A Gazebo Simulator for Continuum Parallel Robots

Andrea Gotelli[1,2], Federico Zaccaria[1,2,3], Olivier Kermorgant[1,2], and Sébastien Briot[1,4]

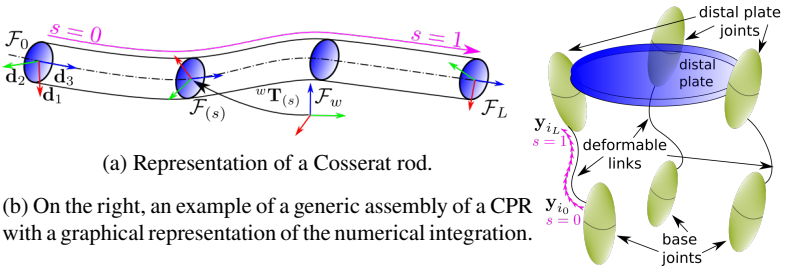
Abstract Continuum Parallel Robots (CPRs) combine properties of continuum and parallel rigid-link robots. They inherit simplicity, compliance, and cost-effectiveness from the first as payload capacity and stiffness from the latter. In this paper, we propose to use Gazebo and ROS to provide a generalized simulator for CPRs, in terms of their joints and geometry, while we use the Cosserat rod theory to model their deformable bodies. We exploit our simulator to solve the direct and inverse geometrico-static models of CPRs and to provide a useful base for simulations.

1 Introduction

Continuum Parallel Robots (CPRs) are manipulators with flexible elastic links, arranged in parallel. Like rigid links parallel robots, they have a good payload capacity and stiffness, but they also show the simplicity, compliance, and cost-effectiveness of continuum robots. Early research on these robots started with [7] and, nowadays, CPRs have a wide range of applications [8] [3].

In this work, we consider robots composed by a single distal plate, a single base and n deformable links, or limbs, ($\{n \in \mathbb{N},\ n > 1\}$), as in [8]. A generic representation of such a robot is given in Figure 1b. In what follows, we consider three possible types of actuated base joints: prismatic, revolute and an actuation that changes the length of the rod, like the one in [8], namely extensible limb. With this layout, the structure of these robots is simple, inexpensive, lightweight, compliant, and easily scalable, making it possible to realize CPRs of a few millimeters [3].

[1]Laboratoire des Sciences du Numérique de Nantes (LS2N), Nantes, France, e-mail: {Andrea.Gotelli,Sebastien.Briot,Olivier.Kermorgant}@ls2n.fr ·
[2]École Centrale de Nantes, France ·
[3]DIN, University of Bologna, Bologna, Italy e-mail: federico.zaccaria3@unibo.it ·
[4]Centre National de la Recherche Scientifique (CNRS)



(a) Representation of a Cosserat rod.

(b) On the right, an example of a generic assembly of a CPR with a graphical representation of the numerical integration.

Fig. 1: Representation of Cosserat rod (on the left) and of a generic CPR (on the right).

The combination of rigid and deformable bodies complicates the identification of the CPRs configuration. It depends on the robot geometry and the efforts among its bodies; this problem is called geometrico-static. In general, as it does not admit an analytical solution, simulations can be used to find a solution numerically.

However, there are no generalized simulators available in the literature. For every case study, researchers develop their own code [8] [3]. These specific simulations can be severely optimized to achieve real-time computation as in [8], but they are not prone to modification of the robot geometry or assembly. Any modification requires modifying the code or, in the worst case, refactoring the whole simulator.

In this paper, we propose a generic simulation for CPRs, based on Gazebo and able to identify the robot assembly automatically. We choose to use Gazebo and ROS as nowadays they are standard in many robotics applications. In particular, Gazebo allows integrating new capabilities, in terms of custom physics and rendering [6].

The paper is structured as follows: in Section 2 we discuss modeling for the deformable links. In Section 3, we present the generalized modeling strategy for CPRs. In Section 4, we briefly introduce the architecture of the simulator. In Section 5, we present some case studies to show the capabilities of our simulator. Finally we discuss our conclusion in Section 6.

