Locking underactuated robots by shrinking their manifolds of free-swinging motion

Adrián Peidró[a,*], Pedro David Pérez-Navarro[a], Rafael Puerto[a], Luis Payá[a] and Oscar Reinoso[a,b]

[a]Systems Engineering and Automation Department, Miguel Hernández University, Avda. de la Universidad s/n, Elche, 03202, Alicante, Spain
[b]Fundación ValgrAI, Camí de Vera s/n, Edificio 3Q, Valencia, 46022, Valencia, Spain

ARTICLE INFO

Keywords:
Underactuated robot
Free swing joint failure
Torque failure
Self-motion manifold
Parallel robot

ABSTRACT

When a robot suffers a Free Swing Joint Failure, it becomes underactuated since some of its actuated joints become passive. If the joints that remain actuated are locked, the robot can perform uncontrollable free-swinging motions that are dangerous due to the risk of collision. This is typically corrected by using brakes or redundant actuators, which are costly solutions. In this paper, we propose an alternative method to secure underactuated robots after Free Swing Joint Failure without brakes or redundant actuators, by taking the healthy actuated joints to what we call "locking configurations". At these configurations, the manifolds along which the underactuated robot performs free-swinging motions degenerate into points, suppressing such uncontrollable motions. We present a discretization and clustering method to find locking configurations by detecting the degeneracy of free-swinging manifolds into points, introducing two criteria to confirm if these configurations are stable, by checking if passive joint velocities are forbidden or if any external wrench can be balanced by the healthy actuators. Several examples illustrate the method, including serial and parallel underactuated robots with varying numbers of failed actuators.

1. Introduction

Underactuated robots have a number of degrees of freedom (DOF) greater than the number of actuators, or to be more accurate, than the number of independent input vector fields [1]. Robots can be underactuated by design, in order to save actuators and reduce weight and energy consumption, at the expense of a higher complexity in motion planning and control [2, 3], as underactuation introduces non-holonomic dynamic constraints between positions, velocities, and accelerations of the actuated and passive joints. Such constraints, which can be integrated only in very specific situations [4], do not restrict the positions and velocities that the robot can achieve, but they constrain the trajectories necessary to reach the desired positions and velocities, which cannot be arbitrary [5].

In other occasions, underactuation occurs undesirably when some actuators completely lose their capability to exert control forces or torques on the joints they drive, which become passive. This situation is called Free Swing Joint Failure (FSJF) or torque failure [6]. The occurrence of FSJF is critical in applications for which accessing the robot to repair it is impossible or very expensive, such as underwater or space robots [7, 8]. Moreover, FSJF is especially dangerous because, after it occurs, part of the mechanical linkage of the robot becomes free to oscillate without control, so there is a risk that the manipulator collides with objects of the environment, including itself, potentially causing damage to people or equipment. FSJF is a more dangerous failure than the other typical failure of the joints of a robot, which is locked (or jammed) joint failure [9], also called position failure [6]. In a locked joint failure, one of the joints becomes locked, producing a rigid union between adjacent links. Unlike a FSJF, a locked joint does not produce uncontrollable motions, but the manipulator loses DOF and dexterity. In this paper we will focus on FSJF, which can be dealt with at three stages: prevention, detection, and correction.

Prevention of FSJF can be addressed during operation of the robot. English and Maciejewski [10] defined an index that assigns a higher probability of suffering FSJF to those joints supporting larger gravitational loads, or an index that measures the angle that a failed joint would freely swing (or the Euclidean distance that the end-effector would sweep [11]) under the effect of gravity after suffering FSJF. Accordingly, to minimize the probability of occurrence of FSJF or its potential effects, the robot should move near the regions of its workspace where these indices are zero [12]. Prevention of FSJF can also be considered when designing the robot. For example, Yi et al. [6] defined an index

*Corresponding author
apeidro@umh.es (A. Peidró); pedrodperez@gmail.com (P.D. Pérez-Navarro); r.puerto@umh.es (R. Puerto); lpaya@umh.es (L. Payá); o.reinoso@umh.es (O. Reinoso)
ORCID(s): 0000-0002-4565-496X (A. Peidró); 0000-0003-2326-2743 (R. Puerto); 0000-0002-3045-4316 (L. Payá); 0000-0002-1065-8944 (O. Reinoso)

of tolerance to torque failure, which quantifies the ability of the healthy actuated joints to control the motion of failed actuated joints, and suggests adding redundant actuators or limbs to increase such tolerance during the design stage.

Regarding detection, Shin et al. [13] detect the occurrence of FSJF when the real position of the end-effector becomes too different from the commanded one. Then, they identify the failed joint by an on-line comparison between the motion of the real robot and the motion of different faulty simulated robots, where each simulated robot has a different faulty joint. Eventually, the real robot is identified with the simulated robot that moves more similarly to it, and a robust non-singular Computed-Torque Controller is used to control the healthy actuated joints and follow the desired end-effector trajectory. Dixon et al. [14] detect FSJF when the real torques are too different from those predicted by the dynamic model of the robot. Actually, they filter torques through 1st-order low-pass filters to avoid estimating joint accelerations. De Luca and Mattone [15] use a similar method based on generalized momenta to avoid using joint accelerations, but they only require filtering part of the dynamic model, obtaining a residual that converges to the fault torque function. Finally, Tinos et al. [16] approximate the dynamic model by a multilayer perceptron and use it to predict the joint velocities, then the error between the predicted and real joint velocities is introduced to a Radial Basis Function classifier to distinguish between FSJF, locked-joint failure, or no failure at all.

After prevention and detection of FSJF comes its correction. Three main corrective strategies can be identified in the literature: redundant actuation, underactuated control through dynamic coupling, and use of brakes.

Considering the first strategy, roughly speaking, a robot is considered redundantly actuated when it has more actuated joints than degrees of freedom. After some actuators of a redundantly actuated robot suffer FSJF, it should retain enough healthy actuators to control all of its DOF, such that it does not become underactuated. In this sense, From and Gravdahl [17] used Screw Theory to determine which joints should have redundant actuators so that the manipulator remains equilibrated (i.e., without passive mobility) when any actuated joint suffers FSJF. Redundant actuation is also adopted by Ting et al. [18], who propose a torque redistribution between all actuated joints to guarantee that they can still provide the torques required to achieve the desired motion, after some joints suffer FSJF. An inconvenient of redundant actuation is that it is expensive because more actuators are required, which also increases weight. Moreover, it makes control more complex before FSJF occurs, i.e., while the robot is redundantly actuated, as it is necessary to coordinate different actuators to limit internal stresses [19].

The second correction strategy is to use dynamic coupling between the (healthy) actuated joints and passive joints (including the failed actuated joints) to control the passive joints and drive them to their desired values during a first phase, locking them with brakes after that, and finally controlling the actuated joints to reach their desired values during a second phase [20]. This strategy requires that there exists enough dynamic coupling between active and passive joints [21]. During the first phase, Maciel et al. [22] only specify the desired trajectories for the passive joints, resulting in an underconstrained system of dynamic equations that does not have a unique solution, providing some freedom or "redundancy" for solving the trajectories of the actuated joints in terms of the desired trajectories for the passive joints, which leaves margin to achieve secondary goals such as minimizing actuation torques. On the contrary, during the first phase, Arai and Tachi [20] specify the desired trajectories both for passive joints and for a subset of actuated joints, so that the evolution of all joints is uniquely defined, without taking advantage of the redundancy exploited in [22].

Control of passive joints through dynamic coupling is also proposed by De Luca et al. [23, 24], but in contrast to the previous approaches, they drive first the actuated joints to their desired values through conventional closed-loop control. After this, periodic open-loop controllers are designed on the basis of a nilpotent approximation of the dynamics of the robot, in order to produce the oscillation of actuated joints around their desired values, with the effect that the passive joints get closer to their desired values after each period, until they converge. Fu et al. [25] present a similar control scheme that iteratively learns the trajectory to execute by the active joints so that friction stops the passive joints at their desired values, to finally lock them with brakes. Finally, Shin and Lee [26] use the dynamic coupling between actuated and passive joints to define a Computed-Torque Controller that directly controls the position and orientation coordinates of the end-effector, as long as the number of such coordinates does not exceed the number of healthy actuated joints, and provided that the robot is far from dynamic singularities (or second-type passive joint singularities [27]), at which a "decoupling matrix", which maps active-joint torques to end-effector accelerations, is not full-rank.

The third correction strategy is the use of brakes to lock the failed actuated joints. One possibility is braking the passive joints after they have been driven to their desired positions by exploiting their dynamic coupling with actuated joints, as explained above. Another possibility is active braking [28], which consists in activating/deactivating the brakes of the passive joints conveniently, taking advantage of gravity to reconfigure the underactuated robot as desired. First, passive joints are locked with brakes and actuated joints move the center of gravity of the robot to a stable equilibrium position. Then, passive joints are released and actuated joints are moved slowly, which automatically produces the motion of passive joints in response, to ensure that the center of gravity remains at equilibrium. Taking advantage of this balancing motion of passive joints, they are driven to their desired values, after which they are braked again so that the actuated joints can also be controlled toward their respective targets.

Most of the previous papers on corrective strategies try to continue operating the robot after FSJF occurs, even with degraded performance, using redundant actuators, dynamic coupling, or braking. Fewer research papers have focused on completely halting and taking the robot to a safe configuration after the failure, aborting its mission to avoid damaging its surroundings due to free-swinging motions. Conventional solutions to halt the robot after FSJF are redundant actuation or braking. However, redundant actuation and braking are expensive and make the robot heavier, and not all robots have brakes. Therefore, in this paper we propose finding configurations to which the healthy actuated joints can be moved after the occurrence of FSJF, so that the robot is stably halted when conventional solutions such as redundant actuation and braking are not available.

Few papers have focused on halting and taking the robot to a safe configuration after FSJF, without using redundant actuators or brakes. English and Maciejewski [29] determined, for a kinematically redundant 3R serial arm with a passive joint, the regions of workspace for which the center of mass is in stable equilibrium under the effect of gravity. Nightingale et al. [30, 31] propose a discontinuous controller to halt underactuated robots starting from arbitrary initial states. This controller is based on a quadratic form which measures the effect that actuated velocities have on the unactuated velocity. When this quadratic form is indefinite, the actuated velocities can be used to increase or decrease the unactuated velocity until driving it to zero from any initial condition. After this, the actuated velocities can also be driven to zero in order to halt the robot. Finally, Pham et al. [32] obtain, for serial and parallel manipulators with failed active joints that have become passive, the values of the active joints for which the robot can resist a given external wrench, that is, so that the twist generated by the passive joints (which depend on the values of the active joints) is reciprocal to the given external wrench. In that case, the external wrench does not produce uncontrolled passive motion, remaining conditionally equilibrated.

By studying several examples, Pham et al. [32] conclude that, typically, while operating underactuated serial robots, it is possible to take their actuated joints to configurations for which the twist generated by the passive joints is reciprocal to a given external wrench (e.g. gravity). For underactuated parallel robots, they conclude that, typically, if the robot resists passively a given wrench, it does so for every configuration of the active joints (or for no configuration at all), which suggests that tolerance to FSJF in underactuated parallel robots should be addressed while designing the robot, not while operating it (unlike with underactuated serial robots).

On the contrary, in the present paper we demonstrate that, while operating underactuated parallel robots that have suffered FSJF, it is also possible to find configurations of the actuated joints for which the robot becomes stably locked, resisting any external wrench (and not only a given external wrench, as in [32]), without redundant actuators or brakes. In this way, tolerance to FSJF in parallel robots can also be addressed during the operational stage (and not only during the design stage), as it occurs with underactuated serial robots [32]. In the present paper, this is done by following a different approach compared to the previous papers that deal with underactuated robots. As we will explain below, the approach followed here is based on the notion of *free-swinging manifolds*, which is analogous to the notion of *self-motion manifolds* of kinematically-redundant robots. The contribution of the present paper is summarized next.

When a robot manipulator (either serial or parallel) suffers FSJF, the number r of actuated joints becomes smaller than the number n of degrees of freedom. As a result, locking its r actuated joints at generic values does not transform the robot into a rigid structure, but into a mechanism with $(n-r)$ DOF. When we lock its actuated joints, the configuration of the underactuated robot is free to slide along a manifold of dimension $(n-r)$, and the motion of the robot along this manifold (which we call *free-swinging manifold*) is uncontrolled, as it depends on factors such as inertia, friction, or external forces like gravity. It is the uncontrolled motion of the robot along these free-swinging manifolds which becomes dangerous, as it may produce the collision of the robot with itself or the environment. In this paper we show that, by driving the actuated joints that remain healthy after FSJF to special configurations (which we call *locking configurations*), it is possible to produce the degeneracy of such manifolds into points, which means that the configuration of the underactuated robot can no longer move uncontrollably along a positive-dimensional manifold. It becomes confined to a point instead, suppressing the free-swinging motions and locking the underactuated robot.

The remainder of this paper is organized as follows. First, Section 2 reviews the notion of self-motion manifolds of kinematically-redundant robots, and introduces the analogous concept of free-swinging manifolds for underactuated robots, explaining how the degeneracy of such manifolds into points can be exploited to lock underactuated robots. Next, Section 3 describes a discretization method to determine the *locking configurations*, which are the positions to which the actuated joints of an underactuated manipulator can be driven so that the free-swinging manifolds degenerate into points. This method is illustrated in Section 4 with a first example involving a planar parallel robot, which has five discrete locking configurations. Following, Section 5 presents two criteria to classify locking configurations into stable or unstable, where stable configurations are those for which the robot can resist any external wrench, remaining rigidly locked. Desirable locking configurations should thus be stable. Then, more examples are presented in order to discuss different situations, such as having positive-dimensional sets constituted by infinitely many locking configurations (Section 6), applying the proposed method to serial underactuated robots (Section 7), or to parallel robots with several failed joints (Section 8). Finally, conclusions are drawn in Section 9, which also proposes future lines of research.

2. Types of self-motions

The term "self-motion" is used in the literature to refer to different concepts that, although related, are not completely equivalent. In this section, this term will be clarified, and it will be explained how it can be used in the proposed method. To that end, in the first place we will analyze fully-actuated kinematically redundant robots, which can be regarded, in a sense, as a dual case to underactuated robots, as it will be argued in this section.

Consider the serial robot of Figure 1a. This is a planar robot with three actuated revolute joints. Therefore, it has three degrees of freedom that allow the robot to control the position $C = (x, y)$ of its gripper, as well as its orientation ϕ, by appropriately commanding the input angles $(\theta_1, \theta_2, \theta_3)$. For a given desired position (x, y) and orientation ϕ, there exist two possible solutions for the inverse kinematic problem, as illustrated in Figure 1a: elbow-up and elbow-down. Now assume that the orientation ϕ of the gripper is irrelevant, and one desires to control only its position (x, y). In that case, the robot of Figure 1a becomes *kinematically redundant*, as it has three DOF but is required to perform a two-dimensional task: controlling the (x, y) coordinates of its gripper. For this kinematically redundant robot, the desired position (x, y) can be reached not only with two different solutions as shown in Figure 1a, but there exist infinitely many different inverse kinematic solutions which, for a generic (x, y), constitute a curve (or a set of curves) of solutions as shown in Figure 1b, which is a one-dimensional manifold that exists in the space of joint coordinates. This is known as a *self-motion manifold* [33], whose name comes from the fact that, when the joint coordinates of the robot slide along these manifolds (i.e, when the robot performs self-motions), the position (x, y) of the gripper does not change, although the overall configuration of the robot changes, as shown in Figures 1c-f. Generically, the dimension of these manifolds equals the degree of kinematic redundancy of the robot. In the example of Figure 1, the robot has three degrees of freedom $(\theta_1, \theta_2, \theta_3)$ and two variables to control (x, y). Therefore, the degree of redundancy in this example is one, and the self-motion manifolds are curves in the space of joint coordinates $(\theta_1, \theta_2, \theta_3)$, as illustrated in Figure 1b. In general, the solution to reach a given (x, y) may consist of multiple manifolds, e.g., different disjoint curves like the one shown in Figure 1b.

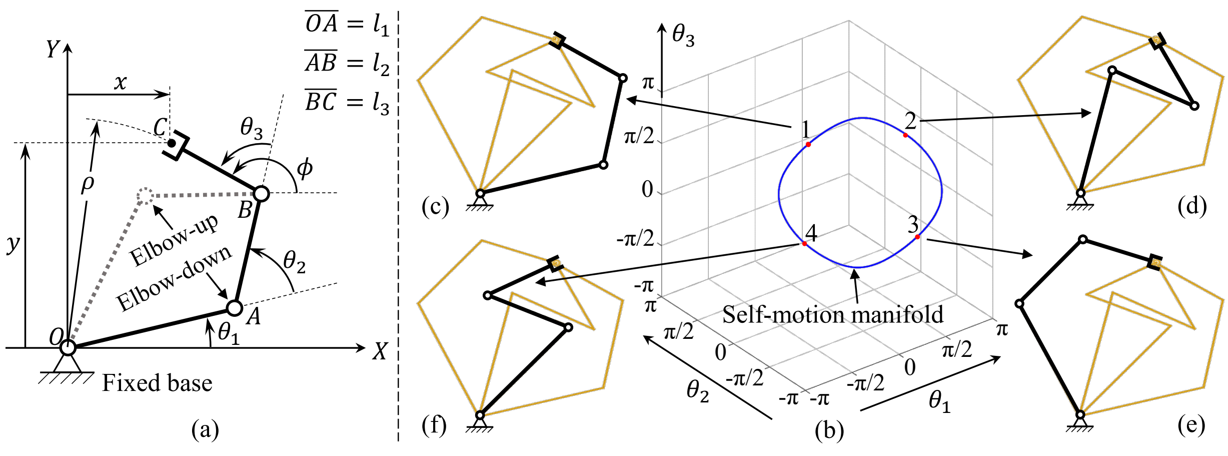

Figure 1: (a) A 3-DOF serial robot with links of lengths $l_1 = 1$, $l_2 = 0.7$, and $l_3 = 0.6$ and end-effector C placed at $x = 0.6$ and $y = 1.2$. (b) Self-motion manifold of the robot, which is a curve consisting of all configurations $(\theta_1, \theta_2, \theta_3)$ that place the end-effector at $(x = 0.6, y = 1.2)$. (c-f) Four particular configurations 1, 2, 3, and 4 that belong to the self-motion manifold, for which the robot adopts different postures placing its end-effector always at $(x = 0.6, y = 1.2)$.

Now consider the closed-chain manipulator shown in Figure 2. This is a three-DOF manipulator obtained by connecting the free gripper of the serial manipulator of Figure 1a to the ground through a PPR kinematic chain (where P means prismatic joint and R means revolute joint), and attaching a new gripper to link BC. Assume that this closed-chain manipulator is fully actuated, with both prismatic joints being actuated, as well as the first revolute joint O connected to the ground. The input kinematic variables of this fully-actuated robot are the displacements (x, y) of the prismatic joints, as well as the rotation angle θ_1 of the first revolute joint.

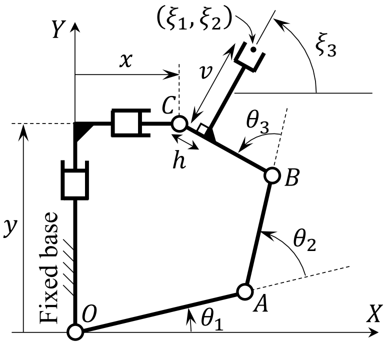

Figure 2: A closed-chain manipulator with the same dimensions as the serial robot of Figure 1a.

Assume next that this closed-chain manipulator suffers a FSJF such that θ_1 becomes passive. Only the displacements (x, y) remain directly controlled by their respective prismatic actuators. In that case, the robot becomes underactuated. Note that the resulting underactuated robot can be seen as a "dual" version of the kinematically redundant robot of Figure 1a. Indeed: if one locks the actuators of joints x and y, then this is equivalent to placing the tip of the gripper of the redundant robot of Figure 1a at the desired position (x, y). With (x, y) locked, the underactuated robot of Figure 2 can adopt infinitely many possible configurations that lie on a manifold that is identical to that of the kinematically redundant robot of Figure 1.

There is, however, a fundamental difference between the self-motions of the kinematically redundant robot of Figure 1 and those of its underactuated counterpart (or dual) of Figure 2. When a kinematically redundant robot moves its configuration along the self-motion manifolds, these movements are controlled at all times. In fact, it is the synchronized motion of the actuated joints of a kinematically redundant robot that keeps the configuration of the robot moving along a self-motion manifold, with the gripper remaining motionless at the desired position (x, y) while the overall configuration of the robot changes. On the contrary, the self-motions of an underactuated robot are not controlled: they depend on external forces such as gravity, and on the dynamics of the robot (inertias, friction, etc.). Indeed, if we lock the healthy actuators (x, y) of the underactuated robot of Figure 2 at $x = 0.6$ and $y = 1.2$, then the angles $(\theta_1, \theta_2, \theta_3)$ can slide freely and uncontrollably along the manifold shown in Figure 1b. These motions will be governed by external forces that are not under control, and they will produce the free swinging of links OA, AB, and BC of the robot of Figure 2, adopting different configurations such as those of Figure 1c-f, which may produce collisions and damage the robot itself or its environment.

Self-motions of kinematically redundant robots are associated to the solution of their inverse kinematic problem, which consists in determining the complete configuration of the robot (i.e., the positions and orientations of all its moving bodies) in order to reach the desired position for the gripper. On the contrary, self-motions of underactuated robots are associated to the solution of their forward kinematic problem, which consists in determining the complete configuration of the robot for given values of its actuated joints, which are fewer in number than the DOF of the robot. For the examples of Figures 1a and 2, both problems can be regarded mathematically as one an the same: finding θ_i for given (x, y). The solution set of this problem for a generic (x, y) consists of a one-dimensional manifold as shown in Figure 1b (or possibly, multiple disjoint manifolds, for other values of x, y). However, although both problems are identical from the mathematical or geometric points of view, they are physically different, as self-motions of kinematically redundant robots are fully controlled, whereas those of underactuated robots are not. In the literature, usually both types of phenomena are indistinctly called self-motions (see Refs. [34, 35], for example), but they should be called differently in order to avoid confusion and reflect the fact that self-motions of underactuated robots are uncontrolled. Accordingly, in what remains of this paper, we will reserve the traditional term "self-motion manifold" for controlled self-motions of kinematically redundant robots that do not alter the position of the gripper, and we will refer to the uncontrollable self-motions of underactuated robots as "free-swinging motions" or "free-swinging manifold" (FSM).

