THESIS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Energy Reduction of Robot Stations with Uncertainties

Mattias Hovgard

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden, 2022

Energy Reduction of Robot Stations with Uncertainties
Mattias Hovgard
ISBN 978-91-7905-719-0

Doktorsavhandlingar vid Chalmers tekniska högskola
Ny serie nr 5185
ISSN 0346-718X

Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg, Sweden
Phone: +46 (0)31 772 1000
www.chalmers.se

Printed by Chalmers Reproservice
Gothenburg, Sweden, 2022

To my family.

Abstract

This thesis aims to present a practical approach to reducing the energy use of industrial robot stations. The starting point of this work is different types of robot stations and production systems found in the automotive industry, such as welding stations and human-robot collaborative stations, and the aim is to find and verify methods of reducing the energy use in such systems. Practical challenges with this include limited information about the systems, such as energy models of the robots; limited access to the stations, which complicates experiment and data collection; limitations in the robot control system; and a general reluctance by companies to make drastic changes to already tested and approved production systems. Another practical constraint is to reduce energy use without slowing down production. This is especially challenging when a robot station contains stochastic variations, which is the case in many practical applications.

Motivated by these challenges, this thesis presents an offline method of reducing the energy use of a production line of welding stations in an automotive factory. The robot stations contain stochastic uncertainties in the form of variations in the robot execution times, and the energy use is reduced by limiting the robot velocities. The method involves collecting data, modeling the system, formulating and solving a nonlinear and stochastic optimization problem, and applying the results to the real robot station. Tests on real stations show that, with only small modifications, the energy use can be reduced significantly, up to 24 percent.

The thesis also contains an online method of controlling a collaborative human-robot bin picking station in a robust and energy-optimal way. The problem is partly a scheduling problem to determine in which orders the operations should be executed, and a timing problem to determine the velocities of the robots. A particular challenge is that some model parameters are unknown and have to be estimated online. A multi-layered control algorithm is presented that continuously updates the operation order and tunes the robot velocities as new orders arrive in the system. Simultaneously, a reinforcement learning algorithm is used to update estimates of the unknown parameters to be used in the optimization algorithms.

Keywords: Energy optimization, robot station, industrial robot, stochastic uncertainties.

List of Publications

This thesis is based on the following publications:

[A] **Mattias Hovgard**, Bengt Lennartson, and Kristofer Bengtsson, "Applied Energy Optimization of Multi-Robot Systems Through Motion Parameter Tuning". *CIRP Journal of Manufacturing Science and Technology, 35, 422-430, 2021.*

[B] **Mattias Hovgard**, Bengt Lennartson, and Kristofer Bengtsson, "Energy Reduction of Stochastic Time-Constrained Robot Stations". *Revised journal submission.*

[C] **Mattias Hovgard**, Bengt Lennartson, and Kristofer Bengtsson, "Energy-Optimal Timing of Stochastic Robot Stations in Automotive Production Lines". *Accepted for the 27th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Stuttgart, September 2022.*

[D] **Mattias Hovgard**, Bengt Lennartson, and Kristofer Bengtsson, "Online Energy-Optimal Timing of Stochastic Robot Stations". *Proceedings of the 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Västerås, September 2021.*

[E] **Mattias Hovgard**, Constantin Cronrath, Kristofer Bengtsson, and Bengt Lennartson, "Adaptive Energy Optimization of Flexible Robot Stations". *Journal submission.*

Other publications by the author, not included in this thesis, are:

[F] **Mattias Hovgard**, Bengt Lennartson, and Kristofer Bengtsson, "Simulation Based Energy Optimization of Robot Stations by Motion Parameter Tuning". *Proceedings of the IEEE 15th International Conference on Automation Science and Engineering (CASE), 456-461, 2019.*

[G] **Mattias Hovgard**, Bengt Lennartson, and Kristofer Bengtsson, "Energy-Optimal Timing of Robot Stations Subject to Gaussian Disturbances". *Proceedings of the 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), 1441-1444, 2019.*

[H] **Mattias Hovgard**, Oscar Jonsson, Nikolce Murgovski, Martin Sanfridson, and Jonas Fredriksson, “Cooperative Energy Management of Electrified Vehicles on Hilly Roads”. *Control Engineering Practice, 73, 66-78, 2018.*

Acknowledgments

Many people have helped make this thesis possible. Firstly, I would like to thank my supervisors Kristofer Bengtsson and Bengt Lennartson for fruitful discussions and for providing valuable feedback. Secondly, I would like to thank the other seniors of the Automation group: Martin, Petter, Knut, Sahar and Emmanuel for their wisdom and advice. I would like to express my gratitude to Nina, Oskar, and Sarmad that helped me get started at the beginning of my PhD and whose work became the basis of my research activities. I would also like to thank my office mates Sabino, Constantin, Zahra, and the rest of the colleagues in my research group and department, for providing great company, inspiration and support during these five years. You are too many to name, but I hope you know who you are. Finally, I would like to thank my friends and family for all their love and support.

This work has been supported by Vinnova FFI through the project SmoothIT and the ITEA3-project SPEAR. This support is gratefully acknowledged. During this time, I have been collaborating with some companies that have provided real applications to study and practical insights that have helped my research; ABB, Volvo Cars, and AB Volvo. I would like to thank the companies as a whole but specifically the employees Martin from ABB; Stefan, Rei, Håkan, Martin, and Göran from Volvo Cars in Gothenburg; and Andreas from AB Volvo in Umeå.

Contents

Part I

Overview

CHAPTER 1

Introduction

With increasing energy prices and the threat of global warming, many actors in society strive to reduce their energy use. According to the International Energy Agency [1] the possible energy savings for the industry is up to 30 %. This thesis specifically looks at the energy reduction potential in the robotics and automation systems, which in many industries, such as the automotive, is significant. The thesis takes a practical approach and investigates challenges in achieving the desired energy reduction, and presents methods that are both effective in terms of energy reduction and possible to apply in practice. It deals with challenges like unknown model parameters, limitations in the robot control system, reluctance by companies to make drastic changes to already tested and approved production systems, competing interests of other performance criteria and stochastic uncertainties in the production systems. This thesis aims to address some of these challenges, find methods to overcome them, and verify the effectiveness of the methods on real examples of robot stations. The content of this thesis is partly a continuation of the work done in AREUS [2] and has been conducted in collaboration with companies, such as AB Volvo, Volvo Cars, and ABB through the projects SmoothIt and SPEAR. The proposed methods are verified on case studies provided by these companies, such as welding stations in an automotive production line, and a flexible robot kitting station.

1.1 Background

Various approaches to the reduction of energy use of robotics and automation systems exist, such as changing the hardware of robots and tools [3], [4], optimizing the physical layout of robot stations [5], operation scheduling [6], turning off inactive machines [7] and trajectory optimization [8]. An overview of energy optimization methods for robots is given by Carabin et al. [9]. Looking specifically at the energy reduction of robot movements, which is the main approach of this thesis, the required steps can roughly be divided into three parts. They are: obtaining data and creating models of the robots and stations, determining the energy optimal behaviour of the robots within given constraints, and applying this optimal behaviour to the robots. Theoretical and practical challenges with each of these steps and existing methods to overcome them are discussed below.

The accuracy of an energy reduction method partly depends on the accuracy of the robot energy models. Ideally, complete models of the mechanical and electrical systems of the robots would be available, which is required in some energy optimization methods [10]–[13]. Finding the parameters (masses, friction coefficients, capacitance etc.), required to create such models, are however not always possible. As opposed to robots specifically designed for research purposes, this information is in general not available for industrial robots. In theory, different types of system identification methods [14], [15] can be used to find these models. However, finding an accurate robot energy model can be challenging, especially if the robot has many degrees of freedom, and if both the mechanical and the electrical systems are unknown. Furthermore, for an industrial robot used in production, it might not be possible to get access to the robot to execute the necessary excitation trajectories. To overcome this lack of complete energy models, methods have been proposed to minimize some quantity correlated to energy use, such as mechanical power or acceleration [2], [16]–[18]. Another simplified method is to limit the dimensionality of the problem by using energy signatures to describe the energy use as a function of the execution time [19], [20].

For industrial robots, a challenge to applying any energy-efficient control strategy in practice is possible limitations in the control system. A common and effective energy reduction approach is to directly change the trajectory of the robot [8], [16], [18], [21]. However, that may not be possible to implement in practice. In many robot control systems, a robot movement is typically defined by a simple motion command that is used by the control system to generate the trajectory. It is often possible to affect the trajectory generation in a limited way by modifying some parameters, but not much insight is given into how the trajectory is created. Methods have been proposed to emulate the behaviour of a robot controller [22] and to use the built-in functionality, such as velocity and acceleration settings, to reduce the energy use of the robot [23]–[25].

An important constraint when reducing the energy use of robot stations in practice is to not negatively affect the production flow of the factory, as that, in the authors' experience, often holds precedence over energy efficiency. When a robot station is taken into production it is thoroughly tested to make sure it works as intended. Any changes made after that point in time introduce risks for negative consequences, like brake-downs due to collisions or missed deadlines. Therefore, energy reduction methods aimed at robot stations already used in production should ideally limit the number of changes required. To guarantee that the productivity is not negatively affected, coordination of multiple robots and robot stations need to be considered [6], [26]–[29]. Then, the problem can be formulated as a scheduling problem of when in the execution of a robot station it is possible to achieve some energy reduction without negatively affecting the production flow. This scheduling problem becomes significantly harder if uncertainties are included in the problem. Different types of uncertainties exist in many practical applications, for example in bin-picking, human-robot collaboration, welding, and vision-based positioning. The inclusion of stochastic uncertainties makes it harder to guarantee that deadlines in the stations are met [30].

Given an energy model, a limited control system, and physical as well as performance constraints that need to be respected, the challenge is to formulate the energy reduction problem into a mathematical optimization problem [31] and to find a solver or solution algorithm to solve it [32], [33]. Depending on the complexity of the problem this may require some simplifications by reformulating the problem into a simpler class of optimization problems [34], [35] or dividing it into multiple optimization problems. Ideally, this should involve some quantification of how close the simplified problem is to the true one.

1.2 Research Approach

This section describes the approach used to conduct the research. The starting point was real robot stations provided by companies through research projects. The goal was to find methods to reduce the energy use of these in the most accessible and effective way possible, while not negatively affecting the production in other ways. The steps taken to achieve this can broadly be divided into modeling, optimization, and verification.

The first step was to create models of the stations, robots, and energy use. That was achieved by collecting as much data as possible from the real stations, either by collecting it directly from employees working with the real stations or by using software to log data during the execution of the stations. This data includes robot code, energy parameters, execution times of the operations, and different types of constraints specific to the particular robot stations. The constraints include physical constraints about robot movements, practical constraints about limitations in the robot control systems, and performance constraints where the deadline of the stations had to be met. During these steps, it became clear that there were some stochastic uncertainties in the stations that needed to be considered.

Extensive literature research was then made to investigate what energy reduction methods already exist, and that may be applicable given the available data and constraints. Based on these inputs some control methods were formulated. They include formulating the problems as mathematical optimization problems. It was then investigated what solvers exist, that are possible to obtain licences to, and that can solve the proposed optimization problems. Several of them were tested and compared in terms of speed, accuracy, and availability.

The final step was to verify that the proposed methods performed as expected, for which some experiments were conducted. This was done in two ways, in simulation and reality. Based on the data collected from the stations, models were built in simulation and the optimization methods were applied to them. Data about energy use, execution times, etc. were collected and analyzed. Where it was possible, the methods were also tested on the real stations. For this experiment, measurement equipment was connected to the power supply of the robot station. The station was then executed several times beforehand to warm up the robots since the energy use is influenced by the temperature of the robot joints. The station was then executed several times with and without optimization, to see how much the energy was reduced. This real data was then compared with the simulation model to verify the results from the simulation.

1.3 Research Questions

Given the existing literature presented above, together with practical challenges from the industry, several research questions have been formulated:

RQ1: What is the most accessible and effective offline method of reducing the energy use of industrial robot stations?

RQ2: How can the problem of incomplete or inaccurate system models be handled?

RQ3: Are there any additional benefits of reducing the energy use of industrial robot stations other than the energy reduction itself?

RQ4: What is the most robust and energy-efficient way of controlling a flexible robot station online?

1.4 Main Contributions

Through the efforts of answering the research questions, several scientific contributions have been made; they are summarized below.

- The first contribution from Papers A, B, and C is to present examples of automotive robot stations that can be used to compare energy reduction methods and to model them mathematically. This includes the overall structure of a robot station, robot energy models, models of the uncertainties in a robot station, constraints between robots and operations, and performance constraints.
- The second contribution is to present an offline method of reducing the energy use of these stations without increasing the cycle time of the

production. This involves creating models of the stations, finding simplified energy models, formulating the mathematical optimization problems, finding suitable solvers, and modifying the optimization problems in such a way that they can be solved by algorithms or solvers. This is done for a deterministic station in Paper A, stochastic stations in Paper B, and a whole production line in Paper C.

- The third contribution is to apply these methods to real examples of robot stations, both to verify the functionality of the methods but also to investigate the potential for energy reduction in the industry today, as well as other potential benefits of the energy reduction method. This is done on a real robot station in Paper A, on three simulation models of real robot stations in Paper B, and on a simulation model of a production line in Paper C.

