Lyu, Tuojian; Lashchev, Andrei; Patil, Sandeep; Atmojo, Udayanto Dwi; Vyatkin, Valeriy

Mechatronic Swarm and its Virtual Commissioning

Published in:
Proceedings - 2023 IEEE International Conference on Mechatronics, ICM 2023

DOI:
10.1109/ICM54990.2023.10101912

Published: 17/04/2023

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Lyu, T., Lashchev, A., Patil, S., Atmojo, U. D., & Vyatkin, V. (2023). Mechatronic Swarm and its Virtual Commissioning. In *Proceedings - 2023 IEEE International Conference on Mechatronics, ICM 2023* (Proceedings - 2023 IEEE International Conference on Mechatronics, ICM 2023). IEEE. https://doi.org/10.1109/ICM54990.2023.10101912

Mechatronic Swarm and its Virtual Commissioning

Tuojian Lyu*, Andrei Laschev†, Sandeep Patil‡ , Udayanto Dwi Atmojo*, Valeriy Vyatkin*‡
*Department of Electrical Engineering and Automation, Aalto University, Espoo, Finland
†Independent researcher
‡Department of Computer Science, Computer and Space Engineering, Lulea Tekniska Universitet, Sweden

Email: tuojian.lyu@aalto.fi, laschev-andrej@yandex.ru, sandeep.patil@ltu.se, udayanto.atmojo@aalto.fi, vyatkin@ieee.org

***Abstract*—This paper presents a swarm-based architecture for the composition of mechatronic systems from smart components complemented by a Virtual Commissioning (VC) environment. The architecture is based on the IEC 61499 standard. The proposed solution enables plug-and-play composition of the system which is ready to operate "out of the box" right after it was composed without extra programming.**

I. Introduction

Production industries experience ever-increasing market pressure in terms of costs, product variability and sustainability of production. The flexibility of production systems is seen as the utmost important characteristic to address these challenges.

Modular machines enabled by information technologies, including artificial intelligence, machine learning, cloud/edge computing, and containerization, are seen more frequently applied towards increasing the flexibility, agility and resilience of smart factories. An example of such a machine is depicted in Fig. 1.

Swarm intelligence [1] is a bio-inspired approach applied to building adaptable multi-robot systems which has gradually been extended to various domains, including industrial automation [2].

A mechatronic swarm refers to a group of autonomous mechatronic systems that work together to achieve a common goal. These systems can be composed of a variety of devices, such as conveyor networks, robotics arms, and sensor networks, that are connected and coordinated through advanced communication and control algorithms. Intelligent mechatronics incorporates advanced technologies such as artificial intelligence, containerization, and VC can further enhance the performance of mechatronic swarm.

In flexible manufacturing environments, mechatronic swarm promises to achieve high efficiency, flexibility, agility, and adaptability.

A typical task for the plug-and-produce composed mechatronic system is illustrated in Fig. 2. The workpiece needs to be carried from the pick-up to the drop-off points (specified with certain tolerances) along a trajectory, again specified with some intermediate points, obeying the specified motion characteristics, such as time, speed or torque, acceleration and jerk, and being within specified tolerances. In addition, dynamically appearing obstacles may need to be avoided.

[1]The drawing is depicting a product of AFAG AG.

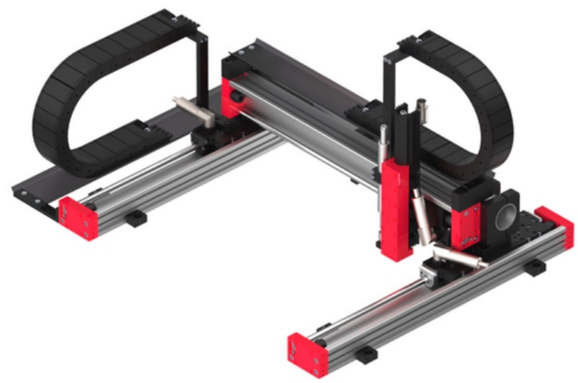

Fig. 1: An example of a manipulator composed of smart mechatronic axes[1].

Besides intelligent mechatronics, smart factory also enhances the enabler - plug-and-play of intelligent mechatronics. Not only should AI/ML or advanced technologies be composed in intelligent devices, but those mechatronic components or control applications should be decentralized and support plug-and-play and reconfigurability. The concept of plug-and-play for smart mechatronics/components allows for flexible and agile integration of these components into a system without needing complicated configuration/reconfiguration. This feature is becoming increasingly important in advanced manufacturing and automation, especially in an agile manufacturing environment.

Another critical step to enable plug-and-play for intelligent mechatronics under a mechatronic swarm is VC. Generally,

[2]The drawing is courtesy of Friedrich Durand of AFAG AG.

