

# On Integrating Manipulability Index into Inverse Kinematics Solver

Kévin Dufour, Wael Suleiman

### ▶ To cite this version:

Kévin Dufour, Wael Suleiman. On Integrating Manipulability Index into Inverse Kinematics Solver. 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2017), Sep 2017, Vancouver, Canada. 10.1109/IROS.2017.8206621 . hal-01588831

## HAL Id: hal-01588831 https://hal.science/hal-01588831

Submitted on 17 Sep 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers. L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

## **On Integrating Manipulability Index into Inverse Kinematics Solver**

Kévin Dufour and Wael Suleiman

Abstract— This paper presents a method to maximize the manipulability index of a redundant industrial manipulator while solving the Inverse Kinematics (IK) problem as an optimization problem. Even though the IK problem is a widely studied topic, the integration of the manipulability index into IK has rarely been taken into account. As the relation between this index and the joint variables of the robot is not straightforward, we have tested different formulations using approximated derivatives. Obstacle avoidance has also been considered and the effect of modifying the Cartesian trajectory during the execution of a task has been thoroughly analyzed.

Different scenarios have been conducted in simulation and have proven that our modified inverse kinematics solver is efficient to maximize the manipulability index, even with the additional constraint of obstacle avoidance. Moreover, the relaxation of the trajectory constraints leads to a greater manipulability while ensuring the motion smoothness and satisfying the robot physical limitations.

#### I. INTRODUCTION

The field of collaborative robots is widely expanding as the manufacturers realize the opportunity of flexibility that this kind of robot offers [1]. A major concern, however, is the safety of the human operators of those robots, as they are designed to operate without a security fence or even in physical interaction with humans. There are different ways of improving the safety: I)- modifying the physical properties of the robot such as the material, the weight [2] or the type of joint actuator, II)- adding a monitoring system [3] in order to make it more human-friendly, III)- integrating some safety constraints into the planning and execution of the task.

Regarding the motion planning phase of a motion, methods such as RRT [4] or PRM [5] are among the very popular ones. These methods try to find the optimal path while considering different criteria. Their main drawback is the prohibitive time of computation that restrains their main use to offline planning, however a recent research [6] on the design of specialized processors to reduce the computation time has shown promising results.

However, when trying to deal with a dynamic environment, it is interesting to generate a preliminary trajectory by motion planning algorithms to benefit from their global planning properties, and then modify it in real-time, by inverse kinematics methods for instance, during the execution. The robot in this case could react to the presence of moving obstacles, such as a human or another robot, while satisfying some constraints, for example the joint limits or a constraint on the velocity or acceleration of the end-effector. A well-known approach to solve IK problems is the reformulation as an optimization problem [7], [8]. An objective function depending on the optimization variable, often the joint velocity, is minimized or maximized subject to different linear equality and inequality constraints. Because of its general formulation, it is easy to add different constraints to meet the safety requirements, as long as it is possible to solve the optimization problem in real-time.

Since the robot is executing a task in a dynamic environment, an important feature is the ability of reacting correctly to unforeseen events. This ability is reflected by the manipulability index, related to the singularities of the Jacobian matrix [9]. When this index becomes small and tends to zero, it means that the robot is close to a singularity and its ability to move away from that pose will be reduced. It is necessary, therefore, to maximize the manipulability index, which can be done by integrating it into the objective function of the IK optimization problem.

The main contribution of this paper is proposing a method to solve the IK problem while integrating the manipulability index, and mainly giving insights into the practical implementation of the new solver as well as the possible strategies to maximize the manipulability index during the task execution.

This paper is organized as follows: Section II presents the formulation of inverse kinematics as an optimization problem with respect to joint limits, obstacle avoidance and trajectory relaxation. In Section III, a new objective function to maximize the manipulability index is proposed. The method is validated through simulation experiments in Section IV.

#### II. INVERSE KINEMATICS PROBLEM FORMULATION

#### A. General Definition

The inverse kinematics problem is generally written in the following form, called quadratic programming (QP) problem:

$$\min_{\dot{\boldsymbol{q}}} \frac{1}{2} \dot{\boldsymbol{q}}^T Q \dot{\boldsymbol{q}} \tag{1}$$

subject to

$$J\dot{\boldsymbol{q}} = \dot{\boldsymbol{r}} \tag{2}$$

$$\hat{\dot{\boldsymbol{q}}}^{-} \leq \dot{\boldsymbol{q}} \leq \hat{\dot{\boldsymbol{q}}}^{+} \tag{3}$$

$$\boldsymbol{b}^{-} \le A \boldsymbol{\dot{q}} \le \boldsymbol{b}^{+} \tag{4}$$

where  $\dot{\boldsymbol{q}} \in \mathbb{R}^n$  is the joint velocity, Q a diagonal and positive semi-definite weighting matrix, J the Jacobian matrix, A,  $\boldsymbol{b^+}, \boldsymbol{b^-}$  the matrix and vectors defining the linear inequality constraints.  $\dot{\boldsymbol{r}}$  is derived from the Cartesian trajectory  $\boldsymbol{r}$  generated by an offline planner.

