

# Increasing Trajectory Tracking Accuracy of Industrial Robots Using SINDYc

Diyar Khalis Bilal\* Mustafa Unel\*

\* Faculty of Engineering and Natural Sciences, Sabanci University,  
Tuzla, 34956 Istanbul, Turkey.  
Integrated Manufacturing Technologies Research and Application  
Center, Sabanci University, Tuzla, 34906 Istanbul, Turkey.  
(e-mail: mustafa.unel@sabanciuniv.edu)

---

**Abstract:** In this work a feedforward control approach based on SINDYc (Sparse Identification of Nonlinear Dynamics with Control) is proposed for increasing the trajectory tracking accuracy of industrial robots. Initially, the dynamic relationship between the desired and the actual trajectory is sparsely identified using polynomial basis functions. Then a new trajectory is created from the desired trajectory using a feedforward controller based on the inverse of the sparsely identified dynamic model. The effectiveness of the proposed approach is evaluated by a simulation study in which 4 different KUKA robots were tasked to follow 16 distinct trajectories based on ISO 9283 standard. The obtained results show that the proposed method successfully models the dynamic relationship between the desired and the actual trajectory with accuracies above 98.09% when all of the robots are considered. Moreover, the developed feedforward controller improves the trajectory tracking accuracy of industrial robots by at least 91.1% and 94.5% for position and orientation tracking, respectively while providing parsimonious models.

*Keywords:* Industrial Robots, Trajectory Tracking, Feedforward Control, Data Driven Modeling, Sparse Regression

---

## 1. INTRODUCTION

Industrial robots are required to perform several repetitive tasks such as assembly, pick and place, loading and unloading, etc., with a very high accuracy. Moreover, they are planned to replace conventional CNC systems in machining tasks in the near future. However, their relatively low trajectory tracking accuracy during machining tasks is the main hindrance in their widespread adoption (Klimchik et al. (2017)). Therefore, there exists many works in literature which deal with increasing the trajectory tracking accuracy of industrial robots by making the output trajectory as close as to the desired trajectory.

One of the most well known approaches for this is the Computed Torque Control (CTC) in which a precise dynamic model of a robot is assumed to be known. Moreover, to overcome uncertainties adaptive CTC methods have been proposed in literature. An example of this approach is the work by Wang (2016) in which a highly accurate task space trajectory tracking was performed under uncertain dynamics using an adaptive CTC. In another work by Chen et al. (2016), an adaptive fuzzy control algorithm was used with CTC to achieve better trajectory tracking while reducing the effect of uncertainties on the robot.

While CTC methods are effective, obtaining precise dynamic models is not an easy task. Therefore, another popular approach in dealing with repetitive tasks is the utilization of feedforward control. In this type of approach, the disturbances in the system are compensated using

prior knowledge about the task. This is often achieved through the utilization of the inverse plant model. This way a new reference signal can be generated that will make the robot to follow the desired trajectory more accurately. An example of this approach is the work by Camoriano et al. (2016), where the inverse dynamics of a multi axis robot is learned using a semi-parametric method. In their work the rigid body dynamics is used to build a parametric model in addition to an incremental kernel method as a nonparametric model. They show that the proposed method makes the robot to have better trajectory tracking accuracy under different conditions. Another approach used in literature for repetitive tasks is the Iterative Learning Control (ILC) method. In the work by Marchal et al. (2014) a machine vision based tracker was used to improve the tracking accuracy of an industrial robot through ILC of a six Degree of Freedom (DOF) industrial robot during robotic manipulation. In another work by Hsiao and Huang (2017) an ILC based method was proposed for increasing the trajectory tracking accuracy of an industrial robot by following a path repetitively.

Moreover, data driven modeling have become increasingly popular in the recent years due to the abundance of data (Aran and Unel (2018), Alcan et al. (2019), Mumcuoglu et al. (2020)). Some examples of these can be seen in the works by Gao et al. (2017) and Doan et al. (2018) where uncertainties for an industrial robot were modeled and compensated using radial basis functions and neural networks. Some works in literature have also been proposed for learning feedforward control using neural net-

works and Gaussian Process Regression (GPR) methods. In the work by Cuong and Minh (2015) the feedforward control is learned through neural networks during free motion of manipulators and the accuracy of the robot is increased using the proposed method. As for GPR based methods, the trajectory tracking accuracy of an industrial robot equipped with a laser cutter is improved through utilization of a GPR based feedforward control (Wang et al. (2015)). Moreover, Meier and Schaal (2016) proposed a drifting GPR based approach to learn the inverse dynamics of an industrial robot and use it for increasing the tracking accuracy of an industrial robot. As seen, there are a multitude of literature on the effectiveness of data driven methods for learning feedforward control to increase tracking accuracy of industrial robots.

