Data-driven modelling of touch-down bearing forces

Benedikt SCHÜßLER [a], Bastian TIGGES [b], Tuomas TIAINEN [c], Stephan RINDERKNECHT [d]

a Technical University of Darmstadt – Institute for mechatronic Systems, Otto-Berndt-Straße 2, 64287 Darmstadt, Germany,
schuessler@ims.tu-darmstadt.de

b Technical University of Darmstadt – Institute for mechatronic Systems, Otto-Berndt-Straße 2, 64287 Darmstadt, Germany

c Aalto University School of Engineering - Department of Mechanical Engineering, Otakaari 24, 02150 Espoo, Finland, tuomas.tiainen@aalto.fi

d Technical University of Darmstadt – Institute for mechatronic Systems, Otto-Berndt-Straße 2, 64287 Darmstadt, Germany

Abstract

To reduce CO2 emissions, the share of renewable energies in the grid increases. At the same time, many sectors like the transport and the building sector are changing to be powered by electricity. Especially electric cars demand high peaks in current from the grid. Storage is needed to balance demand and supply of electric energy. Flywheels can be part of the solution as they can be charged and discharged with high power and do not suffer from losing significant capacity even after thousands of cycles. Minimum loss of energy is crucial for a flywheel therefor active magnetic bearings (AMB) are used. If a malfunction of the AMB occurs the rotor falls into a touch-down bearing (TDB). To decide whether further maintenance in case of a drop-down event is needed information about the forces stressing the TDB is important. To avoid costs for physical sensors soft sensors are a suitable solution. In this research, a data-driven soft sensor based on recurrent neural networks is created to calculate the forces during the drop-down event. As input data only the position of the rotor is used. A test rig with physical sensors applied to every TDB supplies the force data to train, validate, and test the soft sensor model. Three different network architectures are compared. The results show that the sensor can calculate whether the rotor hits a TDB and is also capable of predicting the peaks in the force signal.

Keywords: *soft sensor; AI; neural network; touch down bearing; backup bearing*

1. Introduction

To tackle climate change, the share of renewable energies in the grid increases. At the same time, the overall demand for electrical power rises, and consumers like charging stations for electric cars request high peaks in current from the grid. To balance production and demand storage is needed. Flywheels represent a suitable storage system for several occasions as they can be charged and discharged with high power and do not suffer from losing significant capacity even after thousands of cycles. To lower the loss of energy within the storage the rotor normally runs in an evacuated room and active magnetic bearings (AMB) are used to support the rotor. In the case of a malfunction of the AMB touch-down bearings (TDB) are applied to prevent a destructive rotor-stator contact elsewhere in the system. The design of TDB is a research topic due to the challenging conditions. TDB must endure high forces and rotational speeds caused by the rotor but also by working in evacuated areas which complicates lubrication and hinders heat dissipation. If a drop-down event occurs, one needs to decide whether maintenance of the TDB needs to be conducted. One way would be to disassemble the flywheel and do a visual check of the bearings. Another approach can be to check the forces over time that stressed the TDB during the drop-down event. The forces can be measured by using piezo electric sensors or strain gauges. In both ways, the sensors need to be applied to every TDB unit. When using planetary TDB this can mean that at least six sensors in the overall flywheel are required, which drives up the cost of the system.

Soft sensors or virtual sensors can overcome this problem. They use existing data and calculate a desired output variable. Soft sensors can be divided into model-driven and data-driven soft sensors (Jiang et al., 2021). The model-driven approach can be used if the knowledge about the physical relations between input and out variables is well known and understood. If this is not the case data-driven soft sensors come into play. They can be based on different methods like partial least squares (PLS) (Kano and Nakagawa, 2008), principal component analysis (PCA) (Zhang et al., 2020), or support vector machines (SVM) (Yan et al., 2004). In the last few years, soft sensors based on neural networks became more popular (Yuan et al., 2020). When it comes to the prediction of time-related data recurrent neural networks (RNN) show a good ability to understand and utilize the information which is embedded in the time series (Du and Swamy, 2019). A drawback of an RNN is the vanishing or exploding gradient problem especially when longer series of data is computed (Yuan et al., 2020). To overcome this problem, the long short-term memory (LSTM) architecture was developed by Hochreiter and Schmidhuber (1997).

