Incremental Sheet Forming - Process Simulation with LS-DYNA

J. Zettler[1], H. Rezai[1], B. Taleb-Araghi[2], M. Bambach[2], G. Hirt[2]
[1] EADS Deutschland GmbH, München
[2] Institut für Bildsame Formgebung der RWTH Aachen

Abstract. Incremental CNC sheet forming (ISF) is a relatively new sheet metal forming process for small batch production and prototyping. In ISF, a blank is shaped by the CNC movements of a simple tool in combination with a simplied die. The standard forming strategies in ISF normally use some kind of die in order to increase the accuracy of the final part. Therefore in this paper the main focus is on a process variation that uses an incremental die to replace the die. By using this new process the tooling costs for the die will become minimal and the geometric accuracy is higher as without die. However, an empirical approach to tool path optimization for an increased accuracy will yield an enourmous experimental effort. This motivates the use of process modeling techniques.
This paper deals with finite element modeling of the „ISF process with an incremental die system“ in LS-Dyna. In particular, the simulation setup and the tool path generation is explained in detail. We will also give a short summary of the state of the art in ISF and about the material modeling in LS-Dyna. The paper adresses mainly the industrial user who would like to perform ISF simulation without bothering about the difficulties of the simulation setup.

1 Introduction

With regards to the mass production of sheet metal components, a number of well-established forming methods like e. g. deep drawing and pressing enable a secure, cost-effective and high-capacity manufacturing of complex parts. Economically important fields of application are e.g. body and structural parts of passenger vehicles or cover parts of white goods which are manufactured in very high volumes. Due to the complexity of the employed dies and the elevated forming forces, these processes require enormous investments for tooling and equipment as well as prolongated lead times until utilisability. In automotive industry, typical values for the time to manufacture and set-up a Deep-Drawing-die and the inherent costs are minimum three months and at least 1.5 Mio. Euros respectively [3]. In the past, these high investments could be compensated by huge volumes of one and the same part during long life cycles.
On the other hand, investigations and market analyses of production industries over the past years show an accelerating development towards an increasing number of variants, lower volumes and shorter innovation cycles [6, 9]. As a result, sample parts and prototypes have to be available at very short term especially in early design stages to avoid delays in the development process of a new product [8]. Furthermore, the need for flexible forming processes that can easily be adapted to new shapes or geometric variants without exceeding tooling effort arises in order to cope with the increasing customisation trend and the inherent lower quantity of parts to be produced [5].

2 Process Description

2.1 Traditional ISF

Asymmetric Incremental Sheet Forming (AISF) is a relatively new method for sheet metal forming. It enables a flexible rapid prototyping and economical low batch production of complex parts. This is accomplished through „kinematical“ forming: the final part is obtained by the continuous action of a CNC controlled forming tool. Due to the local action of the forming tool, the forces can be kept low. This enables additional tooling to be made of cheap materials (e. g. wood or resin), reduced in geometric complexity or even omitted, which reduces the set-up costs. It is differentiated among „single-point“ (SPIF) and „two-point“ (TPIF) forming . The process variants have the same forming concept, according to which a metal sheet is clamped with a blank holder and then formed by the continuous movements of a CNC controlled forming tool. The difference, however, is the tool concept, itself. In SPIF, there is no support tool used. TPIF uses a partial or a full die as a support tool. This increases the geometrical accuracy but reduces the flexibility of the method (see Figure 1).

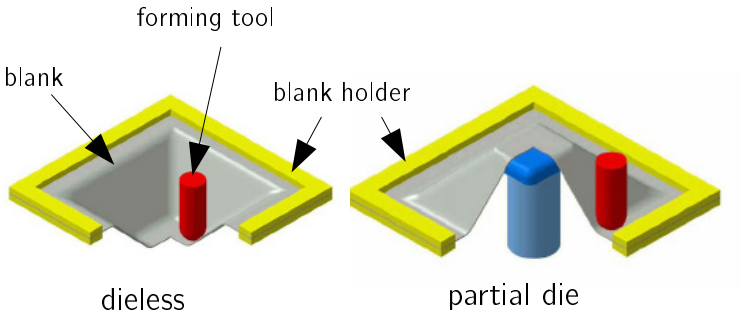

Figure 1: Comparison between Dieless and Partial Die ISF [4]

2.2 ISF with Incremental Die

As we have seen in the previous chapter, if the traditional ISF process is used we either have to use a full die, partial die or form without a die with reduced geometric accuracy.
Therefore a process variation has been developed that uses an Incremental Die as a die replacement (See Figure 2). We will call this variation „Incremental Sheet forming with Incremental Die“ (ISFID).

