Robotics and Autonomous Systems 183 (2025) 104855

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

A sensorless approach for cable failure detection and identification in cable-driven parallel robots

Giovanni Boschetti [a,b], Riccardo Minto [a,*]

[a] Department of Industrial Engineering, University of Padova, Via Venezia 1, Padova, 35131, Italy
[b] Department of Information Engineering, University of Padova, Via Gradenigo 6/b, Padova, 35131, Italy

ARTICLE INFO

Keywords:
Cable robots
Recovery strategy
Digital twin

ABSTRACT

Cable-driven parallel robots (CDPRs) are a particular class of parallel robots that provide several advantages that may well be received in the industrial field. However, the risk of damage due to cable failure is not negligible, thus procedures that move the end-effector to a safe pose after failure are required. This work aims to provide a sensorless failure detection and identification strategy to properly recognize the cable failure event without adding additional devices. This approach is paired with an end-effector recovery strategy to move the end-effector towards a safe position, thus providing for a complete cable failure recovery strategy, which detects the failure event and controls the end-effector accordingly. The proposed strategy is tested on a cable-driven suspended parallel robot prototype composed of industrial-grade components. The experimental results show the feasibility of the proposed approach.

1. Introduction

Cable-driven parallel robots (CDPRs), commonly called cable robots, belong to a particular class of parallel robots which provide several advantages that may be well received in the industrial field. First, the parallel structure places most of the mass in the robot base, allowing for high payload capacity and high speed of the end-effector throughout its entire workspace [1]. Moreover, they improve rigid link parallel robots, whose limited workspace, due to possible collisions between links and the mechanical limits of their joints (both active and passive ones), is one of their shortcomings [2]. In fact, by replacing links with cables wound around winches, it is possible to achieve large workspace manipulation tasks while maintaining a simple and inexpensive design. Given their advantages, it is reasonable to adopt them in a wide variety of fields, ranging from industrial applications [3], rehabilitation fields [4], to entertainment applications [5].

Although the spread of CDPRs in the industrial field is still limited despite their advantages, they can become increasingly common with the digitalization of industrial plants required by the Industry 4.0 concept, along with other technologies such as collaborative robots or Digital Twin [6]. The concept of the digital twin is based on the definition of a model to realize a virtual representation of a physical system that is updated by exchanging information with the physical system [7]. Previous work in industrial robotics has implemented the Digital Twin concept with different goals. Zheng et al. [8] focused on the lack of frameworks for robotic applications, as the digital twin approach was mainly applied to product design and maintenance. After providing guidelines for an effective digital twin to improve the process design phase (optimizing tolerances, location positions, and clamping strategies) and the product manufacturing phase (increasing flexibility and competitiveness), the authors provided a case study application with a digital twin of a welding production line. Their digital twin, consisting of a geometric, kinematic, and mechanical model of an industrial robot, allows monitoring the process and improving the product quality. Kuts et al. [9] realized a digital copy of a real manufacturing system that includes a Motoman GP8 industrial robot in a virtual reality environment. The authors proposed a digital twin that mimics the movement of the real robot, since the motion commands are sent to both the real controller and a digital copy of it. The digital twin allows for improving the awareness of the robot in its workspace, thus reducing the risk of collision with the environment.

Furthermore, a digital representation of the system can be realized with closed-loop state observers, also called estimators. In fact, an observer reconstructs the missing state variable of a system using a system model and measurements of both the input and output of the system [10]. Boschetti et al. [11] adopted an Extended Kalman Filter to estimate the state variables of a spatial cable-suspended parallel robot (CSPR), i.e., with all exit points above the end-effector. In particular, two different formulations have been adopted based on the Udwadia–Kalaba formulation and on the Penalty one, respectively,

[*] Corresponding author.
E-mail address: riccardo.minto@unipd.it (R. Minto).

https://doi.org/10.1016/j.robot.2024.104855
Received 10 January 2024; Received in revised form 29 July 2024; Accepted 31 October 2024
Available online 9 November 2024

showing comparable performance and real-time capabilities. Le Nguyen and Caverly [12] introduced two different Extended Kalman Filter approaches to estimate the robot pose using payload data from an IMU installed in the end-effector. Indeed, using only the IMU data for pose estimation has limited accuracy. The two approaches were tested in simulations considering an overconstrained 6-DOF CDPR, simulating the noise of the IMUs with zero-mean Gaussian white noise.

Although digital twins have been defined for the control and optimization of CDPRs in a variety of fields, e.g., masonry construction [13], according to the authors' knowledge, they have hardly been adopted for cable failure detection. In fact, there is very limited literature on cable failure detection. Cable failure is the biggest disadvantage when working with CDPRs [14], as it is not possible to guarantee the position of the end-effector. Moreover, an emergency stop procedure for these devices is not as simple to implement as in rigid link robots, due to the nature of cables. Indeed, wires resist tension but not compression, i.e., each cable can pull but not push the end-effector, which arises several challenges when controlling cable robots [15]. Therefore, a recent focus in the literature has been the definition of appropriate recovery strategies for the end-effector in the event of cable failure, since abrupt stopping of the actuators does not imply that the end-effector rests in a stable position; in contrast, it could generate uncontrollable motions [16]. A first solution could be to double each cable with a passive auxiliary cable as a safety solution, or at least add a single auxiliary cable. However, increasing the number of cables may not be the optimal approach since it increases the complexity of the system, increases costs, and leads to possible obstruction of the workspace or interference between the cables. Therefore, a possible solution could be to control the remaining healthy cables to recover a lost wrench, since the end-effector applies a wrench different from the expected one in case of cable failure. Moreover, in case the robot does not remain overconstrained after failure, it is necessary to take into account changes in the static equilibrium workspace (SEW), or that the end-effector could lay outside it. Indeed, the SEW (i.e., a set of end-effector poses, for a given motor mount configuration, where static equilibrium can be obtained while applying tension to all cables, assuming infinite maximum cable lengths and tensions [17]) changes for underconstrained parallel robots after failure, as seen in [18], where a feasible motion leads the end-effector toward a Safe Pose (SP) to achieve static equilibrium inside the modified SEW. Alternatively, Berti et al. [19] presented a recovery strategy based on the definition of a piecewise trajectory, composed of an elliptical path, used to drive the platform inside the after-failure SEW, followed by two seventh-degree polynomial trajectories to stop the robot. Instead of defining a recovery trajectory, Boumann and Bruckmann [20] proposed a cable failure recovery strategy that focuses on minimizing kinetic energy, which was implemented and tested within a multibody simulation environment.

However, cable failure detection is not considered in these works, supposing instead that the information is provided by some sensors. Caro et al. [21] state that measuring cable length is not sufficient to detect cable failure and suggest multiple sensor possibilities, such as optical sensors or IMUs, to measure load altitudes or cable angles with respect to the horizontal plane. Cable tensions were not considered as it may be difficult to distinguish a broken cable from a slack one. On the other hand, Salah et al. [22], when evaluating the failure modes for a CDPR used in an automatic storage environment, solved the cable failure detection problem by properly designing the cables. The cables are made of aramid cables mixed with carbon cables; the latter are arranged in order to snap first, causing the electric current that passes through them to stop, thus allowing early cable failure detection. Raman et al. [23] proposed an on-line sensorless failure detection framework for a planar CDPR. In particular, they adopted an Interactive Multi Model Adaptive Estimation Filter for estimating the state variables (pose of the end-effector) and detecting cable failure as it is suitable to detect model changes. In contrast to this work, Raman et al. adopted a bank of Extended Kalman Filters to estimate the pose of the end-effector, and more importantly, the probability of each motion model associated with different failure scenarios. Even more recently, Boumann et al. [24] presented a failure detection algorithm based on three measurements of the cables' state, i.e., cable force, motor angle, and motor velocity. Indeed, if the cable tension is near zero, the cable is too short, and the motor is winding up the cable, it is possible to infer that the cable is disconnected from the platform.

Therefore, the aim of this work is to provide a sensorless failure detection and broken cable identification (FDI) strategy for CSPRs whose number of cables is sufficient to control all the DOFs after failure. The FDI strategy is based on the definition of a digital twin to identify any changes between the real model and the twin, allowing the controller to correctly detect the failure event and identify the broken cable. The novelty of this work is the sensorless nature of the proposed FDI strategy since no other works have proposed a sensorless approach for the cable failure detection and broken cable identification. Indeed, previous works require measurements from multiple sensors or the end-effector tracking information, which may not be feasible to obtain after a cable failure event without adding additional sensors. Indeed, this is the strategy's greatest advantage since it does not require any additional devices than those adopted to control the robot. Moreover, the definition of a digital twin proposed in this work is suitable for applications with low computation time (i.e. 2 ms). This strategy is paired with an end-effector recovery (ER) strategy to move the end-effector towards a safe position, with the aim of minimizing safety risk. The ER strategy originates from a previously defined and tested approach that can be applied to real-time industrial applications. Indeed, an effective recovery strategy should avoid collision and provide a fast FDI and ER strategy for the end-effector even when its computational efficiency is critical and in the presence of only integrated sensors, such as encoders or resolvers. For this reason, the prototype of a three-degree-of-freedom (3-DOF) cable-suspended parallel robot adopted in this work is composed of industrial-grade components, emphasizing the practical implication of the proposed work.

This work is organized as follows: Section 2 presents the underlying equations of the model adopted to represent the CDPR. Section 3 presents the theoretical formulation for both the failure detection algorithm and the recovery strategy. A simulation-based test is also presented. Section 4 describes the experimental setup adopted and Section 5 presents the experimental validation of the theoretical formulation presented in Section 3, presenting a comparison of the digital twin and the real robot, the capabilities of the FDI algorithm, and the effectiveness of the recovery strategy. Lastly, Section 6 concludes the work.

2. Cable robot model

To realize an appropriate digital twin, it is necessary to correctly define both the kinematic and dynamic models of the considered class of CDPRs. The proposed strategy is applied to a CSPR whose number of cables is sufficient to ensure that the robot is at least fully actuated after the failure event. Hence, considering a 3-DOF robot, at least three cables whose exit points do not lie in the same vertical plane must be healthy after the failure event. This requirements entails that the number of cables m is at least 4, as depicted in Fig. 1.

