# Application of Task-to-Method Transform to Laser Seam Welding

Jürgen Pollak
Institut für Angewandte Forschung
Hochschule Karlsruhe - Technik und Wirtschaft
Karlsruhe, Germany
e-mail: juergen.pollak@hs-karlsruhe.de

***Abstract*—Intelligent machines are supposed to automatically set process parameters when faced with a task to be processed. The intelligence is often realized by databases which link the task with process parameters. This paper reviews a flexible and portable (to various processes) system to find (optimized) process parameters which force the process outcome to pre-defined quality under given variable conditions. In addition, extensions to the original system are presented and the whole concept is applied to laser seam welding (LSW). Experimental results based on real process executions demonstrate the applicability in industrial environments.**

***Keywords*–Machine Intelligence; Task-to-Method-Mapping; Support Vector Regression; Data Domain Description; Laser Welding**

## I. Introduction

Intelligent production machines have to flexibly respond to varying tasks by setting their process parameters in such a way that given task goals are reached under given conditions. For this purpose, the machine needs to represent and use knowledge about the relation between process parameters and process goals under given (but varying) process conditions.

This paper reviews a general concept and an implementation of the automatic extraction and application of such process knowledge represented in experimental outcome data. [1] A process goal is represented by quality measure values to be achieved by the process. In experiments, a process is executed with defined process parameters and under known or controlled conditions. The achieved quality is measured after the experimental process execution. The data (process parameter, condition and quality values) may be sampled from real physical, or simulated numerical experiments. The data form the basis for the estimation of a so-called goal function with the process parameters and the condition quantities as independent variables. The goal function defines quantities which describe the desired end state of the process and represents the process knowledge. Once derived from the data, it is used to find the process parameters which yield a desired result. Finding the appropriate process parameter settings (“process methods”) yielding a given goal is then equivalent to finding those parameter values, where the goal function takes on the desired, given goal values. More than one solution exists, the set of all solutions is called ”level set”. To select the best suitable method, it is further proposed to use the level set as a basis to optimise a given cost function which associates cost with the process parameters.

The goal function is constructed by applying non-linear kernel regression to the experimental data. Experiments frequently also deliver process boundaries, beyond which the process will not execute or no result is reached at all. This feasibility boundary is modelled in our approach via a two-class support vector machine. Furthermore, it has to be taken into account that the goal function can only be applied in areas which are supported by experimental data. This so-called confidence domain is modelled by hulls enclosing the experimental data.

The paper is organized as follows: Section II gives a review of the originally developed Task-to-Method Transformation (T2MT). This section is divided into subsections giving an overview over the general concept, followed by details about the process modelling by goal functions and a classifier to constrain the predictions to regions supported by data. The last subsection of Section II presents the procedure how to find process parameters from given tasks and process models. The next part, in Section III, applies the methods to LSW and describes various extensions to the original system. A short introduction to LSW is given in Section III-A. Section III-B extends the process model by multi-valued goals. Acceptable goal ranges are introduced in Section III-C. The calibration of models to new situations (Section III-D) and using a process model to apply small parameter adjustments (Section III-E) are further enhancements. All these extensions convert the original T2MT into an industrial applicable system. An experimental verification using real process data is given in Section III-F.

## II. Summary of T2MT

This chapter gives a summary of the T2MT, which was introduced in [1]. It was developed as a general applicable system to find process parameters from process models derived from experimental data. T2MT was originally verified by numerical simulations for the process of resistance spot welding.

### A. General Concept

Process parameters describe the variable control quantities, which can be set by the process machine in a vector $\vec{p}$. The process conditions represent all fixed quantities in a vector $\vec{c}$, which otherwise govern the process and cannot be set by the machine. They are fixed externally and independently from process execution. The goals are quantities characterising the desired end state of the process in a vector $\vec{g}$. For example, in car seat manufacturing metal sheets are joined by welding seams. The process parameters are in the simplest case laser power, laser focus point and welding speed. Possibly varying conditions are the materials and thicknesses of the two sheets. The goal is the double valued extend of the welding seam, seam width and seam depth, which have to be obtained.

The task is then given by the combination of the goals and the conditions $\vec{t} = [\vec{g}, \vec{c}]$. An intelligent machine has to find at least one method (consisting of process parameters $\vec{p}$) fulfilling a given task $\vec{t}$, or it has to state that the task is not feasible. In other words, the machine has to perform a mapping from $\vec{t}$ to $\vec{p}$. We call this the T2MT.

