

Thesis for the Degree of Licentiate of Engineering

Efficient Simulation and Optimization for Tandem Press Lines

Nima Khansari Nia

Thesis for the degree of Licentiate of Engineering at University West will be defended in public on Tuesday 25 of September 2012 at 10.00 in C118, Gustava Melins gata 2, University West, Trollhättan. Language: English.

Faculty opponent: Associate Professor Amos H.C. Ng,
University of Skövde
School of Technology and Society



Department of Signals and Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2012



Department of Engineering Science
UNIVERSITY WEST
Trollhättan, Sweden 2012

Efficient Simulation and Optimization for Tandem Press Lines

Nima Khansari Nia

Department of Signals and Systems
Automation Research Group
Chalmers University of Technology
SE-412 96 Göteborg, Sweden
Telephone +46 (0) 31 772 10 00

© Nima K. Nia, 2012

Printed by Chalmers Reproservice
Göteborg, Sweden 2012

Today, simulation is needed as one of the tools for optimum utilization of a press line. Utilization includes: high throughput, minimum wear of the line and high quality. High quality includes quality of the produced parts, quality of the stamping dies and quality of the tooling manufactured. In order to perform efficient geometrical simulation and virtual commissioning in stamping, two main fields are investigated namely simulation time and optimization time. Thus, reducing computation time is the main theme of this thesis. An efficient press line simulation model is built and verified, resulting in reduced simulation building time due to the modularization of the model.

To reduce simulation time, collision detection time is reduced by a method based on 3D to 2D geometrical collision detection. The method is based on pre-calculation of all collision points in the environment of interest, and then using a simplified collision detection model in a simulation based optimization. This is less resource consuming than collision checking the original 3D objects for all optimization evaluations. The suggested approach reduces the collision detection from a 3D to a 2D problem, where collision between simplified but moving curves is used in the repeated simulation for optimization. This collision detection approach, together with a simplified implementation of the control code, results in ~200 times reduction of the computation time, compared to the original simulation based on standard 3D collision detection.

The variable parameters in a press line exceed 100, resulting in time/computationally demanding computations. There is also a need of fast optimizations for die design, line tryout and ramp-up. Since the number of evaluations grows exponentially with the number of dimensions in an optimization problem, optimization time is reduced by a decomposition strategy aiming at dimension reduction. Two simulation/optimization strategies were chosen and tested to decrease calculations. The presented results mean that simulation and virtual commissioning can be performed not only for press stations but also for complete press lines, where the complexity increases linearly with the number of stations in the line.

KEYWORDS: Press Line Simulation, Simulation based optimization, Virtual manufacturing, Virtual commissioning, Parameter tuning, Optimization strategies.

Acknowledgments

I would like to thank Fredrik Danielsson my university supervisor for his support, commitment and attention on details. Thanks for once asking me to participate as a PhD student.

I would also like to express my gratitude to Bengt Lennartson my main supervisor. His support and ideas has pushed me in doing better projects, analysis and publications.

I am grateful for support and help from Emile Glorieux. His ideas and vast skill in programming has been invaluable. Thanks also to Bo Svensson, for his help and pedagogical explanations. I am also grateful to Anders Nilsson for his help in this project.

I would like to express my gratitude to my company supervisor Roger Mattsson for his visions for virtual factories which enabled this project and his support during the project. I would also like to thank my local company supervisors in for their support. Without their understanding, this project had been though to complete.

I wish to thank my parents and sister for their advice and support. Above all I wish to thank my lovely wife Anita for her understanding and help during this time of raising our three small and fantastic children, Darian, Tindra and Ariana.

Kristianstad, Sweden August 2012

Nima K. Nia

Publications

The thesis is based on the following three appended papers.

Paper I: Nima K. Nia, Fredrik Danielsson, and Bengt Lennartson, (2011). A faster collision detection method applied on a sheet metal press line. In *Proceedings of FAIM2011, the 21th International Conference on Flexible Automation and Intelligent Manufacturing*, pp 833-840, Taichung, Taiwan.

Paper II: Nima K. Nia, Fredrik Danielsson, and Bengt Lennartson, (2012). Efficient geometrical simulation and virtual commissioning performed in stamping. In *Proceedings of ETFA2012, the 17th International Conference on Emerging Technologies & Factory Automation*
ETFA, Poland

Paper III: Nima K. Nia, Fredrik Danielsson, and Bengt Lennartson, (2012). Toward efficient simulation and optimization strategies in stamping. Submitted to an international journal.

Other publications

The following paper also contributes to the thesis work, but is not appended. This paper contains mainly case study results and/or constitutes first outlines to approaches, methods and concepts in the appended papers.

Bo Svensson, Nima K. Nia, Fredrik Danielsson, and Bengt Lennartson, (2011). Sheet-Metal Press Line Parameter Tuning using a Combined DIRECT and Nelder-Mead Algorithm. In *Proceedings of ETFA2011, the 16th IEEE International Conference on Emerging Technologies and Factory Automation*, pp. 207-214, Toulouse, France.

Abbreviations

| | |
|-----|--|
| CIM | <i>Collision Inspection Method</i> – An efficient method for collision detection aimed for iterative simulations. |
| CU | <i>Collision Curve Upper</i> – A 2D limitation curve for upper part of the die |
| CL | <i>Collision Curve Lower</i> – A 2D limitation curve for lower part of the die |
| CS | <i>Collision Curve Sheet</i> – A 2D limitation curve based on collision between the entering sheet in the die and the extracting sheet |
| TCP | <i>Tool Center Point</i> – A decided point on a robot |

Contents

| | |
|-----------------------|-----|
| Abstract | iii |
| Acknowledgments | v |
| Publications | vii |
| Abbreviations | ix |
| Contents | x |

Introductory Chapters

| | |
|---|----|
| Introduction | 3 |
| 1.1. Objective..... | 6 |
| 1.2. Research questions | 6 |
| 1.3. Main contributions..... | 7 |
| 1.4. Limitations..... | 8 |
| 1.5. Outline of thesis..... | 8 |
| Related research..... | 9 |
| 2.1. Simulation within stamping area | 9 |
| 2.2. Collision detection in simulations | 11 |
| 2.3. Simulation-based optimisation | 12 |
| 2.4. Simulation, optimization and virtual commissioning | 12 |
| 2.5. Summary..... | 13 |
| Press Line Components and Simulation..... | 15 |
| 3.1. Press lines | 15 |
| 3.2. Scanning | 15 |
| 3.3. Die design | 16 |
| 3.4. Die simulation..... | 17 |
| 3.5. Die internal collision detection..... | 17 |
| 3.6. Die external collision detection | 18 |
| 3.7. Gripper..... | 19 |
| 3.8. Summery..... | 20 |
| Tandem Press line | 21 |
| 4.1. Press line description | 21 |
| 4.2. Synchronization | 22 |
| 4.3. Motion | 23 |

| | | |
|--|---|-----------|
| 4.4. | Variable waiting time | 25 |
| 4.5. | Summary..... | 25 |
| Collision Inspection Method | | 27 |
| 5.1. | Collision Inspection Method | 27 |
| 5.2. | CIM part 1: Sampling collision points between objects..... | 28 |
| 5.3. | CIM Part 2: Collision check between TCP and Collision Curve | 30 |
| 5.4. | Creation of Collision Curve for a die | 30 |
| 5.5. | Collision check between TCP, CU, CL and CS | 31 |
| 5.6. | Relative motion curves | 33 |
| 5.7. | Minimizing collision risks | 35 |
| 5.8. | Summary..... | 35 |
| Simulation, Optimization and Strategies | | 37 |
| 6.1. | Simulation model..... | 37 |
| 6.2. | Optimization results for a press station | 40 |
| 6.3. | Multi objective functions..... | 41 |
| 6.4. | Optimization strategies for a press line | 42 |
| 6.5. | Optimization results for a press line | 45 |
| 6.6. | Summary..... | 46 |
| Conclusion and future work | | 49 |
| Summary of appended papers..... | | 51 |

Appended Papers

- Paper I: A faster collision detection method applied on a sheet metal press line
- Paper II: Efficient geometrical simulation and virtual commissioning performed in stamping
- Paper III: Toward efficient simulation and optimization strategies in stamping

Introductory Chapters

Chapter 1

Introduction

Many companies today, strive for preparation and good discipline in order to improve manufacturing result. The time to produce a component is beside well preparation, where production parameters are taken into consideration, also dependent on the time to introduce and ramp up production. The components production time is also dependent on production down time, where lack of consideration of production parameters are one of the reasons.

Stamping is one of the branches of manufacturing where automated manufacturing systems, in particular press lines are used. A frequently used approach to increase capacity utilization rate, or to be more precise to minimize the time per produced component in press line, is to tune the process parameters such as time constants, cam values, velocities and robot paths. Currently this on-line optimization approach is a manual, empirical, trial-and-error method [1].

There are many tools suggested by researchers and industry to reach quality and cost effective manufacturing. Simulation is considered to be one of the tools. There are several simulation types in stamping where finite element method, geometric simulations and virtual commissioning are common. Simulations usage is dependent on the complexity of the stamping process and how well virtual manufacturing tools are incorporated as a quality tool in companies.

Geometrical simulation and virtual commissioning of press stations (press and sheet metal handling equipment) and press lines have not been in the focus of the scientific community. Because of the complexity of the process, no single simulation software exists for handling all of the steps. Consequently, different simulation software is used, which often results in compatibility problems as a consequence. This field is relatively new, and research and development are still to be made in order to fully benefit from simulation as a quality tool.

The frequent usage of simulation in stamping leads to the question: Why are efficient geometrical simulation and virtual commissioning important in stamping?

High throughput, high quality (e.g. few collision risks), and minimum wear of stamping equipment are among key factors in stamping. Many parameters such as die/gripper shape, oil amount on sheet metal part, and purity in production, in turn impact these factors [2]. Simulation is a tool to address some of these parameters. It is utilized as a quality instrument, aiming at cost minimization due to the complexity of the stamping process. It also decreases operator expertise dependency which is a major aspect in tuning press lines in a more objective way. Hence, with the help of simulation the need for on-line tuning can be decreased in favor of off-line tuning.

Operator expertise is important in a successful manufacturing environment. When a new type of component is introduced in the line, not only the mechanical tools in the press station

but also the control parameters have to be modified to match the component [3]. Most automotive press line tuning methods are today performed on-line and highly dependent on the skill and experience of the operator. In most automated industrial equipment the option of storing programs exists. This option is used by the operators to introduce a new product to the press line by using a program most likely compatible to the new product. The program is then manipulated, in most cases manually to an optimum, based on the skills of the operators. The high dependency of the operator has several major risks. The first is the risk of collisions. Since the consequences of collisions are production disturbances and costly damages, safer parameters are chosen resulting in lower cycle time. Machine wear due to poor parameter choices leading to rough robot movements, is another hazard. Other risks include the possibility of losing intellectual property, where experienced operators leave the line for another or leave the company which would mean loss of knowledge and optimum expertise. Losing intellectual property results in delays and knowledge restarts for the line. A computerized optimizer would decrease the dependency of operators' skills and risk of losing intellectual property on operator level.

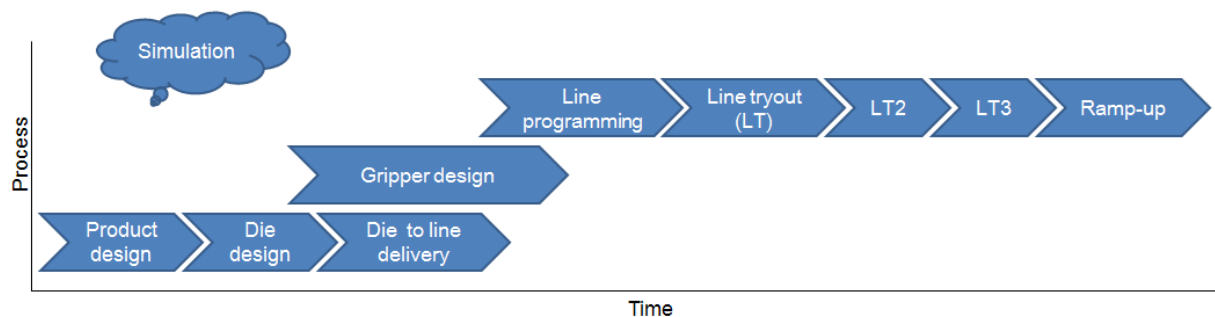


Figure 1 Illustration of different steps of stamping process

Another question that has to be taken into consideration by a stamping company is: When is simulation intended to be used, in the time frame of product/die design to ramp-up?

A variance in the response time of a simulation is acceptable, depending on the intended use of the simulation, in the time frame of product/die design to ramp-up, see Figure 1. This is an important factor for companies. A longer simulation time is acceptable when the simulation is planned for line tryout/ramp-up. The reason is that the time between die design [4] and die delivery to the line can be used for simulation. Die design which has shorter delivery time needs faster simulations in order to compare different design solutions and their impact on the cycle time of the press line.

A press line is complex. Hence, the simulation of a press line when virtual commissioning [5] is incorporated, is complex. Because of the complexity, great time and resources are used first to build simulation models and next to run the simulations. Therefore the time and resource consumption when building and running virtual commissioning simulation models is a factor to take into consideration. According to design methodology the goal of a project is to use the outcome in an industrial environment, Budynas et al. [6]. The outcome of this thesis is aimed for an industrial environment; thus the simulation must have an acceptable response time. There are several alternatives to achieve this goal. One would be to use a cluster server environment also referred to as parallel computing [7], with enough power to fulfill the requirements for the simulation. The other would be finding alternative calculation methods, which permits running the simulation on simpler computers. Method two is chosen due to its improvement potential.

This thesis concentrates on investigating and suggesting methods for simulation and optimization of tandem press lines. This is a difficult task and to the author's knowledge has not been done before, e.g. García-Sedano et al. in [36] and Svensson in [1] simplify a press line optimization problem to a press station optimization problem. A positive outcome of the suggested methods in this thesis is fast building time of press line virtual commissioning models due to the modularization of the suggested model. Simulation building time refers to designing a detailed model of a Press Line. This is a time consuming task. The model of the Programmable Logic Controller (PLC) or control system that controls the Press Line is complex. Svensson et al. have stated in [1] that the PLC logic for a press line can consist of more than 100 000 lines of code, 2000 input/outputs and more than 100 adjustable parameters. Obtaining correct geometry and correct motion of the components of the line: Press, Die, Robot, Gripper [8], and line equipment, have also complexities. Technologies such as Laser scanning [9-11] and Laser tracking [12] help in virtual construction of virtual press shops [13]. Due to complexity, acquiring correct virtual control system and geometry affect the time frame of design and verification of a virtual model. A period of several months up to a year, dependent of the available resources, is not unusual.

The thesis also addresses run time of such models. The focus of this thesis is on the fields of geometrical simulation and virtual commissioning, covering Die Design [4] to Ramp-up [14]. Virtual commissioning in this thesis is referred to pre-programming and off-line programming of press lines. The aim of this thesis is to propose solutions to the following question: How are efficient geometrical simulation and virtual commissioning performed in stamping? In order to address this problem the *total simulation-optimization time* is investigated.

Total simulation-optimization time, which is a critical factor, refers to the total computation time for simulation and optimization of a model, optimizing input parameters e.g. path creation/modification or synchronization of robots and press. The total computation time is dependent on:

Simulation time
Optimization algorithm
Objective function
Optimization strategy

Simulation time in this thesis refers to the total execution time for simulation of press stations in a press line or simulation of a whole press line. The simulation of press line models can be categorized as computationally expensive due to the complexity of the models and consequently, time consuming. Iterative tasks such as optimization could demand several thousand simulation evaluations. Therefore decreasing the *simulation time* is a major aspect. *Simulation time* is dependent on several elements e.g. calculation of press line components, kinematic/geometrical updates and *collision detection*. *Collision detection* is one of the focus areas of this thesis. *Collision detection* is performed between dies, press, sheet metal parts and grippers. It aims at avoiding real collisions in introduction of a new product in the line, also referred as line tryout. In addition collision detection aims in avoiding collisions during ramp up until the decided cycle time is achieved. Collision avoidance is a must in a press station, since collisions between components in the line could lead to destruction of equipment, grippers, dies and even leads to human injuries. A collision may also initiate production delays, since unplanned operations such as gripper, die or robot repair must be performed. Since a 3D virtual representation of die, sheet metal part and grippers can consists of millions of triangles the *collision detection* is a computational expensive task. Therefore, the minimization of the *collision detection time* is one of the goals in this thesis.

