Manuscript version
DOI original paper: 10.1109/ICCMA56665.2022.10011461

Data-Driven Force Control of an Automated Scratch Test

Florian Diepers*, Dominik Polke, Elmar Ahle
Electrical Engineering and Computer Science
University of Applied Sciences Niederrhein
Krefeld, Germany
{florian.diepers, dominik.polke, elmar.ahle}@hs-niederrhein.de

Dirk Söffker
Dynamics and Control
University of Duisburg-Essen
Duisburg, Germany
soeffker@uni-due.de

***Abstract*—The development of chemicals, like coatings, is increasingly automated in high-throughput plants. The assessment of the quality of the coating formulation is also automated to achieve precise and reproducible results. Usually, a cross-cut test is performed to characterize the adhesion of coatings. To obtain comparable conditions and comply with standards, the force during scratching has to be controlled. In this paper, the force control of an automated testing equipment is optimized. There are different substrates, coating formulations and process parameters influencing the quality of the adhesion of the coatings. Because of this, the force controller has to cover a variety of dynamics. The different coatings and nonlinear behavior of the scratching makes modeling the system extremely difficult. Therefore, a data-driven approach is used. Two data-driven methods, the intelligent PID (iPID) and Virtual Reference Feedback Tuning (VRFT), are applied to the field of force control. The results for both control methods are compared. Additionally, both methods are combined with an Iterative Learning Control (ILC) to optimize the control performance over multiple iterations. The qualities of the data-driven controllers are evaluated. All results are compared to the currently implemented controller.**

Keywords—data-driven control, force control, intelligent PID (iPID), Virtual Reference Feedback Tuning (VRFT), Iterative Learning Control (ILC), coating, cross-cut test

I. INTRODUCTION

The development of new formulations of coatings is usually done experimentally. To accelerate the development process and objectify the results, the formulation, application and characterization of these experiments are automated. One important property of coatings is the adhesion to the surface. Various scratch tests [1–3] have been defined and are used to characterize this adhesion.

For good results, the testing conditions must be reproducible. Because of this, a defined force must be guaranteed during a scratch test. To ensure this, an automated testing equipment is used to perform those cross-cut tests with an appropriate force control. The testing equipment is also used in [4] to automate the visual evaluation of the scratch results via image segmentation using Deep Convolutional Neuronal Networks. In this work, the force control is optimized to achieve a better reproducibility. To cover the nonlinear dynamics, different data-driven model-free control methods are evaluated for force control. The results are compared with the performance of the existing controller.

This paper is structured as follows. First, a short introduction and further references to the used control methods intelligent PID, Virtual Reference Feedback Tuning and Iterative Learning Control are given in section II. In section III, the testing equipment for the cross-cut test is presented. The different methods are implemented in section IV and the results are presented in section V. Finally, a short summary and an outlook of future work is given.

II. BACKGROUND AND RELATED WORK

Controller design using only measured data is denoted as data-driven control [5]. There are many different methods for data-driven control, e.g. [5, 6]. In contrast to that, model-based control is using a given model for controller design. In this work, no model but directly measured system data are used. In this section, three different data-driven model-free control algorithms are presented, which are applied to force control.

A. Intelligent PID

As first application example, an intelligent PID controller described by Fliess and Join [7–9] is used. According to [7] the system is considered to have an unknown finite-dimensional linear or nonlinear description of the input-output behavior

$$E\left(t, y, \dot{y}, \cdots, y^{(a)}, u, \dot{u}, \cdots, u^{(b)}\right) = 0, \tag{1}$$

where E is sufficiently smooth. During the control the system description (1) is not available and the simple local model

$$y^{(n)} = F + \alpha u \tag{2}$$

is used instead. This model is locally valid for short time intervals and therefore must be updated each time step. The constant parameter $\alpha \in \mathbb{R}$ and the derivation order n are tuneable parameters, where α is chosen so that $y^{(n)}$ and $\alpha\, u$ have the same order of magnitude.

