2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

STORM: Screw Theory Toolbox For Robot Manipulator and Mechanisms

Keerthi Sagar*, Vishal Ramadoss*, Dimiter Zlatanov and Matteo Zoppi

***Abstract*— Screw theory is a powerful mathematical tool for the kinematic analysis of mechanisms and has become a cornerstone of modern kinematics. Although screw theory has rooted itself as a core concept, there is a lack of generic software tools for visualization of the geometric pattern of the screw elements. This paper presents STORM, an educational and research oriented framework for analysis and visualization of reciprocal screw systems for a class of robot manipulator and mechanisms. This platform has been developed as a way to bridge the gap between theory and practice of application of screw theory in the constraint and motion analysis for robot mechanisms. STORM utilizes an abstracted software architecture that enables the user to study different structures of robot manipulators. The example case studies demonstrate the potential to perform analysis on mechanisms, visualize the screw entities and conveniently add new models and analyses.**

I. INTRODUCTION

Machine theory, design and kinematic analysis of robots, rigid body dynamics and geometric mechanics– all have a common denominator, namely screw theory which lays the mathematical foundation in a geometric perspective. The general motion of a rigid body based on the concept of screw axis was described by Giulio Mozzi and was reviewed in [1]. Based on Plücker's investigation regarding the geometry of lines, the framework of theory of screws and screw systems were formalized by Ball [2]. A comprehensive classification of screw subspaces was described in detail by Hunt in [3], [4], [5]. The fundamentals of theory of screws were revisited in [6], [7].

The theory of screws plays a pivotal role in the analysis of instantaneous kinematics of mechanisms. In robotic mechanisms, the instantaneous motion of a rigid body and the statics of the body are represented by a subspace of the six-dimensional vector space of twists or wrenches. These linear subspaces are denoted as screw systems. A screw can be used to denote the position and orientation of a spatial vector, the linear and angular velocity of a rigid body, or a force and a couple, respectively. Many frontier problems in mechanism theory [8], [9], [10], [11], [12] can be solved elegantly using screw theory due to the clear geometrical concept [13], simple expression and explicit physical meaning.

Due to the comprehensive classification of screw systems, it is quite often difficult to represent and visualize the image in Cartesian three-space. It is crucial for designers to understand the geometric pattern of the underlying screw system defined in the mechanism or a robot and to have a visual reference of them. Existing frameworks [14], [15], [16] target the design of kinematic mechanisms from computational point of view with position, velocity kinematics and identification of different assembly configurations. A practical

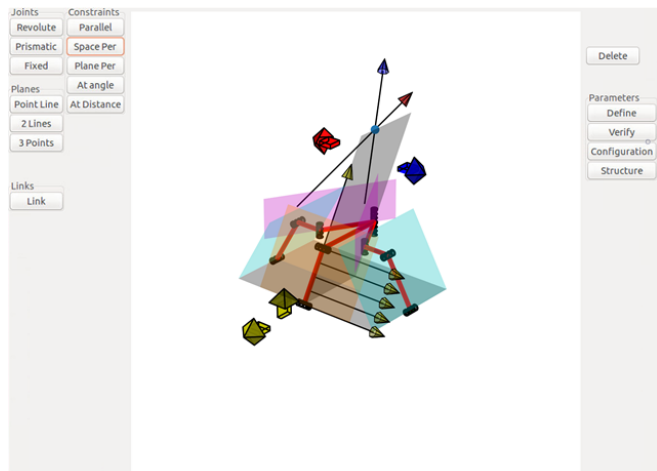


Fig. 1: A simple GUI window for STORM, the zero pitch screws are shown with yellow, red and blue arrowheads and the infinite pitch screws are shown with pyramid arrowheads

framework for representation of geometrical problems was described in [17]. In [16], the kinematics with closed-form geometric algorithms targeting anthropomorphic manipulators was addressed. An approach based on computational kinematics was proposed in [14] which supports visualization of the screw system for any manipulator topology. The software accompanying the textbook [18] elucidates the Lie group representations and serves as a pedagogical tool. However, to the best of the authors' knowledge, no single software platform has been developed to support screw system visualization and reciprocal screw identification for mechanisms of arbitrary type and structure.

There is a need for a software that can complement the theory [19], [20], [18], [21] by providing visualization and simulation for the theoretical bases of the mechanism and robot analysis using screw algebra. This is the motivation behind the development of the Screw Theory Toolbox For Robot Manipulator and Mechanisms (STORM). This paper is developed to address essential elements and features of STORM. A comprehensive documentation will be available with the toolbox as it is progressively updated.

* Equal contribution. K. Sagar, V. Ramadoss, M. Zoppi and D. Zlatanov are with PMAR Robotics Group, Department of Mechanical, Energy, Management and Transportation Engineering, University of Genoa, Genoa (GE), 16145, Italy. Email: {keerthi, zlatanov, zoppi}@dimec.unige.it, {vishal.ramadoss}@edu.unige.it

The application of a wrench on a twist (also called their reciprocal product [4]) measures the power exerted by the system of forces for the instantaneous motion. When a wrench exerts no power on a twist, they are called reciprocal. The projective space underlying a twist or wrench subspace is called a screw system [22], [23]. An n-system underlies an n-dimensional subspace. The reciprocity relation helps the designer to obtain the motion pattern [24], [25], [26], [27] from the related constraint counterpart and it is important to find the reciprocal screw system for a given one. Existing methods for the determination of reciprocal screws can be classified as either algebraically [28], [29], [30], [31] or geometrically [32] based approaches.