Controlled, or traditional, self-motions (associated to the inverse kinematics) of kinematically redundant manipulators have been extensively studied, both for serial and parallel or closed-chain manipulators [34, 36]. On the contrary, free-swinging motions of underactuated manipulators have not been studied as much. Hassan and Notash [9] analyzed the inverse Jacobian matrix that maps gripper velocities to velocities of the healthy actuated joints in underactuated robots that have suffered FSJF, and they identified the null space of this matrix with the set of velocities that the output gripper admits when locking the actuated joints. When mapped to the space of passive joint velocities, this null space is tangent to the free-swinging manifolds studied in the present paper. Apart from reference [9], in which free-swinging motions were locally identified at a velocity level, to the best of the authors' knowledge, free-swinging manifolds of underactuated robots have not been addressed globally at configuration level yet, which will be the basis of the method proposed later in the present paper.

Finally, some papers have dealt with free-swinging motions of *locally underactuated* parallel robots. By *locally underactuated* we mean that the robot is fully actuated for generic configurations, but it can attain special singular configurations at which locking the actuated joints leaves some part of the robot free to perform finite and uncontrollable motions. It should be highlighted that these have traditionally been called *self-motions* (of the forward kinematics) *of parallel robots*, to distinguish them from the traditional *self-motions* (of the inverse kinematics) *of kinematically redundant serial robots*. In this way, when some papers talk about self-motions of (locally underactuated) parallel robots, they actually refer to what we call free-swinging motions, as if the other type of controlled self-motions (associated to the inverse kinematics) were exclusive of kinematically redundant serial robots. However, this term can be confusing because parallel robots can also be kinematically redundant and admit self-motions as the solution to its inverse kinematic problem [34], analogous to those of redundant *serial* robots. Hence the need, as argued above, to use a clearer term to distinguish free-swinging self-motions of underactuated robots from traditional controlled self-motions of kinematically redundant robots, without having to specify if the robot is serial or parallel in either case (we can have serial or parallel underactuated robots, as well as serial or parallel kinematically-redundant robots).

Examples of fully-actuated robots that can reach singular configurations at which they become locally underactuated are presented in [35, 37, 38, 39, 40, 41]. The example of [38], shown in Figure 3, is intuitive to illustrate this kind of local underactuation: a five-bar robot with distal links having the same length L is fully-actuated at generic configurations as shown in Figure 3a, but if the actuated angles θ_1 and θ_2 are chosen so that joints $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$ coincide as in Figure 3b, then the end-effector $\boldsymbol{C}$ can freely rotate without control about $\boldsymbol{B}_1$, describing a circle of radius L, even if the input angles θ_1 and θ_2 are perfectly locked. In contrast to these papers that deal with free-swinging motions due to local underactuation, the present paper will address these motions for globally underactuated robot manipulators, i.e., robots that are underactuated at all configurations after suffering FSJF at some joints that were previously actuated. The basis of the proposed method will be explained next.

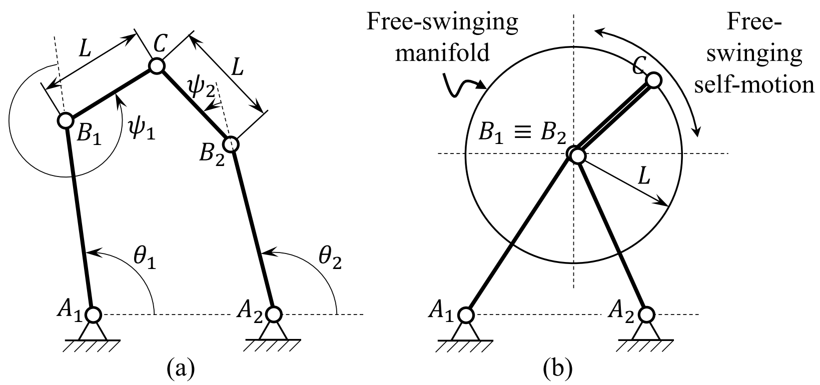

Figure 3: (a) A fully-actuated 2-DOF five-bar parallel robot, at a generic configuration. (b) The same robot, but placed at a singular configuration where it becomes locally underactuated, as its end-effector $\boldsymbol{C}$ can freely rotate about $\boldsymbol{B}_1$ while both input angles (θ_1, θ_2) are locked.

2.1. Suppression of uncontrollable free-swinging motions

The number, shape and size of self-motion manifolds of kinematically redundant robots depend on the desired position (x, y) for the gripper [33]. The position $(x = 0.6, y = 1.2)$ illustrated in Figure 1a can be reached by all configurations that constitute the self-motion manifold shown in Figure 1b, which is a single closed loop. When the desired position (x, y) is changed continuously, this loop will also deform continuously, changing its shape and size. When reaching certain singular positions (x', y'), this loop will suffer topological changes such as splitting into several manifolds or merging with other manifolds. When reaching other singular positions (x'', y''), this loop may even shrink to a point and then vanish. The transformations of self-motion manifolds of kinematically redundant robots when varying the position of the gripper were studied in [33, 42]. In [34], particular attention was paid to transformations that produce the vanishing of self-motion manifolds, as they imply motion impediments for the gripper. The detection of such vanishings is useful when planning trajectories that should avoid motion impediments in kinematically redundant robots. In the present paper, this idea of manifolds that shrink and vanish will be extended to the case of underactauted robots, to suppress their uncontrollable free-swinging motions and lock them stably after they have suffered FSJF. The key idea of the present paper will be explained next.

Let us return again to the closed-chain manipulator of Figure 2, where only the displacements of the prismatic joints x and y remain actuated after suffering a FSJF at joint O, which makes the robot underactuated. The dimensions of the robot of Figure 2 are the same as those of the kinematically redundant serial robot of Figure 1a, and the initial displacements of the actuated joints are $x = 0.6$ and $y = 1.2$, which results in the free-swinging manifold of Figure 4d1, which is identical to the self-motion manifold of Figure 1b. If x and y are locked at $x = 0.6$ and $y = 1.2$, then the actual configuration of the robot will be some point along this manifold, but it is not possible to tell which one if we only lock (and measure) x and y, as the motion along this manifold is free and uncontrolled, subject to external forces such as gravity. For the particular case shown in Figure 4d1, the robot adopts the configuration $\boldsymbol{\theta}^* = (\theta_1 = 0.2279, \theta_2 = 1.1258, \theta_3 = 1.2825)$ rad, which results in the configuration represented in Figure 4d2. Next, we will analyze how the free-swinging manifold of Figure 4d1 changes when the actuated joints (x, y) change, as an analogy to how self-motion manifolds of kinematically redundant robots change when changing the position of their gripper [34]. If we start from the situation of Figure 4d2 and decrease y while keeping x constant, we observe that the free-swinging manifold becomes slightly larger (Figure 4c1, where $y = 1$) until it splits into two components (Figures 4b1 and 4a1, where $y = 0.9$ and $y = 0.8$, respectively). This splitting is one of the possible changes of topology suffered by manifolds as explained in the previous paragraph.

Now, let us return to the configuration of Figure 4d2 to increase y instead of decreasing it, keeping x constant. In that case, as y increases, the free-swinging manifold, which is a closed loop, decreases in size progressively (Figure 4e1 to 4h1), until it degenerates into a point (Figure 4i1). If one tries to further increase y after the manifold has degenerated into a point, this manifold would vanish, meaning that there would not exist real configurations to assemble the manipulator, only imaginary ones. The maximum strokes (x, y) that can be attained by the prismatic actuators are those that keep the kinematic chain $\boldsymbol{O}$-$\boldsymbol{A}$-$\boldsymbol{B}$-$\boldsymbol{C}$ completely extended; larger values of x or y are not possible because this kinematic chain cannot be extended further, as shown in Figure 4i2.

Note that, by driving the actuated joints (x, y) to their limit positions that fully extend the kinematic chain, the free-swinging manifold has degenerated into a point, so that the free swinging has actually been suppressed: there is no positive-dimensional manifold along which the configuration of the robot can uncontrollably slide, only a point. By driving the actuated joints (x, y) to appropriate positions, it has been possible to transform the free-swinging manifold into a single configuration that forbids uncontrollable movements of the robot, at least at a *finite or configuration level*.

The previous example serves to introduce the main aim of this paper: a method to find, if they exist, positions to which the healthy actuated joints of underactuated manipulators can be driven after suffering FSJF, in order to suppress their uncontrollable free-swinging motions. This will be achieved by enforcing the degeneration of free-swinging manifolds into isolated configurations (a point) at which the robot can be locked to prevent uncontrollable motions that may damage the environment or the robot itself. This will be the objective of the following sections.

Note that the degeneracy of a free-swinging manifold into a point is a necessary condition to safely and stably lock an underactuated robot using only its healthy actuated joints, but it is not a sufficient condition. Indeed, taking the example of Figure 4i2 as an example, it is clear from this figure that, when the actuators fully extend the kinematic chain $\boldsymbol{O}$-$\boldsymbol{A}$-$\boldsymbol{B}$-$\boldsymbol{C}$, they neither resist external torques nor forces applied perpendicularly to this kinematic chain, because this kinematic chain is not rigid (points $\boldsymbol{A}$ and $\boldsymbol{B}$ admit velocities perpendicularly to the chain). This is, uncontrollable free-swinging motions may still be possible at *differential or velocity level*, even after suppressing them at finite level.

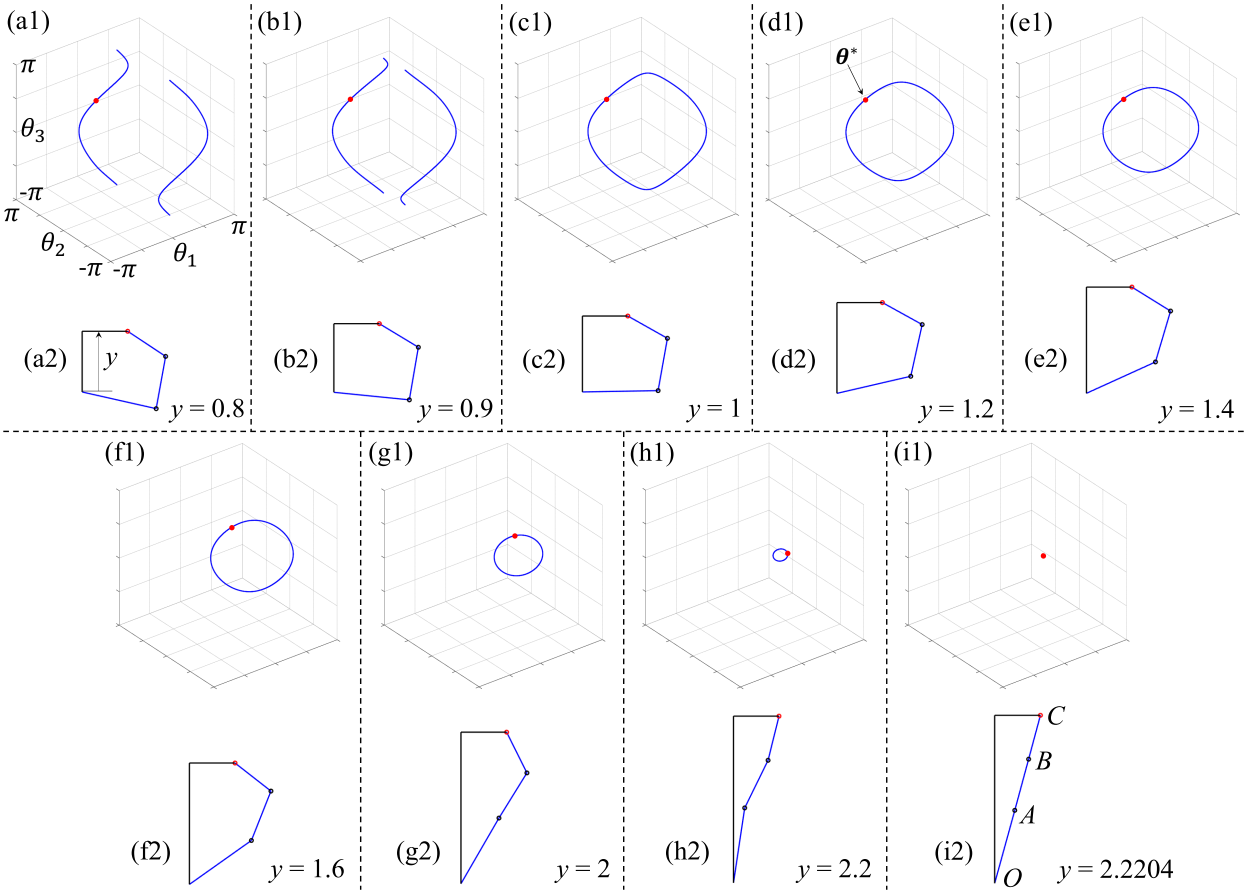

Figure 4: (Upper subfigures a1 ... i1): Evolution of free-swinging manifolds in the space of passive joint displacements $(\theta_1, \theta_2, \theta_3)$ when varying the actuated joint displacement y. (Lower subfigures a2 ... i2): Configuration adopted by the robot when the passive joint coordinates are those of the red point $\boldsymbol{\theta}^*$ shown in the corresponding upper subfigures a1 ... i1. The representation of the gripper is omitted for clarity.

Summing up: when a free-swinging manifold degenerates into a point configuration and the robot adopts this configuration, the robot may not necessarily remain static under all external forces, so it may not be a stable configuration at which to safely lock the underactuated robot. The stability of free-swinging manifolds degenerated into points will be examined in Section 5. However, in the next section, first we will deal with the necessary condition, i.e., finding the positions of the healthy actuated joints for which the free-swinging manifolds shrink to points.

3. Locking configurations

Consider a robot with n degrees of freedom which, initially, is fully-actuated and has n active joints and m passive joints. Assume that, after FSJF occurs, only r $(< n)$ joints remain actuated, whose displacements are grouped in the vector $\boldsymbol{\theta}_a \in \mathbb{R}^{r\times 1}$. In that case, the number of passive joints after FSJF will increase to $s = m + n - r$, and their displacements will be grouped in an s-dimensional vector $\boldsymbol{\theta}_p$. For parallel robots and robots with closed kinematic chains, these active and passive joint displacements are related through a set of m loop-closure constraints as follows:

$$\mathbf{F}\left(\boldsymbol{\theta}_a, \boldsymbol{\theta}_p\right) = \mathbf{0}_{m\times 1} \tag{1}$$

For the example of the underactuated robot of Figure 2 with a FSJF in θ_1, we have that $n = 3$, $\boldsymbol{\theta}_a = [x, y]^T$ and $\boldsymbol{\theta}_p = [\theta_1, \theta_2, \theta_3]^T$, and the loop closure equation for that robot is:

$$\mathbf{F}\left(\boldsymbol{\theta}_a, \boldsymbol{\theta}_p\right) = \begin{bmatrix} -x + l_1 \cos\left(\theta_1\right) + l_2 \cos\left(\theta_1 + \theta_2\right) + l_3 \cos\left(\theta_1 + \theta_2 + \theta_3\right) \\ -y + l_1 \sin\left(\theta_1\right) + l_2 \sin\left(\theta_1 + \theta_2\right) + l_3 \sin\left(\theta_1 + \theta_2 + \theta_3\right) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{2}$$

For a fully-actuated robot with $r = n$ actuated joints (before suffering FSJF), and for fixed displacements $\boldsymbol{\theta}_a$ of the actuated joints, Equation (1) is a system of m nonlinear equations in m unknowns (the displacements $\boldsymbol{\theta}_p$ of the passive joints), which generically admits a finite set of solutions. This is the "ordinary" forward kinematic problem of a fully-actuated robot manipulator: determining the complete configuration of the robot for given displacements of the actuated joints, which generally has different solutions traditionally known as "assembly modes".

For an underactuated robot with r ($< n$) actuated joints, keeping $\boldsymbol{\theta}_a$ constant in Equation (1) yields a system of m equations in the s ($> m$) unknowns $\boldsymbol{\theta}_p$. As the number of unknowns now exceeds the number of equations by $n - r$, the solution set of Equation (1) is, generically, a collection of disjoint manifolds of dimension $n - r$ in the space of passive joint displacements $\boldsymbol{\theta}_p$. These are the Free-Swinging Manifolds (FSM) introduced in Section 2. For example: the solution set of Equation (2) in the space of passive joint displacements $(\theta_1, \theta_2, \theta_3)$ for $x = 0.6$, $y = 1.2$, $l_1 = 1$, $l_2 = 0.7$, and $l_3 = 0.6$, is the one-dimensional manifold shown in Figure 4d1. And when y takes different values, this solution set evolves as shown in the remaining upper subfigures (a1, b1, ..., i1) of Figure 4.

We will also consider that there exist d inequality constraints of the following form:

$$\mathbf{G}\left(\boldsymbol{\theta}_a, \boldsymbol{\theta}_p\right) \geq \mathbf{0}_{d\times 1} \tag{3}$$

Equation (3) means that $G^i \geq 0$ for $i = 1, \ldots, d$, where G^i is the i-th component of $\mathbf{G}$. These inequalities model joint limits or collisions. For a given value of the active joint displacements $\boldsymbol{\theta}_a$, these constraints exclude forbidden regions of the s-dimensional space of passive joint displacements $\boldsymbol{\theta}_p$, i.e., regions where $G^i < 0$ for some i. The parts of the free-swinging manifolds that intersect these forbidden regions must be removed, which may result in the disappearance of complete components of the manifolds or parts of them, which may further split manifolds into smaller disjoint components. As it will become evident later, inequality constraints such as joint limits and collisions will play a central role in this paper. Note that such constraints are always present in practical applications.

In what remains of this section, we will describe a method to determine the points $\boldsymbol{\theta}_a^*$ of the space of active joint displacements $\boldsymbol{\theta}_a$ to which the actuated joints of an underactuated robot can be driven to suppress the uncontrollable free-swinging motions, which occurs when free-swinging manifolds shrink into points as argued in Section 2.1. Hereafter, these $\boldsymbol{\theta}_a^*$ will be called *locking configurations*.

In the following, we will assume that only one actuated joint has suffered FSJF, i.e.: $r = n - 1$ and $s = m + 1$. In that case, the free-swinging manifolds are one-dimensional, i.e., parts of smooth curves. This will allow us to simplify the upcoming computations and explanations. Section 8 will discuss FSJF in multiple joints simultaneously.

3.1. Method for detecting manifolds that degenerate into points

In this section, we will describe a method to find the locking configurations, which are values $\boldsymbol{\theta}_a^*$ of the actuated joint displacements for which free-swinging manifolds degenerate into points, suppressing in this way the uncontrollable motions, at least at a finite level (but not necessarily at differential level, which will be addressed in a later section). The method that will be presented next is an evolution of the method presented in [34], but with some improvements and modifications. In [34] we presented a method for finding the positions of the gripper of kinematically redundant robots for which self-motion manifolds degenerate into points and vanish, producing a barrier for the motion of the gripper. Since computing the self-motion manifolds of a redundant robot is analogous to computing the free-swinging manifolds of an underactuated robot, the method presented in [34] can be extended to the problem addressed here.

The method consists of the following steps. Firstly, it is necessary to discretize the r-dimensional space of $\boldsymbol{\theta}_a$ into a grid. For example, if $r = 1$, we have a single actuated joint displacement θ_{a1}. Assuming that its joint limits are θ_{a1}^{min} and θ_{a1}^{max}, in that case we will discretize the line segment $\left[\theta_{a1}^{min}, \theta_{a1}^{max}\right]$ into a grid of equally spaced nodes, as shown in Figure 5a. Similarly, if we have $r = 2$ actuated joints $\boldsymbol{\theta}_a = [\theta_{a1}, \theta_{a2}]^T$ with their respective joint limits, in that case we will discretize the rectangle $\left[\theta_{a1}^{min}, \theta_{a1}^{max}\right] \times \left[\theta_{a2}^{min}, \theta_{a2}^{max}\right]$ into a two-dimensional grid of nodes, as shown in Figure 5b. In the general case of r actuated joints, this process generalizes to discretizing an r-dimensional box. The resolution $\Delta\theta_{ai}$ of the grid must be chosen according to the desired resolution. If a finer resolution is chosen, there will be more nodes and the method will generate more accurate results but it will also require higher CPU time.

After discretizing the space of active joint displacements $\boldsymbol{\theta}_a$ into a grid with the desired resolution, the following three operations must be done for each node of this grid: mapping (§3.1.1), clustering (§3.1.2), and matching (§3.1.3).

