A convex programming approach to the inverse kinematics problem for manipulators under constraints

Franco Blanchini[a], Gianfranco Fenu[b], Giulia Giordano[c], Felice Andrea Pellegrino[b]

[a]*Dipartimento di Matematica, Informatica e Fisica, Università degli Studi di Udine, Via delle Scienze, 206 33100 Udine, Italy,* franco.blanchini@uniud.it
[b]*Dipartimento di Ingegneria e Architettura, Università degli Studi di Trieste, Via A. Valerio, 10 34127 Trieste, Italy,* fenu@units.it, fapellegrino@units.it
[c]*LCCC Linnaeus Center and Department of Automatic Control, Lund University, Ole Römers väg, 1 SE 223 63 Lund, Sweden,* giulia.giordano@control.lth.se

Abstract

We propose a novel approach to the problem of inverse kinematics for possibly redundant planar manipulators. We show that, 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. Convex constraints in the decision variables (in particular, linear constraints in the workspace) are easily managed with the proposed approach. We also show how to exploit the idea for avoiding obstacles while tracking a reference end-effector trajectory and discuss how to extend the results to some kinds of non-planar manipulators. Simulation results are reported, showing the effectiveness of the approach.

Keywords: robotic manipulators; inverse kinematics; convex programming; constraints.

1. Introduction and Motivation

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, 4]). Closed-form solutions exist only for manipulators having a simple kinematic structure (see [5]). In the other cases, for example when the manipulator is redundant ([6]), it is necessary to resort to numerical methods. Numerical approaches to kinematic inversion may be roughly divided in two categories. A first one is based on differential kinematics and comprises Jacobian-based methods (the Jacobian matrix represents a linear, configuration dependent, map between the joint velocity space and the operational velocity space). Various methods based on differential kinematics have been proposed: Jacobian pseudoinverse ([7]), Jacobian transpose [8, 9], damped least-squares ([10, 11]) and other variations (see also

[12] and the references therein). Iterations are necessary due to the linearized nature of the approach: in all the mentioned methods, the solution results from a process that starts from an assigned configuration and iteratively computes a sequence of kinematically feasible configurations that eventually converges to the desired one. A second category is that of global methods, which explore the whole configuration space and try to find a minimizer for the position and/or orientation error with respect to the prescribed one (see for instance [13, 14] and the more recent [15, 16]). The task is difficult because of the highly nonlinear relationship between joint space variables and operational space variables. These approaches lead to a non-convex nonlinear programming problem, need to resort to heuristics in order not to get stuck in local minima and are, in general, computationally expensive.

In the present paper, we propose a novel optimization-based method for kinematic inversion. The main features of the proposed approach can be summarized as follows.

- The method is global, in the sense that it takes into account all the admissible poses.
- The joints are considered as point masses subject to fictitious gravity forces. Suitable constraints take into account the length of the links.
- The inverse kinematics problem is formulated as a minimization problem whose objective function is the total potential energy of the system of masses.
- The objective function provides a criterion to fruitfully exploit redundancy by selecting the unique minimizing configuration.
- The approach leads to a *convex programming problem*, which can be efficiently solved by means of well-known tools (this is a significant advantage over the existing global methods).
- In some circumstances, the feasible solution of the optimization problem may result non admissible from a kinematic point of view. In these cases, we show how to properly modify the fictitious gravity field so as to achieve physical admissibility.
- Additional convex constraints in the operational space are easily taken into account to avoid collisions.
- The main limitation of the method is that it is valid, in general, for planar manipulators only. However, it can be employed for some particular, but practically important, non-planar manipulators that we characterize in terms of their kinematic structure.

Rather than providing continuity theorems, which would be valid case by case, we support our idea by the following physical intuition: the movements of the robot correspond to those of a *rope subject to gravity*, of which we are moving one of the extrema. This creates smooth transitions, as it can be certified by experiments.

It is worth mentioning that a virtual gravity approach has already been proposed for legged robots by [17], where the fictitious gravity field is employed to enforce a

motion direction that is parallel to the ground, without an explicit design of the gait. Our use of fictitious gravity is different, since it is instrumental to characterize the desired configuration as a minimizer of the potential energy. Our approach can be seen as a fast method for finding a robot configuration that satisfies a prescribed end-effector pose and, possibly, convex constraints. For this reason it is not necessarily alternative to classical navigation approaches, but can be successfully combined with other methods, *e.g.*, those proposed by [18, 19], based on potential functions. Many developments of the basic idea are possible: some of them have already been described in [20].

The paper is organized as follows: in Section 2 we state and solve the problem for planar manipulators and provide some extensions to non-planar cases. Section 3 shows how to include the obstacles in the problem formulation. Various simulation results are presented in Section 4, for both planar and non-planar cases. Finally, a concluding discussion is reported in Section 5.

2. The inverse kinematics problem: the planar case and some extensions

Consider a redundant planar manipulator as represented 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 that provide the desired position; in the case of motion planning, smooth end-effector and joint trajectories need to be obtained. The problem is well known to be difficult for the following reasons:

- the involved equations include trigonometric terms;
- in the presence of obstacles or boundaries, the "forbidden" region in the configuration space may be hard to describe;
- it is not always clear how to cope with redundancy.

The main idea of this paper is to adopt an inversion method based on convex optimization, which can provide a solution that efficiently deals with the above issues. 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 we call

- **admissible**: a configuration $(x_i, y_i),\ i = 1, \ldots, n$, that is compatible with the robot kinematics;
- **feasible**: a configuration that is compatible with the constraints of the optimization problem.

The typical solution to the inverse problem, for instance in the case of manipulators as in Fig. 1, considers directly the following equations:

$$x_n = \sum_{i=1}^{n} r_i \cos(q_i), \qquad y_n = \sum_{i=1}^{n} r_i \sin(q_i),$$

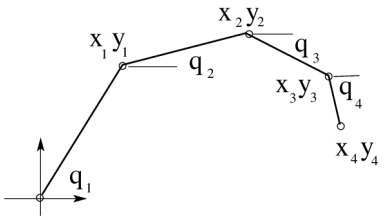


Figure 1: The inverse kinematics problem.

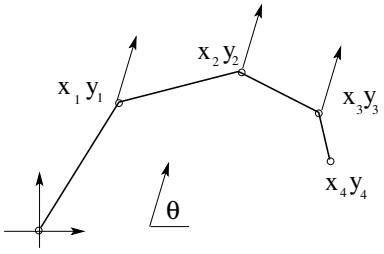


Figure 2: The catenary kinematics problem.

which have to be solved in the free variables q_i. Now let us imagine the same problem solved for a catenary in which a "unit gravity force" $F_G = (\cos(\theta), \sin(\theta))$ is assigned, as in Fig. 2, and 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:

$$\begin{aligned} \min_{x_i,y_i} \quad & \sum_{i=1}^{n} [\cos(\theta)x_i + \sin(\theta)y_i] \\ s.t. \quad & (x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 = r_i^2, \quad i = 1,2,\ldots,n \\ & (x_0, y_0) = (0,0) \\ & (x_n, y_n) \ \text{ assigned} \end{aligned}$$

where θ 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:

$$\min_{x_i,y_i} \quad \sum_{i=1}^{n} [\cos(\theta)x_i + \sin(\theta)y_i] \tag{1}$$

$$s.t. \quad (x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 \leq r_i^2, \quad i = 1,2,\ldots,n \tag{2}$$

$$(x_0, y_0) = (0,0) \tag{3}$$

$$(x_n, y_n) \ \text{ assigned} \tag{4}$$

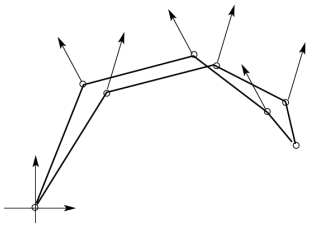


Figure 3: Different configurations.

This relaxation 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. We remind that a constraint is

- **active** if it is satisfied as an equality;
- **inactive** if it is satisfied as a strict inequality.

As we will see, a solution of (1)–(4) corresponds to an admissible configuration if and only if the inequality constraints are active at the optimum.

The optimization problem (1)–(4) has a simple formulation, convex obstacles and boundaries can be easily considered without affecting convexity and, despite redundancy, the solution is unique. 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 (as in Fig. 3).