- In connection with the real experiment in paper A, the fourth contribution is to compare the energy use of the simulation models with the real energy use. This is done to verify that the energy models in the simulation are accurate enough for energy reduction purposes.

- The fifth contribution in paper D is to use a simple model of a robot station to investigate the benefits of using online energy reduction methods as opposed to offline energy reduction methods.

- The sixth contribution in paper E is to model a real example of a flexible kitting robot station and to present an online control architecture capable of controlling it in a robust and energy optimal way, including learning of unknown model parameters.

1.5 Thesis Outline

The rest of the thesis is organized as follows. Chapter 2 contains the basic definitions and models of an industrial robot, robot station, and production line that will be used in the rest of this thesis. Chapter 3 presents methods for reducing the energy use of a deterministic robot station, a stochastic robot station and a production line. In Chapter 4 the methods developed in Chapter 3 are tested on either simulated or real examples of robot stations or a production line and the results of these experiments are presented. In

Chapter 5 an example of a flexible kitting station is presented, a method to control it in an energy-efficient way is developed, and an experiment to verify the method is shown. In Chapter 6 the included papers are summarized and in Chapter 7 conclusions and future work are presented.

CHAPTER 2

Robot Station Model

This chapter presents the necessary background and models of industrial robots, robot stations, and production lines required for the rest of this thesis. It is a summary of the material presented in Paper A, B, and C, together with some additional background.

2.1 Robot Model

This section describes the basic energy model of a typical industrial robot (as seen in Fig. 2.1) [26], [36], [37]. A standard model for the mechanical part of a robot with n joints is

$$\tau = A(\mathbf{q})\ddot{\mathbf{q}} + J(\mathbf{q}, \dot{\mathbf{q}}) + Q(\mathbf{q}) \tag{2.1}$$

where τ is a vector of joint torques, A is the inertia matrix, J is a vector of Coriolis and centrifugal torques, and Q is a vector of gravity torques. A, J, and Q are functions of the geometric and inertial parameters of the robot. $\mathbf{q}$, $\dot{\mathbf{q}}$, and $\ddot{\mathbf{q}}$ are vectors containing the joint angles, angular velocities, and angular accelerations respectively of each of the n joints.

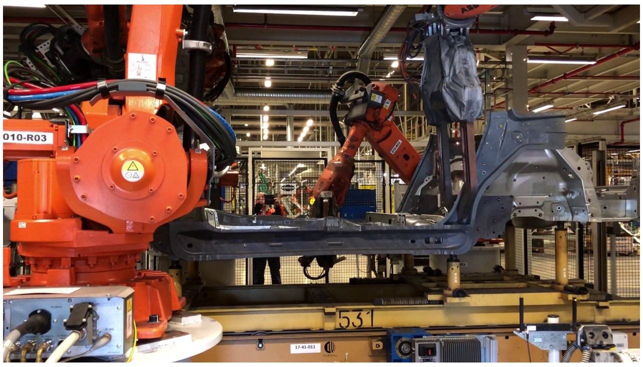


Figure 2.1: A welding station from Volvo Cars.

The mechanical part of the robot is powered by an electrical system. In short, the electrical system is supplied with energy by a three-phase AC system, which is transformed to DC by a rectifier. Connected to the DC-side are n number of Permanent Magnet Synchronous Motors (PSMSs), to power the robot joints; a capacitor, that can supply energy to the DC-bus or store energy recuperated by the motors through deceleration; and a "bleeder", that can dissipate excess energy. Apart from the useful energy to move the robot joints (2.1), the supplied energy also has to account for losses in the rectifier, electrical motors and losses due to friction in the joints. The electrical system also powers the robot control system and brakes for the joints. The brakes are normally closed, but during movement and for some time afterwards, they are kept open by a constant supply of power. Additionally, the robot tool may also use a significant amount of energy, e.g. the welding tool of a welding robot. In this context, it can also be mentioned that the lighting and ventilation of the factory are big users of energy [38]. However, in this thesis, the focus is on the energy use related to robot movements.

2.2 Robot Control System

The robot control system is what controls the behaviour of the robot. The user typically programs the behaviour of the robot control systems by specifying sequences of instructions. This is normally done in simulation software which contains a model of the robot, such as RobotStudio from ABB, KUKA.sim or Process Simulate. An example of a series of instructions written in the RAPID programming language from ABB is, [39],

```
AllocateZone Z1
MoveL p1, v2500, z100
MoveL p2, v2500, z100
SpotL p3, v100, spot1
AllocateZone Z2
MoveL p4, v2500, z100
SpotL p5, v100, spot2
ReleaseZone Z1
ReleaseZone Z2
```

`AllocatZone` and `ReleaseZone` are used to handle shared zones. A shared zone is a physical space that only one robot has access to at a time, to prevent collisions between robots. The first `MoveL` command means that the Tool Center Point (TCP) moves linearly to location `p1` with a speed defined by the *speeddata* `v2500` (maximum linear velocity of $2500\,\mathrm{mm\,s^{-1}}$). `z100` describes how close to `p1` the robot can start deviating to better transition into the next instruction. `SpotL` describes a linear motion ending in a weld. Based on these instructions the robot control system generates a trajectory, which is the joint angles $\mathbf{q}$. So, attempts at reducing the energy use of industrial robots are somewhat limited by the fact that $\mathbf{q}$ is generated by the robot control system. There exist methods to run user-generated trajectories using for example the Fast Robot Interface (FRI) from KUKA or Externally Guided Motion (EGM) from ABB. However, in the author's experience, these are not commonly used in industry as they to some extents are less robust, making them unsuitable to use in many types of critical manufacturing processes.

2.3 Station Model

In this section, a model of a robot station is presented. To complete the task assigned to the station, the robots can execute a set of *operations* $\mathcal{I}$. An operation may for example be a robot movement, performing a weld on a car body or picking a component from a box. For the example of robot code above: the two moves to the first welding location can be considered as one operation and performing the actual weld can be considered as another one.

Each operation i has a starting time S_i, an execution time (duration) D_i, and a completion time C_i, which all in general can be stochastic. In this model, the execution time can be divided into a deterministic part d_i, which determines the energy use, and an additive stochastic part E_i. The stochastic part is assumed to be, or possible to approximate as, normally distributed. The magnitude and type of stochastic variations differ depending on the type of operation, with some operations being completely (within the margin of error) deterministic. For example, a robot movement typically doesn't have much variation, while operations performed by human operators, welding, and cleaning of welding tools do. In Fig. 2.2 the distribution of the execution time of a welding robot is shown. The discrete jumps in the distribution are due to different levels of cleaning of the welding tool (no, small, and big) and the continuous distribution around each peak most likely is due to variations in each of the around 20 welding operations performed by the robot. The variation of a single welding operation may depend on several factors, one being that "dirt" accumulates on the welding tool which increases the time to perform a successful weld (which is why the tools need to be cleaned).

An operation can be modeled as an Extended Finite Automaton (EFA) [40], [41], where the operation may have a number of preconditions. The preconditions are logical constraints that need to be fulfilled for the operation to execute. One of the simplest preconditions can be named *precedence constraint* and occurs when there is a fixed order in which some of the operations need to be executed. The *operation order* is represented by $\mathcal{A}$. Let $\mathcal{W}_i^{\mathcal{A}}$ be a set of operations that need to precede operation i for operation order $\mathcal{A}$. The

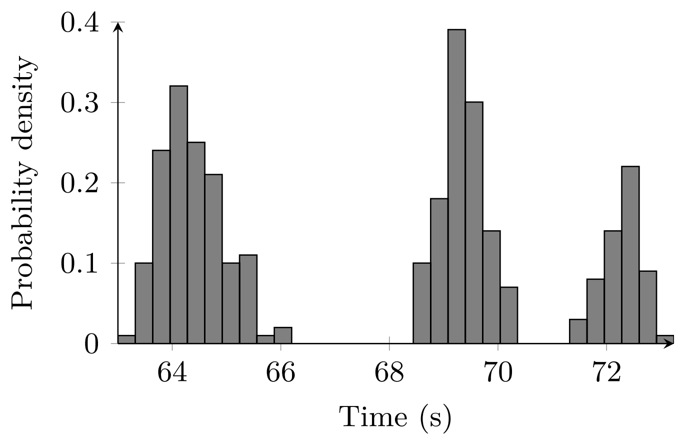


Figure 2.2: The distribution of the execution time of a welding robot in an automotive factory.

precedence constraint for each $\ell \in \mathcal{W}_i^{\mathcal{A}}$ can then be expressed as

$$S_i \geq C_\ell \tag{2.2}$$

where the $\geq$-sign in the expression above and in the rest of this thesis represents greater than or equal "almost surely" [42], i.e. $\mathrm{P}(S_i \geq C_\ell) = 1$, where P is used to denote probability. A difference can be made between necessary precedence constraints, e.g. a component must first be placed before it can be welded into place, and precedence constraints determined by the control strategy that are not necessary for the functionality of the station but that enforces an advantageous behaviour of the station.

The operations require some *resources* to execute, which are contained in the set $\mathcal{R}$. Examples of resources are robots, human operators, tools, or shared zones. Each operation requires at least one resource to execute but it may be more, and each resource can only be used by one operation at a time. For any resource $r \in \mathcal{R}$ the *resource constraint* states that for any two operations i and ℓ requiring resource r to execute it must hold that

$$S_i \geq C_\ell \quad \text{or} \quad S_\ell \geq C_i \tag{2.3}$$

A robot station with precedence and resource constraints can be expressed as a disjunctive graph [43], where the operations are the nodes, precedence constraints are directed edges, and resource constraints between two operations of the same resource are represented by undirected edges. Before the execution of the station, the operation order of all operations using the same resource must be determined. This means replacing each resource constraint formulated above with precedence constraints. In other words, to decide the direction of the undirected edges. This results in a Directed Acyclic Graph (DAG).

The *makespan* of a station is in general a stochastic variable denoted by T^{make} and is defined as

$$T^{\text{make}} = \max_{i \in \mathcal{I}}(C_i) - \min_{i \in \mathcal{I}}(S_i)$$

It is a measure of how long it takes for the station to execute the operations. Assuming that the first operation starts at time $t = 0$, T^{make} simplifies to $T^{\text{make}} = \max_{i \in \mathcal{I}}(C_i)$. As discussed in Chapter 1, an important conflicting interest to energy reduction is *productivity*. Productivity is loosely defined here as the number of produced products or completed tasks per unit of time. To ensure that the productivity is not reduced, the makespan can be constrained by a chance constraint [35] to meet a *deadline* t_d as,

$$\mathrm{P}(T^{\mathrm{make}} < t_d) \geq \gamma \tag{2.4}$$

where γ is close to, or precisely, 1.

Based on the real stations used as case studies for this thesis, the robots in a station are often programmed with the maximum velocity, which uses a lot of energy. This is reasonable if the production only consists of one robot because then the productivity is directly related to the velocity of the robot. However, for a robot station with multiple robots, the precedence constraints between them mean that it is not always necessary for all robots to move with the maximum velocity. Therefore, the concept of *slack* can be introduced [30], [43]. Essentially, the slack is a measurement of how much idle time for the robots there is in the schedule. Slack can be used to quantify how much operations can be delayed without extending the total execution time of the station or production line.

Based on the real robot stations there often is a significant amount of slack, which allows the velocity to be reduced and thereby also the energy, without reducing the productivity of the station. An illustrative example of a robot station containing precedence constraints, resource constraints, and slack is shown in Fig. 2.3.

2.4 Production Line

In many production systems, the robot stations are ordered in production lines. How exactly the production line is configured can vary, what is presented below is based on observations from the automotive factory presented later in Chapter 4. The production line consists of a sequence of robot stations with buffers before and after (see Fig. 2.4). The production line works in cycles and is programmed so that all stations start with the next cycle at the same time. t^{aim} is the desired *cycle time* of the production line and the next cycle will not start before that. The value of t^{aim} is determined to keep the

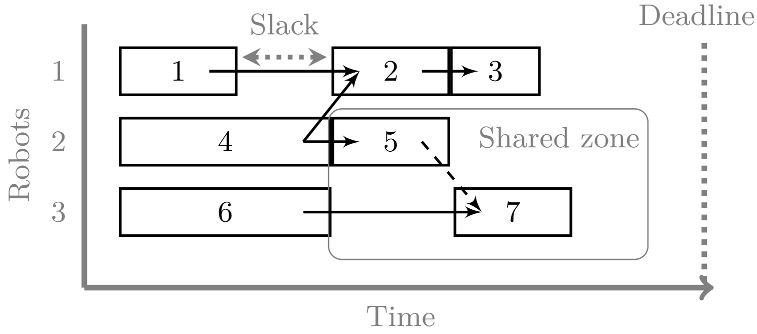


Figure 2.3: An illustrative example schedule of a robot station, containing a set of operations and resources. The operations are shown as numbered boxes and the solid arrows between them are necessary precedence constraints. The station contains four resources: three robots and one shared zone. The robots are listed to the left. The operations on the same row as a robot require that robot to execute. Operation 5 and 7 also require the shared zone to execute. Only two operation orders are possible: the one where Operation 5 precedes Operation 7 (which is shown) or the other way around. The additional precedence constraint determined by the control strategy and related to the shared zone is shown with a dashed arrow.

required productivity of the production line. The actual cycle time T^{cycle} is a stochastic variable and can be expressed as

$$T^{\text{cycle}} = \max\left(\max_{z\in\mathcal{Z}} T_z^{\text{make}}, t^{\text{aim}}, T^{\text{outer}}\right) \tag{2.5}$$

where $\mathcal{Z}$ is the set of stations in the production line and T_z^{make} is the makespan of station z. T^{outer} represents delays caused by disturbances in other parts of the production.

