# DESIGN, CONTROL, AND EXPERIMENTS OF A LOW-COST OPEN-SOURCE PLANAR CABLE-DRIVEN PARALLEL ROBOT

Mikael Arslan[1], Lionel Birglen[2]
[1]*Mechatronics Department, ENS Rennes, Rennes, France*
[2]*Department of Mechanical Engineering, Polytechnique Montréal, QC, Canada*
*Email: mikael.arslan@ens-rennes.fr; lionel.birglen@polymtl.ca*

## ABSTRACT

This paper presents the complete design of a low-cost planar cable-driven parallel robot (CDPR) actuated with four motors, from its mechanical structure to its control scheme as well as its electronic circuitry. This robot is modular in the sense that its actuation modules can be moved along the supporting frame and the robot itself can be configured in both a two and three degree-of-freedom mode. The objective of this work is to share the open design of the robot focusing on low-cost and instrumentation in order to be readily usable for education and research purposes and for the widest range of users as possible without compromising on performance. The general structure and mathematical modeling of the robot is first presented, including the inverse kinematics and computation of the wrench feasible workspace, as well as its control strategy and a novel cable winding model. Finally, actual experiments conducted with the robot are presented and discussed to demonstrate what can be achieved with the machine.

**Keywords:** planar CDPR; prototyping; inverse kinematics; wrench feasible workspace; control; open-source.

# CONCEPTION, CONTRÔLE, ET EXPÉRIMENTATION D'UN ROBOT PARALLÈLE PLAN À CABLES À BAS COÛT ET EN SOURCE LIBRE

## RÉSUMÉ

Cet article présente la conception complète d'un robot parallèle plan à câbles et entraîné par quatre actionneurs, qui possède la particularité d'avoir un bas coût et dont les sources sont ouvertes. Cette conception proposée va de la structure mécanique à son schéma de commande, en passant par son circuit électronique. Ce robot est aussi modulaire dans le sens où ses modules d'actionnement peuvent être glissés le long de sa structure et le robot lui-même peut avoir à deux ou trois degrés de liberté. L'objectif de ce travail est de partager une conception ouverte du robot en mettant l'accent sur le faible coût et l'instrumentation afin d'être facilement utilisable pour l'enseignement et la recherche par le plus grand nombre d'utilisateurs possible sans compromettre les performances. La structure générale et la modélisation mathématique du robot sont d'abord présentées, y compris la cinématique inverse et le calcul de l'espace de travail atteignable, ainsi que sa stratégie de contrôle et un nouveau modèle d'enroulement des câbles du robot. Finalement, des expériences réelles menées avec le robot sont présentées et discutées pour illustrer les possibilités du système.

**Mots-clés :** robot à câbles plan ; prototypage ; modèle géométrique inverse ; espace de travail atteignable ; contrôle.

## 1. INTRODUCTION

While typical robots use rigid links to manipulate an end-effector, cable driven parallel robots (CDPRs) connect their end-effectors to the ground, usually a rigid frame, with several cables whose lengths are controlled by actuators. The number of these actuators determine if the robot is under, fully or over-constrained. Generally, the more a CDPR is constrained, the stiffer it is. Yet, the downside of having more cables and actuators is that it also increases the control complexity and cost of the robot. CDPRs have demonstrated high payload capabilities with minimal moving parts. The low inertia of cables also allows higher accelerations of the end-effector to be achieved resulting in very fast travels. Due to these advantages over classical serial or parallel robots, CDPRs have a broad range of applications spanning from telescope manipulation [1] to pick and place tasks [2]. CDPRs also attracted the interest of the research community since at least the early 2000s [3–5]. Many subsequent works [6, 7] have followed focusing for instance on kinematic modeling and workspace analysis but challenges still exist to this day [8]. Amongst the main issues still being investigated one can find: accurate control of cable lengths in the presence of sag or elasticity, cable nonlinear behaviour modeling, management of interferences, workspace synthesis, and accurate pose determination. Despite these ongoing concerns, the versatility and performances offered by CDPRs led to many architectures and configurations proposed both by academia and the industry. Open-source and/or free software toolboxes have been released to ease the complexity of dealing with CDPRs, e.g. [9–11]. However, there are no equivalent for the hardware of CDPRs of such open-source projects. This paper proposes to fill that gap with a simple, general, and affordable planar CDPR design for which all sources are open and editable. While there are many similar CDPRs reported in the literature, e.g. [12–14], they usually rely on expensive parts, are dedicated to a single task, and/or not publicly available. The paper will first discuss the kinematic model of the robot, its workspace calculations as well as a novel cable winding technique and its model, all needed for trajectory planning. Then, the mechanical, electrical, control, and software design of the robot will be discussed, each in turn. Lastly, experiments using the robot will be shown and discussed.

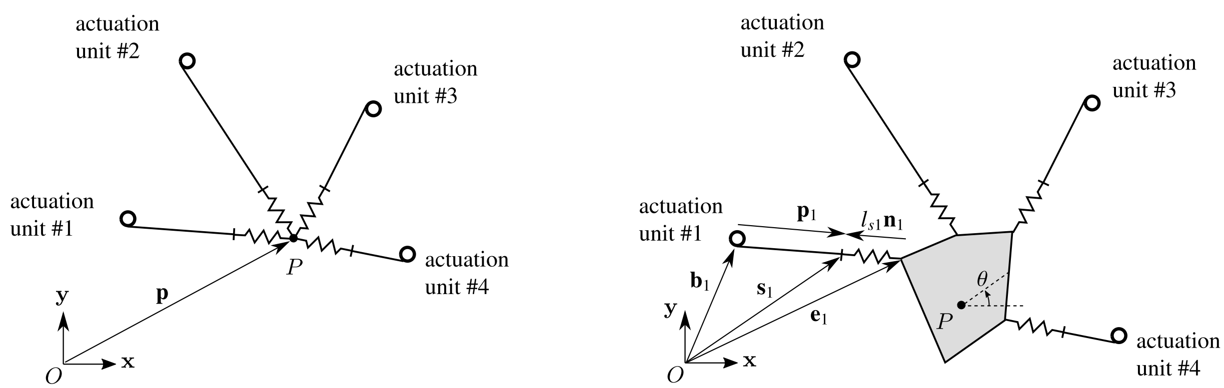



Fig. 1. Geometry of the two (left) and three (right) DOF configurations of the robot

