

Constrained Orientation Control of a Spherical Parallel Manipulator via Online Convex Optimization

Tasbolat Taunyazov, Matteo Rubagotti and Almas Shintemirov

Abstract—This paper introduces a new framework for the closed-loop orientation control of spherical parallel manipulators (SPMs) based on the online solution of a convex optimization problem. The aim of solving a constrained optimization problem is to define a reference position for the SPM that remains as close as possible to the ideal reference (i.e., the one for which the top mobile platform has the desired orientation), at the same time keeping the SPM within the set of configurations in which collisions between links and singular configurations are avoided (the so-called *feasible workspace*). The proposed approach relies on a recently introduced method for obtaining unique inverse kinematics for SPMs, and on a newly proposed method for generating an approximation of the feasible workspace suitable for fast online optimization. The proposed control scheme is experimentally tested on an Agile Wrist SPM prototype, confirming the performance expected from the theoretical formulation.

Index Terms—Spherical parallel manipulators, robot workspace, robot control, real-time optimization, convex optimization.

I. INTRODUCTION

Spherical parallel manipulators (SPMs) can be employed for the design of orientation platforms providing high stiffness and load-carrying capacity, at the same time guaranteeing high-performance dynamic properties [1]–[3]: these characteristics make SPMs suitable candidates for future industrial orienting devices. Nonetheless, the practical application of SPMs is currently limited, due to challenges related to their closed kinematic structure, such as the limitations of their motion ability (including the problem of self-collisions), and the presence of complex kinematic singularities [4].

At present, control system design for SPMs mostly focuses on 3-DOF SPMs with revolute joints (in short, RRR configuration) and artificially limited workspace. Known examples

This research was funded under the target program #0115PK03041 "Research and Development in the Fields of Energy Efficiency and Energy Saving, Renewable Energy Sources and Environmental Protection for Years 2014-2016" from the Ministry of Education and Science of the Republic of Kazakhstan as part of the "Research of Possible Applications of Renewable Energy for Development of Small/Mobile Autonomous Systems" project.

T. Taunyazov and A. Shintemirov are with the Department of Robotics and Mechatronics and National Laboratory Astana, Nazarbayev University, 53 Kabanbay Batyr Avenue, Astana Z05H0P9, Kazakhstan.

M. Rubagotti is with the Department of Engineering, University of Leicester, University Road, Leicester LE1 7RH, United Kingdom.

Corresponding author: A. Shintemirov, ashintemirov@nu.edu.kz.

This paper has supplementary downloadable material available at <http://ieeexplore.ieee.org>.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier

are, among others, the *Agile Eye*, the *Shade* and the Agile Wrist [5]–[7]. Feedback control schemes based on dynamic models, such as PID control with gravity compensation, have been proposed for specific SPM structures in [8]–[12]. The development of new types of SPMs typically includes a design optimization stage, so as to guarantee that the manipulator can safely operate within an a-priori fixed workspace [13]–[16]. Given the need of avoiding self-collisions and singular configurations, inequality constraints in the SPM workspace play a fundamental role. Thus, there is a need for general control laws that can directly take these constraints into account. In [17], an off-line framework was proposed for generating optimal motor trajectories for general 3-DOF RRR SPMs, based on the kinematics analysis reported in [18]. A feasible configuration space for the SPM was first defined by using numerical procedures, in order to guarantee the absence of singularities and collisions between links during the motion of the manipulator. Point-to-point reference trajectories of the actuated joint servomotors were then defined by solving an optimization problem, based on the configuration space and on the servomotor dynamics.

The approach proposed in [17] can only be applied when the task for the SPM is perfectly known a priori. In order to cope with more complex scenarios, the manipulator needs to track trajectories that are not a-priori planned (for instance, in order to follow a moving object with a radar or a camera): ideally, this would require solving a numerical optimization problem at each sampling instant. In past years, this was not a solution typically implemented in mechatronic systems, due to the high computational burden that optimization problems usually carry with them. However, thanks to recent developments in optimization algorithms (especially in the field of *convex optimization* [19]) and to the availability of more powerful microprocessors, the application of embedded optimization in mechatronics and robotics is becoming a reality, as testified, among others, in the recent works [20]–[24].

This paper introduces a framework for constraint-aware real-time SPM position control: the main objective is to acquire an external reference orientation for the top mobile platform of the manipulator, and then track it while avoiding self-collisions and singular configurations. The quaternion-based orientation kinematics used in the proposed control scheme is presented in Section II, together with the overview of a recently proposed method for obtaining unique solutions for the SPM inverse kinematics. A low-complexity estimate of the feasible workspace is obtained in the form of a

polytope \mathcal{P} using a novel experimental approach, which is compared to the fully numerical approach of [17] (Section III). Exploiting the convexity of \mathcal{P} , the joint reference vector lying within the feasible workspace is obtained via online convex optimization, as outlined in Section IV, where the overall control scheme is also reported. Section V outlines and discusses the results of the experimental demonstration of the proposed framework using a 3D printed prototype of the Agile Wrist SPM. The contribution of this work mainly consists of defining a fast optimization-based procedure to correct in real time the position reference signal for an SPM, so as to obtain a collision-free and singularity-free motion. The definition of the feasible workspace as a simple polytope (suitable for being used in a convex optimization routine) using an experimental procedure is an integral part of the contribution. To the best of the authors' knowledge, the use of real-time numerical optimization in SPM control, in order to guarantee the avoidance of self-collisions and singular configurations, has never been proposed in the literature.

II. SPM MODELING AND KINEMATICS

A. Modeling and conditioning index

The kinematic analysis of 3-DOF RRR SPMs has drawn considerable attention in the past years, and several results have been presented (see, e.g., [7], [25]–[33]). In this work, we adopt the conventional model of a general symmetric 3-DOF SPM with revolute joints [17], illustrated in Fig. 1. The SPM structure includes two pyramid-shape platforms: one at the base (fixed) and one at the top (mobile). The platforms are connected by three legs, numbered as $i = 1, 2, 3$, and equally spaced. Each leg is composed of two curved links, referred to as lower (*proximal*) link, and upper (*distal*) link. The axes of all joints, represented by unit vectors \mathbf{u}_i , \mathbf{v}_i , and \mathbf{w}_i , intersect at the so-called *center of rotation*. Angles α_1 and α_2 define the dimensions of proximal and distal links, respectively. Finally, angles β and γ define two regular pyramids of the top mobile and base platforms (see, Fig. 1). This kind of parallel manipulator is called “spherical” because the motion of the center of the top platform is mechanically confined on the surface of a sphere whose center coincides with the center of rotation.

Fig. 1 illustrates the SPM right-handed orthogonal coordinate system with its origin located at the center of rotation. The z -axis is perpendicular to the base pyramid platform; the y -axis belongs instead to the plane spanned by the z -axis and \mathbf{u}_1 . The joint angles, namely θ_i , $i = 1, 2, 3$, are measured respectively from the plane spanned by the z axis and \mathbf{u}_i , to the plane of the corresponding proximal link. The spatial orientation of the whole SPM system (i.e., the orientation of its base platform) is defined in a stationary coordinate system. The latter, for the case of a horizontal position of the SPM base platform, is aligned with the SPM coordinate system in Fig. 1.