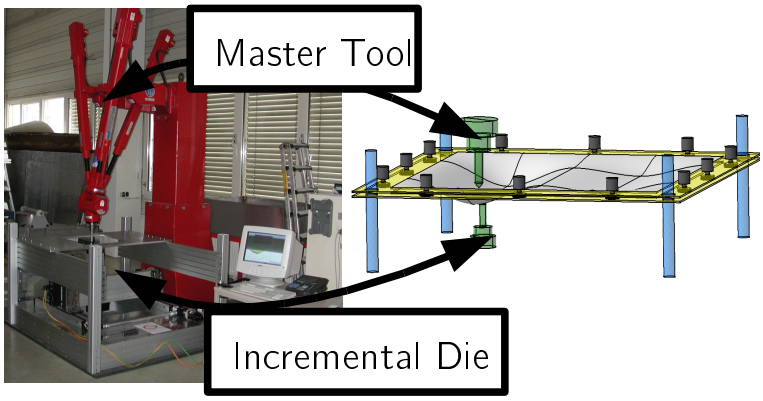

Figure 2: ISF with Incremental Die

By using such a device we have the possibility to support the sheet locally where it might be necessary. This brings us some benefits such as the following

- High flexibility because the tool path can be adapted for both devices independently
- Possibility to form parts with inner isles without removing the part form the blank holder
- Nearly no die tooling costs
- Geometric accuracy is higher then in traditional ISF dieless forming

For improving geometric accuracy of a part a method would be to form the part with a toolpath produced by a CAM software and afterwards measure it with a 3D scanner system. Make a new toolpath based on a springback correction and form the part once again. This procedure is very time consuming and requires a lot of iteration steps until the desired accuracy is reached. For this purpose we investigated the use of FEM methods to simulate the process of ISFID and furthermore reduce the currently necessary hardware trials to a minimum.

3 ISF with Incremental Die - Process Modelling

3.1 Basic Setup and Element Selection

The two tools are modelled as rigid bodies with a very fine mesh compared to the sheet which is modelled with shell elements or continuum elements (Figure 3).

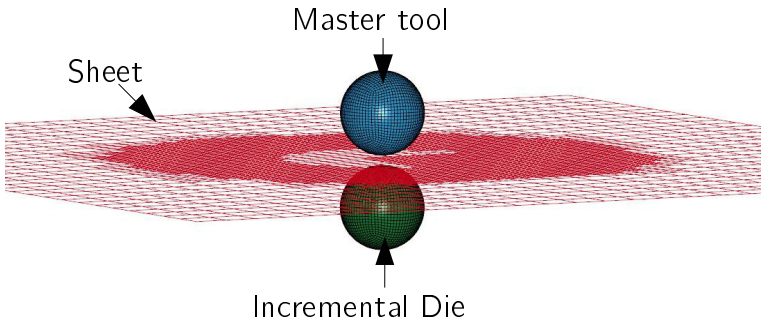

Figure 3: Simulation Setup

As you can see in Figure 3 the sheet is already meshed adaptively in the areas where the tools will move around later on. This mesh refinement is done with the `*CONTROL_ADAPTIVE_CURVE` Keyword which uses the tool path projection as input data. This way, there is no need for adaptive remeshing during the simulation and simulation time is reduced. The shell mesh itself is either of type 2 or 25 for simulations where a springback calculation is not necessary, or of type 16 or 26 where a springback calculation will be performed (type 16/26 are fully-integration elements). The types 25/26 are new types of shells which include thickness stretch to account for stresses which act normal to a shell element. Those elements are mainly used because in the process of ISF most of the stresses introduced by the two tools act normal to the sheet. If the standard type 2 shell is used, those forces are just neglected and therefore the results are not that accuracte. In Figure 4 you can see a comparison of the z-stress at a section cut right through the middle of the sheet at the end of the forming with a type 2 shell element (solid line) and a type 25 shell element (dotted line). If the thickness stretch is considered, as it is with type 25, then the stresses are much higher as with the type 2 element where the stresses are more or less equal to zero. Another possibility to include forces normal to the sheet surface would be to use continuum solid elements. A major drawback when using this kind of elements is the calculation time which increases by a factor of 10 to 20 if we select at least 3 elements through the thickness. A comparison of the FE-results optained by shell 25 and the solid element formulation has also shown that the solid elements can be replaced by the type 25 shell elements without having to worry about the simulation results as far as we only use 3 solids for modeling the sheet thickness. If we use more solid elements for the thickness the results may become more accurate compared to shell type 25/26.

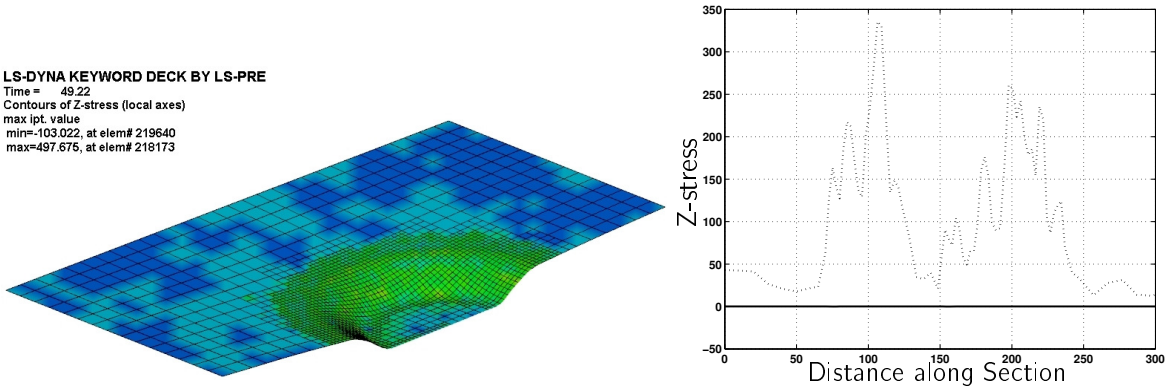

Figure 4: Comparison of local Z-Stress values between Shell element type 2 and type 25