The end-effector was considered as a point mass with only translational DOFs since recovery strategies are generally focused on the end-effector position. This assumption allows simplifying the dynamics and kinematics of the end-effector, which is suitable as the most important goal of the strategy consists of avoiding collision of the end-effector.

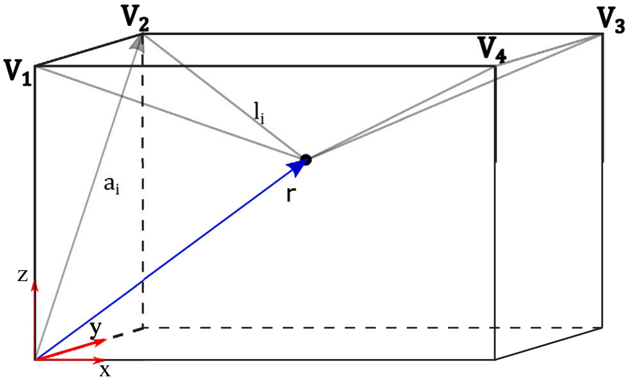

Fig. 1. Structure of the considered cable robot.

2.1. Dynamic model

Focusing on the actuators of the cable robot, each actuator is composed of an electric motor, a drum, and a rotating pulley. To evaluate the torque necessary to perform a certain motion, it is necessary to study the actuator dynamics, which can be described as

$$j_{m,i}\ddot{\theta}_i = -b\dot{\theta}_i + c_i + \rho\tau_i \tag{1}$$

where $j_{m,i}$ is its moment of inertia, θ_i, $\dot{\theta}_i$, and $\ddot{\theta}_i$ are the motor position, velocity, and acceleration, respectively. b is the friction coefficient, c_i is the motor torque, and ρ the drum radius. We assume $j_{m,i}$ and ρ as constants since the mass of the cables is considered negligible with respect to the magnitude of the system.

Expressing it in vector form

$$\mathbf{c} = \mathbf{j_m}\ddot{\boldsymbol{\theta}} + b\dot{\boldsymbol{\theta}} - \rho\boldsymbol{\tau} \tag{2}$$

considering that all the actuators have the same drum radius ρ and friction coefficient b.

To evaluate the required torque for a certain end-effector trajectory $\mathbf{p}$, it is first necessary to consider the required cable tensions. The cable tensions $\boldsymbol{\tau}$ can be evaluated considering the overall wrench $\mathbf{w}$, i.e., the sum of the forces $\mathbf{w_c}$ applied to the end-effector by the m cables and $\mathbf{w_e}$ by the other external forces:

$$\mathbf{w} = \mathbf{w_c} + \mathbf{w_e} = m\ddot{\mathbf{p}} \tag{3}$$

where m is the end-effector mass and $\ddot{\mathbf{p}}$ is the end-effector acceleration.

The cable wrench $\mathbf{w_c}$ is related to the cable tensions $\boldsymbol{\tau}$ by means of the structure matrix $\mathbf{S}$ [25]:

$$\mathbf{S} = \left[\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_m\right] \tag{4}$$

where $\mathbf{u}_i$ are unit vectors oriented from the end-effector towards the corresponding exit point V_i. Hence, it is possible to define:

$$\mathbf{w_c} = \mathbf{S}\boldsymbol{\tau} \tag{5}$$

In this way, it is possible to define the overall wrench as:

$$\mathbf{w} = \mathbf{S}\boldsymbol{\tau} + \mathbf{w_e} = m\ddot{\mathbf{p}} \tag{6}$$

From Eq. (2) it is possible to define $\boldsymbol{\tau}$ as a function of the motor torque $\mathbf{c}$, the motor acceleration $\ddot{\boldsymbol{\theta}}$, and the motor speed $\dot{\boldsymbol{\theta}}$:

$$\boldsymbol{\tau} = \frac{\mathbf{j_m}\ddot{\boldsymbol{\theta}} + b\dot{\boldsymbol{\theta}} - \mathbf{c}}{\rho} \tag{7}$$

which can be substituted in Eq. (6) to obtain:

$$\mathbf{S}\mathbf{j_m}\ddot{\boldsymbol{\theta}} + \mathbf{S}b\dot{\boldsymbol{\theta}} - \mathbf{S}\mathbf{c} + \rho\mathbf{w_e} = m\rho\ddot{\mathbf{p}} \tag{8}$$

To finally complete the dynamic model, it is first necessary to define the kinematic model and relate the end-effector motion $\mathbf{p}$ to the motors' angular motions $\boldsymbol{\theta}$.

Given the hypothesis of point-mass end-effector, it is possible to evaluate the cable length l_i from the actuator angular position θ_i as

$$l_i = l_{i,0} + \rho\theta_i \tag{9}$$

where $l_{i,0}$ is the distance between the actuator and the corresponding exit point B_i, which is set as constant. By deriving the equation, the cable length velocity $\dot{l}_i$ can be evaluated as:

$$\dot{l}_i = \rho\dot{\theta}_i \tag{10}$$

and in matrix form:

$$\dot{\mathbf{l}} = \rho\dot{\boldsymbol{\theta}} \tag{11}$$

The relationship between $\dot{\mathbf{l}}$ and the end-effector velocity $\dot{\mathbf{p}}$:

$$\dot{\mathbf{l}} = \mathbf{J}\dot{\mathbf{p}} = -\mathbf{S}^T\dot{\mathbf{p}} \tag{12}$$

where $\mathbf{J}$ is the Jacobian matrix, which is related to the structure matrix by means of the kinetostatic relation. Hence, it is possible to obtain the velocity equation:

$$\dot{\boldsymbol{\theta}} = -\frac{\mathbf{S}^T\dot{\mathbf{p}}}{\rho} \tag{13}$$

and thus the acceleration equation:

$$\ddot{\boldsymbol{\theta}} = -\frac{\mathbf{S}^T\ddot{\mathbf{p}}}{\rho} - \frac{\dot{\mathbf{S}}^T\dot{\mathbf{p}}}{\rho} \tag{14}$$

Therefore, substituting Eqs. (12) and (14) into Eq. (8), it is possible to evaluate the required motor torque as:

$$\mathbf{c} = \frac{\mathbf{S}^{\dagger}(\mathbf{w_e}\rho^2 - \mathbf{S}\mathbf{j_m}\mathbf{S}^T\ddot{\mathbf{p}} - \mathbf{S}\mathbf{j_m}\dot{\mathbf{S}}^T\dot{\mathbf{p}} - \mathbf{S}b\mathbf{S}^T\dot{\mathbf{p}} - m\rho^2\ddot{\mathbf{p}})}{\rho} \tag{15}$$

where $\mathbf{S}^{\dagger}$ is the Moore–Penrose generalized inverse of $\mathbf{S}$.

2.2. Control model

To ensure that the real robot moves accordingly, a position control algorithm has been implemented as in [26], which is based on a position PD controller. This type of controller was adopted since the simple structure of a PD controller is suitable for real-time applications, avoiding the use of complex and time-consuming algorithms. Moreover, since previous work showed suitable results, no other solutions were investigated.

The position loop compares the position feedback $\theta_{i,r}$ of the actuator sensors to evaluate the error value between the actual position of the actuators and the reference position $\theta_{i,d}$. The reference position is obtained from the inverse kinematic model, allowing us to easily compare the desired position of the end-effector, defined in the Cartesian space, with the actuator feedback, which is defined in the joint space.

To improve the responsiveness of the system, a feedforward term was adopted, as in [26]. Moreover, adopting a feedforward term based on the inverse dynamics equation (Eq. (15) that takes into account the system dynamics allows us to maintain positive cable tensions under disturbance, such as the oscillatory motion of the end-effector. Fig. 2 presents the complete control scheme, consisting of a feedback position control algorithm that provides the corrective motor torque $\mathbf{c}_{FB}$, and an inverse dynamic feed-forward approach that evaluates the motor torque $\mathbf{c}_{ID}$, achieving the overall motor torque $\mathbf{c}$.

2.3. Digital twin

A similar control system provides for the torques of the digital twin, with an inverse-dynamics feedforward term providing for the torque $\mathbf{c}_{ID,dt}$ and a position-based control algorithm providing for the corrective torque $\mathbf{c}_{FB,dt}$. The feedforward torque is the same for both the real robot and the digital twin, since the dynamic model is the same, whereas the PD controller of the digital twin required a proper tuning, similar to the one of the real robot.

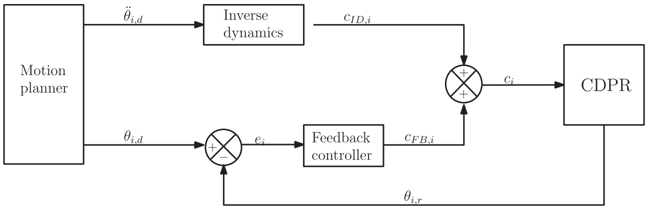

Fig. 2. Control scheme of the considered robot.

Given the actuator torque, to simulate the CSPR behavior, the direct dynamic equation has been adopted to iteratively evaluate the end-effector acceleration $\ddot{\mathbf{p}}_k$ at step k from the torques $\mathbf{c}_{k-1}$ at step $k-1$:

$$\ddot{\mathbf{p}}_k = \frac{\mathbf{w_e}\rho^2 - \mathbf{S}_{k-1}\mathbf{j_m}\dot{\mathbf{S}}_{k-1}^T\dot{\mathbf{p}}_{k-1} - \mathbf{S}_{k-1}\mathbf{c}_{k-1}\rho - \mathbf{S}_{k-1}b\mathbf{S}_{k-1}^T\dot{\mathbf{p}}_{k-1}}{m\rho^2 + \mathbf{S}_{k-1}\mathbf{j_m}\mathbf{S}_{k-1}^T} \tag{16}$$

where the values of the end-effector position and velocity, and therefore the structure matrix, refer all to step $k-1$ and are obtained by integrating the acceleration $\ddot{p}_{k-1}$ at step $k-1$. Given the kinetostatic relationship, it is possible to evaluate the actuator accelerations as in Eq. (14). Furthermore, by integrating the direct dynamic equation, it is possible to evaluate the Cartesian velocity $\dot{\mathbf{p}}$ and position $\mathbf{p}$ of the end-effector. Hence, it is possible to realize a digital twin of the considered CSPR which resembles the control scheme of the real robot.