Let the SPM “home” orientation be defined as the non-singular configuration with the top mobile and base platforms being horizontal and parallel to each other and with all proximal links rotated to the same side, i.e., the configuration shown in Fig. 1. For manipulator control purposes, the rotation

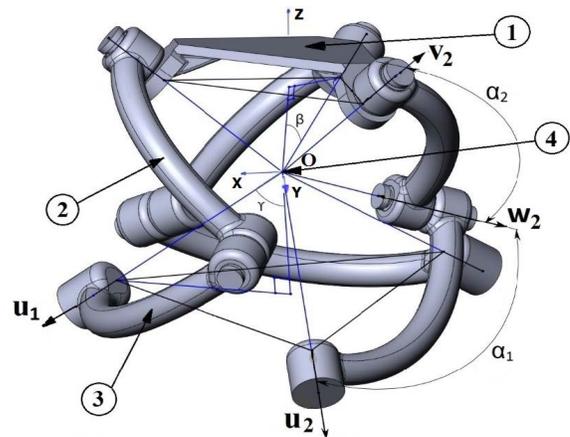


Fig. 1. 3D representation of the kinematic model of a 3-DOF RRR SPM: (1) - top mobile platform, (2) - distal link, (3)- proximal link, (4) - center of rotation.

of the top platform can be described using quaternions [34]. Let quaternion \mathbf{q}_{ref} denote the SPM rotation from the home configuration defined by vectors $\mathbf{v}_{\text{home},i}$ ($i = 1, 2, 3$) to a new reference configuration defined by $\mathbf{v}_{\text{ref},i}$. Similarly, quaternion \mathbf{q}_{base} defines the SPM base platform spatial rotation with respect to the SPM stationary coordinate system, and thus, to vectors $\mathbf{v}_{\text{home},i}$ ($i = 1, 2, 3$).

Employing the quaternion-based rotation sequence rule [34], the resulting SPM orientation quaternion \mathbf{q} is defined as

$$\mathbf{q} = \mathbf{q}_{\text{base}} \cdot \mathbf{q}_{\text{ref}}. \quad (1)$$

The new reference vectors $\mathbf{v}_{\text{ref},i}$ ($i = 1, 2, 3$) are defined as follows:

$$\mathbf{v}_{\text{ref},i} = \mathbf{q} \otimes \mathbf{v}_{\text{home},i} \otimes \mathbf{q}^*, \quad i = 1, 2, 3, \quad (2)$$

where \mathbf{q}^* denotes the quaternion conjugate of \mathbf{q} and \otimes is the quaternion multiplication operator.

Another important concept used in the paper, tightly related to the avoidance of singular configurations during the SPM motion (see, e.g., [7]) is the *conditioning index* $\zeta(\mathbf{J}) \in (0, 1)$, in which \mathbf{J} is the Jacobian matrix of the SPM. A value of $\zeta(\mathbf{J})$ close to zero coincides with a singular configuration, while $\zeta(\mathbf{J})$ close to 1 corresponds to a so-called *isotropic configuration*. The reader is referred to [17] for a simple description of the singularity analysis, which includes the derivation of $\zeta(\mathbf{J})$.

B. SPM inverse kinematics

In order to formulate the control problem, it is necessary to obtain a unique solution for the SPM inverse kinematics, which is usually challenging in parallel manipulators. If this is achieved, it is possible to unambiguously define specific values of the actuated joint angles given a certain orientation of the top mobile platform. In the remainder of the paper, the term *configuration* refers to a particular value for the vector of input joint (i.e., actuator) angles

$$\boldsymbol{\theta} \triangleq [\theta_1 \quad \theta_2 \quad \theta_3]^T, \quad (3)$$

with T being the transposition operator.

Given a known orientation \mathbf{v}_i , $i = 1, 2, 3$, of the SPM mobile platform, the SPM inverse kinematic solutions are defined by three uncoupled equations for obtaining $\boldsymbol{\theta}$, as described in [26], [35]. For any given orientation (four in total) of the SPM top mobile platform, these equations provide two solutions for each angle θ_i on each SPM leg $i = 1, 2, 3$ [26], also referred to as *working modes* [30]. Therefore, eight possible solutions are provided for the inverse kinematic problem. In this paper, the algorithm presented in [18] (to which we refer the reader for details), is used. This algorithm is defined in order to employ only one of the eight solutions at each sampling instant, also maintaining the same working mode during the whole robot motion. For further details on the choice of the working mode for the SPM, the reader is referred to [7], [26].

III. ESTIMATION OF THE FEASIBLE WORKSPACE

The feasible workspace is obtained as a convex set of values of $\boldsymbol{\theta}$ for which singularities and link collisions are avoided. A numerical approach for defining an analogous workspace was presented in [17]: in this work an alternative experimental approach, aimed at reducing conservativity, is instead proposed. In Section V-B it is shown that a larger workspace as compared to the approach in [17] is obtained for the considered SPM by using the newly proposed experimental approach. In order to improve readability, the main concepts of computational geometry needed for the understanding of Section III-B are reported in the Appendix.

A. Experimental methodology

In order to generate experimental data, the SPM prototype is set up to rotate slowly, with no payload, spanning configurations within an over-estimate of its feasible workspace, defined based on the limitations of joint rotation angles imposed by the mechanical structure of the SPM prototype. This set is given as

$$\tilde{\mathcal{P}} \triangleq \{\boldsymbol{\theta} : \boldsymbol{\theta}_\ell \leq \boldsymbol{\theta} \leq \boldsymbol{\theta}_u\}. \quad (4)$$

where $\boldsymbol{\theta}_\ell \in \mathbb{R}^3$ and $\boldsymbol{\theta}_u \in \mathbb{R}^3$ contain lower and upper bounds, respectively, for the three components of $\boldsymbol{\theta}$, and the inequalities are to be intended component-wise. An equally-spaced 3D-grid of points is defined in $\tilde{\mathcal{P}}$, with $\epsilon > 0$ as the distance between two consecutive points of the grid along each axis. The discrete set of points of the grid is referred to as $\tilde{\mathcal{P}}_g$, and can be formally defined as

$$\tilde{\mathcal{P}}_g \triangleq \left\{ \boldsymbol{\theta} \in \tilde{\mathcal{P}} : \boldsymbol{\theta} = \boldsymbol{\theta}_\ell + \epsilon [m_1 \ m_2 \ m_3]^T \right\}, \quad (5)$$

where m_1 , m_2 and m_3 are positive integers. The intermediate goal of the proposed methodology for ultimately defining \mathcal{P} consists of splitting $\tilde{\mathcal{P}}_g$ into two subsets \mathcal{V} and \mathcal{F} , including the configurations that are or are not, respectively, contained in the feasible workspace. It is important to notice that, due to the symmetric structure of the manipulator, given three angles θ_α , θ_β and θ_γ , the three configurations $\boldsymbol{\theta}_a = [\theta_\alpha \ \theta_\beta \ \theta_\gamma]^T$, $\boldsymbol{\theta}_b = [\theta_\gamma \ \theta_\alpha \ \theta_\beta]^T$, and $\boldsymbol{\theta}_c = [\theta_\beta \ \theta_\gamma \ \theta_\alpha]^T$ are either all contained in \mathcal{V} or all contained in \mathcal{F} . Thus, only a subset of the points in $\tilde{\mathcal{P}}_g$ has to be checked (about one third if the lower and upper bounds in $\tilde{\mathcal{P}}$ are the same for all three angles).

