IDENTIFICATION AND PREDICTIVE CONTROL OF LASER BEAM WELDING USING NEURAL NETWORKS

A. Bollig [*], D. Abel [*], Ch. Kratzsch [†], S. Kaierle [†]

[*] Institute of Automatic Control
Aachen University, 52056 Aachen, Germany
Phone: ++49-241-8027480, Fax: ++49-241-8022296
E-Mail: A.Bollig@irt.rwth-aachen.de
[†] Chair of Laser Technology
Aachen University, 52056 Aachen, Germany

Keywords: Neural networks, Predictive control, System identification, Linearization, Laser beam welding

Abstract

Welding with laser beams is an innovative technique, which leads to higher penetration depth and a narrower seam compared to conventional welding techniques. One significant criterion of the quality of a junction is the penetration depth. Within this article a predictive control scheme is presented that optimises the process' input laser power by taking the future welding speed into account. For modelling the non-linear process an Artificial Neural Network (*ANN*) is applied. The GPC-algorithm with a linear model obtained by instantaneous linearization of the network is used. For this reason, an extended training of the ANN is introduced. First results of the application on a real laser welding system are described.

1 Introduction

Welding with laser beams is characterized by high specific power input into the workpiece by focussing the laser beam in the vicinity of the workpiece surface. Thereby the material is not only molten, but also vaporized forming a capillary with plasma in it. The result is a narrow seam with a higher penetration depth compared to conventional welding techniques as the laser beam can penetrate the material through the capillary easily. As a result, a minimum of thermal stress of the workpiece can be maintained. From a control engineer's point of view laser beam welding is a complex multi-dimensional problem. Besides process parameters (e.g. laser power, welding speed, position of the focus, angle of incidence in 3D material processing, inert gas) also geometry parameters of the joint (e.g. material thickness, gap, surface properties) influence the seam quality.

Simple models of the process can be obtained from energy balances and approximate the static behaviour of the process. A detailed mathematical description of the process is extremely difficult as it has to take into account the sub-processes radiation absorption and multiple reflections, heat conduction, melting, hydrodynamics, evaporation and optical emission of plasma [14]. Analytical modelling of laser beam welding is a field of intensive research, but so far no self-contained model of the process covering all necessary details is found. As the dominating system input variables are welding speed and laser power, these are suited for controlling the penetration depth. In 3D laser beam welding the welding speed depends on the path planning, the machine dynamics and perhaps a seam tracking system and can therefore not be used for controlling purposes. The laser power, however, is instantaneously alterable; therefore it can be used as the manipulated variable. Further input variables mentioned above can be kept constant. As the future welding speed is known from the path planning, the use of a predictive controller which takes future variations of the welding speed into account seems appropriate.

As the penetration depth cannot be measured without destroying the workpiece, the intensity of the plasma emission is used as the controlled variable instead. Temporally and spatially resolved observations of the keyhole are acquired with a CCD-camera and interpreted using physical insight [7],[14]. The camera is mounted directly on the welding head and its optical path goes coaxial with the laser beam path through the focussing optic. The design of the system allows an easy adaption to both CO_2 and Nd:YAG laser optics.

In this article, a predictive controller taking the welding speed into account is described [1]. As the fast, complex, non-linear process cannot be analytically described in a suitable way for MPC, experimental system identification is carried out to obtain a model for the controller.

2 Identification

The reduced data-driven model of the process has the two input variables laser beam power p and welding speed v. Its output variable is the (measurable) intensity of the plasma emission I. As the specific energy applied to the process is p/v, the process is expected to show non-linear behaviour. Identification of two linear *SISO*-models from p to I and v to I, respectively, will not result in a suitable model for the controller. To cover the whole operating range, a global non-linear model is needed. During different experiments measurements of the laser beam power p, the welding speed v and the filtered intensity of the plasma emission I were collected.

