Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 133 (2025) 114–119

www.elsevier.com/locate/procedia

20th CIRP Conference on modelling of Machining Operations

Performance and efficiency of co-simulation for milling operations in robotic machining

Hugo Dantinne[a,*], Valentin Dambly[b], Olivier Verlinden[a], François Ducobu[b], Edouard Riviere-Lorphevre[b], Bryan Olivier[a]

[a] *Department of Theoretical Mechanics, Dynamics and Vibrations, University of Mons 20, Place du Parc, 7000 Mons, Belgium*
[b] *Department of Machine design and Production Engineering, University of Mons, 20, Place du Parc, 7000 Mons, Belgium*

* Corresponding author *E-mail address: hugo.dantinne@umons.ac.be*

Abstract

In robotic machining, tool/workpiece interaction generates forces that impact the performance of the robotic arm. This leads to deflections due to its lower stiffness compared to dedicated machining machines, particularly at its joints. These deflections can degrade the quality of the machining operation, making trajectory and orientation optimization essential. Accurate optimization requires knowledge of the transient force field for the operation and a model of the robot. Currently, the simulations are performed using two in-house simulation frameworks : *EasyDyn* for the multibody dynamics of robot modelling and *Dystamill* for the modelling of the cutting forces. To enhance the simulation's modularity, co-simulation between the multibody system (MBS) of the robot and the workpiece (with calculation of cutting forces for the latter) is proposed, allowing for better integration of the coupled dynamics. The decision on where to partition the model is critical. Given that the workpiece transmits force elements while the robot transmits position data, a displacement-force coupling approach is proposed. Co-simulation offers several advantages, including the ability to use dedicated software tailored to different parts of the model and facilitating the interfacing of various robots with different machining models. However, the modularity offered by co-simulation techniques is counterbalanced by less accurate results with respect to those obtained with monolithic models. This research aims to evaluate the influence of the accuracy of the results on a simple model.

Keywords: Co-Simulation; Robotic Machining; Multibody Dynamics simulation, Tool/Workpiece Interaction

1. Introduction

In recent times, industrial robots (IRs) have been widely used for various tasks such as assembly, quality inspection but also machining [1, 2, 3]. Growing interest for IRs is due to easier setup on production lines, higher adaptability and working space, a wider range of use, and lower costs than dedicated machining machines [4, 5]. However, it has been determined that IRs are 50 times less rigid than conventional milling machines (Computer Numerical Control - CNC) due to their open-loop structures [6]. This flexibility is an obstacle on the use of IRs in machining as it leads to deflections of the robot, which increases issues in achieving dimensional and geometrical tolerances. A second issue linked to the lack of stiffness of those machines is self-excited vibrations. They can cause lower accuracy and workpiece surface quality, as well as lowering tool life [4, 7], and therefore, must be eliminated. In order to compensate for these issues, extensive research has been conducted on improving robotic machining recently [1]. This can be achieved by identifying machine positions with higher stiffness (posture optimization), through optimized path planning (i.e. selecting feed directions where stiffness coefficients remain more consistent), respecting stability conditions derived from specific stability diagrams, or even incorporating online/offline control strategies, among other approaches [1]. From these researches, it can be highlighted that models of robotic machining are mainly composed of 2 parts interacting at their boundary : the robot itself with its milling tool and the workpiece.

Different models are used to model the dynamics of IRs, depending on factors such as desired complexity and accuracy. Moreover, all models are not relevant for all robots as they come from different manufacturers and have different intrinsic properties due to their components and building techniques. With

10.1016/j.procir.2025.02.021

the exception of control-oriented models, robots are mainly represented as multibody systems (MBS). The kinematics of the robot can be established by the modified Denavit Hartenberg convention [8], while its dynamics are obtained by a Lagrangian method [9] with or without flexible-body dynamics (such as flexible joint modelling) [10] or with the virtual work (or power) principle [11].

The same principle applies to the calculation of cutting forces. In this paper, a mechanistic cutting force model was used, based on a linear relationship between undeformed chip thickness and cutting forces, combined with end mill discretization, as implemented in the Dystamill software [12]. This approach, which slices the workpiece to determine the cutter-workpiece engagement (CWE) and, subsequently, chip thickness, was also adopted by Chen et al. [5]. Alternative methods for modelling the workpiece include solid modelling, volume discretization, point-based methods, and vector modelling techniques such as tri-dexels [4].

