2018 26th Mediterranean Conference on Control and Automation (MED)
Zadar, Croatia, June 19-22, 2018

System Identification of a CNC machining center with Support Vector Machines

Muzaffer Ay[1], Sebastian Stemmler[1], Dirk Abel[1], Max Schwenzer[2] and Fritz Klocke[2]

Abstract— **Support Vector Machines (SVM) is a machine learning algorithm with inherent generalization ability and a convex optimization problem. This paper studies the application of the SVM method for the online identification of the nonlinear dynamic behavior of the feed velocity in a CNC machining center. Both blackbox and greybox modeling approaches are tested for this purpose. Within the German Cluster of Excellence "Integrative Production Technology for High-Wage Countries", a modelbased predictive control (MPC) strategy with a linear state-space model is already implemented for the feed velocity of the CNC machining center. Due to nonlinearities, the model of the controlled system has to be identified and updated during the process. Therefore, the SVM method should be used to recurrently identify a model in every time-step. Additionally, the identified models should be capable of being formulated in a linear state-space model. The methodology is validated with measured data from the CNC machining center. The gained results for the blackbox and greybox approaches show only small deviations from the measured behavior of the system.**

I. INTRODUCTION

The German Cluster of Excellence "Integrative Production for High-Wage Countries" aims at establishing a modelbased self-optimizing approach, which allows to control quality measures and higher order objectives [6]. One of the application fields within the cluster is the milling. For milling, one of the objectives is to minimize the processing time without exceeding the maximum endurable force for the working tool. For this purpose, a linear modelbased predictive control (MPC) strategy was first formulated in [4]. Since the first formulation, the MPC has been further developed. The current formulation of the MPC for the modelbased self-optimization of the milling process is described in [16]. For the linear MPC strategy, a linear model of the controlled system is explicitly used to generate the system input. The MPC approach is a very popular advanced control technique, partly due to its ability to optimize given objectives against the desired system behavior. Further, the MPC is able to fulfill possible constraints for the controller and the controlled process [1].

But in some cases, the behavior of the system changes during the process. Due to the changed behavior, the model within the MPC becomes invalid. A possible reason for the changing behavior can be the change of the working point, the change of the constraints or nonlinearities within the system. A possibility to avoid the invalidity of the model within the MPC is to identify the system behavior during the process and update the existing model.

There is a substantial amount of applicable methods for the online system identification during the process [12]. Methods of gaining popularity in recent years are machine learning methods. Machine learning methods are often applicable to both linear and nonlinear systems without much knowledge about the system [9]. A machine learning method with many favorable characteristics is the "Support Vector Machines" (SVM) method. Compared to otherwise highly researched and used method of "Neural Networks" (NN), SVM has a higher inherent generalization ability and provides a convex optimization problem [8].

In [13] the SVM method is used for the offline identification of a model, which would be applicable in a linear MPC strategy. However, the offline identification does not consider a sliding horizon. The identification with a sliding horizon requires the recurrent identification of the system in every time-step with changing training data. In [11] the "Least-Squares SVM" (LS-SVM), a variation of the SVM method, is used for the recurrent identification of a nonlinear model for a nonlinear MPC. Further, both of the works use a blackbox modeling approach. The blackbox modeling only utilizes data to identify a model, without a physical description of the system. With the greybox approach, a known model structure and known parameters can be preset before the unknown parameters of the model are identified through data. A greybox modeling approach with LS-SVM is followed in [2], although the used LS-SVM method is only formulated for systems, where the parameters of the system are directly measurable. However in practice, the parameters are in general not directly measurable. In [3] the same greybox approach is combined with genetic algorithms (GA) and expanded for systems with unmeasurable parameters.

This paper studies the applicability of the SVM method for the online identification of a blackbox and a greybox model for the existing MPC strategy.

