

Object-oriented Digital Twins of parallel manipulators

Paolo Campanini^a, Francesco Casella^b, Gianni Ferretti^b, Massimo Goletti^a, and Bruno Scaglioni^a

^aMUSP Lab, Strada Torre della Razza, 29122 Piacenza, Italy

^bPolitecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria, Piazza Leonardo da Vinci 32, 20133 Milano, Italy

ARTICLE HISTORY

Compiled October 1, 2020

ABSTRACT

Abstract.

KEYWORDS

Object-oriented modelling; simulation; parallel manipulators; Modelica; DAE systems; closed chains.

1. Introduction

Owing to their superior performance with respect to serial manipulators in terms of stiffness, positioning accuracy and speed, and load-to-weight ratio, parallel manipulators have recently received growing interest also in industrial applications, for example for machining tasks such as drilling and milling [1], and for PCBs assembly [2].

However, the complex kinematics of parallel manipulators, based on multiple closed loops, if on one hand simplifies the inverse kinematics, on the other hand limits the achievable workspace and considerably complicates the calculation of the direct kinematics, and above all the dynamic modeling.

The key concept applied in developing the dynamic model of a parallel manipulator consists in cutting the kinematic loops, modelling the resulting serial tree of subchains and introducing the constraint reactions.

In order to model the serial tree, two approaches can be followed: the Lagrange-Euler (LE) formulation and the Newton-Euler (NE) formulation. The Lagrange-Euler formulation is based on the computation of the kinetic and potential energy of the tree as a whole, while the Newton-Euler formulation computes the dynamics of each link of the tree separately. The NE results naturally in large number of equations and, in this respect, is considered as poorly efficient with respect to the LE formulation. The question however is debatable, in fact, the complexity of the computation of the Lagrangian largely increases with the number of bodies involved while, on the other hand, a large number of the equations involved in the NE formulation are actually assignments [3] and it is not a case that the most efficient method to compute the

dynamics of serial manipulator is based on the NE formulation [4].

The modelling approaches then essentially differ with respect to the way the kinematic constraints and the reaction forces are taken into account.

The Newton-Euler approach has been applied in [5] and in [6], where the principle of virtual powers has been considered to remove the constraint forces. The principle of virtual work has been also applied in [7] and, in [8], to a Kane's formulation of motion equations [9–11]. An efficient formulation of the dynamics of a Stewart parallel manipulator, based on the screw theory, has been recently proposed in [12], shown to be suitable for application to dynamic model-based control.

The Lagrange-D'Alembert formulation has been applied in [13] and, recently, in [14], where the Jacobian/Hessian matrices of the constraint equations are derived from the kinetic energy. With this approach the constraint forces (Lagrange multipliers) are removed from the motion equations, by projecting the motion of the system into the directions allowed by the kinematic constraints. Similarly, the Natural Orthogonal Complement (NOC) approach has been proposed in [15], where the constraint forces are eliminated by multiplying the unconstrained dynamical equations by an orthogonal complement, derived from velocity constraints. The NOC approach has been recently applied to a Newton-Euler formulation of the dynamic model of a parallel Schönflies-motion generator [16]. Lagrangian formulation and virtual work principle have been applied in [17], with the goal to derive an efficient control-oriented dynamic model.

In all the cited approaches, the parallel robot model is developed considering its dynamics as a whole, the derivation process is rather complex (the use of symbolic manipulation tools is often suggested, i.e. MAPLE in [17]), and it is generally difficult to integrate into a multi-domain model, taking into account not only the dynamics of the multibody system but also the dynamics of electromechanical or hydraulic actuation devices, elasticity and friction, which have a significant effect on the dynamics of real manipulators. In particular, it has been shown in [18] that friction compensation yields significant improvements on control performance.

