https://doi.org/10.20965/jaciii.2022.p0125

Paper:

Synthesis and Prototyping of a 6-dof Parallel Robot for the Automatic Performance of Assembly Tasks

Jérôme Landuré[*], **Clément Gosselin**[*], **Thierry Laliberté**[*], **and Muhammad E. Abdallah**[**]

[*]Laboratoire de Robotique, Department of Mechanical Engineering, Université Laval
1065 Avenue de la médecine, Québec, Qc G1V0A6, Canada
E-mail: Clement.Gosselin@gmc.ulaval.ca
[**]Manufacturing Systems Research Lab., General Motors Research and Development
GM Technical Center, Warren, MI 48092, USA
E-mail: muhammad.abdallah@gm.com

[Received October 16, 2020; accepted November 29, 2021]

This paper presents the development of a 6-dof parallel robot for the performance of assembly tasks in a human-robot collaborative environment. The architecture and design of the robot are selected such that the robot is mechanically backdrivable. Thereby, the robot can physically interact with an environment or with humans without requiring the use of a force/torque sensor, which is the main objective of this work. The architecture of the robot is first described and its kinematic model is established. The Jacobian matrices are derived and an algorithm is presented for the determination of its workspace. The force capabilities of the robot are then established based on a proposed formulation. A prototype of the robot is presented and control schemes are developed, including a controller based on a vision system. Finally, a video demonstrating the experimental validation of the robot accompanies this paper. The video qualitatively demonstrates the interaction capabilities of the robot.

Keywords: assistive robot, vision control, impedance control, RCC

1. Introduction

Assembly tasks are involved in many industrial processes and humans are very effective at such tasks because of their adaptability. However, the repetitive nature of assembly tasks raises some issues in terms of ergonomy and comfort of the human operators. In order to alleviate these drawbacks, the concept of collaborative assembly, which involves humans and robots working together, has emerged [1]. The main objective of collaborative assembly is to reduce the physical and cognitive load on the human operators, while maintaining their engagement in the work. Additionally, introducing robots in the process can improve the productivity and tracability.

Collaborative robotics is used in several industrial settings. For instance, [2] presents a mechanism which helps for the intuitive lifting and moving of heavy objects. [3] presents several control schemes for actuated collaborative lifting mechanisms to improve their behaviour. [4] presents an impedance control scheme with underactuated degrees of freedom to improve the behaviour and bandwith of a mechanism compared to more conventional admittance methods. Also, several studies deal with the control of a robot working with a human operator in order to make collaborative work more intuitive. For instance, [5] presents a control scheme which introduces new parameters linked to the human perception such as the perceived weight and uses these human criteria as optimization factors.

The assistive robot must receive enough information to perform its task properly. Sensors coupled with a proper control strategy are a solution to entitle the robot to its needs to assess its environment. Computer vision is an attractive sensing approach because it is minimally invasive in a context of assembly task. The environment detection is a common vision problem, [6] exposes a method of segmentation of the environment with 3D cameras to semantically identify the objects in the scene. [7] proposes a 'on network' method to detect obstacles for a mobile robot to make the robot switch its configuration to adapt to its environment. In [8], the advantages of mounting a vision system on the mobile parts of a robot rather than on fixed components are discussed. [9] introduces vision with a robotic arm to prevent unexpected collisions, with a human for instance.

One of the issues with assembly operations is that some of the tasks must be performed in cluttered or difficult to reach configurations. Such situations generate ergonomic stressors because human operators must then adopt unfavourable poses. In order to alleviate this problem, it is proposed to use a small six-degree-of-freedom (6-dof) robot mounted on a supporting moveable structure that is controlled – manually or through an interface – by a human operator. The operator brings the robot close to the task to be performed, the robot detects is location with respect to the task using a vision system and then performs the task. This approach allows to keep the human operator in favourable – e.g., standing upright – poses, thereby

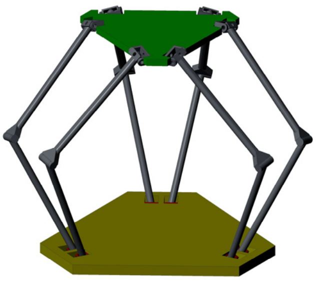


Fig. 1. Conceptual model of the spatial 6-URS robot.

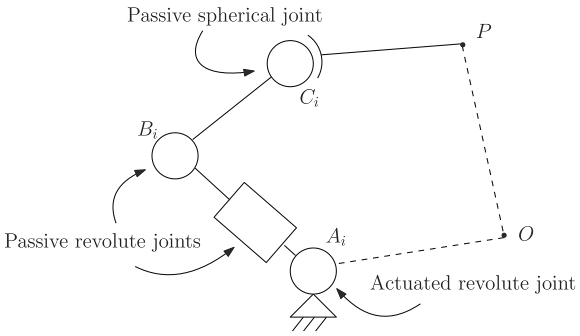


Fig. 2. Kinematic model of one leg of the 6-URS robot.

limiting the ergonomic stress. Since the robot is directly in contact with the task, it is desired to ensure that it has proper interaction capabilities. In order to address this issue, it is proposed here to design a mechanically backdrivable robot so that impedance control can be used to interact with the environment or with humans. The main advantage of this approach is that it does not require the use of a force/torque sensor and that it makes the interaction natural and easier to control. Additionally, the robot must have sufficient stiffness and force capabilities to perform assembly tasks. Combining these two requirements – backdrivability and force capabilities – is difficult for a serial robot. Therefore, a parallel robot architecture is proposed in this work.

This paper is structured as follows. In Section 2, the architecture and the kinematic model of the 6-dof parallel robot is described, including the solution to the inverse and direct kinematic problems. Section 3 presents the derivation of the Jacobian matrices and Section 4 provides an algorithm to determine the workspace of the robot, for design purposes. The force capabilities of the robot are then studied in Section 5 based on a proposed approach. Section 6 describes the prototype that was built as part of this work and Section 7 presents the control algorithms that were implemented to demonstrate the concept of the backdrivable parallel robot. Finally, Section 9 briefly describes the experimental validation and presents an accompanying video. In Section 10 some conclusions are drawn.

2. Kinematic Architecture of the Robot

The robot is composed of 6 parallel kinematic chains or “legs,” each of similar design and actuated by one motor.

The 6-dof robot is represented schematically in **Fig. 1** and the kinematic chain of one leg is presented at **Fig. 2**. The i-th leg is attached to the fixed base of the robot at point A_i and to the moving platform (end-effector) at point C_i. The leg is composed of two rigid bars linked at point B_i. The architecture of the robot is described as 6-$\underline{U}RS$, namely the joint at A_i is a universal joint with its proximal dof actuated, the joint at B_i is a passive revolute joint and the joint at C_i is a passive spherical joint. It should be noted that the axis of the second (passive) revolute joint of the universal joint located at point A_i is oriented along the line connecting point A_i to point B_i. As a consequence, the position of point B_i is dependent only on the i-th actuated joint coordinate and not on the motion of the passive dof of the U joint.

A large variety of parallel mechanisms can be synthesized based on a number of methods (see [10] for instance). The architecture described above is chosen to avoid the use of prismatic actuators – which are generally not backdrivable – and to limit the possibilities of mechanical interferences. Point P is the reference point on the end-effector while point O is the reference point on the base, i.e., the origin of the fixed reference frame. The position vector of point A_i with respect to the origin of the fixed reference frame is noted $\mathbf{a}_i$. Similarly, the position vectors of points B_i and C_i are noted $\mathbf{b}_i$ and $\mathbf{c}_i$. To control the robot, the solution of the inverse kinematic problem (IKP) as well as the Jacobian matrices of the robot (see [11] for instance) are needed.

2.1. Inverse Kinematic Problem (IKP)

The goal of the IKP is to determine the angles of the revolute joints at the motors for a given position and orientation of the end-effector. The IKP is derived from the constraint equation arising from the length of the distal part of each of the legs. Using the law of cosines on the triangle defined by points A_i, B_i, and C_i, one has:

$$(\mathbf{c}_i-\mathbf{b}_i)^2=(\mathbf{b}_i-\mathbf{a}_i)^2+(\mathbf{c}_i-\mathbf{a}_i)^2 \\ -2(\mathbf{c}_i-\mathbf{a}_i)^T\mathbf{Q}_i(\mathbf{b}_{i0}-\mathbf{a}_i) \quad . \quad . \quad . \quad . \quad (1)$$

where:

1. $\mathbf{Q}_i$ is the rotation matrix describing the motion controlled by the ith actuator, i.e., the rotation associated with the first dof of the U joint at point A_i

2. $(\mathbf{c}_i-\mathbf{b}_i)^2=(\mathbf{c}_i-\mathbf{b}_i)^T(\mathbf{c}_i-\mathbf{b}_i)$ is a constant value, referred to as h_i^2 in what follows. It is the square of the length of the distal link

3. $(\mathbf{b}_i-\mathbf{a}_i)^2=(\mathbf{b}_i-\mathbf{a}_i)^T(\mathbf{b}_i-\mathbf{a}_i)$ is a constant value, referred to as b_i^2 in what follows. It is the square of the length of the proximal link

4. $\mathbf{b}_{i0}$ is the position vector of point B_i corresponding to the reference orientation of the i-th actuator, used to set the 0 for the rotation angle of the actuated dof

In Eq. (1) and in what follows, one has $i = 1, \ldots, 6$, unless otherwise specified. The rotation matrix $\mathbf{Q}_i$ can be written in terms of the i-th joint coordinate θ_i as

$$\begin{aligned}\mathbf{Q}_i &= \mathbf{u}_i\mathbf{u}_i^T + \cos(\theta_i)\left[\mathbf{I}_3 - \mathbf{u}_i\mathbf{u}_i^T\right] \\ &\quad + \sin(\theta_i)\left[\mathbf{I}_3 \times \mathbf{u}_i\right] \quad \ldots\ldots\ldots\ldots \quad (2)\end{aligned}$$

where $\mathbf{u}_i$ is a unit vector in the direction of the axis of the i-th actuated joint and $\mathbf{I}_3$ stands for the 3×3 identity matrix.

