



CHALMERS

Chalmers Publication Library

Efficient Geometrical Simulation and Virtual Commissioning Performed in Stamping

This document has been downloaded from Chalmers Publication Library (CPL). It is the author's version of a work that was accepted for publication in:

17th IEEE International Conference on Emerging Technologies & Factory Automation (ETFA)

Citation for the published paper:

Nia, N. ; Danielsson, F. ; Lennartson, B. (2012) "Efficient Geometrical Simulation and Virtual Commissioning Performed in Stamping". 17th IEEE International Conference on Emerging Technologies & Factory Automation (ETFA)

<http://dx.doi.org/10.1109/ETFA.2012.6489592>

Downloaded from: <http://publications.lib.chalmers.se/publication/171166>

Notice: Changes introduced as a result of publishing processes such as copy-editing and formatting may not be reflected in this document. For a definitive version of this work, please refer to the published source. Please note that access to the published version might require a subscription.

Chalmers Publication Library (CPL) offers the possibility of retrieving research publications produced at Chalmers University of Technology. It covers all types of publications: articles, dissertations, licentiate theses, masters theses, conference papers, reports etc. Since 2006 it is the official tool for Chalmers official publication statistics. To ensure that Chalmers research results are disseminated as widely as possible, an Open Access Policy has been adopted. The CPL service is administrated and maintained by Chalmers Library.

(article starts on next page)

Efficient geometrical simulation and virtual commissioning performed in stamping

Nima Khansari Nia
Volvo Cars IT
nnia@volvocars.com

Fredrik Danielsson
Dept. of Engineering Science
University West

Bengt Lennartson
Dept. of Signals and Systems
Chalmers University of Technology

Abstract

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.

1. Introduction

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 stamp metal sheets to different components through successive operations. Since the stamping process is complex, geometrical simulations are used more and more on daily basis as a quality tool, aiming at cost minimization.

The purpose of using simulations in stamping is: high throughput, high quality (few collision risks), minimum operator expertise dependency and minimum wear of real stamping equipment. 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. Thus, different simulation software is used, often with 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.

This paper concentrates on the fields of geometrical simulation and virtual commissioning [1], covering Die Design to Ramp up. Virtual commissioning in this paper is referred to pre-programming also referred to offline programming of press lines. The aim of this paper is to

propose solutions to the following question: How is efficient geometrical simulation and virtual commissioning performed in stamping? In order to address this problem, three fields are investigated namely: A. Simulation building time, B. Collision detection time and C. Optimization time. These fields are described in detail below.

Building a detailed model of a Press Line 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 [2] that the PLC can consist of more than 100 000 lines of PLC code, 2000 input/outputs where 100 could be adjustable. Obtaining correct geometry and correct motion of the line (Press, Die, Robot, Gripper, and Line equipment) have also complexities. Technologies such as Laser scanning [3-5] and Laser tracking [6] help in virtual construction of virtual press shops [7]. Due to complexity, acquiring correct virtual control system and geometry affect the time frame of designing and verifying a virtual model. A period of several months up to a year, dependent of the available resources, is not unusual.

Collision detection time is the second field of study in this paper. 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. Collision detection aims furthermore in avoiding collisions during ramp up until the decided cycle time is achieved. Collision avoidance is a must in a press station, where collisions between components online could lead to destruction of equipment, grippers, dies and human injuries. A collision also leads to production delays, since unplanned operations such as gripper, die or robot repair must be performed.

Total Optimization time which is the third proposed critical factor, refers to the total computation time for an optimization, simulating and optimizing input parameters e.g. path creation/modification or synchronization of robots and press. The computation time is dependent on three major parts namely chosen optimization algorithm, objective function and simulation time. Reducing time in A-C is the main theme of this paper. As a solution to time reduction, a collision detection method bases on 3D to 2D reduction and an

enhanced simulation method are presented and utilized on a press station. Both these methods aim at addressing the question discussed above and decreasing building time, collision detection time and optimization time.