3.1.1. Mapping

First, it is necessary to map the free-swinging manifolds at the current node $\boldsymbol{\theta}_a^A$. This means obtaining a sufficiently dense point cloud that approximates these manifolds accurately. This can be achieved by sweeping the first passive joint

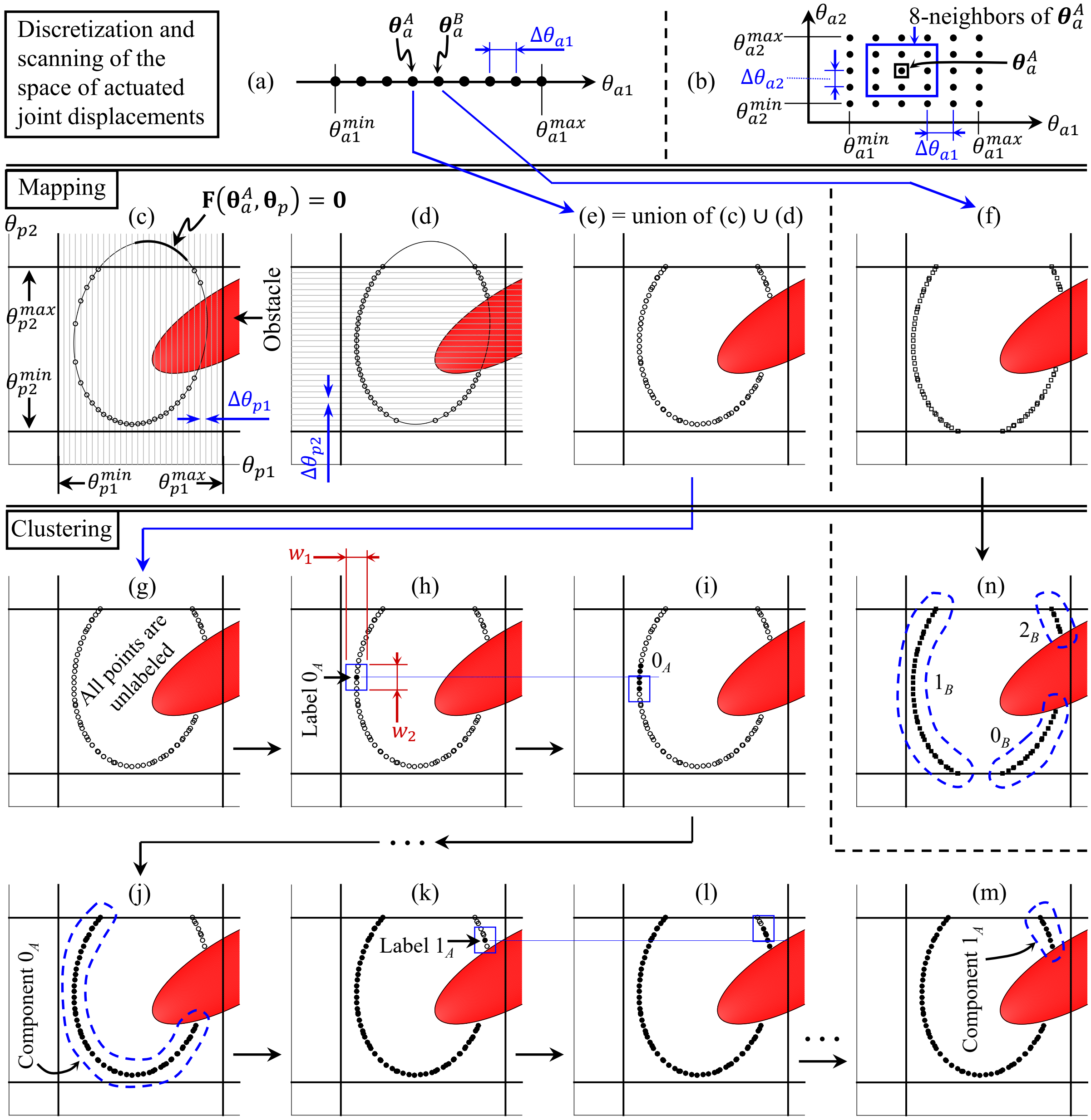

Figure 5: (a) Discretization of a one-dimensional space of actuated joints. (b) Discretization of a two-dimensional space of actuated joints. (c-f) Steps for mapping the free-swinging manifolds. (g-n) Steps for clustering these manifolds and identify different components.

coordinate θ_{p1} between its joint limits $[\theta_{p1}^{min}, \theta_{p1}^{max}]$ with a desired step size $\Delta\theta_{p1}$ as shown in Figure 5c, obtaining a sequence $\theta_{p1}^{i} = \theta_{p1}^{min} + i \cdot \Delta\theta_{p1}$ ($i = 0, 1, \ldots$). For each θ_{p1}^{i}, Equation (1) becomes a nonlinear system with the same number of equations and unknowns, which can be solved by any suitable method (preferably algebraic elimination methods) to obtain all real solutions, which are finite in number. For each θ_{p1}^{i}, these real solutions are the intersection points (shown as empty circles in Figure 5c) between the curves implicitly defined by $\mathbf{F}\left(\theta_a^A, \theta_p\right) = \mathbf{0}$ and the hyperplane $\theta_{p1} = \theta_{p1}^{i}$. By sweeping θ_{p1} from θ_{p1}^{min} to θ_{p1}^{max} and storing all these intersection points, we obtain a collection of points that approximate these implicitly-defined one-dimensional manifolds. During this process, the intersection points that violate joint limits for θ_{p2} or fall in collision regions (elliptical shaded red region in Figure 5c) are discarded, as these are unfeasible configurations.

After sweeping θ_{p1}, this process is repeated but sweeping θ_{p2} between its joint limits, as illustrated in Figure 5d. This is necessary because sweeping only θ_{p1} yields a point cloud that poorly approximates the manifold curves in the regions where these curves have vertical tangent, where the density of intersection points is lower, as seen in Figure 5c. Similarly, sweeping only θ_{p2} yields point clouds with low density near the parts of the curves with horizontal tangent. If there were more passive joints, this sweeping process would be repeated for each of them. The point clouds obtained after sweeping all passive joints are joined as shown in Figure 5e, obtaining an everywhere-dense point cloud that accurately approximates the manifolds for $\boldsymbol{\theta}_a = \boldsymbol{\theta}_a^A$, without low-density regions. This mapping process is repeated for each node $\boldsymbol{\theta}_a^B$ neighbor of $\boldsymbol{\theta}_a^A$, obtaining a dense point cloud represented by empty squares in Figure 5f, which approximates the free-swinging manifolds for $\boldsymbol{\theta}_a = \boldsymbol{\theta}_a^B$.

3.1.2. Clustering

After mapping the manifolds for the current node $\boldsymbol{\theta}_a^A$, it is necessary to identify the different connected components of these manifolds, and this is done through a clustering process that will assign a label to each intersection point obtained, indicating the manifold to which each point belongs. Initially, all points are unlabeled, which are represented by empty circles in Figure 5g. One of these unlabeled points is selected randomly, and is given a label 0_A, which means that this point belongs to the component 0 of the manifolds for $\boldsymbol{\theta}_a^A$. This selected point is shown filled in Figure 5h, which also shows a small box centered at this point, with size w_i along each axis θ_{pi}. Next, the set $\mathcal{P}$ of all the unlabeled points that fall in this box are searched efficiently using k-dimensional (k-d) trees [43], and these points are given the same label (0_A) as the center of the box, as they are considered to be path-connected to the center of the box due to proximity. These points $\mathcal{P}$, which are no longer unlabeled, are shown filled in Figure 5i. Now this process is repeated: for each point $p \in \mathcal{P}$, the same box is centered at p (Figure 5i) and all unlabeled points falling in it are given the same label as the center of the box. This process is repeated recursively until no more unlabeled points fall in the box, which occurs when the complete connected component 0_A has been identified, as shown in Figure 5j.

After a connected component of the manifolds has been completely identified, another unlabeled point is selected randomly and is given the next label 1_A (Figure 5k), and the process described in the previous paragraph is repeated until the connected component 1_A is completely identified (Figures 5l and 5m). For the current example node $\boldsymbol{\theta}_a^A$, the free-swinging manifolds have two connected components: 0_A identified in Figure 5j, and component 1_A identified in Figure 5m. This clustering process is repeated for the neighbor node $\boldsymbol{\theta}_a^B$ in Figure 5n to identify three components: 0_B, 1_B and 2_B.

3.1.3. Matching

The third and last operation of the method requires comparing and *matching* the free-swinging manifolds between the current node $\boldsymbol{\theta}_a^A$ and each of its neighbors. When $\boldsymbol{\theta}_a^A$ belongs to a one-dimensional grid as in Figure 5a, the neighbors of $\boldsymbol{\theta}_a^A$ are the nodes at its left and right sides. When $\boldsymbol{\theta}_a^A$ belongs to a two-dimensional grid as in Figure 5b, the neighbors of $\boldsymbol{\theta}_a^A$ are the 8-neighbors that surround $\boldsymbol{\theta}_a^A$, as illustrated in Figure 5b. For three-dimensional grids, we would pick the 26 neighbors surrounding the central box of a $3 \times 3 \times 3$ three-dimensional array, and so on for higher dimensions.

The objective of the matching operation is to compare the free-swinging manifolds identified for the current node $\boldsymbol{\theta}_a^A$ to the manifolds of each of its neighbors $\boldsymbol{\theta}_a^B$, to discover if any manifold degenerates into a point while moving the actuated joints between $\boldsymbol{\theta}_a^A$ and $\boldsymbol{\theta}_a^B$. Continuing with the example of Figure 5, take the manifolds 0_A and 1_A identified for $\boldsymbol{\theta}_a^A$, these are represented in Figure 6a as filled circles. Similarly, the manifolds 0_B, 1_B, and 2_B identified for $\boldsymbol{\theta}_a^B$ are represented as filled squares in Figure 6a. By inspecting Figure 6a, it appears that, when moving the actuated joints from $\boldsymbol{\theta}_a^A$ to $\boldsymbol{\theta}_a^B$, the manifolds of $\boldsymbol{\theta}_a^A$, which are parts of an elliptic closed curve, expand outwards until becoming the manifolds of $\boldsymbol{\theta}_a^B$. During this expansion, manifold 1_A transforms into 2_B, while manifold 0_A splits into manifolds 1_B and 0_B. This splitting occurs after part of manifold 0_A, while expanding, intersects the forbidden area below θ_{p2}^{min}.

The transformation of manifolds from $\boldsymbol{\theta}_a^A$ to $\boldsymbol{\theta}_a^B$, which was deduced by visual inspection of Figure 6a as explained in the previous paragraph, can be estimated by the following matching algorithm, which is very similar to the clustering operation described in Section 3.1.2. First, take a small box with size w_i' along each axis θ_{pi}, and center it at any point of the manifolds of $\boldsymbol{\theta}_a^A$, like the red point c_A shown in Figure 6a, which belongs to manifold 0_A. Next, use k-d trees to efficiently retrieve the set $\mathcal{P}_B$ of points of the manifolds of $\boldsymbol{\theta}_a^B$ (i.e., the filled blue squares of Figure 6a) that fall in this box. Each retrieved point $p_i \in \mathcal{P}_B$, will have a (possibly different) label $L(p_i) \in \{0_B, 1_B, 2_B, \dots\}$ of the manifold

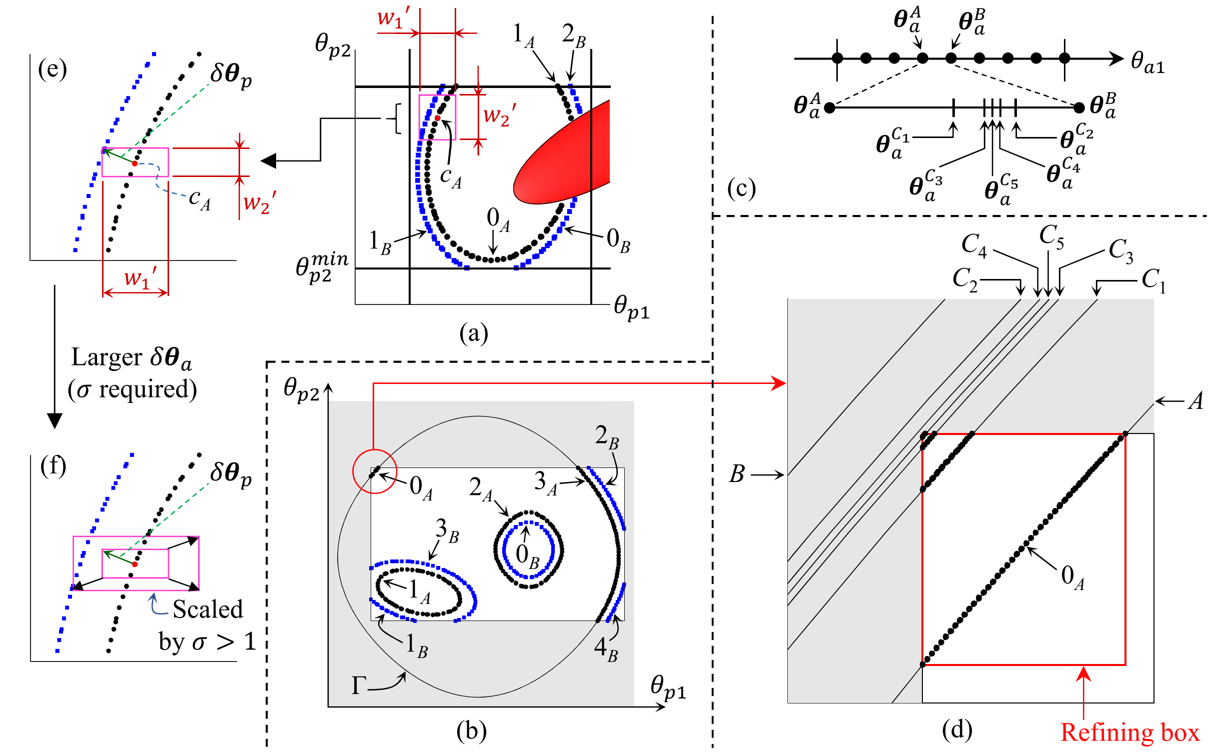

Figure 6: (a) Example of matching operation for the manifolds of Figures 5e and 5f. (b) Another example of matching that includes the vanishing of a manifold after it degenerates into a point. (c-d) Bisection method to refine the shrinking of a manifold into a point. (e-f) A possible choice for the size w_i' of the matching box.

to which it belongs (for the example box shown in Figure 6a, all points in this box belong to manifold 1_B). Thus, due to proximity, we can assume that the manifold 0_A, to which c_A belongs, transforms into the set of manifolds $\{L(p_1), L(p_2), L(p_3), \dots\}$, for all $p_i \in \mathcal{P}_B$. This process is repeated by iterating the center c_A over all points of the manifolds of $\boldsymbol{\theta}_a^A$, and the result of this operation is summarized in a *matching matrix* **M** as explained next.

The matching matrix **M** is a binary matrix having as many rows and columns as there are manifolds for $\boldsymbol{\theta}_a^A$ and $\boldsymbol{\theta}_a^B$, respectively. The entry of **M** at row i and column j is 1 if the i-th manifold of $\boldsymbol{\theta}_a^A$ transforms into the j-th manifold of $\boldsymbol{\theta}_a^B$, and 0 otherwise. These entries are filled during the execution of the matching algorithm described in the previous paragraph. For the example of Figure 6a, the resulting matching matrix is:

$$\mathbf{M} = \begin{array}{c} \begin{array}{ccc} 0_B & 1_B & 2_B \end{array} \\ \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array} \begin{array}{c} \\ 0_A \\ 1_A \end{array} \tag{4}$$

The two ones in the first row of (4) mean that manifold 0_A transforms into manifolds 0_B and 1_B after splitting, and the one in the second row means that manifold 1_A transforms into manifold 2_B. The use of a binary matrix to represent the results of the matching operation is an improvement with respect to the method presented in [34], where this result was represented using a more obscure notation based on Set Theory.

Now consider Figure 6b, which shows a more elaborate example of the matching operation. It exhibits the matching between manifolds of two hypothetical neighbor nodes $\boldsymbol{\theta}_a^A$ and $\boldsymbol{\theta}_a^B$. Manifolds for $\boldsymbol{\theta}_a^A$ are shown as black circles, whereas manifolds for $\boldsymbol{\theta}_a^B$ are shown as blue squares, and the gray shaded area is forbidden, being out of joint limits. For $\boldsymbol{\theta}_a^A$, two manifolds (0_A and 3_A) are open curves that are the parts of a bigger curve Γ within the joint limits, and other two manifolds (1_A and 2_A) are closed curves. For $\boldsymbol{\theta}_a^B$, four manifolds ($2_B, 4_B, 3_B$ and 1_B) are open curves, and one manifold (0_B) is a closed curve. By visual analysis of Figure 6b, it appears that, when moving the actuated joints from $\boldsymbol{\theta}_a^A$ to $\boldsymbol{\theta}_a^B$, the bigger curve Γ expands outward, splitting manifold 3_A into 2_B and 4_B, and producing the progressive shrinking of manifold 0_A until it degenerates into a point and vanishes. Regarding manifold 2_A, this closed curve contracts to become 0_B. Finally, the closed curve 1_A expands and exceeds the joint limits, splitting into manifolds 1_B and 3_B.

For the hypothetical example of Figure 6b, the matching algorithm described previously would produce the following matching matrix:

$$\mathbf{M} = \begin{array}{c} \begin{array}{ccccc} 0_B & 1_B & 2_B & 3_B & 4_B \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix} \end{array} \begin{array}{c} \\ 0_A \\ 1_A \\ 2_A \\ 3_A \end{array} \tag{5}$$

which summarizes the visual analysis performed above.

Recall that we are interested in detecting the degeneracy or shrinking of manifolds into points, which suppresses uncontrollable free-swinging motions at finite level. Typically, the shrinking of a manifold into a point occurs just before it vanishes, like manifold 0_A in Figure 6b. This is what occurs in the vast majority of cases, for generic situations. Therefore, we decide to detect the shrinking of manifolds into points by detecting their vanishing, which is easier to detect. There are, however, two situations where this decision may fail, namely:

- A positive-dimensional manifold may suddenly vanish without shrinking into a point first. This would happen, for example, if we had a manifold in Figure 6b that was a straight horizontal segment that translated vertically as the actuated joints are moved from $\boldsymbol{\theta}_a^A$ to $\boldsymbol{\theta}_a^B$. If we had such a horizontal segment manifold in Figure 6b, when this segment crossed the upper or lower joint limits of θ_{p2} into the forbidden region, the whole segment would disappear suddenly, without shrinking into a point first. A real example of a one-dimensional self-motion manifold that vanishes without degenerating first into a point appears when studying the positioning self-motions of the AAI redundant arm [44]. This situation is undesirable in our study, as we want manifolds to shrink into points just before vanishing, since these point manifolds suppress uncontrollable free-swinging motions. This undesirable situation can be detected and discarded by the bisection method that will be described later, which detects the vanishing of a manifold when it is sufficiently small to be regarded as a point.

- A manifold shrinks into a point without vanishing. To illustrate this, imagine a hypothetical underactuated manipulator with two passive joints $(\theta_{p1}, \theta_{p2})$ and one actuated joint θ_{a1}, whose constraint equation (1) is: $\theta_{p1}^2 + \theta_{p2}^2 = \theta_{a1}$. For a given value of $\theta_{a1} > 0$, the free-swinging manifold of such a manipulator is a circle of radius $\sqrt{\theta_{a1}}$. As $\theta_{a1} > 0$ approaches zero, this circle shrinks until becoming a point for $\theta_{a1} = 0$. After this, if θ_{a1} becomes negative, the constraint equation has no real solutions and the manifold vanishes. Now consider the same problem, but with a modified kinematic constraint where θ_{a1} appears squared: $\theta_{p1}^2 + \theta_{p2}^2 = \theta_{a1}^2$. In this modified situation, if we start with $\theta_{a1} > 0$, the manifold is a circle of radius θ_{a1} that shrinks into a point as θ_{a1} decreases and approaches zero. After this, if θ_{a1} becomes negative, the manifold is again a circle of radius $|\theta_{a1}|$. This example illustrates the shrinking of a manifold into a point without ever vanishing, which makes it impossible to detect the degeneracy of the manifold into a point by detecting its vanishing as suggested in this paper. However, this situation is unusual and will be omitted in this paper, which will focus on locking configurations for which the free-swinging manifolds vanish after shrinking into a point.

Let us return now to the matching example of Figure 6b, summarized by the matching matrix of (5). As explained above, the manifolds of interest in this paper are those that vanish after degenerating into a point, like manifold 0_A of Figure 6b. Note in (5) that this situation is reflected by the first null row in the matching matrix $\mathbf{M}$, which means that manifold 0_A does not transform into any manifold when moving the actuated joints from $\boldsymbol{\theta}_a^A$ to $\boldsymbol{\theta}_a^B$. Analogously, a null column of $\mathbf{M}$ would also denote the vanishing of a manifold when moving in the reverse direction, from $\boldsymbol{\theta}_a^B$ to $\boldsymbol{\theta}_a^A$.

Whenever $\mathbf{M}$ has a null row or column, we know that, for some point $\boldsymbol{\theta}_a^*$ between $\boldsymbol{\theta}_a^A$ and $\boldsymbol{\theta}_a^B$, a manifold shrinks into a point and then vanishes. This $\boldsymbol{\theta}_a^*$ is the *locking configuration* sought in this paper, to which we can drive the actuated joints after suffering FSJF in order to suppress the free-swinging motions of the underactuated manipulator. In the original version of this method [34], which was applied to fully-actuated kinematically-redundant robots, this point $\boldsymbol{\theta}_a^*$ was taken as the midpoint of $\boldsymbol{\theta}_a^A$ and $\boldsymbol{\theta}_a^B$ for simplicity. However, this is insufficient for the present paper, where we need a more precise estimate of $\boldsymbol{\theta}_a^*$ in order to properly assess the stability of the locking configuration according to Section 5. To that end, we introduce a refinement step based on a bisection algorithm, as explained next.

