2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)
June 28-30, 2023. Seattle, Washington, USA

A Digital Twin Framework for Virtual Re-Commissioning of Work-Drive Systems Using CAD-based Motion Co-Simulation

1st Rémy Carlier[1,2]
Remy.Carlier@UGent.be

2nd Joris Gillis[3,4]
Joris.Gillis@KULeuven.be

3rd Erwin Rademakers[5]
Erwin.Rademakers@FlandersMake.be

4th Gianni Borghesan[3,6]
Gianni.Borghesan@KULeuven.be

5th Pieter De Clercq[5]
Pieter.DeClercq@FlandersMake.be

6th Chris Ganseman[5]
Chris.Ganseman@FlandersMake.be

7th Kurt Stockman[1,2]
Kurt.Stockman@UGent.be

8th Jeroen D. M. De Kooning[1,2]
Jeroen.DeKooning@UGent.be

[1]*Department of Electromechanical, Systems & Metal Engineering*, Ghent University, Belgium
[2]*FlandersMake@UGent - Corelab MIRO*, Belgium
[3]*Department of Mechanical Engineering*, KU Leuven, Belgium
[4]*FlandersMake@KULeuven - Corelab MPRO*, Belgium
[5]*MotionS*, Flanders Make, Belgium
[6]*FlandersMake@KULeuven - Corelab M&A*, Belgium

***Abstract*—Modular systems offer increased system versatility and efficiency. Work-drive systems are a subset of mechatronic systems, often modular, featuring a working function and a driving function. Efficient module changes are a core challenge of modularity as software adaptations are needed to accommodate the various module functionalities, geometries and dynamics. Traditional virtual commissioning reduces the development time of products and systems. Additionally, quality improvements are achieved by virtue of earlier and more thorough testing in a risk-free environment. Specifically, control software developed and tuned using a system's virtual replica can be deployed to the system with little or no modifications. Current virtual commissioning solutions lack automated capabilities or solely employ kinematic models. The digital twin framework proposed in this paper for automated virtual re-commissioning uses a dynamic multibody model to adapt software during module changes in an operating environment with little to no help from a human operator. The proposed framework is demonstrated in simulation for low-level control loop re-commissioning. A sample task sequence is executed in three test cases: 1) baseline 2) system parameter change 3) low-level controller re-tune. The simulation results show an increase in tracking error in 2) compared to 1), illustrating the need for controller re-tune. This subsequently brings back similar performance as the baseline case, i.e. a decrease of 10% tracking error in 3) compared to 2).**

***Index Terms*—Digital twins, Virtual commissioning, Work-drive systems, CAD, Motion simulation**

I. Introduction

In recent years, there has been a demand from the manufacturing industry for increased efficiency and versatility of production lines [1], [2]. Mechatronic systems are widely used in this field and are historically integrated systems with proprietary components. Modern modular mechanic systems allow quick re-configuration and component sharing. Advantages of modular mechatronic systems include reduced equipment acquisition costs, improved work flexibility and more optimal use of time and physical resources. Work-Drive systems are a sub-type of mechatronic systems featuring both a driving function and a working function. They are prime candidates to benefit from modularity. Examples of such systems are a robotic arm with end-effector as the working function, attached to an automated guided vehicle (AGV) as the driving function. In this paper, the studied system is a 6 degree of freedom (DOF) robotic arm attached to a sliding linear platform. Fig. 1 shows the CAD representation of this modular Work-Drive system.

A particular challenge of modular industrial systems is time efficient module changes. Module swaps imply changes in system kinematics and dynamics. These changes call for adaptations of control parameters. For simple systems with few module swaps and when modules are well known, state of the art adaptive control may be successfully employed [3]. However, when the number of compatible modules is large and when this pool of modules is likely to constantly evolve, system re-commissioning is needed. This means model and control parameters must be adapted. Traditionally, a human operator familiar with the system would do this. However, there is potential to accelerate this process through automated virtual re-commissioning.

Increased computing power, access to large amounts of data and high-end software tools have been a driving force in

This research is part of the Work-Drive SBO project funded and supported by Flanders Make, the strategic research centre for the manufacturing industry.