The modeling approaches described in the literature are therefore difficult to apply to the creation of Digital Twins (DT), compliant with the Industry 4.0 paradigm. In fact, a DT is something more than a simulation model. Having to be as faithful a replica as possible of the physical device, it must also replicate its structure in the connection of components, often belonging, in whole or in part, to physical domains other than the mechanical domain only. In particular, the development of a DT should take place in the same way as the assembly of the components of the physical system. In this respect, the Lagrangian approach to tree structure modeling is clearly not applicable, as it is based on the calculation of the kinetic and potential energy of the entire mechanical system as a whole. Furthermore, even the manipulations of the system of equations necessary to eliminate the constraint reactions from the equations of motion are in contrast with a true modular approach. On the other hand, object-oriented modeling seems to be particularly suitable for the creation of DT, in which it is able to guarantee a true modular multi-domain approach to modeling [19].

This paper describes the development of DTs of parallel manipulators using an object-oriented approach where:

- Only components of the standard Modelica library are used (apart from the explicitly developed inverse kinematics models), connected through the graphical interface of the interpreter (OpenModelica).
- The management of closed kinematic chains is completely transparent to the user and is carried out directly by the symbolic manipulation process during the

model compilation phase.

- The constructs of the Modelica language have been used to associate the state variables of the model to the actual degrees of freedom only, i.e. actuators coordinates, and to guide the symbolic manipulation and the calculation of the initial configuration of the simulations.

The description of the mechanical dynamics is therefore distributed over the various components (links), through the Newton-Euler approach, which is often considered in the literature as inefficient, as it implies the explicit calculation of the constraint reactions (useless for control-oriented models) but, on the other hand, this calculation is still important, especially for the use of the DT in the design phase.

At first, the dynamic model of a Delta robot is illustrated and validated with respect to experimental results. Then the model of a Stewart platform is discussed and used to automatically generate the inverse dynamics, thanks to the symbolic manipulation capabilities of the environment. In other words, an algebraic function calculating the motor torques from the position, velocity and acceleration of motor coordinates is generated, suitable to be used in model-based control strategies. Similar functions have been developed in [12], just with reference to a Stewart platform, and in [17], where the MATLAB function implementing the inverse dynamics of the 3-DOF parallel manipulator presented in [20] has been developed. The cycle time of the said MATLAB function was of 0.52695 ms, therefore compatible with the servo control cycle time of commercial control hardware. **In this work, the cycle time of the MATLAB function, computing the inverse dynamics of the Stewart platform, generated by the OpenModelica environment was of ?.**

The developed models are freely available at [link GitHub](#).

The paper is organized as follows. In Section 2 the model of a Delta robot is described and validated with respect to experimental results. In Section 3 the model of a Stewart platform is described. In Section 4 the process of generating the inverse dynamics function of the Stewart platform is illustrated and its validation and cycle time are discussed. Finally, some Conclusion is given in Section 5.

2. Model of a Delta robot

The Delta robot here considered is the result of project developed in cooperation between *Logicon* and *Mitsubishi Electric Italia*. The servomotors and the relevant motion hardware and software were provided by Mitsubishi Electric Italia, while the mechanical components were purchased by third parties. Logicon was responsible for the assembly of the machine and for all other aspects related to the project, such as the design of the electrical cabinet, wiring, technical drawing and FEM analysis. Fig. 1(a) shows a picture of the platform, while Fig. 1(b) illustrates its kinematics.

The structure of the robot consists of three identical legs connecting a fixed base with a moving platform. Each leg is composed of an upper arm and a parallelogram structure; three actuated revolute joints connect the upper arms with the base and provide motion, while all other joints are spherical joints. However, the two upper spherical joints have been replaced by two universal joints, in order to avoid the long sides of the parallelogram to rotate along their axes.