II. SUPPORT VECTOR MACHINES

The SVM method was originally introduced for pattern recognition by Vapnik and Lerner in [18]. SVM is a machine learning algorithm from the category of "supervised learning" methods. For supervised learning, exemplary data is used to mirror the system behavior and to learn a mathematical description of the system. Vapnik was also the first to describe the SVM method for regression problems [19]. For regression, the function to identify is formulated as

[1]Institute of Automatic Control, RWTH Aachen University, Germany, 52056 Aachen M.Ay@irt.rwth-aachen.de
[2]Laboratory for Machine Tools and Production Engineering, RWTH Aachen University, Germany, 52056 Aachen M.Schwenzer@wzl.rwth-aachen.de

$$y_k = w^{\mathrm{T}} \cdot \Phi(x_k) + \rho, \tag{1}$$

where y_k is the output at time-step k, x_k the input at time-step k, w the weight vector and ρ the bias. With the function Φ the input x is mapped onto higher dimensions to provide a linear formulation of the regression function on the mapped dimensions. In the literature, a large number of possible definitions can be found for such a function [9]. There is no distinct function for a specific problem. For each use case, the appropriate function has to be found through exploration. For the case, that the regression problem is already expressible with a linear function in the original input space, Φ should be defined as

$$\Phi(x_k) = x_k. \tag{2}$$

To describe a function as in (1), w and ρ has to be identified first. The SVM method determines this unknowns through a Lagrange optimization with the help of training data. The optimization problem for this purpose is described as

$$\min_{w,\rho,\xi_k,\xi_k^*} J = \frac{1}{2} w^{\mathrm{T}} \cdot w + C \sum_{k=1}^{N} (\xi_k + \xi_k^*) \tag{3}$$

s.t.

$$y_k - w^{\mathrm{T}} \cdot \Phi(x_k) - \rho \leq \varepsilon + \xi_k, \tag{4}$$

$$w^{\mathrm{T}} \cdot \Phi(x_k) + \rho - y_k \leq \varepsilon + \xi_k^*, \tag{5}$$

$$\xi_k, \xi_k^* \geq 0. \tag{6}$$

The optimization problem penalizes the absolute value of w and the deviation between the trained model and the training data with the slack variables ξ_k and ξ_k^* for every time-step k at N time-steps. The parameter C serves as a weight for ξ_k and ξ_k^* in the optimization problem. The optimization drives two contrary objectives. First, the minimization of the absolute value of w. This has the purpose to obtain a smoother regression function, since the smoothness of the identified function provides a more generalizable model [9]. Further, a more generalizable model is more applicable to untrained data. Second, the minimization of the deviation between the regression model and the training data. Hence, the model describes the data more accurately. Through the factor C, the two contradictory objectives can be prioritized against each other. For feasibility and more robustness, a soft margin is defined with $\varepsilon > 0$, inside of which deviations between model and data are allowed [13].

For an exemplary case, Fig. 1 shows the regression problem as it is described with SVM. The training data is visualized through circles on an one-dimensional input- and an one-dimensional output-space. The regression problem has to find a function like the one plotted solid, to describe the data within a ε margin between the dotted plots. In the illustrated scenario, the data points at time-step $\tilde{k}$ and $\hat{k}$ can not be described by the solid line inside the given ε margin. To limit the outliers, the deviations to the ε environment get captured through $\xi_{\tilde{k}}^*$ and $\xi_{\hat{k}}$ and penalized in the optimization.

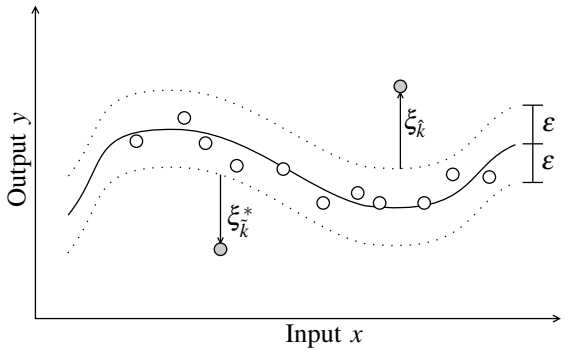

Fig. 1. Visualization of the regression problem for the Support Vector Machines method

To obtain a global optimization and a convex problem the optimization problem (3) and the constraints (4)-(6) are further described and optimized through the Lagrange function $\mathscr{L}$ in

$$\begin{aligned} \min_{w,\rho,\xi_k,\xi_k^*} \max_{\alpha_k,\alpha_k^*,\beta_k,\beta_k^*} \mathscr{L} = J &- \sum_{k=1}^{N} (\beta_k \xi_k + \beta_k^* \xi_k^*) \\ &- \sum_{k=1}^{N} \alpha_k \left(\varepsilon + \xi_k - y_k + w^{\mathrm{T}} \cdot \Phi(x_k) + \rho\right) \\ &- \sum_{k=1}^{N} \alpha_k^* \left(\varepsilon + \xi_k^* + y_k - w^{\mathrm{T}} \cdot \Phi(x_k) - \rho\right) \end{aligned} \tag{7}$$

with the Lagrange coefficients α_k and α_k^* for the k-th constraints (4) and (5). If the value of the regression function is higher than the value in the training data, then it leads to $\alpha_k^* = 0$ and the other way around to $\alpha_k = 0$. The same way, the Lagrange coefficients β_k and β_k^* follow the constraint (6) at the k-th time-step.