In the FE setup the sheet is clamped on all four sides so no relative movement of the sheet is possible. The tool movement is realised using the `*BOUNDARY_PRESCRIBED_MOTION_RIGID` Keyword for each of the three

coordinate directions X,Y,Z. As the toolpath is one of the most critical parts in the simulation setup we will focus on this topic in section 4. The contact between the tools and the sheet is realised using a Surface to Surface forming contact (*CONTACT_FORMING_SURFACE_TO_SURFACE). Other contact descriptions like *CONTACT_AUTOMATIC_SURFACE_TO_SURFACE have shown to entail a larger calculation time for the process without any benefits concerning the results. The material model we use for the process depends mainly on the investigated material. For Al99.5 a different model then for TiGr3 or Ti6.4 is necessary. Section 3.2 focuses on this issue in more detail.

3.2 Material Modell selection

The mat model used for the simulation is *MAT_024 (*MAT_PIECEWISE_LINEAR_PLASTICITY) and *MAT_036 (3_PARAMETER_BARLAT). For both material models, tensile tests on strips were conducted to obtain the yield curve. In the tensile tests, rupture occurs at around 30% engineering strain. Therefore, an extrapolation of the obtained curve up to a true strain value of one is necessary because ISF entails such a large strain.
Furthermore, the anisotropic behavior for the BARLAT model is described through the „Lankford coefficient“ or the „R-value“. For practical purposes, measurements are taken of strains in width and length direction and the R-value is computed by the following equation, assuming a constant volume.

$$R = \frac{\ln(\frac{W}{W_0})}{\ln(\frac{W}{W_0}) + \ln(\frac{L}{L_0})}$$

Here, W and L are original width and length, respectively. Investigations on benchmark parts have shown that the anisotropy phenomena do not affect the forming by AISF since the forming zone is relatively small in AISF [2]. In most cases, the simulation results are accurate enough using the isotropic material model *MAT_024 for all kind of materials (TiGr3 or Al99.5 or Stainless Steel).

3.3 Mass scaling

As we use LS-Dyna (mainly an explicit FE solver), we have to take care of dynamic effects that may affect the simulation results. Especially if we want to speed up the whole simulation with methods like mass or time scaling. Normally the rigid tools move around at a maximum speed of 0.6m/s. This is related to the maximum speed of the system in reality. For some benchmark parts the process time in reality is around 20-30 minutes. If we use a time vs. position curve that equals exactly the movements in reality then the simulation time would be 2-3 weeks, compared to 20-30 minutes in reality. To speed up the simulation the first idea is a speed up for the tool movements, this results in a time scaling for the time vs. position curve. The second option is mass scaling where the minimum time step is fixed to a lower bound. If we use shell elements for the modeling of the sheet we are able to set the minimum time step to a value of 10^{-4}s compared to 10^{-8}s for the initial time step used without mass scaling. This reduces the simulation time significantly and our benchmark problem was solved in 2 hours. As we speed up the process, the dynamic effects influence the results more and more. To evaluate the simulation results, an important value is the variable `energy ratio` in the results file GLSTAT. It should be equal to 1 throughout the process. If the dynamic effects introduced by mass scaling or time scaling have a negative influence on the simulation results, then the energy ratio is the first variable to look at.
To combine a process speed up evoked by mass or time scaling with a good energy balance (the energy ratio is ≈ 1) the description of the tool path is one of the main influence parameters. If we can make shure that the description of it is „smooth“ then the dynamic effects will also be minimized. In section 4 a tool path development is shown that can be used to combine process speed up techniques with a perfect energy balance.

4 Tool Path generation

The tool path mainly determines the final accuracy of the final geometry. Therefore care has to be taken during its generation.
Generally we would like to have local support of our incremental die everywhere the master tool is currently in contact with the sheet. Therefore we use the master tool path as a basis to construct the tool path for the incremental die. In Figure 5 the relationship between the master tool and the incremental die is shown. If the tool path of the master tool is known then the tool center point of the incremental die XS,YS,ZS can be calculated (eq. 1).