The top level Modelica model is shown in Fig. 2(a), it includes the world reference (1), the global parameters record (2), the motion planner (3), the motion controllers (4), the platform (6) and the base (7) models and the aggregate model of the legs (5),

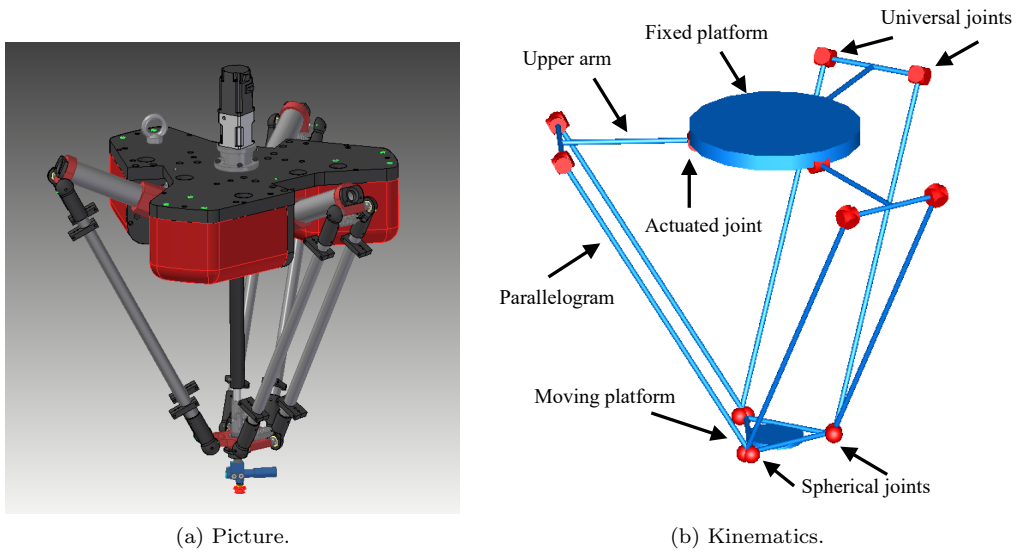


Figure 1. Delta robot.