The makespans of two stations can be seen in Fig. 2.5 together with the desired and actual cycle times. A large part of the values of the actual cycle

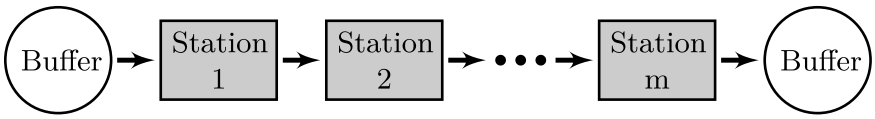


Figure 2.4: An automotive production line

times are close to the desired one, but there is a considerable amount that is much larger. These values are most often due to outer disturbances. The two stations shown in Fig. 2.5 are one of the ones with the lowest makespan and one of the ones with the highest makespan. It can be noted that there is a large amount of slack for Station 9.

2.5 Summary

In this chapter, the most relevant theoretical background for the topic of energy reduction of industrial robot stations has been presented. This includes models of a robot, station, and production line, as well as a short analysis of the energy reduction problem.

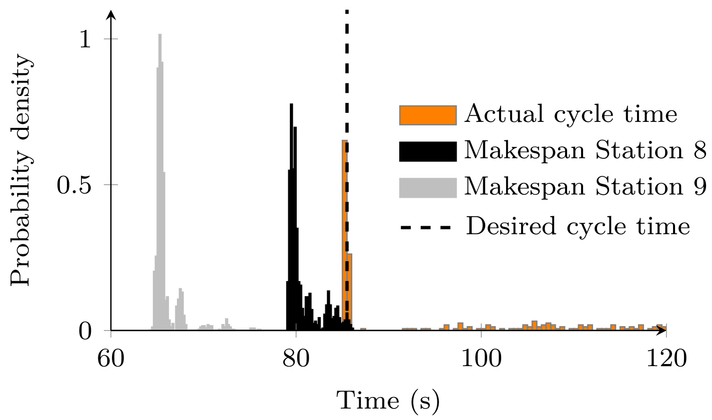


Figure 2.5: The distributions of the makespans of two stations together with the actual and desired deadline for a production line in an automotive factory.

CHAPTER 3

Offline Energy Reduction Method

This chapter presents an offline method for reducing the energy use of robot stations and production lines. Offline in this context means that changes to the stations can only be made before the execution. The method is based on the particular case studies presented later in Chapter 4 but can be used for other types of robot stations as well. This chapter is a summary of Papers A, B, and C, as well as some additional content to put the work into perspective.

3.1 Method Overview

This section briefly describes the necessary steps to achieve the desired energy reduction of the robot, station, or production line. The first step is to collect the necessary data, such as robot models, robot code, and data about the execution times. The execution times are collected during multiple cycles and are used to model the stochastic variations in the station. The robot code is analyzed to identify the operations and resources, which then can be translated into precedence and resource constraints. The robot station is then built in RobotStudio, which contains energy models of the robots, and simplified energy models of the robot movement operations are found. With the identified operations, constraints, and energy functions, an optimization problem is formulated and solved. Based on the optimized solution the robot code is modified and implemented back in the station. More details about the energy models and optimization are given below.

3.2 Energy Model

As already discussed, there are some practical limitations to finding the mechanical and electrical parameters required to create the full energy model described in Section 2.1. This in combination with limitations in the robot control system, which limits the freedom of designing the robot trajectory, leads to the decision of describing the energy use of the robots using energy signatures, which has shown to be effective [19], [20], [25], [26]. An energy signature g_i of a movement operation i is a function of the execution time d_i. This allows the energy use of a robot to be described for every operation separately, without the need for a complete energy model of the robot and does not require changing the path of the robot movement. The energy functions are parameterized as

$$g_i = \psi_1^i \exp(\psi_2^i d_i) + \psi_3^i \exp(\psi_4^i d_i) \tag{3.1}$$

where $\psi_1^i, \psi_2^i, \psi_3^i$, and ψ_4^i are parameters of g_i. In this thesis the energy functions are found by modifying motion parameters: linear velocity, angular velocity, acceleration, and jerk (time derivative of acceleration) of the robot movements; details can be found in Paper A. The velocities are modified by changing the `speedata` setting, and the acceleration and jerk are modified by adding the command `AccSet` before a robot instruction. In Paper A a comparison is made between the different motion parameters. The overall result is that linear velocity and angular velocity are most effective in terms of energy reduction. However, between individual operations, there is a large variation in terms of which motion parameter is the best. Examples of energy functions for robot movements can be seen in Fig. 3.1 and it can be noted that they are convex. $\underline{d}_i$ is used to denote the shortest possible execution time of operation i. As already mentioned, robots are often programmed with $d_i = \underline{d}_i$.

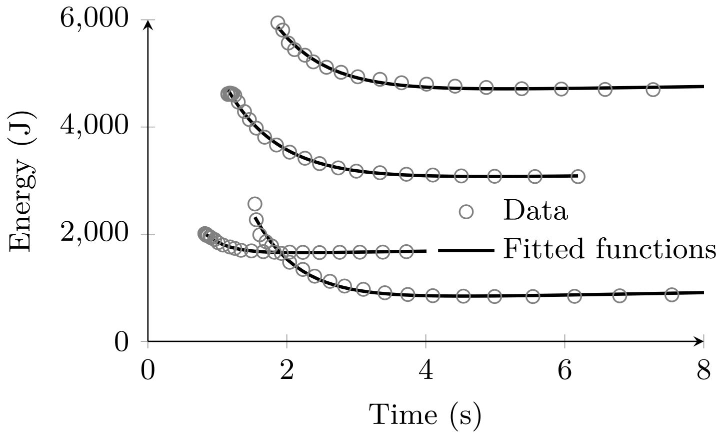


Figure 3.1: Examples of the energy data and fitted functions of four operations.

3.3 Single Station Optimization

The energy optimization problem for a single robot station with precedence constraints, resource constraints, and the energy signatures (3.1) can, for illustrative purposes, be described as the following stochastic programming problem [35], [44]–[46]:

Optimization Problem 1

$$\min \sum_{i \in \mathcal{I}} g_i(d_i)$$

subject to:

$$C_i = S_i + E_i + d_i \qquad i \in \mathcal{I} \tag{3.2}$$

$$S_i \geq \max_{\ell \in \mathcal{W}_i^{\mathcal{A}}} (C_\ell) \qquad i \in \mathcal{I} \tag{3.3}$$

$$T^{\text{make}} = \max_{\ell \in \mathcal{L}^{\mathcal{A}}} (C_\ell) \tag{3.4}$$

$$\mathrm{P}(T^{\text{make}} < t_d) \geq \gamma \tag{3.5}$$

$$\underline{d_i} \leq d_i \qquad i \in \mathcal{I}$$

$\square$

where $\mathcal{L}^{\mathcal{A}}$ is the set of operations that for operation order $\mathcal{A}$ have no operations succeeding them on any of the resources. The decision variables at this point are S_i, $\mathcal{A}$, and d_i.

Optimization Problem 1 is both a scheduling problem of determining the operation order and a timing problem of determining d_i. It can be classified as a chance-constrained programming problem [35], and a stochastic and project scheduling problem [42], [47]. The combination of the scheduling part and the stochastic variables makes Optimization Problem 1 hard to solve to optimality in a reasonable time, except for trivial examples [35]. Examples of commonly used stochastic optimization methods to solve such problems are simulated annealing, genetic programming, particle swarm, and ant colony optimization. [48]–[51]. Many solution methods to chance constraint programming problems involve reformulating [35] the problem into a deterministic mathematical optimization problem [31]. Typically, in a way that the solution to the reformulated problem is feasible for the original problem as well. Although, this does not guarantee optimally. Commercial solvers for stochastic programming problems are for example SAMPL ("Stochastic AMPL").

In this thesis, Optimization Problem 1 will be approximated as a deterministic mathematical optimization problem, modeled in the optimization tool CasADi [52], and solved using IPOPT [53] together with the Branch and Bound algorithm of BONMIN [32]. BONMIN solves the problem to optimality in the case that the problem is a convex Mixed-Integer Nonlinear Programming (MINLP) problem [34], i.e. the LP-relaxation of the optimization problem is a convex optimization problem. In the case that the problem is nonconvex the solution from BONMIN is a heuristic. Other solvers exist [33] such as SCIP, Couenne, and BARON. Exactly what type of solution method, solver, and level of simplification of Optimization Problem 1 required depends on the type of problem, size, available solving time, and the requirement on the goodness of the solution. What is presented in this chapter are approaches that proved to be suitable for the problems considered in this thesis.

Mean Approximation

A trivial simplification of Optimization Problem 1 is to use the expected values of S_i, E_i, and C_i and formulate it as a convex MINLP using the Big M Method [54]. This includes both the scheduling and the timing part of Optimization Problem 1 and can be solved efficiently to optimality using BONMIN. The complete optimization formulation is presented in Paper A and using the solution to that problem will be referred to as the *mean approximation* method. This simplification is accurate assuming that the stochastic variance of E_i is negligible. If not, the following can be shown: let T^{make} be the makespan of an actual robot station and t^{make} be the corresponding approximate version in the mean approximation method. It has been shown by Möhring [55] that $T^{\mathrm{make}} \geq t^{\mathrm{make}}$ and that the difference between them can grow arbitrarily large with increasing numbers of operations or variance of the operations execution times. This means that (3.5) likely will not hold if the variance of E_i is significant.

Normal Approximation

In Paper B a less trivial approximation of Optimization Problem 1 is presented. It is based on approximating the distributions of the stochastic variables in Optimization Problem 1 as normal and is summarized below. The approximation will first be done under the assumption that there is a fixed operation order, focusing on the timing optimization.

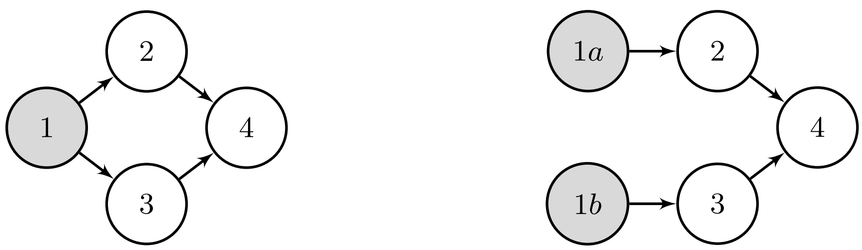
Timing Optimization

With a fixed operation order the system can be described by a DAG [55]. It can be concluded that the optimal solution to Optimization Problem 1 is where equality holds for (3.3). For any solution to Optimization Problem 1 where equality does not hold for one i, (3.3) can be written as $S_i = \Delta_i + \max_{\mathcal{W}(\mathcal{A})_i}(C_i)$, where $\Delta_i > 0$. Then it is possible to find another feasible solution that is equal to the first one in every sense except that $\Delta_i = 0$. This new solution will have the same value of the cost function and the probability of meeting the deadline will be at least as high.

In general the distributions of S_i, E_i, C_i, and T^{make} are not normal; even if every execution time E_i is, the starting time S_i is not (because of (3.3)). Nevertheless, they will be approximated as such, because it reduces the complexity of the problem [35], and at the same time is accurate enough to be usable in practice. With that assumption, evaluation of the sum in (3.2), which in general is too complicated to be practically feasible, because it involves the solution of an integral, is simplified to the sum of the expected values and variances of S_i, E_i, and d_i [56]. Regarding the accuracy of the approximation: for the robot stations considered in this paper, it is often the case that one of the C_ℓ in (3.3) is significantly larger than the others, which means that $S_i \approx C_\ell$. So, the whole schedule reduces to a series of sums of E_i. This in combination with the central limit theorem [57] means the approximation holds, as will be seen in Section 4.2.

The stochastic variables C_ℓ involved in (3.3) for an i are not independent of each other but will be approximated as such. Looking at Optimization Problem 1, the chance constraint (3.5) is essentially the max function of a sum of E_i. Depending on the precedence constraints in the station, an E_i may be involved in more than one sum. To make C_ℓ in (3.3) independent of each other, every time an E_i would appear more than once, a duplicate with the same distribution but that is independent of the first one is used instead; an illustration of this is shown in Fig. 3.2. This approximation makes the problem easier to solve and it can be shown that the solution to the approximate problem will fulfill (3.5) in the original problem formulation as well [55].

