

# Object-Oriented Models of Parallel Manipulators

Paolo Campanini<sup>1</sup> Gianni Ferretti<sup>2</sup>

<sup>1</sup>MUSP Lab, Strada Torre della Razza, 29122 Piacenza, Italy, [paolo.campanini@musp.it](mailto:paolo.campanini@musp.it)

<sup>2</sup>Politecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria, Piazza Leonardo da Vinci 32, 20133 Milano, Italy, [gianni.ferretti@polimi.it](mailto:gianni.ferretti@polimi.it)

## Abstract

In this paper, the development of models of parallel manipulator is described, based on components of the Modelica standard library only. At first, the dynamic model of a Delta robot is illustrated and validated with respect to experimental data. Then, the model of a Stewart platform is discussed. Thanks to the symbolic manipulation capabilities of the OpenModelica compiler, the model is then used to automatically generate the inverse dynamics, which is in general is a quite difficult task.

*Keywords:* Object-oriented modelling; simulation; parallel manipulators; Modelica; DAE systems; closed chains

## 1 Introduction

Owing to their superior performance compared 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 (Escorcia-Hernández et al. 2020), and for PCBs (Printed Circuit Board) assembly (Hesselbach and Kerle 1997).

However, the complex kinematics of parallel manipulators, based on multiple closed loops, 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 sub-chains 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 compared 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 formu-

lation are actually assignments (Elmqvist and Otter 1994) and it is not a case that the most efficient method to compute the dynamics of serial manipulator is based on the NE formulation (Walker and Orin 1982).

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 (Dasgupta and Mruthyunjaya 1998) and in (Briot and Khalil 2015), where the principle of virtual powers has been considered to remove the constraint forces. The principle of virtual work has been also applied in (Tsai 2000) and, in (Jiao et al. 2019), to a Kane's formulation of motion equations (Kane and Levinson 1983; Yang et al. 2016; Lieh 1994). An efficient formulation of the dynamics of a Stewart parallel manipulator, based on the screw theory, has been recently proposed in (Hou, Zhang, and Zeng 2020), shown to be suitable for application to dynamic model-based control.

The Lagrange-D'Alembert formulation has been applied in (Nakamura and Ghodoussi 1989) and, recently, in (Abo-Shanab 2020), 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 (Angeles and Lee 1988), 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 (Karimi Eskandary and Angeles 2018). Lagrangian formulation and virtual work principle have been applied in (Xin, Deng, and Zhong 2016), 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 (Xin, Deng, and Zhong 2016)), 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 (Grotjahn, Heimann, and Abdellatif 2004) 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 (Scaglioni and Ferretti 2018).

This paper describes the development of models of parallel manipulators<sup>1</sup>, suitable for the creation of DTs, using an object-oriented approach where:

- Only components of the standard Modelica (Fritzson et al. 2020) library are used (no equations have been written, 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.

<sup>1</sup>The developed models are freely available at [https://github.com/looms-polimi/Parallel\\_manipulators](https://github.com/looms-polimi/Parallel_manipulators).

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 (Hou, Zhang, and Zeng 2020), just with reference to a Stewart platform, and in (Xin, Deng, and Zhong 2016), where the MATLAB function implementing the inverse dynamics of the 3-DOF parallel manipulator presented in (Huang et al. 2005) has been developed.

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 3.2 the process of generating the inverse dynamics function of the Stewart platform is illustrated. Finally, some Conclusion is given in Section 4.

## 2 Model of a Delta robot

The Delta robot considered here 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. Figure 1 shows a picture of the robot, while Figure 2 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.

The spherical joints add an additional degree of freedom, corresponding to the rotation of the rods around an axis passing through the centers of the spherical joints. In reality, this degree of freedom is constrained by a system of springs (Figure 3), which however only exerts internal forces and therefore has no influence on the motion. Wanting to describe the rods as rigid bodies, without introducing the modeling of the spring system, it would be necessary to replace two spherical joints with two universal joints. However, since the rods, made of carbon, are thin and light, it is possible to concentrate their mass in the middle of the rod and thus avoid the introduction of the additional degree of freedom.