Typically, in the aforementioned works the robotic manipulator is assumed to be open in terms of control, i.e. the user has full access to the control algorithm deployed in the manipulator. However, this is not always the case, especially for industrial robots where the manufacturers have deployed their own control algorithms and the users have no access to the control architecture of the robot. The only thing that the users have access to are the desired trajectories to be followed by the robot. This desired trajectory is typically in the form of the pose of the end effector with six components, where three of them represent the position and the other three represent the orientation of the end effector. Moreover, in an industrial setting, robot manipulators are typically tasked with following a trajectory repeatedly. Thus, the same level of accuracy is expected for these repetitive tasks. Therefore, it is imperative to make use of this information in order to improve the robot's accuracy when tasked with the same work. Motivated by these facts, in this work an approach based on Sparse Identification of Nonlinear Dynamics with Control (SINDYc, Brunton et al. (2016)) is proposed to obtain dynamic models of repetitive tasks and then generate feedforward controllers based on this dynamic model. SINDYc is a data driven method known to provide parsimonious models which are easy to train. To train these models, the ground truth data for trajectory tracking can be obtained from a highly accurate sensor such as a laser tracker. The generated feedforward controller modifies the input trajectory and generates a new one so that the robot follows the desired trajectory more accurately without having to modify the control algorithm of the robot. Therefore, the objective of the proposed approach is to manipulate the reference trajectories in terms of the end effector's pose fed to the robot so that the robot follows a desired trajectory accurately and precisely.

The rest of this paper is structured as follows: In Section 2, the proposed methodology for increasing tracking accuracy is presented in detail. The effectiveness of the proposed approach is validated through a simulation study for four distinct robots in Section 3, followed by the conclusion in Section 4.

## 2. IMPROVED TRAJECTORY TRACKING USING SPARSE REGRESSION

In this section a sparse regression based approach is proposed for increasing the trajectory tracking accuracy

of industrial robots. Typically the desired trajectories to be followed by an industrial robot are given as the pose of the end effector with respect to a reference frame. This pose input contains six components of which the first three denoted as  $x, y,$  and  $z$  are for translation along  $X, Y,$  and  $Z$  axes, respectively and the other three denoted as  $\phi, \theta,$  and  $\psi$  are for orientation around  $X, Y,$  and  $Z$  axes, respectively. This work proposes an approach based on Sparse Identification of Nonlinear Dynamics with Control (SINDYc) proposed by Brunton et al. (2016). The SINDYc algorithm is a data driven approach which can be used to determine the sparse vectors of coefficients that define the dynamics of a physical system in the presence of an external input. In this work we formulate the SINDYc algorithm to obtain a dynamic process model relating the desired and the actual trajectories followed by a robot as follows:

$$P_A^t = \Omega \Delta(P_A^{t-1}) + \Gamma P_D^t \quad (1)$$

where

$$P_A^t = \begin{bmatrix} x_A(t_1) & x_A(t_2) & \cdots & x_A(t_n) \\ y_A(t_1) & y_A(t_2) & \cdots & y_A(t_n) \\ z_A(t_1) & z_A(t_2) & \cdots & z_A(t_n) \\ \phi_A(t_1) & \phi_A(t_2) & \cdots & \phi_A(t_n) \\ \theta_A(t_1) & \theta_A(t_2) & \cdots & \theta_A(t_n) \\ \psi_A(t_1) & \psi_A(t_2) & \cdots & \psi_A(t_n) \end{bmatrix} \quad (2)$$

$$P_D^t = \begin{bmatrix} x_D(t_1) & x_D(t_2) & \cdots & x_D(t_n) \\ y_D(t_1) & y_D(t_2) & \cdots & y_D(t_n) \\ z_D(t_1) & z_D(t_2) & \cdots & z_D(t_n) \\ \phi_D(t_1) & \phi_D(t_2) & \cdots & \phi_D(t_n) \\ \theta_D(t_1) & \theta_D(t_2) & \cdots & \theta_D(t_n) \\ \psi_D(t_1) & \psi_D(t_2) & \cdots & \psi_D(t_n) \end{bmatrix} \quad (3)$$