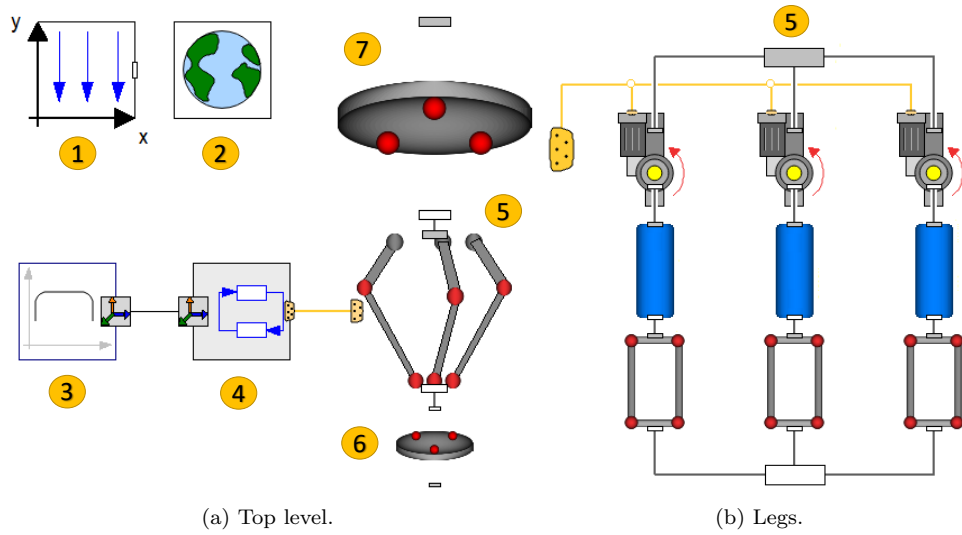


Figure 2. Delta robot: top level and legs models

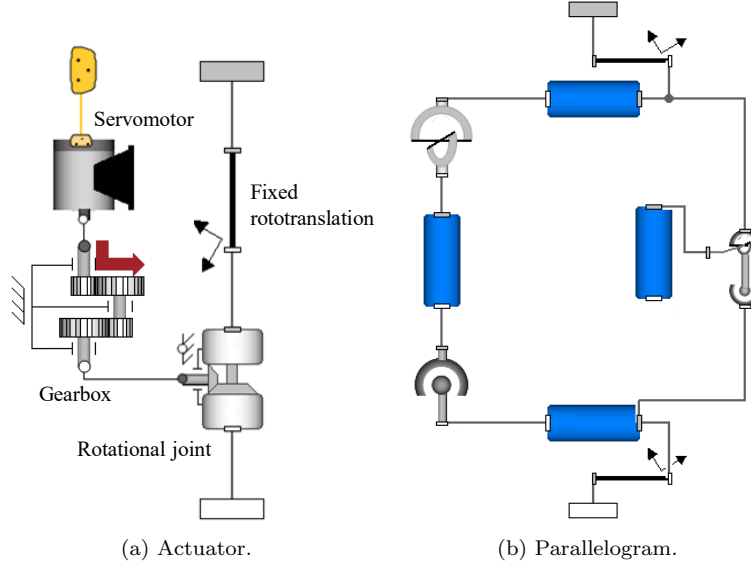


Figure 3. Delta robot: leg model

shown in Fig. 2(b).

The world reference model (1) defines the inertial reference frame and the gravity field (it must be always present when the `package MultiBody` is used [21]), while the global parameters record (2) collects all the main parameters of the model to improve readability and modifiability.

The motion planner (3) defines the trajectory of the origin of the platform reference frame; so far linear, trapezoidal and cubic trajectories can be assigned, as well as the pick-and-place trajectory considered for validation.

The motion controller model (4) first implements the inverse kinematics, thus computing the reference signals for the controllers of the motor coordinates: 3 identical classical independent PID controllers have been implemented.

It must be noted that controllers are connected to servomotors through an `expandable connector`, denoted by the yellow cable connector. This (input/output) connector models the role of the communication bus on the real machine, collecting the control signals, thus the encoder measurements from the servomotors and the current setpoints from the controllers.

The leg model is defined by the actuator model (Fig. 3(a)) connected to the upper arm (rigid body), in turn rigidly connected to the upper short side of the parallelogram (Fig. 3(b)).

The model of the electrical motor could have been easily included [22] in the servomotor model, but this would have required the adoption of very short integration step sizes, needed to follow the electrical dynamics of the windings. Since in this work the focus is on the much slower mechanical dynamics, the whole current (torque) control loop has been approximated with a first order transfer function, modelling the torque control loop bandwidth.

The gearbox model has been taken directly from the Modelica standard library [23]. In particular, the (lossy) gearbox model models the gear ratio and the losses of a standard gear box, including the stuck phases that may occur at zero speed, due to the friction in the gear teeth and/or in the bearings. The loss terms, efficiencies and friction torques, can be arbitrarily defined through lookup tables as functions of the

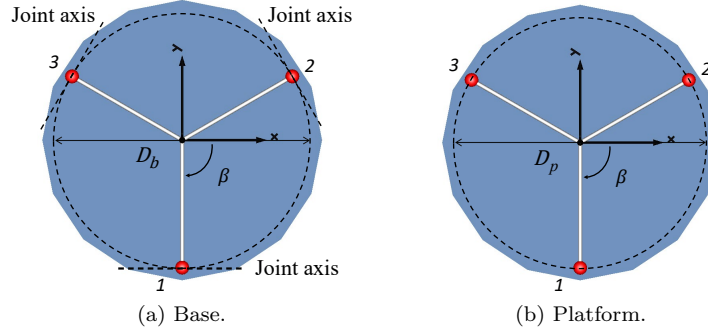


Figure 4. Deltacon base and platform.

absolute value of the input shaft speed and of the power flow direction.

The base (6) and the platform (7) can have different shapes and geometries, however, the kinematics is only defined by their attached reference frames, while dynamics is only defined by the inertial parameters of a single rigid body; in this case both base and platform have been modelled as discs (cylinders).

