQP methods are used)	Calculation of maximum moment (simply supported)
\	Practical numerical values of constraint limitations (e.g., maximum allowable angular twist of shaft)

## 4.4 Problem Formulation and Results

In this section, the formulation of the important equations coded into the DOAO, such as the objective function and the constraints, is demonstrated for each case. The numerical and graphical (if applicable) optimization results are presented for select cases, and their implication on the size of the motor selection is discussed.

### 4.4.1 Hollow shaft under static torque

#### *Problem Formulation:*

If the hoisted load is static, the hollow shaft connecting the driven side pulley of the belt pulley system and the rope winch is under constant torque. The torque depends on the radius of the winch pulley and the weight of the hoisted load. If the shaft is simply supported, the vertical force is balanced by the bearing reactions. The weight of the shaft

itself is not considered in this case study (it will be considered in section 4.4.2). Since the side of the shaft that is connected to the belt-pulley transmission is fixed (the weight of the load alone should not rotate the pulleys and motor rotor) and the winch pulley is applying a constant torque, the problem is simplified to a hollow shaft cantilever under torsion (case 1). The weight of the shaft is to be minimized, which would also minimize the material cost. The objective function is

$$\text{minimize } f = \frac{\pi}{4}(d_o^4 - d_i^4)L\rho g \quad (4.17)$$

There are three constraints limiting the reduction of the design variables ( $d_o$  and  $d_i$ ) beyond structural limits, as given below.

*Stress:*

$$\tau_{max} = \frac{Tr_o}{J} \quad \text{or} \quad \frac{T\left(\frac{d_o}{2}\right)}{\pi\left(\frac{d_o^4 - d_i^4}{32}\right)} - \tau_{allow} \leq 0 \quad (4.18)$$

*Twist:*

$$\theta = \frac{T}{GJ}L \quad \text{or} \quad \frac{32 \cdot T \cdot L}{G \cdot \pi(d_o^4 - d_i^4)} - \theta_{allow} \leq 0 \quad (4.19)$$

*Buckling:*

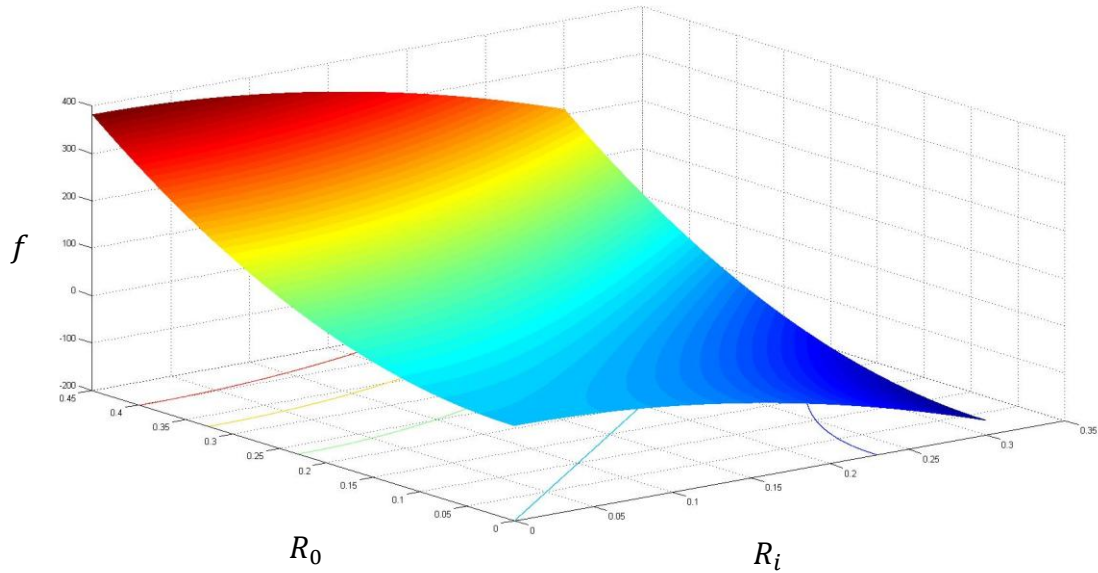
$$T \leq (T_{cr})_{buckling} \quad \text{or} \quad T - \frac{\pi d_o^2 E}{12\sqrt{2}(1 - \nu^2)^{0.75}} \left(1 - \frac{d_i}{d_o}\right)^{2.5} \leq 0 \quad (4.20)$$

The limits on the design variables are established based on human judgment and expertise in the DOAO. To avoid getting a thin-walled shaft as the solution, a constraint is placed on how close the inner diameter of the shaft can be to the outer diameter

$$\begin{aligned} 1 \text{ cm} &= d_{i_{min}} \leq d_i \leq d_{i_{max}} = 6 \text{ cm} \\ 2 \text{ cm} &= d_{o_{min}} \leq d_o \leq d_{o_{max}} = 7 \text{ cm} \\ d_i &\leq 0.9d_o \end{aligned} \quad (4.21)$$

### **Results:**

Since there are only two design variables in this case, a graphical solution is possible. The three-dimensional plot of the weight as a function of the design variables is given in figure 4.6. The only practical region of the plot is the left half, where the outer radius is larger than the inner radius. Without the constraints however, this plot cannot show the

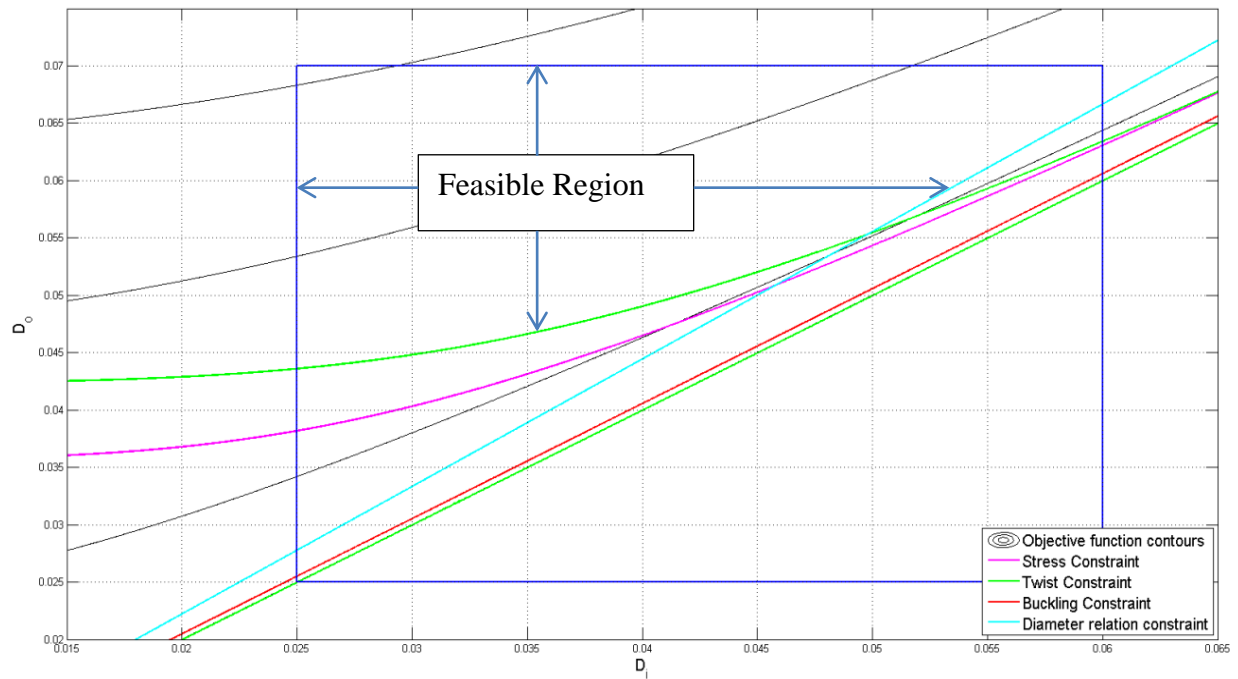


**Figure 4.6: 3D plot of the objective function for Case 1.**

feasible region where an optimal solution may occur. Figure 4.7 shows a plot of the projected contours of the objective function contours along with the constraints developed for this case. It should be noted that the axes are diameters and that the blue rectangle border represents the limits or range of the design variables (upper and lower limits for each). The necessary parameters given in Table 4.2 were taken from the MCDES GUI or inferred based on the choices by the user. The allowable twist of the shaft was set at  $\theta = 1^\circ$ .