The LSTM architecture was applied to several soft sensor applications. ElSaid et al. (2016) used an LSTM network to predict the vibration of an aircraft engine up to ten seconds into the future by using several input variables measured already by the engine. Bellone et al. (2020) showed that an LSTM is capable of estimating states like the NOx values of a combustion engine of a car as a real-time application. In the work of Miettinen et al. (2021) the displacement of rotors used in the paper industry was estimated by a bidirectional LSTM using the bearing reaction forces.

This paper presents an approach where a soft sensor based on different neural networks was trained to predict the forces on the TDB.

2 Method

This section describes the test rig to explain the overall setup and to show where the data used for the NN is measured. Afterward, the preprocessing of the data is explained. The different network architectures are further described and the methodology to analyze the results is explained.

2.1 Test Rig

To test the planetary TDB of the flywheel, a special TDB test rig for drop-down experiments in planetary TDB has been built at the Institute for Mechatronic Systems at the Technical University of Darmstadt. Even if the TDB test rig is built as a robust inner rotor system to withstand multiple drop-downs it has similar rotor dynamic properties as the flywheels (*Proceedings of ISMB 16*, 2018; Quurck et al., 2017; Quurck et al., 2018). The rotor has a mass of 18 kg and a maximal rotational speed of 20,000 rpm. This results in a surface velocity of the rotor at the planetary TDB of 230 m/s. Figure 1 shows a partial section view of the test rig without its base and a detailed view of the planetary TDB itself.

As the flywheels, the TDB test rig is levitated magnetically in all axes and driven by a permanent magnetic synchronous machine. For a drop-down experiment, the radial AMB are switched off, while the axial AMB remains active. This is because in the flywheel the axial levitation is performed by a passive permanent magnetic bearing which is not expected to fail. Since the purpose of the TDB test rig is to test the planetary TDB till failure the system has a secondary TDB which only gets in contact with the rotor if the planetary TDB has failed. The system is equipped with eddy current position sensors from eddylab, which are needed for the AMB and the drop-down evaluation. For a deeper analysis of the drop-down experiments, the system is equipped with multiple further sensors. For example, the force is measured at every TDB unit. In all, in this paper investigated drop-down experiments one TDB unit in each plane is equipped with a piezoelectric force sensor from pcb of type 211B. The charge amplifier is of type 5073A411 from Kistler. At the remaining TDB units, the forces in radial and tangential direction are measured with structurally integrated strain gauges. For the data acquisition, a system from National Instrument is used.

