# Inverse kinematics by means of convex programming: some developments.

Franco Blanchini[1], Gianfranco Fenu[2], Giulia Giordano[1], Felice Andrea Pellegrino[2]

*Abstract*— **A novel approach to the problem of inverse kinematics for redundant manipulators has been recently introduced: by considering the joints as point masses in a fictitious gravity field, and by adding proper constraints to take into account the length of the links, the kinematic inversion may be cast as a convex programming problem. Such a problem can be solved in an efficient way and may be easily modified to include constraints due to obstacles. Here we present further developments of the idea. In particular, for the case of planar robots, we show (i) how to impose hard constraints on the joint angles while preserving the convexity of the problem, and (ii) how to add constraints due to objects (for instance, a load carried by the robot) that are rigidly attached to some part of the robot.**

## I. INTRODUCTION

As is well-known, for robotic manipulators, the inverse kinematics problem consists in finding a joint configuration that corresponds to a given position and/or orientation of the end–effector. The problem arises because, usually, the task to be performed by the robot is expressed in the operational space (Cartesian space), while the robot is controlled in the configuration space (joint space). Solving the inverse kinematics problem allows the application to manipulators of planning and navigation techniques available for the configuration space (see for instance [1], [2], [3] and [4]). In [5], a novel global (i.e. not relying on the Jacobian), optimization–based, method for kinematic inversion has been proposed. Basically, the configurations of the robot correspond to those of a rope subject to gravity, of which one of the extrema (representing the end-effector) is moved. In particular, the joints are considered as point masses subject to fictitious gravity forces. Hence a minimum energy problem is formulated, where suitable constraints take into account the length of the links. Instead of the more usual joint angles, in the mentioned formulation, the decision variables are the Cartesian coordinates of each of the joints. The objective function of the optimization problem is the total potential energy of the system of masses. The major benefit of the proposed approach is that it leads (thanks to a proper relaxation) to a convex programming problem, which can be efficiently solved by means of well–known tools (see [6]).

Here, for planar robots, we present two developments of the approach described above:

(i) we show how to impose hard constraints on the joint angles while preserving the convexity, even for non zero-symmetrical angular ranges;

(ii) we show how to impose hard constraints on parts of the robots other than the links or on objects rigidly attached to the robot, while preserving the convexity.

Both of the above developments have practical relevance. In the case of redundant robots, infinite configurations lead, generally, to the desired position and orientation of the end–effector. However, since mechanical joint limits are always present, the problem arises to choose a configuration that is compliant with the constraints. In the context of Jacobian–based methods [7], the problem may be faced by minimizing, in the null–space of the Jacobian matrix, a suitable scalar objective function, which is often of the form $(q-\bar{q})^\top W(q-\bar{q})$. The idea is that of keeping the joint variables as close as possible to the center $\bar{q}$ of their admissible range. The method (see for instance [8] and the survey [9]) is iterative and local by nature and, depending on the initial configuration, may fail to provide a solution that is compliant with the constraints even if it exists. Other objective functions may be used for enforcing satisfaction of constraints of type (ii), for instance for obstacle avoidance purposes [10], but suffering from the same limitations described above. Unlike local methods, global methods explore the configuration space and try to find a minimizer for the position and/or orientation error with respect to the prescribed one (see for instance [11], [12] and the more recent [13], [14]). Mechanical joint limits and constraints of type (ii) may be included in the resulting nonlinear programming problem, in the form of constraints on the decision variables. However, due to the highly nonlinear relationship between joint space variables and operational space variables, solving it requires heuristics in order not to get stuck in local minima and, generally, the computational burden is high.

## II. INVERSE KINEMATICS AS A CONVEX PROGRAMMING PROBLEM

We briefly remind the formulation of the problem, as presented in [5]. Consider a redundant planar manipulator as in Fig. 1, composed of several links connected by revolute joints (also called nodes in the following). Let the position of the end–effector $(x_E, y_E) = (x_4, y_4)$ be assigned, as well as the link lengths $r_i$. The inverse kinematics problem consists in determining suitable angles $q_i$ which assure the desired position. In order to formulate the inverse kinematics as a convex optimization problem, it is convenient to describe the configuration of the robot by the $n$–tuple $(x_i, y_i)$, $i = 1, \ldots, n$, instead of the more common $q_i$, $i = 1, \ldots, n$. In the following, a configuration $(x_i, y_i)$, $i = 1, \ldots, n$, that is compatible with the robot kinematics will be called *admissible*; a configuration that is compatible with the constraints of

*corresponding author, fapellegrino@units.it

[1]Dipartimento di Matematica e Informatica, Università degli Studi di Udine, Via delle Scienze 206, 33100 Udine, Italy.

