# Determining normalized friction torque of an industrial robotic manipulator using the symbolic regression method

Baressi Šegota Sandi[1], Mrzljak Vedran[1], Prpić-Oršić Jasna[1], Zlatan Car[1]
[1]Faculty of Engineering, University of Rijeka, Vukovarska 58, 51000 Rijeka, Croatia
E-mail: sbaressisegota@riteh.hr, vedran.mrzljak@riteh.hr, jasna.prpic-orsic@riteh.hr, car@riteh.hr

***Abstract:*** *The goal of the paper is estimating the normalized friction torque of a joint in an industrial robotic manipulator. For this purpose a source data, given as a figure, is digitized using a tool WebPlotDigitizer in order to obtain numeric data. The numeric data is the used within the machine learning algorithm genetic programming (GP), which performs the symbolic regression in order to obtain the equation that regresses the dataset in question. The obtained model shows a coefficient of determination equal to 0.87, which indicates that the model in question may be used for the wide approximation of the normalized friction torque using the torque load, operating temperature and joint velocity as inputs.*

***KEYWORDS:*** *GENETIC PROGRAMMING, FRICTION TORQUE PREDICTION, INDUSTRIAL ROBOTIC MANIPULATOR FRICTION, MACHINE LEARNING, SYMBOLIC REGRESSION*

## *1. Introduction*

Artificial intelligence (AI) is one of the key tools used in robotics [1] and other fields such as medicine [2], epidemiology [3], complex system engineering [4] and others today. AI can be used in many different applications in robotics – path optimization [5], determining kinematics [6] and dynamics [7] of robotic manipulators, predicting grasp strength [8] or fault detection [9].

The goal of the presented research is to apply symbolic regression [10] to the problem of determining the joint torque of a robotic manipulator, when given the temperature, torque load of the joint, and velocity as inputs. As lowering of normalized friction torque can have a positive effect on energy use [11] and equipment durability [12] not just in robotics, but other fields as well, developing friction models is an important task [13]. Symbolic regression was chosen because it generates models shaped like equations, which allows for easier integration and later analysis of models when compared to the so-called black-box models [14].

The goals of this paper are to obtain a dataset from a previously performed research, given as a figure. After a simple formal analysis is performed on the extracted data, the genetic algorithm will be applied in order to perform symbolic regression on the collected data. The final goal of the paper is to demonstrate that symbolic regression can be applied to the dataset of this type, in order to regress an equation describing the relation of joint radial velocity, joint torque load, and working temperature as inputs to the normalized joint friction torque.

This paper will initially present the methodology of obtaining the data, followed by the short description of the symbolic regression process, with the results being finally presented and discussed.

## *2. Dataset generation*

The data used in the presented research is obtained from an article by Garcia et al., dealing with determining the relevant factors for the energy consumption of industrial robots [15], in a form of a graph figure. The graph provides shows the relation of normalized friction torque to the radial velocity of the robotic manipulator joint. Four curves are given for the four different operating points – with combinations of two different variables. First variable is the torque load of the joint at two different values (0.01, and 0.70 Nm). The second variable is the operating temperature, also given at two different values (33 and 80 °C). The separate operating points will be marked separately in the software, and the above variables will be added manually when the final dataset is combined. To convert the graphed figure into the numeric data, a software WebPlotDigitizer [16] has been applied. Fig. 1 shows the interface of the WebPlotDigitizer software, loaded in a browsing window. The left and right side show the markup tools, while the middle shows the source image [15], with the manually marked points given in the middle.