2. Background

An evolutionary path-planning approach is used by Liao et al. in [8] 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 on 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 [7]. A modelling 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 [1].

A virtual commissioning scenario for stamping is investigated by García-Sedano et al. in [9]. 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. The line is simplified though to a press station consisting of two robots and a press.

A press line simulation model is designed by Svensson et al. in [2] including PLC and geometry with collision detection possibility. The system runs on a time synchronized environment [10]. The line is restricted to a station, only due to its complexity with focus on different optimization methods. The optimization tool consists of several optimization methods among others Nelder-Mead [11] and Direct [12]. 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) [13]. This architecture is designed to handle time and synchronization problems as well as distributed simulations. The collision check evaluates collisions according to Collision Inspection Method (CIM), proposed by Nia et al. in [14]. CIM is a method for simplification of complex 3D geometry to 2D.

The simulation model uses real PLCs including all electrical signals, signal paths and hardware 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 [2]. Although this model would replicate the real model well, the model execution is time and computer resource consuming. As an example running the simulation on a medium sized server had a response evaluation time of 139.7s before and 8.5s after the geometry simplification suggested in [15]. 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 showed that the collision calculation time based on CIM was 2.2% of 8.5s. Hence, there is still improvement potential in building an efficient model. By removing unnecessary signals and control code (e.g. oil pressure and line security), the model should result in shorter calculation time.

According to design methodology the goal of a project is to use the outcome in an industrial environment, Budynas et al. [16]. The outcome of this paper 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 with enough power to fulfil 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, discussed earlier.

3. Tandem Press Station

The tandem press station studied in this paper consists of a mechanical press with fixed displacement-time curves, associated die and robots.

The material handling is performed by two 2D robots and intermediate stations. The robots are responsible for transport of the sheets metal part by grippers in the Y direction, see Figure 1 .

Robots follow a programmed path in the YZ plane. The robot feeding the press is called feeder and the robot extracting sheets from the press is called extractor. Press position is measured in Z and feeder and extractor positions are measured in YZ.

Intermediate stages work as sheet rest. 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.

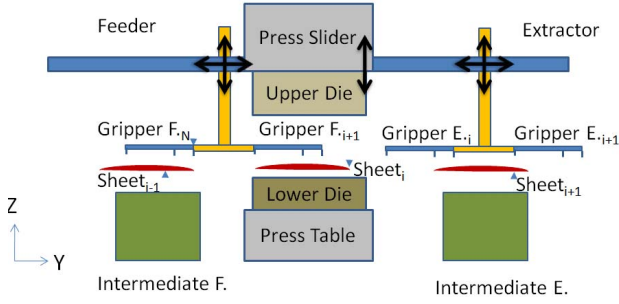


Figure 1 Press Station description

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 equipments in the line. Modern equipments such as robots and hydraulic presses lack 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 components from start to end and back to starting position measured from 0 to 360 degrees.

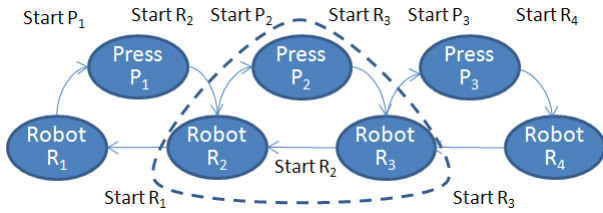


Figure 2 Combination of two stations and signal description

Several stations combined form a linear formation called 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 2 describes the combination of three stations and their signal transmission. Each machine starts the next by a start signal at a programmed cam value. Robot R_1 starts press P_1 . Press P_1 starts robot R_2 which starts both press P_2 and robot R_1 . Robot R_2 , which is an extractor, is also working as a feeder for the press P_2 in the next station.

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 1 .