2 Cosserat rod theory

The Cosserat rod theory is widely used and discussed in the literature to model the geometrico-static properties of continuum robots [7], as it has been proven to be reliable and precise [2]. A rod can be represented as a sequence of cross-sections stacked on top of each other along the rod centerline, as shown in Figure 1a. This line is parameterized by $s \in [0\,1]$ being a normalized curvilinear abscissa, defined by $s = p/\ell$, with $p \in [0\,\ell]$ the non-normalized curvilinear abscissa along the beam, and ℓ the beam length at rest. Following the assumptions discussed in [8], we attach a frame $\mathcal{F}_{(s)}$ to every rod cross-section, defined with the principal axis of inertia of the cross-section at s, namely $\mathbf{d}_1$, $\mathbf{d}_2$ and $\mathbf{d}_3 = \mathbf{d}_1 \times \mathbf{d}_2$. The frame configuration is

defined by an homogeneous transformation ${}^{w}\mathbf{T}_{(s)} \in SE(3)$ that describes its pose in terms of the position of its origin ${}^{w}\mathbf{p}_{(s)} \in \mathbb{R}^3$ and orientation ${}^{w}\mathbf{R}_{(s)} \in SO(3)$ of its frame axes with respect to a reference frame $\mathcal{F}_w$:

$$ {}^{w}\mathbf{T}_{(s)} = \langle {}^{w}\mathbf{p}_{(s)}, {}^{w}\mathbf{R}_{(s)} \rangle = \begin{bmatrix} {}^{w}\mathbf{R}_{(s)} & {}^{w}\mathbf{p}_{(s)} \\ \mathbf{0}_{[1\times3]} & 1 \end{bmatrix} \tag{1} $$

In this paper, we define a derivative with respect to the arc length $\frac{d}{ds}$ with the symbol $(\cdot)'$. In order to establish the evolution of the continuum body with respect to the arc length, we compute the derivative of position and orientation respectively as ${}^{w}\mathbf{p}'_{(s)}$ and ${}^{w}\mathbf{R}'_{(s)}$. Their counterparts, in the local coordinate of the cross-section, are: ${}^{s}\mathbf{v}_{(s)} = {}^{w}\mathbf{R}_{(s)}{}^{T}\,{}^{w}\mathbf{p}'_{(s)}$ and ${}^{s}\mathbf{u}_{(s)} = \left[{}^{w}\mathbf{R}_{(s)}{}^{T}\,{}^{w}\mathbf{R}'_{(s)}\right]^{\vee}$, where we used the map $(\cdot)^{\vee} : \mathbb{R}^{3\times3} \rightarrow \mathbb{R}^3$ such that: $\mathbf{a} \in \mathbb{R}^3, (\hat{\mathbf{a}})^{\vee} = \mathbf{a}$, $\hat{\mathbf{a}}$ being the cross-product matrix associated with the vector $\mathbf{a}$. In the following, we drop the prefix ${}^{w}(\cdot)$ for the quantities expressed with respect to $\mathcal{F}_w$.

In our work, we implement the system of ordinary differential equations (ODEs) for a Cosserat rod [3] for every i^{th} limb of a CPR, with $i = 1, \ldots, n$.

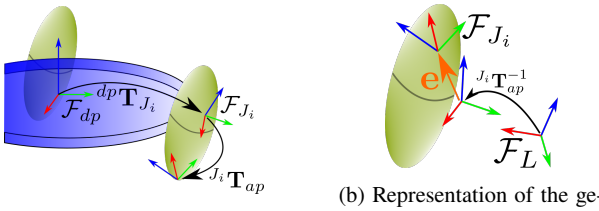
$$ \begin{cases} \mathbf{p}_{i}{}'_{(s)} = \ell_i \left(\mathbf{R}_{i_{(s)}}\, {}^{s}\mathbf{v}_{i_{(s)}} \right) \\ \mathbf{R}_{i}{}'_{(s)} = \ell_i \left(\mathbf{R}_{i_{(s)}}\, {}^{s}\mathbf{u}_{i_{(s)}} \right) \\ \mathbf{n}_{i}{}'_{(s)} = -\ell_i \bar{\mathbf{f}}_{i_{(s)}} \\ \mathbf{m}_{i}{}'_{(s)} = -\ell_i \left(\mathbf{p}_{i}{}'_{(s)} \times \mathbf{n}_{i_{(s)}} + \bar{\mathbf{l}}_{i_{(s)}} \right) \end{cases} \tag{2} $$