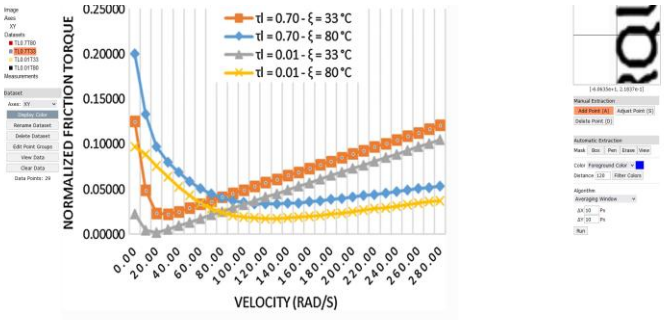


***Fig. 1.*** *WebPlotDigitizer interface with the data points manually marked for extraction*

The process starts by marking two known points on x axis (in this case 60.0 and 240.0 have been used), and y axis (in this case 0.05 and 0.2 have been used). Then, each of the datapoints of the selected series are marked and their values are exported. This process is repeated with all of the separate data series, with the exports later merged in Microsoft Excel with the values of torque load and temperature added, as given in the original figure legend [15]. It should be noted that WebPlotDigitizer allows for automatic detection and extraction of data points, but this has yielded poor results – possibly caused due to data series overlap, so manual marking has been used. One thing to note regarding the manual extraction is the need for extreme precision and consistency in the manual marking of the data points, as mislabeling the datapoints, even by a little amount, will cause wrong values to be extracted. To confirm the proper extraction has been performed the data points are separately plotted and compared to the original data. Extracted data points are given in Fig. 2. The comparison between the original and extracted data shows that the extracted data is accurate.

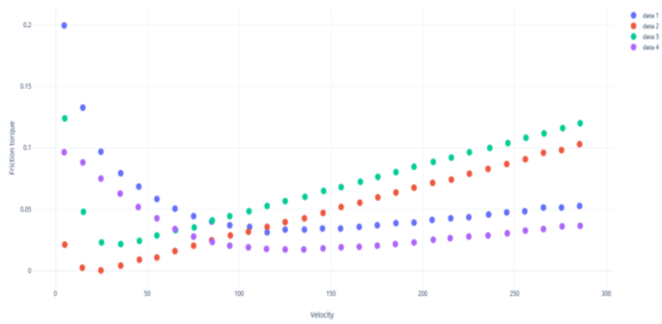


***Fig. 2.*** *Data points extracted using WebPlotDigitizer*

An example of 10 data points extracted from the figure are given in the Table 1. The data points in question are the first ten values given for the operating point of 0.01 Nm torque load and the operating temperature of 33 °C. As it can be seen the data extracted for the x-axis shows a high precision, which is another indication of precise value extraction.

| Torque Load | Temperature | Velocity | Friction Torque |
|---|---|---|---|
| 0.01 | 33 | 5 | 0.021137026 |
| 0.01 | 33 | 15 | 0.002332362 |
| 0.01 | 33 | 25 | 0.000145773 |
| 0.01 | 33 | 35 | 0.004081633 |
| 0.01 | 33 | 45 | 0.008892128 |
| 0.01 | 33 | 55 | 0.010641399 |
| 0.01 | 33 | 65 | 0.015889213 |
| 0.01 | 33 | 75 | 0.020262391 |
| 0.01 | 33 | 85 | 0.024635569 |
| 0.01 | 33 | 95 | 0.028571429 |

***Table 1.*** *Example of the extracted data points*

Statistical analysis can be performed on the extracted data. For each of the columns (torque load, operating temperature, velocity, and normalized friction torque) statistical descriptors (minimum, maximum, average value, standard error value and correlation to the output) are given in Table 2, along with other relevant information such as unit.

| | Torque Load | Temperature | Velocity | Norm. Friction Torque |
|---|---|---|---|---|
| **Minimal value** | 0.01000 | 33.00000 | 4.76190 | 0.00015 |
| **Maximal Value** | 0.70000 | 80.00000 | 285.71429 | 0.19956 |
| **Average** | 0.35500 | 56.50000 | 145.39135 | 0.05178 |
| **Standard deviation** | 0.34650 | 23.60195 | 84.24268 | 0.03275 |
| **Output correlation** | 0.31778 | -0.22761 | 0.20264 | 1.00000 |
| **Output covariance** | 0.00361 | -0.17596 | 0.55914 | 0.00107 |
| **Unit** | Nm | °C | rad/s | - |

***Table 2.*** *Descriptive statistics of the extracted data*

In the extracted data the first three columns (torque load, temperature, joint velocity) are defined as inputs, while the normalized friction torque is defined as the output for the AI methods described in the following section. Torque load and temperature have only two possible values (0.01 and 0.07 for torque load and 33 and 80 for the working temperature). Torque load shows the highest correlation, but the lowest covariance to the output. Temperature shows negative correlation and covariance indicating that a higher work temperature equals a lower friction torque. Finally, the joint velocity ranges between 4.76 and 285.71 rad/s. It shows the weakest correlation amongst the input values, but the highest covariance to the output variable. The output variable itself, normalized friction torque, ranges between 0.00015 and 0.19956. The average value is 0.05178 with a standard deviation of 0.03275.