The SPM prototype rotates with only one actuated joint in motion at a time with a predefined joint angle step equal to ϵ , through a set of configurations ensuring the SPM symmetrical motion with respect to $\boldsymbol{\theta}_{\text{home}}$. As the motion proceeds, the grid points through which the SPM moves are added to \mathcal{V} . However, when a singular configuration or a collision is detected, all further points in that direction are assigned to \mathcal{F} and the manipulator is positioned back to its home assembly configuration, from which it will move towards parts of $\tilde{\mathcal{P}}_g$ which have not been explored yet. At the end of the experiment, all configurations in $\tilde{\mathcal{P}}_g$ have been classified as belonging to either \mathcal{V} or \mathcal{F} . The procedures for detecting singular configurations and collisions are hereafter explained.

1) *Detection of singular configurations*: The grid points corresponding to singular (or near-singular) configurations are detected numerically as those satisfying condition

$$\zeta(\mathbf{J}) < \zeta(\mathbf{J})_{\min}. \quad (6)$$

where the *minimum threshold index* $\zeta(\mathbf{J})_{\min}$ is a tuning parameter.

2) *Collision detection*: The SPM configurations corresponding to collisions between links are instead detected experimentally. Whenever a link collision occurs, this causes the stalling of the joint actuators (servomotors) which leads to a sharp increase of at least one of the motor supply currents. This can be detected by high-pass filtering the current signals in order to extract their high-frequency components: the SPM configurations at which such events are detected indicate the boundaries of the feasible workspace. To prevent the damage of the SPM setup, the actuator velocities are kept low and, as soon as a link collision event is detected, the SPM reverses its motion and returns to its home position.

The collision detection method originally proposed in [36] for human-robot collisions is here employed for detecting self-collisions. The motor currents $\mathbf{I} \in \mathbb{R}^3$ are processed using a digital 3rd-order Chebyshev filter with experimentally-tuned cut-off frequency. The filtered currents are then compared with time-varying thresholds τ_{HPF} , defined as follows [36]:

$$\tau_{\text{HPF}} = \tau_{\text{H}_{\min}} + k_{\text{H}_v} \frac{|\dot{\theta}|}{v_{\max}} + k_{\text{H}_a} \frac{|\ddot{\theta}|}{a_{\max}} > 0. \quad (7)$$

where $\dot{\theta}$ and $\ddot{\theta}$ are the angular speed and acceleration, respectively, of the considered actuator. Firstly, the positive parameter $\tau_{\text{H}_{\min}}$ is chosen as the smallest value of any filtered currents in static conditions ($\dot{\theta} = \ddot{\theta} = 0$), for all SPM configurations. Then, the positive gain k_{H_v} is selected so that the value of $\tau_{\text{H}_{\min}} + k_{\text{H}_v} \frac{|\dot{\theta}|}{v_{\max}}$ is an upper bound for the value of the motor current when the actuator runs at its maximum (constant) velocity $v_{\max} > 0$. Finally, $k_{\text{H}_a} > 0$ is tuned so that the whole expression in the right-hand side of the equation in (7) is an upper bound for the motor current, at maximum actuator acceleration $a_{\max} > 0$. Since $\tau_{\text{HPF}}(\boldsymbol{\theta})$ is positive and defines the critical upper bound, the same value with negative sign is taken as the critical lower bound. The reader is referred to [36] for further details.

B. Convex approximation of the feasible workspace

A possible numerical estimate of the feasible workspace would be the union of cubes, all with edge length equal to ϵ , each of them centered at a point of \mathcal{V} . Using such an estimate, however, one would need to verify if a given point, provided as online external reference, belongs to any of the cubes, which can be increasingly computationally demanding as ϵ is reduced to improve precision. The proposed solution consists of a method to easily detect online if a given point is in the feasible workspace, by solving a convex optimization problem. This requires providing an estimate of the feasible workspace as a convex set: the computational effort is therefore moved from the online computation to offline calculations. Different methods to generate an internal convex approximations of a non-convex set have been proposed when an analytical expression is given for the non-convex set [37]–[39], while, to the best of the authors' knowledge, no standard approaches exists when an analytical expression is not available. In the remainder of this section, we propose an iterative method to determine a convex inner approximation of the feasible workspace in the form of a polytope (Definition 4, Appendix A). As a preliminary step, a collection of polytopes \mathcal{C} is defined as a list containing all cubes with edge length ϵ centered around the points of \mathcal{F} , namely C_k , $k = 1, \dots, |\mathcal{F}|$, with $|\cdot|$ standing for cardinality of a set. Since the home configuration tends to be located at a central position in the feasible workspace, θ_{home} is taken as the first point of its temporary approximation, namely $\mathcal{P}_{\text{temp}} = \theta_{\text{home}}$, and the list of vertices of $\mathcal{P}_{\text{temp}}$ is initialized as $\mathcal{V}_{\mathcal{P}} = \theta_{\text{home}}$. The six grid points that are nearest to θ_{home} in the 3D space are included in a set named \mathcal{V}_N , and then the first set of points to be tested is defined as $\mathbf{v}_j \in \mathcal{V}_N \cap \mathcal{V}$. After applying an ordering criterion for points \mathbf{v}_j , $\mathcal{P}_{\text{temp}}$ is updated as $\mathcal{P}_{\text{temp}} = \text{co}(\mathcal{P}_{\text{temp}}, \mathbf{v}_1)$, with $\text{co}(\cdot)$ standing for ‘‘convex hull’’ (Definition 2, Appendix A). This is simply a segment, given the initialization $\mathcal{P}_{\text{temp}} = \theta_{\text{home}}$. Then, the other points are tested following their order, and each time a polytope is recursively defined as $\mathcal{P}_{\text{temp}} = \text{co}(\mathcal{P}_{\text{temp}}, \mathbf{v}_j)$. In this case, the so-called *V-representation* (Definition 5, Appendix A) is the most suitable (Remark 1, Appendix A). Indeed, at each iteration, it is necessary to verify if the newly defined polytope intersects any of the cubes contained in \mathcal{C} , which is carried out by checking if the intersection between $\mathcal{P}_{\text{temp}}$ and all sets of \mathcal{C} is an empty set. This operation is carried out by using the so-called H-representation (Definition 6 and Remark 1, Appendix A). If there is no intersection, \mathbf{v}_j is added to $\mathcal{V}_{\mathcal{P}}$. Instead, if an intersection is detected, then \mathbf{v}_j is removed from \mathcal{V} , and $\mathcal{P}_{\text{temp}}$ is not updated. Once all points \mathbf{v}_j have been tested, a new set \mathcal{V}_N of nearest points to $\mathcal{P}_{\text{temp}}$ has to be determined. For all points $\mathbf{v}_k \in \mathcal{V} \setminus \mathcal{V}_{\mathcal{P}}$, the distance to $\mathcal{P}_{\text{temp}}$, defined as

