CASPR: A Comprehensive Cable-Robot Analysis and Simulation Platform for the Research of Cable-Driven Parallel Robots

Darwin Lau[1], Jonathan Eden[2], Ying Tan[2], Denny Oetomo[2]

***Abstract*— The study of cable-driven parallel robots (*CDPRs*) has attracted much attention in recent years. However, to the best of the authors' knowledge, no single software platform exists for researchers to perform different types of analyses for CDPRs of arbitrary structure. In this paper, the Cable-robot Analysis and Simulation Platform for Research (*CASPR*) of CDPRs is introduced. Using this platform, arbitrarily types and structures of CDPRs, such as single and multi-link CDPRs, can be studied for a wide range of analyses, including kinematics, dynamics, control and workspace analysis. CASPR achieves this using a general CDPR model representation and an abstracted software architecture. Moveover, CDPRs can be defined using Extensible Markup Language (*XML*) with out-of-the-box availability of an extensive range of robots and analysis tools. The open-source platform aims to provide both a communal environment for the researchers to use and add models and algorithms to. The example case studies demonstrate the potential to perform analysis on CDPRs, directly compare algorithms and conveniently add new models and analyses.**

I. INTRODUCTION

Since the first cable-driven parallel robot (*CDPR*) proposed by Landsberger [1], CDPRs have gained interest due to their unique and favourable characteristics: high payload to weight ratio, potential to span large distances, ease of reconfigurability and their bio-inspired nature. As such, CDPRs have been used for a range of interesting applications: high-speed/payload manipulation [2]–[5], search and rescue [6], exoskeletons [7] and musculoskeletal robots [8]–[10].

An important characteristic of CDPRs is that the actuating cables can only apply forces in tension (*positive cable force*). This constraint results in the need of actuation redundancy for a CDPR to produce motion in all degrees-of-freedom without requiring external wrenches [11]. These characteristics create many challenges in the modelling [8], design [12], analysis [13]–[15] and control [16], [17] of CDPRs.

In the vast number of CDPR studies, two interesting traits in the research methodology can be observed. First, developed analysis techniques can often be applied to a general class of CDPRs but are typically implemented and evaluated only on the CDPRs that are available to the research group. Second, it is difficult and time-consuming to perform different types of analyses and compare different implementations, particularly as the CDPR literature grows.

[1]D. Lau is with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong `darwinlau@mae.cuhk.edu.hk`

[2]J. Eden, Y. Tan and D. Oetomo are with the Melbourne School of Engineering, the University of Melbourne, Melbourne, Australia `jpeden@student.unimelb.edu.au`, `{yingt,doetomo}@unimelb.edu.au`

These can inhibit the advance of CDPR research in several ways: 1) new techniques are not validated on a wide range of robots, for example, CDPRs with varying number of degrees-of-freedom, number of cables and cable attachments; 2) no benchmarking algorithms exist for newly developed methods to be compared with; and 3) researchers have to *reinvent the wheel* by re-implementing existing models and algorithms to compare with the state-of-the-art.

These issues have motivated the development of software platforms to study CDPRs. For the control of CDPRs, a simulation software [18] was developed within the closed-source dynamics simulator XDE. The simulation was performed only on the CoGiRo [4] and the addition of other CDPRs is not trivial. The ARACHNIS software [19] was developed to perform workspace analysis on planar and spatial CDPRs using the capacity margin [20]. In both works, only one type of analysis can be performed, control [18] and workspace analysis [19], respectively. Furthermore, for each analysis only one algorithm was implemented. Also, out-of-the-box these software do not support arbitrary types of CDPRs, including multilink CDPRs (*MCDRs*).

Identifying these issues, the development of a *comprehensive open-source software platform* for the study of CDPRs would allow cable robotics researchers to easily validate and share their research. The platform should achieve four main objectives. First, it should contain a range of representative state-of-the-art CDPR models, including MCDRs, for researchers to test their algorithms with. Second, the software should be *comprehensive* in that it contains different types of analyses important to the study of CDPRs, including the kinematics, dynamics, workspace analysis and control. Third, it should allow direct comparison of different implementations for each type of analysis. Finally, the platform should be expansible, such that the community can actively and conveniently contribute new models and algorithms.

In this paper, the Cable-robot Analysis and Simulation Platform for Research (*CASPR*) is presented. To achieve the above objectives, the platform consists of three main elements: the CDPR model, the analysis tools, and the simulation framework. Using a generalised CDPR model [8], robots of arbitrary structure can be added using Extensible Markup Language (*XML*) and then studied. Developed in MATLAB using a modular object-oriented programming (*OOP*) paradigm, the platform allows the research community to conveniently use and contribute to it. The convenience and comprehensiveness of the platform will be demonstrated through examples of different types of analyses performed on a range of different CDPRs.

To the best of the authors' knowledge, this is the first comprehensive platform for the study of CDPRs and is open-source, easily accessible and allows the community to contribute to[1]. This paper aims to only describe fundamental elements and features of CASPR. More detailed documentation will be available with the software as it is progressively updated. The first released version will already contain a range of well-known CDPR models, including single link CDPRs such as the NIST RoboCrane [2], CoGiRo [4], IPAnema family [5], ACROBOT [21], SEGESTA [22], FAST [23], KNTU planar CDPR [24], passive spring planar CDPR [25] and the MyoArm [26]; and also the MCDRs such as the CAREX [7] and a human neck model [8]. Furthermore, the platform also contains a diverse collection of analysis algorithms from many different research groups [8], [13], [15]–[17], [19], [20], [27]–[37]. The remainder of this paper will describe the various aspects of the software that has been developed.