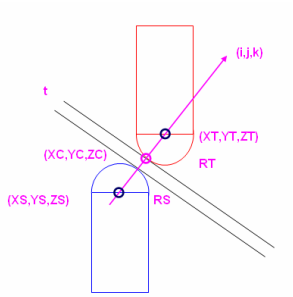

Figure 5: Relationship between Master Tool and Incremental Die

$$[XS, YS, ZS] = ([i, j, k] * (RS + RT + t)) + [XT, YT, ZT] \quad (1)$$

As the `*BOUNDARY_PRESCRIBED_MOTION_RIGID` requires a description of the tool path either in time vs. velocity, time vs. position or time vs. acceleration we did some trials with the time vs. position input curve as this is the easiest one to transfer to from a CAM software. Unfortunately most of the FE calculations had errors due to oscillations of the sheet. One reason is the toolpath description which is just not „smooth“ enough because internally LS-Dyna is differentiating from position to speed and finally to acceleration. As a result, if our time vs. position curve is not at least C^2-continuous we will get a time vs. acceleration curve that leads to the oscillations of the sheet. So we decided to directly input the tool path data in the form time vs. acceleration. By using this description we also make sure that the acceleration will never exceed values that the machines in reality also could not attain. A similar approach has already been performed for the AISF process [1]. In this paper the authors used a time vs. velocity approach for the tool path description of geometries that can be divided in simple primitives like lines or circle segments. As we need a general approach to generate the tool path for free form surfaces that cannot be divided into simple primitives a new development as we show here is necessary.
We will not focus directly on how we generated the basic toolpath for both machines as this depends a lot on the CAM software. The input we use to generate the time vs. acceleration curves for LS-Dyna is based on tool path curves in the B-Spline format for the master tool as well as the incremental die. We generate those B-Spline curves with a self written software working more or less like a CAM software e.g. the final tool path is like a z-level milling operation with a constant z-pitch (see Figure 6).

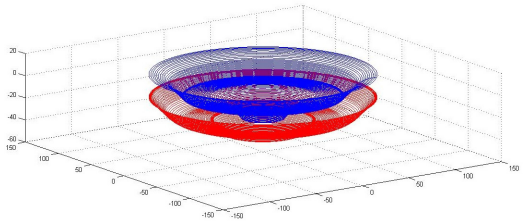

Figure 6: Master toolpath and Incremental Die toolpath

4.1 The velocity function

In this part we construct a function $v(t)$ wich describes the velocity of the master tool movement along a given curve φ. On the basis of $v(t)$ and the time-derivatives of the the input curve we compute the acceleration values for the output in X,Y and Z direction. To make sure that the absolute velocity and acceleration of the tool movement do not exceed the given specifications in form of v_{max} and a_{max}, we have to compute the parameter set wich defines the velocity function in dependence of φ. For our computations we used the following Parameters:

$\varphi(u)$	circle ($r = 200$mm)
v_{max}	$600\frac{mm}{s}$
a_{max}	$3000\frac{mm}{s^2}$
Δz	1mm
N	4000

Table 1: Parameter

4.1.1 Parameter Determination of the velocity-function

Let $\varphi(u) \in \mathbb{R}^3$, $u \in [u_0, u_1]$ be an arbitrary C^1-Curve wich describes the toolpath. Let t_0 represent the start time and $\varphi(u_0)$ the according start position.
First we have to compute the arc length S of φ before we are able to build the velocity function. The arc length is defined as

$$S = \int_{u_0}^{u_1} \|\dot{\varphi}(u)\| du \tag{2}$$

We will now build the velocity-functions v(t) and compute the timevalue T in such a way that it fullfills the following conditions:

$$\begin{aligned}
&(a) \quad \int_{t_0}^{T} v(t) = S \\
&(b) \quad v(t_0) = v(T) = 0 \\
&(c) \quad \dot{v}(t_0) = \dot{v}(T) = 0 \\
&(d) \quad \max_{t\in[t_0,T]} v(t) \leqq v_{max} \\
&(e) \quad \max_{t\in[t_0,T]} \dot{v}(t) \leqq a_{max}
\end{aligned}$$

We construct v(t) as a combination of three functions as follows.

$$v(t; v_0, a_0) := \begin{cases} v_a(t; v_0, a_0), & \text{if } t \in [t_o, t_1] \\ v_0, & \text{if } t \in [t_1, t_2] \\ v_b(t; v_0, a_0), & \text{if } t \in [t_2, T] \end{cases}$$

$$\begin{aligned}
\text{with} \quad & v_a(t; v_0, a_0) := c_1 * (\frac{\sin(c_2(t - t_0) - \pi)}{c_2} + (t - t_0)) \\
& v_b(t; v_0, a_0) := v_0 - v_a(t - t_2; v_0, a_0)
\end{aligned}$$

$$\text{and} \quad c_1 := \frac{a_0}{2} \quad , \quad c_2 := \frac{\pi a_0}{v_0}$$

Note: The Notation $v(t)$ is equivalent to $v(t; v_0, a_0)$
To make sure that $v(t)$ is C^1-continuous we choose

$$t_1 = \frac{2v_0}{a_0} + t_0 \tag{3}$$

Through derivation we obtain

$$a(t) := \frac{\partial v(t)}{\partial t} \tag{4}$$

and now it follows that

$$\max_{t\in[t_0,T]} v(t) \leqq v_0 \quad \text{and} \quad \max_{t\in[t_0,T]} a(t) \leqq a_0 \qquad \text{(see figure 7)}$$