Substituting Eq. (2) into Eq. (1), one obtains:

$$K_i + L_i \cos\theta_i + M_i \sin\theta_i = 0 \quad \ldots\ldots\ldots \quad (3)$$

where:

$$\begin{aligned}K_i &= h_i^2 - b_i^2 - (\mathbf{c}_i - \mathbf{a}_i)^2 \\ &\quad + 2(\mathbf{c}_i - \mathbf{a}_i)^T(\mathbf{u}_i\mathbf{u}_i^T)(\mathbf{b}_{i0} - \mathbf{a}_i) \quad \ldots\ldots \quad (4)\\ L_i &= 2(\mathbf{c}_i - \mathbf{a}_i)^T\left(\mathbf{I}_3 - \mathbf{u}_i\mathbf{u}_i^T\right)(\mathbf{b}_{i0} - \mathbf{a}_i) \quad \ldots\ldots \quad (5)\\ M_i &= 2(\mathbf{c}_i - \mathbf{a}_i)^T(\mathbf{I}_3 \times \mathbf{u}_i)(\mathbf{b}_{i0} - \mathbf{a}_i) \quad \ldots\ldots \quad (6)\end{aligned}$$

Substituting the following identities in Eq. (3):

$$\cos(\theta_i) = \frac{1 - \tan^2\left(\dfrac{\theta_i}{2}\right)}{1 + \tan^2\left(\dfrac{\theta_i}{2}\right)} \quad \ldots\ldots\ldots\ldots \quad (7)$$

$$\sin(\theta_i) = \frac{2\tan\left(\dfrac{\theta_i}{2}\right)}{1 + \tan^2\left(\dfrac{\theta_i}{2}\right)} \quad \ldots\ldots\ldots\ldots \quad (8)$$

then yields

$$(K_i - L_i)\tan^2\left(\frac{\theta_i}{2}\right) + 2M_i\tan\left(\frac{\theta_i}{2}\right) + (K_i + L_i) = 0 \quad \ldots\ldots\ldots\ldots \quad (9)$$

Therefore, the 2 solutions for the actuated joint coordinates are:

$$\theta_{i1} = 2\arctan\left(\frac{-M_i + \sqrt{M_i^2 + K_i^2 - L_i^2}}{K_i - L_i}\right) \quad . \quad (10)$$

$$\theta_{i2} = 2\arctan\left(\frac{-M_i - \sqrt{M_i^2 + K_i^2 - L_i^2}}{K_i - L_i}\right) \quad . \quad (11)$$

This means that the IKP has two solutions for each leg, and hence that for a given configuration (position and orientation) of the end-effector, the IKP for the whole mechanism has $2^6 = 64$ solutions.

2.2. Direct Kinematic Problem (DKP)

The IKP gives the joint coordinnates knowing the pose of the end-effector. However, some situations require the opposite, i.e., to obtain the pose of the end-effector knowing the joint coordinnates. This is the problem referred to as direct kinematic problem or DKP.

For convenience, the rotation matrix representing the orientation of the end-effector is partitioned into three column vectors as $\mathbf{Q} = \begin{bmatrix}\mathbf{q}_1 & \mathbf{q}_2 & \mathbf{q}_3\end{bmatrix}$ and a vector that includes 12 parameters is formed with the translations and rotation parameters, of the platform, namely

$$\mathbf{s} = \begin{bmatrix}\mathbf{p}\\ \mathbf{q}_1\\ \mathbf{q}_2\\ \mathbf{q}_3\end{bmatrix}_{12\times 1} \quad \ldots\ldots\ldots\ldots\ldots \quad (12)$$

Vector $\mathbf{s}$ is a redundant representation of the pose, which introduces robustness in the solution of the direct kinematic problem (DKP). A set of constraint equations must then be written, including the loop closure equations and the constraint equations on the column vectors forming matrix $\mathbf{Q}$. These equations are written as

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix}(\mathbf{c}_1 - \mathbf{b}_1)^T(\mathbf{c}_1 - \mathbf{b}_1) - l_1^2\\ (\mathbf{c}_2 - \mathbf{b}_2)^T(\mathbf{c}_2 - \mathbf{b}_2) - l_2^2\\ (\mathbf{c}_3 - \mathbf{b}_3)^T(\mathbf{c}_3 - \mathbf{b}_3) - l_3^2\\ (\mathbf{c}_4 - \mathbf{b}_4)^T(\mathbf{c}_4 - \mathbf{b}_4) - l_4^2\\ (\mathbf{c}_5 - \mathbf{b}_5)^T(\mathbf{c}_5 - \mathbf{b}_5) - l_5^2\\ (\mathbf{c}_6 - \mathbf{b}_6)^T(\mathbf{c}_6 - \mathbf{b}_6) - l_6^2\\ \mathbf{q}_1^T\mathbf{q}_2\\ \mathbf{q}_1^T\mathbf{q}_3\\ \mathbf{q}_2^T\mathbf{q}_3\\ \mathbf{q}_1^T\mathbf{q}_1 - 1\\ \mathbf{q}_2^T\mathbf{q}_2 - 1\\ \det(\mathbf{Q}) - 1\end{bmatrix}_{12\times 1} = \mathbf{0}. \quad . \quad (13)$$

The first 6 equations are related to the lengths of links B_iC_i. The following three equations represent the orthogonal nature of the vectors composing a rotation matrix. The next two equations correspond to the fact that the vectors composing the rotation matrix must be unit vectors. Finally, the last equation ensures that the last column of the rotation matrix is a unit vector and that the matrix is a proper orthogonal matrix (not a reflection). The solution of the DKP consists in finding a pose of the platform that satisfies all the equations in Eq. (13), from given encoder readings. From the encoder readings, vectors $\mathbf{b}_i$ are readily computed. Also, from an estimated pose of the platform, vector $\mathbf{s}$ and position vectors $\mathbf{c}_i$ are readily computed. The limited Taylor expansion of Eq. (13) is then written as

$$\mathbf{f}(\mathbf{s}) = \mathbf{f}(\mathbf{s}_0) + \frac{d\mathbf{f}}{d\mathbf{s}}\Delta_\mathbf{s} \quad \ldots\ldots\ldots\ldots \quad (14)$$

Since it is desired to have $\mathbf{f}(\mathbf{s}) = \mathbf{0}$, one has

$$\Delta_\mathbf{s} = -\left(\frac{d\mathbf{f}}{d\mathbf{s}}\right)^{-1}\mathbf{f}(\mathbf{s}_0) \quad \ldots\ldots\ldots\ldots \quad (15)$$

where $d\mathbf{f}/d\mathbf{s}$ is the (12×12) Jacobian matrix of the system of equations in Eq. (13), which is readily obtained analytically. Eq. (15) is used iteratively, until a solution to Eq. (13) is found within a prescribed error threshold. With appropriate conditioning for the equations, the algorithm is fast and robust.

3. Jacobian Matrices

The Jacobian matrices provide the velocity relationships between the controlled joint coordinates and the end effector coordinates. To obtain these matrices, Eq. (1) is first rewritten as

$$\begin{aligned} h_i^2 &= b_i^2 + (\mathbf{c}_i - \mathbf{a}_i)^T(\mathbf{c}_i - \mathbf{a}_i) \\ &\quad -2(\mathbf{c}_i - \mathbf{a}_i)^T(\mathbf{b}_i - \mathbf{a}_i) \end{aligned} \qquad (16)$$

Differentiating Eq. (16) with respect to time and noting that $\dot{\mathbf{a}}_i = \mathbf{0}$ then yields

$$(\mathbf{c}_i - \mathbf{a}_i)^T \dot{\mathbf{c}}_i = (\mathbf{c}_i - \mathbf{a}_i)^T \dot{\mathbf{b}}_i + (\mathbf{b}_i - \mathbf{a}_i)^T \dot{\mathbf{c}}_i \qquad (17)$$

which can be rearranged as

$$(\mathbf{c}_i - \mathbf{b}_i)^T \dot{\mathbf{c}}_i = (\mathbf{c}_i - \mathbf{a}_i)^T \dot{\mathbf{b}}_i. \qquad (18)$$

Furthermore, one has

$$\dot{\mathbf{b}}_i = \theta_i \mathbf{u}_i \times (\mathbf{b}_i - \mathbf{a}_i) \qquad (19)$$

and,

$$\dot{\mathbf{c}}_i = \dot{\mathbf{p}} + \boldsymbol{\omega} \times (\mathbf{c}_i - \mathbf{p}) \qquad (20)$$

where $\dot{\mathbf{p}}$ is the velocity of vector point P of the end effector and $\boldsymbol{\omega}$ is the angular velocity of the end effector.