ISBN: 978-1-61208-437-4

In experimental process investigations, a variety of process conditions is explored. For each specific condition, a set of methods $\vec{p}$ is applied and the resulting goal values $\vec{g}$ are measured. Each single experiment gives a vector triple $[\vec{g}, \vec{p}, \vec{c}]$ and the available experimental series give a set of such triples. We propose to build an abstraction of the experimental data by the formation of a goal function $\vec{g}(\vec{p}, \vec{c})$. It represents the knowledge contained in the experimental data.

Furthermore, the goal function should only be applied in areas supported by experimental data. This so-called confidence domain is modelled by hulls enclosing the experimental data. A support function $s(\vec{p}, \vec{c}) > 1$ can be defined inside the hulls, $s(\vec{p}, \vec{c}) = 1$ on the hulls and dropping continuously to $s(\vec{p}, \vec{c}) = 0$ within some distance outside. This support function defines some space around the experimental data, which may be accepted as a region for inter- and extrapolations of new (yet unseen) tasks $\vec{t}$ and methods $\vec{p}$.

The goal function $\vec{g}(\vec{p}, \vec{c})$ and the support function $s(\vec{p}, \vec{c})$ finally form the process knowledge model, extracted from the experimental data.

The goal function is then used to perform the T2MT. The condition vector $\vec{c}_{\vec{t}}$ is a constant, when a specific task $\vec{t}$ is given. In this case, the goal function is only a function over the corresponding subspace of $\vec{p}$. The level set of parameter vectors $\vec{p}$ defined by $\vec{g}(\vec{p}, \vec{c}) = \vec{g}_{\vec{t}}$ represents the set of methods fulfilling the task. Finding the level set of the goal function is thus the core component of the T2MT. Afterwards, the other model function $s(\vec{p}, \vec{c})$ is applied to the level set to exclude unsupported method solutions.

The resulting restricted solution set forms the search space for the minimisation of a cost function. Based on external knowledge, the cost function assigns cost to the process parameters and process goals.

The solution for the vector-valued goal function $\vec{g}(\vec{p}, \vec{c}) = \vec{g}_{\vec{t}}$ with $\vec{g}, \vec{g}_{\vec{t}} \in \mathbb{R}^M$ can be broken down into the solution for $M$ single-valued goal functions $g_i(\vec{p}, \vec{c}) = g_{\vec{t},i}, \ \ i = 1, \ldots, M$. Each of them has a level set $\{\vec{p}\}_i$ as a solution. The level set satisfying all equations is given by the intersection of all single sets $\{\vec{p}\}_1 \cap \{\vec{p}\}_2 \cap \cdots \cap \{\vec{p}\}_M$. It is therefore sufficient to construct a method for single-valued goal functions.

### B. Goal Function Approximation

A central part, when looking for a method $\vec{p}$ solving a given task $\vec{t} = [\vec{g}, \vec{c}]$ under constraints $\vec{c}$, is a model description of the physical process. The process is modelled by construction of a goal function $\vec{g}(\vec{p}, \vec{c})$ which comprises the whole necessary knowledge about the process. In any experiment, the conditions $\vec{c}$ and the method parameters $\vec{p}$ are set. The process is then executed and the outcome is measured. The outcome quantities are identical with the goal describing quantities $\vec{g}$, which describe the desired final properties of the process result. Experiments are conducted under many different $\vec{c}_i$ and $\vec{p}_i$ and corresponding $\vec{g}_i$ are measured. This gives an experimental sample of triples $\{\vec{p}_i, \vec{c}_i, \vec{g}_i\}_{i=1}^N$, which is used to create an abstraction in the form of a goal function. Subsequently, this goal function can be inverted to find appropriate parameters for given goals and conditions. For a given task $\vec{t}$ the conditions $\vec{c}$ are fixed and the goal function $\vec{g}(\vec{p})$ depends only on process parameters $\vec{p}$. In most cases there is no explicit prior model available to form this function. Therefore, the goal function has to be extracted from experimental (real or simulated) data.

To represent the goal function, basically any regression method can be used. All methods build up the regression function by a weighted superposition of base functions, which itself may need parametrisation. A fitting algorithm is applied to determine the weights and parameters of the base functions so that the superposition approximates the observed data as accurately as possible. For most methods, the number of base functions must be specified in advance (and by association, the complexity of the representable function).