II. MODEL

This section describes how the model of the system kinematics and dynamics are represented in CASPR. For an m-cable n-DoF (*degrees-of-freedom*) p-link CDPR, the *generalised coordinates* can be represented by $\mathbf{q} \in \mathbb{R}^n$. The *cable space* can be described by the lengths of the cables $\mathbf{l} = [l_1 \quad \ldots \quad l_m] \in \mathbb{R}^m$ and the magnitudes of the cable forces $\mathbf{f} = [f_1 \quad \ldots \quad f_m] \in \mathbb{R}^m$, where $l_i, f_i \geq 0$ denote the length and force of cable i, respectively. The fundamental equations for the system kinematics and dynamics are

$$\dot{\mathbf{l}} = L(\mathbf{q})\dot{\mathbf{q}} \tag{1}$$

$$\begin{gathered} M(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\dot{\mathbf{q}}, \mathbf{q}) + \mathbf{G}(\mathbf{q}) + \mathbf{w}_e = -L^T(\mathbf{q})\mathbf{f} \\ \mathbf{0} \leq \mathbf{f}_{min} \leq \mathbf{f} \leq \mathbf{f}_{max} \ , \end{gathered} \tag{2}$$

where $L \in \mathbb{R}^{n\times m}$ is the cable Jacobian matrix. For the equation of motion (EoM) (2), M is the mass-inertia matrix, $\mathbf{C}$ is the centrifugal and Coriolis vector, $\mathbf{G}$ is the gravitational vector and $\mathbf{w}_e$ is the external wrench acting on the system. The vectors $\mathbf{f}_{min}$ and $\mathbf{f}_{max}$ are the minimum and maximum cable force bounds, respectively. In CASPR, the kinematics and dynamics of a CDPR are contained within the `SystemModel` object. The model can be described in two parts: the rigid bodies and the cables.

A. Rigid Body Model

The kinematics and dynamics of the system's rigid bodies are represented by the `SystemModelBodies` object. Consider the general multilink system shown in Figure 1. Frame 0 is the inertial frame and $\{F_i\}, \ i = 1, \ldots, p$ is the non-inertial frame for link i. The joint and centre of mass (*CoM*) for link i are located at points P_i and G_i, respectively.

The generalised coordinates for the MCDR is $\mathbf{q} = [\mathbf{q}_1^T \ \ldots \ \mathbf{q}_p^T]^T$, where $\mathbf{q}_i \in \mathbb{R}^{n_i}$ denotes the generalised coordinates for joint i and n_i is the joint's number of DoFs.

[1]CASPR can be downloaded at https://github.com/darwinlau/CASPR.

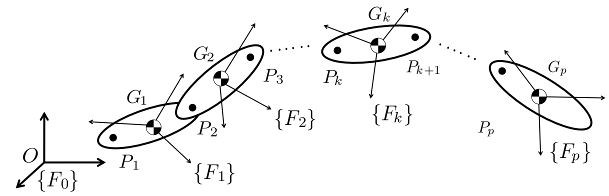


Fig. 1: General rigid body model for MCDRs, where G_i and P_i are the CoM and joint locations for link i, respectively.

1) Joints: A general representation is adopted for the robot's joints such that it is easy for new joint types to be added. All joints must inherit from the abstract `JointBase` parent class. The state of a joint is defined by $\mathbf{q}_i$, $\dot{\mathbf{q}}_i$ and $\ddot{\mathbf{q}}_i$. Every joint requires the properties $\mathbf{r}_{rel}(\mathbf{q}_i)$, $R(\mathbf{q}_i)$, $S(\mathbf{q}_i)$ and $\dot{S}(\mathbf{q}_i, \dot{\mathbf{q}}_i)$ to be defined.

- $\mathbf{r}_{rel}$: The relative translation of the joint/body.
- R: The rotation matrix between frames $\{F_{i-1}\}$ and $\{F_i\}$.
- S: The linear mapping between the relative translational velocity $\mathbf{r}'_{rel}$, relative angular velocity $\boldsymbol{\omega}_{rel}$ and $\dot{\mathbf{q}}_i$

$$\begin{bmatrix} \mathbf{r}'_{rel} \\ \boldsymbol{\omega}_{rel} \end{bmatrix} = S(\mathbf{q}_i)\dot{\mathbf{q}}_i \ , \tag{3}$$

  where $S(\mathbf{q}_i) \in \mathbb{R}^{6\times n_i}$.
- $\dot{S}(\mathbf{q}_i, \dot{\mathbf{q}}_i)$: The time derivative of the matrix S.

These properties are updated automatically when the joint's `update(q, q_dot, q_ddot)` function is called. Examples of joints that have been implemented in CASPR include:

1) *Revolute* (1 DoF): A rotation about a defined axis of rotation (e.g. `RevoluteX`)
2) *Universal* (2 DoFs): A set of two defined rotations (e.g. `UniversalXY`)
3) *Planar* (3 DoFs): Motion confined to a specified plane (e.g. `PlanarXY`)
4) *3-D Cartesian* (3 DoFs): `TranslationalXYZ`
5) *Spherical* (3 DoFs): using quaternions (`Spherical`), Euler angles (e.g. `SphericalEulerXYZ`) and fixed angles (e.g. `SphericalFixedXYZ`)
6) *Spatial* (6 DoFs): Translation and rotation of spatial bodies (`Spatial`).

Each joint defines the type of motion for a link. As such, systems with one joint are single link CDPRs (e.g. a single `PlanarXY` joint results in a planar CDPR), and MCDRs could be formed by using any combination of joints. In addition to the available joints, new joint types could be easily added to the platform by inheriting from the `JointBase` class and defining the required properties.

2) Equations of Motion: Each rigid link of the system can be described by the following properties:

- type of joint as presented in Section II-A.1
- centre of mass location defined with respect to its joint location in its local frame
- link's parent link and the location of the joint relative to the parent link (for MCDRs)

- mass of the link and the moment of inertia matrix about its centre of mass

The properties for each link are contained within `BodyModel`. Using these properties and the method proposed in [8], the LHS terms of the EoM (2), such as `M`, `C` and `G`, are computed and available to be used for analysis when the `update(...)` function of `SystemModelBodies` is called.

B. Cable Model

Like the rigid body model in Section II-A, the cable model is represented by the `SystemModelCables` object. The important elements of the cable model are the cable lengths $\mathbf{l}(\mathbf{q})$ and the Jacobian matrix $L(\mathbf{q})$. These are automatically computed when `update(...)` is called [8].

The abstract class `CableModelBase` contains all of the basic kinematics and dynamics information for a cable required for (1) and (2), such as the cable routing, cable vectors, stiffness, length l_i, force f_i, and the minimum $f_{i,min}$ and maximum $f_{i,max}$ force bounds where $f_{i,min} \leq f_i \leq f_{i,max}$. The kinematics of a cable are defined by a set of attachment locations defined by A_{ijk}, the attachment of cable i on segment j on link k. Figure 2 shows the convention for the attachment locations and cable vectors used in CASPR. This model can be used for MCDRs and single link CDPRs.