The derivative order n does not have to be equal to the highest order a of the system. Generally, the order is chosen to 1 or 2 [7, 9]. The control law for a time discrete iPID controller

The authors acknowledge gratefully the program FH Basis of the German federal country North Rhine-Westphalia for partly funding the cross-cut testing equipment.

is obtained by combining a classical PID controller with the local model (2) to

$$u(k) = \frac{1}{\alpha}\Big(-F(k) + y_{\mathrm{d}}^{(n)}(k) + K_{\mathrm{p}}\, e(k) + K_{\mathrm{i}} \sum e(k)\, \Delta T + K_{\mathrm{d}}\, \Delta e(k)\Big), \tag{3}$$

where $K_{\mathrm{p}}, K_{\mathrm{i}}, K_{\mathrm{d}} \in \mathbb{R}$ are the tuning parameters, $y_{\mathrm{d}}^{(n)}$ denotes the n^{th} derivative of the reference output, $e(k) = y_{\mathrm{d}}(k) - y(k)$ the output tracking error, $\Delta e(k) = e(k) - e(k-1)$ the difference of the tracking error, and k represents the current time step. The value $F(k)$ describes the whole unknown system dynamics. It is estimated in every time step based on (2) using the estimation of the n^{th} derivative of the measured output $\hat{y}^{(n)}(k)$ by

$$\hat{F}(k) = \hat{y}^{(n)}(k) - \alpha u(k-1), \tag{4}$$

where $\hat{F}(k)$ denotes the estimated value. The system input u is shifted by one time step to avoid algebraic loops [8].

In the following, the local model (2) is described by the first derivative $\dot{y}$. To calculate the estimated derivative $\hat{\dot{y}}$ a robust differentiator based on sliding mode technique from [10] is used. To differentiate the system output $y(k)$ the difference (sliding surface) $s(k) = x(k) - y(k)$ is considered. The estimated derivate results from the super twisting algorithm

$$\begin{aligned} \dot{z}(k) &= -\omega\, \mathrm{sgn}(s(k)) \\ \hat{\dot{y}}(k) &= z(k) - \mu |s(k)|^{\frac{1}{2}} \mathrm{sgn}(s(k)), \end{aligned} \tag{5}$$

with $z(k) = z(k-1) + \dot{z}(k)\Delta T > 0$, $\omega > \Theta > 0$ with the Lipschitz constant Θ, $\mu^2 \geq 4\,\Theta\, \frac{\omega+\Theta}{\omega-\Theta}$, and $x(k) = x(k-1) + \hat{\dot{y}}(k)\Delta T$.

B. Virtual Reference Feedback Tuning

The Virtual Reference Feedback Tuning is a method to determine the parameters of a fixed-structure controller with measured input and output data. The method is described in [11–14]. Applications can be found in [15] and [16].

The VRFT controller design [11] is based on a desired system behavior described by a reference model. The reference model is given as the time-discrete *z*-transformed transfer function $M(z)$. The goal is to determine the controller

$$C(z, \theta) = \beta^{\mathrm{T}}(z)\, \theta \tag{6}$$

so that the closed loop with the unknown plant dynamics $P(z)$ equals the reference model $M(z)$. The control loop is shown in Fig. 1.

The aim is to determine the optimal controller parameters $\theta = [\theta_1 \quad \theta_2 \quad \cdots \quad \theta_m]^{\mathrm{T}}$ for the fixed structure $\beta^{\mathrm{T}}(z) = [\beta_1(z) \quad \beta_2(z) \quad \cdots \quad \beta_m(z)]$ of the controller with m linear discrete-time transfer functions. Because the transfer function $P(z)$ of the plant is unknown the system is described by the measured data of the system input u and the system output y of the plant. With this, the VRFT can be summarized in three steps [11]:

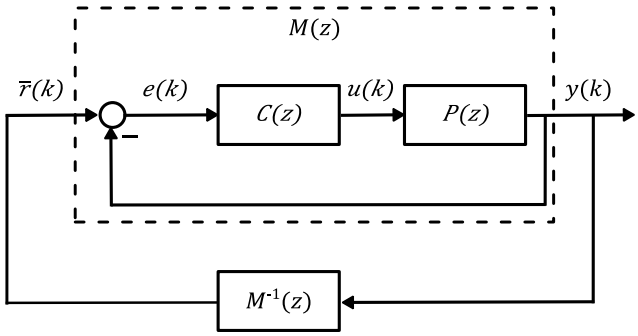

Fig. 1 Block diagram of the desired system [13]

