Maintenance Reduction of Medical Robotic Manipulators through Automatic Data-Driven Updates of Feedforward Control

Václav Helma[1], Martin Goubej[1], Pavel Březina[1], Henry Stoutjesdijk[2], and Marco Alonso[2]

[1]NTIS Research Centre, University of West Bohemia, Pilsen, Czechia
Email: {helma, mgoubej}@ntis.zcu.cz, brezina1@students.zcu.cz

[2]Philips Healthcare, Eindhoven, Netherlands
Email: {henry.stoutjesdijk, marco.alonso}@philips.com

***Abstract*—The paper presents a new method for data-driven feedforward compensation of static and quasi-static forces acting on a multi-axis medical robotic manipulator. The proposed approach uses a look-up current calibration table (CCT) and an adaptive algorithm updating the CCT to ensure that the manipulator maintains accurate, fast, and safe performance over time. The key aspect of our control strategy is called data assimilation step, which involves modelling the CCT using an approximating function. We use the NURBS (Non-uniform rational basis spline) technique, which has desirable properties such as high accuracy and flexibility in approximating and even interpolating complex functions. The technique allows the manipulator to compensate for external disturbances such as gravity, friction and gear or cabling resistance. This can improve the precision and reduce the downtime of the manipulator due to periodic feedforward recalibration.**

***Index Terms*—quasi-static forces compensation, data-driven feedforward control, look-up table adaptation, NURBS approximation and interpolation**

I. Introduction

Mechatronic systems, such as robotic manipulators, have become increasingly widespread in various industries, ranging from manufacturing to healthcare, but they often suffer from performance limitations due to external disturbances, such as friction, gravity, gear and cable resistance and other environmental factors. Feedback control techniques have traditionally been used to compensate for these disturbances, but they introduce inherent design trade-offs regarding stability, robustness, noise amplification etc. As mechatronic systems become more complex, their control strategies must also evolve to ensure efficient and accurate operation. As a result, feedforward control has emerged as a complementary technique for improving the performance of mechatronic systems.

Feedforward control is a method that estimates the effect of external disturbances on a system and applies a control action to compensate for them, without relying on feedback signals. The key advantage of feedforward control is that it can be used to compensate for disturbances that are not directly observable or cannot be measured with high accuracy. This can lead to better overall system performance and stability. This approach can be particularly useful in applications where feedback control alone is insufficient due to system limitations or environmental factors.

The development of feedforward control in mechatronics has been the subject of extensive research over the past few decades, with contributions from researchers in academia and industry. The term feedforward systems has its origin in fields of biology [1] and electronic systems [2]. However, the concepts of feedforward control were largely developed in the 1980s [3], [4], [5], [6].

One area of application for feedforward control is in robotic manipulators, where it can be used to compensate for static and quasi-static forces, such as friction [7], [8] [9], [10], gravity [11], [12], and other disturbances (gear or cabling resistance, etc.) [13], [14]. Various methods have been proposed to address each of these disturbance sources, including both model-based [15], [16], [10] and data-driven [17], [18] approaches. Additionally, some methods use adaptive [19] or learning algorithms [20] to continuously adjust the compensation based on the current operating conditions. All these methods are often combined with feedback control techniques to achieve better overall system performance.

Another area of application for feedforward control is in aerospace engineering [19], [21], where it can be used to compensate for atmospheric disturbances and other external factors that affect the flight of aircraft and spacecraft. Feedforward control can also be used in automotive and transportation systems [22], [23] to compensate for road disturbances and other external factors that affect vehicle performance.

In this paper we focus on a specific use-case in the healthcare domain of Image Guided Therapy (IGT) robots manufactured by Philips Healthcare [24] used for minimally invasive procedures like coronary catheterization. This requires a large complex multi-axes robotic arm to position a typically a C-shaped form carrying the radiography emitter and collector (see Figure 1) with respect to the patient. This environment and application require stringent demands on the robotic performance in for instance accurate positioning and motion to enable optimal 2D and 3D imaging and at the same time fulfill high demands on safe operation within a healthcare environment working near humans. This requires amongst

Fig. 1: Philips Azurion 8DoF medical manipulator

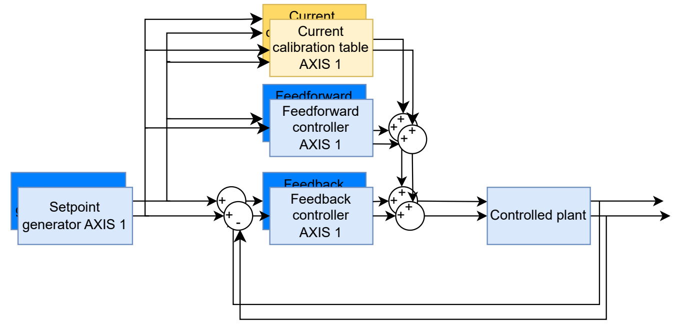


Fig. 2: Structure of control strategy for two-axes

others motion control that can accurately predict the required robotic actuator currents under all conditions and a feature in this motion control called Current Calibration Tables (CCT) and is the topic considered in this paper.