As the teams work with different robots and models depending on their needs, having the possibility of modifying a subsystem - here, the robot or the workpiece - with inconsequential impact on the other parts of the system is useful. It allows to use dedicated software - and dedicated expertise - for each part of the global model to get more appropriate results. Co-simulation, which is a numerical simulation approach in which a whole system is split into different subsystems that are modelled and simulated separately, offers this modularity [13]. These subsystems and so their solvers interact between themselves through data exchange during the simulation. However, there are drawbacks to co-simulation as it can decrease the accuracy of the results due to discrete data exchange between subsystems contrasting with monolithic models [14], which violates the principle of energy conservation and for some techniques, the action/reaction principle [15].

At this time, monolithic models, which combine robot/workpiece displacements and cutting forces calculation in a single algorithm, are the main solution found in the literature in the field of IRs. Co-simulation can however be found in other fields such as railway dynamics [15] or machining [16]. A monolithic simulation model refers to a simulation approach in which all the components of the system are integrated into a single, unified framework with a unique numerical solver. Instead of dividing the system into modular or weakly coupled sub-models (co-simulation), the entire system's behaviour is represented cohesively within one comprehensive model.

In this paper, a simplified co-simulated model is presented, combining two in-house software : Dystamill [12], for the calculation of cutting forces, and Easydyn [17], for the calculation of the robot's and piece's dynamics (Figure 1). The robot/workpiece system is divided in two multibody subsystems, one for the robot and one for the workpiece. The latter uses Dystamill to compute the cutting forces [12]. Those two subsystems exchange information at each macro time step (h). The macro time step is defined as the interval of time between two consecutive exchanges of information between the subsystems of the model. The robot part gives its position ($q(t)$) and velocities ($\dot{q}(t)$) in the 3 directions of space to the workpiece subsystem and the latter gives the cutting forces (F_c) to the robot. After each discrete exchange of data, both subsystems perform their numerical integration on their own until the next exchange of information. The aim of this paper is to determine a benchmark to test the influence of co-simulation on robot/tool/workpiece coupling. It is a proof of concept for co-simulation validation, the dynamics of the model are simplified at the extreme.

Chen et al. created a coupling algorithm between a MBS and a cutting forces algorithm [5]. The approach presented in this article differs from the method developed by the other team, as it integrates both subsystems iteratively, performing one after the other, until the difference between successive iterations becomes sufficiently small.

In section 2, the multibody model of the system is described and the cutting forces model is overviewed. In the 3rd section, the simulations parameters are given. The results and the comparison between both models are developed in the 4th section. Then, the article ends with a conclusion and perspective of future works.

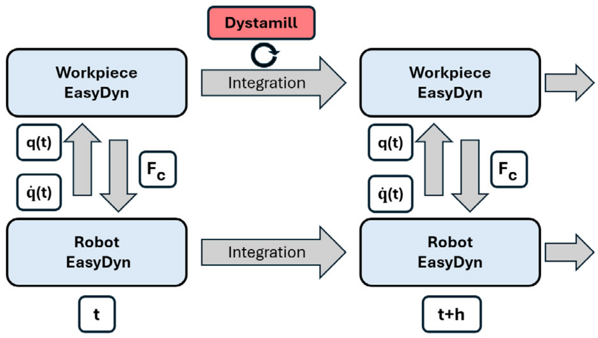

Fig. 1. Co-simulation methodology for one macro time step (h)