Assume that a manifold like 0_A in Figure 6b vanishes. Figure 6d shows a magnified view of the area near 0_A. In general, the vanishing manifold 0_A will not be small enough to be considered as a point yet. For the neighbor node $\boldsymbol{\theta}_a^B$, this manifold does not exist, which is reflected by the curve $\boldsymbol{B}$ (which is a solution of (1) for $\boldsymbol{\theta}_a = \boldsymbol{\theta}_a^B$) in Figure 6d not having points in the feasible region within the joint limits. A bisection process illustrated in Figure 6c-d is now applied, where manifolds will be mapped recursively using the algorithm described in Section 3.1.1 inside the minimum box that contains 0_A ("Refining box" in Figure 6d). The midpoint $\boldsymbol{\theta}_a^{C_1}$ between $\boldsymbol{\theta}_a^A$ and $\boldsymbol{\theta}_a^B$ is computed as shown in Figure 6c, and its manifold $\boldsymbol{C}_1$ is sampled in the refining box. Since this manifold has points in the refining box, it substitutes 0_A as a better approximation of the shrunk manifold, node $\boldsymbol{\theta}_a^{C_1}$ substitutes node $\boldsymbol{\theta}_a^A$, and a new midpoint $\boldsymbol{\theta}_a^{C_2}$ between $\boldsymbol{\theta}_a^{C_1}$ and $\boldsymbol{\theta}_a^B$ is chosen. Then, since the manifold $\boldsymbol{C}_2$ has no points in the refining box, node $\boldsymbol{\theta}_a^{C_2}$ substitutes $\boldsymbol{\theta}_a^B$ and a new midpoint $\boldsymbol{\theta}_a^{C_3}$ is chosen between $\boldsymbol{\theta}_a^{C_1}$ and $\boldsymbol{\theta}_a^{C_2}$. This process continues recursively, by computing always the midpoint ($\boldsymbol{\theta}_a^{C_4}$, $\boldsymbol{\theta}_a^{C_5}$...) between two nodes in the space of $\boldsymbol{\theta}_a$: one node for which the manifold has valid samples in the refining box, and another node for which the manifold has no valid samples in the box. This process stops when the manifold has a small number of sampled points in the refining box (e.g., ≤ 2 points), like manifold $\boldsymbol{C}_5$ in Figure 6d. The midpoint that stops the algorithm ($\boldsymbol{\theta}_a^{C_5}$ in Figure 6c) is taken as an approximation of the locking configuration for which the vanishing manifold 0_A has shrunk into a point, suppressing free-swinging motions. This process must be repeated for each vanishing manifold, i.e., for each null row or column of the matching matrix $\mathbf{M}$.

In the following sections, this method will be illustrated with several examples, discussing also the stability of locking configurations in Section 5.

4. Example 1: underactuated R$\underline{\text{P}}$RRR parallel robot

In this section, we will illustrate the method with a two-DOF manipulator that suffers FSJF. Consider the $\underline{\text{RP}}$RRR manipulator shown in Figure 7a, which consists of a closed kinematic chain with the following revolute (R) and prismatic (P) joints: O (R), A (P), B (R), C (R), and D (R). Initially, the manipulator is fully actuated, where joints O and A are actuated (that is why they are underlined in $\underline{\text{RP}}$RRR). The actuated joint displacements are $\rho > 0$ and ϕ, whereas angles θ_2 and θ_3 are passive, with limits -2.508 rad $\leq \theta_2 \leq 3.023$ rad and -1.911 rad $\leq \theta_3 \leq 2.419$ rad, as in [42]. A gripper is rigidly mounted on link l_3 to perform manipulation tasks.

Now, assume that joint ϕ suffers FSJF, so that joint O becomes passive. In that case, the manipulator can be denoted as R$\underline{\text{P}}$RRR, with one actuated joint $\boldsymbol{\theta}_a = [\rho]^T$ and three passive joints $\boldsymbol{\theta}_p = [\theta_2, \theta_3, \phi]^T$. For a fixed ρ, the manipulator becomes an unactuated four-bar linkage that can perform uncontrollable motions along the free-swinging manifolds. The shape of these manifolds will depend on ρ. In this section, the method described in Section 3.1 will be used to find the locking configurations ρ^* for which free-swinging manifolds shrink into points.

The interest in this closed-chain manipulator is justified next. In [42], Luck and Lee studied the evolution of self-motion manifolds of the kinematically-redundant manipulator of Figure 1a. They studied how these self-motion manifolds change in the space of angles (θ_2, θ_3) when varying the distance ρ between the gripper and the origin (θ_1 is neglected from this study because it does not affect ρ). The mathematical relation between ρ and (θ_2, θ_3) is the same for the kinematically-redundant manipulator of Figure 1a and for the underactuated robot of Figure 7a. Thus, the self-motion manifolds of the former robot are the same as the free-swinging manifolds of the latter, and this will allow us to compare our results with those reported in [42].

Particularizing Equation (1) for this robot yields:

$$\mathbf{F}(\boldsymbol{\theta}_a, \boldsymbol{\theta}_p) = \begin{bmatrix} l_1 + l_2 \cos\theta_2 + l_3 \cos(\theta_2 + \theta_3) - \rho\cos\phi \\ l_2 \sin\theta_2 + l_3 \sin(\theta_2 + \theta_3) - \rho\sin\phi \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{6}$$

where $l_1 = 0.06$ m, $l_2 = 0.03$ m, and $l_3 = 0.02$ m. Inequalities (3) are:

$$\mathbf{G}(\boldsymbol{\theta}_a, \boldsymbol{\theta}_p) = \begin{bmatrix} \theta_2 - \theta_2^{min} \\ -\theta_2 + \theta_2^{max} \\ \theta_3 - \theta_3^{min} \\ -\theta_3 + \theta_3^{max} \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \text{with:} \quad \begin{array}{ll} \theta_2^{min} = -2.508 \text{ rad}, & \theta_2^{max} = 3.023 \text{ rad}, \\ \theta_3^{min} = -1.911 \text{ rad}, & \theta_3^{max} = 2.419 \text{ rad}. \end{array} \tag{7}$$

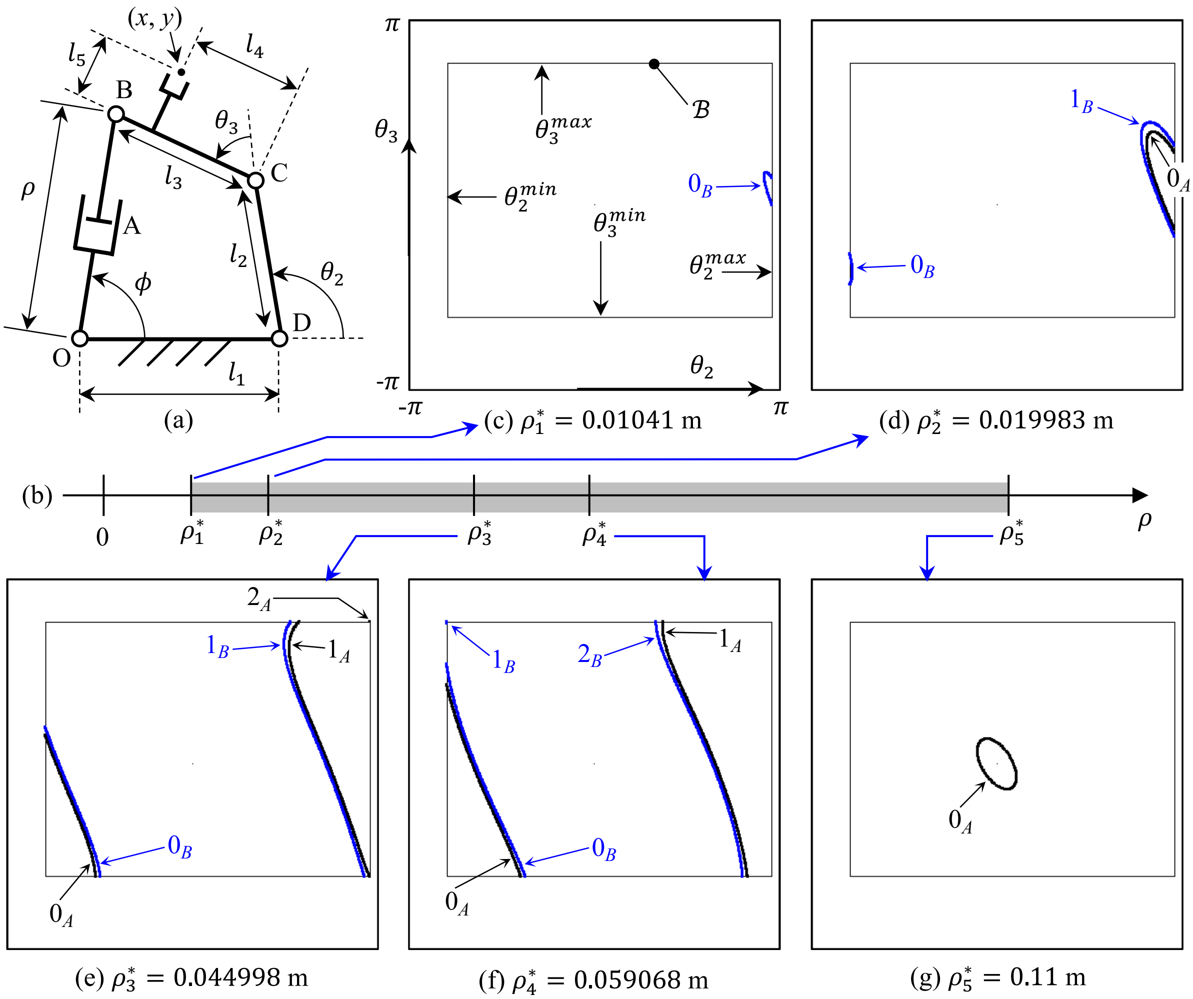

Figure 7: (a) Underactuated RPRRR robot. (b) Locking configurations ρ_i^* identified in the space of the only actuated joint ρ. (c-g) Free-swinging manifolds in the vicinity of each locking configuration.

For a generic value of $\rho > 0$, the solution set of Equations (6) and (7) is a collection of disjoint one-dimensional manifolds in the space of passive joint displacements $(\theta_2, \theta_3, \phi)$. Actually, since the inequalities (7) do not involve ϕ, it is sufficient to study the projection of these manifolds on plane (θ_2, θ_3), as the value of ϕ is uniquely determined by (θ_2, θ_3) as follows: $\phi = \text{atan2}\left(l_2 \sin\theta_2 + l_3 \sin(\theta_2+\theta_3), l_1 + l_2\cos\theta_2 + l_3\cos(\theta_2+\theta_3)\right)$. Thus, for simplicity and clarity, and to ease the comparison with [42], in this example we will visualize the free-swinging manifolds projected on plane (θ_2, θ_3).

Next, the method described in Section 3.1 will be used for obtaining the locking configurations ρ^* for which some free-swinging manifolds degenerate into points. The method was implemented in Java language and run on an Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz processor with 16GB RAM[1], under the following conditions:

- The ρ axis is discretized between $\rho^{min} = 0$ and $\rho^{max} = l_1 + l_2 + l_3 + 0.1l_1$ into a grid of $N = 50$ nodes, i.e., using a step of: $\Delta\rho = (\rho^{max} - \rho^{min})/(N-1) = 0.0023673$ m. The upper bound ρ^{max} is selected considering that the maximum feasible value of ρ occurs when all links are aligned ($\rho = l_1 + l_2 + l_3$).

[1]All examples presented in this paper were run on this computer. It is necessary to disclose the features of this computer in order to have a reference when reporting the CPU time taken by the method for each example. Note that the original method presented in [34] relied heavily on parallel computation in order to speed-up the calculations, but parallelization of computations was not needed at all in the present paper.

- For each value of ρ, the solution set of Equation (6) in the space of $(\theta_2, \theta_3, \phi)$ is a set of curves that are sampled using the mapping operation described in Section 3.1.1. To that end, each passive joint is varied on a grid of 200 equally spaced points between its joint limits. When each passive joint is varied, the other two passive joints are analytically solved from (6), which can be reduced to a quadratic equation.

- The size w_i of the box (shown in Figure 5h) used for the clustering algorithm of Section 3.1.2 equals four times the step used for sweeping each passive joint during the mapping operation.

- To choose the sizes w_i' of the matching box of Figure 6a centered at c_A, we first differentiate (1):

$$\mathbf{F}_{\theta_a}\delta\boldsymbol{\theta}_a + \mathbf{F}_{\theta_p}\delta\boldsymbol{\theta}_p = \mathbf{0} \xrightarrow{\text{min. norm}} \delta\boldsymbol{\theta}_p = \left[\delta\theta_{p1}, \delta\theta_{p2}, \dots, \delta\theta_{ps}\right]^T = -\mathbf{F}_{\theta_p}^T\left(\mathbf{F}_{\theta_p}\mathbf{F}_{\theta_p}^T\right)^{-1}\mathbf{F}_{\theta_a}\delta\boldsymbol{\theta}_a \tag{8}$$

  where $\delta\boldsymbol{\theta}_a = \boldsymbol{\theta}_a^B - \boldsymbol{\theta}_a^A$, while $\mathbf{F}_{\theta_a} \in \mathbb{R}^{m\times r}$ and $\mathbf{F}_{\theta_p} \in \mathbb{R}^{m\times s}$ $(s > m)$ are the Jacobians of $\mathbf{F}(\boldsymbol{\theta}_a, \boldsymbol{\theta}_p)$ with respect to $\boldsymbol{\theta}_a$ and $\boldsymbol{\theta}_p$, respectively. These Jacobians are evaluated at $\boldsymbol{\theta}_a = \boldsymbol{\theta}_a^A$ and $\boldsymbol{\theta}_p = c_A$. When the difference $\delta\boldsymbol{\theta}_a$ between the neighboring nodes $\boldsymbol{\theta}_a^A$ and $\boldsymbol{\theta}_a^B$ is very small, the minimum-norm solution of $\delta\boldsymbol{\theta}_p$ in (8) approximates the shortest vector that connects c_A with the manifolds of node $\boldsymbol{\theta}_a^B$, as illustrated in Figure 6e. The components of this shortest vector are chosen as (half of) the sides w_i' of the matching box, i.e.: $w_i' = 2|\delta\theta_{pi}| \cdot \sigma$. In general, if the difference between $\boldsymbol{\theta}_a^A$ and $\boldsymbol{\theta}_a^B$ is small but not sufficiently, the linear approximation of (8) may not be valid and $\delta\boldsymbol{\theta}_p$ may be too short to connect c_A with the manifolds of $\boldsymbol{\theta}_a^B$ as shown in Figure 6f, so this box should be slightly scaled by $\sigma > 1$, which we will choose as $\sigma = 2$.

Under these conditions, the method takes an average of 0.18 seconds to obtain the five locking configurations ρ_i^* shown in Figure 7b. The solution set of Equations (6) and (7) is non-empty only for values of ρ in the interval $[0.01041, 0.11]$, shown in gray in Figure 7b.

Figures 7c-g represent the plane (θ_2, θ_3) for $-\pi \le \theta_2 \le \pi$ and $-\pi \le \theta_3 \le \pi$, where a box $\mathcal{B}$ that represents the joint limits (7) is also represented. Each of the Figures 7c-g, which is associated to each locking configuration ρ_i^*, represents the free-swinging manifolds corresponding to a value of ρ slightly smaller than ρ_i^* (manifolds 0_A, 1_A... represented by black curves) and for a value slightly greater than ρ_i^* (manifolds 0_B, 1_B... represented by blue curves). According to Figures 7c-g, we can make the following observations:

- Figure 7c: when ρ is slightly greater than ρ_1^*, there is a single manifold 0_B that degenerates into a point when $\rho \to \rho_1^{*+}$ (i.e., when ρ tends to ρ_1^* from the right), and vanishes when ρ decreases below ρ_1^*. Since there are no manifolds for $\rho < \rho_1^*$, the manipulator cannot be assembled for $\rho < \rho_1^*$ while satisfying (7).

- Figure 7d: when ρ is slightly smaller than ρ_2^*, there is a single manifold 0_A which is an open curve. When ρ is slightly above ρ_2^*, there are two manifolds 0_B and 1_B, where 0_B is very small and about to shrink into a point. When crossing ρ_2^* in the negative direction, manifold 1_B transforms into manifold 0_A, whereas manifold 0_B degenerates into a point and vanishes.

- Figure 7e: when ρ is slightly smaller than ρ_3^*, there are three manifolds: 0_A and 1_A are large open curves, whereas 2_A is a tiny open curve about to shrink into a point. When ρ increases above ρ_3^*, manifold 2_A degenerates into a point and vanishes, whereas manifolds 0_A and 1_A transform into the manifolds 0_B and 1_B, respectively.

- Figure 7f: this situation is similar to the previous one. When ρ is slightly greater than ρ_4^*, there are three manifolds: two large open curves $(0_B, 2_B)$, and a tiny open curve (1_B). This curve 1_B degenerates into a point and vanishes when ρ decreases below ρ_4^*, whereas manifolds 0_B and 2_B transform into 0_A and 1_A, respectively.

- Figure 7g: when ρ is slightly smaller than ρ_5^*, there is a single manifold which is a small loop 0_A around the origin of plane (θ_2, θ_3). When crossing ρ_5^* in the positive direction, 0_A shrinks into the origin and vanishes. The manipulator cannot be assembled for $\rho > \rho_5^*$ because the solution set of (6) is empty in the real domain.

In the supplementary video ex1.mp4 attached to this paper, the reader can observe the evolution of free-swinging manifolds as ρ increases from 0 to 0.12, observing the transition between Figures 7c-g in motion.

Figures 8b-f show the configurations adopted by the robot when ρ equals each of the locking configurations ρ_i^* of Figure 7b, with (θ_2, θ_3) taking the value of the manifold that degenerates into a point in each case. In these figures, the joint limits of (θ_2, θ_3) have been represented as circular sectors, as illustrated in Figure 8a. Although all these locking configurations suppress uncontrollable free-swinging motions at *finite* level through the shrinking of free-swinging manifolds into points, not all of them suppress such motions at *differential* or velocity level, as it can easily seen in Figures 8b-f.

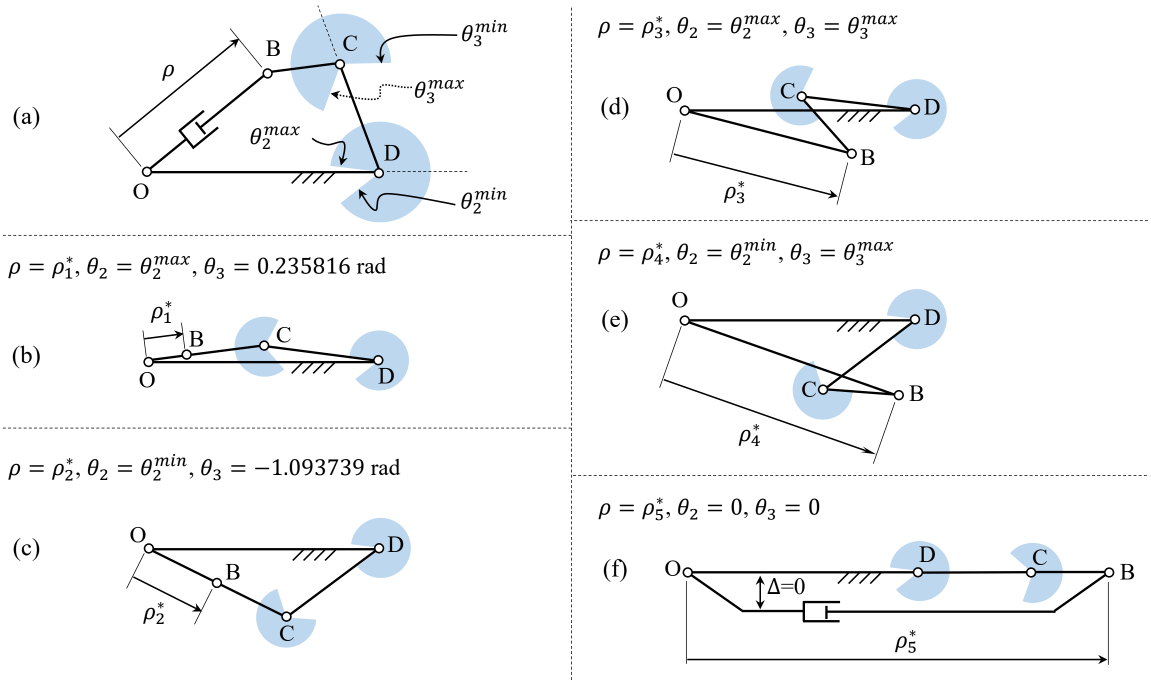

Figure 8: (a) Joint limits represented as circular sectors (the gripper is omitted). (b-f) Postures adopted by the robot for the five locking configurations of Figure 7b. In (b-e), the symbol of the prismatic joint (⊏⊢) is omitted in link OB for clarity of the drawings, but OB is a linear actuator as in (a). In (f), the linear actuator OB should actually be aligned with ODCB, but an artificial offset Δ is represented also for clarity.

Take the locking configuration ρ_2^* illustrated in Figure 8c for example. Any external force or torque exerted on bar CD trying to rotate it about D in either direction, can be compensated by the active force exerted by the linear actuator OB. Or, if the external wrench tries to rotate CD clockwise, it can even be balanced passively by the unidirectional reaction exerted by the joint limit θ_2^{min}, which is a mechanical stop that prevents CD from rotating clockwise. However, the aligned kinematic chain OBC is not rigid at all: it cannot resist forces perpendicular to OBC or torques, and joint B can be moved infinitesimally in the direction perpendicular to OBC, even if we lock ρ, O, and C. This is, the robot admits passive joint velocities ($\dot{\theta}_3 \neq 0$) even if the actuated joint is locked ($\dot{\rho} = 0$). For analogous reasons, the locking configurations ρ_1^* and ρ_5^* of Figures 8b and 8f also admit passive joint velocities and cannot resist external wrenches even if the actuated joint ρ is perfectly locked. We will call these locking configurations *unstable*.

In the following section, we will provide two criteria to classify locking configurations into *unstable* or *stable*, depending whether they admit uncontrollable free-swinging motions at velocity level, or not.

Before proceeding to the next section, however, let us compare the results of Figure 7 to [42]. In [42], Luck and Lee obtained the critical values of ρ for which the self-motion manifolds of the redundant manipulator of Figure 1a suffer changes of topology, such as splitting or merging of different manifolds, or opening of a closed curve into an open one. As shown in Table 1, among these critical values, they obtained our locking configurations ρ_i^* of Figure 7b except for ρ_4^*, which was not reported in [42]. Moreover, the plots of self-motion manifolds in the plane (θ_2, θ_3) shown in [42] coincide with our plots of free-swinging manifolds shown in Figure 7. Thus, this comparison validates our analysis.