So far the Lagrange function (7) states the optimization problem in a dual formulation as it still minimizes the parameters of the original optimization problem (3) and maximizes the Lagrange coefficients to provide the constraints (4)-(6) as much as possible. For such a dual problem the optimal solution is given with the saddle point condition, that the derivatives of $\mathscr{L}$ has to vanish with respect to the primal variables w, ρ, ξ_k and ξ_k^* [15]:

$$\frac{\partial \mathscr{L}}{\partial p} = 0, \quad \forall p \in \{w, \rho, \xi_k, \xi_k^*\}. \tag{8}$$

After the definition of

$$\alpha = [\alpha_1, \cdots, \alpha_N]^{\mathrm{T}}, \tag{9}$$

$$\Omega_{ij} = \Phi^{\mathrm{T}}(x_i) \cdot \Phi(x_j) \tag{10}$$

the substitution of the derivatives (8) in (7) leads to the quadratic optimization problem

$$\max_{\alpha} \mathscr{L} = -\frac{1}{2} \alpha^{\mathrm{T}} H \alpha + f^{\mathrm{T}} \cdot \alpha \tag{11}$$

with

$$\tilde{H} = \begin{bmatrix} \Omega & -\Omega \\ -\Omega & \Omega \end{bmatrix}, \tag{12}$$

$$H = \tilde{H} + \gamma \cdot I, \tag{13}$$

$$f = \left[y^{\mathrm{T}} - \varepsilon \cdot I^{\mathrm{T}}, -y^{\mathrm{T}} - \varepsilon \cdot I^{\mathrm{T}} \right]^{\mathrm{T}}, \tag{14}$$

and I^{T} as a transposed column vector with all ones. Because the determinant of $\tilde{H}$ is zero, a simple regularization with a small $\gamma > 0$ is used to obtain a feasible quadratic optimization problem [13]. The solution of the quadratic optimization are the Lagrange coefficients α_k, which can be used in (8) for $p = w$ as

$$\frac{\partial \mathscr{L}}{\partial w} = w - \sum_{k=1}^{N} (\alpha_k - \alpha_k^*) \Phi(x_k) = 0 \tag{15}$$

to calculate the weight vector, which is required for the regression function (1). For any given solution, ρ can be calculated over the Karush-Kuhn-Tucker conditions

$$\alpha_k \left(\varepsilon + \xi_k - y_k + w^{\mathrm{T}} \cdot x_k + \rho \right) = 0, \tag{16}$$

$$\alpha_k^* \left(\varepsilon + \xi_k^* + y_k - w^{\mathrm{T}} \cdot x_k - \rho \right) = 0, \tag{17}$$

$$(C - \alpha_k) \cdot \xi_k = 0, \tag{18}$$

$$(C - \alpha_k^*) \cdot \xi_k^* = 0, \tag{19}$$

that for any point of the solution the product between the primal variables and the constraints has to vanish [15].

III. LEAST-SQUARES SUPPORT VECTOR MACHINES

LS-SVM is first formulated by Suykens in [17] as a variation of the standard SVM formulation. The main difference between the two methods is, that LS-SVM solves a system of linear equations instead of an optimization problem. Therefore, the error

$$e_k = y_k - \hat{y}_k \tag{20}$$

is defined as the deviation between the value from the regression model (1) $\hat{y}_k$ and the output value within the training data y_k at time-step k. Before the formulation as a system of linear equations, the LS-SVM method also formulates a optimization problem as

$$\min_{w,\rho,e_k} J = \frac{1}{2} w^{\mathrm{T}} w + \frac{C}{2} \sum_{k=1}^{N} e_k^2 \tag{21}$$

s.t.

$$y_k = w^{\mathrm{T}} \cdot \Phi(x_k) + \rho + e_k, \tag{22}$$

with C as the trade-off constant. Fig. 2 shows an exemplary case for how the regression problem is described for the LS-SVM method. The data points are visualized as circles, with the input and output space both being one-dimensional. LS-SVM looks for a function like the one plotted solid and minimizes the absolute value of w and the quadratic sum of the error at every time-step k. Again the minimization of the absolute value of w generates a smoother function and the minimization of the error generates a more data driven model. Both objectives can be prioritized against each other through the trade-off constant C.