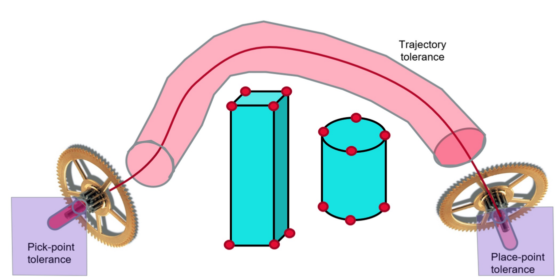

Fig. 2: A task for the composed mechatronic system[2].

VC refers to debugging control code or commission automation systems in an agile manner while saving cost and time. Since intelligent mechatronics composes advanced technologies, VC is an essential phase for the mechatronic swarm to work properly, ensuring composed mechatronic components cooperate ideally as designed.

To achieve plug-and-play of smart mechatronics in mechatronic swarm, there are two critical challenges:

- Challenge 1: What is the tool or standard could be used to provision decentralized automation systems running on distributed devices from various vendors.
- Challenge 2: How to conduct VC for smart mechatronic components containing distributed control applications.

Under the first challenge, there are several tasks. Since intelligent mechatronic components require plug-and-play, these components are distributed and should be functioning in either centralized or decentralized style. This feature requires tools or platforms which include capabilities such as modeling systems, programming control applications, and distributed deployment together. Due to its support of distributed computing and system-level modeling, IEC 61499 is chosen as the standard to model the control system, and its Integrated Development Environment (IDE) 4diac is the development platform.

The second challenge addresses questions about VC of distributed IEC 61499 applications. Control applications in the ROS nodes are first commissioned via a simulator in ROS. After migrating joint controllers to 4diac from ROS, joint controllers should also be commissioned before deploying to real PLCs. In the section of evaluation, we presented the differences of performance under various parameters. VC results from experimental data prove that the performance of distributed control in IEC 61499 of a 3-axis manipulator could be improved after VC.

II. Related works

In response to the increasing demand for manufacturing flexibility Arai *et al.* have proposed the concept of plug-and-produce assembly systems [3]. Intelligent mechatronics was considered as an important enabling technology as summarised by Lastra [4]. The latter is enabled by a number of mechanical, communication and computing technologies, among which the multi-agent software architecture plays an important role [5], [6].

The IEC 61499 standard [7] introduces a component reference automation architecture that extends the traditional PLC programming architecture of IEC 61131-3 with the features needed in the plug-and-produce systems, such as support of distributed computing and better compatibility with information technologies [8]. Due to its compatibility with advanced information technologies, IEC 61499 control applications could be virtually commissioned via digital twins or simulators either on the local or on the edge/cloud platforms [9] [10].

Firstly, regarding the term plug-and-play, two key topics were introduced in the work [11] to discuss how to achieve component-level plug-and-play for the modern IO-OT hybrid control systems:

- How to model the overall hybrid automation system in one language/standard.
- How to enable the hybrid deployment if devices/runtime are from different vendors.

Those two issues are addressed by using IEC 61499 standard because it is a system-level modelling language as well as control application programming language supporting distributed deployment.

Sorouri *et al.* [12] discussed the increasing demand for plug-and-play service in the medical devices domain. In their work, they mentioned the higher degree of freedom robots and medical devices require more flexibility and interoperability. Decentralized control seems to be better than centralized control in some scenarios where those robots or devices are used. To increase the interoperability and flexibility, IEC 61499 is used to model and create control applications in a distributed manner. This work was followed by [13], where Sorouri *et al.* present a multi-agent control of a pick-and-place manipulator ready for mechatronic plug-and-produce operation and enabled with IEC 61499.

Sinha *et al.* have considered requirements to the decentralised control architecture for intelligent mechatronic systems in [14], having compared SystemJ and IEC 61499, which confirmed some fundamental aspects of the distributed control architecture important for plug-and-produce.

The works mentioned above were majorly focused on discrete automation systems. Application of the plug-and-produce principle in continuous motion domain was started by using decentralised interpolation in works of Dripke [15], and Beller *et al.* [16], who used IEC 61499. However the presented IEC 61499 solution did not have a general and reproducible method for decentralised interpolation.

Other related supporting technologies were also addressed. Thus, Oliviera *et al.* [17] presented an approach on automating the integration of IEEE 1451 compliant smart transducer in IEC 61499 environment based on plug and play concept.

Jirkovský *et al.* [18] proposed a solution for plug and play utilizing Semantic Web and OPC UA.

Profanter *et al.* [19] presented a system architecture to facilitate plug and produce based on semantics OPC UA skills concept.

Ye *et al.* [20] presented and demonstrated how Asset Administration Shell (AAS) can be used to implement the plug-and-produce scenario.

Yang *et al.* [21] presented a use case of applying IEC 61499 to the next-generation power distribution infrastructure. For this advanced infrastructure, one of the critical features is plug-and-play device integration.