Table 1 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 Path</i>	Variable	Fixed	Variable
<i>Signal</i>	N_p Position variables	Fixed	N_p Position variables
	Start feeder	Start press	Start extractor

Motion stands for parameters: velocity, acceleration, deceleration and jerk, all affecting the motion 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 of the velocity set points. The feeder and extractor have several positions along their path, namely: Fetch, Leave and Position $1 \cdot N_p$ 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 after. 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 instead [2].

4. Motion

In this section the relationships between the motions and signals in Table 1 , are explained for the components in the station.

4.1. Press Motion

Press motion is dependent on a press motion curve and a press start time signal $t_{P\ start}$ (the time instance when the press starts a press strike). $t_{P\ start}$ is dependent on the feeder see Figure 2 . Press motion is measured in height (mm) / degrees. In the studied station the press is a mechanical press with a fixed press motion curve. The motion curves of mechanical presses (crank, linkage knuckle-joint) are fixed compared to force controlled presses [17]. A cycle in a press starts at the top position normally 0 degrees down to the lowest position at 180 degrees and up to 360 degrees. The Combination of press motion and press start signal $t_{P\ start}$, results in a function $f_{Press\ Motion}(t, t_{P\ start})$, which generates the height of the slider.

4.2. Feeder/Extractor Motion

The function $f_{Feeder\ Motion}(t)$ returns the position of the tool centre point (TCP) of a robot along the path in relation to time, and is exemplified in Figure 3 .

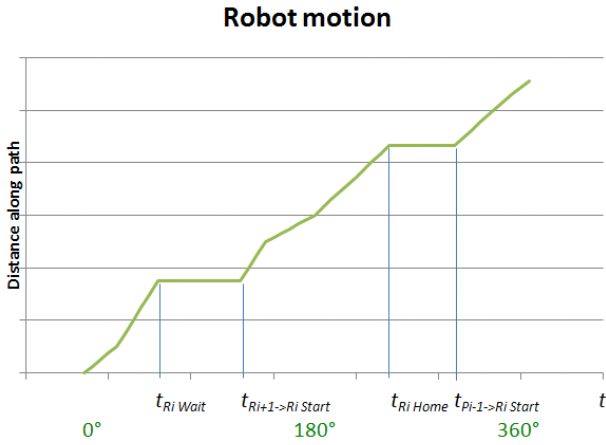


Figure 3 Illustration of robot motion curve with waiting time

The starting signals in the station affect the motion of the feeder Robot. To emphasize this, the time of the function is divided as follows:

$$\begin{aligned}
 t_{P_{i-1} \rightarrow R_i \text{ Start}} &= \text{Start time for robot } R_i \text{ to pick sheet metal part from press } P_{i-1} \text{ (Signal from press } P_{i-1}) \\
 t_{R_i \text{ Wait}} &= \text{Signal wait for robot } R_i \text{ before entering Press } P_i \\
 t_{R_{i+1} \rightarrow R_i \text{ Start}} &= \text{Start time for robot } R_i \text{ to enter press } P_i \text{ (Signal from robot } R_{i+1} \text{ to robot } R_i) \\
 t_{R_i \text{ Home}} &= \text{Signal home before entering press } P_{i-1}
 \end{aligned} \tag{1}$$

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

5. Motion relation

The relationships between the motion curves are dependent on the tuning of the station. A non-optimized station results in loss of production and could also mean ware of equipment due to unnecessary accelerations, deceleration and jerks. The goal in an optimization of the 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 3).

$$\delta t_1 = t_{R_{i+1} \rightarrow R_i \text{ Start}} - t_{R_i \text{ Wait}} \tag{2}$$

$$\delta t_2 = t_{P_{i-1} \rightarrow R_i \text{ Start}} - t_{R_i \text{ Home}} \tag{3}$$

$$\delta t_3 = t_{P_i \text{ Start Cycle } cn} - t_{P_i \text{ Start Cycle } cn-1} + T \tag{4}$$

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

Formula (2) describes the time difference between start signal of robot R_{i+1} (extractor) to R_i (feeder) and feeders wait signal. Formula (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. Formula (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 ($\delta t_3 = 0$) 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.