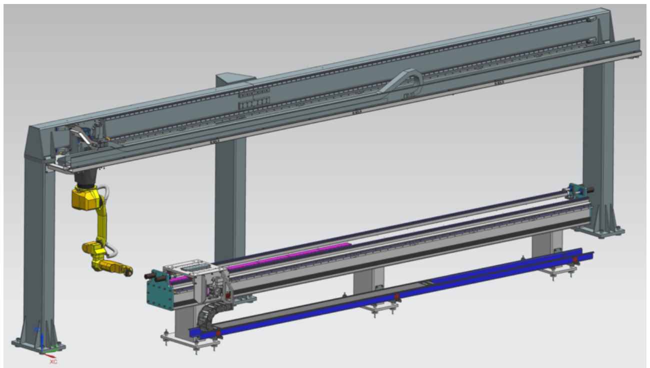

Fig. 1. CAD rendering of the linear track & 6 DOF robot arm

Industry 4.0 [4]. Central to this industrial revolution is the digitalization of processes and systems. The traditional design cycle of products and systems called for time intensive use of costly prototypes. Virtual commissioning has been used in recent years as a means of accelerating this design cycle [5]. Virtual commissioning can be defined as the verification and validation of system components with their control algorithms in simulation before implementation on the physical system [6]. These simulations can be software-in-the-loop (SIL) or hardware-in-the-loop (HIL). The appeal of virtual commissioning is reduced development and validation time, the possibility for more thorough testing with increased variety of validation cases, the ability to test the system before the hardware is available, increased safety and reduced risk of damage during physical commissioning later on [6]. In [7], it was shown experimentally that virtual commissioning increased the quality of a control logic program while simultaneously reducing the debugging time. In complement to digitalizing the commissioning cycle, the system supervision can also be digitalized. A trending way of doing so is through a digital twin (DT). Digital twins are typically linked with Internet of Things, data driven modelling and recent developments in communication technology [8]. Typical digital twin tasks are real-time monitoring, optimal scheduling, model-based control, prediction, fault diagnosis and predictive maintenance [9], [10]. A network of digital twins may even expand the capabilities of smart factories as presented in [11].

Previously, [12] developed a SIL simulation environment to test and validate human-robot collaboration for AGVs. The authors in [13] used virtual reality to commission multi-robot manufacturing cells. These two applications used the game engine Unity to create the virtual reality representation of systems. A digital twin for a robot arm was developed in [14] using the same gaming and VR tools. The kinematic control was validated against the real robot. However, these three applications lack mechanical dynamic modelling, i.e. they only include kinematic motion and do not account for forces in the system. In [15], a TwinCAT PLC motion program for a 6 DOF robot is virtually commissioned through ABB's RobotStudio software. RobotStudio makes use of rigid body dynamics but does not allow modifications to the torque, velocity and position controllers. [6], proposes a CAD-based model for virtual commissioning of a delta pick and place robot. The trajectory tracking is then tested against the physical system. However, this CAD-based model approach fails to provide real-time supervisory capabilities. Current modular mechatronic systems either lack automated virtual re-commissioning or solely consider kinematics when re-commissioning software components. The proposed digital twin framework for virtual re-comissioning contributes to existing literature by featuring a CAD-based multibody model that allows automated re-commissioning of low-level control loops during module changes. Fig. 2 graphically presents this framework for the 6 DOF robot & linear track system. The virtual replica is comprised of a CAD-based co-simulation and Docker containers hosted in an Ubuntu virtual machine. When this virtual replica runs in parallel to the system, it offers valuable insights. For typical mechatronic systems, virtual replicas monitor tracking errors, working point and parameter changes in addition to component condition using sensor signals. A supervising entity then acts based on these observations to optimally adapt control parameters, end a process and signal the repair or replacement of components. In the proposed framework, it will signal the need for virtual re-commissioning of software components. Specifically in this paper, the re-commissioning of low-level controllers will be demonstrated.

This paper is structured as follows: the NX - Simulink co-simulation is presented in § II-C, while the task control algorithm and trajectory planner are presented in § II-D and § II-E. § II-F describes the world model and object tracking. Simulation results and the accompanying discussions are presented in § III. Conclusions and future research topics are available in § IV.