The choice of *optimization algorithm* and *objective function* specifically the use of multi objective functions in press line simulation, impact result and resource consumption. The performance of an algorithm i.e. the efficiency of the code in finding an optimum impacts the computation time, since different algorithms use different strategies. Therefore different optimization algorithms could use different number of evaluations to reach a solution. Since the number of evaluations grows exponentially with the number of dimensions in an optimization problem, Svensson et al. in [1] suggests a new algorithm CoLis, which is based on combinations of optimization algorithms to reduce the number of evaluations.

The amount of dimensions in an optimization problem impacts the computation exponentially; hence different *optimization strategies* can be used in order to decrease optimization time. Dividing an extensive problem into sub problems (decomposition) or dimensionality reduction (Screening/mapping) are methods used when it is possible. A few of the questions regarding strategy have to be answered: Is it possible to use decomposition? Which strategies fit press line optimization and which of these strategies utilizes minimum time and resources, maintaining accuracy?

In order to answer these questions several simulation/optimizations strategies have been suggested and evaluated in this thesis with the goal to decrease computation cost.

1.1. Objective

The objective of this thesis is to investigate and suggest methods for simulation and optimization of tandem press lines. The objective is also to minimize *time* for simulation and optimization of press lines. Therefore the center of attention is on reduction of simulation evaluation time by efficient collision detection and reduction of evaluations through investigation of optimization strategies.

The industrial aim for this thesis is a useful fast off-line press line simulation parameter tuning method with directly transferable parameters from a simulation to the manufacturing plant.

1.2. Research questions

The main research questions are:

Q1: How are efficient geometrical simulation and virtual commissioning performed in stamping?

Q2: How can a feasible collision detection method be formulated to simplify complex and computationally intensive press line simulations?

Q3: Is it possible to establish an optimization strategy for press line simulations?

1.3. Main contributions

The main contributions in this thesis can be summarized in the following four items:

C1: A simulation concept including geometry, collision detection and optimization for press line simulation has been proposed.

C2: A module based approach for building press line models where design time is minimized has been established. The simulation model is parameterized and therefore applicable for other tandem press lines.

C3: A generic collision detection method based on 3D to 2D simplification for automated material handling with motion restricted to 2D has been proposed and developed.

C4: An optimization strategy for computationally expensive sequential automated models such as press lines has been proposed.

In Table 1 the relationship between the main contributions and each research question is illustrated. Further, the table shows in which appended papers the research questions are addressed and the contributions are presented.

Table 1. Illustration of the relationships: research questions – main contributions – appended papers.

| | | <i>Research question</i> | | |
|---------------------|----|--------------------------|---------|-----------|
| | | Q1 | Q2 | Q3 |
| <i>Contribution</i> | C1 | Paper II | | Paper III |
| | C2 | Paper II | | |
| | C3 | | Paper I | |
| | C4 | | | Paper III |

1.4. Limitations

There are some limitations in this thesis constraining the scope of investigation.

The proposed collision inspection method is limited to 2D automated manufacturing processes,

Manual optimization is not investigated in this thesis.

Only process parameter tuning is taken into account as an optimization method, even though there are other ways of improving performance such as modified control strategy and control code optimization, etc [1].

The proposed optimization strategies are carried out on a virtual automotive sheet-metal tandem press line only. Hence, they are not verified on a real press line.

1.5. Outline of thesis

This introduction is followed by Chapter 2, which summarizes related work. Chapter 3 describes the use of simulation in die, tooling design and the utilization of press line simulation. Chapter 4 describes the characteristics of a tandem pres line and Chapter 5 discusses the suggested collision inspection method. Chapter 6 enlightens the suggested optimization strategies for press line simulation. Finally, Chapter 7 contains conclusions and discussion of future work, followed by a summary of the appended papers in Chapter 8.

Related research

Geometrical simulation and virtual commissioning of press stations (press and sheet metal handling equipment) and press lines have not been in the focus of the scientific community. This field is relatively new, and research and development are still to be made in order to fully benefit from simulation as a quality tool.

2.1. Simulation within stamping area

Several examples of usage of simulation within stamping are described here showing the requirement of a clear strategy to be able to benefit from the wide range of alternatives within stamping simulation. Simulations are used within several different stamping fields from press and die design to maintenance as shown in Table 1.

Finite Element Method (FEM) simulations are common and are used in analysis of sheet metal formability including material properties, die design and springback [15, 16]. The aim is ensuring high quality, parameters such as formability, strength and processing without low quality parameters such as wrinkles due to overflow of material and cracks due to insufficient material flow.

In Press design, simulation of the press mechanism is used by He in [17] to control a hybrid mechanical press, including trajectory planning, trajectory optimization and real time feedback. The terminology hybrid within stamping is used in several different meanings dependent on the stamping level. In press machine level, He refers hybrid to the combination of a constant speed motor and a servo motor for driving the press.

Schuler in [2] refers to hybrid as mechanical-hydraulic presses. The hybrid drive system allows gentle impact of the top die on the part and also optimum control of the force exerted during the forming process. In contrast to hydraulic presses, hybrid presses offer higher output due to their mechanical basic drive system. In the press line level, hybrid refers to the combination of a lead hydraulic press and mechanical follow up presses.

Discrete Event Simulation (DES) is used in scheduling and logistic simulations. Press line scheduling simulations (material handling between press lines) problems are described and investigated in [18, 19] through implementation of algorithms to minimize the completion time of the jobs and using maximum machine capacity. Through dynamic simulation models, material handling is studied in [20]. The result is efficient throughput levels considering several input parameters such as schedules, production rate, racks etc.

Table 1. List of stamping simulation application, level of usage, simulation type and description

| <i>Application</i> | <i>Level</i> | <i>Type</i> | <i>Description</i> |
|---|-------------------------|--------------------|---|
| <i>Sheet Metal Part Design</i> | Pre-Production | FEM Geometrical | Sheet Metal Formability, Springback, Material Blank and Strip Layout |
| <i>Die Design</i> | Pre-Production | FEM | Analysis of: Structure: Material use, Strength, Face design |
| <i>Press Design</i> | Pre-Production | FEM Geometrical | Analysis of Structure, Forces Design of crank, shaft, ram, table, etc |
| <i>Scheduling</i> | Plant/Line | DES | Scheduling of jobs |
| <i>Logistic</i> | Plant/Line | DES | Planning, Buffer analysis |
| <i>Die Design (Internal)</i> | Pre-Production | Geometrical | Die Internal Functionality, Cam-Driver interferences etc. |
| <i>Die Design (External=Press Line)</i> | Line/ Station/Device | Geometrical | Die External Functionality, Gripper, Scrap interferences |
| <i>Packaging</i> | Line/ Station | Ergonomic | Design of packaging, work forces, overload health issues |
| <i>Offline Programming</i> | Line/ Station | Commissioning | Creation and Modification of programs offline without disturbance of the running production |
| <i>Service</i> | Line/ Station | Commissioning | Web, Distance Monitoring and Support |

Geometrical simulation within stamping is used as a tool of improvement striving higher quality through dynamic studies, time and interference analysis. Examples of dynamic studies are sheet metal vibration or scrap simulation. Time studies are used in die design by analyzing die internal mechanism motion, station or line cycle time. Interference studies are used for example in die design internal simulations to check for collision between mechanical components in the dies such as cam and drivers.

Die design external collision checks are performed between press line components such as grippers and dies. Grippers are used for sheet metal part transportation from one station to the other. The terms tooling and fixtures are frequently used in industry and scientific community instead of gripper. Grippers use different approaches in grasping parts. Vacuum cups, shovels or pneumatic pliers are examples of grasping mechanisms to hold on the part during motion. Gripper simulations for stamping are important since deformation during handling process can cause part dimensional variation jeopardizing the end result and limit transfer speed.

Gripper simulations are discussed in several papers. Hoffman et al. [21] studies ideal part holding positions by analyzing dynamic behavior of a the part during transportation in a crossbar transfer presses. They proposed a strategy to optimize material handling in automotive crossbar transfer presses. To reduce the holding forces and component deflections, they developed algorithms to automatically determine the vacuum-cup layout for a sheet metal part.

Ceglarek et al. uses in [22] a generic methodology for modeling and optimization of rigid tooling based on dynamic analysis of parts. The result is minimizing part deformation during transport. The methodology links FEM with optimization and includes part, tooling, and dynamic parameters.

Li et al. advances in [8] Ceglareks results from calculation on rigid grippers. Li proposes a dexterous model with the advantage of predicting part deformation more accurately. The

method calculates dynamic behavior of large sheet metal parts transformed with vacuum cups combining FEM with statistical data. Experiment is done on a blank sheet before press operations.

Callies et al. suggest in [23] a mathematical approach for modeling and control of multi link vacuum grippers. The gripper model is based on rigid body formulation and force calculations of vacuum cups are included during transfer.

Hossein et al. in [24] investigates flexible grippers for optimum positioning of suction cups. The term flexible refers to the possibility of using the same gripper for several different sheet metal parts due its motorized links.

Research in [25] by Moore et al. and Ng et al. [26] is focused on an internet-enabled 3D based virtual engineering framework, providing valuable functions including service and maintenance. The solution is applicable for machine monitoring and diagnostics enabling remote service support of discrete manufacturing machine systems.

2.2. Collision detection in simulations

Collision detection has been in the focus of the research community for a long time but to the author's knowledge no research paper handles efficient collision detection in press lines and more specifically collision detection in press line simulations and optimizations.

The 2D projection method mentioned in [27] is based on the idea that a precaution field including a moving object and an obstacle would function as a warning system for collisions. When the moving object enters the precaution field the alarm goes off. The obstacle has itself a repulsive field. The repulsive field of the obstacle would then force the moving object to choose another path. These fields are represented as the smallest circles where an object in a simulation could fit within. Although this method works well in scenarios where high collision accuracy, is not in focus, press line simulation demands high collision detection accuracy which makes this method inappropriate.

The method used in [28] is based on grouping objects close to the path of a moving object and then using the grouped objects as one unit in collision detection against the object in motion. In simulation cases where the amount of objects (solid or mock-up) is considerable, the clustering method mentioned in [28] would work excellent. But in simulation cases with few complex objects, the method would reach its limitation. The sweep collision detection method mentioned in [29] was used in order to detect collision for soft material such as cloth. The sweep technology is interesting though but was abandoned for a simpler solution. The swept volume is made by positioning a face first at time t and second in $t + \Delta t$ and storing the resulted volume.

A combination of [27-29] would give a new solution strategy for collision detection. The idea is based on slicing the objects in risk of a collision, projecting the slices in 2D, sweeping the 2D projection back to 3D and grouping the projections as one unit. This idea would have decreased the simulation model significantly but was neglected in this work for the benefit of a simpler method. Simplification of objects before collision testing is also discussed in [30]. Simplification methods usability is dependent on the simulation case. If the simulation is accuracy dependent, the simplification method would reach its limitation. In many industrial applications such as press line simulation high simulation accuracy is standard and necessary to obtain robust and well behaved solutions.

Inevitable Collision State named in [31] meets real time constraint imposed by dynamic environment which differs from the approach in this paper. The Inevitable Collision State methods efficiency is obtained by applying the principles 2D slicing, pre-computing and graphic processing. The pre-computing method used in collision detection algorithm in this

this thesis reminds of second principle in [31]. The pre-computing method determines whether a given state is to be avoided or not. In the same manner one of the algorithms developed in this thesis seeks in advance for possible collisions in the model taken into consideration the limitations such as path and direction.

Collision pre-computing is possible due to the fact that the transferred part in a press line follows the same path and the motion is repetitive. This means that a simpler representation of collision areas can be calculated and reused in optimization. By pre-computing, using a visualizing program for collision detection during iterative optimization will be unnecessary giving the benefit of decreasing time and resource consumption.

2.3. Simulation-based optimisation

Approaches of simulation-based optimization combined with varying types of virtual manufacturing systems can be found in the literature. Fu et al. [32] provide a descriptive review of the main approaches for carrying out simulation optimization and exemplifies the use of optimization simulations in several examples. Li et al. [33] have developed an optimization systems based on virtual machining where optimization is realized via modifying NC programs. Using metamodels and evolutionary algorithm Persson et al. [34] combined simulation and soft computing techniques to successfully optimize a manufacturing system.

2.4. Simulation, optimization and virtual commissioning in stamping

Geometrical simulation in stamping and virtual commissioning is focused on in several papers, while press line optimization has not been in the focus of science.

An evolutionary path-planning approach is used by Liao et al. in [35] for robot-assisted handling of sheet metal parts. Extracting robots path in a press station for sheet metal bending is studied. The proposed approach globally searches the motion path space to identify collision free paths. The focus of the paper is on the evolutionary algorithm. The question of how the simplified 2D station can represent 3D and time/resource consumptions in calculations is yet to be answered.

Virtual construction of a press shop, collision detection and material flow analysis are explored in [13]. A modeling standard for constructions is established and applied. The expectation of achieving great saving in time and cost is not further examined though.

Virtual commissioning is a step further, compared to the above examples. Its intention is to test manufacturing systems and associated control programs through simulation, before the real systems are realized [5].

A virtual commissioning scenario for stamping is investigated by García-Sedano et al. in [36]. Commercial software is used together with a generic algorithm. The aim is to use the generic algorithm to optimize industrial robots trajectory and minimizing cycle time in a stamping line. Kinematic constraints such as velocities, accelerations and jerk limitation are taken into consideration. The line is simplified though to a press station consisting of two robots and a press without considering the important synchronization of the components of the station.

A press line simulation model is designed by Svensson et al. in [1] including PLC and geometry with collision detection possibility. The system runs on a time synchronized environment [37]. The line is restricted to a station, due to its complexity with focus on different optimization methods. Simulating and optimizing the press station is

computationally expensive due to its complexity. The optimization tool consists of several optimization methods among others Nelder-Mead [38] and Direct [39]. The focus is on PLC parameters. Input parameters include robot paths, path related zones and start signals. Output parameters include production rate, collisions, velocity, velocity set points, accelerations and jerks. Path optimizations are possible with modifications of the existing model. The core of the simulation architecture is called synchronized distributed simulation protocol (SDSP) [40]. This architecture is designed to handle time and synchronization problems as well as distributed simulations. The simulation model uses real PLCs including all electrical signals, signal paths and hardwire logics in the manufacturing plant. A great advantage of using real PLCs to execute the real control code is the opportunity to directly transfer the tuned control parameters to the manufacturing plant [1]. Although this model would replicate the real model well, the model execution is time and computer resource consuming.

The optimization problem in this thesis is categorized as high dimensional design problems. Songqing et al. [41] discuss strategies for tackling high dimensional problems in the scientific community. These are divided into five different strategies: Decomposition, screening, mapping, space reduction and visualizations are the most common methods used see Table 2. Songqing et al. [41] recommend mapping and decomposition as promising methods. Mapping is described as transformation of a problem from an original high dimensional space to low dimensional preserving the optimum of the original function. Decomposition is described as transforming a problem to sub problems. Each sub problem can be addressed with suitable methods.

In this thesis several optimization/simulation strategies are suggested which can be categorized as decomposition. The aims of the strategies are to decrease the time for computationally expensive press line simulation models. The collision method used can be categorized as screening method where important collision points are sampled once and reused in the iterative simulation.

Table 2. Five different optimization strategies

| <i>Strategies</i> | <i>Description</i> |
|------------------------|---|
| <i>Decomposition</i> | Converting a problem into sub problems |
| <i>Screening</i> | Identifications of important variables |
| <i>Mapping</i> | Dimensionality reduction by transforming correlated variables into smaller set of uncorrelated variables. |
| <i>Space Reduction</i> | Reduction of ranges of variables |
| <i>Visualizations</i> | Finding key trends and relationships among variables and visualizing them to the user. |

2.5. Summary

Stamping simulation is a wide term and can include many areas such as sheet metal forming simulation, scheduling or virtual commissioning. Simulation and optimization in stamping has not been in the focus of the scientific community in the area of virtual commissioning. Although some papers discuss virtualization of press shops, material handling and press station optimization, to the author's knowledge no paper has been found on efficient press line optimization where effective building and running of such models are discussed and improved. It is the author's belief that the outcome of this thesis can be a significant

contribution in this field since for the first time, several successful simulations and optimizations of press lines will be performed and presented in this thesis.