1. Determine the virtual reference $\bar{r}(k)$ for the desired system $y(k) = M(z)\bar{r}(k)$ based on the measured outputs $y(k)$. This results in the tracking error $e(k) = \bar{r}(k) - y(k)$.
2. Use $L(z)$ to filter $e(k)$ and $u(k)$

   $$e_{\mathrm{L}}(k) = L(z)\, e(k) \text{ and } u_{\mathrm{L}}(k) = L(z)\, u(k), \tag{7}$$

   where the index L describes the filtered elements.
3. Calculate the specific controller parameter vector $\hat{\theta}_N$ minimizing the criterion

   $$J_{\mathrm{VRFT}}^{N}(\theta) = \frac{1}{N}\sum_{k=1}^{N} (u_{\mathrm{L}}(k) - C(z,\theta)\, e_{\mathrm{L}}(k))^2 \tag{8}$$

   for N data points, which results in

   $$\hat{\theta}_N = \left[\sum_{k=1}^{N} \varphi_{\mathrm{L}}(k)\, \varphi_{\mathrm{L}}(k)^{\mathrm{T}}\right]^{-1} \sum_{k=1}^{N} \varphi_{\mathrm{L}}(k)\, u_{\mathrm{L}}(k), \tag{9}$$

   with $\varphi_{\mathrm{L}}(k) = \beta(z)\, e_{\mathrm{L}}(k)$.

The goal of the controller is to generate the correct system input $u(k)$ to achieve the desired output $y(k)$. Therefore, the virtual reference $\bar{r}(k)$ (which was never used to generate the data) is used to determine the necessary tracking error $e(k)$. The controller is optimized to generate the correct $u(k)$ for a given error.

C. Iterative Learning Control

The Iterative Learning Control (ILC) is based on the idea of learning from tracking errors of previous iterations. The method is described in [17] and in the extensive survey [18]. Some newer developments can be found in [19, 20]. Applications are repetitive tasks like robot movements. It learns repeating disturbances and suppresses them in the future.

The ILC is mostly used in a discrete setting because it uses stored values from previous iterations which are always sampled. The plants dynamics are often represented in a lifted-system representation [17, 18], where the system inputs $u(k)$ and outputs $y(k)$ are interpreted as N-dimensional vectors u and y of a SISO system over a finite time interval $0 \leq k < N$ with N sampled data points per iteration. The matrix G denotes the single impulse responses of the system

$$\underbrace{\begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(N-1) \end{bmatrix}}_{y} = \underbrace{\begin{bmatrix} \delta_0 & 0 & \cdots & 0 \\ \delta_1 & \delta_0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ \delta_{N-1} & \dots & \delta_1 & \delta_0 \end{bmatrix}}_{G} \underbrace{\begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(N-1) \end{bmatrix}}_{u}. \tag{10}$$

A typical update law for ILC [17] is

$$u_{i+1}(k) = Q\,[u_i(k) + L(e_i(k))], \tag{11}$$

where i denotes the iteration index, k the time index within an iteration, L a learning function, and Q an optional filter. The system input for the current time step k of the next iteration $i+1$ is based on the corresponding system input of the current iteration and the resulting tracking error.

Different types of learning functions are used in literature [18]. In this paper, the P-type learning function as

$$u_{i+1}(k) = u_i(k) + \Psi_\mathrm{p}\, e_i(k), \tag{12}$$

with the filter $Q = 1$ and the learning function $\Psi_\mathrm{p} \in \mathbb{R}$ as tuning parameter of the controller, is used.

The resulting concept is visualized in Fig. 2. The system input depends on the stored values of the system input and tracking error of the previous iteration. With a growing number of iterations, the system input is refined and the tracking error for repeating disturbances vanishes. This is only valid, if the initial conditions are equal for every iteration and the system is asymptotically stable. Stability can be achieved if the condition

$$\left|1 - \Psi_\mathrm{p}\,\gamma\right| < 1 \tag{13}$$

is met.

III. Experimental Setup

The testing equipment executes automated cross-cut tests. As preparation, different coatings are applied to metal sheets with a high-throughput equipment. These metal sheets are inserted in the testing equipment and fixed by vacuum. Then, the metal sheet is moved below a needle and a pattern according to the standards [1–3] is scratched into the coating material. Before and after the scratch test a photo of the coated surface is taken. To evaluate the coating properties those photos are analyzed, e.g. in [4]. A photo of the setup is shown in Fig. 3.