where  $P_A^t \in R^{6 \times n}$  is the actual trajectory followed by the end effector for  $n$  times steps and it is typically provided by a laser tracker,  $P_A^{t-1} \in R^{6 \times n}$  is the actual trajectory at the previous time step,  $P_D^t \in R^{6 \times n}$  is the control input defining the desired trajectory to be followed by the end effector for  $n$  time steps,  $\Delta(\cdot) \in R^{K \times n}$  is a library containing  $K$  candidate functions of  $P_A^{t-1}$  for  $n$  time steps,  $\Omega \in R^{6 \times K}$  contains the sparse vectors of coefficients for  $K$  candidate functions for each component of the trajectory and  $\Gamma \in R^{6 \times 6}$  is the input matrix.

Using this formulation one can learn the dynamic model ( $\Omega$ ) relating  $P_D$  to  $P_A$  through the sequential thresholded least-squares algorithm (Brunton et al. (2016)). In this algorithm only two parameters need to be provided, namely the sparsification knob denoted as  $\lambda$  and the number of iterations. However, in order to use this algorithm Eq.1 will need be rewritten as follows:

$$P_A^t = \bar{\Omega} \bar{\Delta}(P_A^{t-1}, P_D^t) \quad (4)$$

where  $\bar{\Omega} = [\Omega | \Gamma] \in R^{6 \times (K+6)}$ .

With this formulation  $\bar{\Delta}(\cdot, \cdot) \in R^{(K+6) \times n}$  now contains  $K + 6$  candidate functions of the actual trajectory at the previous time step ( $P_A^{t-1}$ ) along with the control input ( $P_D^t$ ) at the current time step. This way, a dynamic model relating the desired and the actual trajectory followed by a robot can be obtained. Each row of the augmented library  $\bar{\Delta}(\cdot, \cdot)$  represents a candidate function for defining

the relationship between the desired and the ground truth trajectory. There is total freedom in choosing these functions which can be polynomials, trigonometric functions, etc.

Now that the process model is obtained, the next step is to generate a feedforward controller based on the pseudo inverse of this dynamic model as follows:

$$\bar{\Delta}(P_A^{t-1}, P_D^t) = \bar{\Omega}^+ P_A^t \quad (5)$$

Then, assuming the obtained model ( $\bar{\Omega}$ ) is fixed for the desired operation, which is quite true for industrial robots having very high repeatability, one can use the pseudo inverse of ( $\bar{\Omega}$ ) as a feedforward controller to generate a new optimized trajectory as follows:

$$\bar{\Delta}(P_D^{t-1}, \bar{P}_D^t) = \bar{\Omega}^+ P_D^t \quad (6)$$

where  $\bar{P}_D^t \in R^{6 \times n}$  is the optimized trajectory for current time step.

In this work the utilization of 1<sup>st</sup> order polynomials in the augmented library was determined to be sufficient to obtain satisfactory models, therefore the augmented matrix is defined as follows:

$$\bar{\Delta}(P_D^{t-1}, \bar{P}_D^t) = \begin{bmatrix} \mathbf{1}_{1 \times n} \\ P_D^{t-1} \\ \bar{P}_D^t \end{bmatrix} \quad (7)$$

Here, the optimized trajectory ( $\bar{P}_D^t$ ) corresponds to the last 6 rows of  $\bar{\Delta}(P_D^{t-1}, \bar{P}_D^t) \in R^{(K+6) \times n}$ . We should note that in the case of 1<sup>st</sup> order polynomials  $K = 7$ . One can use the new optimized trajectory ( $\bar{P}_D^t$ ) along with the identified process model ( $\bar{\Omega}$ ) to obtain and validate the new output trajectory as follows:

$$\hat{P}^t = \bar{\Omega} \bar{\Delta}(\hat{P}^{t-1}, \bar{P}_D^t) \quad (8)$$

where  $\hat{P}^t \in R^{6 \times n}$  is the estimated output trajectory when the optimized trajectory ( $\bar{P}_D^t$ ) is used at the input of the robot.