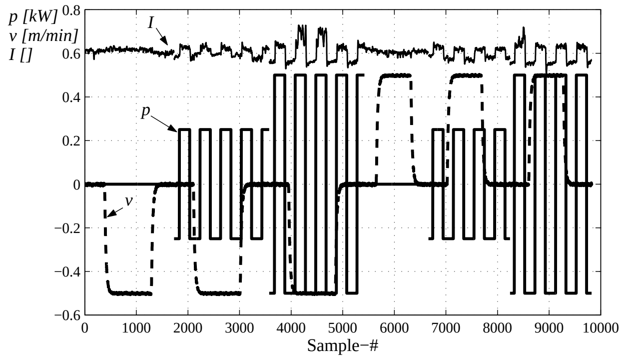

Figure 1: Process behaviour

Figure (1) shows a composition of different measurements of the dynamic process behaviour as deviations from an operating point. Using a calibration curve, the measured intensity can be mapped to the penetration depth to describe the quasi-steady penetration depth with time-averaged measurements.

Because of their approximation capabilities Artificial Neural Networks are well suited for system identification when a black-box model of a non-linear dynamic process is to be derived from measured data. In 1990, Narenda and Parathasarathy proposed the use of neural networks for system identification and control [8]. Nowadays, neural networks are established as approximators in many fields. A widely-used type of network is the Multi-Layer Perceptron network (*MLP*) consisting of one hidden layer of neurons with non-linear activation functions and one output layer with linear neurons. For the hidden layer neurons with the hyperbolic tangent activation function are frequently used. Non-linear time discrete dynamic systems can often be represented by non-linear difference equations. Therefore a tapped delay line for nb by at least d samples delayed system input variables and na by at least one sample delayed output variables is used. These so-called *NARX*-structures (Non-linear AutoRegressive, eXogenous input) are intuitively deduced from linear system identification [15], [9] and with neural networks often used to represent non-linear dynamic systems besides more complex structures with internal dynamics. The resulting neural network has the structure depicted in figure (2). Depending on which output signal is fed back, one distinguishes between the serial-parallel (*NARX*) and the parallel (*NOE*) structure. For predicting more than one step ahead, the NARX-structure cannot be used, and so for use in the controller the NOE-model is mainly applied. Nevertheless the NARX-structure can be used during the identification.

The mapping of the network at discrete time t can be described by

$$\hat{I}(t) = \sum_{j=1}^{S_1} w_{1j}^2 \cdot \tanh\left(\sum_{i=1}^{na+2nb} w_{ji}^1 \varphi_i(t) + b_j^1\right) + b_1^2 \qquad (1)$$

where the network inputs for the prediction of time instant $t+k$

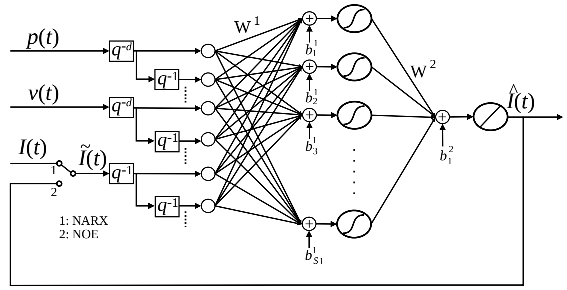

Figure 2: MLP-network with external dynamics

are given as a regression vector

$$\begin{aligned} \varphi(t+k) \; = \; [\; & p(t+k-d) \quad \dots \\ & p(t+k-d-nb+1) \\ & v(t+k-d) \quad \dots \\ & v(t+k-d-nb+1) \\ & \tilde{I}(t+k-1) \quad \dots \\ & \tilde{I}(t+k-na) \quad]^T \end{aligned} \qquad (2)$$

in which the signals $\tilde{I}$ are measurements for time arguments negative with respect to t and predictions otherwise. The weight w_{ij}^k denotes the connection of the ith neuron in the kth layer to its jth input.

Using neural networks in this framework means that two problems have to be dealt with: finding the right regression vector for the mapping (which means determining the system order) and finding a sufficient number of hidden neurons and determine optimal weights. The first problem is also known in linear system identification theory, the latter is a typical neural network training task. Starting with a considerably higher system order than the applied filter has, time series of p, v and I were arranged as regressors and presented to the network as N training patterns. Using a Levenberg-Marquardt algorithm to adjust the network parameters, the one-step-ahead-prediction error is minimized (also referred to as *Prediction Error Method* and corresponds to figure (2), NARX).