Sections II and III describe a control topology employing so called Current calibration tables (CCT), essentially a multi-dimensional lookup tables responsible for providing actuator feedforward torque for the compensation of external disturbances (see Figure 1).

The CCT compensates for the static part of friction and external torques due to gravity, cabling, gear resistance, and other factors affecting the manipulator's movement in a non-linear and time-varying manner. The current calibration table is parameterised initially during machine commissioning, but its periodic recalibration is needed to cope with varying disturbance characteristics through the machine lifespan. This requires intensive human assistance, which is inconvenient and time-consuming. Therefore, the goal of this work, which is described in more detail in Section IV, is to propose a solution to minimise the need for periodic recalibrations by means of a suitable adaptive CCT update algorithm, aiming at increasing the efficiency and reliability of the manipulator and decreasing the downtime of the robot.

In particular, a B-spline based interpolator/approximator is proposed as a unified framework for storing, adapting and evaluating CCTs. Interpolation and approximation are fundamental techniques, enabling proper values to be derived at points between the mesh of sampled data points stored in the look-up table. These topics are therefore addressed in Sections V-VIII, which first describe the general concepts and then the application of these approaches to the CCT update problem.

The main focus of the paper lies in Section IX, where CCT update algorithm is introduced together with examples performed on real data obtained from the medical manipulator. The proposed solutions could significantly improve the calibration process, ultimately resulting in more reliable and efficient performance during medical procedures.

The concluding remarks and summary of the work are given in Section X, while future research directions are outlined in the last Section XI.

II. Structure of Motion Control Loops

The motion control loops of the medical manipulator (see Figure 2 for two axes controller) are structured to incorporate various controllers and compensators to ensure accurate, fast and smooth movement. These include feedback controllers, feedforward compensators, and a current calibration table (CCT).

- The *feedback controllers* are responsible for tracking the reference position trajectory, which is the desired path of the manipulator's movement. These controllers continuously compare the actual position of the manipulator with the desired trajectory and make any necessary adjustments to ensure that the manipulator moves along the desired path as closely as possible.
- To compensate for inertia and partly friction of the manipulator moving parts, the *feedforward compensation* is implemented. It provides additional input to the controller based on the expected effects of these factors in a model-based manner, allowing for more precise and responsive control without any unexpected jerks or delays.
- The *current calibration table* is used to compensate for other factors not compensated for by the model-based feedforward compensation. These include non-linear position dependent friction and external torques due to for instance cabling, gear resistance, and other wear factors that may change over the lifetime of the system. This table contains calibration values that are used to complement the current as calculated by the feedforward compensation and provided to the manipulator's motors.

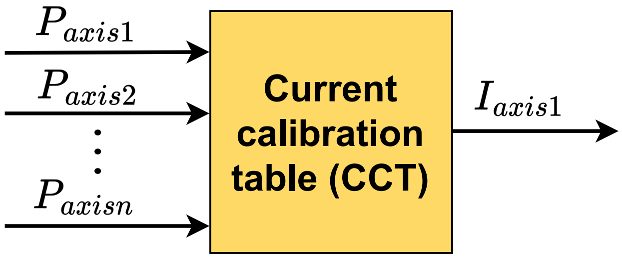


Fig. 3: CCT diagram

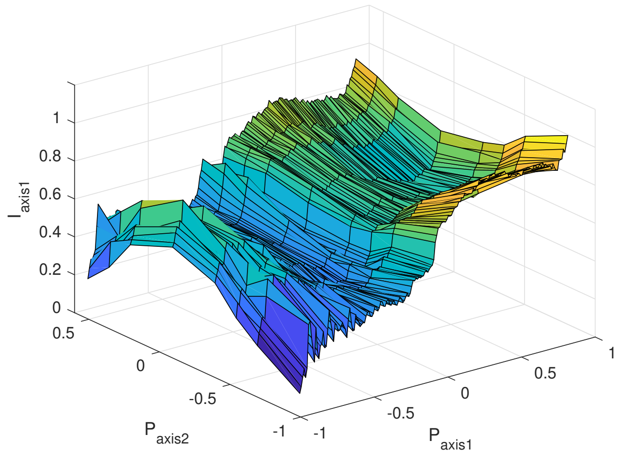


Fig. 4: CCT as a function of two variables

By combining all these controllers and compensators, the motion control loops of medical manipulator are able to provide highly precise and responsive control over the manipulator's workspace. This allows for a wide range of medical procedures to be performed with greater accuracy and safety.