Substituting Eqs. (19) and (20) into Eq. (18) then leads to

$$\begin{aligned} &((\mathbf{b}_i - \mathbf{a}_i) \times (\mathbf{c}_i - \mathbf{a}_i))^T \mathbf{u}_i \dot{\theta}_i \\ &\quad = (\mathbf{c}_i - \mathbf{b}_i)^T \mathbf{t} + ((\mathbf{c}_i - \mathbf{p}) \times (\mathbf{c}_i - \mathbf{b}_i))^T \boldsymbol{\omega} \ . \end{aligned} \qquad (21)$$

This equation can be written in matrix form as:

$$\mathbf{K}\dot{\boldsymbol{\theta}} = \mathbf{J}\dot{\mathbf{x}} \qquad (22)$$

with $\dot{\boldsymbol{\theta}} = \begin{bmatrix} \dot{\theta}_1 & \dot{\theta}_2 & \dot{\theta}_3 & \dot{\theta}_4 & \dot{\theta}_5 & \dot{\theta}_6 \end{bmatrix}^T$ the joint velocity vector and $\dot{\mathbf{x}} = \begin{bmatrix} \boldsymbol{\omega} & \dot{\mathbf{p}} \end{bmatrix}^T$ the generalised end-effector velocity, and where the Jacobian matrices are written as

$$\begin{aligned} \mathbf{K} = diag(((\mathbf{b}_i - \mathbf{a}_i) \times (\mathbf{c}_i - \mathbf{a}_i))^T \mathbf{u}_{\mathbf{i}}, \\ i = 1, \ldots, 6) \end{aligned} \qquad (23)$$

$$\mathbf{J} = \begin{bmatrix} ((\mathbf{c}_1 - \mathbf{p}) \times (\mathbf{c}_1 - \mathbf{b}_1))^T & (\mathbf{c}_1 - \mathbf{b}_1)^T \\ ((\mathbf{c}_2 - \mathbf{p}) \times (\mathbf{c}_2 - \mathbf{b}_2))^T & (\mathbf{c}_2 - \mathbf{b}_2)^T \\ ((\mathbf{c}_2 - \mathbf{p}) \times (\mathbf{c}_2 - \mathbf{b}_2))^T & (\mathbf{c}_2 - \mathbf{b}_2)^T \\ ((\mathbf{c}_2 - \mathbf{p}) \times (\mathbf{c}_2 - \mathbf{b}_2))^T & (\mathbf{c}_2 - \mathbf{b}_2)^T \\ ((\mathbf{c}_2 - \mathbf{p}) \times (\mathbf{c}_2 - \mathbf{b}_2))^T & (\mathbf{c}_2 - \mathbf{b}_2)^T \\ ((\mathbf{c}_2 - \mathbf{p}) \times (\mathbf{c}_2 - \mathbf{b}_2))^T & (\mathbf{c}_2 - \mathbf{b}_2)^T \end{bmatrix} \qquad (24)$$

4. Workspace

In this application, the robot is brought close to the task. Hence, the target workspace is a sphere with a diameter of approximately 12 cm. To ensure that this goal is reached, design tools were developed to assess and adjust the capabilities of the robot. The static workspace of the robot is considered, meaning that singularities cannot be overcome with momentum and that the workspace must be free from singularities.

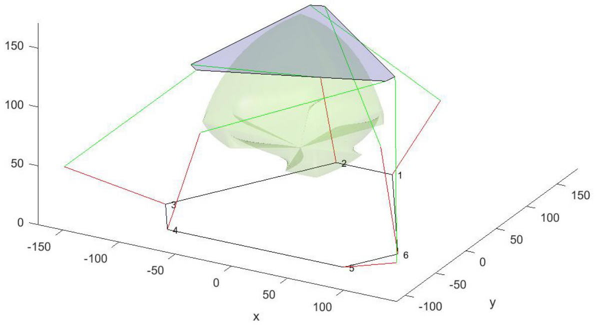


Fig. 3. Representation of the geometric workspace of the robot in a 80° torsion orientation from its reference orientation.

In order to estimate the workspace of the robot, a numerical method based on the geometric equations derived above is used. The workspace is defined here as the set of possible poses (position and orientation) of the end-effector that can be reached with a non-singular configuration of the robot. These poses correspond to all the potential solutions of the IKP and thus all the solutions of Eq. (9). The IKP produces a solution when its discriminant Δ is greater than or equal to zero, thus

$$\Delta_i = 4(M_i^2 - K_i^2 + L_i^2) \geq 0. \qquad (25)$$

This condition must be verified for all 6 legs of the robot for a given pose to be a potential configuration. A numerical approach has been selected for the determination of the workspace because it allows one to include as many constraints as needed, in addition to the limitations given by Eq. (25), such as joint limits, mechanical interferences, singularities, and others. The robot has six degrees of freedom, and hence its workspace is embedded in a six-dimensional space, which cannot be easily visualized. Therefore, the orientation of the platform is prescribed and then the algorithm to generate the position workspace is used. The result is a region of the space within which it is required to evaluate the performance of the robot. This exercise is repeated for different targeted orientations. An example result is shown in **Fig. 3** where the workspace of the robot is shown for an orientation of the platform that corresponds to a 80° torsion from the reference orientation. The estimated workspace corresponds to all reachable positions for a reference point on the end-effector, for the prescribed orientation.

However, it is known that the workspace of a robot is not only limited by the reachable space. Singular configurations (also referred to as singularities, see [12–14] for general insights on sigularities for parallel manipulators or [15–17] for more specific examples) are a limiting working condition for robots, especially parallel robots. These configurations are configurations in which the be-

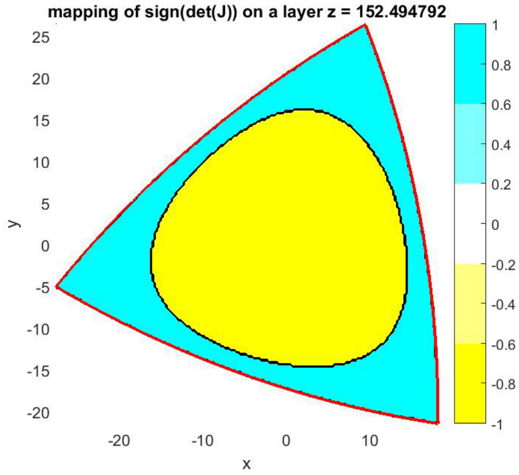


Fig. 4. Singularity locus detection in a section of the workspace of the robot, the robot is in a 80° Torsion orientation from its reference orientation.

haviour of the robot can change unexpectedly. For instance, in singular configurations, the force output possible in some direction drops considerably. Parallel robots are known to have singular configurations that can be located within the boundaries of their reachable workspace. Therefore, a separate analysis must be conducted in order to identify the singular configurations. The workspace is sliced into layers and in each of the layers, the determinant of matrix $\mathbf{J}$ is computed. A singular configuration is reached when the determinant reaches zero (see [18–21] for analysis of the roots of the determinant to study singular configurations). The numerical method to search for zeros is to check the sign of the determinant.

Figure 4 shows an example of the representation of the sign of the determinant. The region in which the determinant is positive is shown in yellow while the region in which the determinant is negative is shown in blue. The black boundary between yellow and blue regions represents the singularity locus. In this example, only the yellow zone could be exploited. Hence, the actual workspace may be considerably reduced by singularities, as illustrated in **Fig. 4**. In a context of design, the goal is to find design parameters that will yield the largest possible useful workspace with simple shapes, in order for the robot to be easy to exploit in practice. The parameters of the robot, especially the location of points A_i and direction of motor axes $\mathbf{u}_i$ greatly affect the results. Several iterations on the design were completed in order to obtain the best possible results.