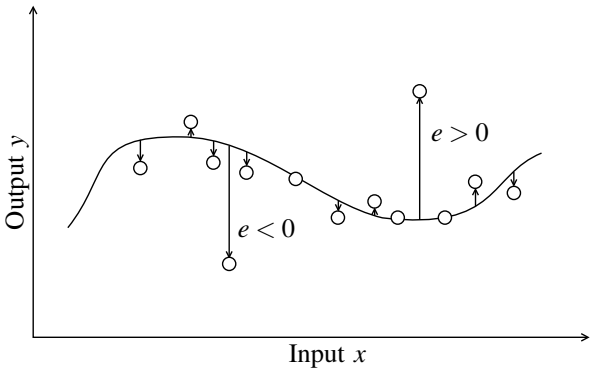

Fig. 2. Visualization of the regression problem for the Least-Squares Support Vector Machines method

The Lagrange function $\mathscr{L}$ and the dual problem are defined similar to the SVM method as

$$\min_{w,\rho,e_k} \max_{\alpha_k} \mathscr{L} = J - \sum_{k=1}^{N} \alpha_k \left(e_k - y_k + w^{\mathrm{T}} \cdot \Phi(x_k) + \rho \right). \tag{23}$$

Through the optimality conditions, that

$$\frac{\partial \mathscr{L}}{\partial p} = 0, \quad \forall p \in \{w, \rho, e_k, \alpha_k\} \tag{24}$$

$\frac{\partial \mathscr{L}}{\partial w} = 0$ and $\frac{\partial \mathscr{L}}{\partial e_k} = 0$ can be substituted in $\frac{\partial \mathscr{L}}{\alpha_k} = 0$ [17]. As a result, the modeled output at time-step k can be described as

$$y_k = \frac{\alpha_k}{C} + \sum_{l=1}^{N} \alpha_l \Phi^{\mathrm{T}}(x_l) \cdot \Phi(x_k) + \rho. \tag{25}$$

Along with $\frac{\partial \mathscr{L}}{\partial \rho} = 0$, (25) can be written in a system of linear equations as

$$\begin{bmatrix} 0 \\ y \end{bmatrix} = \begin{bmatrix} 1^{\mathrm{T}} & 0 \\ \frac{1}{C} I + \Omega & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ \rho \end{bmatrix} \tag{26}$$

for all time-steps, with

$$y = [y_1, \cdots, y_N]^{\mathrm{T}} \tag{27}$$

$$\alpha = [\alpha_1, \cdots . \alpha_N]^{\mathrm{T}} \tag{28}$$

$$\Omega_{ij} = \Phi^{\mathrm{T}}(x_i) \cdot \Phi(x_j). \tag{29}$$

The solution of (26) delivers w and ρ of the regression function (1).

IV. BLACKBOX IDENTIFICATION

The identification of dynamic systems in the time domain requires the input x to contain former values of the exogenous input u_k and the output y_k of the system [10]. This is due to the characteristic of dynamic systems, that there is often no direct mapping between an explicit input and output value. The output rather depends on the form of the input signal. Hence, the input vector for system identification is defined as

$$x_k = [u_{k-L}, \cdots u_{k-1}, y_{k-L}, \cdots y_{k-1}]^{\mathrm{T}} \tag{30}$$

with L lags of former inputs and outputs. In this paper, the number of lags for input and output are chosen to be identical to simplify the model tuning. But in general, it is possible to choose the number of lags for input and output independently. The number of lags to use for modeling, can also be determined over a large number of initial lags, that are pruned through model validation [10]. Although, with pruning additional time for computation has to be expected. Through the identified number of lags, the blackbox approach inherently describes the time delay of the system without additional identification of the time delay itself.