2. Model

A simplified model of the robot/workpiece is used for the comparison between the monolithic model and the co-simulated one (Figure 2 - 3D system with a 2D figure for simplicity). To help the reader visualize the system, it is represented at figure 3. The system used for simplification is a bi-mass system. The springs are noted k and the dampers, d. The IR is represented by a mass m_1 giving the vibration and deflection of the robot attached by a spring and a damper to a fictional ground, the robot itself, to which the user can give "perfect" moving instructions. At this stage, the robot dynamic elements (k_1 and d_1) are considered constant in space for simplification purpose, even if it is known that they vary in the workspace. For the small displacements of the robot in the simulations, this hypothesis of linearity does not deteriorate the results significantly. The workpiece (noted pc) is represented by the same system, but its ground is fixed. Its mass, spring and damper are noted respectively m_2, k_2 and d_2. Between the robot and the workpiece, we find the tool, represented by a spring k_c and a damper d_c, which are the co-simulation variables. A half of the tool mass is given to each subsystem. As found in the literature, the cut between the two subsystems in the co-simulated model is done at the tool level [16]. Choosing the interface between the tool and the robot allows the use of software tailored for either machining operations or simulation of the dynamics of the robots. This system features 9 dofs, all displacements. Three of them are used for the imposed x, y and z displacements of the robot ($q_{4,5,6}$) and 3 are given to each vibrating mass ($q_{1,2,3}$ for m_1 and $q_{7,8,9}$ for m_2). As Dystamill is a 2.5D software (working with layers of 2D simulations), simulation of rotations of the robot is not needed. The monolithic model is similar as the co-simulated one, the difference lies in the fact that there is no cut, therefore, no sub-models. The whole model is in a single code. Hence, since the monolithic model encounters less numerical error, it is taken as the reference to compare the co-simulated model with.

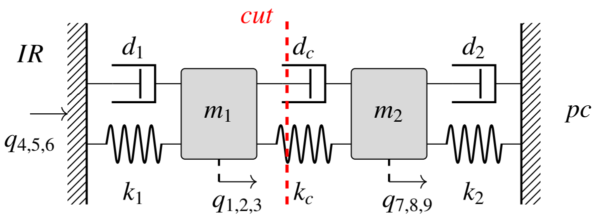

Fig. 2. Double mass-spring-damper system

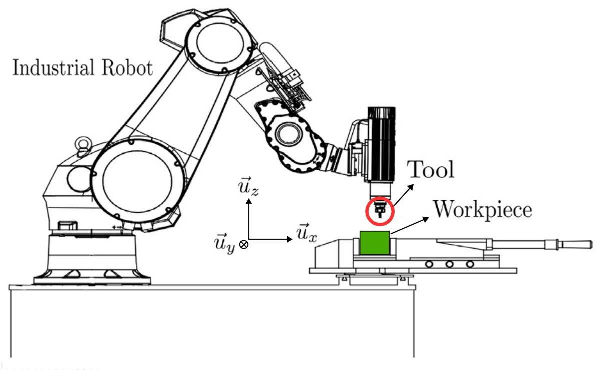

Fig. 3. Scheme of the robot/workpiece system

As explained in the previous section, for each macro time step, during the integration of the dynamics of the workpiece subsystem, the calculation of the cutting forces is processed by Dystamill. To compute the milling forces, the software first detects if there is contact between the tool and workpiece, then the quantity of matter cut by the tool. As the software works with layers for the tool (the planes of the layers are perpendicular to the axis of the tool) and the workpiece, this is done for each layer. Knowing the area of matter cut, elementary cutting forces for each layer can be estimated using Altintas' model. The global cutting forces are the sum of all the elementary cutting forces. The value of the cutting forces is then sent to the robot subsystem at the end of the macro time step. More complete information is given in Huynh's et al. article on Dystamill [12].

3. Benchmark parameters

The parameters used for the simulations are given in table 1.

Table 1. Simulation parameters for the comparison between monolithic and co-simulated models - from [4]

Operation parameters	Unit	Value
Operation type	/	Side Milling
Tool Type	/	Flat-end mill
Tool Material	/	Carbide
Diameter	[mm]	10
N° of edges	/	2
Pitch/run out	[°/μm]	170-190
Helix Angle	[°]	30
Rotation Speed	[RPM]	11250
Axial - Radial Depth of Cut	[mm]	2 - 4
Feed per Tooth	[mm/tooth]	0.13
Number of slices (tool)	/	14
Macro Time Steps (h)	[s]	10^{-5}
Simulation time	[s]	5
Workpiece Material	/	Al6082
Workpiece's Length	[mm]	185
Cutting Force Coefficients $K_{t/r/a,c}$	[MPa]	733.5/346.5/127.9
$K_{robot,x/y/z}$	[N/mm]	301/1932/1652
$K_{tool,x/y/z}$	[N/mm]	3613/3613/69376
$K_{workpiece,x/y/z}$	[N/mm]	5658/5658/37259