Real production processes may show very complicated non-linear dependencies on process parameters. But piecewise, in the small surrounding of an assumed operating point, the process model function behaves quite smooth. Support Vector Regression (SVR) is an universal method to find the smoothest regression function representing observed data. [2] The regression function is build up by a superposition of more or less localized non-linear functions (depending on kernel choice) pinned at (measured or simulated) data vectors. It is beneficial that SVR picks out only the relevant subset of the whole data set to describe the smooth goal function. These data vectors, which determine the function to represent this experimental knowledge in a generalized way by the goal function $\vec{g}(\vec{p}, \vec{c})$, are called Support Vectors.

The goal function by SVR representation takes the form

$$g(\vec{x}, \vec{\alpha}) = \sum_{i=1}^{l} \alpha_i K(\vec{x}, \vec{x}_i) - \rho, \quad \vec{\alpha} = (\alpha_1, \ldots, \alpha_l), \tag{1}$$

where the parameters $\alpha_i$ and $\rho$ are determined by an quadratic optimization algorithm from the data. [2]

Two kernels, used in calculation for the present paper, were

$$\begin{aligned} \text{Polynomial } K(\vec{p}, \vec{q}) &= (\gamma\langle\vec{p}, \vec{q}\rangle + c)^d \quad \text{and} && (2)\\ \text{Gaussian RBF } K(\vec{p}, \vec{q}) &= \exp(-\gamma\|\vec{p} - \vec{q}\|^2). && (3) \end{aligned}$$

The free parameters $(\gamma, d)$ in these kernels are found and fixed by exploring the $(\gamma, d)$-space for values of minimum residual fitting error by cross validation. [3]

When doing the numerics, especially with polynomial kernels, one will face numerical issues when using the raw values of the process quantities. They are more significant if some quantities have very small values, others very high values, and if the dynamic ranges are very different. These issues can be circumvented by normalizing the training data to range $[0, 1]$ and de-normalizing the results accordingly.

Processes usually have sharp boundaries in the space of parameters and conditions, beyond which the process collapses or exhibits unacceptable behavior. This region of unfeasible processes could be represented by a special goal value. But this would result in a discontinuity of the goal function (1) and consequently in fitting problems. We propose to represent the feasibility region within the boundary by a separate step function, changing value at the boundary. The feasibility region is then represented by a two-class (feasible / unfeasible) classifier. The model of the feasibility region is formed by the training of a Support Vector Machine [2], which requires training data covering both classes.

### C. Data Support Region

Experimental data will usually explore some finite areas in the $(\vec{p}, \vec{c})$-space, while the goal function covers the whole space. In order to get a reliable functional approximation of the goal function, it is necessary to restrict the goal function to areas supported by experimental data.

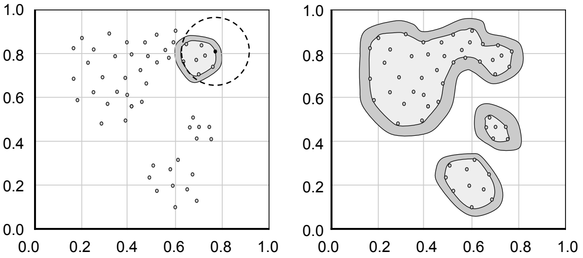

Figure 1. Local (a) and global (b) SVRM boundaries.

The region in input space, defined by the hull enclosing the training data $\{\vec{p}_i, \vec{c}_i, \vec{g}_i\}_{i=1}^{N}$, will be described by the support function $s(\vec{p}, \vec{c})$. The generalization (interpolation and extrapolation) of the experimental data by the goal function is only valid inside this hull and in a small region around it. We call this whole trusted region the data supported region.

In T2MT, this region is modelled by a one-class support vector machine [4] with parameters calculated from the training data set, inspired by the work [5].

According to [4], the problem to describe the supported region may be solved by a mapping of the training data to a feature space where they are linearily separated from the origin. The separating hyperplane with maximum distance from the origin determines the boundary.