II. Model description

A. Application description

The Work-Drive system of interest is comprised of a 6 DOF robot arm attached to a sliding linear track. A second linear track with sliding platform is located underneath and emulates a conveyor belt. Fig. 1 shows a CAD representation of this experimental setup. The robotic arm and its end-effector execute work functions such as gripping objects from the bottom linear track. Various end-effectors can be attached to the robotic arm, enabling a wide range of work functions to be performed. The top linear track executes the drive function. During task execution, the robot arm and end-effector must avoid collisions with the two tracks and their supporting elements. Two ceiling-mounted cameras and a robot arm mounted camera monitor the position of the target object. The latter can then be placed at various locations on the sliding platform, replicating the stochastic positioning of objects on a conveyor belt in certain industrial applications.

B. Architecture

In Fig. 3, the developed software architecture is shown. It is based on two virtual machines (VMs) running on a

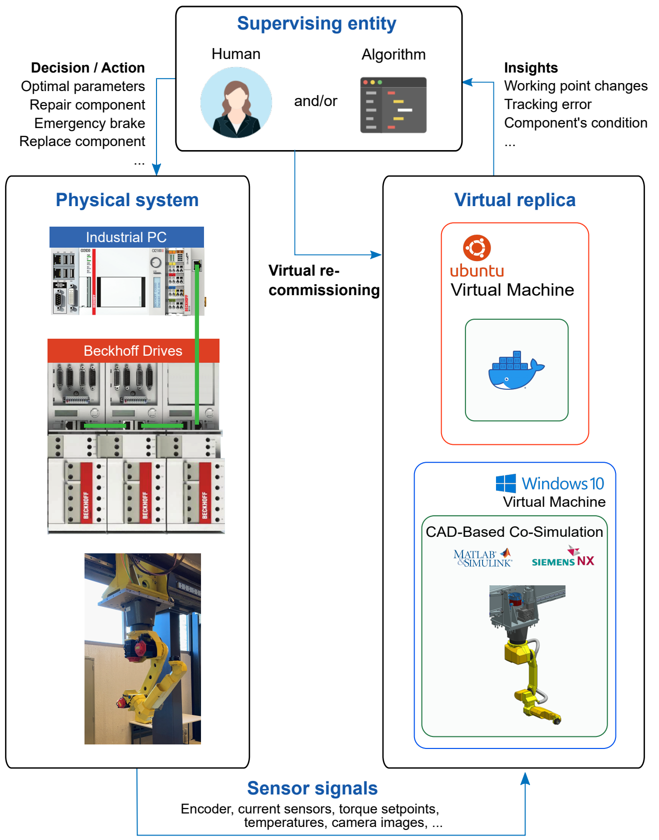

Fig. 2. Proposed Digital Twin Framework for virtual commissioning. Human and algorithm icons by shivaniga_jipara and mithun on https://www.freeicons.io (accessed 10 November 2021).

common host computer. The Ubuntu VM contains the optimal trajectory planner, the task controller, the world model and the web graphical user interface (GUI). These software modules run as Docker images. The first three are coded in Python and the last in HTML5. After virtual commissioning, they are implemented on the real system with minimal structural modifications. The Simulink - Siemens NX co-simulation runs inside the Windows 10 VM. Siemens NX runs a CAD based motion simulation while Simulink runs the cascaded velocity and position control loops. The two VMs run on a single physical machine and are forced onto the same IP range. The web GUI, task controller, trajectory planner, world model and Simulink each have their own ROS2 node. They publish and subscribe to each other's ROS2 topics. This is denoted by the pink lines in Fig. 3. All messages are time-stamped, each computer clock is periodically re-synchronized, ensuring system synchronization. On the physical system, it is proposed that the virtual replica shown in Fig. 3 runs in parallel to the real system along with a supervising entity as displayed in Fig. 2. When this supervising entity detects changes, for example a full module swap, the virtual replica is used to re-commission the controllers for the new system kinematics and dynamics. The new parameters are then sent to the physical system, on the left side of Fig. 2. During the time re-commissioning takes place, the physical system enters a safe mode with stricter motion constraints. Once re-commissioning is fully automated, this would be in the order of tens of seconds.