## 3. Genetic programming

Symbolic Regression with the use of Genetic programming is a hybrid technique combining elements of both evolutionary computing [17] and machine learning [18]. It works by generating random equations, which are tested against the existing data to determine the quality of each individual equation [19]. Then, three operations are applied to these equations, with those equations that provide a better fit being more likely to be selected [20]. These operations are crossover, mutation, and reproduction. Crossover combines two solutions into one, with the goal of obtaining a better overall solution by combining two good ones [21]. Mutation randomly changes a solution to introduce a random change, which allows stagnation of solutions, and to possibly obtain a higher quality solution through a slight modification to an existing good solution [22]. Finally, reproduction allows for a good existing solution to be copied into the next iteration of the algorithm, in order to preserve good solutions for further use [23].

The equations generated by the genetic programming are given in the tree shape, with each function being given as a root node for the operation with the operands as the child nodes. For example, a simple equation such as $y = \sin(x) + x$, would be written as $add(\sin(x), x)$. For that reason, the equations that are generated can be converted and simplified using automatic software, such as Sympy [24], that will allow us to render the obtain equation in a more commonly understood format.

In the presented paper the genetic programming was implemented using GPLearn Python library [25]. The number of possible solutions in each of the algorithm iterations is set to 5000, while the number of algorithm iterations is set to a maximum of 200, with the possibility of the algorithm stopping in the case of early convergence to a satisfactory solution ($R^2 > 0.99$). Probability of crossover, mutation and reproduction are set to 0.7, 0.2 and 0.1 – respectively. All of the other settings are left at default values. The manual tuning was selected due to the GP not being overly sensitive to the hyperparameter selections [26], and the belief that these parameters, set based on the experience obtained during the previous research in the field could be used to obtain a satisfactory solution [27, 28, 29].

The models are evaluated using coefficient of determination ($R^2$). $R^2$ provides the information of how much of the variance in the original data is contained in the predicted data, or in other words – how well does the model predict the data as the output varies [30]. It is a commonly used metric for determining the quality of predictive models [31]. The value of $R^2$ is a float which ranges from 0 to 1, where 0 indicates no variance of the original dataset is present in the predicted data, while 1 indicates all the variance in the original dataset is present in the predicted data, or in other words – a perfect prediction [32]. This means that higher values of $R^2$ indicate a better prediction. $R^2$ is calculated according to:

$$R^2 = 1 - \frac{SS_{RES}}{SS_{TOT}}, \quad (1)$$

where $SS_{RES}$ and $SS_{TOT}$ are calculated as:

$$SS_{RES} = \Sigma(y_i - \hat{y}_i), \quad (2)$$

and

$$SS_{TOT} = \Sigma(y_i - \bar{y}). \quad (3)$$

In the above equations, $\hat{y}_i$ represents the predicted value corresponding to the value $y_i$ in the dataset, while $\bar{y}$ is the mean output value of the dataset.

## 4. Results and discussion

The results show a coefficient of determination equal to 0.86. This score indicates a model that is good for a general approximation, but it is not precise enough to obtain a high-quality predicted value for the given input parameters. So, while the model may not be used to get the exact value, it can be used for a general approximation of the expected normalized friction torque.

At the end of the execution the model is saved in the shape of the equation, in the previously described tree shape – which may not be optimal for further implementation as it cannot be directly integrated in most programming languages or tools commonly used. The obtained equation from the model is given as:

$$div(div(add(mul(div(X0, X2), div(add(X1, X2), mul(X2, X1))), sub(mul(add(X0, X2), div(X0, X0)), sub(mul(X1, 0.400), add(X2, X1)))), add(sub(div(add(0.129, X2), sub(X2, X2)), mul(add(-0.216, -0.932), div(X0, X0))), sub(mul(div(X0, -0.896), div(X2, X0)), div(sub(X2, 0.129), sub(X1, 0.987))))), mul(div(sub(sub(add(0.889, 0.692), mul(X1, X1)), div(sub(X2, X2), sub(X2, -0.071))), sub(mul(mul(X0, -0.640), div(X1, X0)), sub(add(X1, X2), mul(-0.579, X2)))), div(add(mul(mul(0.645, X0), div(X0, X1)), div(div(X1, 0.320), add(X0, X1))), div(div(mul(-0.825, X1), div(X1, X1)), add(add(X0, X0), add(X1, X1)))))) \tag{4}$$