K. Dufour and W. Suleiman are with Electrical and Computer Engineering Department, Faculty of Engineering, University of Sherbrooke, Canada. {Kevin.K.Dufour, Wael.Suleiman}@USherbrooke.ca

Equation (3) expresses in a single form the joint range and velocity limits using the generalized boundaries  $\hat{q}^-$ ,  $\hat{q}^+$ defined by the velocity damper formula [10], that formula combines the joint position and velocity so that the velocity smoothly decreases when approaching the joint boundaries and does not violate the limits.

The inequality constraint (4) can be used for collision avoidance by using the velocity damper definition, introduced initially in [11]: when the distance d between two convex objects, one being a part of the robot ( $\mathcal{O}_1$ ) and the other an obstacle ( $\mathcal{O}_2$ ), becomes smaller than a constant  $d_i$ , the following formula is applied:

$$-\boldsymbol{n}^{T}J(q,p_{1})\boldsymbol{\dot{q}} \leq \xi \frac{d-d_{s}}{d_{i}-d_{s}} \text{ if } d \leq d_{i}$$
(5)

where n is the unit vector between the closest points  $p_1$ and  $p_2$  on the two objects, as described in Fig. 1,  $\xi$  is a positive coefficient and  $d_s$  is a security distance that cannot be crossed. Let k be the number of distances between the robot and a set of obstacles satisfying  $d \leq d_i$ , the constraint (4) can then be rewritten as:

$$A\dot{q} \leq b^+$$

where  $A \in \mathbb{R}^{k \times n}$  and  $b^+ \in \mathbb{R}^k$ , such as their  $j^{th}$  line is defined by:

$$A_{j} = -\boldsymbol{n}_{j}^{T} J_{j}(q, p_{1})$$

$$\boldsymbol{b}_{j}^{+} = \xi \frac{d_{j} - d_{s}}{d_{i} - d_{s}}$$
(6)



Fig. 1. Parameters for collision avoidance

It should be pointed out that this problem can be adapted to consider the acceleration or even jerk limits of the joints by replacing the optimization variable to the relevant derivative of q and rewriting the constraints consequently [10].

#### B. Relaxed Problem Formulation

The particularity of a redundant robot is that the solution of the IK problem is not unique, thus the best solution among those admissible is chosen by the optimization algorithm. However, in case of collision avoidance, there might be no solution if the obstacle is in collision with every configuration corresponding to a given pose in the trajectory. It is then interesting to allow the robot to deviate from the pre-defined Cartesian trajectory, if the path taken between the initial and the final poses is not important. A task like "pick-and-place" could be an example. This can be achieved by adding a slack variable vector  $\boldsymbol{\delta}$  to the optimization variable, which was previously  $\dot{\boldsymbol{q}}$  and then becomes  $\mathcal{Z} = \begin{bmatrix} \dot{\boldsymbol{q}} & \boldsymbol{\delta} \end{bmatrix}^T$ . Indeed, if we write the Cartesian path constraint (2) with the new variable:  $J\dot{\boldsymbol{q}} = \dot{\boldsymbol{r}} - \boldsymbol{\delta}$ , it appears that the trajectory is relaxed and the new one is, in velocity terms,  $\dot{\boldsymbol{r}} - \boldsymbol{\delta}$ . The IK problem formulation thus becomes [12], [10]:

$$\min_{\mathcal{Z}} \frac{1}{2} \mathcal{Z}^T Q_{\mathcal{Z}} \mathcal{Z}$$

 $egin{aligned} \mathcal{J}\mathcal{Z} &= \dot{m{r}} \ \mathcal{A}\mathcal{Z} &\leq m{b}^+ \ \mathcal{Z}^- &< \mathcal{Z} &< \mathcal{Z}^+ \end{aligned}$ 