The top level Modelica model is shown in Figure 4, it includes the world reference (1), the global parameters

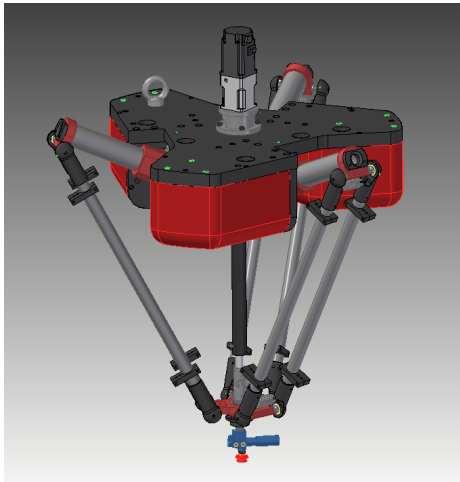


Figure 1. Picture of the Delta robot.

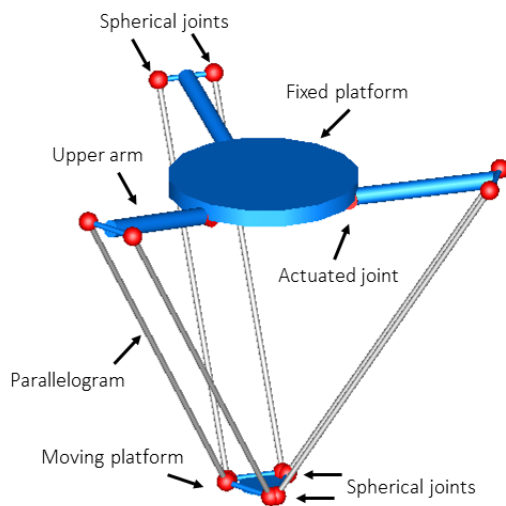


Figure 2. Kinematics of the Delta robot.

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), shown in Figure 5. 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 (Otter, Elmquist, and Mattsson 2003), 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) con-



Figure 3. Springs system

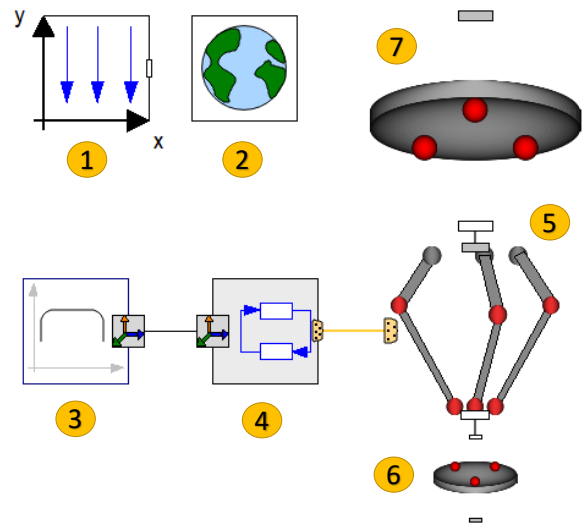


Figure 4. Delta robot model top level

nectors 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 (Figure 6(a)) connected to the upper arm (rigid body), in turn rigidly connected to the upper short side of the parallelogram (Figure 6(b)). The model of the electrical motor could have been easily included (Ferretti et al. 2002) 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 (Pelchen, Schweiger, and Otter 2002). 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 absolute value of the input shaft speed and of the power flow direction. The base