The contributions of this work can be summarized as follows:

- unified software framework for different manipulator topologies (i.e, serial, parallel, hybrid manipulators, cable robots);
- a combination of algebraic and geometric methods for determination of screw systems;
- representation of screw systems with a common geometric set;
- incorporation of result visualization for kinematic analysis;
- quick observation of the motion pattern and capabilities of the designed mechanism architecture;

In what follows, we first illustrate the STORM software architecture, including the overall workflow and key algorithms in Section II. Section III introduces four use cases as examples to verify the usability of the software. The concluding remarks are drawn in Section IV.

II. THE STORM FRAMEWORK

For screw system visualization, the STORM is based on Python 2.7. The high-level overview of the software environment is shown in Fig. 2. The main design paradigm in STORM is in the implementation of four core program issues is possible.

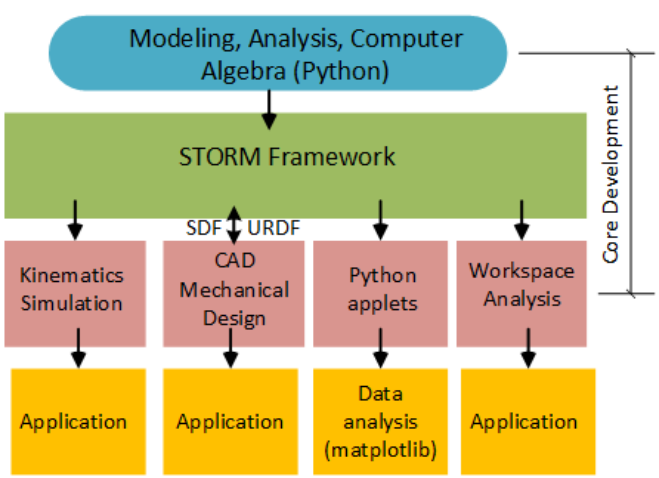


Fig. 2: STORM software environment

The mathematics and algorithms are encapsulated in core library STORMlib. The results and states are stored in XML files wherever applicable. Python language enables the data transfer to be managed via a top level object through bindings. In this way, the computation, configuration and coordination, 3'Cs are taken care in the software implementation. Interfacing between custom and third party tools are possible by providing Simulation Description Format (SDF), Universal Robot Description Format (URDF), Standard Tessellation Language (STL) file formats to the toolbox and to compute the results for a desired architecture. The bi-directional arrow shown in the Fig. 2 is to indicate that the model descriptions obtained from the 3D computer-aided design (CAD) software are imported into the STORM framework enabling visualization of screw systems and kinematic analyses.

The STORM framework incorporates a generic implementation by separating the data model and algorithms. The robot geometry and configuration is defined using XML thereby allowing generic joint models and different kinematic structures. The main features of STORMlib are the base data model for robot geometry, loading and saving robots, algorithm for reciprocal screw determination and handling computer algebra for kinematic analysis. The computed results are exported as Comma Separated Values (CSV) files, which stores the parameters from kinematic analysis.

A user-friendly Graphical User Interface (GUI) is provided for the designer, which enables user to utilize the already contained well-known mechanisms and robot models and also allow the user to build interactively the kinematic chain of robot from XML-based library of joints and rigid bodies to output the screws and their geometric parameters. This enables users to effectively verify their theoretical model by visualizing the configuration and generating its corresponding screw elements. This modular framework consists of four major parts: i) System model, ii) Determination of reciprocal screws, iii) Screw system visualization, iv) Motion pattern. The various aspects of the framework and its associated GUI are illustrated below.

A. Model

This section describes how the primitives of the system are represented in STORM. The basic screw elements and their physical meanings are shown in Tab. I. The corresponding definitions are as follows: line with curved arrows denotes zero pitch screw, line with pyramid arrows denotes infinite pitch screw; line with curved arrows and transparent color denotes finite pitch screw; red color denotes constraint; black color denotes freedom/ twist. These five basic elements can be used to describe an overview of the motion and constraint of a mechanism.

B. Geometric approach to determine reciprocal screws

Definition 1. *A line l with a pitch h is a geometric element called a* ***screw***[4]. A screw (geometric) is not a vector. The screws form the projective space underlying the space of twists and wrenches.

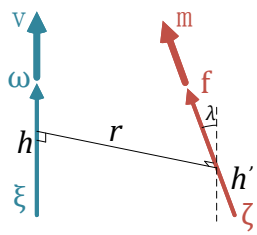


Fig. 3: Reciprocal screws

In general, a screw $\$$ can be represented as:

$$\$ = \begin{bmatrix} \boldsymbol{s} \\ \boldsymbol{r} \times \boldsymbol{s} + h\boldsymbol{s} \end{bmatrix} \tag{1}$$

where $\boldsymbol{s}$ is the direction vector of the screw, h is the pitch of the screw and $\boldsymbol{r}$ is the position vector of any point on the screw. The canonical representation of a screw (twist/ wrench) describes the axis closest to the origin $\boldsymbol{r}$ and pitch h in the following way, where $\boldsymbol{\omega}$, $\boldsymbol{v}$ denote instantaneous angular and translational velocity respectively :

$$\boldsymbol{r} = \frac{\boldsymbol{\omega} \times \boldsymbol{v}}{\boldsymbol{\omega} \cdot \boldsymbol{\omega}}, h = \frac{\boldsymbol{\omega} \cdot \boldsymbol{v}}{\boldsymbol{\omega} \cdot \boldsymbol{\omega}}, \boldsymbol{s} = \frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|} \tag{2}$$

Similarly, the parameters are defined for a wrench with ($\boldsymbol{f}$, $\boldsymbol{m_O}$). Depending on the pitch, the following classifications of screws are defined:

1) *Zero-pitch screws*: $h = 0$, $\$_0 = \begin{bmatrix} \boldsymbol{s} \\ \boldsymbol{r} \times \boldsymbol{s} \end{bmatrix}$
2) *Infinite-pitch screws*: $h = \infty$, $\$_\infty = \begin{bmatrix} \boldsymbol{0}_{3\times1} \\ \boldsymbol{s} \end{bmatrix}$
3) *Finite-pitch screws*: $h \neq 0$, $h \neq \infty$ expressed as linear combination of zero-pitch and infinite pitch screws.