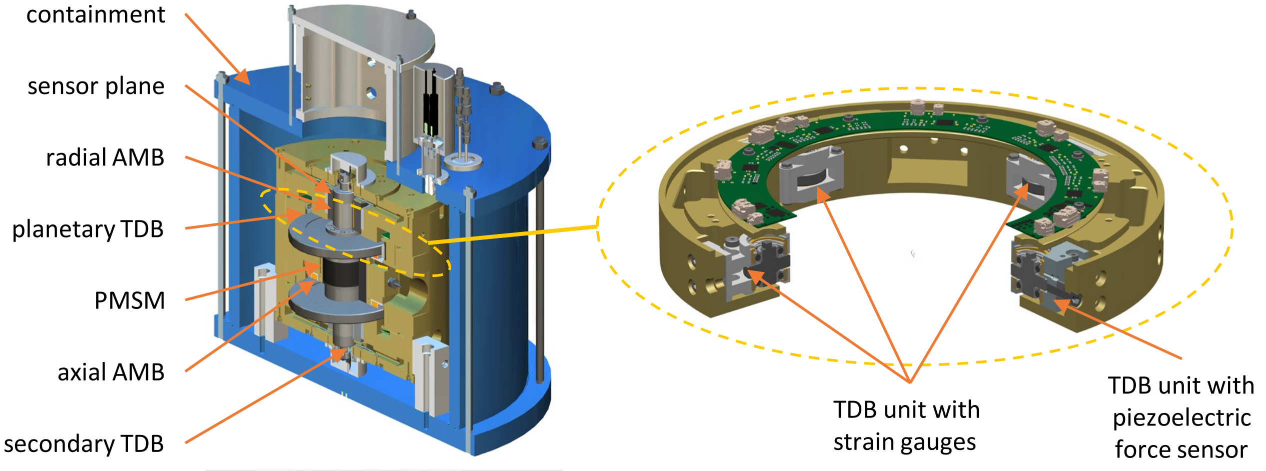

Figure 1 labeled partial section view of the TDB test rig without the base and a detailed view of the planetary TDB with 3 TDB units with strain gauges and one TDB unit with a piezoelectric force sensor.

2.2 Modeling

The test series that was used for the training, validation, and testing of the neural network contains 57 drop-down event experiments. The rotational speed when the drop-down was initiated varies between 2500 and 20,000 rpm. The force data given by the strain gauges were sampled at 20 kHz and the position data was sampled at 10 kHz. The length of one drop-down experiment varies with the initial rotational speed and goes up to 300 s, the higher peaks in force on the TDBs normally occur during the first 100 s. This leads to a maximum amount of 6,000,000 samples per sensor signal per test series. For the modelling the goal was to use only sensor signals for the soft sensor which are already available in flywheels. Therefore, even if the TDB test rig is equipped with multiple sensors, only the position measurement is used. For the training, validation, and testing 23 test series were used, 14 for training, five for validation, and four for testing.

The main usage of the soft sensor is to calculate the forces after a drop-down event to then decide whether maintenance is required. This also means that the sensor does not need to be real-time capable. The position data of the four sensors were fed in sequences into the soft sensor and the output of the soft sensor is a sequence of force data with the same length. It was investigated whether it is beneficial building one soft sensor that estimates the force on every TDB or build one soft sensor for each TDB. As the in- and output sequences of data are used the task can be described as a sequence-to-sequence regression task. The length of the input and output sequence was changed in length between 512 and 4096 samples to figure out what gives the best results. To increase the amount of training data the stride length between two sequences which are extracted from the test series was not equal to the length of the sequences. The stride length was changed in different tests from 32 samples to 1024 samples. Next to gaining more training data, the goal was to utilize high peaks in the force data in more than one sequence. As peaks in the force data can occur in more than one sequence it is important to not pick sequences randomly and distribute them to the training, validation, and test dataset. This would lead to a situation where the same data appears in training and testing. To avoid this case the different experiments were distributed into training, validation, and test data. Only 23 of the 57 experiments were chosen because the other ones were conducted with a lower rotational speed of the rotor at the drop-down event. To reduce computation effort sequences without any peaks in force were deleted. This was done by comparing the maximum force data of a sequence to a threshold. As there is noise on the force data this threshold was set at 500 N. As described before the sampling rate for the position data and the force data measured by the strain gauges are not the same. In this case, up-sampling for the position data was conducted. This was achieved by adding values between every other position data point by linear interpolation. In Figure 2 one can see an example sequence of the in- and output data. On the left, the orbit of the rotor during an example sequence is shown for the upper sensor plane. The green and red lines determine the part of the trajectory where the rotor hits the TDB and a force is measured. The measured force during the sequence can be seen on the right side of the figure. In this case, there is an offset on the sensor signal of about 100 N and some noise with an amplitude of around 50 N. The peak in Force reaches nearly 3200 N. The peak has a length of around 50 samples which represents 2.5 ms.