**Table 4.2: Parameter values for Figure 4.7.**

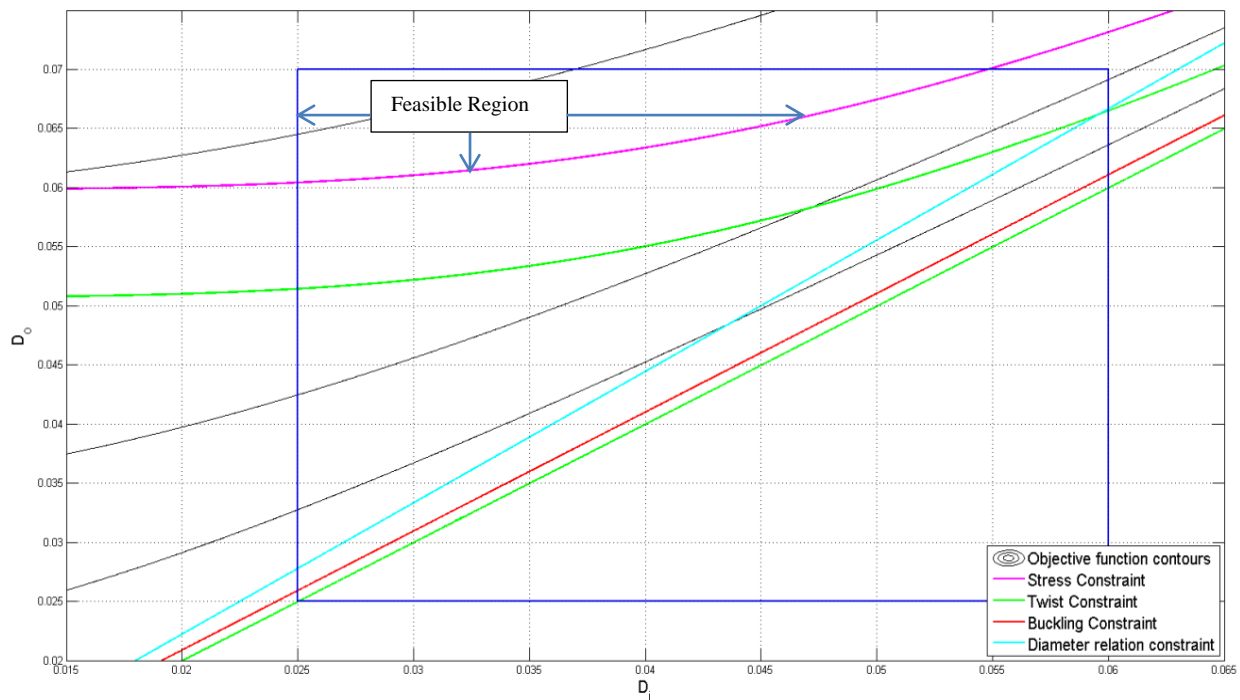
Parameter	Value
Length of shaft $L$	0.3 m
Mass of load $M$	1000 kg
Radius of winch pulley $R$	0.15 m
Shaft material: 4140 Alloy Steel	
Material density $\rho$	7850 kg/m <sup>3</sup>
Allowable shear strength $\tau_{allow}$	275 MPa
Young's Modulus $E$	210 GPa
Shear Modulus $G$	80 GPa
Poisson's Ratio $\nu$	0.3



**Figure 4.7: Objective function contour and constraint plot for Case 1 using 4140 Alloy Steel.**

It can be seen that the limiting factor on the design when using 4140 Alloy Steel and the parameters in Table 4.2 is the twist constraint and sometimes the diameter relation constraint (equation 4.21). Any design variable choice below the top twist

constraint, the stress constraint, or the diameter relation constraint violates the structural integrity of the shaft. The feasible region is between the top twist constraint, the diameter relation constraint, and the top and left design variable range constraints. To illustrate the importance of the DOAO, the material choice was switched from 4140 Alloy Steel to Magnesium Alloy A261 (which changes all the inferred parameters), the mass was increased to 1500 kg, and the allowable twist of the shaft (which ultimately determines the accuracy of the design in positioning the load) was increased slightly to  $\theta = 1.5^\circ$ . With these relatively simple design adjustments, the active constraints are overwhelmingly different than before. Figure 4.8 demonstrates these changes.



**Figure 4.8: Objective function contour and constraint plot for Case 1 using Magnesium A261 Alloy.**

In figure 4.8, the limiting factor on the design throughout the design area is the stress constraint. In addition, the feasible region is significantly smaller than in figure 4.7. It would be very difficult and time consuming for a designer to determine such changes

in constraints and determine optimum design variable values for any simple change in the details of the design. For more complicated constraints and objective functions, such a process would be almost impossible. That being said, although the graphical representations presented above can be useful for a human designer, a single optimal solution cannot be easily identified. A feasible solution can be chosen within the feasible region; however, the optimum solution requires some extra analysis. This extra graphical and numerical analysis cannot be afforded in the integrated system of figure 4.5. For this reason, numerical optimization routines were used for the integrated system.

The SQP method was used for the advantages listed in section 4.2.2. To initiate the optimization process, a starting point for the design variables must be identified. Since this starting point is not the optimized value and may be infeasible (if SQP method is used), the values are taken from the MCDES. The following tables show the results obtained from running simulations with varied design specifications (such as material choice) from the MCDES. The starting points for the routine were randomly chosen to simulate a real design scenario where the user does not know the constraints and design details. It should be noted that the stopping criteria used (equation 4.17) only ensures that a local minimum is reached. Thus, depending on the starting point, the final solution might or might not correspond to the global minimum. Nevertheless, an improvement will often be made on the starting point, and numerical results for the design variables that do not violate the constraints will be found. These can be automatically fed into the MCDES for motor selection, and the impact the component optimization has on the torque requirements is evident.

**Table 4.3: Parameter values for results of Table 4.4.**

Parameter	Value
$L, M, R$	0.3 m, 1500 kg, 0.15 m
$\rho, \tau_{allow}, E, G, \nu$	Inferred from material choice
$[d_{i-min}, d_{o-min}, d_{i-max}, d_{o-max}]$	[0.02, 0.025, 0.1, 0.15]
$d_i \leq 0.75d_o$	

**Table 4.4: Case 1 optimum solutions vs. shaft material & impact on required  $T_{motor}$ .**

Starting Point $[d_{i1}, d_{o1}]$	Material	Optimum Solution $[d_{i*}, d_{o*}]$	$T_{motor}$ change (all else constant)
[0.06, 0.07]	4140 Alloy Steel	<b>[0.035, 0.047]</b>	-13.4%
[0.06, 0.07]	Aluminum Alloy 24 ST4	<b>[0.05, 0.067]</b>	+1.96%
[0.06, 0.07]	Magnesium Alloy A261	<b>[0.058, 0.077]</b>	+5.69%
[0.06, 0.07]	Beryllium	<b>[0.047, 0.063]</b>	~ 0%
[0.06, 0.07]	Titanium	<b>[0.045, 0.061]</b>	-2.98%

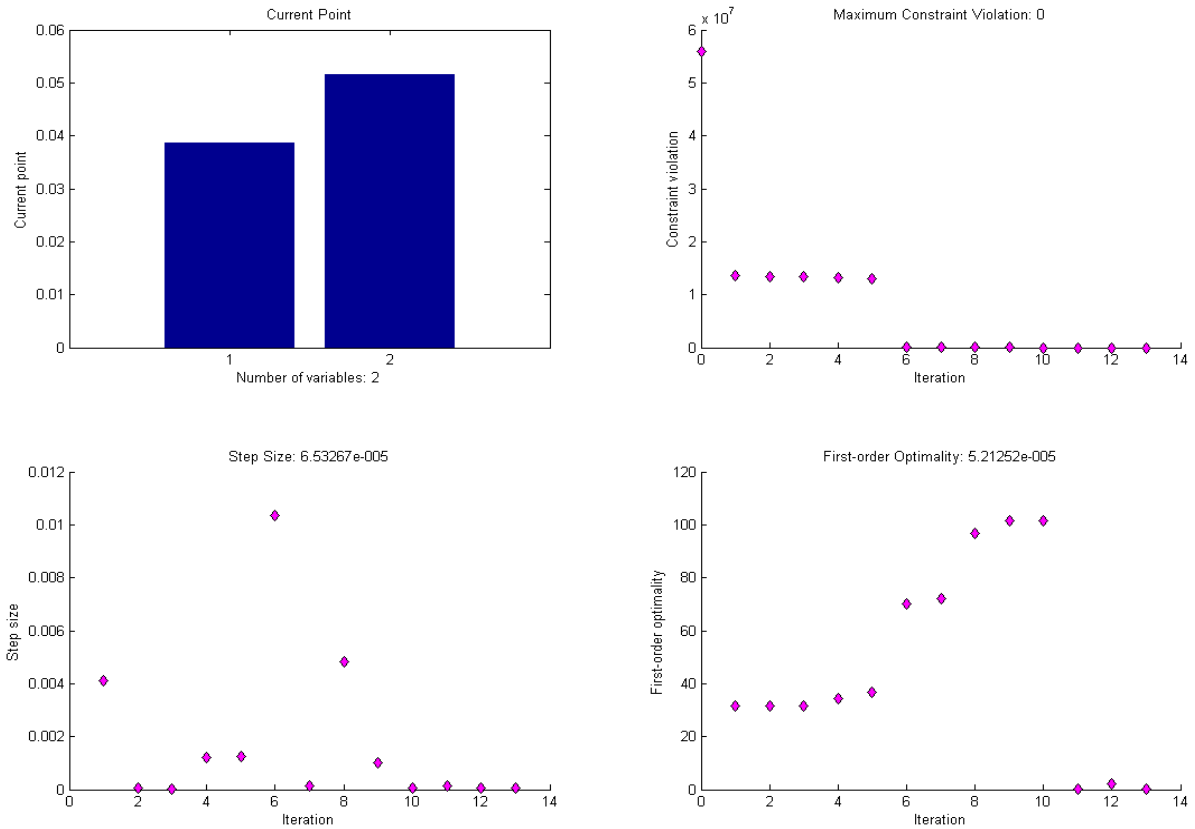
In table 4.5, the material is kept the same (4140 Alloy Steel) and the core design parameters ( $L, M, R$ ) are increased at varied rates. The starting point values are increased at a steady rate. This is because the user knows that larger diameters will be needed if the core design parameters are increased. The optimal solution for each case is simulated. The difference between the starting points determined by human judgment and the optimized points from the simulation is more than 50%. The effect of this reduction in size on the required motor torque is found through integration with the MCDES.

Figure 4.9 shows graphs of key optimization performance indicators versus iterations during an optimization routine, such as constraint violations and step size. For each iteration, the values of the design variables are displayed. The stopping criterion of the algorithm depends on the number of constraint violations, the step size of the upcoming iteration, and the optimality conditions. As seen from iterations 6 to 11, even if

there are no constraint violations, the optimal solution is not identified if the step size and the optimality conditions are larger than a determined threshold. On a final note, the excellent convergence rate of the SQP method is evident from figure 4.9 and is an important feature of the DOAO.