Although the optimization problem involved can be solved very efficiently, it is not ensured that, for a given end-effector position, the optimization problem provides an admissible solution. On the one hand, an admissible configuration may not exist at all, *e.g.*, due to the distance of the end-effector from the reference origin $(x_0, y_0) = (0, 0)$. Obviously this distance has to be smaller or equal to the sum of the link lengths:

$$\sqrt{x_n^2 + y_n^2} \leq \sum_{i=1}^{n} r_i = \rho_{\max}.$$

Otherwise, no admissible configuration exists which guarantees the desired end-effector position. On the other hand, except for particular cases, there exists also a minimum distance under which admissible configurations exist, but are not produced by any selection of a common gravity force. This is the case of Fig. 4 (left), where the admissible configuration (solid line) is different from the one achieved by the optimization problem (dashed line). The solutions provided by the optimization problem, in fact, physically correspond to solutions that would be achievable if the nodes were connected by *strings* rather than rigid arms. The situation can be fixed if we solve the same problem by adopting different gravitational fields for each node. This boils down to solving the optimization problem with the same constraints, but assigning different angles θ_i in the

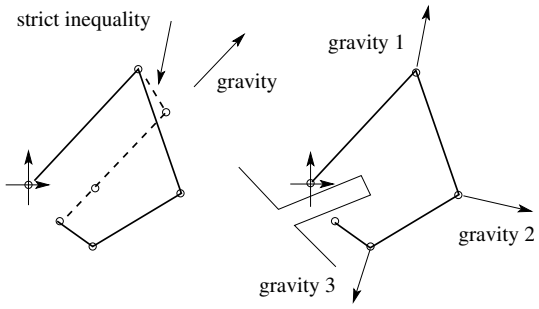


Figure 4: Left: an admissible configuration (solid line) not recovered by the solution of the optimization problem with a single gravity (dashed line). Right: the same configuration recovered with different gravity forces.

objective function, namely replacing (1) with the (linear) objective:

$$\min_{x_i,y_i} \quad \sum_{i=1}^{n} \left[\cos(\theta_i)x_i + \sin(\theta_i)y_i\right].$$

Remark 1. *The functional with a node-specific gravity can be exploited in order to "shape" the robot,* e.g.*, to surround obstacles (see Fig. 4, right). The problem of associating a suitable gravity field with each node will be considered later.*

2.1. Formalization of the problem

In the planar case, the problem may be formulated in the following general form:

$$\min \quad c^\top x + d^\top y \tag{5}$$
$$s.t. \quad (x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 \leq r_i^2, \quad i = 1,2,\ldots,n \tag{6}$$
$$(x_0, y_0) = (0,0) \tag{7}$$
$$(x_n, y_n) \text{ assigned} \tag{8}$$
$$Mx + Ny \leq q \tag{9}$$
$$Qx + Ry = s \tag{10}$$

where x and y are the vectors of the node coordinates, M, N, Q and R are assigned constraint matrices, while q and s are assigned constraint vectors. The weight vectors c and d define the linear cost and can be seen as a possibly node-specific gravitational field. A possible choice of c and d is associated with the gravity potential encountered before: $c^\top = \cos(\theta)[1\ 1\ \ldots 1]$ and $d^\top = \sin(\theta)[1\ 1\ \ldots 1]$. Equality constraints (10) are introduced, *e.g.*, if a cart which carries the robot is constrained on a track. Fixing the attitude of the end-effector also corresponds to a linear equality constraint. Inequality constraints (9) for the points (x_i, y_i) are due to the environment (ceiling, floor, walls) and may take into account global (*i.e.*, acting on all nodes) or local (*i.e.*, acting on a subset of nodes) constraints.

In the configuration space q, the constraints would be non-convex and hard to describe. On the contrary, (5)–(10) is convex. The only problem is that a feasible solution does not necessarily correspond to a physically admissible configuration. A feasible solution is physically admissible only if the inequality constraints are active at the optimal solution.

Observation 1. *A feasible solution of problem (5)–(10) corresponds to a physically admissible configuration if and only if the inequality constraints (6) are active at the optimum.*

Since our aim is to use convex optimization to solve the inverse kinematics problem, we need to handle the case in which a feasible solution is found with some non-equality (strict inequality) constraints. The first preliminary result states that any physically realizable (admissible) configuration is the solution of the convex optimization provided that the functionals c and d are properly chosen.

Proposition 1. *Let $\tilde{x} = [\tilde{x}_0\ \tilde{x}_1 \dots \tilde{x}_n]^\top$ and $\tilde{y} = [\tilde{y}_0\ \tilde{y}_1 \dots \tilde{y}_n]^\top$ be vectors that satisfy the constraints (6)–(10), with (6) satisfied as equality. Then there exist vectors c and d in (5) such that the optimal solution of the problem (5)–(10) is equal to $(\tilde{x}, \tilde{y})$.*

Proof Since the constraints (6) are active, $\tilde{x}$ and $\tilde{y}$ are on the boundary of the feasibility domain (6)–(10), which is a compact and convex set. Then for each boundary point a linear functional exists, corresponding to a choice of c and d in (5), for which such a point is the optimal solution of (5)–(10). ■

In practical implementations it is often necessary to determine a trajectory, rather than a single configuration. This problem can be efficiently solved by considering the initial and final end-effector positions and a finite number of intermediate positions. For each of the considered positions, the convex optimization problem is solved; all the obtained configurations are eventually interpolated using regular spline functions in the angle space, which ensure the desired smoothness of the trajectory[1]. Note that the convex optimization problems are independent of one another and each provides a unique solution. In the presence of redundancy, however, abrupt changes of configuration from one solution to the subsequent are avoided, in view of the rope analogy.

2.2. Guaranteed admissibility via hierarchical optimization

We have seen that in some cases, when the end-effector is too close to the reference origin, no admissible configuration can be found by solving the optimization problem with a common gravity force applied to all nodes. For instance, Fig. 5, right panel, shows that the third link constraint is inactive (*i.e.*, satisfied as a strict inequality), leading to a configuration which is not physically admissible, because the actual link length is as in the left panel (admissible configuration).

To achieve admissibility, we can consider a functional $\alpha f_c + \beta f_a$, with coefficients $\alpha > 0$ and $\beta \geq 0$. The *common* gravity functional f_c is chosen as in (1), with a suitable

[1] As an alternative solution, a dynamic tracking problem can be solved using the computed optimized configuration as a reference.

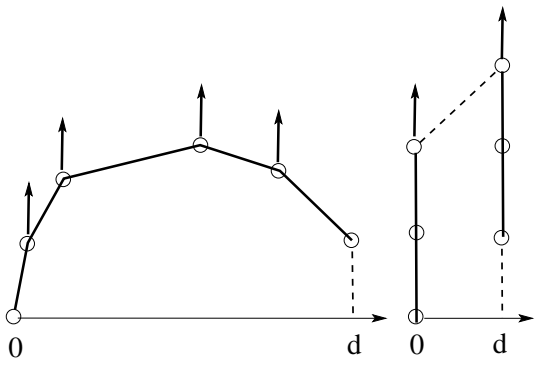


Figure 5: Admissible (left) and non-admissible (right) solutions

choice of θ that provides the desired robot orientation, and is applied as a first option. The node-specific *auxiliary* functional f_a is introduced, if necessary, to guarantee that an admissible configuration can be achieved, *i.e.*, that all the length constraints are satisfied as equalities. More in general, the coefficients α and β can be tuned to privilege f_c or f_a, depending on the situation. A pseudocode procedure will be suggested later.

In this section we provide a guaranteed method to ensure that, possibly after the introduction of a proper node-specific gravitational functional, the solution of the optimization problem always corresponds to an admissible robot configuration. We first show that, if we apply a common gravity field to all nodes, corresponding to a choice of the angle θ in (1), then, in the case of feasible but non-admissible solution, the problem concerns at most one link. To this end, a preliminary lemma is in order.

Lemma 1. *The optimal solution of the problem*

$$\min_{x_i,y_i} \quad \sum_{i=1}^{n} [\cos(\theta)x_i + \sin(\theta)y_i] \tag{11}$$

$$s.t. \quad (x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 \leq r_i^2, \quad i = 1,2,\ldots,n \tag{12}$$

$$(x_0, y_0) = (0,0) \tag{13}$$

$$(x_n, y_n) \ \textit{free}, \tag{14}$$

where the n-th node position is free, is such that the inequality constraints are all active, and each link is aligned with the gravity vector.

Proof Consider the contribution $\cos(\theta)x_1 + \sin(\theta)y_1$ to the objective function. Clearly, such a quantity is minimized, under the constraint $(x_1 - x_0)^2 + (y_1 - y_0)^2 \leq r_1^2$, when the vector (x_1, y_1) is directed as the antigradient $(-\cos(\theta), -\sin(\theta))$ and takes its maximum allowed length, which is r_1. For such a choice, call it $(\bar{x}_1, \bar{y}_1)$, the first link is aligned with the gravity vector and the first inequality constraint is active. Now consider the contribution $\cos(\theta)x_2 + \sin(\theta)y_2$ to the objective function. Such a contribution is minimized when the vector (x_2, y_2) is directed as the antigradient $(-\cos(\theta), -\sin(\theta))$ and takes its maximum allowed length. Given the constraints (12), for $i = 1,2$, the maximum allowed length is $r_1 + r_2$ that is indeed achieved when

$(x_1, y_1) = (\bar{x}_1, \bar{y}_1)$ and $(x_2 - x_1, y_2 - y_1)$ is directed as $(-\cos(\theta), -\sin(\theta))$, leading to the second link being aligned with the gravity vector and the second inequality constraint being active. By repeating the reasoning for the subsequent links, the lemma is proved.

Proposition 2. *Consider a planar robot with n sequential links and the associated optimization problem (1)–(4). Then at most one constraint in (2) may result inactive at the optimum.*

Proof Assume that the optimal solution has the *i*th constraint in (2) inactive (*i.e.*, satisfied as a strict inequality). Since we are dealing with a convex optimization problem, the solution is not changed if the *i*th constraint is removed from (2). Being the functional linear, removing such a constraint splits the optimization problem in two independent convex problems in the variables $(x_0, y_0) \dots (x_{i-1}, y_{i-1})$ and $(x_i, y_i) \dots (x_n, y_n)$. This is equivalent to considering two separate arms in which the first node (x_0, y_0) and the last one (x_n, y_n) respectively have assigned position[2]. In view of Lemma 1, the optimal solutions correspond to these two branches both directed as the gravity vector, with all remaining constraints satisfied as equalities. ■

Another property can be immediately seen with the support of Fig. 5 and formally stated in the following proposition.