Table 1
Comparison of the locking configurations shown in Figure 7b of the present paper and the corresponding critical values reported in Figure 6 of [42]. *Actually, Figure 6 of [42] reports these values multiplied by 100 and rounded to one decimal place, because they studied a scaled robot 100 times larger than ours, with $l_1 = 6$, $l_2 = 3$, and $l_3 = 2$.

	Our value	Value reported in [42]*
ρ_1^*	0.01041	0.0104
ρ_2^*	0.019983	0.02
ρ_3^*	0.044998	0.045
ρ_4^*	0.059068	Missing
ρ_5^*	0.11	0.11

5. Stability of locking configurations

In the example of Section 4, it was shown that, for some critical values of the actuated joint displacements (the so-called *locking configurations*), some free-swinging manifolds degenerate into points, suppressing the *finite* uncontrollable motions. However, the examples of Figure 8 revealed that not all of these locking configurations imply a stable locking of the underactuated robot, as the robot may admit nonzero passive joint velocities, or, from a statics perspective, it cannot resist external forces or torques. Further analysis of each locking configuration is required, so that they can be classified as stable or unstable. This will be done in the present section by means of two criteria. Both criteria assume that the robot has been taken to a locking configuration, and analyze if the robot can be kept stably at that configuration.

5.1. Static criterion

Assume that the task to be performed by an n-DOF robot can be described by n position and/or orientation output coordinates of its gripper, and these coordinates are grouped in a vector $\boldsymbol{\xi} = [\xi_1, \dots, \xi_n]^T$, which can be explicitly written as a function $\mathbf{H}$ of the active and passive joint displacements:

$$\xi = \mathbf{H}(\boldsymbol{\theta}_a, \boldsymbol{\theta}_p) \tag{9}$$

Both equality constraints (1) and (9) can be combined into a single equality constraint:

$$\mathbf{L} = \begin{bmatrix} \mathbf{F}(\boldsymbol{\theta}_a, \boldsymbol{\theta}_p) \\ \mathbf{H}(\boldsymbol{\theta}_a, \boldsymbol{\theta}_p) - \xi \end{bmatrix} = \mathbf{0}_{(m+n)\times 1} \tag{10}$$

Now consider a Lagrangian dynamical system constrained by $m + n$ equality constraints (10) and d inequalities (3). The static equilibrium is achieved if the following equations (11)-(15) are satisfied [45], together with (10) and (3):

$$\boldsymbol{\tau}_\xi + \mathbf{L}_\xi^T \cdot \boldsymbol{\lambda} + \mathbf{G}_\xi^T \cdot \boldsymbol{\gamma} = U_\xi \tag{11}$$

$$\boldsymbol{\tau}_{\theta_a} + \mathbf{L}_{\theta_a}^T \cdot \boldsymbol{\lambda} + \mathbf{G}_{\theta_a}^T \cdot \boldsymbol{\gamma} = U_{\theta_a} \tag{12}$$

$$\boldsymbol{\tau}_{\theta_p} + \mathbf{L}_{\theta_p}^T \cdot \boldsymbol{\lambda} + \mathbf{G}_{\theta_p}^T \cdot \boldsymbol{\gamma} = U_{\theta_p} \tag{13}$$

$$\boldsymbol{\gamma} \geq \mathbf{0}_{d\times 1} \tag{14}$$

$$\gamma_j \cdot G^j = 0 \quad (j = 1, \dots, d) \quad \text{(recall that } G^j \text{ is the } j\text{-th component of } \mathbf{G} \text{ in (3))} \tag{15}$$

Where:

- U is the potential energy of the system.
- $\mathbf{L}_{\mathbf{v}}$ and $\mathbf{G}_{\mathbf{v}}$ are the Jacobian matrices of functions $\mathbf{L}$ and $\mathbf{G}$ with respect to $\mathbf{v}$ (with vector $\mathbf{v}$ being substituted by $\boldsymbol{\xi}$, $\boldsymbol{\theta}_a$, or $\boldsymbol{\theta}_p$). Similarly, $U_{\mathbf{v}}$ is the (column vector) gradient of U with respect to $\mathbf{v}$.
- $\boldsymbol{\tau}_{\xi}$ are external forces and torques exerted on the gripper along the directions of its output coordinates $\boldsymbol{\xi}$, whereas $\boldsymbol{\tau}_{\theta_a}$ are the driving forces/torques exerted by the actuators that drive the r actuated joints that remain healthy. $\boldsymbol{\tau}_{\theta_p}$ are the torques associated to the passive joint displacements. Since these joints do not have actuators, we have: $\boldsymbol{\tau}_{\theta_p} = \mathbf{0}_{s\times 1}$.
- $\boldsymbol{\lambda} = [\lambda_1, \dots, \lambda_{m+n}]^T$ are the Lagrangian multipliers associated to the equality constraints given in (10).
- $\boldsymbol{\gamma} = [\gamma_1, \dots, \gamma_d]^T$ are the Karush-Kuhn-Tucker (KKT) multipliers associated to the inequality constraints given in equation (3). These multipliers satisfy the non-negativity condition (14), which means that each $\gamma_j \geq 0$.
- Equation (15), together with (3) and (14), defines the *complementarity condition* [45]. According to this condition, if an inequality constraint $G^j \geq 0$ is not active (i.e., it is satisfied as a strict inequality: $G^j > 0$), then its corresponding KKT multiplier must be zero ($\gamma_j = 0$), and vice versa.

The next objective will be to determine the actuation forces and torques $\boldsymbol{\tau}_{\theta_a}$ that are necessary to keep static equilibrium at a locking configuration after FSJF, balancing the conservative forces and torques (U_{ξ} , U_{θ_a}, U_{θ_p}) derived from the potential U, as well as the external forces and torques $\boldsymbol{\tau}_{\xi}$ exerted on the gripper. To this end, first we group Equations (12) and (13), forming a system of $r + s = n + m$ scalar equations in the $n + m$ Lagrange multipliers $\boldsymbol{\lambda}$:

$$\begin{bmatrix} \mathbf{I}_r \\ \mathbf{0}_{s\times r} \end{bmatrix} \boldsymbol{\tau}_{\theta_a} + \begin{bmatrix} \mathbf{L}_{\theta_a}^T \\ \mathbf{L}_{\theta_p}^T \end{bmatrix} \boldsymbol{\lambda} + \begin{bmatrix} \mathbf{G}_{\theta_a}^T \\ \mathbf{G}_{\theta_p}^T \end{bmatrix} \boldsymbol{\gamma} = \begin{bmatrix} U_{\theta_a} \\ U_{\theta_p} \end{bmatrix} \tag{16}$$

Next, we solve the Lagrangian multipliers $\boldsymbol{\lambda}$ from (16), assuming that the $(n+m)\times(n+m)$ matrix that multiplies $\boldsymbol{\lambda}$ in (16) is nonsingular, obtaining:

$$\boldsymbol{\lambda} = \begin{bmatrix} \mathbf{L}_{\theta_a}^T \\ \mathbf{L}_{\theta_p}^T \end{bmatrix}^{-1} \left(\begin{bmatrix} U_{\theta_a} \\ U_{\theta_p} \end{bmatrix} - \begin{bmatrix} \mathbf{I}_r \\ \mathbf{0}_{s\times r} \end{bmatrix} \boldsymbol{\tau}_{\theta_a} - \begin{bmatrix} \mathbf{G}_{\theta_a}^T \\ \mathbf{G}_{\theta_p}^T \end{bmatrix} \boldsymbol{\gamma} \right) \tag{17}$$

Inserting (17) into (11) and rearranging terms yields:

$$\boldsymbol{\tau}_{\xi} + \mathbf{L}_{\xi}^T \begin{bmatrix} \mathbf{L}_{\theta_a}^T \\ \mathbf{L}_{\theta_p}^T \end{bmatrix}^{-1} \begin{bmatrix} U_{\theta_a} \\ U_{\theta_p} \end{bmatrix} - U_{\xi} = \mathbf{N} \begin{bmatrix} \boldsymbol{\tau}_{\theta_a} \\ \boldsymbol{\gamma} \end{bmatrix}, \quad \text{with:} \quad \mathbf{N} := \mathbf{L}_{\xi}^T \begin{bmatrix} \mathbf{L}_{\theta_a}^T \\ \mathbf{L}_{\theta_p}^T \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{I}_r & \mathbf{G}_{\theta_a}^T \\ \mathbf{0}_{s\times r} & \mathbf{G}_{\theta_p}^T \end{bmatrix} \in \mathbb{R}^{n\times(r+d)} \tag{18}$$

The objective is to balance *any* external force or torque $\boldsymbol{\tau}_{\xi}$ applied on the gripper. If $\boldsymbol{\tau}_{\xi}$ can take any value, the left hand side of (18) can be any vector of $\mathbb{R}^n$, regardless of the potential U. Therefore, the matrix product on the right-hand side of the same equation should be able to generate any vector of $\mathbb{R}^{n\times 1}$ in order to balance any external $\boldsymbol{\tau}_{\xi}$. If $\boldsymbol{\tau}_{\theta_a}$ and $\boldsymbol{\gamma}$ were not constrained, this balance would be guaranteed if rank($\mathbf{N}$) $= n$. However, according to (14), each component γ_j of $\boldsymbol{\gamma}$ should be non-negative. In addition, these components must satisfy the complementarity condition (15). Therefore, it is necessary to consider these constraints (14) and (15) when analyzing if the linear mapping on the right-hand side of (18) can generate any vector of $\mathbb{R}^n$, and this requires studying the range space of $\mathbf{N}$ with constrained domain.

The range of linear mappings with constrained domain was studied by Luck and Lee [42] in the context of velocity analysis of kinematically redundant robots with joint limits. In that context, when some joints reach their limits, they only admit velocities of one sign (analogously to our non-negativity constraint of Eq. 14), and in that case the Jacobian matrix that maps joint velocities to gripper velocities in redundant robots becomes a linear mapping with constrained domain, analogously to our matrix $\mathbf{N}$ in (18). One of the methods proposed in [42] to analyze the range of such constrained mappings is the *Column Extension*. We will use this method as explained in the following steps:

1. Identify each inequality constraint that is satisfied strictly ($G^j > 0$). Due to the complementarity condition (15), the associated KKT multiplier must be zero ($\gamma_j = 0$), so the column of $\mathbf{N}$ multiplied by $\gamma_j = 0$ can be removed.

2. After removing the columns indicated in the previous step, the right-hand side of (18) can be written as follows:

$$\mathbf{N} \cdot \begin{bmatrix} \boldsymbol{\tau}_{\theta_a} \\ \boldsymbol{\gamma} \end{bmatrix} = \left[\mathbf{N}_{\tau_{\theta_a}} \mid \mathbf{N}_{\gamma}\right] \cdot \begin{bmatrix} \boldsymbol{\tau}_{\theta_a} \\ \boldsymbol{\gamma} \end{bmatrix} = \left[\mathbf{N}_{\tau_{\theta_a}} \mid \hat{\mathbf{N}}_{\gamma}\right] \cdot \begin{bmatrix} \boldsymbol{\tau}_{\theta_a} \\ \hat{\boldsymbol{\gamma}} \end{bmatrix} \tag{19}$$

   where $\mathbf{N}_{\tau_{\theta_a}}$ comprises the columns of $\mathbf{N}$ that are multiplied by $\boldsymbol{\tau}_{\theta_a}$, $\mathbf{N}_{\gamma}$ comprises the columns of $\mathbf{N}$ that are multiplied by $\boldsymbol{\gamma}$, $\hat{\boldsymbol{\gamma}}$ only contains the nonzero entries of $\boldsymbol{\gamma}$ after removing the null γ_j in step 1, and $\hat{\mathbf{N}}_{\gamma}$ only comprises the columns of $\mathbf{N}_{\gamma}$ that are multiplied by $\hat{\boldsymbol{\gamma}}$. The right-hand side of (19) is a mapping where part of the domain is unconstrained ($\boldsymbol{\tau}_{\theta_a}$), whereas $\hat{\boldsymbol{\gamma}} \geq \mathbf{0}$. To rewrite (19) as a mapping where all domain variables must be non-negative, we apply the *Column Extension* method [42], by which the matrix is *extended* by repeating the columns that multiply $\boldsymbol{\tau}_{\theta_a}$, as follows:

$$\left[\mathbf{N}_{\tau_{\theta_a}} \mid \hat{\mathbf{N}}_{\gamma}\right] \cdot \begin{bmatrix} \boldsymbol{\tau}_{\theta_a} \\ \hat{\boldsymbol{\gamma}} \end{bmatrix} = \left[\mathbf{N}_{\tau_{\theta_a}} \mid -\mathbf{N}_{\tau_{\theta_a}} \mid \hat{\mathbf{N}}_{\gamma}\right] \cdot \begin{bmatrix} \boldsymbol{\tau}^+_{\theta_a} \\ \boldsymbol{\tau}^-_{\theta_a} \\ \hat{\boldsymbol{\gamma}} \end{bmatrix} = \tilde{\mathbf{N}} \cdot \begin{bmatrix} \boldsymbol{\tau}^+_{\theta_a} \\ \boldsymbol{\tau}^-_{\theta_a} \\ \hat{\boldsymbol{\gamma}} \end{bmatrix} \tag{20}$$

   where $\boldsymbol{\tau}^+_{\theta_a} \geq \mathbf{0}$ and $\boldsymbol{\tau}^-_{\theta_a} \geq \mathbf{0}$. All variables in the domain of the mapping $\tilde{\mathbf{N}}$ in (20) are nonnegative, and the range of this mapping can be studied more easily as explained in the next step, which was also proposed in [42].

3. To determine the range space of $\tilde{\mathbf{N}}$, first its columns are regarded as points $v_1, v_2, \dots v_k$ in $\mathbb{R}^n$. Let V denote the set of these points, including in V also the origin $\mathcal{O}$ of $\mathbb{R}^n$, i.e.: V$=\{v_1, v_2, \dots v_k, \mathcal{O}\}$. The convex hull of V is a polytope $\mathcal{P}$ in $\mathbb{R}^n$. Since the domain of $\tilde{\mathbf{N}}$ consists of nonnegative variables ($\boldsymbol{\tau}^+_{\theta_a}, \boldsymbol{\tau}^-_{\theta_a}, \hat{\boldsymbol{\gamma}}$), its range (or image) will consist of all points that can be contained in any positive scaling of $\mathcal{P}$, where the origin $\mathcal{O}$ is the only point of $\mathcal{P}$ that remains fixed after being scaled [42]. If the origin of $\mathbb{R}^n$ belongs to the interior of $\mathcal{P}$, then it is possible to scale $\mathcal{P}$ keeping $\mathcal{O}$ fixed so that any point of $\mathbb{R}^n$ belongs to the scaled polytope, and in this case the mapping is full-rank (Figure 9a). On the contrary, if the origin of $\mathbb{R}^n$ belongs to the boundary of $\mathcal{P}$, then positive scalings of this polytope with $\mathcal{O}$ kept fixed will always leave parts of $\mathbb{R}^n$ uncovered, and the mapping will not be full-rank in this case (Figure 9b). If the mapping $\tilde{\mathbf{N}}$ is full-rank, then it can generate any vector of $\mathbb{R}^n$ to balance the left-hand side of (18), and in that case the locking configuration is classified as stable.

It should be remarked that the static analysis presented here is worst-case or conservative, in the sense that it neglects stiction (static friction) effects that may contribute to balancing external forces and torques more easily. In general [45], Equations (11)-(13) may contain static friction terms that have been omitted here for simplicity, but including these terms in the model can only benefit the stabilization of a locking configuration against external forces and torques.

5.2. Velocity criterion

As an alternative to the static criterion presented in Section 5.1, this section presents a simpler criterion based on admissible passive joint velocities. Differentiating (1) with respect to time, and evaluating $\mathbf{F}_{\theta_p}$ at a locking configuration $(\boldsymbol{\theta}^*_a, \boldsymbol{\theta}^*_p)$ yields:

$$\mathbf{F}_{\theta_p}(\boldsymbol{\theta}^*_a, \boldsymbol{\theta}^*_p) \cdot \dot{\boldsymbol{\theta}}_p = \mathbf{0} \tag{21}$$

where $\mathbf{F}_{\theta_p}$ is the Jacobian matrix of $\mathbf{F}$ with respect to $\boldsymbol{\theta}_p$, and where it has been assumed that the actuated joints have been locked at $\boldsymbol{\theta}^*_a$, i.e., $\dot{\boldsymbol{\theta}}_a = \mathbf{0}$.

Solving (21) yields the nullspace tangent to the free-swinging manifolds at $\boldsymbol{\theta}^*_p$, i.e., the passive joint velocities $\dot{\boldsymbol{\theta}}^n_p$ that are admissible when the actuated joints have been locked at $\boldsymbol{\theta}^*_a$. However, the passive velocities $\dot{\boldsymbol{\theta}}^n_p$ that solve (21) must also satisfy unidirectional constraints that arise from the inequalities (3) that are active. Assume that, for the locking configuration $(\boldsymbol{\theta}^*_a, \boldsymbol{\theta}^*_p)$, some inequalities of (3) are active, i.e., some components G^i of $\mathbf{G}$ are zero when evaluated at $(\boldsymbol{\theta}^*_a, \boldsymbol{\theta}^*_p)$. In that case, the passive joint velocities must satisfy:

$$G^i_{\theta_p}(\boldsymbol{\theta}^*_a, \boldsymbol{\theta}^*_p) \cdot \dot{\boldsymbol{\theta}}_p \geq 0 \quad \forall\, G^i \text{ such that } G^i(\boldsymbol{\theta}^*_a, \boldsymbol{\theta}^*_p) = 0 \tag{22}$$

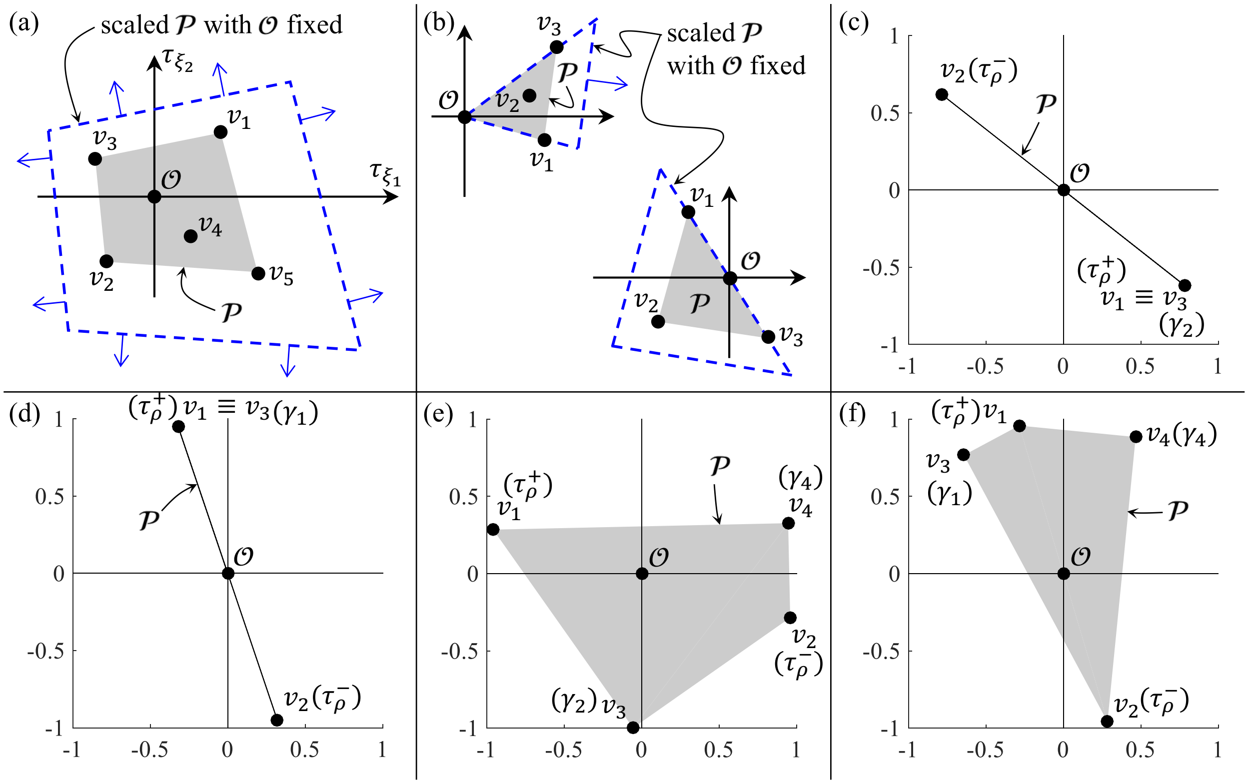

Figure 9: (a) Example where scaling $\mathcal{P}$ with $\mathcal{O}$ fixed can fully cover $\mathbb{R}^2$. (b) Examples where scaling $\mathcal{P}$ with $\mathcal{O}$ fixed cannot fully cover $\mathbb{R}^2$. (c-f) Polytopes for the locking configurations ρ_1^*, ...ρ_4^* of Figure 8.

where $G^i_{\theta_p}$ is the (row vector) gradient of G^i with respect to $\boldsymbol{\theta}_p$. Equation (22) means that the projection of the passive joint velocity $\dot{\boldsymbol{\theta}}_p$ on the line perpendicular to the hypersurface $G^i = 0$ at $(\boldsymbol{\theta}_a^*, \boldsymbol{\theta}_p^*)$ should point to the side of the hypersurface where $G^i > 0$. Otherwise, the constraint $G^i \geq 0$ would be violated.