3. Failure detection algorithm and recovery strategy

3.1. Failure detection and identification algorithm

To properly detect the failure event, it is first necessary to properly identify the effects of the cable failure event on the CSPR. In addition to the change in the SEW due to cable failure, when a cable fails, the kinematic relationship between the motor angles and the position of the end-effector described above is no longer valid. In fact, when a cable breaks, the structure matrix $\mathbf{S}$, which starts as a $n \times m$ matrix, becomes a $n \times m-1$ matrix. Hence, considering the inverse kinematic from Eq. (14):

$$\ddot{\mathbf{p}} = -(\mathbf{S}^T)^{\dagger}(\ddot{\theta}\rho + \dot{\mathbf{S}}^{\mathbf{T}}\dot{\mathbf{p}}) \tag{17}$$

due to the changes in the structure matrix, there is a difference between the real end-effector acceleration, which is given by $m-1$ cables, and the reference one given by m cables. Therefore, there is a difference between the acceleration of the end-effector of the real robot and of the digital twin, which follows the reference trajectory because it is not affected by cable failure.

Moreover, it is possible to observe that the acceleration $\ddot{\theta}_f$ of the motor corresponding to the broken cable increases due to the change in the applied torque c_f. Indeed, since there is no cable tension to balance it, Eq. (2) becomes:

$$c_f = j_{m,f}\ddot{\theta}_f + b\dot{\theta}_f \tag{18}$$

where $j_{m,f}$ is the rotor inertia of the actuator corresponding to the broken cable and $\dot{\theta}_f$ its velocity. If we neglect the friction term for the sake of simplicity and consider a fixed value of c_f, it is possible to observe that the motor acceleration will increase after the failure event:

$$\ddot{\theta}_f = \frac{c_f}{j_{m,f}} \tag{19}$$

Hence, the failure event not only causes a change in the end-effector acceleration, but also an increase in the acceleration of the motor corresponding to the broken cable. This is shown in Fig. 3, where the actuator acceleration corresponding to the broken cable is represented.

Due to the variation in the acceleration, as the cable fails at 3.5 s, the feedback controller of the real robot behaves differently from the controller of the digital twin, since the dynamics of the real robot is no longer represented by the feedforward term, whereas the digital twin does not observe any difference since its cables are still healthy. Therefore, a first approach may suggest that it may be possible to detect the failure event by detecting the spike in the real acceleration (in red) and to identify the broken cable.

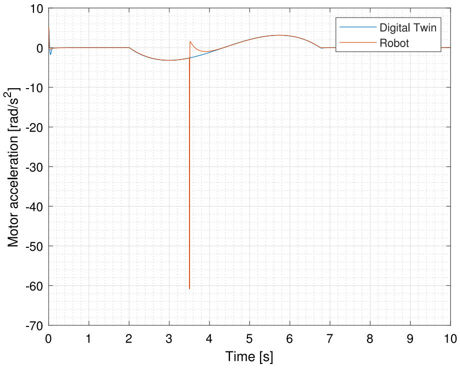

Fig. 3. Comparison of the motor acceleration between the digital twin and the robot during cable failure: corresponding to the failure event at 3.5 s, the acceleration of the actuator corresponding to the broken cable quickly changes, increasing by 30 times. However, due to the feedback controller, the acceleration error is quickly recovered.

However, since the robot is position controlled, the changes in acceleration are quickly recovered by the PID-based feedback controller, and it may be difficult to distinguish the acceleration error from an external noise in a real application. Hence, to better detect and identify the failure event, it is preferable to compare the torques between the real robot and the digital twin. Indeed, as per Eq. (18), the torque c_f of the failed actuator required to follow the desired trajectory decreases, and this is proved in Fig. 4, which compares the torques for each actuator of the digital twin and the simulated robot. Indeed, it is possible to observe that, after an initial oscillation, c_f settles to 0. On the other hand, the torque of the other actuators changes, with two cables increasing their torques to sustain the end-effector weight and the last one reducing the torque since the opposite torque given by the broken cable is lost.

Therefore, it is possible to compare the torques of the digital twin with the robot to observe a change in the robot's state and to identify the failure event. To identify the broken cable, the system first detects if there is any difference between the torques of the digital twin $\mathbf{c}_{DT}$ and of the real robot $\mathbf{c}$ (*detection condition*):

$$|c_{DT,i} - c_i| > \epsilon \quad \forall i \in [1, m] \tag{20}$$

where ϵ is a threshold value properly defined to avoid detecting disturbances. Indeed, after a cable failure scenario of a CSPR, all the actuators' torques need to change and an error is observable when comparing to the digital twin. Since in a failure event the actuators' torques of the real robot are different from the ones of the digital twin, at least one actuator provides a much lower torque since there is no cable tension to balance. Hence, a commanded torque near zero means that the corresponding actuator requires only a minimum torque to reach the target position, thus no cable forces are applied to the motor, as in a cable failure event. However, the effect on commanded torque is not due to the feedforward term, but to the position control loop. Hence, to identify the broken cable among the others, we search for an actuator whose commanded torque is very similar to c_f in Eq. (18) (*identification condition*); the detection condition avoids a false positive signal when a minimum tension is applied to the cable. Indeed, the system first detects the cable failure event in the presence of a difference in torque with the digital twin, which can simulate cable slackness and is different from the nominal torque. These two conditions can be better described

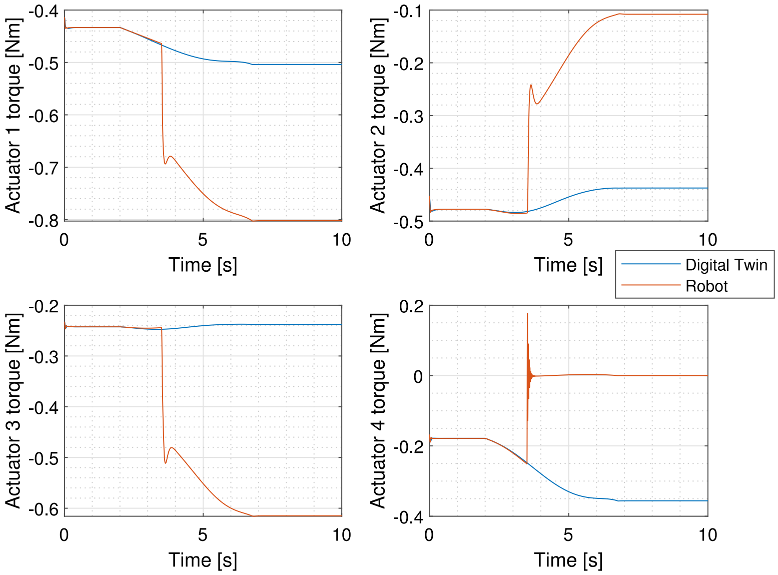

Fig. 4. Comparison of the motor torques between the digital twin (blue) and the robot (red): after the failure event at 3.5 s, the actuators' torques quickly change, with the one corresponding to the broken cable (actuator 4) dropping to zero. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

as identifying the set f of indexes i that satisfies both the detection condition and identification one:

$$f = \{i | 1 \le i \le m \wedge |c_{DT,k} - c_k| > \epsilon \quad \forall k \in [1,m] \wedge |c_i - c_f| < \delta\} \tag{21}$$

where δ is a small arbitrary value. This condition allows to consider also the possible scenario where multiple cables fail, providing that the robot is at least fully actuated after the failure event.

The advantage of this approach is that it does not require any sensor signal except for the position sensors of the actuators used for the position control algorithm. Hence, by adopting a properly modeled digital twin, it is possible to compare the torques to detect the cable failure event, and by identifying the actuator applying near-zero torque, it is possible to identify the broken cable.

3.2. Recovery strategy

As the cable failure event has been detected, it is possible to proceed with a cable failure recovery strategy. Hence, it is possible to pair the FDI algorithm with a recovery strategy. We decided to develop a strategy based on tension control of the residual cables since it is suitable for real-time applications. However, other end-effector recovery strategies may be adopted without affecting the validity of the FDI algorithm.

This strategy aims to move the end-effector towards a predefined safe point (SP) inside the residual SEW, whose position depends on the broken cable. SP can be defined on the basis of several criteria, such as the robot pose at failure or the presence of obstacles inside the workspace, as already highlighted in other works. As the considered workspace is occupied by a table, the main focus of this work was to keep the end-effector at a suitable height to avoid collision with the table; without any further constraints, the x, y coordinates of SP were defined as the barycenter of the area obtained by intersecting the residual SEW with the horizontal plane at the chosen height.

After the offline definition of the SP, it is necessary for the algorithm to plan the trajectory towards the safe pose online. This can be achieved by planning a linear trajectory whose simple definition is suitable for a real-time application. However, to avoid discontinuities in cable tensions that could lead to further cable failures, the end-effector velocity should be null or already oriented toward the SP [18]. Previous approaches to define a connecting path before the linear trajectory to orient the end-effector velocity have proven effective, but they may also be expensive in terms of computational time.