Proposition 3. *Consider a reference frame having the origin in the point* $(x_0, y_0) = (0, 0)$ *and the x axis orthogonal to the direction of the gravity, which is common to all nodes; namely, the functional* (1) *is adopted with* $\theta = \pi/2$. *Let* $d = |x_n|$ *be the distance from the origin of the projection of the extremal point* (x_n, y_n) *onto the x axis. If d is greater than the largest link, then there are no inactive constraints (*i.e.*, constraints satisfied as strict inequalities).*

Proof Assume by contradiction that, at the optimum, the *i*th constraint is satisfied as a strict inequality: $(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 < r_i^2$. As in the proof of Proposition 2, since the optimization problem is convex, then the solution is unchanged if we remove this constraint from (6). Again the solution corresponds to the case of two separate chains whose extrema (x_0, y_0) and (x_n, y_n), respectively, have assigned position (see Fig. 5, right). In view of Lemma 1 the optimal solutions correspond to the two chains both directed as the gravity vector. With obvious considerations, we would have $(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 \geq d^2 > r_i^2$, and the *i*th length constraint would be violated. ■

If there exists an inactive constraint (unique, in view of Proposition 2), an admissible solution can be achieved by introducing an auxiliary functional, so that the gravity vectors are different for each node. How can these node-specific vectors be chosen?

Consider a reference frame having the origin in the point (x_0, y_0) (denoted by *A* in Fig. 6), one axis (*z* in the figure) parallel to the segment joining the origin and the end effector, and the other axis orthogonal (*w* in the figure). Auxiliary forces p and $-p$ parallel to z are applied to the second and last but one node (*B* and *E*), respectively, in opposite directions. A force h in direction w is applied to all remaining nodes. The

[2]Intuitively, "breaking the chain" by removing the *i*th link; see Fig. 5, right.

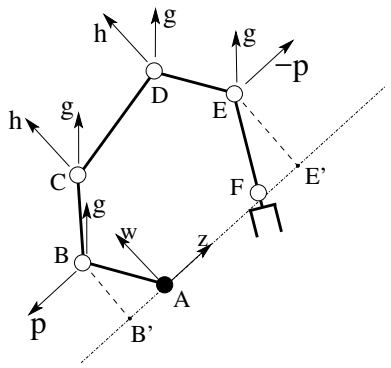


Figure 6: The common gravity g and the auxiliary forces.

following proposition guarantees that admissibility is always achieved by taking p large enough, provided that there is no high disproportion among the link lengths.

Proposition 4. *Assume that each link length is smaller than the sum of any other two. Let p and h be the auxiliary forces as defined above. Then, for p large enough and $h \neq 0$, the solution of the optimization problem has all length constraints satisfied as equalities (hence it is admissible).*

Proof Consider Fig. 6 and let B' and E' be the projections of B and E on the line for A and F. At the optimum, for p large enough, the first and last links are stretched: $\mathrm{dist}(A,B) = r_1$ and $\mathrm{dist}(F,E) = r_N$. Indeed, consider the first link and suppose by contradiction that, at the optimum, $\mathrm{dist}(A,B) < r_1$. Now, by moving the point B to B'' along the circle of radius BC having C as center, we get a feasible solution such that $\mathrm{dist}(A,B'') = r_1$. The value of the objective function corresponding to such a feasible solution is less than the optimal, provided that the displacement vector BB'' has a positive component along $p+g$, which is always the case for p large enough. The existence of a feasible solution with a better value of the objective function contradicts the supposed optimality. The same holds for the last link. Then, for p increasing, the distance $\mathrm{dist}(B',E')$ gets larger and converges to:

$$\begin{aligned} &\mathrm{dist}(A,F) + \mathrm{dist}(A,B') + \mathrm{dist}(E',F) \\ &= \mathrm{dist}(A,F) + r_1 + r_N > r_i, \qquad \text{for any } i, \end{aligned}$$

by assumption. If $\mathrm{dist}(B',E')$ is large enough, we can apply the reasoning of Proposition 3 to the intermediate part of the robot (*B–C–D–E* in the figure). Indeed, if we consider a fictitious convex problem in which we fix the second and last but one nodes to their optimal positions, the remaining nodes optimal positions are clearly unchanged. Then the proof follows from Proposition 3, being the intermediate nodes subject to a common gravity field ($g+h$ in the figure). ■

In the following we report a pseudocode that shows how the *common* functional αf_c is applied and the *auxiliary* functional βf_a is added (the total functional being the sum $\alpha f_c + \beta f_a$), whose weight β is progressively increased by δ_β if necessary.

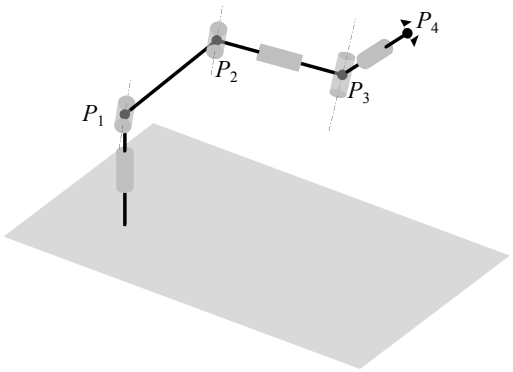


Figure 7: Anthropomorphic arm with spherical wrist.

Pseudocode
Inputs: End-effector position (x_n, y_n); step δ_β; maximum value β_{max}; gravity direction θ; weight $\alpha > 0$.
Outputs: Angles q_i.

1. Set $\beta := 0$. Set $p = [p_x, p_y] := [x_n, y_n]/\|[x_n, y_n]\|$.
2. Solve problem (5)–(10) with functional
   $c^\top := \alpha[\cos(\theta)\ \cos(\theta)\ldots\cos(\theta)] + \beta[0\ \ p_x\ 0\ldots 0\ -p_x\ 0]$
   $d^\top := \alpha[\sin(\theta)\ \sin(\theta)\ \ldots\sin(\theta)] + \beta[0\ \ p_y\ 0\ldots 0\ -p_y\ 0]$
3. IF the *length constraints* (6) are satisfied *as equalities*, then GOTO step 5; ELSE $\beta \leftarrow \beta + \delta_\beta$.
4. IF $\beta \leq \beta_{max}$, then GOTO step 2, ELSE STOP (unsuccessfully).
5. Compute the angles $q_i := \text{atan2}(y_i - y_{i-1}, x_i - x_{i-1})$, $i = 1, \ldots, n$.

Remark 2. *The assumption in Proposition 4 rules out the extremal condition in which, due to a long link, pulling the extremal links is not sufficient to accommodate the chain, because the inequality constraint for the long link remains inactive at the optimum. To fix this problem, further auxiliary forces can be subsequently added, so as to "activate" the unique inactive constraint:* e.g.*, forces directed as p and $-p$ could be applied to nodes C and D in Fig. 6.*

Remark 3. *The propositions of this section are also valid for three-dimensional robots moving in a plane which rotates along the vertical axis. It is not difficult to see that they can be extended,* e.g.*, to DLR manipulators (considered in the following) with a possibly non-vertical gravity field, which can be integrated by an auxiliary functional.*

2.3. Guidelines for the choice of the gravity vector

The direction of the gravity vector is a parameter that governs the solution of the inverse kinematics problem as formulated in the proposed approach. In particular, it allows us to enforce a "shape" of the robot, given an end-effector position. It is worth

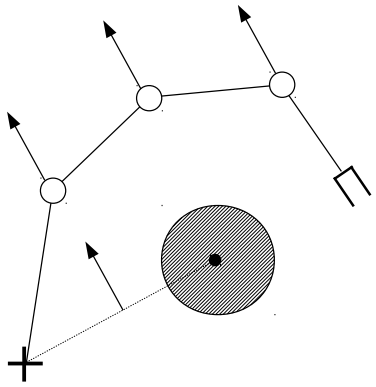


Figure 8: To circumvent an obstacle, the gravity vector may be chosen orthogonal to the segment joining the base to the center of the obstacle.

mentioning that, in the case of redundant robots (*i.e.*, robots that admit many configurations resulting in the same end-effector pose), any method for inverse kinematics must include, either explicitly or implicitly, a parameter that allows to determine a single configuration among the many configurations corresponding to the same end effector pose. For the method proposed in the present paper, this parameter is precisely the gravity vector. Due to the analogy of the rope subject to gravity, the meaning of the parameter and its effect on the obtained configuration are intuitive and this very intuition can help in choosing it. The choice of the gravity has to be made on a case-by-case basis and may help in forcing the selection of collision free configurations, in the presence of obstacles (another possibility for avoiding collision, based on the introduction of suitable constraints in the optimization problem, will be considered in Section 3). As an example of how the gravity vector may be chosen in different situations, for avoiding obstacles, we report three cases (Figures 8-10).

a : an obstacle is located in the interior of the workspace (Figure 8). By choosing the gravity vector to be orthogonal to the segment joining the base of the robot to the center of the obstacle, the robot is forced to stay either on one side or the other with respect to the mentioned segment, thus avoiding collisions;

b : a conic region has to be avoided in Figure 9, and the gravity vector may be chosen orthogonal to the segment joining the base to the vertex of the region;

c : the admissible workspace is a vertical stripe (Figure 10). In this case, a reasonable direction for the gravity vector is vertical.

2.4. Extension to non-planar manipulators

The approach described so far is not suitable for generic non-planar manipulators, *i.e.*, manipulators represented by a sequence of nodes that do not necessarily lie in the same plane. Thanks to the analogy of the rope subject to gravity, it is easy to recognize that different gravity vectors acting in different points are necessary in order for

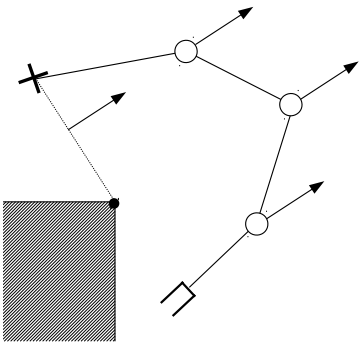


Figure 9: In this case, the gravity vector may be chosen orthogonal to the segment joining the base to the vertex of the polygon to be avoided.