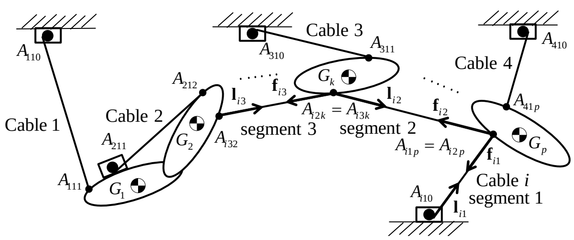


Fig. 2: Cable attachments for MCDRs, where $\mathbf{l}_{ij}$ and $\mathbf{f}_{ij}$ are the cable and force vectors of segment j of cable i, respectively [8].

The abstract base class allows different cable models to be implemented, such as rigid cables, spring or variable stiffness cables, sagging cables and physiological muscles. Inheriting from abstract base classes, the `update(...)` method for the cable kinematics and dynamics must be implemented for each specific cable model. Currently, four different types of cable models exist within the framework and demonstrate how more cable models can be constructed.

1) Ideal Cables: The ideal cable refers to the massless rigid cable model with infinite stiffness. The properties that need to be defined for this model are a set of attachment locations and the minimum and maximum force bounds for the cable.

2) Linear Spring Cables: The linear spring cable refers to a linear spring in series with the cable and/or a massless cable with linear elasticity. Two additional properties need to be defined: the spring and/or cable stiffness (K_{cable}) and the resting or nominal length of the spring (l_{s0}).

3) Variable Stiffness Cables: The variable stiffness cable refers to spring cables for which the stiffness of the spring can be adjusted. Examples of such cables include [38], [39].

4) Hill-type Muscle Cables: The Hill-type muscle model is a well accepted model of human muscles [40]. The inclusion of this model shows the potential to use CASPR and analysis for CDPRs in the study of musculoskeletal systems [33].

C. Creating CDPRs Using XML

In CASPR, a collection of well-known CDPRs has already been included to be immediately used, including: NIST RoboCrane [2], CoGiRo [4], IPAnema family [5], ACROBOT [21], SEGESTA [22], FAST [23], KNTU planar CDPR [24] and the MyoArm [26], CAREX [7]. Using XML, new CDPR models can also be added to the platform. Code Sample 1 shows an example XML file for the specifications of a single-link spatial manipulator, where the orientation is described using the xyz-Euler angle representation.

Code Sample 1: XML for example spatial robot structure.

```
<links display_range="-1.0 1.0 -1.0 1.0 -1.0 1.0">
<link_rigid num="1" name="Example Spatial CDPR">
  <joint_type>SPATIAL_EULER_XYZ</joint_type>
  <physical>
    <mass>10.0</mass>
    <com_location>0.0 0.0 0.5</com_location>
    <end_location>0.0 0.0 0.0</end_location>
    <inertia ref="com">
      <Ixx>3.0</Ixx><Iyy>3.0</Iyy><Izz>3.0</Izz>
      <Ixy>0.0</Ixy><Ixz>0.0</Ixz><Iyz>0.0</Iyz>
    </inertia>
  </physical>
  <parent>
    <num>0</num><location>0.0 0.0 0.0</location>
  </parent>
</link_rigid>
</links>
```

From the code sample, it can be seen that the defined CDPR has a mass of $m = 10$ kg, moment of inertia $I_G = \text{diag}([3.0, 3.0, 3.0])$ and CoM at ${}^1\mathbf{r}_{PG} = [0 \;\; 0 \;\; 0.5]^T$ relative from the joint location P. Code Sample 2 shows an extract from an example XML file for the CDPR cable properties.

Code Sample 2: XML code for example cable arrangements.

```
<cables default_cable_set="basic">
  <cable_set id="basic">
    <cable_ideal name="cable 1"
        attachment_reference="joint">
      <properties>
        <force_min>0</force_min>
        <force_max>80</force_max>
      </properties>
      <attachments>
        <attachment>
          <link>0</link>
          <location>-4.0 3.0 5.0</location>
        </attachment>
        <attachment>
          <link>1</link>
          <location>-0.65 0.125 0.25</location>
        </attachment>
      </attachments>
    </cable_ideal>
  </cable_set>
</cables>
```

In this example, one ideal cable is shown within a `cable_set`, with an ID of "basic". The minimum and maximum force bounds are set to be $f_{min} = 0$ N and $f_{max} = 80$ N, respectively. Two cable attachments are specified, with the first attached on link 0 (the base) at position ${}^{0}\mathbf{r}_{OA_{110}} = [-4.0 \ \ 3.0 \ \ 5.0]^T$ and the other attached on link 1 (end effector) at ${}^{1}\mathbf{r}_{P_1 A_{111}} = [-0.65 \ \ 0.125 \ \ 0.25]^T$. The `cable_set` tag allows multiple sets of cable arrangements to be defined within the same file, where the desired arrangement can be selected through its `id`. Cable sets allow users to easily switch between and perform analysis on different cable arrangement designs. Furthermore, cables of different types, such as ideal cables, linear spring and Hill-type muscle, could be defined within the same `cable_set`, allowing for more flexibility when constructing CDPRs.

In summary, this section shows how CDPR models are represented within CASPR. Several key features should be noted regarding the generic implementation. First, it allows generic joint models and arbitrary multilink serial/branched kinematic structure. Second, in addition to the already implemented joints, new joint types can be easily added. Third, more complex cable models can be easily developed. Finally, implemented joint and cable models can be used to easily construct CDPRs through XML. In CASPR, users do not need to derive the equations for the kinematics and dynamics of CDPRs as all of the terms in (1) and (2) are automatically computed when the `update(...)` function of the `SystemModel` object is called.

III. ANALYSIS

With the already developed CDPR model, users of CASPR can focus on the development and testing of CDPR analysis algorithms. This section describes the already implemented framework for CDPR analysis and the algorithms that are already available in CASPR.