$$\begin{aligned} \mathcal{Z} &= \begin{bmatrix} \dot{\boldsymbol{q}} \\ \boldsymbol{\delta} \end{bmatrix}, \quad \mathcal{Q}_{\mathcal{Z}} = \begin{bmatrix} Q & 0 \\ 0 & Q_{\delta} \end{bmatrix}, \quad \mathcal{J} = \begin{bmatrix} J & I \end{bmatrix} \\ \mathcal{A} &= \begin{bmatrix} A & 0 \end{bmatrix}, \quad \mathcal{Z}^{+} = \begin{bmatrix} \hat{\boldsymbol{q}}^{+} \\ \boldsymbol{\delta}^{+} \end{bmatrix}, \quad \mathcal{Z}^{-} = \begin{bmatrix} \hat{\boldsymbol{q}}^{-} \\ \boldsymbol{\delta}^{-} \end{bmatrix} \end{aligned}$$

The amplitude of the deviation is controlled by the bounds  $\delta^+$  and  $\delta^-$  of the slack variable and its associated weighting matrix  $Q_{\delta}$ , which is defined as follows:

$$Q_{\delta}(t) = f_e(t) \boldsymbol{I}_n$$

w

subject to

with

$$f_e(t) = \begin{cases} f_1(t) & \text{if } 0 \le t \le T_\rho \\ Q_{low} & \text{if } T_\rho \le t \le T_f - T_\rho \\ f_2(t) & \text{otherwise} \end{cases}$$

with  $I_n \in \mathbb{R}^{n \times n}$  the identity matrix,  $f_1$  and  $f_2$  polynomial functions of degree 5 defined to obtain a profile similar to Fig. 2, allowing a smooth, continuous behavior for  $\delta$ , and as a result for  $\dot{q}$ , by setting their first and second derivatives to zero at their limits. Their value,  $Q_{high}$ , is very high at the beginning and at the end of the task so  $\delta$  is close to 0. Refer to [12] for a detailed description and explanation.



Fig. 2. Profile of constraint stiffness

#### III. MANIPULABILITY INDEX

#### A. Theoretical Definition

In order to have a motion that has enough admissible movements to escape unforeseen events, it is important to stay away as far as possible from the robot joint singularities. This ability, named manipulability, is defined in [9] as:

$$m(\boldsymbol{q}) = \sqrt{\det(JJ^T)} = \sigma_1 \sigma_2 \cdots \sigma_n \tag{7}$$

with J the Jacobian matrix of the robot's end-effector for the configuration q and  $(\sigma_i)_{1 \le i \le n}$  the singular values of J.

This index is particularly useful because when it increases, the manipulability of the robot does as well, and when it gets close to 0, the robot is close to a singularity. Other definitions, however, do not have this property: usually they only tell if the robot is close to a singularity but they do not give any information about the relative distance to the singularity and then cannot be used to get away from it.

#### **B.** Practical Implementation

By integrating the manipulability index into the IK formulation, the following optimization problem is obtained:

$$\min_{\dot{\boldsymbol{q}},\boldsymbol{q_t}} \frac{1}{2} \, \dot{\boldsymbol{q}}^T Q \, \dot{\boldsymbol{q}} - \alpha \, m(\boldsymbol{q_t}) \tag{8}$$

subject to

$$J\dot{q} = \dot{r}$$
  
 $\hat{\dot{q}}^{-} \leq \dot{q} \leq \hat{\dot{q}}^{+}$   
 $b^{-} \leq A\dot{q} \leq b^{+}$ 

where  $\alpha$  is a positive coefficient, preceded by a negative sign in order to maximize  $m(q_t)$ .

As  $m(q_t)$  is a nonlinear function in q, Eq (8) becomes a nonlinear optimization problem which is hard to solve within a fixed time and therefore the hard real-time constraints might not be satisfied.

In order to transform Eq (8) into a QP problem, the manipulability index is linearized as follows:

$$m(\boldsymbol{q_t}) = m(\boldsymbol{q_{t-1}}) + \nabla m^T \Delta \boldsymbol{q} + \varepsilon \frac{1}{2} \Delta \boldsymbol{q}^T H_m \Delta \boldsymbol{q}$$
  
=  $m(\boldsymbol{q_{t-1}}) + T \nabla m^T \dot{\boldsymbol{q}} + \varepsilon \frac{1}{2} T^2 \dot{\boldsymbol{q}}^T H_m \dot{\boldsymbol{q}}$  (9)

where  $H_m$  is the Hessian matrix of m,  $\nabla m$  is its gradient, T is the control loop sampling time of the robot and  $\varepsilon$  is a constant equal to 0 (first-order approximation) or 1 (second-order approximation).