A *zero-pitch twist* and an *infinite-pitch twist* are represented as $\boldsymbol{\xi}_0$ and $\boldsymbol{\xi}_\infty$, respectively, while a *zero-pitch wrench* and an *infinite-pitch wrench* are represented as $\boldsymbol{\zeta}_0$ and $\boldsymbol{\zeta}_\infty$.

Definition 2. *Two screw systems are reciprocal when any wrench acting on a screw in one system exerts no power on any twist on a screw in the other system*[4].

In Fig. 3, a body is constrained about the instantaneous screw axis (ISA) $\boldsymbol{\xi}$ with pitch h. The pitch h satisfies the requirement $\boldsymbol{v} = h\boldsymbol{\omega}$. The screw $\boldsymbol{\zeta}$ contains a wrench with pitch h' has its intensity $\boldsymbol{f}$ and the moment $\boldsymbol{m} = h'\boldsymbol{f}$. The shortest distance between $\boldsymbol{\xi}$ and $\boldsymbol{\zeta}$ is r and the angle between them is λ.

The work done by the wrench $\boldsymbol{\zeta}$ on the twist $\boldsymbol{\xi}$ can be represented as:

$$\boldsymbol{\zeta} \circ \boldsymbol{\xi} = \boldsymbol{f} \cdot \boldsymbol{v} + \boldsymbol{m} \cdot \boldsymbol{\omega} \tag{3}$$

where, $\circ$ is the *reciprocal product* between two screws. For Fig. 3, when no work is done, the following relationship is obtained [4]:

$$\boldsymbol{\zeta} \circ \boldsymbol{\xi} = \boldsymbol{f}\boldsymbol{\omega}((h + h')\cos(\lambda) - r\sin(\lambda)) = 0 \tag{4}$$

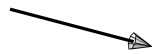
Primitive	Mathematical symbol	Physical meaning
	$\$_0$	Rotational DOF
	$\$_0$	Constraint force
	$\$_\infty$	Translational DOF
	$\$_h$	Finite pitch - Helical DOF
	$\$_\infty$	Constraint couple

TABLE I: Basic screw elements (primitives) and their physical meanings

Screw 1	Screw 2	Reciprocity condition
$\$_{10}$	$\$_{20}$	Coplanar axes
$\$_{1\infty}$	$\$_{2\infty}$	Always reciprocal
$\$_{10}$	$\$_{2\infty}$	Perpendicular axes
$\$_{1\infty}$	$\$_{20}$	Perpendicular axes

TABLE II: Reciprocity conditions

If the locations and pitches of the screws $\boldsymbol{\xi}$ and $\boldsymbol{\zeta}$ are such that,

$$\begin{aligned} (h + h')\cos(\lambda) - r\sin(\lambda) &= 0 \\ h' &= h - r\tan(\lambda) \end{aligned} \tag{5}$$

then, irrespective of the intensity of the applied wrench or the amplitude of the instantaneous twist, the contribution the wrench makes to the instantaneous working rate is zero and the screws are reciprocal. Some of the reciprocity conditions are listed in the Tab. II. The geometric conditions can be expressed explicitly by taking instantaneous motions and system of forces. Simply by utilizing the reciprocity conditions, one can obtain the reciprocal system, thereby determining the constraints and freedoms of the robot/mechanism.

C. Algebraic approach to determine reciprocal screws

In this section we describe the algorithm utilized to determine the reciprocal screws. The underlying principle is to obtain a set of homogeneous equations from the reciprocal conditions and obtain the null space of the equation. We utilize the concept of augmentation approach as established in [31] and extend the algorithm with recursive partitioning for rank deficient chains. The visual screw systems are generated algebraically. Then we validate the obtained reciprocal screws for different systems using the geometric conditions that are established in Tab. II.

1) Mathematical Background: The null space can also be derived in an efficient way as explained in [29]. The algorithm for the linear algebraic approach is based on [31]. For a mechanism with L number of legs, the number of screws pertaining to each leg L is initialized and the linearly independent screws are stored. The rank of the system is determined. To map the reciprocal screw product, a dictionary is created. An extra row to map while shifting the column vectors. The array for reciprocal screws are initialized and the partitioning of matrix into sub-matrices is performed. The sub-matrix is checked for rank deficiency conditions. A value of zero is assigned to $(6-n-1)$ elements of the reciprocal screw. The column vector is isolated to get the cofactors. Using the determinant of cofactor matrix, the elements of reciprocal screw systems are calculated. The pseudocode of the methodology is given in Algorithm 1.