The upper arm rotational joints are attached to the base along a circumference of diameter D_b , displaced by 120° , with the rotation axes tangent to the circumference (Fig. 4(a)). The platform is attached to the lower sides of the parallelograms through frames placed along a circumference of diameter D_p , still displaced by 120° (Fig. 4(b)).

It must be pointed out that the upper and lower connectors of the legs model in Fig. 2(b) are actually vector connectors. In other words, all the servomotors connectors and the connectors attached to the lower sides of the parallelograms are collected in a vector one, in turn connected to another vector connector of the base and platform. The base and platform models then define the correct geometrical displacements among the legs connectors through fixed rototranslation models.

The model was built using library joints, which potentially introduce the state variables associated with the introduced degrees of freedom. As a consequence of the closure of the kinematic chains, however, the degrees of freedom are only those associated with the actuators and, properly, the state variables of the model should be associated with the actual degrees of freedom only. To implement this choice and guide the symbolic manipulation, the construct `StateSelect.always` was used for the variables of the actuators (position and speed) and `StateSelect.never` for the variables of all the other joints.

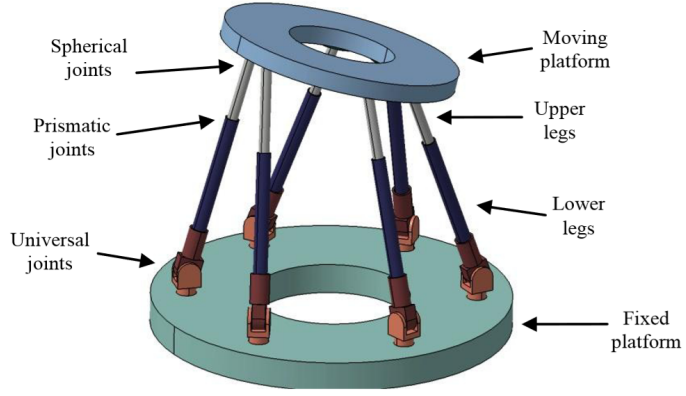
Another problem concerns the initialization of the model, in particular the need to solve closed kinematic chains. Also in this case it was decided to set the actuator variables using the `fixed = true` attribute, and then define the initial values of all the other variables as attempt values.

Validazione sperimentale.

- Userei i dati della Simulazione 1, se fosse quella relativa al confronto fra animazione e video, metterei il link al video.
- Provarei ad imporre le coppie misurate e a fare il confronto fra le posizioni e le velocità del modello e del robot
- Se non fosse un granchè metterei invece il confronto della tesi.



(a) Picture.



(b) Kinematics.

Figure 5. Stewart platform.

3. Model of a Stewart platform

The Stewart platform considered in this work has been designed for an innovative labeller for bottles of different size and dimension, Fig. 5(a) shows a picture of the platform, while Fig. 5(b) illustrates its kinematics.

It consists of a base and a platform, the former is usually fixed to the ground, while the latter can be positioned in space, in both position and orientation (6 d.o.f.), through 6 identical legs. Each leg is extendable through an electric cylinder, connected to the base through universal joints and to the moving platform through spherical joints. The electric cylinder is a mechanical linear drive unit with a piston rod, the driving component consists of an electrically actuated spindle converting the rotary motion of the motor into a linear motion of the piston rod.

The top level Modelica model is shown in Fig. 6(a) and is very similar to the Delta robot top model, while the aggregate model of the legs (5) is shown in Fig. 6(b).

In this case the motion planner (3) defines the trajectory of platform pose, in turn defined by the position of the origin of the platform reference frame and by the Euler angles relating the orientation of the platform frame with respect to the world frame; so far linear, trapezoidal and cubic trajectories can be assigned. Accordingly, 6 identical classical independent PID controllers have been implemented for the motion of the cylinders.

The leg model, whose hierarchy is shown in Fig. 7, is defined by the servomotor model connected to the cylinder model. In turn, the cylinder model is defined by the connection of two rigid bodies (stator and rod) through a prismatic joint, connected to the servomotor model through a gearbox model and a screw drive model. The Modelica standard library model of the ideal gear transforming rotational into translational motion has been slightly modified, in order to account for power losses, by simply defining two different transmission ratios for displacements and forces.