By replacing (9) into (8), the following equivalent QP problem is obtained:

$$\min_{\dot{\boldsymbol{q}}} \frac{1}{2} \, \dot{\boldsymbol{q}}^T Q \, \dot{\boldsymbol{q}} - \alpha \, \left( T \, \nabla m^T \, \dot{\boldsymbol{q}} + \varepsilon \frac{1}{2} T^2 \, \dot{\boldsymbol{q}}^T H_m \, \dot{\boldsymbol{q}} \right) \quad (10)$$

subject to

$$J\dot{q} = \dot{r}$$
  
 $\dot{\hat{q}}^{-} \le \dot{q} \le \dot{\hat{q}}^{+}$   
 $b^{-} \le A\dot{q} \le b^{+}$ 

**-** •

Because of the definition of m (7), it is not evident to find a direct analytical relationship between m and q and by consequence a proper definition for  $\nabla m$  and  $H_m$ . Thus, the gradient of m is approximated numerically by:

$$(\nabla m)_i = \frac{\partial m}{\partial q_i}$$
$$= \frac{m(\boldsymbol{q} + \delta q_i \boldsymbol{E}_i) - m(\boldsymbol{q} - \delta q_i \boldsymbol{E}_i)}{2\delta q_i}$$

where  $(\nabla m)_i$  is the *i*<sup>th</sup> element of vector  $\nabla m$ , and  $E_i \in \mathbb{R}^n$ ,

and the Hessian matrix by:

$$(H_m)_{i,j} = \frac{\partial^2 m}{\partial q_i \partial q_j} = \frac{(\nabla m)_j \left( \boldsymbol{q} + \delta q_i \boldsymbol{E}_i \right) - (\nabla m)_j \left( \boldsymbol{q} - \delta q_i \boldsymbol{E}_i \right)}{2\delta q_i}$$

where:

$$(\nabla m)_j (\boldsymbol{q} + \delta q_i \boldsymbol{E}_i) = \frac{m(\boldsymbol{q} + \delta q_i \boldsymbol{E}_i + \delta q_j \boldsymbol{E}_j) - m(\boldsymbol{q} + \delta q_i \boldsymbol{E}_i - \delta q_j \boldsymbol{E}_j)}{2\delta q_j}$$
$$(\nabla m)_j (\boldsymbol{q} - \delta q_i \boldsymbol{E}_i) = \frac{m(\boldsymbol{q} - \delta q_i \boldsymbol{E}_i + \delta q_j \boldsymbol{E}_j) - m(\boldsymbol{q} - \delta q_i \boldsymbol{E}_i - \delta q_j \boldsymbol{E}_j)}{2\delta q_j}$$

Using these formulations, it is then easy to compute the objective function since only the manipulability index of different configurations is computed, which indirectly implies, according to (7), Jacobian matrix calculations.

#### **IV. SIMULATION RESULTS**

The optimization problem can be written in different ways depending on the desired constraints. In order to evaluate the performance of each definition, several scenarios have been carried out, showing the efficiency of the different approaches. In the first scenario, the manipulability index is already high, and we aim at proving that we can improve it nonetheless while precisely tracking the desired Cartesian trajectory all along and even avoiding an obstacle. The second scenario focus on the case of a trajectory getting close to a singularity and the effect of our algorithm to get the robot away from that singularity thanks to the robot redundancy. Finally, we will allow the relaxation of the desired trajectory in both scenarios and present its benefits.

The different approaches have been tested in simulation using a Baxter research robot from Rethink Robotics. A preliminary test has been conducted offline to measure the evolution of the manipulability index of the robot depending on the joint positions and to find an approximation of its maximal value. Thus, all the manipulability index shown in this section will be expressed relatively to that maximal value. The matrix Q in (10) is equal to the identity matrix, and T = 0.01s. The optimization problem is solved by qpOASES [13], which has been proven to be efficient and real-time compatible.

#### A. First Scenario

In this first scenario, the robot executes a simple cubic trajectory in the Cartesian space along the vertical axis. In this case, the slack variable  $\delta$  (the relaxation variable) is not considered and the robot should strictly follow the trajectory.

First, the order of approximation of m will be studied by comparing the first-order ( $\varepsilon = 0$ ), Fig. 3, and the second-order ( $\varepsilon = 1$ ), Fig. 4, for different values of the coefficient  $\alpha$ ,  $\alpha = 0$  meaning that the manipulability index is not maximized.