## 2. MODELING

### 2.1. Inverse Kinematics

The general model of a planar 2 and 3 degree-of-freedom (DOF) CDPR is illustrated in Fig. 1. The inverse kinematics of this robot relates the end-effector/platform Cartesian coordinates to the corresponding cable lengths. These Cartesian coordinates can be written as a vector $\begin{bmatrix}\mathbf{p}^T & \theta\end{bmatrix}^T$ where $\mathbf{p}$ is position of a selected point $P$ of the end-effector, typically its geometric center, and $\theta$ is the angle of this end-effector relative to a fixed frame. If the mobile platform is of negligible dimensions, this latter angle can be neglected and the robot has then only 2 degrees of freedom. In the more general case, it has 3-DOF and vectors $\mathbf{e}_i$ $(i = 1, \ldots, 4)$ are used here to represent the attachment points of the cables from their respective actuation unit to the mobile platform. Four cables and actuation units are used in the robot shown in this paper since it is the minimal number required to fully constrain a 3-DOF end-effector due to the unidirectional nature of the forces that cables can transmit. Springs are inserted in the robot between one end of each cable and the mobile platform for safety, to simplify control, and also to model and study the impact of various cable stiffness on the behaviour of the robot. Geometric loop closure equations yield:

$$\mathbf{p}_i = \mathbf{s}_i - \mathbf{b}_i \tag{1}$$

where $\mathbf{p}_i$ is the vector going from the $i^{th}$ cable attachment point to the base frame at an attachment point defined by vector $\mathbf{b}_i$ and $\mathbf{s}_i$ is the vector from point $P$ to the spring attachment point to the cable. The latter equation can be developed into:

$$\mathbf{p}_i = \mathbf{e}_i + l_{si}\mathbf{n}_i - \mathbf{b}_i \tag{2}$$

with:

$$\mathbf{e}_i = \mathbf{p} + \mathbf{R}(\theta)\mathbf{e}'_i, \quad \mathbf{R}(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}, \quad l_{si} = l_{si0} + \frac{t_i}{k}, \quad \mathbf{n}_i = \frac{\mathbf{s}_i - \mathbf{b}_i}{||\mathbf{s}_i - \mathbf{b}_i||} \tag{3}$$

where $\mathbf{e}'_i$ is the coordinates of the attachment point of the spring on the mobile platform expressed in a frame attached to the latter. Matrix $\mathbf{R}(\theta)$ is the rotation matrix of the end-effector, $l_{si}$ is the spring length with a direction described by the unit vector $\mathbf{n}_i$ ($l_{si0}$ is the length of the spring when zero force is applied), $t_i$ is the cable tension and $k$ the spring stiffness. All springs are assumed identical. Cable lengths can then be computed as $l_i = ||\mathbf{p}_i||$ for a defined cable tension. In the subsequent part of the paper a 2-DOF version of the robot will be considered for illustration purposes, i.e. only the center of the end-effector will be controlled, but the machine is fully capable of handling the 3-DOF case with a varying orientation.

### 2.2. Cable Tension

In this work, it is assumed that the mobile platform of the robot is moving at modest velocities so inertias are neglected in favor of a quasi-static approach to estimate the cable tensions. The static equilibrium of the platform is then given by:

$$\sum_i t_i\mathbf{n}_i = \mathbf{W}\mathbf{t} = -\mathbf{w}_e, \quad \text{with} \quad \mathbf{W} = \begin{bmatrix} \mathbf{n}_1 & \mathbf{n}_2 & \mathbf{n}_3 & \mathbf{n}_4 \end{bmatrix} \tag{4}$$

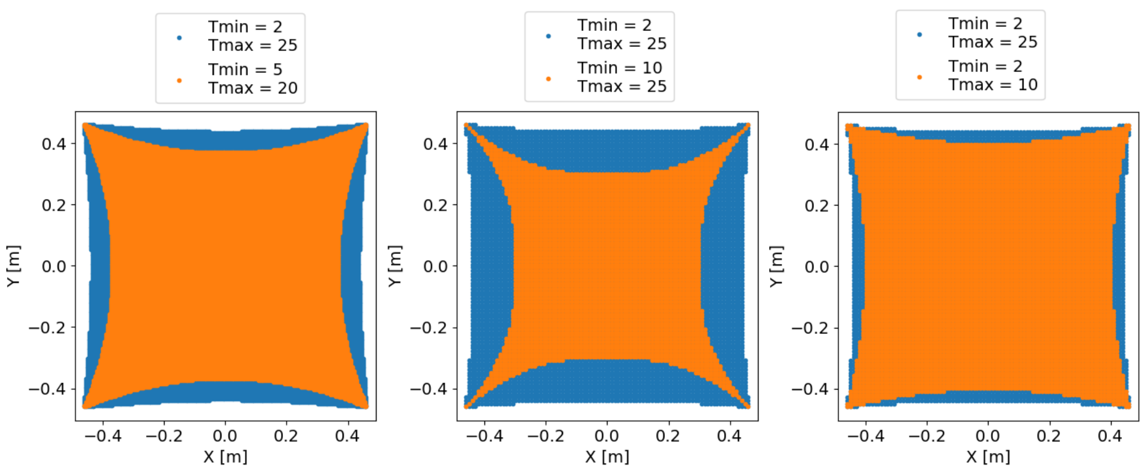
where $\mathbf{W}$ is the wrench matrix of the robot and $\mathbf{w}_e = \begin{bmatrix} \mathrm{f}x_{ext} & \mathrm{f}y_{ext} \end{bmatrix}$ is the external wrench vector. It becomes possible to determine, for a given pose, the cable tensions leading to static equilibrium. For instance for no applied external wrench, one can use:

$$\mathbf{t} = \min(\mathbf{t}^T\mathbf{W}^T\mathbf{W}\mathbf{t}) = \min(||\mathbf{W}\mathbf{t}||) \tag{5}$$

under the constraint that each component of $\mathbf{t}$ must stay between a set minimum and maximum values and assuming the weight of the end-effector to be negligible compared to cable tensions. This algorithm minimizing cable tensions can lead to small values of the latter so proper care must be taken to ensure that the minimal allowed tension if sufficient. Other algorithms exist to compute cable tensions, e.g. [15–18], although at the price of a more complex computation.

### 2.3. Workspace

The determination of the workspace of the robot is a critical step for the actual usage of the machine in order to ensure its safe operation. Two types of workspaces are commonly used when designing CDPR, namely the Wrench-Closure Workspace (WCW) and the Wrench-Feasible Workspace (WFW). The former is defined as the set of poses where the CDPR can create any desired wrench at its mobile platform by means of non-negative cable tensions [7]. The latter workspace (WFW) is an extension of the concept of WCW introduced as the set of mobile platform poses where a desired wrench set can be produced with cable tensions bounded by given (non-negative) lower and upper limits [19]. In the context of the cable robot discussed in this paper, the workspace of the robot is defined as the WFW for which the robot can balance the weight of the mobile platform, i.e. the set of poses where the end-effector can be moved without reaching excessive levels of cable tensions. Minimal tensions are also to be considered to avoid loss of control due to a cable becoming slack and is also useful to reduce cable sag. The springs added to the mobile platform are also in a sense detecting excessive cable tensions since if large forces are applied they will enter plastic deformation, which can be readily observed by inspection. Therefore, these springs act as a mechanical fuse for the robot avoiding damage to other more expensive parts such as the mobile platform or the pulleys. In this work and considering the selected springs, the maximal range of cable tension is between 2 N and 25 N and a range of $[5, 20]$ N is selected for the workspace available to the user. Assuming a square frame of 1 m for the robot and actuation pulleys at the corners of this frame, the corresponding workspace of a 2-DOF robot is shown in Figs. 2(a)-2(b) for different ranges of cable tensions. The effect of the weight of the mobile platform is negligible in these plots since the weight of the one used with the prototype, depicted in Fig. 5(a), is only 0.3 N.