$$\text{dist}(\mathbf{v}_k, \mathcal{P}_{\text{temp}}) \triangleq \inf \{ \|\mathbf{v}_k - \theta\|_2 : \theta \in \mathcal{P}_{\text{temp}} \}, \quad (8)$$

can be determined numerically by solving the following QP [19, Sec. 8.1.1]:

$$\text{dist}(\mathbf{v}_k, \mathcal{P}_{\text{temp}}) = \arg \min_{\theta} \|\mathbf{v}_k - \theta\|_2^2 \quad (9a)$$

$$\text{subj. to } \mathbf{A}_p \theta \leq \mathbf{b}_p, \quad (9b)$$

where $\mathbf{A}_p \theta \leq \mathbf{b}_p$ is the H-representation of $\mathcal{P}_{\text{temp}}$. Then, the points \mathbf{v}_j are redefined as those to which the minimum distance, for this specific realization of $\mathcal{P}_{\text{temp}}$, is associated. The procedure already detailed is repeated for the new set of points \mathbf{v}_j , which are the nearest to the newly generated polytope $\mathcal{P}_{\text{temp}}$. The same procedure is repeated until $\mathcal{V} \equiv \mathcal{V}_{\mathcal{P}}$. At this point the inner approximation of the feasible workspace $\mathcal{P} = \mathcal{P}_{\text{temp}}$ is defined, in its H-representation, as

$$\mathcal{P} = \{ \theta \in \mathbb{R}^3 : \mathbf{A}_p \theta \leq \mathbf{b}_p \}. \quad (10)$$

A more formal description of the method is given as pseudo-code in Algorithm 1. In the algorithm, it is assumed for simplicity that also the nearest six points to θ_{home} are found by solving (9b) for all points of $\mathcal{V} \setminus \mathcal{V}_{\mathcal{P}}$, which can be avoided, as previously described, to reduce the computational load at least for the first iteration. Also, a graphical example for a toy 2D case is shown in Fig. 2.

Algorithm 1: Convex estimate of the feasible workspace

Input: $\mathcal{V}, \mathcal{F}, \theta_{\text{home}}$

Output: \mathcal{P}

```

/* collection of ``forbidden'' areas */
 $\mathcal{C} \leftarrow \emptyset;$ 
for  $k \leftarrow 1$  to  $|\mathcal{F}|$  do
     $\mathcal{C} \leftarrow \{\mathcal{C}, C_k\};$ 
/* initialization of  $\mathcal{P}_{\text{temp}}$  */
 $\mathcal{P}_{\text{temp}} \leftarrow \{\theta_{\text{home}}\};$ 
/* initialization of vertex list of  $\mathcal{P}_{\text{temp}}$  */
 $\mathcal{V}_{\mathcal{P}} \leftarrow \{\theta_{\text{home}}\};$ 
 $i \leftarrow 0;$ 
while  $\mathcal{V} \setminus \mathcal{V}_{\mathcal{P}} \neq \emptyset$  do
     $i \leftarrow i + 1;$ 
    for  $k \leftarrow 1$  to  $|\mathcal{V} \setminus \mathcal{V}_{\mathcal{P}}|$  do
         $\mathbf{v}_k \leftarrow k$ -th element of  $\mathcal{V} \setminus \mathcal{V}_{\mathcal{P}}$ ;
        /* distance computed as in (9b) */
         $d_k \leftarrow \text{dist}(\mathbf{v}_k, \mathcal{P}_{\text{temp}});$ 
     $d_{\min} \leftarrow \min \{d_k\}_{k=1}^{|\mathcal{V} \setminus \mathcal{V}_{\mathcal{P}}|};$ 
     $\mathcal{V}_N \leftarrow \{\mathbf{v}_j : d_j == d_{\min}\};$ 
    for  $j \leftarrow 1$  to  $|\mathcal{V}_N|$  do
         $\mathcal{P}'_{\text{temp}} = \text{co}(\mathcal{P}_{\text{temp}}, \mathbf{v}_j);$ 
        intersect  $\leftarrow$  false;
        for  $k \leftarrow 1$  to  $|\mathcal{F}|$  do
            if  $\mathcal{P}'_{\text{temp}} \cap C_k \neq \emptyset$  then
                intersect  $\leftarrow$  true;
                break;
        if intersect == false then
             $\mathcal{P}_{\text{temp}} \leftarrow \mathcal{P}'_{\text{temp}};$ 
             $\mathcal{V}_{\mathcal{P}} \leftarrow \mathcal{V}_{\mathcal{P}} \cup \{\mathbf{v}_j\};$ 
        else
             $\mathcal{V} \leftarrow \mathcal{V} \setminus \{\mathbf{v}_j\};$ 
     $\mathcal{P} \leftarrow \mathcal{P}_{\text{temp}};$ 
return  $\mathcal{P}.$ 

```

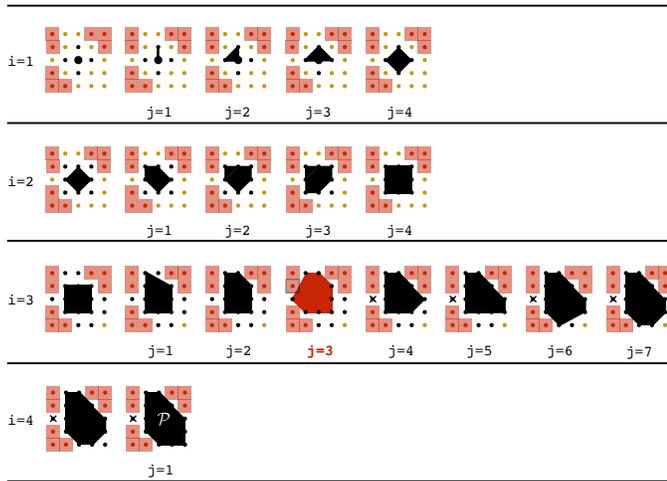


Fig. 2. Graphical 2D representation of the procedure described in Algorithm 1. The red dots and the surrounding squares represent sets C_k , and the central big black dot represents θ_{home} . At each iteration $i = 1, \dots, 4$ of the outer while loop in Algorithm 1, the black dots are the elements of \mathcal{V}_N , determined as those that minimize the distance to $\mathcal{P}_{\text{temp}}$, and the yellow dots are the elements of $\mathcal{V} \setminus \mathcal{V}_P$. Notice that the polytope $\mathcal{P}_{\text{temp}}$ grows in size at each iteration j of the for loop, apart from the cases when an intersection is detected (for this example, only when $(i, j) = (3, 3)$). The points removed from \mathcal{V} when an intersection is detected are indicated with a “x”.

IV. CONSTRAINED SPM CONTROL FRAMEWORK

The definition of \mathcal{P} is finally inserted in the proposed overall control framework, outlined in Fig. 3. At each sampling instant, the position reference for the SPM top platform is either fixed a priori (for inertial stabilization) or obtained from an external device (e.g., joystick). On the other hand, the SPM spatial orientation is obtained from an IMU sensor mounted on the SPM base platform. Both values are provided in the form of rotation quaternions \mathbf{q}_{ref} and \mathbf{q}_{base} , respectively. The SPM reference is then represented in the form of a Cartesian reference vector \mathbf{v}_{ref} , still accounting for the orientation of the top mobile platform, and subsequently into a joint-space vector Θ^* , representing the reference angles for the SPM joints, actuated by servomotors with low-level position control. This is computed using the quaternion-based orientation kinematics and the approach for obtaining unique inverse kinematic solution, all recalled in Section II.