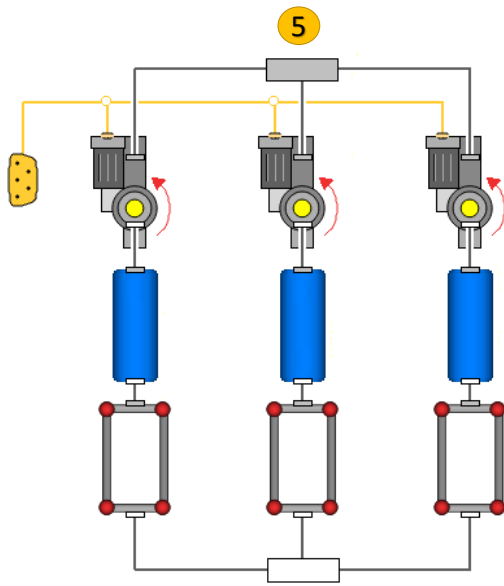


Figure 5. Delta robot legs model

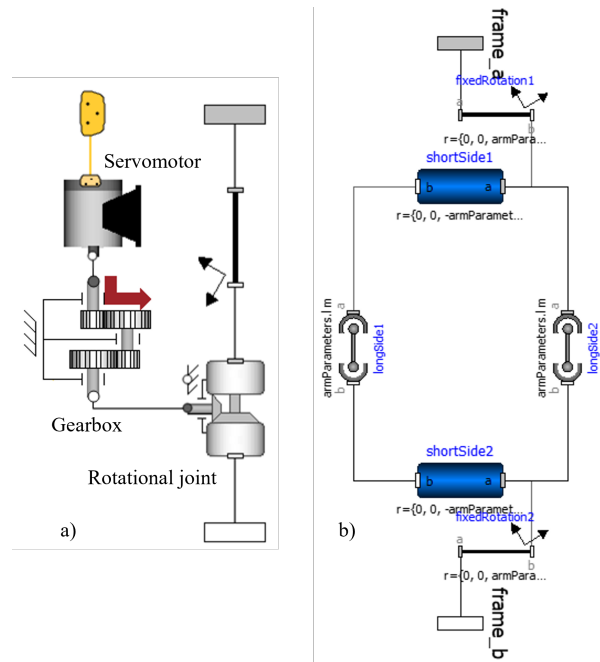


Figure 6. Delta robot leg model

(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  $120^\circ$ , with the rotation axes tangent to the circumference (Figure 7(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  $120^\circ$  (Figure 7(b)).

It must be pointed out that the upper and lower connectors of the legs model in Figure 5 are actually vectors of frames. In other words, all the servomotors frames and the frames attached to the lower sides of the parallelograms are collected in a vector, in turn connected to another vector of frames in 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 actu-

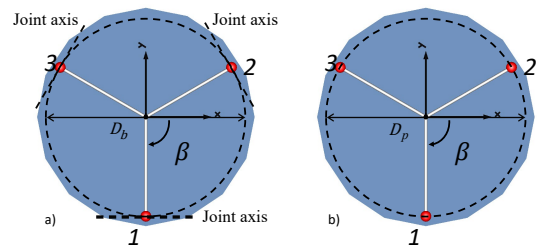


Figure 7. Delta robot base and platform

ator variables using the `fixed = true` attribute, and then define the initial values of all the other variables as attempt values.

The model has been validated with reference to a fast pick-and-place trajectory, depicted in Figure 8, where the end effector repeatedly travels along a linear trajectory of 82.4 cm in 0.44 s, therefore at an average speed of 1.87 m/s (video available). Since the structure and settings of joint controllers were not available, the validation has been performed by imposing the measured joint velocities to the model (thus computing the inverse dynamics), while comparing the measured joint torques to the simulated ones. Figure 9 shows the measured joint velocities, while Figures 10, 11 and 12 show the comparison between the measured and the simulated torque for joint 1, 2 and 3 respectively.

Although a good correspondence has been obtained between the simulation results and the measurements, some discrepancies are highlighted, in particular in Figure 10 at about time 0.3 s, in the form of a vibratory mode in the experimental data, and in Figures 11 and 12 at about 0.15 s, where the difference between simulations and measurements appears particularly evident.