The region of known data (corresponding to the set of known tasks) can be formulated as [1]

$$\frac{1}{\rho}\sum_{i}^{S_N} c_i K(\vec{x}, \vec{x}_i) \geq 1, \tag{4}$$

where

$$K(\vec{x}, \vec{y}) = \exp\left(-\gamma\, |\vec{x} - \vec{y}|^2\right) \tag{5}$$

and

$$\rho = \sum_{j}^{S_N} \alpha_j K(\vec{x}_i, \vec{x}_j), \tag{6}$$

with $S_N =$ number of support vectors.

In order to accept a new task some distance away from the already known task (which means to accept the value of our goal function in an extrapolated region), we soften the boundary definition:

$$\begin{aligned} \frac{1}{\rho}\sum_{i}^{S_N} c_i K(\vec{x}, \vec{x}_i) &\geq p & (7)\\ p &\approx 0.9 \qquad (0 < p \leq 1) & (8) \end{aligned}$$

The setting of parameter $p$ determines the range of how far extrapolations of the goal function are accepted. The value of $p$ depends on the sample density of the training data and on pecularities of the process under consideration. It has to be found by experiments. In Figure 1, the boundary of the light grey region is defined by $p = 1$. In this area data is available and the goal function interpolates in between. In the dark grey region, defined by $p < 1$, the goal function extrapolates to regions not supported by training data. In this region, the results of the goal function are assumed to be uncertain but the result shall be accepted. Outside the dark region results of the goal function are rejected as untrustable.

A still open question is how to choose the width $\gamma$ of the Gaussian Kernel function (5). If these functions are highly localized, the boundary around the training data will be very sharp with poor generalization performance and a large number of support vectors. If the kernel functions on the other hand are too broad, the resulting boundary may be too smooth (only a small part of the training vectors are considered as support vectors) and occlude essential structures. It is therefore crucial to find a value $\gamma$ producing a good description of the training data.

Cross-validation cannot be used, because there is no false-class in our training set. We follow the procedure presented in [5], which is summarized shortly.

In a first step all training vectors which might lie on the data domain boundary are identified (Figure 1a) by local one-class Support Vector Machine Classifiers (in [5], [6] this is called SVRM - Support Vector Representation Machine). Every training vector is considered with its surrounding vectors inside a sphere of a given radius. It can be assumed that this restricted sub-sample follows a simple distribution, therefore it is justified to choose a local $\gamma_i = d_i$. If the training vector $\vec{x}_i$ lies on the boundary (or very near the bounday) of the sphere, this vector is stored in a list of 'local' boundary vector candidates of the total sample.

The second step is to train several global SVRMs using all training data, each with different $\gamma$. For every cycle, the training vectors lying on or very close to the boundary of the global SVRM are selected and stored in lists of 'global' boundary vector candidates.

In a last step (Figure 1b) the global $\gamma$ is chosen, for which the best match is found between the set of 'global' boundary vector candidates and the set of potential (local) boundary vector candidates. Further details about the algorithm can be found in [1].

### D. Parameter Extraction (Level Set)

The goal function $\vec{g}(\vec{p}, \vec{c})$ (1), defined in Section II-B, represents a surface embedded into a high-dimensional space spanned by the process parameters $\vec{p}$ and process conditions $\vec{c}$. A specific task is then given by the demand to reach a task goal $\vec{g}_{\vec{t}} = \vec{g}(\vec{p}, \vec{c}_{\vec{t}})$. This can be viewed as the $\vec{p}$-dependent intersection of a hyperplane $\vec{g}_{\vec{t}} =$ const with the curved surface $\vec{g}(\vec{p}, \vec{c}_{\vec{t}})$. The level set is the set of solutions

$$\text{level set: } \{\, \vec{p} \,|\, \vec{g}_{\vec{t}} = \vec{g}(\vec{p}, \vec{c}_{\vec{t}}) \,\}\,. \tag{9}$$

In the present case, the level set can be found by meshing the high-dimensional feature space. The mesh is refined by incrementally subdividing cells, which are intersected by $\vec{g}_{\vec{t}}$, until the desired accuracy is reached. The level set is afterwards given by a discrete set of solutions. It may contain solutions outside the region supported by data (Section II-C). Therefore, the level set has to be confined to this region to form the final set of feasible methods.