where the symbols *X0*, *X1*, and *X2* indicate torque load, temperature and the joint velocity. As the above can be hard to read and interpret we can simplify it and convert it to a more commonly used equation form using Sympy [24]. This allows us to obtain the simpler equation given below:

$$\frac{A}{B}. \tag{5}$$

Due to the length, the equation is expressed with two coefficients *A* and *B*, with the coefficients in question being defined as:

$$A = -0.825\xi \cdot (-1.64\xi - 1.579\omega)(\tau\_l + \tau\_l\ (\xi + \omega)/(\xi + \omega^2\ ) + 0.6\xi + 2\omega), \tag{6}$$

and

$$B = \left((1.581 - \xi^2)(2\tau_l + 2\xi)\left(0.645\tau_l \frac{2}{\xi} + \frac{3.125\xi}{\tau_l + \xi}\right)\right) \cdot \left(1.15\omega + 1.148 - \frac{\omega - 0.129}{\xi - 0.987}\right). \tag{7}$$

For easier understanding, in equations 6 and 7, the symbols *X0*, *X1*, and *X2* have been replaced using the appropriate physical symbols used in the original paper [15], according to the following:

- Torque load on the joint of the manipulator – $\tau_l$,
- Operating temperature – $\xi$, and
- Radial velocity – $\omega$.

## 5. Conclusions

The presented results show that achieving a regressive model with some degree of accuracy is possible using an algorithm that provides an open model that is easy to integrate. Still, the results do show that the precision achieved is not high enough to be used in application where an extremely high precision of the model is necessary. The achieved results could be improved in a couple of different ways. First is the application of different regression methods to the dataset which have shown to perform better on similar datasets, such as multilayer perceptron artificial neural networks, or gradient boosted trees. The second is collection of more data-points, either through direct measurement, collection of data from other papers with similar themes as the one the data for the presented research has been collected, or synthetic data generation. Finally, the precision could additionally be improved by adding additional input variables into the dataset, as this could provide additional information to the model that would allow for a higher precision regression.

In the addition to the above, the article shows that it is possible to prepare datasets for machine learning from previously existing figures that visualize potential input data using the data extraction tool like WebPlotDigitizer. This opens up possibilities of using a significant amount of existing non-digitized data from previous research, allowing for a wider application of data-dependent techniques such as machine learning.

Finally, the results demonstrate that it is possible to achieve a somewhat precise models of normalized friction torque using torque load, operating temperature and the radial speed of the robotic manipulator joint using genetic programming, although future work should focus on significantly improving the regression quality of these models through previously described approaches.

## 6. Acknowledgment

This research has been (partly) supported by the Croatian Science Foundation under the project IP-2018-01-3739, CEEPUS network CIII-HR-0108, European Regional Development Fund under the grant KK.01.1.1.01.0009 (DATACROSS), project CEKOM under the grant KK.01.2.2.03.0004, Erasmus+ project WICT under the grant 2021-1-HR01-KA220-HED-000031177, University of Rijeka scientific grant uniri-tehnic-18-275-1447, University of Rijeka scientific grant uniri-tehnic-18-18-1146 and University of Rijeka scientific grant uniri-tehnic-18-14.

## 7. References

**[1]** Murphy, R. R. (2019). Introduction to AI robotics. MIT press. (doi: https://doi.org/10.1108/ir.2001.28.3.266.1)

**[2]** Lorencin, I., Baressi Šegota, S., Anđelić, N., Blagojević, A., Šušteršić, T., Protić, A., ... & Car, Z. (2021). Automatic evaluation of the lung condition of covid-19 patients using x-ray images and convolutional neural networks. Journal of Personalized Medicine, 11(1), 28. (doi: https://doi.org/10.3390/jpm11010028)