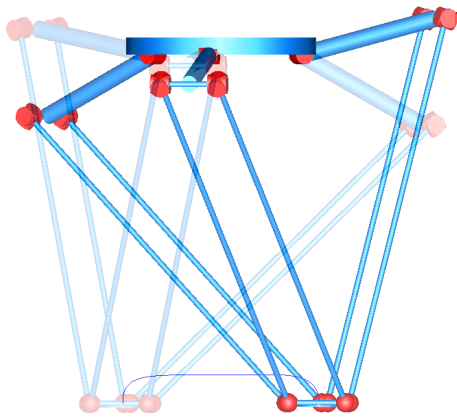


Figure 8. Pick and place trajectory

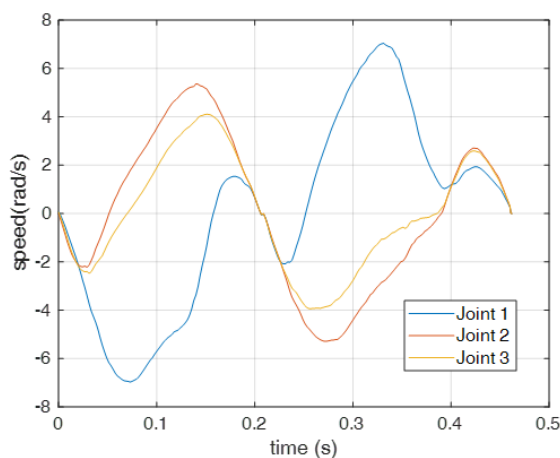


Figure 9. Measured joint velocities

The main cause of these discrepancies is probably attributable to the adoption of ideal gear and spherical joint models, not taking into account friction, backlash and elasticity. The vibratory modes detected are most likely due to the compliance of the parallelogram rods, modeled as rigid. It is also possible that at high operating speeds the same structure on which the robot is fixed may introduce vibrations.

### 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, Figure 13(a) shows a picture of the platform, while Figure 13(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

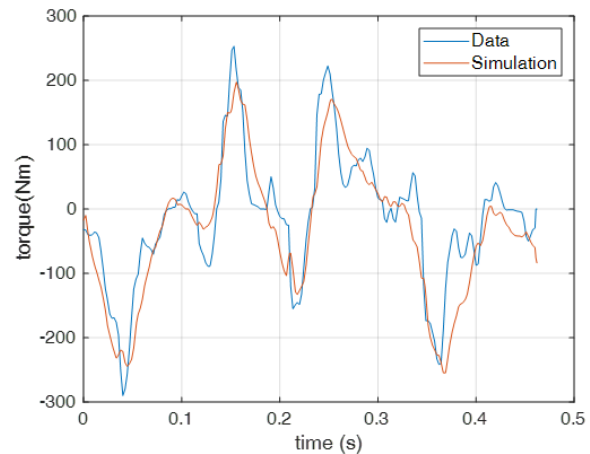


Figure 10. Meas. (blue) and simulated torque (red) on joint 1

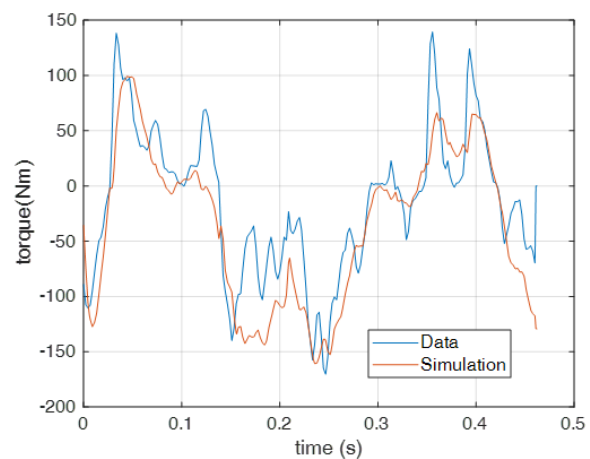


Figure 11. Meas. (blue) and simulated torque (red) on joint 2

motor into a linear motion of the piston rod.