The proposed method was coded in MATLAB software and optimized using sequentially thresholded least squares algorithm. The proposed approach can be used with an industrial robot that does not allow any modifications to its control algorithm, as is the case with many of them in the market. The user only needs to modify the reference trajectories which is the only adjustable parameter in closed systems. Therefore, it does not alter the embedded closed-loop control algorithm. Typically, in an industrial setting the robots perform the same task repeatedly, thus one can train models using the proposed method for each task using a highly accurate external sensor. Moreover, a single external tracker is sufficient to train models for multiple robots, thus reducing the need for an external tracker for each robot. This way, the proposed method can be used to generate optimized trajectories for each task individually for each robot. The overall flowchart of the proposed method for increasing the trajectory tracking accuracy via sparse regression is illustrated in Fig. 1.

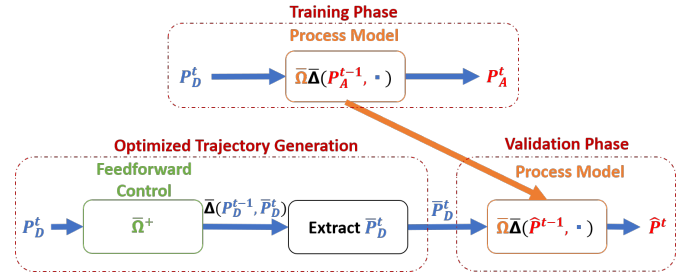


Fig. 1. The proposed SINDYc based approach

### 3. SIMULATION RESULTS AND DISCUSSIONS

In order to evaluate the effectiveness of the proposed approach we assume the availability of one to one correspondences between the desired and the actual trajectory followed by an industrial robot. Typically, the desired trajectory for industrial robots are obtained via CAD/CAM software which provide the set points to be followed by the robot. As for the actual trajectory followed by the robots, they can be obtained through highly accurate laser trackers. In order to perform a high fidelity simulation, the dataset from our previous work (Bilal et al. (2020)) was used as a basis for the simulations of this work. This dataset contains 16 distinct trajectories followed by a KUKA KR240 R2900 ultra robot's end effector based on ISO 9283 standard. The end effector was tasked to move to 5 specific points in each of these trajectories while continuously changing its orientation. The trajectory of the end effector was tracked in realtime using a Leica AT960 laser tracker and 63551 trajectory points were acquired as shown in Fig. 2. Then, this data was used as the desired trajectory to be followed by various industrial robots in simulation environment. In this work, various KUKA industrial robots were considered where their repeatability provided by the manufacturer was utilized as given in Table 1. To perform the simulations in MATLAB, filtered white noise, i.e. colored noise, was added to the trajectories obtained through the ISO 9283 experiment by setting the white noise's mean and variance based on the robots' specifications given by Table 1.

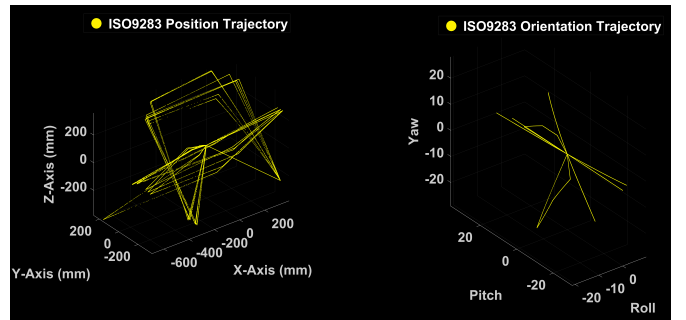


Fig. 2. The position and orientation trajectories used in simulations based on ISO 9283 standard

For the training and validation dataset, we used only 30% of the data for training the models. We should note that by using only 30% of the data for training and the remaining 70% for validation we are still getting successful models. The process model was identified by setting the sparsification parameter ( $\lambda$ ) and the number of iterations

Table 1. Specifications of the KUKA industrial robots used in the simulations

Robot Type	Robot ID	Repeatability (mm)
KR 3 R540	1	$\pm 0.02$
KR 22 R1610-2	2	$\pm 0.04$
KR 70 R2100	3	$\pm 0.05$
KR 240 R2900 ultra	4	$\pm 0.06$

to 0.002 and 10, respectively in the sequential thresholded least-squares algorithm. As for learning the feedforward control, it was performed based on the pseudo inverse of the process model as described in Section 2. Training the process and feedforward models required only 0.012458 and 0.00045 seconds, respectively<sup>1</sup>. The accuracy of the obtained process model between the desired and actual trajectory of the robot for each of the individual axis are given in Table 2 using the best fit criterion. As seen, the proposed method provides process models with accuracies of at least 98.09% when validated on the remaining 70% of the dataset.