The final step is to find a function that approximates the max-functions of Optimization Problem 1 under the assumption that the stochastic input variables are independent and normally distributed. This is done by first analytically solving the integral for the case with two input variables. The resulting nonconvex function can then approximately be applied recursively to the case with multiple input variables. These three steps result in Optimization Problem 4 in Paper B. It is a nonlinear and nonconvex optimization problem, but still is possible to solve using BONMIN for the problem sizes



(a) Original system **(b)** Independent system

Figure 3.2: An illustration of the independence assumption. The figures represent operations and precedence constraints. Both $1a$ and $1b$ have the same distribution of their execution times as 1 but are otherwise independent. The completion times of operation 4 are equal to $C_4 = \max(E_1{+}E_2, E_1{+}E_3){+}E_4$ and $\widehat{C}_4 = \max(E_{1a}{+}E_2, E_{1b}{+}E_3){+}E_4$ for the two examples respectively. It can be shown that $\mathrm{P}(C_4 < t_d) \geq \mathrm{P}(\widehat{C}_4 < t_d)$.

considered in this paper. Using this solution will be referred to as the *normal approximation* method.

Operation Scheduling

To include the operation order into the optimization formulation just described there are some options, which once again depend on the problem size, available time, and required goodness of the solution. One option is to find the operation order by solving a separate optimization problem, that does not include the energy use or stochastic variations, and then use that solution to solve the normal approximation problem. This could be done iteratively, to hopefully converge to the true optimum. An example of this would be to use the mean approximation method to find an operation order to use in the normal approximation method. This will be referred to as the *mean + normal* method.

Based on the fact that the goal of the optimization is to minimize both the energy use and the probability of meeting the deadline, another approach is to find the operation order by minimizing the expected makespan ($\min_{i \in \mathcal{I}} \mathrm{E}[C_i]$)[42] since there is a correlation between a low makespan and a high probability of meeting the deadline. This problem is more common in the literature and can be solved more efficiently than the complete problem since it can be formulated as a Mixed-Integer Linear Programming (MILP) problem. Using the minimum makespan operation order in the normal approximation method will be referred to as the *makespan + normal* method.

However, solving the operation scheduling and the timing optimization separately may result in the solution being suboptimal. In paper B an approach to include the operation order into the normal approximation is presented. It is a nonconvex MINLP that essentially includes all precedence constraints for every possible operation order and then uses a set of discrete decision variables that enables or disables them. It increases the complexity of the problem significantly but can be solved for small problem instances. This will be referred to as the *complete* method.

3.4 Production Line Optimization

Energy optimization for a whole production line of the type described in Section 2.4 is done in detail in Paper C and presented in short here. Since the conflicting objectives are energy reduction and productivity, a first step is to investigate if it is possible to improve the cycle time T^{cycle}, which potentially leaves more room for energy reduction.

Cycle Time Optimization

In the list below some different cycle time policies are described. This is under the assumption that the buffers before and after the production line are ideal, i.e. they have unlimited storage capacities and never run out of products. This assumption is made to make the analysis possible without having to analyze T^{outer} in detail.

- **Policy 0:** This is the policy that is used in the investigated production lines, that waits for t^{aim} before starting with the next cycle. $T_s^{\text{cycle}} = \max(\max_{z\in\mathcal{Z}} T_z^{\text{make}}, t^{\text{aim}})$.
- **Policy 1:** In Policy 1 the wait for t^{aim}, which seems unnecessary, has been removed. The cycle time would then be $T^{\text{cycle}} = \max_{z\in\mathcal{Z}} T_z$.
- **Policy 2:** In Policy 2 every station operates independently. Each station starts on the next cycle as soon as possible, i.e. when the next station in line can accept the finished product and the previous station can forward the next one. For this policy, the cycle time of the production line is defined as the time between two finished consecutive products for the last station.

Energy Optimization

To reduce the energy use of the production line, each station can be optimized separately using any of the methods previously described in this chapter, but with some modifications of γ_z (γ for station z). For a whole production line to meet a deadline with probability γ, γ_z of the individual stations need to be higher. The following relation defines a lower bound on γ_z for this requirement to hold [45].

$$\gamma \leq 1 - \sum_{z\in\mathcal{Z}} (1 - \gamma_z)$$

The deadline t_d in (3.5) is typically chosen as $t_d = t^{\mathrm{aim}}$. However, if the cycle time policy manages to reduce the cycle time, then t_d can be chosen higher, leaving more room for energy reduction while keeping a high productivity.

3.5 Summary

Related to **RQ1**, this chapter has presented a method for reducing the energy use of robot stations and production lines. The method involves: modeling the robot stations in simulation; identifying operations and constraints; obtaining simplified energy models; and formulating and solving an optimization problem. For a single robot station, it was noted that the true optimum to the optimization problem cannot be easily found. Several simplifications were presented, allowing the optimization problem to be solved. For the whole production line, several cycle time policies were presented, that are aimed at reducing the cycle time and thereby leaving more room for energy reduction. Then it was shown how the energy use of a production line can be reduced.

CHAPTER 4

Experiments and Results

This chapter contains descriptions and results of the experiments conducted to verify that the proposed method described in Chapter 3 works as intended and to investigate the potential for energy reduction in a typical automotive factory today.

4.1 Real Deterministic Experiment

This section describes an experiment on the station shown in Fig. 2.1, which is a welding station from Volvo Cars in Gothenburg; full details can be found in Paper A. The station is part of a production line that produce car bodies, and the cycle time is around one minute. In the station, there are four robots from ABB of the type IRB6640, each one equipped with a spot-welding tool. In every cycle, each robot performs a series of spot welds on the car body. There are four shared zones, that prevent the robots from colliding. The station also contains precedence constraints between every consecutive operation of one robot. Meaning that the only operation order that is not fixed, that is left to determine by the optimization, is in which order the robots access the shared zones. The operations are either movement operations, which use energy, or welding operations, that have stochastic execution times and whose energy use is not included. However, for simplicity, the stochastic variations will not be included in the optimization. In total each robot executes around 20 operations per cycle (see Fig. 4.1).

Experimental Setup

First, the robot station is optimized with the mean approximation method using the maximum makespan of the unoptimized station as the deadline. The result of the optimization is then implemented in the robot code, which means adjusting the velocity settings for each operation. An experiment is then performed using the optimized robot code to compare it to the original robot code. This is partly done to see how much energy reduction can be made but also to verify the accuracy of the energy models used during the optimization. Three different types of energy data are collected from the station during the experiment (see [58] for reference). *Simulated data*: this is the standard data obtained when simulating in RobotStudio. The energy use is based on a "nominal robot operating under typical conditions". This type of data is also what is used to find the energy functions. *Torque-based data*: by having live connections to the robots, the power and energy use can be more

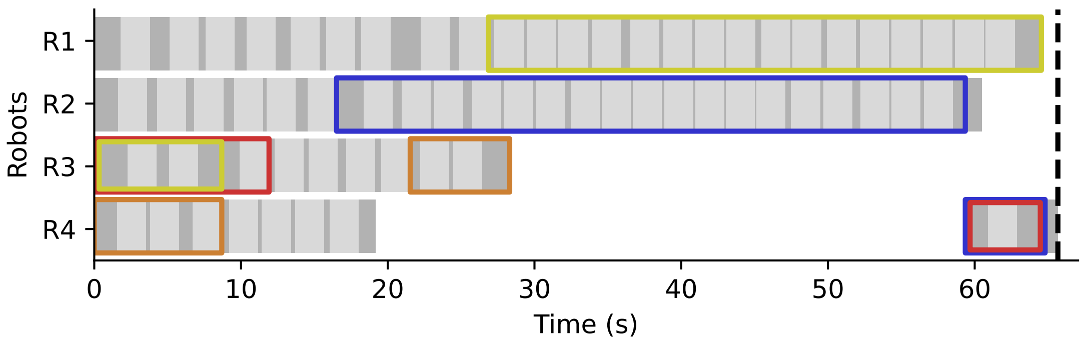


Figure 4.1: A time schedule of the unoptimized welding station, showing the four robots and the execution times of their respective operations. The welding operations are light gray, movement operations are grey, and the deadline is shown as a black dashed line. The colours indicate resource constraints between sequences of operations from different robots. Two sequences of operations with the same colour cannot be executed at the same time.

accurately predicted based on torque data from the real robots. The simulated and torque-based data are "measured" on the DC-side before each motor (cf. Section 2.1). *Measured data*: current and voltage measurement equipment are connected to the three-phase AC system that supplies the robot cabinet of one of the robots. The total power and energy use are found by measuring the current and voltage at a rate of 10 000 Hz.

Results

The results are divided into two parts. The first part is a comparison between the three data sets and the second part is a comparison of the energy use between the original and optimized settings.

Verification of Energy Models

In Table 4.1 the peak power, energy use, and energy reduction of the three data sets of one robot are shown. The peak power is of interest because it determines how powerful electrical components must be used when setting up a robot station. Using less powerful components means that money can be saved. The energy use for the simulated and torque-based data is calculated without including the negative parts of the power use. That is, recuperation between the motors and the capacitor is not included. The peak power is calculated as the maximum power of one cycle. The measured data only includes the energy use to power the robot, not the idle energy use of the brakes and control system.

Table 4.1: A comparison between the simulated, torque-based, and measured power/energy use of Robot 4.

		Simulated	Torque-based	Measured (-idle use)
Peak power (kW)	Original	13.0	13.6	14.3
	Optimized	10.3	10.9	11.2
Energy use (kJ)	Original	29.4	30.2	30.5
	Optimized	24.6	23.8	25.0
Energy reduction (%)		16.3	21.0	18.0

The result shows that when disregarding the idle energy use the simulated and torque-based data slightly underestimate the measured data but is otherwise fairly accurate. To conclude, although some differences can be seen between the simulated and torque-based data compared to the measured data, the accuracy of the total energy use presented in Table 4.1 justifies the use of the simulated energy use in the optimization (which already has been presented), and the use of the torque-based energy use during calculations of energy savings (which will be presented in the next section).

Energy reduction

In this section, the result of the whole station is presented. The values presented are the average of the torque-based data from 44 cycles. A detailed Table of the energy use, peak power, and jerk can be found in Paper A, but in short, the results are as follows. The biggest energy reductions are for Robots 3 and 4 with 33 % and 20 % in energy reduction respectively. This is expected because they had more amounts of slack than the other robots before the optimization. During the optimization, this slack has been used to lower the velocities of their operations and in turn reduce their energy use (see Fig. 4.2). As can be seen in Fig. 3.1, the energy functions have minima, which means that at a certain point a larger amount of slack will not result in more energy reduction. That is why Robot 3 has more energy reduction than Robot 4 even though Robot 4 has a larger amount of slack. For the station

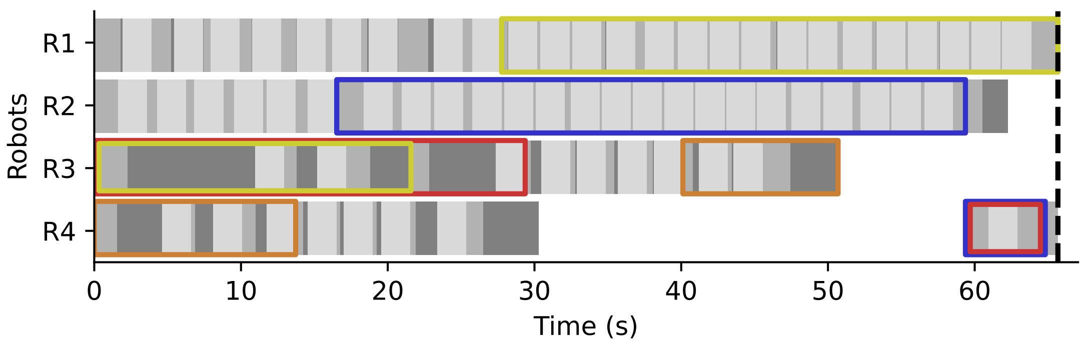


Figure 4.2: A time schedule of the optimized welding station. The darkest gray areas show the execution time extensions (compared to $\underline{d}_i$) after the optimization. Otherwise, the notations are the same as in Fig. 4.1.

as a whole, the result shows a reduction of 12 % in energy use between the original and optimized settings.

The energy reduction is the most important outcome of this experiment, but the peak power and jerk are also of interest. Looking at Fig. 4.3 the power peaks for Robots 3 and 4 are significantly lower after the optimization. For the station as a whole, the peak power is reduced by 12 %.