5. Force Capabilities

In this section, a mathematical model is developed in order to determine the force capabilities of the robot. Also, a procedure is proposed to investigate the capability of the robot to apply forces in different directions and

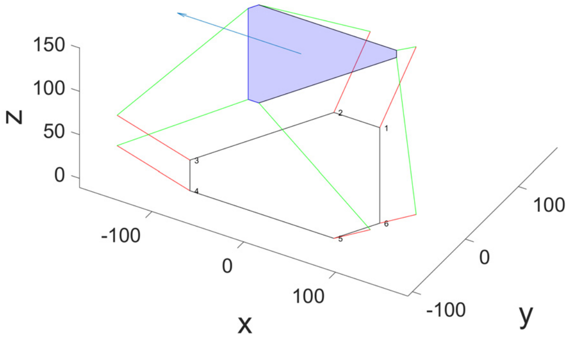


Fig. 5. Model of the robot used for the force analysis. The arrow at the top indicates the direction of the force.

to assess the impact of such forces on the actuators. This analysis is important in order to assess the tasks that can be performed by the manipulator. The model used to estimate the forces is the static model, obtained from the principle of virtual power. We can write that the input and output virtual powers must be equal, namely

$$P_{in} = P_{out} \quad \text{(26)}$$

$$\dot{\boldsymbol{\theta}}^T \mathbf{c} = \dot{\mathbf{x}}^T \mathbf{f} \quad \text{(27)}$$

where $\dot{\boldsymbol{\theta}}$ is the vector of actuated joint velocities, $\mathbf{c}$ is the vector of joint torques, $\dot{\mathbf{x}}$ is the vector of Cartesian velocities and $\mathbf{f}$ is the vector of Cartesian forces and torques. Substituting Eq. (22) into Eq. (27), one then has

$$\dot{\boldsymbol{\theta}}^T \mathbf{c} = \left[\mathbf{J}^{-1}\mathbf{K}\dot{\boldsymbol{\theta}}\right]^T \mathbf{f} \quad \text{(28)}$$

which yields

$$\dot{\boldsymbol{\theta}}^T \mathbf{c} = \dot{\boldsymbol{\theta}}^T \mathbf{K}^T \mathbf{J}^{-T} \mathbf{f}. \quad \text{(29)}$$

For this relation to remain true for any value of $\dot{\boldsymbol{\theta}}$, the following relation must be satisfied:

$$\mathbf{c} = \mathbf{K}^T \mathbf{J}^{-T} \mathbf{f} \quad \text{(30)}$$

This is a simplified expression of the generalized formula shown in [22]. From this relation, for a known load applied on the effector, we can estimate the required actuator torques. For instance, the estimated weight of the mobile parts is 200 g, overestimated at 300 g. It leads to a 3 N force applied at the effector. The direction of the gravity is set in different possibilities.

Figure 5 represents the robot in one of the configurations and the arrow at the top on the end-effector shows the direction of the force applied. **Fig. 6** shows a computation of the required torque along a displacement on the vertical axis. It shows that the worst configuration requires a little more than 0.1 Nm. No reduction stage was introduced here at the actuated joints in order to keep the transmission easily backdrivable. With such an arrangement, the mechanism can allow compliant movements, which is desirable for interactions with the task. Furthermore, end force impacts on the motor current is not scaled down by the reduction ratio and could be measured with

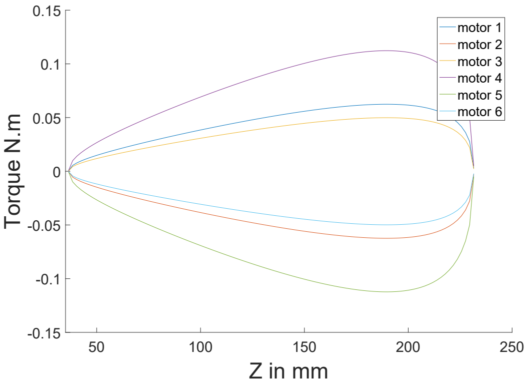


Fig. 6. Torques required at the motors with a 3 N force applied at the effector in the $-X$ direction.

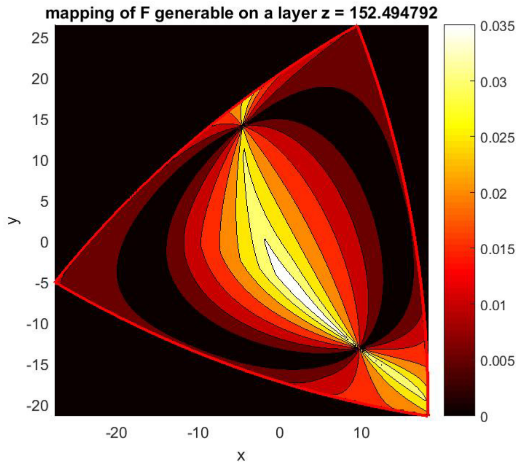


Fig. 7. Force ratio potency of the robot in the X direction in a section of its workspace, the robot is in a 80° Torsion orientation from its reference orientation.

enough accuracy to be included in the control scheme.

Figure 7 shows the same section of the workspace as **Fig. 4** but presents the force transmission ratio in the "X" direction. It can be observed that the force transmission ratio is close to zero in the areas close to the singular configurations.

The above design tools were used to tune the geometric parameters of the robot. Among others, the force ratio was used to ensure the capabilities of the robot. The method of computation is as follows. First, a force/moment direction is chosen. This is the direction along which the force ratio of the robot is to be computed. It is comprised in vector **f**, for instance, a force in the "Y" direction corresponds to $\mathbf{f} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}^T$. The magnitude of **f** is not important at this step. Then, the actuator torques are computed using Eq. (30). The result is then normalized, i.e., the result is adjusted so that when multiplied by the maximum available torque, the maximum possible force in the chosen direction is obtained. Force ratios are defined as:

$$\frac{F_{output}}{F_{input}} = R \quad \ldots\ldots\ldots\ldots \quad (31)$$

However, there are six inputs instead of one for one output. Therefore, for a set of intended similar actuators with the same maximum available torque, the above results are multiplied by a coefficient so that the maximum absolute value of torque is 1 (in the wanted unit, Nm for instance). If actuators with different force capabilities are intended, coefficients should be applied accordingly. Then, Eq. (30) is applied in the reverse direction:

$$\mathbf{f} = \mathbf{J}^T \mathbf{K}^{-T} \mathbf{c} \quad \ldots\ldots\ldots\ldots \quad (32)$$

The result gives a coefficient when multiplied by the maximum force of the chosen type of actuator (maximum continuous torque for a motor/maximum peak torque depending the intended process of use) gives the potential maximum force that the robot can produce in this direction. Because the Jacobian matrices change with the configuration, the result changes with the location. That is why this method is computed for all the configurations in the previously bounded workspace. An example of result is presented through a contour plot in **Fig. 7** for a 80° twist orientation. Overall, we estimate the real workspace as the reachable space not crossed by singularities in which sufficient force capabilities can be deployed. Because of the 6 dofs, the impact of the design parameters on the properties of the parallel architecture can be difficult to predict. However, the above analysis allowed an iterative design exercise that led to an architecture for which the workspace and force capabilities where deemed sufficient. Using the tilt and torsion convention (proposed originally in [23] as a novel three-angle orientation representation), we estimate that the rotational workspace of the robot allows tilt angles of ±30° and torsion angles of ±60° throughout the position workspace.

6. Prototyping

The CAD model of the robot is shown in **Fig. 8** and a photograph of the manufactured prototype is shown in **Fig. 9**. Bearings, bushings, retaining rings and screws are used with care to produce an easy to build mechanism with low friction motions. No gear reduction is used at the actuated joints in order to obtain a backdrivable mechanism, therefore, the motors need to be able to produce high torques with a low need for high velocities. Indeed, the parallel robot is designed to produce an intuitive interaction with the environment or with humans and not to produce high velocities. The selected motors are Maxon EC90 Flat, brushless, 260 W with 4096 count per turns encoders, two channels. They can reach 1 Nm constant torque output. The motors are controlled using DZEANTU-020B080 servo drives from Advanced Motion Controls and the Simulink Real-Time environment from MathWorks.

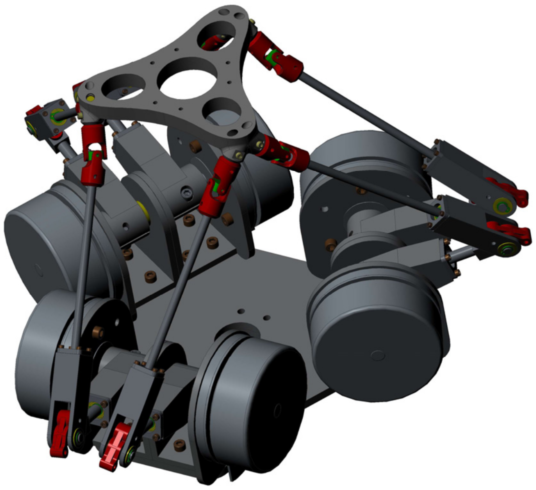


Fig. 8. CAD model of the final design of the 6-URS robot.

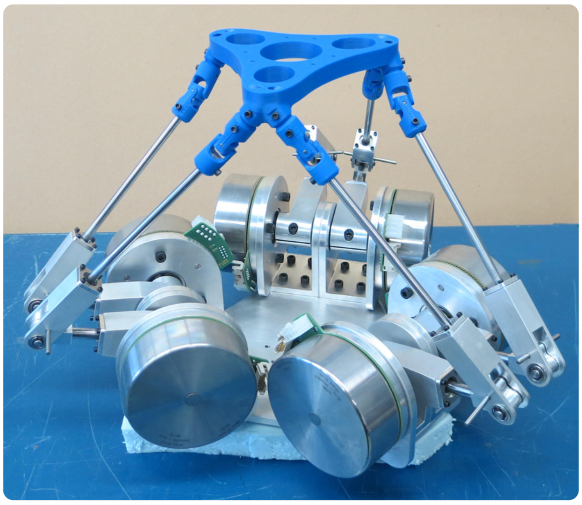


Fig. 9. Photograph of the prototype of the 6-URS robot.

7. Control Schemes

Standard control algorithms are implemented to demonstrate the capabilities of the robot. The algorithms are taken from the literature (see [24] for instance) and they are briefly described in this section.