Table 2. Accuracy of the process models obtained using the SINDYc approach

Accuracy (%)	Robot ID			
	1	2	3	4
$x$	98.96	98.93	98.91	98.89
$y$	98.94	98.88	98.86	98.83
$z$	98.95	98.90	98.87	98.85
$\phi$	98.36	98.36	98.35	98.35
$\theta$	98.12	98.14	98.12	98.12
$\psi$	98.10	98.10	98.11	98.09

To quantify the performance of the proposed method for improving trajectory tracking, the maximum, mean, minimum and the standard deviation of absolute errors between the desired and the estimated trajectories of the robots were utilized. The absolute errors for position and orientation were calculated as follows:

$$E_T = \sqrt{E_x^2 + E_y^2 + E_z^2} \quad (9)$$

$$E_R = \sqrt{E_\phi^2 + E_\theta^2 + E_\psi^2} \quad (10)$$

where  $E_x$ ,  $E_y$ ,  $E_z$ ,  $E_\theta$ ,  $E_\phi$ , and  $E_\psi$  are the absolute errors between the desired and the estimated trajectory, and  $E_T$  and  $E_R$  are the absolute errors for position and orientation trajectories.

The effectiveness of the obtained feedforward controllers is observed by the results plotted in Figures 3 to 6, which show the maximum, mean, minimum and standard deviation of absolute errors, respectively when validated on the remaining 70% of the dataset. The proposed method was evaluated on the aforementioned 4 KUKA robots and the black lines represent the original absolute errors between the desired and the output trajectory of the robot's end effector and the red lines represent the results obtained through the proposed approach. Moreover, the absolute tracking errors for the last 200 samples of X, Y and Z axes are shown in Fig. 7 for the KR 240 robot. As observed, the proposed method provides consistent improvements for each individual axis.

<sup>1</sup> Tested on a laptop with Intel Core i7-10750H CPU with 16 GB of RAM.

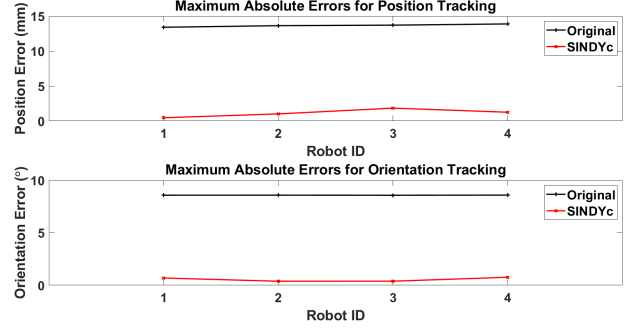


Fig. 3. Maximum absolute errors for position and orientation tracking

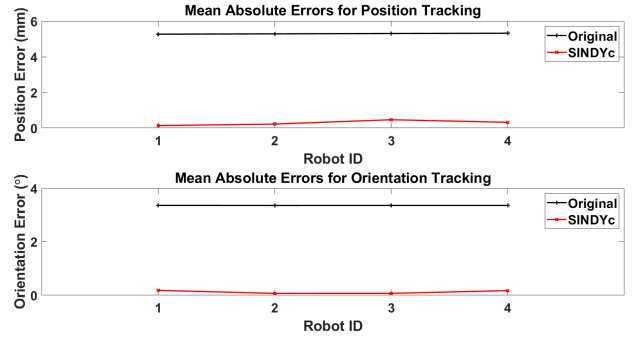


Fig. 4. Mean of absolute errors for position and orientation tracking

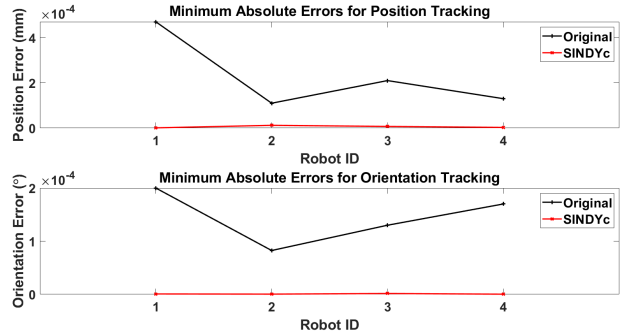


Fig. 5. Minimum absolute errors for position and orientation tracking

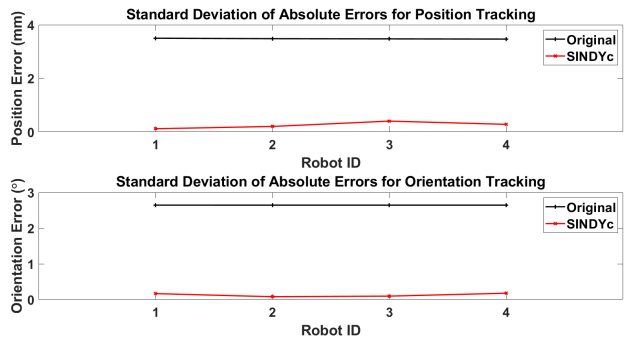


Fig. 6. Standard deviation of absolute errors for position and orientation tracking