On the other hand, previous studies [27] have shown that it is possible to control the end-effector outside the SEW as long as all cables have positive tensions. This leads to the result that it is possible to control the end-effector by planning in the tension-space by defining the initial and target values of the tension that are both positive. The definition of a target tension state can be easily evaluated as the static tensions at the SP

$$\mathbf{t_{SP}} = \mathbf{S_R^{\dagger}} \mathbf{w_e} \tag{22}$$

where $\mathbf{S_R}$ is the structure matrix evaluated considering the remaining cables and $\mathbf{w_e}$ is evaluated as the gravity force $[0, 0, -mg]^T$. On the other hand, the initial set of cable tensions $\mathbf{t_R}$ is evaluated on the basis of the forces applied to the end-effector as the failure event is detected. Indeed, we aim to apply the same force $\mathbf{f_f}$ at the instant of failure detection to avoid further discontinuities in the end-effector acceleration, which could lead to further failures or to instability of the end-effector. Therefore, the algorithm evaluates $\mathbf{f_f}$ as

$$\mathbf{f_f} = m\ddot{\mathbf{p}} - \mathbf{w_e} + \mathbf{f_{FB}} \tag{23}$$

where $\mathbf{f_{FB}}$ are the forces due to the feedback controller, evaluated as:

$$\mathbf{f_{FB}} = \mathbf{S_R} \mathbf{t_{FB}} = \mathbf{S_R} \frac{\mathbf{c_{FB}}}{\rho} \tag{24}$$

where $\mathbf{t_{FB}}$ and $\mathbf{c_{FB}}$ are the components of the cable tensions applied by the remaining cables and of the respective actuators' torques due to the feedback controller, respectively.

Hence, it is possible to evaluate $\mathbf{t_R}$ as

$$\mathbf{t_R} = \mathbf{S_R}^{\dagger} \mathbf{f_f} \tag{25}$$

By defining a motion law in the tension space, it is possible to control the end-effector outside the SEW and move it inside it. Although this strategy does not ensure the motion of the end-effector inside a cluttered environment, it may be possible to add waypoints in the tension space to avoid obstacles in the workspace. However, while planning in the tension space allows controlling the end-effector outside the SEW, the approach moves the end-effector with only a dynamic feed-forward term, thus it does not ensure that the end-effector reaches SP with the proper precision. Hence, a second phase with a motion towards the SP is required, similar to previous recovery strategies [26]. This motion inside the residual SEW is defined as a simple linear trajectory that starts when the end-effector velocity is sufficiently low

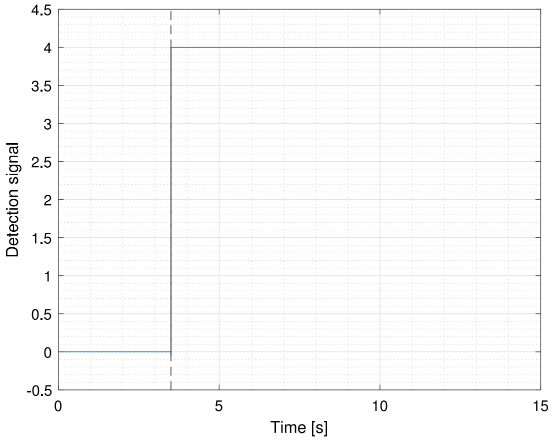

Fig. 5. Detection signal of the FDI strategy. The dashed line identifies the real time instant of the failure event.

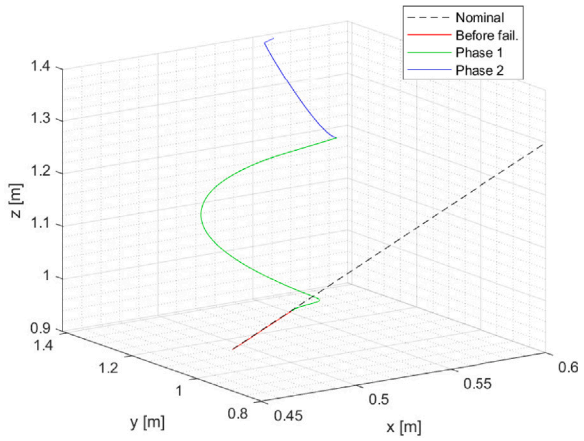

Fig. 6. Spatial trajectory of the end-effector during the recovery strategy: the red line is the trajectory before cable failure, whereas the green and blue lines represent the two phases of the recovery strategy. The dashed line is the nominal trajectory. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

and stops the end-effector at the desired SP thanks to the feedback controller, which can now be used since the end-effector is inside the SEW.

3.3. Approach application

Before the experimental tests, the approach was tested in a simulation model developed in Mathworks Simulink, representing both the cable robot and its controller. In addition, a digital twin of the simulated cable robot is implemented to test the FDI algorithm. The model is simulated with a fixed time step of 0.002 s. This value was chosen in order to be similar to the time step of the experimental setup, whose PLC runs at 500 Hz. The model is designed in three parts, starting with a motion generator algorithm that evaluates the motor position required to reach a certain pose in the Cartesian space using an inverse kinematics algorithm, the CSPR models (both the simulated robot and the digital twin), and the FDI and ER algorithms (the latter only applied to the simulated robot).

Both CSPR models are defined as in Section 2, with actuator torques as input data, evaluating their position and velocity, which allows evaluating the cables length and velocity. Moreover, in the simulation environment, it is possible to evaluate additional outputs, i.e., the end-effector position and velocity in the absolute reference frame, and the cable tensions, which cannot be measured in an experimental setup without additional sensors but must be evaluated using a direct kinematics algorithm.

To simulate the failure event, the algorithm changes the evaluation of the structure matrix, considering only the first three cables, thus affecting the acceleration of the end-effector $\ddot{\mathbf{p}}$ according to Eq. (16), and of the actuators $\ddot{\theta}$ according to Eq. (14). The acceleration of the fourth actuator, hereafter considered as corresponding to the broken cable, is evaluated according to Eq. (19).

By adopting the proposed FDI strategy, the algorithm is capable of correctly identifying the broken cable, as seen in Fig. 5, where the signal of the failure event is presented, showing the difference in time between the failure event (identified by the dashed line) and the rise of the detection signal. The algorithm is defined to indicate that the value of the detection signal value corresponds to the broken cable.

The failure event, which happens at 3.5 s, is detected at 3.504 s, showing a delay of 4 ms. This result is possible due to the low threshold in the torque error, set at 0.03 Nm, which can be achieved when working in simulation, as the modeled disturbances are the same.

Given a detection signal different from zero, the recovery strategy algorithm first acquires all the information regarding the robot trajectory, i.e., the end-effector reference position and acceleration, the actuators' angular position and velocity, and the current torques provided by the PD feedback controller in order to evaluate the initial value of the cable tensions as seen previously. Given the initial value of the cable tensions and the position of the SP corresponding to the broken cable, the approach designs a 5th degree polynomial motion law in the tension space, which are converted in the motor torques required to apply the desired tensions.

For the second part of the ER strategy, a feedback loop is required to know the actuators' position at every time step needed for the recovery approach to ensure that the end-effector reaches the desired position. The resulting trajectory of the end-effector in the Cartesian space is represented in Fig. 6.

Fig. 6 highlights the different phases of the recovery strategy. Before the cable failure event, the end-effector follows the planned linear trajectory (red). As the FDI algorithm detects that cable 4 has broken, the ER strategy starts in its first phase, moving the end-effector in the tension space (green). In this scenario, no via points have been placed, leading to the arched trajectory to a first SP ([0.56, 1.28, 1.22] m). Then, after the actuator speeds are sufficiently low (less than 0.01 rad/s), the second phase moves the end-effector in the second SP ([0.56, 1.41, 1.38] m). The end-effector coordinates during the test can be better observed in Fig. 7, with an RGB-color norm to represent the coordinates.

The two dashed lines identify the different phases of the proposed approach: the first identifies the failure detection event, and the second identifies the end of the first phase of the recovery strategy. Although the actuators' velocity is sufficiently low, it is possible to observe that some residual oscillation is the x-coordinate. However, since the end-effector velocity cannot be measured without additional sensors, the actuator speeds are still adopted, accepting a low residual velocity on the end-effector.

The different phases of the recovery strategy can be better appreciated by observing the behavior of the actuator torques in Fig. 8. Again, two dashed lines identify the phases of the proposed approach.

In the initial phase, the torque behavior corresponds to the movement of the end-effector actuated by the four cables, which starts at 2 s. The first vertical dashed line in the figure shows the time of the detection event by the FDI algorithm (t = 3.5 s). Then, the first phase of the recovery strategy began. A sudden variation in the actuator torques can be observed, and which is expected due to the failure event to sustain the end-effector. The motion planning in the tension space leads to the initial behavior of the torques which follows a polynomial law. At t = 4.75 s, the actuator torques reach the desired values and are

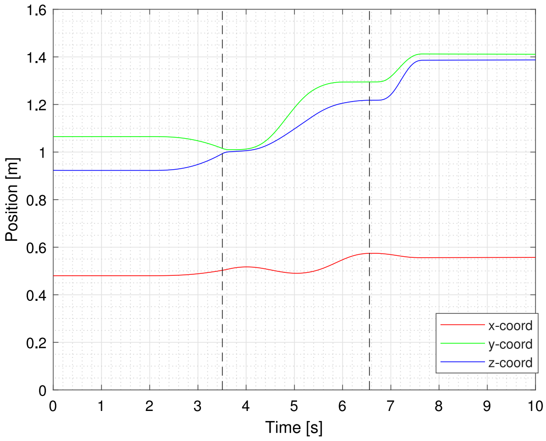

Fig. 7. End-effector coordinates during the recovery approach: x (**red**), y (**green**), z (**blue**). The dashed lines identify the failure detection event and the end of the first recovery phase. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

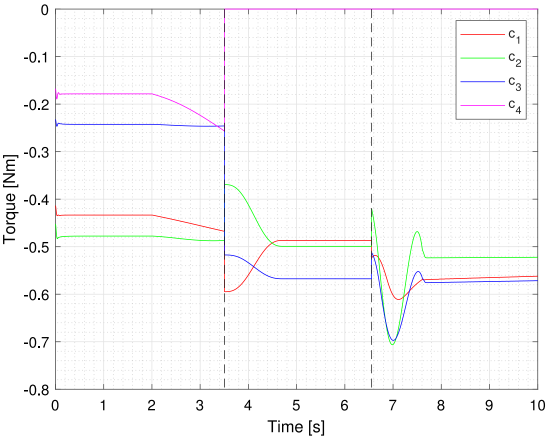

Fig. 8. Motor torques during the recovery approach for motor 1 (**red**), motor 2 (**green**), motor 3 (**blue**) and motor 4 (**magenta**). The dashed lines identify the failure detection event and the end of the first recovery phase. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

kept constant until $t = 6.556$ s, which corresponds to low actuator speeds. Actuator 4 torque, which corresponds to the broken cable, is kept at zero during the recovery strategy. The final phase of the recovery strategy begins after the second vertical dashed line. The actuator torques correspond to the linear trajectory planned to make the end-effector reach the second SP. The controller defines appropriate values of actuator torques during the recovery to guarantee that the cable tensions remain positive, and hence that the cables are always in tension.