The convex estimate of the feasible SPM workspace \mathcal{P} , obtained as described in Section III (or, alternatively, using the numerical approach proposed in [17]), is used for computing the constrained joint reference vector θ^* , defined as the point of \mathcal{P} that is closest to Θ^* . In this way, the control system aims at providing a reference configuration as close as possible to the ideal one, while avoiding collisions and singular configurations. The value of θ^* is obtained by projecting Θ^* onto \mathcal{P} : this is a QP, in the same form of (9b), and precisely

$$\theta^* = \arg \min_{\theta} \|\theta - \Theta^*\|_2^2 \quad (11a)$$

$$\text{subj. to } \mathbf{A}_p \theta \leq \mathbf{b}_p. \quad (11b)$$

Thanks to the current performance of microprocessors and to the latest refinements in QP solvers, the online solution of (11a) for the considered problem size typically requires a time

interval of tens of microseconds. This result is obtained at the expense of spending several hours for generating \mathcal{P} , which however has to be done only once, and offline.

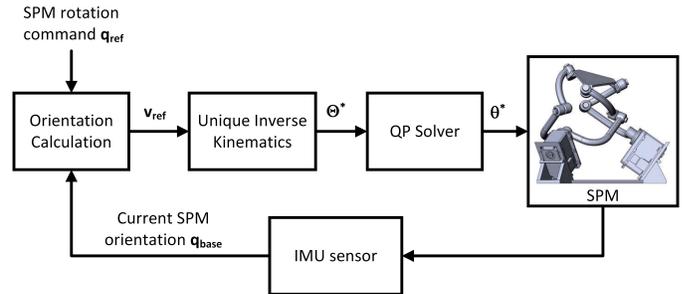


Fig. 3. Block diagram of the SPM orientation control framework.

V. RESULTS AND DISCUSSION

A. Experimental setup

The experimental demonstration of the proposed framework is carried out on an “Agile Wrist” SPM prototype. The mechanical part of the prototype is designed in SolidWorks CAD and manufactured using 3D printing technology with ABS plastic. The Agile Wrist SPM prototype is actuated by three Dynamixel MX-106 servomotors with internal PID position control loop fixed to the base platform. The SPM has three identical legs and $\alpha_1 = \alpha_2 = 90^\circ$ (see, Section II-A). The three unit vectors \mathbf{u}_i and \mathbf{v}_i , $i = 1, 2, 3$, are mutually orthogonal [7], which implies $\beta = \gamma = 54.7^\circ$ [17].

The home configuration of the Agile Wrist corresponds to $\theta_{\text{home}} = [135^\circ \ 135^\circ \ 135^\circ]^T$. In this configuration the axes of base and top joints are aligned as $\mathbf{u}_1 = -\mathbf{v}_3$, $\mathbf{u}_2 = -\mathbf{v}_1$, $\mathbf{u}_3 = -\mathbf{v}_2$ and the SPM top mobile and base platforms are parallel to each other. The angles in θ are set from the vertical plane generated by the z axis and vectors \mathbf{u}_i , to the plane of the corresponding proximal links in the servomotor counterclockwise direction, as shown in [18].

B. Estimation of the feasible workspace

Set $\tilde{\mathcal{P}}$, as defined in (4), was defined by setting all entries of θ_ℓ equal to 60° , and all elements of θ_u equal to 170° , according to the prototype mechanical limitations. A set of uniformly-sampled input SPM configurations θ with spacing $\epsilon = 2^\circ$ was adopted for the calculation of \mathcal{P}_g , as defined in (5).

Following the methodology described in Section III-A, all SPM configurations θ were tested to detect near-singular configurations by checking condition (6) with $\zeta(\mathbf{J})_{\min} = 0.3$. Self collisions were detected using the actuator current measurements processed with the Chebyshev filter

$$H(z) = \frac{0.1782 - 0.5345z^{-1} + 0.5345z^{-2} - 0.1782z^{-3}}{1 - 0.0241z^{-1} - 0.5080z^{-2} + 0.1067z^{-3}}, \quad (12)$$

whose cutoff frequency of 30 Hz was determined based on trial and error.

Fig. 4 demonstrates an example of self-collision detection when the servomotor of Joint 3 was being actuated. One can

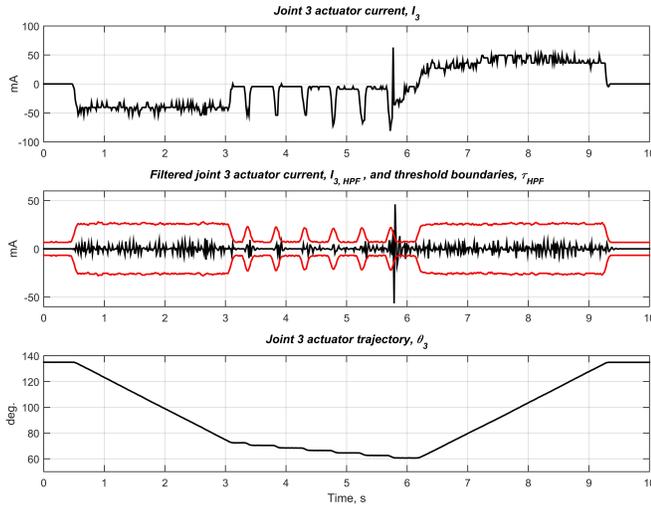


Fig. 4. From top to bottom, time evolutions of, respectively: measured currents of the Joint 3 servomotor (Dynamixel MX-106); high-pass filtered currents (in black) and time-varying threshold (in red); joint position θ_3 .

notice that, while θ_3 moves towards its lower bound, the high-pass filtered current $I_{3,HPF}$ at $t = 5.8$ s exceeds the time-varying threshold τ_{HPF} , computed according to (7) with $\tau_{H,min} = 6.7$ mA, $k_{H_v} = 2.1$ mA and $k_{H_a} = 0.65$ mA, using real-time measurements of velocity and acceleration. As soon as the link collision event is detected, the SPM prototype reverses its motion and returns to its home position, i.e. $\theta_{3,home} = 135^\circ$. The experimental testing of all grid configurations in $\tilde{\mathcal{P}}_g$ lasted 13 hours. The whole experiment and the computation of \mathcal{P} are conducted on a PC workstation with 3.5 GHz CPU, 8 Gb RAM and 64-bit Windows operating system. Fig. 5(a) illustrates all tested configurations: the points corresponding to feasible configurations (i.e., those contained in \mathcal{V}) are depicted in blue, whereas the infeasible configurations (i.e., those contained in \mathcal{F}) are depicted in red.

Following Algorithm 1, \mathcal{P} is determined as shown in Fig. 5(b) by using the functions provided with the Multi-Parametric Toolbox [40] for MATLAB, which implements all the required computational geometry calculations. The H-representation of \mathcal{P} , given by $\mathbf{A}_p\boldsymbol{\theta} \leq \mathbf{b}_p$, is composed of $n_p = 28$ linear inequalities, corresponding to an equal number of facets.