Press Line Components and Simulation

In this chapter a brief general explanation of press line components and simulation is given. Information gathering, the base of a successful and reliable simulation is discussed. Two important components of the press line: dies and grippers are explained and analyzed. Interferences between these two components together with the press are the reason of press line simulation. Therefore a design strategy for dies and tooling where consideration is taken to press and robot motion limitations is one of the important factors in successful stamping production. Simulation as design tool quality checker, helps improving verification of dies and gripper before production start and during production.

3.1. Press lines

A press line is a set of presses in a linear formation. Press lines have been used in industry for a long time. Their main work is to reshape metal sheets to different components through successive operations. The goals in a press line are high output and high quality. A line has a short cycle time with high repetition accuracy and full control of the complex flow in the process [42]. The initial costs of a production line have to be recovered from the revenue for the components produced in the line. For that reason it is important that the production line has a high throughput rate, not only in the short run but also throughout the entire economic life of the line. Otherwise there would be a risk that the cost per produced component would exceed the possible calculated revenue [43]. 3D simulation of press lines, as a process planning tool, is one of the methods to keep and improve the throughput. Simulations are used in die design and tooling design with the goal of achieving high quality and production rate.

Press line components can be divided in three major groups: Presses, dies and material handling equipment which in turn can consist of robots and tooling.

Several sources of information are used when building a successful model over a press shop: Scanning, manual measurement, 2D and 3D CAD models, documentation and interviews with operation personnel of the press line of interest. A deeper explanation of scanning as information gathering method is explained below.

3.2. Scanning

Scanning as a source of data information gathering is used to complete geometrical models or to build models where no other information is available. In this chapter a brief introduction in

scanning within press shops is given. Two state of the art types of scanning are applicable for press line model building: 3D laser scanning and laser tracking.

3D laser scanning

Various techniques can be used to detail scan an object such as laser scanning to capture surface data of the tools and stampings [44]. Among other methods applicable for press shops are optical scanners. High accuracy is used to capture information about the object of interest. One problem with laser scanners and optical scanners is that they depend on line-of-sight; they cannot see undercuts or hidden surfaces. To completely scan in an object requires multiple scans from different views. The individual scans must then be registered together using available software and techniques [45]. This leads to the fact that typically 70-100 scans were made at strategic positions to collect data needed for mapping the complex geometries in a press line in [46].

Press shop scanning is used with the goal of making detailed computer models based on 3D scans, which will later be used in planning and simulation process to reduce down time in the press shop. The demands for accuracy are 5 times higher (2mm) in the press lines compared to traditional plant level scanning [46]. Today with accurate industrial computerized tomography complex surfaces could now be interrogated quickly and accurately. This is performed with non-contact laser and optical scanners with promising speed. [45]. 3D scene modeling based on using a laser scanner together with different digital cameras (texturing and stereo processing in different scales), and laser line projectors (structured light) are an example for multiple 3D recovering. Combining different sensors allows for acquiring any object with different levels of detail, e.g. fast digitization of a large object at medium resolution and refining some parts with higher accuracy afterwards [47].

Laser tracking

The laser tracker was developed as a portable coordinate measuring machine that can measure large or irregular structures [48] as well as complicated motions for multi-axis robotic machines.

A laser tracking system including position and orientation measurement constitutes an instrument to accurately determine robot performance as well as to acquire hints on how to improve robot models and control algorithms [49]. By the help of laser tracker motion of material handling system can be tracked and analyzed resulting in input data in simulation.

3.3. Die design

To construct and design dies is in many cases a demanding and complicated task, where the outcome is highly dependent on experience and craftsmanship of the designer. The final shape of the sheet metal part must be taken into consideration since it also influences the choice of press type and how many forming stages are needed, which in turn is also dependent on the product shape [50].

A press die consists of two major parts, upper part of a die; Die Upper and lower part of the die; Die Lower shown in Figure 2. A sheet metal part is placed between the die parts and the two parts are pressed together which leads to changes in sheet metal. Changes can be forming, trimming, flanging, piercing, etc.

The upper part and the lower part in turn consist of movable mechanical parts e.g. cams and drivers. The motions of these cams are dependent of the motion and contact of the Die Upper and Die Lower. Blank holders are used to grasp the sheet metal part while the matrix (form) is pressurized forcing the upper part against the lower.

Production of dies are both time consuming and expensive, involving: design, casting, milling, assembly, function control and line tryout. This process requires high quality in all steps. As a consequence simulations of dies are used as a quality tool.

3.4. Die simulation

Today die design is done in 3D with the help of CAD programs. The design of the die could take from a couple of weeks to 2-3 month. The sheet metal part which is the product which the die will eventually form is prepared in a first step by a process engineer. The final product is sent to a die designer where the designer, designs the die either from the beginning or with the help of a template. Step by step, the die is formed virtually with the necessary parts, cams, drivers, pneumatics and electric sensors.

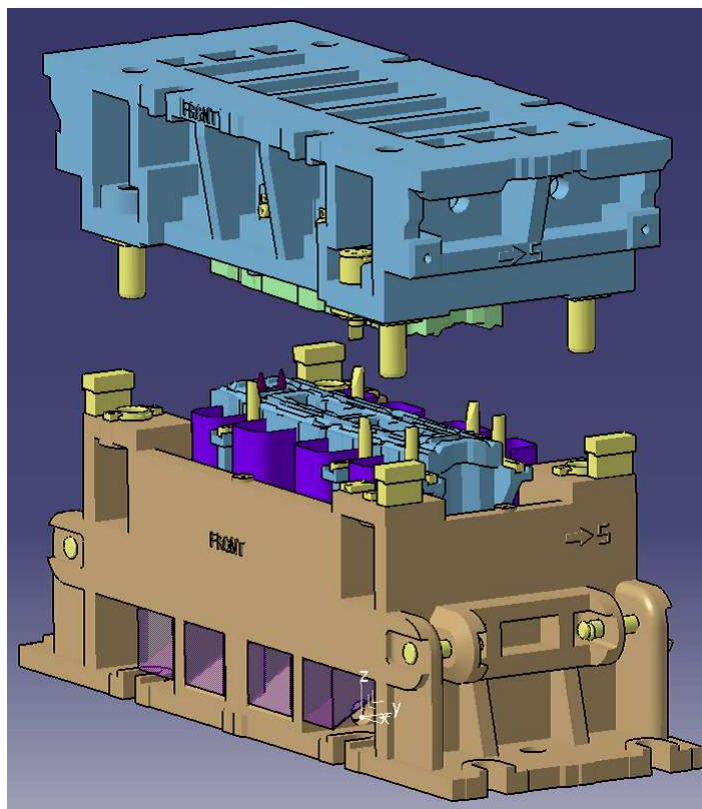


Figure 2 Press die consisting of two major parts, Die Upper and Die Lower

If simulation is an incorporated method, it is used in two major scenarios namely: internal collision detection and external collision detection (Press line simulation).

3.5. Die internal collision detection

In internal collision detection, simulation is used to help the designer in time scheduling and internal part collision detection. Time scheduling refers to tuning and adjustment of the angles of the cams and drivers which in turn will affect the interactions of the cams, blank holder and other movable parts in the die. Other examples are the adjustment of the cam slider length and damper length.

Internal collision detection is used while the design of the die is preceded and as a last step before the virtual model is sent for casting and refers to check for collisions between the movable parts in the die.

3.6. Die external collision detection

External collision detection which is also referred to as press line simulation could be divided in three different cases: Before Die Design, Die Design, After Die Design.

Before die design

Before a die is designed, the designer gathers information about the press the die is intended for. Force capability of the press, maximum die size and clamp limitations are among parameters the designer searches for. Motion of the mechanical equipment of the press is an important parameter to take into consideration. Information is available either in documents or in virtual tools. The motion of material handling equipment can be visualized through collision risk volume also referred to as sweep volume [51].

A sweep volume is created with the help of the movement of sheet metal part and the tooling from one station to the other based on the motion of the material handling robots or equipment. The sweep volume is created by transferring sheet metal part, tooling and parts of the material handling equipment in risk of collision. These parts are transferred in the planned path incrementally while the program records the volume in space occupied. The result is a swept volume which could be used as a reference.

When designing the lower part of the die this swept volume must not be in collision with the die otherwise a collision will be inevitable. This method could be used also to avoid collision with the upper part of the die by setting the upper part as a reference, which means that the swept volume will be the result of the relative movement of the upper part of the die, gripper and the sheet metal part. In the same manner this volume can be used as a reference in the design of the upper part of the die.

Die design

In transfer press lines with a master axis system where the motion of the material handling equipment is fixed the designer can not impact the cycle time of the press through alternative die solution. The die designer task is to make sure the die fits the specifications of the press avoiding collision with press and with material handling equipment. The die designer has an extra dimension to consider for dies intended for tandem press lines. While a die is being created, the designer can impact the cycle time of a tandem press line by choosing different die solutions. This opportunity needs reliable press line simulations which in turn help creating high quality dies for production.

After die design

Press lines simulations are also done late in the production process in so called line tryout. Line tryout is the stage where die are ready for production and are introduced for the first time to the press line. Tooling is either pre-fabricated or finalized online. For programming of the press line an existing program is used or a new program is created. Press line simulation including offline capability and optimization can help programming the line to its best performance based on the shape of the dies and tooling. A change in this stage of the dies in order to achieve better line performance can result in delays in the production. Changes can include casting, milling, assembly and function control.

Although simulations are possible in this stage an early simulation in designing dies and tooling can reduce cost and lead to higher line performance.

3.7. Gripper

The importance of tooling is emphasized in a test done by [52]. The result of the test showed that most of the problems in product and process design, are caused by gripper installation, gripper maintenance and incoming material variation. This result demonstrates the significance of press line simulation. Each station in a tandem press line handles one or two grippers. One gripper is for the incoming part and one for the pressed part. If i is the number of press stations in a line, this fact means that an automotive made by 200-300 hundred sheet metal parts needs $2*i*(200-300)$ grippers. Hossein et al. [24] suggests a flexible transfer system, which is able to handle various parts without changing any components hence decreasing the number of grippers needed.

Dependent of the goal for a press line simulation, different resolutions in tooling design can be used. Three different types of tooling is suggested here: Light, Rough and Detailed tooling design

Light gripper design

Light tooling design refers to a volume being used as a tooling. In its simplest form the light tooling design has the shape of a box and as a more advanced form the shape of the light form will remind of the finished desired tooling. The light tooling stands for the far most allowable limits in the desired tooling. This means that when a simulation is completed with a light tooling, the final tooling's shape must fit within the limit of the light tooling shape, otherwise risk of collision is inevitable. The results could be used directly on shop floor when building the tooling or as a pre step when virtually building the tooling. This method offers a simple shape for use in simulations. Due to its simplicity, complex motions and collision detections can be performed at a reduced time and calculation cost. It also offers the freedom of design within its limitations for gripper designers or shop personnel.

Rough gripper design

Rough gripper design refers to using templates. This means that rough shapes of gripper can be used dependent of the final product of interest. In the simulation scenario, the simulator chooses a rough gripper which reminds of the shape of sheet metal part the gripper is intended for. When using a template the rough shape of the gripper covers up for the most dangerous collisions. A gripper designer or line personnel can in a later stage complete the gripper with extra equipment such as suction cups. The benefit of this method is the ability to use simple base (template) gripper fulfilling the specifications of a complete gripper in fast simulations and visualizations intend for analysis.

Detailed gripper design

Detailed gripper design refers to using a complete virtual designed gripper. Although these models are computationally demanding, they offer the benefit of a detailed CAD model. In a simulation scenario a more detailed model results in a more accurate simulation result which in turn means higher quality. When using a detailed model two forms of design methods could be used: Forward kinematic or Inverse kinematic.

Forward kinematics [53] in gripper design refers to placing a suction cup or a movable part of the gripper joint by joint in the grasping point of intersect on the sheet metal part.

Inverse kinematics [54] refers to pointing out the grasping point of interest and the grasping point on the gripper and letting the computer calculate the joint positions of the gripper. Inverse kinematics could also be combined with morphed solids. This means that not only the joint are computed automatically but also the shape of the gripper is morphed along its axis to fit the grasping links and joints. This result in fast gripper creation where inputs for gripper creation consists of: grasping points and gripper template.

The three suggested gripper design methods above could be used either as a sequence in gripper design based on a simulation or separately in designing grippers for the press shop. The benefit of the first two suggested methods is computation speed. Since collision detection is time consuming these two methods offer faster collision detection due to their simplified forms. In this thesis light gripper design is used as input in collision method.

3.8. Summery

Press, dies, robots, tooling and sheet metal parts are major geometrical components in a press line simulation. Due to their complexity, simulation will ease analysis of the process. In order to perform geometrical simulation, data about the process and line components are needed. In this chapter scanning was pointed out as one of many information gathering tools.

Late changes on dies and tooling are costly. Hence simulations allow die and process designers to forecast problems leading to higher efficiency, higher quality and revenue.

Tandem Press line

A deeper understanding of the motion and control of the presses and material handling equipment is necessary in order to understand optimization possibilities and limitations of the line. This section gives a detail explanation of the tandem press line used in this thesis. Signal dependency and motion is analyzed and discussed.

4.1. Press line description

In this study a sheet-metal press line, no. 53 at the Volvo Car Manufacturing plant in Gothenburg, is used as a case study. This actual press line has five press stations. Figure 5 visualizes in detail the combination of stations in the press line.

The stations embody a mechanical press with fixed displacement-time curve, associated die and robot. In the press line there are four specialized 2D-robots feeder/extractors, so called Unifeeders, supplied by Binar AB (Binar). The robot feeding the press is called feeder and the robot extracting sheets from the press is called extractor. Furthermore, there are four intermediate stations, which individually can move vertically and rotate around horizontal and vertical axes [1]. The station shown in Figure 3 consists of an upper/lower die, press, 2D robot feeder called Unifeeder, intermediate station, gripper and sheet metal part.

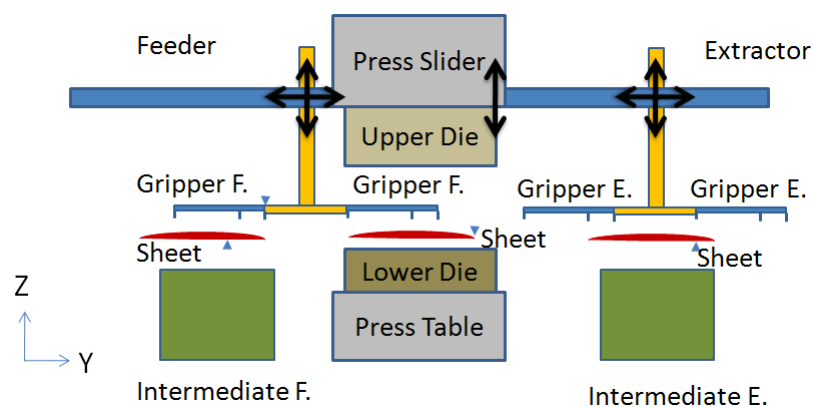


Figure 3 Press Station description

Press position is measured in the Z direction and feeder and extractor positions are measured in the YZ-plane. The motion for each component in the station is a function of an angular position referred to as cam value. Thus, for a specific cam value, a specific position of

the component in space exists. The Unifeeder handles the movement of the part. It picks up the sheet metal part from the intermediate station and places it along a programmable path in to the die. Movement is limited to the YZ-plane, giving two degrees of freedom.

While the press is moving down from top position, the Unifeeder puts in a sheet metal part in the press. The Unifeeder must leave the press before the press reaches its lowest position. The path of the Unifeeder is settled before production starts as well as the critical synchronization between the Unifeeder and the Press. The Unifeeder could be as near as 2mm to the die when entering (feeder) or leaving (extractor) the die. This forces the virtual models and methods to be as accurate as possible.

The cam value measurement technique is a heritage of the mechanical presses where the position of the press ram is a function of angular position of the crank. This technique is used for synchronization of the equipment in the line. Modern equipment such as robots and hydraulic presses lacks real cam values. However, the technique is often used for synchronization purposes, but with virtual cam values instead. A full cycle describes the motion of the equipment from start to end and back to starting position measured from 0 to 360 degrees, see Figure 4.