4. Experimental setup

This section will present the setup adopted to address the failure detection and recovery strategy problem of a cable suspended parallel robot. To prove the possible industrial applications of the proposed method, industrial-grade components were adopted.

4.1. Control system

The servo system tasked with controlling the cable robot is composed of an industrial PC acting as the controller, two servo drives Beckhoff AX5206, four B&R 8LVB23ee004LjFn00 servo gear motor, each equipped with resolvers to observe their position and speed, with each servo drives controlling two actuators. More data on the servo drives are presented in Table 1.

Table 1
Characteristics of the adopted servo drive.

Drive Property	Value	Unit
Rated supply voltage	3 × 100 …480	V (AC)
DC-link voltage (max)	875	V
Rated output current (channel)	6	A
Peak output current (channel)	13	A

Table 2
Characteristics of the adopted actuators.

Actuator Property	Value	Unit
Nominal speed n_n	3000	rpm
Nominal torque τ_n	1.3	Nm
Moment of inertia J	5.12×10^{-4}	kgm^2
Drum radius ρ	36	mm
Static friction torque	0.2	Nm
Friction coefficient b	0.0015	–

In this work, we adopted a soft-PLC, i.e., a system composed of a PC (the hardware support platform) and a software that allows achieving the main functionality of a traditional (hard) PLC [28], for the control unit. This technology, while not as common as traditional PLCs, is becoming increasingly common in industrial applications as it provides several benefits, e.g., open architecture, network communication capability, and better data processing.

As in our previous work [26], TwinCAT 3 software was installed on a Windows 10 PC in kernel mode to turn it into a real-time controller. This software features the option to import Matlab Simulink projects into the PLC project, building the Simulink block diagram intro C++ code, allowing for sufficient computational effort for a real-time application. Indeed, the setup works with a cycle time of 2 ms. However, given the amount of signals required for testing and validation, the sampling time is fixed at 10 ms, with the results provided as MATLAB formatted dats files (.mat). The ability to incorporate a Simulink project into the PLC allows us to define the control model in the Simulink environment, increasing the system flexibility and allowing us to quickly simulate the system and validate the developed approaches. This is especially useful when developing the digital twin of the real robot, since it is possible to quickly change the simulation parameters. Moreover, the digital twin is integrated into the Simulink block diagram, thus it is implemented into the robot controller to cyclically monitor the robot behavior.

Lastly, as in [26], the B&R 8LVB23ee004LjFn00 servo gear motors were adopted considering that each motor is required to compensate for cable tension τ due to the weight of the end-effector. It is important to note that, for the recovery strategy to be effective, at least two motors should be able to support the weight of the entire platform.

The characteristics of the adopted actuators are presented in Table 2. The velocity data provided by the resolvers allow for the implementation of the three terms of the PID control algorithm without deriving the position information, avoiding the need for a filter for the derivative of the position data.

4.2. Robot architecture

Fig. 9 shows the prototype 3-DOF CSPR with four cables that was developed in the Robotics and Automation Laboratory at the Department of Industrial Engineering of the University of Padova.

The size of the robotic workcell, along with the mass m_e and radius r_e of the end-effector, are presented in Table 3. In the proposed setup, the end-effector is a steel ball with a ring, which was used to attach the four cables. This could lead to a difference from the adopted point-mass model. Indeed, as previously stated, we assumed the end-effector

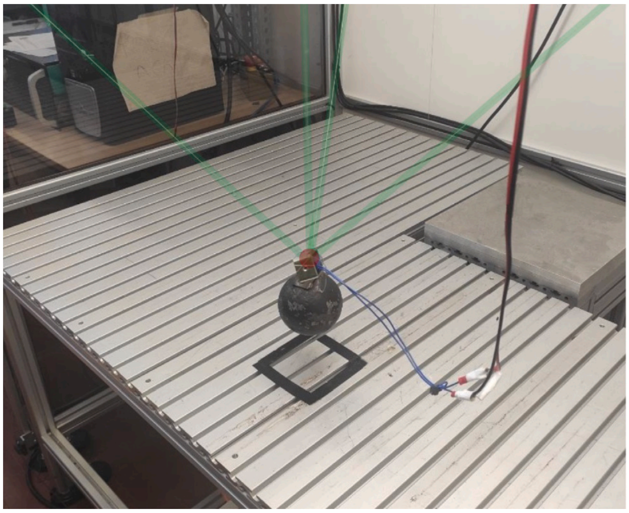

Fig. 9. Prototype cable suspended parallel robot (CSPR) used during the tests. The green lines emphasize the cables, which unite in the anchor point represented by the red dot.

Table 3
Size of the robot workcell and of the end-effector.

Parameter	Value	Unit
a	1688	mm
b	1930	mm
h	2438	mm
m_e	2.941	kg
r_e	90	mm

as a point mass, which allows us to simplify both the kinematics and the dynamics of the cable robot without lack of generality. This is a reasonable approximation when dealing with the failure detection and recovery problem. However, this difference was considered an acceptable compromise between the model hypothesis and the technological constraints. Moreover, the introduction of the inertia in the model does not provide for significant changes when comparing with the digital twin.

Each cable is wound around a metal drum with grooves for the cables fixed on the motor axis, with each motor placed on the lower vertices of the workcell. Each exit point is placed at the upper vertices of the cell and is composed of a movable pulley, whose axis rotates around a vertical axis to allow the cable to change direction. The movable pulleys have a radius of 5 mm. The cables, made of Dyneema fibers, are tied to the ring of the end-effector to form a single attach point. Lastly, to effectively simulate the cable failure, an electromagnet is used to connect two extremities of the cable. To power the electromagnet, a 12 V external power supply is adopted, represented by the blue cables in Fig. 9. To control the electromagnet, an external signal from the controller is adopted. The electromagnet is provided with a spring to quickly separate the parts without any significant delay.

In conclusion, the presented system represents the typical setup of an industrial CSPR, since it has been designed with industrial grade components and does not require additional sensors for the failure detection and recovery strategy other than the ones embedded in the actuators, i.e., the resolvers.

5. Experimental results

5.1. Comparison between the digital twin and real setup

For the approach to be effective, the digital twin of the robot should be properly designed. For the digital twin to be effective, the difference with the real robot should be minimized, represented by three Key Performance Indicators (KPI) evaluated as the maximum error in the Cartesian position (K_c), in the joint position (K_j), and in the actuators' torque (K_t). Indeed, it is important that both the kinematic and the dynamic properties are similar. Hence, the digital twin was designed to consider and simulate disturbances affecting the real robot, such as friction and the electromagnet. Fig. 10 compares the spatial trajectory of the digital twin (green) and of the real robot (red) when following a circular trajectory as a reference (blue). Since the system adopts only resolvers, the measured Cartesian position of the end-effector is evaluated by means of direct kinematics; however, previous work on the same experimental setup showed a good matching between the two [26].

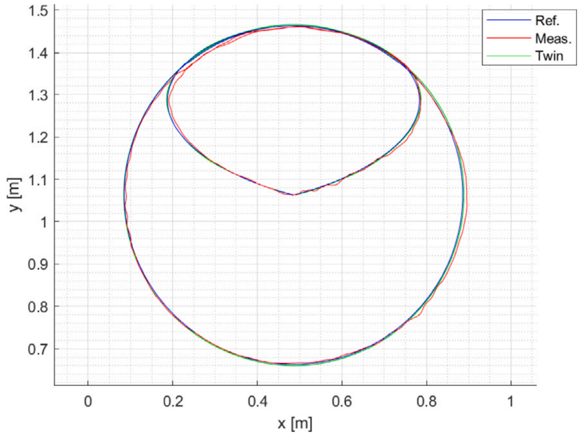

Fig. 10. Spatial trajectory of the end-effector: comparison between the reference motion (**blue**), the measured motion (direct kinematics, **red**), and the digital twin one (**green**). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

These results show a close matching between the Cartesian trajectories of the digital twin and the real robot, with a value of K_c of 0.0143 m.

Regarding the evaluation of the second indicator, i.e., the KPI K_j concerning the maximum joint angular position error, Fig. 11 shows the angular position of each actuator when performing the circular trajectory in the Cartesian space.

The red line refers to the measured angular position provided by the sensors, the green line is the angular position estimated by the digital twin; the reference trajectory is also provided in blue. Again, it is possible to observe a good matching between the trajectories, with the value of K_j equal to 0.0583 rad.

Lastly, Fig. 12 compares the nominal torques of the real robot (red) and the digital twin (green). We did not present the measured torque from the drivers (evaluated from the measured current), since it is very similar to the reference torque comprised of the feedback torque and the feedforward term, and the inevitable noise only affects the readability of the figure.

Despite the similar behavior, some discrepancies between the reference torque and the digital twin ones can be observed, especially in the fourth actuator, which is equipped with the electromagnet. Indeed, the differences in the two torques are lower than 0.1 Nm for all actuators, except for motor 4, causing the value of K_t to increase to a value of 0.1566 Nm for the presented case. Although the electromagnet has been modeled as a point mass connected to the cable, these discrepancies are due to disturbances which could not be properly modeled. These differences between the two systems lead to a higher value of the threshold for the FDI algorithm; however, a higher value for the threshold increases the detection time and may even impede the detection of the failure event. Several tests have been carried out to identify the proper value of the difference that satisfies this trade-off, and we the threshold ϵ equal to 0.3 Nm, which allows us to detect the failure event in a reasonable time without the risk of false positives.