[2]Dipartimento di Ingegneria e Architettura, Università degli Studi di Trieste, Via Alfonso Valerio 10, 34127 Trieste, Italy.
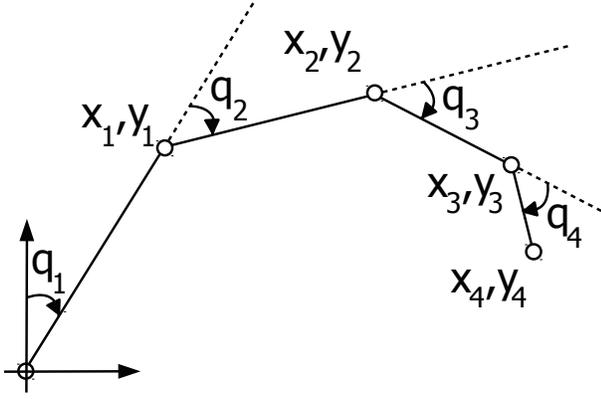
Fig. 1: The inverse kinematics problem.
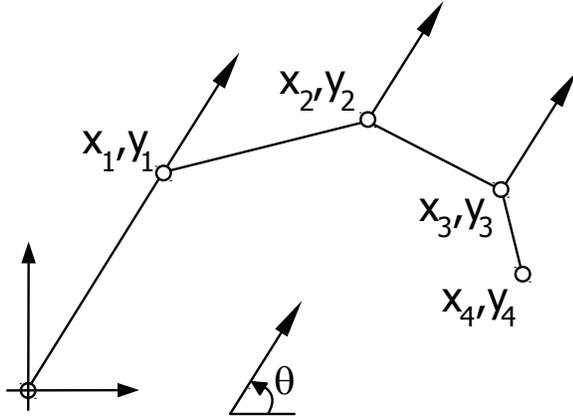


Fig. 2: The catenary kinematics problem.

the optimization problem (but not necessarily with the robot kinematics) will be called *feasible*. Now let us introduce a "unit gravity force" $F_G = (\cos(\theta), \sin(\theta))$, as in Fig. 2, that affects unitary fictitious masses placed at the joints. Such a force can be arbitrarily (*i.e.*, not necessarily vertically) directed.

Denoting by $(x_i, y_i)$ the joint positions, and taking into account the gravity, the following optimization problem arises:

$$\min_{x_i, y_i} \quad \sum_{i=1}^{n} [\cos(\theta) x_i + \sin(\theta) y_i]$$
$$s.t \quad (x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 = r_i^2, \quad i = 1, \dots, n$$
$$(x_n, y_n) \quad \text{assigned}$$

where $\theta$ is a fixed parameter, representing the gravity direction, and we aim at minimizing potential energy. The above problem is not convex, but it can be convexified by replacing the equality constraints with inequality constraints:

$$\min_{x_i, y_i} \quad \sum_{i=1}^{n} [\cos(\theta) x_i + \sin(\theta) y_i] \tag{1}$$
$$s.t \quad (x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 \le r_i^2, \quad i = 1, \dots, n \tag{2}$$
$$(x_n, y_n) \quad \text{assigned} \tag{3}$$

This formulation physically corresponds to the case of unitary masses subject to gravity and connected not by rigid arms, but by strings which can be not completely stretched. While the problem with equality constraints is strictly equivalent to the inverse kinematics problem, the "relaxed" problem is not: a solution of (1)–(3) corresponds to an admissible configuration if and only if the inequality constraints are active, namely *satisfied as equalities*, at the optimum.