If the only solution that satisfies both (21) and (22) is the zero vector $\dot{\boldsymbol{\theta}}_p = \mathbf{0}$, then the locking configuration $(\boldsymbol{\theta}_a^*, \boldsymbol{\theta}_p^*)$ is classified as stable, because it means that locking the actuated joints also locks the passive ones, suppressing uncontrollable free-swinging motions not only at finite level but also at differential level. Otherwise, uncontrollable passive velocities are admissible and the locking configuration $(\boldsymbol{\theta}_a^*, \boldsymbol{\theta}_p^*)$ is classified as unstable.

5.3. Example 1 (conclusion)

Next, we will conclude Example 1 started in Section 4 by classifying the five locking configurations shown in Figure 8 into stable and unstable, using the two criteria explained above. In order to apply the static criterion, assume that the RPRRR robot has a gripper mounted on link BC as illustrated in Figure 7a, and the output coordinates are the position coordinates (x, y) of this gripper. The particularization of Equation (9) for this robot is:

$$\xi = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} l_1 + l_2\cos(\theta_2) + l_4\cos(\theta_2+\theta_3) + l_5\cos(\theta_2+\theta_3-\pi/2) \\ l_2\sin(\theta_2) + l_4\sin(\theta_2+\theta_3) + l_5\sin(\theta_2+\theta_3-\pi/2) \end{bmatrix} = \mathbf{H}\left(\boldsymbol{\theta}_a, \boldsymbol{\theta}_p\right) \tag{23}$$

where $l_4 = l_5 = 0.01$ m. In this example, $\boldsymbol{\tau}_{\theta_a} = \tau_\rho$ is the input force exerted by the linear actuation OB, whereas $\boldsymbol{\tau}_\xi = [\tau_x, \tau_y]^T$ are external forces applied on the end-effector along the x and y directions.

Next, the two stability criteria presented will be illustrated with the five locking configurations shown in Figure 8:

1. For the locking configuration ρ_1^* of Figure 8b, we have: $\mathbf{G}(\boldsymbol{\theta}_a^*, \boldsymbol{\theta}_p^*) = [5.531, 0, 2.1468, 2.1832]^T$. According to (15), this means: $\gamma_1 = \gamma_3 = \gamma_4 = 0$. Accordingly, the right-hand side of Equation (20) becomes:

$$\begin{bmatrix} 0.3514183404 & -0.3514183404 & 50.08944255 \\ -0.2769271862 & 0.2769271862 & -39.54229645 \end{bmatrix} \begin{bmatrix} \tau_\rho^+ \\ \tau_\rho^- \\ \gamma_2 \end{bmatrix}, \quad \text{with: } \tau_\rho^+ \geq 0, \tau_\rho^- \geq 0, \gamma_2 \geq 0 \tag{24}$$

   The columns of this matrix are represented in Figure 9c as points v_i in $\mathbb{R}^2$, indicating in parentheses the generalized force that each column multiplies according to (24). Actually, to ease visualization, due to the great difference of magnitudes between the third and remaining columns in (24), the points v_i represented in Figure 9c are "normalized" so that they all have a unit distance from the origin. This normalization, which will be used in Figures 9c-f, does not affect the results of the stability criterion. As Figure 9c shows, the polytope $\mathcal{P}$ obtained as the convex hull of these points (plus the origin $\mathcal{O}$) is a line segment, since all points are collinear. Therefore, the mapping of (24) is not full-rank and it cannot balance arbitrary external forces applied on the end-effector, or arbitrary gravitational torques. Thus, ρ_1^* is not a stable locking configuration according to the static criterion. If we apply the velocity criterion to this configuration, Equation (22) particularizes to $\dot{\theta}_2 \leq 0$, because $\theta_2 = \theta_2^{max}$. This inequality is satisfied by a null vector of $\mathbf{F}_{\theta_p}$ such as: $\dot{\boldsymbol{\theta}}_p^n = [\dot{\theta}_2, \dot{\theta}_3, \dot{\phi}]^T = [0, -0.52, 1]^T$. Thus, this locking configuration is also unstable according to the velocity criterion.
2. For the locking configuration ρ_2^* of Figure 8c, we have $\mathbf{G}(\boldsymbol{\theta}_a^*, \boldsymbol{\theta}_p^*) = [0, 5.531, 0.817261, 3.512739]^T$, which implies: $\gamma_2 = \gamma_3 = \gamma_4 = 0$ (15). The corresponding (normalized) columns of the mapping $\tilde{\mathbf{N}}$ are represented in Figure 9d, together with the origin $\mathcal{O}$ and their convex hull $\mathcal{P}$. Again, $\mathcal{P}$ is a line segment, whose scaling cannot span $\mathbb{R}^2$. Thus, this locking configuration is not stable according to the static criterion. Regarding the velocity criterion, we have $\theta_2 = \theta_2^{min}$, which implies: $\dot{\theta}_2 \geq 0$. This inequality is satisfied by a null vector of $\mathbf{F}_{\theta_p}$ such as: $\dot{\boldsymbol{\theta}}_p^n = [0, 1, -1]^T$. Thus, this locking configuration is also unstable according to the velocity criterion.
3. For the locking configuration ρ_3^* of Figure 8d, we have: $\mathbf{G}(\boldsymbol{\theta}_a^*, \boldsymbol{\theta}_p^*) = [5.531, 0, 4.33, 0]^T$, which implies: $\gamma_1 = \gamma_3 = 0$. Figure 9e shows the (normalized) columns of $\tilde{\mathbf{N}}$ and the polytope $\mathcal{P}$. As $\mathcal{O}$ belongs to the interior of $\mathcal{P}$, it is possible to scale $\mathcal{P}$ with $\mathcal{O}$ kept fixed so that it spans the whole plane $\mathbb{R}^2$. Thus, this locking configuration is stable according to the static criterion. Regarding the velocity criterion, we have $\theta_2 = \theta_2^{max}$ and $\theta_3 = \theta_3^{max}$, which implies: $\dot{\theta}_2 \leq 0$ and $\dot{\theta}_3 \leq 0$. The nullspace vectors of $\mathbf{F}_{\theta_p}$ for this configuration are proportional to $\dot{\boldsymbol{\theta}}_p^n = [-0.5338, 0.7308, 0.4255]^T$, which have different sign in $\dot{\theta}_2$ and $\dot{\theta}_3$, so they cannot satisfy these inequalities. Thus, the only feasible solution is $\dot{\boldsymbol{\theta}}_p^n = \mathbf{0}$, and this configuration is also stable according to the velocity criterion.
4. For the locking configuration ρ_4^* of Figure 8e, we have: $\mathbf{G}(\boldsymbol{\theta}_a^*, \boldsymbol{\theta}_p^*) = [0, 5.531, 4.33, 0]^T$, which implies: $\gamma_2 = \gamma_3 = 0$. Figure 9f shows the (normalized) columns of $\tilde{\mathbf{N}}$ and the polytope $\mathcal{P}$. As in the previous case, this configuration is stable according to the static criterion, because $\mathcal{O}$ belongs to the interior of $\mathcal{P}$. Regarding the velocity criterion, since $\theta_2 = \theta_2^{min}$ and $\theta_3 = \theta_3^{max}$, then it must be: $\dot{\theta}_2 \geq 0$ and $\dot{\theta}_3 \leq 0$. The nullspace vectors of $\mathbf{F}_{\theta_p}$ are proportional to $\dot{\boldsymbol{\theta}}_p^n = [0.2282, 0.9222, 0.3121]^T$, which implies that $\dot{\theta}_2$ and $\dot{\theta}_3$ have the same sign. Thus, the only solution is $\dot{\boldsymbol{\theta}}_p^n = \mathbf{0}$, so this locking configuration is also stable according to the velocity criterion.
5. For the final locking configuration ρ_5^* of Figure 8f, we have: $\mathbf{G}(\boldsymbol{\theta}_a^*, \boldsymbol{\theta}_p^*) = [2.508, 3.023, 1.911, 2.419]^T$ which means that $\gamma_1 = \gamma_2 = \gamma_3 = \gamma_4 = 0$. Accordingly, the right-hand side of Equation (20) becomes: $\mathbf{0}_{2\times 2}\left[\tau_\rho^+, \tau_\rho^-\right]^T$. This means that no external forces or torques can be balanced at all, so this locking configuration is unstable according to the static criterion. This is also true for the velocity criterion, as the null space of $\mathbf{F}_{\theta_p}$ for this configuration is parameterized by $\dot{\boldsymbol{\theta}}_p^n = \alpha[-0.4, 1, 0]^T + \beta[2.2, 0, 1]^T$, with $\alpha, \beta \in \mathbb{R}$. Since $\dot{\boldsymbol{\theta}}_p^n$ is not constrained because no inequalities (7) are active, α and β can take any value, so $\dot{\boldsymbol{\theta}}_p^n$ admits nonzero values.

Summing up, for this example there are only two stable locking configurations (ρ_3^* and ρ_4^*), to which the actuated joint ρ can be taken after the robot has become underactuated due to suffering FSJF, in order to completely suppress uncontrollable free-swinging motions at finite and velocity levels, resisting any external force applied to the gripper.

The video ex1_control.mp4 attached to this paper shows a simulation of the control of this underactuated robot, in order to take it to each of the five locking configurations ρ_i^* of Figure 8. In this simulation, only ρ is controlled through a Proportional-Integral-Derivative (PID) controller, leaving ϕ unactuated. This PID controller tries to drive ρ to different setpoints manually specified by the user that commands the simulation. The user chooses these setpoints through trial-and-error, by visually monitoring the passive joints (θ_2, θ_3) and taking advantage of inertia and gravity to drive these passive joints to the free-swinging manifolds that degenerate into points, reaching each of the five locking configurations ρ_i^*. To avoid having to command manually the actuated joints through trial-and-error, it would be more efficient to plan a trajectory to the desired locking configuration, and use a model-based controller that exploits dynamic coupling between actuated and passive joints to follow this trajectory. However, this will be part of future work, as explained in Section 9.

6. Example 2: PPRRRR manipulator

In the example of the RPRRR manipulator studied in Sections 4 and 5.3, the number of joints that remain actuated after the occurrence of FSJF is $r = 1$. Accordingly, the space of actuated joint displacements is 1-dimensional, and the locking configurations $\boldsymbol{\theta}_a^*$ at which some free-swinging manifolds shrink into points form a finite set of points $(\rho_1^*, \ldots, \rho_5^*)$ scattered along the ρ axis. As these locking configurations constitute a finite and small set, each of these configurations can be studied manually in order to determine its stability using the static and velocity criteria, as illustrated in Section 5.3. However, when the number of joints that remain actuated after FSJF is $r \geq 2$, the number of locking configurations $\boldsymbol{\theta}_a^*$ is not finite: in general, we may have positive-dimensional sets consisting of infinitely many different locking configurations. For example, for $r = 2$ actuated joints θ_{a1} and θ_{a2}, the locking configurations $\boldsymbol{\theta}_a^*$ will generically lie along curves in plane $(\theta_{a1}, \theta_{a2})$. In that case, it will be necessary to check every point of these curves in order to determine their stability. This will be illustrated in the present section with another example.

The example studied in this section is the underactuated manipulator shown in Figure 2, where $n = 3$, $\boldsymbol{\theta}_a = [x, y]^T$, $\boldsymbol{\theta}_p = [\theta_1, \theta_2, \theta_3]^T$, $l_1 = \overline{OA} = 1$, $l_2 = \overline{AB} = 0.7$, and $l_3 = \overline{BC} = 0.6$. The loop-closure equation that relates the actuated and passive joint displacements for this robot was given in (2). The inequality constraints that will be imposed to this example are as follows:

$$\mathbf{G}(\boldsymbol{\theta}_a, \boldsymbol{\theta}_p) = \begin{bmatrix} \theta_1 + 2.3562 \\ -\theta_1 + 1.7453 \\ \theta_2 + 2.0944 \\ -\theta_2 + 1.6581 \\ \theta_3 + 1.7453 \\ -\theta_3 + 1.2217 \\ -0.2\left(l_1 \cos\theta_1\right) + 1.3\left(l_1 \sin\theta_1\right) + 1 \\ -0.2\left(l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2)\right) + 1.3\left(l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2)\right) + 1 \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{25}$$

The first six entries of $\mathbf{G}$ in (25) are joint limits, in radians. The last two entries of $\mathbf{G}$ model a collision constraint, so that link AB (modeled as a segment with zero thickness) does not penetrate into the semi-plane defined by $-0.2x + 1.3y + 1 < 0$, which models a wall obstacle. In order for link AB not to penetrate into this semi-plane, it is sufficient to impose the condition that both of its endpoints A and B satisfy $-0.2x + 1.3y + 1 \geq 0$: this is imposed by the last two entries of $\mathbf{G}$. Note that it is very simple to *globally* model this collision constraint using inequalities because the geometry of the colliding bodies is very simple: a segment that must not intersect a semi-plane. In the more general case where the geometry of the links or the obstacle is more complex, this collision constraint cannot be *globally* modeled so easily as an inequality, and *local* analyses are needed instead. This will be illustrated in Section 7.

In order to study the stability of locking configurations using the static criterion described in Section 5.1, it will be assumed that the output position/orientation coordinates of the gripper are $\boldsymbol{\xi} = [\xi_1, \xi_2, \xi_3]^T$ as shown in Figure 2, and they are functions of the actuated and passive joint displacements as given by the following equation:

$$\boldsymbol{\xi} = \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{bmatrix} = \begin{bmatrix} x - h\cos(\theta_1 + \theta_2 + \theta_3) + v\cos(\theta_1 + \theta_2 + \theta_3 - \pi/2) \\ y - h\sin(\theta_1 + \theta_2 + \theta_3) + v\sin(\theta_1 + \theta_2 + \theta_3 - \pi/2) \\ \theta_1 + \theta_2 + \theta_3 - \pi/2 \end{bmatrix} = \mathbf{H}(\boldsymbol{\theta}_a, \boldsymbol{\theta}_p) \tag{26}$$

where h and v define the position of the gripper in link BC relative to joint C, as shown in Figure 2.

Next, the method described in Section 3.1 will be used for obtaining the locking configurations, i.e., the values of the actuated prismatic joints (x^*, y^*) for which some free-swinging manifolds degenerate into points. The method is run with the following configuration parameters:

- Both actuated joint displacements (x, y) are swept in the square region $[-2.3, 2.3] \times [-2.3, 2.3]$ (where $2.3 = l_1 + l_2 + l_3$ is the maximum reach of x or y) with a resolution of 150 nodes per joint. For each node (x, y), the solution set of (2) is a set of curves in the space of passive joints $(\theta_1, \theta_2, \theta_3)$. These curves are mapped using the method of Section 3.1.1, by sweeping each passive joint on a grid of 250 nodes between its limits. When each passive joint is swept, the other two are solved from (2), which can be reduced to a quadratic equation.

- The sizes w_i of the box used for the clustering operation of Section 3.1.2 are chosen as four times the step with which each passive joint is swept during the mapping operation. The sizes w'_i used for the matching operation of Section 3.1.3 are chosen as explained immediately after Equation (8).

Running the method under these conditions requires 77 seconds to compute the locking configurations shown in Figure 10a, which are curves in the plane (x, y), such that taking the actuated joints to these curves produces the shrinking of some free-swinging manifolds into points. Both stability criteria are applied to each point of these curves, and the points classified as stable are highlighted in Figure 10b. Both criteria classify as stable the same set of points.

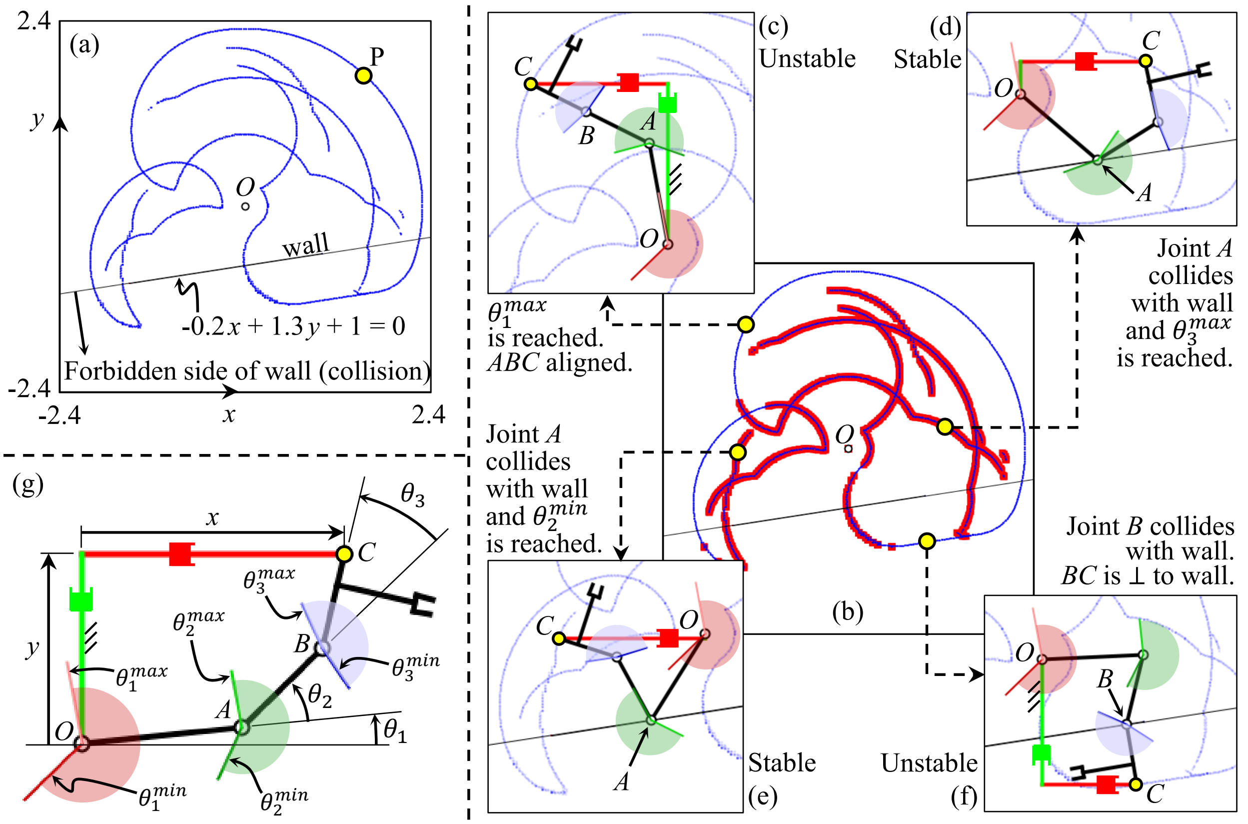

Figure 10: (a): Curves of locking configurations (in blue) in the (x, y) plane of the underactuated manipulator of Figure 2. The line $-0.2x + 1.3y + 1 = 0$ is the boundary of the forbidden region where link AB is not permitted due to collisions. (b): Stable locking configurations highlighted in red. (c and f): Two examples of unstable configurations. (d and e): Two examples of stable configurations. (g): Representation of joint limits as circular sectors.

Figures 10c, 10d, 10e, and 10f illustrate the configuration adopted by the robot when the actuated joints (x, y) place joint C at different stable and unstable configurations taken as examples, for which different inequalities of (25) are active as indicated in these figures. In order to understand the representation of the robot in Figures 10c-f, Figure 10g shows that the actuated joints x and y are represented simplified as red and green bars respectively, and the joint limits for θ_i are represented as circular sectors, similarly to the example of Figure 8.

Regarding the operation to classify locking configurations into stable or unstable, its execution time (0.11 seconds) is negligible with respect to the time necessary to obtain these locking configurations (77 seconds). The application of both stability criteria for each locking configuration is automated as follows. First, note that both criteria require identifying which components G^i (≥ 0) of $\mathbf{G}$ are zero, but due to small numeric errors they will never be exactly zero. Thus, in this example we consider that G^i is zero if $G^i < 0.01$. After this, we proceed as follows:

- For the static criterion: we determine the polytope $\mathcal{P}$ defined in Section 5.1, modeled as a set of faces defined by three or more coplanar points. For each face of this polytope, defined by at least three coplanar vertices v_i, v_j and v_k, we check the coplanarity between these three vertices and the origin $\mathcal{O}$, by computing the determinant of $[v_i - v_j, v_i - v_k, v_i]$. If the absolute value of this determinant is below 0.001 for at least one face of the polytope, then we consider that the origin belongs to the boundary of $\mathcal{P}$ and the locking configuration is not stable. Otherwise, it is stable.
- For the velocity criterion: for this criterion, first it is necessary to find the dimension of the null space of $\mathbf{F}_{\theta_p}$.
  - If the dimension is 1, we proceed as in the example of Section 5.3. In this case, there is only one unit null vector $\dot{\boldsymbol{\theta}}_p^n$ (or $-\dot{\boldsymbol{\theta}}_p^n$). If $\dot{\boldsymbol{\theta}}_p^n$ satisfies (22), then the configuration is unstable, because there exist admissible passive velocities that are compatible with the unidirectional constraints (22). If not, then it is necessary to check $-\dot{\boldsymbol{\theta}}_p^n$. If $-\dot{\boldsymbol{\theta}}_p^n$ satisfies (22), then the configuration is unstable. If not, then the configuration is stable, as no admissible passive joint velocities exist that are compatible with these unidirectional constraints.
  - If the dimension is 2, we can proceed as we will explain for the example of Section 8. This happens, for example, when the actuated joints (x, y) are moved to points like P of Figure 10a, where the kinematic chain $OABC$ is fully aligned. In that case, the FSM is an increasingly smaller loop that shrinks into a point as in Figure 4i, and the dimension of the null space increases to two.

Finally, the supplementary video ex2.mp4 attached to this paper shows how the free-swinging manifolds in the space of passive joint displacements $(\theta_1, \theta_2, \theta_3)$ change when varying the actuated joints, and in particular, the video shows how two of these manifolds degenerate into points when moving the actuated joints to locking configurations, suppressing free-swinging motions.