The unknown parameters w and ρ can be solved the same way, as it is described in chapter II. Additionally, to gain an ARX formulation of the regression, a function as in (2) is used to define the parameters as

$$[b_L \cdots b_1] := [w_1 \cdots w_L], \tag{31}$$

$$[a_L \cdots a_1] := [w_{L+1} \cdots w_{2L}]. \tag{32}$$

Similar to the formulation in [13], with (31) and (32), the regression problem can be formulated as

$$y_k = \sum_{i=1}^{L} a_i \cdot y_{k-i} + \sum_{j=1}^{L} b_j \cdot u_{k-j} + \rho, \tag{33}$$

which is transformed into a linear state-space model

$$z_{k+1} = A \cdot z_k + B \cdot u_k \tag{34}$$

$$y_k = C \cdot z_k + D \cdot u_k \tag{35}$$

with the definition of the state as

$$z_k = [\rho, u_{k-L}, \cdots u_{k-1}, y_{k-L}, \cdots y_k]^{\mathrm{T}}, \tag{36}$$

$$z_0 = [\rho, 0, \cdots, 0]^{\mathrm{T}} \tag{37}$$

and the matrices as

$$A = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 1 \\ 1 & b_L & b_{L-1} & \cdots & b_2 & a_L & \cdots & a_1 \end{bmatrix} \tag{38}$$

$$B = [0, \cdots, 0, 1, 0, \cdots, b_1]^{\mathrm{T}} \tag{39}$$

$$C = [0, \cdots, 0, 1], \tag{40}$$

$$D = 0. \tag{41}$$

V. GREYBOX IDENTIFICATION

The greybox approach allows the preset of the model structure in (33) with a defined order for inputs and outputs. It is further allowed to preset known parameters a_i and b_j. In contrast to the blackbox approach, the greybox approach requires the a priori identification of the time delay with external methods. That way the model order for inputs has to be increased through $b_j = 0$ for $j < d$, with d being the number of delay time steps. The remaining parameters are modeled with the LS-SVM method as

$$a_i\left(x_{k-i}^{a_i}\right) = w_i^{a^{\mathrm{T}}} \cdot \Phi_{a_i}\left(x_{k-i}^{a_i}\right), \tag{42}$$

$$b_j\left(x_{k-j}^{b_j}\right) = w_j^{b^{\mathrm{T}}} \cdot \Phi_{b_j}\left(x_{k-j}^{b_j}\right), \tag{43}$$

with $x_{k-i}^{a_i}$ being the scheduling variable of the parameter a_i at time-step $k-i$ and $x_{k-j}^{b_j}$ being the scheduling variable of b_j at time-step $k-j$. The scheduling variables are the influences, which lead to the changing behavior of the system during the process. The vector w_i^a defines the weight for the parameter a_i and the vector w_j^b defines the weight for b_j. Contrary to the blackbox approach, the greybox approach does not learn the behavior between the input and output of the system, but rather the unknown parameters of the preset model. Generally, the parameters are not measurable during the process. Therefore, the model should be trained indirectly over the error at the system output as in (20). Hence, the optimization problem (21) reads as

$$\min_{w_i^a, w_j^b, \rho, e_k} J = \frac{1}{2}\left(\sum_{i=1}^{P} ||w_i^a||^2 + \sum_{j=1}^{Q} ||w_j^b||^2\right) + \frac{C}{2} \cdot \sum_{k=k_0}^{N} e_k^2 \tag{44}$$

s.t.

$$y_k = e_k + \sum_{i=1}^{P} w_i^{a^{\mathrm{T}}} \cdot \Phi_{a_i}\left(x_{k-i}^{a_i}\right) y_{k-i} + \sum_{j=1}^{Q} w_j^{b^{\mathrm{T}}} \cdot \Phi_{b_j}\left(x_{k-j}^{b_j}\right) u_{k-j} + \rho, \quad k_0 \le k \le N \tag{45}$$

with $k_0 = \max\{P+1, Q+1\}$, P being the model order for the output and respectively Q being the model order for the input. The optimality conditions and calculations compare to chapter III with the difference, that there are additional derivatives due to the existence of different weights w_i^a and w_j^b. The resulting linear equation system reads the same as (26), with

$$\Omega_{kl} = \sum_{i=1}^{P} y_{k-i} \Phi_{a_i}^{\mathrm{T}}\left(x_{k-i}^{a_i}\right) \cdot \Phi_{a_i}\left(x_{l-i}^{a_i}\right) y_{l-i} + \sum_{j=1}^{Q} u_{k-j} \Phi_{b_j}^{\mathrm{T}}\left(x_{k-j}^{b_j}\right) \cdot \Phi_{b_j}\left(x_{l-j}^{b_j}\right) u_{l-j}. \tag{46}$$