The reciprocal product between a n-screw system and its $(6-n)$-screw system is given as

$$\mathbf{A}\Delta\mathbf{\$}^r = 0 \tag{6}$$

where $\mathbf{A}$ is the screw matrix of an assembly of n screws as row vectors, $\mathbf{\$}^r$ is the reciprocal screw matrix as its column vectors, and Δ is the elliptic polar operator for interchanging the primary part with the secondary part of a screw. Suppose the Plücker homogeneous coordinates of the reciprocal screw $\mathbf{\$}^r$ has the following form:

$$\mathbf{\$}^r = \begin{bmatrix} l_r & m_r & n_r & p_r & q_r & r_r \end{bmatrix}^{\mathsf{T}} \tag{7}$$

Substituting Eqn. 7 into Eqn. 6 yields

$$\begin{bmatrix} l_1 & m_1 & n_1 & p_1 & q_1 & r_1 \\ l_2 & m_2 & n_2 & p_2 & q_2 & r_2 \\ & & & \vdots & & \\ l_n & m_n & n_n & p_n & q_n & r_n \end{bmatrix} \begin{bmatrix} p_r \\ q_r \\ r_r \\ l_r \\ m_r \\ n_r \end{bmatrix} = 0 \tag{8}$$

In linear algebraic form, the problem of determining the reciprocal screw system is equivalent as obtaining the null space from matrix $\mathbf{A}$. The algebraic procedure [31] is explained in brief for the two-system of reciprocal screws for the sake of clarity. Suppose $\$_1$, $\$_2$, $\$_3$ and $\$_4$ represent the four-system of screws defining a robot. The screws are expressed in Plücker coordinates.

$$\mathbf{A} = \begin{bmatrix} l_1 & m_1 & n_1 & p_1 & q_1 & r_1 \\ l_2 & m_2 & n_2 & p_2 & q_2 & r_2 \\ l_3 & m_3 & n_3 & p_3 & q_3 & r_3 \\ l_4 & m_4 & n_4 & p_4 & q_4 & r_4 \end{bmatrix} \tag{9}$$

Two reciprocal screws can be obtained by partitioning the matrix twice. The first partition and augmentation is :

$$\mathbf{A} = \begin{bmatrix} l_1 & m_1 & n_1 & p_1 & q_1 & * \\ l_2 & m_2 & n_2 & p_2 & q_2 & * \\ l_3 & m_3 & n_3 & p_3 & q_3 & * \\ l_4 & m_4 & n_4 & p_4 & q_4 & * \\ - & - & - & - & - & - \end{bmatrix} \tag{10}$$

Assign $n_1^r = 0$, the augmented 5×5 matrix is utilized to obtain the reciprocal screws. The cofactors of augmenting row gives the elements of reciprocal screws.

$$p_1^r = \begin{vmatrix} m_1 & n_1 & p_1 & q_1 \\ m_2 & n_2 & p_2 & q_2 \\ m_3 & n_3 & p_3 & q_3 \\ m_4 & n_4 & p_4 & q_4 \end{vmatrix} q_1^r = -\begin{vmatrix} l_1 & n_1 & p_1 & q_1 \\ l_2 & n_2 & p_2 & q_2 \\ l_3 & n_3 & p_3 & q_3 \\ l_4 & n_4 & p_4 & q_4 \end{vmatrix} \tag{11}$$

$$r_1^r = \begin{vmatrix} l_1 & m_1 & p_1 & q_1 \\ l_2 & m_2 & p_2 & q_2 \\ l_3 & m_3 & p_3 & q_3 \\ l_4 & m_4 & p_4 & q_4 \end{vmatrix} l_1^r = -\begin{vmatrix} l_1 & m_1 & n_1 & q_1 \\ l_2 & m_2 & n_2 & q_2 \\ l_3 & m_3 & n_3 & q_3 \\ l_4 & m_4 & n_4 & q_4 \end{vmatrix} \tag{12}$$

$$m_1^r = \begin{vmatrix} l_1 & m_1 & n_1 & p_1 \\ l_2 & m_2 & n_2 & p_2 \\ l_3 & m_3 & n_3 & p_3 \\ l_4 & m_4 & n_4 & p_4 \end{vmatrix} \tag{13}$$

The second independent reciprocal screw is produced by shifting the partition :

$$\mathbf{A} = \begin{bmatrix} * & m_1 & n_1 & p_1 & q_1 & r_1 \\ * & m_2 & n_2 & p_2 & q_2 & r_2 \\ * & m_3 & n_3 & p_3 & q_3 & r_3 \\ * & m_4 & n_4 & p_4 & q_4 & r_4 \\ - & - & - & - & - & - \end{bmatrix} \tag{14}$$

Assign $p_2^r = 0$, the augmented 5×5 matrix is utilized to obtain the reciprocal screws. The two-system of reciprocal screws is obtained. This procedure is utilized for finding reciprocal screws from five-, four-, three- and two-systems found in robot manipulator and mechanisms.

D. Screw System Visualization

A screw system, of order n $(1 \leq n \leq 5)$, also called n-system of screws, consists of all the screws that are linearly dependent on n linearly independent screws. Any set of n linearly independent screws within a n-system forms a basis of this system. Some of the examples of typical screw systems based on line geometry are shown in Tab. III. The first screw represented in the table with dimension 1 is a zero-pitch screw and the other being infinite-pitch screw. The 2-systems comprises of planar concurrent pencil of $\mathbf{\$}_0$ screws, planar parallel pencil of $\mathbf{\$}_0$ screws and planar concurrent pencil of $\mathbf{\$}_\infty$ screws. The 3-system comprises of 3-$\mathbf{\$}_\infty$ screws orthogonal to each other as shown in the table and spatial concurrent pencil of $\mathbf{\$}_0$ screws. The 4-system is a combination of 3-$\mathbf{\$}_\infty$ orthogonal screws and 1-planar pencil of $\mathbf{\$}_0$ screws. Exhaustive list of screw systems can be found in literature [5].