A number of $S_1 = 8$ neurons in the hidden layer gave good results during the training. To obtain a parsimonious model which does not tend to overfitting, the last delay terms were successively pruned and the network was retrained until a remaining system order of $na = 2$, $nb = 3$ and $d = 1$ for both system inputs was left. Further pruning of network inputs decreased considerably the network performance. For further information about structure selection see e.g. [10] and [6]. If the neural network model is to be used as a multi-step-ahead-predictor, like in the case of a predictive controller, some additional properties of the model besides input- to output-mapping might be interesting. If one e.g. can claim the process to be identified to be stable, or physical insight allows to specify a range for the "local" gains of the process, also the model should show these properties. In the following it is shown how this can be taken into account during the training as a NARX-structure.

As for nonlinear systems properties like gain, damping and even stability might vary with respect to the operating point, a local linearization allows an interpretation of the model. By Taylor series expansion at time τ one obtains as a linear approximation of eq. (1)

$$\hat{I}(t) \approx \hat{I}(\tau) + \left.\frac{\partial \hat{I}(t)}{\partial \varphi_1(t)}\right|_{t=\tau} (\varphi_1(t) - \varphi_1(\tau)) + \left.\frac{\partial \hat{I}(t)}{\partial \varphi_2(t)}\right|_{t=\tau} (\varphi_2(t) - \varphi_2(\tau)) + \dots \quad (3)$$

Using $o_j^1(t)$ to denote the outputs of the hidden layer, the partial derivatives

$$\frac{\partial \hat{I}(t)}{\partial \varphi_k(t)} = \sum_{j=1}^{S_1} w_{1j}^2 \cdot \left(1 - \left(o_j^1(t)\right)^2\right) \cdot w_{jk}^1 \quad (4)$$

are easily calculated from the non-linear prediction at time instant $t = \tau$ and one obtains (cp. eq. (2))

$$b_i = \frac{\partial \hat{I}(t)}{\partial p(t-d-i)}, \quad i = 0 \dots nb-1, \quad (5)$$

$$c_i = \frac{\partial \hat{I}(t)}{\partial v(t-d-i)}, \quad i = 0 \dots nb-1 \quad (6)$$

and

$$a_i = -\frac{\partial \hat{I}(t)}{\partial \tilde{I}(t-i)}, \quad i = 1 \dots na \quad (7)$$

as coefficients for a linear ARX model and thereby

$$\hat{I}(t) \approx -\sum_{i=1}^{na} a_i \tilde{I}(t-i) + \sum_{i=0}^{nb-1} b_i p(t-d-i) + \sum_{i=0}^{nb-1} c_i v(t-d-i) + e(\tau) \quad (8)$$

approximates eq. (1) with

$$e(\tau) = \hat{I}(\tau) + \sum_{i=1}^{na} a_i \tilde{I}(\tau-i) - \sum_{i=0}^{nb-1} b_i p(\tau-d-i) - \sum_{i=0}^{nb-1} c_i v(\tau-d-i) \quad (9)$$

being the difference between non-linear and linearized model at τ resulting from the bias values of the network.

Some properties like stability and damping can be read off the coefficients a_i, as they represent the denominator of the z-transfer function.

The Levenberg-Marquardt-Algorithm is often used during the identification to minimize the functional

$$V(\theta) = \frac{1}{2N} \sum_{t=1}^{N} \varepsilon(t,\theta)^2 \quad (10)$$

which includes the network prediction errors $\varepsilon(t,\theta) = I(t) - \hat{I}(t,\theta)$ of the respective data set t depending on the actual network weights and bias values θ.