C. Co-Simulation & Low-Level Control

For virtual commissioning of control algorithms, a highly realistic model is required. In this application, the CAD software Siemens NX was employed as it features multibody dynamics solvers that accurately model the system forces and movements. In Siemens NX, the motion bodies and joints are parametrized. Additionally masses, moments of inertia and friction characteristics are tuned. Finally, the NX block is prepared for co-simulation by specifying the timestep and the block in- and outputs. It is compiled into a Simulink block and integrated into the control scheme. The outputs of the NX block are the velocity Θ and position Ω of each controlled joint, emulating the output of each joint's drive motion object. The inputs are torque commands of the joints T_{SP}, the same as are fed to the servo drives.

Given an accurate model and sufficient stability margins, the low-level control scheme presented in Fig. 3 can be transferred from the virtual commissioning environment to a PLC or drive piloting the physical system. This control scheme is a state of the art servo drive positioning loop with cascaded position and velocity feedback controllers. A velocity feedforward action Ω_{FF} improves the tracking performance for spline shaped trajectories. The sampling rates of the signals and controllers are selected to be the same as those implemented inside the servo drives. Controller saturations enforce actuator limits, i.e. torque and velocity are constrained to stay within motor and gearbox operating limits. The trajectories received from the optimal trajectory planner are bounded to stay within track and joint positioning limits.

D. Task-Level controller

The task controller (TC) configures the trajectory planner described in § II-E using pre-defined application description files. The TC itself takes instructions from an operator through an GUI. Communication in and out of the TC happens through the remote procedure call (RPC) protocol implemented in the *ActionLib* inter-process communication mechanism of ROS2. Specifically, the TC sends the planner as a `json` file. In this file, the states, initial states, transitions and RPC parameters are specified.

Petri Nets are a well-known formalism to represent plans where concurrent activities take place. In the case of the TC, the simultaneous actions are the work and drive functions. Petri Nets are characterised by places and transitions. Tokens in each place represent the state of the system. In this case, the number of tokens present is either 0: no action is requested, 1: an action is requested in which a single planning action coordinates both the work and drive actors, 2: work and drive are commanded to act in a concurrent way. The planner signals the end of the action. The Petri Net is represented with a graph, encoded with the *dot* language. This is shown in Fig. 4.