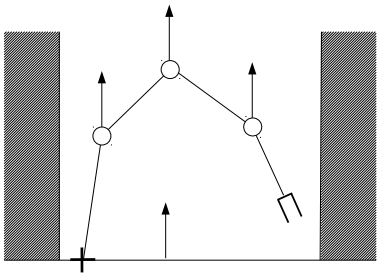


Figure 10: To avoid collision with vertical boundaries, the gravity vector may be chosen as in the figure.

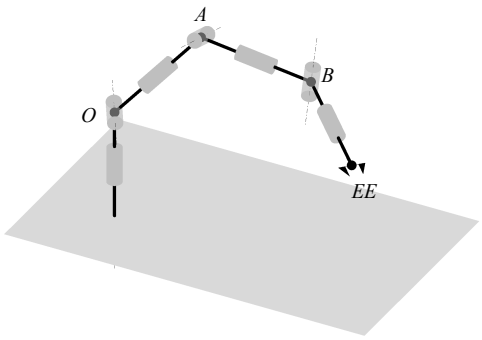


Figure 11: Kinematic structure of the DLR manipulator described by [5].

the robot to assume a non-planar configuration. Moreover, in general, additional constraints must be included in the optimization problem to enforce the consistency of the solution with respect to the robot structure. Unfortunately, due to these constraints, the relaxed problem[3] might become non-convex. As an example, consider the anthropomorphic arm with spherical wrist, whose kinematic structure is shown in Fig. 7. Node positions are denoted by $P_i,\ i = 1,\ldots,4$. For generic values of the joint variables, the nodes do not lie in the same plane. Consider now the position P_3 of the wrist's center. For an assigned P_2, having $||P_3 - P_2||_2^2 = r_3^2$ ($||\cdot||_2$ is the Euclidean norm) does not guarantee that P_3 is compliant with the robot structure: since the joint located in P_2 is not spherical, the further constraint $[(P_3 - P_2) \times (P_2 - P_1)] \cdot (P_3 - P_2) = 0$ is necessary, where the symbols $\times$ and $\cdot$ denote cross product and dot product, respectively. The latter constraint is non-convex, rendering non-convex the whole relaxed problem. However, in some cases of practical significance, the technique can be employed even for non-planar manipulators. As an example, consider the DLR manipulator described by [5], whose kinematic structure is depicted in Fig. 11. Note that the links *O*–*A*, *A*–*B* and *B*–*EE* may have arbitrary orientations. For this system, given any position for the points[4] *EE*, *B* and *A*, compatible with the link lengths, there exists an admissible configuration in terms of joint variables: as a consequence, no additional constraints are required for complying with the robot structure and the relaxed problem remains convex. In general, the relaxed problem remains convex (and thus the method is viable) for all manipulators that can be represented by a sequence of nodes $P_i,\ i = 1,\ldots,n$, whose mutual positions are only constrained by $||P_i - P_{i-1}||_2^2 = r_i^2,\ i = 2,\ldots,n$. In particular, any open kinematic chain consisting of a sequence of an arbitrary number of spherical joints enjoys this property, thus being suitable for the proposed method. A simulation of a DLR-like manipulator is reported in Section 4.

In other cases the physical realizability of a configuration can be imposed if one restricts the external forces to have special directions. For instance, in the case of

[3] By *relaxed problem* we mean the optimization problem where the equality constraints $(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 = r_i^2,\ \ i = 1,2,\ldots,n$ are replaced by inequality constraints $(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 \leq r_i^2,\ \ i = 1,2,\ldots,n$.

[4] That are actually the nodes, *i.e.*, the optimization variables.

Table 1: The Denavit-Hartenberg parameters for the robot of Fig. 12.

Link	a_i	α_i	d_i	ϑ_i
1	a_0	0	0	0
2	0	$-\frac{\pi}{2}$	d_2	ϑ_2
3	a_3	0	0	ϑ_3
4	a_4	0	0	ϑ_4
5	a_5	0	0	ϑ_5
6	a_6	0	0	ϑ_6
7	a_7	0	0	ϑ_7
8	a_8	0	0	ϑ_8
9	a_9	0	0	ϑ_9

a planar vertical arm which lies on a plane rotating with respect to the vertical axis, admissibility can be ensured by assuming forces lying in that plane (see the first example in Section 4.2). Simulation results are reported in Section 4, in which significant examples of non-planar robot are considered.

Remark 4. *For* planar robots*, a desired end-effector attitude can be considered in the optimization problem without difficulties. Indeed, if the robot is planar, the attitude is actually a direction, since no rotation around the direction is possible. If the end-effector position is given, then the problem involves only the remaining links. Conversely, if we can freely choose the end-effector position, its orientation is a linear constraint on the last link of length* r_n*:* $(x_n - x_{n-1} \;\; y_n - y_{n-1} \;\; z_n - z_{n-1})^\top = r_n \nu$*, where* ν *is a unit vector with the assigned orientation. This does not affect convexity.*
For non-planar robots*, the situation is different. In particular, two cases may occur:*

1) *the approach is not suitable for the robot, because the constraints that enforce the kinematic consistency are not convex (this is one of the limitations of our approach);*

2) *the approach is suitable, implying that the wrist of the robot is spherical (otherwise additional non-convex constraints would be present).*

If case 2) occurs, the optimization problem involves only the first portion of the kinematic chain (up to the wrist), since the location of the wrist is determined by the location and attitude of the end-effector.

2.4.1. Two examples of non-planar manipulators

In the following we formulate the optimization problem for two non-planar redundant robots. For the same robots, we report simulations in Section 4.

The first example is a redundant robot on a cart; the kinematic structure is reported in Fig. 12 and the Denavit-Hartenberg parameters are in Table 1. The robot is a planar manipulator having 7 d.o.f. mounted on a cart moving along the x-axis. The whole planar structure can rotate above the first revolute joint. The gravity vector has a vertical component which can point up or down, to obtain a robot configuration with upward or downward concavity, and a horizontal component which can be zero or

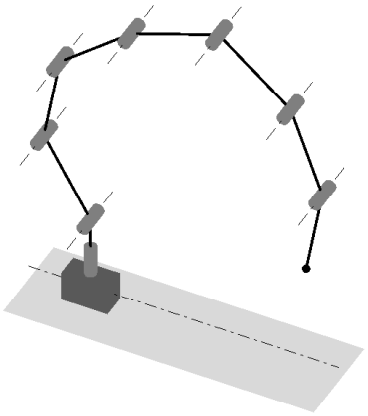


Figure 12: The kinematic structure of the redundant robot on a cart. The robot has 9 d.o.f. (one prismatic joint and 8 revolute joints).

can incline the robot forward or backward. Given the desired end-effector position (x_E^d, y_E^d, z_E^d), the convex optimization problem can be formulated as:

$$\min_{x_i,y_i,z_i} \quad \pm\gamma_V \sum_i z_i + \gamma_N \|[x_n - x_E^d \quad y_n - y_E^d \quad z_n - z_E^d]\|_2 \tag{15}$$

$$+\gamma_K(\eta+\xi) + \gamma_H(\nu_{H,x}\sum_i x_i + \nu_{H,y}\sum_i y_i) \tag{16}$$

$$s.t. \quad (x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 \le r_i^2, \quad i = 1,2,\ldots,n \tag{17}$$

$$z_i \ge 0, \quad i = 1,2,\ldots,n \tag{18}$$

$$y_0 = z_0 = 0 \tag{19}$$

$$\eta, \xi \ge 0 \tag{20}$$

$$\eta - \xi \le x_0 \tag{21}$$

$$\eta - \xi \ge x_0, \tag{22}$$

where $\gamma_V, \gamma_N, \gamma_K, \gamma_H$ are proper weights, respectively, on vertical gravity, on the distance of the end-effector from its desired position, on the distance of the cart from its central position, and finally on horizontal gravity. η and ξ are nonnegative auxiliary variables such that $x_0 = \eta - \xi$, $|x_0| = \eta + \xi$; $\nu_H = [\nu_{H,x} \quad \nu_{H,y}]^\top$ is the horizontal gravity vector; the term in γ_V is taken with positive or negative sign depending on the desired concavity (upward or downward, respectively).

Remark 5. *In this formulation, the cost functional is a combination of more terms, each expressing a different requirement. Of course, different robot configurations may be obtained by properly changing the weights associated with each term. For high-precision positioning, the end-effector coordinates can be assigned as constraints:* $x_n = x_E^d$, $y_n = y_E^d$, $z_n = z_E^d$. *This can cause "unnatural" trajectories and even feasibility problems if the target point is too far. Clearly these equality constraints can be*

Table 2: The Denavit-Hartenberg parameters for the robot of Fig. 11.