(a) Selected workspace (orange) (b) Minimum cable tension variation (c) Maximum cable tension variation

Fig. 2. Workspace comparisons of the robot for different cable tension ranges

### 2.4. Cable Winding and Model

As mentioned earlier, a novel winding pulley is used in the robot in order to facilitate displacement of the actuation unit along the base frame and offer a compact design. While classical winding drums in CDPR or other cable machines (elevators, cranes, etc.) relies on a helix path of the cable around these drums, e.g. [13], a planar spiraling pattern has been chosen here, as illustrated in Fig. 3. A similar pattern was illustrated in [20] as a potential alternative to winding drums but not actually used in this work.

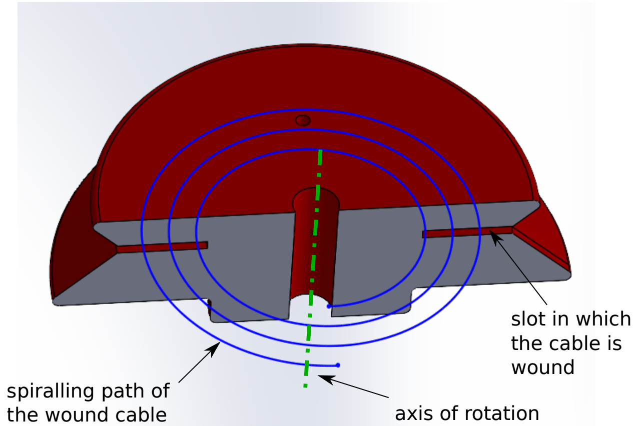



Fig. 3. Pulley design and winding pattern

This design choice results in a very compact solution but complicates the kinematic model of the robot since the cable is winding itself on what becomes a variable radius pulley. Assuming the variation of the radius to be linear, one has:

$$r(\alpha) = r_0 + \frac{D\alpha}{2\pi} \tag{6}$$

where $r(\alpha)$ is the instantaneous winding radius, $r_0$ is the minimal radius of the pulley, $D$ is the cable diameter, and $\alpha$ is the rotation angle of the pulley. The relationship between $\Delta l$, the magnitude of the cable length variation at the output of the actuation unit, and $\Delta\alpha$, the variation of the angle of the pulley, is:

$$\Delta l = \int_{\alpha_i}^{\alpha_i+\Delta\alpha} \left( r_0 + \frac{D\alpha}{2\pi} \right) d\alpha \tag{7}$$

which yields:

$$\Delta l = \left( r_0 + \frac{D\alpha_i}{2\pi} \right) \Delta\alpha + \frac{D}{4\pi}\Delta\alpha^2 \tag{8}$$

where $\alpha_i$ is the initial value of angle $\alpha$ (in the initial position of the mobile platform). To solve the previous equation, this initial pulley angle $\alpha_i$ is needed but it is difficult to measure accurately in practice. An easier way is to measure the initial radius $r_i$ of the pulley and deduce $\alpha_i$ from:

$$r_i = r_0 + \frac{D\alpha_i}{2\pi} \tag{9}$$

by substituting $\alpha_i$ in Eq. (8), it becomes:

$$\frac{D}{4\pi}\Delta\alpha^2 + r_i\Delta\alpha - \Delta l = 0 \tag{10}$$

and the actual solution corresponds to the positive root of the previous quadratic equation, namely:

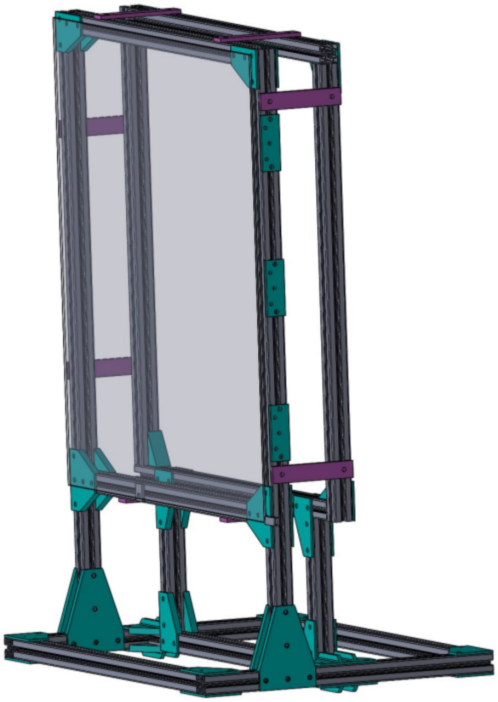
$$\Delta\alpha = \frac{2\pi}{D} \left( \sqrt{r_i^2 + \frac{D\Delta l}{\pi}} - r_i \right). \tag{11}$$

Please note that the initial position of the robot is assumed to correspond to the center of the frame. To move to a specific position, the difference of cable length to produce is thus calculated from this origin using Eq. (11) providing a target angle for each pulley and thus, each actuator, and considering the desired cable tension computed from Section 2.2.

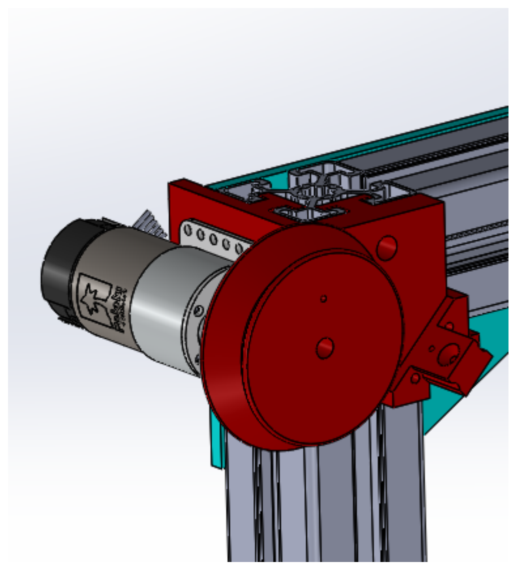
## 3. DESIGN OF THE ROBOT

### 3.1. Mechanical Design

The main structure of the proposed robot is composed of two frames made from Bosch-Rexroth 45 mm square aluminum extrusions, shown in Fig. 4(a), and assembled with 3D printed brackets (blue parts in Fig.4(a)). The two frames are offset using aluminum bars (pink parts in Fig. 4(a)) and are sandwiching the actuation units, cables, and mobile platform. The front and back faces of these frames are covered by impact-resistant polycarbonate sheets for safety, one of which is hinged to give access to the components. The 3D brackets were made from ABS with extruded guides and slots designed to fit easily into the aluminium extrusions. These two frames are fixed on top of another support structure, also made with aluminium extrusion connected by 3D printed parts, where the electric control cabinet is located and at the bottom of which four caster wheels were placed (not shown in CAD) to be able to move the robot with ease.