The positions of the joints on the base and platform are placed on a circumference, of diameters D_b and D_p , in couples displaced by 120° , with α_b , α_p being the angular displacements between the positions of each couple (Fig. 8). The right and left connectors of the legs model in Fig. 6(b) are again vector connectors, connected to other vector connectors on the base and platform.

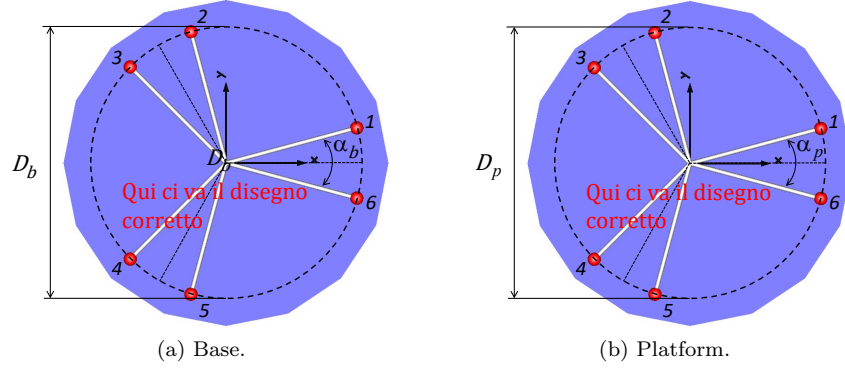


Figure 8. Stewart platform kinematics parameters.

4. Computation of the inverse dynamics for the Stewart platform

- (1) Si considera il modello controllato della piattaforma di Stewart, le cui uscite, posizioni, velocità ed accelerazioni dell'end effector saranno gli ingressi del modello inverso.
- (2) Si crea il modello inverso e si verifica che le coppie calcolate dal modello diretto e inverso siano identiche.
- (3) Si crea l'FMU del modello inverso.
- (4) Si verifica che le uscite del modello inverso OpenModelica e FMU siano identiche.
- (5) Si incapsula l'FMU in un file C per verificare i tempi di calcolo.

5. Conclusion

Conclusion.

References

- [1] J.M. Escorcia-Hernández, A. Chemori, H. Aguilar-Sierra, and J. Arturo Monroy-Anieva, *A new solution for machining with RA-PKMs: Modelling, control and experiments*, Mechanism and Machine Theory 150 (2020), pp. 1–18.
- [2] J. Hesselbach and H. Kerle, *Placing SMD parts on electronic circuit boards with parallel robots - Concepts and calculatory design*, in *Proceedings of IFAC Workshop on Intelligent Manufacturing Systems, IMS'97*, Vol. 30, July, Seoul, Korea. 1997, pp. 223–228.
- [3] H. Elmqvist and M. Otter, *Methods for tearing systems of equations in object oriented modeling*, in *Proceedings of the 1994 European Simulation Multiconference, ESM94*, Jun., Barcelona, Spain. pp. 1–7.
- [4] M.W. Walker and D.E. Orin, *Efficient dynamic computer simulation of robotic mechanisms*, Journal of Dynamic Systems, Measurement, and Control 104 (1982), pp. 205–211.
- [5] B. Dasgupta and T. Mruthyunjaya, *Closed-form dynamic equations of the general Stewart platform through the Newton–Euler approach*, Mechanism and Machine Theory 33 (1998), pp. 993–1012.
- [6] S. Briot and W. Khalil, *Dynamics of parallel robots - From rigid bodies to flexible elements*, Springer, 2015.