$$ \mathbf{n}_{i_{(s)}} = \mathbf{R}_{i_{(s)}} \mathbf{K}_{SE_i} \left[{}^{s}\mathbf{v}_{i_{(s)}} - {}^{s}\mathbf{v}^{0}_{i_{(s)}} \right] \tag{3} $$

$$ \mathbf{m}_{i_{(s)}} = \mathbf{R}_{i_{(s)}} \mathbf{K}_{BT_i} \left[{}^{s}\mathbf{u}_{i_{(s)}} - {}^{s}\mathbf{u}^{0}_{i_{(s)}} \right] \tag{4} $$

Where $\bar{\mathbf{f}}_{i_{(s)}}$ and $\bar{\mathbf{l}}_{i_{(s)}}$ are respectively some external distributed forces and moments along the centerline of the i^{th} limb, while $\mathbf{n}_{i\,(s)}$ and $\mathbf{m}_{i\,(s)}$ are respectively the internal forces and moments that act from a cross-section s to the next one $s + ds$ having ds as an infinitesimal increment of s. The ODEs (2) are paired with Equations (3) and (4) linking the deformation with the internal stresses of the rod (Hooke's law). The deformation is a difference between the current rod curvatures and the ones in a load free state: ${}^{s}\mathbf{u}^{0}_{(s)}$ and ${}^{s}\mathbf{v}^{0}_{(s)}$. The two diagonal matrices $\mathbf{K}_{SE} = diag\,(GA, GA, EA)$ and $\mathbf{K}_{BT} = diag\,\left(EI_{xx}, EI_{yy}, E\left(I_{xx} + I_{yy}\right)\right)$ account for geometrical and physical properties of the rod: E and G are the rod Young and shear modules, A is the cross-section area while I_{xx} and I_{yy} the principal moment of inertia.

We define the state of a rod as $\mathbf{y}_{i_{(s)}} = \left(\mathbf{p}_{i_{(s)}}, \mathbf{R}_{i_{(s)}}, \mathbf{n}_{i_{(s)}}, \mathbf{m}_{i_{(s)}}\right)$, where the operator $(\cdot, \ldots, \cdot)$ is a vertical concatenation of the vector listed between parentheses, in the case of a matrix, it vertically concatenates each of its columns. For every i^{th} limb, starting from the state at the rod base: $\mathbf{y}_{i_{(s=0)}} = \mathbf{y}_{i_0}$, the numerical integration of (2) gives the state at the rod tip $\mathbf{y}_{(s=1)} = \mathbf{y}_{L}$.This final state contains the pose of the last cross section $\mathbf{T}_{i_{(s=1)}} = \mathbf{T}_{i_L}$, and the wrench of the internal stresses $\mathbf{W}_{i_{(s=1)}} = \mathbf{W}_{i_L}$.



(a) Distal plate representation.

(b) Representation of the geometrical constrains.

Fig. 2: Representation of the distal plate joint. On the left its definition, on the right considerations about the geometrical constrains.

3 Generic Modeling of CPRs

We assume that base joints are responsible for the motion of the deformable links and that they define the initial state of the rod: $\mathbf{y}_{i_0}$, or the rod length ℓ_i. At the distal plate, every rod tip state $\mathbf{y}_{iL}$ must comply with the assembly constraints coming from the coupling between the tip of the rod and the joint. Moreover, the static balance of the distal plate must be satisfied. We detail these computations in what follows.