To extend functional (10) and by this formulate further goals of the training, some additional conditions are introduced. This is illustrated by the demand for stable local models. For e.g. a second-order system two conditions have to be fulfilled by the denominator coefficients according to Schur-Cohn-Jury [5]

$$\alpha : \quad -1 < a_2 < 1 \quad (11)$$

$$\beta : \quad -1 - a_2 < a_1 < a_2 + 1 \quad . \quad (12)$$

From condition α one can derive a measure δ_α showing the violation of the condition:

$$a_2 > 1 : \quad \delta_\alpha = a_2 - 1 \quad , \quad (13)$$

$$a_2 < -1 : \quad \delta_\alpha = a_2 + 1 \quad . \quad (14)$$

Analogous a measure δ_β is found if the coefficients of the actual linearization violate condition β. Now the functional can be written in an extended form

$$V^+(\theta) = \frac{1}{2N} \sum_{t=1}^{N} \varepsilon(t,\theta)^2 + \delta_\alpha(t,\theta)^2 + \delta_\beta(t,\theta)^2 \quad (15)$$

where δ_α and δ_β only contribute to the functional if a linearization leads to a violation. The Levenberg-Marquardt-Algorithm is a combination of Gauss-Newton-Algorithm and gradient descent [10], where in eq. (15) the prediction error and any violation due to the actual choice of the parameters θ^i are used in a linearized form $\tilde{()}$.

$$\tilde{\varepsilon}(t,\theta) = \varepsilon(t,\theta^i) + \left[\frac{\mathrm{d}\varepsilon(t,\theta)}{\mathrm{d}\theta}|_{\theta^i}\right]^T (\theta - \theta^i) = \varepsilon(t,\theta^i) - \psi^T(t,\theta^i)(\theta - \theta^i)$$

$$\tilde{\delta_k}(t,\theta) = \delta_k(t,\theta^i) + \left[\frac{\mathrm{d}\delta_k(t,\theta)}{\mathrm{d}\theta}|_{\theta^i}\right]^T (\theta - \theta^i) = \delta_k(t,\theta^i) - \rho_k^T(t,\theta^i)(\theta - \theta^i) \quad (16)$$

For δ_α applies for instance

$$\rho_\alpha(t,\theta^i) = -\frac{\mathrm{d}\delta_\alpha(t,\theta)}{\mathrm{d}\theta}|_{\theta^i} = -\frac{\mathrm{d}a_2(t,\theta^i)}{\mathrm{d}\theta} \quad (17)$$

with (in this case set $k = 2$)

$$\frac{\partial a_k}{\partial b^2} = 0 \quad , \quad (18)$$

$$\frac{\partial a_k}{\partial w_{1,j}^2} = w_{j,nb+k}^1 \left[\left(o_j^1\right)^2 - 1\right] \quad , \quad (19)$$

$$\frac{\partial a_k}{\partial b_j^1} = 2 w_{1,j}^2 w_{j,nb+k}^1 o_j^1 \left(1 - \left(o_j^1\right)^2\right) \quad , \quad (20)$$

$$\frac{\partial a_k}{\partial w_{j,i}^1} = \frac{\partial a_k}{\partial b_j^1} \varphi_i \quad k \neq nb + i \quad , \quad (21)$$

$$\frac{\partial a_k}{\partial w_{j,i}^1} = w_{1,j}^2 \left[\left(o_j^1\right)^2 - 1\right] + \frac{\partial a_k}{\partial b_j^1} \varphi_i \quad k = nb + i \quad . \quad (22)$$

as contributions to ρ_α. The respective argument (t, θ^i) indicating data set t in training iteration i has dropped for the sake of simplicity.