III. Current Calibration Table

The medical manipulator control system relies on a current calibration table to achieve accurate joint movements. The table uses a look-up function to map desired joint positions to drive torques or currents respectively (as shown on diagram in Figure 3), which is crucial for precise motion control. In this particular case, the CCT is a dependence of the required drive current on the position of the two joints and can therefore be visualized (Figure 4) as a surface plot. The position axes can be of arbitrary units of rotation or translation and therefore all axes in the plot have been normalized to maintain a generic representation throughout this paper. The look-up table is initially calculated during the machine commissioning phase, which is done to ensure that the manipulator is calibrated to perform optimally. However, due to the natural wear and tear of the machine, periodic recalibration is required. This is because factors such as joint and gear friction, as well as cabling flexibility, can change over time. By regularly recalibrating the current calibration table, the medical manipulator can maintain

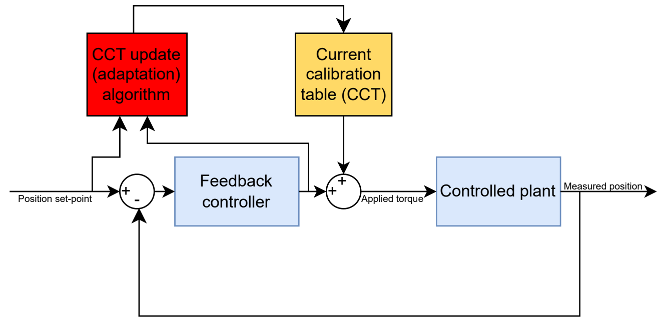


Fig. 5: CCT adaptation diagram

its accuracy and precision, ensuring reliable and consistent performance in medical procedures.

IV. The Goal and Possible Solutions

The goal is to eliminate or minimize the need for periodic recalibrations of CCT. By eliminating or reducing the need for periodic re-calibrations, the medical manipulator would require less downtime for maintenance and thereby increase the number of possible patient treatments.

The proposed CCT adaptation process begins by identifying the appropriate motion states from recorded data, where static forces/torques are predominant. Then, we calculate calibration table errors based on known applied torque and update the calibration table accordingly. The procedure is illustrated schematically in Figure 5. It is important to note that a continuous adaptation of the calibration table is not desirable due to stability concerns. Instead, a single-shot update during normal machine downtime is preferred to ensure stability and prevent any potential issues. We ommit a detailed description of the data-processing step and focus solely on the CCT adaptation algorithm in this actual work.

V. Interpolation and Approximation

The paper often refers to three key types of problems in data analysis: interpolation, approximation (regression), and extrapolation. These are fundamental concepts required for the described approach. They are sometimes used interchangeably in the literature. Therefore, we define the terminology used further in the text for the sake of clarity.

- *Interpolation* is the problem of finding or estimating new data points within the range of a known set of discrete data points. The goal is to create an interpolating function that goes through the known data set. This problem is often encountered in signal processing, where it is known as resampling.
- *Approximation (or regression)* is the process of finding or estimating new data points within the range of a set of approximately known data points. This time, the approximating function does not exactly match the known data set, while accounting for measurement or observation errors. Metrics of data fitness, such as the sum of squared errors criterion, are often used to evaluate the quality of

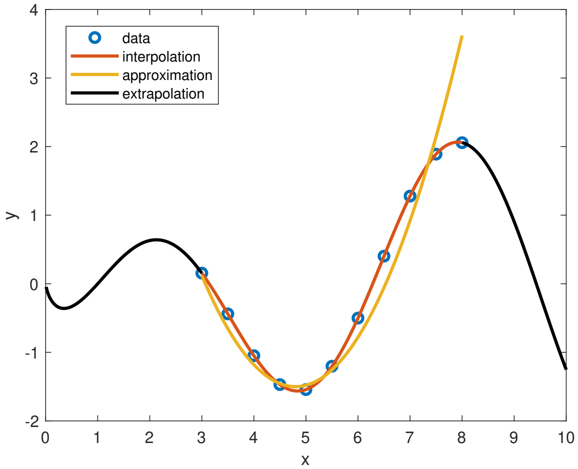


Fig. 6: Interpolation, approximation and extrapolation comparison

the approximating function. This problem is common in statistics and modelling, where it is used to analyze and understand complex data sets.