The work [22] introduced enablers of plug-and-play service based on web service description language and IEC 61499. A building automation system was used as an example to demonstrate the plug-and-play software services.

Dai *et al.* [11] proposed a more flexible software architecture composing edge computing platforms, microservices, and containerization technologies. Their presented an architecture that enables plug-and-play of intelligent components via pulling a set of atomic services from edge nodes.

From many related works, their use cases prove IEC 61499 is suitable for enabling plug-and-play of intelligent mechatronics of mechatronic swarm. Compared to the legacy IEC 61131-3, it is more flexible and easier to integrate advanced information technologies in IEC 61499. This capability matches the principle of intelligent mechatronics. Moreover, distributed deployment also matches the idea of mechatronic swarm.

III. Inverse kinematics solution for multi-axis manipulator

As a case study, we use a 3-axis manipulator depicted in Fig.3. The manipulator has three axis (J1-J3), each of which has an independent controller to control the drive.

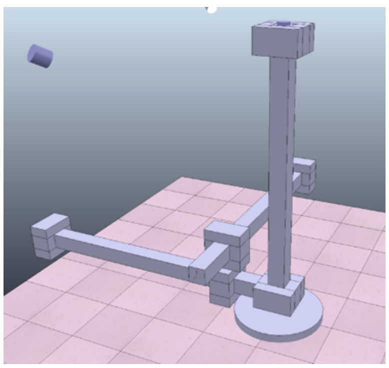

Fig. 3: Case study model: 3-axis manipulator.

Direct kinematics is a method that uses the kinematic equations of the robot to calculate the position of the working body (e.g., gripper) in some basic coordinate system (BCS) based on the set values of the parameters of the manipulator links [23].

The calculation of the coordinate system of the last link of the manipulator from the BCS of the manipulator can be performed as (1) using the transformation matrix (2).

$$r_0 = M_0^n r_g = M_0^1 M_0^1 ... M_{n-1}^n r_g = \begin{pmatrix} & R_0^n & & T_0^n \\ 0 & 0 & 0 & 1 \end{pmatrix} r_g \quad (1)$$

$$M_{n-1}^n = \begin{pmatrix} cos\theta_n & -sin\theta_n cos\alpha_n & sin\theta_n sin\alpha_n & a_n cos\theta_n \\ sin\theta_n & cos\theta_n cos\alpha_n & -cos\theta_n sin\alpha_n & a_n sin\theta_n \\ 0 & sin\alpha_n & cos\alpha_n & d_n \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (2)$$

where: R is a 3×3 submatrix, which describes the rotational movement, T is 3×1 submatrix, which describes translation movement, and N is the number of joints [24], M_{n-1}^n is the resulting transformation matrix from the n-link coordinate system to the manipulator base coordinate system, r_g is a 4×1 vector, which represents the coordinates of the manipulator grip in the coordinate system of the n-th link.

The gripper coordinates of the manipulator in the BCS can be derived as:

$$p = \begin{pmatrix} T_0^g(1) & T_0^g(2) & T_0^g(3) & \varphi & \theta & \psi \end{pmatrix}, \quad (3)$$

where: $\varphi = arctan\left(\frac{R_{2,3}}{R_{1,3}}\right)$, $\theta = arctan\left(\frac{R_{1,3}cos\varphi + R_{2,3}sin\varphi}{R_{3,3}}\right)$, $\psi = arctan\left(\frac{R_{3,2}}{-R_{3,1}}\right)$.

The inverse kinematics (IK) problem is the calculation of the generalized coordinates of joints for given linear and angular coordinates of the end effector of the manipulator. The main task of this method is to solve the following equation:

$$\hat{\beta} \in argmin_\beta S(\beta) \equiv argmin_\beta \sum_{i=1}^{N} [y_i - f(x_i, \beta)]. \quad (4)$$

As applied to the IK of a manipulator, y_i is to be substituted with the coordinates of the target position of a gripper of the manipulator (2) in BCS, and $f(x_i, \beta)$ is to be substituted with (1).

The IK problem can also be approximated using heuristic methods. These methods perform simple, iterative operations to gradually lead to an approximation of the solution. The heuristic algorithms have low computational cost (return the final pose very quickly), and usually support joint constraints. The most popular heuristic algorithms are: Cyclic Coordinate Descent (CCD), and Forward And Backward Reaching Inverse Kinematics (FABRIK) and Levenberg–Marquardt algorithm [25].

IV. Implementation architecture

At the first step of implementation prototyping, the Robotic Operating System (ROS) was used as an implementation framework. ROS provides standard operating system services, such as hardware abstraction, low-level device control, implementation of frequently used functions, message passing between processes, and package management. ROS is based on a graph architecture, where data processing takes place in nodes that can exchange messages among themselves. All these characteristics of the ROS framework make it suitable for use as a distributed computation framework.