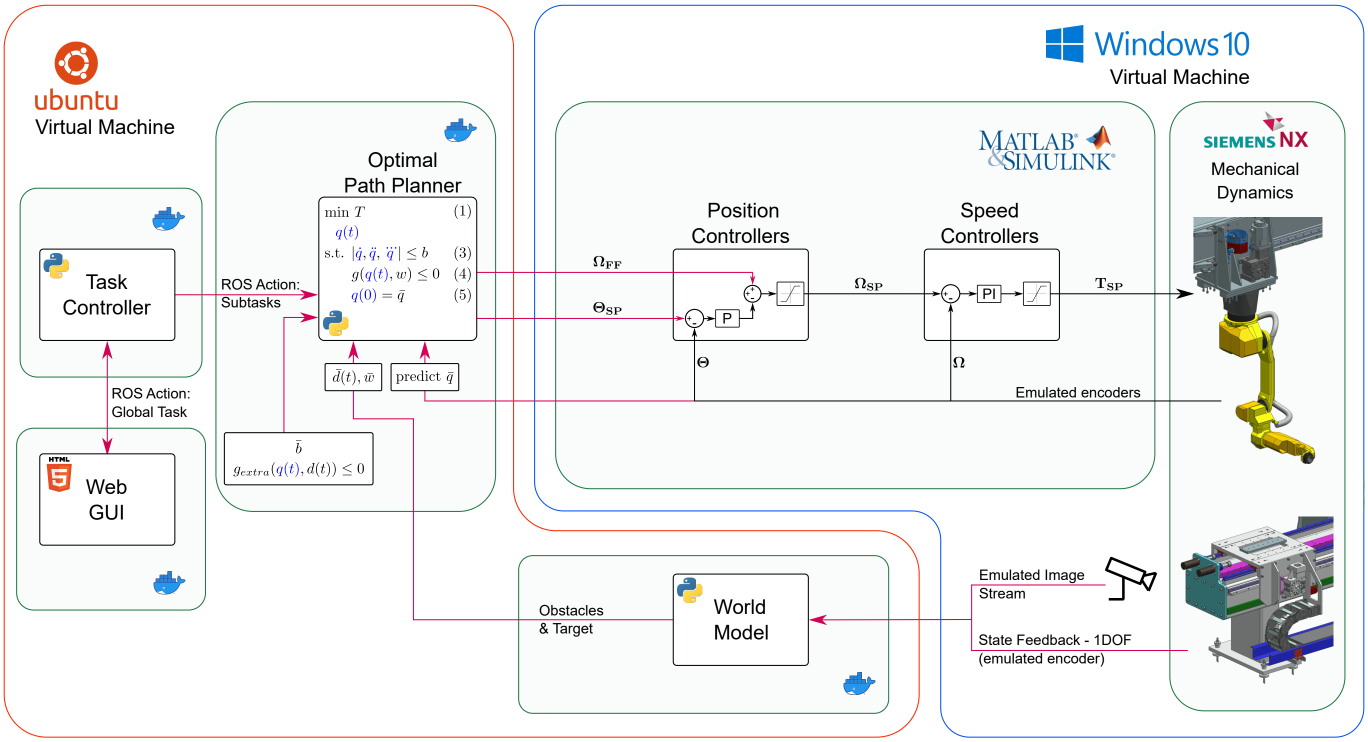

Fig. 3. Schematic representation of the control scheme developed for the virtual commissioning of mechatronic systems

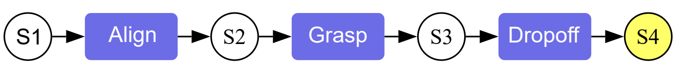

Fig. 4. Petri Net representing the example plan reported in § III. The state reported is the final step, when one Token is state `s4`

E. Optimal Trajectory Planner

Trajectory planning is approached as a nonlinear optimal control problem (OCP) with a free end-time that is numerically solved. The various tasks commanded by the task-level controller of § II-D give rise to concrete numerical settings for the parametric constraints of a template OCP, next to a possibility for structural additions to the set of constraints. The template OCP is comprised of:

1) the planning horizon time T as objective
2) a kinematic model of the linear track derived from the CAD dynamic model & robot arm
3) trajectory constraints that limit joint positions, velocities, accelerations and jerks to arbitrary bounds b
4) nonlinear trajectory constraints for collision avoidance, with input robot pose $q(t)$ and obstacles w
5) a boundary constraint for the initial configuration $\bar{q}$ of the robot

In a task to align a gripper with a (moving) target d, the structural additions consist of boundary constraints on the position, velocity and orientation misalignment error at the end of the planning horizon. When the planner is started, a prediction is made about the configuration of the robot at that time in the future where a numerical solution to the planner OCP is expected to be finished. This prediction is taken as the numerical value of the parametric constraint 5) in the template. Time-stamped messages from the world model inform the planner about current and past poses and velocities of the robot, target and obstacles.

The specification of the OCP is performed using Rockit [16] which implements various numerical solution strategies. A direct approach is used (first discretise, then optimise) with IPOPT [17] to solve the resultant nonlinear program, and CasADi [18] to handle the algorithmic differentiation and code-generation of the required constraint and objective evaluation functions and their derivatives. These compiled evaluation functions are parametric and hence constructed offline for each possible task individually. Since the kinematic model gives rise to a trivial set of differential equations, the system dynamics are eliminated from the OCP. Instead, joint position trajectories $q(t)$ are represented directly as 4^{th} order B-splines of which the coefficients serve as decision variables for the nonlinear program. As collision constraints, signed distances between robot links and obstacles w are computed (both represented as capsules) and summarise non-collision into a single scalar trajectory constraint using the LogSumExp function [19].