Fig. 4.4 shows a comparison of velocity, acceleration, and jerk between the original and optimized settings for one operation. It can clearly be seen how the velocity is limited and thereby the acceleration and jerk are also reduced. The jerk is calculated as the mean of the absolute jerk of the TCP. The jerk is reduced by 75 % for Robot 3 and by 25 % for the whole station. Reducing

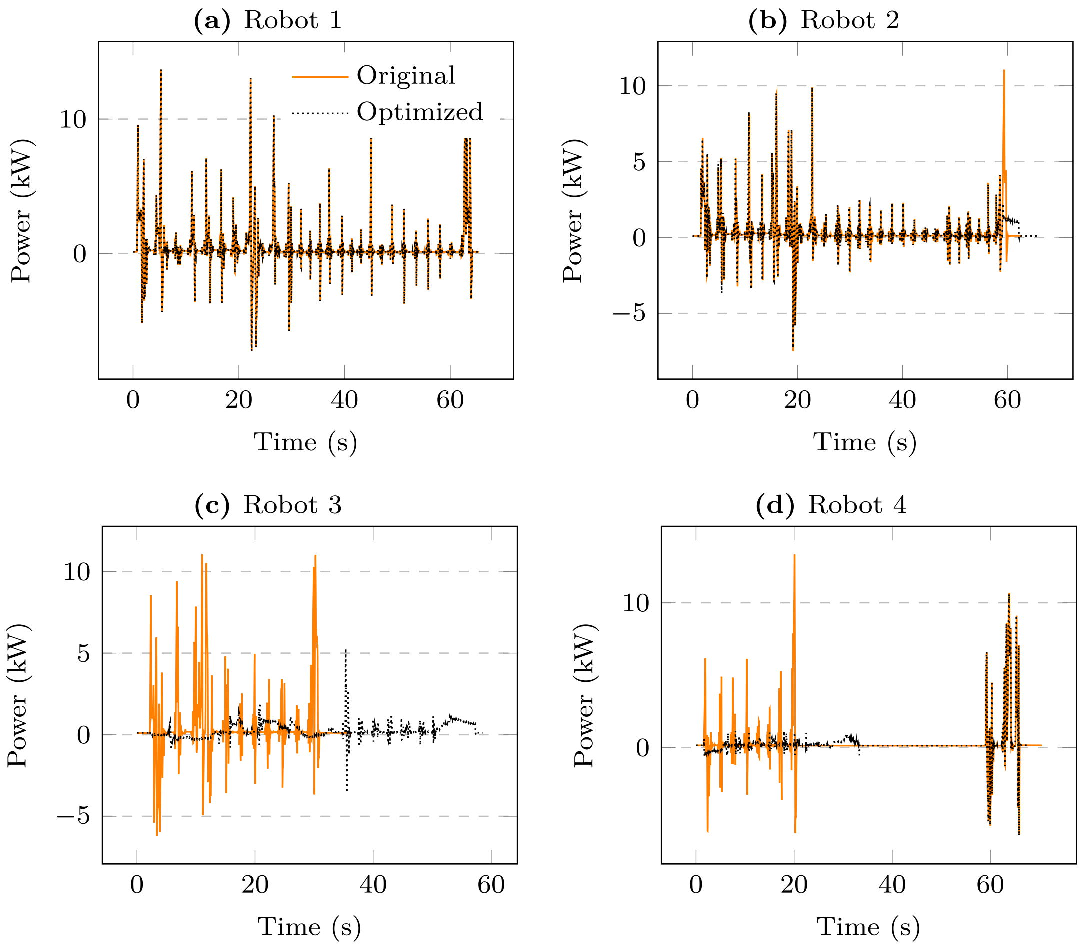
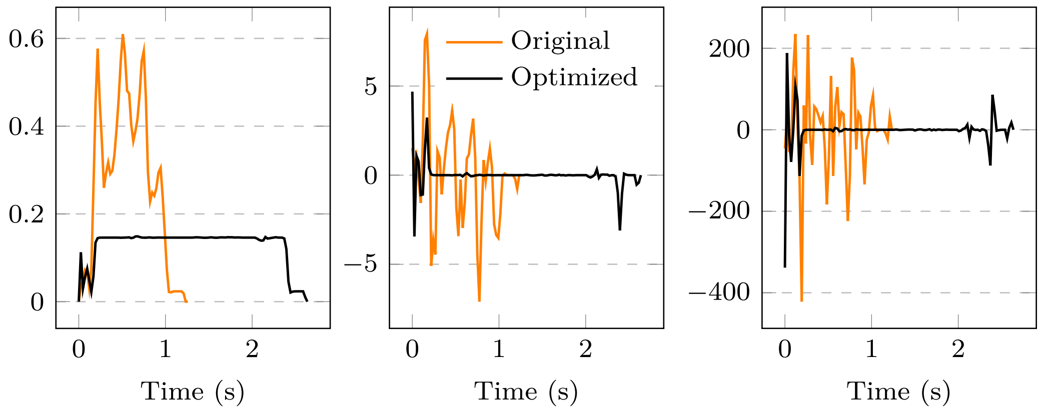


Figure 4.3: The power use of the original and optimized settings for the four robots.



(a) Velocity (m/s) **(b)** Acceleration (m/s^2) **(c)** Jerk (m/s^3)

Figure 4.4: The velocity, acceleration, and jerk of one operation with the original and optimized settings

the jerk increases the precision of the robot movements [59] [60], but it is also believed that there is a correlation between jerk and the wear and tear of the dress packs attached to the robots. The dress packs contain cables and hoses required to operate the welding tool. Sometimes they break and must be replaced. The stress that they experience depend on many factors [61], but it is probable that there is a correlation with the aggressiveness of the robot movements.

4.2 Simulated Stochastic Stations

This section is a summary of the experiments from paper B and describes the experiments on three welding stations of a similar type as presented in Section 4.1 but that is part of another production line. The main difference is that these experiments are done only in simulation and that the stochastic variations of the welding operations are included in the optimization. The use of simulation rather than real tests is motivated by the accuracy of the simulated energy use, as was verified in Section 4.1.

Experimental Setup

A schedule of one of the stations can be seen in Fig. 4.5, and it can be noted that it contains a significantly smaller amount of slack than the station in

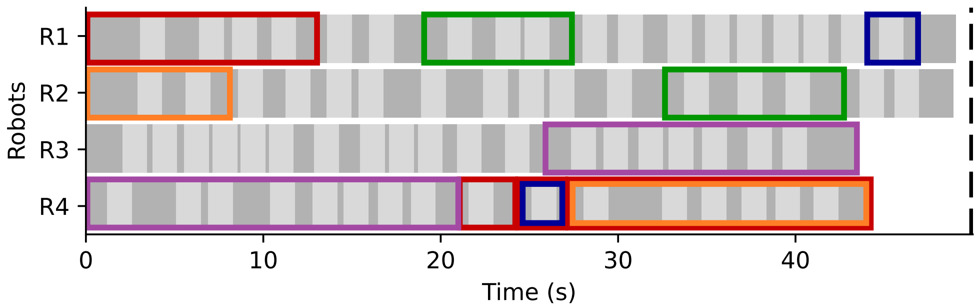
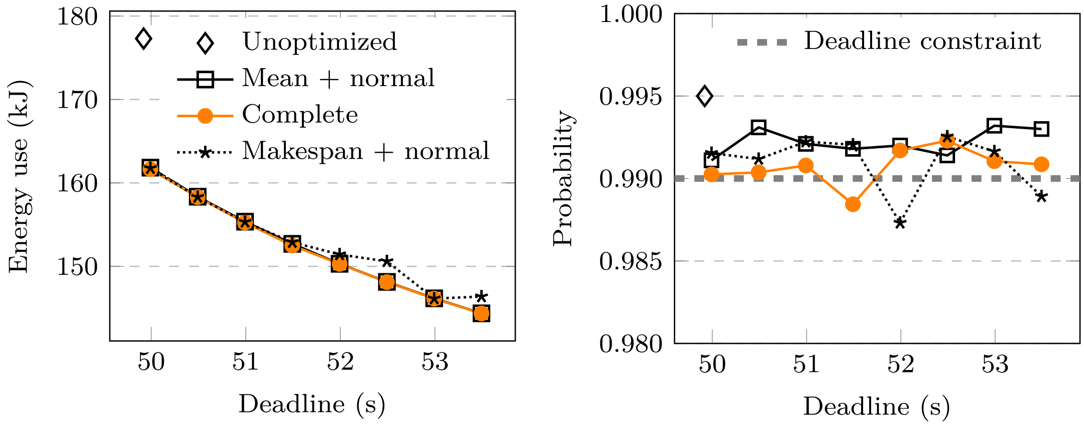


Figure 4.5: A time schedule of a robot station of the same type as shown in Fig. 4.1. The execution times of the operations shown in the schedule are the mean values for the welding operations and the lower bounds $\underline{d}_i$ for the movement operations.

Section 4.1, as is the case for the other two stations. The same type of energy functions are used for the movement operations but the welding operations are modeled as having uniformly distributed execution times $E_i \sim U(1.25, 1.55)$, instead of being constant. The experiment of each station is conducted by first determining the deadline $t_{d,0}$ so that they are met 99.5 % of the time for the original unoptimized stations. Each station is then optimized with $\gamma = 0.99$ for a number of $t_d \geq t_{d,0}$ using the optimization methods described in Section 3.3. After that, the resulting operation order and velocity settings are applied to the simulation models, and the stations are simulated again to determine the probability of meeting the deadline.

Results

The results presented here are of the station shown in Fig. 4.5, the results of the other two stations can be found in Paper B and overall the conclusions are similar. The results of the experiment can be seen in Fig. 4.6 and an optimized schedule can be seen in Fig. 4.7. The energy use of the unoptimized station, when $d_i = \underline{d}_i$, are 177 kJ. Comparing this with the optimized energy use shown in Fig. 4.6a, there are some things to note. Firstly, the result shows that significant energy reduction can be made, even with the lowest deadlines and even though the station has a significantly smaller amount of slack than the station in Section 4.1. The energy reduction with the lowest deadline is roughly 9 %. The reason why the reduction is so big despite the limited



(a) Station 1, energy use. **(b)** Station 1, probability to meet deadline.

Figure 4.6: The simulated energy use and probability to meet deadline for one of the three stations before and after optimization.

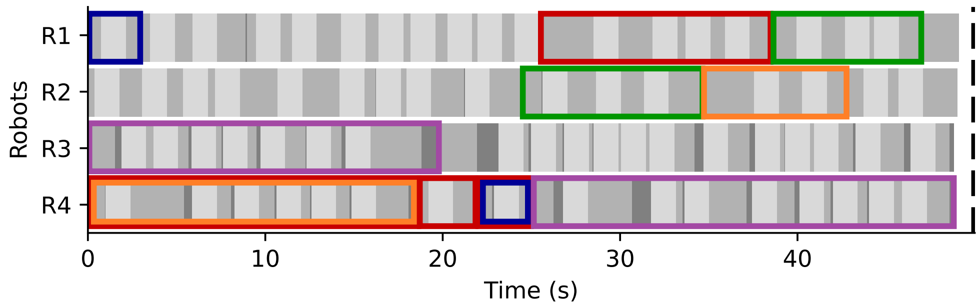


Figure 4.7: The optimized version of Fig. 4.5.

amount of slack is because of the steep gradients of the energy functions around $\underline{d}_i$ (see Fig. 3.1). The energy use of the station is further reduced by around 2-3 percent per additional second that the deadline is extended (compared to the unoptimized energy use). However, this effect is declining as the deadline is increasing.

Fig. 4.6b shows the simulated probability to meet the deadline. The result shows that the probability to meet the deadline is very close to, and most of the time above, the constraint of 0.99. This is a good result considering the heavy approximations used when deriving the optimization methods and the fact that the distributions of the execution times of the operations are not normal (as modeled) but uniform. This is partly because there are enough operations and the uniform distribution is close enough to the normal distribution for the central limit theorem to hold. Comparing the optimization methods: the mean + normal method almost reduces the energy use as much as the complete method, which is a positive result since the former is much less complex. The makespan + normal method sometimes finds the same solution but not always.

4.3 Simulated Production Line

This section contains experiments on a production line with 10 stations, one of which is the one in Section 4.1, including buffers before and after. The experiments are done in simulation and full details can be found in paper C. For this experiment, stochastic data about the execution times of the robot stations were obtained. This includes data about the movement and welding operations that were used in Sections 4.1 and 4.2 but also data about transportation and setup times between the stations, and cleaning of the welding tools (see Fig. 2.2).

Experimental Setup

It was not possible to obtain full station models of all 10 stations. Therefore, the station models from Section 4.1 and Section 4.2, named Station models A and B respectively, are used to represent all 10 stations. The energy functions, operations, and constraints of Station models A and B are combined with the real stochastic data about the execution times of the 10 stations to create the 20 simulation models used in this experiment. The simulation model of each station is adjusted so that the amount of slack is similar to the real station, to accurately represent the energy reduction potential in the real system. The experiment can be divided into two parts with the same experimental setup, one for each station model, and can be described as follows. The 10 stations are first optimized as described in Section 3.4. The results are implemented in the simulation models of the stations and the stations are then simulated to find the makespans of the optimized stations. The new cycle times are then calculated using Policies 0,1, and 2. For comparison, a default case is also included, where no optimization is performed and policy 0 is used to calculate the cycle time.

Results

The main results are shown in Table 4.2 for station model A. The data shown is the relative energy use, which is the optimized energy use compared to the unoptimized energy use, and the cycle time mean and variance (minus outliers). To put it into perspective, an unoptimized production line with 10 stations of Station model A operating continuously for 24 h would use around 271 kW h.

The results show that a significant energy reduction of 24 % can be made. Using policy 0 this energy reduction comes at the cost of a higher cycle time mean and variance compared to the default case. Policy 1 counteracts the increase in cycle time mean and Policy 2 even reduces it. However, this leads to an increase in cycle time variance. A high variance in the production is sometimes undesirable as it makes it harder to detect deviations in the production flow. Still, the increase in variance is small in comparison to the variance caused by the outer disturbances (see Fig. 2.5). Overall, it can be argued that the negative effect on the cycle time is small in comparison to the large reduction in energy use.