The Cartesian stiffness coefficients of the robot's simplified model have been determined based on the joint stiffnesses of the robot model developed by Huynh [18]. Once a more comprehensive model of the robot is integrated, pose-dependent joint stiffnesses will be used. For the stiffness of the tool, it has been considered as a beam element with a force at the tip. For the stiffness in the z direction, its value might be high, nevertheless, it will be overshadowed by the robot's stiffness in the same direction. And for the piece, we have to look at local deformations as it can be considered as infinitely rigid. In fact, if it is considered as a beam, its length would be the depth of cut and its section, the area of the base of the uncut part of the piece. This would lead to values several orders of magnitude larger than every other stiffness in the model. To calculate the local stiffness, Hertz contact between a cylinder and a plane (x and y directions) and the one between a sphere and a plane (z direction) have been used. Carbide (the material of the tool) being stiffer than the aluminium alloy constituting the workpiece, it is considered as an approximation that all the deformation happens on the latter. As the stiffness is force dependent, it was chosen to take the maximum value of the force in the calculation of the cutting forces by Dystamill. At this stage, this work is a proof of concept, consequently, the values for damping factors being complicated to obtain, they have been tuned to get an admissible behaviour. The masses of the system have been taken as the masses of the bodies (1000 kg for the robot and 10 kg for the workpiece). In reality, the whole robot does not vibrate at the same amplitude, the tip of the robot vibrates more than its base. It explains why the results are not similar as the ones found by Huynh [10].

4. Results

In figure 4 the evolution of the position of the workpiece during the simulation is shown. It can be seen that co-simulation follows nearly exactly the monolithic model even in the transient states (entering and exiting the workpiece). The reader will find an illustration of the position of the robot at the beginning of the simulation and its direction of movement at figure 5. Getting null displacement values for the workpiece when there is no contact between the tool and the workpiece is needed, so, in the model, the coupling spring/damper must be deactivated during these times. Figure 6 gives the same evolutions but for the deflection of the robot. As for the workpiece, both models provide similar results. Robot deflection is not null even when there is no contact with the workpiece, this is a reaction to its acceleration. In this simulation, the robot is moved along the x axis (q_4) and its acceleration evolution is given in figure 7. This kind of evolution allows to get smoother velocities evolutions than LSPB (Linear Segments with Parabolic Blends) trajectories as it gives triangular accelerations instead of steps [4].

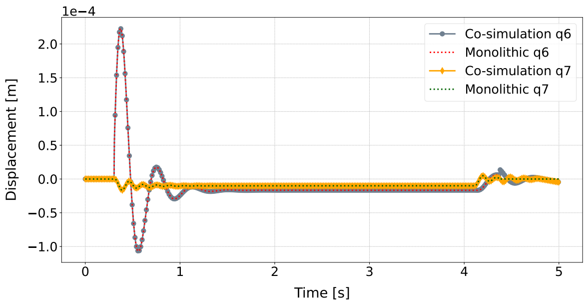

Fig. 4. Comparison of the x and y (q_6 and q_7) displacements of the workpiece for both models

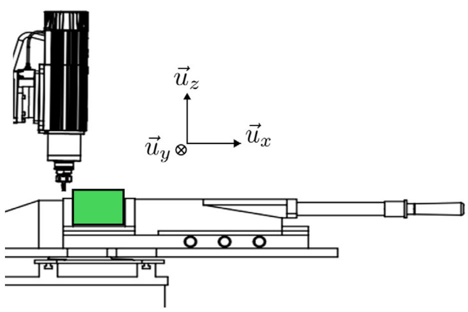

Fig. 5. Position and direction of the tool compared to the workpiece

When comparing two methods, it is essential to calculate errors. In this work, as commonly done in the literature, the monolithic model has been used as the reference [15, 19]. Accuracy indicators such as simple configuration parameter comparison error (Equation 1), displacement mean total error (Equation 2) and RMSE (Root Mean Square Error) (Equation 3) were used to assess the computation fidelity.