Link	a_i	α_i	d_i	ϑ_i
1	0	$-\frac{\pi}{2}$	d_1	ϑ_1
2	0	$\frac{\pi}{2}$	0	ϑ_2
3	0	$-\frac{\pi}{2}$	d_3	ϑ_3
4	0	$\frac{\pi}{2}$	0	ϑ_4
5	0	$-\frac{\pi}{2}$	d_5	ϑ_5
6	0	$\frac{\pi}{2}$	0	ϑ_6
7	0	0	d_7	ϑ_7

activated ad hoc *when the robot is sufficiently close to the target.*

The second example is the DLR 7 d.o.f. non-planar manipulator represented in Fig. 11, whose Denavit-Hartenberg parameters are reported in Table 2. For this robot any solution of the optimization problem (consisting in the coordinates (x_A, y_A, z_A) and (x_B, y_B, z_B) of the points A and B) is physically realizable (provided that, as usual, inequality constraints are satisfied as equalities). Given the desired end-effector position (x_E^d, y_E^d, z_E^d), the gravity vectors $f_A = (f_A^x, f_A^y, f_A^z)$ and $f_B = (f_B^x, f_B^y, f_B^z)$, acting respectively on the points A and B (see Figure 11), the convex optimization problem can be formulated as:

$$\begin{aligned} \min_{x_A, y_A, z_A, x_B, y_B, z_B} \quad & f_A^x x_A + f_A^y y_A + f_A^z z_A + f_B^x x_B + f_B^y y_B + f_B^z z_B \\ s.t. \quad & (x_A - x_O)^2 + (y_A - y_O)^2 + (z_A - z_O)^2 \leq r_1^2, \\ & (x_B - x_A)^2 + (y_B - y_A)^2 + (z_B - z_A)^2 \leq r_2^2, \\ & (x_{EE} - x_B)^2 + (y_{EE} - y_B)^2 + (z_{EE} - z_B)^2 \leq r_3^2, \end{aligned}$$

where $r_i \quad (i = 1, 2, 3)$ are the lengths of the robot links.

A desired end-effector attitude can be obtained easily by imposing the relation: $(x_{EE} - x_B \quad y_{EE} - y_B \quad z_{EE} - z_B)^\top = r_3 \nu$, where ν is a unit vector with the assigned orientation. If this is the case, the optimization problem becomes

$$\begin{aligned} \min_{x_A, y_A, z_A, x_B, y_B, z_B} \quad & f_A^x x_A + f_A^y y_A + f_A^z z_A \\ s.t. \quad & (x_A - x_O)^2 + (y_A - y_O)^2 + (z_A - z_O)^2 \leq r_1^2, \\ & (x_B - x_A)^2 + (y_B - y_A)^2 + (z_B - z_A)^2 \leq r_2^2. \end{aligned}$$

2.5. An application to the forward kinematics of parallel robots: the case of the Delta robot

Now we show how the same idea (minimizing a potential energy subject to properly relaxed kinematics constraints) can be applied to a *forward* kinematics problem. Parallel robots (described by [21]) consist of kinematic closed chains; the loop closure equations are usually nonlinear expressions of the joint coordinates. As a consequence, the forward kinematic problem of a parallel robot is usually much more complex than the inverse kinematic problem. Although we do not claim that the proposed approach

may be employed to any kind of parallel robot (the problem is currently under investigation), we show an application to the Delta robot. The Delta robot (described by [22]) is a well-known parallel robot having 3 degrees of freedom: a moving platform can translate along the three axes based on the three active revolute joints attached to a fixed platform (see Fig. 13). The forward kinematics problem, in this case, can be for-

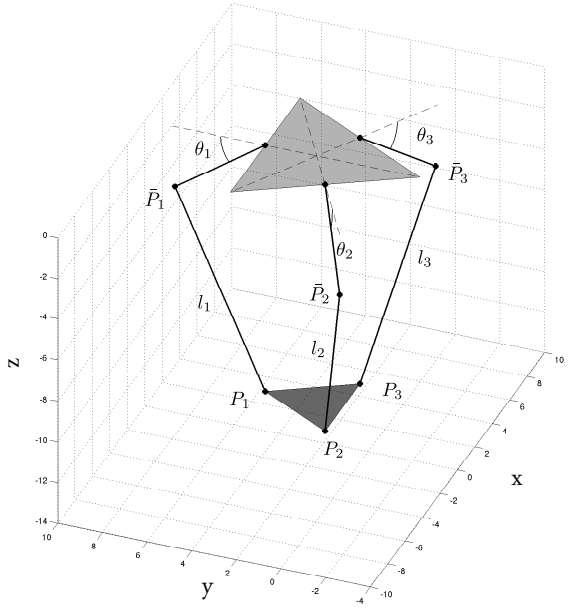


Figure 13: The Delta robot.

mulated as follows: find the coordinates of the center of mass of the moving platform, given the angles of the three revolute joints. Referring to Fig. 13, let $l_i = ||\bar{P}_i - P_i||_2$, $i = 1,2,3$, be the (fixed) lengths of the links attached to the moving platform. The coordinates $(\bar{x}_i, \bar{y}_i, \bar{z}_i)$ of points $\bar{P}_i$, $i = 1,2,3$, are uniquely determined by the known values of θ_1, θ_2 and θ_3. By assigning a fictitious mass to the points $P_i = (x_i, y_i, z_i)$, $i = 1,2,3$, and choosing a vertical gravity (pointing downwards), the following convex programming problem can be formulated, where $d_{ij} = ||\bar{P}_i - P_i||_2$:

$$\min_{x_i, y_i, z_i \;\; i=1,2,3} z_1 \tag{23}$$

$$s.t. \quad (\bar{x}_i - x_i)^2 + (\bar{y}_i - y_i)^2 + (\bar{z}_i - z_i)^2 \leq l_i^2, \quad i = 1,2,3 \tag{24}$$

$$(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2 \leq d_{12}^2 \tag{25}$$

$$(x_2 - x_3)^2 + (y_2 - y_3)^2 + (z_2 - z_3)^2 \leq d_{23}^2 \tag{26}$$

$$(x_1 - x_3)^2 + (y_1 - y_3)^2 + (z_1 - z_3)^2 \leq d_{13}^2 \tag{27}$$

$$z_1 = z_2 \tag{28}$$

$$z_2 = z_3. \tag{29}$$

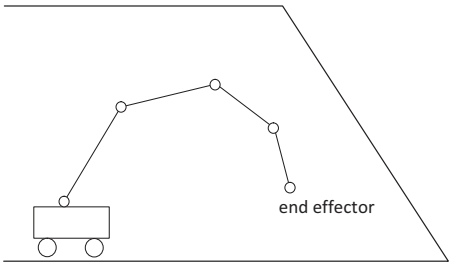


Figure 14: Obstacle avoidance: a convex admissible workspace.

Constraints (28) and (29) express the absence of rotational degrees of freedom (the moving platform remains parallel to the fixed one). The Matlab code implementing the above method for forward kinematics of a Delta robot is available at the URL `http://control.units.it/en/catenary`.

3. Dealing with obstacles

The linear inequality constraints (9) may be used, by a proper choice of M, N and q, for enforcing obstacle avoidance. In particular, we will address two cases as follows.

3.1. Convex admissible workspace

In this case, enforcing obstacle avoidance is straightforward. Indeed, a convex admissible workspace (as that in Fig. 14) can be described to an arbitrary degree of accuracy by the intersection of a sufficient number m of half-planes:

$$\mathscr{A} = \left\{(x,y) : a_j x + b_j y + c_j \leq 0,\ j = 1,\ldots,m\right\}. \tag{30}$$

Assuming that the desired end-effector position is within the admissible workspace (otherwise no admissible configuration exists), the constrained inverse kinematics problem may be formulated as (5)–(10), where M, N and q are such that the constraints (30) hold for each of the free nodes:

$$a_j x_i + b_j y_i + c_j \leq 0,\ j = 1,\ldots,m,\ i = 1,\ldots,n-1. \tag{31}$$

In this way the nodes are guaranteed to lie in the admissible workspace. Since the workspace is convex, no collision can occur for all the points of the links.

3.2. Convex obstacles within a convex region

Consider now the case when the robot is required to stay within a convex region $\mathscr{A}$ (described as in the previous subsection) while avoiding collisions with convex obstacles $\mathscr{O}_k$, $k = 1,\ldots,l$. In other words, the admissible workspace is:

$$\tilde{\mathscr{A}} = \mathscr{A} \setminus \bigcup_{k=1}^{l} \mathscr{O}_k. \tag{32}$$

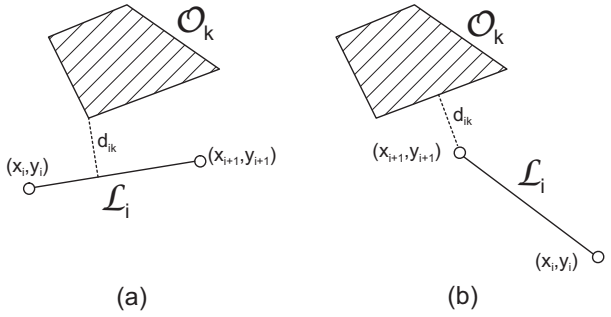


Figure 15: Distance link-obstacle.