The cross-cut machine uses three synchronous motors to move in the three cartesian coordinates. The metal sheet is located on a carriage which moves in the x-y-plane. Two force sensors are installed in this carriage. The needle moves up and down in the z-direction and applies the force on the coating, which is measured by the force sensors. A spring is already used to reduce the stiffness of the system and achieve better control results.

The whole testing equipment is controlled via a programmable logic controller (PLC). The PLC code runs with a cyclic time of 1 ms. At the beginning, the force values are read from both sensors as measured system outputs and added to a single value for each cartesian axis. The resulting force value is used by a controller to determine the motor position increment as system input. The controller must have a short reaction time

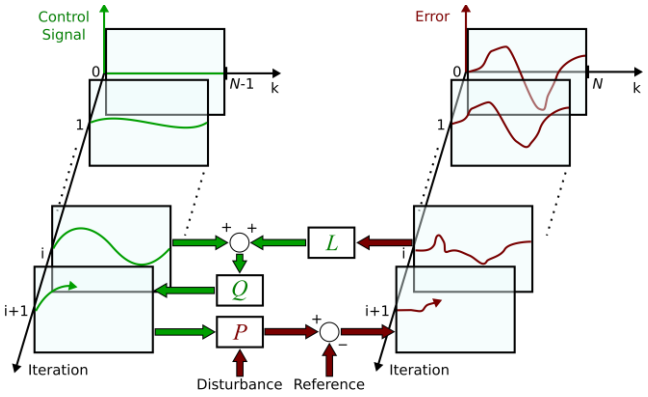

Fig. 2 Development of the control signal and tracking error over the iterations [18]

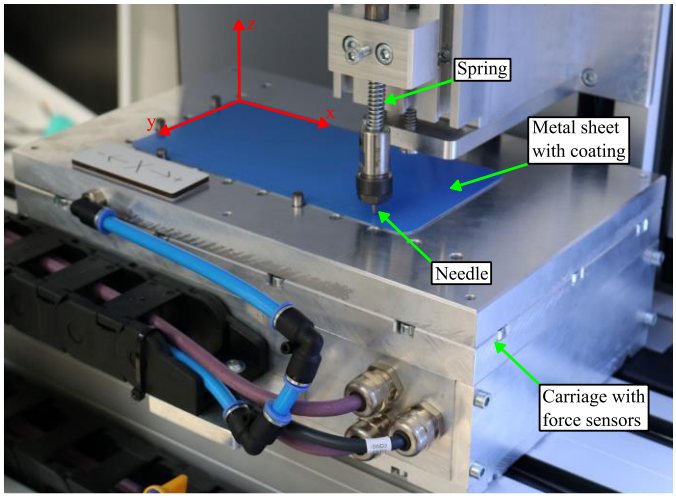

Fig. 3 Cross-cut testing equipment

because there is only a small distance where the resulting force is in a permitted range. The force controller is the outer part of a cascade controller. The inner control loop is a position control directly implemented on the PLC. The current force controller for the cross-cut tests is implemented as an adaptive proportional controller controlling the position increment.

IV. Application of the Data-Driven Methods

The data-driven control methods from section II are implemented on the PLC of the experimental setup. The system output is the measured force in z-direction and the system input the position increment in z-direction. Therefore, a positive system input results in a decreasing system output and vice versa. The position increment has to be quite small, because the range of positions with an acceptable force response is also small.

A. Intelligent PID

The iPID (3) is implemented with a derivative order $n = 1$ and a modification of the derivative part. The difference of the tracking error $K_\mathrm{d}\,\Delta e(k)$ is replaced by the estimated derivate $-K_\mathrm{d}\,\hat{\dot{y}}(k)$. This holds because

$$\hat{\dot{e}} = \hat{\dot{y}}_\mathrm{d} - \hat{\dot{y}} = 0 - \hat{\dot{y}} \tag{14}$$

and the reference can be considered as constant. The numerical derivation (based on (5)) is chosen because the measured signals have high noise which results in poor performance of the simple difference $\Delta e(k)$. The chosen parameter values are given in Table I.