Svensson et al. in [2] suggests:

$$f = c_1 g_1 + c_2 g_2 + c_3 g_3 + \dots \tag{5}$$

as an 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 [12] optimization with 100 iterations, each including 10 unknown parameters was performed for $g_1 - g_3$ based on the model described in section 7. $c_1 - c_3$ are set as constant values, g_1 equals production rate, g_2 and g_3 are production performances for robot R_2 with and without sheet metal part according to:

$$g_{2,3} = \frac{1}{N} \sum_{k=1}^N |e_k| \tag{6}$$

where e_k represents different criterion terms described in Table 2 .

Table 2 Simulation results for robot R and press P with criteria e_k , acc=Acceleration, J=Jerk

Criterion(e_k)	Cycle Time(s)	Mean(Acc) m/s^2	Mean(J) m/s^3	δt_2 (s)	δt_3 (s)	Nr. of Evaluations
acc _k	4.35	4.7	31.6	0.31	1.02	1996
acc _k ²	4.67	3.8	28.4	0.1	1.34	5194
J _k	4.40	4.5	28.9	0.32	1.08	5076
J _k ²	4.62	5.2	36.6	0.86	1.29	8264
acc _k ² + J _k ²	4.31	4.9	34.6	0.42	0.99	9750

The same set of reference parameters were used at optimization start. Highest cycle time and δt_3 was produced by using $e_k = acc_k^2$ which also produced low acceleration values. Table 2 shows also that the more complex the objective functions are, higher evaluations

are needed for the Direct algorithm to converge. δt_2 was neglectable for all cases due to the robot was fed continuously.

Equation (5) and (6) 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. Relative Motion

Since several objects are in motion in a press station, relative motion curves can be used to visualize objects activity and collision risks. This knowledge is usable in die design process, where known limitations of a press are essential to avoid collisions. In a tandem press line a known motion relationship is required in order to benefit from relative motion function.

6.1. Collision Curves

Before relative motion curves are discussed in more detail, an understanding of collision checks in a press station is required. An effective method for collision check in a press station is the collision inspection method (CIM), proposed by K. Nia et al. in [15]. 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.

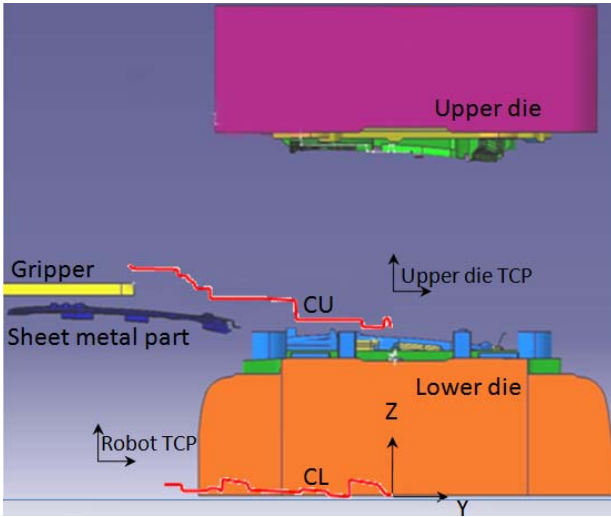


Figure 4 Result of CIM in a collision scenario

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 an iterative simulation.

Figure 4 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.

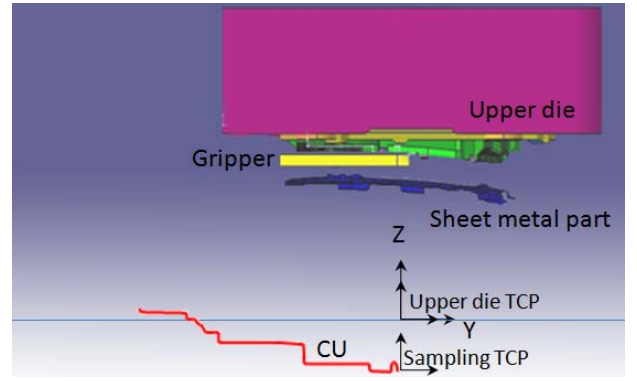


Figure 5 Creation of the curve CU

Figure 5 illustrates the creation of the curve CU. The chosen sampling 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 5 the sheet metal part, the gripper and its local axis system (the sampling 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.