Convexity of the obstacles is assumed to ensure numerical efficiency in computing the minimum distance point. The assumption can be relaxed by adopting convex over-bounding volumes of the physical obstacles. Since $\tilde{\mathscr{A}}$ is in general not convex, the convexity of the problem would be lost by applying the constraints $(x_i, y_i) \in \tilde{\mathscr{A}}$, $i = 1, \ldots, n-1$. However there is a scenario, of practical significance, where convexity may be retained and kinematic inversion is possible within a workspace of the form (32). Let the robot be in a given, collision-free, configuration at time t' and assume that the end-effector has to move from the current position (x'_E, y'_E) to a new (close) position. Such a situation occurs, for example, when the end-effector has to track a reference generated by a human operator: the operator "pushes" the end-effector towards the goal and the robot has to accommodate its pose to track the reference while avoiding collisions with obstacles. The idea here is to use *local* convex constraints, where the meaning of "local" is twofold: indeed the mentioned constraints are included in the optimization problem (i) only when the robot is close to an obstacle and (ii) only for those links of the robot that are close to the obstacle. Notice that, provided that the geometry of the robot and of the environment is exactly known, judging the occurences of (i) and (ii) may be performed systematically and automatically by means of proper algorithms, *e.g.*, the well-known method proposed by [23]. Let the ith link at time t' be the segment $\mathscr{L}_i = \left\{\lambda(x'_i, y'_i) + (1-\lambda)(x'_{i+1}, y'_{i+1}),\ \lambda \in [0,1]\right\}$. We define the distance between the ith link and the kth obstacle as:

$$d_{ik} = \min\left\{||a-b||_2 : a \in \mathscr{O}_k,\ b \in \mathscr{L}_i\right\}.$$

From Fig. 15 it is clear that, by allowing the joints i and $i+1$ to move less than d_{ik}, no collision between $\mathscr{O}_k$ and $\mathscr{L}_i$ can occur. In other words, by adding to the optimization problem the following constraints:

$$(x_i - x'_i)^2 + (y_i - y'_i)^2 < d_{ik}^2 \tag{33}$$
$$(x_{i+1} - x'_{i+1})^2 + (y_{i+1} - y'_{i+1})^2 < d_{ik}^2, \tag{34}$$

the problem remains convex and the solution (if any) is such that link $\mathscr{L}_i$ does not collide with obstacle $\mathscr{O}_k$. By taking into account the closest obstacle to each link, the

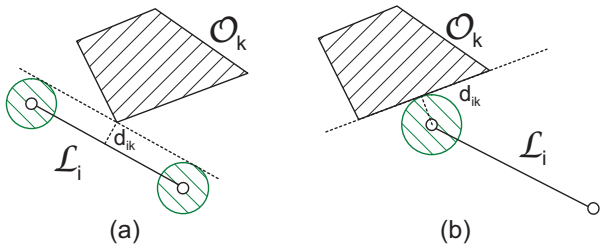


Figure 16: Restricting the movements of the joints to less than d_{ik} may be very conservative.

optimization problem may be formulated as:

$$\min \quad c^\top x + d^\top y \tag{35}$$

$$s.t. \quad (x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 \leq r_i^2, \quad i = 1,2,\ldots,n \tag{36}$$

$$(x_0, y_0) = (0,0) \tag{37}$$

$$(x_n, y_n) \text{ assigned} \tag{38}$$

$$(x_i - x_i')^2 + (y_i - y_i')^2 < \min_k d_{ik}^2, \quad i = 1,2,\ldots,n-1 \tag{39}$$

$$(x_{i+1} - x_{i+1}')^2 + (y_{i+1} - y_{i+1}')^2 < \min_k d_{ik}^2, \tag{40}$$

$$i = 1,\ldots,n-1 \tag{41}$$

$$(x_i', y_i') \quad i = 1,2,\ldots,n \text{ assigned} \tag{42}$$

$$Mx + Ny \leq q \tag{43}$$

$$Qx + Ry = s. \tag{44}$$

Hence the next proposition, whose proof follows immediately from continuity arguments, can be stated.

Proposition 5. *Given a set of convex obstacles $\mathcal{O}_k$, $i = 1,2,\ldots,m$, and a collision-free configuration (x_i', y_i'), $i = 1,2,\ldots,n$, corresponding to the end-effector position $(x_E', y_E') = (x_n', y_n')$, the problem (35)–(44), where $(x_n, y_n) = (x_E', y_E') + \delta(x_E - x_E', y_E - y_E')$, admits a feasible solution for $\delta > 0$ sufficiently small. Such a solution, if admissible, is collision-free.*

The proposition may be exploited to allow navigation in a constrained environment by solving a sequence of convex optimization problems, whose constraints are adapted to the current configuration of the robot via suitable "local" constraints.

Remark 6. *The nature of the method is local, and resembles that of well-known approaches such as [24, 25, 18, 19]. The mentioned approaches rely on the linearized kinematics, in particular on the null-space of the Jacobian: the motion of the whole robot is governed by two contributions, one that pushes the end-effector towards the goal, the other that accommodates the robot in order to stay away from the closest obstacle. Although all the approaches deal with the distance from the robot to the obstacles in 3D Cartesian space, none of them is based on convex programming. Moreover, our method is different in nature because it is not based on linearized kinematics.*

Finally, it does not require to set a significant parameter, namely the relative weight between the two mentioned control actions.

A drawback of the strategy described above is that, in some cases, it may result in too conservative constraints. Consider, for instance, Fig. 16. It is clear that, in both the cases (a) and (b), the link could move parallel to the dashed lines while avoiding collision. Unfortunately, due to the small distance d_{ik}, the constraints (33)–(34) would allow only unnecessarily small movements in such a direction. It is possible to alleviate this problem by slightly modifying the previous strategy. Let $\bar{k}_i = \arg\min_k d_{ik}$, namely the index of the closest obstacle to link i, and modify the constraints (39)–(41) as follows:

$$(x_i - x_i')^2 + (y_i - y_i')^2 < \min_{k \neq \bar{k}_i} d_{ik}^2 \tag{45}$$

$$(x_{i+1} - x_{i+1}')^2 + (y_{i+1} - y_{i+1}')^2 < \min_{k \neq \bar{k}_i} d_{ik}^2, \tag{46}$$

where $i = 1, 2, \ldots, n-1$. In other words, the new constraints do not take into account the closest obstacle to link i. Instead, the requirement that no collision occurs with the closest obstacle is enforced by properly chosen linear constraints (the red lines in Fig. 17) that require that link i remains to the same side with respect to the closest obstacle. In particular, two cases may occur, as shown in Fig. 17: either the minimum distance is attained by a vertex of the object, as in case (a), or not, as in case (b). In the former case, the constraints may be chosen by taking the line to which $\mathscr{L}_i$ belongs and translating it towards the closest obstacle of an amount $d_{i\bar{k}_i}$. More precisely, let $ax + by + c = 0$ be the normal (*i.e.*, such that $a^2 + b^2 = 1$) form of the equation of the line passing through (x_i', y_i') and (x_{i+1}', y_{i+1}'). Then c is the distance from the origin to the line to which $\mathscr{L}_i$ belongs. The linear constraints relative to the ith link take the form:

$$ax_i + by_i + c \pm d_{i\bar{k}_i} \lesseqgtr 0 \tag{47}$$

$$ax_{i+1} + by_{i+1} + c \pm d_{i\bar{k}_i} \lesseqgtr 0, \tag{48}$$

where the sign on the first members and the verse of the inequalities depend on whether the obstacle belongs to the same half-plane of the origin or not. Such constraints express the requirement that the ith link does not cross the closest line parallel to itself and passing through the frontier of the closest obstacle. As far as case (b) is concerned, the linear constraint may be chosen in the same way as above, with the only difference that the line is now the one to which the closest edge (of the closest obstacle) belongs. Proposition 5 still holds, since the constraints (45)–(46) guarantee that no collision can occur with obstacles other than the closest.

4. Simulation results

4.1. Planar redundant robot

We report a simulation example obtained by applying the proposed technique to a 7-link planar robot. The initial configuration of the robot is shown in black in Fig. 18:

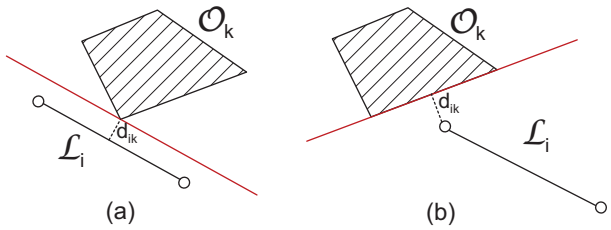


Figure 17: A linear constraint may be employed for avoiding collision with the closest obstacle, reducing conservativeness.

the base is located in $(0,0)$ and the end-effector in $(10,45)$. The gravity vector is $(\cos\theta, \sin\theta)$ with $\theta = \frac{5\pi}{8}$, kept constant during the whole simulation and chosen to obtain a concavity of the robot directed towards the fourth quadrant. Note that the value of θ is not critical: any gravity vector pointing towards the second quadrant will result in a similar preferred concavity orientation of the robot. Two obstacles are present, represented by the gray boxes. Moreover, three linear constraints define the admissible region for the robot as the intersection of three half-planes delimited by the red lines. The simulation was performed as follows: by means of a graphical user interface, an operator specified, on-line, the desired motion of the end-effector, chosen among {stand by, (N)orth, (E)ast, (S)outh, (W)est, NE, SE, SW, NW}. A fixed step size of 0.75 units of length in each direction was set. Then, at each time step, the new configuration of the robot was automatically computed by solving (35)–(44). We stress that the role of the operator was only to specify the desired motion of the end effector, while the proposed method automatically guarantees the kinematics inversion and collision avoidance. A stroboscopic view of the whole trajectory is shown in Fig. 18: the constrained trajectory from the first pose (black) to the last (blue) was obtained by solving a sequence of convex optimization problems, one per time step. As for the constraints, this is the case of convex obstacles (the boxes) within a convex region (the intersection of the three half-planes delimited by the red lines), described in Section 3.2. Hence, some local convex constraints of the type (47)–(48) need to be activated depending on the current configuration of the robot (thus they change in time). An example of such constraints is reported in Fig. 19: the two magenta lines represent a pair of linear constraints acting either on the last link or the penultimate when the robot is in the blue configuration. It is clear that the type of constraints is that of Fig. 17(a). Constraints of the type of Fig. 16, although not shown in the figure, are present for all the (x_i, y_i).