#### 3.1 Direct dynamics

The top level Modelica model is shown in Figure 14(a) and is very similar to the Delta robot top model, while the aggregate model of the legs (5) is shown in Figure 14(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 Figure 15, 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 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  $120^\circ$ , with  $\alpha_b$ ,  $\alpha_p$  being

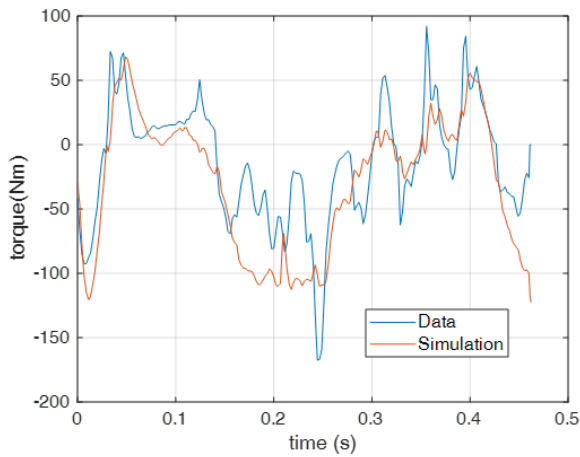


Figure 12. Meas. (blue) and simulated torque (red) on joint 3

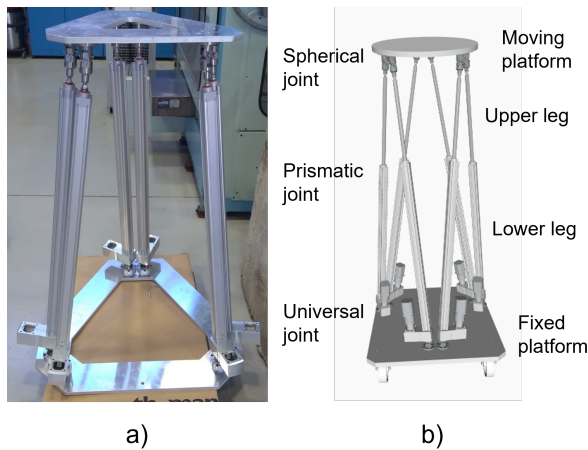


Figure 13. Stewart platform

the angular displacements between the positions of each couple (Fig. 16). The right and left connectors of the legs model in Fig. 14(b) are again vectors of frames, connected to other vectors of frames on the base and platform.

### 3.2 Inverse dynamics

The use of the model of the Stewart platform has been investigated for the automatic generation of the inverse dynamics, namely of an algebraic function computing the torques as a function of joints positions, velocities and accelerations, mainly used in control and trajectory planning (Balafoutis and Patel 1991). In control applications, inverse dynamics is usually applied to convert positions, velocities and accelerations, computed according to some desired trajectory, into the joint generalized forces which will achieve the desired motion. In trajectory planning, inverse dynamics can be used to check that the desired trajectory can be executed without exceeding the actuators' limits. Analytically computing the inverse dynamics is a difficult task, particularly in the case of closed kinematic chains but, thanks to the symbolic manipulation capabilities of the Modelica interpreter, the inverse dynamics can