In a simulation case, the robot TCP of Figure 4 is tested for collision against CL and CU according to (7) where CL is fixed while CU follows the press motion curve.

The equations that determine when collision occurs according to CIM are:

$$\begin{aligned}
 & z_p + z_{CU}(y_R) \leq z_R(y_R) \\
 & \min(y_{CU}) \leq y_R \leq \max(y_{CU}) \\
 & \quad \quad \quad \text{OR} \\
 & z_{CL}(y_R) \geq z_R(y_R) \\
 & \min(y_{CL}) \leq y_R \leq \max(y_{CL}) \\
 & z_p = \text{Press } z \text{ values } z_p(t) \\
 & y_{CL}, z_{CL} = \text{Collision Lower } y, z \text{ values} \\
 & y_{CU}, z_{CU} = \text{Collision Upper } y, z \text{ values}
 \end{aligned} \tag{7}$$

$y_R, z_R =$ Robot y, z TCP values $y_R(t), z_R(t)$

The first part of the equation describes that z_{CU} follows the press motion z_p while the robot TCP must be below CU to avoid collisions. The second part explains that the robot TCP must be above CL to avoid collisions. Collision curves for different components in the line can be used at the same time in a simulation.

6.2. Relative motion curves

To graphically illustrate a possible collision scenario, the motion curve of a real industrial press station including a feeder robot is shown below.

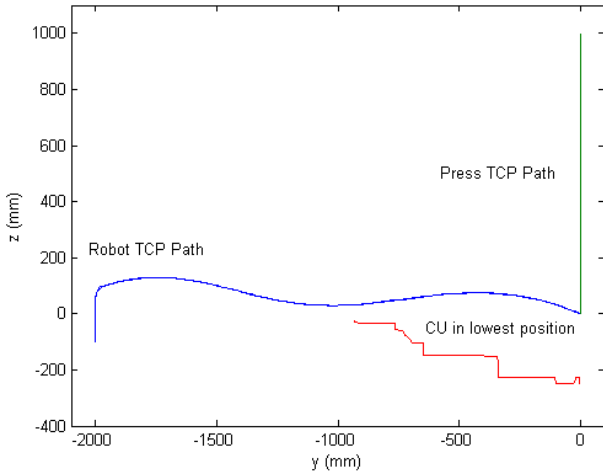


Figure 6 Motion path of press, robot and collision curve CU

Figure 6 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 4 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 7.

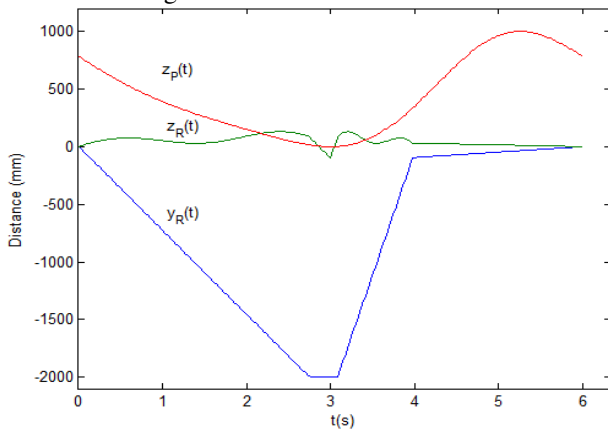


Figure 7 Motion curves of press P and robot R

Robot TCP values y_R and z_R follow the robot path in Figure 6 according to robot motion curves of Figure 7. 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 7. The CU in Figure 7 is located at its lowest position, meaning the ram of press is at its lowest position, see also Figure 5.