**Table 4.5: Case 1 optimum solutions vs. human judgment and impact on required  $T_{motor}$ .**

Starting Point $[d_{i1}, d_{o1}]$	Core design parameters $[L, M, R]$	Optimal Solution $[d_{i*}, d_{o*}]$	$T_{motor}$ change (all else constant)
[0.05, 0.08]	[0.1, 500, 0.05]	<b>[0.02, 0.027]</b>	-11.9%
[0.06, 0.09]	[0.15, 750, 0.1]	<b>[0.025, 0.033]</b>	-4.86%
[0.07, 0.1]	[0.175, 980, 0.105]	<b>[0.028, 0.037]</b>	-5.63%
[0.08, 0.11]	[0.18, 1500, 0.145]	<b>[0.044, 0.058]</b>	-2.68%
[0.09, 0.12]	[0.3, 2500, 0.16]	<b>[0.046, 0.061]</b>	-2.96%



**Figure 4.9: Sample screenshot of Case 1 numerical optimization.**

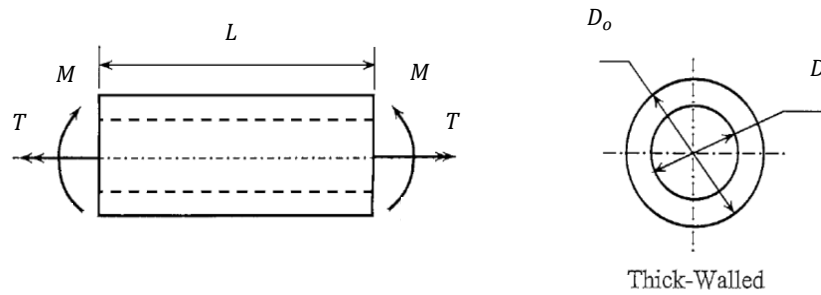
#### 4.4.2 Hollow shaft with combined loading

##### ***Problem Formulation:***

In case 1, the hollow shaft connecting the driven side pulley of the belt pulley system and the rope winch was subjected to a constant torque only. The weight of the shaft itself was not considered. In reality, the weight of the shaft is important and should be accounted, especially for large shafts used in heavy-duty applications. In case 2, the objective function of the problem remains the same, but a constraint which accounts for the combined loading effect is added. The shaft is still simply supported on bearing housings, so the vertical force from the hoisted load is supported by the nearest bearing and is not felt by the shaft. It is assumed that the nearest bearing is sufficiently close to the hoisted load application point so that it takes the majority of the load. The winch pulley is still applying a constant torque to the shaft, but there is also an added constant distributed load along the length of the shaft, representing the weight of the shaft. This creates a transverse shear and a moment in the shaft as a function of position  $x$ .

$$V = \frac{F_{weight}}{L}(L - x) \quad \& \quad M = \frac{\frac{F_{weight}}{L}x}{2}(L - x) \quad (4.22 \ \& \ 4.23)$$

Assuming a uniform cross-sectional area, the maximum moment occurs at half-length and is given by  $M = \frac{F_{weight} \cdot L}{8}$ . The constraints are formulated on this basis, as shown in figure 4.10.



**Figure 4.10: Problem basis of Case 2 optimization.**

In the constraints, the maximum normal stress in the shaft due to bending moment and the maximum shear stress due to torque are:

$$\sigma = \frac{32MD_o}{\pi(D_o^4 - D_i^4)} \quad \& \quad \tau = \frac{16TD_o}{\pi(D_o^4 - D_i^4)} \quad (4.24 \& 4.25)$$

The principal stress for this 2D stress state is found and the Tresca failure criterion is applied to find the nonlinear inequality constraint (equation 4.27) as a function of the design variables ( $D_i, D_o$ ).

$$\sigma_{PP} = \frac{\sigma}{2} \pm \sqrt{\left(\frac{\sigma}{2}\right)^2 + \tau^2} = \frac{16MD_o}{\pi(D_o^4 - D_i^4)} \pm \frac{1}{2} \left[ \left( \frac{32MD_o}{\pi(D_o^4 - D_i^4)} \right)^2 + 4 \left( \frac{16TD_o}{\pi(D_o^4 - D_i^4)} \right)^2 \right]^{\frac{1}{2}} \quad (4.26)$$

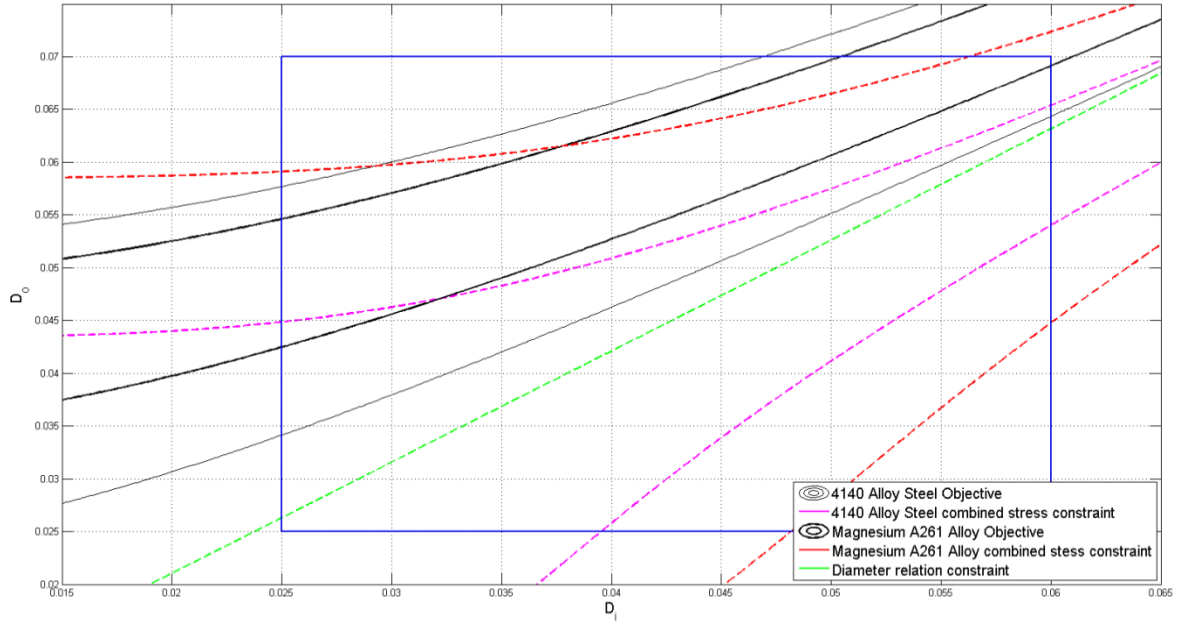
*Tresca:*

$$\sigma_{11} - \sigma_{33} \leq \sigma_{allow} \quad \therefore \left[ \left( \frac{32MD_o}{\pi(D_o^4 - D_i^4)} \right)^2 + 4 \left( \frac{16TD_o}{\pi(D_o^4 - D_i^4)} \right)^2 \right]^{\frac{1}{2}} - \sigma_{allow} \leq 0 \quad (4.27)$$

The limits on the design variables and the dimension relation constraint are the same as in equation 4.21.

### **Results:**

The number of design variables has not increased, so a graphical representation is still possible. Figure 4.11 gives a plot of the weight contours and the new combined loading stress constraint for case 2. It should be noted that in this comparison graph, two objective function contours are plotted on the same plane. This is because the objective function is not only dependent on the design variables; in this case, it changes for different materials due to the change in density.



**Figure 4.11: Graphical comparison of 4140 Alloy Steel and Magnesium A261 Alloy in Case 2 optimization.**

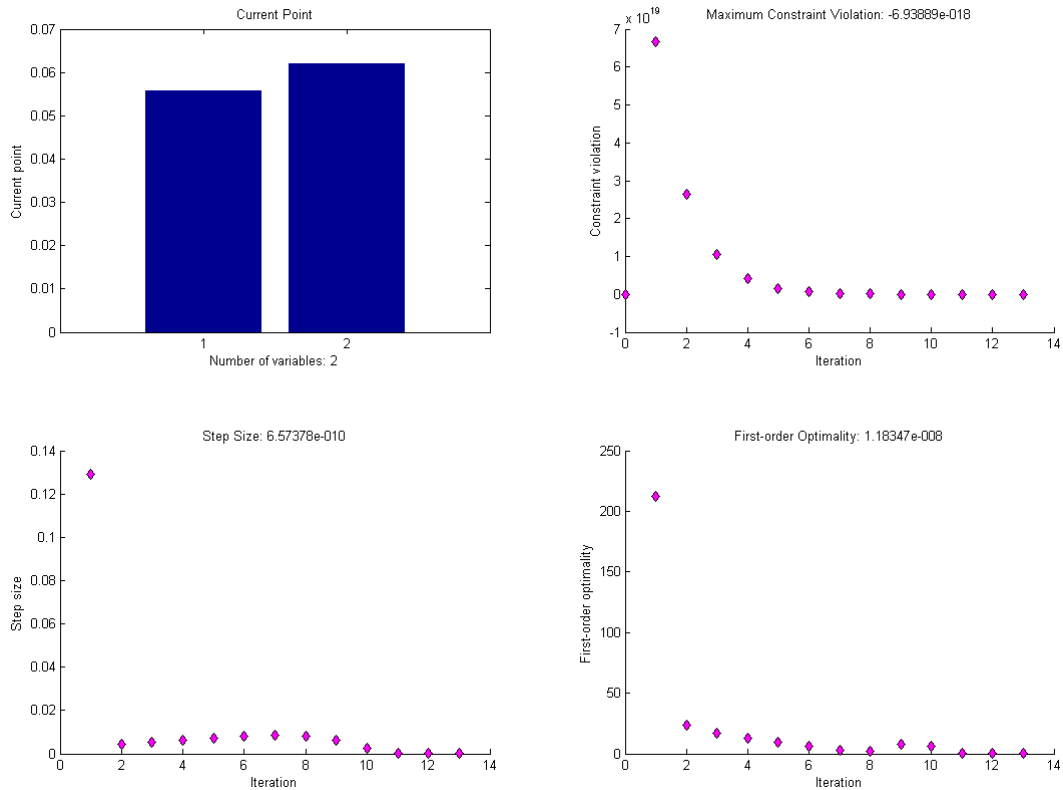
Numerical results were also obtained for case 2, and are presented in table 4.6. Here, the starting point was chosen randomly (infeasible to simulate worst case scenario) and kept constant. The change in optimal solution based on the material choice is observed, along with the effect the optimization has on the required  $T_{motor}$ . Compared to a conventional design process, the interactive MCDES with the DOAO is significantly faster in evaluating design changes. Also, it is fully autonomous from the moment the user builds the high-level structure and chooses a starting point for the design variables.