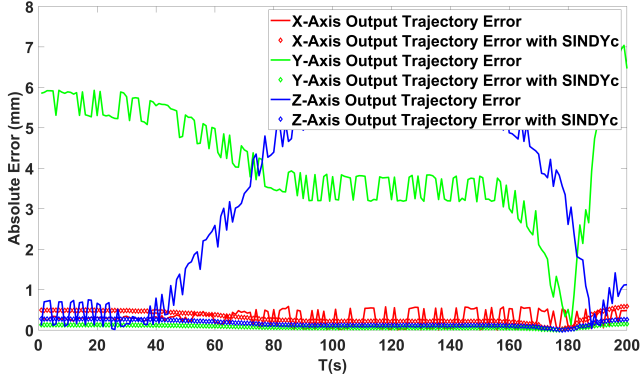


Fig. 7. Absolute trajectory tracking errors for X, Y and Z axes

The magnitudes of the absolute errors are given in Tables 3 to 5. As seen from these results, the proposed approach reduces the absolute errors significantly for all of the robots which have varying position accuracy and precision. As seen from Table 3, the proposed method reduces the maximum position errors by at least 86.6% and upto 96.6% when all of the robots are considered. As for the maximum orientation errors, they are improved by at least 91.1% and upto 95.6%. Moreover, as observed from Table 4, the proposed approach reduces the mean of position errors by at least 91.1% and upto 97.4%. The proposed method reduces the mean of orientation errors by at least 94.5% and upto 97.9%. As for reducing the minimum of position errors, from Table 5 it is seen that the proposed approach reduces them by at least 89.1% and upto 99.8%, while the orientation errors are reduced by at least 98.9% and upto 99.8% when all of the robots are considered.

Table 3. Maximum of absolute errors for position and orientation tracking results

Robot Type	Position Errors (mm)		Orientation Errors (°)	
	Original	SINDYc	Original	SINDYc
KR 3 R540	13.3935	0.4593 (96.6%)	8.5750	0.6876 (91.9%)
KR 22 R1610-2	13.6191	1.0139 (92.6%)	8.5776	0.3808 (95.6%)
KR 70 R2100	13.6928	1.8375 (86.6%)	8.5695	0.3858 (95.5%)
KR 240 R2900 ultra	13.8669	1.2535 (90.9%)	8.5849	0.7600 (91.1%)

The (%) shows the improvement percentage.

Table 4. Mean of absolute errors for position and orientation tracking results

Robot Type	Position Errors (mm)		Orientation Errors (°)	
	Original	SINDYc	Original	SINDYc
KR 3 R540	5.2706	0.1365 (97.4%)	3.3619	0.1824 (94.5%)
KR 22 R1610-2	5.2896	0.2254 (95.7%)	3.3604	0.0693 (97.9%)
KR 70 R2100	5.3084	0.4679 (91.1%)	3.3609	0.0710 (97.8%)
KR 240 R2900 ultra	5.3227	0.3224 (93.9%)	3.3606	0.1735 (94.8%)

The (%) shows the improvement percentage.

Table 5. Minimum of absolute errors for position and orientation tracking results

Robot Type	Position Errors (mm)		Orientation Errors (°)	
	Original	SINDYc	Original	SINDYc
KR 3 R540	0.00047	1.06E-06 (99.8%)	0.0002	6.55E-07 (99.7%)
KR 22 R1610-2	0.00011	1.21E-05 (89.1%)	8.24E-05	3.28E-07 (99.6%)
KR 70 R2100	0.00021	7.23E-06 (96.5%)	0.00013	1.49E-06 (98.9%)
KR 240 R2900 ultra	0.00013	2.86e-06 (97.8%)	0.00017	2.23E-07 (99.8%)

The (%) shows the improvement percentage.