(a) Frames and support structure

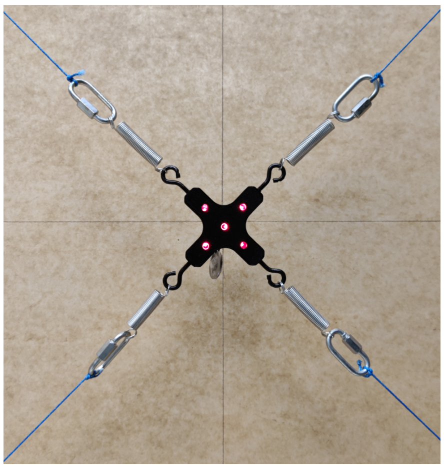


(b) Actuation unit: gearmotor and winding pulley assembly with cable redirector

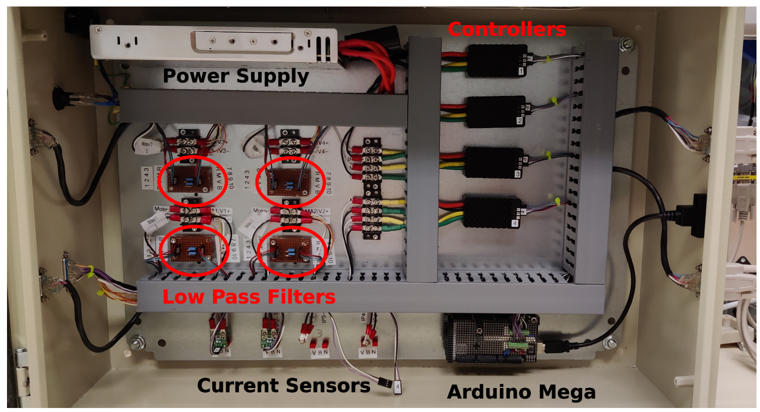
Fig. 4. CAD illustrations

The actuation unit is comprised of a 12 V DC gearmotor from Pololu with a reduction ratio 102.1:1 and a 64 count-per-turn magnetic encoder (using quadrature decoding). These gearmotors have a stall current of 5.5 A corresponding to a maximal torque of 34 kg·cm but have a maximal output power of 8 W only. The motors are housed on a 3D printed mounting bracket sliding into the aluminium extrusions and locked in place by a screw, and they each drive a winding pulley (cf. Section 2.4) out of which the cables are guided to a cable redirector (red parts in Fig. 4(b)). The forces exerted on the assembly are small enough to allow for screws to be directly threaded into the 3D printed part. The cable redirector allows for a fixed reference point of the cables making the calibration of the inverse kinematic model simpler. The pulley is composed of two halves assembled together with a gap of 1.5 mm that corresponds to the diameter of the cable used. These cables are made from ultra-high-molecular-weight polyethylene fibers which are known for their abrasion resistance, limited elongation under load, and very low cost.

The main end-effector proposed is here constituted by a small 3D printed cross-shaped part with 5 LEDs powered by an embedded CR2032 battery, see Fig. 5(a). These LEDs allow for easier video tracking of the end-effector during experiments to quantify the accuracy of the robot, see Section 4. Please note that the illustrated cable connection pattern is only valid for a 2-DOF version of the robot as it would result in a singular configuration if a 3-DOF end-effector is considered: all cables indeed intersect at a common point in the depicted pose and the mobile platform cannot oppose/produce any torque. The springs connecting this end-effector to the cables are 8.7×46.8 mm and made from steel spring with a measured stiffness of 264 N/m.



(a) End-Effector w/ springs and LEDs




(b) Inside view of the electrical cabinet

Fig. 5. Details of the robot

### 3.2. Electrical Design

The overall topology of the electrical circuit of the robot is shown in Fig. 6 and is built around an Arduino Mega 2560 Rev3 main board sending commands to four Roboclaw 30 A Solo motor controllers from BasicMicro which are handling the position control of each pulleys. The Arduino Mega and Solo drives are communicating through a TTL-level serial protocol in a line topology. Current sensor carriers, Pololu ACS711EX, are inserted at the outputs of these controllers to measure the currents fed to the motors and thus, estimate the torques they produce. The electrical power is provided to the robot by a 12 V power supply capable of delivering 30 A continuously and whose output is equipped with a bulk capacitance of 22,000 $\mu$F (25 V) added to absorb regenerative braking energy and eliminate voltage swings. All components and associated wiring are contained inside an 8x16x24 in electrical cabinet, see Fig. 5(b). Input and output wires of the cabinet go through standard DB-9 (encoder and motor power lines) and DB-15 (current and voltage signals) connectors. Due to the relatively long cables used to connect the cabinet to the actuation units and weak line drivers, encoder signals are unacceptably noisy (resulting in abnormal motions) but a simple passive RC filter with a cutoff frequency of 2 kHz on each data line, see Fig. 5(b), was sufficient to achieve accurate encoder counts even at higher velocities. A complete and detailed electrical circuit is available in the online repository of the project.