The Matlab code for the above simulation is available at the URL `http://control.units.it/en/catenary`.

4.2. Non-planar redundant robots

In the following we provide simulations regarding the two examples of non-planar robots introduced in Section 2.4.

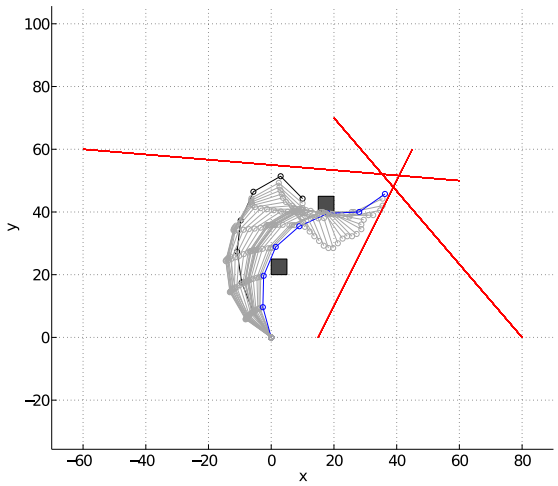


Figure 18: A stroboscopic view of the whole constrained trajectory of a planar robot.

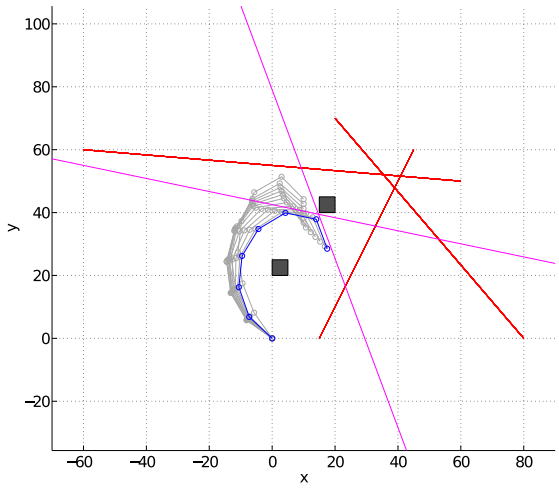


Figure 19: Example of local constraints (magenta), which are configuration-dependent and active on a subset of links: here the constraints are relative to the blue configuration and acting one on the last link, the other on the penultimate.

4.2.1. Redundant robot on a cart

The robot is a planar manipulator having 7 d.o.f. mounted on a cart moving along the x-axis. The whole planar structure can rotate above the first revolute joint. The gravity vector has a vertical component which can point up or down, to obtain a robot configuration with upward or downward concavity, and a horizontal component which can be zero or can incline the robot forward or backward. The robot arm can also be stretched in order to form a straight line. The difference among the horizontal gravity modes is shown in Fig. 20. To move the end-effector, the user needs to specify its desired position. For computing the robot configuration, two different solution methods are used: the first solves the convex optimization problem at each step (the desired end-effector position moves on along the segment joining the initial and final positions); the second (*fast mode*) solves the convex optimization problem just for the final position, while intermediate configurations are obtained via linear interpolation. Given the desired end-effector position (x_E^d, y_E^d, z_E^d), the convex optimization problem can be formulated as (16)-(22). If the solution is not physically admissible, a new optimization problem is solved: the cost functional has an additional term (due to auxiliary forces applied as proposed in Section 2.2), whose weight is progressively increased until all length constraints are satisfied as equalities.

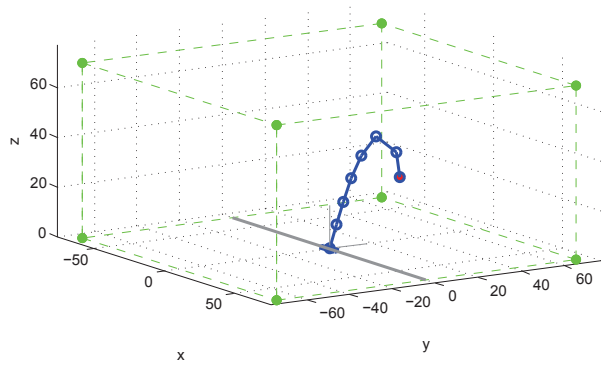
Figs. 21 (a), (b) and (c) show the stroboscopic view of some trajectories obtained with the fast mode; Fig. 21 (d) shows instead a trajectory obtained by solving at each step the convex optimization problem. The Matlab code for this non-planar example is available at the URL `http://control.units.it/en/catenary`.

4.2.2. DLR-like manipulator

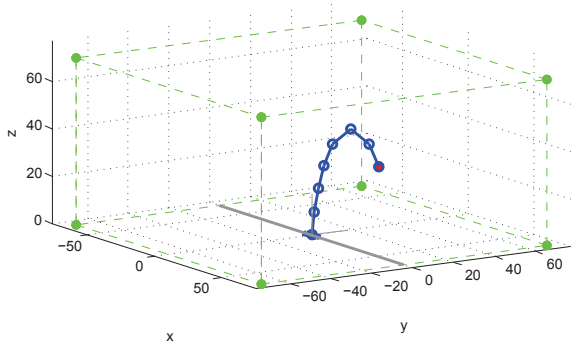
Consider the 7 d.o.f. non-planar manipulator represented in Fig. 11, whose Denavit-Hartenberg parameters are reported in Table 2. The proposed kinematic inversion method has been applied to a commercial robot, the Mitsubishi PA-10, having the same structure of the DLR robot. Precisely, the end-effector has been constrained to lie on a particular closed path (the ellipse in Figs. 22 and 23), keeping a chosen orientation (in particular, the Z-Y-Z Euler angles of the end-effector orientation are $[\varphi\,,\,\vartheta\,,\,\psi] = [0\,,\,\frac{\pi}{2}\,,\,0]$). The end-effector orientation has been imposed by determining the position of the last spherical wrist of the robot, given each desired position of the end-effector, resolving the optimization problem keeping fixed both the end-effector and last spherical wrist positions and then using redundancy to obtain the desired orientation.

5. Concluding discussion

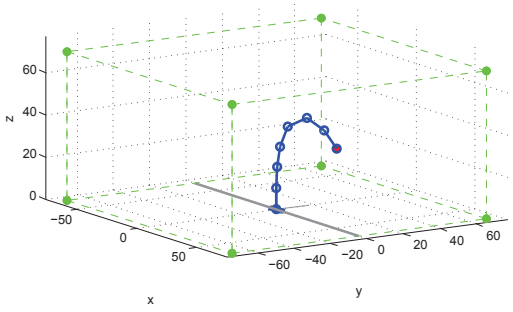
In this paper we propose an approach to the kinematic inversion problem which is based on convex optimization. The essential idea is relaxing equality constraints to achieve convex inequality constraints. The optimization problem is formulated so that the constraints are in fact satisfied as equalities, which ensures the physical admissibility of the solution, and can be solved in a very efficient way. The method is valid, in general, for planar manipulators only but it can be employed for some particular, practically important, non-planar manipulators. Compared to the available methods,



(a) Forward horizontal gravity

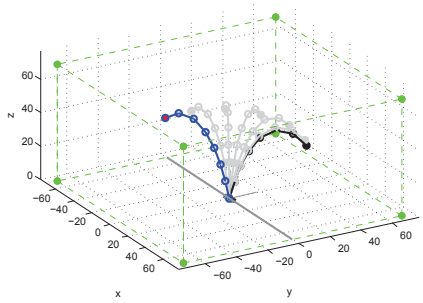


(b) No horizontal gravity

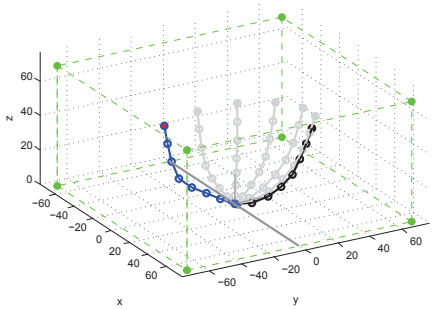


(c) Backward horizontal gravity

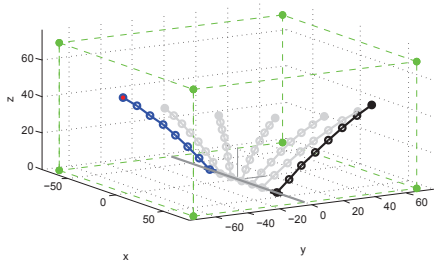
Figure 20: Effect of different horizontal gravity modes on the robot configuration with end-effector position $(20, 20, 30)$.



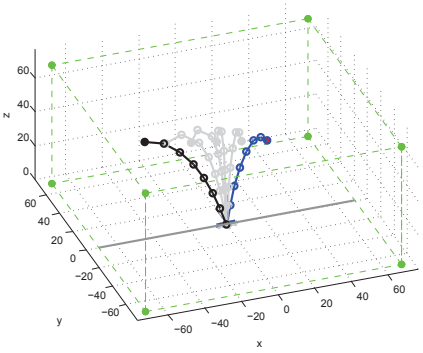
(a) Downward concavity; the end-effector moves from $(30,30,30)$ to $(-30,-30,40)$.



(b) Upward concavity; the end-effector moves from $(30,30,50)$ to $(-40,-30,30)$.



(c) The end-effector moves from $(55,50,45)$ to $(-55,-50,45)$, the cart from $(0,0,29)$ to $(0,0,-33)$.



(d) The end-effector moves from $(-40,10,50)$ to $(40,40,20)$.