Further simulations can be performed for different parameter changes. Many times during the analysis of the simulation results, it was observed that the values for the optimal solution depended on the initial values. That is, the algorithm will give back the feasible local minimum of the objective function nearest to the initial point. Further work on this algorithm may attempt to record all local minimum results for a range of initial points and return the least value. In many cases, this can further optimize the values and

provide even better torque reduction and cost benefits. Another possible avenue for tackling this problem will be to build extra heuristic rules into the DOAO which can determine the most promising initial point which the existing routine can start at.

**Table 4.6: Case 2 optimum solutions vs. shaft material & impact on required  $T_{motor}$ .**

Starting Point [ $d_{i1}$ , $d_{o1}$ ]	Material	Optimum Solution [ $d_{i*}$ , $d_{o*}$ ]	$T_{motor}$ change (all else constant)
[0.06, 0.07]	4140 Alloy Steel	[0.056, 0.062]	-11.5%
[0.06, 0.07]	Aluminum Alloy 24 ST4	[0.052, 0.06]	-3.79%
[0.06, 0.07]	Magnesium Alloy A261	[0.058, 0.07]	+0.78%
[0.06, 0.07]	Beryllium	[0.052, 0.06]	-2.62%
[0.06, 0.07]	Titanium	[0.046, 0.051]	-9.59%

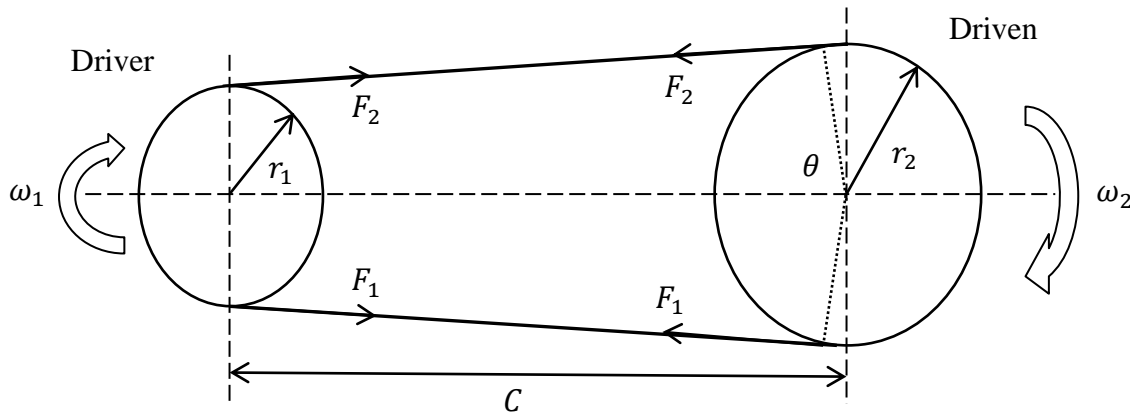


**Figure 4.12: Sample screenshot of case 2 numerical optimization.**

### 4.4.3 Belt Pulley Drive

#### *Problem Formulation:*

In the high-level design structure shown in figure 4.4, a belt pulley drive (BPD) connects the motor rotor (or gearbox) to the hollow shaft optimized in the two previous sections. The design of the BPD is essential to the whole system, and minimization of its weight does not only reduce its individual cost but also reduces the size of the motor and thus the overall cost of the system. In order for the DOAO to optimize the BPD, it must contain



**Figure 4.13: Geometric and design parameters for flat BPD design and optimization.**

knowledge of proper design procedures and the necessary constraints. In BPD design, many parameters are empirical and can be highly interdependent. Thus, some design constraints are modified to reduce difficulties in the optimization. For example, some equality constraints are converted to inequality constraints within small ranges, due to the difficulties equality constraints create in the optimization. These modifications do not significantly affect the overall performance of the design.

**Table 4.7: Case 3 geometric and design parameters.**

$r_1, r_2$	Radii of pulleys
$\omega_1, \omega_2$	Angular velocities of pulleys
$P, F_1, F_2$	Horsepower and belt tensions
$\theta, C$	Angle of contact and distance between centers
$T_1, T_2$	Torques at driver and driven pulley
$F_c$	Centrifugal force and coefficient of friction
$t, b, \gamma$	Belt thickness, width (into page), and specific weight
$\mu, f'$	Coefficient of friction and friction development

The design variables for optimization are  $r_1, r_2, F_1, F_2$  and  $C$ . Other parameters in Table 4.7 such as  $\theta$  are excluded due to their dependency on one or more of the chosen design variables (they should be as independent as possible). The immediate objective of the optimization is to minimize the weight of BPD

$$\text{minimize } f = w_1 tb + w_2(r_1 + r_2) \quad (4.28)$$

where  $w_1$  and  $w_2$  are scalar weights which can be excluded, especially since the belt weight is not affected by the design parameters. The main assumption here is that the overall weight of the BPD is a linear function of the individual pulley geometries. The constraints which ensure the minimization of the pulleys does not violate the design integrity are:

*Speed ratio:*

$$\frac{r_1}{r_2} - \frac{\omega_2}{\omega_1} = 0 \quad (4.29)$$

*Power:*

$$(F_1 - F_2)r_1(2\pi\omega_1) \geq P \quad (4.30)$$

*Belt strength:*

$$\left(\frac{F_1}{bt}\right) - \sigma_0 \leq 0 \quad \text{or} \quad F_1 - F_{allow} \leq 0 \quad (4.31)$$

*Balance of belt forces:*

$$\left( \frac{F_1 - \left( \frac{w}{g} \omega_1^2 \right) r_1^2}{F_2 - \left( \frac{w}{g} \omega_1^2 \right) r_1^2} \right) - e^{\mu\theta} = 0 \quad (4.32a)$$

$$\theta = \pi - 2\sin^{-1} \left( \frac{r_2 - r_1}{C} \right) \quad (4.32b)$$

$$w = 12\gamma bt \quad (4.32c)$$

*Central Distance:*

$$\max\{(3r_1 + r_2), 2r_2\} - c \leq 0 \quad (4.33)$$

*Friction:*

$$f' \leq \mu \quad (4.34a)$$

$$\frac{1}{\theta} \ln \left( \frac{(F_1)_a - F_c}{F_2 - F_c} \right) \leq \mu \quad (4.34b)$$

$$F_c = \frac{w}{g} \left( \frac{\pi r_1 \omega_1}{720} \right)^2 \quad (4.34c)$$

In the optimization routine, Equation 4.32a is converted into two inequality constraints which allow for a small region of feasibility, rather than a strict equality. Otherwise, it has been observed that the optimization might stall or fail to return a feasible solution. Finally, the range or side constraints for each of the design variables must be set. For the belt tensions, this range depends on the chosen belt material and size. For the driving pulley, the lower bound also depends on the belt material and size (found in design tables). The upper bound of the driving pulley and both bounds of the remaining two design parameters are not limited by anything except that they must be nonzero and must satisfy the applicable constraints given by equations 4.33 and 4.29.

**Results:**

For this case, there are five design variables. Hence a graphical representation of the objective function contours and the constraints is not possible. The objective function and all the constraints were used in the numerical optimization routine. The source of the necessary design parameters such as the input angular velocity  $\omega_1$  and the required speed ratio were taken from the MCDES GUI. Some of these values were inferred based on the belt material and size chosen by the user. An example of this is shown in table 4.8.

**Table 4.8: Parameter values for BPD optimization.**

Parameter	Value
Input angular velocity $\omega_1$	1750 rpm
Output angular velocity $\omega_2$	583 rpm
Speed ratio (input : output)	3:1
Belt width $b$	6 in
Power	1 hp
Belt: Polyamide A-2	
Specific weight $\gamma$	0.037 lbf/in <sup>3</sup>
Allowable tension per unit width $F_{allow}$	60 lbf/in
Coefficient of friction $\mu$	0.4
Belt thickness $t$	0.11 in
Minimum driving pulley diameter $d_{1min}$	2.4 in

The ranges of the design variables or side constraints are once again defined beforehand in the optimization routine. These can be changed by the user, depending on the application. For the results presented later in this section, the side constraints were:

$$d_{1min}(\text{Belt spec. dependent}) \leq d_1 \leq d_{1max} = 20 \text{ in}$$

$$2 \text{ in} = d_{2min} \leq d_2 \leq d_{2max} = 60 \text{ in}$$

$$100 \text{ lbf} = F_{min} \leq F_1, F_2 \leq F_{max} = 2000 \text{ lbf}$$

$$5 \text{ in} = C_{min} \leq C \leq C_{max} = 100 \text{ in}$$

Numerical results were obtained for the BPD optimization with the constraints and formulation presented above. The starting point was chosen randomly (infeasible to simulate worst case scenario) and kept constant:

$$[d_1, d_2, F_1, F_2, C] = [12 \text{ in}, 36 \text{ in}, 500 \text{ lbf}, 400 \text{ lbf}, 40 \text{ in}]$$

The change in the optimal solution based on the chosen belt (material and specification) is observed, along with the effect the optimization has on the required  $T_{motor}$ . The effect on the motor torque is determined by running the MCDES with the starting point design and comparing the required torque with its counterpart after the optimization and MCDES is run. The optimization using the DOAO and the consequent selection of the motor using the MCDES are found to be significantly faster.