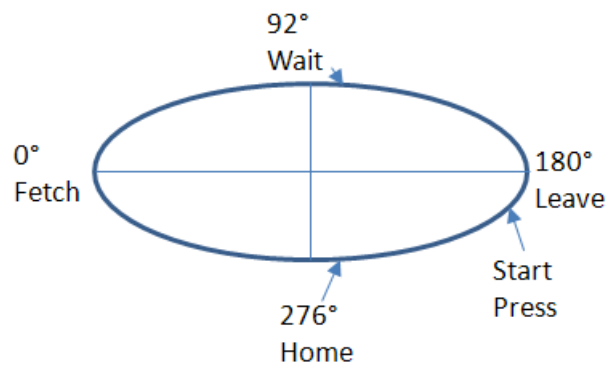


Figure 4 Representation of a full cycle for a robot in a tandem press line including cam values, path position and signals

4.2. Synchronization

Several stations combined in a linear formation are called a Tandem Press Line. In a press line a set of four to six presses is normal. An extractor in a station works also as a feeder in the next station. This fact increases the complexity of tuning the line dramatically.

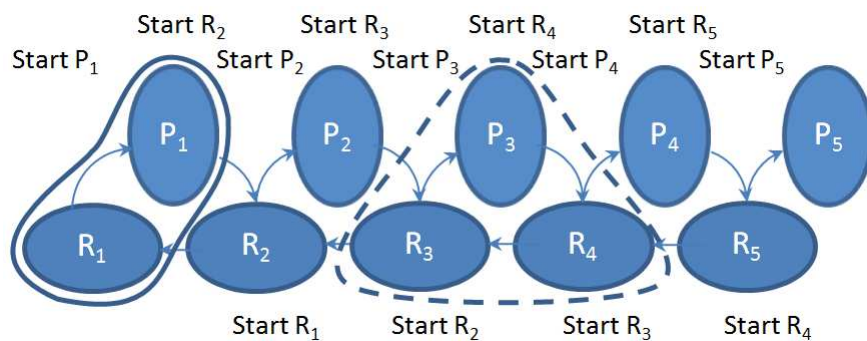


Figure 5 Representation of a tandem press line with signal description. The solid marking represents a station. The dashed line represents the combination of two stations. Robot R, Press P

Figure 5 describes the combination of five stations and their signal transmission. Each machine starts the next by a start signal at a programmed cam value.

Figure 6 illustrates the signal S transmission of Press P and Robot R of station i :

- Signal $S_{R_i P_i}$ represents Robot R_i starting press P_i . R_i is also working as an extractor for the robot R_{i-1} in the former station.
- Signal $S_{P_i R_{i+1}}$ represents press P_i starting robot R_{i+1} .
- Signal $S_{R_{i+1} R_i}$ represents robot R_{i+1} starting robot R_i which is an extractor. R_{i+1} is also working as a feeder for the press P_{i+1} in the next station.

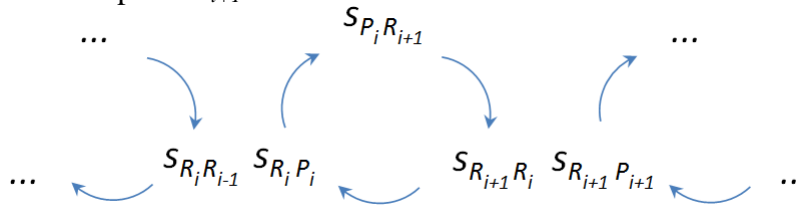


Figure 6 Signal transmission in a press and its closest robots

Tuning of start signals plays a major role in synchronization between the components. Since the components in the station are not mechanically combined, cam values are the sharing variables used for synchronization. Besides the starting signals between the components in the station, several other parameters affect the motion of the components and their relationship. The movement of feeder, press and extractor could be divided into three major parameter groups namely motion, path and signal, see Table 3.

Table 3. Input variables and signals affecting synchronization in the station. (N_p =number of positions)

| | <i>Motion feeder</i> | <i>Motion press</i> | <i>Motion extractor</i> |
|---------------|--------------------------|---------------------|--------------------------|
| <i>Motion</i> | Variable | Fixed | Variable |
| <i>Path</i> | N_p Position variables | Fixed | N_p Position variables |
| <i>Signal</i> | Start feeder | Start press | Start extractor |

Motion stands for parameters: velocity, acceleration, deceleration and jerk, all affecting the movement of the components in the line. In this case study, due to the real press mechanical attribute, the motion press curve is fixed, meaning that motion parameters are fixed for different cam values. Feeder and extractor motions are dependent on the velocity set points. The feeder and extractor have several positions along their path, namely: Fetch, Leave and Position $1 \rightarrow N_p$ where N_p is the number of positions between Fetch and Leave. Fetch is the position where the sheet metal part is grasped in the press before. Leave is the position where the sheet metal part is left in the press, waiting for a press strike. The start signals and velocity set points are critical concerning performance and are today tuned by operators. The goal is to determine these start values by optimization [1].

4.3. Motion

In this section the relationships between the motions and signals in Table 3, are explained for the components in the station.

Press motion

Press motion depends on a press motion curve m_{P_i} and the press start signal $S_{R_i P_i}$ initiated from the robot of the station. At the time instance $t(S_{R_i P_i})$ the press starts a press strike. Press motion is measured in height (mm) / degrees. In the studied line the presses are mechanical with fixed press motion curves. The motion curves of mechanical presses (crank, linkage knuckle-joint) are fixed compared to force controlled presses [55]. A cycle in a press starts at the top position normally a cam value of 0 degrees down to the lowest position at 180 degrees and up to 360 degrees. The combination of press motion curve m_{P_i} and press start signal $S_{R_i P_i}$, results in a function $f_{Press Motion}(m_{P_i}, S_{R_i P_i})$, which generates the height of the slider.

Feeder/Extractor motion

The function $f_{Feeder Motion}(m_{R_i}, S_{P_{i-1} R_i}, S_{R_{i+1} R_i})$ returns the position of the tool centre point (TCP) of a robot along the path where m_{R_i} is the motion controller of the robot.

The starting signals in the station affect the motion of the feeder robot. If needed, R_i waits at a wait position set by cam values until signal $S_{R_{i+1} R_i} = 1$, meaning that the extractor has grasped a sheet metal part and is on the move out of the press. R_i waits also at a home position set by cam values until signal $S_{P_{i-1} R_i} = 1$ before entering press in the station before.

Figure 7 illustrates $f_{Feeder Motion}$ in relation to time where the time is divided as follows:

- | | | |
|----------------------------|---|-----|
| $t(S_{R_{i+1} R_i} = 0)$: | The time when robot R_i stops in Wait position before entering Press P_i | |
| $t(S_{R_{i+1} R_i} = 1)$: | Start time for robot R_i to enter press P_i (Signal from robot R_{i+1} to robot R_i) | |
| $t(S_{P_{i-1} R_i} = 0)$: | The time when robot R_i stops in Home position before entering press P_{i-1} | (1) |
| $t(S_{P_{i-1} R_i} = 1)$: | Start time for robot R_i to pick sheet metal part from press P_{i-1} (Signal from press P_{i-1}) | |

Robot motion

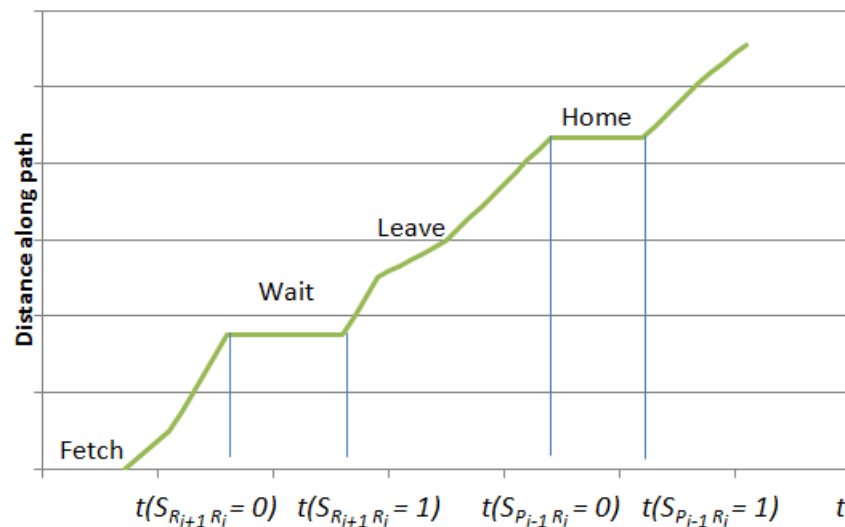


Figure 7 Illustration of robot motion curve with waiting time

The times in equation (1) are not constants. They are dependent on the starting signals. Hence they can vary and are dependent of other robots and presses of the press line.

Velocity, acceleration, deceleration and jerk of the robots are calculated by derivation of $f_{Feeder Motion}$. The motion curve of the extractor resembles feeders curve and therefore is not described in detail. The time intervals $t(S_{R_{i+1}R_i} = 1) - t(S_{R_{i+1}R_i} = 0)$ and $t(S_{P_{i-1}R_i} = 1) - t(S_{P_{i-1}R_i} = 0)$ in equation (1) are waiting times, since no motion is conducted during these intervals.

4.4. Variable waiting time

The goal of the optimization of a station is to decrease the cycle time and ware for the robots and the press. If waiting time is δt_j , then the total waiting time in the station is $\sum_1^3 \delta t_j$, where j is the index for the different delays, (See Figure 7).

$$\delta t_1 = t(S_{R_{i+1}R_i} = 1) - t(S_{R_{i+1}R_i} = 0) \quad (2)$$

$$\delta t_2 = t(S_{P_{i-1}R_i} = 1) - t(S_{P_{i-1}R_i} = 0) \quad (3)$$

$$\delta t_3 = t(S_{R_i P_i})_{cn+1} - t(S_{R_i P_i})_{cn} - T \quad (4)$$

i = Station number, cn = Cycle number, T = Press Cycle time

Equation (2) describes the time difference between start signal of robot R_{i+1} (extractor) to R_i (feeder) and feeders wait signal. Equation (3) describes the waiting time for extractor until press P_i gives start signal. Notice that the extractor works also as a feeder for the next station in the line. Equation (4) expresses the waiting time for the press before start. The time should be zero in a cycle time optimal press line (high throughput) implying that the press strokes continuously without delays between each stroke. A scenario with minimum (δt_3) indeed results in high production rate, with the down side of increasing ware on the feeder and extractor due to high accelerations and jerk. Hence this results in higher requirements on the sheet metal handling equipment.

4.5. Summary

Tuning of start signals plays a major role in synchronization between the components. The movement of feeder, press and extractor are divided into three major parameter groups namely motion, path and signal. The relationships between the motion curves are dependent on the tuning of the station for tandem press lines. The components in the station are not mechanically combined, cam values are shared variables used for synchronization.

Collision Inspection Method

Since several objects are in motion in a press, checking for collisions virtually is computationally heavy. Hence, for iterative tasks such as optimization, visualizing object activity and collision is a demanding task. Therefore an innovative collision detection method is suggested below.

5.1. Collision Inspection Method

The Collision Inspection Method (CIM) is a method based on the following approach: Pre-calculating all collision points in an environment of interest, and using the simplified result in an optimization simulation, is less resource consuming than collision checking the original objects for all optimization iterations. If these points are known in advance, efforts can be concentrated on e.g. finding optimal robot path or optimal start and stopping degrees for the robot. CIM can eliminate the need of visualization. It is a general method that can be used in all scenarios where parts are transferred in 2D or 3D with an optional small stochastic variation in the third dimension.

CIM depends on a general path/direction, solid geometry (none flexible, cloth etc.) and no stochastic rotation of objects. The first part of CIM handles detecting and storing collision points between objects in risk of collision and compressing the result to collision curves. This part of CIM is a pre-process and is only calculated once. The second part detects collisions between a chosen point on the moving object and the collision curves.

Depending on the sampling rate in CIM in 3D and six degrees of freedom, the result of CIM will be a simplification of the original objects. In 3D with motion restricted along a surface, collision points would result in 3D curves. In 2D motion, which is common in press lines, the outcome of CIM is 2D collision curves. Each pair of components in a collision set in CIM results in a chosen TCP and a collision curve. The TCP position is checked against the pre-calculated collision curves in a simulation scenario, instead of complete geometry collision checks. Hence, 2D collision curves are less resource demanding in a repeated simulation.

Figure 8 illustrates the two parts of CIM. Part one handles collision detection and sampling between objects of interest and part two handles collision detection between the simplified collision curves.

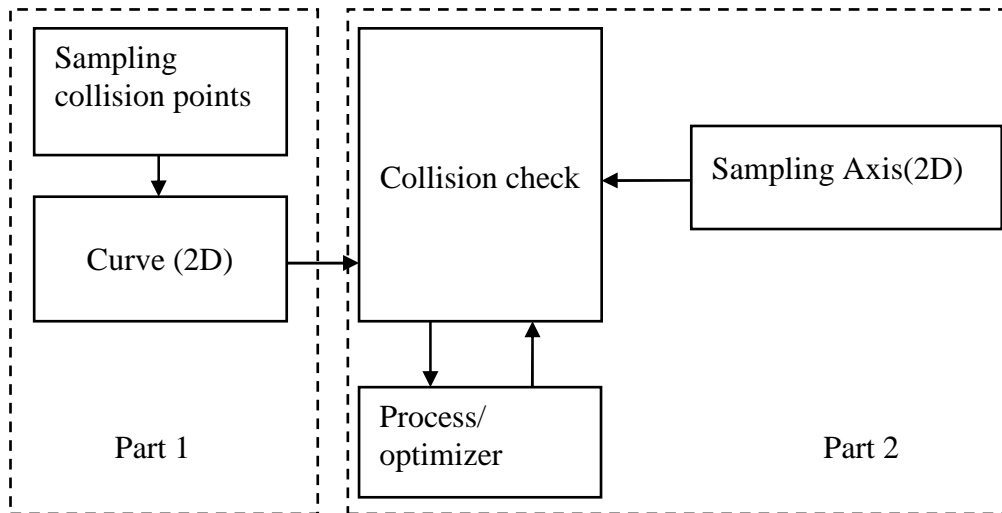


Figure 8 Description of the CIM method

5.2. CIM part 1: Sampling collision points between objects

Sampling collision points is the first step of CIM. Collision points are sampled from all objects of interest and then transformed to collision curves.

While sampling, the collision points build up a collision cloud, which is a group of points representing collision points between Object 1 and Object 2. The collision cloud can either be produced around Object 1 and/or around Object 2 as visualized in Figure 9. Although Object 1 and Object 2 can be reduced to their simplified collision clouds representations in a simulation scenario, a further simplification is possible in this case due to movement limitations in the YZ-plane.

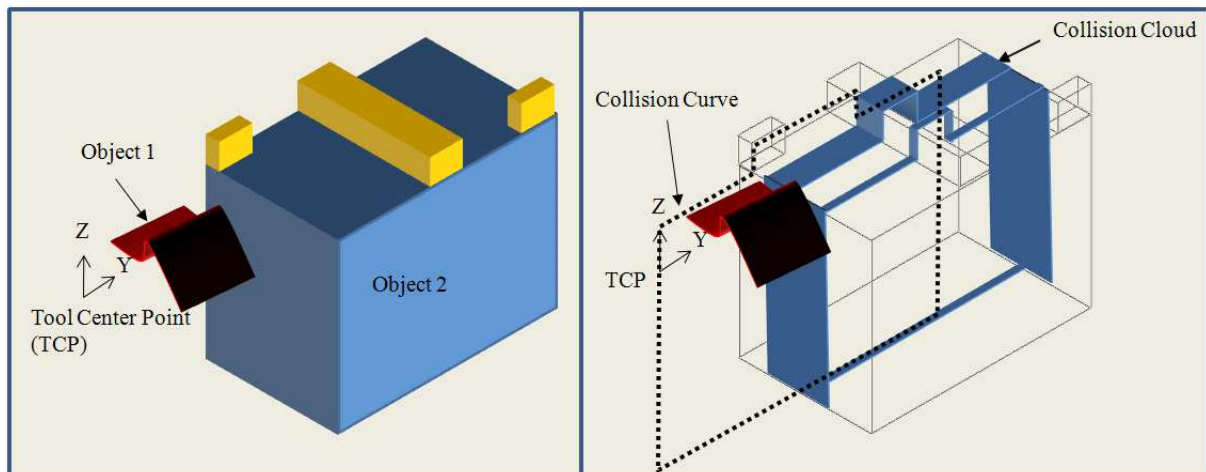


Figure 9 Collision check between Object 1 and Object 2, resulting in a collision cloud around Object 2 and collision curve sampled in TCP

When a point is registered as a collision point, a new point is stored in TCP coordinate. Because of the stochastic variables in real life contributing to displacement of the Object 1 during transportation, $\pm x$ in movements in X-direction must be taken into consideration. These variables include e.g. fluctuation in oil amount on the sheets and vibrations in a press

line. This leads to the fact that in the virtual model, a collision detection with $\pm xmm$ in X-direction is also done for every $+y$ value with d as sampling value.