The combination of the collision detection in (7) paths in Figure 6 and the motion curves in Figure 7 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 8 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 8 complies with elevation of the press and entrance of the robot in the press and represents the collision condition:

$$z_P + z_{CU}(y_R) - z_R(y_R) \leq 0 \quad (8)$$

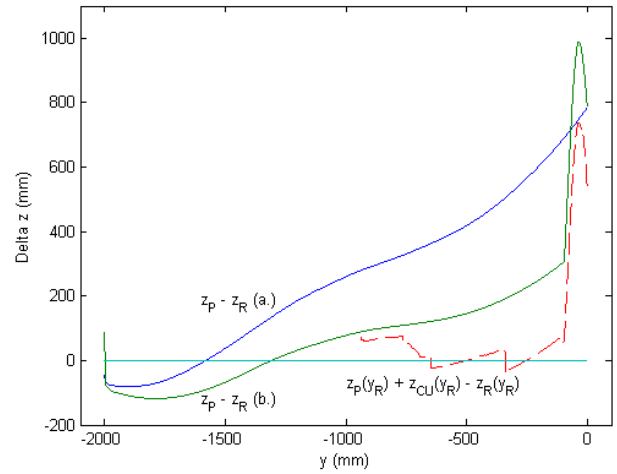


Figure 8 Relative motion curve for a press, robot and collision curve CU

As seen in Figure 8, some parts of the dashed relative motion curve are negative. According to (7) the condition $z_P + z_{CU}(y_R) - z_R(y_R) \leq 0$ represents collisions. Hence, the dashed relative motion curve in Figure 8 easily illustrates the collision scenario between the press and the robot.

There are three possibilities to minimize the collision risk: (i) a change of the die/gripper geometry, (ii) path adjustment and (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 station, the robots are not aware of the position of each other or the press/die. Their control is signal dependent as described in Section 3. 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.

7. Simulation Method

The original simulation model, which is described in detail in Section 2-3, consists of emulated control systems, emulated user interface, logic functions etc. Overall, several different programs are needed in order to run the simulation. In the new proposed model the PLC and Press Line Logic and emulators are replaced by a compiled Matlab [18] program. The solution is generic and could be adapted to any 2D transporter system.

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 set time step, in this case a sampling of every 5ms of virtual time, a call is sent from the control function to the feeder, the press and the extractor. The motion planner in the feeder, the extractor and the press 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. The collision detection is performed by a collision function based on CIM section 6.1 and is visualized according to the

relative motion curve in Section 6.2. Totally five collision curves are used to detect clashes. 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 curve in Figure 4 represent sheet metal part in. Figure 8 represents sheet metal part in-upper. The collision detection is zone-based meaning; 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.

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.

A Matlab stand alone simulation showed a simulation time of 0.7s including collision detection. This fact indicates that the 8.45s which was the result from K. Nia et al. [15] is now further improved more than 10 times. Compared to the original simulation time, which was 139.7s, the enhancement is 200 times. The Matlab simulation model works directly with an optimization tool. The results from the simulation model are analyzed by the optimizer and new parameters are fed back to the simulation model iteratively until optimum is reached. A combined DIRECT/Nelder-Mead optimization comparable with Svensson et al. [19] was performed with evaluation numbers of 800, 1200 and 1400. The average calculation time was improved 10 times, as shown in Table 3 .

Table 3 Calculation time compression between reference and new method

<i>Evaluation Number</i>	<i>Reference (Calculation time)</i>	<i>New Simulation Method(Calc. time)</i>
800	8h	0.8h
1200	11h	1.1h
1400	13h	1.3h

8. Conclusions

Simulation building time, collision detection time and optimization time are investigated in this paper for the purpose of press station simulation. The suggested simulation method of this paper speeds up virtual construction time, collision detection time and optimization time, leading to high throughput, high quality (few collision risks), minimum operator expertise dependency and minimum wear of real stamping equipment. Hence, these methods reduce the total engineering project time by predicting collisions e.g.

avoiding unplanned milling of dies and predicting cycle time before arrival of the dies to the press stations.