7.1. Position Control

Position control is a standard first step in the process of controlling a robot. It validates the mathematical model and the tools developed for it are useful for further models.

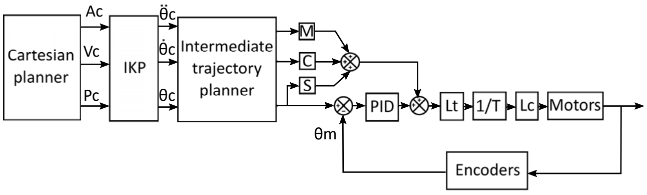


Fig. 10. Position control scheme.

The position control scheme is shown in **Fig. 10**. It works as follows. First, a pose must be input as the desired pose of the robot, a pose being the location of the reference point as well as the orientation of the end-effector. This can be extended as a trajectory, meaning a set of poses. That is why the input of the control scheme is referred to as a planner. In the case of a trajectory, acceleration and velocity information should be provided as well to include the complete kinematic behaviour in the scheme. The Cartesian pose or trajectory is converted to joint coordinates using the solution of the IKP (Eqs. (10) and (11) determined in Section 2.1). Then, a closed control loop is applied using the data collected by the encoders. The angle difference between the desired position of the actuated joints and the measured one is fed to the PID regulator. The output is considered as a torque, to which an estimated required torque based on the desired trajectory is added. Then, a torque limiter L_t limits the maximum torque sent. Afterwards, this torque value is converted to a current value with the motor current to torque ratio T, using its inverse in this case. Then, another limiter is added for the current L_c. Two limiters are redundant but they cater to different limiting reasons and are therefore useful as a simple means of guaranteeing basic motor protection. The vector of joint velocities $\dot{\boldsymbol{\theta}}$ is obtained using Eq. (22), namely

$$\dot{\boldsymbol{\theta}} = \mathbf{K}^{-1}\mathbf{J}\dot{\mathbf{x}} \quad \dots\dots\dots\dots\dots \quad (33)$$

with $\mathbf{K}$ and $\mathbf{J}$ the matrices developed in Eqs. (23) and (24). The joint accelerations $\ddot{\boldsymbol{\theta}}$ are obtained by differentiating Eq. (33) with respect to time, which yields

$$\ddot{\boldsymbol{\theta}} = \mathbf{K}^{-1}(\mathbf{J}\ddot{\mathbf{x}} + \dot{\mathbf{J}}\dot{\mathbf{x}} - \dot{\mathbf{K}}\dot{\boldsymbol{\theta}}) \quad \dots\dots\dots \quad (34)$$

with the following expressions for $\dot{\mathbf{J}}$ and $\dot{\mathbf{K}}$,

$$\dot{\mathbf{J}} = \begin{bmatrix} \left((\dot{\mathbf{c}}_1-\dot{\mathbf{p}})\times(\mathbf{c}_1-\mathbf{b}_1)+(\mathbf{c}_1-\mathbf{p})\times(\dot{\mathbf{c}}_1-\dot{\mathbf{b}}_1)\right)^T & (\dot{\mathbf{c}}_1-\dot{\mathbf{b}}_1)^T \\ \left((\dot{\mathbf{c}}_2-\dot{\mathbf{p}})\times(\mathbf{c}_2-\mathbf{b}_2)+(\mathbf{c}_2-\mathbf{p})\times(\dot{\mathbf{c}}_2-\dot{\mathbf{b}}_2)\right)^T & (\dot{\mathbf{c}}_2-\dot{\mathbf{b}}_2)^T \\ \left((\dot{\mathbf{c}}_3-\dot{\mathbf{p}})\times(\mathbf{c}_3-\mathbf{b}_3)+(\mathbf{c}_3-\mathbf{p})\times(\dot{\mathbf{c}}_3-\dot{\mathbf{b}}_3)\right)^T & (\dot{\mathbf{c}}_3-\dot{\mathbf{b}}_3)^T \\ \left((\dot{\mathbf{c}}_4-\dot{\mathbf{p}})\times(\mathbf{c}_4-\mathbf{b}_4)+(\mathbf{c}_4-\mathbf{p})\times(\dot{\mathbf{c}}_4-\dot{\mathbf{b}}_4)\right)^T & (\dot{\mathbf{c}}_4-\dot{\mathbf{b}}_4)^T \\ \left((\dot{\mathbf{c}}_5-\dot{\mathbf{p}})\times(\mathbf{c}_5-\mathbf{b}_5)+(\mathbf{c}_5-\mathbf{p})\times(\dot{\mathbf{c}}_5-\dot{\mathbf{b}}_5)\right)^T & (\dot{\mathbf{c}}_5-\dot{\mathbf{b}}_5)^T \\ \left((\dot{\mathbf{c}}_6-\dot{\mathbf{p}})\times(\mathbf{c}_6-\mathbf{b}_6)+(\mathbf{c}_6-\mathbf{p})\times(\dot{\mathbf{c}}_6-\dot{\mathbf{b}}_6)\right)^T & (\dot{\mathbf{c}}_6-\dot{\mathbf{b}}_6)^T \end{bmatrix} \quad \dots\dots\dots\dots\dots\dots\dots\dots \quad (35)$$

and

$$\dot{\mathbf{K}} = diag\Big((\dot{\mathbf{b}}_i \times (\mathbf{c}_i - \mathbf{a}_i) + (\mathbf{b}_i - \mathbf{a}_i) \times \dot{\mathbf{c}}_i)^T \mathbf{u}_i, i = 1,\dots,6\Big) \quad \dots \quad (36)$$

where the expressions for $\dot{\mathbf{b}}_i$ and $\dot{\mathbf{c}}_i$ are given in Eq. (19) and in Eq. (20).

The estimation of the motor positions, velocities and

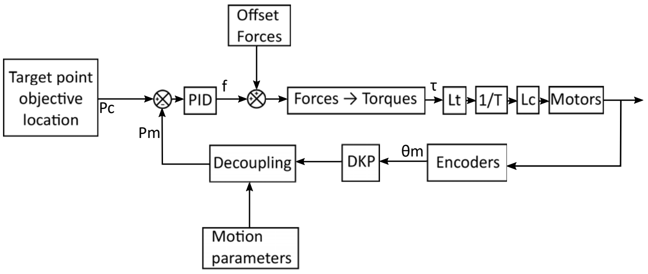


Fig. 11. Hybrid force/position control scheme.

accelerations enables to estimate the required torque during the trajectories with the following expression:

$$\mathbf{t}_q = \mathbf{M}\ddot{\boldsymbol{\theta}} + \mathbf{C}\dot{\boldsymbol{\theta}} + \mathbf{S}\boldsymbol{\theta} + \mathbf{g}(\boldsymbol{\theta}). \quad \ldots\ldots\ldots \quad (37)$$

with **M** the inertia matrix of the mobile parts, **C** the damping matrix, **S** the stiffness matrix and $\mathbf{g}(\boldsymbol{\theta})$ the gravity term. The system does not include viscous or elastic components and the material properties that could introduce such variables were considered negligible. The inertia matrix includes the inertia reflected for each motor, the parts being mobile, these values change while the robot is moving. An average inertia was estimated from the design software. Albeit less accurate than a complete analysis, it was still an interesting addition to the control while not requiring much time to implement. This predictive control allows the regulator to be less solicited and more efficient thanks to a more accurate understanding of the physics of the mechanism.

7.2. Hybrid Control

A hybrid position/force control scheme was also implemented in order to demonstrate the interaction capabilities of the robot.

This control scheme is shown in **Fig. 11**. As mentioned above, the robot is backdrivable. This property can be used to enhance the collaborative behaviour of the robot. Indeed, the interaction between the environment (operator, external objects) and the end-effector of the robot can be used to produce behaviours that mimic the action of passive spring-based RCC mechanisms (see [25] as reference) with the benefit of being programmable and supporting additional control features such as vision detection. This approach is similar to admittance control (discussed in [26] for instance) but no force sensors are involved. The "force" is determined by how much the end part is moved from the desired controlled position, i.e., impedance control is implemented. In other words, this deviation is assimilated to a force similar to how a spring would behave. For this reason, it is important to determine the pose of the end-effector, which is obtained through the solution of the direct kinematic problem (determined in Section 2.2), from the encoder readings.