The optimization problem (1)–(3) has a simple formulation, convex obstacles and boundaries can be easily considered without affecting convexity and, despite redundancy, the solution is unique (as suggested by the masses-and-strings analogy). The degrees of freedom can be fruitfully exploited in the actuation by choosing the fictitious gravity force $F_G$, which can be arbitrarily oriented in order to shape the robot chain and cope with environmental constraints: different orientations of $F_G$ produce different configurations with the same end–effector position. Of course, due to the relaxation, it is not assured that, for a given end–effector position, the optimization problem provides an admissible solution (i.e. compatible with the robot kinematics). This potential drawback is analyzed in the detail in [5], where a node-specific gravity field is employed to avoid non admissible solutions.

By posing $p_i = [x_i, y_i]^\top$, and $p = [p_1^\top, \dots, p_n^\top]^\top$, the problem may be written in the following more general form:

$$\min \quad F_G^\top p \tag{4}$$
$$s.t. \quad \|p_i - p_{i-1}\| \le r_i, \quad i = 1, 2, \dots, n \tag{5}$$
$$p_n \quad \text{assigned} \tag{6}$$
$$g^\top p \le h \tag{7}$$
$$l^\top p = k \tag{8}$$

for properly chosen $F_G \in \mathbb{R}^{2n}$, $g \in \mathbb{R}^{m \times 2n}$, $h \in \mathbb{R}^m$, $l \in \mathbb{R}^{s \times 2n}$, $k \in \mathbb{R}^s$ and where the equalities and inequalities involving vectors have to be intended component–wise. Here, $m$ is the number of inequality constraints and $s$ the number of equality constraints. Inequality constraints (7) are due to the environment (ceiling, floor, walls) and may be global (*i.e.*, acting on all nodes) or local (*i.e.*, acting on a subset of nodes). Equality constraints (8) may for instance take into account the attitude of the end–effector. Needless to say, getting the joint angles $q_i$ from an admissible solution of problem (4)-(8) is a matter of straightforward trigonometric calculations.

## III. DEALING WITH ANGULAR CONSTRAINTS

In practical situations, joint angles are subject to constraints, due for instance to mechanical reasons. Suppose that a joint angle $q_i$ is such that

$$q_i^- \le q_i \le q_i^+, \quad q_i^- = -q_i^+. \tag{9}$$

We will refer to this situation as zero-symmetric admissible range. The constraint (9) can be included in the optimization

problem by observing that any constraint of the form:

$$F(M_1 p_1 + M_2 p_2 + \cdots + M_n p_n) \leq \gamma, \tag{10}$$

where $F : \mathbb{R}^2 \to \mathbb{R}$ is convex, $\gamma$ is a constant coefficient and $M_1, M_2, \ldots, M_n$ are constant $2 \times 2$ matrices, will preserve the convexity of the problem.

Consider Fig. 3, representing two subsequent links $i$ and $i+1$, of length $r_i$ and $r_{i+1}$, and suppose that $q_i^- = -\alpha$ and $q_i^+ = \alpha$. We denote by $O$ the point lying on the bisector of the angular range, and whose distance from $p_i$ is $r_{i+1}$. It is easy to see that, by construction, the constraints (9) will be satisfied if and only if the distance from $p_{i+1}$ to the point $O$ is less or equal to the length $\overline{OC}$ of segment $OC$. As a consequence, the constraint may be expressed as:

$$\left\| p_{i+1} - \left[ p_i + (p_i - p_{i-1}) \frac{r_{i+1}}{r_i} \right] \right\| \leq 2 r_{i+1} \sin \frac{\alpha}{2}. \tag{11}$$

Clearly, inequality (11) is of type (10), hence it can be added to the optimization problem while preserving its convexity. The case of non zero-symmetric admissible range, in which

$$q_i^- \leq q_i \leq q_i^+, \qquad q_i^- \neq -q_i^+, \tag{12}$$

may be treated in a similar way. First we denote by $\alpha$ the angle such that $2\alpha = q_i^+ - q_i^-$. Then we observe that the point $O$, that lies on the bisector of the admissible angular range, can still be expressed as a linear combination of decision variables, by taking a proper rotation matrix $\Omega_\phi$. Referring to Fig. 4, the point $O$ may be obtained by adding to $p_i$ the vector $(p_i - p_{i-1}) \frac{r_{i+1}}{r_i}$ *rotated by an angle $\phi$*. By defining

$$\Omega_\phi = \left[ \begin{array}{cc} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{array} \right],$$

the constraint for non zero–symmetric angular range may thus be expressed as:

$$\left\| p_{i+1} - \left[ p_i + \Omega_\phi (p_i - p_{i-1}) \frac{r_{i+1}}{r_i} \right] \right\| \leq 2 r_{i+1} \sin \frac{\alpha}{2}. \tag{13}$$