$$\epsilon_{sub,k} = q_k^{CS} - q_k^{Mono} \tag{1}$$

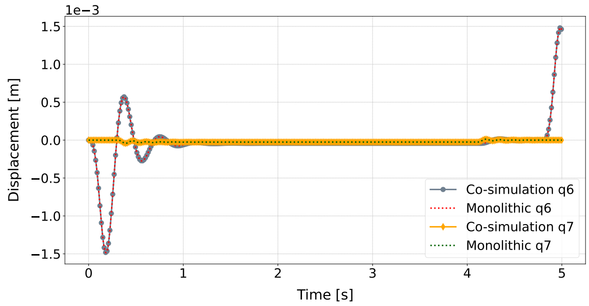

Fig. 6. Comparison of the x and y deflections of the robot for both models

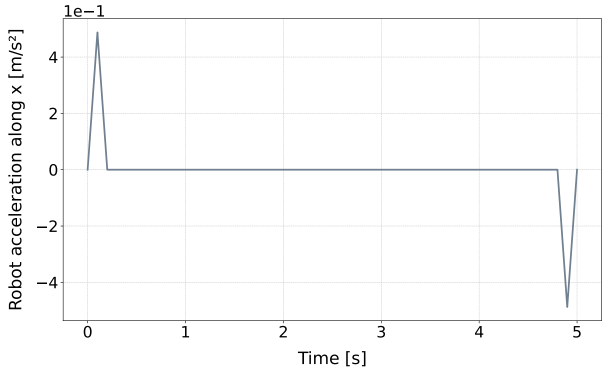

Fig. 7. Acceleration of the robot along the x axis

$$\epsilon_{disp_{MTE}} = \frac{\sum_{t_a=t0}^{tf} \sum_{k=0}^{n_{dof}} q_k^{CS}(t_a) - q_k^{Mono}(t_a)}{N_{inc}} \tag{2}$$

$$RMSE_k = \sqrt{\frac{\sum_{i=1}^{N_{inc}} (q_k^{CS} - q_k^{Mono})^2}{N_{inc}}} \tag{3}$$

With *CS* referring to Co-Simulated and *Mono* to monolithic. The other notations are given :

- k : the k^{th} degree of freedom
- t_a : Time during simulation
- t_0 : Initial simulation time
- t_f : Final simulation time
- n_{dof} : Number of degrees of freedom
- $N_{inc} = (t_f - t_0)$/h, the number of steps in the simulation

Stability indicators, such as the normalized energy divergence, focusing more on the total energy of the system also exist [15, 20]. Those indicators come from the residual energy that can be observed due to the violation of the principle of energy conservation. Nevertheless, energy dissipation in the robot due to damping is difficult to evaluate at this stage. As a result, those indicators are not used because they would lead to inaccurate estimations of errors.

Figures 8 to 11 give the simple configuration parameter comparison error for the robot and the workpiece deflection in the x and y direction. As the macro time step is small 10^{-5} s, in co-simulation, both models exchange regularly information, thus the error is contained to the 10^{-10} m order of magnitude, for all directions during the milling operation. During the transient states (entering and exiting the workpiece), as the variation of the displacement of the robot and piece is greater, the calculated errors are larger. The fact that the errors when exiting the piece are greater than when entering, especially for workpiece displacement, might be caused by the simplicity of the model, and the many approximations taken.

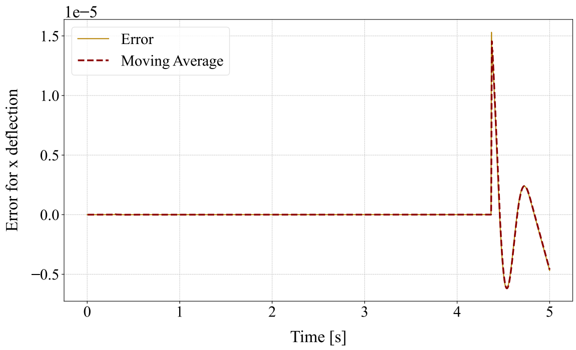

Fig. 8. Simple configuration parameter comparison error for x displacement of the workpiece