The distributed multi-agent control architecture of the IK implementation is shown in Fig. 4. It is assumed that each Joint Controller computer runs IK and PID block controllers for its respected axis.

All individual components of the demonstration stand exchange data using the ROS communication protocols. Based on the current position of the manipulator grip, the trajectory generating unit sets a set of points, velocities and accelerations in the base coordinate system, which forms the target trajectory of movement for the manipulator gripping. After receiving the "Ready" signal from the controllers, the trajectory generation unit sends this data to the inverse kinematics (IK) nodes of each link to recalculate the coordinates of the trajectory points from the base coordinate system to the generalized coordinate systems of each link. Then, after recalculating the coordinates, each controller polls the states of other controllers, and if all controllers send a "Ready signal" i.e., they are ready to

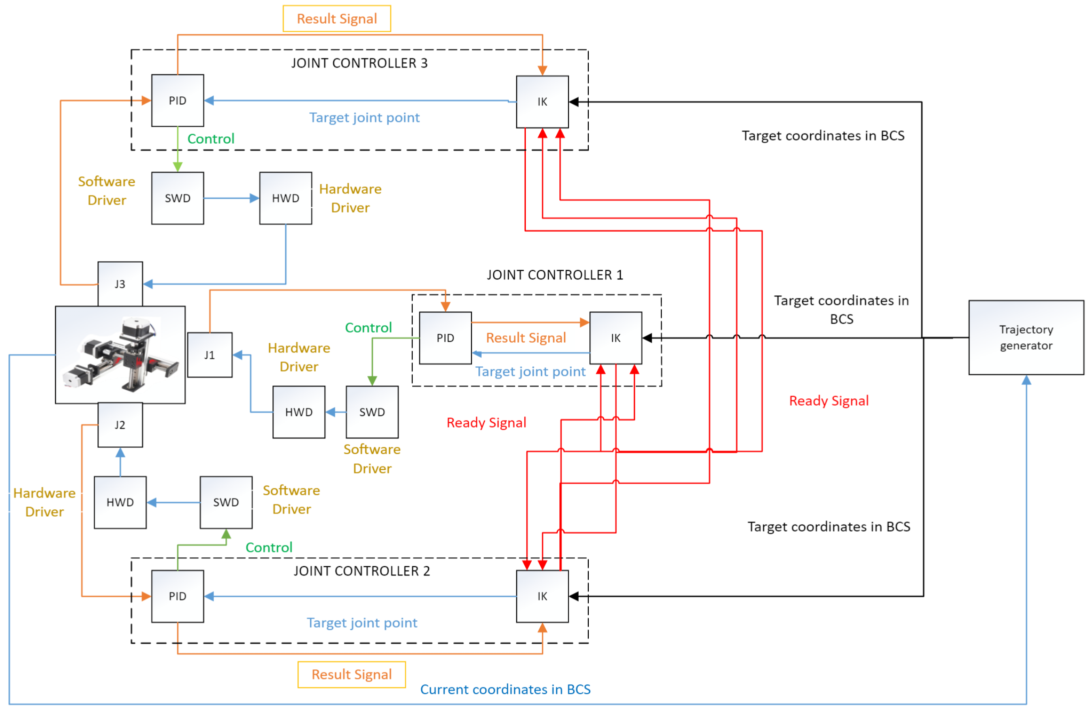

Fig. 4: Functional diagram of the architecture of the proposed distributed multi-agent control system.

send a command to their PID controllers. This command is a ROS-action "FollowJointTrajectory" that contains target point, speed and acceleration data. This way the controllers are synchronized from one point to another.

Using functions from the API of the CoppelliaSim simulator [26][3], PID regulators receive data on the current position, speed, moment of inertia of their link, respectively. And the drives are controlled using the $simSetJointPosition()$, $simSetJointTargetVelocity()$ and $simSetJointForce()$ functions. After the PID controllers bring their link to the target coordinate, a signal is sent back to the inverse kinematics node. Then the nodes of inverse kinematics are synchronized again and send the parameters of the next point of the trajectory to the regulators and the regulators again bring their link to the desired coordinate.

V. Demo use-case

The manipulator for the use-case demonstration was assembled from three axes FSL40XYZ-X(J1-J3) as shown in Fig.5.

Each axis has a stepper motor controlled by a TB6600 stepper motor hardware driver (HWD) and stepper motor software driver (SWD). The IK algorithm for each axis is deployed to a BeagleBoneBlack (BBB) micro-controller, three BBBs (one per axis) are communicating via Ethernet with each other and with a computer providing the target trajectory to follow.