Note that $\phi$ depends only on the structure of the robot (i.e. the length of the links and the admissible angular range) and not on the particular configuration. Hence $\Omega_\phi$ is constant, rendering the constraints of type (10) and thus preserving the convexity of the problem.

## IV. Rigid transformations constraints

Consider a point $v$ whose position with respect to link $i+1$ is fixed. The point $v$ may belong to the robot itself or to an object (for instance a load that is being carried by the robot) rigidly connected to the link. The point may be expressed as a linear combination of the components of $p_i$ and $p_{i+1}$, having the same coefficients independent of the actual configuration of the robot. For instance, referring to Fig. 5, point $v$ may be expressed as

$$v = p_i + \lambda \Omega_\phi (p_{i+1} - p_i)$$

where $\lambda$ is the (constant) ratio between $||v - p_i||$ and $r_{i+1}$. Being $v$ a convex function of $p_i$ and $p_{i+1}$, inequality con-
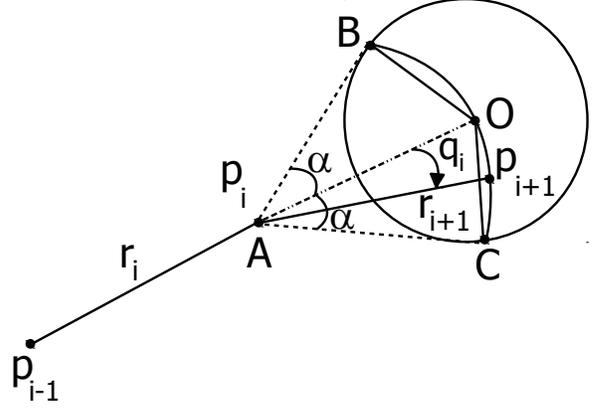


Fig. 3: The case of zero–symmetric admissible angular range. The dashed lines $AB$ and $AC$ delimit the admissible angular range.
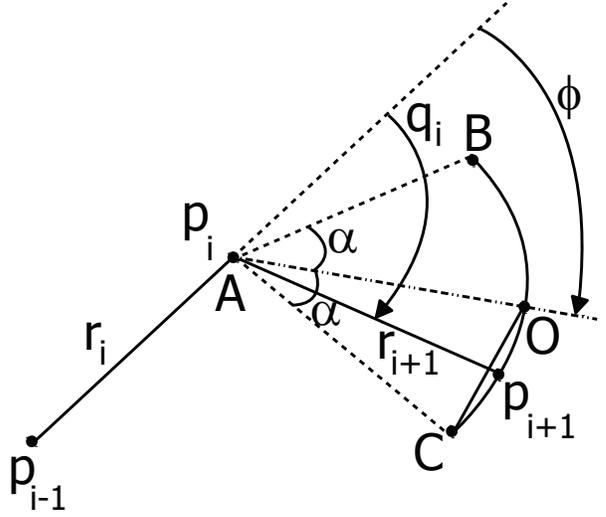


Fig. 4: The case of non zero–symmetric admissible angular range. The dashed lines $AB$ and $AC$ delimit the admissible angular range.
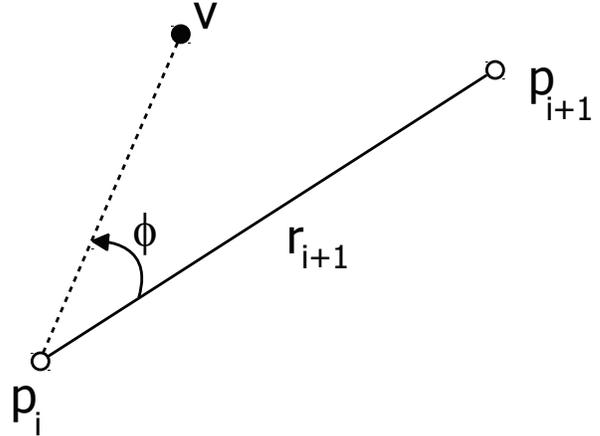