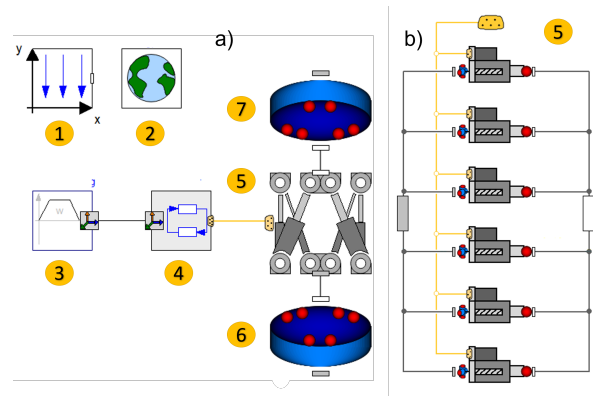


Figure 14. Stewart platform

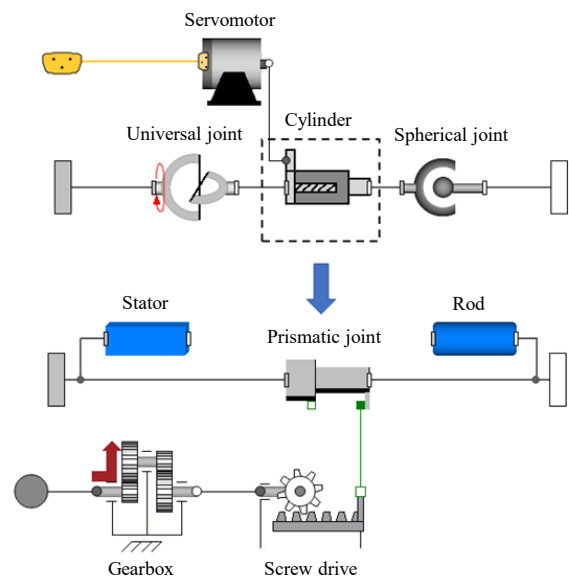


Figure 15. Stewart platform: leg model

be automatically generated from the model.

The torques computed by the inverse dynamics have been checked along the trajectory depicted in Figure 17, obtained by controlling joint positions through 6 independent PID controllers. The model of the Stewart platform has been then copied and pasted (Figure 18 and 19), while modifying the input structure. Thanks to the AngleToTorqueAdaptor model (Figure 20), instead of the joint torques, the inputs are defined by the joint positions, velocities and accelerations computed by the direct dynamics. The differences  $\delta_i$  ( $i = 1, \dots, 6$ ) between the torques computed by the direct dynamics and the torques computed by the the inverse dynamics is shown in Figure 21, as it can be seen it is equal to zero (numerically  $\pm 1.5 \times 10^{-16}$  N). A standalone model can be also generated (Figure 22).

It is important to point out that the inverse dynamic model was obtained directly, without any manipulation, from the direct dynamic model, which in turn was built in a modular way using only library models. Instead, all the approaches mentioned in the introduction required a



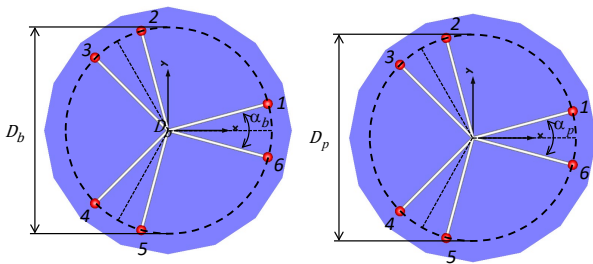


Figure 16. Stewart platform base and kinematic parameters

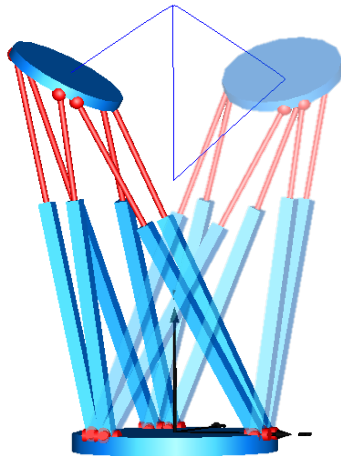


Figure 17. Stewart platform trajectory

specific analytical approach, and in some cases the use of symbolic manipulators. However, if on the one hand the efficiency in the construction of the inverse dynamic model can be considered as optimal, on the other hand the question of the computational efficiency of the generated model remains open, to be investigated in future works.

## 4 Conclusion

In this paper, the development of models of parallel manipulators based on an object-oriented modelling approach is discussed. The models have been developed based on components of the Modelica standard library only, without writing a single line of code. Two different manipulators have been considered: a Delta robot and a Stewart platform. The model of the Delta robot has been validated on the basis of experimental data, collected on a real robot. The model of the Stewart platform has been considered in order to automatically generate the inverse dynamics model, which is in general a quite difficult task to be performed manually. The main future development seems to be the verification of the numerical efficiency of the automatically generated inverse dynamics model, for the purpose of real time control and planning.