TABLE I VALUES OF THE PARAMETER IN THE IPID ALGORITHM

Parameter	iPID				Derivation	
	α	K_p	K_i	K_d	ω	μ
Value	$-2 \cdot 10^6$	0.003	0	-0.975	1.0	0.4

The tuning factor α has to be negative because a negative system input corresponds to an increasing force response. The absolute amount of α has to be large to damp the system input. The system already has an integral behavior, therefore the integral part of the controller K_i is set to zero. Hence, the implemented controller is an iPD. The derivative gain K_d is chosen near to minus one because with a smaller difference $-1 - K_\mathrm{d}$ the influence of the derivative on the system input is reduced.

B. Virtual Reference Feedback Tuning

For the VRFT controller the fixed PD-type controller structure

$$\beta = \begin{pmatrix} 1 & \frac{2}{T_\mathrm{s}} \frac{1 - z^{-1}}{3 - z^{-1}} \end{pmatrix}^\mathrm{T} \tag{15}$$

with the sampling time $T_\mathrm{s} = 1$ ms is chosen based on the PID example from [21]. The goal is to determine the parameter vector $\hat{\theta} = [\hat{\theta}_1 \quad \hat{\theta}_2]^\mathrm{T}$ for the controller $C(z, \theta)$ of the fixed structure β. Measured data for the system input $u(k)$ and the system output $y(k)$ are required for this purpose. To obtain these measurements, a single scratch with a velocity of 1 mm/s is performed using the existing controller. The resulting data are shown in Fig. 4.

Beside the measured variables and the controller structure the VRFT algorithm needs the reference model, which is chosen to

$$M(z) = \frac{0.01}{z - 0.99} \ . \tag{16}$$

Also, the filter $L(z)$ is chosen to $L(z) = \frac{W(z)\,(1 - M(z))\,M(z)}{\sqrt{\Phi_\mathrm{u}}}$ with Φ_u being the spectral density of $u(k)$ and $W(z)$ the weighting transfer function which is chosen to $W(z) = 1$. The resulting parameter $\hat{\theta} = [-23.8 \cdot 10^{-6} \quad 20.7 \cdot 10^{-9}]^\mathrm{T}$ is implemented.

C. Iterative Learning Control

The ILC is applied in addition to the other two methods to further optimize the control performance. The groups iPID+ILC and VRFT+ILC are formed. The two methods iPID and VRFT work against disturbances during a scratch. In contrast, the ILC optimizes the control performance over multiple scratches if there are repeating disturbances like differences in the coating

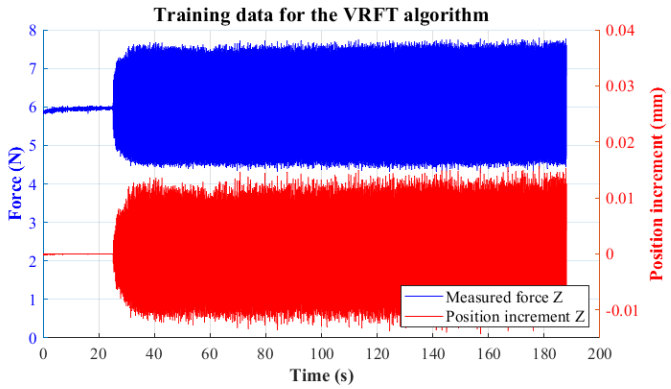

Fig. 4 Training data for VRFT: Position increment $u(k)$ and measured force $y(k)$

thickness. The resulting system input by the combined controllers is realized as the sum of the individual inputs as

$$u_i(k) = u_{i,\,\mathrm{ILC}}(k) + u_{\mathrm{iPID\ or\ VRFT}}(k) \tag{17}$$

based on [22].

The ILC is implemented based on (12) using the tuning parameter $\Psi_\mathrm{p} = -2 \cdot 10^{-6}$. The factor has to be negative because of the inverse behavior between system input and output.

V. RESULTS AND DISCUSSION

The controller performance of the methods described above are compared to the existing adaptive proportional controller. For this, some experiments are performed with the cross-cut testing equipment described in section III. For simplicity, the results presented here are for a single scratch along the x-axis with a velocity of 1 mm/s. The application of other movements like a scratch in y-direction can be done analogously. Only with ILC the correct sorting of the sampled values of the last iteration to the current movement is important and has to be adapted.