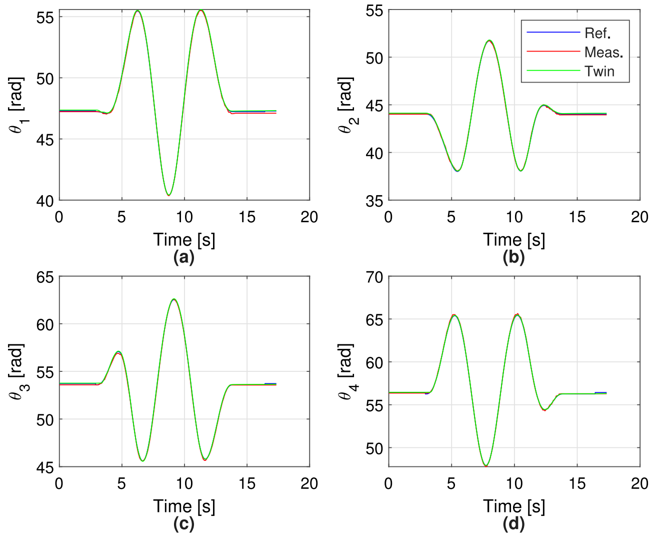

Fig. 11. Comparison of the angular position for each actuator: reference position (**blue**), measured position (**red**), digital twin position (**green**). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

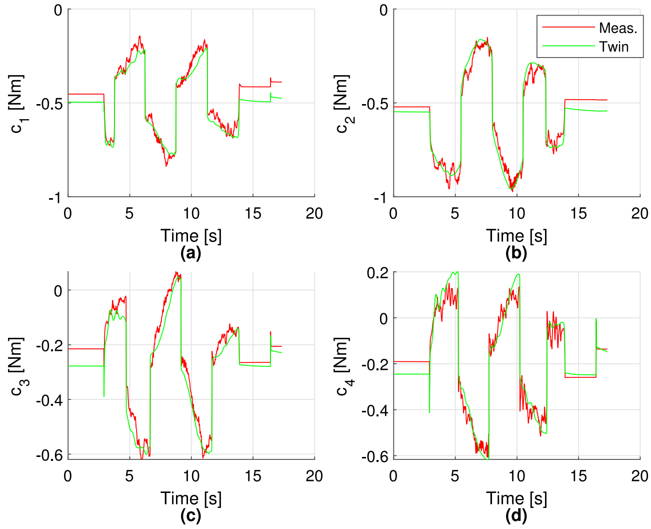

Fig. 12. Comparison of the torques for each actuator: reference torque (**red**), digital twin torque (**green**). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5.2. Failure detection algorithm

Given the torque error required to identify the failure event, it is necessary to identify how the cable tensions affect the FDI algorithm. Indeed, the tension distribution clearly affects the capability of the algorithm to identify the failure event: if the cable tension of the broken cable before failure is approximately equal to the minimum tension value, the failure event does not affect the tension distribution greatly, nor does reducing the capability of the FDI algorithm. On the other hand, when the contribution of the broken cable to the tension distribution before failure is significant, the algorithm should be more capable of detecting the failure.

The contribution of a cable on the tension distribution during a motion depends on the position of the end-effector in the workspace and on the motion direction. Indeed, if the end-effector is moving towards an exit-point, the corresponding cable tension increases, consequently changing the tension distribution. On the other hand, a motion orthogonal to the cable does not provide much change in the tension distribution. Hence, it is necessary to verify the capability of the FDI algorithm to detect the failure event for both different points in the workspace and for different directions.

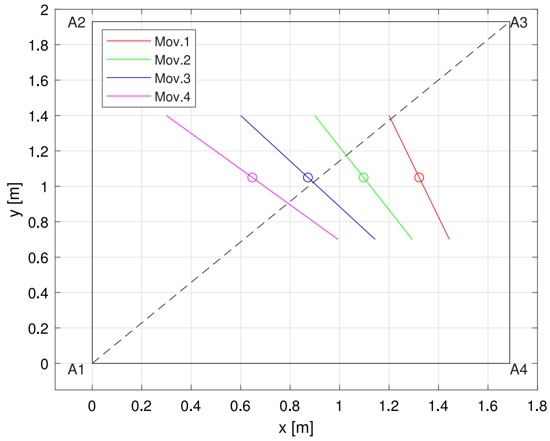

Fig. 13. Direction of motion for testing the FDI algorithm in a direction along the broken cable. For each direction, the FDI was tested for both sense of motion. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

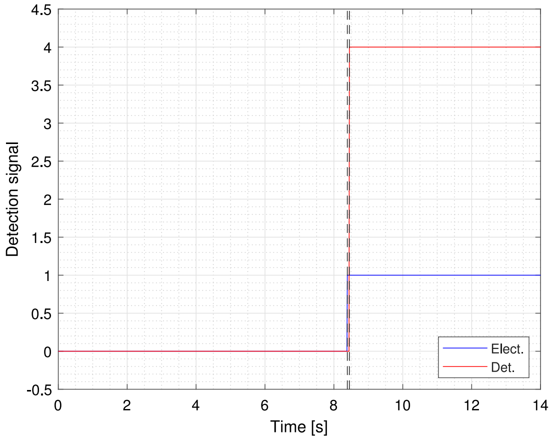

Fig. 14. Detection signal of the FDI strategy for motion 2 along the cable (red). The blue line identifies the electromagnet signal to simulate the failure event. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 4
Coordinates of the points defining the tangential movement towards exit-point 4.

Motion	Start [m]	Finish [m]
1	[1.2, 1.4, 1.25]	[1.4928, 0.56, 1.9628]
2	[0.9, 1.4, 1.25]	[1.3728, 0.56, 1.9628]
3	[0.6, 1.4, 1.25]	[1.2528, 0.56, 1.9628]
4	[0.3, 1.4, 1.25]	[1.1328, 0.56, 1.9628]

We first considered 4 movements directed towards the exit point A_4 as in Fig. 13, in order to obtain a motion along the cable to be broken. Each movement was tested in both senses of direction.

The coordinates of the starting and end point of each motion are presented in Table 4. For each movement, the failure point was defined as the midpoint of the segment, identified by the circles in Fig. 13.

Fig. 14 compares the detection signal from the FDI algorithm (red) with the signal used to turn off the electromagnet (blue) and separate the two extremities of cable 4, thus simulating the failure event. The failure event is activated when the end-effector is at the midpoint,

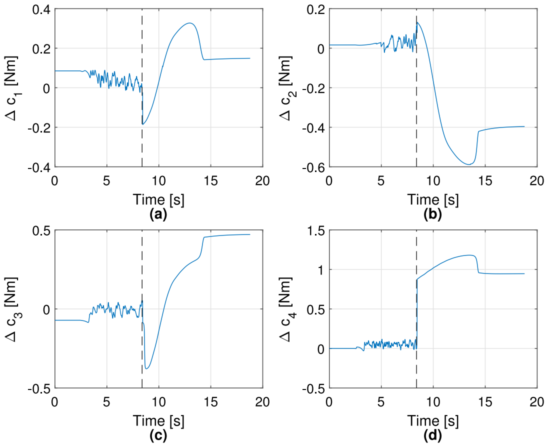

Fig. 15. Differences between the motor torques of the digital twin and the real robot. It is possible to identify a sudden spike in the difference in motor torque 4 after the failure event (identified by the dashed line).

Table 5
Detection time for the four motion along cable 4. $t_{forward}$ is the detection time with a motion towards the exit-point 4, $t_{backward}$ is the detection time in the opposite direction.

Motion	$t_{forward}$ [ms]	$t_{backward}$ [ms]
1	50	80
2	60	30
3	60	40
4	50	60

which corresponds at time 8.390 s and is detected at 8.450 s, i.e., after 60 ms.

Two dashed lines identify the interval between the failure event and the detection, highlighting the time interval. This small interval is given by the particular condition of motion, since the robot is pulling towards the corresponding exit-point, thus, a failure event creates a great change in the tension distribution. This is clearly visible in Fig. 15, which shows the differences between the motor torques of the digital twin and the real robot. After the failure event, identified by the dashed line, the difference increases rapidly, reaching a value of 0.874 Nm for cable 4 at 8.45 s.

Table 5 presents the detection time for the 4 motion in both the forward direction $t_{forward}$, i.e., towards the exit-point, and $t_{backwards}$ in the opposite direction.

The algorithm successfully detected the broken cable for all motions without errors.

Similarly, the FDI algorithm was tested in four directions orthogonal to the cable, in order to test the performance of the algorithm in the most critical condition. Fig. 16 shows the 4 directions considered, and the point of failure is identified by a circle. Again, each movement was tested in both directions of motion.

Table 6 presents the coordinates of the starting and end points of each movement. The coordinates were defined in order to obtain a motion orthogonal to the cable and belonging to the horizontal plane, i.e., the motion vector **b** was defined as

$$\mathbf{v} = \mathbf{n} \times \mathbf{u_4} \tag{26}$$

where $\mathbf{u_4}$ is the unit vector oriented from the end-effector towards exit-point A_4 and $\mathbf{n}$ is the normal vector to the horizontal plane, i.e., $[0 \quad 0 \quad 1]^T$.

Fig. 17 compares the detection signal from the FDI algorithm (red) with the signal used to break the cable by means of the electromagnet (blue). The failure event is activated when the end-effector is at the midpoint, which corresponds at time 4.850 s and is detected at 5.050 s, i.e., after 200 ms.