## References

Abo-Shanab, R. F. (2020). “Dynamic modeling of parallel manipulators based on Lagrange–D’Alembert formulation and

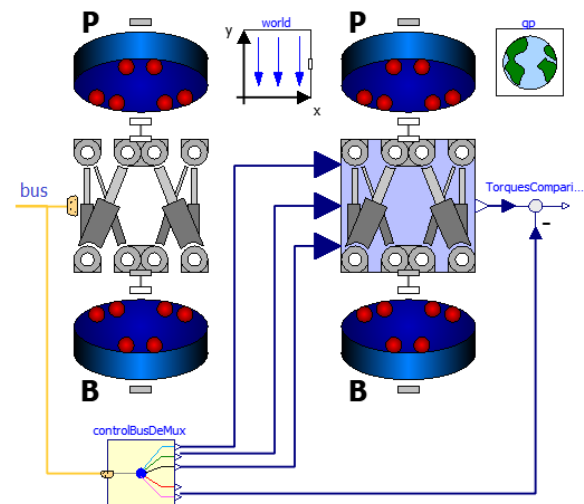


Figure 18. Stewart platform: direct and inverse dynamics



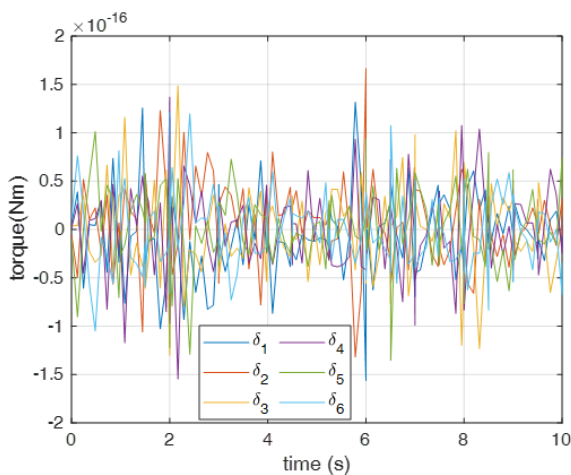
Figure 19. Stewart platforms

- Jacobian/Hessian matrices”. In: *Multibody System Dynamics* 48.4, pp. 403–426.
- Angeles, J. and S. K. Lee (1988). “The formulation of dynamical equations of holonomic mechanical systems using a natural orthogonal complement”. In: *Journal of Applied Mechanics* 55.1, pp. 243–244.
- Balafoutis, C. A. and R. V. Patel (1991). “Manipulator Inverse Dynamics”. In: *Dynamic Analysis of Robot Manipulators: A Cartesian Tensor Approach*. Boston, MA: Springer US, pp. 117–182.
- Briot, S. and W. Khalil (2015). *Dynamics of parallel robots - From rigid bodies to flexible elements*. Springer.
- Dasgupta, B. and T. S. Mruthyunjaya (1998). “Closed-form dynamic equations of the general Stewart platform through the Newton–Euler approach”. In: *Mechanism and Machine Theory* 33.7, pp. 993–1012.
- Elmqvist, H. and M. Otter (1994). “Methods for tearing systems of equations in object oriented modeling”. In: *Proceedings of the 1994 European Simulation Multiconference, ESM94*. Barcelona, Spain, pp. 1–7.
- Escorcia-Hernández, J. M. et al. (2020). “A new solution for machining with RA-PKMs: Modelling, control and experiments”. In: *Mechanism and Machine Theory* 150, pp. 1–18.