F. World Model

Planning and control modules require knowledge about the physical environment in order to retrieve suitable key points that avoid collisions with (potentially dynamic) obstacles. Environment knowledge can come from many sources of information. Static objects that belong to the setup can be measured a priori and collected in a static map, while moving or unknown objects need to be monitored during execution using perception sensors such as cameras. The task of the world model is to consume all these sources of information and abstract the data into a structured and semantic representation of the environment. As such, it needs to fuse incoming object measurements that stem from multiple sensors with model-based object predictions provided by the world model itself, based on previous measurements and/or prior information. In the digital twin, the input data streams are provided to the world model module by simulated cameras with object detection capabilities, using the same communication paradigm as in the physical setup. This allows to easily interchange between simulation and real-life operation, without any adaptations to the world model needed.

Additional to the sensor fusion capabilities, the world model also acts as a database server that can be queried for information. Indeed, the goal of the world model is to provide the planning and control modules with the necessary information about the environment, i.e. planner constraints. To this end, several services are implemented to retrieve relevant information about obstacles from specific classes, obstacles that are located in a certain spatial region of interest or even prediction of obstacle positions for a given timestamp.

III. Results

To illustrate the benefits of virtual re-commissioning, a sample task sequence was run using the full simulation solution. This task sequence is comprised of an aligning task during which the robot's end-effector is aligned with the moving target then tracks it, a pick task during which the target object is grasped and a drop-off task to a pre-determined location. Three test cases are performed. 1) is the baseline in which the controller parameters are tuned to the nominal robot arm parameters. 2) simulates a work module change, the robot arm's links and platform are made heavier by a factor 1.3. During 2) the same task sequence is ran with the baseline controller parameters. In 3) the low-level controllers are re-tuned to the new system dynamics. Other modifications to the work and drive functions like kinematic parameters, additional degrees of freedom or a different end-effector would necessitate virtual re-commissioning of the task-level, planner and world model modules. However, the focus of this paper is to showcase virtual re-commissioning of the low-level controllers.

The low-level controllers are evaluated based on their tracking performance. Due to the multi-joint motion of this modular system, this tests both the setpoint tracking performance and disturbance rejection of the controllers. Table I presents the root mean squared error (RMSE) of the position controller for each joint and for the three test cases. The tracking performance of the position controllers are used since they are the external controllers in the position-velocity controller cascade. A poor tracking performance of the internal (velocity) controller will be apparent in the external (position) controller's tracking RMSE.