**Table 4.9: Case 3 optimum solutions vs. belt specification & impact on  $T_{motor}$ .**

Material	Optimum Solution (rounded) [ $d_1^*$ , $d_2^*$ , $F_1^*$ , $F_2^*$ , $C^*$ ]	$T_{motor}$ change (all else constant)
Polyamide F1	[1.2, 3.6, 210, 110, 4]	-14.0%
Polyamide F2	[2.5, 7.5, 360, 315, 7.9]	-10.4%
Polyamide A2	[2.45, 7.35, 360, 318, 7.8]	-10.6%
Polyamide A3	[4.5, 13.5, 600, 577, 13.9]	-8.7%
Polyamide A4	[10, 30, 1050, 1040, 30]	-3.8%
Polyamide A5	[14, 42, 1650, 1643, 42]	+11.6%
Leather 2-ply $\frac{23}{64}$	[9.3, 27.9, 360, 350, 28.4]	-4.8%
Leather 2-ply $\frac{20}{64}$	[6.2, 18.6, 300, 284, 19.3]	-6.6%
Leather 2-ply $\frac{18}{64}$	[4.7, 14.1, 246, 223, 14.6]	-8.4%
Leather 1-ply $\frac{13}{64}$	[4, 12, 198, 173, 12.6]	-9.1%
Leather 1-ply $\frac{11}{64}$	[3.1, 9.3, 180, 147, 9.8]	-10.0%

## **Chapter 5**

### **Conclusions**

#### **5.1 Synopsis and Contributions**

This thesis developed a Design Expert System (DES) which is able to guide the evolution of optimal design alternatives for a motion control (MC) system and assess the feasibility of these alternatives. A comprehensive knowledge base (KB) for the DES was generated by using principles and procedures of MC design and motor selection and representing them in a proper if-then rule format. A systematic motor selection methodology which can incorporate generic or custom mechanical structures together with generic or custom duty cycles of the load was proposed and incorporated into the DES.

Practical design features such as s-curve profiling, geometric trajectory blending, and motor thermal response were systematically integrated into the DES. A knowledge acquisition (KA) facility for the DES was created to allow for automatic retrieval of specifications for commercially available motors and gears from an external database which can be easily updated and modified. Implementation of all the DES features in a coherent program with a convenient graphic user interface (GUI) was performed using a commercial software package. The DES results were verified against examples of human designer results and the specifications of an existing practical engineering setup. It was found that the DES offered significant benefits in time reduction, completeness, and decision explanation, while the results obtained through it were within 10% of the compared values.

The DES was built for smooth integration within a larger Evolutionary Design Framework (EDF). In the EDF, along with other prescribed duties, the DES is intended to guide the automated evolution process based on its built-in heuristic rules and design expertise. A model guidance process was proposed to achieve this specific task. Using a Sequential Quadratic Programming (SQP) optimization technique, a pilot version of this process has been developed. In this development, the MC design parameters are optimized using practical design constraints and criteria supplied by the DES. The optimized data are then used by the DES for further design analysis and actuator selection. The entire process was found to flow seamlessly between the developed systems. The performance of the developed guidance process was studied using a practical design case, and benefits of the developed process were discussed. The influence the guided optimization has on the final motor-gear selection was quantified.

The main contributions of this work may be summarized as follows:

- Development of an extensive computer KB related to MC system design and motor selection by incorporating practical and applied design principles.
- Successful implementation of technologies of design analysis and motor selection, with the developed KB, in a DES complete with a KA scheme, explanation facility, and GUI.
- Creation of a guided design evolution system using a SQP routine in Matlab® and the developed DES, where SQP optimization is steered using practical design constraints and knowledge supplied by the DES.
- Verification of the DES using hypothetical and actual design cases, and of the guided design evolution system using a practical design case-study.

## 5.2 Significance

The contributions of this research are useful in the area of automated engineering design, optimization, evaluation, and industrial automation at large. The benefits of the contributions of the thesis are highlighted below:

- Increasing the breadth or coverage of digital rule-based knowledge to include a vital new domain: MC design and actuator selection.
- Using computer-based systems instead of human experts to guide and set up iterative optimization processes and assess the practicality and real-world feasibility of the results.
- Facilitating the development of fully autonomous engineering design and redesign methods, where existing engineering systems are systematically monitored for faults or design weaknesses, diagnosed, and redesigned according to feasible alternatives.

## 5.3 Suggestions for Future Work

The creation of efficient and robust automated design and optimization systems is a complex undertaking which requires continuous investigation and research. The following points are suggested as directions and areas for future research and development:

- Using the DES to assist the MHMS in identifying design weaknesses (not only faults) in the monitored machine. For this task, the DES must be updated with

practical knowledge of the trends and indicators observed by human experts to recognize design deficiencies. Such knowledge must be acquired for a range of design domains (e.g., structural or electrical) for the system to be beneficial.

- Improvement of the interaction between the DES and the MEDOS by:
  - Developing the GP further to output more concrete design parameters such as specific component sizes or weights.
  - Implementing an autonomous method of translating a proposed design alternative by the MEDOS from BG or LG models to parameters that the DES can handle.
- Expanding the KB of the DES to other domains such as hydraulic system design and increasing the breadth of the current DES for the selection of other actuators such as hydraulic motors or cylinders and transmissions such as fluid transmissions and continuously variable transmissions (CVT).
- Additional experimentation, verification, and validation on existing engineering designs with the main purpose of smoothening collaboration between the components of the EDF, which were developed separately.

## References

- [1] Budynas, R. G., and Nisbett, J. K., *Shigley's Mechanical Engineering Design – 8<sup>th</sup> ed.* 2008, New York: McGraw Hill.
- [2] Ulrich, K., and Eppinger, S., *Product Design and Development – 4<sup>th</sup> ed.* 2003, New York: McGraw Hill.
- [3] Ullman, J. D., *The Mechanical Engineering Design Process – 3<sup>rd</sup> ed.* 2003, New York: McGraw Hill.
- [4] Committee on Engineering Design Theory and Methodology, N.R.C.U.S., *Improving engineering design: designing for competitive advantage.* , ed. M.S.B. Committee on Engineering Design Theory and Methodology, Commission on Engineering and Technical Systems, National Research Council. 1991, Washington, D.C.: National Academy Press, 107 p.
- [5] Matelli, J. A., Bazzo, E., and da Silva, J. C., 2009, “An expert system prototype for designing natural gas cogeneration plants, *Expert Systems with Applications*, 36, pp. 8375-8384.
- [6] Proctor, P., 1995, “Boeing adopts Expert Design System,” *Aviation week and space technology*, 127(17), pg. 27.
- [7] Mittal, S., Dym, C. L., Morjaria, M., 1986, “PRIDE: An expert system for the design of paper handling systems,” *IEEE Computers*, 19(7), pp. 102-114.
- [8] Razavi, B., *Condition Monitoring in a Hydraulic System of an Industrial Machine using Unscented Kalman Filter.* M.A.Sc Thesis, 2009, University of British Columbia: Vancouver.
- [9] de Silva, C.W., “A Paradigm Shift in Machine Health Monitoring: A Unified System Framework Including Intelligent Supervisory Control and On-Line Design Evolution with Networked Remote Operation,” *Tier 1 Canada Research Chair in Mechatronics and Industrial Automation*, the University of British Columbia, Vancouver, BC, Canada, 2008.
- [10] Campos, J. R., *Design Evolution of Engineering Systems through Modeling, On-line Monitoring, and Evolutionary Optimization.* M.A.Sc Thesis, 2010, University of British Columbia: Vancouver.
- [11] W. J. Wang and P. D. McFadden, “Application of wavelets to gearbox vibration signals for fault detection,” *J. Sound Vibrat.*, vol. 192, pp. 927-939, 5/23. 1996.

- [12] Raman, S., *Condition Monitoring of Industrial Machines using Wavelet Packet and Intelligent Multisensor Fusion*. M.A.Sc Thesis, 2009, University of British Columbia: Vancouver.
- [13] Lang, H., *Remote Monitoring and Fault Diagnosis of an Industrial Machine through Sensor Fusion*. M.A.Sc Thesis, 2007, University of British Columbia: Vancouver.
- [14] Koza, J.R., *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. 1992, The MIT Press.
- [15] Koza, J.R., *Genetic Programming II - Automatic Discovery of Reusable Programs*. 1994, The MIT Press.
- [16] Holland, J.H., *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. 1975, Ann Arbor, MI: University of Michigan Press.
- [17] Seo, K., et al., *Toward a Unified and Automated Design Methodology for Multi-domain Dynamic Systems Using Bond Graphs and Genetic Programming*. *Mechatronics*, 2003.13(8-9). p. 851-885.
- [18] Behbahani, S., *Practical and Analytical Studies on the Development of Formal Evaluation and Design Methodologies for Mechatronic Systems in Mechanical Engineering*. 2007, University of British Columbia: Vancouver. p. xx, 164.
- [19] Samarakoon, B. and de Silva, C. W., 2011, "Controlled evolution of engineering systems using bond graph and genetic programming," *Proc. 23<sup>rd</sup> Canadian Congress of Applied Mechanics*, Vancouver, Canada, June 5-9.
- [20] Alipourazadi, S. and de Silva, C. W., 2011, "Modeling of one-dimensional distributed system using linear graph," *Proc. 23<sup>rd</sup> Canadian Congress of Applied Mechanics*, Vancouver, Canada, June 5-9.
- [21] De Silva, C.W., *Mechatronics: An Integrated Approach*. 2005, Boca Raton, FL: CRC Press, Taylor & Francis.
- [22] De Silva, C.W., *Mechatronics: A Foundation Course*. 2010, Boca Raton, FL: CRC Press, Taylor & Francis.
- [23] De Silva, C.W., *Modeling and Control of Engineering Systems*. 2009, Boca Raton, FL: CRC Press, Taylor & Francis.
- [24] Giarratano, J., Riley, G. *Expert Systems: Principles and Programming – 3<sup>rd</sup> ed.* 1998, PWS, Boston, MA.