A. Kinematics

1) Inverse Kinematics: The IK problem is defined as solving for the cable lengths $\mathbf{l}$ for at a pose $\mathbf{q}$

$$\mathbf{l} = \mathbf{g}(\mathbf{q}) \ , \tag{4}$$

where $\mathbf{g} : \mathbb{R}^n \to \mathbb{R}^m$ is the IK function. In CASPR, the IK (4) is computed using the approach proposed in [8] when the `update(...)` function of `SystemModel` is called. The resulting length vector $\mathbf{l}$ and its derivative $\dot{\mathbf{l}}$ are stored within the `cableLengths` and `cableLengthsDot` variables of the object, respectively.

2) Forward Kinematics: The FK problem, the inverse problem to (4), is to determine the CDPR pose $\mathbf{q}$ for a given set of cable lengths $\mathbf{l}$. This problem is non-trivial and on-going research is actively being performed on this topic. As such, the FK analysis in CASPR is implemented in a generic manner such that different FK algorithms could be easily implemented, tested and compared for different CDPR models. In CASPR, the abstract base class for FK analysis is `FKAnalysisBase`, and different FK analyses need to implement the `computeFunction(...)` function. Examples of implemented FK algorithms in CASPR include: the Jacobian pseudoinverse integration method [15] and the the non-linear least squares optimisation approach [27].

B. Dynamics

1) Forward Dynamics: The forward dynamics (*FD*) problem for CDPRs at time $t = t_k$ is to determine $\mathbf{q}(t_k)$ and $\dot{\mathbf{q}}(t_k)$ given the cable forces $\mathbf{f}(t_k)$. In CASPR, this is determined by performing double integration on the joint coordinates acceleration determined from (2)

$$\ddot{\mathbf{q}}(t_k) = M^{-1}\left(-L^T\mathbf{f} - \mathbf{C} - \mathbf{G} - \mathbf{w}_e\right) \ , \tag{5}$$

where the RHS of (5) is computed using the states $\mathbf{q}(t_{k-1})$, $\dot{\mathbf{q}}(t_{k-1})$ and $\mathbf{f}(t_k)$. Different MATLAB *ode* solvers, such as *ode45* and *ode113*, can be used to solve the FD.

2) Inverse Dynamics: The inverse dynamics (*ID*) problem is defined as the determination of positive cable forces $\mathbf{f}$, subject to constraints (2), to produce a desired trajectory $\mathbf{q}$, $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$. Due to the actuation redundancy of CDPRs, the ID problem has attracted great interest from the research community. Similar to the FK analysis in Section III-A.2, the abstract base class `IDSolverBase` is provided for different ID analyses to inherit from. The ID algorithm is implemented in the `resolveFunction(...)`. To illustrate how different ID algorithms can be implemented, the following ID approaches are already implemented in CASPR:

- `IDSolverLinProg`[2]: Solves the ID as a *linear program* (*LP*) optimisation problem [8]
- `IDSolverQuadProg`[2]: Solves the ID as a *quadratic program* (*QP*) optimisation problem [8]
- `IDSolverMinInfNorm`[2]: Solves for the minimum *infinity norm* of a vector function [28]
- `IDSolverOptimallySafe`: Solves for the *optimally safe* cable force distribution [29]
- `IDSolverClosedForm`: A computationally efficient method to solve for the closed-form solution [30], [31]
- `IDSolverFeasiblePolygon`: A versatile ID algorithm for n-DoF CDPRs actuated by $n+2$-cables [32]

Some of these formulations can be regarded as *generic*[2], where the ID problem is described as an optimisation problem with an *objective* and subject to *constraints*. Example objective functions include: the family of linear ($L(\mathbf{f}) = \mathbf{b}^T\mathbf{f}$) and quadratic ($Q(\mathbf{f}) = \mathbf{f}^T A\mathbf{f} + \mathbf{b}^T\mathbf{f}$) functions of cable forces [8], the minimisation of interaction wrenches for MCDRs [13] and optimally safe cable forces [28].

C. Control

The control of CDPRs refers to the generation of positive cable forces, subject to the force constraints (2), such that the CDPR can track a reference joint space trajectory $\mathbf{q}_{ref}$. Different control algorithms can be implemented and tested by inheriting from the abstract base class `ControllerBase`. The method `executeFunction(...)` must be implemented, where it accepts the current CDPR state $\mathbf{q}$, $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$

[2]Generic ID formulations constructed by *objectives* and *constraints*

and the reference trajectory $\mathbf{q}_{ref}$, $\dot{\mathbf{q}}_{ref}$ and $\ddot{\mathbf{q}}_{ref}$ and determines the cable forces $\mathbf{f}$. Example controllers implemented in CASPR include: the *computed torque control* approach [16] and the Lyapunov function based controller proposed in [17].

D. Workspace

Compared with traditional serial and rigid link parallel robots, the workspace analysis of CDPRs is more complex due to the actuation redundancy and positive force constraints. As such, CDPR workspace analysis is typically performed at a kinetics level and different CDPR specific workspace conditions and metrics have been proposed. This section describes the workspace analysis architecture in CASPR and the approaches that are already implemented.

1) Workspace Conditions: Workspace conditions refer to a binary requirement that defines the capability of a CDPR. One of the most generic workspace definitions is the *wrench-feasible condition* (*WFC*), defined as the ability to produce a given set of wrenches $\mathbf{w} \in \mathcal{B}$ for cable force bounds $\mathbf{f} \in [\mathbf{f}_{min}, \mathbf{f}_{max}]$ and can be mathematically expressed as

$$WFC(\mathbf{q}) \leftrightarrow \forall \mathbf{w} \in \mathcal{B}\ \exists \mathbf{f} \in [\ \overline{\mathbf{f}}, \underline{\mathbf{f}}\ ] : \mathbf{w} = -L^T(\mathbf{q})\mathbf{f}\ , \quad (6)$$

where $\overline{\mathbf{f}} = \mathbf{f}_{min}$ and $\underline{\mathbf{f}} = \mathbf{f}_{max}$. Other workspace conditions include the *static equilibrium condition* (*SEC*), the ability to resist against the gravity force in equilibrium when $\mathcal{B} = \mathbf{G}(\mathbf{q})$, and the *wrench-closure condition* (*WCC*), the ability to produce arbitrary wrench $\mathcal{B} = \mathbb{R}^m$ given unbounded positive cable forces of $\mathbf{f}_{min} = \mathbf{0}$ and $\mathbf{f}_{max} = \boldsymbol{\infty}$.