7. Example 3: an underactuated serial manipulator

So far, the method proposed in this paper has been illustrated for manipulators with closed kinematic chains. This method is best suited for closed-chain manipulators and parallel robots, in which the motion of some joints or links can “interfere” or “obstruct” the motion of other joints or links, to the point of locking the mechanism, even if neglecting collisions. For example, consider the five-bar mechanism of Figure 3a: if A_1 and A_2 are actuated joints, the robot is fully actuated. If joint A_2 suffers FSJF, and only A_1 remains actuated, the robot becomes underactuated, and, for a fixed generic value of the actuated joint coordinate θ_1, the robot can freely and uncontrollably move along a one-dimensional free swinging manifold in the space of the passive joints (θ_2, ψ_1, ψ_2). It is possible to suppress uncontrollable motions along this manifold (at least at finite level) by rotating link A_1-B_1 counterclockwise until the chain B_1-C-B_2-A_2 becomes fully aligned. After this, θ_1 cannot rotate counterclockwise because the chain B_1-C-B_2-A_2 cannot stretch more. This produces the shrinking of the FSM of this manipulator into a point that does not allow for finite uncontrollable motions. As argued in Section 5, if we can additionally ensure that the robot does not admit passive joint velocities when the manifold has degenerated into a point, then the locking is stable.

This is generally not possible for serial manipulators, where the motion of different joints is independent and one actuated joint cannot be obstructed by other joints, at least when collisions are neglected. As a result, suppressing the uncontrollable free-swinging motions of an underactuated serial manipulator using only the healthy actuated joints is more difficult. However, next we will show that, with the help of appropriate collision constraints, it is possible to use the proposed method to stably lock a serial robot that has suffered FSJF.

Note that, from a different perspective, Pham et al. [32] already noted that parallel robots are more robust to FSJF than serial robots, by using Screw Theory to analyze the motion space of the passive joints after the healthy actuated joints are locked. This higher robustness to free-swing joint failure may be regarded as another advantage of parallel robots over their serial counterparts, in addition to their higher payload-to-weight ratio and stiffness.

Now, consider the serial manipulator shown in Figure 1a. Initially, this is a fully-actuated manipulator with actuated joints $(\theta_1, \theta_2, \theta_3)$ and output pose coordinates $\boldsymbol{\xi}$, related as follows:

$$\boldsymbol{\xi} = \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{bmatrix} = \begin{bmatrix} l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3) \\ l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3) \\ \theta_1 + \theta_2 + \theta_3 \end{bmatrix} = \mathbf{H}(\boldsymbol{\theta}_a, \boldsymbol{\theta}_p) \tag{27}$$

where (ξ_1, ξ_2) are the (x, y) coordinates of the gripper C, whereas ξ_3 is the angle ϕ between link BC and the X axis. Assume that θ_2 suffers FSJF, therefore: $\boldsymbol{\theta}_a = [\theta_1, \theta_3]^T$ and $\boldsymbol{\theta}_p = [\theta_2]$. Since this is a serial manipulator, there are no loop-closure constraints relating the actuated and passive joint displacements, i.e., the constraint function $\mathbf{F}$ defined in (1) does not exist for this robot, and for the statics model (10) we simply have: $\mathbf{L} = \mathbf{H}(\boldsymbol{\theta}_a, \boldsymbol{\theta}_p) - \boldsymbol{\xi}$.

Assume first that only the failed joint θ_2 has joint limits: $\theta_2^{min} \leq \theta_2 \leq \theta_2^{max}$, where $\theta_2^{min} = -2.0944$ rad and $\theta_2^{max} = 1.6581$ rad, as in Section 6. The lengths of the links are again: $l_1 = 1$, $l_2 = 0.7$, $l_3 = 0.6$. Note that, as there are no loop-closure constraints between the actuated and passive joints of this robot, the free-swinging manifold for fixed values of the actuated joints (θ_1,θ_3) will always be the line segment $\{\theta_2 \in [\theta_2^{min}, \theta_2^{max}]\}$ regardless of the values of θ_1 and θ_3, and this will not change if joint limits are imposed to θ_1 and θ_3. This means that this underactuated robot cannot be locked and stabilized by moving the actuated joints (θ_1,θ_3) to appropriate positions where this line segment manifold degenerates into a point, unlike in the closed-chain manipulators studied in the previous examples.

However, this situation changes if collision constraints are considered. Assume that we have a static rectangular obstacle centered at $(1.6, -0.6)$, with an orientation of 2 rad, width 0.7 and height 0.4, as shown in Figure 11a. This obstacle should not collide with link l_3, which is modeled also as a rectangle with length $l_3 + 2e$ and width a as shown in Figure 11a, with $e = 0.1$ and $a = 0.2$. We will show next that, if this collision constraint is imposed, then it is possible to find values of (θ_1, θ_3) where some FSM degenerate into points and the manipulator is locked stably.

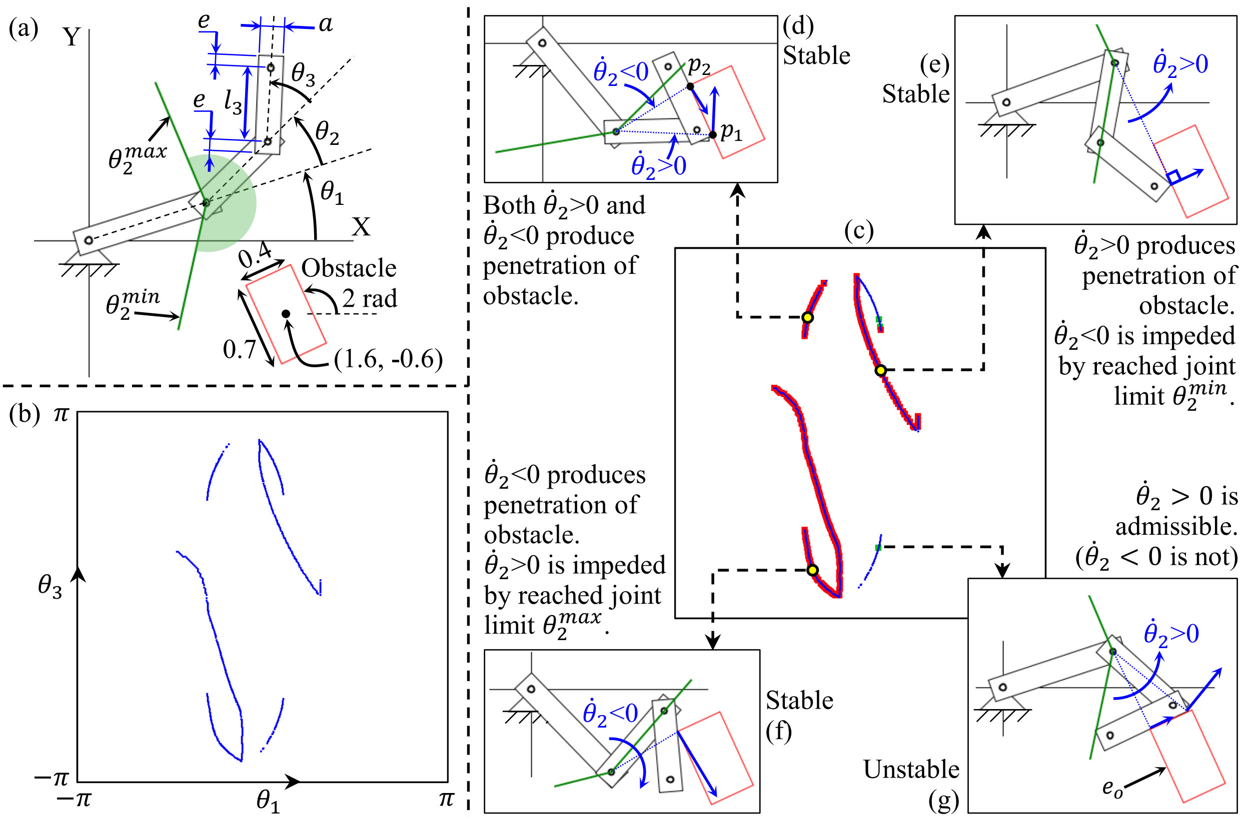

Figure 11: (a): Rectangular shapes of the obstacle and link l_3, which move in the same plane and may collide. (b): Unclassified locking configurations in the actuated plane (θ_1, θ_3). (c): Stable locking configurations highlighted in red. (d,e,f): Examples of stable locking configurations. (g): Example of unstable locking configuration.

The way to include this collision constraint in the computations of the method described in Section 3.1 is as follows. During the mapping stage described in Section 3.1.1, when sampling points from the free-swinging manifold between θ_2^{min} and θ_2^{max}, the sampled points that generate mechanical interferences between the obstacle and link l_3 are discarded. Mechanical interferences between these two rectangular bodies can be detected by using the Separating Axis Theorem [46]. After this, the other stages of the method (clustering §3.1.2 and matching §3.1.3) are applied in order to detect the locking configurations of the (θ_1, θ_3) plane at which some free-swinging manifolds degenerate into points. For this particular example, the proposed method computes the locking configurations shown in Figure 11b. Obtaining these results required a time of 91 seconds when using the following parameters for the method:

- The actuated joints (θ_1, θ_3) are swept in the square region $[-\pi, \pi] \times [-\pi, \pi]$ with a resolution of 150 nodes per joint. For each node (θ_1, θ_3), the interval $\{\theta_2 \in [\theta_2^{min}, \theta_2^{max}]\}$ is discretized into 500 points, discarding the points that produce mechanical interference between link l_3 and the obstacle. The mapping operations described in Section 3.1.1 are thus simplified for this serial robot, as there is no loop-closure equation (1) to solve.
- The size w_i of the box used for the clustering operation of Section 3.1.2 is chosen as 2.4 times the step with which θ_2 is swept in the interval $[\theta_2^{min}, \theta_2^{max}]$ during the mapping operation. Regarding the size w_i' used for the matching operation of Section 3.1.3, in this case it cannot be chosen according to Equation (8) because there is no loop-closure function $\mathbf{F}$ defined for this robot. Therefore, w_i' is chosen equal to w_i for this particular example.

After obtaining the locking configurations, it is necessary to classify them into stable or unstable, applying both criteria introduced in Section 5. Both criteria require the derivatives of the inequality functions G^i that vanish, i.e., inequalities $G^i \geq 0$ that are active. While it was easy to globally model the collision constraint in the example of Section 6, because the colliding bodies were a line segment and a straight line, for bodies with arbitrarily complex shapes it is not generally possible to write down an inequality constraint that is globally valid for all possible relative poses between the colliding bodies. In general, local models that are only valid in the vicinity of the collision must be used. In those cases, the G^i function must be defined as the signed distance between the colliding objects when they are on the verge of colliding [45]. This G^i function, called *gap function*, must be defined as the signed distance between the near-to-colliding bodies along the common normal at the points of contact, as illustrated in Figure 12a. Thus, these points will largely depend on the geometric shape of the bodies, and on their relative pose.

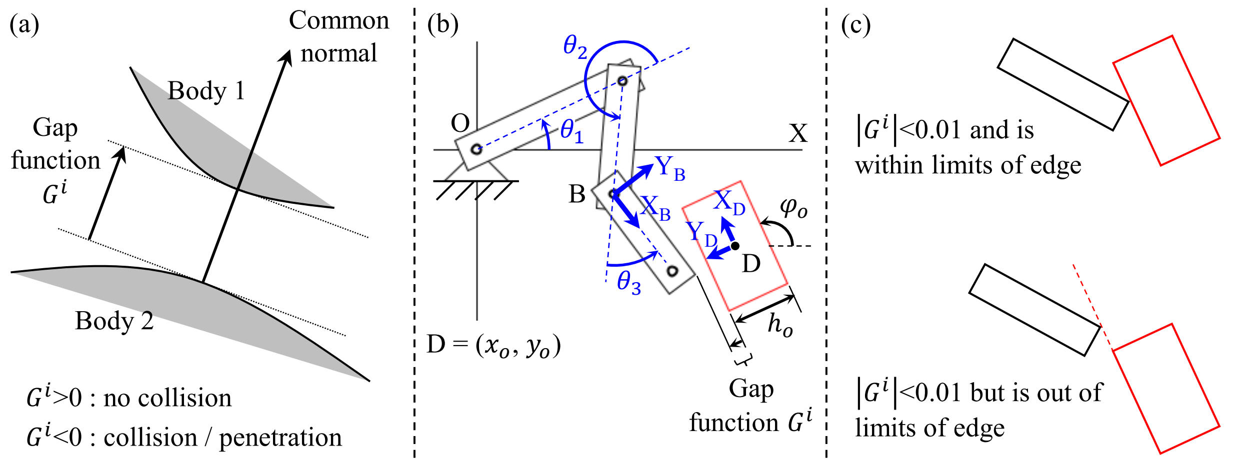

Figure 12: (a) Definition of gap function. (b) Example of gap function when one of the vertices of link l_3 is incident on one of the edges of the obstacle. (c) Difference between a valid true incidence (top), and an invalid virtual incidence out of the limits of the incised edge (bottom).

In the particular case that we are considering in this section, the colliding bodies (link l_3 and the obstacle) are rectangles. For the collision of two rectangles, there are two main cases:

1. A vertex of link l_3 is incident on an edge of the obstacle. As there are four incident vertices and four incised edges, there are 16 possibilities for this case.
2. A vertex of the obstacle is incident on one of the edges of link l_3. Again, there are 16 possibilities, corresponding to all possibilities of four vertices incident on four edges.

Accordingly, 32 gap functions must be locally defined in order to model the collision constraint between link l_3 and the obstacle, depending on which vertex is incident on which edge. Each gap function consists of the signed distance from the incised edge to the incident vertex, considering all the possibilities enumerated above. For example, Figure 12b illustrates the computation of the gap function when one of the vertices of link l_3 is incident on one of the edges of the obstacle. This gap function is obtained using planar homogeneous transform matrices, by first obtaining the coordinates of the incident vertex in the local frame BX_BY_B attached to link l_3, then mapping them to global coordinates in frame OXY, then mapping them to local coordinates in the frame DX_DY_D attached to the obstacle, then extracting the y-coordinate in frame DX_DY_D, and finally subtracting half of the height h_o of the obstacle (where D is the center of the obstacle). All of these operations are represented next, from right to left:

$$G^i = -h_o/2 + \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} c_{\varphi_o} & -s_{\varphi_o} & x_o \\ s_{\varphi_o} & c_{\varphi_o} & y_o \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} c_{\theta_1+\theta_2+\theta_3} & -s_{\theta_1+\theta_2+\theta_3} & l_1 c_{\theta_1} + l_2 c_{\theta_1+\theta_2} \\ s_{\theta_1+\theta_2+\theta_3} & c_{\theta_1+\theta_2+\theta_3} & l_1 s_{\theta_1} + l_2 s_{\theta_1+\theta_2} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} l_3 + e \\ a/2 \\ 1 \end{bmatrix} \quad (28)$$

where $c_x = \cos(x)$, $s_x = \sin(x)$, $x_o = 1.6$, $y_o = -0.6$, $\phi_o = 2$, and $h_o = 0.4$. For valid configurations in the vicinity of the posture shown in Figure 12b, the gap function G^i defined in (28) should be positive or zero. Otherwise, a mechanical interference occurs. For the other 31 combinations of edge-vertex incidence, an analogous gap function can be built.

Note that, in addition to these 32 gap functions due to collisions, there are also two inequalities corresponding to the joint limits of θ_2, which are the same as the third and fourth inequalities of (25). This makes 34 inequalities for this robot, but these will not be shown for briefness.

Having said that, the velocity and statics criteria of Section 5 are applied as follows. For each locking configuration (θ_1^*, θ_3^*) obtained (with its corresponding value of θ_2^*), all 34 G^i-functions are evaluated to check which of the corresponding inequalities are active. It is considered that an inequality is active if $|G^i| < 0.01$. Additionally, for a vertex-edge incidence to be considered active, it is also necessary to check if the incidence truly occurs within the limits of the incised edge, or virtually at its prolongation, see Figure 12c. For those G^i-functions that are deemed active, their partial derivatives are computed, and these are used in the static and velocity criteria presented in Section 5, to classify (θ_1^*, θ_3^*) into stable or unstable according to these criteria.

For the present example, the statics criterion is applied as in Section 6, i.e., checking that the determinant of coplanarity between the origin $\mathcal{O}$ and three vertices that define each face of the polytope $\mathcal{P}$ is below 0.001 in absolute value. Regarding the velocity criterion, since the serial robot studied in this section has no loop-closure function $\mathbf{F}$, Equation (21) is not used, whereas Equation (22) is simplified to: $G^i_{\theta_2}(\theta_a^*, \theta_p^*) \cdot \dot{\theta}_2 \geq 0 \quad \forall G^i$ such that $G^i(\theta_a^*, \theta_p^*) = 0$.

After applying this to classify the locking configurations of Figure 11b, the configurations that are classified as stable by both criteria are those highlighted in red in Figure 11c. In this case, there are a few locking configurations that are erroneously classified as stable only by the statics criterion; these are highlighted in green in Figure 11c. If the threshold of the determinant of coplanarity of the statics criterion is increased to 0.002 (instead of 0.001, as mentioned in the previous paragraph), this criterion classifies these configurations as unstable, like the velocity criterion.

Figure 11d shows an example of stable locking configuration, for which link l_3 has an edge p_1-p_2 in contact with an edge of the obstacle. If θ_1 and θ_3 are locked and the velocity of the passive joint is positive ($\dot{\theta}_2 > 0$), the velocity of some points of the contacted edge of link l_3, like point p_1, is such that they penetrate the obstacle. On the contrary, if $\dot{\theta}_2 < 0$, points like p_2 penetrate the obstacle. Thus, according to this graphical analysis, the only admissible passive velocity is $\dot{\theta}_2 = 0$ and this locking configuration is stable, which coincides by both criteria presented in Section 5.

The stability of the example configurations of Figures 11e-g can be graphically studied in a similar way, by locking θ_1 and θ_3 while trying to increase or decrease θ_2. In Figure 11e, clockwise rotations of θ_2 ($\dot{\theta}_2 < 0$) are impeded because $\theta_2 = \theta_2^{min}$, whereas counterclockwise rotations ($\dot{\theta}_2 > 0$) would require the vertex of link l_3 that contacts the obstacle to penetrate it. Similarly, in Figure 11f, counterclockwise rotations of θ_2 are impossible because $\theta_2 = \theta_2^{max}$, whereas clockwise rotations would require the edge of l_3 contacted by the obstacle to penetrate it. Finally, in Figure 11g, which represents one of the aforementioned locking configuration classified erroneously as stable by the statics criterion, counterclockwise rotations of θ_2 are possible because the velocities of all contact points of l_3 point away from the obstacle, whereas these velocities would have components directed to the interior of the obstacle if $\dot{\theta}_2 < 0$, which forbids clockwise rotations. Since the passive joint velocity can be nonzero, this configuration is indeed unstable.

Additionally, a supplementary video ex3.mp4 is attached to this paper, which shows how the free-swinging manifolds in the space of the passive joint coordinate θ_2 change when varying the actuated joints (θ_1, θ_3). This video shows how two of these manifolds degenerate into points when moving the actuated joints to locking configurations, suppressing free-swinging motions.

8. Example 4: more than one joint suffers FSJF

So far, we have assumed that only one actuated joint fails and becomes free. As a result of this, the free-swinging manifolds are one-dimensional, i.e., curves in the space of the passive joint displacements. A natural question to raise is what happens if two or more joints fail simultaneously. In that case, the dimension of the free-swinging manifolds increases, but the basic idea proposed in this paper still remains valid in those cases.

For example, assume that, for the closed-chain manipulator of Figure 2, both θ_1 and y fail simultaneously and become passive, and only x remains actuated. In that case, the dimension of the free-swinging manifolds increases to two, i.e., for a fixed x, the passive joints $(\theta_1, \theta_2, \theta_3, y)$ can freely slide along a surface (or set of disjoint surfaces) in the four-dimensional space of passive joint displacements. These surfaces will change in shape and size when varying the only joint that remains actuated (x). As in the case of one-dimensional free-swinging manifolds, it may occur that, for some values of x, some of these surfaces degenerate into points at which passive velocities are inadmissible, stably locking the underactuated manipulator against external wrenches. By applying the proposed method, considering the same constraints as in Section 6, but now regarding y also as a passive joint displacement, seven locking configurations shown in Figure 13 are obtained in 60 seconds, where Figure 13 indicates which configurations are classified as stable (both stability criteria coincide in their classification for each locking configuration). Also, a video (ex4.mp4) is attached, which shows the evolution of the two-dimensional manifolds projected on the subspace of $(\theta_1,\theta_2,\theta_3)$ when sweeping x from -2.4 to 2.4. In this example, since y does not appear in the inequalities (25), it is possible to omit it and simply analyze the free-swinging manifolds projected on the subspace of $(\theta_1,\theta_2,\theta_3)$, because y is uniquely defined by $(\theta_1,\theta_2,\theta_3)$ in the second row of Equation (2). In fact, since we omit y, the loop-closure constraint (1) of this manipulator is simply the first row of $\mathbf{F}$ defined in (2), which relates the actuated joint x to the passive joints $(\theta_1,\theta_2,\theta_3)$.