The final level set is then a list of process parameter vectors. Each of them will produce the result $\vec{g}_{\vec{t}}$ as requested by the task:

$$\{\, \vec{p}_k \,|\, \vec{g}_{\vec{t}} = \vec{g}(\vec{p}_k, \vec{c}_{\vec{t}}) \,\}\,, \quad k = 1, 2, \ldots \tag{10}$$

In the special case of a quadratic polynomial kernel

$$K(\vec{p}, \vec{q}) = \left(\gamma\langle\vec{p}, \vec{q}\rangle + c\right)^2, \tag{11}$$

the solution can be found analytically by direct calculation

$$\begin{aligned} g_t &= g(\vec{x}, \vec{\alpha}), & (12)\\ x_k &= f\left(g_t, \vec{\alpha}, (x_1, x_2, x_{k-1}, x_{k+1}, \ldots, x_n)\right). & (13) \end{aligned}$$

Every solution in the found level set is associated with some cost such as energy, wear of tools, production cycle time and so on. To select the most efficient process method from the level set, one should be able to define a cost function depending on process parameters $\vec{p}$ and process conditions $\vec{c}$.

If the resulting level set is given as a discrete set of only a few hundred or thousand points, and if the computational effort to calculate the cost function is low, it is sufficient to do a complete search.

## III. Application to LSW

The main goal of this work is to demonstrate the applicability of T2MT to real industrial processes. LSW was chosen as a sample process because the setup of a specific laser welding machine or welding task is a very time consuming procedure.

A software library incorporating all the methods of Section II was developed. To fulfil additional functional requirements which emerged during application and testing, extensions to this original system were developed and implemented. In the following part of the paper, these extensions and the results of the verification procedure are presented.

### *A. Introduction to LSW*

In order to weld work pieces by laser, the work pieces have to be held in fixed positions. For that purpose, a laser welding cell is equipped with complex jigs composed of many pneumatic cylinders, limit switches, proximity switches, mechanical stops and grippers. Such jigs are usually mounted on a turn table which moves the fixed work pieces into a completely enclosed welding cabin. Inside this cabin, one or more robots are equipped with laser welding heads. These are optical devices with fixed or adjustable focal length. An optical fibre guides the laser light from the laser device to the welding head.

To make a seam, the laser light has to be focused on the work piece. The focus point has to be moved along the target line, it can be exactly on the surface of the work piece, some millimetres above, or inside or below the work piece. The corresponding parameter, called defocus, can be used to control the ratio between welding seam width and the penetration depth. Laser power is in the range of up to 6000 W. One of the most important advantages of LSW is the distance of the welding head to the work piece. In the presented examples, this distance (approximately equal to the focal length) is about 60 cm. Another benefit is the huge processing speed, the welding progress can be more than 200 mm/s.

The result of the welding process can be described by weld width and penetration depth (Figure 2). The customer usually wants to specify these values. Additionally, more quality constraints must to be satisfied: Undercut, root cavity, excess penetration, excess weld metal (Figure 3).

### *B. Two Goal Functions*

Penetration depth and weld width are two quantities, which define the goal values to be fulfilled by the process. Each quantity is modelled separately by SVR. In order to find the appropriate process parameters, one needs to search for the overlap of the two level sets for each goal value. One method to achieve this, is (1) to determine the level set from only one goal model and then (2) restrict this level set by the evaluation of the second goal model and force it to be equal to the second goal value.

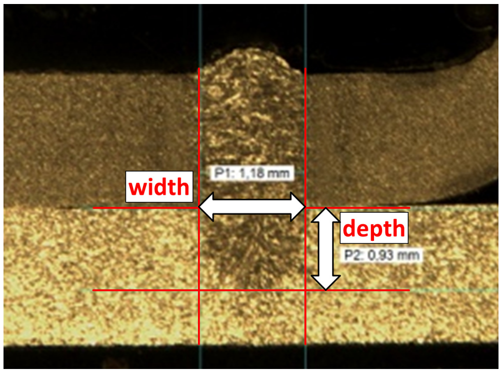

Figure 2. Micrograph showing the main quality quantities in LSW: penetration depth and weld width (with permission from AWL [7]).

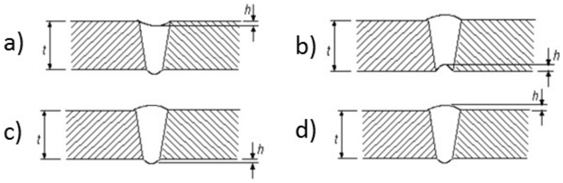

Figure 3. Additional quality measures: a) Undercut, b) Root concavity, c) Excess penetration and d) Excess weld metal.