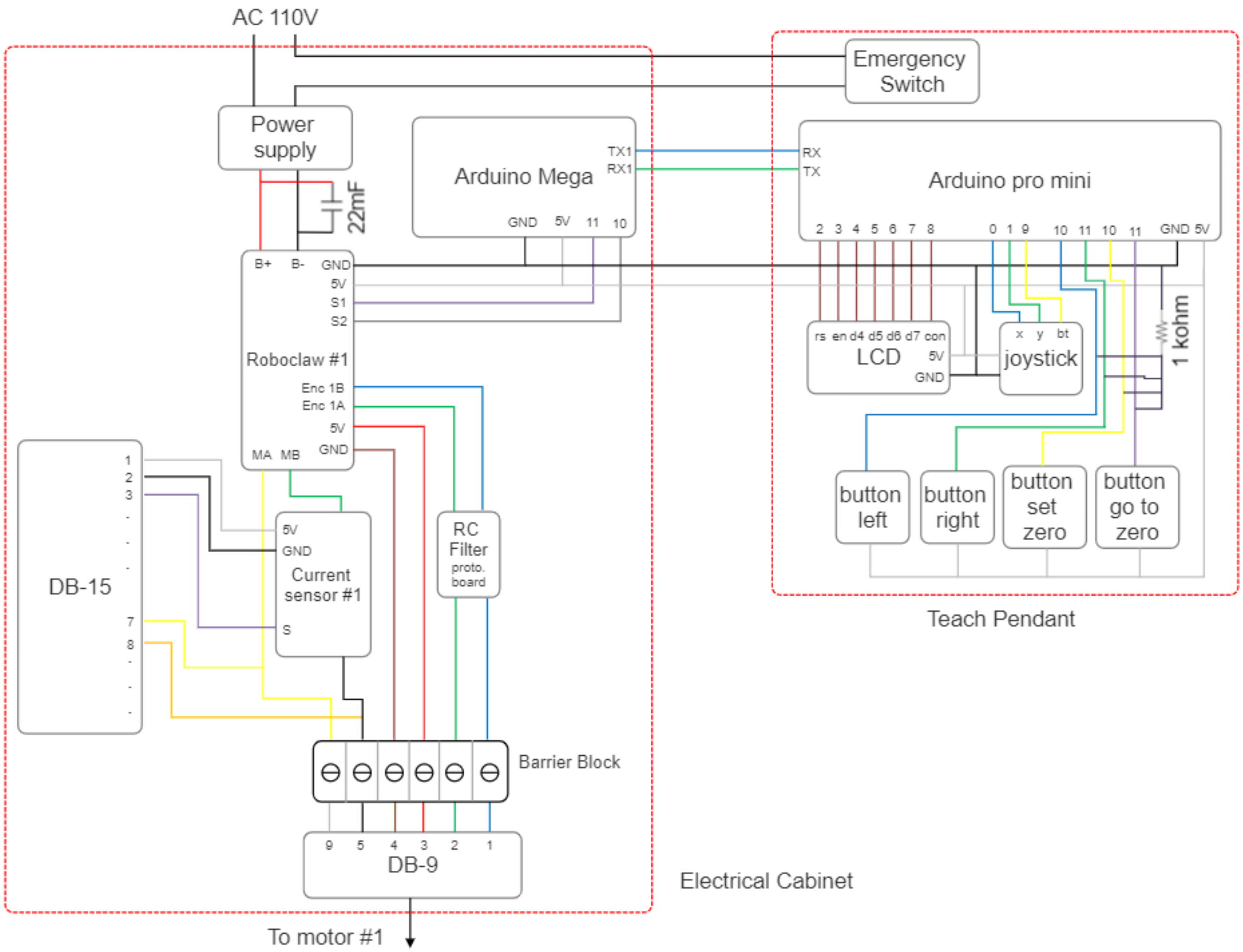



Fig. 6. General topology and pinout of the electrical circuit of the robot (one of four actuation units shown)

### 3.3. Control and Software Design

The Roboclaw controllers are provided with position and speed commands from the Arduino Mega which stores precomputed joint trajectories in its memory. These trajectories are defined as arrays of long integers corresponding to desired waypoints expressed in encoder counts. One array stores the list of all positions while the second one stores corresponding velocities. A Python script has been created to assist with the definition of trajectories with a graphical user interface allowing the user to indicate desired waypoints of the trajectory and travel speed for the end effector (see Fig. 7). Intermediate travel points are then computed by interpolating the user defined data. Using the inverse kinematic model, joint coordinates are subsequently calculated for each motor. Finally, velocities are obtained by computing the time derivative of position data assuming equal duration between each step of the trajectory. The graphical user interface also shows the trajectory estimated total duration as well as unreachable areas using workspace calculations. The low-level PID position control of each actuation unit is handled by the Roboclaw controller after the autotune of its parameters (a feature offered by the Roboclaw). The flowchart of the main program run by the Mega board is shown in Fig. 8, gtz is the acronym of "go to zero" while sz stands for "set zero".

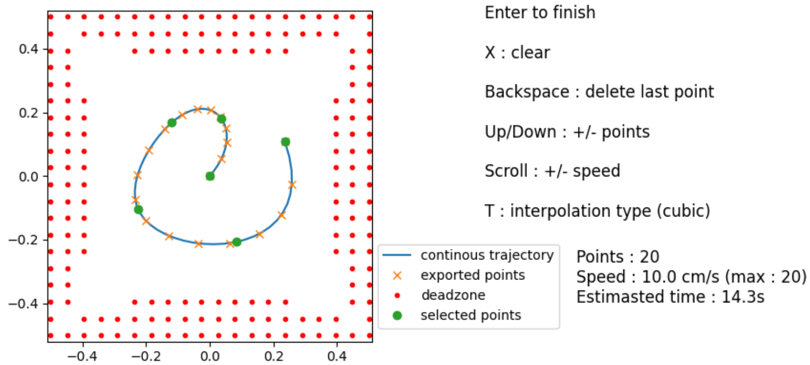



Fig. 7. Graphical user interface for trajectory planning

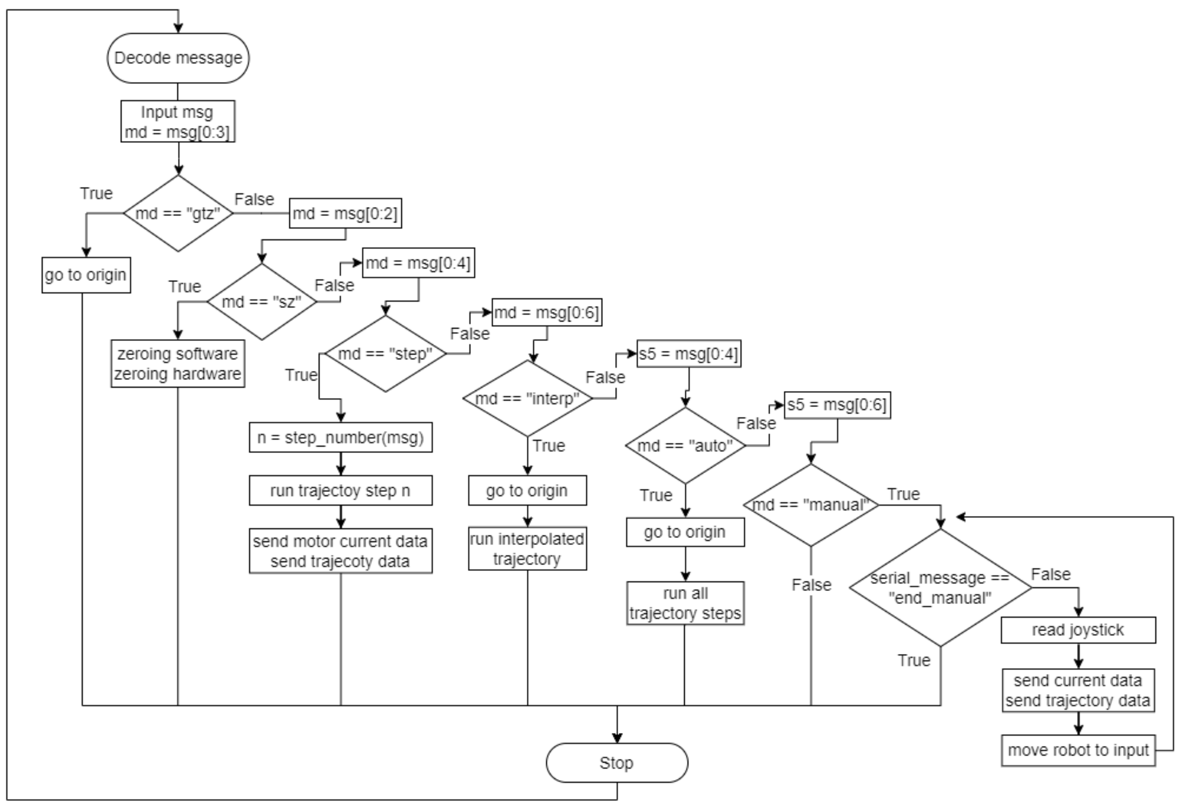
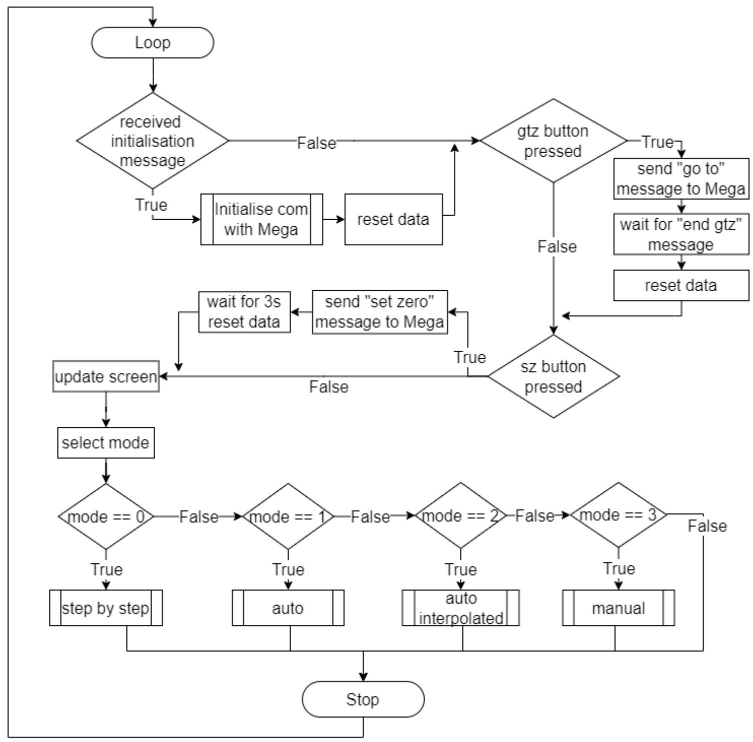