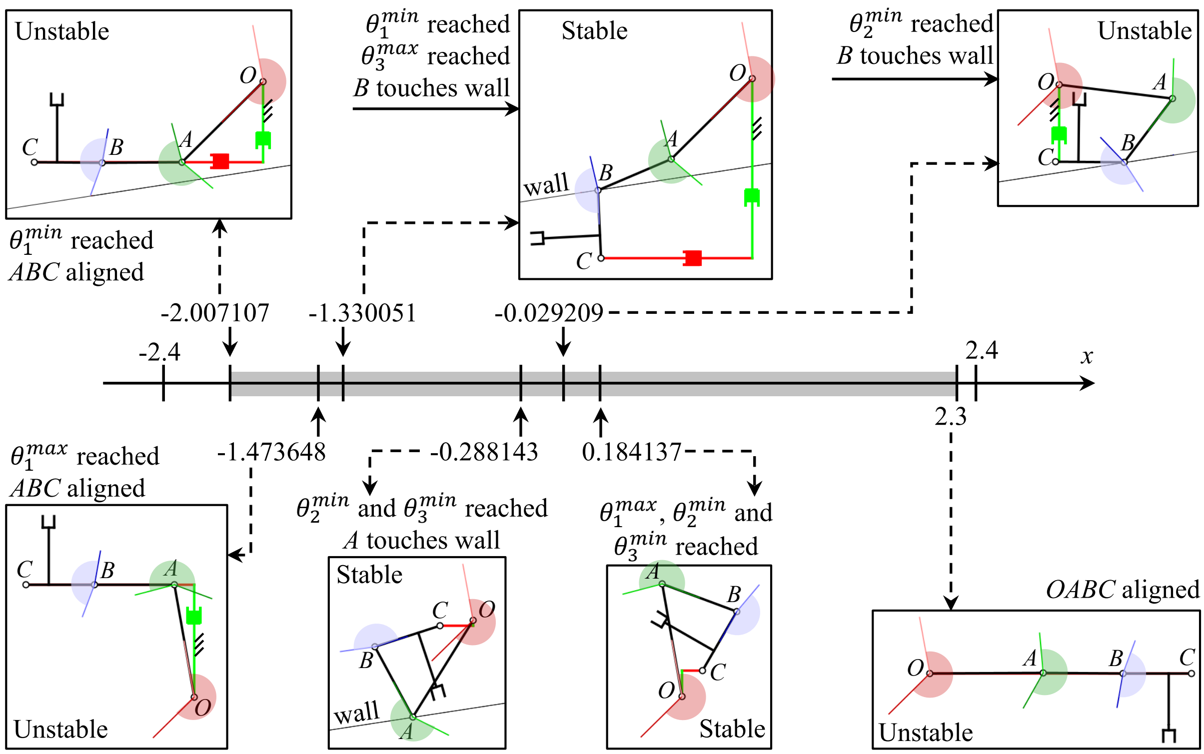

Figure 13: Locking configurations for which free-swinging manifolds degenerate into points, for the manipulator of Figure 2 with only x actuated. The prismatic joints (x, y) are not clearly visible in some configurations, but recall that (x, y) are the Cartesian coordinates of C with respect to O, with x actuated and y passive. Link AB cannot penetrate the wall.

The seven locking configurations shown in Figure 13 are obtained by running the proposed method with the following configuration parameters, but some modifications with respect to the method described in Section 3.1 are required, because the free-swinging manifolds are now two-dimensional:

- The actuated joint x is swept from -2.3 to 2.3 with a resolution of 100 equally-spaced values. For each value of x, the first row of Equation (2) generically defines a set of disjoint surfaces in the space of $(\theta_1,\theta_2,\theta_3)$; these surfaces are the free-swinging manifolds in this example.
  - In order to map these manifolds using the method described in Section 3.1.1, a modification is needed. The method described in Section 3.1.1 assumed that the FSM are one-dimensional, so that they can be mapped by sweeping one of the coordinates of the ambient space of the manifolds while solving the remaining coordinates from (1) (and this sweeping must be repeated for each ambient coordinate, to ensure that manifolds are densely sampled everywhere [34]). Now that the FSM are two-dimensional, in order to map these manifolds for a given value of x, we must simultaneously sweep two of the coordinates of the ambient space $(\theta_1,\theta_2,\theta_3)$ of the FSM, while solving the remaining one from the first row of Equation (2). To be more concrete, first we sweep (θ_1, θ_2) in the box $[\theta_1^{min}, \theta_1^{max}]\times[\theta_2^{min}, \theta_2^{max}]$ with a resolution of 75 nodes per joint (i.e., 75^2 nodes), and for each node we solve θ_3 from the first row of (2), which can be transformed into a quadratic equation. Then, we repeat this, but sweeping joints (θ_1, θ_3) while solving θ_2. Finally, (θ_2, θ_3) are swept while θ_1 is solved. The three sweepings combined yield a point cloud that approximates the free-swining manifold surfaces accurately and densely everywhere, as shown in the attached video ex4.mp4.
- The sizes w_i of the box used for the clustering operation of Section 3.1.2 are chosen as four times the step with which each passive joint is swept during the mapping operation. The sizes w_i' used for the matching operation of Section 3.1.3 are chosen as explained immediately after Equation (8).
- For the present example, the statics stability criterion is implemented as in Sections 6 and 7, i.e., checking that the determinant of coplanarity between the origin $\mathcal{O}$ and three vertices that define each face of the polytope $\mathcal{P}$ is below 0.001 in absolute value.
- For each locking configuration, the velocity stability criterion is implemented as explained next. First, using the singular value decomposition, a basis $\mathbf{B}$ of the null space of $\mathbf{F}_{\theta_p}$ is obtained, and the nullspace passive velocities that satisfy (21) are written as: $\dot{\boldsymbol{\theta}}_p^n = \mathbf{Bc}$, where $\mathbf{c} \in \mathbb{R}^{b\times 1}$ is a vector of free parameters and b is the dimension of the null space. Substituting $\dot{\boldsymbol{\theta}}_p^n = \mathbf{Bc}$ into (22) yields a set of inequalities in $\mathbf{c}$ of the form: $\mathbf{b}^i\,\mathbf{c} \geq 0$. Thus, the problem boils down to finding if there exists some $\mathbf{c} \neq \mathbf{0}$ that satisfies these inequalities (in which case the locking configuration is unstable), or, on the contrary, the only solution is $\mathbf{c} = \mathbf{0}$ (stable). To find out if a nonzero $\mathbf{c}$ exists that satisfies all inequalities, each inequality i is written as an equivalent equation $\mathbf{b}^i\,\mathbf{c} = t_i^2$, introducing a new auxiliary free unknown t_i, and the resulting under-determined system is solved using Newton-Raphson method with the minimum-norm pseudoinverse of the Jacobian, starting from random values of $\mathbf{c}$ and t_i. If the Newton-Raphson iterations always converge to $\mathbf{c} = \mathbf{0}$ after 100 attempts starting from different random $\mathbf{c}$ and t_i, the studied locking configuration is classified as stable.

Having concluded this example with two-dimensional manifolds, one may ask what happens if the dimensions of the problem scale. There are two main dimensions that must be analyzed:

- Dimension r of the space of healthy actuated joints. For all examples shown in this paper, we have assumed the situations shown either in Figure 5a or Figure 5b, i.e., the number r of joints that remain actuated after suffering FSJF is one or two. In those cases, finding the locking configurations requires sweeping or scanning the space of actuated joints in grids of dimension one or two, as illustrated in Figure 5a-b. Grids of dimension $r = 1$ or $r = 2$ can be scanned quickly, but when $r > 2$, this scanning may take too much time. A possible solution to this problem is as follows: if the robot has $r > 2$ healthy actuated joints after suffering FSJF, lock $r - 2$ of these actuated joints and use the remaining two to drive the robot to a stable locking configuration, which can be searched in a two-dimensional grid as in the examples of Figures 10 and 11.
- Dimension of the free-swinging manifolds. The example presented in this section has demonstrated the feasibility of finding locking configurations when the free-swinging manifolds are two-dimensional. If the dimension of these manifolds is higher than two, i.e., if three or more actuated joints fail and become passive, the proposed method would take too much time to apply all the operations described in Section 3.1. For example, mapping three-dimensional manifolds by generalizing the operations described in Section 3.1.1 would be computer-intensive, as it would require to sweep three of the passive joints in a three-dimensional grid whose number

of nodes would be N^3, where N is the number of values swept by each joint, and this should be repeated for every possible combination of three passive joints to guarantee a sufficiently dense sampling of the manifold [34].

9. Conclusions and future work

In this paper, we have studied robots that become underactuated after suffering Free-Swing Joint Failure in some of their actuated joints. It has been shown that, after this failure occurs, locking the joints that remain actuated allows the robot to move freely along free-swinging manifolds, which are analogous to the self-motion manifolds of kinematically-redundant robots. However, there is a fundamental difference: motion along the self-motion manifolds of a redundant robot is achieved through the synchronized and controlled motion of its joints, whereas motion along the free-swinging manifolds introduced in this paper is uncontrolled as it depends on inertial and unknown external forces, friction, etc.

Based on the notion of free-swinging manifolds, we have presented a method to find "locking configurations" at which to lock or secure underactuated robots lacking brakes or redundant actuators. These are the configurations to which the healthy actuated joints of the robot can be taken so that some free-swinging manifolds degenerate into points. If the free-swinging manifold along which the robot moves uncontrollably is reduced to a point, such uncontrollable motion is suppressed, at least at finite or configuration level. The method presented in this paper to find these locking configurations consists in scanning the r-dimensional space of healthy actuated joints, mapping free-swinging manifolds at each point of this space, and comparing such manifolds between neighboring points in order to detect the degeneracy of manifolds into points.

We have also shown that not all locking configurations are stable, in the sense that uncontrollable free-swinging motions may be possible at differential or velocity level even if the free-swinging manifold has degenerated into a point, which means that the robot cannot resist external wrenches that may try to drive it away from the locking configuration. A couple of criteria have been presented: the velocity criterion and the static one. Through examples, both criteria have been shown to coincide in their classification of locking configurations into stable or unstable.

The proposed method has been validated with several examples including serial and parallel underactuated robots, with different types of constraints (joint limits and collisions of different complexity), and different number of failed joints. As shown in the examples, computation of locking configurations is fairly quick, considering that they can be precomputed offline and stored in memory so that the controller can use this information when it is needed. It is important to stress that our method only uses kinematic or geometric information, as it only involves a static analysis at most. No dynamic information or inertial parameters are needed. Static friction has been neglected for simplicity. Nevertheless, incorporating static friction in our method can only benefit the stable locking of the underactuated robot. Note that, as the examples show, our method typically requires the existence of inequality constraints such as joint limits or collisions in order to provide stable lockings. However, such constraints are always present in practical applications.

In the future, we will study the problem of driving the robot to a stable locking configuration after FSJF occurs, assuming that stable locking configurations have been computed in advance using the method proposed in this paper, and they have been stored in the memory of the controller of the robot. First, occurrence of FSJF will have to be detected using methods like [14, 15]. After detecting the occurrence of FSJF, the controller will retrieve from memory the locking configuration that is closest to the current configuration that the robot adopts when FSJF is detected, and a feasible trajectory to the closest locking configuration will be planned in the space of actuated and passive joints. Finally, this trajectory will be followed using controllers that exploit dynamic coupling between actuated and passive joints of underactuated robots, like [20, 22]. This problem will be studied on simulated and real robots.

Funding

This work was supported by the Valencian Conselleria d'Innovació, Universitats, Ciència i Societat Digital [grant number CIGE/2021/177]; and the Spanish Ministry of Science and Innovation [grant number PID2020-116418RB-I00].

References

[1] A. Müller, On the terminology and geometric aspects of redundant parallel manipulators, Robotica 31 (1) (2013) 137–147. doi:10.1017/S0263574712000173.

[2] A. Peidró, J. Gallego, L. Payá, J. M. Marín, Ó. Reinoso, Trajectory analysis for the MASAR: A new modular and single-actuator robot, Robotics 8 (3) (2019) 78. doi:10.3390/robotics8030078.

[3] I. Soto, R. Campa, S. Sánchez-Mazuca, Modeling and control with friction compensation of a pendubot system, Revista Iberoamericana de Automática e Informática Industrial 18 (1) (2021) 39–47. doi:10.4995/riai.2020.13083.

[4] G. Oriolo, Y. Nakamura, Control of mechanical systems with second-order nonholonomic constraints: Underactuated manipulators, in: Proceedings of the 30th IEEE Conference on Decision and Control, Vol. 3, 1991, pp. 2398–2403. doi:10.1109/CDC.1991.261620.

[5] M. Reyhanoglu, A. van der Schaft, N. H. McClamroch, I. Kolmanovsky, Dynamics and control of a class of underactuated mechanical systems, IEEE Transactions on Automatic Control 44 (9) (1999) 1663–1671. doi:10.1109/9.788533.

[6] Y. Yi, J. E. McInroy, Y. Chen, Fault tolerance of parallel manipulators using task space and kinematic redundancy, IEEE Transactions on Robotics 22 (5) (2006) 1017–1021. doi:10.1109/TRO.2006.878973.

[7] Q. Jia, B. Yuan, G. Chen, Y. Fu, Kinematic and dynamic characteristics of the free-floating space manipulator with free-swinging joint failure, International Journal of Aerospace Engineering 2019 (2019) 2679152. doi:10.1155/2019/2679152.

[8] H.-L. Pei, Y. Xu, Control of underactuated robots with unactuated joints, IFAC Proceedings Volumes 32 (2) (1999) 695–700. doi:10.1016/S1474-6670(17)56118-2.

[9] M. Hassan, L. Notash, Analysis of active joint failure in parallel robot manipulators, Journal of Mechanical Design 126 (6) (2004) 959–968. doi:10.1115/1.1798071.

[10] J. D. English, A. A. Maciejewski, Fault tolerance for kinematically redundant manipulators: anticipating free-swinging joint failures, IEEE Transactions on Robotics and Automation 14 (4) (1998) 566–575. doi:10.1109/70.704223.

[11] J. D. English, A. A. Maciejewski, Measuring and reducing the Euclidean-space effects of robotic joint failures, IEEE Transactions on Robotics and Automation 16 (1) (2000) 20–28. doi:10.1109/70.833184.

[12] J. English, A. Maciejewski, Fault tolerance for kinematically redundant manipulators: anticipating free-swinging joint failures, in: Proceedings of IEEE International Conference on Robotics and Automation, Vol. 1, 1996, pp. 460–467 vol.1. doi:10.1109/ROBOT.1996.503819.

[13] J.-H. Shin, C.-Y. Lee, J.-J. Lee, Robust fault-tolerant control framework for robot manipulators with free-swinging joint failures: fault detection, identification and accommodation, in: Proceedings 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vol. 1, 1999, pp. 185–190 vol.1. doi:10.1109/IROS.1999.813002.

[14] W. Dixon, I. Walker, D. Dawson, J. Hartranft, Fault detection for robot manipulators with parametric uncertainty: a prediction-error-based approach, IEEE Transactions on Robotics and Automation 16 (6) (2000) 689–699. doi:10.1109/70.897780.

[15] A. De Luca, R. Mattone, Actuator failure detection and isolation using generalized momenta, in: 2003 IEEE International Conference on Robotics and Automation, Vol. 1, 2003, pp. 634–639 vol.1. doi:10.1109/ROBOT.2003.1241665.

[16] R. Tinós, M. H. Terra, M. Bergerman, A fault tolerance framework for cooperative robotic manipulators, Control Engineering Practice 15 (5) (2007) 615–625. doi:10.1016/j.conengprac.2006.10.018.

[17] P. J. From, J. T. Gravdahl, On the mobility and fault tolerance of closed chain manipulators with passive joints, Modeling, Identification and Control 29 (4) (2008) 151–165. doi:10.4173/mic.2008.4.3.

[18] Y. Ting, S. Tosunoglu, R. Freeman, Torque redistribution and time regulation methods for actuator of fault-tolerant parallel robots, Journal of robotic systems 12 (12) (1995) 807–820. doi:10.1002/rob.4620121205.

[19] F. Caccavale, M. Uchiyama, Cooperative Manipulators, Springer, Berlin, Heidelberg, 2008, pp. 701–718. doi:10.1007/978-3-540-30301-5_30.

[20] H. Arai, S. Tachi, et al., Position control of manipulator with passive joints using dynamic coupling, IEEE Transactions on Robotics and Automation 7 (4) (1991) 528–534. doi:10.1109/70.86082.

[21] M. Bergerman, C. Lee, Y. Xu, A dynamic coupling index for underactuated manipulators, Journal of Robotic Systems 12 (10) (1995) 693–707. doi:10.1002/rob.4620121006.

[22] B. C. Maciel, M. H. Terra, M. Bergerman, Optimal robust control of underactuated manipulators via actuation redundancy, Journal of Robotic Systems 20 (11) (2003) 635–648. doi:10.1002/rob.10113.

[23] A. De Luca, R. Mattone, G. Oriolo, Control of underactuated mechanical systems: application to the planar 2R robot, in: Proceedings of 35th IEEE Conference on Decision and Control, Vol. 2, 1996, pp. 1455–1460. doi:10.1109/CDC.1996.572718.

[24] A. De Luca, R. Mattone, G. Oriolo, Stabilization of underactuated robots: theory and experiments for a planar 2R manipulator, in: Proceedings of International Conference on Robotics and Automation, Vol. 4, 1997, pp. 3274–3280. doi:10.1109/ROBOT.1997.606788.

[25] Y. Fu, Q. Jia, G. Chen, Y. Wang, Regulation of space manipulators with free-swinging joint failure based on iterative learning control, in: International Conference on Intelligent Robotics and Applications, 2021, pp. 360–370. doi:10.1007/978-3-030-89092-6_33.

[26] J.-H. Shin, J.-J. Lee, Dynamic control of underactuated manipulators with free-swinging passive joints in Cartesian space, in: Proceedings of the 1997 IEEE International Conference on Robotics and Automation, 1997, pp. 3294–3299. doi:10.1109/ROBOT.1997.606791.

[27] R. G. Roberts, The dexterity and singularities of an underactuated robot, Journal of Robotic Systems 18 (4) (2001) 159–169. doi:10.1002/rob.1013.

[28] J. D. English, A. A. Maciejewski, Failure tolerance through active braking: a kinematic approach, The International Journal of Robotics Research 20 (4) (2001) 287–299. doi:10.1177/02783640122067408.

[29] J. D. English, A. A. Maciejewski, Robotic workspaces after a free-swinging failure, Journal of Intelligent and Robotic Systems 19 (1) (1997) 55–72. doi:10.1023/A:1007993332111.

[30] J. Nightingale, R. Hind, B. Goodwine, A stopping algorithm for mechanical systems, in: G. S. Chirikjian, H. Choset, M. Morales, T. Murphey (Eds.), Algorithmic Foundation of Robotics VIII, Springer, Berlin, Heidelberg, 2010, pp. 167–180. doi:10.1007/978-3-642-00312-7_11.

[31] J. Nightingale, B. Goodwine, An algorithm for stopping a class of underactuated nonlinear mechanical robotic systems, in: 21st Mediterranean Conference on Control and Automation, 2013, pp. 531–536. doi:10.1109/MED.2013.6608773.

[32] C. D. Pham, P. J. From, J. T. Gravdahl, A geometric approach to the design of serial and parallel manipulators with passive joints, Applied Mathematics 5 (16) (2014) 2585–2601. doi:10.4236/am.2014.516247.

[33] J. Burdick, On the inverse kinematics of redundant manipulators: characterization of the self-motion manifolds, in: Proceedings, 1989 International Conference on Robotics and Automation, 1989, pp. 264–270 vol.1. doi:10.1109/ROBOT.1989.99999.

[34] A. Peidró, O. Reinoso, A. Gil, J. M. Marín, L. Payá, A method based on the vanishing of self-motion manifolds to determine the collision-free workspace of redundant robots, Mechanism and Machine Theory 128 (2018) 84–109. doi:10.1016/j.mechmachtheory.2018.05.013.

[35] D. Chablat, E. Ottaviano, S. Venkateswaran, Self-motion conditions for a 3-PPPS parallel robot with delta-shaped base, Mechanism and Machine Theory 135 (2019) 109–114. doi:10.1016/j.mechmachtheory.2019.02.006.

[36] E. J. Haug, A. Peidro, Redundant manipulator kinematics and dynamics on differentiable manifolds, Journal of Computational and Nonlinear Dynamics 17 (11) (2022) 111008. doi:10.1115/1.4055313.

[37] N. Rakotomanga, I. A. Bonev, A 3-RPR parallel mechanism with singularities that are self-motions, Journal of Mechanisms and Robotics 2 (3) (2010) 034502. doi:10.1115/1.4001737.

[38] J. Cervantes-Sánchez, J. Hernández-Rodríguez, J. Rendón-Sánchez, On the workspace, assembly configurations and singularity curves of the RRRRR-type planar manipulator, Mechanism and Machine Theory 35 (8) (2000) 1117–1139. doi:10.1016/S0094-114X(99)00061-0.

[39] S. Briot, V. Arakelian, I. A. Bonev, D. Chablat, P. Wenger, Self-motions of general 3-RPR planar parallel robots, The International Journal of Robotics Research 27 (7) (2008) 855–866. doi:10.1177/0278364908092466.

[40] D. Chablat, P. Wenger, I. A. Bonev, Self motions of special 3-RPR planar parallel robot, in: Advances in robot kinematics, 2006, pp. 221–228. doi:10.1007/978-1-4020-4941-5_24.

[41] I. A. Bonev, D. Chablat, P. Wenger, Working and assembly modes of the agile eye, in: Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006, pp. 2317–2322. doi:10.1109/ROBOT.2006.1642048.

[42] C. L. Luck, S. Lee, Topology-based analysis for redundant manipulators under kinematic constraints, in: Proceedings of 1995 34th IEEE Conference on Decision and Control, Vol. 2, 1995, pp. 1603–1608. doi:10.1109/CDC.1995.480367.

[43] A. Peidró, L. Payá, S. Cebollada, V. Román, Ó. Reinoso, Solution of the forward kinematics of parallel robots based on constraint curves, in: O. Gusikhin, K. Madani, J. Zaytoon (Eds.), Informatics in Control, Automation and Robotics, Springer International Publishing, Cham, 2022, pp. 386–409. doi:10.1007/978-3-030-92442-3_20.

[44] C. Luck, S. Lee, Redundant manipulator self-motion topology under joint limits with an 8-DOF case study, in: Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vol. 2, 1993, pp. 848–855 vol.2. doi:10.1109/IROS.1993.583191.

[45] B. Brogliato, Nonsmooth mechanics - Models, Dynamics and Control, Springer Cham, 2016. doi:10.1007/978-3-319-28664-8.

[46] C. Ericson, Real-time collision detection, CRC Press, 2004. doi:10.1201/b14581.