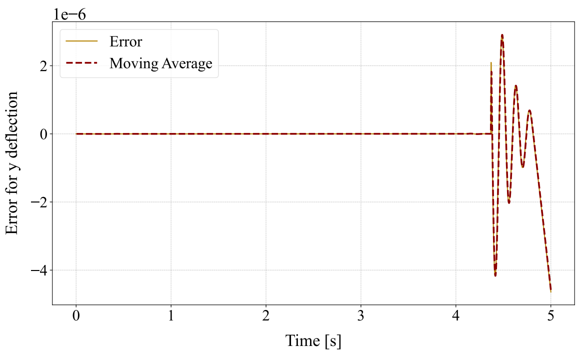

Fig. 9. Simple configuration parameter comparison error for y displacement of the workpiece

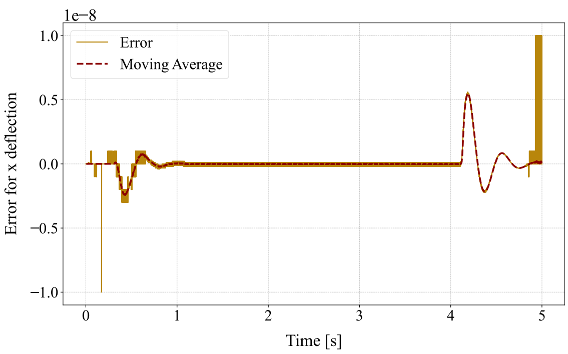

Fig. 10. Simple configuration parameter comparison error for x displacement of the robot

Even though looking at errors at sub-micron scale when deflections are usually at the tenth of mm scale might seem irrelevant, this type of error shows that, with a simplified model,

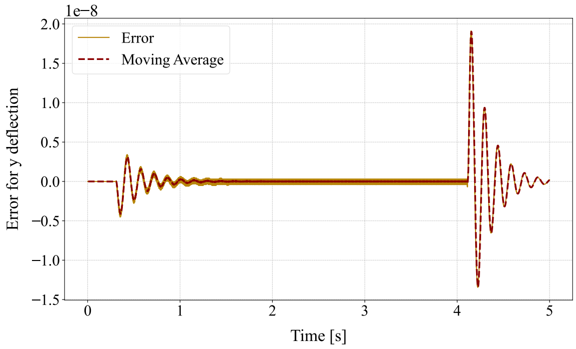

Fig. 11. Simple configuration parameter comparison error for y displacement of the robot

co-simulation can be used. It is worth mentioning that errors on velocities and especially on accelerations are higher but on the whole simulation, both models produce similar results. This could be caused by a small shift in results in time caused by co-simulation as variations in these quantities are more important than in position. It would be interesting to investigate it.

The values for the other indicators are given in table 2.

Table 2. Values for displacement mean total error and RMSE

Error type	$RMS\,E_1$	$RMS\,E_2$	$RMS\,E_7$	$RMS\,E_8$	$\epsilon_{dispMTE}$
Value [m]	$8.7*10^{-10}$	$2.3*10^{-9}$	$1.6*10^{-6}$	$7.2*10^{-7}$	$7.9*10^{-7}$

These indicators give the same conclusions as the first one and show that co-simulation can be used for simple systems. This approach is likely to be applicable to more complex systems, including those incorporating non-linearities, although doing so may result in increased errors.

At this stage, we have only discussed results for a macro step time of 10^{-5} s. Figure 12 shows how the displacement mean total error evolves for different values of h (macro time step) for co-simulation, still compared to a time step of 10^{-5} s for the monolithic simulation. As expected, when the time step is increased, as there are fewer exchanges of information between the sub systems, the error increases. Above h = 10^{-3} s, the evolution becomes erratic as the macro step time is too high to accurately model the phenomenon. For some values of h, the coupled solver cannot process the entire simulation The increase in step time also causes Dystamill to be less accurate. Less accurate cutting forces lead to worse displacement values. It is interesting to see that under a macro step time of 10^{-4} s, the error does not seem to evolve substantially. At 10^{-4} s, the simulation lasts 26.26 s for co-simulation and 13.48 s for the monolithic model, with an Intel core i9-9900 CPU.