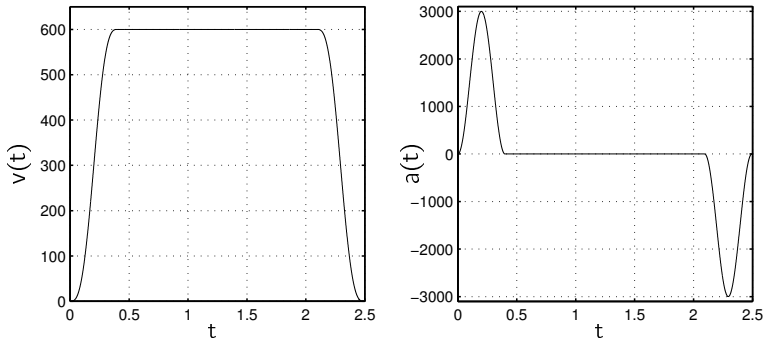

Figure 7: v(t) and a(t)

Based on the arc length S we determine the parameter values v_o, t_2 and T. The computation of a_0 is more complex and will be discussed in 4.2.3. However we need a_0 first to determine the other parameters.

(We only compute this parameter set for the master tool path φ. For the computation of the parameter values for the incremental die see 4.3)

Define the distance function

$$s(t; v_0, a_0) := \int_{t_0}^{t} v(\tau; v_0, a_0) d\tau$$

Set $a_0 = \hat{a}_{max}$ as computed in 4.2.3

A distinction is made between the following two cases:

1. $s(t_1; v_{max}, \hat{a}_{max}) = \int_{t_0}^{t_1} v_a(t; v_{max}, \hat{a}_{max}) dt \leqq \frac{S}{2}$

   If this condition is fulfilled then set

   $$v_0 = v_{max}$$

   t_1 is determined as in (3) and t_2, T will be computed as follows:

   $$t_2 = t_1 + \frac{S - 2 * s(t_1; v_{max}, \hat{a}_{max})}{v_0}$$
   $$T = t_2 + (t_1 - t_0)$$

2. $s(t_1; v_{max}, \hat{a}_{max}) = \int_{t_0}^{t_1} v_a(t; v_{max}, \hat{a}_{max}) dt > \frac{S}{2}$

   We determine v_0 from the equation

   $$s(t_1; v_0, \hat{a}_{max}) = \frac{S}{2}$$

   t_1 is now determined as in (3) and set

   $$t_2 = t_1$$
   $$T = 2t_1 - t_0$$

4.2 Time vs. Acceleration Output

4.2.1 Acceleration components

Define $\varphi_t^x(t) := \frac{(\frac{\partial \varphi_1(u(t))}{\partial t})}{\|(\frac{\partial \varphi_1(u(t))}{\partial t})\|}$ and $\varphi_{tt}^x(t) := \frac{\partial \varphi_t^x(t)}{\partial t}$

The acceleration function $\hat{a}_x(t)$ in x-direction is determined by

$$\hat{a}_x(t) = \frac{\partial(v(t) * \varphi_t^x(t))}{\partial t} = a(t) * \varphi_t^x(t) + v(t) * \varphi_{tt}^x(t) \tag{5}$$

$\hat{a}_y(t)$ and $\hat{a}_z(t)$ the same way. For the computation of the components, we need in general an approximation for the time derivatives of the input curve as shown in 4.2.2.

4.2.2 The time derivative of the input curve

Set $\Delta t := \frac{T - t_0}{N}$ where $N \in \mathbb{N}$ is the desired number of output values for a single path.

$$\text{Define} \quad \hat{T} = \{\hat{t}_0, ..., \hat{t}_N\} \quad \text{with} \quad \hat{t}_i := t_0 + i * \Delta t \quad , \quad i \in \{0, .., N\}$$

The appropriate distances result in

$$\hat{S} = \{\hat{s}_0, ..., \hat{s}_N\} \quad \text{with} \quad \hat{s}_i := s(\hat{t}_i; v_0) \quad , \quad i \in \{0, .., N\}$$

According to $\hat{S}$ we compute the parameter values $\hat{u}_i \in [u_0, u_1]$ (e.g. using the Jacobi-algorithm) wich fulfill the following condition

$$\hat{s}_i = \int_{u_0}^{\hat{u}_i} \|\varphi(u)\| du$$

and wich leads to

$$\hat{U} = \{\hat{u}_0, ..., \hat{u}_N\} \quad , \quad i \in \{0, .., N\}$$

Finally we can define the numerical derivatives $\hat{\varphi}_t^x(t)$,$\hat{\varphi}_t^x(t)$ as an approximation of $\varphi_t^x(t)$,$\varphi_{tt}^x(t)$ through

$$\hat{\varphi}_t^x(\hat{t}_i) := \frac{\frac{(\varphi_1(\hat{u}_{i+1}) - \varphi_1(\hat{u}_{i-1})}{2 * \Delta t}}{\|\frac{(\varphi_1(\hat{u}_{i+1}) - \varphi_1(\hat{u}_{i-1})}{2 * \Delta t}\|} \quad \text{and} \quad \hat{\varphi}_{tt}^x(\hat{t}_i) := \frac{(\hat{\varphi}_t^x(\hat{t}_{i+1}) - \hat{\varphi}_t^x(\hat{t}_{i-1}))}{2 * \Delta t} \tag{6}$$