E. Motion Pattern

The number of degrees of freedom defines the mobility of the mechanism. Screw theory provides a more accurate analysis of a mechanism's mobility compared to the conventional Chebychev-Grübler-Kutzbach criterion. Considering the scenario of a parallel manipulator, firstly the screw systems are identified by finding the leg constraints and

Algorithm 1: Generate Reciprocal Screws

function Generate_Screws.ScrewsReciprocal(Screws)
Input : XML - Leg/Limb architecture
Output: k linearly independent reciprocal screws (leg L)
for *Leg L= 1 to n* **do**
  Initialize no. of screws;
  Keep linearly independent screws;
  Compute rank to define *n_system* of screws;
  n_system_r = 6 - n_system; // Reciprocal
  no_r_found = n_system_r; //Screws computed
  r_screws(L) = [];
  Screw_matrix = Screws(L);
  while *Matrix_rank(r_screws(L)) < n_system_r* **do**
    sub_matrix = Screw_matrix[:, n_system+1]
    if *Matrix_rank(sub_matrix) == n_system* **then**
      r_screw = *Screw_Mat_Augment(sub_matrix,n_system, no_r_found,r_screws,Screws)*
      r_screws(L).append(r_screw)
      no_r_found = no_r_found - 1
    else
      r_screw = *Recursive_Partitioning(sub_matrix, n_system, no_r_found, r_screws(L), Screws(L))*
      r_screws(L).append(r_screw)
      no_r_found = no_r_found - 1
    Screw_matrix= shift_right(Screw_matrix)
r_screws_architecture = $\uplus_{i \in L}$ r_screws(i)
Mobility =Matrix_rank(r_screws_architecture)
return r_screws_architecture

then platform constraints. The constraints of the manipulator is the reciprocal screws that are determined using the above explained algorithm. Once the platform constraints are determined, the platform freedoms are the reciprocal system to the set of constraint wrenches. The idea is of great interest when new mechanism topologies are to be analyzed, and the framework allows to visualize the reciprocal screw system swiftly. The motion pattern may be full-cycle or instantaneous. In the software presented, the instantaneous mobility [24] is obtained using screw theory.

F. Simulation and interface

The high-level interface is responsible for specifying the mechanism, simulator and analysis algorithm to simulate and compute the results. Two possibilities of high-level interface are utilized: Pyscripts and the graphical user interface (GUI).

1) Scripts: Different simulations can be performed in a flexible manner using python scripts. An example script to determine the reciprocal screw using recursive partitioning approach is shown in code sample 1.

Code sample 1: Script to generate screws and constraint wrenches

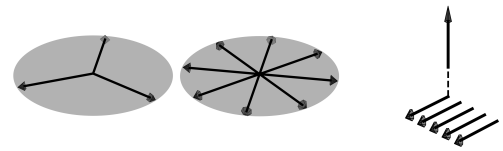
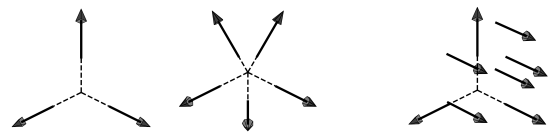
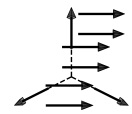
Dimension	Screw system
1	
2	
3	
4	

TABLE III: Example of typical screw systems based on line geometry

```
1: Generate and visualize screws and constraint wrenches
Create limb/leg objects based on end-effector choice
leg_obj = Generate_Screws.getLeg(mechanism_joints, mechanism_EF)
2: Plucker coordinate of screws
plucker_screws = Generate_Screws.plucker(leg_obj)
plucker_reciprocal_screws= Generate_Screws.ScrewsReciprocal(plucker_screws, leg_obj)
mechanism_screw_sim = Visualization_Build.Screw.Sim (plucker_screws,plucker_reciprocal_screws,plot_type)
```

In the script, the reconfigurable mechanism with defined joint from the model library is first created. The reciprocal screw (RS) solver will use the recursive partition method. The RS simulator is then formed using the model and the solver. Finally, the run command will simulate the mechanism and shows the corresponding screws for visualization. In the GUI, this corresponds to the configuration button.

2) GUI Interface: In addition to scripts, the GUI is interactive and user friendly for analyzing the models. Figure 1 shows the main window for the GUI with the three-fold reconfigurable mechanism. The main window allows a mechanism to be imported and defined from the library of mechanisms. The parameters of the mechanism can be verified, modified and the configuration gives the reciprocal screws of the desired architecture. The user can also define joints, planes and links using the widgets at the top. The GUI shows the defined planes and reciprocal screws as per the definition in Tab. I.

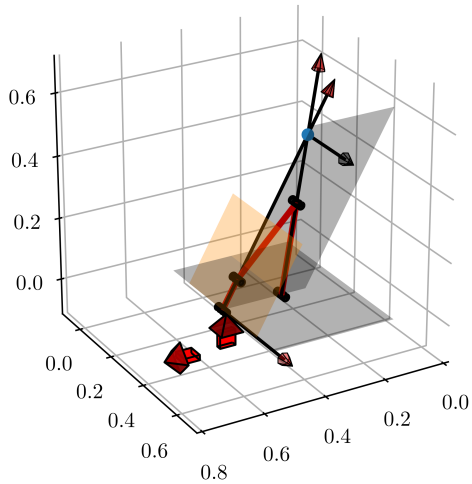


Fig. 4: Wrenches and Twist of a four bar mechanism

III. CASE STUDIES

A. Closed Chain Mechanism

Figure 4 shows the constraint wrenches and twists for a planar four-bar mechanism. This closed loop mechanism can be analysed as a parallel manipulator consisting of two legs, with end effector on the middle link. For each leg the reciprocal screws will be computed, assuming them as a 2R mechanism. These represent the constraint wrenches of each legs. The wrench system of the four bar mechanism is the linear combination of the wrench systems of all its serial chains. One can obtain two constraint couples and two constraint forces for each leg. The constraints exerted on the end-effector link will be the linear combination of individual constraint systems.