3.1 Assembly Constrains

The distal plate joints are attached to the platform body, each one with its own frame $\mathcal{F}_{J_i}$, as represented in Figure 2a. These frames are described with a rigid body transformation $^{dp}\mathbf{T}_{J_i}$ with respect to the distal plate frame $\mathcal{F}_{dp}$. For every joint, $^{J_i}\mathbf{T}_{ap}$ describes how the rod tip should connect with the joint body. If the rod tip does not reach the attach point, we define an error as shown in Figure 2b. The projection of $^{J_i}\mathbf{T}_{ap}^{-1}$ through $\mathbf{T}_L$ gives a pseudo joint origin $\mathcal{F}_{\bar{J}_i} = \langle \mathbf{p}_{\bar{J}_i}, \mathbf{R}_{\bar{J}_i} \rangle$. We then define: $\mathbf{e}_i = \left(\mathbf{e}_{i\,pos}, \mathbf{e}_{i\,ori}\right)$, $\mathbf{e}_i \in \mathbb{R}^6$; the geometrical error in $SE(3)$ between the rod tip and the joint, with $\mathbf{e}_{i\,pos} = \mathbf{p}_{\bar{J}_i} - \mathbf{p}_{J_i}$ and $\mathbf{e}_{i\,ori} = \left(\mathbf{R}_{J_i}^T \mathbf{R}_{\bar{J}_i} - \mathbf{R}_{J_i} \mathbf{R}_{\bar{J}_i}{}^T\right)^\vee$. These formulations are detailed in [8], where all joints are assumed ideal and bodyless. Note that $\mathbf{e}_i$ can be computed in different ways. However, we believe that our formulation is more computationally efficient and allows to account for the joint geometry and DoFs generically. If the joint has some DoFs, the corresponding part of $\mathbf{e}_i$ is neglected, as there are no constraints on that motion. We introduce the vector $^{J_i}\mathbf{a} = (a_x, a_y, a_z) \in \mathbb{R}^3$, for which each component can have two possible values: 1 if the joint is free along the corresponding axis and 0 otherwise. It defines a matrix $\tilde{I} = I - diag\left(^{J_i}\mathbf{a}\right)$, with I the identity matrix, that we use to compose a block-diagonal matrix $^{J_i}\mathcal{I}_{dof} \in \mathbb{R}^{6\times6}$. This last matrix cancels the component of $\mathbf{e}_i$ along the joint DoFs. As an example, we detail the calculation for a spherical joint. As it can rotate in all directions, we have to consider only the translation part of the error. It follows that $^{J_i}\mathbf{a} = (1, 1, 1)$ and $\tilde{I} = \mathbf{0}_{3\times3}$, and the error is computed as follows:

$$\mathbf{e}_{p_i}^{jnt} = \mathcal{R}_{J_i}\, {}^{J_i}\mathcal{I}_{dof}\mathcal{R}_{J_i}^T\mathbf{e}_i \tag{5}$$

$$\begin{aligned}\mathbf{e}_{p_i}^{jnt} &= \mathcal{R}_{J_i}\begin{bmatrix}\mathbb{1}_{3\times3} & \mathbf{0}_{3\times3}\\ \mathbf{0}_{3\times3} & \tilde{I}\end{bmatrix}\mathcal{R}_{J_i}^T\begin{bmatrix}\mathbf{e}_{i\,pos}\\ \mathbf{e}_{i\,ori}\end{bmatrix}\\ &= \mathcal{R}_{J_i}\begin{bmatrix}\mathbb{1}_{3\times3} & \mathbf{0}_{3\times3}\\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3}\end{bmatrix}\begin{bmatrix}{}^{J_i}\mathbf{e}_{i\,pos}\\ {}^{J_i}\mathbf{e}_{i\,ori}\end{bmatrix}\\ &= \mathcal{R}_{J_i}\begin{bmatrix}{}^{J_i}\mathbf{e}_{i\,pos}\\ \mathbf{0}_{3\times1}\end{bmatrix} = \begin{bmatrix}\mathbf{e}_{i\,pos}\\ \mathbf{0}_{3\times1}\end{bmatrix}\end{aligned} \tag{6}$$

Table 1: Definition of $\mathcal{I}_{dof}$ based on joint type

type	axis	$\mathcal{I}_{dof}$	${}^{J_i}\mathbf{a}^T$
revolute	x	$diag\left(I, \tilde{I}\right)$	$(1,0,0)$
	y	$diag\left(I, \tilde{I}\right)$	$(0,1,0)$
	z	$diag\left(I, \tilde{I}\right)$	$(0,0,1)$
prismatic	x	$diag\left(\tilde{I}, I\right)$	$(1,0,0)$
	y	$diag\left(\tilde{I}, I\right)$	$(0,1,0)$
	z	$diag\left(\tilde{I}, I\right)$	$(0,0,1)$
fixed		$diag\left(I, I\right)$	$(0,0,0)$
spherical		$diag\left(I, \tilde{I}\right)$	$(1,1,1)$

Where $\mathcal{R}_{J_i} = diag\left(\mathbf{R}_{J_i}, \mathbf{R}_{J_i}\right)$ is a block-diagonal matrix to project an error in $SE(3)$ from the joint frame to the world frame. In Equation (5), we provide the the general computation of the actual error while, in Equation (6), there is the formula applied to a spherical joint. For other joints, the Table 1 refers for the various cases.