- [7] L.W. Tsai, *Solving the inverse dynamics of a Stewart-Gough manipulator by the principle of virtual work*, Journal of Mechanical Design 122 (2000), pp. 3–9.
- [8] J. Jiao, Y. Wu, K. Yu, and R. Zhao, *Dynamic modeling and experimental analyses of Stewart platform with flexible hinges*, Journal of Vibration and Control 25 (2019), pp. 151–171.
- [9] T.R. Kane and D.A. Levinson, *The use of Kane’s dynamical equations in robotics*, The International Journal of Robotics Research 2 (1983), pp. 3–21.
- [10] J. Yang, Z. Xu, Q. Wu, M. Zhu, S. He, and C. Qin, *Dynamic modeling and control of a 6-dof micro-vibration simulator*, Mechanism and Machine Theory 104 (2016), pp. 350–369.
- [11] J. Lih, *Computer-oriented closed-form algorithm for constrained multibody dynamics for robotics applications*, Mechanism and Machine Theory 29 (1994), pp. 357–371.
- [12] Y. Hou, G. Zhang, and D. Zeng, *An efficient method for the dynamic modeling and analysis of Stewart parallel manipulator based on the screw theory*, Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 234 (2020), pp. 808–821.
- [13] Y. Nakamura and M. Ghodoussi, *Dynamics computation of closed-link robot mechanisms with nonredundant and redundant actuators*, IEEE Transactions on Robotics and Automation 5 (1989), pp. 294–302.
- [14] R.F. Abo-Shanab, *Dynamic modeling of parallel manipulators based on Lagrange–D’Alembert formulation and Jacobian/Hessian matrices*, Multibody System Dynamics 48 (2020), pp. 403–426.
- [15] J. Angeles and S.K. Lee, *The formulation of dynamical equations of holonomic mechanical systems using a natural orthogonal complement*, Journal of Applied Mechanics 55 (1988), pp. 243–244.
- [16] P. Karimi Eskandary and J. Angeles, *The dynamics of a parallel Schönflies-motion generator*, Mechanism and Machine Theory 119 (2018), pp. 119–129.
- [17] G. Xin, H. Deng, and G. Zhong, *Closed-form dynamics of a 3-DOF spatial parallel manipulator by combining the Lagrangian formulation with the virtual work principle*, Nonlinear Dynamics (2016), pp. 1329–1347.
- [18] M. Grotjahn, B. Heimann, and H. Abdellatif, *Identification of friction and rigid-body dynamics of parallel kinematic structures for model-based control*, Multibody System Dynamics 11 (2004), pp. 273–294.
- [19] B. Scaglioni and G. Ferretti, *Towards Digital Twins through object-oriented modelling: a machine tool case study*, in *Proceedings of the 9th Vienna International Conference on Mathematical Modelling, MATHMOD2018*, F. Breitenecker, W. Kemmetmüller, A. Körner, A. Kugi, and I. Troch, eds., IFAC-PapersOnLine Vol. 51, Feb. IFAC, 2018, pp. 613–618.
- [20] T. Huang, M. Li, X.M. Zhao, J.P. Mei, D.G. Chetwynd, and S.J. Hu, *Conceptual design and dimensional synthesis for a 3-DOF module of the TriVariant-a novel 5-DOF reconfigurable hybrid robot*, IEEE Transactions on Robotics 21 (2005), pp. 449–456.
- [21] M. Otter, H. Elmqvist, and S.E. Mattsson, *The new Modelica multiBody library*, in *Proceedings of the 3rd International Modelica Conference*, P. Fritzson, ed., November, Linköping, Sweden. 2003, pp. 311–330.
- [22] G. Ferretti, G. Magnani, P. Rocco, L. Bonometti, and M. Maraglino, *Simulating permanent magnet brushless motors in DYMOLA*, in *Proceedings of the 2nd International Modelica Conference*, Oberpfaffenhofen, Germany. 2002, pp. 109–115.
- [23] C. Pelchen, C. Schweiger, and M. Otter, *Modeling and simulating the efficiency of gearboxes and of planetary gearboxes*, in *Proceedings of the 2nd International Modelica Conference*, Oberpfaffenhofen, Germany. 2002, pp. 257–266.