Given CAD models $O_{1,2}$ and minimum sampling distance d , a collision detection algorithm f_C returns:

$$C = f_C(O_1, O_2, d, D, TCP(O))$$

C = Collision curve
 $O_{1,2}$ = Object 1 and Object 2
 d = Sampling distance dx, dy and dz
 D = Direction vector in $\pm Y, \pm Z$
 $TCP(O)$ = Chosen TCP of object

(5)

The collision points in C are sampled in a chosen Tool Center Point (TCP) for a chosen object. To sample C for $O_{1,2}$ in the YZ-plane four direction vectors $D(\pm Y, \pm Z)$ are used. Sampling distance $d(dx, dy, dz)$ is a chosen distance between sampling points in X, Y, and Z-direction. d impacts the accuracy of the collision curve and is adjustable. As a last step the collision curve is segmented and neighboring points with the same y or z values are removed except the first and last point in the neighboring segment. In this way further unnecessary points are removed.

A simplified part of the code for CIM is visualized in Figure 10. The code describes creation of a collision curve between lower part of Object 1 (O_1) and upper part of Object 2 (O_2). O_2 is fixed while O_1 is movable. Collision points are sampled and saved in the chosen TCP until O_1 reaches a point in y where no collisions are detected in $\pm Z$ -direction.

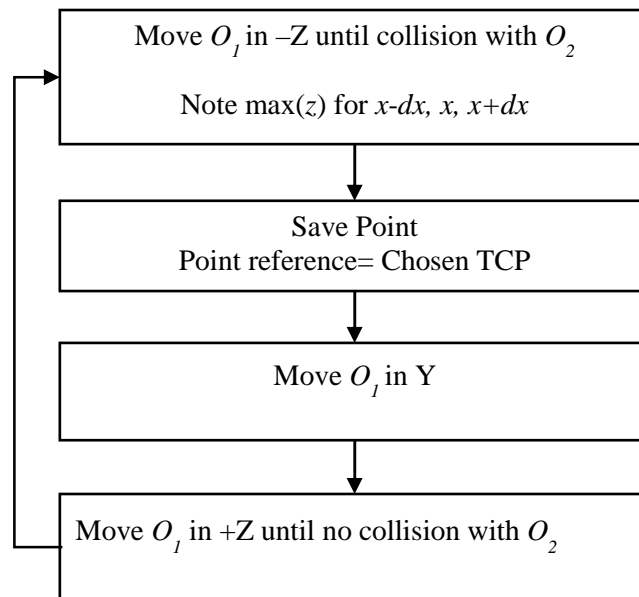


Figure 10 Part of algorithm f_C describing creation of a collision curve on the top of object 2

Object 1 is moved in $+Z$ until it is collision free. Then Object 1 is transformed in $-Z$ for $x-dx, x$ and $x+dx$ until collision is detected. The highest z value is noted. The same procedure is repeated until all collision points are sampled in TCP from the top of the Object 2.

In the same manner other sides of the Object 2 are sampled in TCP. x values of the points are set to zero and TCP points are then saved in a vector, resulting in a 2D collision curve.

5.3. CIM Part 2: Collision check between TCP and Collision Curve

The chosen sampling point TCP is used for collision detection instead of its geometry and collision curve is used as a representation for Object 2, which will drastically reduce computation time. This means that in a simulation scenario TCP of Object 1 is checked against the collision curve of Object 2 instead of geometry of Object 1 against geometry of Object 2.

Several collision curves can be used as input values together with TCP in a simulation scenario. In a simulation scenario TCP and collision curves can be movable independently in the YZ-plane.

5.4. Creation of Collision Curve for a die

As the first step of CIM, collision points are sampled from objects of interest in the station: Die Upper (DU), Die Lower: (DL) and sheet metal part. Sampling points are then transformed to collision curves. The accuracy of the collision curves is dependent of the sampling distance. A sampling of 2 mm is used in this case since 2 mm is the accepted geometry difference on the real press line.

Because of the uncertainties in real life contributing to displacement of the model sheet metal part during transportation, ± 2 mm in movements in x direction must be taken into consideration. These variables include fluctuation in oil amount on the sheets and vibrations. x values of the points are set to zero resulting in two 2D curves called collision curve upper and collision curve lower. A third collision curve is also produced originating from collisions between sheet metal part In and sheet metal part Out and their grippers. This collision curve is called Collision Sheet (CS).

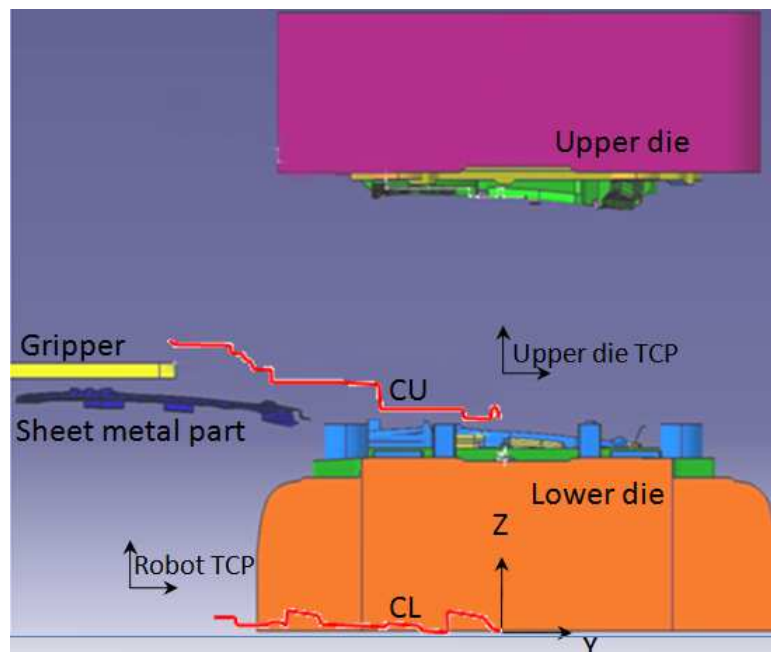


Figure 11 Result of CIM in a collision scenario

Figure 11 illustrates a press die, with its main parts, upper die and lower die. The lower die is fixed, while the upper die follows the press motion. A sheet metal part is held by a robot gripper.

The collision sets in CIM include collisions between the gripper, the sheet metal part and the lower die, which generates the curve Collision Lower (CL), and collisions between the gripper, the sheet metal part and the upper die, which produces the curve Collision Upper (CU). The collision curves are created by repeatedly sampling collisions in a chosen TCP, while objects are colliding in the collision set.

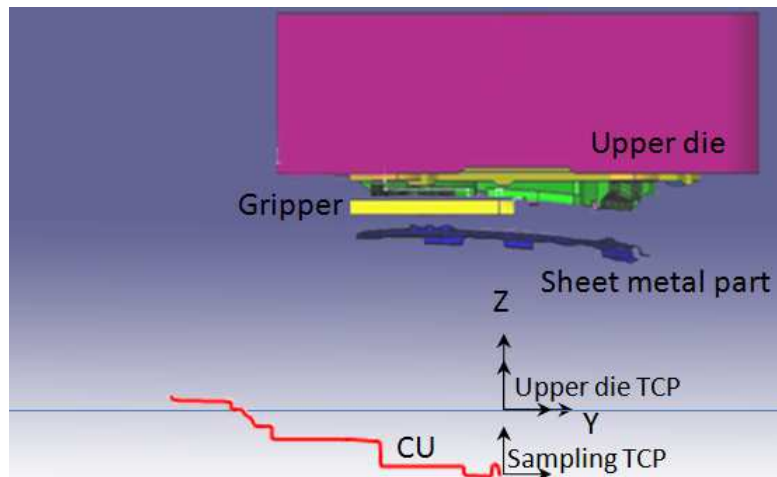


Figure 12 Creation of the curve CU

Figure 12 illustrates the creation of the curve CU. The chosen sampled TCP, which has the same local axis system as the sheet metal part, is located at (0,0) when the part is placed in the die. The sampled TCP is used to create the curve CU. The CU position and form is dependent on the height and shape of the gripper and the upper die. In Figure 12 the sheet metal part, the gripper and its local axis system, the sampled TCP are shown at the first created point of curve CU. This means that the gripper, the sheet metal part and sampled TCP had to be lowered due to gripper height from $z=0$ in order to detect the first collision at the top-right of the gripper.

5.5. Collision check between TCP, CU, CL and CS

The chosen sampling point TCP is used for collision detection instead of full geometry of the sheet metal part. The collision curves CU and CL are used as representations instead of upper die and lower die. In a simulation scenario TCP is checked against CU and CL as seen in Figure 13. The curves CU, CL and chosen sampling point on the robot TCP are set as input values in the algorithm. The algorithm checks if point TCP_z is between CL and CU while TCP follows the robot path.

The curve CU is a moving curve in the Z-direction, following the same direction as the press and the curve CL is fixed. This means that the collision free area is flexible in Z-direction and TCP must get out of the press (collision free area) before CU reaches its lowest value otherwise a collision is inevitable.

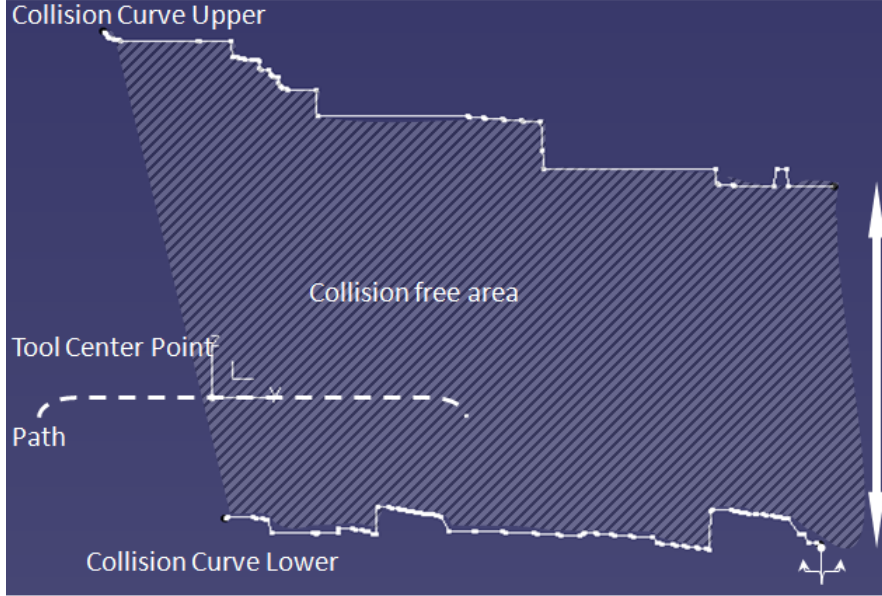


Figure 13 collision detection between a robot and a press in a virtual press station,

The equations that determine when collision occurs in station i , according to CIM are:

$$\begin{aligned} z_{P_i} + z_{CU_i}(y_{R_i}) &\leq z_{R_i}(y_{R_i}) \\ z_{P_i} + z_{CU_i}(y_{R_{i+1}}) &\leq z_{R_{i+1}}(y_{R_{i+1}}) \end{aligned} \quad (6)$$

$$\begin{aligned} z_{CL_i}(y_{R_i}) &\geq z_{R_i}(y_{R_i}) \\ z_{CL_i}(y_{R_{i+1}}) &\geq z_{R_{i+1}}(y_{R_{i+1}}) \end{aligned} \quad (7)$$

$$\begin{aligned} f_{inpolygon}(y_{R_i}, z_{R_i}, y_{CS_i} + y_{R_{i+1}}, z_{CS_i} + z_{R_{i+1}}) &= True \\ i \in \{1 \dots n_s\} \text{ where } n_s &= \text{last station number} \end{aligned} \quad (8)$$

$$\begin{aligned} z_{P_i} &= \text{Press } z \text{ values } z_{P_i}(t) \\ y_{CU}, z_{CU} &= \text{Collision Upper } y, z \text{ values} \\ y_{CL}, z_{CL} &= \text{Collision Lower } y, z \text{ values} \\ y_{CS}, z_{CS} &= \text{Collision Sheets } y, z \text{ values} \\ y_{R_i}, z_{R_i} &= \text{Feeder Robot } y, z \text{ TCP values } y_{R_i}(t), z_{R_i}(t) \\ y_{R_{i+1}}, z_{R_{i+1}} &= \text{Extractor Robot } y, z \text{ TCP values } y_{R_{i+1}}(t), z_{R_{i+1}}(t) \end{aligned}$$

Equation (6) describes that z_{CU} follows the press motion z_{P_i} while robot R_i and R_{i+1} TCPs must be below CU to avoid collisions. Equation (7) explains that the robots TCP must be above CL to avoid collisions. Equation (8) describes the collision criteria for collision sheets (CS) which is a collision curve based on collisions between the incoming sheet and the outgoing sheet. The equation checks for collisions between feeder robot TCP values and CS which is following the TCP of the extractor robot. This function is applied while y_{R_i}, z_{R_i} is located between min and max of y_{CS}, z_{CS} while y_{CS}, z_{CS} follow extractor robot TCP y values.

Figure 14 illustrates several collision curves in a simulation scenario against a TCP. A collision occurs when the TCP is within the limitations of the curves. The figure also illustrates collision curves In and Out which are based in collisions between sheet metal part In/Out verses die.

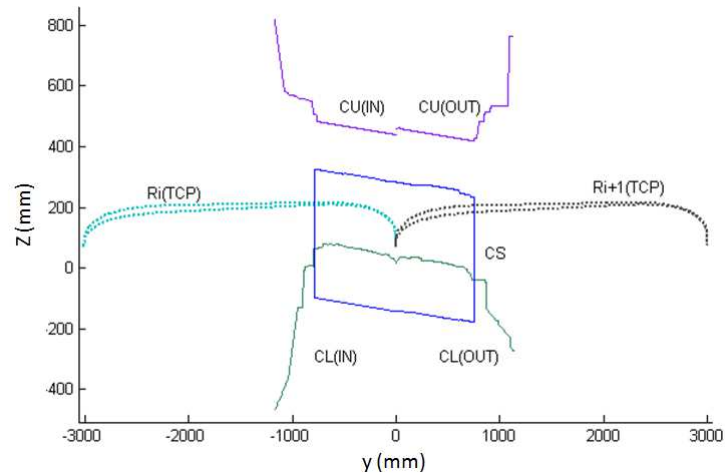


Figure 14 Motion path of robot R_i and R_{i+1} TCP and collision curves for a station

5.6. Relative motion curves

To graphically illustrate a possible collision scenario, relative motion curves can be used. Relative motion curves are a combination of motion curves and path and collision curves. Figure 15 shows the motion curve of a real industrial press and a virtual feeder robot.

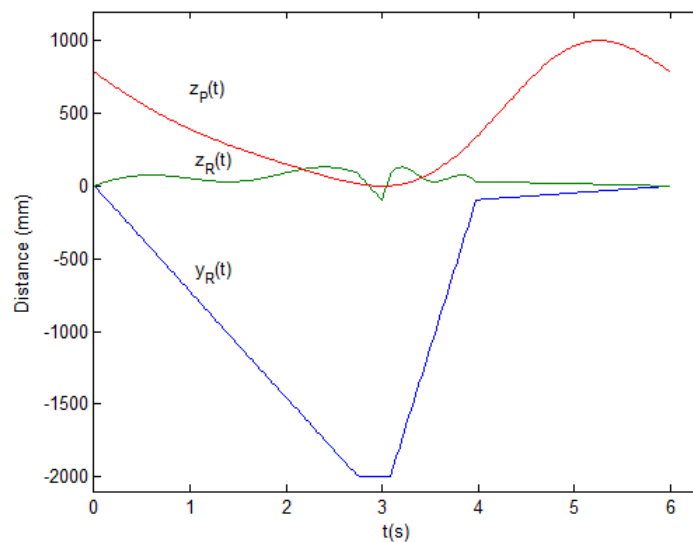


Figure 15 Motion curve of press P and robot R

Figure 16 illustrates the path for the press and the robot. The lowest point of the press is positioned at (0,0), meaning that the upper die TCP of Figure 12 follows the press path and reaches its lowest position at $z=0$. The upper die, which is attached to the ram of the press, follows the path according to its motion curve $z_p(t)$, illustrated in Figure 15.