The objective of this control scheme is to emulate the behaviour of a RCC mechanism. It means that the robot must be able to produce an elastic behaviour near a chosen point for the six degrees of freedom. The advantages of the emulation of the RCC are that the location of the target point can be changed arbitrarily and the elastic stiffness can be changed independently with any value. The control scheme is shown in **Fig. 11**. The location of the target Cartesian configuration is chosen in the Cartesian block, specified in the "Motion parameters" block. The real location of the robot is measured with the encoders and the computation of the solution of the direct kinematic problem. The position differences must generate proportional forces multiplied by the elastic coefficient; for the PID, only a P regulator is used if the desired behaviour is purely elastic. However, some derivative coefficient can be added to damp the oscillations. Then, these Cartesian forces are converted to motor torques with the forces to torques relation (special case of [22]), one has:

$$\tau = \mathbf{K}^T \mathbf{J}^{-T} \begin{bmatrix} \mathbf{f}_t \\ \mathbf{f}_r \end{bmatrix} \quad \ldots\ldots\ldots\ldots \quad (38)$$

where f_t is the vector of translational Cartesian forces and f_r the vector of Cartesian torques. These efforts are chosen to have proportional behaviour with the deviation between the targeted location and the measured one:

$$\mathbf{f}_t = \mathbf{K}_p \Delta\mathbf{t} \quad \ldots\ldots\ldots\ldots\ldots \quad (39)$$

$$\mathbf{f}_r = \mathbf{K}_\theta \Delta\boldsymbol{\theta} \quad \ldots\ldots\ldots\ldots\ldots \quad (40)$$

with

$$\mathbf{K}_t = \begin{bmatrix} k_{tx} & 0 & 0 \\ 0 & k_{ty} & 0 \\ 0 & 0 & k_{tz} \end{bmatrix} \quad \ldots\ldots\ldots\ldots \quad (41)$$

$$\mathbf{K}_\theta = \begin{bmatrix} k_{\theta x} & 0 & 0 \\ 0 & k_{\theta y} & 0 \\ 0 & 0 & k_{\theta z} \end{bmatrix} \quad \ldots\ldots\ldots\ldots \quad (42)$$

with "k_{xx}" the chosen elastic coefficient.

One important aspect is the handling of the rotations. Preferably, the rotations should be decoupled from one another to get a proper consistent behaviour, which is not compatible with Euler angles where the angles are multiplied to get the orientation matrix. To get decoupled rotations around the Cartesian axes, quaternions are used instead. The transformation formula from the orientation matrix to quaternions and reversed are presented in [27]. Quaternions are useful to produce a continuous model for the representation of orientation space, as discussed in [28], which is important for the convergence of the algorithm. The quaternions yield three independent angles around the axes of the reference frame, and the deviation between these angles and the targeted angles produce the torques to reproduce the behaviour of torsion springs around the axes.

Another aspect explored was the introduction of damping in the control scheme. Passive RCC mechanisms do not have damping elements but it was easy to introduce them here by adding another force based on the derivative of Δp and $\Delta\theta$ with other coefficient matrices $\mathbf{D}_t$ and $\mathbf{D}_\theta$. Eqs. (39) and (40) become

$$\mathbf{f}_t = \mathbf{K}_t \Delta t + \mathbf{D}_t \Delta \dot{\mathbf{t}} \quad \text{. . . . .} \quad (43)$$

$$\mathbf{f}_r = \mathbf{K}_\theta \Delta \theta + \mathbf{D}_\theta \Delta \dot{\boldsymbol{\theta}} \quad \text{. . . . .} \quad (44)$$

with

$$\mathbf{D}_t = \begin{bmatrix} d_{tx} & 0 & 0 \\ 0 & d_{ty} & 0 \\ 0 & 0 & d_{tz} \end{bmatrix} \quad \text{. . . . .} \quad (45)$$

$$\mathbf{D}_\theta = \begin{bmatrix} d_{\theta x} & 0 & 0 \\ 0 & d_{\theta y} & 0 \\ 0 & 0 & d_{\theta z} \end{bmatrix} \quad \text{. . . . .} \quad (46)$$

with "d_{xx}" the chosen damping coefficient. The introduction of damping significantly reduces the oscillations produced by high stiffness coefficients but lessens the similarities with the behaviour of a conventional RCC.

The hybrid control scheme simulates the behaviour of a RCC mechanism, with the adding benefits of changeable coefficients for the stiffness. The location of the centre of rotation can also be changed. This is a very interesting feature because a centre of rotation far from the contact point is difficult to achieve with passive mechanisms. Moreover, the centre of rotation of the original RCC design is not fixed, only allowing small motions to keep a consistent behaviour, which is not a problem for the control scheme. However, the robot presented here is not suited to very far virtual centre of rotation because of singularities, different designs should be explored to benefit from this aspect of the control scheme.

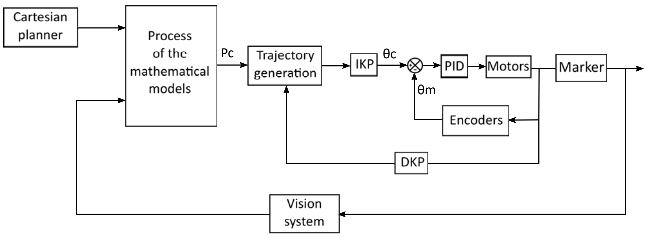


Fig. 12. Vision based control scheme.

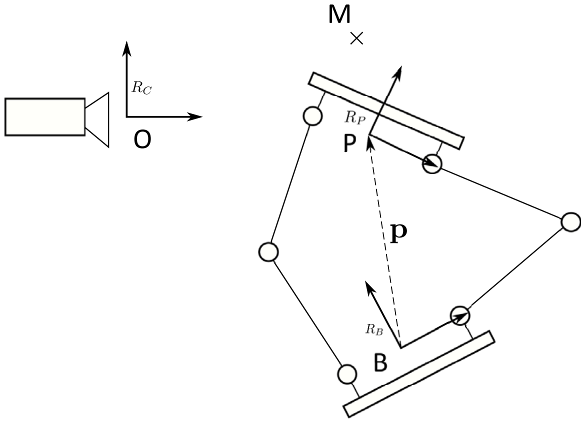


Fig. 13. Geometric model to solve the relative positions and orientations required for the vision based control scheme.

7.3. Vision-Based Control

The third control scheme implemented in this work makes use of a camera and computer vision algorithms to guide the motion of the robot. In the intended work scenario, the human operator brings the robot close to the task to be performed and then the robot performs the task using computer vision to guide the motion of the end-effector.

Some state-of-the-art methods were explored to detect the pose of the parts like SIFT (see [29] for the original reference) and ORB (see [30] for the original reference) feature detection. Although these tools have proven very effective for detection problems in general, ARUCO marker detection (see [31]) was selected here due to its robustness and ease of implementation despite requiring hardware constraints. Indeed, in an industrial environment, robustness is of paramount importance. Several options exist on the placement of the ARUCO markers and the camera. For example, the camera could be mounted on the end-effector of the robot and markers could be placed in known locations in the environment. In this work, we chose to place the camera in a set known location in the environment and one marker is placed on the end-effector of the robot. The detection of the marker is handled by a python script that uses the ARUCO library. The script creates packages of the location and orientation data and sends them to a port in the target node computer used to process the Simulink Realtime control scheme of the robot. A "UDP retrieve" block in simulink collects the data package in the port, unpacks it and uses it for control. The marker detection and the control process run on two different machines and are not synchronized. Therefore, the processing speed of one does not influence the processing speed of the other. One main challenge is to solve the mathematical problem giving the desired inputs for the command of the robot. The control scheme is shown in **Fig. 12**. The Cartesian planner processes the desired poses of the robot, the vision system measures the pose of the marker in the camera frame. Because the marker's location on the end-effector is known, the measurement of the vision system gives indirectly the pose of the end-effector.

The solution of the mathematical model requires the knowledge of the pose of the robot relative to the base because the pose of the base (which is mobile) must be determined. Therefore, instead of using another sensor, the pose of the robot is determined using the solution of direct the kinematic problem presented above, making use of the encoders. This information is taken from the output of the trajectory generation block. Indeed, in order to smoothen the movement of the robot, several points are generated between the current pose of the robot and the desired pose. **Fig. 13** illustrates the geometric modelling of the robot and camera. Three reference frames are used, namely: one attached to the camera referred to as R_C, one attached to the base referred to as R_B and one attached to the effector/platform referred to as R_P. The main points of interest are M_m, the current location of the marker on the platform, and M_o, the targeted location for the marker, which corresponds to the desired pose of the robot. We introduce $\mathbf{B}_{i/j}$ the rotation matrix from frame i to frame j. For instance, $\mathbf{B}_{B/P}$ is the rotation matrix from R_B to R_P. We have the following result:

$$[\mathbf{m}_o]_{R_B} = [\mathbf{o}]_{R_B} + \mathbf{B}_{\frac{C}{B}}[\mathbf{m}_o]_{R_C} \quad \dots\dots (47)$$

$$\mathbf{B}_{\frac{C}{B}} = \mathbf{B}_{\frac{C}{P}}\mathbf{B}_{\frac{P}{B}} \quad \dots\dots (48)$$

where $\mathbf{m}_o$ and $\mathbf{o}$ are respectively the position vectors of points M_o and O and where the subscripts of the brackets indicate in which frame the vectors are expressed. The kinematic closed loop then leads to

$$[\mathbf{o}]_{R_B} = [\mathbf{p}]_{R_B} + \mathbf{B}_{\frac{P}{B}}[\mathbf{m}_m]_{R_P} - \mathbf{B}_{\frac{C}{B}}[\mathbf{m}_m]_{R_C} \quad \dots \quad (49)$$

where $\mathbf{p}$ and $\mathbf{m}_m$ are the position vectors of points P and M_m respectively. Then, substituting Eq. (47) into Eq. (49) and using Eq. (48) yields the final expression for $[\mathbf{m}_o]_{R_B}$, namely

$$\begin{aligned}[\mathbf{m}_o]_{R_B} &= [\mathbf{p}]_{R_B} + \mathbf{B}_{\frac{P}{B}}[\mathbf{m}_m]_{R_P} \\ &\quad + \mathbf{B}_{\frac{C}{P}}\mathbf{B}_{\frac{P}{B}}\left([\mathbf{m}_o]_{R_C} - [\mathbf{m}_m]_{R_C}\right) \quad \dots\dots (50)\end{aligned}$$

where all quantities are known. The desired pose is sent to the IKP block that provides the corresponding angles for the motors. Then, the previous control method is applied for a closed-loop control of the angles of the motors with the encoders. The vision-guided control mode can be used to provide a very effective assistance to the human operator.