- [25] De Silva, C.W. *Sensors and Actuators - Control System Instrumentation*. 2009, CRC Press, Boca Raton, FL.
- [26] Durkin, J., *Expert System Design and Development*, 1994, New Jersey: Prentice Hall.
- [27] Karray, F. and de Silva, C.W., *Soft Computing and Intelligent Systems Design*, Pearson Education, UK, Soft Cover Edition, 2008.
- [28] Turban, E., *Decision support systems and expert systems 4<sup>th</sup> ed.* 1995, Englewood Cliffs, NJ: Prentice Hall.
- [29] Hart, A., *Knowledge acquisition for expert systems*. 1989, London, UK: Kogan Page.
- [30] Kendal, S.L., Creen, M., *An introduction to knowledge engineering*. 2007, London: Springer
- [31] Friederich, S., and Gargano, M., *Expert Systems Design and Development using VP-Expert*. 1989, New York: Wiley.
- [32] Liao, S., 2004, "Expert system methodologies and applications – a decade review from 1995 to 2004," *Expert Systems with Applications*, 28(1), pp. 93-103.
- [33] Agrawal, D., and Vasudevan, P. T., 1993, "PLASSEX: An Expert System for Plastics Selection," *Advances in Polymer Technology*, 12(4), pp. 419-428.
- [34] Sapuan, S. M., 2001, "A knowledge-based system for materials selection in mechanical engineering design," *Materials and Design*, 22, pp. 687-695.
- [35] Zarandi, M. H. F. et al, 2011, "A material selection methodology and expert system for sustainable product design," *International Journal of Advanced Manufacturing Technology*, 57, pp. 885-903.
- [36] Ganguli, R., and Bandopadhyay, S., 2002, "Expert System for Equipment Selection," *International Journal of Mining, Reclamation and Environment*, 16(3), pp. 163-170.
- [37] Er, A., and Dias, R., 2000, "A rule-based expert system approach to process selection for cast components," *Knowledge-Based Systems*, 13, pp. 225-234.
- [38] Ferguson, G. L. et al, 1999, "Expert System for selecting speed reduction components for a power transmission," *Journal of Manufacturing Systems*, 18(1), pp. 66-74.

- [39] Milani, A. A., and Hamed, M., 2008, "A Knowledge-based System for Selecting Fastening Tools in Automobile Assembly Lines," *Proc. IEEE Int. Conference on Industrial Technology*, Chengdu, China, April 21-24.
- [40] Liu, T. I., Singonahalli, J. H., and Iyer, N. R., 1996, "Detection of roller bearing defects using expert system and fuzzy logic," *Mechanical Systems and Signal Processing*, 10(5), pp. 595-614.
- [41] Franklin, R., and Halabe, U. B., 1997, "Knowledge-Based Assistant for Ultrasonic Inspection in Metals," *Proc. of SPIE - The International Society for Optical Engineering*, v 3203 (Architectures, Networks, and Intelligent Systems for Manufacturing Integration), pp. 44-48.
- [42] Yang, B., Lim, D., and Tan, A., 2005, "VIBEX: an expert system for vibration fault diagnosis of rotating machinery using decision tree and decision table," *Expert Systems with Applications*, 28, pp. 735-742.
- [43] Ebersbach, S., and Peng, Z., 2008, "Expert system development for vibration analysis in machine condition monitoring," *Expert Systems with Applications*, 34, pp. 291-299.
- [44] Al-Taani, A., 2005, "An Expert System for Car Failure Diagnosis," *World Academy of Science, Engineering and Technology*, 12, pp. 4-7.
- [45] Moshage, R. et al., 1993, *Application of Expert Systems for Diagnosing Equipment failures at Central Energy Plants*, U.S. Army Construction Engineering Research Laboratories (USACERL), Champaign, IL.
- [46] Hoole, S. R. H. et al., 2003, "An Expert Design Environment for Electrical Devices and its Engineering Assistant," *IEEE Trans. On Magnetics*, 39(3), pp. 1693-1696.
- [47] Abersek, B., Flaker, J., and Balic, J., 1996, "Expert System for Designing and Manufacturing of a Gear Box," *Expert Systems with Applications*, 11(3), pp.397-405.
- [48] Matelli, J. A., Bazzo, E., and da Silva, J. C., 2009, "An expert system prototype for designing natural gas cogeneration plants," *Expert Systems with Applications*, 36, pp. 8375-8384.
- [49] Sakr, K. M., and Hosain, M. U., 1991, "Applications of expert system building tools in structural design," *Canadian Journal of Civil Engineering*, 18, pp. 493-503.
- [50] Manninen, J., & Zhu, X. X. (2001). Level-by-level flowsheet synthesis methodology for thermal system design. *AIChE Journal*, 47(1), 142–159.

- [51] Saggu, J. S., and Morris, A. J., 1990, "Design for Manufacture: A Knowledge-Based Approach," *Proc. Of IEEE Second Int'l Conf. on Data and Knowledge Systems for Manufacturing and Engineering*, pp. 198-205.
- [52] Liu, T. I., Yang, X. M., and Kalambur, G. J., 1995, "Design for Machining Using Expert System and Fuzzy Logic Approach," *Journal of Materials Engineering and Performance*, 4(5), pp. 599-605.
- [53] Liu, T. I., and Yang, X. M., 1999, "Design for quality and reliability using expert system and computer spreadsheet," *Journal of the Franklin Institute*, 336, pp. 1063-1074.
- [54] Fritz, C., and MacCleery, B., 2009, *Lower your energy bill with field-oriented motor control*, NI Instrumentation Newsletter, Austin, TX.
- [55] Voss, W., 2007, *A Comprehensible Guide to Servo Motor Sizing*, CopperHill, Greenfield, MA.
- [56] McCoy, G. A. et al., 1993, *Energy-Efficient Electric Motor Selection Handbook*, U.S. Department of Energy, Washington, DC.
- [57] [http://industrial.panasonic.com/ww/e/25000/fa\\_pro\\_sg\\_eard\\_shing1\\_e/fa\\_pro\\_sg\\_eard\\_shing1\\_e/ctlg\\_geared\\_e\\_9.pdf](http://industrial.panasonic.com/ww/e/25000/fa_pro_sg_eard_shing1_e/fa_pro_sg_eard_shing1_e/ctlg_geared_e_9.pdf) consulted 20 April, 2011.
- [58] Harrington, R. L., *Marine Engineering*. 1992, New Jersey: The Society of Naval Architects and Marine Engineers.
- [59] Marine Technology Consultants (VERITEC), *Vibration Control in Ships*. 1985, Oslo, Norway.
- [60] Saunders, H. E., *Hydrodynamics in ship design*. 1957, New Jersey: The Society of Naval Architects and Marine Engineers.
- [61] Kyriakopoulos, K.J., Saridis, G.N., "Minimum Jerk Path Generation", *Proc. 1988 IEEE Int. Conf. Rob. & Aut.*, 1988, pp.364-369.
- [62] Constantinescu, D., Croft E.A., "Smooth and Time-Optimal Trajectory Planning for Industrial Manipulators along Specified Paths", *J. Rob. Sys.*, Vol. 17, No. 5, 2000, pp. 233-249.
- [63] Llyod, J. and Hayward, V., 1993, "Trajectory generation for sensor-driven and time-varying tasks," *The International Journal of Robotics Research*, 12(4), pp. 380.

- [64] Bonnet, A. H., 2001, "Operating Temperature Considerations and Performance Characteristics for IEEE 841 Motors," *IEEE Transactions on Industry Applications*, 37(4), pp. 1120-1131.
- [65] Arora, J. S., *Introduction to Optimum Design*. 1989, New York: McGraw Hill.
- [66] Belegundu, A. D. and Chandrupatla, T. R., *Optimization concepts and applications in Engineering*. 1999 New Jersey: Prentice Hall.
- [67] Venkataraman, P., *Applied optimization with Matlab programming*. 2002, New York: Wiley.

## Appendix A – Inertia Equations

**Table A.1: Equivalent polar moments of inertia of MCDES GA.**

Generic application	Equivalent polar moment of inertia $J_{eq}$ [kgm <sup>2</sup> ]
Hoister	$J_{eq_i} = J_{m_i} + J_d + r^2(W + m_R)$ <p> <math>J_m</math>: motor rotor inertia [kgm<sup>2</sup>]  <math>J_d</math>: pulley inertia <math>\left(J_d = \frac{1}{8}m_p D^2\right)</math> [kgm<sup>2</sup>]  <math>m_p</math>: winch pulley weight [kg]  <math>D</math>: winch pulley diameter [m]  <math>r</math>: winch pulley radius [m]  <math>W</math>: mass of load [kg]  <math>m_R</math>: mass of rope (if considerable) [kg] </p>
Flywheel	$J_{eq_i} = J_{m_i} + J_d + J_{coupling}$ <p> <math>J_m</math>: motor rotor inertia [kgm<sup>2</sup>]  <math>J_d</math>: flywheel inertia <math>\left(J_d = \frac{1}{8}m_d D^2\right)</math> [kgm<sup>2</sup>]  <math>m_d</math>: flywheel weight [kg]  <math>D</math>: flywheel diameter [m]  <math>J_{connection}</math>: coupling inertia (if applicable) [kgm<sup>2</sup>] </p>
Belt Conveyor	$J_{eq_i} = J_{m_i} + J_d + J_s + r^2(nW + m_c)$ <p> <math>J_m</math>: motor rotor inertia [kgm<sup>2</sup>]  <math>J_d</math>: driving cylinder inertia <math>\left(J_d = \frac{1}{8}m_d D_d^2\right)</math> [kgm<sup>2</sup>]  <math>m_d</math>: driving cylinder weight [kg]  <math>D_d</math>: driving cylinder diameter [m]  <math>J_s</math>: driven cylinder inertia <math>\left(J_s = \frac{1}{8}m_s D_s^2\right)</math> [kgm<sup>2</sup>]  <math>m_s</math>: driven cylinder weight [kg]  <math>D_s</math>: driven cylinder diameter [m]  note: <math>D_s = D_d = D</math>  <math>r</math>: driving cylinder radius [m]  <math>n</math>: number of weights on conveyor  <math>W</math>: mass of load [kg]  <math>m_c</math>: mass of conveyor belt (if considerable) [kg] </p>