[3] $simGetJointPosition()$, $simGetObjectFloatParameter(m_vrepJointsHandle, 2012, vel)$, $simGetJointForce()$

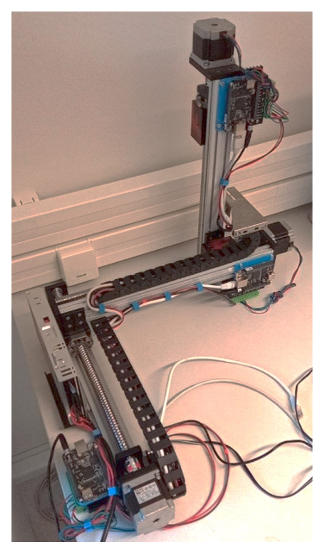

Fig. 5: Testbed: 3-axis manipulator with distributed control.

VI. Implementation in IEC 61499

The distributed control application was migrated from ROS joint control nodes to IEC 61499 function blocks.

Fig. 6 illustrates the migration by comparing the ROS nodes control application with the control application in IEC 61499. In ROS, there are three nodes (joint_controller) written in C++ that control the robot in the x, y and z axes. The control application for each axis is identical. The difference is the data extraction for trajectory input. Since each node is written in

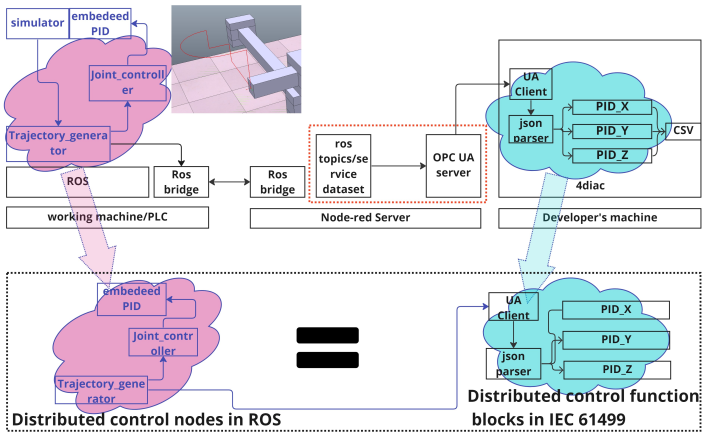

Fig. 6: Migrating distributed control of 3-axis robot from ROS nodes to IEC 61499.

C++ and the IEC 61499 supports creating function blocks also in C++, it is possible to create the same joint controller in the IEC 61499 as function blocks. To control the 3-axis robots moving along the same trajectory, the trajectory generated by "Trajectory_generator" in the ROS can be sent to the IEC 61499 via the OPC UA protocol. PID_X, PID_Y, and PID_Z can then use the same control code as the ROS nodes (joint_controller) to process the same generated trajectory received from the OPC UA server.

Based on the designed architecture, the implementation of this three-axis robot control in 4diac is shown in Fig. 7. The green boxes in the diagram divide the entire control application into six sections.

The starting section is the top left section, which function is to read the generated trajectory from the OPC UA server into the 4diac application for processing in subsequent FBs.

The function of the second part is to initialize the block and control the PID calculation of each target point on the trajectory. For each control loop, the first section passes each target to PID controllers in the third section.

The third section has three PID controllers that control the motion of the three-axis robot in the X, Y, and Z direction. For each target, each of the three PID controllers conducts closed-loop control, i.e., it received error feedback from the fourth section to correct the following output.

The fourth part is responsible for calculating the deviation between the distance of the PID output and the corresponding target point. The calculated error is used as feedback to the PID controller for output correction.

The fifth part is to check whether the current position reaches to the target point, i.e., whether the error between the current position and the target point is within the given error limit.

The last section records detailed data for each motion control in a CSV file, such as LMS, time, and index. Those recorded data will be used to visualize the motion path and PID controller performance in the section of VC and its evaluation.

VII. Virtual Commissioning Architecture

The controller in the ROS was commissioned via a Coppeliasim simulator. The performance of the commissioned control application in ROS may be different after being migrated to the IEC 61499 application. Many factors can affect the performance of a migrated IEC 61499 control application, such as communication between FBs, between runtimes, and between distributed controllers. Apart from different communication protocols in the 61499 application, one of the essential factors is VC against the PID controllers, i.e., PID tuning. This PID tuning is also one of the essential parts affecting the control performance. Therefore, in this section, VC refers to PID tuning of the 61499 control application.

Fig. 8 shows the workflow of the VC. In each loop of motion control, there are some variables generated for tuning PID controllers by analyzing its performance matrics:

- kp: Proportional term.
- ki: Derivative term.
- kd: Integral term.
- LIM_H: The highest limit of PID output.
- LIM_L: The lowest limit of the PID output.
- REACH_LIM: If the error is within this limit, it indicates that the current position reaches the target and should fetch the next target.
- Interval: It is the time interval for PID calculation.

Generated variables will be stored in OPC UA server so that the 4diac FBs can pull those variables and calculate performance metrics after each VC. For each experiment, only three variables will be randomly generated, e.g., kp, ki, and kd. Those variables are the main factors affecting performance.