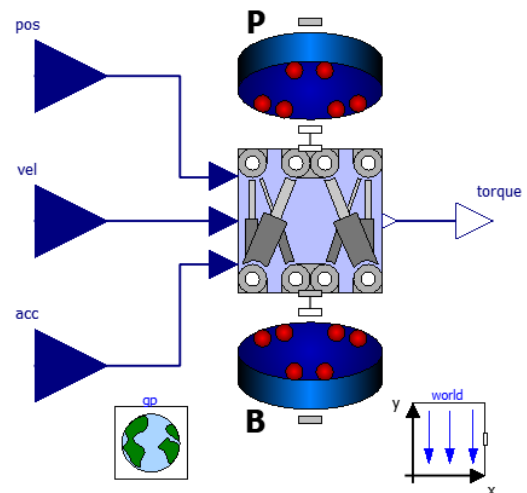
## AngleToTorqueAdaptor



**Figure 20.** AngleToTorqueAdaptor model



**Figure 21.** Difference between direct and inverse torques



**Figure 22.** Inverse dynamics model

- Ferretti, G. et al. (2002). “Simulating permanent magnet brushless motors in DYMOLA”. In: *Proceedings of the 2nd International Modelica Conference*. Oberpfaffenhofen, Germany, pp. 109–115.
- Fritzson, Peter et al. (2020). “The OpenModelica Integrated Environment for Modeling, Simulation, and Model-Based Development”. In: *Modeling, Identification and Control* 41.4, pp. 241–295.
- Grotjahn, M., B. Heimann, and H. Abdellatif (2004). “Identification of friction and rigid-body dynamics of parallel kinematic structures for model-based control”. In: *Multibody System Dynamics* 11.3, pp. 273–294.
- Hesselbach, J. and H. Kerle (1997). “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. 14. Seoul, Korea, pp. 223–228.
- Hou, Y., G. Zhang, and D. Zeng (2020). “An efficient method for the dynamic modeling and analysis of Stewart parallel manipulator based on the screw theory”. In: *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 234.3, pp. 808–821.
- Huang, T. et al. (2005). “Conceptual design and dimensional synthesis for a 3-DOF module of the TriVariant-a novel 5-DOF reconfigurable hybrid robot”. In: *IEEE Transactions on Robotics* 21.3, pp. 449–456.
- Jiao, Jian et al. (2019). “Dynamic modeling and experimental analyses of Stewart platform with flexible hinges”. In: *Journal of Vibration and Control* 25.1, pp. 151–171.
- Kane, T. R. and D A. Levinson (1983). “The use of Kane’s dynamical equations in robotics”. In: *The International Journal of Robotics Research* 2.3, pp. 3–21.

- Karimi Eskandary, P. and J. Angeles (2018). “The dynamics of a parallel Schönflies-motion generator”. In: *Mechanism and Machine Theory* 119, pp. 119–129.
- Lieh, J. (1994). “Computer-oriented closed-form algorithm for constrained multibody dynamics for robotics applications”. In: *Mechanism and Machine Theory* 29.3, pp. 357–371.
- Nakamura, Y. and M. Ghodoussi (1989). “Dynamics computation of closed-link robot mechanisms with nonredundant and redundant actuators”. In: *IEEE Transactions on Robotics and Automation* 5.3, pp. 294–302.
- Otter, M., H. Elmqvist, and S. E. Mattsson (2003). “The new Modelica multiBody library”. In: *Proceedings of the 3rd International Modelica Conference*. Linköping, Sweden, pp. 311–330.
- Pelchen, C., C. Schweiger, and M. Otter (2002). “Modeling and simulating the efficiency of gearboxes and of planetary gearboxes”. In: *Proceedings of the 2nd International Modelica Conference*. Oberpfaffenhofen, Germany, pp. 257–266.
- Scaglioni, B. and G. Ferretti (2018). “Towards Digital Twins through object-oriented modelling: a machine tool case study”. In: *Proceedings of the 9th Vienna International Conference on Mathematical Modelling, MATHMOD2018*, pp. 613–618.
- Tsai, L. (2000). “Solving the inverse dynamics of a Stewart-Gough manipulator by the principle of virtual work”. In: *Journal of Mechanical Design* 122.1, pp. 3–9.
- Walker, M. W. and D. E. Orin (1982). “Efficient dynamic computer simulation of robotic mechanisms”. In: *Journal of Dynamic Systems, Measurement, and Control* 104.3, pp. 205–211.
- Xin, G., H. Deng, and G. Zhong (2016). “Closed-form dynamics of a 3-DOF spatial parallel manipulator by combining the Lagrangian formulation with the virtual work principle”. In: *Nonlinear Dynamics* 86, pp. 1329–1347.
- Yang, J. et al. (2016). “Dynamic modeling and control of a 6-DOF micro-vibration simulator”. In: *Mechanism and Machine Theory* 104, pp. 350–369.