The base plane is depicted in grey color. The plane (denoted by beige color) is shown to visualize the constraint force passing through the joint axes and the crank. The two constraint wrenches passing through the two joints of each leg are in the plane (denoted by grey color), the only motion the intermediate link can perform is an instantaneous rotation about an axis that is passing through the intersection of the constraint wrenches as shown by a zero pitch screw in the Fig. 4, as rotation is reciprocal to forces when the axis is either intersecting or is coplanar to the forces.

B. Deployable Mechanism

A three-fold reconfigurable four bar mechanism was developed for a specific use-case scenario of human-robot collaboration [33]. Three orthogonal 4-bar units can be deployed and stowed in three directions using a 1-DOF actuator. Fig. 5, shows the proposed mechanism, as the four bar operation is geometrically intuitive, we show only the crank units of the four bar parallelogram mechanism. Different colored planes are shown to depict the plane of the rotating joints which serves as a base for visualizing constraint forces passing through the joint axes and a planar pencil of screws through the rotation axis of twists.

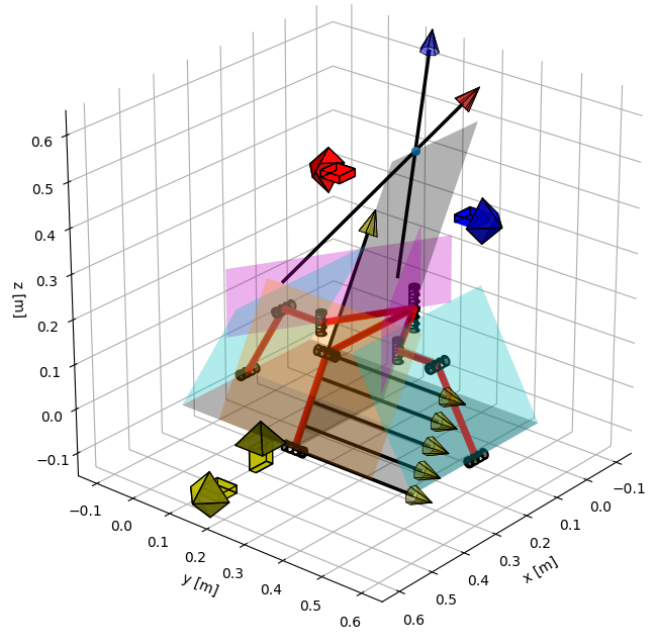


Fig. 5: Mathematical mock-up of the deployable mechanism

The elevation of all three orthogonal cranks must be the same with respect to the base to achieve full-cycle mobility. The full-cycle mobility of this mechanism can be easily deduced by considering the mechanism as a parallel manipulator. The horizontal bar can be considered as a mobile platform with two 2-system and one 4 - system constraint wrench system. The platform is mobile only if the two 2-system of the leg-freedoms have a non-zero intersection. Each of these 2-systems consists of a plane of vertical parallel screws. The (vertical) line of intersection of the two planes is the only possible axis of the putative intersection screw. However, it is required that on this axis the screw belonging to each of the two systems is of the same pitch. This is true only if (1) all screws are of the same (zero) pitch or (2) the intersection line is equidistant from the R joints. Hence, a full-cycle mobility is achieved for this particular geometric arrangement of the deployable mechanism. The

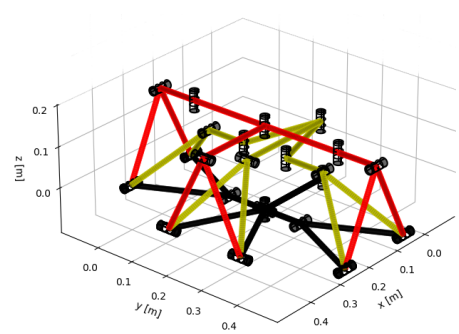


Fig. 6: Motion pattern: Simulation of the deployment process: fully folded state (black color), partially folded state (yellow color) and fully deployed state (red color)

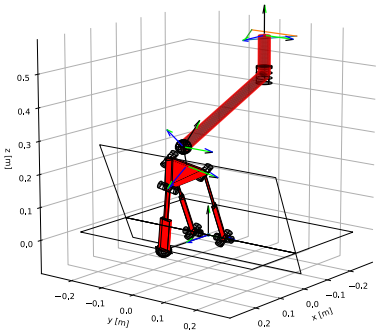


Fig. 7: Hybrid mechanism with Exechon manipulator supporting a Spherical-revolute serial chain

motion pattern of the mechanism is illustrated in Fig. 6.

C. Hybrid Parallel Manipulator

A hybrid parallel-serial mechanism is shown in Fig. 7, where the base of the robot is an Exechon robot mounted with a serial arm for orientation purpose. The Exechon has 2-RRPR legs and 1-SPR leg where R, P, S denote the revolute, prismatic and spherical joint respectively. Constraint wrenches of each leg are shown in Fig. 8. The reciprocal twist system forms a 3-DOF parallel manipulator with mixed motion of rotation and translation.