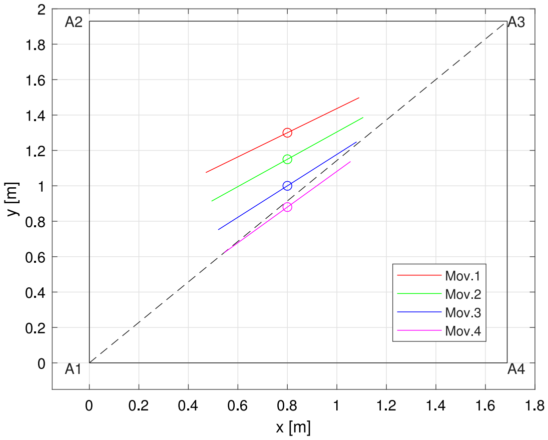

Fig. 16. Direction of motion for testing the FDI algorithm in a direction orthogonal to the broken cable. For each direction, the FDI was tested for both sense of motion. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 6
Coordinates of the points defining the orthogonal movement with respect to exit-point 4.

Motion	Start [m]	Finish [m]
1	[1.09, 1.4981, 1.25]	[0.4704, 1.0749, 1.25]
2	[1.1064, 1.3866, 1.25]	[0.4936, 0.9134, 1.25]
3	[1.0795, 1.2482, 1.25]	[0.5205, 0.7518, 1.25]
4	[1.0551, 1.1374, 1.25]	[0.5449, 0.6226, 1.25]

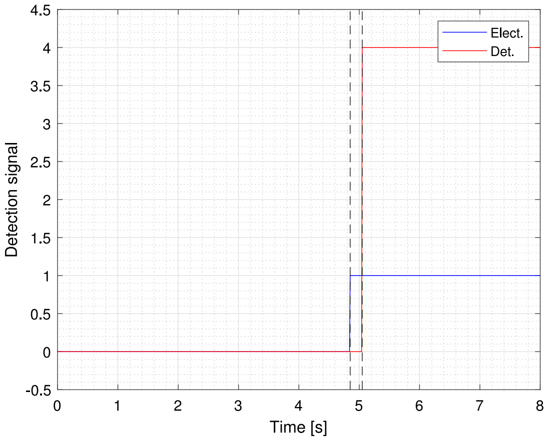

Fig. 17. Detection signal of the FDI strategy for motion 4 orthogonal to the cable (red). The blue line identifies the electromagnet signal to simulate the failure event. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Two dashed lines identify the interval between the failure event and the detection, highlighting the time interval. Since the cable does not pull during motion given the considered direction, the effect of the failure is less visible on the tension distribution, thus reducing the capability of the FDI algorithm. Moreover, except for the fourth motion (magenta in Fig. 16), for all failure points, the further exit-point is A_4, which means that tension τ_4 is minimum. These conditions greatly affected the ability of the algorithm to detect a change in the tension distribution. This is clearly visible in Fig. 18, which shows the differences between the motor torques of the digital twin and the real robot.

After the failure event, identified by the dashed line, the difference increases slowly, reaching at most a difference of about 0.2 Nm for most cables after 200 ms, except for cable 3 which reaches an error of 0.7 Nm

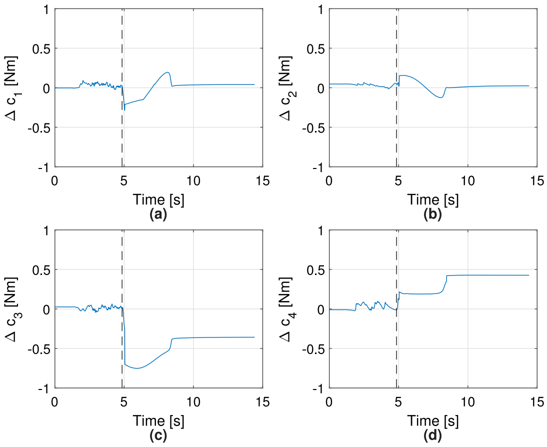

Fig. 18. Differences between the motor torques of the digital twin and the real robot. It is possible to observe that the error after the failure event (dashed line) is lower than the previous scenario, increasing the detection time.

Table 7
Detection time for the four motion orthogonal to cable 4. $t_{forward}$ is the detection time with a motion from the starting point to the finish one, $t_{backward}$ is the detection time in the opposite direction.

Motion	$t_{forward}$ [ms]	$t_{backward}$ [ms]
1	230	210
2	290	230
3	270	210
4	200	240

at 5.05 s. Despite the algorithm detects the failure event thanks to the difference δc_3 in the third torque, the FDI algorithm correctly detects the failure in cable 4, as it was the one with positive torque difference and minimum real torque.

Table 7 presents the detection time for the four movements in both the forward direction $t_{forward}$, i.e., from the starting point to the finish, and $t_{backwards}$ in the opposite direction. The values confirm that in unfavorable conditions the detection time greatly increases, requiring a very low threshold for the difference of the torques. Lastly, for all motions, the algorithm correctly identified the broken cable.

In conclusion, while the detection time for orthogonal movement is relatively high, it should be noted that it is a safe condition where it is less likely that the cable fails, as it does not effectively pull. Moreover, improvements in the digital twin can reduce the detection time.

5.3. Recovery strategy

The effectiveness of the FDI algorithm was further tested in the experimental setup when paired with the proposed ER strategy. A circular trajectory of radius 0.4 m was adopted for the test, as seen in Fig. 19, where the dashed line represents the nominal trajectory. A 2D view in Fig. 20 highlights the position of the end-effector with respect to the residual SEW, whose border is indicated in magenta.

The failure event was planned at time instant 5.87 s, i.e., when the end-effector has covered an arc of 135°. This point is placed near the exit-point 4, which is a favorable condition for the FDI algorithm, but an adverse one for the ER strategy, since the end-effector is completely outside the residual SEW.

The failure is detected at 6.06 s, i.e., after 190 ms, as seen in Fig. 21, which compares the electromagnet signal (blue) with the detection signal in red.

The different phases of the recovery strategy are highlighted in Fig. 19, with the red curve representing the motion before failure, which closely follows the nominal one, the green curve is the first phase, where the end-effector is moved in the tension space, and lastly, the blue line represents the second phase, where the end-effector is moved toward the SP placed at coordinate [0.62, 1.28, 1.2] m. The

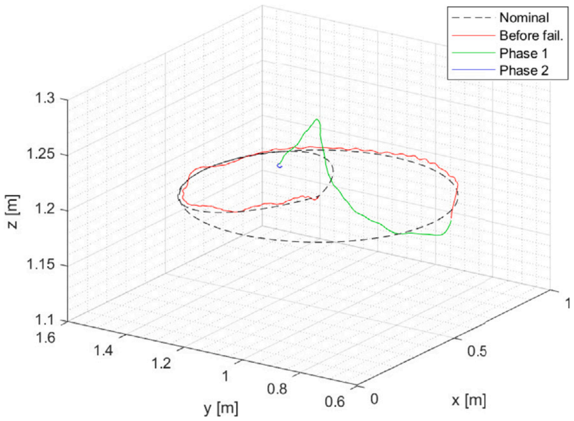

Fig. 19. Spatial trajectory of the end-effector during the recovery strategy. The red line is the trajectory before cable failure, whereas the green and blue lines represent the two phases of the recovery strategy. The dashed line is the nominal trajectory. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

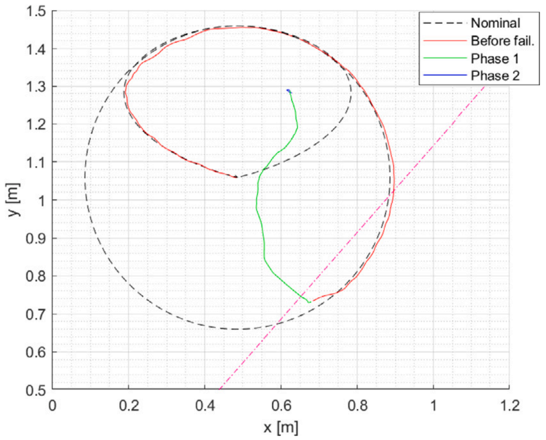

Fig. 20. 2D trajectory of the end-effector during the recovery strategy. The magenta line highlights the border of the residual SEW. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

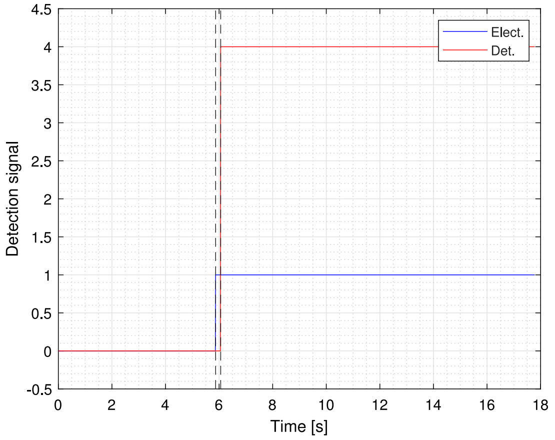

Fig. 21. Detection signal of the FDI strategy for the circular motion. The blue line identifies the electromagnet signal used to simulate the failure event. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

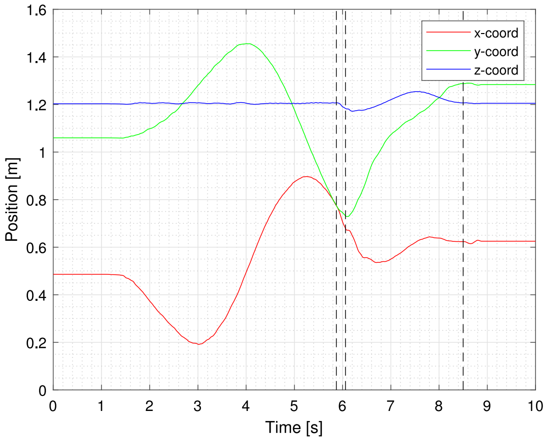

Fig. 22. End-effector coordinates during the recovery approach: x (**red**), y (**green**), z (**blue**). The dashed lines identify the failure event, the detection, and the end of the first recovery phase. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

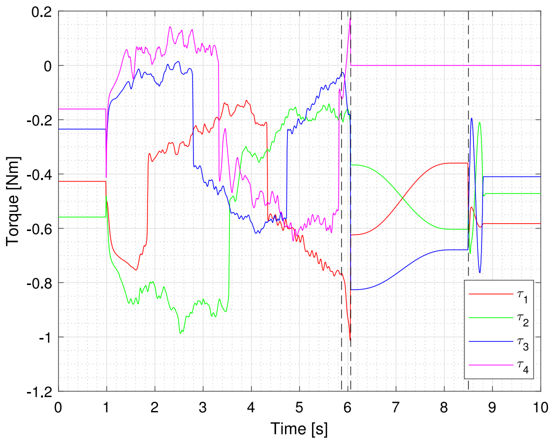

Fig. 23. Motor torques during the recovery approach for motor 1 (**red**), motor 2 (**green**), motor 3 (**blue**) and motor 4 (**magenta**). The dashed lines identify the failure detection interval and the end of the first recovery phase. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

end-effector coordinates can be observed in Fig. 22, with an RGB-color norm to represent the coordinates and three dashed lines to identify the different phases of the proposed approach.