The result for station model B is shown in paper C. Overall the same conclusions can be drawn except that the energy reduction is less (22 %) because it has a smaller amount of slack, although the difference is small. The question is how these results differ compared to if the actual station models (instead of Station models A and B) had been used. That is not possible to say for certain, but on average the amount of slack of the actual stations is somewhere between Station model A and B, meaning that the energy reduction can be expected to follow the same trend.

In Fig. 4.8, the energy use of the individual stations for one cycle are shown. For station 8 no energy reduction is possible because it already violates the

Table 4.2: The result of the experiment for Station model A

	Default	Optimized		
	Policy 0	Policy 0	Policy 1	Policy 2
Relative energy use	1	0.757	0.757	0.757
Cycle time mean (s)	97.7	98.3	97.5	96.0
Cycle time var. (s^2)	0.398	0.537	0.905	0.871

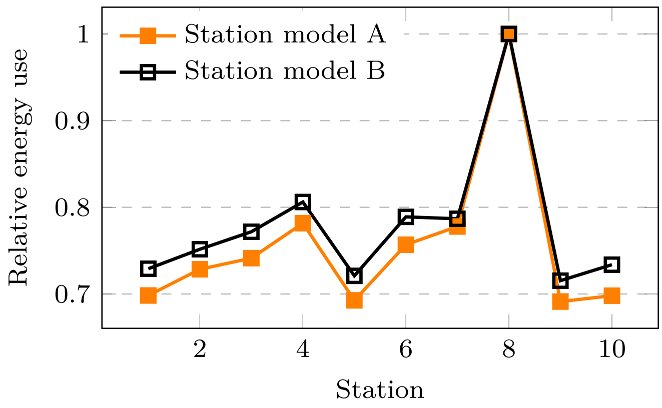


Figure 4.8: The optimized energy for each station for the two station models.

deadline (see Fig. 2.5). Conversely, Station 9 shows a lot of energy reduction, because of the large amount of slack.

4.4 Summary

In this chapter, the methods presented in Chapter 3 for reducing the energy use of industrial robots station have been verified. They have been tested both on real robot stations and in simulation models based on real data. The results showed that the energy use of robot stations and production lines could be reduced significantly with barely affecting the productivity. This verifies that the proposed method to answer **RQ1** works as intended. For **RQ2**, the results showed that the energy models in the simulation software are accurate enough to be usable for the purpose of energy reduction. Relating to **RQ3**, the results also showed that reducing the energy use using the proposed methods also results in significant reductions of peak power and jerk.

CHAPTER 5

Online Energy Reduction Method

This chapter presents an online method for energy reduction and parameter estimation of industrial robot stations. Online in this context means that decisions about the station are made during the execution of the station. It is a summary of the material presented in Papers D and E but presented from the perspective of the latter, if not stated otherwise, as the former essentially is a simplified version of the latter.

5.1 Problem Description

The station used as a case study for this work can be seen in Fig. 5.1. The structure is similar to the type of station described in Chapter 2 but with some key differences. In the station, there is a set of components located in boxes, one per component type. There is a robot tasked with picking these components. Each component type requires a specific tool to be picked, although some component types require the same tool. The tools are in a tool rack together with a camera. The robot can equip the camera and use it to take a picture of a box, after which an algorithm is executed which identifies the location and orientation of several components, making them possible to

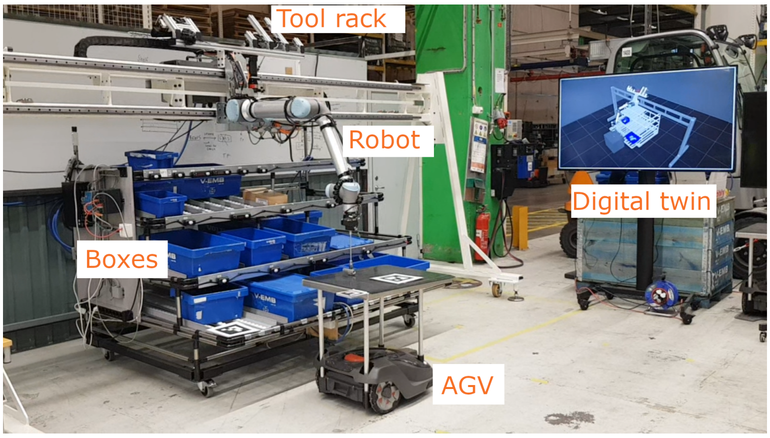


Figure 5.1: The kitting station used as a case study.

pick. When the robot has successfully picked a component, it can be placed on one or more Automated Guided Vehicles (AGVs); each AGV corresponds to an order. An order contains several components that need to be picked and a corresponding deadline t_d^o for when this needs to be accomplished.

The control actions in the stations are a set of operations that the robot can execute to complete the orders, including decisions on the execution times of these operations, which must be tuned to meet the deadlines of the orders. The operations are: move between positions, try to pick a component, equip/unequip a tool, or scan and identify components. The state s of the system at time index k contains information about the time since the start of the system, the equipped tool, the component currently gripped by the robot, the operation being executed, and the position of the robot. For each operation to execute, there is a number of preconditions that need to be fulfilled, which may be more complex than precedence or resource constraints. For example, a component cannot be picked before it has been identified, and the robot needs to be in the right position to equip, unequip, pick, and place.

The stochastic variations in the station are both continuous and discrete. It is assumed that all operations have stochastic execution times that can be approximated by normal distributions with mean μ_i and variance v_i. How big the variations are depend on the type of operation. The discrete stochastic variations come from the fact the robot can fail when trying to pick a component, in which case it has to try again. Another source of stochastic variations is that the orders can arrive in the system at any time. The energy functions of the operations are of the same type as (3.1) but with the difference that the parameters are not known, and no accurate simulation models of the robot exist to determine them.

5.2 Optimization and Control

The goal is to find the operation order and robot velocities such that all orders are completed in time with a high probability and that the energy use is reduced. To achieve that, it is also necessary to find the unknown parameter values of the energy functions. To that end, a control architecture is used that consists of a Model Predictive Controller (MPC) and a Reinforcement Learning (RL) algorithm. The MPC contains a scheduling algorithm to find the optimal operation sequence and a robust optimizer to determine the timing of the robot operations.

The scheduling happens when a new order arrives in the system or the robot is unsuccessful in picking a component. The timing optimization happens when an operation is completed. The learning happens in batches with even intervals but uses data from each timing optimization.

Timing Optimization

The starting point of the timing optimization is the current state s_k and the operation sequence $\mathcal{J}_k$ from the scheduling algorithm. The goal of the optimization is to minimize the energy use by finding the optimal execution times of the operations while satisfying the following chance constraint:

$$\mathrm{P}\left(\bigcup_{o\in\mathcal{O}_k} C_o > t_d^o\right) \leq \epsilon \tag{5.1}$$

where $\mathcal{O}_k$ is the set of active orders at time index k and C_o is the stochastic completion time of the last operation of order o. $\epsilon = 1-\gamma$ and is the maximum allowed probability of missing any of the deadlines. (5.1) is in general hard to express in a closed form but is possible here because every operation duration is assumed to be normally distributed. This assumption means that C_o also is normally distributed and its distribution can be determined by the means and variances of the operations preceding it [55], [56].

The timing optimization presented here is similar to Optimization Problem 1 with a fixed operation order. Given the operation sequence J_k, the problem presented here can also be described by a DAG and can essentially be solved in the same way as Optimization Problem 1 in Section 3.3. However, because this application is time critical, an alternative and less complex approach is used. The optimization problem at time index k can be formulated as

Optimization Problem 2

$$\min_{d_i, w} \sum_{i \in \mathcal{J}_k} g_i(d_i) + Kw \tag{5.2}$$

subject to:

$$\tilde{d}_o = \sum_{i \in \mathcal{J}_{ko}} d_i \qquad o \in \mathcal{O}_k \tag{5.3}$$

$$\tilde{\mu}_o = \mu_{i_k|t_k} + \sum_{i \in \mathcal{J}_{ko}} \mu_i \qquad o \in \mathcal{O}_k \tag{5.4}$$

$$\tilde{v}_o = v_{i_k|t_k} + \sum_{i \in \mathcal{J}_{ko}} v_i \qquad o \in \mathcal{O}_k \tag{5.5}$$

$$t_k + \tilde{\mu}_o + \tilde{u}_o + \sqrt{2\tilde{v}_o \log(1/\epsilon_o)} - t_d^o \leq w \qquad o \in \mathcal{O}_k \tag{5.6}$$

$$\sum_{o \in \mathcal{O}_k} \epsilon_o \leq \epsilon \tag{5.7}$$

$$\underline{d}_i \leq d_i \qquad i \in \mathcal{J}_k \tag{5.8}$$

$$0 \leq w \tag{5.9}$$

which is based on [45] by Nemirovski et al. and [35] by Ben-Tal et al.; more details can be found in Paper D. w is a slack variable and K is chosen big enough so that $w > 0$ only when the problem would be infeasible otherwise. t_k is the time since the start of the system and $\mathcal{J}_{ko} \subseteq \mathcal{J}_k$ is the operation sequence up until order o is completed. $\mu_{i_k|t_k}$ and $v_{i_k|t_k}$ are the mean and variance respectively of the currently executing operation conditioned on the fact that the time is t_k and the operation is not yet completed [62]. ϵ_o can either be chosen beforehand or determined as part of the optimization problem. Optimization Problem 2 is a convex and conservative version of (5.1) in the sense that a feasible solution to Optimization Problem 2 is feasible with respect to (5.1) as well. To conclude the timing optimization, it finds optimal operation durations by solving Optimization Problem 2 for a given operation sequence $\mathcal{J}_k$ from the operation scheduling described in the next section.

Operation Scheduling

The aim of the operation scheduling is to find the operation sequence $\mathcal{J}_k$ to be executed by the robot that takes the system from the current state s_k to a state where all orders are completed. This scheduling problem is more complex than the one in Chapter 3 because the number of required operations is not known in advance and the problem contains constraints (related to tool changing and scanning) that are difficult to model mathematically in a standard optimization formulation. Therefore, a version of the graph search algorithm A^* named Anytime Repairing A^* (ARA^*) [63] is used.

In short, the standard A^* searches through the state space guided by $\xi_h(s)$, which is the expected cost to reach the goal state from any state s, and $\xi_g(s)$, which is the actual cost to go from the current state s_k to state s. For the application in this chapter, the cost is the probability of missing any of the deadlines t_d^o. Essentially, what is minimized in the scheduling is a modified version of Optimization Problem 2 where the energy use is disregarded by setting $d_i = \underline{d}_i$, the cost function is ϵ, and the decision variable is $\mathcal{J}_k$. The algorithm is guaranteed to find the optimal operation sequence if ξ_h is admissible, i.e. smaller or equal to the actual cost. The drawback of A^* is that it may take too long for the algorithm to terminate and return a solution. For the application in this chapter, if the optimization is started during the execution of an operation, it should ideally be completed before the end of the operation, to avoid delays. Therefore, it is desirable to relax the constraint of finding the optimal solution and settle for a sub-optimal solution that is available in time. The idea behind anytime A^* is to solve A^* multiple times, starting with a very relaxed version where it behaves as a greedy best path first algorithm and then reducing the relaxation gradually until ξ_h is admissible. With this approach, there is always a solution available when needed, even though it may not be the optimal one. To conclude the operation scheduling, using a version of ARA^* it finds an operation sequence $\mathcal{J}_k$ that fulfills (5.1) by using a modified version of Optimization Problem 2 as the cost to minimize.

Solving the timing optimization and operation scheduling separately means that the true optimum of the complete problem is not always found (cf. Section 4.2). However, it was found that it is not practically feasible to solve the complete problem in an acceptable time. The proposed division of the optimization problem prioritises meeting the deadlines, which is the most important criterion, and the author believes that this division is a good compromise between speed and accuracy.

Learning and Adaptation

The learning and adaptation algorithm is used to update the estimates of the unknown parameters of the energy functions. The algorithm needs to be fast enough to be usable online and the same be able to estimate the unknown parameters accurately enough. Essentially, the problem of controlling the kitting station can be modeled as a Markov decision process [64]. A reinforcement learning method based on [65] by Gros and Zanon is used, which has shown to be effective for similar types of systems. It requires measurements of the actual energy use and duration of the last operation and uses the values of the cost function of Optimization Problem 2 as the value function. It then calculates parameter updates using the gradient with respect to the unknown parameters of the Lagrangian relaxation of Optimization Problem 2. More details are found in Paper E.

5.3 Numerical Experiments

In this section, the proposed control architecture is applied to numerical examples of robot stations, to verify it.