**[3]** Car, Z., Baressi Šegota, S., Anđelić, N., Lorencin, I., & Mrzljak, V. (2020). Modeling the spread of COVID-19 infection using a multilayer perceptron. Computational and mathematical methods in medicine, 2020. (doi: https://doi.org/10.1155/2020/5714714)

**[4]** Baressi Šegota, S., Lorencin, I., Anđelić, N., Mrzljak, V., & Car, Z. (2020). Improvement of marine steam turbine conventional exergy analysis by neural network application. Journal of Marine Science and Engineering, 8(11), 884. (doi: https://doi.org/10.3390/jmse8110884)

**[5]** Baressi Šegota, S., Anđelić, N., Lorencin, I., Saga, M., & Car, Z. (2020). Path planning optimization of six-degree-of-freedom robotic manipulators using evolutionary algorithms. International Journal of Advanced Robotic Systems, 17(2), 1729881420908076. (doi: https://doi.org/10.1177/1729881420908076)

**[6]** Šegota, S. B., Anđelić, N., Mrzljak, V., Lorencin, I., Kuric, I., & Car, Z. (2021). Utilization of multilayer perceptron for determining the inverse kinematics of an industrial robotic manipulator. International Journal of Advanced Robotic Systems, 18(4), 1729881420925283. (doi: https://doi.org/10.1177/1729881420925283)

**[7]** Baressi Šegota, S., Anđelić, N., Šercer, M., & Meštrić, H. (2022). Dynamics Modeling of Industrial Robotic Manipulators: A Machine Learning Approach Based on Synthetic Data. Mathematics, 10(7), 1174. (doi: https://doi.org/10.3390/math10071174)

**[8]** Anđelić, N., Car, Z., & Šercer, M. (2022). Prediction of Robot Grasp Robustness using Artificial Intelligence Algorithms. Tehnički vjesnik, 29(1), 101-107. (doi: https://doi.org/10.17559/tv-20210204092154)

**[9]** Anđelić, N., Car, Z., & Šercer, M. (2021). Neural Network-Based Model for Classification of Faults During Operation of a Robotic Manipulator. Tehnički vjesnik, 28(4), 1380-1387. (doi: https://doi.org/10.17559/tv-20201112163731)

**[10]** Koza, J. R., & Poli, R. (2005). Genetic programming. In Search methodologies (pp. 127-164). Springer, Boston, MA. (doi: https://doi.org/10.1007/0-387-28356-0_5)

**[11]** Park, J. Y., & Salmeron, M. (2014). Fundamental aspects of energy dissipation in friction. Chemical reviews, 114(1), 677-711. (doi: https://doi.org/10.1021/cr200431y)

**[12]** Khan, Z. A., Chacko, V., & Nazir, H. (2017). A review of friction models in interacting joints for durability design. Friction, 5(1), 1-22. (doi: https://doi.org/10.1007/s40544-017-0143-0)

**[13]** Amanov, A., & Pyun, Y. S. (2018). Lowering friction of ball screws made of different steel grades through ultrasonic impact treatment. Tribology International, 123, 105-119. (doi: https://doi.org/10.1016/j.triboint.2018.03.008)

**[14]** Rudin, C., & Radin, J. (2019). Why are we using black box models in AI when we don't need to? A lesson from an explainable AI competition. (doi: https://doi.org/10.1162/99608f92.5a8a3a3d)

**[15]** Garcia, R. R., Bittencourt, A. C., & Villani, E. (2018). Relevant factors for the energy consumption of industrial robots. Journal of the Brazilian Society of Mechanical Sciences and Engineering, 40(9), 1-15. (doi: https://doi.org/10.1007/s40430-018-1376-1)

**[16]** Drevon, D., Fursa, S. R., & Malcolm, A. L. (2017). Intercoder reliability and validity of WebPlotDigitizer in extracting graphed data. Behavior modification, 41(2), 323-339. (doi: https://doi.org/10.1177/0145445516673998)

**[17]** Anđelić, N., Baressi Šegota, S., Lorencin, I., Poljak, I., Mrzljak, V., & Car, Z. (2021). Use of Genetic Programming for the Estimation of CODLAG Propulsion System Parameters. Journal of Marine Science and Engineering, 9(6), 612. (doi: https://doi.org/10.3390/jmse9060612)