The solution of the linear equation system provides w_i^a, w_j^b and ρ. With $x_{l-i}^{a_i}$, $x_{l-j}^{b_j}$, y_{l-i} and u_{l-j} being out of the training data, the identified system is described as

$$y_k = \sum_{i=1}^{P} \sum_{l=k_0}^{N} \alpha_{l-k_0+1} \Phi_{a_i}^{\mathrm{T}} \left(x_{l-i}^{a_i}\right) \Phi_{a_i} \left(x_{k-i}^{a_i}\right) y_{l-i} y_{k-i} \qquad (47)$$

$$+ \sum_{j=1}^{Q} \sum_{l=k_0}^{N} \alpha_{l-k_0+1} \Phi_{b_j}^{\mathrm{T}} \left(x_{l-j}^{b_j}\right) \Phi_{b_j} \left(x_{k-j}^{b_j}\right) u_{l-j} u_{k-j} + \rho.$$

The formulation in a linear state-space model for the greybox approach can be written the same way as in (36)-(41) for the blackbox approach. Further, the product between two nonlinear mappings is substituted as

$$K(a,b) := \Phi^{\mathrm{T}}(a)\,\Phi(b) \qquad (48)$$

with a kernel function $K(a,b)$. A wide variety of possible definitions for such a function exist in the literature, which have to be chosen in dependence of the respective system and data [14]. The main advantage of the kernel function is, that it requires less computational resources [9].

VI. EXPERIMENTAL RESULTS

The described methods in the previous chapters were validated on data, which are gained from a CNC machining center of the model "Mazak Variaxis i-600" with five axis with a sample-time of $T_s = 0.01$s. This paper aims at the identification of the dynamic behavior while generating the tool velocity according to the feed velocity. Therefore, the input of the system is the feed velocity and the output is the resulting tool velocity. The dynamic behavior was tested with different step signals of different heights. For the blackbox approach, the "LibSVM" library [7] and the hyperparameters $L = 10$, $C = 0.01$ and $\varepsilon = 0$ were used for the identification. In the literature, usually a gridsearch with k-fold crossvalidation is used for the selection of the last mentioned hyperparameters [9]. In this work, results with less deviation could be gained through manual testing of parameter combinations. For the greybox identification, $C = 10$ was also chosen manually as the LS-SVM hyperparameter, since this way more accurate results could be gained. It demands further studies, if the worse results with a gridsearch and k-fold crossvalidation link to the weaknesses of the method for time-series predictions [5]. As scheduling variables for the greybox approach, $x_{k-i}^{a_i}$, $x_{k-j}^{b_j}$ were all set to $\Delta u_k = u_k - u_{k-1}$ for every time-step k, since the nonlinearities are assumed to correlate to the step height. Therefore, the kernel was chosen as a Fourier kernel

$$K(x_1,x_2) = \frac{1-q^2}{2\left(1-2q\cos\left(||x_1-x_2||\right)+q^2\right)}, \qquad (49)$$

since the kernel function has to reflect the difference between two values. The parameter of the Fourier kernel was chosen through manual testing as $q = 0.005$. The greybox approach was also tested with various other kernels, namely with linear, RBF, polynomial and hyperbolic kernels. Between the tested kernels, the Fourier kernel delivered the results with the least deviation. The model structure for the greybox identification was given as $P = 2$, $Q = 1$, since former works suggest similarities to a second order system [16]. Through the a priori identified time delay of 5 time-steps the input order increases to $Q = 6$, like it is described in chapter V.

Both, the blackbox and the greybox approaches were used in such a manner, that at every step measured values of previous steps are used to train a model. In general, the amount of training data can be limited to a certain number of previous steps. However, in this work data from all previous steps were used for the training. For the greybox approach in specific, better results were received without the offset of the data from zero. Following the training, the models were used to predict the future value of the system. Fig. 3 shows a part of the resulting signal plots. The input signal of the system is plotted as a dotted and the measured output of the system as a straight line. Ideally, the identified system behavior has to reproduce the measured output. The predictions of the blackbox approach are plotted as a dash-dotted and the predictions of the greybox approach as a dashed line.