By using $\tilde{\varepsilon}$, $\tilde{\delta_\alpha}$ and $\tilde{\delta_\beta}$ according to eq.(16) now the functional $\tilde{V}^+$ is used to approximate eq.(15). It can easily be differentiated with respect to θ:

$$\begin{aligned} G(\theta^i) &= \frac{\mathrm{d}\tilde{V}^+}{\mathrm{d}\theta}\,|_{\theta^i} \\ &= -\frac{1}{N}\sum_{t=1}^{N}\psi(t,\theta^i)\varepsilon(t,\theta^i) \\ &+ \rho_\alpha(t,\theta^i)\delta_\alpha(t,\theta^i) + \rho_\beta(t,\theta^i)\delta_\beta(t,\theta^i) \qquad (23) \\ H(\theta^i) &= \frac{\mathrm{d}^2\tilde{V}^+}{\mathrm{d}\theta^2}\,|_{\theta^i} \\ &= \frac{1}{N}\sum_{t=1}^{N}\psi(t,\theta^i)\psi(t,\theta^i)^T + \rho_\alpha(t,\theta^i)\rho_\alpha(t,\theta^i)^T \\ &+ \rho_\beta(t,\theta^i)\rho_\beta(t,\theta^i)^T \quad . \qquad (24) \end{aligned}$$

By this a modified gradient $G(\theta^i)$ and Hessian $H(\theta^i)$ for the Levenberg-Marquardt-algorithm are found, that allow by minimizing eq.(15) to find a similar input-to-output-mapping taking the desired properties into account. Clearly this method cannot supersede e.g. a stability proof for the global NOE-structure in the Lyapunov sense as it only consideres the linearized models of the training data, but it revealed to be a quite convenient enhancement for the identification.

3 Predictive Control

Having an internal controller's model capable of modelling non-linear dynamic processes allows on one hand a wider operating range of the controller, on the other hand a demanding non-linear, perhaps constrained optimization problem has to be solved at every time instant. In e.g. [16] non-linear optimization is used with a neural network model applying a quasi-Newton algorithm, in [2], also feedback linearization is used to ease the optimization problem. In e.g. [17], neural networks were also applied to learn the optimal solution instead of computing it analytically. As the computational burden is still high for those approaches and the time behaviour might even be non-deterministic, a quite intuitive way, described in [10] as approximate predictive control (*APC*) is followed for this application as short sample times of 2 ms are required due to the fast process dynamics. The approximate predictive controller uses a linear predictor. By instantaneous linearization at each sample step the good approximation capabilities of the neural network and the deterministic behaviour of a linear optimization are then combined.

The static non-linear behaviour will be covered quite exactly by the linearized model, while depending on the degree of nonlinearity transients will only be approximated. Therefore, care has to be taken in trusting the 'optimal' solution that can only be valid for the linear model. Setting the weighting factor ρ of future control changes in eq. (33) to a relative high value ensures a smoother control action that is not that strongly dependent on the properties of a single linearized model during one sample step.

The coefficients of the linearized model can be written in polynomial form

$$\begin{aligned} A\left(q^{-1}\right) &= \left[1 + a_1 q^{-1} \ldots + a_{na} q^{-na}\right] \qquad (25) \\ B\left(q^{-1}\right) &= \left[b_0 + b_1 q^{-1} \ldots + b_{nb-1} q^{-nb+1}\right] \qquad (26) \\ C\left(q^{-1}\right) &= \left[c_0 + c_1 q^{-1} \ldots + c_{nb-1} q^{-nb+1}\right] \qquad (27) \end{aligned}$$

in the time shift operator q in order to use the GPC-algorithm [3, 4]. The GPC-algorithm exploits the linearity of the model to separate the prediction of the controlled variable into two parts. The first part, the so-called "free" response Φ, describes the estimated behaviour of the controlled variable when the manipulated variable remains constant. The second part represents the estimated future impact of a change in the manipulated variable. The prediction of the controlled variable for the time instants $t + d \ldots t + N_2$ can then be written as

$$\hat{I} = \Gamma\tilde{P} + \Phi(\tilde{V}) \quad , \qquad (28)$$

with

$$\begin{aligned} \hat{I} &= \left[\hat{I}(t+d)\,\hat{I}(t+d+1)\ldots\hat{I}(t+N_2)\right]^T \qquad (29) \\ \tilde{P} &= \left[\Delta p(t)\,\Delta p(t+1)\ldots\Delta p(t+N_u-1)\right]^T \qquad (30) \\ \tilde{V} &= \left[\Delta v(t)\,\Delta v(t+1)\ldots\Delta v(t+N_2-d)\right]^T \qquad (31) \\ \Phi &= \left[\varphi(t+d)\,\varphi(t+d+1)\ldots\varphi(t+N_2)\right]^T \qquad (32) \end{aligned}$$

For the presented case of laser beam welding the known future velocity can be used in the calculation of Φ as changing system input, because only the manipulated variable remains constant in Φ and is modelled by $\Gamma\tilde{P}$ separately.