Initial Example

The first example is from paper D and is a simple robot station of the same type as in Section 4.1, with operations with stochastic execution times and a fixed operation order. The purpose of the experiment is to investigate the benefits of performing the optimization online as opposed to offline. The experiment is conducted by simulating the station many times. During each simulation, a version of Optimization Problem 2 is solved n_{opt} times with an interval of $\Delta t = \frac{t_d}{n_{\text{opt}}}$. The result can be seen in Fig. 5.2 and shows that the more often the optimization is done the more energy reduction can be made, which intuitively can be explained as follows. The optimizations early in each simulation have to leave margins in the schedule for uncertainties in the

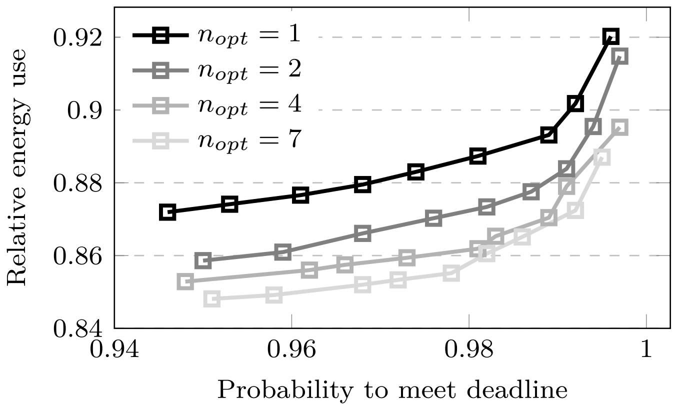


Figure 5.2: The relative energy use as a function of the simulated probability to meet the deadline, for some values of n_{opt}.

execution times. As the simulation progresses more and more uncertainties are realized, meaning that fewer and fewer margins have to be left in the schedule, which enables energy reduction. The result also shows that the additional effect decreases as n_{opt} increases. An additional plot presented in Paper D shows that the additional benefit of performing the optimization online increases as the variance of the execution times does. It also shows that the variance of the makespan is decreased with an increasing number of n_{opt}; as already mentioned, a low variance in the makespan is something which is desirable in many production systems.

Experimental Setup

This section describes the setup of the numerical example in Paper E. The experimental setup is as follows. 500 orders are randomly generated, each order consisting of 1-5 components. Which component types they are and how many of each component type is random. During the execution of the station, new orders are added to the station randomly, such that 1-3 orders are always active. The optimization and learning are executed according to Section 5.2 until all orders are done. This procedure, starting from the 500 orders is repeated 5 times

Detailed Results

The results of the experiment can be seen in Fig. 5.3 and 5.4. Fig 5.3 shows the energy use per operation over episodes of the experiment, where an episode is defined between the completion of two orders. As baseline energy use, three values are shown: the average of 100 episodes without any optimization, with optimization using the initial energy parameter values, and with optimization using the true parameter values. The actual result of the experiment with optimization using the learned parameter values is shown as the minimum and maximum values of all five repetitions and as the rolling mean of 50 episodes for each of the five repetitions.

The result shows that the learning and optimization work as expected: in the beginning, the energy use cannot be reduced because the parameters are wrong, but as the parameters are learned, it converges to the true optimum. Fig. 5.4 shows a comparison between the true parameters and the learned parameters for one of the operations. Note that the goal of the parameter estimation is not to find the true values, but to find values that allow the minimum energy use to be achieved. In Fig. 5.4, the true and learned optimum is roughly the same. The fact that some of the estimated functions have steeper gradients for values greater than the optimum does not matter, the optimum will still be found. Similar conclusions can be made for the other

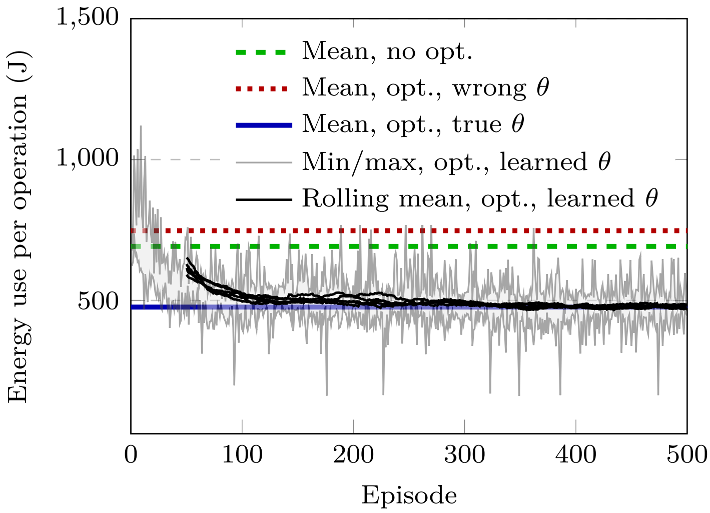


Figure 5.3: The average energy use per operation and episode of the experiment.

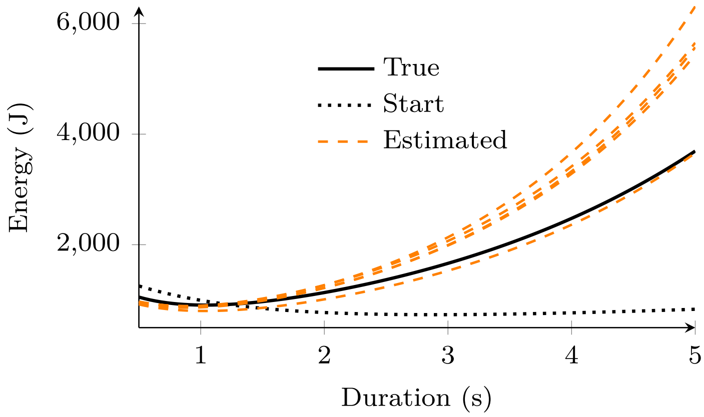


Figure 5.4: The energy function of the move operation with the start, estimated and true parameters values.

operations. In paper E the distributions of the completion times of the orders are shown. Only between 0.5-1.5 percent of the values miss the deadline, which is acceptable.

5.4 Summary

In this chapter, a case study of a flexible robot station has been shown. To answer **RQ4**, a method of controlling this station has been presented. The method contains operation scheduling, timing optimization, and parameter estimation. Experiments showed that the method successfully reduced the energy use, while the deadlines were met. Relating to **RQ2**, it was also shown that the method was able to estimate the unknown parameters of the energy models such that the optimal energy use could be found.

CHAPTER 6

Summary of included papers

This chapter provides a summary of the included papers.

6.1 Paper A

Mattias Hovgard, Bengt Lennartson, and Kristofer Bengtsson
Applied Energy Optimization of Multi-Robot Systems Through Motion Parameter Tuning
CIRP Journal of Manufacturing Science and Technology, 35, 422-430, 2021

In this paper a case study of a real robot station is investigated. The station is modeled and an energy reduction method is presented. It involves formulating and solving an optimization problem and tuning of motion parameters in the robot code. The motion parameters are compared to find the one that is most effective in terms of energy reduction. The optimized motion parameters are then implemented in the robot code in the real station and experiments are done to verify how accurate the energy models are and how much energy reduction can be made. The result shows that the energy models are accurate enough to be used for the purpose of energy reduction. It also shows that the energy use can be reduced by 12 %, the peak power by 12 %, and the jerk by 25 %.

6.2 Paper B

Mattias Hovgard, Bengt Lennartson, and Kristofer Bengtsson
Energy Reduction of Stochastic Time-Constrained Robot Stations
Revised journal submission

This paper is a continuation of the previous one with the difference that stochastic uncertainties are included and that three other robot stations are used as case studies. Because of the inclusion of stochastic variables, the optimization problem becomes harder to solve. An approach is presented to approximate the optimization problem and make it solvable. Experiments on the stations are done in simulations built using real data. The result shows that the energy use of the stations can be reduced significantly by 9, 11 and 18 % respectively while meeting the deadlines with high probabilities.

6.3 Paper C

Mattias Hovgard, Bengt Lennartson, and Kristofer Bengtsson
Energy-Optimal Timing of Stochastic Robot Stations in Automotive Production Lines
Accepted for the 27th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Stuttgart, September 2022

This paper investigates the problem of reducing the energy use of robot stations with stochastic execution times in production lines in automotive factories. It combines real stochastic cycle time data with robot and station models from the previous two papers. Firstly, policies to reduce the cycle time of the production line are presented, to leave more room for energy reduction. Secondly, energy is reduced by using the method presented in Paper B. The result shows that up to 24 % of the energy use can be reduced with only marginally affecting the cycle time variance of the production line.

6.4 Paper D

Mattias Hovgard, Bengt Lennartson, and Kristofer Bengtsson
Online Energy-Optimal Timing of Stochastic Robot Stations
Proceedings of the 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Västerås, September 2021

This paper looks at online energy reduction of similar types of industrial robot stations as in Papers, A, B, and C. In it, an MPC-type method is presented for tuning robot velocities such that the energy is reduced but that the deadline is met. The result shows that it is beneficial to perform the optimization online compared to offline but that the additional benefit is decreasing with the frequency of optimization. The result also shows that the benefit of performing the optimization online increases with the variance in operation execution times.

6.5 Paper E

Mattias Hovgard, Constantin Cronrath, Kristofer Bengtsson, and Bengt Lennartson
Adaptive Energy Optimization of Flexible Robot Stations
Journal submission

This paper proposes an adaptive control strategy that is applied to a flexible robot kitting station. The station contains several stochastic uncertainties, such as variations in execution times, probability of failure and random arrival of orders. An additional challenge is that the parameters of the energy models are unknown. The aim of the control strategy is to reduce energy use while meeting deadlines. A control architecture is presented that consists of a type of MPC and an RL algorithm. The MPC contains a scheduling algorithm to find the optimal operation sequence and a robust optimizer to determine the timing of the operations. The RL algorithm uses information from the MPC to update estimates of the unknown model parameters. The proposed control architecture is applied in simulation to the case study. The result shows that the unknown model parameters are estimated accurately, the energy is reduced, and the deadlines are met.

CHAPTER 7

Concluding Remarks and Future Work

Below follows some concluding remarks about the material presented in this thesis and ideas for future work.

7.1 Concluding remarks

The first part is about the offline method and the corresponding experiments (Chapters 3 and 4). Related to **RQ1**, an offline method for reducing the energy use of industrial robot stations in a production line was presented. It contains collecting data, model building, finding energy functions, optimization, and implementation in robot code. It was verified, using four examples of real robot stations, that the method works as intended. It is simple enough to be used in practice and achieves energy reduction of up to 24 % for a production line, without lowering productivity. Related to **RQ2**, real experiments have been conducted to investigate the accuracy of commercially available energy models. It was shown that the difference is small enough to allow the energy models to be used for energy reduction purposes. Related to **RQ3**, it was shown through the same experiments that the energy reduction method also resulted in reduced peak power by 12 % and jerk by 25 %.

It was surprising for the author how much energy reduction potential there was in the investigated production lines. The impression was that the only optimization that had been done to the production lines was to reduce the cycle time. A possible reason for this is the combination of a relatively low energy cost in comparison to the cost of employing engineers to work on energy efficiency and the cost of having a breakdown in the production line due to some mistake caused by energy reduction efforts.

A potential weakness of the proposed method is one related to the temperature of the robot joints. As shown previously [16] the energy use of an industrial robot decreases with increasing temperatures in the robot joints. Before the real experiment, the robot was executed at maximum speed to reach the appropriate joint temperatures. It is possible that only executing the robot at lower energy optimized velocities would not be enough to reach the appropriate joint temperature and thereby lead to an increase in energy use instead. This could also be a drawback when it comes to the wear and tear of the robot. It was argued in Section 4.1 that a lower velocity is beneficial for the wear and tear of the dress pack attached to the robot. However, for the robot itself, too low joint temperatures (resulting from too low velocities) may result in the joint lubrication having too high viscosity and thereby increase the joint friction and wear [66].

The second part is about the online method presented in Chapter 5. Related to **RQ4**, a control architecture for controlling a flexible robot station in a robust and energy-optimal way was presented. The method includes a scheduling algorithm, an MPC-type of timing optimization, and a learning algorithm for estimating unknown model parameters. An example was used to show that the method can control the system, reducing energy use, and meeting deadlines. Related to **RQ2**, a method was presented that uses reinforcement learning to estimate unknown model parameters. This step was necessary as there, as opposed to the robots in the welding stations, were no simulation models available. Results showed that the method could estimate the parameters well enough so that the optimum of the problem could be found.

The problem of controlling the flexible robot station in Chapter 5 is an interesting research problem. It contains many different types of stochastic variations: continuous execution time variations, discrete failing probabilities, and arrivals of orders. To completely understand the problem and find the best control strategy more accurate data about the stochastic variations would be needed, something that has not yet been possible to obtain.

7.2 Future work

Firstly, the stations that the methods presented in this thesis have been applied to have been relatively similar. It would be interesting to investigate the energy reduction potential for other kinds of robot stations, such as stations from other types of industries or collaborative robot stations where human operators are more involved.

Secondly, a bottleneck in this research about energy reduction methods was access to real stations. Both in terms of data to create the necessary models, and the usefulness of the results of the research. With more access to real robot stations, it would probably be possible to design more effective methods and the results would be more useful for the industry.

Related to the same topic, the energy reduction approach of the welding stations was somewhat limited by the fact that they were already used in production. If the energy reduction effort had been done at an earlier stage, before it was taken into production, there would be more freedom in applying energy reduction methods. In that case, more advanced methods could be used, such as modifying the paths of the robot movements, and it would be possible to divide the tasks more freely between the robots; this would probably increase the potential for energy reduction.