We are now able to compute (5) for the discretisation $\hat{T}$ (see Figure 8).

$$\begin{aligned}
\hat{a}_x(\hat{t}_i) &= a(\hat{t}_i) * \hat{\varphi}_t^x(\hat{t}_i) + v(\hat{t}_i) * \hat{\varphi}_{tt}^x(\hat{t}_i) \quad , \quad i \in \{0, .., N\} \\
\hat{a}_y(\hat{t}_i) &= a(\hat{t}_i) * \hat{\varphi}_t^y(\hat{t}_i) + v(\hat{t}_i) * \hat{\varphi}_{tt}^y(\hat{t}_i) \quad , \quad i \in \{0, .., N\} \\
\hat{a}_z(\hat{t}_i) &= a(\hat{t}_i) * \hat{\varphi}_t^z(\hat{t}_i) + v(\hat{t}_i) * \hat{\varphi}_{tt}^z(\hat{t}_i) \quad , \quad i \in \{0, .., N\}
\end{aligned}$$

In Figure 9, 10 and 11 we can see the acceleration-fucntions in X,Y and Z direction (Figure 9) and the

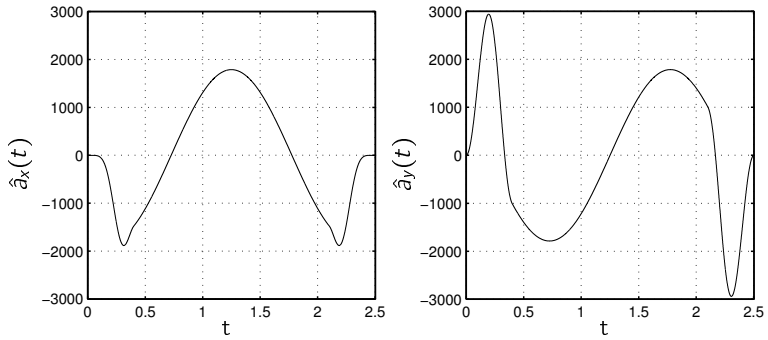

Figure 8: acceleration in X,Y direction for one circle

corresponding velocity- (Figure 10) and distance-functions (Figure 11) (which result through integration) for

three connected circles with constant z-levels $z_1 = -1$, $z_2 = -2$, $z_3 = -3$ and the radii $r_1 = 200$, $r_2 = 199$, $r_3 = 198$ (see Table 1 for the used parameters)

4.2.3 An estimation for a_0

We want to be sure that the absolute acceleration $\hat{a}(t)$ of the master tool is not exceeding the user-specified value a_{max}. Hence we try to estimate a_0 in such a way that the initial condition

$$\hat{a}(t) = \sqrt{\hat{a}_x^2(t) + \hat{a}_y^2(t) + \hat{a}_z^2(t)} \leqq a_{max} \quad , \quad \forall t \in [t_0, T] \tag{7}$$

will be fullfilled. In the first place we compute an approximation of $\hat{\varphi}_{tt}$ with $a_0 = a_{max}$ as in 4.1.1 - 4.2.2. We set

$$\begin{aligned} m_1 &:= \min_{t\in[t_0,T]} (\hat{\varphi}^x_{tt}(t) + \hat{\varphi}^y_{tt}(t) + \hat{\varphi}^z_{tt}(t)) \\ m_2 &:= \min_{t\in[t_0,T]} (\hat{\varphi}^x_{tt}(t)^2 + \hat{\varphi}^y_{tt}(t)^2 + \hat{\varphi}^z_{tt}(t)^2) \\ M &:= \max_{t\in[t_0,T]} (\hat{\varphi}^x_{tt}(t)^2 + \hat{\varphi}^y_{tt}(t)^2 + \hat{\varphi}^z_{tt}(t)^2) \end{aligned}$$

For the case that the maximum of $\hat{a}(t)$ occur in $[t_1, t_2]$ where the tool runs with constant velocity we have to adjust v_{max} first.
That means if $v_{max}^2 * M > a_{max}^2$ then set $v_{max} := \frac{a_{max}}{\sqrt{M}}$.
Now we estimate a_0 at $t = \frac{t_1}{2}$ where the function a(t) (see (4)) has its maximum. We obtain with (3) that $v(\frac{t_1}{2}) = \frac{v_{max}}{2}$ and $a(\frac{t_1}{2}) = a_{max}$. With $t = \frac{t_1}{2}$ and the equations (5),(7) we receive