Fig. 9 illustrates the visualized motion paths under various randomly generated PID parameters. In the figure, the path in blue is the original trajectory generated in ROS and stored in OPC UA server. And, all the paths in red are the results under different PID parameters. The results indicate the fact that PID tuning and VC are essential to the 3-axis control performance in IEC 61499. By analyzing and visualizing all the experimental results from VC, it is clear that to achieve the best control performance for the 3-axis robot in IEC 61499, VC is essential and helpful for finding the most optimal parameters for PID controllers.

For the detailed evaluation of VC, it is introduced in the following section.

VIII. Evaluation

The plug-and-play is to reduce the engineering effort for customers. In this 3-axis manipulator case, we enable plug-and-play for 3 identical joint motors, which are controlling XYZ respectively. Since those motors are identical, plug-and-play helps to just program and commission one motor and then clone commissioned control applications to others to achieve plug-and-play as an assembled system. By utilizing this method, it doesn't require additional engineering effort to assemble identical components multiple times.

Fig. 10 presents the experimental results of finding the most optimal PID parameters for one joint motor. The performance

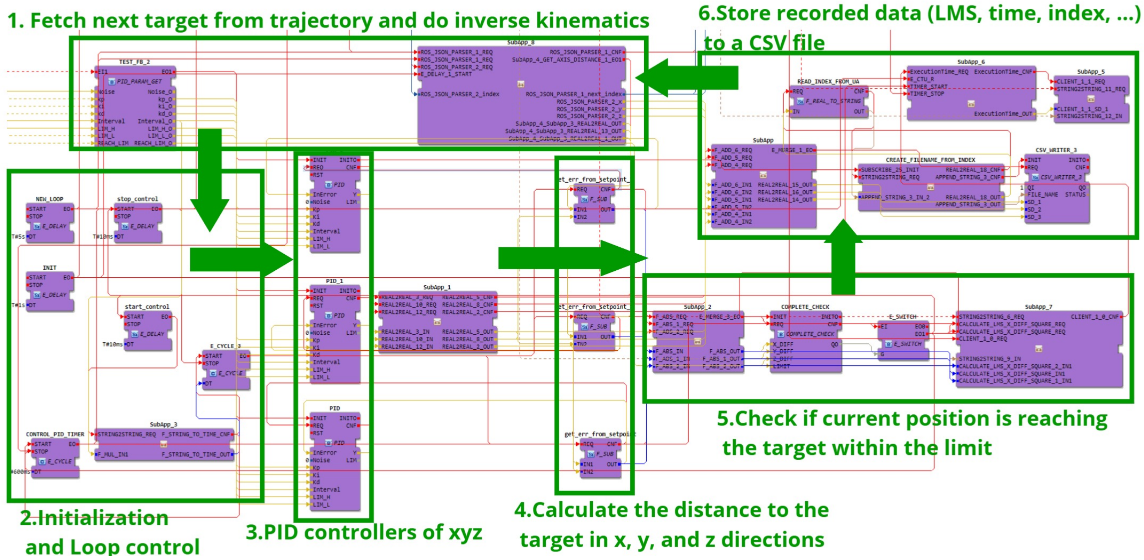

Fig. 7: The implementation of 3-axis robot control in IEC 61499 in 4diac IDE.

evaluation metric is the Lest Mean Square (LMS) value. The least LMS value corresponds to the best performance of joint control.

Fig. 11 illustrates the relationship between LMS and simulation execution time. The expectation during VC is if PID parameters have not been tuned optimally, the LMS value is high as well as the execution time since PID controllers need a longer time to control the end-effector to the target position. Via this figure, the results prove our hypothesis is correct that the more optimally commissioned controllers, the better performance and lest LMS can be achieved.

IX. Open questions and future work

A limitation of the current IK-solution is that it is calibrated to the configuration of mechatronic components. To solve such a problem in the conditions of a system configuration unknown to agents, it is necessary to somehow calculate the internal parameters of the manipulator, such as the parameters D-X for each link of the manipulator. That is, to perform a calibration after the new configuration is assembled.