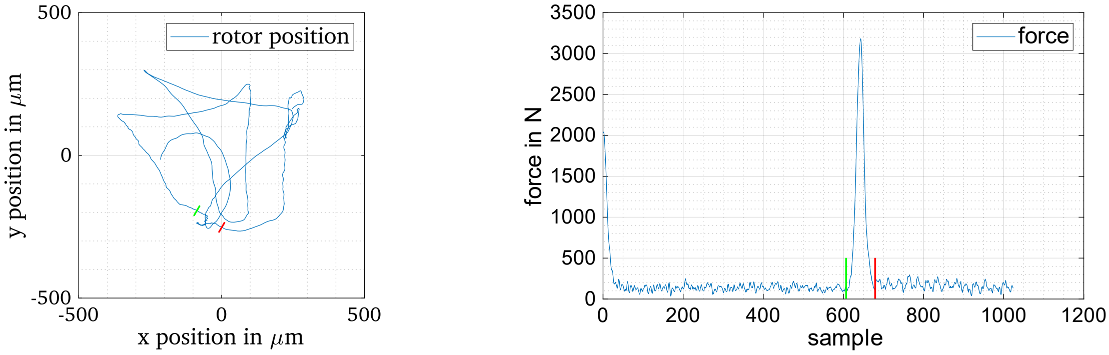

Figure 2 left: rotor trajectory in the upper sensor plane; right: measured normal force on TDB in the upper plane.

As the task is a sequence-to-sequence regression where the relation in time between different data points is important RNN were chosen. More specifically the LSTM and the gated recurrent unit (GRU) architecture were compared. In an RNN the output of one cell of the network is fed into the network in the next timestep again. This can lead to a vanishing or exploding gradient problem when it comes to training the network. Both the LSTM and the GRU architecture tackle this problem by adding a path where information can flow without being multiplied by any weights which could lead during the training to the vanishing or exploding gradient problem. The GRU architecture needs less computational effort which can be beneficial if a real-time application is on hand (Han et al., 2023). The bidirectional LSTM architecture is an addition to the normal LSTM architecture. Here, the input sequence is computed forward through and network and backward. In real-time applications, this can be crucial as a leg in time for the prediction appears. The number of layers and the number of hidden states were changed during the trials. After the RNN layers, fully connected layers were added to the network. In the end, the loss was determined. Here, the mean absolute error (MAE), the mean squared error (MSE), and the mean cubic error (MCE) were chosen. For the training of the networks, the Adam solver was used with a learning rate decreasing over time. The validation dataset was evaluated four times per epoch. The training stopped when the loss of the validation dataset did not decrease five times in a row.

To compare the results of the different architectures two methods were used. First, the overall root mean cubic error (RMCE) between the predicted values and the measured values from the test rig was computed according to (*1*). Here, n represents the total number of sequences and s the total number of samples per sequence. The RMCE was chosen because in the calculation of the lifetime of a grooved ball bearing the force affects the lifetime with the third power (Childs, 2019).

$$RMCE = \frac{1}{n\,s}\sqrt[3]{\sum_{1}^{n}\sum_{1}^{s}|(F_{measured_{n,s}} - F_{predicted_{n,s}})|^3} \tag{1}$$

Next to the RMCE further attention was paid to the height of the peaks in force. Therefore, the peaks were categorized into different groups according to their height. Every group has a range of 500 N, with the first group going from 500 to 1000 N. The number of groups depends on the height of the highest peak of the test data. Then, the peaks in the test data in every sequence were localized. The height of the peak and sample number of the location in the sequence were stored. Afterward, in the corresponding predicted sequence, the highest predicted value around the stored location was searched. For every group of peaks, the root mean squared error (RMSE) was computed by comparing the heights of the measured peaks from the test rig with the heights of the predicted peaks. Figure 3 shows an exemplary diagram of the accuracy of a network for different groups of peaks. In this case, the calculated RMSE for each group was divided by the force value of the group to gain a deviation in percentage. As the groups have a width of 500 N the median of the group boundaries was used to calculate the percental deviation. In the measurement shown in Figure 3, there was no peak with a value between 3500 and 4000 N and therefore there is no bar visible.