$$0 \geqq \hat{a}(\frac{t_1}{2}) - a_{max}$$

$$\begin{aligned} \Rightarrow 0 \geqq &((\hat{a}_{max} * \hat{\varphi}^x_t(\frac{t_1}{2}) + \frac{v_{max}}{2} * \hat{\varphi}^x_{tt}(\frac{t_1}{2}))^2 \\ &+(\hat{a}_{max} * \hat{\varphi}^y_t(\frac{t_1}{2}) + \frac{v_{max}}{2} * \hat{\varphi}^y_{tt}(\frac{t_1}{2}))^2 \\ &+(\hat{a}_{max} * \hat{\varphi}^z_t(\frac{t_1}{2}) + \frac{v_{max}}{2} * \hat{\varphi}^z_{tt}(\frac{t_1}{2}))^2) - a_{max}^2 \end{aligned}$$

$$\begin{aligned} \Rightarrow 0 &\geqq 3\hat{a}_{max}^2 + v_{max}\hat{a}_{max} * (\hat{\varphi}^x_{tt}(\frac{t_1}{2}) + \hat{\varphi}^y_{tt}(\frac{t_1}{2}) + \hat{\varphi}^z_{tt}(\frac{t_1}{2})) + \frac{v_{max}^2}{4}(\hat{\varphi}^x_{tt}(\frac{t_1}{2})^2 + \hat{\varphi}^y_{tt}(\frac{t_1}{2})^2 + \hat{\varphi}^z_{tt}(\frac{t_1}{2})^2) - a_{max}^2 \\ &\geqq 3\hat{a}_{max}^2 + v_{max} * m_1 * \hat{a}_{max} + \frac{v_{max}^2}{4} * m_2 - a_{max}^2 \end{aligned}$$

Finally set $\hat{a}_{max} := \frac{-b+\sqrt{b^2-4ac}}{2a}$ with $a = 3$, $b = v_{max} * m_1$, $c = (\frac{v_{max}^2}{4} * m_2) - a_{max}^2$

4.3 Parameter values for the incremental die tool path

For the incremental die tool path $\gamma(w) \in \mathbb{R}^3$, $w \in [w_0, w_1]$ we compute the arc length $\hat{S}$ as in (2). Afterwards we determine the maximum velocity $\hat{v}_0$ and the parameter $\hat{a}_0$ with

$$\begin{aligned} \hat{v}_0 &= \frac{\hat{S}}{T - t_0} \\ \hat{a}_0 &= a_0 * \frac{\hat{v}_0}{v_0} \end{aligned}$$

using the time-value T computed already for the master tool path. As for φ we just continue at 4.2.1 and then both tool paths are in a form that can be used as a time vs. acceleration curve for the LS-Dyna solver.