D. Multi link cable driven robot

The multi-link cable driven robot (MCDR) is an extension of the cable robots where the moving platform is replaced by a multi-body chain. It is typically an open-chain structure with multiple links and complex cable routing. This design introduces the advantages of having a serial kinematic structure and preserves the benefits associated with cable-driven parallel mechanism. Due to the unilateral driving

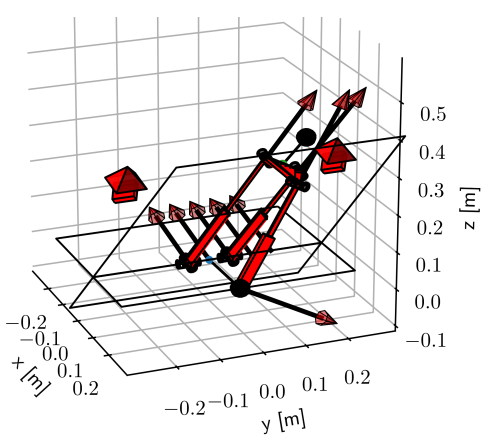


Fig. 8: Reciprocal Leg constraint wrench system of Exechon: Planar pencil of screws, couple, and the constraint force passing though the spherical joint constitute for geometrical constraints. Constraint force passing through the actuated/locked prismatic joint together with geometric constraints represent actuated constraints

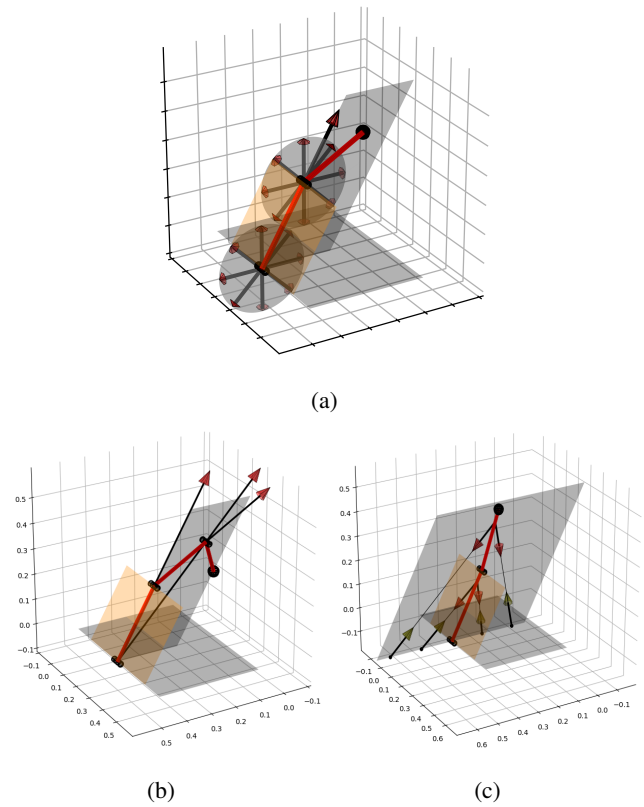


Fig. 9: (a) Planar 2R deficient multi link cable driven robot with reciprocal screw system, pencil of screws through the rotation axis of twists and lying in the plane of rotating joint (b) Reciprocal screws of planar 3R multi link cable driven robot (c) 2R cable driven robot with cable wrenches

property of the cables, maintaining positive cable tension is essential in controlling the moving platform. As a result, the number of actuators must be more than the number of degrees of freedom (DOF) of the moving platform to obtain force-closure. Force closure refers to the ability for a cable driven parallel robot (CDPR) to produce arbitrary wrenches in all DOFs. There has been numerous efforts in force-closure analysis of cable-driven platforms and more recently of cable-driven multi-body systems with open and closed structures.

We use STORM toolbox to determine the reciprocal screws of the cable driven robot and further define the wrenches acting on the multi link system as a linear combination and obtain the net required torques at each joint. The joint torque provided by the cable forces are equated with the joint torques that are required by the external wrenches under the static equilibrium condition.

Figure 9(a) shows the architecture of 2R serial link driven by cables. Figure 9(b) shows the 3R planar chain. For the 3R cable driven chain, the reciprocal screw forms a pencil of screws radiating from the zero-pitch twist axis and lies in the plane of joints. The reciprocal screw for joint 1 must be reciprocal to joints 2 and 3, due to the coplanar conditions. Therefore, the reciprocal screw must pass through

joint axes 2 and 3. Similarly other two reciprocal screws can be identified. The three reciprocal screws are shown in the figure. The 2R chain has only two zero-pitch screws as twists. There will be one independent reciprocal screw in the direction of line joining the center of two joints that is coplanar and will be reciprocal to these two twists. There will be a two-system of screws which will be reciprocal to all twists except being self-reciprocal. These pencil of screws lie on the same plane and passes through the rotation axes of the twits. Any one of the screw from the set can be selected as basis. Figure 9(c) shows the 2R multi-link cable driven robot driven by 4 cables. The cable tensions are depicted in the figure. A broad spectrum of robots and mechanisms ranging from folding mechanism to cable robots are studied using STORM toolbox.

IV. CONCLUSIONS

This article provides a high-level overview of the architecture and features of the STORM simulation and visualization framework. The platform aims to address the lack of a comprehensive screw-theoretic analysis software by achieving five main objectives as described in the introduction. A wide range of known mechanisms, hybrid architectures and algorithms are presented. Furthermore, the modular design protocol of STORM enables the user to add new models and algorithms without reinventing the wheel. The presented case studies elucidate the utility and flexibility in computing reciprocal screws and assists in kinematic analysis on different topologies of robots. The developed code will be supported for Python 3.5 and MATLAB. Future work for STORM will focus on increasing the type of analyses based on screw theory and extend it to non-purely parallel mechanisms.

REFERENCES

[1] M. Ceccarelli, "Screw axis defined by giulio mozzi in 1763 and early studies on helicoidal motion," *Mechanism and Machine Theory*, vol. 35, no. 6, pp. 761–770, 2000.

[2] R. S. Ball, *A Treatise on the Theory of Screws*. Cambridge university press, 1998.