If the level set of the first goal value is determined by the analytical method (11)-(13), the resulting process parameters will reproduce exactly the first goal value. But there are situations in which no single parameter set out of this level set will produce the second goal with acceptable accuracy. The user has to specify the allowed deviations in the goal values and the feasible resolution of the process parameters. Additionally, to accept small deviations it showed to be advantageous to run the determination of the level set twice, with changed roles of the first and second goal quantities.

Determination of the level set by the subdivision algorithm (Section II-D) does not suffer from this issue, because it is internally already working in a discretizied parameter and goal space.

### *C. Parameter Extraction Allowing Goal Ranges*

In LSW, it is not always appropriate to match both goal values of Penetration Depth and Weld Width exactly. A customer may require the penetration depth to be equal to the thickness of the lower sheet. Additionally, he may only set the requirement on the weld width to be greater than a given minimum value or to lie in a given range. For this, the goal range is disretisized according to the specified resolution of the goal value and a levelset is determined for each of these descrete levels. After that, the union of all found levelsets is build and repetitions of parameters are deleted.

### *D. Model Mapping and Model Calibration*

A laser welding cell is typically build up at the vendors facility, where also process parameters for good quality products are determined. Test sheets are welded with different process parameter settings, cross sections are cut and polished. Penetration depth, weld width and other parameters, which characterise the welding seam, are measured by micrography (Figure 2 and Figure 3). First products are produced, and if the customer is satisfied with the quality, the welding cell will be dismantled and rebuilt at the customers factory.

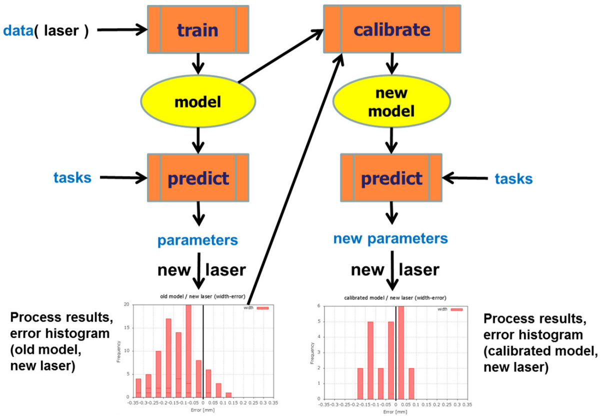
data( laser )
train
model
calibrate
new model
tasks
predict
predict
tasks
parameters
new parameters
new laser
new laser
Process results, error histogram (old model, new laser)
old model / new laser (width-error)
calibrated model / new laser (width-error)
Process results, error histogram (calibrated model, new laser)
Frequency
Error [mm]

Figure 4. Model mapping or model calibration.

When the first products are produced at the customer, it is not unusual that the results are slightly different to the previously results at the vendor. Something small has changed in the whole setup, which has an influence on the results but the cause might not be obvious. Because the physics of the process has not changed, it is a good assumption that a process model developed by the vendor is still correct. Then only a small affine transformation of the combined process parameter and goal quantity space will shift the process model, so that it now captures the new situation with sufficient accuracy. The new measurements at the customer are used to calibrate the process model (Figure 4).

A new task may require to weld a material combination which is similar to an other material combination with an already existing process model. It can be assumed that the physics does not behave very different and therefore, the existing model can be used as a basis for the new welding task. Because the qualitative behaviour is already modelled, only a small number of additional experiments have to be done to capture deviations. The original model can then be transformed into a new model for the new task by the same procedures as in the case of calibration. In this way, an existing process model is mapped to a new process model for a new task.

Both procedures, model calibration and model mapping, are algorithmic equivalents.

### *E. Parameter Adjustment*

The previous chapter dealt with the calibration of a process model to slightly different boundary conditions. The idea was to create an adjusted process model, which again is capable to describe the whole process space.

But sometimes it is enough to just find better process parameters for a given task. Again, under the assumption of similar physical behaviour, an existing process model can be used to calculate gradients in parameter space which yield better goal values. The process model does not have to be very precise in an absolute sense, but it should exhibit the same qualitative behaviour.

### *F. Experimental Verification*

In the I-RAMP[3] project (see Section ACKNOWLEDGEMENT), the changed conditions (after dismantling and rebuilding a laser welding cell) were simulated by the exchange of the laser source and the optical fibre connecting the source with the laser head mounted on the robot. Theoretically, there should be no change in the process results, if the laser source and the fibre are exact replacements. But it is nearly sure that these components are a bit different, e.g., the source is build by an other vendor and produces slightly different laser output power.