A. Comparison of iPID and VRFT with exsiting controller

First, the iPID and VRFT are individually compared to the existing adaptive proportional controller. The results for an exemplary scratch test are given in Fig. 5. The adaptive proportional controller oscillates strongly around the setpoint of 5 newton with an root mean square error (RMSE) of 0.9644. The iPID and VRFT controller oscillate with a significantly less amplitude and a smaller frequency. Both new control algorithms are disturbed by the start of movement but compensate the disturbance with time. This disturbance is handled slightly better by the VRFT in comparison to the iPID. After approximately 20 s the oscillations have reached a constant error range. Then, the iPID oscillates around the setpoint but with the VRFT a permanent control error remains. The RMSE of all three control algorithms are given in Table II. The error values are given once for the whole scratch and once after the settling time of 20 s.

TABLE II ROOT MEAN SQARE ERRORS FOR A SINGLE SCRATCH

Time	Original	iPID	VRFT
$t_0 \le t \le t_\mathrm{end}$	0.9644	0.1834	0.2428
$t_{20\,\mathrm{s}} \le t \le t_\mathrm{end}$	0.9634	0.1243	0.2394

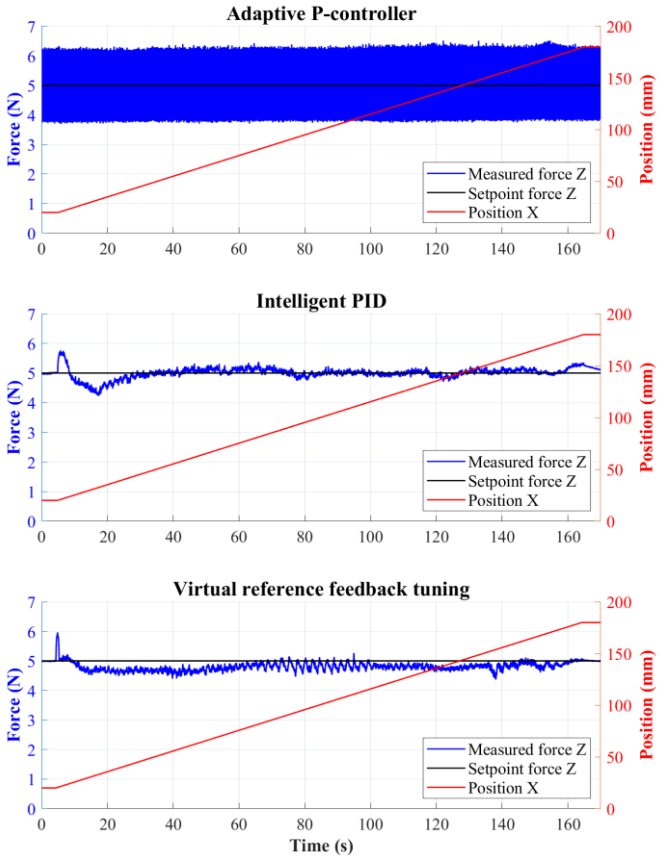

Fig. 5 Results of the force control for the existing controller, iPID and VRFT

B. *Combination of the new control methods with ILC*

The results of the combined control methods iPID+ILC and VRFT+ILC are shown in Fig. 6. The adaptive proportional controller will not be combined with ILC because of its poor performance in single use. Ten iterations are done in a row for both combinations. Each iteration is a single scratch along the x-axis. Between the iterations the force drops to zero, because the needle is lifted and moved back to the start position in x- and the next line in y-direction.

The ILC is supposed to improve the control performance over time by suppressing repeated disturbances. In this example, the implemented ILC does not achieve the desired purpose. No significant improvement over multiple iterations can be observed in Fig. 6. For example, the characteristic force reduction at the beginning of a scratch are not minimized. Only for VRFT+ILC an effect can be observed. The output of the system controlled with this combination is systematical below the reference at start. With increasing number of iterations, the average of a single iteration approaches the setpoint.

Another characteristic is the RMSE for each iteration as shown in Table III. The error for both combinations first decreases slightly and remain constant afterwards. For the iPID+ILC the ILC seems to have no effect. For the VRFT+ILC combination the constant errors (except for the outlier in iteration eight) can be explained with two effects. First, an increasing error caused by the rising amplitude of the measured force. Second, the decreasing error caused by the convergence