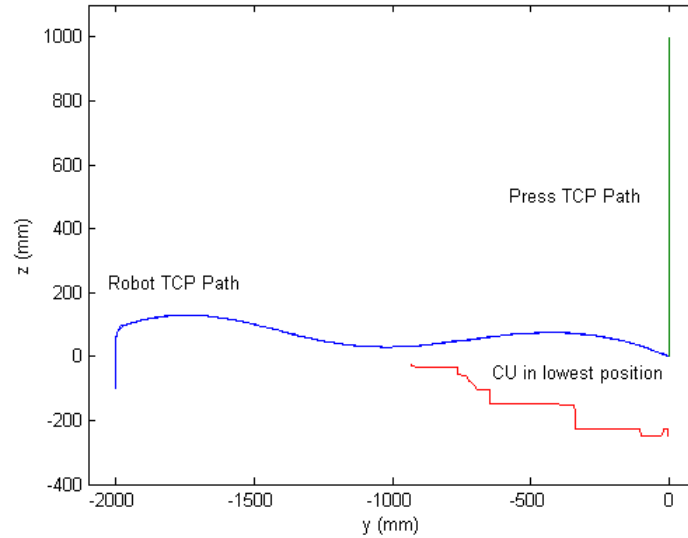


Figure 16 Path of press, robot

Robot TCP values y_R and z_R follow the robot path in Figure 16 according to robot motion curves of Figure 15. The sheet metal parts local axis (Sampling TCP) is located at (0,0) when it is delivered by the robot to the press. Hence the robot TCP path is positioned so that its delivery point is located in (0,0) as illustrated in Figure 16. The CU in Figure 16 is located at its lowest position, meaning the ram of press is at its lowest position, see also Figure 12.

The combination of the collision detection in Figure 13, the motion curves in Figure 15 and paths in Figure 16 could be visualized in a single curve, here referred to as the relative motion curve. The difference between the z values of the press and the robot $z_P - z_R$ is shown in Figure 17 for different y values of the robot. Plot (a) describes lowering of the press while the robot exits the press. The elevation of the press and entrance of the robot in the press is shown in plot (b).

The dashed curve in Figure 17 complies with elevation of the press and entrance of the robot in the press and represents the collision condition:

$$\begin{aligned} & z_P + z_{CU}(y_R) - z_R(y_R) \\ & \min(y_{CU}) \leq y_R \leq \max(y_{CU}) \end{aligned} \quad (9)$$

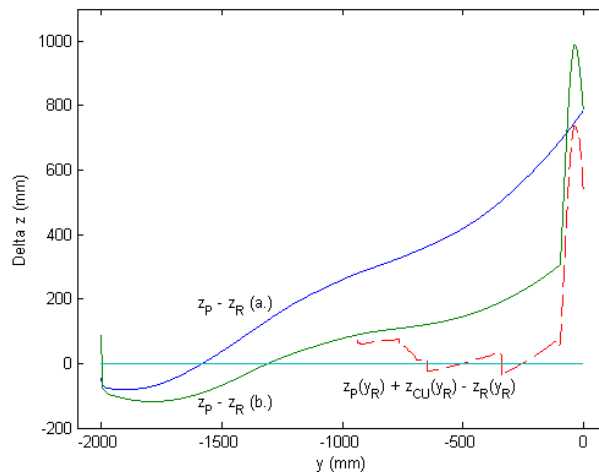


Figure 17 Relative motion curve for a press, robot and collision curve CU

As seen in Figure 17, some parts of the dashed relative motion curve are negative. According to (6) the condition $z_P + z_{CU}(y_R) - z_R(y_R) \leq 0$ represents collision. Hence, the dashed relative motion curve in Figure 17 easily illustrates the collision scenario between the press and the robot.

5.7. Minimizing collision risks

There are three possibilities to minimize the collision risk:

- (i) a change of the die/gripper geometry
- (ii) path adjustment
- (iii) a modification of the synchronization parameters and the velocity of the robot.

These changes could be done either separately or combined as described below. A change of the die/gripper geometry which in turn affects collision curves can be done in several stages of the die design. At the latest stages line tryout and ramp up, the dies are both software and hardware designed and ready for production. An unplanned change of the hardware is costly and inefficient. Thus, early efficient die geometry is desirable. Relative motion curves could be used early in the die design process to visualize collision risk areas for the die engineer.

A path adjustment and a modification of synchronization parameters, requires efficient simulation models in order to detect risks early in the die design process. An efficient solution is presented in the next section.

A modification of synchronization parameters and slowing down the velocity of the robot would allow the press to elevate and therefore lowering the risk of collision on the entrance of the robot in the press. A combined use of i-iii would allow a die designer to calculate cycle time of the line and visualize the impact different die solutions could have on the cycle time.

By introducing the collision curves in the system, the third solution opens the possibility of dynamic robot velocity control. In the studied case, the robots are not aware of the position of each other or the press/die. Their control is signal dependent. Hence, by making the robots aware of the position of each other and the press/die by using collision curves, the possibility of controlling the station or a line by dynamically adjusting velocities arises. This is an interesting alternative to the current less flexible signal based control strategy. Such a solution should result in smooth machine run with less machine wear and online momentary adjustments and tuning.

5.8. Summary

For iterative tasks such as optimizations visualizing collisions is computationally expensive. Therefore an innovative collision detection method is suggested. The first part of CIM handles detecting and storing collision points in a TCP between objects in risk of collision and compressing the result to collision curves. This part of CIM is a pre-process and is only calculated once. The second part detects collisions between the TCP on the moving object and the collision curves. This method reduces geometry drastically producing simple 2D curves which results in faster computations.

Simulation, Optimization and Strategies

A PLC-based simulation model consists of control systems, user interface, logic functions, kinematics, collision detection etc. Overall, several different programs are needed in order to run the simulation. As an example running the simulation on a medium sized server had a simulation measuring cycle time t_m of 139.7s before and 8.5s after geometry simplification using CIM [56] for the model used in [1]. Although decreasing the time by a factor of 16 is considerable, 8.5s per simulation evaluation is still high due to the need of thousands of evaluations in the optimization. It is also important to mention that 8.5s is the computation time for one station. Several stations in a press line would increase the computation time considerably. Evaluations show that the collision calculation time based on CIM is now only 2.2% of 8.5s. Hence, there is still improvement potential in building an efficient control model. By removing unnecessary signals and control code (e.g. oil pressure and line security), the model should result in shorter calculation time.

In a new proposed simulation model the PLC and press line logic are replaced by a Matlab [57] model. The solution is generic and could be adapted to any 2D transporter system.

6.1. Simulation model

The Matlab simulation model works directly with an in house developed optimization tool PLCOpt. The results from the simulation model are analyzed by the optimizer and new parameters are fed back to the simulation model iteratively until a sub-optimum is reached, see Figure 18. The solution consists of three main functions: control, robot and press and their sub functions. The robot function is used both as feeder and extractor. The motion planner function is based on real robot motion control and is verified. Logic, such as sheet metal part movement, gripper pneumatic and start signals of the components in the station, is encapsulated in the control function.

At a time step, in this case a sampling of every 5ms of virtual time, a call is sent from the control function to the robots and, presses. The motion planner in the robots and presses returns the position, time and cam values of each machine. A warm up cycle is performed before a measuring cycle, ensuring the line is filled with sheet metal parts before the measuring cycle is performed.

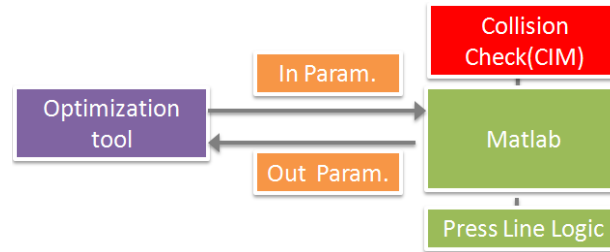


Figure 18 Simulation model and optimization tool interaction

Time for one evaluation t_e in the simulation model is defined as:

$$t_e = t_w + t_m$$

$$t_w = \text{simulation warm up cycle time}$$

$$t_m = \text{simulation measuring cycle time}$$
(10)

Equation (10) shows the need for efficient simulation methods leading to minimum t_e . For a minimum t_e two methods have been suggested, a Matlab based simulation model and the collision inspection method CIM. The use of both these methods is described in the following sections.

The motion planner block

The motion planner function for the robots used in this virtual line is the same as real robots motion control. Velocity, geometrical and path parameters are input parameters of the function. The result is the motion of the specified robot. However, for other robot types, such as Doppin material handling robots, the control is based on the handling equipments physical motor properties. Maximum jerk, acceleration and speed values are a combination of motor data and process requirements.

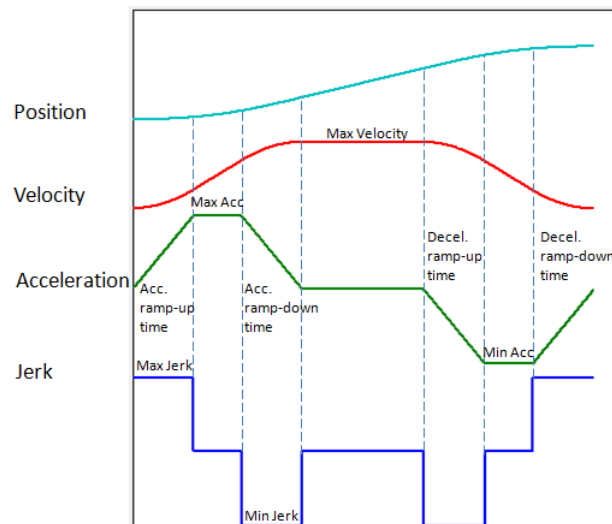


Figure 19 Illustration of position, velocity, acceleration curves base on piecewise constant jerk and their relationship

Process parameters are measurable online and are dependent on several different parameters, among others oil amount on sheet, suction force of the cups and flexibility of the

sheet metal part. Motor data is either noted from the manufacturer or possible to obtain through analysis or sampling techniques. Knowing these values opens the possibility of mathematically calculating a motion profile by knowing the path and through integral calculation of jerk, acceleration and velocity.

In Figure 19 an example scenario of change of speed from low to high and back to low again in a path segment is illustrated. The velocity change is dependent on jerk. Starting with a square-wave profile of jerk also known as the rate of change of acceleration, a trapezoidal shaped acceleration profile is calculated as illustrated in Figure 19. The benefit of a trapezoidal acceleration profile is that jerk can be controlled. Controlling the jerk in a mechanical structure is important because high jerk values can cause the mechanism to vibrate. In press handling equipment this can lead to displacement of the sheet metal part in relation to gripper deformation during handling process. This could also lead to higher collision hazard, drop of the part, unnecessary risks, unplanned repairs and maintenance. Minimizing structural vibrations is also important since oscillations can adversely affect the cycle time of the robot. Lower velocity is set for the robots as a precaution to avoid oscillations leading to lower cycle time.

The magnitude of the jerk for a square-wave acceleration profile is always infinite and for stiff, strong mechanisms, a square-wave profile may result in shorter cycle times. This scenario is not applicable for press station transfer equipment since these cannot be considered as stiff. As shown in Figure 19, the curves are normalized for easy comparison. For a general trapezoidal profile, there are four acceleration values that can be specified: the ramp up to maximum acceleration, the ramp down from maximum acceleration, the ramp up to maximum deceleration, and the ramp down to zero acceleration. Each of these four acceleration values can be individually specified and set. By knowing the path of the robot and calculating the integral of the s-shaped velocity profile, the motion of TCP of the robot along the path in relation to time can be determined. This method leads to a reliable virtual motion comparable to the real mechanism.

Collision detection

The collision detection is performed by a collision function based on CIM and can be visualized according to the relative motion curve. Totally five collision curves are used to detect clashes in one station. The collision curves are sheet metal part in/out vs. lower/upper die, which result in four different collision curves, and finally sheet metal part in vs. sheet metal part out. The collision detection is zone-based meaning that no collision calculation is performed until TCP is between minimum and maximum y values of the collision curves. This leads to even higher calculation time efficiency.

Table 4. Compression between number of triangles in the line and the resulting CIM calculation

| <i>Station Number</i> | <i>Nr. of Triangles (Millions)</i> | <i>Nr. of Points (TCP included)</i> |
|-----------------------|--|---|
| Station 1 | 0.4 | 400 |
| Station 2 | 1.3 | 553 |
| Station 3 | 1.0 | 431 |
| Station 4 | 1.8 | 462 |

The lower collision curves CL are stationary while the upper collision curves CU are following the press motion curve. The collision curves In, are checked against the TCP of the feeder robot, while the collision curves Out are checked against the TCP of the extractor

robot. The resulting sheet metal collision curve In/Out follows TCP of the extractor robot and is checked against the TCP of the feeder robot. Table 4 describes a comparison between the amounts of triangles in the original press line 3D model compared to using CIM. The table shows that the geometry of the dies, grippers and sheet metal parts in the press line are simplified considerably. A total of 4.5M triangles are reduced to 1837 points using CIM. This reduction is huge, using less computer power during collision detection calculations in iterative optimizations.

A Matlab standalone simulation, including collision detection resulted in a computation time of 0.7s, for the measuring cycle of one station of the press line t_m . This fact indicates that the 8.45s which was the result from K. Nia et al. [58] is now further improved more than 10 times. Compared to the original simulation time, which was 139.7s, the enhancement is ~200 times. A simulation run of the whole press line (four stations) showed a t_m computation time of 1.9s.

6.2. Optimization results for a press station

A series of Direct [39] optimizations with 100 iterations and Differential Evolution [59] with 2040 evaluations N_e were performed aiming to minimize δt_j . A scenario with minimum $\sum_1^3 \delta t_j$ results in high production rate, see Table 5. Hence this results in higher requirements on the sheet metal handling equipment.

Table 5. Optimization results for minimization of δ_i compared to cycle time using the methods Direct (D) and Differential Evolution (DE).

| Minimization Criterion | Cycle Time(s) | Mean(Acc) m/s^2 | Mean(J) m/s^3 | $\delta_2(s)$ | $\delta_3(s)$ | Nr. of Evaluations N_e |
|---------------------------|---------------|-------------------|-----------------|---------------|---------------|--------------------------|
| $\delta t_1 + \delta t_2$ | 3.74 | 6.3 | 44,7 | 0.34 | 0.38 | D:1412 |
| δt_3 | 3.74 | 6.3 | 44,5 | 0.08 | 0.41 | D:1602 |
| $\sum_1^3 \delta t_j$ | 3.63 | 6.8 | 56,6 | 0.02 | 0.30 | D:2238 |
| Cycle time R | 3.56 | 7.4 | 59,4 | 0.21 | 0.24 | D:1988 |
| δt_3 | 3.56 | 7.0 | 53,7 | 0.08 | 0.24 | DE:2040 |
| $\sum_1^3 \delta t_j$ | 3.47 | 5.6 | 43,0 | 0.00 | 0.15 | DE:2040 |
| Cycle time R | 3.36 | 7.4 | 51,4 | 0.07 | 0.04 | DE:2040 |

As shown in Table 5 comparing $\sum_1^3 \delta t_j$ and Cycle time, Differential Evolution is more efficient than Direct for the same amount of evaluations. One interesting fact is that δt_3 does not perform well for both methods, compared to other objective functions in Table 5 with the same N_e . A duplication of N_e was performed for both optimization methods with δt_3 as objective function without extreme improvement. This shows the complexity of the landscape due to nonlinearity and multi dimensionality of the optimization problem and the importance of the choice of objective function in the optimization.

6.3. Multi objective functions

Svensson in [1] suggests,

$$f = c_1 g_1 + c_2 g_2 + c_3 g_3 + \dots \quad (11)$$

as objective function f in an optimization based on simulations. $g_1, g_2, g_3 \dots$ are actual production performance values e.g. production rate, energy consumption, smooth motion, etc. $c_1, c_2, c_3 \dots$ are weight values to achieve a good compromise between the desired production performances g_i .

A series of Direct [39] optimizations with 10 unknown parameters was performed for $g_1 - g_3$ based on the press station model. The weight values $c_1 - c_3$ are set as constant values, g_1 equals cycle time, g_2 and g_3 are production performances for robot R_2 with and without sheet metal part according to,

$$g_{2,3} = \frac{1}{N} \sum_{k=1}^N |e_k| \quad (12)$$

where e_k represents different criterion terms described in Table 6.

The weight values are set as $\{1, 0.6, 0.2\}$, indicating an aim for more smooth run when the robot is carrying a sheet metal part than without. The same set of reference parameters were used at optimization start. Compared to Table 5 the cycle times are higher due to the choice of objective function. Table 6 also shows that the more complex the objective functions are, the higher numbers of evaluations are needed for the Direct algorithm to converge.