To determine the cost of the future manipulated variable, a functional is set up which includes the estimated future control deviation $r(t+i) - \hat{I}(t+i)$ as well as the control efforts $\Delta p(t+i)$,

$$\begin{aligned} J(t,\tilde{P},\tilde{V},R) &= \sum_{i=N_1}^{N_2}\left[r(t+i) - \hat{I}(t+i)\right]^2 \\ &+ \rho\sum_{i=1}^{N_u}\left[\Delta p(t+i-1)\right]^2 \quad . \qquad (33) \end{aligned}$$

The weighting ρ can be used to affect the speed of the controller. Moderate settings lead to a smooth controller response and by this compensate poorly fitting linearized models while too high settings decrease the overall performance. Additionally, low pass filtering of the coefficients of the linearized models can attenuate the effects of measurement noise. By combining the future reference into the vector R and minimizing the functional the optimal solution

$$\tilde{P}_{\text{opt.}} = \left[\Gamma_1^T\Gamma_1 + \rho I_{N_u}\right]^{-1}\Gamma_1^T(R - \Phi) \qquad (34)$$

for the manipulated variable laser power is found.

If constraints are present in the minimization of eq.(33), a quadratic optimization problem has to be solved, and the solution according to eq.(34) might not be feasible. For a stable process like in the presented case, besides state-of-the-art QP-solvers the one-degree-of-freedom algorithm (*ONEDOF*) [12] can be applied to overcome this problem. It is outlined briefly in the following. The main idea consist of a combination of two control strategies: a guaranteed feasible, slowly converging one and a fast, unconstrained "optimal" one. The latter is represented in this case by eq.(34), while the first is chosen to be an open loop control strategy based on an inverse static mapping. As the identified network was trained with a constraint on the local laser power gain to be positive, a second, static ANN can be trained afterwards which performs a mapping

$$P_{\text{stat.}} = ANN(V_{\text{actual}}, I_{\text{reference}}) \quad . \tag{35}$$

By using an MLP with an output neuron with saturation, the feasibility of the constant solution $P_{\text{stat.}}$ is guaranteed, and since the process is stable, the controlled variable will tend towards the reference, allowing a steady control deviation caused by model offset and disturbances. Now the optimal control sequence $P_{\text{opt.}}$ in absolute values is given by

$$P_{\text{opt.}} = (1 + q^{-1})\tilde{P}_{\text{opt.}} \tag{36}$$

and a violation of the constraints $P_{\text{min.}}$, $P_{\text{max.}}$ of dimension $N_u \times 1$ on the controller output can be avoided by choosing

$$P_{\text{feas.}} = (1-\alpha)P_{\text{opt.}} + \alpha P_{\text{stat.}} \quad 0 \leq \alpha \leq 1 \tag{37}$$

with a minimal value of the single degree of freedom α. By introducing

$$V = \begin{bmatrix} P_{\text{stat.}} - P_{\text{opt.}} \\ -P_{\text{stat.}} + P_{\text{opt.}} \end{bmatrix} \tag{38}$$

and

$$W = \begin{bmatrix} P_{\text{max.}} - P_{\text{opt.}} \\ -P_{\text{min.}} + P_{\text{opt.}} \end{bmatrix} \tag{39}$$

the problem is reformulated as

$$\min(\alpha) \quad \text{s.t.} \quad \alpha V \leq W \tag{40}$$

which can easily be solved by initializing $\alpha = 0$ and setting

$$\alpha = \max\left(\alpha, \frac{W(i)}{V(i)}\right) \quad i = 1 \ldots 2N_u \tag{41}$$

which has to be evaluated only if $W(i) \leq 0$, i.e. a constraint violation occurs. Clearly if $\alpha < 1$ – which can be achieved with sensible settings for the controller – the integral action of the original GPC is maintained and a steady control deviation vanishes. Rate constraints and constraints on the controlled variable can be included in a similar way. For further reading see [12], [13].