**[18]** Anđelić, N., Baressi Šegota, S., Lorencin, I., Mrzljak, V., & Car, Z. (2021). Estimation of COVID-19 epidemic curves using genetic programming algorithm. Health informatics journal, 27(1), 1460458220976728. (doi: https://doi.org/10.1177/1460458220976728)

**[19]** Koza, J. R. (2010). Human-competitive results produced by genetic programming. Genetic programming and evolvable machines, 11(3), 251-284. (doi: https://doi.org/10.1007/s10710-010-9112-3)

**[20]** Hancock, P. J. (1994, April). An empirical comparison of selection methods in evolutionary algorithms. In AISB workshop on evolutionary computing (pp. 80-94). Springer, Berlin, Heidelberg. (doi: https://doi.org/10.1007/3-540-58483-8_7)

**[21]** Nazif, H., & Lee, L. S. (2012). Optimised crossover genetic algorithm for capacitated vehicle routing problem. Applied Mathematical Modelling, 36(5), 2110-2117. (doi: https://doi.org/10.1016/j.apm.2011.08.010)

**[22]** Hong, T. P., & Wang, H. S. (1996, October). A dynamic mutation genetic algorithm. In 1996 IEEE International Conference on Systems, Man and Cybernetics. Information Intelligence and Systems (Cat. No. 96CH35929) (Vol. 3, pp. 2000-2005). IEEE. (doi: https://doi.org/10.1109/icsmc.1996.565436)

**[23]** Bridges, C. L., & Goldberg, D. E. (1987). An analysis of reproduction and crossover in a binary-coded genetic algorithm. Grefenstette, 878, 9-13.

**[24]** Meurer, A., Smith, C. P., Paprocki, M., Čertík, O., Kirpichev, S. B., Rocklin, M., ... & Scopatz, A. (2017). SymPy: symbolic computing in Python. PeerJ Computer Science, 3, e103. (doi: https://doi.org/10.7287/peerj-cs.103v0.1/reviews/2)

**[25]** Ferreira, J., Pedemonte, M., & Torres, A. I. (2019). A genetic programming approach for construction of surrogate models. In Computer Aided Chemical Engineering (Vol. 47, pp. 451-456). Elsevier. (doi: https://doi.org/10.1016/b978-0-12-818597-1.50072-2)

**[26]** Koza, J. R., Andre, D., Keane, M. A., & Bennett III, F. H. (1999). Genetic programming III: Darwinian invention and problem solving (Vol. 3). Morgan Kaufmann. (doi: https://doi.org/10.1108/k.2000.06729aae.006)

**[27]** Anđelić, N., Šegota, S. B., Lorencin, I., Jurilj, Z., Šušteršič, T., Blagojević, A., ... & Car, Z. (2021). Estimation of covid-19 epidemiology curve of the united states using genetic programming algorithm. International Journal of Environmental Research and Public Health, 18(3), 959.. (doi: https://doi.org/10.3390/ijerph18030959)

**[28]** Anđelić, N., Baressi Šegota, S., Lorencin, I., & Car, Z. (2020). Estimation of gas turbine shaft torque and fuel flow of a CODLAG propulsion system using genetic programming algorithm. Pomorstvo, 34(2), 323-337. (doi: https://doi.org/10.31217/p.34.2.13)

**[29]** Lorencin, I., Anđelić, N., Mrzljak, V., & Car, Z. (2019). Genetic algorithm approach to design of multi-layer perceptron for combined cycle power plant electrical power output estimation. Energies, 12(22), 4352. (doi: https://doi.org/10.3390/en12224352)

**[30]** Ozer, D. J. (1985). Correlation and the coefficient of determination. Psychological bulletin, 97(2), 307. (doi: https://doi.org/10.4324/9781315266404-33)

**[31]** Nagelkerke, N. J. (1991). A note on a general definition of the coefficient of determination. Biometrika, 78(3), 691-692. (doi: https://doi.org/10.1093/biomet/78.3.691)

**[32]** Baressi Šegota, S., Lorencin, I., Šercer, M., & Car, Z. (2021). Determining residuary resistance per unit weight of displacement with Symbolic Regression and Gradient Boosted Tree algorithms. Pomorstvo, 35(2), 287-296. (doi: https://doi.org/10.31217/p.35.2.11)