5. Conclusion and perspectives

In conclusion, using a sufficiently small time step validates the use of co-simulation for this simplified model of machining, as the resulting errors are orders of magnitude smaller than the deflections of the subsystems. However, several approximations

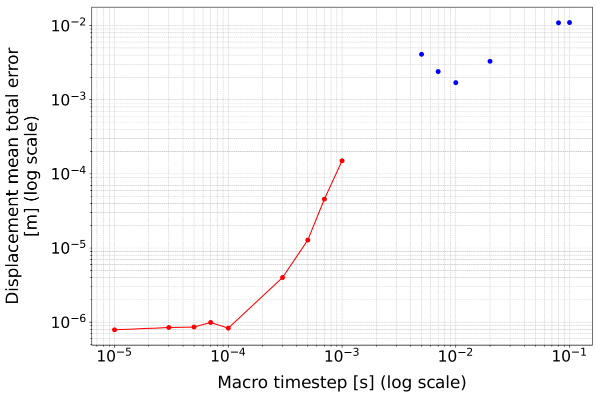

Fig. 12. Evolution of the displacement mean total error with macro time step

were necessary to use this simple system, leading to differences in deflection results compared to the ones obtained with more complex models [12]. This is not a limitation in the context of this work, which primarily aims to introduce the concept of co-simulation in machining. The main advantage of co-simulation is that, even if simulations take longer, different software (and therefore their own modelling features and solvers) can be used for the sub-sections of the model, allowing for different ways to model the robot/tool/workpiece interaction.

This study demonstrates that the tools developed can be used as a basis to prepare the way for future research. Incorporating a more complete robot model (with an increasing complexity for the modelling of flexibility (first with transmission flexibility and later with full flexible [10])) and accounting for variations in its dynamic properties across the workspace in both co-simulation and monolithic models will be essential. Another exciting perspective would be to extend this framework to include workpiece models and cutting force estimations along the z-axis as well, which is necessary for robotic machining. Ultimately, comparing these advanced simulations with experimental data will help assessing the accuracy and usefulness of co-simulation in robotic machining, especially for subsystems with greater number of degrees of freedom.

Acknowledgments

The authors would like to acknowledge the Belgian National Fund for Scientific Research (FNRS-FRS) and the UMONS Risk Institute for the grant allocated to H. Dantinne.

References

[1] Alexander Verl, Anna Valente, Shreyes Melkote, Christian Brecher, Erdem Ozturk, and Lutfi Taner Tunc. Robots in machining. *CIRP Annals*, 68(2):799–822, 2019. doi: 10.1016/j.cirp.2019.05.009.

[2] Huynh Hoai Nam. *Robotic machining: Development and validation of a numerical model of robotic milling to optimise the cutting parameters*. PhD thesis, UMONS, 2019. url: http://rgdoi.net/10.13140/RG.2.2.32268.46726.

[3] Achim Buerkle, William Eaton, Ali Al-Yacoub, Melanie Zimmer, Peter Kinnell, Michael Henshaw, Matthew Coombes, Wen-Hua Chen, and Niels Lohse. Towards industrial robots as a service (IRaaS): Flexibility, usability, safety and business models. *Robotics and Computer-Integrated Manufacturing*, 81:102484, June 2023. doi: 10.1016/j.rcim.2022.102484.

[4] Valentin Dambly. *Tool trajectories adaptation to correct the dynamic and static inaccuracies*. PhD thesis, UMONS, 2023.

[5] Kun Chen, Peng Xu, and Bing Li. Interactive coupling of structural dynamics and milling forces for high-frequency stability prediction in robotic milling. *Robotics and Computer-Integrated Manufacturing*, 86:102676, April 2024. doi: 10.1016/j.rcim.2023.102676.

[6] Hui Zhang, Jianjun Wang, G. Zhang, Zhongxue Gan, Zengxi Pan, Hongliang Cui, and Zhenqi Zhu. Machining with flexible manipulator: toward improving robotic machining performance. In *Proceedings, 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics.*, pages 1127–1132, Monterey, CA, 2005. IEEE. doi: 10.1109/AIM.2005.1511161.