[3] C. Gibson and K. Hunt, "Geometry of screw systems—1: Screws: genesis and geometry," *Mechanism and Machine theory*, vol. 25, no. 1, pp. 1–10, 1990.

[4] K. H. Hunt, *Kinematic geometry of mechanisms*. Oxford University Press, USA, 1978, vol. 7.

[5] C. Gibson and K. Hunt, "Geometry of screw systems—2: classification of screw systems," *Mechanism and Machine Theory*, vol. 25, no. 1, pp. 11–27, 1990.

[6] H. Lipkin and J. Duffy, "Sir robert stawell ball and methodologies of modern screw theory," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 216, no. 1, pp. 1–11, 2002.

[7] J. K. Davidson, K. H. Hunt, and G. R. Pennock, "Robots and screw theory: applications of kinematics and statics to robotics," *J. Mech. Des.*, vol. 126, no. 4, pp. 763–764, 2004.

[8] J. D. Adams and D. E. Whitney, "Application of screw theory to constraint analysis of assemblies of rigid parts," in *Proceedings of the 1999 IEEE International Symposium on Assembly and Task Planning (ISATP'99)(Cat. No. 99TH8470)*. IEEE, 1999, pp. 69–74.

[9] A. Müller, "Screw theory–a forgotten tool in multibody dynamics," *PAMM*, vol. 17, no. 1, pp. 809–810, 2017.

[10] S. K. Mustafa and S. K. Agrawal, "On the force-closure analysis of n-dof cable-driven open chains based on reciprocal screw theory," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 22–31, 2011.

[11] Q. Li, "Type synthesis theory of lower-mobility parallel mechanisms and synthesis of new architectures," *Dissertation, Yanshan University*, 2003.

[12] S. K. Mustafa and S. K. Agrawal, "Reciprocal screw-based force-closure of cable-driven closed chains," in *ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers Digital Collection, 2011, pp. 1243–1251.

[13] H. Pottmann, M. Peternell, and B. Ravani, "An introduction to line geometry with applications," *Computer-aided design*, vol. 31, no. 1, pp. 3–16, 1999.

[14] V. Petuya, E. Macho, O. Altuzarra, C. Pinto, and A. Hernandez, "Educational software tools for the kinematic analysis of mechanisms," *Computer Applications in Engineering Education*, vol. 22, no. 1, pp. 72–86, 2014.

[15] X. Iriarte, J. Aginaga, and J. Ros, "Teaching mechanism and machine theory with geogebra," in *New Trends in Educational Activity in the Field of Mechanism and Machine Theory*. Springer, 2014, pp. 211–219.

[16] J. M. Pardos-Gotor, "Screw theory for robotics: A practical approach for modern robot kinematics," 2018.

[17] L. Dorst, D. Fontijne, and S. Mann, *Geometric algebra for computer science: an object-oriented approach to geometry*. Elsevier, 2010.

[18] K. M. Lynch and F. C. Park, *Modern Robotics*. Cambridge University Press, 2017.

[19] J. M. Selig, *Geometric fundamentals of robotics*. Springer Science & Business Media, 2004.

[20] R. M. Murray, *A mathematical introduction to robotic manipulation*. CRC press, 2017.

[21] J. Angeles, *Fundamentals of robotic mechanical systems*. Springer, 2002.

[22] J. S. Dai and J. R. Jones, "Interrelationship between screw systems and corresponding reciprocal systems and applications," *Mechanism and machine theory*, vol. 36, no. 5, pp. 633–651, 2001.

[23] D. Zlatanov, "The representation of the general three-system of screws by a sphere," *Mechanism and machine theory*, vol. 49, pp. 315–331, 2012.

[24] J. S. Dai, Z. Huang, and H. Lipkin, "Screw system analysis of parallel mechanisms and applications to constraint and mobility study," in *ASME 2004 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers Digital Collection, 2004, pp. 1569–1582.

[25] D. Zlatanov, I. A. Bonev, and C. M. Gosselin, "Constraint singularities of parallel mechanisms," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, vol. 1. IEEE, 2002, pp. 496–502.

[26] M. Zoppi, D. Zlatanov, and R. Molfino, "On the velocity analysis of interconnected chains mechanisms," *Mechanism and machine theory*, vol. 41, no. 11, pp. 1346–1358, 2006.

[27] J. Zhao, Z. Feng, F. Chu, and N. Ma, *Advanced theory of constraint and motion analysis for robot mechanisms*. Academic Press, 2013.

[28] K. Sugimoto and J. Duffy, "Application of linear algebra to screw systems," *Mechanism and machine theory*, vol. 17, no. 1, pp. 73–83, 1982.

[29] J. S. Dai and J. R. Jones, "Null–space construction using cofactors from a screw–algebra context," *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 458, no. 2024, pp. 1845–1866, 2002.

[30] J. Zhao, H. Zhou, Z. Feng, and J. Dai, "An algebraic methodology to identify the principal screws and pitches of screw systems," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 223, no. 8, pp. 1931–1941, 2009.

[31] J. S. Dai and J. R. Jones, "A linear algebraic procedure in obtaining reciprocal screw systems," *Journal of robotic systems*, vol. 20, no. 7, pp. 401–412, 2003.

[32] J. Zhao, B. Li, X. Yang, and H. Yu, "Geometrical method to determine the reciprocal screws and applications to parallel manipulators," *Robotica*, vol. 27, no. 6, pp. 929–940, 2009.

[33] K. Sagar, V. Ramadoss, M. Jilich, M. Zoppi, D. Zlatanov, and A. Zanella, "Development of a reconfigurable four-bar mechanism for a human robot collaborative gripper," in *International Symposium on Advances in Robot Kinematics*. Springer, 2020, pp. 311–318.