Fig. 8. Flowchart of the main program

### 3.4. Teach Pendant

A teach pendant was also designed, see Fig. 9(a). It is the main user interface of the robot and allows operation without a computer connected to the machine. This pendant has a four-line LCD screen, a joystick, and several buttons: one for setting the origin at the current position, one to go to the origin, and two to control the orientation of the end effector. The joystick is used to both navigate between menus on the LCD screen and also control the displacement of the end-effector when it is in manual mode. Information about current trajectory step, motor positions and currents are displayed on the screen when in operation. All these interface components are connected to an Arduino Pro mini embedded inside the teach pendant. This Pro mini is communicating with the Arduino Mega inside the electrical cabinet through a second UART serial connection. An emergency stop button shutting down the power directly at the 30 A power supply from the teach pendant was also added for safety. The Arduino Pro mini can send various commands to the Arduino Mega for the latter to handle, e.g.: moving the robot to the origin (reset position); execute one step of a previously stored trajectory (step-by-step mode); move the end-effector an increment from its current position (joystick/manual control). The flowchart of the program run by the Pro mini board is shown in Fig. 9(b)

(a) Picture




(b) Flowchart

Fig. 9. Teach pendant design and programming

The complete robot including all its elements is shown in Fig. 10. The total cost of the required parts adds up to approximately $2,300 USD. A BOM (bill of materials) listing all the required purchases to build the robot with their costs is shared along all design files at:

https://github.com/LionelBirglen/OpenSourcePlanarCableRobot

The most expensive items for building the robot are: the aluminium extrusions, the motors with their controllers, and the polycarbonate sheets. Only these four parts have a cost greater than $200 USD.

## 4. EXPERIMENTS

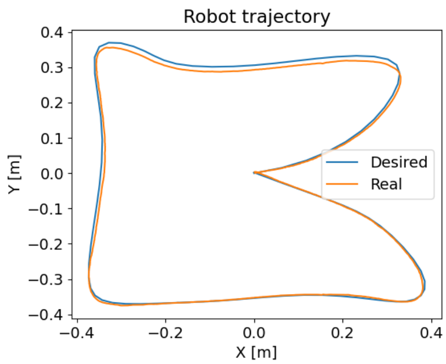
To measure the positional accuracy of the robot in Cartesian space during experiments, video tracking was used. A digital camera with video recording capability is placed in front of the robot to capture the entirety of its workspace. The 5 LEDs of the end-effector allow for easier tracking and detection of video distortion if any occurs (the distance between the LEDs has to stay constant). Tracker (`https://physlets.org/tracker/`), a free and powerful video tracking software is then used to post process the video, detect the platform through its LEDs, and export positions of selected points during motion. The automatic object

Fig. 10. Complete robot

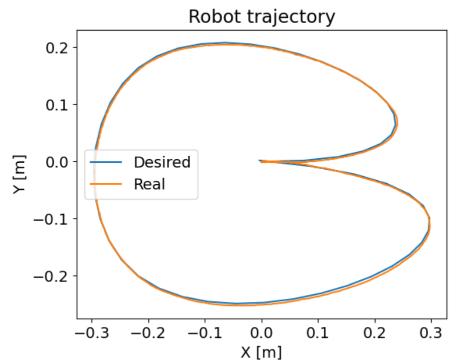
tracking feature of this software allows for a very fast and convenient data processing. After calibration, the exported data corresponding to the Cartesian coordinates of the robot can be compared to the target trajectory, see Figs. 11(a) and 11(b) for two examples of such a comparison. As can be seen from these plots, the accuracy of the robot is excellent. To quantify the performance of the robot the Root-Mean-Square Error (RMSE) between the experimental and target trajectories along the **x** and **y** axes, see Figs 11(c)-11(f), can be computed. This error, listed in Tab. 1, represents the end-effector average deviation from the desired trajectory along their respective axis with respect to time. As can be seen from Tab. 1, this RMSE error is actually larger than one could expect by looking at Figs. 11(a) and 11(b). This is because most of the errors are due to a (relatively) constant time lag in the trajectory. If this delay is of little importance other performance metrics should be used. For instance, another measure of the accuracy of the robot proposed here is to compute Pearson's correlation coefficient (PCC) between the target and measured trajectories. As shown in Tab. 1, both trajectories are indeed highly correlated which is more representative of the closeness of the theoretical and experimental trajectories. Yet another performance measure that is, this time, completely independent of temporal aspects is to compute the maximal distance between the two trajectories, namely:

$$\max_{\forall t}\min_{\forall t'}(||\mathbf{p}(t)_{th}-\mathbf{p}(t')_{ex}||) \tag{12}$$

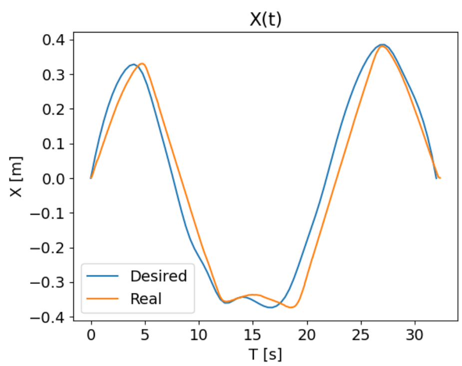
where $\mathbf{p}(t)_{th}$ is the theoretical position of the end-effector at time $t$ and $\mathbf{p}(t')_{ex}$ is the experimental position of the end-effector at time $t'$. Essentially, this metric find the closest experimental point to a defined theoretical position, compute the distance between these two positions, and output the maximal value of this distance over the complete trajectory. Applying this metric to the two example trajectories yields the results shown in Tab. 1 again. As one can see this maximal distance is small in both cases highlighting the closeness of the experiments to the theoretical trajectories. Additionally, the metric proposed in Eq. (12) quantify the worst-case situation. If instead one looks at the average values of $\min_{\forall t'}(||\mathbf{p}(t)_{th} - \mathbf{p}(t')_{ex}||$, values of 5.7 mm and 2.5 mm are obtained for Trajectories #1 and #2 respectively. The latter values better reflect what the user should expect when using the robot and again illustrate that it follows its prescribed trajectory with an excellent accuracy for a cable-driven machine.