Table 6. Optimization results for robot R and press P for different choices of the criterion e_k , where acc =acceleration and J =jerk.

| Criterion(e_k) | Cycle Time(s) | Mean(Acc) m/s^2 | Mean(J) m/s^3 | Std(Jerk) | Max(J) m/s^3 (g_2) | $\delta r_2(s)$ | $\delta r_3(s)$ | Nr. of Evaluations |
|--------------------|---------------|-------------------|-------------------|-----------|----------------------------|-----------------|-----------------|--------------------|
| acc_k | 4.35 | 4.7 | 31.6 | 48.4 | 140 | 0.31 | 1.02 | 1996 |
| acc_k^2 | 4.67 | 3.8 | 28.4 | 48.2 | 234 | 0.10 | 1.34 | 5194 |
| J_k | 4.40 | 4.5 | 28.9 | 44.6 | 143 | 0.32 | 1.08 | 5076 |
| J_k^2 | 4.62 | 5.2 | 36.6 | 62.2 | 270 | 0.86 | 1.29 | 8264 |
| $acc_k^2 + J_k^2$ | 4.31 | 4.9 | 34.6 | 56.4 | 274 | 0.42 | 0.99 | 9750 |

Highest cycle time was produced by using $e_k = acc_k^2$ which also produced low acceleration values. As shown in the table, a low mean acceleration does not necessarily mean a smooth motion for g_2 (with sheet metal part), see $\max(|J|)$ for $e_k = acc_k^2$. $e_k = acc_k$ resulted in the lowest number of evaluations in the optimization, hence the lowest calculation time. The smoothest motion results from using $e_k = J$ since low mean acceleration, jerk and low max jerk and standard deviation are produced. The downside is the number of evaluations needed, which is more than two times higher compared to the best value when using Direct.

Another test was performed with the objective function of equation (11) with the aim of lowering max jerk for the carrying of sheet metal part of the robot where

$g_2 = \text{Max}(\sum_{k=1}^N |J_k|)$ $g_3 = 0$. The Direct algorithm succeeded to lower the max jerk to 250m/s^3 after ~ 12500 evaluations. Compared to $e_k = J$ or $e_k = acc_k$ that produced high jerk and high number of evaluations. The waiting time δt_2 was neglectable for all cases, since the robot had infinite source (fed continuously).

A comparison between Table 5 and Table 6 shows that for the same 100 Direct iterations, twice and even triple evaluations are used when using multi objective functions. It is also shown in Table 5 and Table 6 that there is a linearity between the cycle time and δt_3 . Comparing the number of evaluations N_e it is also shown that the lowest N_e were produced by using δt_j and cycle time as objective functions, hence resulting in lowest calculation times. This implies the need of known acceleration and jerk limits of the robots based on maintenance plan. Known limitations of the equipment could lead to using δt_j as objective function resulting in time efficient optimizations.

Equations (11) and (12) are industry dependent and must e.g. be balanced based on expected production rate and maintenance level. High values for accelerations and jerks might be permitted dependent of quality of equipment, maintenance plan and equipment expected life time.

6.4. Optimization strategies for a press line

The total amount of variable parameters in the press line of this case study is ~ 100 . The amount of parameters in one station is 14 without including path parameters for the 2D robots, where 12 parameters belong to the feeder robot and two belong to the press. Since a station is dependent of an extractor robot (feeder robot in the next station), the 12 parameters of the extractor must be included in the optimization. Consequently 26 parameters are needed in the optimization of one station. Four of the five press stations are used in this case study which results in a total amount of 104 parameters. Path parameters for the robots consist of a minimum of five parameters for each robot which results in a total 60 path parameters. In this case study path parameters are not included in the optimization, which implies that robot paths are fixed.

The variable parameters in this case study exceed 100, resulting in time/computationally demanding computations. There is also a need of fast optimizations for die design, line tryout and ramp-up. This is a complex task and to the author's knowledge this type of line optimization has not been performed before. Scientific papers written in this subject reduce the line optimization problem to a station optimization problem [1] and [36]. The aim is to solve the complex task of optimizing press line optimization through optimization strategies.

For this reason eight combined simulation/optimization strategies were suggested and considered to decrease the computationally expensive calculations. These are listed in three main categories: *Push*, *Pull* and *Bottleneck*.

Push is defined as the method of sequentially optimizing stations starting from station one to the last station n_s .

Pull is defined as the method of sequentially optimizing stations starting from the last station to the first.

Bottleneck is defined as the method of detecting and optimizing the worst performing station in the line iteratively.

| Strategies | Station By Station | Station By Station Additive | Station By Station Complete Line |
|-------------------|--------------------|-----------------------------|----------------------------------|
| <i>Push</i> | ✓ | ✓ | ✓ |
| <i>Pull</i> | ✓ | ✓ | ✓ |
| <i>Bottleneck</i> | ✓ | - | ✓ |

Table 7. List and combination of eight different optimization strategies

Three sub categories combinable with the main categories are considered. These are: *Station By Station (SBS)*, *SBS Additive* and *SBS Complete Line*.

SBS refers to the method of simulating and optimizing one station at a time until the whole line is simulated. The optimized parameters are kept and reused as start parameters in the next simulation e.g. station ones optimized parameters are used as start parameters in the next iteration when station one is being optimized. This continues iteratively until stop criteria is fulfilled.

SBS Additive refers to the method of first, simulating and optimizing one station. Second, the optimized station is simulated at the same time as the next station is being simulated and optimized. This additive behavior continues until all stations are simulated and optimized.

SBS complete line refers to the method of simulating the whole line while optimizing one station at a time.

Two simulation/optimization strategies were chosen and tested to decrease the calculations based on: *Push*, and *Bottleneck*.

In order to describe *Push* and *Bottleneck* mathematically a definition of the cost function is required. The cost function from station i to station j , where $i \leq j$ is defined as:

$$f_{i,j}(p_i, \dots, p_j) \quad (13)$$

The cost function for one station i is $f_{i,i}(p_i)$, while the cost function for all stations $1, \dots, n_s$ (the whole line) is $f_{1,n_s}(p_1, \dots, p_{n_s})$.

Push additive-complete line

The first strategy *Push additive-complete line* is a combination of two strategies. *Push additive* and *Push complete line*.

Push additive is performed by sequentially simulating, optimizing and adding stations starting from station one to the last station. The method starts with simulating and optimizing station 1 until a convergence criterion e.g. number of evaluations N_e . Next, station 1 and station 2 are simulated with the optimized parameters of station 1, while station 2 is optimized. This is continued until the whole line is simulated and optimized.

Push additive is defined as:

$$p_i := \arg \min_{p_i} f_{1,i}(p_1, \dots, p_i) \quad i = 1, 2, \dots, n_s \quad (14)$$

where p_i is the vector of parameters of station i and n_s is the total number of stations.

Push complete line method is performed by simulating the whole line while sequentially optimizing one station at a time and is defined as:

$$p_i := \arg \min_{p_i} f_{1,n_s}(p_1, \dots, p_{n_s}) \quad i = 1, 2, \dots, n_s \quad (15)$$

Equation (15) is then iterated N_l times.

Bottleneck additive-complete line

The second suggested strategy *Bottleneck additive-complete line* is defined as the method of detecting and optimizing the worst performing station in the line iteratively. The performance measure is case dependent. It could be the total cycle time of a robot, the cycle time of the robot without waiting time δt_j , or cycle time of a station with infinite source or sink (infinite sheet metal delivery and extraction without delays). Source in this case is defined as the press before station and sink is defined as press after.

In this study bottleneck is computed by first using *push additive* resulting in a sub optimized press line, followed by a *bottleneck complete line* optimization based on (16) and (17)

The *bottleneck* station i_b is the station with the worst performance and is defined as:

$$i_b \triangleq \{i \in \Omega_s | \forall j \in \Omega_s, j \neq i: f_{i,i}(p_i) \geq f_{j,j}(p_j)\} \quad (16)$$

where $\Omega_s = \{1, \dots, n_s\}$ is the index set including all stations.

Optimization of i_b is defined as:

$$p_{i_b} := \arg \min_{p_{i_b}} f_{1,n_s}(p_1, \dots, p_{n_s}) \quad (17)$$

The station i_b is iterated N_l times, or until $f_{1,n_s}(p_{new}) \leq (1 + \varepsilon)f_{1,n_s}(p_{old})$, where ε is a small number. Hence the iteration continues until the improvement is very small. Observe that the iteration stops immediately if the new solution gives worse performance than the old one.

Total simulation-optimization time

The amount of evaluations in one station optimization is dependent of expected optimization accuracy, time/computer power availability or objective function. The press line can be simulated and optimized iteratively N_l times until a satisfactory result is achieved. Hence the number of optimization evaluations times the number of press line iterations result in different total simulation-optimization evaluations N_{et} .

$$N_{et} = \sum_{i=1}^{n_s} N_{ea_i} + N_l \sum_{i=1}^{n_s} N_{ec_i} \quad (18)$$

n_s = Number of stations

N_{ea} = Number of evaluations in each station additive part

N_{ec} = Number of evaluations in each station complete line part

N_l = Number of line iterations

The total simulation-optimization time t_{so} is defined as:

$$t_{so} = \sum_{k=1}^{N_{et}} t_{e_k} \quad (19)$$

t_e = Time for one evaluation

A low t_{so} is desirable in order to use the results in die design or line tryout. Therefore, two optimizations were performed based on the suggested strategies: *Push additive-complete line* and *Bottleneck additive-complete line*.

6.5. Optimization results for a press line

Below the results for push additive-complete line and bottleneck additive-complete line are presented.

Push additive-complete line with Differential Evolution

A set of twelve optimizations was performed with the push strategy using differential evolution as optimization algorithm. Each optimization point in Figure 20 was performed four times.

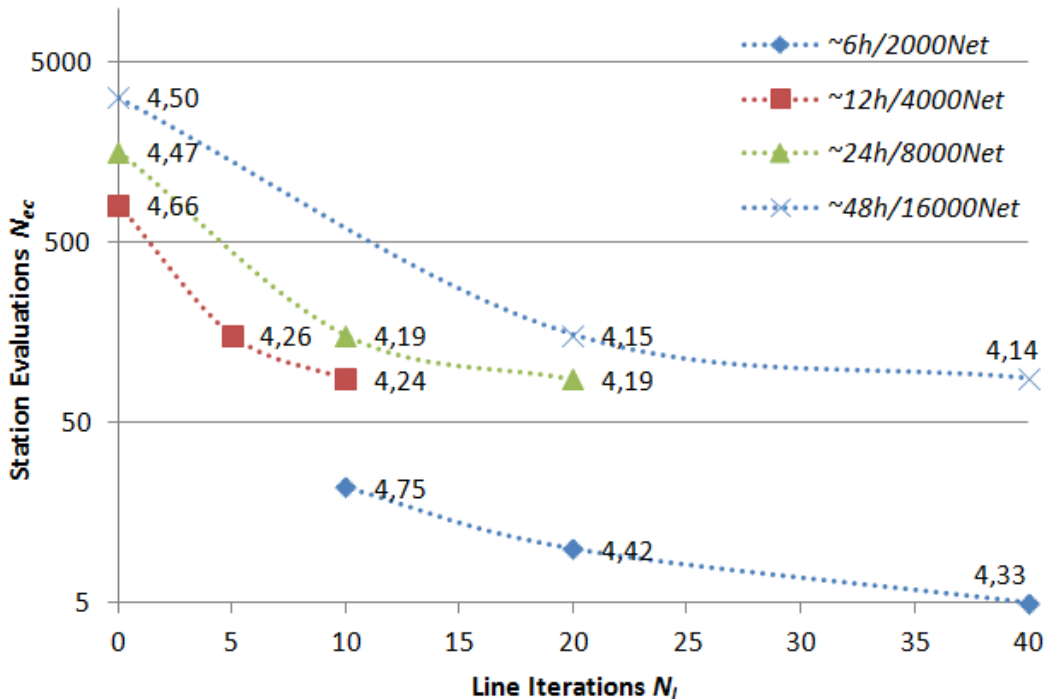


Figure 20 Push strategy with Differential Evolution. Press line cycle time in seconds

The cycle time, which is the average of the four optimizations, is labeled in Figure 20 in seconds. The line was first optimized and built with push additive method using N_{ea} values: {5,10,22,88,154,1606,3212}. Next push complete line was performed with $N_{ec} = N_{ea}$ and N_l values: {0,5,10,20,40}. Figure 20 shows twelve optimization sets divided in four series with total N_{et} of $\sim\{2000,4000,8000,16000\}$ and an approximate calculation time t_{so} of

{6h,12h,24h,48h} for each point of the series. In Figure 20, a trend is shown, where $N_{ec}=\{88,154\}$ and $N_l=\{5,10,20,40\}$ produce improving results for high N_l . This indicates the need of using a combination of N_{ec} and N_l . The $\sim 6h/2000N_{et}$ run was the fastest run but shows the worst cycle time due to low N_{ec} . However the $\sim 6h/2000N_{et}$ run produces a rough estimation of the optimized cycle time. This method is successful resulting in better performance for a combination of high $N_{ec} = N_{ea}$ and N_l .

Bottleneck additive-complete line with Differential Evolution

Another set of nine optimizations were performed with the same N_{ea}, N_{ec}, N_l settings as in the push experiment resulting in almost 12 days of calculations and a total of $\sim 100\,000$ evaluations.

Both push and bottleneck methods share the same first push additive phase. In order to detect the bottleneck in the line, the performance was measured by calculating the cycle time of the robots in each station by using a source and sink (infinite sheet metal delivery and retreat).

The objective of using a source and a sink in front and in the end of the station being optimized was to detect the stations real performance without having to consider the possible delays/restrictions caused by other stations. The bottleneck station i_b was then optimized while it was simulated together with the rest of the line.

The bottleneck method did not perform well. The hypothesis is that, during performance check the station was extracted from the line and sink and source methodology was used. Hence no restrictions (delays) were on the station, which in many cases resulted in collisions. The station was though optimized while being a part of the line. Thus, the same station with the same parameters in a line run was collision free. This scenario is normal since in a line simulation e.g. the feeder R_2 of the station has to wait for the strike of press before P_1 meaning that the press of the station P_2 has time to rise resulting in a collision free path. In a source and sink scenario the feeder R_2 does not need to wait for the press stroke of the press before, P_1 , meaning that it picks up the plate and tries to leave the plate in the press P_2 before P_2 has raised up enough for a collision free path. This means that a collision is inevitable.

Bottleneck additive-complete line methodology was unfortunately not successful. Extracting stations from the line for performance check and retrieving them in the line for optimization is not an optimum optimization method. An interesting setup for future work is bottleneck station by station strategy explained in Table 7 where both performance check and optimization is done station by station. The difficulty of this method lies in cooperation of the single optimized stations in a line.

6.6. Summary

Since the number of variable parameters in this case study exceeds ~ 100 , the optimization is computationally demanding. Two *optimization strategies* based on decomposition were defined and tested to decrease the calculations namely *Push* and *Bottleneck*.

It is shown that with the setup used in this thesis the *push* strategy fits press line optimization problems. The amount of evaluations N_{ec} in one station optimization is dependent on expected optimization accuracy, time/computer power availability and objective function. The press line can be simulated and optimized iteratively N_l times until a satisfactory result is achieved. It is shown that using a combination of N_{ec} and N_l is necessary when using the push strategy in order to get an adequate result.

The bottleneck method did not perform well. The hypothesis is that, during performance check the station was extracted from the line, and sink and source methodology was used. The station was however optimized while being a part of the line. This means that although the performance of a station measured with infinite sink and source is high, the same station in a press line interacting with other stations could result in a low performing station. This is due to the dependency between the stations. Probably there is a stronger link (dependency) between stations close to each other than stations further apart. The low performance of the *bottleneck* method is due to performance checking and optimization of not neighboring stations. The *push* method on the other hand optimizes gradually, station by station and using line iterations N_l . This methodology improves the performance of the stations step by step, hence improving the interaction of the neighboring stations, resulting in a better performing line.

Conclusion and future work

There is a need of fast simulations and optimizations for die design, line tryout and ramp-up in stamping. This thesis concentrates on investigating and suggesting methods for simulation and optimization of tandem press lines. This is a difficult task and to the author's knowledge has not been done before. A fast simulation method for press lines including innovative collision detection is developed. Two optimization strategies are presented and utilized on a virtual press line. The simulation time improvement in this case study is considerable. The time reduction and the minimization of programs needed for a simulation, opens the possibility of desktop simulation solutions for real line operative personnel.