Fig. 5: A point $v$ rigidly attached to the link $i+1$.

straints such as

$$F(M_0v + M_1p_1 + M_2p_2 + \cdots + M_np_n) \leq \gamma, \qquad (14)$$

where $F : \mathbb{R}^2 \rightarrow \mathbb{R}$ is convex, $\gamma$ is a constant coefficient and $M_0, M_1, \ldots, M_n$ are constant $2 \times 2$ matrices, may be added to the problem without losing convexity. Even equality constraints of the form

$$M_0v + M_1p_1 + M_2p_2 + \cdots + M_np_n = \gamma \qquad (15)$$

will preserve convexity. The above facts may be exploited in various ways, for instance:

- avoiding collision of objects rigidly attached to the robot with the environment (see the simulation reported in Subsection V-B);
- formulating inverse kinematics problems where, instead of $p_n$ assigned, one has $v$ assigned;
- formulating "proximity problems" where, instead of assigning $p_n$ one specifies that $||v - \bar{v}|| \leq \varepsilon$, meaning that the distance between some point(s) belonging to the robot and some other point(s) must be below a given threshold, while satisfying the other constraints.

## V. SIMULATION RESULTS

### A. Angular constraints

We report an example obtained by applying the proposed technique to a planar robot of 7 links, having equal length $r_i = 10$, $i = 1, \ldots, 7$. In Figs. 6 and 7, stroboscopic views of two trajectories are reported. In both cases the gravity vector is $(\cos\theta, \sin\theta)$ with $\theta = \dfrac{5\pi}{8}$, kept constant during the whole simulation and chosen to obtain a concavity of the robot directed towards the fourth quadrant. The initial and final configurations of the robot are shown in black and blue, respectively. The red line represents a constraint due to the environment (all the nodes are constrained to stay on the half–plane delimited by the line and containing the origin). In both cases the configurations belonging to the trajectory are obtained by solving a convex optimization problem for each of the desired positions of the end-effector, from the top one $(38, 33)$ to the bottom one $(38, 9)$. In Fig. 6, joint angles are not constrained. On the contrary, the trajectory of Fig. 7 is obtained by requiring that

$$-\frac{1}{10} \leq q_4 \leq \frac{1}{10}$$

and

$$-\frac{1}{10} \leq q_5 \leq \frac{1}{10},$$

meaning that the angles between the links 3 and 4, and between links 4 and 5 are less than $\approx 6°$. This is achieved by adding the following constraints of type (11):

$$||p_5 - [p_4 + (p_4 - p_3)]|| \leq 20\sin\frac{1}{20}, \qquad (16)$$

$$||p_6 - [p_5 + (p_5 - p_4)]|| \leq 20\sin\frac{1}{20}. \qquad (17)$$
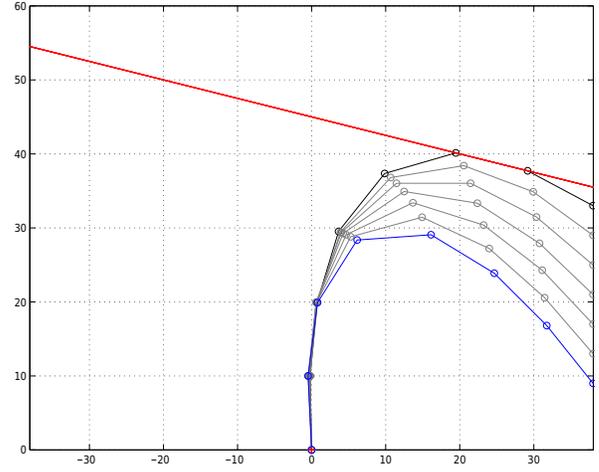