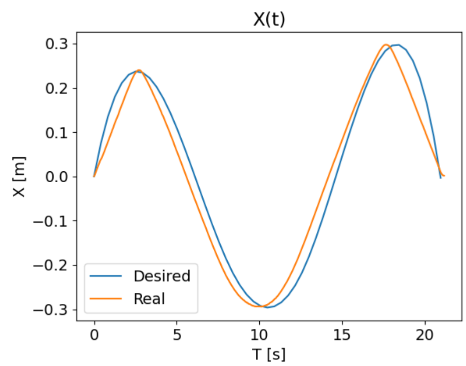

(a) Trajectory #1



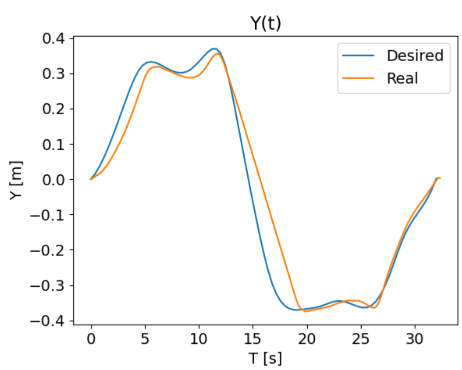

(b) Trajectory #2



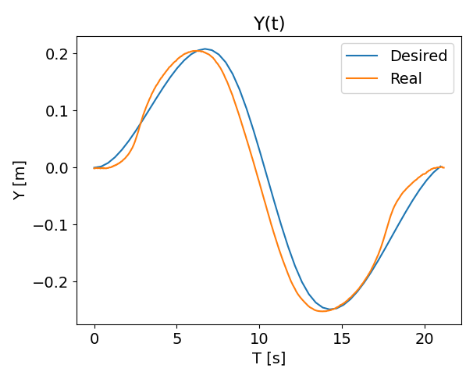

(c) X(t) Trajectory #1




(d) X(t) Trajectory #2




(e) Y(t) Trajectory #1




(f) Y(t) Trajectory #2

Fig. 11. Experimental data

| | RMSE (cm) | | PCC | | maxmin (cm) |
|---|---|---|---|---|---|
| trajectory | x | y | x | y | - |
| #1 | 5.9 | 6.2 | 0.977 | 0.978 | 1.57 |
| #2 | 3.9 | 2.6 | 0.983 | 0.987 | 0.47 |

Table 1. RMSE, PCC, and maxmin values for the two test trajectories

Motor currents during motions were also acquired through a low-cost VINT Hub Phidget (a simplistic data acquisition card) at a frequency of 1kHz per line. These measurements in addition to the torque estimation they provide could also be used to make sure that the motors are used in their allowable range of current. For instance, Pololu recommends not using them at more than 25% of the stall current for any extended period of time. An example of current measurement is shown in Fig. 12. As can be seen in this figure, while the data is indeed available, low-pass filtering of the signals is highly desirable due to the large amount of noise present.

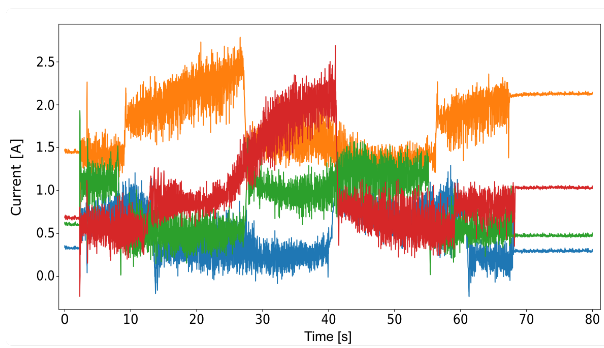
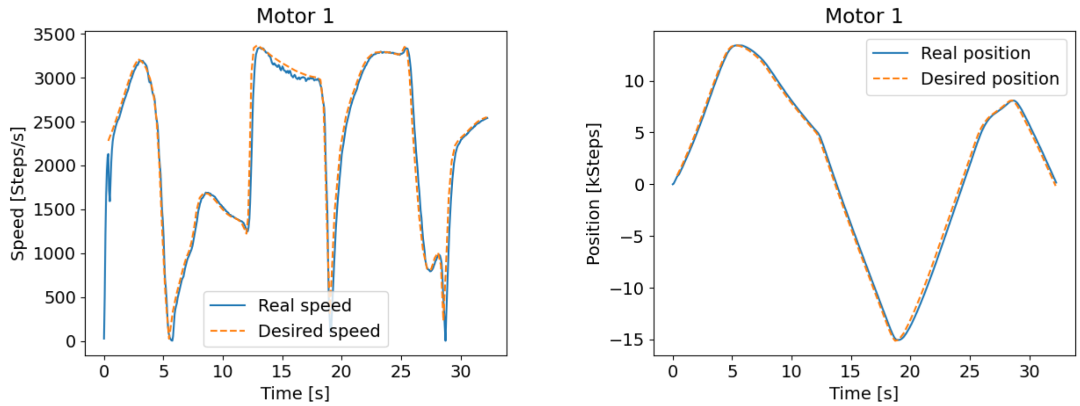



Fig. 12. Example of current measurements during a sample trajectory

Joint coordinates are also accessible and easier to obtain than Cartesian coordinates. The Arduino Mega is getting the angular position and velocity of each motor from their respective controller in real-time and can thus export them through its serial connection to a computer (if present). Doing so provides the user with the joint coordinate trajectories of the robot which can then also be superposed with the desired trajectories, see Figs. 13(a) and 13(b) for an example.

### 4.1. Discussion

Tab. 1 shows a noticeable difference between Trajectories #1 and #2. Trajectory #1 RMSE and min-max values are indeed greater than trajectory #2. This points to the fact that the closer to the center of the workspace the trajectory is, the more accurate the robot is. The suspected largest influence for this loss of accuracy at greater distance is the cable winding model which assumes perfect winding. In practice when the cable is wound/unwound under tension, its diameter slightly changes. This might cause an accumulation of errors over a large distance of travel potentially explaining the difference between trajectories, see also [14] for a similarly observed issue. The previously mentioned phase shifts of the X- and Y-axis trajectories are also an obvious issue. As can be seen from the plots the actual Cartesian trajectory seems sometimes to be leading the desired one which seems impossible. However, it should be reminded that a delay in joint trajectories could result in an apparent opposite effect in Cartesian space due to the nonlinear relationship between the two workspaces. As Fig. 13(a) also shows the motor velocity can be out of sync of up to 0.2 s




(a) Example of the joint velocity data of motor #1 (b) Example of the joint position data of motor #1

during high acceleration phases and this is due to the position controller not being able to keep up with the command.