Three questions were raised in the introductory chapter:

Q1: How are efficient geometrical simulation and virtual commissioning performed in stamping?

In order to perform efficient geometrical simulation and virtual commissioning several major areas have been addressed, namely: *simulation evaluation time (collision detection)*, *optimization algorithm*, *objective function* and *optimization strategy*.

Simulation building time is also improved due to modification and modularization capability of the simulation method. The proposed simulation model is easy to modify to fit other similar press lines. Therefore the suggested simulation method of this paper speeds up virtual construction time.

Q2: How can a feasible collision detection method be formulated to simplify complex and computationally intensive iterative press line simulations?

An innovative collision detection method CIM is suggested, converting a 3D problem to a 2D solution. The first part of CIM handles detecting and storing collision points between objects in 3D in risk of collision and compressing the result to 2D. This part of CIM is a pre-process and is only calculated once. The second part detects collisions between the TCP of the moving object and the collision curves. This method reduces geometry drastically producing simple 2D curves, which results in much faster computations.

Q3: Is it possible to establish an optimization strategy that fulfils the necessary requirements for press line simulations?

Optimization time is dependent on *optimization algorithm*, *objective function* and *optimization strategy*. Several tests were performed stating the performances of different *optimization algorithms*. Different *objective functions* were tested and the vast time consumption of the multiple objective functions was stated. Press line components waiting

time and cycle time were used as *objective functions*, and the differences in performance was stated. This implies the need of known acceleration and jerk limits of the robots based on maintenance plan. Known limitations of the equipment could lead to using robot/press waiting time as objective function resulting in time efficient optimizations.

Since the variable parameters in this case study exceeds ~100, the optimization is computationally demanding. Two *optimization strategies* based on decomposition were defined and tested to decrease the calculations namely *Push* and *Bottleneck*. It is shown that with the setup used in this thesis the *push* strategy fits press line optimization problems.

Hence, efficient *total simulation-optimization time* opens up for efficient optimization, resulting in high throughput, high quality (e.g. few collision risks), minimum operator expertise dependency and minimum wear of real stamping equipment.

Future work

An interesting area for future work is parallel computing. An optimization of a press line takes roughly 24h. This time might be acceptable in an ideal industrial environment where designers produce perfect dies before line tryout die, and no late changes occur. Since a production environment is not ideal, late changes do occur and there is an analysis desire for multiple die/gripper solutions in stamping. Hence, the total simulation time must be further improved. A solution might be to run parallel simulation-based optimizations, and later join them to a final optimization.

Summary of appended papers

In this section a brief presentation of the appended papers is given. There are three included papers appearing in logical order, reflecting the development of the performed research activities.

Paper I

Nima K. Nia, Fredrik Danielsson, and Bengt Lennartson, (2011). A faster collision detection method applied on a sheet metal press line, FAIM, Taiwan.

Geometrical collision detection is a time and resource consuming simulation task. In order to decrease time and resources, a general method has been developed. The method is useful in simulation cases where 3D CAD data is part of an iterative method, e.g. optimization. The method is based on a transformation of a general 3D problem to a 2D problem, eliminating the need of 3D CAD models. Press Line simulations during the last decade have been accepted as a quality improvement method. Today simulations of automated press lines are done for internal collision checks in dies and external collision checks against dies and material handling equipment. If these collisions are not detected in simulations, they result in delays when a new product is introduced in the line, so called line tryout or later on when the line is ramped up to decided rate. The results of these collisions are used for pre-die design, design of grippers, maintenance and production planning. In this paper a new method, based on 2D simplifications, is developed and tested successfully in a virtual model of a press line at Volvo Car Manufacturing. Die Uppers 2 917 708 triangles and Die Lowers 602 686 triangles where reduced to 58 and 90 points. The result of the method shows substantial reduction of geometry data and a considerable improvement in collision detection evaluation time over general 3D algorithms in the tested case.

Paper II

Nima K. Nia, Fredrik Danielsson, and Bengt Lennartson, (2012).

Efficient geometrical simulation and virtual commissioning performed in stamping optimization. ETFA, Poland

In order to perform efficient geometrical simulation and virtual commissioning in stamping, three fields are investigated namely: simulation building time, collision detection time and optimization time. Hence, reducing time is the main theme of this paper. To reduce simulation building time and optimization time, an efficient stamping simulation model is built and tested. Collision detection time is examined by a relative motion method based on 3D to 2D geometrical collision detection. The presented results mean that simulation and virtual commissioning can be performed at least ten times faster compared to standard approaches.

Paper III

Nima K. Nia, Fredrik Danielsson, and Bengt Lennartson, (2012).

Toward efficient simulation and optimization strategies in stamping. Submitted to an international journal.

To reduce simulation time, collision detection time is reduced by a method based on 3D to 2D geometrical collision detection. The method is based on pre-calculation of all collision points in the environment of interest, and then using a simplified collision detection model in a simulation based optimization. This is less resource consuming than collision checking the original 3D objects for all optimization evaluations. The suggested approach reduces the collision detection from a 3D to a 2D problem, where collision between simplified but moving curves is used in the repeated simulation for optimization. This collision detection approach, together with a simplified implementation of the control code, results in ~200 times reduction of the computation time, compared to the original simulation based on standard 3D collision detection.

The variable parameters in a press line exceed 100, resulting in time/computationally demanding computations. There is also a need of fast optimizations for die design, line tryout and ramp-up. Since the number of evaluations grows exponentially with the number of dimensions in an optimization problem, optimization time is reduced by a decomposition strategy aiming at dimension reduction. Two simulation/optimization strategies were chosen and tested to decrease calculations. The presented results mean that simulation and virtual commissioning can be performed not only for press stations but also for complete press lines, where the complexity increases linearly with the number of stations in the line.

Bibliography

- [1] B. Svensson, "A Simulation-Based Optimisation Method for PLC Systems," Doctor Of Philosophy, University West, 2012.
- [2] S. GmbH, *Metal Forming Handbook*, 1998.
- [3] B. Svensson, "Optimisation of Manufacturing Systems Using Time Synchronised Simulation," Licentiate Of Engineering, Signal and Systems, Chalmers University of Technology, 2010.
- [4] I. Suchy, *Handbook of Die Design*, Second ed., 2004.
- [5] P. Hoffmann, *et al.*, "Virtual commissioning of manufacturing systems a review and new approaches for simplification," presented at the Modelling and Simulation, 2010.
- [6] R. G. Budynas and J. K. Nisbett, *Shigley's Mechanical Engineering Design*, 8 Rev ed.: McGraw-Hill Education - Europe, 2006.
- [7] K. Asanovic, *et al.*, "A View of the Parallel Computing Landscape," *Commniactions of the Acm*, vol. 52, pp. 56-67, 2009.
- [8] H. F. Li, *et al.*, "A Dexterous Part-Holding Model for Handling Compliant Sheet Metal Parts," *Manufacturing Science and Engineering*, vol. 124, pp. 109-118, 2002.
- [9] O. Wulf and B. Wagner, "Fast 3D Scanning Methods for Laser Measurement Systems," presented at the Control Systems and Computer Science, 2003.
- [10] X. Lian, *et al.*, "Reconstructing Indoor Environmental 3D Model Using Laser Range Scanners and Omnidirectional Camera," presented at the World Congress on Intelligent Control and Automation, Chongqing, China, 2008.
- [11] P. Biber, *et al.*, "3D Modeling of Indoor Environments by a Mobile Robot with a Laser Scanner and Panoramic Camera," presented at the Intelligent Robots and Systems, Sendai, Japan, 2004.
- [12] M. Vincze, *et al.*, "A Laser Tracking System to Measure Position and Orientation of Robot End Effectors Under Motion," *The International Journal of Robotics Research*, vol. 4, pp. 305-314, 1994.
- [13] S. Kuk, *et al.*, "Construction and Application of a Virtual Press Shop," presented at the International Multi Conference of Engineers and Computer Scientists, Hong Kong, 2009.
- [14] P. D. Ball, *et al.*, "Modelling production ramp-up of engineering products," *Journal of Engineering Manufacture*, vol. 225, pp. 959-971, 2011.
- [15] B.-T. L. a. C.-C. Kuo, "Application of an integrated CAD/CAE/CAM system for stamping dies for automobiles," *THE INTERNATIONAL JOURNAL OF ADVANCED MANUFACTURING TECHNOLOGY*, vol. 35, pp. 1000-1013, 2008.
- [16] D. Ling, *et al.*, "Integrated Stamping Simulation Using State Of The Art Techniques To Fulfill Quality Assessment Requirements," presented at the Numerical Simulation of 3D Sheet Metal Forming Process, 2005.
- [17] K. He, *et al.*, "Trajectory Planning, Optimization and Control of a Hybrid Mechanical Press " *WSEAS Transactions on Systems*, vol. 8, pp. 614-627, 2009.

- [18] S. A. Ozalp, "A Genetic Algorithm for Scheduling of Jobs on Lines of Press Machines," presented at the Large-Scale Scientific Computing, 2005.
- [19] D. Yu, *et al.*, "The scheduling for press shop based on constraints parallel machine model," presented at the Natural Computation, 2009.
- [20] E. J. Williams, *et al.*, "Material Handling Resource Utilization Simulation Study for Stamping Plant," in *Winter Simulation Conference*, 2006, pp. 1940-1945.
- [21] H. Hoffmann and M. Kohnhauser, "Strategies to Optimize the Part Transport in Crossbar Transfer Presses," *CIRP Annals - Manufacturing Technology*, vol. 51, pp. 27-32, 2002.
- [22] D. Ceglarek, *et al.*, "Modeling and Optimization of End Effector Layout for Handling Compliant Sheet Metal Parts," *Manufacturing Science and Engineering*, vol. 123, pp. 473-480, 2001.
- [23] R. Callies and S. Fronz, "Recursive modeling and control of multi-link manipulators with vacuum grippers," presented at the 5th Symposium on Mathematical Modelling, Vienna, Austria, 2006.
- [24] R. Hossein, *et al.*, "Optimal design of a flexible transferring system in press automation using fuzzy clustering," *Assembly Automation*, vol. 25, pp. 53-58, 2005.
- [25] P. R. Moore, *et al.*, "Advanced machine service support using Internet-enabled three-dimensional-based virtual engineering," *International Journal of Production Research*, vol. 46, pp. 4215-4235, 2008.
- [26] A. H. C. Ng, *et al.*, "Virtual manufacturing for press line monitoring and diagnostics," *International Journal of Machine Tools & Manufacture*, vol. 48, pp. 565-575, 2008.
- [27] Y. Chunyan, *et al.*, "Two-Level 2D Projection Maps Based Horizontal Collision Detection Scheme for Avatar in Collaborative Virtual Environment," *Computational Science And Its Applications – ICCSA*, vol. 3480/2005, pp. 875-885, 2005.
- [28] J.-L. Chen, *et al.*, "A recursive algorithm of obstacles clustering for reducing complexity of collision detection in 2D environment," presented at the Robotics and Automation. Proceedings ICRA. IEEE International Conference, 2001.
- [29] S. Huh, *et al.*, "Collision resolutions in cloth simulation," presented at the The Fourteenth Conference on Computer Animation, Seoul, South Korea, 2001.
- [30] S.-E. Yoon, *et al.*, "Fast Collision Detection between Massive Models using Dynamic Simplification," presented at the ACM International Conference Proceeding Series 2004.
- [31] L. Martinez-Gomez and T. Fraichard, "An efficient and generic 2D Inevitable Collision State-checker," presented at the Intelligent Robots and Systems, IROS. IEEE/RSJ International Conference 2008.
- [32] M. C. Fu, *et al.*, "Simulation optimization: a review, new developments and applications," in *Winter Simulation Conference*, 2005.
- [33] J. G. Li, *et al.*, "Off-line optimization on NC machining based on virtual machining," *Adv Manuf Technol*, vol. 36, pp. 908-917, 2008.
- [34] A. Persson, *et al.*, "A Metamodel-Assisted Steady-State Evolution Strategy for Simulation-Based Optimization," *Electrical Engineering*, vol. 6, pp. 1-13, 2008.
- [35] X. Liao and G. G. Wang, "Evolutionary path planning for robot assisted part handling in sheet metal bending," *Robotics and Computer-Integrated Manufacturing*, vol. 19, pp. 425-430, 2003.
- [36] J. A. García-Sedano, *et al.*, "Stamping Line Optimization Using Genetic Algorithms and Virtual 3D Line Simulation," in *Hybrid Artificial Intelligence Systems*, 2010, pp. 319-326.
- [37] H. Carlsson, *et al.*, "General Time Synchronisation Method for PLC Programs " presented at the The International Federation of Automatic Control, Korea, 2008.

- [38] J. A. Nelder and R. Mead, "A Simplex Method for Function Minimization," *The Computer Journal*, vol. 7, pp. 308-313, 1965.
- [39] M. H. Wright, "Direct search methods: once scorned, now respectable," presented at the Dundee Biennial Conference on Numerical Analysis, 1995.
- [40] F. Danielsson, "A distributed system architecture for optimizing control logic in complex manufacturing system," presented at the ISCA 12th International Conference, Atlanta USA, 1999.
- [41] S. Shan and G. G. Wang, "Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions," *Structural and Multidisciplinary Optimization*, vol. 41, pp. 219-241, 2010.
- [42] APT. (2011, 16th august). *Heat Exchanger Lines*. Available: <http://www.aptgroup.com/products/ProductionLines/HeatExchangerLines1/>
- [43] B. Svensson, *et al.*, "A Virtual Real-Time Model for Control Software Development - applied on a Sheet-Metal Press Line," presented at the 3rd International Industrial Simulation Conference, Berlin, Germany, 2005.
- [44] S. Bickerstaffe. Pressing simulation system reduces toll trial and error. *Automotive Engineer*.
- [45] A. Andersen, *et al.*, "Three-dimensional techniques for capturing and building virtual models of complex objects for use in scientific and industrial applications, data archiving, and the entertainment industry," presented at the Stereoscopic Displays and Virtual Reality Systems X, Santa Clara, CA, USA, 2003.
- [46] F. EUROPE. Press Line Simulation based on LS 3D scanning. *Faro Short Story*.
- [47] K. Scheibe, *et al.*, "Multi-scale 3D-Modeling," *Advances in Image and Video Technology*, vol. 4319, pp. 96-107, 2006.
- [48] K. Harding, "Application issues when using optical 3D systems in place of CMMs," *Machine Vision and Three-Dimensional Imaging Systems for Inspection and Metrology II*, vol. 4567, pp. 1-10, 2002.
- [49] J. P. Prenninger, *et al.*, "Real-time contactless measurement of robotpose in sixdegrees of freedom," *Measurement*, vol. 14, pp. 255-264.
- [50] H. Lundh, *et al.*, *Formningshandboken* vol. II. Borlange, 1997.
- [51] K. Abdel-Malek and J. Yang, "Swept volumes: fundation, perspectives, and applications," *International Journal of Shape Modeling*, vol. 12, pp. 87-127, 2006.
- [52] D. Ceglarek and S. J. "Dimensional Variation Reduction for Automotive Body Assembly," *Manufacturing Review*, vol. 8, pp. 139-154, 1995.
- [53] L. Kelmar and P. K. Khosla, "Automatic generation of kinematics for a reconfigurable modular manipulator system," in *Robotics and Automation*, 1988, pp. 663-668.
- [54] J. Zhao and N. I. Badler, "Inverse Kinematics Positioning Using Nonlinear Programming for Highly Articulated Figures," *ACM Transactions on Graphics*, vol. 13, pp. 313-336, 1994.
- [55] K. Osakada, *et al.*, "Mechanical servo press technology for metal forming," *CIRP Annals - Manufacturing Technology*, vol. 60, pp. 651-672, 2010.
- [56] N. K. Nia, *et al.*, "A faster collision detection method applied on a sheet metal press line," presented at the FAIM, Taiwan, 2011.
- [57] Matlab. Available: www.mathworks.com
- [58] N. K. Nia, *et al.*, "A faster collision detection method applied on a sheet metal press line," presented at the FAIM, Taiwan, 2010.
- [59] R. Storn and K. Price, "Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, pp. 341-359, 1997.

- [60] J. A. Nelder and R. Mead, "A Simplex Method for Function Minimization," *The Computer Journal*, vol. 7, pp. 308-313, 1965.
- [61] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *IEEE International Neural Networks*, 1995, pp. 1942-1948.