- *Extrapolation*, on the other hand, involves finding or estimating new data points outside the range of a set of known or partially known data points. In this case, we are guessing or predicting new points that lie beyond the domain of the original data. A practical example of extrapolation is the prediction of COVID-19 disease spread.

Visual comparison is given in the form of a simple 1D example in Figure 6.

VI. Relevance of the Interpolation and Approximation to CCT Problem

Interpolation plays a crucial role in addressing the CCT problem for the medical manipulator. The discrete set of data points is stored in the form of a look-up table. However, to achieve optimal performance, the system needs to derive accurate values for points that lie between the sampled data points and that is where interpolation finds its place. To achieve this, we need to find a static n-dimensional map between position and current spaces. Mathematically, this can be expressed as:

$$I_{Axis_i} = \mathbf{f}(P_{Axis_1}, \ldots, P_{Axis_n}). \tag{1}$$

By finding a suitable function that can interpolate between these known data points, we can generate accurate current values for any desired position within the range of motion.

Approximation is another critical aspect of addressing the CCT update problem. In some cases, new information may become available from measurements that was not present in the original look-up table. This means that the stored table may now be partially known and possibly obsolete. To form a new table that incorporates the new information, a data merging step is required. Since both the old calibration table and the measurements contain some error, this data merging step is essentially an approximation problem. By introducing a fitness metric, such as the sum of squared errors criterion, an approximating function can be found that adequately fits the available data points. Overall, approximation is essential for maintaining the accuracy and reliability of the look-up table, especially when new information becomes available. By leveraging appropriate approximation techniques, we can ensure that the medical manipulator performs optimally and safely in all operating conditions and over time.

VII. Constructing the Interpolating Function

There are several methods for constructing an interpolating function. Next, the most commonly used ones will be briefly introduced.

- The first method, *linear interpolation*, involves connecting adjacent data points with line segments. This method is simple to construct and evaluate and can be simply extended to multi-dimensional spaces. However, sharp transitions may not be desirable, as they can induce vibrations in motion trajectories and feedforwards.
- The second method is *polynomial interpolation*, which involves finding a polynomial function that goes through the data points, solving the output smoothness problem. This method is simple to construct as well. However, it is more computationally expensive and requires high-degree polynomials for long datasets, which can lead to oscillatory artifacts.
- The third method, *spline interpolation*, fits low-degree polynomials in each of the intervals, preserving smoothness, avoiding unwanted behavior of high-degree functions, and keeping computational burden low. However, it is more difficult to construct, and one needs to ensure the compatibility of derivatives in the knot points. Furthermore, spline interpolation can be numerically ill-conditioned until treated properly. Despite these challenges, spline interpolation has become a standard in various fields of science and engineering due to its many advantages.

A comparison of the above methods is depicted in Figure 7. The high-order polynomial interpolation is deliberately chosen to demonstrate the undesired chattering effects.

VIII. Basis-Spline (B-Spline) Functions

B-spline functions, short for Basis-spline functions, are piece-wise polynomial functions with several unique properties. The theory of NURBS and B-splines in general is well described in [25]. These functions define a polynomial basis of a chosen degree with a minimal support, making them numerically robust compared to the power basis. Additionally, any spline function can be generated from a B-spline, and its "shape" is obtained from a linear combination of the basis functions and a "control polygon" that determines the output dimension.