The first interval represents the nominal behavior of the robot, the second one identifies the motion in the interval between the failure event and detection, the third one the motion during the first recovery phase, and the last one the motion towards the SP. Given that the SP chosen was the same for the two phases, the last motion is minimal.

Lastly, Fig. 23 shows the behavior of the motor torques during the recovery strategy.

The behavior corresponds to the simulation results, with the first vertical dashed line indicating the failure event. Given the increased delay for detection in comparison to the simulation, it is interesting to note the torques in this time interval. While actuator 1 (red) and actuator 3 (blue) need to increase the torque value to sustain the end-effector weight, actuator 4 unwinds to recover the position error due to the feedforward term. After the failure event has been detected, the controller provides the torques required to move in the tension space with a polynomial law. The third dashed line highlights the end of the first phase and the start of the second recovery phase, where the actuator torques correspond to the linear trajectory planned to make the end-effector reach the SP.

6. Conclusions

This work presents a sensorless failure detection and identification algorithm, whose sensorless nature makes it suitable for industrial approaches. Moreover, this approach is suitable for control systems even when computational efficiency is critical since it does not require complex algorithms. The algorithm is paired with an end-effector recovery strategy to provide for a complete cable failure recovery strategy. The experimental tests were carried out with industrial-grade equipment, further stating the practical implications of the proposed work. An electromagnet was adopted to simulate the failure scenario. The FDI algorithm was tested for different directions and positions in the workspace to verify the capabilities of the approach. The experiments proved the capability of the approach to correctly detect the failure event in a limited amount of time, allowing the recovery strategy to efficiently move the end-effector while maintaining a positive cable tension during the process.

CRediT authorship contribution statement

Giovanni Boschetti: Writing – review & editing, Writing – original draft, Supervision, Resources, Methodology, Formal analysis, Conceptualization. **Riccardo Minto:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation.

Acknowledgment

This study was carried out within the PNRR research activities of the consortium iNEST (Interconnected North-Est Innovation Ecosystem) funded by the European Union Next-GenerationEU (Piano Nazionale di Ripresa e Resilienza (PNRR) - Missione 4 Componente 2, Investimento 1.5 - D.D. 1058 23/06/2022, ECS_00000043). This manuscript reflects only the Authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

Data availability

Data will be made available on request.

References

[1] J.-P. Merlet, Parallel Robots, Vol. 128, Springer Science & Business Media, 2005.
[2] H.D. Taghirad, Parallel Robots: Mechanics and Control, CRC Press, 2013.
[3] A. Pott, H. Mütherich, W. Kraus, V. Schmidt, P. Miermeister, A. Verl, Ipanema: A family of cable-driven parallel robots for industrial applications, in: Cable-Driven Parallel Robots, Springer, 2013, pp. 119–134.
[4] G. Zuccon, A. Doria, M. Bottin, G. Rosati, Planar model for vibration analysis of cable rehabilitation robots, Robotics 11 (6) (2022) 154.
[5] C. Chesher, Robots and the moving camera in cinema, television and digital media, in: International Workshop on Cultural Robotics, Springer, 2015, pp. 98–106.
[6] K. Fyraj, P. Firsching, Using augmented reality to enhance the capabilities of human-machine interaction in industry, in: Tagungsband 1. Symposium Elektronik Und Systemintegration ESI 2018: Von Der Sensorik Bis Zur Aktorik in Interdisziplinärer Anwendung, 2018.
[7] M. Frasheri, H. Ejersbo, C. Thule, C. Gomes, J.L. Kvistgaard, P.G. Larsen, L. Esterle, Addressing time discrepancy between digital and physical twins, Robot. Auton. Syst. 161 (2023) 104347, http://dx.doi.org/10.1016/j.robot.2022.104347.
[8] Y. Zheng, S. Yang, H. Cheng, An application framework of digital twin and its case study, J. Ambient Intell. Humaniz. Comput. 10 (3) (2019) 1141–1153.
[9] V. Kuts, T. Otto, T. Tähemaa, Y. Bondarenko, Digital twin based synchronised control and simulation of the industrial robotic cell using virtual reality, J. Mach. Eng. 19 (2019).
[10] I. Palomba, D. Richiedei, A. Trevisani, Kinematic state estimation for rigid-link multibody systems by means of nonlinear constraint equations, Multibody Syst. Dyn. 40 (1) (2017) 1–22.

[11] G. Boschetti, F. González, G. Piva, D. Richiedei, B. Rodríguez Frade, A. Trevisani, Synthesis of an extended Kalman filter for cable-driven parallel robots, in: ECCOMAS Thematic Conference on Multibody Dynamics, Budapest University of Technology and Economics, 2021, pp. 277–288.

[12] V. Le Nguyen, R.J. Caverly, Cable-driven parallel robot pose estimation using extended Kalman filtering with inertial payload measurements, IEEE Robot. Autom. Lett. 6 (2) (2021) 3615–3622.

[13] T. Bruckmann, R. Boumann, Simulation and optimization of automated masonry construction using cable robots, Adv. Eng. Inform. 50 (2021) 101388.

[14] R.G. Roberts, T. Graham, T. Lippitt, On the inverse kinematics, statics, and fault tolerance of cable-suspended robots, J. Robot. Syst. 15 (10) (1998) 581–597.

[15] J. Bolboli, M.A. Khosravi, F. Abdollahi, Stiffness feasible workspace of cable-driven parallel robots with application to optimal design of a planar cable robot, Robot. Auton. Syst. 114 (2019) 19–28, http://dx.doi.org/10.1016/j.robot.2019.01.012.

[16] G. Boschetti, G. Carbone, C. Passarini, Cable failure operation strategy for a rehabilitation cable-driven robot, Robotics 8 (1) (2019) 17.

[17] A.T. Riechel, I. Ebert-Uphoff, Force-feasible workspace analysis for underconstrained, point-mass cable robots, in: IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004, Vol. 5, IEEE, 2004, pp. 4956–4962.

[18] G. Boschetti, C. Passarini, A. Trevisani, A recovery strategy for cable driven robots in case of cable failure, Int. J. Mech. Control 18 (2017) 41–48.

[19] A. Berti, M. Gouttefarde, M. Carricato, Dynamic recovery of cable-suspended parallel robots after a cable failure, in: Advances in Robot Kinematics 2016, Springer, 2018, pp. 331–339.

[20] R. Boumann, T. Bruckmann, Simulation and model-based verification of an emergency strategy for cable failure in cable robots, Actuators 11 (2) (2022) 56.

[21] S. Caro, J.-P. Merlet, Failure analysis of a collaborative 4-1 cable-driven parallel robot, in: European Conference on Mechanism Science, Springer, 2020, pp. 440–447.

[22] B. Salah, O. Janeh, T. Bruckmann, B. Noche, Improving the performance of a new storage and retrieval machine based on a parallel manipulator using fmea analysis, IFAC-PapersOnLine 48 (3) (2015) 1658–1663.

[23] A. Raman, I. Walker, V. Krovi, M. Schmid, A failure identification and recovery framework for a planar reconfigurable cable driven parallel robot, IFAC-PapersOnLine 55 (37) (2022) 369–375.

[24] R. Boumann, C. Jeziorek, T. Bruckmann, Validation of emergency strategies for cable-driven parallel robots after a cable failure, in: International Conference on Cable-Driven Parallel Robots, Springer, 2023, pp. 209–220.

[25] G. Boschetti, A. Trevisani, Cable robot performance evaluation by wrench exertion capability, Robotics 7 (2) (2018) 15.

[26] G. Boschetti, R. Minto, A. Trevisani, Experimental investigation of a cable robot recovery strategy, Robotics 10 (1) (2021) 35.

[27] C. Gosselin, Global planning of dynamically feasible trajectories for three-DOF spatial cable-suspended parallel robots, in: Cable-Driven Parallel Robots, Springer, 2012, pp. 3–22.

[28] Q. Liang, L. Li, The study of soft PLC running system, Procedia Eng. 15 (2011) 1234–1238.

Giovanni Boschetti is a professor of Industrial Robotics at the Department of Industrial Engineering of the University of Padova. He currently is the president of the IFToMM Italy Association, the Italian branch of the International Federation for the promotion of Mechanisms and Machine Science. His main research interests are focused on the kinematics and on the performance evaluation of Serial and Parallel Industrial Robots, on the dynamics and the performance evaluation of Cable Direct Driven Robots and on several aspects of interaction between humans and Collaborative Robots.

Riccardo Minto received his B.Sc. Degree in Mechanical Engineering in 2016 and his M.Sc. Degree in Mechanical Engineering in 2018 at the Department of Industrial Engineering of the University of Padova. In December 2022 he obtained the Ph.D. in Mechatronics and Product Innovation Engineering at the University of Padova. Since March 2023 he is Assistant Professor of Machine Mechanics at the Department of Industrial Engineering of the University of Padova. His research interests include human–robot interaction (rehabilitation robotics and industrial collaboration), flexible assembly systems, and cable-direct-driven robots.