C. Comparison with the workspace of [17]

Fig. 5(c) and Table I show a comparison with the numerical approach of [17], for the same Agile Wrist SPM prototype. In [17], the space occupied by the links was over-approximated using nine cylinder-like shapes in total. From this geometrical description, a set analogous to \mathcal{V} (here referred to as \mathcal{V}') containing feasible points within a 3D grid, was obtained via numerical simulation. Finally, a polytope \mathcal{P}' (analogous to \mathcal{P}) was generated based on \mathcal{V}' by defining its facets following an ad-hoc manual procedure. In Table I, one can see that the proposed experimental approach provides a 5.5% improvement in volume as compared to the method proposed in [17]. This can be attributed to the fact that, in [17], the space occupied by the links was over-approximated, which led to

defining \mathcal{V}' in a conservative way as compared to \mathcal{V} . Although the conservativity of the approach of [17] can be decreased by using a more precise approximation of the space occupied by the links, the method proposed in this paper eliminates this kind of conservativity thanks to its experimental nature. The disadvantages of this approach are the large time needed for running the experimental procedure, and the risk of damaging the SPM when self collisions occur. Therefore, the method proposed in [17] would be more suitable when dealing with a fragile or high-precision platform, that could be damaged when executing the procedure described in Section III-A. As a final consideration, from Table I one can notice that the number of facets of \mathcal{P} (coinciding with the number of inequalities contained in $\mathbf{A}_p\boldsymbol{\theta} \leq \mathbf{b}_p$) is considerably lower than that of \mathcal{P}' . The procedure defined in Algorithm 1 is not aimed at minimizing n_p , so one cannot conclude that this improvement is due to its use rather than using an ad-hoc manual procedure. Nonetheless, reducing n_p leads to a smaller computation time for solving the QP (11), which is crucial for the real-time implementation of the control routine.

TABLE I
COMPARISON OF \mathcal{P} AND \mathcal{P}' FOR THE AGILE WRIST SPM PROTOTYPE

Properties	\mathcal{P}' [17]	\mathcal{P}
Number of facets n_p	61	28
Polytope volume	6.5309e+05	6.9167e+05

D. Comparison with an “optimization-free” method

As already mentioned in Section I, the mechanical structure of an SPM can be designed so as to guarantee that the manipulator can reach an a-priori fixed set of configurations (see, e.g., [13]–[16] for the design optimization of different parallel robots than the Agile Wrist considered in this work). In the following, we propose a simple method to obtain a constrained SPM control strategy that does not require solving any real-time optimization problem. Assume that the Agile Wrist SPM prototype was designed to guarantee that all three joint angles can independently vary between 110° and 150° , without incurring into any self-collision or singular configuration. The resulting workspace in the joint space, represented in Fig. 5(c) in blue, would be the cube

$$\mathcal{S} \triangleq \{\boldsymbol{\theta} \in \mathbb{R}^3 : \theta_i \in [110^\circ, 150^\circ], i = 1, 2, 3\}. \quad (13)$$

Using directly set \mathcal{S} instead of \mathcal{P} to enforce constraint satisfaction would not require solving any optimization problem, as one could simply determine Θ^* by saturating the components of $\boldsymbol{\theta}$ component-wise. Also, no analytical or experimental procedure would be needed to find a workspace estimate, as \mathcal{S} is by design a feasible workspace. However, the direct use of \mathcal{S} would lead to a more conservative imposition of the constraints, since \mathcal{P} could in fact be much larger. The proposed QP-based method thus allows for an enlargement of the set of feasible configuration already guaranteed from the design optimization phase, but can also be used when no a-priori design-optimization has been carried out, determining the workspace \mathcal{P} completely a posteriori.

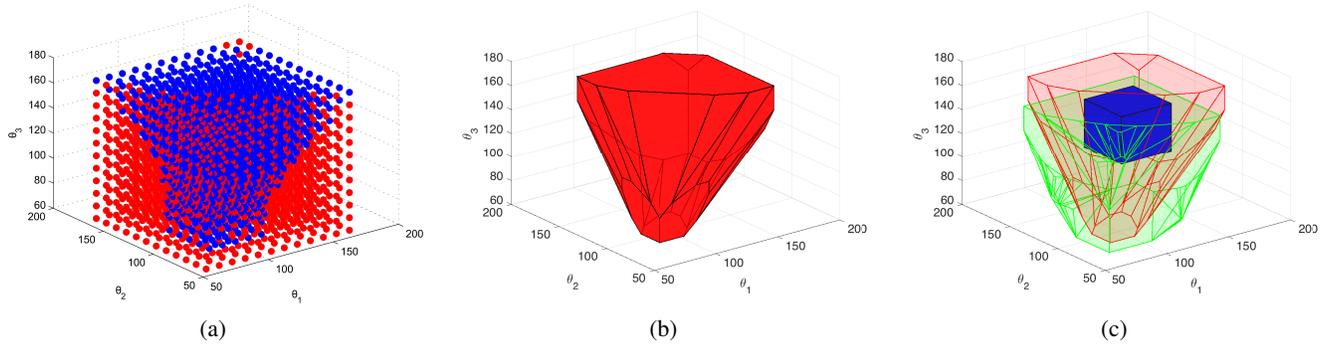


Fig. 5. Estimation of the feasible SPM workspace for the Agile Wrist SPM prototype: (a) - Experimentally obtained SPM configurations, divided into sets \mathcal{V} (blue) and \mathcal{F} (red); (b) - Graphical representation of polytope \mathcal{P} ; (c) - Comparison of \mathcal{P} (red) with \mathcal{S} (blue) and with \mathcal{P}' (green), the latter obtained in [17] for the same SPM prototype.

E. Constrained SPM control

For the implementation of the orientation control, the Agile Wrist SPM prototype was additionally equipped with an UM7-LT orientation sensor from CHRobotics [41], attached to the SPM base platform. This sensor combines triaxial accelerator, rate gyro, and magnetometer data using an extended Kalman filter to produce an estimate of the orientation of the base platform in the form of an orientation quaternion at each sampling instant.

The closed-loop control framework outlined in Fig. 3 was implemented in C code and included the calculation of the SPM orientation and unique inverse kinematic solution, in addition to the QP solver. In particular, the CVXGEN tool [42] was used to obtain the C code implementation of the QP solver routine (based on interior point methods) embedded into the control program, which determines the value of θ^* . The control system was implemented on the previously-mentioned PC workstation using the cross-platform Qt C++ development environment. The SPM actuators communicated with the PC via UART-USB, while the IMU sensor was connected with a Wixel microcontroller via USB. The sampling time of the control process, fixed at 25 ms, was strictly enforced and relied on the PC's system clock. Employing the experimentally-defined workspace polyhedron \mathcal{P} , described in the CVXGEN interface as a set of $n_p = 28$ inequalities, the worst-case QP computation time did not exceed $50 \mu\text{s}$, which was significantly smaller than the worst-case overall control loop cycle time of about 23 ms, as illustrated in Fig. 6.

Two different experimental scenarios are considered in the remainder of this section.