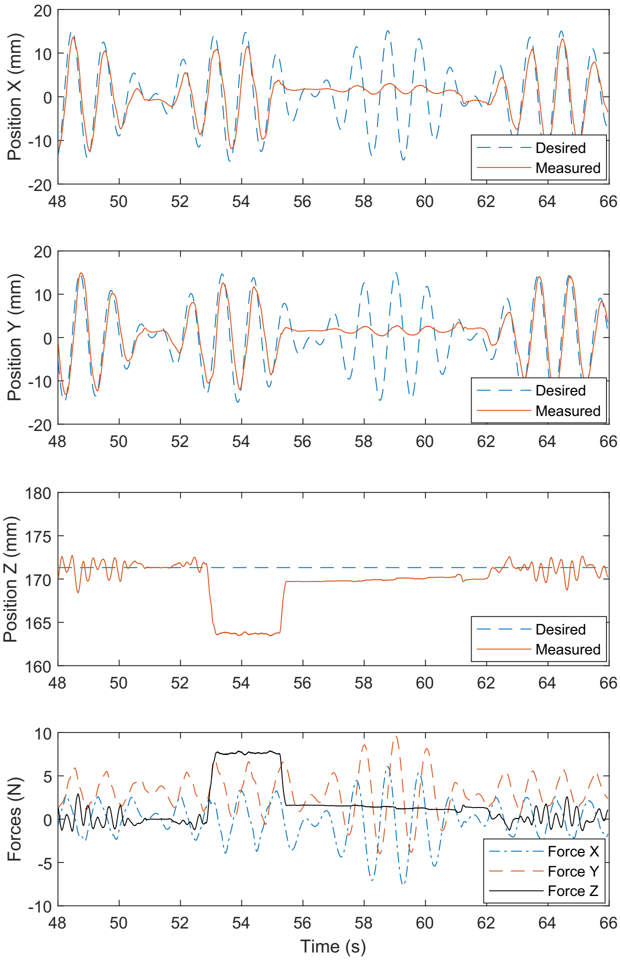


Fig. 14. Desired and measured positions and forces during an insertion task with a spiral search approach.

8. Experimental Results

In order to illustrate the hybrid position-force control, Cartesian position and computed forces at the end effector during a peg in a hole insertion with the help of a spiral trajectory are presented. The task is similar to what is shown in the video between 1:45 and 2:00. The spiral trajectory is in the X–Y plane, the insertion is along the Z axis and the vertical direction is along the Y axis. The control compliances are set to 500 N/m along the X and Y axes and to 1000 N/m along the Z axis, which are relatively low. The stiffness along Z is set to a larger value in order to obtain a more effective insertion. The damping is set to 10 Ns/m (low value) and there is no integral term in the controller (a PD controller is used). The results are shown in **Fig. 14**. Referring to **Fig. 14**, the sequence begins with a free spiral motion of the end-effector (time 48 s to 53 s). Because of the low stiffness, the desired trajectories are not perfectly followed by the robot, but the tracking is close enough for the search task to be effective. Then, the robot is pushed against the surface (time 53 s to 55.5 s). Due to the compliance, the end-effector moves backwards, which increases the force at the end-effector along the Z axis, but only by a limited amount because of the significant compliance. Also, because of the friction between the surface and the peg, the spiral trajectory is harder to follow, but nevertheless sufficiently performed for the search task to be effective. When the target is found, the peg is suddenly inserted in the hole and the end-effector moves forward (time 55.5 s). From (time 55.5 s to 61 s), the end-effector is almost maintained in position in the X and Y directions against the desired spiral trajectory. This increases the forces along the X–Y plane, but with a limited amount because of the significant compliance. Also, the robot is still slightly pushed against the wall, which causes a small and relatively steady force along the Z axis. When robot is moved away from the wall (time 61 s), the peg is extracted and the robot goes back to its free spiral motion. It is noted that the positive shift of force along the Y axis corresponds to the gravity compensation.

9. Multimedia Material

The implementation of the algorithms discussed above is presented in a video. The video can be found at: https:

//youtu.be/25f1Rli_Q5U. The first part of the video shows the position control discussed in Section 7.1 through pick&place type trajectories and a circular trajectory.

Then, the hybrid control is presented, manipulation of the end-effector with hands illustrates the backdrivable property of the robot. Moreover, pulling the end-effector in a direction produces a return motion followed by a back-and-forth around a neutral configuration, which illustrates the spring-like behaviour expected regarding the control scheme discussed in Section 7.2. The robot is mounted on a passive mechanism to simulate the remote aspect of the assembly task set in this context. The passive mechanism allows the three translations as well as one rotation close to the base of the robot, controlled with a pulley-driving belt that enables the human operator to control the rotation remotely. The passive mechanism also acts as a weight support, reducing greatly the weight that the human operator feels, at the cost of added inertia. A first peg-in-hole type insertion test follows to show that the compliance of the robot already helps the human operator in achieving the task. Afterwards, the target pose of the robot is changed from a fixed pose to a spiral trajectory with a constant orientation. The idea is to introduce a search strategy to the robot, similar to the work in [32], to ease the search of the hole. However, albeit the following test shows some kind of help, the context is here different than in [32]. Indeed, the base of the robot is not fixed, the robot does not work by itself but in collaboration with a human operator, making search trajectories inconsistent because if the human operator moves the base before the robot has finished searching, it is impossible to guarantee the work area is correctly explored. The search trajectory can even go against the intent of the human operator, making this strategy a potential detriment for him/her. Then, the different parameters of the model are changed to explore the flexibility of the robot as a configurable RCC over the fixed-by-design passive alternative. A model with the centre of rotation set 15 cm below the end-effector is shown first, then the centre of rotation is set 15 cm above the end-effector. To illustrate that the centre of rotation can be placed everywhere, it is placed in the plane of the effector, shifted on the side upwards. The behaviour of the robot in an insertion scenario is then compared with the centre of rotation below and above the end-effector. The fundamental benefit of RCC mechanisms is to set the centre of rotation further in front of the tool, to obtain an alignment change produced by the reaction force between the parts favourable for the insertion. The alignment change is generally unfavourable if the centre of rotation is below, despite being the easiest behaviour to design, and make orientation a challenge in assembly tasks. The close-up views in the video show this phenomenon: with a centre of rotation set below, the peg does not align correctly when touching the surface, whereas with a centre of rotation above, the peg aligns itself correctly. However, the design of this robot is not ideal for large orientation changes, this control strategy would work better with a different architecture.

The last part of the video deals with the vision control scheme discussed in Section 7.3. The robot attempts to reach a pose defined in the frame of the camera, which would be a pose required to perform an assembly task correctly. This target position is set in the centre of the camera at a 55 cm distance from the optical centre for the test. The workspace of the robot is fixed in a constant orientation and constrained in a 10 cm diameter sphere in the control to avoid undesirable configurations. The introduction of changes in orientation as well as optimization of the workspace would greatly increase the task of implementing this control scheme. One can notice that if the robot is pulled or pushed too far, the spherical constraint prevents it from reaching the target position. The robot follows a trajectory in the direction of the constantly changing target objective in the frame of the robot with a Cartesian velocity limitation. Two different velocity limitations are presented. The higher the speed limit, the faster the accuracy correction in theory. However, this parameter requires to reconsider the regulation parameters of the PID to keep the robot efficient. The last segment of the video emphasizes the planar correction performed to clearly see the behaviour of the robot in the camera feed.

10. Conclusion

In this article, a novel 6-dof backdrivable parallel robot designed to work in a collaborative assembly environments is presented. The kinematics of the robot are introduced, then a numerical algorithm is used to compute the geometrical workspace of the robot. The singularities of the robot are discussed, followed by an analysis of its force capabilities. A prototype is then presented that implements the characteristics previously discussed. Finally, three control schemes are implemented and demonstrated, each for different application purposes.

Acknowledgements

Funding for this project was provided by the Natural Sciences and Engineering Research Council of Canada (NSERC) and General Motors of Canada. The authors would also like to acknowledge the support of Advanced Motion Controls through the University Outreach program. The help of Simon Foucault in the design of the set-ups and mechanisms presented in this paper is also gratefully acknowledged.

References:

[1] J. Krüger, T. K. Lien, and A. Verl, "Cooperation of human and machines in assembly lines," CIRP annals, Vol.58, No.2, pp. 628-646, 2009.

[2] A. Campeau-Lecours, P.-L. Belzile, T. Laliberté, S. Foucault, B. Mayer-St-Onge, D. Gao, and C. Gosselin, "An articulated assistive robot for intuitive hands-on-payload manipulation," Robotics and Computer-Integrated Manufacturing, Vol.48, pp. 182-187, 2017.