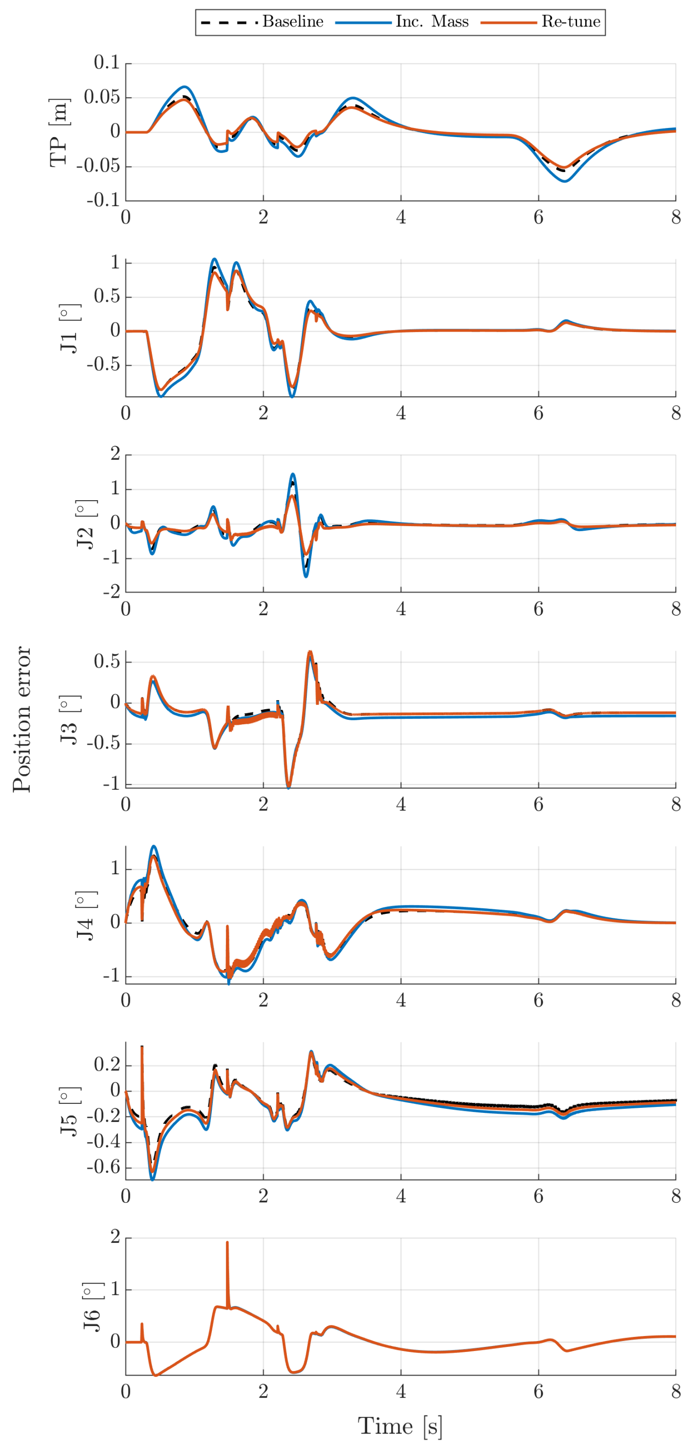

Fig. 5. Position tracking error for the task sequence described in Fig. 4

Decreased setpoint tracking performance is illustrated graphically in Fig. 5. Whereas the blue curve (increased weight case) has larger deviation from the zero axis, indicating larger tracking error, than the baseline case (black dashed line) for all joints except J6. The latter has very low weight and is a simple cylinder spinning around its revolution axis. Therefore the effects of a mass increase are smaller compared to the other joints, i.e. the three curves are super-imposed in Fig 5. Re-tuning the position and velocity controllers of each joint to have similar overshoot and settling time as the baseline case reduces the RMSE of each joint to within 10 percent or lower than the baseline value. In Fig. 5, this is shown by the orange line (re-tuned controllers) being closer or super-imposed to the black curves of the baseline case. From Table I, it is apparent when comparing the baseline case to the increased link mass case that the modified system dynamics degrade the tracking performance as indicated by the increased RMSE for all joints. Specifically, the RMSE increases by at least 10 percent for all joint except J5 and J6, the two smallest and lightest joints.

Without re-commissioning the low-level controllers, tracking performance is degraded, this means rougher positioning accuracy and increased time to reach position setpoints. For industrial applications the combination of these two have further consequences on overall process efficiency, mainly due to increased task completion time and increased failure rate.

TABLE I
ROOT MEAN SQUARED ERROR OF TEST CASES [10^{-3}]

Joint Number	Baseline	Increased Mass	Controllers Re-Tune
TP [mm]	22.4	28.5	20.4
J1 [°]	317.6	359.2	313.2
J2 [°]	239.7	296.0	219.3
J3 [°]	205.1	225.9	205.8
J4 [°]	363.0	424.5	362.6
J5 [°]	138.9	182.0	156.6
J6 [°]	279.8	279.9	279.8

IV. CONCLUSIONS

The benefits of virtual re-commissioning were illustrated for work-drive systems using a multibody CAD-based simulation framework. First, the commissioning of software components was described. Then a concrete example of virtual re-commissioning is presented for the low-level controller. Re-commissioning allowed position tracking to return to the baseline performance after increasing by 10% or more when changing the system dynamics, i.e. increasing the weight of each joint by 30%. However, virtual re-commissioning is not solely limited to low-level controller tuning. Certain system changes such as end-effector changes may require adjustments to other software components. Future work will focus on implementation on a live system and automated detection & re-commissioning.

REFERENCES

[1] C. A. Barrera Diaz, T. Aslam and A. H. C. Ng, "Optimizing reconfigurable manufacturing systems for fluctuating production volumes: A simulation-based multi-objective approach," IEEE Access, vol. 9, pp. 144195–144210, Oct. 2021.