1) *Inertial stabilization*: The goal is to stabilize the top mobile platform of the Agile Wrist prototype at a horizontal position while the SPM base platform is rotated arbitrary, as demonstrated in Fig. 7(a). The SPM reference configurations Θ^* are determined online with a sampling time $T_s = 25$ ms taking into account the platform orientation feedback from the IMU sensor and the SPM configuration obtained from the angular position feedback of the servomotors.

Fig. 7(b) shows a case in which the base platform is very inclined, and the reference configuration Θ^* , computed by the inverse kinematics block, is directly provided to the servomotors. As a result, a self-collision occurs. Fig. 7(c) shows instead

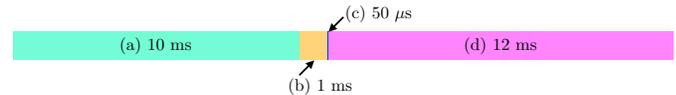


Fig. 6. Worst-case execution time diagram of one cycle of the SPM control loop: a) data acquisition time, b) orientation calculation time, c) QP solution time, d) actuation communication time.

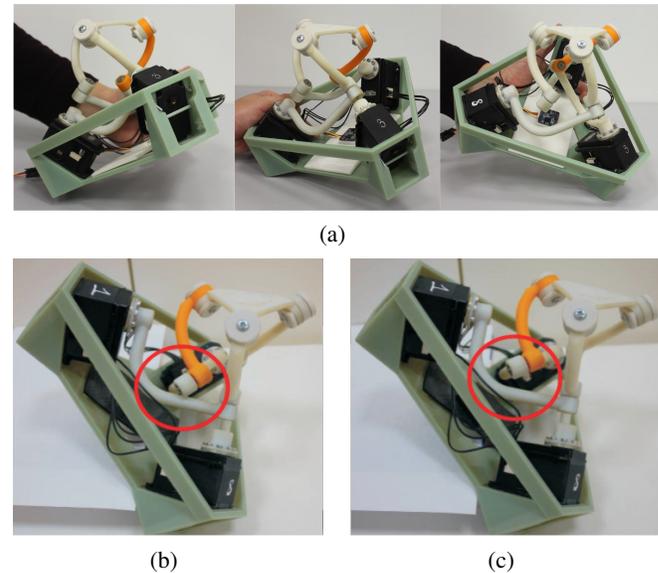


Fig. 7. Inertial stabilization of the Agile Wrist SPM prototype (a); large displacement of the SPM base platform: (b) - the reference Θ^* is directly imposed, resulting in link collision; (c) - the constrained reference θ^* is used, and collision avoidance is achieved.

the case when the constrained reference configuration θ^* is provided. The outcome, as expected, is that the link stops right before a collision would have occurred. Fig. 8 shows instead the time evolution of Θ^* and θ^* during an experiment. In the time intervals when $\Theta^* \in \mathcal{P}$, then $\theta^* \equiv \Theta^*$, while the values differs when Θ^* exits \mathcal{P} . In Fig. 8, we also reported the corresponding simulated evolution of the joint angles in case \mathcal{S} in (13) is employed to impose the constraints. As expected, the corresponding joint angle references are more conservative with respect to those obtained solving the QP in (11) when the unconstrained references exceed the feasible workspace. The

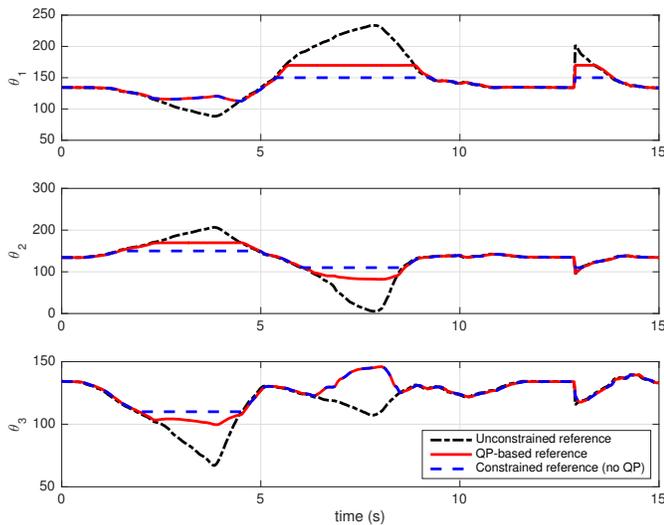


Fig. 8. Experimental time evolutions of signals Θ^* (dash-dot black line) and θ^* (solid red line) applied to the Agile Wrist SPM prototype for inertial stabilization. Also, the corresponding simulated evolution of θ by using workspace \mathcal{S} is shown as a dashed blue line.

advantage of using set \mathcal{S} is a reduced computation time, but, as shown in Fig. 6, the computation time needed for solving the QP was already negligible when compared with the overall time needed for data acquisition, orientation calculation, and communication to the actuators.

Finally, Fig. 9 shows a graphical representation of Θ^* and θ^* in the configuration space, for the specific case in which

$$\Theta^* \triangleq [187.12^\circ \quad 230.71^\circ \quad 191.60^\circ]^T, \quad (14)$$

$$\theta^* \triangleq [157.76^\circ \quad 170.00^\circ \quad 162.24^\circ]^T. \quad (15)$$

2) *Tracking an external orientation reference for the top platform:* A further degree of difficulty can be added to the inertial stabilization scenario, in a case when the top platform should not be kept flat, but rather follow an orientation profile provided by an external device. In this case, the value of Θ^* will depend on both the external reference, and on the orientation of the base platform. This scenario perfectly fits into the proposed framework, and the same exact procedure is followed in order to obtain a reference vector θ^* inside the feasible workspace. Fig. 10 demonstrates the SPM orientation control implemented on the Agile Wrist SPM prototype at different SPM base platform positions. A flight simulation game joystick is used for imposing SPM orientation references. As shown in Fig. 10(c) when the imposed orientation reference Θ^* is defined outside the manipulator workspace and directly applied to the SPM, the SPM top mobile platform motion causes the link collision similarly to the previous scenario. On the other hand, the reference θ^* computed for the same value of Θ^* constrains the manipulator motion within its workspace, thus preventing link collisions and singularities as seen in Fig. 10(d).

A video, provided as supplementary multimedia content, shows additional experimental results and scenarios for the two problems described in this section.

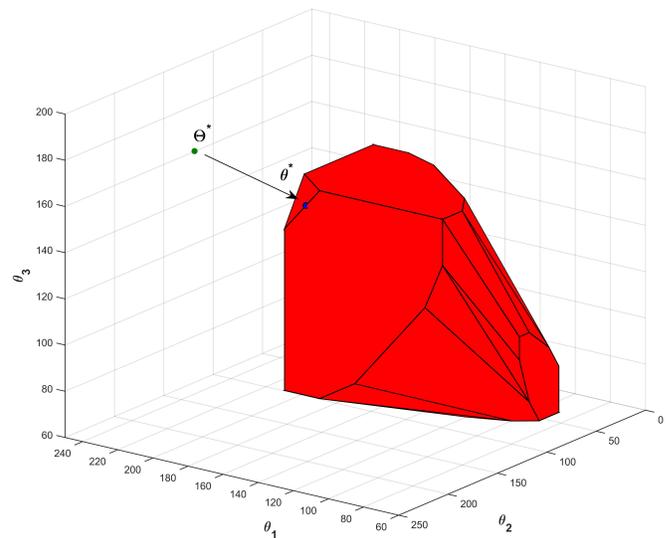


Fig. 9. Graphical representation, in the configuration space, of the generation of θ^* via QP: the point corresponding to Θ^* is projected onto \mathcal{P} , thus obtaining θ^* .