$$\Phi(u) = \sum_{i=0}^{n} N_{i,p}(u) P_i. \tag{2}$$

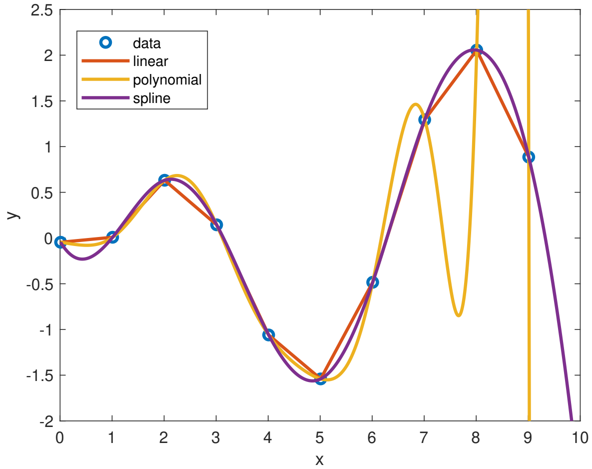
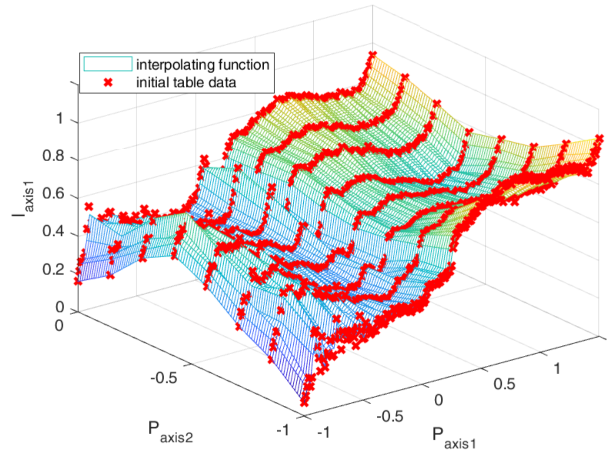
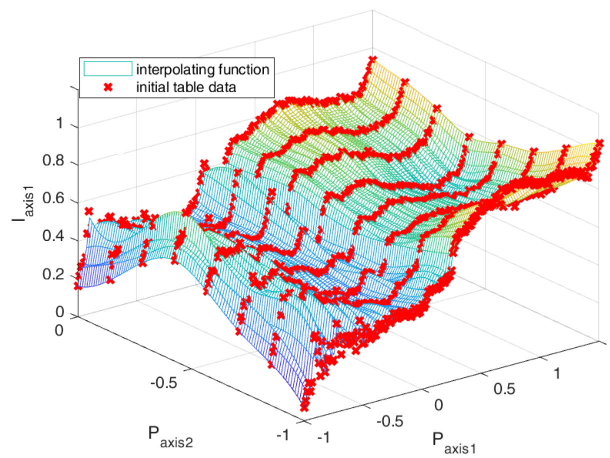


Fig. 7: Linear, polynomial and spline interpolation



(a) Linear interpolation



(b) Cubic interpolation

Fig. 8: Initialization step - data interpolation

The linearity with respect to control points can be exploited in fitting problems, making B-spline functions highly useful in many fields, including robotics and motion control systems for describing motion trajectories and/or feedforwards. B-spline functions can also be adapted to high-dimensional spaces, such as for a bivariate case

$$P(u, v) = \sum_{j=0}^{m} \sum_{i=0}^{n} \mathbf{P}_{i,j} N_{i,k}(u) N_{j,k}(v). \tag{3}$$

Because of their robust and reliable data handling, evaluation, and versatility in interpolation and approximation problems, B-spline functions have a wide range of applications across many fields. One of the most common uses of B-spline functions is in computer graphics and CAD systems, where they are used to create B-spline surfaces. They are also highly relevant in other fields such as robotics and motion control systems for trajectory planning and feedforward design. Overall, the properties of B-spline functions make them a powerful tool in a variety of contexts.

IX. CCT UPDATE ALGORITHM

The proposed algorithm for adapting the current calibration table (CCT) involves following steps:

1) An interpolation function is constructed from the calibration table data and stored. Mathematically, we solve the following problem. Having a set of n data points

$$(I_1, P1_1, P2_1), (I_2, P1_2, P2_2), \ldots, (I_n, P1_n, P2_n) \tag{4}$$

satisfying

$$P1_1 < P1_2 < \cdots < P1_n, \tag{5}$$
$$P2_1 < P2_2 < \cdots < P2_n \tag{6}$$

and

$$I_1 = f(P1_1, P2_1), \tag{7}$$
$$I_2 = f(P1_2, P2_2), \tag{8}$$
$$\vdots$$
$$I_n = f(P1_n, P2_n), \tag{9}$$