The abstract base class `WorkspaceConditionBase` is provided for different workspace conditions to be inherited. The `evaluateFunction(...)` should be implemented to check if the particular workspace condition is satisfied. In the current version of CASPR, different algorithms and implementations of the SC [19], [33], WCC [19], [34]–[36] and WFC [19] have been included.

2) Workspace Metrics: Compared with workspace conditions that are binary in nature, workspace metrics provide a measure of quality at different poses. These qualities are useful in the design and synthesis of CDPRs. Like the workspace conditions, the abstract class `WorkspaceMetricBase` is provided for different workspace metrics to be inherited. The `evaluateFunction(...)` should be implemented and calculates the value of the measure. The following metrics have already been implemented:

- `CapacityMarginMetric`: Measures the robustness in producing a given workspace condition [20]
- `TensionFactorMetric`: Measures the relative tension distribution of the cables at a specific pose [36]
- `UnilateralDexterityMetric`: Approximates the dexterity given the unilateral force constraint [37]
- `SEACM`: measures the capability to accelerate from static equilibrium [41]

In summary, it can be observed that the framework for CDPR analysis in CASPR allows different algorithms to be easily implemented. This is achieved by developing abstract base classes that only require the computational method particular to the algorithm to be implemented.

IV. SIMULATION AND INTERFACE

In Sections II and III, the models and analyses within CASPR, respectively, were presented. This section describes the simulation and interface to interact with the software.

A. Simulators

The analysis approaches presented in Section III all perform the analysis for either one joint pose $\mathbf{q}$ or instance in time. *Simulators* in CASPR allow the analysis approaches to be performed over a trajectory or a set of joint poses. For example, the `run(...)` function of the `ForwardKinematicsSimulator` object accepts a trajectory of cable lengths $\mathbf{l}(t)$ and its derivative $\dot{\mathbf{l}}(t)$, performs the FK analysis on the entire trajectory and stores the resulting joint space trajectory $\mathbf{q}(t)$. The FK analysis method is specified when constructing the simulator object and is stored as the `FKSolver` variable. In addition to removing the need for users to implement the use of the analysis methods themselves, the simulators also provide useful functionalities such as the recording of information and plotting of results. Currently, simulators have already been created for the IK, FK, ID, FD, control and workspace analysis of CDPRs.

B. High-Level Interface

The high-level interface is responsible for specifying the CDPR model, simulator and analysis algorithm to simulate and record the results. Two different high-level interfaces to interact with CASPR will be described: MATLAB scripts and the graphical user interface (*GUI*).

1) Scripts: Using MATLAB scripts, different simulations can be created in a flexible manner. Code Sample 3 shows an example script to simulate the ID of a planar CDPR.

Code Sample 3: Script to run an inverse dynamics simulation.

```
% Step 1: Setup the KNTU planar CDPR model
model_type = ModelConfigType.M_KNTU_PLANAR;
model_config = ModelConfig(model_type);
% Step 2: Create the model objects from XML
modelObj = model_config.getModel('set1');
% Step 3: Create the input trajectory from XML
traj = model_config.getTrajectory('traj1');
% Step 4: Create the IDSolver
id_type = ID_CF_SolverType.IMPROVED_CLOSED_FORM;
solver = IDSolverClosedForm(modelObj, id_type);
% Step 5: Create and run InverseDynamicsSimulator
sim = InverseDynamicsSimulator(modelObj, solver);
sim.run(traj);
```

In the script, the planar CDPR with a `PlanarXY` joint from the `ModelConfigType` library is first created and the cable set `'set1'` and the trajectory `'traj1'` from the XML configuration files will be loaded. The ID solver will use the `IDSolverClosedForm` with the improved closed form method [30]. The ID simulator is then constructed using the model and the ID solver. Finally, the `run(...)` command will simulate the ID for each point of the trajectory.

2) GUI Interface: In addition to scripts, the GUI is a more user friendly method to interact with the models and analysis tools in CASPR. Figure 3 shows the main window for the GUI with the IPAnema2 robot [5] selected.

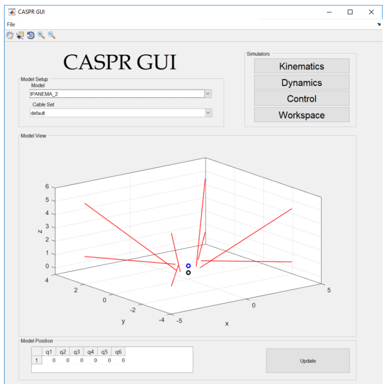


Fig. 3: The main GUI window for CASPR, the red lines corresponds to the cables of the CDPR.

The main window allows a robot to be selected from the XML library of CDPRs. After selecting the CDPR, the cable arrangement set can then be chosen. For the specified CDPR and cable arrangement, the type of analysis can be chosen and executed. The GUI also shows the resulting figures, plots and graphs. Furthermore, users could add their developed analysis tools to the GUI by modifying XML scripts that hold the GUI settings.

V. Case Studies

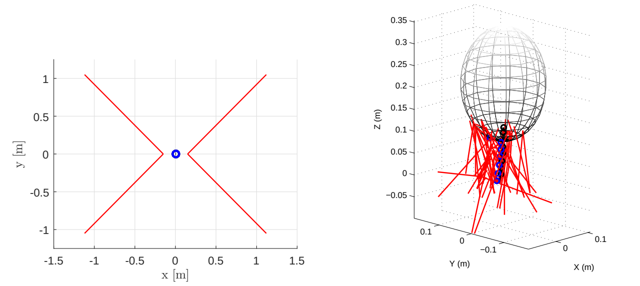
Two case studies are presented to illustrate the ease and potential in performing simulations using CASPR. CASPR has also been used in the generation of results for [10], [13], [33], [41].