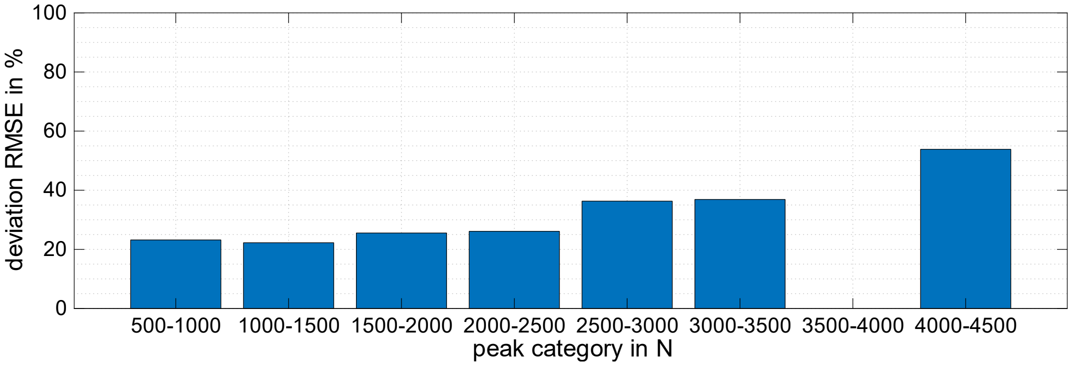

Figure 3 example of the accuracy in predicting peaks of different heights in percentage.

3 Results

Different architectures were compared on the data of the test rig. For the comparison the sequence length of the in- and output data was set to 1024 samples and the stride length between two sequences to 512 samples. The architectures can be seen in Table 1. "3 x 512 LSTM" means that three LSTM layers with 512 hidden units are used. "1 x 128 FC" represents one fully connected (FC) layer with 128 neurons. For every network, an MSE regression layer was added.

Table 1 network architectures

LSTM 1	LSTM 2	BiLSTM 1	BiLSTM 2	GRU 1	GRU 2
3 x 512 LSTM 1 x 128 FC 1 x 64 FC 1 x 32 FC 1 x 16 FC	3 x 256 LSTM 1 x 256 FC 1 x 128 FC 1 x 64 FC 1 x 32 FC 1 x 16 FC 1 x 8 FC	3 x 512 BiLSTM 1 x 128 FC 1 x 64 FC 1 x 32 FC 1 x 16 FC	3 x 256 BiLSTM 1 x 256 FC 1 x 128 FC 1 x 64 FC 1 x 32 FC 1 x 16 FC 1 x 8 FC	3 x 512 GRU 1 x 128 FC 1 x 64 FC 1 x 32 FC 1 x 16 FC	3 x 256 GRU 1 x 256 FC 1 x 128 FC 1 x 64 FC 1 x 32 FC 1 x 16 FC 1 x 8 FC
MSE regression layer					

In Figure 4, an example sequence for the force on one TDB is shown. The sequence is 1024 samples long which represents 51.2 ms. The force on the TDB measured in the test rig is visualized in blue and the force predicted by the network in orange. One can observe that especially the positions of the peaks in force are accurately predicted by the network. This means that the network can predict whether contact between the rotor and the TDB occurred or not. The heights of the peaks by trend seem to be predicted a bit lower compared to the measured ones. On the right side of Figure 4 the accuracy of the network for predicting peaks in different groups is visualized. The deviation is given in percentage as it was shown in Chapter 2.2. It is noticeable that the accuracy rises when the height of the peaks decreases. What needs to be mentioned is that the number of peaks that could be used to calculate the deviation in the prediction was rather low for the groups with high peaks. The value for the group from 4000 to 4500 N was only calculated on the one peak shown in the plot on the right. For the groups 2500 to 3000 N and 3000 to 3500 N, less than ten peaks were available.