With this approach for the predictive controller, the following scheme results for the closed control loop. The motion planning provides reference trajectories for the welding speed and the desired penetration depth. The latter is mapped to a desired intensity. The motion control loop of the welding system contains the drive dynamics and shows first order lagging behaviour. This is also considered by a prefilter to $\tilde{V}$ in the MPC. The MPC calculates the optimal laser power taking the future welding speed into account and passes $p_{\text{ref.}}$ as the first element of $P_{\text{feas.}}$ in eq.(37) as actual command value to the cascaded laser power control loop.

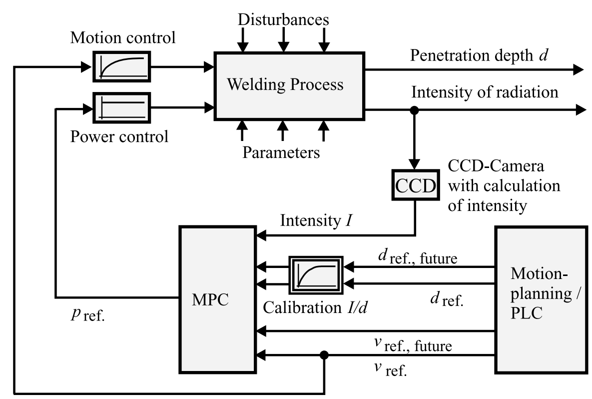

Figure 3: Predictive control scheme

For practical application, a conservatively tuned PI-controller is operated in parallel to the predictive controller and is switched on if the process is leaving the domain covered by the training data.

4 Application

The described predictive control scheme has been implemented on an open OSACA-based control environment [11]. A CCD-Camera [7] was used to observe the plasma. A PC-system is processing the images from the camera and calculating the intensity value corresponding to the actual penetration depth. This intensity value, the welding speed and the desired penetration depth is then used to perform the control task. Various experiments have been conducted to investigate the predictive controller's performance. As one example the compensation of a step-like change of the welding speed by 1.3 m/min is shown. Figure (4) shows the change of the measured intensity I, the welding speed v and the laser power p without controller. The longitudinal section of the weld shows the change in the penetration depth. Figure (5) shows the measured intensity, welding speed, laser power and a longitudinal section for an experiment with controller. Note that the penetration depth is almost constant. The fluctuations in the penetration depth and the measurements can be compared with the fluctuations in figure (4).

5 Summary

Welding with laser beams is an innovative technique for joining materials which experiences a strong increase in a wide variety

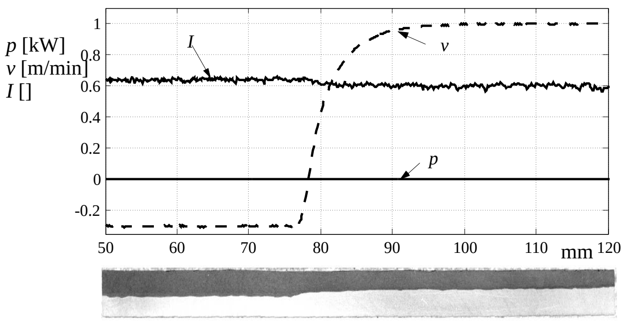

Figure 4: Plasma intensity and penetration depth for an uncontrolled weld with constant laser power and change in welding speed