A. Inverse Dynamics

This case study illustrates two different capabilities of the platform: 1) to run different ID algorithms to directly compare them; and 2) to use the same algorithm implementations on different CDPRs. Figure 4 shows two different CDPRs used within the ID case study, the single link 3-DoF 4-cable KNTU planar CDPR [24] and the 8-link 24-DoF 76-cable neck-inspired MCDR [8]. In all of the simulations, the minimum and maximum cable force bounds for all cables were $f_{min} = 0.001$ N and $f_{max} = 60$ N, respectively.

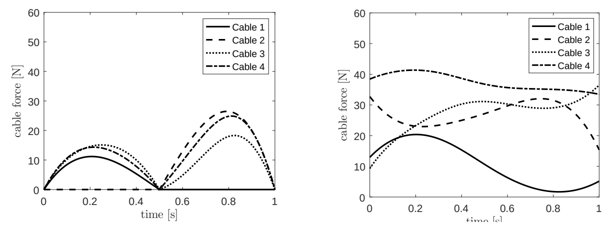
For the KNTU CDPR, three different ID algorithms presented in Section III-B.2 were simulated using the `InverseDynamicsSimulator`. The trajectory used for the robot was a quintic interpolated spline between the coordinates $\mathbf{q}_s = [0.3\ \ 0.8\ \ 0.1]^T$ and $\mathbf{q}_e = [0.7\ \ 0.2\ \ -0.2]^T$, where $\mathbf{q} = [x \quad y \quad \theta]^T$ represents the x and y position and the orientation of the robot, respectively, for zero initial and ending velocities and accelerations. Figure 5 shows the resulting cable force solutions to perform the trajectory.

Figure 5(a) shows the solution obtained from using the ID solvers `IDSolverQuadProg` [8] and `IDSolverMinInfNorm` [28], to minimise the two and



(a) KNTU planar robot [24] (b) 8-link 8S 76-cable model [8]

Fig. 4: Two different CDPRs used in the ID case study.



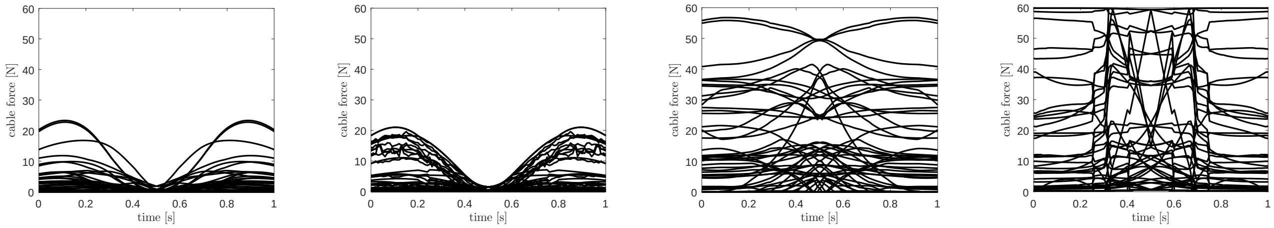
(a) Minimum infinity and two-norm (b) Improved closed-form ID [30]

Fig. 5: Cable force solutions for the inverse dynamics of the KNTU planar CDPR.

infinite norms of the cable forces, respectively. It should be noted that both methods resulted in the same solution for this problem. Figure 5(b) is the solution obtained using the `IDSolverClosedForm` [30] (as shown in Sample Code 3) for the same trajectory. To change between different solvers, only the solver definition in step 4 of Sample Code 3 needs to be modified.

Figure 6 shows the cable force solutions for the inverse dynamics performed on the 8-link neck-inspired MCDR. To change the model from the KNTU to the neck-inspired model, only step 1 of Sample Code 3 needs to be modified, while keeping the remaining steps the same. The desired trajectory is the roll motion, that is, a left and right tilting movement of the neck, as simulated in [8]. Figures 6(a), 6(b) and 6(c) show the results using the minimum sum squared, minimum infinity norm and the closed form method from the KNTU example, respectively. Figure 6(d) is obtained using the `IDSolverOptimallySafe` method [29].

Using CASPR, the characteristics of different ID solvers can be easily observed and compared, as shown in Figures 5 and 6. For example, it can be observed in Figures 6(b) and 6(d) that discontinuities exist in the solution since both methods result in a Linear Program (LP). Furthermore, the difference in the solution due to the use of different objective functions can be observed. For example, the minimum sum squared and infinity norm solutions result in significantly lower cable forces than closed-form approach that tries to achieve cable forces that are in the middle of the force bounds.

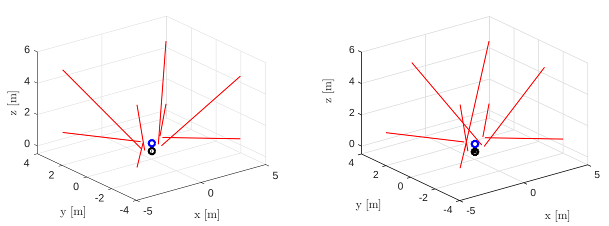


(a) Minimum sum squared (b) Minimum infinity norm (c) Improved closed-form [30] (d) Optimally safe form [29]

Fig. 6: Cable force solutions for the inverse dynamics of the neck-inspired MCDR.

B. Workspace Analysis

In the second case study, the wrench-closure workspace (*WCW*) analysis using the `WorkspaceSimulator` is performed on the 6-DoF 8-cable IPAnema2 CDPR [5]. From the script in Sample Code 3, workspace analysis on CDPRs could be performed by simply modifying steps 4 and 5 to perform WCW analysis. In this study, different WCW analyses are performed on two different arrangements of cable attachments, as shown in Figure 7. The arrangements are defined as two `cable_set` sets within the XML file, and simulations on different cable sets can be performed just by modifying step 2 of Code Sample 3.

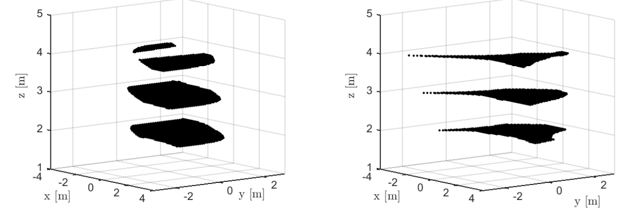


(a) Cable set 1: as defined in [5] (b) Set 2: modified arrangement

Fig. 7: The IPAnema2 for the workspace analysis study.