Dual considerations apply for the static equilibrium of the distal plate. The wrench from the rod tip transfers on the joint origin: ${}^{J_i}\mathbf{W}_{J_i} = -\mathbf{Ad}_{\mathbf{T}_{rJ_i}}\mathbf{W}_L$, where $\mathbf{Ad}_{\mathbf{T}_{rJ_i}}$ is the adjoint transformation from the rod tip frame and the joint frame defined in [5]. It can be dived in two parts: ${}^{j}\mathbf{W}_{J_i}^r = {}^{J_i}\mathcal{I}_{dof}\,{}^{J_i}\mathbf{W}_{J_i}$ that is reciprocal to the joint twist and ${}^{J_i}\mathbf{W}_{J_i}^p = \left[I - {}^{J_i}\mathcal{I}_{dof}\right]{}^{J_i}\mathbf{W}_{J_i}$ that develops a non zero power.

The contribution of the i^{th} limb on the distal plate balance is obtained again with the shifting law: $\mathbf{W}_{i\,dp} = \mathcal{R}_{dp}\ \mathbf{Ad}_{\mathbf{T}_{J_i dp}}\,{}^{J_i}\mathbf{W}_{J_i}^r$.

3.2 Unique Solver

In order to find the configuration of a CPR, the geometrico-static problem is divided into two sub problems: the direct and inverse geometrico-static problem (DGM and IGM, respectively). The DGM consists in finding the pose of the distal plate knowing the wrench applied on it and the base joint actuations. In the IGM, knowing the distal plate pose and wrench, the joints coordinates are found.

In the literature, the CPRs geometrico-static problem is typically solved with the shooting method as in [8]. The shooting method consists in manipulating two vectors: the guess and the residual vector. The former, denoted as $\mathbf{x}_0$, is a vector containing all the variables in the geometrico-static problem while the second, denoted as $\mathbf{r}$ represents a cost function which module must be null. They are defined as:

$$\mathbf{x}_0 = \left(\mathbf{W}_{1\,(0)}, \ldots, \mathbf{W}_{n\,(0)}, \ldots, \mathbf{q}, \mathbf{p}_{dp}, \mathbf{ea}_{dp}\right) \tag{7}$$

$$\mathbf{r} = \left(\mathbf{e}_{p_1}^{jnt}, \mathbf{e}_{W_1}^{pow}, \ldots, \mathbf{e}_{p_n}^{jnt}, \mathbf{e}_{W_n}^{pow}, \ldots, \mathbf{e}_{W}^{bal}, (\mathbf{r}_{DGM}|\mathbf{r}_{IGM})\right) \tag{8}$$

Where $\mathbf{q}$ is the vector of joints values, $\mathbf{W}_{i\,(0)}$ the wrench at the i^{th} rods base and $\mathbf{p}_{dp}$ and $\mathbf{ea}_{dp}$ describes the platform position and Euler angles orientation respectively. The residual vector $\mathbf{r}$ contains the constrains $\mathbf{e}_{p_i}^{jnt}$ and $\mathbf{e}_{W_i}^{pow} = \mathbf{W}_i^p$ and the static equilibrium of the distal plate $\mathbf{e}_W^{bal}$, which satisfies the following equation.

$$\mathbf{e}_W^{bal} = \mathbf{W}_{dp}^{\mathbf{g}} + \sum_{i=0}^{n} \mathbf{W}_{i\,dp} + \mathbf{W}_{dp}^{\mathbf{ext}} = \mathbf{0} \tag{9}$$

Where $\mathbf{W}_{dp}^{\mathbf{g}}$ and $\mathbf{W}_{dp}^{\mathbf{ext}}$ are gravity and external wrenches acting on the distal plate.