Moreover, the proposed method improves the robots' precision as well as seen from the standard deviation of absolute errors given in Table 6. From these results it is observed that the proposed approach reduces the standard deviation of errors by at least 88.3% and upto 96.5% for position and by at least 93.2% and upto 96.9% for orientation tracking, respectively.

Table 6. Standard deviation of absolute errors for position and orientation tracking results

Robot Type	Position Errors (mm)		Orientation Errors (°)	
	Original	SINDYc	Original	SINDYc
KR 3 R540	3.4991	0.1200 (96.5%)	2.6443	0.1665 (93.7%)
KR 22 R1610-2	3.4819	0.2057 (94.1%)	2.6437	0.0796 (96.9%)
KR 70 R2100	3.4745	0.4048 (88.3%)	2.6438	0.0964 (96.3%)
KR 240 R2900 ultra	3.4668	0.2840 (91.8%)	2.6436	0.1779 (93.2%)

The (%) shows the improvement percentage.

The process model parameters obtained via SINDYc are given in Table 7 for the KR240 robot. As observed, the proposed method provides sparse coefficients where most of the parameters are inactive i.e. they have zero coefficients. The same pattern is observed for the model obtained for the feedforward control as shown in Table 8. From this table it is observed that most of the coefficients are close to zero. Therefore, the proposed approach provides parsimonious models where only the most effective parameters are active.

Table 7. Process model obtained via SINDYc for KR240 robot

Parameters	$x_A^t$	$y_A^t$	$z_A^t$	$\phi_A^t$	$\theta_A^t$	$\psi_A^t$
1	0	-0.0052	0	0	0	0
$x_A^{t-1}$	0	0	0	0	0	0
$y_A^{t-1}$	0	0	0	0	0	0
$z_A^{t-1}$	0	0	0	0	0	0
$\phi_A^{t-1}$	0	-0.01023	0.029123	0.022664	0	0
$\theta_A^{t-1}$	0	0.014491	-0.00348	0	0	0
$\psi_A^{t-1}$	0.014711	0.018288	0.009626	0.00392	0	0.015931
$x_D^t$	0.999998	0.014995	0.00892	0	0	0
$y_D^t$	0	1.000076	0.014739	0.008724	0	0
$z_D^t$	0	0	0.999589	0.014653	0.008904	0
$\phi_D^t$	0	0.008675	-0.02887	0.977815	0.015352	0.009448
$\theta_D^t$	0.008928	-0.01524	0.003025	0	0.999893	0.014998
$\psi_D^t$	0	-0.01004	-0.00984	-0.00381	0	0.984303

Table 8. Feedforward control obtained via SINDYc for KR240 robot

Parameters	$\bar{x}_D^t$	$\bar{y}_D^t$	$\bar{z}_D^t$	$\bar{\phi}_D^t$	$\bar{\theta}_D^t$	$\bar{\psi}_D^t$
$x_D^t$	0.99979	-0.01525	-0.00883	0.00021	7.54E-05	-0.00024
$y_D^t$	-0.00027	0.99932	-0.01478	-0.00851	0.00026	-0.00021
$z_D^t$	-6.20E-05	0.00019	0.99902	-0.0157	-0.00866	0.00013
$\phi_D^t$	8.63E-05	-0.00902	0.02885	1.02172	-0.01594	-0.00963
$\theta_D^t$	-0.00893	0.01521	-0.0033	-0.00012	1.00014	-0.01523
$\psi_D^t$	-0.00024	0.00986	0.00979	0.00366	-0.00014	1.01564

#### 4. CONCLUSION

In this work a sparse regression based feedforward control approach for increasing the trajectory tracking accuracy of industrial robots is proposed. The proposed method is based on Sparse Identification of Nonlinear Dynamics with Control (SINDYc). First, the SINDYc algorithm was used to obtain a dynamic model relating the desired trajectory to the output one, where the desired trajectory is provided by the user and the output trajectory is obtained using a laser tracker. Then, a feedforward controller was developed based on the inverse of the process model which takes in the desired trajectory and provides an optimized one. This optimized trajectory was used along with the identified process model in order to show that the robot follows the original desired trajectory more accurately when the optimized trajectory is used at the input of the robot.