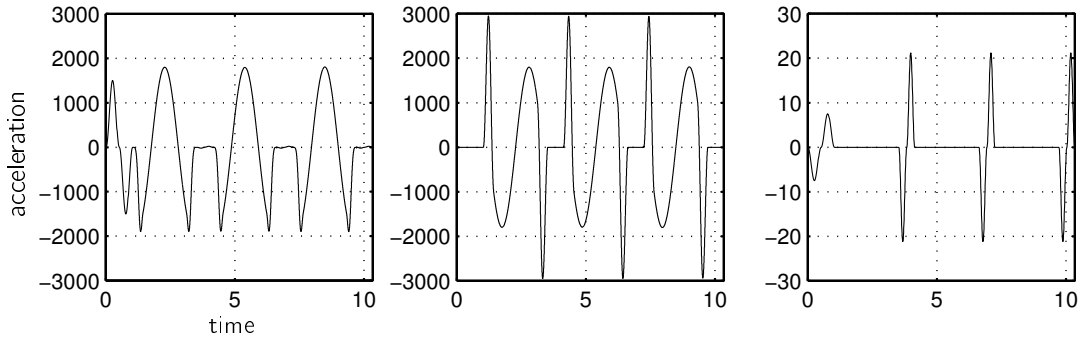

Figure 9: acceleration curves in X, Y and Z direction

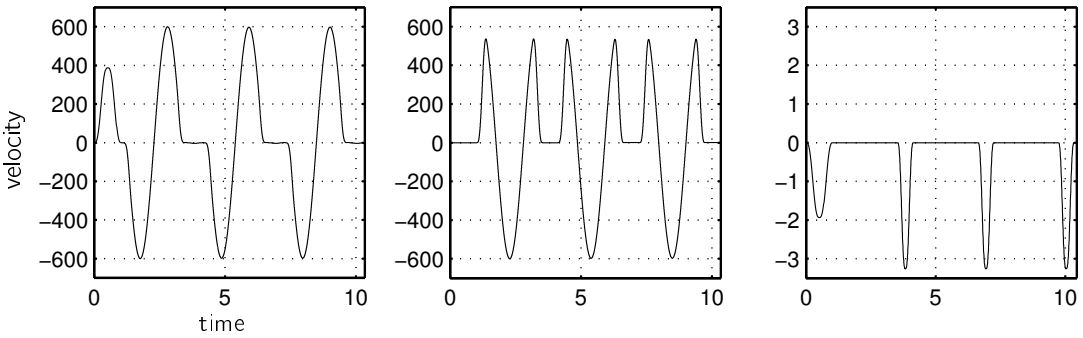

Figure 10: velocity curves in X, Y and Z direction

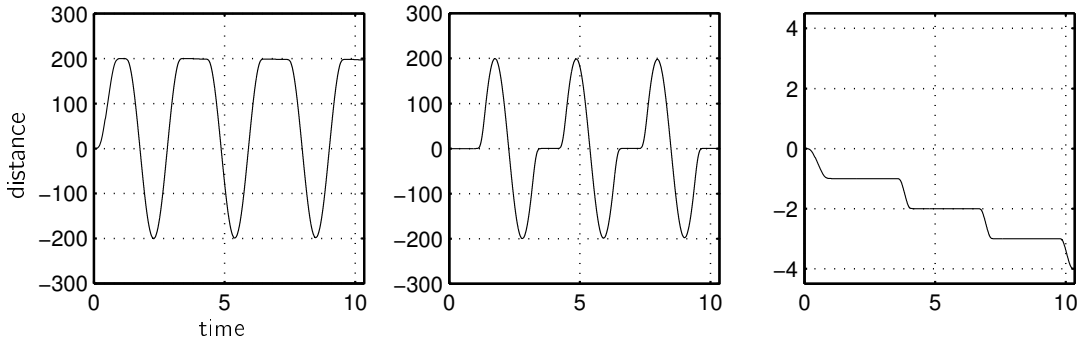

Figure 11: distance curves in X, Y and Z direction

5 Summary and Outlook

The main focus of this paper is the creation of a „smooth“ C^2-contigous tool path as the toolpath is an important aspect in the ISF process simulation. The current approach we use is suitable for all kind of input curves as long as they are represented as B-Spline curves. In this paper we only presented the tool path generation for the incremental die based on the tool path of the master die. We realise this with a simple vector multiplication that can be seen in Figure 5. The result is a synchronised movement between the both tools. A lokal support for the master tool is realised in this way. We managed to use the acceleration of the tools as an input value in the time vs. acceleration curve. With our approach the most possible continuous tool path for arbitrary curves is usable for the simulation of the SPIF or AISF or ISFID process. Simulation speed ups with mass scaling will now be possible with an excellent energy ratio.
As the ISFID process is very flexible we are also doing some variations where only one tool is moving downwards in z-direction and the other one is staying at the same z-level throughout the forming. With this variation sharp corners can be formed much better then in the standard SPIF process [7].
In the near future a focus is on the springback behaviour in ISFID and the correction of it. Especially the ISFID system is suited very well for the correction of geometries as either the incremental die or the master tool can push and support the sheet. No new die has to be manufactured. The idea is to perform a few iteration steps by simulation of corrected tool paths until the desired geometry is reached. After the simulation a „spring back optimized“ tool path is used for the machine to form the part.

6 Acknowledgement

We would like to thank the European Government for the support of this work in the framework of the „SCULPTOR“ project in which the authors currently work on the development of an incremental die system in combination with CAD/CAM software development.

7 Literatur

[1] Bambach, M. ; Hirt, G. : Error Analysis In Explicit Finite Element Analysis Of Incremental Sheet Forming. In: Cesar de Sa, J. M. A. (Hrsg.) ; Santos, A. D. (Hrsg.): *American Institute of Physics Conference Series* Bd. 908, 2007 (American Institute of Physics Conference Series), S. 859–864

[2] Bambach, M. ; Hirt, G. ; Junk, S. : Modelling and Experimental Evaluation of the Incremental CNC Sheet Metal Forming Process. In: *7th International Conference on Computational Plasticity*, 2003

[3] Breun, F. ; Wohnig, W. : Werkzeugbau für Prototypen, Klein- und Mittelserien. In: *Neuere Entwicklungen in der Blechumformung* (1990)

[4] Hirt, G. ; Ames, J. ; Bambach, M. : Economical and ecological benefits of CNC Incremental Sheet Forming (ISF). In: *ICEM*, 2003

[5] Hirt, G. ; Ames, J. ; Bambach, M. : Basic Investigation into the Characteristics of dies and Support Tools used in CNC-Incremental Sheet Forming. In: *IDDRG*, 2006

[6] Liewald, M. : New Developments in Forming Technologies for Aluminium. In: *Aluform Congress*, 2006

[7] Maidagan, E. ; Zettler, J. ; Bambach, M. ; Rodríguez, P. P. ; Hirt, G. : A new incremental sheet forming process based on a flexible supporting die system. In: Trans Tech Publications (Hrsg.): *Key Engineering Materials* Bd. 344, 2007, S. 607–614

[8] McKinsey&Company ; TU Darmstadt: *HAWK 2015 - Herausforderung Automobile Wertschöpfungskette*. 2003. – Materialien zur Automobilindustrie, Eigendruck, Verband der Automobilindustrie e. V.

[9] McKinsey&Company ; WZL of RWTH Aachen University: *Tomorrow's Automotive Production*. 2006