The resulting WCW in the XYZ coordinates (translational) for the two different cable arrangements at a constant orientation of $\theta_x = 30°$, $\theta_y = -30°$ and $\theta_z = 5°$ (using xyz-Euler angles) are shown in Figure 8. As expected, it can be observed that different cable arrangements would result in a very significant workspace. This case study serves to show the ease in which the analysis of different CDPRs and cable arrangements, can be performed in CASPR.

Furthermore, the support of multiple implementations for the same type of analysis, such as WCW, allows the performance of different algorithms to be directly compared on the same CDPR using the same computer. For this case study, four different WCW implementations that are already available within CASPR were executed on the IPAnema2. Table I shows the resulting computational time required to generate the WCW shown in Figure 8(a) for a constant orientation for $Z = 2, 3, 4$ m and $X \in [-4, 4], Y \in [-3, 3]$ with $\Delta X = \Delta Y = 0.1$ m using a numerical approach. All runs were performed using MATLAB 2014b on the same computer with an Intel® Core™ i7-4770 3.4 Ghz processor and 8 GB RAM.



(a) Cable set 1 (b) Cable set 2

Fig. 8: WCW in XYZ for the IPAnema2 at an orientation of $\theta_x = 30°$, $\theta_y = -30°$ and $\theta_z = 5°$ (xyz-Euler angles).

TABLE I: Computational costs for the different inverse dynamics problems and examples

	Methods			
	From [36][3]	From [35]	From [34]	QP method[4]
Total time[5] (s)	454.89	94.61	89.38	133.57

VI. CONCLUSION

A high-level overview of the architecture and features of the CASPR simulation platform for CDPRs was presented. The platform aims to address the lacking of a comprehensive CDPR analysis software by achieving the four main objectives as described in the introduction. Out-of-the-box, a wide range of state-of-the-art CDPRs and algorithms are already available. Furthermore, the modular design of CASPR makes it convenient to develop new models and analysis algorithms, without needing to *re-invent the wheel*. The presented case studies illustrate the usefulness and flexibility in performing analysis and comparison on different CDPRs and algorithms. As an open-source platform, it is hoped that the community would be able to both benefit from CASPR while contributing their work for others to use. Accordingly, future work for CASPR will look to increase the types of analyses provided by adding new CDPR models such as sagging cable models. It is also desired that through community engagement, the coverage of CDPR research available in CASPR is increased.

[3] Generated through computation of the tension factor metric.

[4] This method uses a QP solver to find a valid solution to the problem $\mathbf{0} = -L^T\mathbf{f}$, $\mathbf{f} \geq \mathbf{0}$ under the condition that L^T is of full rank.

[5] All times are implementation and computer dependent.

REFERENCES

[1] S. E. Landsberger and T. B. Sheridan, "A new design for parallel link manipulator," in *Proc. IEEE Int. Conf. Syst. Man. Cybern.*, 1985, pp. 812–814.

[2] J. S. Albus, R. V. Bostelman, and N. Dagalakis, "The NIST robocrane," *J. Robot. Syst.*, vol. 10, no. 5, pp. 709–724, 1993.

[3] S. Kawamura, W. Choe, S. Tanaka, and S. R. Pandian, "Development of an ultrahigh speed robot FALCON using wire drive system," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1995, pp. 215–220.

[4] J. Lamaury and M. Gouttefarde, "Control of a large redundantly actuated cable-suspended parallel robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013, pp. 4659–4664.

[5] A. Pott, H. Mütherich, W. Kraus, V. Schmidt, P. Miermeister, and A. Verl, "Ipanema: A family of cable-driven parallel robots for industrial applications," in *Cable-Driven Parallel Robots*, ser. Mechanisms and Machine Science, A. Pott and T. Bruckmann, Eds. Springer International Publishing, vol. 12, pp. 119–134.

[6] J.-P. Merlet and D. Daney, "A portable, modular parallel wire crane for rescue operations," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2010, pp. 2834–2839.

[7] Y. Mao and S. K. Agrawal, "Design of a cable-driven arm exoskeleton (CAREX) for neural rehabilitation," *IEEE Trans. Robot.*, vol. 28, no. 4, pp. 922–931, 2012.

[8] D. Lau, D. Oetomo, and S. K. Halgamuge, "Generalized modeling of multilink cable-driven manipulators with arbitrary routing using the cable-routing matrix," *IEEE Trans. Robot.*, vol. 29, no. 5, pp. 1102–1113, 2013.

[9] T. Kozuki, H. Mizoguchi, Y. Asano, M. Osada, T. Shirai, J. Urata, Y. Nakanishi, K. Okada, and M. Inaba, "Design methodology for thorax and shoulder of human mimetic musculoskeletal humanoid kenshiro - a thorax with rib like surface -," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2012, pp. 3687–3692.

[10] D. Lau, J. Eden, S. Halgamuge, and D. Oetomo, "Cable function analysis for the musculoskeletal static workspace of a human shoulder," in *Cable-Driven Parallel Robots*, ser. Mechanisms and Machine Science, A. Pott and T. Bruckmann, Eds. Springer International Publishing, vol. 32, pp. 263–274.

[11] A. Ming and T. Higuchi, "Study on multiple degrees-of-freedom positioning mechanism using wires (part 1) - concept, design and control," *Int. J. Japan Social Eng.*, vol. 28, no. 2, pp. 131–138, 1994.

[12] D. Lau, K. Bhalerao, D. Oetomo, and S. K. Halgamuge, "On the task specific evaluation and optimisation of cable-driven manipulators," in *Advances in Reconfigurable Mechanisms and Robots I*, J. S. Dai, M. Zoppi, and X. Kong, Eds. Springer London, 2012, ch. 63, pp. 707–716.

[13] D. Lau, D. Oetomo, and S. K. Halgamuge, "Inverse dynamics of multilink cable-driven manipulators with the consideration of joint interaction forces and moments," *IEEE Trans. Robot.*, vol. 31, no. 2, pp. 479–488, 2015.

[14] ——, "Wrench-closure workspace generation for cable driven parallel manipulators using a hybrid analytical-numerical approach," *J. Mech. Des.*, vol. 133, no. 7, pp. 071 004/1–7, 2011.