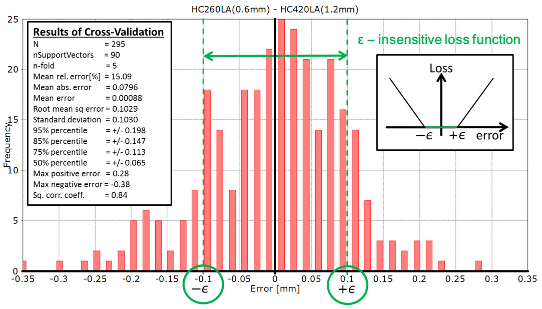

Figure 5. 5-fold cross validation error histogram of process model.

In the following, the steps performed to demonstrate the usability and applicability of T2MT to LSW are described in details, please refer to Figure 4. All experiments were done by AWL. [7]

*1) Experimental Sampling of Process Space:* Experiments were done on three different material and thickness combinations:

- HC260LA-0.6mm on HC420LA-1.2mm,
- HC420LA-1.2mm on HC380LA-1.5mm, and
- DC04-1.5mm on HC380LA-1.5mm.

The process parameter space was sampled on a regular grid in following ranges:

- laser power: 3500 W to 5500 W, in steps of 500 W,
- focus: -20 mm to +20 mm, in steps of 5 mm and
- weld velocity: 80 mm/s to 220 mm/s, in steps of 10 mm/s.

In the case of DC04 and low laser power, the lower limit of weld velocity was reduced to 30 mm/s.

All in all 1485 experiments were made. Each welding seam was cut, sanded and measured by micrography. The measured quantities were penetration depth, weld width, undercut, root concavity, excessive penetration and excess weld metal (Figure 2 and Figure 3).

*2) Generation of Process Models:* Based on these data, process models for penetration depth and weld width were calculated using SVR (Sections II-B and III-B). The boundary of the space supported by the data was modelled by SVRM (Section II-C). All calculations regarding support vectors are based on the library libsvm [8].

Only data which produced good quality was used to build up the process models. The conditions to specify good quality were set to penetration depth $\geq 0.1$ mm, weld width $\geq 0.5$ mm, undercut $\leq 0.25 \times$ 'upper sheet thickness', and root cavity $\leq 0.25 \times$ 'lower sheet thickness'.

An example of the error distribution of such a model is shown in Figure 5. The Figure shows the histogram for the first material combination (HC260LA-0.6mm on HC420LA-1.2mm). It is the overlay of 5-fold cross validation. The inlay on the left side shows some statistical quantities, e.g., 75% of the errors are in the range ±0.113 mm. The inlay on the right side shows the $\epsilon$-insensitive loss function used in the SVR algorithm to weight the errors. In the case shown, $\epsilon$ was set to 0.1 mm. Errors in the range of $\pm\epsilon$ are weighted by zero, they have no influence on the optimization algorithm used to determine the regression coefficients. Errors outside this range are weighted linearly. These two facts are responsible for the very robust behaviour of SVR with respect to outliers.

*3) Parameter Prediction for Sample Tasks (Original Model):* New welding tasks were specified by selection of a material combination, specification of demanded values of penetration depth and weld width and in some cases also specification of one of the process parameters laser power, laser focus, or weld velocity. The remaining process parameters were determined by calculation of the level set for the given goals of penetration depth and weld width.

In order to select one parameter set out of the level set, a cost function was defined which prefers smaller cycle time (faster speed):

$$\text{cost} = \sqrt{(d-d_0)^2 + (w-w_0)^2 + (v/100)^2},$$

where $d$ = penetration depth, $D_0$ = demanded penetration depth, $w$ = weld width, $w_0$ = demanded weld width and $v$ = weld velocity.

*4) Measurements with a new Laser:* Changed production conditions were simulated by exchange of the laser by an other laser made by an other vendor. Also, the fibre connecting the laser source with the laser head in the robot was exchanged.

The parameters predicted in the previous step were used to perform weld processes. Again, all produced welding seams were cut, sanded and measured by micrography. All measured penetration depths and weld widths were found to be smaller than requested.

As a cross-check, some additional measurements were made with process parameters taken from the original experiments, from which the process models were created. Also in these cases, the results were too small.