At this scale, the plots of the blackbox and greybox approaches are hard to distinguish from the measured output. Nonetheless, a over- and undershooting of the greybox model is still observable after steps. Due to its model structure, the greybox model behaves like a system of second order. The blackbox approach avoids the same effect with its greater degree of freedom and therefore its free choice of the model order. Further, Fig. 4 shows the relative error

$$e_{rel_k} = \frac{|y_{measure_k} - y_{model_k}|}{y_{measure_k}} \qquad (50)$$

between the measured output $y_{measure_k}$ and the model output y_{model_k} at every time-step k. The relative error between the measured output and the prediction of the blackbox approach is plotted as a solid and the relative error between the measured output and the greybox approach as a dashed line. The mean relative error $\overline{e_{rel}}$ over the plotted area is for the blackbox approach 0.0196 and for the greybox approach

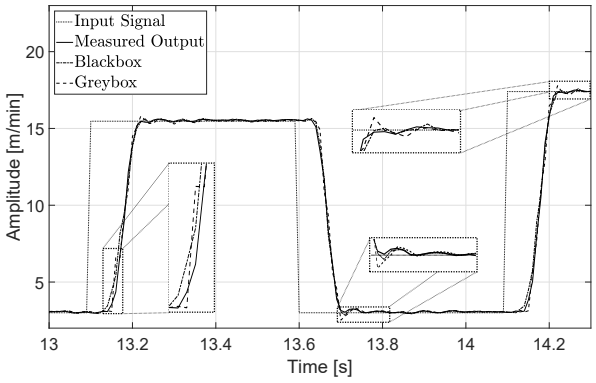

Fig. 3. Plots of the input and output signal of the CNC machining center compared to the results of the blackbox and greybox approach

0.0252. The greybox approach has a higher error, mainly due to higher peaks. The error for both approaches peak at the edges of the steps. This parts of the signal are most challenging to reproduce, since even small differences at reflecting the system behavior lead to big errors. The peaks of the greybox approach are higher than the peaks of the blackbox approach, due to a smoother approximation of the blackbox approach through its higher degree of freedom. However, due to a higher number of lags, the blackbox model needs more time to adapt to the measured data. Hence, at 13.16s and 14.6s the deviation of the blackbox approach decreases slower than the deviation of the greybox approach. The greybox approach is adapting faster to measured data, since the learning effects a smaller number of parameters with big influence on the output. This results in bigger changes of the model prediction and the deviation between two time-steps. A demonstration of this behavior can be observed especially at 13.16s, where a big misprediction is corrected immediately at the next time-step. For both approaches, the error at the respective edges decreases with the increasing amount of data. Outside of the edges both approaches describe the system nearly ideally with only very small deviations including measured noise.

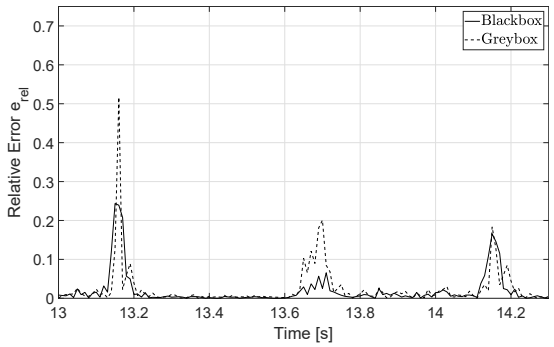

Fig. 4. Comparison of the relative error for the blackbox and greybox approach

VII. CONCLUSIONS

The SVM and the LS-SVM methods were used to generate a blackbox and respectively a greybox model. Both models were formulated in such a manner, that the application within a MPC strategy is possible. It was further possible to use the SVM and LS-SVM approaches for a recurrent system identification with changing data. The results were validated with respect to measured data from the CNC machining center. The models from both approaches were able to describe the dynamic behavior with a low relative error. The blackbox model led to a mean relative error of 0.0196 and the greybox approach to 0.0252. The blackbox approach was further able to generate a smoother model due to its higher degree of freedom. However, the blackbox model needed more time to adapt to the changing training data. The greybox model was able to follow preset configurations of the model structure, but was difficult to tune. The choice of the parameters, kernels and scheduling variables had a big influence on the resulting model quality. Due to higher peaks of the error at the edges, a bigger overall error resulted for the greybox model compared to the blackbox model. Interesting for future research are reliable parameter tuning algorithms and the integration of the studied methods in the existing MPC strategy. A more sophisticated choice of the kernel function and a direct comparison with other identification methods in regards of accuracy and computational resources are also possible fields for future research.