On a larger scale, the question is if the energy reduction of robots is the most effective approach for the industry to reduce its energy use. For example, in the automotive industry the lighting and ventilation [38] use a large amount of energy; in particular the ventilation in a paint shop. It remains to be investigated and is a question of the required work effort and risk compared to the gain in reduced energy use.

Finally, the parameter estimation algorithm in Chapter 5 was only used to identify the energy functions. It would make the energy reduction method more usable in practice if the learning method was able to approximate the full energy model of a robot or the stochastic variations in the execution times of the operations.

References

[1] IEA, "Energy efficiency 2019," IEA, 2019.

[2] M. Pellicciari, A. Avotins, K. Bengtsson, *et al.*, "Areus innovative hardware and software for sustainable industrial robotics," in *2015 IEEE International Conference on Automation Science and Engineering (CASE)*, 2015, pp. 1325–1332.

[3] D. Meike and L. Ribickis, "Energy efficient use of robotics in the automobile industry," in *2011 15th International Conference on Advanced Robotics (ICAR)*, IEEE, 2011, pp. 507–511.

[4] L. Kroll, P. Blau, M. Wabner, U. Friess, J. Eulitz, and M. Klärner, "Lightweight components for energy-efficient machine tools," *CIRP Journal of Manufacturing Science and Technology*, vol. 4, pp. 148–160, 2011.

[5] M. Gadaleta, G. Berselli, and M. Pellicciari, "Energy-optimal layout design of robotic work cells: potential assessment on an industrial case study," *Robotics and Computer-Integrated Manufacturing*, vol. 47, pp. 102–111, 2017.

[6] S. Gürel, H. Gultekin, and V. E. Akhlaghi, "Energy conscious scheduling of a material handling robot in a manufacturing cell," *Robotics and Computer-Integrated Manufacturing*, vol. 58, pp. 97–108, 2019.

[7] C. Cronrath, B. Lennartson, and M. Lemessi, "Energy reduction in paint shops through energy-sensitive on-off control," in *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, IEEE, 2016, pp. 1282–1288.

[8] H. Lyu, X. Song, D. Dai, J. Li, and Z. Li, "Time-optimal and energy-efficient trajectory generation for robot manipulator with kinematic constraints," in *2017 13th IEEE Conference on Automation Science and Engineering (CASE)*, 2017, pp. 503–508.

[9] G. Carabin, E. Wehrle, and R. Vidoni, "A review on energy-saving optimization methods for robotic and automatic systems," *Robotics*, vol. 6, no. 4, 2017.

[10] P. Boscariol and D. Richiedei, "Energy-efficient design of multipoint trajectories for Cartesian robots," *The International Journal of Advanced Manufacturing Technology*, vol. 102, pp. 1853–1870, 2019.

[11] N. Sinaga, P. Paryanto, S. A. Widyanto, R. Rusnaldy, A. Hetzner, and J. Franke, "An analysis of the effect of gravitational load on the energy consumption of industrial robots," *Procedia CIRP*, vol. 78, pp. 8–12, 2018.

[12] O. Wigstrom, B. Lennartson, A. Vergnano, and C. Breitholtz, "High-level scheduling of energy optimal trajectories," *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 1, pp. 57–64, 2013.

[13] M. Pellicciari, G. Berselli, F. Leali, and A. Vergnano, "A method for reducing the energy consumption of pick-and-place industrial robots," *Mechatronics*, vol. 23, no. 3, pp. 326–334, 2013.

[14] M. Gautier, A. Janot, and P. O. Vandanjon, "A new closed-loop output error method for parameter identification of robot dynamics," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 2, pp. 428–444, 2013.

[15] K. Paes, W. Dewulf, K. Vander Elst, K. Kellens, and P. Slaets, "Energy efficient trajectories for an industrial ABB robot," *Procedia CIRP*, vol. 15, pp. 105–110, 2014.

[16] S. Riazi, O. Wigström, K. Bengtsson, and B. Lennartson, "Energy and peak power optimization of time-bounded robot trajectories," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 2, pp. 646–657, 2017.

[17] L. Wang, A. Mohammed, X. V. Wang, and B. Schmidt, "Energy-efficient robot applications towards sustainable manufacturing," *International Journal of Computer Integrated Manufacturing*, vol. 31, no. 8, pp. 692–700, 2018.

[18] E. Glorieux, S. Riazi, and B. Lennartson, "Productivity/energy optimisation of trajectories and coordination for cyclic multi-robot systems," *Robotics and Computer-Integrated Manufacturing*, vol. 49, pp. 152–161, 2018.

[19] A. Vergnano, C. Thorstensson, B. Lennartson, *et al.*, "Modeling and optimization of energy consumption in cooperative multi-robot systems," *IEEE Transactions on Automation Science and Engineering*, vol. 9, no. 2, pp. 423–428, 2012.

[20] L. Bukata, P. Sucha, Z. Hanzálek, and P. Burget, "Energy optimization of robotic cells," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 1, pp. 92–102, 2017.

[21] G. Pastras, A. Fysikopoulos, and G. Chryssolouris, "A theoretical investigation on the potential energy savings by optimization of the robotic motion profiles," *Robotics and Computer-Integrated Manufacturing*, vol. 58, pp. 55–68, 2019.

[22] D. Gleeson, C. Larsen, J. S. Carlson, and B. Lennartson, "Implementation of a rapidly executing robot controller," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, 2019, pp. 1341–1346.

[23] D. Gleeson, S. Björkenstam, R. Bohlin, J. S. Carlson, and B. Lennartson, "Optimizing robot trajectories for automatic robot code generation," in *2015 IEEE International Conference on Automation Science and Engineering (CASE)*, 2015, pp. 495–500.

[24] R. R. Garcia, A. C. Bittencourt, and E. Villani, "Relevant factors for the energy consumption of industrial robots," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 40, no. 9, 2018.

[25] M. Gadaleta, M. Pellicciari, and G. Berselli, "Optimization of the energy consumption of industrial robots for automatic code generation," *Robotics and Computer-Integrated Manufacturing*, vol. 57, pp. 452–464, 2019.

[26] D. Meike, M. Pellicciari, and G. Berselli, "Energy efficient use of multi-robot production lines in the automotive industry: detailed system modeling and optimization," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 3, pp. 798–809, 2014.

[27] G. Chen, L. Zhang, J. Arinez, and S. Biller, "Energy-efficient production systems through schedule-based operations," *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 1, pp. 27–37, 2013.

[28] Q. Chang, G. Xiao, S. Biller, and L. Li, "Energy saving opportunity analysis of automotive serial production systems (March 2012)," *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 2, pp. 334–342, 2013.

[29] M. Faraji Amiri and J. Behnamian, "Multi-objective green flowshop scheduling problem under uncertainty: estimation of distribution algorithm," *Journal of Cleaner Production*, vol. 251, 2020.

[30] N. Sundström, O. Wigström, and B. Lennartson, "Conflict between energy, stability, and robustness in production schedules," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 2, pp. 658–668, 2017.

[31] S. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge University Press, 2004.

[32] P. Bonami, L. T. Biegler, A. R. Conn, *et al.*, "An algorithmic framework for convex mixed integer nonlinear programs," *Discrete Optimization*, vol. 5, no. 2, pp. 186–204, 2008.

[33] J. Kronqvist, D. E. Bernal, A. Lundell, and I. E. Grossmann, "A review and comparison of solvers for convex MINLP," *Optimization and Engineering*, vol. 20, pp. 397–455, 2019.

[34] P. Belotti, C. Kirches, S. Leyffer, J. Linderoth, J. Luedtke, and A. Mahajan, "Mixed-integer nonlinear optimization," *Acta Numerica*, vol. 22, pp. 1–131, 2013.

[35] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *Robust Optimization.* Princeton University Press, 2009.

[36] W. Khalil and E. Dombre, *Modeling, Identification and Control of Robots.* Elsevier Science & Technology, 2004.

[37] M. Gadaleta, G. Berselli, M. Pellicciari, and M. Sposato, “A simulation tool for computing energy optimal motion parameters of industrial robots,” *Procedia Manufacturing*, vol. 11, pp. 319–328, 2017.

[38] J. M. Rödger, N. Bey, and L. Alting, “The Sustainability Cone - A holistic framework to integrate sustainability thinking into manufacturing,” *CIRP Annals - Manufacturing Technology*, vol. 65, no. 1, pp. 1–4, 2016.

[39] ABB, *Technical reference manual, RAPID Instructions, Functions and Data types*. ABB, 2017.

[40] B. Lennartson, K. Bengtsson, C. Yuan, *et al.*, “Sequence planning for integrated product, process and automation design,” *IEEE Transactions on Automation Science and Engineering*, vol. 7, no. 4, pp. 791–802, 2010.

[41] M. Sköldstam, K. Åkesson, and M. Fabian, “Modeling of discrete event systems using finite automata with variables,” in *2007 46th IEEE Conference on Decision and Control*, 2007, pp. 3387–3392.

[42] X. Cai, X. Wu, and X. Zhou, *Optimal Stochastic Scheduling*. Springer New York, 2014.

[43] M. L. Pinedo, *Scheduling*. Springer Cham, 2016.

[44] S. L. Janak, X. Lin, and C. A. Floudas, “A new robust optimization approach for scheduling under uncertainty. II. uncertainty with known probability distribution,” *Computers and Chemical Engineering*, vol. 31, no. 3, pp. 171–195, 2007.

[45] A. Nemirovski and A. Shapiro, “Convex approximations of chance constrained programs,” *SIAM Journal on Optimization*, vol. 17, no. 4, pp. 969–996, 2007.

[46] J. R. Birge and F. Louveaux, *Introduction to Stochastic Programming*. Springer New York, 2011.

[47] P. Brucker, A. Drexl, R. Möhring, K. Neumann, and E. Pesch, “Resource-constrained project scheduling: notation, classification, models, and methods,” *European Journal of Operational Research*, vol. 112, pp. 3–41, 1999.

[48] M. Wahde, *Biologically Inspired Optimization Methods: An Introduction*. WIT Press, 2008.

[49] X. Li, K. Xing, Y. Wu, X. Wang, and J. Luo, "Total energy consumption optimization via genetic algorithm in flexible manufacturing systems," *Computers and Industrial Engineering*, vol. 104, pp. 188–200, 2017.

[50] K. Nonoyama, Z. Liu, T. Fujiwara, M. M. Alam, and T. Nishi, "Energy-efficient robot configuration and motion planning using genetic algorithm and particle swarm optimization," *Energies*, vol. 15, no. 6, 2022.

[51] N. R. Ortiz-Pimiento and F. J. Diaz-Serna, "The project scheduling problem with non-deterministic activities duration: A literature review," *Journal of Industrial Engineering and Management*, vol. 11, no. 1, pp. 116–134, 2018.

[52] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

[53] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, pp. 25–57, 2006.

[54] I. Griva, S. G. Nash, and A. Sofer, *Linear and Nonlinear Optimization*, 2. ed. Philadelphia: SIAM, 2008.

[55] Möhring R.H., "Scheduling under uncertainty: bounding the makespan distribution," in *Alt H. (eds) Computational Discrete Mathematics. Lecture Notes in Computer Science*, vol. 2122, Berlin, Heidelberg: Springer, 2001, pp. 79–97.

[56] E. W. Weisstein, "Normal sum distribution," *From MathWorld–A Wolfram Web Resource*,

[57] ——, "Central limit theorem," *From MathWorld–A Wolfram Web Resource*,

[58] ABB, *Operating manual RobotStudio*, 2018.

[59] V. Zanotto, A. Gasparetto, A. Lanzutti, P. Boscariol, and R. Vidoni, "Experimental validation of minimum time-jerk algorithms for industrial robots," *Journal of Intelligent and Robotic Systems*, vol. 64, pp. 197–219, 2011.

[60] A. Olabi, R. Béarée, O. Gibarua, and M. Damakc, "Feedrate planning for machining with industrial six-axis robots," *Control Engineering Practice*, vol. 18, no. 5, pp. 471–482, 2010.

[61] J. S. Carlson, J. Kressin, T. Hermansson, R. Bohlin, M. Sundbäck, and H. Hansson, "Robot station optimization for minimizing dress pack problems," *Procedia CIRP*, vol. 44, pp. 389–394, 2016.

[62] M. Hovgard, B. Lennartson, and K. Bengtsson, "Online energy-optimal timing of stochastic robot stations," in *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, IEEE, 2021, pp. 01–06.

[63] M. Likhachev, G. Gordon, and S. Thrun, "Ara*: Anytime a* with provable bounds on sub-optimality," in *Proceedings of (NeurIPS) Neural Information Processing Systems*, 2003, pp. 767–774.

[64] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[65] S. Gros and M. Zanon, "Data-driven economic nmpc using reinforcement learning," *IEEE Transactions on Automatic Control*, vol. 65, no. 2, pp. 636–648, 2019.

[66] A. C. Bittencourt, *Modeling and diagnosis of friction and wear in industrial robots*. Linköping University Electronic Press, 2014.