VI. CONCLUSIONS

In this paper, a novel framework for constrained feedback control of a 3-DOF RRR SPM using convex optimization has been proposed. The innovative method to find an estimate \mathcal{P} of the feasible workspace as a polytope, based partially on an experimental procedure and partially on computational geometry tools, presents a reduction of conservativeness as compared to a previously-proposed approach. Generating \mathcal{P} typically requires a computation time of several hours, but allows one to move all the computational complexity offline, thus being able to determine a fast online solution to the constrained control problem. The proposed methods for workspace generation and closed-loop constrained control have been experimentally tested on an Agile Wrist prototype actuated by servomotors, and the expected results have been confirmed, as shown in the paper and in the multimedia attachment.

ACKNOWLEDGEMENTS

The authors would like to thank Mr. Aibek Niytkaliyev for the technical support with SPM modelling and Agile Wrist prototype manufacturing, and Dr. Zhandos Yessenbayev and Mr. Bukeikhan Omarali for their contribution to the preparation of the accompanying multimedia attachment.

APPENDIX A

BASIC CONCEPTS OF COMPUTATIONAL GEOMETRY

The needed basic definitions and concepts of computational geometry (restricted to the specific case of a 3D Euclidean space) are recalled to give a clear outline of the concepts used in Section III-B. The reader interested in a more in-depth treatment of the topic is referred to [19].

Definition 1 (convex set): A set \mathcal{S} is *convex* if the line segment between any two points of \mathcal{S} lies in \mathcal{S} , i.e., for any $\theta_1, \theta_2 \in \mathcal{S}$ and any α with $0 \leq \alpha \leq 1$, it is verified that

$$\alpha\theta_1 + (1 - \alpha)\theta_2 \in \mathcal{S}. \quad (16)$$

- [21] X. Zhang, X. Chen, J. L. Alarcon-Herrera, and Y. Fang, "3-D model-based multi-camera deployment: A recursive convex optimization approach," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 6, pp. 3157–3169, 2015.
- [22] C. Park, J. Pan, and D. Manocha, "Real-time optimization-based planning in dynamic environments using GPUs," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 4090–4097.
- [23] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel, "Motion planning with sequential convex optimization and convex collision checking," *Int. J. Robot. Res.*, vol. 33, no. 9, pp. 1251–1270, 2014.
- [24] A. Zhakatayev, M. Rubagotti, and H. A. Varol, "Closed-loop control of variable stiffness actuated robots via nonlinear model predictive control," *IEEE Access*, vol. 3, pp. 235–248, 2015.
- [25] C. Gosselin, J. Sefrioui, and M. J. Richard, "On the direct kinematics of spherical three-degree-of-freedom parallel manipulators of general architecture," *ASME J. Mech. Des.*, vol. 116, no. 2, pp. 594–598, 1994.
- [26] C. Gosselin and E. Lavoie, "On the kinematic design of spherical three-degree-of-freedom parallel manipulators," *Int. J. Robot. Res.*, vol. 12, no. 4, pp. 394–402, 1993.
- [27] K. Al Widyan, X. Q. Mab, and J. Angeles, "The robust design of parallel spherical robots," *Mech. Mach. Theory*, vol. 46, no. 3, pp. 335–343, 2011.
- [28] T. A. Hess-Coelho, "A redundant parallel spherical mechanism for robotic wrist applications," *ASME J. Mech. Des.*, vol. 129, no. 8, pp. 891–895, 2007.
- [29] S. Bai, M. R. Hansen, and T. O. Andersen, "Modelling of a special class of spherical parallel manipulators with Euler parameters," *Robotica*, vol. 27, no. 2, pp. 161–170, 2008.
- [30] I. A. Bonev, D. Chablat, and P. Wenger, "Working and assembly modes of the Agile Eye," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, Orlando, Florida, 2006, pp. 2317–2322.
- [31] X. Kong and C. Gosselin, "A formula that produces a unique solution to the forward displacement analysis of a quadratic spherical parallel manipulator: the Agile Eye," *ASME J. Mech. Robot.*, vol. 2, no. 4, pp. 1–4, 2010.
- [32] C. Gosselin and M. Gagne, "A closed-form solution for the direct kinematics of a special class of spherical three-degree-of-freedom parallel manipulators," in *Proc. Workshop on Computational Kinematics*, 1995, pp. 231–240.
- [33] S. Bai and M. R. Hansen, "Forward kinematics of spherical parallel manipulators with revolute joints," in *Proc. IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 2008, pp. 527–532.
- [34] J. B. Kuipers, *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace and Virtual Reality*. Princeton University Press, 2002.
- [35] Y. Jian and J. Angeles, "Kinematics simulation and control design of the Agile Wrist in a dual-arm robotic mechanical systems," in *Proc. Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2012, pp. 39–42.
- [36] M. Geravand, F. Flacco, and A. D. Luca, "Human-robot physical interaction and collaboration using an industrial robot with a closed control architecture," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 4000–4007.
- [37] D. Henrion and C. Louembet, "Convex inner approximations of non-convex semialgebraic sets applied to fixed-order controller design," *Int. J. Control*, vol. 85, no. 8, pp. 1083–1092, 2012.
- [38] Q. T. Dinh, W. Michiels, S. Gros, and M. Diehl, "An inner convex approximation algorithm for BMI optimization and applications in control," in *Proc. IEEE Conference on Decision and Control (CDC)*, 2012, pp. 3576–3581.
- [39] D. Simon, J. Löfberg, and T. Glad, "Nonlinear model predictive control using feedback linearization and local inner convex constraint approximations," in *Proc. European Control Conference (ECC)*, 2013, pp. 2056–2061.
- [40] M. Herceg, M. Kvasnica, C. N. Jones, and M. Morari, "Multi-Parametric Toolbox 3.0," in *Proc. European Control Conference (ECC)*, 2013, pp. 502–510.
- [41] "UM7-LT Orientation Sensor," Website. [Online]. Available: <http://www.chrobotics.com/shop/um7-lt-orientation-sensor>
- [42] J. Mattingley and S. Boyd, "CVXGEN: A code generator for embedded convex optimization," *Optimization and Engineering*, vol. 13, no. 1, pp. 1–27, 2012.
- [43] D. Avis, "LRS: A revised implementation of the reverse search vertex enumeration algorithm," in *Polytopes, Combinatorics and Computation*, ser. DMV-Seminar 29, G. Kalai and G. Ziegler, Eds. Birkhauser Verlag, 2000, pp. 177–198.
- [44] K. Fukuda and A. Prodon, "Double description method revisited," in *Combinatorics and Computer Science*, ser. Lecture Notes in Computer Science. Springer, 1996, pp. 91–111.
- [45] K. Fukuda, *cdd/cdd+ Reference Manual*, Institute for operations Research ETH-Zentrum, 1997.