Generic application	Equivalent polar moment of inertia $J_{eq}$ [kgm <sup>2</sup> ]
Horizontal puller	$J_{eq_i} = J_{m_i} + J_d + r^2 W (\mu + 1)$ <p> <math>J_m</math>: motor rotor inertia [kgm<sup>2</sup>]  <math>J_d</math>: pulley inertia <math>\left(J_d = \frac{1}{8} m_p D^2\right)</math> [kgm<sup>2</sup>]  <math>m_p</math>: pulley weight [kg]  <math>D</math>: pulley diameter [m]  <math>r</math>: pulley radius [m]  <math>W</math>: mass of load [kg]  <math>\mu</math>: frictional coefficient </p>
Ball screw drive/ Linear actuator	$J_{eq_i} = J_{m_i} + J_A + \frac{(W_L + W_T)}{g * e} \left(\frac{1}{2\pi * P}\right)^2$ <p> <math>J_m</math>: motor rotor inertia [kgm<sup>2</sup>]  <math>J_A</math>: feedscrew inertia [kgm<sup>2</sup>]  <math>W_L</math>: mass of load [kg]  <math>W_T</math>: mass of table or slide [kg]  <math>P</math>: lead of feedscrew [m]  <math>e</math>: mechanism efficiency </p>
Nip roll	$J_{eq_i} = J_{m_i} + \frac{m_m r_m^2}{2} + \left(\frac{1}{N_r}\right)^2 \frac{m_l r_l^2}{2e}$ <p> <math>J_m</math>: motor rotor inertia [kgm<sup>2</sup>]  <math>m_m</math>: mass of motor side roller [kg]  <math>r_m</math>: radius of motor side roller [kg]  <math>m_l</math>: mass of load side roller [kg]  <math>r_l</math>: radius of load side roller [m]  <math>e</math>: mechanism efficiency  <math>N_r</math>: Transmission ratio <math>\left(\frac{r_l}{r_m}\right)</math> </p>

Generic application	Equivalent polar moment of inertia $J_{eq}$ [kgm <sup>2</sup> ]
Rack and pinion	$J_{eq_i} = J_{m_i} + \frac{m_p r_p^2}{2} + \frac{(W_L + W_R)}{g * e} (r_p)^2$ <p> <math>J_m</math>: motor rotor inertia [kgm<sup>2</sup>]  <math>m_p</math>: mass of pinion [kg]  <math>r_p</math>: radius of pinion [m]  <math>W_L</math>: mass of load [kg]  <math>W_R</math>: mass of rack [kg]  <math>e</math>: mechanism efficiency </p>
Rotary table	$J_{eq_i} = J_{m_i} + \left(\frac{1}{N_r}\right)^2 \frac{(m_d + m_l) r_d^2}{2e} + J_G$ <p> <math>J_m</math>: motor rotor inertia [kgm<sup>2</sup>]  <math>m_d</math>: mass of disk [kg]  <math>m_l</math>: mass of load [kg]  <math>r_d</math>: radius of disk [m]  <math>e</math>: mechanism efficiency  <math>N_r</math>: Transmission ratio from table manufacturer or  <math>\left(\frac{\# \text{ of teeth input gear}}{\# \text{ of teeth output gear}}\right)</math>  <math>J_G</math>: Gear inertia (from table manufacturer) [kgm<sup>2</sup>] </p>
Center Driven Winder/ Surface Driven Winder	<p>Center driven:</p> $J_{eq_i} = J_{m_i} + \frac{m_w r_w^2}{2} + 2 \left(\frac{1}{N_r}\right)^2 \frac{(m_s r_s^2)}{e}$ <p>Surface driven:</p> $J_{eq_i} = J_{m_i} + \frac{(m_s r_s^2)}{2} + \frac{m_s r_s^2}{e} + \left(\frac{1}{N_r}\right)^2 \frac{m_w r_w^2}{e}$ <p> <math>J_m</math>: motor rotor inertia [kgm<sup>2</sup>]  <math>m_w</math>: mass of winding/unwinding roller [kg]  <math>r_w</math>: radius of winding/unwinding roller [m]  <math>m_s</math>: mass of 1 support roller [kg] (other assumed the same)  <math>r_s</math>: radius of 1 support roller [m] (other assumed the same)  <math>e</math>: mechanism efficiency  <math>N_r</math>: Transmission ratio  Center: <math>\left(\frac{r_s}{r_w}\right)</math> Surface: <math>\left(\frac{r_w}{r_s}\right)</math> </p>

Generic application	Equivalent polar moment of inertia $J_{eq}$ [kgm <sup>2</sup> ]
<p>Propeller/Fan</p> <p>[56 - 58]</p>	$J_{eq_i} = J_{m_i} + J_P + J_E$ $J_P = \frac{WD^2}{K_1}$ $J_E = K_2 J_P \text{ (VERITEC estimate)}$ <p> <math>J_m</math>: motor rotor inertia [kgm<sup>2</sup>]  <math>J_P</math>: propeller mass moment of inertia (from weight alone) [kgm<sup>2</sup>]  <math>J_E</math>: propeller torsional entrained water moment of inertia [kgm<sup>2</sup>]  <math>W</math>: propeller weight [lb]  <math>D</math>: propeller diameter [in] </p> <p>If weight not known: <math>W = K_3 D^3 (MWR)(BTF)</math></p> <p> <math>K_3</math>: material density factor  0.26 for Mn – Brz  0.25 for Ni – Mn – Brz  0.235 for Ni – Al – Brz  0.235 for Mn – Ni – Al – Brz  0.225 for cast iron  0.245 for stainless steel </p> <p> <math>MWR</math>: mean width ratio = <math>\frac{\text{developed area per blade}}{D(\text{blade radius} - \text{hub radius})}</math> </p> <p> <math>BTF</math>: blade thickness fraction  <math display="block">= \frac{\text{maximum blade thickness extrapolated to shaft axis [in]}}{D}</math> </p> <p><math>K_1</math>: blade outline factor: <math>19 &lt; K_1 &lt; 28</math></p> <p><math>K_2</math>: entrained water factor: <math>0.25 &lt; K_2 &lt; 0.3</math></p>

**Table A.2: Equivalent polar moments of inertia of MCDES custom components**

Component	Polar moment of inertia $J$ [kgm <sup>2</sup> ]
Solid shaft	<p>Along longitudinal axis</p> $J = \frac{mr^2}{2} = \frac{Wr^2}{2g} = \frac{\pi L \rho r^4}{2g}$ <p><math>r</math>: cylinder radius [m]  <math>m</math>: cylinder mass [kg]</p>
Hollow shaft	<p>Along longitudinal axis</p> $J = \frac{m}{2}(r_o^2 + r_i^2) = \frac{W}{2g}(r_o^2 + r_i^2) = \frac{\pi L \rho}{2g}(r_o^4 - r_i^4)$ <p><math>r_o</math>: cylinder outer radius [m]  <math>r_i</math>: cylinder inner radius [m]  <math>m</math>: cylinder mass [kg]</p>
Coupling	<p>Same as hollow shaft above except that the inner radius of the coupling equals the outer radius of the shaft. The outer radius of the coupling and the new material density are entered accordingly</p>
Chain-sprocket transmission	$J = \frac{m_{SM}r_{SM}^2}{2} + \left(\frac{1}{N_r}\right)^2 \frac{m_{SL}r_{SL}^2}{2e} + \frac{W_C}{ge}(r_{SM})^2 + \left(\frac{1}{N_r}\right)^2 \frac{J_L}{e}$ <p><math>J_L</math>: Load inertia [kgm<sup>2</sup>]  <math>e</math>: mechanism efficiency  <math>N_r</math>: Transmission ration <math>\left(\frac{r_{SL}}{r_{SM}}\right)</math>  <math>r_{SM}</math>: motor – side sprocket radius [m]  <math>r_{SL}</math>: load – side sprocket radius [m]  <math>m_{SM}</math>: motor – side sprocket mass [kg]  <math>m_{SL}</math>: load – side sprocket mass [kg]  <math>W_C</math>: Chain weight [kg]</p>

Component	Polar moment of inertia $J$ [kgm <sup>2</sup> ]
Belt-pulley transmission	$J = \frac{m_{PM}r_{PM}^2}{2} + \left(\frac{1}{N_r}\right)^2 \frac{m_{PL}r_{PL}^2}{2e} + \frac{W_B}{ge}(r_{PM})^2 + \left(\frac{1}{N_r}\right)^2 \frac{J_L}{e}$ <p> <math>J_L</math>: Load inertia [kgm<sup>2</sup>]  <math>e</math>: mechanism efficiency  <math>N_r</math>: Transmission ration <math>\left(\frac{r_{PL}}{r_{PM}}\right)</math>  <math>r_{PM}</math>: motor – side pulley radius [m]  <math>r_{PL}</math>: load – side pulley radius [m]  <math>m_{PM}</math>: motor – side pulley mass [kg]  <math>m_{PL}</math>: load – side pulley mass [kg]  <math>W_B</math>: Belt weight [kg] </p>
Multi-output gearbox	$J = \frac{m_{GM}r_{GM}^2}{2} + \left(\frac{1}{N_r}\right)^2 \frac{m_{GL}r_{GL}^2}{2e} + \left(\frac{1}{N_r}\right)^2 \frac{J_L}{e}$ <p> <math>J_L</math>: Load inertia [kgm<sup>2</sup>]  <math>e</math>: mechanism efficiency  <math>N_r</math>: Transmission ration <math>\left(\frac{r_{GL}}{r_{GM}}\right)</math>  <math>r_{GM}</math>: motor – side gear radius [m]  <math>r_{GL}</math>: load – side gear radius [m]  <math>m_{GM}</math>: motor – side gear mass [kg]  <math>m_{GL}</math>: load – side gear mass [kg] </p>

Component	Polar moment of inertia $J$ [kgm <sup>2</sup> ]
Roller bearing	$J = J_{InnerRace} + J_{RE}n_{RE}$ $J = \frac{m_{total}}{6}(r_o^2 + r_i^2) + J_{RE}n_{RE}$ <p>For spherical roller element, <math>J_{RE} = \frac{2m_{total}r_{RE}^2}{15}</math>  For cylindrical roller element, <math>J_{RE} = \frac{m_{total}r_{RE}^2}{6}</math>  For needle roller element, <math>J_{RE} = \frac{m_{total}r_{RE}^2}{8}</math></p> <p><math>m_{total}</math>: total mass of the bearing [kg]  <math>r_o</math>: radius of the outer face of the inner race [m]  <math>r_i</math>: radius of the inner face of the inner race [m]  <math>r_{RE}</math>: approximate radius of rolling element [m]  <math>n_{RE}</math>: number of rolling elements in the bearing</p> <p>Note:</p> <ul style="list-style-type: none"> <li>- Outer race and housing are stationary and thus not considered</li> <li>- Inner race, outer race, and rolling elements are assumed to have equal masses = <math>\frac{m_{total}}{3}</math></li> </ul>