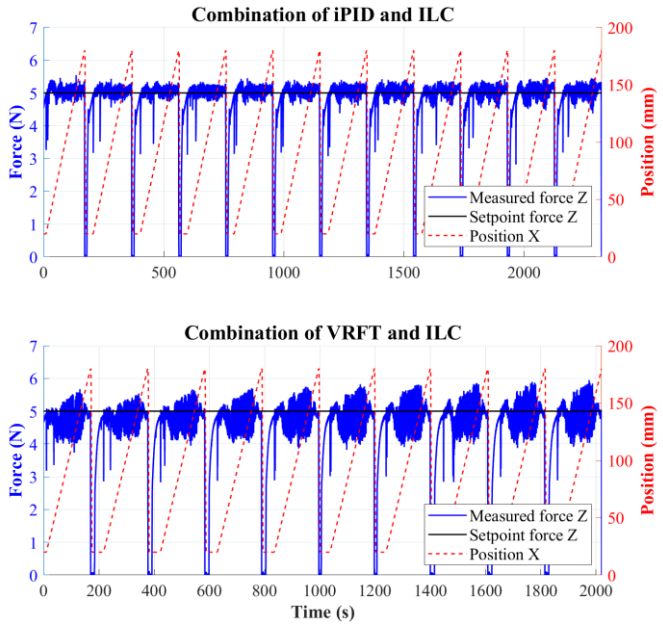

Fig. 6 Results of the force control for the combinations iPID+ILC and VRFT+ILC

TABLE III ROOT MEAN SQARE ERRORS FOR EACH ITERATION

Iterations	**1**	**2**	**3**	**4**	**5**
iPID+ILC	0.2120	0.2022	0.1963	0.1970	0.1965
VRFT+ILC	0.4079	0.3469	0.3656	0.3665	0.3899

Iterations	**6**	**7**	**8**	**9**	**10**
iPID+ILC	0.1926	0.1974	0.1946	0.2006	0.1967
VRFT+ILC	0.3921	0.3957	0.4434	0.3794	0.3850

of the average measured force and the setpoint. Both effects cancel each other out.

VI. SUMMARY AND CONCLUSION

The goal of this contribution is the controller optimization for the cross-cut testing equipment using data-driven methods. For this, a comparison of the iPID and VRFT and the combinations iPID+ILC and VRFT+ILC with the existing controller is done. The existing controller is implemented as an adaptive proportional controller with a poor control performance. For the comparison a simplified individual scratch in x-direction is used. The results show that both iPID and VRFT outperform the existing controller. The iPID results in smaller tracking errors over the scratch and is therefore recommended. The combination of the new control methods with the ILC yields no further improvement. The control performance remains comparable.

In this paper, only model-free control methods are discussed. In the future, data-driven model-based control methods will be investigated additionally. Furthermore, more information can be used to improve the controller, for example the current position or the forces in x- and y-direction. In addition, the different control methods should be applied to the cross cut test. Scratching over existing scratches may introduce a repetitive disturbance, where iterative learning control may have an advantage.

REFERENCES

[1] *DIN EN ISO 4628-8:2013-03, Beschichtungsstoffe- Beurteilung von Beschichtungsschäden- Beurteilung der Menge und der Größe von Schäden und der Intensität von gleichmäßigen Veränderungen im Aussehen- Teil-8: Bewertung der von einem Ritz oder einer anderen künstlichen Verletzung ausgehenden Enthaftung und Korrosion (ISO 4628-8:2012); Deutsche Fassung EN ISO 4628-8:2012*, Berlin.

[2] *DIN EN ISO 2409:2020-12, Beschichtungsstoffe-Gitterschnittprüfung (ISO 2409:2020); Deutsche Fassung EN ISO 2409:2020*, Berlin.

[3] *Test Method for Film Hardness by Pencil Test*, D01 Committee, West Conshohocken, PA.

[4] G. Zhang, C. Schmitz, M. Fimmers, C. Quix, and S. Hoseini, "Deep learning-based automated characterization of crosscut tests for coatings via image segmentation," *J. Coat. Technol. Res. (Journal of Coatings Technology and Research)*, vol. 19, no. 2, pp. 671–683, 2022, doi: 10.1007/s11998-021-00557-y.

[5] Z.-S. Hou and Z. Wang, "From model-based control to data-driven control: Survey, classification and perspective," *Information Sciences*, vol. 235, pp. 3–35, 2013, doi: 10.1016/j.ins.2012.07.014.