[15] T. Bruckmann, L. Mikelsons, T. Brandt, M. Hiller, and D. Schramm, "Wire robots part I: Kinematics, analysis & design," in *Parallel Manipulators New Developments*, ser. ARS Robotic Books. I-Tech Education and Publishing, 2008.

[16] R. L. Williams II, M. Xin, and P. Bosscher, "Contour-crafting-cartesian-cable robot system: Dynamics and controller design," in *DETC 2008: 32nd Annual Mechanisms and Robotics Conference, Vol. 2, Parts A & B*, 2009, pp. 39–45.

[17] A. B. Alp and S. K. Agrawal, "Cable suspended robots: Design, planning and control," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2002, pp. 4275–4280.

[18] M. Michelin, C. Baradat, D. Q. Nguyen, and M. Gouttefarde, "Simulation and control with XDE and matlab/simulink of a cable-driven parallel robot (CoGiRo)," in *Cable-Driven Parallel Robots*, ser. Mechanisms and Machine Science, A. Pott and T. Bruckmann, Eds. Springer International Publishing, vol. 32, pp. 71–83.

[19] 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*, ser. Mechanisms and Machine Science, A. Pott and T. Bruckmann, Eds. Springer International Publishing, vol. 32, pp. 293–305.

[20] F. Guay, P. Cardou, A. L. Cruz-Ruiz, and S. Caro, "Measuring how well a structure supports varying external wrenches," in *New Advances in Mechanisms, Transmissions and Applications: Proceedings of the Second Conference MeTrApp 2013*, V. Petuya, C. Pinto, and E.-C. Lovasz, Eds. Springer Netherlands, vol. 12, pp. 385–392.

[21] L. Gagliardini, S. Caro, M. Gouttefarde, and A. Girin, "A reconfiguration strategy for reconfigurable cable-driven parallel robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 1613–1620.

[22] C. Reichert and T. Bruckman, "Unified contact force control approach for cable-driven parallel robots using an impedance/admittance control strategy," in *IFToMM World Congress*, 2015, pp. 645–654.

[23] 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.

[24] R. Babaghasabha, M. A. Khosravi, and H. D. Taghirad, "Adaptive control of KNTU planar cable-driven parallel robot with uncertainties in dynamic and kinematic parameters," in *Cable-Driven Parallel Robots*, ser. Mechanisms and Machine Science, A. Pott and T. Bruckmann, Eds. Springer International Publishing, 2015, vol. 32, pp. 145–159.

[25] Q. Duan, V. Vashita, and S. K. Agrawal, "Effect on wrench-feasible workspace of cable-driven parallel robots by adding springs," *Mech. Mach. Theory*, vol. 86, pp. 201–210, 2015.

[26] C. Richter, R. H. Sören Jentzsch, J. A. Garrido, E. Ros, A. Knoll, F. Röhrbein, P. van der Smagt, and J. Conradt, "Scalability in neural control of musculoskeletal robots," *IEEE Robot. Autom. Mag.*, 2016.

[27] A. Pott and V. Schmidt, "On the forward kinematics of cable-driven parallel robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2015, pp. 3182–3187.

[28] C. Gosselin and M. Grenier, "On the determination of the force distribution in overconstrained cable-driven parallel mechanisms," *Meccanica*, vol. 46, no. 1, pp. 3–15, 2011.

[29] P. H. Borgstrom, B. L. Jordan, G. S. Sukhatme, M. A. Batalin, and W. J. Kaiser, "Rapid computation of optimally safe tension distributions for parallel cable-driven robots," *IEEE Trans. Robot.*, vol. 25, no. 6, pp. 1271–1281, 2009.

[30] A. Pott, "An improved force distribution algorithm for over-constrained cable-driven parallel robots," in *Computational Kinematics*. Springer Netherlands, 2014, pp. 139–146.

[31] K. Müller, C. Reichert, and T. Bruckmann, "Analysis of a real-time capable cable force computation method," in *Cable-Driven Parallel Robots*, ser. Mechanisms and Machine Science, A. Pott and T. Bruckmann, Eds. Springer International Publishing, 2015, vol. 32, pp. 227–238.

[32] 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 Trans. Robot.*, vol. 31, no. 6, pp. 1444–1457, 2015.

[33] D. Lau, J. Eden, D. Oetomo, and S. K. Halgamuge, "Musculoskeletal static workspace of the human shoulder as a cable-driven robot," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 2, pp. 978–984, 2015.

[34] B. Ouyang and W.-W. Shang, "A new computation method for the force-closure workspace of cable-driven parallel manipulators," *Robotica*, vol. 33, no. 3, pp. 537–547, 2015.

[35] W. B. Lim, G. Yang, S. H. Yeo, and S. K. Mustafa, "A generic force-closure analysis algorithm for cable-driven parallel manipulators," *Mech. Mach. Theory*, vol. 46, no. 9, pp. 1265–1275, 2011.

[36] C. B. Pham, S. H. Yeo, G. Yang, and I.-M. Chen, "Workspace analysis of fully restrained cable-driven manipulators," *Robot. Auton. Syst.*, vol. 57, no. 9, pp. 901–912, 2009.

[37] R. Kurtz and V. Hayward, "Dexterity measures with unilateral actuation constraints: the n+1 case," *Adv. Robot.*, vol. 9, no. 5, pp. 561–577, 1994.

[38] S. Yeo, G. Yank, and W. Lim, "Design and analysis of cable-driven manipulators with variable stiffness," *Mech. Mach. Theory*, vol. 69, pp. 230–244, 2013.

[39] K. Yang, G. Yang, J. Wang, T. Zheng, and W. Yang, "Design analysis of a 3-dof cable-driven variable-stiffness joint module," in *IEEE Int. Conf. Robot. and Biomimetics*. IEEE, 2015, pp. 529–534.

[40] F. E. Zajac, "Muscle and tendon: Properties, models, scaling, and application to biomechanics and motor control," *Crit. Rev. in Biomed. Eng.*, vol. 17, no. 4, pp. 359–411, 1989.

[41] J. Eden, D. Lau, Y. Tan, and D. Oetomo, "Available acceleration set for the study of motion capabilities for cable robots," *Mech. Mach. Theory*, vol. 105, pp. 320–336, 2016.