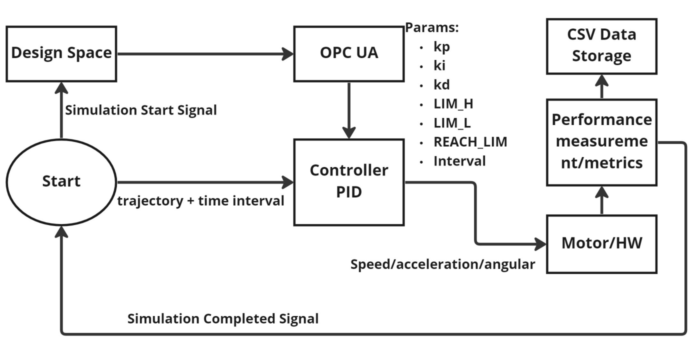

Fig. 8: The workflow of VC for the PID controllers in IEC 61499

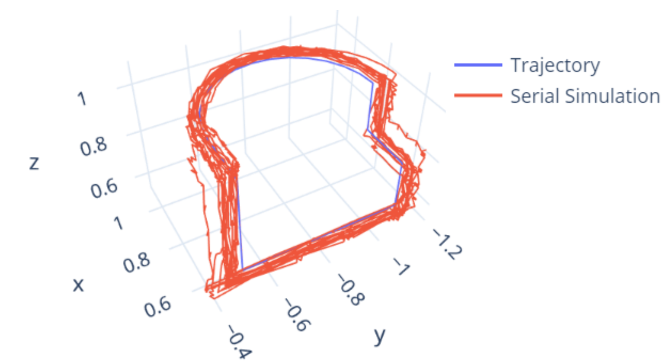

Fig. 9: Visualized 3-axis motion paths under various sets of variables

The future work will continue with the generalisation of the IK solution, demonstration of the plug-and-play capabilities, measuring and improving performance characteristics, and experimentation with the execution of the IEC 61499 application on different run-time platforms to demonstrate heterogeneity and portability. Extensive experimentation with various VC architectures and the use of wireless communication is also planned.

Acknowledgments

This work was supported, in part, by the HORIZON 2020 project 1-SWARM funded by the European Commission (grant agreement: 871743) and Horizon Europe project Zero-SWARM funded by the European Commission (grant agreement: 101057083). The authors are enormously grateful for the inspiration, collaboration and support of Friedrich Durand, Alois Zoitl and Gennady Veselov.

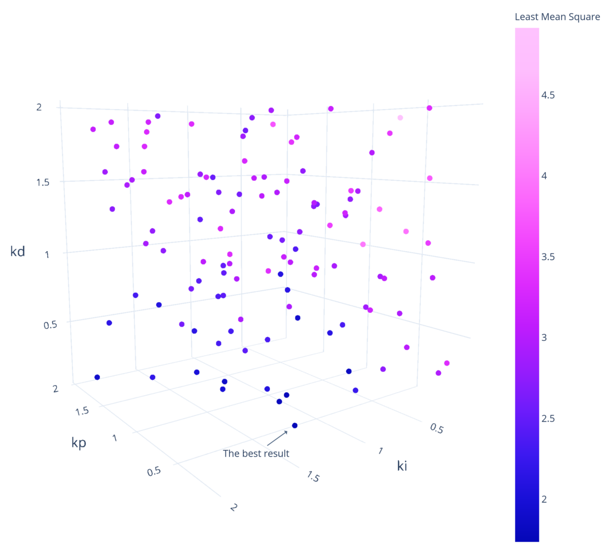

Fig. 10: Performance metrics LMS calculation via kp, ki, and kd.

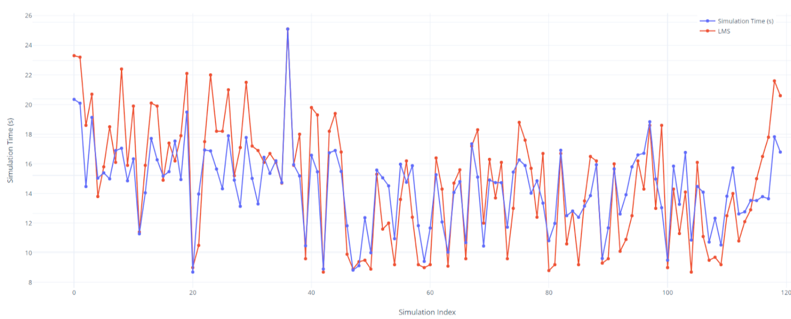

Fig. 11: The relationship between LMS and execution time.

REFERENCES

[1] J. Kennedy, "Swarm intelligence," in *Handbook of nature-inspired and innovative computing*. Springer, 2006, pp. 187–219.

[2] I. Kaliaev, S. Kapustjan, and D. Ivanov, "Decentralized control strategy within a large group of objects based on swarm intelligence," in *2011 IEEE 5th International Conference on Robotics, Automation and Mechatronics (RAM)*. IEEE, 2011, pp. 299–303.

[3] T. Arai, Y. Aiyama, Y. Maeda, M. Sugi, and J. Ota, "Agile assembly system by "plug and produce"," *CIRP Annals*, vol. 49, no. 1, pp. 1–4, 2000. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0007850607628832

[4] J. M. Lastra, *Reference mechatronic architecture for actor-based assembly systems*. Tampere University of Technology, 2004.