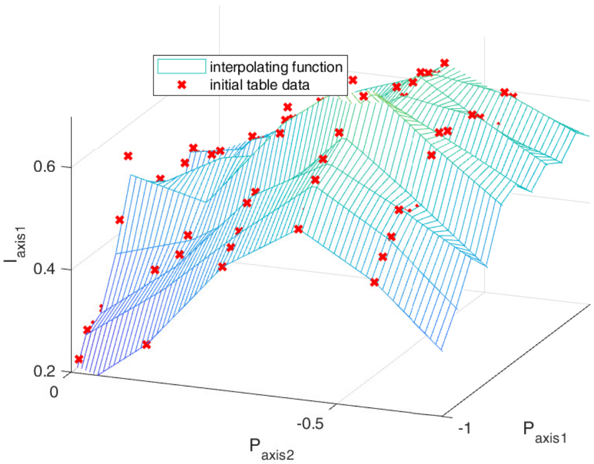
then

$$I = f(P1, P2) \tag{10}$$

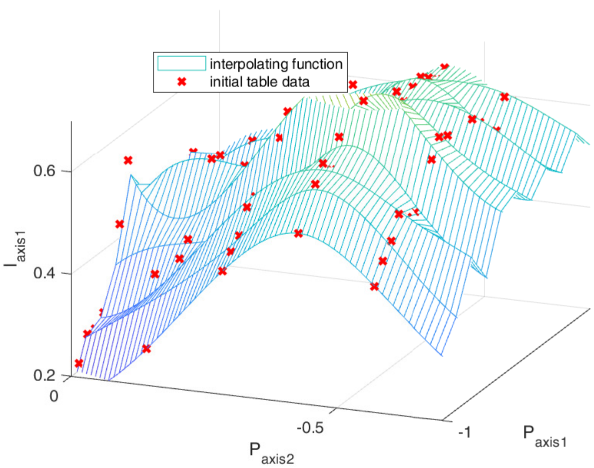
is the desired interpolation function we are looking for, if at the same time

$$P1_1 < P1 < P1_n, \tag{11}$$
$$P2_1 < P2 < P2_n \tag{12}$$

hold. This is shown in Figure 8 on real CCT data for linear and cubic interpolation, respectively. For better clarity, these surfaces are zoomed in Figure 9. This

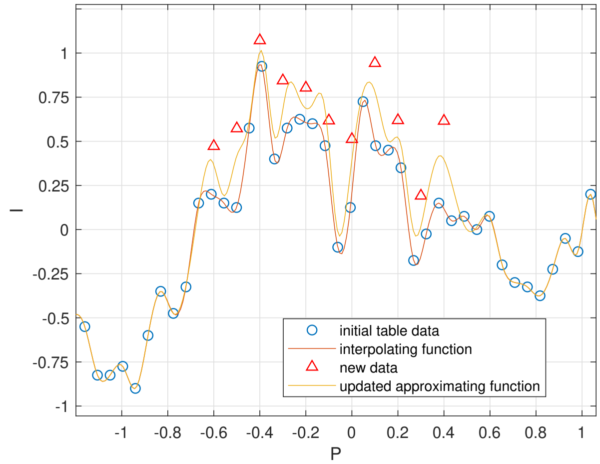


(a) Linear interpolation

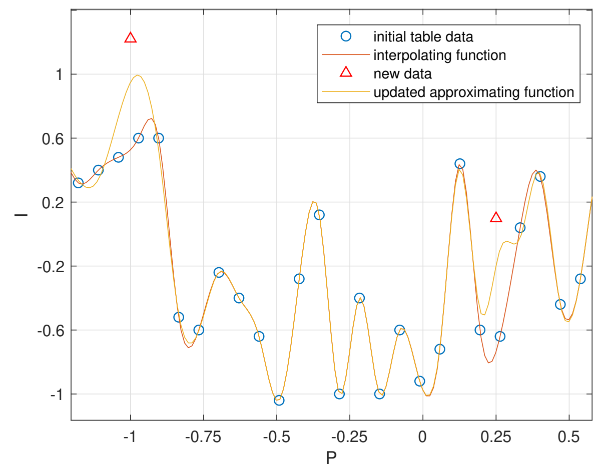


(b) Cubic interpolation

Fig. 9: Initialization step - data interpolation (zoomed)



(a) Example 1



(b) Example 2

Fig. 10: Data assimilation step shown on 1D slice of the CCT data

interpolation function is in the form of a multivariate spline.

2) The interpolation function from the previous step is used to generate current feedforwards in the optional second step. This step has the potential to improve motion smoothness when compared to (bi)linear interpolation.
3) The third step involves off-line processing of motion data to extract new data points. This step has not been carried out yet, but it involves scanning the readings of the axes positions and commanded currents, identifying parts suitable for extraction of quasi-static components covered by the CCT model (preferably low non-zero constant velocity), and collecting new data points to merge with the existing table.
4) An approximation problem is formulated and solved to update the interpolating function, which involves recomputing the control points of the B-spline. We call this as a *data assimilation step*. Mathematically, we solve the following problem. Suppose we now have $n+p$ data points (the original n data points and p new data points to update the table)

$$(I_1, P1_1, P2_1), \dots, (I_n, P1_n, P2_n), \dots, (I_{n+p}, P1_{n+p}, P2_{n+p}), \tag{13}$$