Finally, the section $(\mathbf{r}_{DGM}|\mathbf{r}_{IGM})$ allows to handle both DGM and IGM with the same numerical framework [9]. It constrains either $\mathbf{q}$ or $\langle \mathbf{p}_{dp}, \mathbf{ea}_{dp} \rangle$ depending on the problem type. Using $\cdot^*$ for the desired values, we can describe $\mathbf{r}_{DGM}|\mathbf{r}_{IGM}$ as follows: $\mathbf{r}_{DGM} = (\mathbf{q}^* - \mathbf{q})$ and $\mathbf{r}_{IGM} = \left(\mathbf{p}_{dp}^* - \mathbf{p}_{dp}, \left[\mathbf{R}_{dp}^{*T}\mathbf{R}_{dp} - \mathbf{R}_{dp}^{*}\mathbf{R}_{dp}^{T}\right]^{\vee}\right)$.

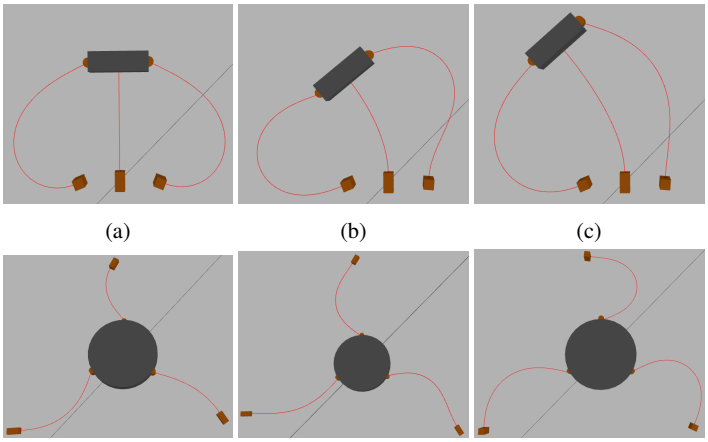
With this method, the variables in the vector $\mathbf{x}_0$ are computed in order to cancel the norm of $\mathbf{r}$. This problem requires a nonlinear solver: we used here the Levenberg-Marquardt algorithm implemented in the Ceres library [1].

4 The Gazebo simulator

Gazebo is a widely used simulator for the dynamic of rigid bodies [4]. It is open-source and based on the Open Dynamic Engine or Bullet physics. It hosts many robotics applications and, with our work, we want to bring the CPRs in this worldwide framework. Even if it does not support deformations, it provides extensions for its physics and rendering through plugins. Plugins allow users to control the Gazebo environment and physics. We developed our own plugin which accounts for the physics of the deformable links, applies the effects on the rigid bodies and displays the rod shapes using Gazebo rendering capabilities.

In order to simulate a CPR we need to specify its properties: deformable links are described in a dedicated *.yaml* file while the rigid bodies are described in a *.sdf* (simulation description format) file. This file contains all the physical, geometrical and visualization properties of the bodies.

We developed three main routines in our plugin. The first one is activated when all the parts of the CPR are loaded in Gazebo; our plugin starts retrieving their needed properties in order to construct the robot model: joints poses, geometries and axis ${}^{J_i}\mathbf{a}$ for all $i = 1, \ldots, n$. A second routine initializes the simulation finding a solution for the robot model. The shooting method requires an initial guess for begin the iterative process. This guess can be given by the user or can be found automatically. In the last case, the solver will start assuming all the forces and actuation values as zero. If no solution is found, then bounded random values for wrench and joints actuation are used to find a suitable solution. If no solution is found after a fixed number of tentative, the simulator exits with error. Once the robot is initialized, the third routine allows the user to move the robot. For this purpose, we created a Graphical User



(a) (b) (c)

(d) 3EFR moved in IGM. (e) 3PFR moved in IGM. (f) 3RFR moved in IGM.

.

Fig. 3: Simulation of four planar CPRs with different actuations. In Figures 3a, 3b and 3c, we show a robot with two actuated revolute joints and an extensible limb which is then fixed on the distal plate. In Figure 3d, we present a robot with three extensible limbs (3EFR). In Figure 3e, we depict a robot with prismatic joints at the base (3PFR). A robot with revolute joints at the base (3RFR) in Figure 3f.