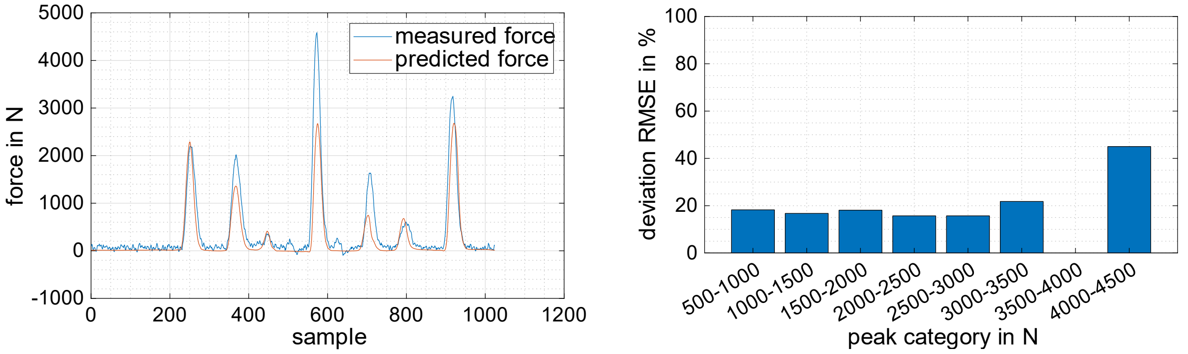

Figure 4 left: predicted (using 1. BiLSTM network) and measured force sequence of the test data; right: peak accuracy in percentage.

To compare the different architectures the overall RMCE was computed as described in Chapter 2.2. At the same time, the percental accuracies as shown in Figure 4 were computed for each network on the test data. Afterward, the percental accuracies for all six different architectures were visualized in one boxplot shown in Figure 5. The percental accuracy for the group of peaks from 4000 to 4500 N in Figure 4 of around 45 % can be spotted as an outlier value in Figure 5 in the box of the first BiLSTM network. On the right side of Figure 5, the overall RMCE for each network architecture is visualized. Here, one can see that the lowest RMCE values are achieved by the two BiLSTM networks.

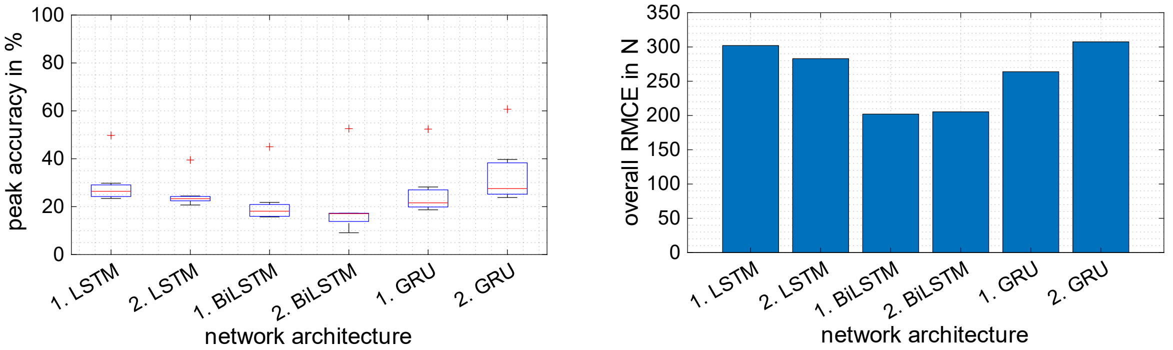

Figure 5 left: boxplot of the different peak accuracies of each network; right: overall RMCE of each network