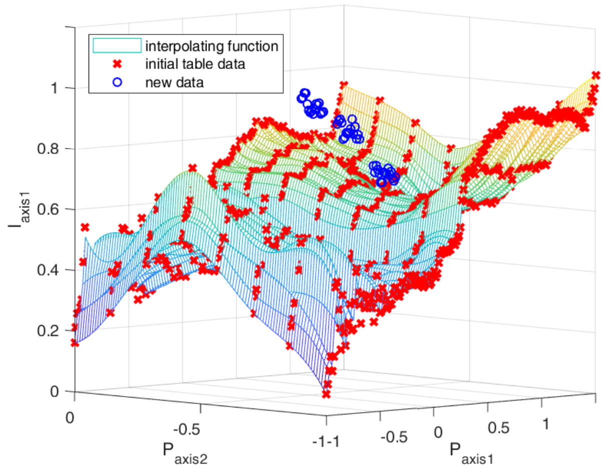
we need to find such a function

$$I = f(P1, P2) \tag{14}$$

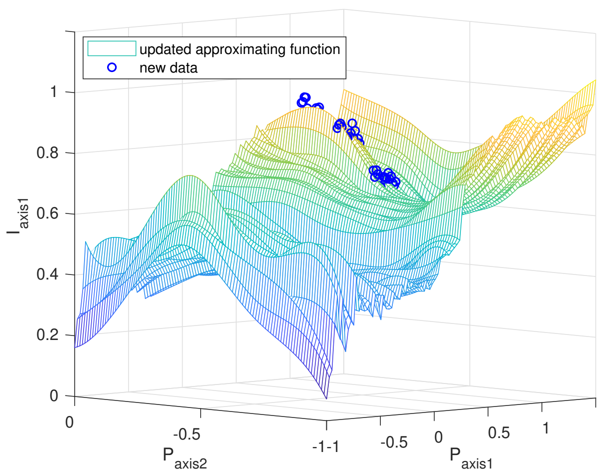
which minimizes the least-squares criterion

$$J = \sum_{i=1}^{n+p} w_i (I_i - f(P1_i, P2_i))^2 \tag{15}$$

where w_i are the optional weights. For better illustration, the procedure is first shown in Figure 10 by means of one-dimensional slice of a real CCT data. Figure 11 then represent an example using full data set.



(a) Before CCT update



(b) After CCT update

Fig. 11: Data assimilation step shown on CCT data

5) In the table update step, the updated interpolating function is evaluated on the grid of the CCT to update it with new values. This prepares the machine for the next run.
6) Finally, the optional step involves storing the difference to the initial CCT on its grid, which can be used for detecting machine malfunction through the fault detection signature step.

X. Summary

The work done so far includes the development of a mathematical framework that can interpolate, store, generate, and adaptively update current calibration table (CCT) models. This framework allows for systematic and automated blending of "old" CCT with new data in an optimal manner. Thanks to the unique properties of the B-spline basis, the resulting interpolant preserves the consistency of the CCT and only changes shape locally in the vicinity of new data points. This approach can be used for an arbitrary number of axes, from n to m dimensions, and is computationally and data efficient. It requires only approximately 5-15% more data storage compared to traditional look-up tables. In addition to these benefits, the developed framework can serve for fault detection and diagnostics. All this makes it a powerful tool for improving machine performance and ensuring that it remains reliable and robust over time.

XI. Future Work

There are several next steps that need to be taken in order to advance this work further. First and foremost, the data extraction step needs to be designed carefully, with rules defined for finding relevant motion data. This will require some filtering or averaging techniques in order to obtain meaningful results. Once the data extraction problem has been optimized, the next step will involve testing and validating the entire methodology, preferably using real machine data to ensure that the approach performs as expected in practice. However, it may be possible to carry out some of this testing offline, depending on the specifics of the data available. One potential avenue for further exploration is the use of a second neural network-based solution. However, this approach has several limitations that must be taken into account. For example, there are many degrees of freedom and hyperparameters to consider, including the network structure, number of layers, type of activation functions, and learning/adaptation rates, which make the solution appear inferior. Standard training methods do not guarantee convergence, and once trained, it becomes a complex multivariate nonlinear function, making it difficult to understand how it works and interpret the results. Overtraining can also produce unwanted chattering effects that are comparable to high-degree polynomials. Overall, the next steps will involve careful optimization of the approach, testing and validation using real-world data, and continued exploration of alternative solutions.

Acknowledgment

This research was performed within the 'Intelligent Motion Control under Industry4.E' (IMOCO4.E) project [26], [27] which has received funding from the ECSEL Joint Undertaking under grant agreement No. 101007311.

References

[1] D. M. MacKay, "Cerebral organization and the conscious control of action," *Brain and Conscious Experience: Study Week September 28 to October 4, 1964, of the Pontificia Academia Scientiarum*, pp. 422–445, 1965.
[2] U.S. Patent US1 686 792A, 1925.
[3] P. H. Meckl and W. P. Seering, "Minimizing residual vibration for point-to-point motion," 1985.
[4] ——, "Feedforward control techniques to achieve fast settling time in robots," in *1986 American Control Conference*. IEEE, 1986, pp. 1913–1918.
[5] N. C. Singer and W. P. Seering, "Using acausal shaping techniques to reduce robot vibration," in *Proceedings. 1988 IEEE International Conference on Robotics and Automation*. IEEE, 1988, pp. 1434–1439.
[6] K. W. Oosting, "Simulation of control strategies for a two degree-of-freedom lightweight flexible robotic arm," Ph.D. dissertation, Georgia Institute of Technology, 1987.