[5] P. Leitao, S. Karnouskos, L. Ribeiro, J. Lee, T. Strasser, and A. W. Colombo, "Smart agents in industrial cyber–physical systems," *Proceedings of the IEEE*, vol. 104, no. 5, pp. 1086–1101, 2016.

[6] L. Ribeiro, J. Barata, M. Onori, and J. Hoos, "Industrial agents for the fast deployment of evolvable assembly systems," in *Industrial Agents*. Elsevier, 2015, pp. 301–322.

[7] IEC 61499, "Function blocks–Part 1: architecture, second edition," *International Electrotechnical Commission, Geneva, Switzerland*, 2012.

[8] V. Vyatkin, "The IEC 61499 standard and its semantics," *IEEE Industrial Electronics Magazine*, vol. 3, no. 4, pp. 40–48, 2009.

[9] T. Lyu, U. D. Atmojo, and V. Vyatkin, "Towards cloud-based virtual commissioning of distributed automation applications with iec 61499 and containerization technology," in *IECON 2021–47th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2021, pp. 1–7.

[10] J. Cabral, M. Wenger, and A. Zoitl, "Enable co-simulation for industrial automation by an fmu exporter for iec 61499 models," in *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1. IEEE, 2018, pp. 449–455.

[11] W. Dai, Y. Zhang, L. Kong, J. H. Christensen, and D. Huang, "Design of industrial edge applications based on iec 61499 microservices and containers," *IEEE Transactions on Industrial Informatics*, 2022.

[12] M. Sorouri, V. Vyatkin, S. Xie, and Z. Salcic, "Plug-and-play design and distributed logic control of medical devices using iec 61499 function blocks," *International Journal of Biomechatronics and Biomedical Robotics*, vol. 2, no. 2-4, pp. 102–110, 2013.

[13] M. Sorouri, S. Patil, Z. Salcic, and V. Vyatkin, "Software composition and distributed operation scheduling in modular automated machines," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 4, pp. 865–878, 2015.

[14] R. Sinha, V. Vyatkin, Z. Salcic, and H. J. Park, "Competitors or cousins? studying the parallels between distributed programming languages systemj and iec61499," in *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*. IEEE, 2014, pp. 1–7.

[15] C. Dripke, Z. Ye, and A. Verl, "Synchronization of a distributed interpolation application via cross-coupled control," in *2019 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*. IEEE, 2019, pp. 1–8.

[16] M. Beller, B. Schneider, and A. Zoitl, "Skill-based motion control with opc ua and deterministic ethernet," in *Computer Aided Systems Theory–EUROCAST 2019: 17th International Conference, Las Palmas de Gran Canaria, Spain, February 17–22, 2019, Revised Selected Papers, Part II 17*. Springer, 2020, pp. 461–468.

[17] D. Oliveira, J. Pinheiro, L. Neto, V. H. Pinto, and G. Gonçalves, "A plug-and-play solution for smart transducers in industrial applications based on ieee 1451 and iec 61499 standards," *Sensors*, vol. 22, no. 19, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/19/7694

[18] V. Jirkovský, M. Obitko, P. Kadera, and V. Mařík, "Toward plugplay cyber-physical system components," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 6, pp. 2803–2811, 2018.

[19] S. Profanter, A. Perzylo, M. Rickert, and A. Knoll, "A generic plug produce system composed of semantic opc ua skills," *IEEE Open Journal of the Industrial Electronics Society*, vol. 2, pp. 128–141, 2021.

[20] X. Ye, J. Jiang, C. Lee, N. Kim, M. Yu, and S. H. Hong, "Toward the plug-and-produce capability for industry 4.0: An asset administration shell approach," *IEEE Industrial Electronics Magazine*, vol. 14, no. 4, pp. 146–157, 2020.

[21] C.-W. Yang, J. Yan, and V. Vyatkin, "Towards implementation of plug-and-play and distributed hmi for the freedm system with iec 61499," in *IECON 2013-39th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2013, pp. 5347–5353.

[22] W. Dai, W. Huang, and V. Vyatkin, "Enabling plug-and-play software components in industrial cyber-physical systems by adopting service-oriented architecture paradigm," in *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2016, pp. 5253–5258.

[23] R. P. Paul, *Robot manipulators: mathematics, programming, and control: the computer control of robot manipulators*. Richard Paul, 1981.

[24] A. Aristidou, J. Lasenby, Y. Chrysanthou, and A. Shamir, "Inverse kinematics techniques in computer graphics: A survey," in *Computer graphics forum*, vol. 37, no. 6. Wiley Online Library, 2018, pp. 35–58.

[25] A. Nubiola and I. A. Bonev, "Absolute calibration of an abb irb 1600 robot using a laser tracker," *Robotics and Computer-Integrated Manufacturing*, vol. 29, no. 1, pp. 236–245, 2013.

[26] (2023) Coppeliasim simulator. [Online]. Available: https://www.coppeliarobotics.com/