The deviations produced by the new laser with respect to the original process models on penetration depth was -0.16±0.16 mm, the deviation on weld width was -0.13±0.10 mm.

*5) Calibration of the Process Models:* The process models were calibrated using the data of the previous step, where all results are out of demanded ranges. Only 37 experiments were used to calibrate the process models, whereas the original models was created by 295 (good quality) experiments out of about 490. It can be expected that the number of required calibration experiments can be further reduced by application of intelligent sampling algorithms.

*6) Parameter Prediction for Sample Tasks (Calibrated Model):* Based on the calibrated model again new tasks were specified and corresponding process parameters are determined in the same manner as in Subsection III-F3.

*7) Verification of the Predictions by new Laser:* Experiments with the new parameters were executed and evaluated. The deviations produced by the new laser with respect to the calibrated process models on penetration depth was +0.05±0.11 mm, the deviation on weld width was -0.05±0.07 mm. It can be stated that these results are a good improvement compared to the original model. This improvement was reached by only few additional experiments with the new laser. It must be kept in mind, that the evaluation of each experiment is very time consuming because it involves cutting, sanding and micrography.

## IV. Conclusion

In [1], a concept (called T2MT) was presented for the automatic extraction and representation of process knowledge from experimental data. It was used to derive process parameters to reach a given goal under given process conditions. The concept was demonstrated in that paper by numerical simulations on resistance spot welding.

In Section II of the current paper, a short review of T2MT is given. Section III applies the methods to LSW and describes additional extensions, which converted the T2MT into a system usable in industry.

The whole concept was now demonstrated to be ready to be applied in industrial environments by experimental verification with real data, sampled from the LSW process. The focus was to demonstrate the advantages by finding good process parameters using T2MT with highly reduced time effort. This time-saving aspect becomes more and more impressive if more process models are available. Data should be gathered from the setup of new machines and from processing of new tasks and should be stored in a database. Process models derived from this database are candidates for the calibration to slightly different tasks, they are the starting point for the generation of new models.

It is worthwhile here to mention the flexibility and portability of the T2MT. The whole framework makes no assumption about the underlying processes, it is exclusively driven by experimental data. The T2MT can also be integrated into machines and perform the automatic parameter finding on-line. In this case, the user needs to describe the demanded task in terms of goal values and process boundary conditions, e.g., materials and sheet thicknesses. The process parameters are determined automatically in this task-driven operation.

## Acknowledgment

This work was supported by the EU Project I-RAMP$^3$ (Intelligent Network Devices for fast Ramp-up), project homepage `http://www.i-ramp3.eu/`. I-RAMP$^3$ is co-financed by the European Commision DG Research under the 7th Framework Programme.

## References

[1] J. Pollak, A. Sarveniazi, and N. Link, "Retrieval of process methods from task descriptions and generalized data representations," The International Journal of Advanced Manufacturing Technology, vol. 53, no. 5-8, pp. 829–840, 2011. [Online]. Available: http://dx.doi.org/10.1007/s00170-010-2874-1 [retrieved: July, 2015]

[2] V. Vapnik, The Nature of Statistical Learning Theory. Berlin: Springer-Verlag, 1995.

[3] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," Statistics and Computing, vol. 14, pp. 199–222, 2004.

[4] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the Support of a High-Dimensional Distribution," Neural Computation, vol. 13, pp. 1443–1471, 2001.

[5] C. Yuan and D. Casasent, "Support vector machines for class representation and discrimination," in Proceedings of the International Joint Conference on Neural Networks, vol. 2, July 20-24, 2003, pp. 1611–1616, DOI 10.1109/IJCNN.2003.1223940.

[6] J.-C. Wang and D. Casasent, "Hierarchical K-means Clustering Using New Support Vector Machines for Multi-class Classification," in International Joint Conference in Neural Networks, JUL 16-21, 2006, pp. 3457–3464.

[7] AWL-Techniek B.V., Nobelstraat 37, NL-3846 CE Harderwijk, (postal address: P.O. Box 245, NL-3840 AE Harderwijk), The Netherlands, Web: http://www.awl.nl [retrieved: July, 2015].

[8] C.-C. Chang and C.-J. Lin, LIBSVM: a library for support vector machines, 2001, software available at http://www.csie.ntu.edu.tw/˜cjlin/libsvm [retrieved: July, 2015].