Fig. 6: A stroboscopic view of the trajectory of a planar robot. Joint angles are not constrained. The red line represents a fixed linear constraint.
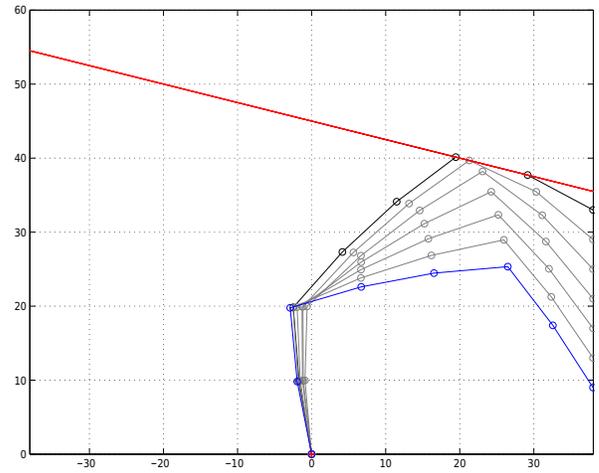


Fig. 7: A stroboscopic view of the trajectory of a planar robot. Two joint angles are constrained to be not greater than $\frac{1}{10}$ rad. The red line represents a fixed linear constraint.
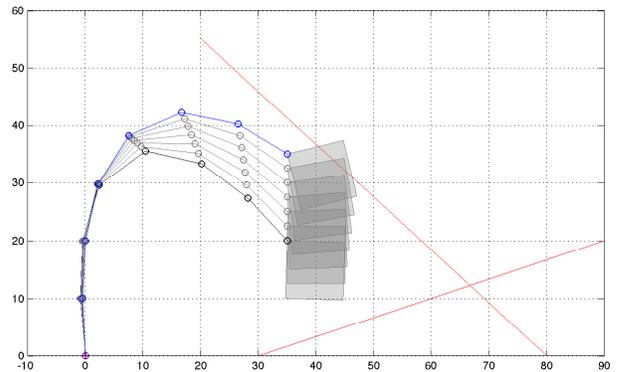


Fig. 8: A stroboscopic view of the trajectory of a planar robot. The square represents a grasped object and the red lines represent fixed linear constraints. A collision occurs between the object and the environment when moving the end–effector from bottom to top, because the shape of the object is not taken into account.
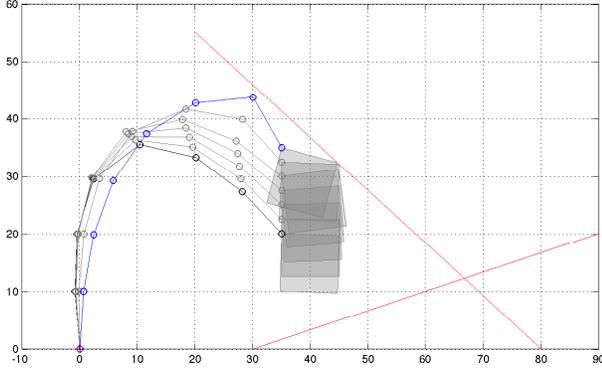
Fig. 9: A stroboscopic view of the trajectory of a planar robot. The square represents a grasped object and the red lines represent fixed linear constraints. As opposite to the case reported in Fig. 8, here the shape of the grasped object is taken into account by properly formulated rigid constraints. As a result, no collision occurs as the end–effector moves from bottom to top.



Fig. 10: A redundant *quasi-planar* robot having 8 revolute joints.

### B. Other rigid constraints

Rigid constraints allow for taking into account the shape of objects rigidly attached to the robot, for instance for avoiding collisions. As an example, consider the same planar robot of Subsection V-A, carrying a square–shaped object of size $l = 10$. The position of the object relative to the end–effector is assumed to be fixed, as is the case for grasped objects. The initial and final configurations of the robot are shown in black and blue, respectively. In Fig. 8 an undesirable trajectory, obtained by neglecting the grasped object, is shown: the end–effector moves from bottom to top and, at some time, a collision occurs between the object and the environment. On the contrary, in Fig. 9, *for the same positions of the end–effector*, no collision occurs, because the shape of the object is taken into account in the optimization problem. Precisely, three vertices $v_1$, $v_2$, $v_3$ of the square (the other coincides with the end–effector and may be neglected) can be expressed as:

$$v_i = p_n + \lambda_i \Omega_{\phi_i} (p_n - p_{n-1}),$$

for $\lambda_1 = 1$, $\lambda_2 = \sqrt{2}$, $\lambda_3 = 1$ and $\phi_1 = \frac{\pi}{4}$, $\phi_2 = 0$, $\phi_3 = -\frac{\pi}{4}$. Then, by adding to the optimization problem constraints of the form

$$g_j^\top v_i \le h_j, \quad j = 1, 2, \quad i = 1, 2, 3,$$

for properly chosen $g_1$, $g_2$, $h_1$, $h_2$, the three vertices are forced to belong to the admissible workspace (delimited by the two red lines). Note that the $v_i$ are not decision variables, but depend linearly on the decision variables $p_1, \ldots, p_n$.

## VI. CONCLUDING DISCUSSION

We have shown how to exploit the properties of rigid transformations for taking into account various types of constraints in the kinematic inversion of redundant planar manipulators based on convex optimization [5]. The method described in the present paper is not suitable for generic non–planar manipulators. The reason is the following: for non–planar manipulators a rigid transformation implies a rotation around an axis that, in general, depends on the current configuration (on the contrary, for planar manipulators the axis is always the same, orthogonal to the plane of the robot). As a consequence, convexity with respect to the decision variables is lost. The method, however, may be applied whenever the kinematic inversion problem, for a given position of the end-effector, can be reduced to a planar one. An example is given in Fig. 10: for a given position of the end-effector, the angle of the first revolute joint (vertical axis) is defined, hence a planar problem[1] can be formulated. The possible adaptation of the method to other special classes of non–planar manipulators is currently under investigation.

### REFERENCES

[1] S. M. Lavalle, *Planning algorithms*. Cambridge University Press, 2006.

[2] E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 501–518, 1992.

[3] F. Blanchini, F. A. Pellegrino, and L. Visentini, "Control of manipulators in a constrained workspace by means of linked invariant sets," *International Journal of Robust and Nonlinear Control*, vol. 14, no. 1314, pp. 1185–1205, Sept. 2004.

[4] F. Blanchini, S. Miani, F. A. Pellegrino, and B. Van Arkel, "Enhancing Controller Performance for Robot Positioning in a Constrained Environment," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 5, pp. 1066–1074, 2008.

[5] F. Blanchini, G. Fenu, G. Giordano, and F. A. Pellegrino, "A convex programming approach to the inverse kinematics problem for manipulators under constraints," Automatica [Accepted provisionally as Regular Paper]. [Online]. Available: http://control.units.it/en/catenary

[6] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

[7] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Springer, 2009.

---

[1]in the vertical plane defined by the end-effector position and the first joint axis.

[8] A. Liégeios, "Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 7, no. 12, pp. 868–871, 1977.

[9] B. Siciliano, "Kinematic control of redundant robot manipulators: A tutorial," *Journal of Intelligent and Robotic Systems*, vol. 3, no. 3, pp. 201–212, 1990.

[10] A. A. Maciejewski and C. A. Klein, "Obstacle Avoidance for Kinematically Redundant Manipulators in Dynamically Varying Environments," *The International Journal of Robotics Research*, vol. 4, no. 3, pp. 109–117, Sept. 1985.

[11] L. C. T. Wang and C. C. Chen, "A combined optimization method for solving the inverse kinematics problems of mechanical manipulators," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 4, pp. 489–499, 1991.

[12] J. Zhao and N. I. Badler, "Inverse kinematics positioning using nonlinear programming for highly articulated figures," *ACM Transactions on Graphics*, vol. 13, no. 4, pp. 313–336, Oct. 1994.

[13] J.-X. Xu, W. Wang, and Y. Sun, "Two optimization algorithms for solving robotics inverse kinematics with redundancy," *Journal of Control Theory and Applications*, vol. 8, no. 2, pp. 166–175, 2010.

[14] K. Kinoshita, K. Murakami, and M. Isshiki, "Solution of Inverse Kinematics by PSO Based on Split and Merge of Particles," in *Artificial Intelligence and Soft Computing*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2013, vol. 7894, pp. 108–117.