Interface (GUI) to ease the usage of the simulator, which is available online in our GitHub repository[1].A video demonstration can be found on YouTube[2].

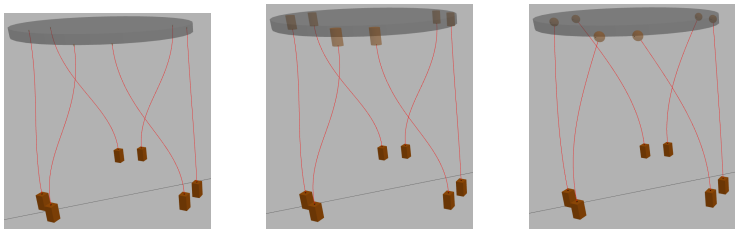
5 Simulations results

In this section, we want to present the simulation of some CPRs starting with a version of the simulator for planar CPRs for which loads, deformations and displacements happen in the plane only. As a result, their model is simpler compared to the one of a spatial CPR such as [8]. Illustrations of DGM and IGM simulations are represented respectively in Figure 3. Another version of the simulator is dedicated to the general case of spatial CPRs. Figure 4 shows the examples of three Stewart-Gough like platforms with different types of joints at the distal plate.

We used the simulator developed in [8] and carried out an identical robot model to be loaded in Gazebo and simulated with our simulator. We compared the length obtained for the extensible limb for different configurations of the robot. The relative error is negligible as its order is $\propto 10^{-4}$ m for rods length of $\propto 10^{0}$ m, which shows the accuracy of our simulator.

[1] https://github.com/aGotelli/A_Gazebo_Simulator_For_Continuum_Parallel_Robots.git

[2] https://youtu.be/6k5aZPOQjQ8



(a) Fixed joints. (b) Revolute joints. (c) Spherical joints.

Fig. 4: Simulation of three Stewart-Gough like platforms with different joint types.

6 Conclusions

In conclusion, we developed a simulator for CPRs, with ROS interfaces and designed for non-programmer users. Nonetheless, it can be used, extended, or modified in order to simulate specific CPRs. There are no other generic simulators in the literature and thus every researcher had to develop their own simulator in their own language, starting from scratch. We generalized the shooting method in order to deal with a wide range of CPRs, accounting for the geometry and dimensions of the joints.

Future works will target the implementation of a theory different from the one we used. We aim to accomplish real-time performances in Gazebo simulation in to order efficiently simulate CPRs, while preserving our generalized approach.

References

1. Agarwal, S., Mierle, K., Others: Ceres solver. http://ceres-solver.org (online)
2. Antman, S.: Nonlinear Problems of Elasticity. Applied Mathematical Sciences, Springer New York (2006), `https://books.google.fr/books?id=\_MagNyCXNqMC`
3. Black, C.B., Till, J., Rucker, D.C.: Parallel continuum robots: Modeling, analysis, and actuation-based force sensing. IEEE Transactions on Robotics **34**(1), 29–47 (2018)
4. Koenig, N., Howard, A.: Design and use paradigms for gazebo, an open-source multi-robot simulator. 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566) **3**, 2149–2154 (2004)
5. Murray, R.M., Li, Z., Sastry, S.S.: A mathematical introduction to robotic manipulation. CRC press (2017)
6. Okoli, F., Lang, Y., Kermorgant, O., Caro: Cable-driven parallel robot simulation using gazebo and ros. Robot Design, Dynamics and Control. CISM International Centre for Mechanical Sciences pp. 288–295 (2019)
7. Rucker, D.C., Jones, B.A., Webster III, R.J.: A geometrically exact model for externally loaded concentric-tube continuum robots. IEEE Transactions on Robotics **26**(5), 769–780 (2010)
8. Till, J., Bryson, C.E., Chung, S., Orekhov, A., Rucker, D.C.: Efficient computation of multiple coupled cosserat rod models for real-time simulation and control of parallel continuum manipulators. International Conference on Robotics and Automation (ICRA) pp. 5067–5074 (2015)
9. Zaccaria, F., Briot, S., Chikhaoui, M.T., Idá, E., Carricato, M.: An Analytical Formulation for the Geometrico-static Problem of Continuum Planar Parallel Robots (512–520) (2020)