Another issue coming from the trajectory generation is that the initial joint velocities of the trajectories are not set to zero (as they should), the robot has therefore to catch up to the desired velocity command at startup as clearly visible in the beginning of Fig. 13(a). Then, as all the motors pull on the end effector simultaneously, an overshoot appears before eventually reaching the desired velocity. In that same figure a part of the trajectory shows an unexpected disturbance. It corresponds to the trajectory #2 lower straight line when the motor #1 unwinds. During the experiments, a little bit of visible slack appeared in the cable connected to this motor leading to an undesired end effector oscillation. This is due to the interpolation algorithm between waypoints not considering cable tension. Additionally, the latter is also not considered in the low-level control of the actuator which is solely based on position information from the encoder of the motor.

## 5. CONCLUSION

This goal of this paper is to present and share the design of a low-cost planar CDPR aimed at research and education purposes. First, its inverse kinematics and workspace determination were presented as well as a model for the novel cable winding proposed. Then, all aspects of the design of the robot were discussed. Finally, experiments were conducted with the robot demonstrating that the robot despite its simplicity and affordable price can achieve a very high level of performance in terms of accuracy. Future work will deal with improved winding pulley, cable routing and resulting stiffness (as discussed in [21–23]), as well as improving control performances. Many design revisions are also under consideration based on experimental results. All design files of the robot are publicly available at: https://github.com/LionelBirglen/OpenSourcePlanarCableRobot

## REFERENCES

1. R. Nan, D. Li, C. Jin, Q. Wang, L. Zhu, W. Zhu, H. Zhang, Y. Yue, and L. Qian, “The five-hundred-meter aperture spherical radio telescope (fast) project,” *International Journal of Modern Physics D*, vol. 20, no. 06, pp. 989–1024, 2011.
2. L. Barbazza, F. Oscari, S. Minto, and G. Rosati, “Trajectory planning of a suspended cable driven parallel robot with reconfigurable end effector,” *Robotics and Computer-Integrated Manufacturing*, vol. 48, pp. 1–11, 2017.

3. R. L. Williams Ii and P. Gallina, “Translational planar cable-direct-driven robots,” *Journal of Intelligent and Robotic systems*, vol. 37, pp. 69–96, 2003.
4. R. L. Williams and P. Gallina, “Planar cable-direct-driven robots: Part i—kinematics and statics,” in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 80234, pp. 1233–1240, American Society of Mechanical Engineers, 2001.
5. P. Gallina, A. Rossi, and R. L. Williams, “Planar cable-direct-driven robots: Part ii—dynamics and control,” in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 80234, pp. 1241–1247, American Society of Mechanical Engineers, 2001.
6. *Kinematic Analysis and Design of Planar Parallel Mechanisms Actuated With Cables*, vol. Volume 7A: 26th Biennial Mechanisms and Robotics Conference of *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 09 2000.
7. M. Gouttefarde and C. Gosselin, “Analysis of the wrench-closure workspace of planar parallel cable-driven mechanisms,” *IEEE Transactions on Robotics*, vol. 22, no. 3, pp. 434–445, 2006.
8. C. Gosselin, “Cable-driven parallel mechanisms: state of the art and perspectives,” *Mechanical Engineering Reviews*, vol. 1, no. 1, pp. DSM0004–DSM0004, 2014.
9. A. Pott, “Wirex–an open source initiative scientific software for analysis and design of cable-driven parallel robots,” in *Proceedings of Fourth International Conference on Cable-driven Parallel Robots. Krakow*, 2019.
10. A. L. C. Ruiz, S. Caro, P. Cardou, and F. Guay, “Arachnis: Analysis of robots actuated by cables with handy and neat interface software,” in *Cable-Driven Parallel Robots: Proceedings of the Second International Conference on Cable-Driven Parallel Robots*, pp. 293–305, Springer, 2015.
11. D. Lau, J. Eden, Y. Tan, and D. Oetomo, “CASPR: A comprehensive cable-robot analysis and simulation platform for the research of cable-driven parallel robots,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2016.
12. M.-C. Kim, E.-S. Kim, J.-O. Park, E. Choi, and C.-S. Kim, “Robotic localization based on planar cable robot and hall sensor array applied to magnetic capsule endoscope,” *Sensors*, vol. 20, no. 20, 2020.
13. V. Mattioni, E. Ida’, and M. Carricato, “Design of a planar cable-driven parallel robot for non-contact tasks,” *Applied Sciences*, vol. 11, no. 20, 2021.
14. F. Ennaiem, A. Chaker, M. A. Laribi, J. Sandoval, S. Bennour, A. Mlika, L. Romdhane, and S. Zeghloul, “Task-based design approach: Development of a planar cable-driven parallel robot for upper limb rehabilitation,” *Applied Sciences*, vol. 11, no. 12, 2021.
15. D. Song, L. Zhang, and F. Xue, “Configuration optimization and a tension distribution algorithm for cable-driven parallel robots,” *IEEE Access*, vol. 6, pp. 33928–33940, 2018.
16. A. F. Côté, P. Cardou, and C. Gosselin, “A tension distribution algorithm for cable-driven parallel robots operating beyond their wrench-feasible workspace,” in *2016 16th International Conference on Control, Automation and Systems (ICCAS)*, pp. 68–73, 2016.
17. E. Picard, S. Caro, F. Plestan, and F. Claveau, “Stiffness oriented tension distribution algorithm for cable-driven parallel robots,” in *Advances in Robot Kinematics 2020*, pp. 209–217, Springer, 2021.
18. M. Gouttefarde, J. Lamaury, C. Reichert, and T. Bruckmann, “A versatile tension distribution algorithm for $n$-dof parallel robots driven by $n+2$ cables,” *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1444–1457, 2015.
19. P. Bosscher, A. T. Riechel, and I. Ebert-Uphoff, “Wrench-feasible workspace generation for cable-driven robots,” *IEEE Transactions on Robotics*, vol. 22, pp. 890–902, Oct 2006.
20. C. B. Pham, G. Yang, and S. H. Yeo, “Dynamic analysis of cable-driven parallel mechanisms,” in *Proceedings, 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics.*, pp. 612–617, IEEE, 2005.
21. H. Khakpour, L. Birglen, and S. Tahan, “Synthesis of differentially driven planar cable parallel manipulators,” *IEEE Transactions on Robotics*, vol. 30, pp. 619–630, June 2014.
22. H. Khakpour, L. Birglen, and S.-A. Tahan, “Analysis and optimization of a new differentially-driven cable parallel robot,” *Journal of Mechanisms and Robotics*, vol. 7, pp. 034503–034503–6, Aug 2015.
23. L. Birglen and M. Gouttefarde, “Stiffness of planar 2-dof 3-differential cable-driven parallel robots,” in *Cable-Driven Parallel Robots* (A. Pott and T. Bruckmann, eds.), (Cham), pp. 61–71, Springer International Publishing, 2019.