[3] G. R. Luecke, K.-L. Tan, and S. Mahrt, "Control of lift assist devices for performance enhancement," Mobile Robots XV and Telemanipulator and Telepresence Technologies VII (Proc. of SPIE), Vol.4195, pp. 260-272, 2001.

[4] P. D. Labrecque, T. Laliberte, S. Foucault, M. E. Abdallah, and C. Gosselin, "uMan: A Low-Impedance Manipulator for

Human-Robot Cooperation Based on Underactuated Redundancy," IEEE/ASME Trans. on Mechatronics, 2017.

[5] S. M. Rahman, R. Ikeura, and H. Yu, "Novel biomimetic control of a power assist robot for horizontal transfer of objects," 2011 IEEE Int. Conf. on Robotics and Biomimetics (ROBIO), pp. 2181-2186, 2011.

[6] G. Liu, Z. Xi, and J.-M. Lien, "Nearly convex segmentation of polyhedra through convex ridge separation," Computer-Aided Design, Vol.78, pp. 137-146, 2016.

[7] A. H. Adiwahono, B. Saputra, T. W. Chang, and Z. X. Yong, "Autonomous stair identification, climbing, and descending for tracked robots," 2014 13th Int Conf. on Control Automation Robotics & Vision (ICARCV), pp. 48-53, 2014.

[8] J. Mure-Dubois and H. Hügli, "Embedded 3D vision system for automated micro-assembly," Two- and Three-Dimensional Methods for Inspection and Metrology IV (Proc. of SPIE), Vol.6382, pp. 1-11, 2008.

[9] R. Ahmad and P. Plapper, "Safe and Automated Assembly Process using Vision assisted Robot Manipulator," Procedia CIRP, Vol.41, pp. 771-776, 2016.

[10] G. Gogu, "Structural synthesis of parallel robots," Solid Mechanics and its Applications book series, Volume 930, Springer, 2008.

[11] K. Waldron, S.-L. Wang, and S. Bolin, "A study of the Jacobian matrix of serial manipulators," J. of Mechanical Design, Vol.107, No.2, pp. 230-237, 1985.

[12] K. Cleary and T. Arai, "A prototype parallel manipulator: Kinematics, construction, software, workspace results, and singularity analysis," Proc. of the 1991 IEEE Int. Conf. on Robotics and Automation, pp. 566-567, 1991.

[13] G. Liu, Y. Lou, and Z. Li, "Singularities of parallel manipulators: A geometric treatment," IEEE Trans. on Robotics and Automation, Vol.19, No.4, pp. 579-594, 2003.

[14] Z. Huang, L. Chen, and Y. Li, "The singularity principle and property of Stewart parallel manipulator," J. of Robotic Systems, Vol.20, No.4, pp. 163-176, 2003.

[15] X. Kong and C. M. Gosselin, "Kinematics and singularity analysis of a novel type of 3-CRR 3-DOF translational parallel manipulator," The Int. J. of Robotics Research, Vol.21, No.9, pp. 791-798, 2002.

[16] I. Ebrahimi, J. A. Carretero, and R. Boudreau, "3-PRRR redundant planar parallel manipulator: Inverse displacement, workspace and singularity analyses," Mechanism and Machine Theory, Vol.42, No.8, pp. 1007-1016, 2007.

[17] J. Gallardo-Alvarado, J. M. Rico-Martínez, and G. Alici, "Kinematics and singularity analyses of a 4-dof parallel manipulator using screw theory," Mechanism and Machine Theory, Vol.41, No.9, pp. 1048-1061, 2006.

[18] D. Pai and M.-C. Leu, "Genericity and singularities of robot manipulators," IEEE Trans. on Robotics and Automation, Vol.8, No.5, pp. 545-559, 1992.

[19] P. Ben-Horin and M. Shoham, "Application of Grassmann–Cayley algebra to geometrical interpretation of parallel robot singularities," The Int. J. of Robotics Research, Vol.28, No.1, pp. 127-141, 2009.

[20] Y. Zhao, "Singularity, isotropy, and velocity transmission evaluation of a three translational degrees-of-freedom parallel robot," Robotica, Vol.31, No.2, pp. 193-202, 2013.

[21] D. Kim, W. Chung, and Y. Youm, "Singularity analysis of 6-DOF manipulators with the analytical representation of the determinant," Proc. 1999 IEEE Int. Conf. on Robotics and Automation, Volume 2, pp. 889-894, 1999.

[22] J. K. Salisbury and J. J. Craig, "Articulated hands: Force control and kinematic issues," The Int. J. of Robotics Research, Vol.1, No.1, pp. 4-17, 1982.

[23] I. A. Bonev and J. Ryu, "Orientation workspace analysis of 6-DOF parallel manipulators," Proc. of the ASME 1999 Design Engineering Technical Conf., pp. 1-8, 1999.

[24] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, "Robotics: Modelling, planning and control," Springer Science & Business Media, 2010.

[25] D. E. Whitney, "Quasi-static assembly of compliantly supported rigid parts," J. of Dynamic Systems, Measurement, and Control, Vol.104, No.1, pp. 65-77, 1982.

[26] A. Q. Keemink, H. v. d. Kooij, and A. H. Stienen, "Admittance control for physical human–robot interaction," The Int. J. of Robotics Research, Vol.37, No.11, pp. 1421-1444, 2018.

[27] F. Zhang, "Quaternions and matrices of quaternions," Linear Algebra and its Applications, Vol.251, pp. 21-57, 1997.

[28] D. J. Evans, "On the representatation of orientation space," Molecular physics, Vol.34, No.2, pp. 317-325, 1977.

[29] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," Int. J. of Computer Vision, Vol.60, No.2, pp. 91-110, 2004.

[30] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," 2011 Int. Conf. on Computer Vision, pp. 2564-2571, 2011.

[31] J. Rekimoto, "Matrix: A realtime object identification and registration method for augmented reality," Proc. of the 3rd Asia Pacific Computer Human Interaction, pp. 63-68, 1998.

[32] T. Matsuno, T. Fukuda, and Y. Hasegawa, "Insertion of long peg into tandem shallow hole using search trajectory generation without force feedback," Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA'04), Volume 2, pp. 1123-1128, 2004.

Name:
Jérôme Landuré

Affiliation:
Laboratoire de Robotique, Department of Mechanical Engineering, Université Laval

Address:
1065 Avenue de la médecine, Québec, Qc G1V0A6, Canada

Brief Biographical History:

2015 Bachelor's degree in Mechanical Engineering from École Nationale Supérieure d'Arts et Métiers

2020 Ph.D. in Mechanical Engineering from Université Laval

Main Works:

- J. Landuré and C. Gosselin, "Kinematic Analysis of a Novel Kinematically Redundant Spherical Parallel Manipulator," J. of Mechanisms and Robotics, Vol.10, No.2, 021007, 2018.

Name:
Clément Gosselin

Affiliation:
Laboratoire de Robotique, Department of Mechanical Engineering, Université Laval

Address:
1065 Avenue de la médecine, Québec, Qc G1V0A6, Canada

Brief Biographical History:

1988 Ph.D. in Mechanical Engineering from McGill University

1989 Joined Université Laval

2001- Canada Research Chair in Robotics and Mechatronics

Main Works:

- K. Wen, T. S. Nguyen, D. Harton, T. Laliberté, and C. Gosselin, "A Backdrivable Kinematically Redundant $(6+3)$-Degree-of-Freedom Hybrid Parallel Robot for Intuitive Sensorless Physical Human–Robot Interaction," IEEE Trans. on Robotics, Vol.37, No.4, pp. 1222-1238, 2021.

Membership in Academic Societies:

- American Society of Mechanical Engineers (ASME), Fellow
- The Institute of Electrical and Electronics Engineers (IEEE), Fellow
- Royal Society of Canada, Fellow

Name:
Thierry Laliberté

Affiliation:
Laboratoire de Robotique, Department of Mechanical Engineering, Université Laval

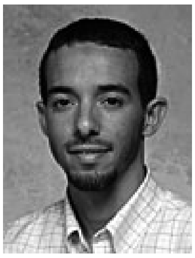
Address:
1065 Avenue de la médecine, Québec, Qc G1V0A6, Canada

Brief Biographical History:

1992 Bachelor's degree in Mechanical Engineering from Université Laval
1994 Master's degree in Mechanical Engineering from Université Laval
1994 Joined Université Laval as a Research Engineer

Main Works:

- L. Birglen, T. Laliberté, and C. Gosselin, "Underactuated Robotic Hands," Springer Tracts in Advanced Robotics book series, Volume 40, Springer, 2007.



Name:
Muhammad E. Abdallah

Affiliation:
Manufacturing Systems Research Lab., General Motors Research and Development

Address:
GM Technical Center, Warren, MI 48092, USA

Brief Biographical History:

2007 Ph.D. in Mechanical Engineering from Stanford University
2009 Joined General Motors Research and Development

Main Works:

- M. A. Diftler, J. S. Mehling, M. E. Abdallah et al., "Robonaut 2 – The first humanoid robot in space," 2011 IEEE Int. Conf. on Robotics and Automation, pp. 2178-2183, 2011.