[2] J. Wan, X. Li, H. -N. Dai, A. Kusiak, M. Martinez-Garcia and D. Li, "Artificial-intelligence-driven customized manufacturing factory: Key technologies, applications, and challenges," Proceedings of the IEEE, vol. 109, pp. 377–398, Apr. 2020.

[3] A. M. Annaswamy and A. L. Fradkov, "A historical perspective of adaptive control and learning," Annual Reviews in Control, vol. 52, pp. 18–41, Nov. 2021.

[4] R. Y. Zhong, X. Xu, E. Klotz and S. T. Newman, "Intelligent manufacturing in the context of Industry 4.0: A review," Engineering, vol. 3, pp. 616–630, Nov. 2017.

[5] T. Lechler, E. Fischer, M. Metzner, A. Mayr and J. Franke, "Virtual commissioning – scientific review and exploratory use cases in advanced production systems," Procedia CIRP, vol. 81, pp. 1125–1130, June 2019.

[6] M. Noga, M. Juhás and M. Gulan, "Hybrid virtual commissioning of a robotic manipulator with machine vision using a single controller," Sensors, vol. 22, Feb. 2022.

[7] M. Schamp, S. Hoedt, A. Claeys, E. H. Aghezzaf and J. Cottyn, "Impact of a virtual twin on commissioning time and quality," IFAC-PapersOnLine, vol. 11, pp. 2405–8693, Sept. 2018.

[8] M. Liu, S. Fang, H. Dong and C. Xu, "Review of digital twin about concepts, technologies, and industrial applications," Journal of Manufacturing Systems, vol. 58, pp. 346–361, Jan. 2021.

[9] Z. Huang, Y. Shen, J. Li, M. Fey and C. Brecher, "A survey on AI-driven digital twins in Industry 4.0: Smart manufacturing and advanced robotics," Sensors, vol. 21, Sept. 2021.

[10] J. D. M. De Kooning, K. Stockman, J. De Maeyer, A. Jarquin-Laguna and L. Vandevelde, "Digital twins for wind energy conversion systems: a literature review of potential modelling techniques focused on model fidelity and computational load," Processes, vol. 9, Dec. 2021.

[11] Y. Wu, K. Zhang and Y. Zhang, "Digital twin networks: A survey," IEEE Internet of Things Journal, vol. 8, pp. 13789–13804, Sept. 2021.

[12] C. Allmacher, M. Dudczig, S. Knopp and P. Klimant, "Virtual reality for virtual commissioning of automated guided vehicles," in 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), Mar. 23 - Mar. 27, 2019.

[13] L. Pérez, S. Rodrìguez-Jiménez, N. Rodriguez, R. Usamentiaga and D. F. Garcia, "Digital twin and virtual reality based methodology for multi-robot manufacturing cell commissioning," Applied Sciences, vol. 10, May 2020.

[14] G. Garg, V. Kuts, and G. Anbarjafari, "Digital twin for FANUC robots: Industrial robot programming and simulation using virtual reality," Sustainability, vol. 13, Sept. 2021.

[15] R. Raffaeli, P. Bilancia, F. Neri, M. Peruzzini and M. Pellicciari, "Engineering method and tool for the complete virtual commissioning of robotic cells," Applied Sciences, vol. 13, Mar. 2022.

[16] J. Gillis, B. Vandewal, G. Pipeleers, and J. Swevers, "Effortless modeling of optimal control problems with rockit," in 39th Benelux Meeting on Systems and Control, Date: 2020/03/10-2020/03/12, Location: Elspeet, The Netherlands, 2020.

[17] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," Mathematical programming, vol. 106, no. 1, pp. 25–57, 2006.

[18] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," Mathematical Programming Computation, vol. 11, no. 1, pp. 1–36, 2019.

[19] J. Gillis, G. Pipeleers, and J. Swevers, "Effortless modeling of optimal control problems with rockit," in 41th Benelux Meeting on Systems and Control, Date: 2020/07/05-2020/07/07, Location: Brussels, Belgium, 2022.