Figure 21: Stroboscopic views of the whole trajectory of a non-planar robot, obtained with the *fast mode* solution (a), (b), (c) or by solving at each time step a convex optimization problem (d). The first pose is depicted in black, the last in blue.

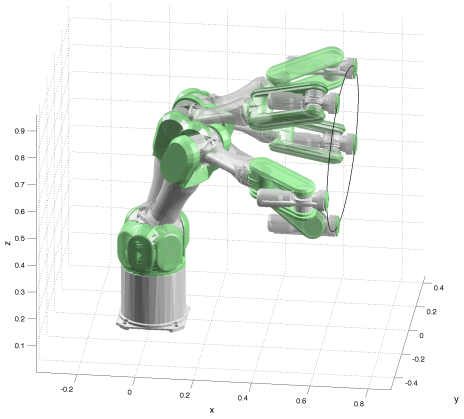


Figure 22: A stroboscopic view of the whole trajectory of the PA10 Mitsubishi manipulator.

the scheme allows to easily include space constraints. Moreover, the optimization allows to exploit the redundancy of solutions in order to ensure a proper orientation of the robot. However, in the case of a high number of constraints, solving a constrained optimization problem might be difficult and require *ad hoc* hardware. We believe that the approach is complementary to the existing ones and can be successfully combined with them. From a numerical point of view, a tolerance ε has to be introduced to check if the constraints are satisfied as equalities. However, this produces an error in the position of the end-effector, which is the sum of the vectors corresponding to the robot links: hence, such an error is upper bounded by $n\varepsilon$, where n is the number of consecutive links. The tolerance can then be chosen so as to ensure the required precision in positioning the end-effector. The optimization problem we have to cope with is a standard convex programming problem with quadratic constraints. Length constraints $(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2 \leq r_i^2$ can be expressed in the quadratic form $v^\top L_i v \leq r_i^2$, where $v = (x^\top \ y^\top \ z^\top)^\top$ is a vector including all coordinates and L_i is a symmetric matrix having three 2×2 blocks of the form $\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$ in the proper position. The other constraints are linear. This type of problems can be solved in fractions of seconds with ordinary hardware and public domain software such as CVX, a package for specifying and solving convex programs ([26, 27]). For instance, a time of about a hundred milliseconds is sufficient to solve the convex problem for a non-planar robot with 7 links, on an Intel Core i7 processor with 4 cores and a base frequency of 2.3 GHz. Such a computation time is certainly problematic for a real-time implementation where an optimization problem has to be solved at each sampling instant. However, we stress the fact that, being the proposed method *global*, it has to be considered primarily as a method for finding robot configurations satisfying some constraints: it is not necessary to solve an optimization problem at each sampling instant, since intermediate configurations can be found by interpolation. On the other hand, there exist

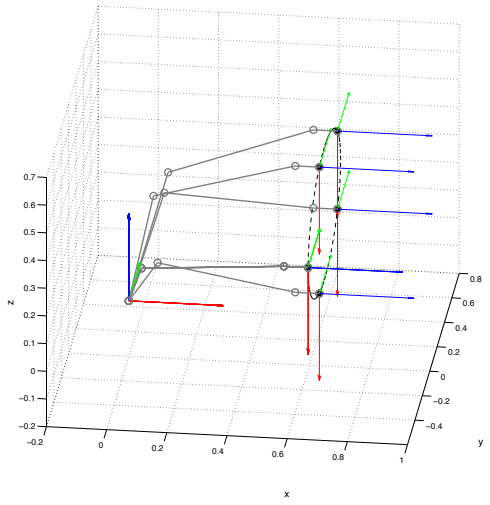


Figure 23: The same trajectory of Fig. 22, but using a stylized robot and visualizing the end-effector orientation.

techniques for creating C-code based solvers for specific instances of convex optimization problems, see for instance [28, 29, 30]. Results have been reported of times of about 1 ms for solving problems having hundreds of decision variables (far more than in our case, in which the number of decision variables is twice the number of joints). Hence we believe that a C-code implementation, tailored for the problem at hand, could significantly reduce the computation time.

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

[2] E. Rimon, D. E. Koditschek, Exact robot navigation using artificial potential functions, IEEE Transactions on Robotics and Automation 8 (5) (1992) 501–518.

[3] F. Blanchini, F. A. Pellegrino, L. Visentini, Control of manipulators in a constrained workspace by means of linked invariant sets, International Journal of Robust and Nonlinear Control 14 (1314) (2004) 1185–1205.

[4] F. Blanchini, S. Miani, F. A. Pellegrino, B. Van Arkel, Enhancing Controller Performance for Robot Positioning in a Constrained Environment, IEEE Transactions on Control Systems Technology 16 (5) (2008) 1066–1074.

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

[6] S. Chiaverini, G. Oriolo, I. D. Walker, Kinematically redundant manipulators, in: B. Siciliano, O. Khatib (Eds.), Springer Handbook of Robotics, Springer, 2008, pp. 245–268.

[7] D. E. Whitney, Resolved motion rate control of manipulators and human prostheses, IEEE Transactions on Man-Machine Systems 10 (2) (1969) 47–53.

[8] A. Balestrino, G. De Maria, L. Sciavicco, Robust control of robotic manipulators, in: Proceedings of the 9th IFAC World Congress, Vol. 5, 1984, pp. 2435–2440.

[9] W. A. Wolovich, H. Elliott, A computational technique for inverse kinematics, in: Proceedings of the 23rd IEEE Conference on Decision and Control, 1984, pp. 1359–1363.

[10] C. Wampler, Manipulator Inverse Kinematic Solutions Based on Vector Formulations and Damped Least-Squares Methods, IEEE Transactions on Systems, Man, and Cybernetics 16 (1) (1986) 93–101.

[11] S. R. Buss, J. S. Kim, Selectively damped least squares for inverse kinematics, Journal of Graphics, GPU, and Game Tools 10 (3) (2005) 37–49.

[12] S. R. Buss, Introduction to Inverse Kinematics with Jacobian Transpose, Pseudoinverse and Damped Least Squares methods (2009).
URL `http://math.ucsd.edu/~sbuss/ResearchWeb`

[13] L. C. T. Wang, C. C. Chen, A combined optimization method for solving the inverse kinematics problems of mechanical manipulators, IEEE Transactions on Robotics and Automation 7 (4) (1991) 489–499.

[14] J. Zhao, N. I. Badler, Inverse kinematics positioning using nonlinear programming for highly articulated figures, ACM Transactions on Graphics 13 (4) (1994) 313–336.

[15] J.-X. Xu, W. Wang, Y. Sun, Two optimization algorithms for solving robotics inverse kinematics with redundancy, Journal of Control Theory and Applications 8 (2) (2010) 166–175.

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

[17] F. Asano, M. Yamakita, Virtual gravity and coupling control for robotic gait synthesis, IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans 31 (6) (2001) 737–745.

[18] J.-O. Kim, P. Khosla, Real time obstacle avoidance using harmonic potential functions, in: Int. Conf. on Robotics and Automation, 1991, pp. 790–796.

[19] W.-J. Cho, D.-S. Kwon, A sensor-based obstacle avoidance for a redundant manipulator using a velocity potential function, in: 5th IEEE International Workshop on Robot and Human Communication, 11-14 Nov 1996, 1996, pp. 306–310.

[20] F. Blanchini, G. Fenu, G. Giordano, F. A. Pellegrino, Inverse kinematics by means of convex programming: Some developments, in: IEEE International Conference on Automation Science and Engineering (CASE2015), Gothenburg, 2015, pp. 515–520.

[21] J.-P. Merlet, Parallel robots, Springer, 2006.

[22] R. Clavel, Delta, a fast robot with parallel geometry, in: C. W. Burckhardt (Ed.), Proc of the 18th International Symposium on Industrial Robots, Springer-Verlag, New York, 1988, pp. 91–100.

[23] E. G. Gilbert, D. W. Johnson, Distance functions and their application to robot path planning in the presence of obstacles, IEEE Journal of Robotics and Automation 1 (1) (1985) 21–30.

[24] A. A. Maciejewski, C. A. Klein, Obstacle Avoidance for Kinematically Redundant Manipulators in Dynamically Varying Environments, The International Journal of Robotics Research 4 (3) (1985) 109–117.

[25] L. Sciavicco, B. Siciliano, A solution algorithm to the inverse kinematic problem for redundant manipulators, IEEE Journal on Robotics and Automation 4 (4) (1988) 403–410.

[26] M. Grant, S. Boyd, CVX: Matlab software for disciplined convex programming, version 2.1, `http://cvxr.com/cvx` (Mar. 2014).

[27] M. Grant, S. Boyd, Graph implementations for nonsmooth convex programs, in: V. Blondel, S. Boyd, H. Kimura (Eds.), Recent Advances in Learning and Control, Lecture Notes in Control and Information Sciences, Springer-Verlag Limited, 2008, pp. 95–110, `http://stanford.edu/~boyd/graph_dcp.html`.

[28] J. Mattingley, S. Boyd, Cvxgen: a code generator for embedded convex optimization, Optimization and Engineering 13 (1) (2012) 1–27.

[29] A. Domahidi, A. U. Zgraggen, M. N. Zeilinger, M. Morari, C. Jones, Efficient interior point methods for multistage problems arising in receding horizon control, in: Proceedings of the 51st IEEE Conference on Decision and Control, no. EPFL-CONF-181938, 2012.

[30] C. Jones, A. Domahidi, M. Morari, S. Richter, F. Ullmann, M. N. Zeilinger, et al., Fast predictive control: Real-time computation and certification, in: Nonlinear Model Predictive Control, Vol. 4, 2012, pp. 94–98.