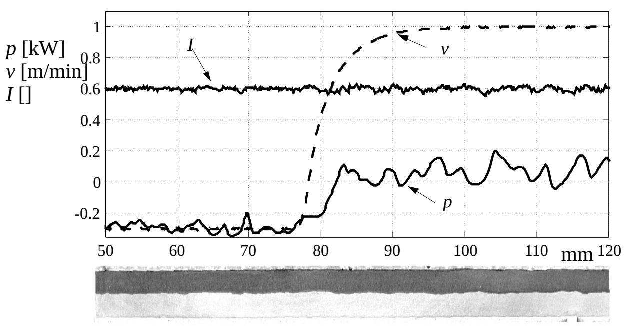

Figure 5: Plasma intensity and penetration depth for a controlled weld with change in welding speed

of industrial applications (e.g. manufacturing car bodies) because of high welding speed and the good quality of the seam. As it is in fact a complex process, means and ways for monitoring and closed loop control have to be applied. Within this article, a predictive control scheme using a dynamic neural network as process model is introduced. The laser power is used to affect the process, taking the welding speed into consideration. For monitoring purposes a CCD-camera is used, calculating the penetration depth from the intensity of the spatially and temporally resolved emission of the plasma. A NARX-MLP-network with external dynamics is used to identify the process and provide parameters for the linear predictor of the controller. Additional attention should be paid to the properties of the identified model, and an approach to extend the training during the identification is introduced. Thereby a non-linear neural network model and a well-known linear predictive control scheme are combined and used to control a very fast, non-linear process.

References

[1] A. Bollig, H. Rake, Ch. Kratzsch, S. Kaierle, "Application of neuro-predicitve control to laser beam welding", In: *Proceedings of the 15th IFAC world congress, 2002*

[2] M.A. Botto, J.S. da Costa, "A comparison of nonlinear predictive control techniques using neural network models", *Journal of Systems Architecture 44, 1998*

[3] D.W. Clarke, C. Mohtadi, P.S. Tuffs, "Generalized predictive control, Part I and II", *Automatica 23(2), 1987*

[4] D.W. Clarke, C. Mohtadi, P.S. Tuffs, "Properties of generalized predictive control", *Automatica 25(5), 1989*

[5] O. Föllinger, *Lineare Abtastsysteme, Oldenbourg Verlag, 1974*

[6] J.B. Gomm, S.K. Doherty, D. Williams, "Experiment design considerations for nonlinear system identification using neural networks", *Computers and chemical engineering, 21(3), 1997*

[7] Ch. Kratzsch, P. Abels, S. Kaierle, R. Poprawe, W. Schulz, "Online process control for laser beam materials processing", *Proceedings of the ISATA2000, Epsom, England, 2000*

[8] K. S. Narenda, K. Parathasarathy, "Identification and control of dynamical systems using neural networks", *IEEE Transactions on Neural Networks, Vol. 1, 1990*

[9] O. Nelles, *Nonlinear System Identification, Springer, Berlin, 2001*

[10] M. Nørgaard, O. Ravn, N.K. Poulen, L.K. Hansen, *Neural Networks for Modelling and Control of Dynamic Systems, Springer, London, 2000*

[11] J. Ortmann, S. Kaierle, E.W. Kreutz, R. Poprawe, A. Kahmen, M. Weck, "Open architecture control for laser materials processing." In: *Proceedings of the Photonics. Boston, 2001*

[12] J. A. Rossiter, M.J. Rice, J. Schuurmanns and B. Kouvaritakis, "A computationally efficient constrained predictive control law." In: *Proceedings of the ACC, 1998*

[13] J. A. Rossiter, "Extending the stability guarantees for an efficient predictive control law." In: *Proceedings of the ISSC. Ireland, 2001*

[14] W. Schulz, J. Michel, M. Niessen, V. Kostrykin, P. Abels, S. Kaierle, "Modelling, dynamical simulation and on-line monitoring in laser beam welding", *Proceedings of the LASER2001, München, 2001*

[15] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P.-Y. Glorennec, H. Hjalmarsson, A. Juditsky, "Nonlinear black-box modelling in system identification: a unified overview.", *Automatica 31(12), 1995*

[16] P.H. Sørensen, M. Nørgaard, O. Ravn, N.K. Poulsen, "Implementation of neural network based non-linear predicitve control", *Neurocomputing 28, 1999*

[17] L.-X. Wang, F. Wan (2001), "Structured neural networks for constrained model predictive control." *Automatica 37, 1235–1243, 2001*