The proposed method was evaluated in a high fidelity simulation where the data obtained via a laser tracker while tracking an industrial robot's end effector was used as a basis for the simulations. The simulation data consisted of 16 distinct trajectories based on ISO 9283 standard where both the position and orientation of the end effector of the robot changed continuously along the trajectory. To evaluate the effectiveness of the proposed approach, 4 KUKA industrial robots with varying accuracies and precisions were considered. From the results it is seen that the proposed approach provided process models relating the desired and actual trajectories with accuracies above 98.09%. Moreover, the developed feedforward controllers decreased the mean of absolute errors during trajectory tracking by at least 91.1% for position and by 94.8% for orientation tracking when all of the robots were considered. Additionally, the standard deviation of absolute errors were reduced at least by 88.3% and 93.2% for position and orientation tracking, respectively. It should be noted that even with less estimation data (30% of the total dataset), the proposed technique works without any problem. As shown, the proposed method significantly increases the trajectory tracking accuracy and precision of various industrial robots. Moreover, obtaining the process and feedforward models via the proposed method is remarkably fast while providing parsimonious models.

#### REFERENCES

Alcan, G., Yilmaz, E., Unel, M., Aran, V., Yilmaz, M., Gurel, C., and Koprubasi, K. (2019). Estimating soot emission in diesel engines using gated recurrent unit networks. *IFAC-PapersOnLine*, 52(5), 544–549.

Aran, V. and Unel, M. (2018). Gaussian process regression feedforward controller for diesel engine airpath. *International Journal of Automotive Technology*, 19(4), 635–642.

Bilal, D.K., Unel, M., and Tunc, L.T. (2020). Improving vision based pose estimation using lstm neural networks. In *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, 483–488. IEEE.

Brunton, S.L., Proctor, J.L., and Kutz, J.N. (2016). Sparse identification of nonlinear dynamics with control (sindy). *IFAC-PapersOnLine*, 49(18), 710–715.

Camoriano, R., Traversaro, S., Rosasco, L., Metta, G., and Nori, F. (2016). Incremental semiparametric inverse dynamics learning. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 544–550. IEEE.

Chen, Y., Ma, G., Lin, S., Ning, S., and Gao, J. (2016). Computed-torque plus robust adaptive compensation control for robot manipulator with structured and unstructured uncertainties. *IMA Journal of Mathematical Control and Information*, 33(1), 37–52.

Cuong, N.D. and Minh, T.X. (2015). Learning feedforward control for a two-link rigid robot arm. *International Journal of Electronics and Electrical Engineering*, 3(4), 279–284.

Doan, Q.V., Le, T.D., Le, Q.D., and Kang, H.J. (2018). A neural network-based synchronized computed torque controller for three degree-of-freedom planar parallel manipulators with uncertainties compensation. *International Journal of Advanced Robotic Systems*, 15(2), 1729881418767307.

Gao, W., Shi, J., Wang, W., and Sun, Y. (2017). Research on sliding mode control for robotic manipulator based on rbf neural network. In *2017 29th Chinese Control And Decision Conference (CCDC)*, 4934–4938. IEEE.

Hsiao, T.S. and Huang, P.H. (2017). Iterative learning control for trajectory tracking of robot manipulators. *International Journal of Automation and Smart Technology*, 7(3), 133–139.

Klimchik, A., Ambiehl, A., Garnier, S., Furet, B., and Pashkevich, A. (2017). Efficiency evaluation of robots in machining applications using industrial performance measure. *Robotics and Computer-Integrated Manufacturing*, 48, 12–29.

Marchal, P.C., Sörnmo, O., Olofsson, B., Robertsson, A., Ortega, J.G., and Johansson, R. (2014). Iterative learning control for machining with industrial robots. *IFAC Proceedings Volumes*, 47(3), 9327–9333.

Meier, F. and Schaal, S. (2016). Drifting gaussian processes with varying neighborhood sizes for online model learning. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 264–269. IEEE.

Mumcuoglu, M.E., Alcan, G., Unel, M., Cicek, O., Mutluergil, M., Yilmaz, M., and Koprubasi, K. (2020). Driver evaluation in heavy duty vehicles based on acceleration and braking behaviors. In *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, 447–452. IEEE.

Wang, C., Zhao, Y., Chen, Y., and Tomizuka, M. (2015). Nonparametric statistical learning control of robot manipulators for trajectory or contour tracking. *Robotics and Computer-Integrated Manufacturing*, 35, 96–103.

Wang, H. (2016). Adaptive control of robot manipulators with uncertain kinematics and dynamics. *IEEE Transactions on Automatic Control*, 62(2), 948–954.