[7] Zhiwu Liu, Zhaohui Deng, Lingxiao Yi, Jimin Ge, and Pengcheng Yang. A review of research on robot machining chatter. *The International Journal of Advanced Manufacturing Technology*, September 2024. doi: 10.1007/s00170-024-14533-4.

[8] Shi Guihong, Liang Zhiqiang, Du Yuchao, Yue Yi, Chen Sichen, Gao Zirui, Zheng Haoran, Liu Zhibing, Qiu Tianyang, and Wang Xibin. Stability prediction for robotic milling based on tool tip frequency response prediction by considering the interface stiffness of spindle-tool system. *Journal of Sound and Vibration*, 585:118471, September 2024. doi: 10.1016/j.jsv.2024.118471.

[9] E. Bayo and A. Avello. Singularity-free augmented Lagrangian algorithms for constrained multibody dynamics. *Nonlinear Dynamics*, 5(2):209–231, March 1994. doi: 10.1007/BF00045677.

[10] Hoai Nam Huynh, Hamed Assadi, Edouard Rivière-Lorphèvre, Olivier Verlinden, and Keivan Ahmadi. Modelling the dynamics of industrial robots for milling operations. *Robotics and Computer-Integrated Manufacturing*, 61:101852, February 2020. doi: 10.1016/j.rcim.2019.101852.

[11] Jaime Gallardo-Alvarado, Ramón Rodríguez-Castro, and Pedro J. Delossantos-Lara. Kinematics and dynamics of a 4-prur schönflies parallel manipulator by means of screw theory and the principle of virtual work. *Mechanism and Machine Theory*, 122:347–360, 2018. doi: 10.1016/j.mechmachtheory.2017.12.022.

[12] Hoai Nam Huynh, Edouard Rivière-Lorphèvre, François Ducobu, Abdullah Ozcan, and Olivier Verlinden. Dystamill: a framework dedicated to the dynamic simulation of milling operations for stability assessment. *The International Journal of Advanced Manufacturing Technology*, 98(5-8):2109–2126, September 2018. doi: 10.1007/s00170-018-2357-3.

[13] Francisco Gonzalez, Miguel Angel Naya, Alberto Luaces, and Manuel Gonzalez. On the effect of multirate co-simulation techniques in the efficiency and accuracy of multibody system dynamics. *Multibody System Dynamics*, 25(4):461–483, April 2011. doi: 10.1007/s11044-010-9234-7.

[14] M. Arnold, B. Burgermeister, C. Führer, G. Hippmann, and G. Rill. Numerical methods in vehicle system dynamics: state of the art and current developments. *Vehicle System Dynamics*, 49(7):1159–1207, July 2011. doi: 10.1080/00423114.2011.582953.

[15] Bryan Olivier. *Subsystem coupling using co-simulation methods : Application to railway dynamics*. PhD thesis, UMONS, 2020.

[16] Tony L. Schmitz, Kevin Powell, Dongki Won, G. Scott Duncan, W. Gregory Sawyer, and John C. Ziegert. Shrink fit tool holder connection stiffness/damping modeling for frequency response prediction in milling. *International Journal of Machine Tools and Manufacture*, 47(9):1368–1380, July 2007. doi: 10.1016/j.ijmachtools.2006.08.009.

[17] Olivier Verlinden, Georges Kouroussis, and Calogero Conti. Easydyn: A framework based on free symbolic and numerical tools for teaching multibody systems. *MULTIBODY DYNAMICS*, 2005.

[18] Hoai Nam Huynh, Hamed Assadi, Valentin Dambly, Edouard Rivière-Lorphèvre, and Olivier Verlinden. Direct method for updating flexible multibody systems applied to a milling robot. *Robotics and Computer-Integrated Manufacturing*, 68, 04 2021. doi: 10.1016/j.rcim.2020.102049.

[19] Lars T. Kyllingstad, Severin Sadjina, and Stian Skjong. Error estimation and step size control with minimal subsystem interfaces, 2024. doi: 10.48550/ARXIV.2406.17353.

[20] Laurane Thielemans, Bernhard H C Sputh, Roland Pastorino, and Wim Desmet. Industrial application of residual power method as co-simulation performance indicator for e-motor hybrid-physical-virtual test setup. Leuven, Belgium, 2024.