[7] B. Armstrong-Hélouvry, P. Dupont, and C. C. De Wit, "A survey of models, analysis tools and compensation methods for the control of machines with friction," *Automatica*, vol. 30, no. 7, pp. 1083–1138, 1994.
[8] B. Armstrong-He´ louvry, P. Dupont, and C. Canudas de Wit, "Friction in servo machines: Analysis and control methods," 1994.
[9] S. Huang, W. Liang, and K. K. Tan, "Intelligent friction compensation: A review," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 4, pp. 1763–1774, 2019.
[10] Z. Jamaludin, H. Van Brussel, and J. Swevers, "Friction compensation of an XY feed table using friction-model-based feedforward and an inverse-model-based disturbance observer," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 10, pp. 3848–3853, 2009.
[11] A. D. Luca and S. Panzieri, "Learning gravity compensation in robots: Rigid arms, elastic joints, flexible links," *International journal of adaptive control and signal processing*, vol. 7, no. 5, pp. 417–433, 1993.
[12] S. Ito, S. Nishio, Y. Fukumoto, K. Matsushita, and M. Sasaki, "Gravity compensation and feedback of ground reaction forces for biped balance control," *Applied bionics and biomechanics*, vol. 2017, 2017.
[13] S. Bhattacharyyta, "Disturbance rejection in linear systems," *International Journal of Systems Science*, vol. 5, no. 7, pp. 633–637, 1974.
[14] C.-L. Lin and Y.-H. Hsiao, "Adaptive feedforward control for disturbance torque rejection in seeker stabilizing loop," *IEEE Transactions on Control Systems Technology*, vol. 9, no. 1, pp. 108–121, 2001.
[15] M. Grotjahn and B. Heimann, "Model-based feedforward control in industrial robotics," *The International Journal of Robotics Research*, vol. 21, no. 1, pp. 45–60, 2002.
[16] M. Boerlage, M. Steinbuch, P. Lambrechts, and M. van de Wal, "Model-based feedforward for motion systems," in *Proceedings of 2003 IEEE Conference on Control Applications, 2003. CCA 2003.*, vol. 2. IEEE, 2003, pp. 1158–1163.
[17] F. Boeren, T. Oomen, and M. Steinbuch, "Iterative motion feedforward tuning: A data-driven approach based on instrumental variable identification," *Control Engineering Practice*, vol. 37, pp. 11–19, 2015.
[18] X. Li, S.-L. Chen, C. S. Teo, and K. K. Tan, "Data-based tuning of reduced-order inverse model in both disturbance observer and feedforward with application to tray indexing," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 7, pp. 5492–5501, 2017.
[19] A. Wildschek, "An adaptive feed-forward controller for active wing bending vibration alleviation on large transport aircraft," Ph.D. dissertation, Citeseer, 2009.
[20] T. J. de Vries, W. J. Velthuis, and J. van Amerongen, "Learning feed-forward control: A survey and historical note," *IFAC Proceedings Volumes*, vol. 33, no. 26, pp. 881–886, 2000.
[21] Y. Wang, A. Da Ronch, and M. Ghandchi Tehrani, "Adaptive feedforward control for gust-induced aeroelastic vibrations," *Aerospace*, vol. 5, no. 3, p. 86, 2018.
[22] C. Nouillant, F. Assadian, X. Moreau, and A. Oustaloup, "Feedforward and crone feedback control strategies for automobile ABS," *Vehicle System Dynamics*, vol. 38, no. 4, pp. 293–315, 2002.
[23] T. Shim and D. Margolis, "Using μ feedforward for vehicle stability enhancement," *Vehicle System Dynamics*, vol. 35, no. 2, pp. 103–119, 2001.
[24] [Online]. Available: https://www.philips.nl/healthcare/e/image-guided-therapy
[25] L. Piegl and W. Tiller, *The NURBS book*. Springer Science & Business Media, 1996.
[26] M. Čech, A.-J. Beltman, and K. Ozols, "Digital twins and ai in smart motion control applications," in *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 2022, pp. 1–7.
[27] ——, "Pushing mechatronic applications to the limits via smart motion control," *Applied Sciences*, vol. 11, no. 18, p. 8337, 2021.