[6] Z. Hou, R. Chi, and H. Gao, "An Overview of Dynamic-Linearization-Based Data-Driven Control and Applications," *IEEE Trans. Ind. Electron.*, vol. 64, no. 5, pp. 4076–4090, 2017, doi: 10.1109/TIE.2016.2636126.

[7] M. Fliess and C. Join, "Intelligent PID controllers," in *2008 16th Mediterranean Conference on Control and Automation*, Ajaccio, France, 2008, pp. 326–331.

[8] M. Fliess and C. Join, "Model-free control and intelligent PID controllers: towards a possible trivialization of nonlinear control?," Apr. 2009. [Online]. Available: http://arxiv.org/pdf/0904.0322v1

[9] M. Fliess and C. Join, "Model-free control," *International Journal of Control*, vol. 86, no. 12, pp. 2228–2252, 2013, doi: 10.1080/00207179.2013.810345.

[10] E. Madadi and D. Söffker, "Towards an Optimal Learning for Robust Iterative-Based Intelligent PID," in *ASME 2016 Dynamic Systems and Control Conference*, Minneapolis, Minnesota, USA, 2016.

[11] M. C. Campi, A. Lecchini, and S. M. Savaresi, "Virtual reference feedback tuning: a direct method for the design of feedback controllers," *Automatica*, vol. 38, no. 8, pp. 1337–1346, 2002, doi: 10.1016/S0005-1098(02)00032-8.

[12] M. C. Campi and S. M. Savaresi, "Direct Nonlinear Control Design: The Virtual Reference Feedback Tuning (VRFT) Approach," *IEEE Trans. Automat. Contr.*, vol. 51, no. 1, pp. 14–27, 2006, doi: 10.1109/TAC.2005.861689.

[13] M. Parigi Polverini, S. Formentin, L. Merzagora, and P. Rocco, "Mixed Data-Driven and Model-Based Robot Implicit Force Control: A Hierarchical Approach," *IEEE Trans. Contr. Syst. Technol.*, vol. 28, no. 4, pp. 1258–1271, 2020, doi: 10.1109/TCST.2019.2908899.

[14] M. C. Campi and S. M. Savaresi, "Virtual Reference Feedback Tuning for non-linear systems," in *Proceedings of the 44th IEEE Conference on Decision and Control*, Seville, Spain, 2005, pp. 6608–6613.

[15] M. P. Polverini, S. Formentin, L. A. Dao, and P. Rocco, "Data-driven design of implicit force control for industrial robots," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, Singapore, Singapore, 2017, pp. 2322–2327.

[16] S. Formentin, M. C. Campi, and S. M. Savaresi, "Virtual Reference Feedback Tuning for industrial PID controllers," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 11275–11280, 2014, doi: 10.3182/20140824-6-ZA-1003.01260.

[17] M. Volckaert, M. Diehl, and J. Swevers, "Generalization of norm optimal ILC for nonlinear systems with constraints," *Mechanical Systems and Signal Processing*, vol. 39, 1-2, pp. 280–296, 2013, doi: 10.1016/j.ymssp.2013.03.009.

[18] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control," *IEEE Control Syst.*, vol. 26, no. 3, pp. 96–114, 2006, doi: 10.1109/MCS.2006.1636313.

[19] H. Tao, J. Li, Y. Chen, V. Stojanovic, and H. Yang, "Robust point‐to‐point iterative learning control with trial‐varying initial conditions," *IET Control Theory & Applications*, vol. 14, no. 19, pp. 3344–3350, 2020, doi: 10.1049/iet-cta.2020.0557.

[20] X. Xu, H. Xie, and J. Shi, "Iterative Learning Control (ILC) Guided Reinforcement Learning Control (RLC) Scheme for Batch Processes," in *2020 IEEE 9th Data Driven Control and Learning Systems Conference (DDCLS)*, Liuzhou, China, 2020, pp. 241–246.

[21] A. Care, F. Torricelli, M. C. Campi, and S. M. Savaresi, "A Toolbox for Virtual Reference Feedback Tuning (VRFT)," in *2019 18th European Control Conference (ECC)*, Naples, Italy, 2019, pp. 4252–4257.

[22] E. Madadi and D. Söffker, "Model-Free Approaches Applied to the Control of Nonlinear Systems: A Brief Survey with Special Attention to Intelligent PID Iterative Learning Control," in *ASME 2015 Dynamic Systems and Control Conference*, Columbus, Ohio, USA, 2015.