## Appendix B – Motor and Gear Data

**Table B.1: Full specifications list of motors used in MCDES.**

Motor #	$J_m$	$\omega_{nom}$	$T_{nom}$	$\eta$	$R_T$	$K_T$	$R_{TH_a}$	$T_{a\_max}$	$R_{TH_w}$	$T_{w\_max}$
<b>118628</b>	6.77E-08	11300	0.00122	0.68	0.544	0.00219	33	65	7	85
<b>118630</b>	5.27E-08	11300	0.00195	0.66	0.98	0.0027	33	65	7	85
<b>118638</b>	5.37E-08	10300	0.00244	0.67	14.1	0.0104	33	65	7	85
<b>110143</b>	3.97E-07	5550	0.00582	0.65	2.2	0.0059	20	85	6	125
<b>110151</b>	4.11E-07	7430	0.00697	0.73	21	0.0212	20	85	6	125
<b>226801</b>	1.43E-06	6690	0.011	0.79	0.571	0.0109	15.8	85	4	125
<b>226805</b>	1.09E-06	7760	0.0197	0.85	1.69	0.019	15.8	85	4	125
<b>226806</b>	1.35E-06	7680	0.0269	0.87	2.36	0.0258	15.8	85	4	125
<b>226815</b>	1.25E-06	5050	0.0299	0.87	19.7	0.0747	15.8	85	4	125
<b>236644</b>	3.90E-06	2980	0.0331	0.75	1.76	0.0171	7.5	85	2.1	125
<b>236645</b>	4.33E-06	2860	0.036	0.77	3.02	0.0241	7.5	85	2.1	125
<b>236666</b>	4.39E-05	3140	0.0384	0.77	0.577	0.0116	7.5	85	2.1	125
<b>236668</b>	4.14E-06	3100	0.0421	0.8	2.23	0.0243	7.5	85	2.1	125
<b>236670</b>	4.40E-06	4710	0.0444	0.83	7.17	0.0461	7.5	85	2.1	125
<b>323219</b>	5.54E-07	14700	0.0505	0.87	1.24	0.0211	9.08	100	0.904	155
<b>311535</b>	8.91E-07	15600	0.059	0.89	0.202	0.0102	8.01	100	1	155
<b>311537</b>	8.91E-07	16700	0.0631	0.89	0.61	0.0192	8.01	100	1	155
<b>309755</b>	1.83E-06	16800	0.0697	0.86	0.135	0.00958	7.95	100	0.831	155
<b>310006</b>	3.57E-06	7900	0.0755	0.87	0.362	0.0199	6	100	1.7	125
<b>310008</b>	3.29E-06	7810	0.0834	0.87	1.53	0.0398	6	100	1.7	125
<b>310009</b>	3.45E-06	7750	0.0882	0.88	2.52	0.0538	6	100	1.7	125
<b>148866</b>	1.39E-05	6370	0.0949	0.88	0.117	0.0164	4.65	100	1.93	155

Motor #	$J_m$	$\omega_{nom}$	$T_{nom}$	$\eta$	$R_T$	$K_T$	$R_{TH_a}$	$T_{a_{max}}$	$R_{TH_w}$	$T_{w_{max}}$
305013	3.33E-06	16200	0.112	0.88	0.102	0.0135	5.3	100	0.209	155
136204	1.19E-05	9360	0.163	0.79	0.308	0.0214	1.9	100	0.9	125
136196	1.19E-05	4430	0.182	0.77	0.262	0.02	1.9	100	0.9	125
136210	2.09E-05	8380	0.285	0.85	0.143	0.025	1.7	100	1.1	125
136209	2.09E-05	5770	0.323	0.85	1.01	0.0704	1.7	100	1.1	125
323772	3.06E-04	2650	0.387	0.83	0.363	0.0705	1.89	100	2.99	125
370357	5.60E-05	2470	0.452	0.92	3.9	0.242	3.81	100	1.24	125
353295	1.29E-04	3640	0.51	0.843	0.0809	0.0567	1.3	100	1.85	125
353296	1.38E-04	3550	0.654	0.873	0.173	0.0891	1.3	100	1.85	125
353297	1.34E-04	3270	0.746	0.885	0.363	0.13	1.3	100	1.85	125
353300	1.28E-04	2820	0.83	0.891	1.06	0.219	1.3	100	1.85	125

**Table B.2: Full specifications list of gears used in MCDES.**

Gear #	$J_g$	$p$	Gear group	$\eta$
110313	2.50E-09	4.1	1	0.91
110314	1.50E-09	17	1	0.83
110315	1.50E-09	67	1	0.75
134516	5.00E-08	3.8	2	0.84
134158	4.00E-08	14	2	0.7
134163	4.00E-08	53	2	0.59
166930	1.50E-07	3.7	3	0.8
166933	8.00E-08	14	3	0.75
166938	8.00E-08	33	3	0.75
203113	1.40E-06	3.5	4	0.9
203115	1.50E-06	12	4	0.81
203119	9.10E-06	26	4	0.81
223080	2.07E-06	3.5	5	0.91
223083	1.76E-06	12	5	0.83
223089	1.73E-09	43	5	0.75
110408	1.65E-05	3.7	6	0.8
110409	1.55E-05	14	6	0.75

## **Appendix C – MCDES Code**

Due to the exceedingly large size of the MCDES code (~ 100 pages), it will not be included here. The core part of this code can be found using the internet link below.

Please contact the author for auxiliary sections of the code.

[http://www.sites.mech.ubc.ca/~ial/docs/MCDES\\_core\\_code.pdf](http://www.sites.mech.ubc.ca/~ial/docs/MCDES_core_code.pdf)

# Appendix D – Conveyor Specifications

<b>Motion Metrics</b> <i>International Corp</i>		Head Office: 289-2366 Main Mall, Vancouver, British Columbia, Canada V6T 1Z4 Tel +1 604.822.5842 Fax +1 604.677.5191 www.MotionMetrics.com	
By: Daniel Ko		Date: 2012-02-20	

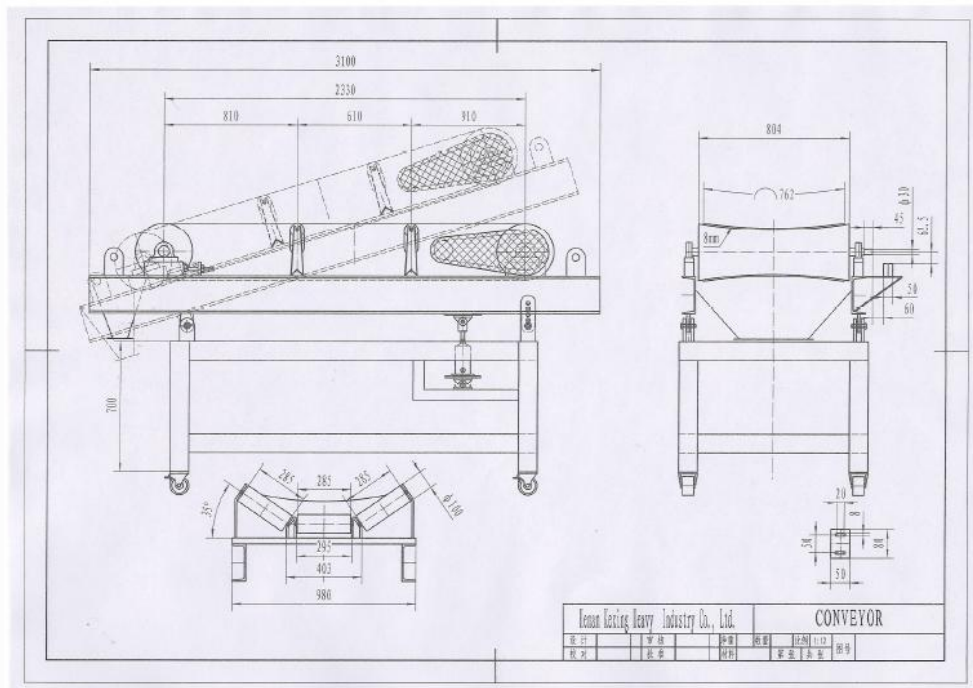
Name	Belt Width	Belt Thickness	Length (pulley shaft to pulley shaft)	Pulley Diameter	DC Motor With Speed Control	Power Cord
Conveyor (Design 2)	30 Inches	8 mm	2,330 mm	1 Foot	1 set	White Color

## Composition of the Conveyor:

1. Belt Pulley: 2 pieces
2. Belt: 1 set
3. Frame: 1 set
4. DC Motor (Made in China): 1 piece
5. Motion Controller (Made in China) : 1 piece
6. Hopper: 1 piece
7. Plug Seat: 1 set
8. White Color Power Cord: 1 set
9. Idlers: 6 pieces (two sets idlers, and one set idlers includes 3 idlers)
10. Sprocket : the diameter of bigger sprocket is 30 cm (1 piece),  
the diameters of the smaller sprockets(3 pieces):  
(1).Small sprocket 1: diameter 6 cm  
(2).Small sprocket 2: diameter 7,5 cm  
(3).Small sprocket 3: diameter 9 cm
11. Chain: 1 set
12. Adjustable Belt Scraper: 1 set
13. Chain Cover: 1 set
14. Transportation wheels with Locking mechanism: 4 pieces
15. Adjustable Bearing Housing: 1 set
16. Adjustable Lift jack mechanism: 1 set
17. Plexiglass Sidewall: 1 set
18. Encoder Mounting Platform: 1 set

**Figure D.1: Geometric parameters of custom conveyor setup used for verification.**

**A Drawing of Design 2 (Dimensions: mm)**



**Figure D.2: Engineering drawing of custom conveyor setup used for verification.**