The methods forces press-shops to a standardized methods e.g. gripper building and forces die designers to consider cycle time impact due to die design. It is the authors belief that the proposed simulation model is easy to modify to fit other similar press stations. Therefore due to modification and the modularization capability of the simulation method, building a complete press line is one of the future projects of the authors.

The relative motion method with its ability to visualize motion and collisions of objects is a useful technique to envision complex motions in a single curve. With it, collisions are predicted in master axis systems, due to known relative motion and visualized for systems without a master axis.

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.

The possibility of using collision curves as a way of controlling a tandem stations is raised. Robots could be made aware of their positions and the position of press/die, by using collision curves. Velocity is then directly controlled instead of a signal based system, leading to harmonic motion with less machine wear and online tuning.

9. Acknowledgement

The authors wish to thank Anders Nilsson, Emile Glorieux and Bo Svensson at university West for their contributions to the project.

References

- [1] P. Hoffmann, *et al.*, "Virtual commissioning of manufacturing systems a review and new approaches for simplification", *Modelling and Simulation*, 2010.
- [2] B. Svensson, "A Simulation-Based Optimisation Method for PLC Systems", Doctor Of Philosophy, University West, 2012.
- [3] O. Wulf and B. Wagner, "Fast 3D Scanning Methods for Laser Measurement Systems", *Control Systems and Computer Science*, 2003.
- [4] X. Lian, *et al.*, "Reconstructing Indoor Environmental 3D Model Using Laser Range Scanners and Omnidirectional Camera", *World Congress on Intelligent Control and Automation*, Chongqing, China, 2008.
- [5] P. Biber, *et al.*, "3D Modeling of Indoor Environments by a Mobile Robot with a Laser Scanner and Panoramic Camera", *Intelligent Robots and Systems*, Sendai, Japan, 2004.
- [6] M. Vincze, J. P. Prenninger, and H. Gander, "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.
- [7] S. Kuk, *et al.*, "Construction and Application of a Virtual Press Shop", *International Multi Conference of Engineers and Computer Scientists*, Hong Kong, 2009.
- [8] 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.
- [9] 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.
- [10] H. Carlsson, F. Danielsson, and B. Lennartson, "General Time Synchronisation Method for PLC Programs ", *The International Federation of Automatic Control*, Korea, 2008.
- [11] J. A. Nelder and R. Mead, "A Simplex Method for Function Minimization", *The Computer Journal*, vol. 7, pp. 308-313, 1965.
- [12] M. H. Wright, "Direct search methods: once scorned, now respectable", *Dundee Biennial Conference on Numerical Analysis*, 1995.
- [13] F. Danielsson, "A distributed system architecture for optimizing control logic in complex manufacturing system", *ISCA 12th International Conference*, Atlanta USA, 1999.
- [14] N. K. Nia, F. Danielsson, and B. Lennartsson, "A faster collision detection method applied on a sheet metal press line", *FAIM*, Taiwan, 2010.
- [15] N. K. Nia, F. Danielsson, and B. Lennartsson, "A faster collision detection method applied on a sheet metal press line", *FAIM*, Taiwan, 2011.
- [16] R. G. Budynas and J. K. Nisbett, *Shigley's Mechanical Engineering Design*, 8 Rev ed.: Mcgraw-Hill Education - Europe, 2006.
- [17] K. Osakada, *et al.*, "Mechanical servo press technology for metal forming", *CIRP Annals - Manufacturing Technology*, vol. 60, pp. 651-672, 2010.
- [18] Matlab. Available: www.mathworks.com
- [19] B. Svensson, *et al.*, "Sheet-Metal Press Line Parameter Tuning using a Combined DIRECT and Nelder-Mead Algorithm", in *16th IEEE International Conference on Emergency Technologies and Factory Automation*, Toulouse, France, 2011, pp. 207-214.