4 Conclusion

This paper investigated utilizing soft sensors to measure the force on planetary TDB during drop down events. The data was collected on a test rig which was built for research on different planetary TDB designs. As input data for the soft sensor only the position data of the rotor in the two sensor planes was used. For the calculation of the force signal, different RNN were tested and compared by their overall RMCE and their accuracy in predicting peaks of the force signal on a test data set. Here, the accuracy of the network based on a BiLSTM structure performed best. The bidirectional path leads to leg in prediction which is not a problem in this case, because this is not a real-time application. An accuracy of predicting the heights of the peaks of around 20 % can be achieved for lower peaks. In these results, the accuracy for higher peaks goes down to 40 %. As the number of peaks with this height in test data was very low this value is less meaningful. The motivation of this work was to measure the force on the TDB to be able to take decisions on whether maintenance is necessary after a drop-down event. Therefore, a future research question could be, which force impact leads to damages in the system that need to be fixed before running the system again. In this work, the networks were trained on data measured on a test rig. In the future, it could be beneficial to use data that is gained from a simulation of the test rig for training and then testing the network on measured data from the actual test rig. This could be useful for developing flywheels where no sensors for the force on the TDB are applied.

Acknowledgment

This research was funded by the German Federal Ministry for Economic Affairs and Energy, grant numbers 03ET6064A and 03EI3000A.

References

Bellone M, Faghani E and Karayiannidis Y (2020) Comparison of CNN and LSTM for Modeling Virtual Sensors in an Engine. In: *SAE Technical Paper Series:* APR. 21, 2020. SAE International400 Commonwealth Drive, Warrendale, PA, United States.

Childs PRN (2019) *Mechanical design engineering handbook*. Oxford, Cambridge, MA: Butterworth-Heinemann.

Du K-L and Swamy MNS (2019) *Neural Networks and Statistical Learning*. London: Springer London.

ElSaid A, Wild B, Higgins J, et al. (2016) Using LSTM recurrent neural networks to predict excess vibration events in aircraft engines. In: *2016 IEEE 12th International Conference on e-Science:* Baltimore, MD, USA, 23.10. - 27.10., pp. 260–269. IEEE.

Han S, Yoon K, Park G, et al. (2023) Hybrid State Observer Design for Estimating the Hitch Angles of Tractor-Multi Unit Trailer. *IEEE Transactions on Intelligent Vehicles* 8(2): 1449–1458.

Hochreiter S and Schmidhuber J (1997) Long short-term memory. *Neural computation* 9(8): 1735–1780.

Jiang Y, Yin S, Dong J, et al. (2021) A Review on Soft Sensors for Monitoring, Control, and Optimization of Industrial Processes. *IEEE Sensors Journal* 21(11): 12868–12881.

Kano M and Nakagawa Y (2008) Data-based process monitoring, process control, and quality improvement: Recent developments and applications in steel industry. *Computers & Chemical Engineering* 32(1-2): 12–24.

Miettinen J, Tiainen T, Viitala R, et al. (2021) Bidirectional LSTM-Based Soft Sensor for Rotor Displacement Trajectory Estimation. *IEEE Access* 9: 167556–167569.

Proceedings of ISMB 16 (2018).

Quurck L, Franz D, Schussler B, et al. (2017) Planetary backup bearings for high speed applications and service life estimation methodology. *Mechanical Engineering Journal* 4(5): 17-00010-17-00010.

Quurck L, Viitala R, Franz D, et al. (2018) Planetary Backup Bearings for Flywheel Applications. In: *Proceedings of ISMB 16:* Beijing, China, August 13.-17.

Yan W, Shao H and Wang X (2004) Soft sensing modeling based on support vector machine and Bayesian model selection. *Computers & Chemical Engineering* 28(8): 1489–1498.

Yuan X, Li L and Wang Y (2020) Nonlinear Dynamic Soft Sensor Modeling With Supervised Long Short-Term Memory Network. *IEEE Transactions on Industrial Informatics* 16(5): 3168–3176.

Zhang B, Han Y, Yu B, et al. (2020) Novel Nonlinear Autoregression with External Input Integrating PCA-WD and Its Application to a Dynamic Soft Sensor. *Industrial & Engineering Chemistry Research* 59(35): 15697–15706.