ACKNOWLEDGMENT

The authors would like to thank the German Research Foundation DFG for the support of the depicted research within the Cluster of Excellence "Integrative Production Technology for High-Wage Countries".

REFERENCES

[1] ABEL, Dirk: Regelungstechnik und Ergänzungen (Höhere Regelungstechnik). 2017. – 488 S. – ISBN 978–3–8107–0067–4

[2] ACUNA, Gonzalo ; CURILEM, Millaray: Time-variant parameter estimation using a SVM Gray-Box model: Application to a CSTR Process. In: 2013 3rd International Conference on Systems and Control, ICSC 2013 (2013), S. 414–418. DOI 10.1109/ICoSC.2013.6750892.

[3] ACUÑA, Gonzalo ; MÖLLER, Hans: Indirect training of Gray-Box Models using LS-SVM and genetic algorithms. DOI 10.1109/LA–CCI.2016.7885719.

[4] ADAMS, Oliver: Modellbasierte prädiktive Kraftregelung beim Fräsen, RWTH Aachen, 2016.

[5] BERGMEIR, Christoph: Validity of Cross-Validation for Evaluating Time Series Prediction. In: Computational Statistics and Data Analysis (2018), Volume 120, 70-83 DOI 10.1016/j.csda.2017.11.003.

[6] BRECHER, Christian: Integrative Production Technology for High-Wage Countries. 2012. 1096 S. DOI 10.1007/978-3-642-21067-9

[7] CHANG, Chih-chung ; LIN, Chih-jen: LIBSVM : A Library for Support Vector Machines. DOI 10.1145/1961189.1961199

[8] CHEN, Wun-Hwa ; HSU, Sheng-Hsun ; SHEN, Hwang-Pin: Application of SVM and ANN for intrusion detection. In: Computers and Operations Research 32 (2005), Nr. 10, S. 2617–2634. DOI 10.1016/j.cor.2004.03.019

[9] DU, Ke-Lin ; SWAMY, M. N. S.: Neural Networks and Statistical Learning. 2014 (October 2015). ISBN 978–1–4471–5570–6

[10] ESPINOZA, Marcelo ; FALCK, Tillmann ; SUYKENS, J.A.K. ; MOOR, BD: Time series prediction using ls-svms. In: European Symposium on Time Series Prediction, ESTSP 8 (2008), 159–168. DOI 10.1.1.660.9723

[11] IPLIKCI, S.: Support vector machines-based generalized predictive control. In: International Journal of robust and nonlinear control (2006), 843–862. DOI 10.1002/rnc. ISBN 9120256922

[12] ISERMANN, Rolf: Identification of dynamic systems. 2011. 705 S. ISBN 978–3–540–78878–2

[13] ROJO-ALVAREZ, Jose L. ; MARTINEZ-RAMON, Manel ; FIGUEIRAS-VIDAL, Anibal: Support vector method for robust ARMA system identification. In: IEEE Trans. Signal Proc. 52 (2004), Nr. 1, S. 155–164. DOI 10.1109/TSP.2003.820084

[14] SCHÖLKOPF, Bernhard ; SMOLA, Alexander J.: Learning with Kernels. 2001. – 648 S. – ISBN 9780262194754

[15] SMOLA, Alex J. ; SCHÖLKOPF, Bernhard: A Tutorial on Support Vector Regression. In: Statistics and Computing 14 (2004), Nr. 3, S. 199–222. DOI 10.1023/B:STCO.0000035301.49549.88. –

[16] STEMMLER, Sebastian ; ABEL, Dirk ; SCHWENZER, Max ; ADAMS, Oliver ; KLOCKE, Fritz: Model Predictive Control for Force Control in Milling. In: IFAC-PapersOnLine 50 (2017), jul, Nr. 1, 15871–15876. DOI 10.1016/j.ifacol.2017.08.2336. ISSN 24058963

[17] SUYKENS, J. A. ; VANDEWALLE, J.: Least squares support vector machine classifiers In: Neural Processing Letters (1999), 9-293. DOI 10.1023/A:1018628609742 ISSN 1573-773X

[18] VAPNIK, V ; LERNER, A: Pattern recognition using generalized portrait method. In: Automation and remote control 24 (1963), Nr. 6, S. 774–780. DOI citeulike–article–id:619639. – ISSN 0005–1179

[19] VAPNIK, Vladimir N.: The nature of statistical learning theory. 2000. – 314 S. ISBN 978–0–387–98780–4