Fig. 3 shows the impact of  $\alpha$  and that there is an optimal value for this coefficient: a maximum normalized value of m, 0.91, is reached for  $\alpha = 5000$ . However increasing  $\alpha$  beyond 5000 makes the manipulability profile decrease, as, for instance, the resulting profile for  $\alpha = 25000$  shows, this behavior can help find the optimal coefficient for a given trajectory. Moreover when  $\alpha$  becomes too high, the QP problem (10) becomes infeasible.

Fig. 4 compares the results obtained when considering the second-order approximation to the optimal result found with the first-order approximation, it shows that considering the second-order approximation does not improve the optimal result obtained by only considering the first-order approximation. Moreover, the calculation of the Hessian implies the evaluation of  $4 \times n \times n$  Jacobian matrices which seriously increases the total calculation time. For that reason, the manipulability index will thereafter be approximated by only its first-order formulation ( $\varepsilon = 0$ ).



Fig. 3. Normalized manipulability index evolution while executing the task in scenario 1, in this case only the first-order approximation of m is used

The case of collision avoidance is then addressed, as shown in Fig. 7 a spherical obstacle that is located close to the elbow during the movement has been added.

The minimal distance between the robot and the obstacle is presented in Fig. 5 for different cases considering alternatively the obstacle constraint and/or the manipulability optimization. It shows that the robot successfully avoids the obstacle and does not cross the security distance  $d_s$  while still be able to reach the goal. However, Fig. 6 also points that the maximization of the manipulability index and avoiding the collision with the obstacle are two opposite constraints,



Fig. 4. Normalized manipulability evolution during task depending on coefficient in scenario 1 using second-order approximation of m

it shows that when the arm is far enough from the obstacle, the manipulability increases but when the robot becomes too close to the obstacle, the distance constraint has the priority and the manipulability index has the same value as for the non-optimized case.

Fig. 7 illustrates the effect of the collision avoidance constraint on the configuration of the robot for a given time during the motion: the elbow, which becomes too close to the obstacle represented by a green sphere, is pushed away to satisfy the security distance. It should be noticed that the end-effector has the same pose in both configurations as the Cartesian trajectory constraint is always satisfied.



Fig. 5. Minimal distance between the robot and the obstacle with different constraints

#### B. Second Scenario

In this scenario, a trajectory is generated offline to represent a critical situation as the manipulability index decreases during the motion and comes close to a singularity.

Fig. 8 shows the evolution of the manipulability without optimization and the effect of our algorithm depending on the coefficient  $\alpha$ . It shows that our method is generally able to increase the normalized manipulability but has a very slight impact near the singularity.



Fig. 6. Normalized manipulability evolution in obstacle avoidance scenario



(a) Without obstacle constraint (b) With obstacle constraint

Fig. 7. Effect of the obstacle avoidance constraint on the configuration of the robot for a given state

In this case, the improvement of manipulability index is limited, this is mainly because only the redundancy is exploited and thus restricted by the robot physical limits. However, this limitation can be overcome by using the trajectory relaxation as shown in the following subsection.



Fig. 8. Normalized manipulability evolution while executing the task in scenario  $\mathbf{2}$ 

#### C. Study of the Trajectory Relaxation

Both previous scenarios have been tested using the trajectory relaxation. For the first scenario, Fig. 9 shows that the relaxation method allows an increase of about 0.05 of the normalized manipulability index maximum to a value of 0.96. Moreover, the manipulability increases much faster since it reaches 0.80 in 0.87s instead of 1.72s without the slack variable. Fig. 10 shows the reference Cartesian trajectory of the end-effector and the modified one, pointing out the small deviations and showing that the robot resumes the task at the end to successfully reach the goal pose for the end-effector.



Fig. 9. Normalized manipulability evolution in scenario 1 while relaxing the constraints



Fig. 10. Reference and computed trajectories in scenario 1 while relaxing the constraints

For the second scenario, Fig. 11 shows a very important improvement of the manipulability index thanks to the relaxation. Thus, near the singularity, the manipulability index becomes close to 0.93 for  $\alpha = 90000$  and to 0.82 for  $\alpha = 15000$ , which is a significant improvement compared to the value obtained in Section IV-B, proving that the robot succeeds to completely escape from the singular configuration. Fig. 12 presents the influence of  $\alpha$  on the relaxation of the trajectory, and as it can be seen increasing  $\alpha$  increases the deviations. However, even limited deviations as with  $\alpha = 15000$  leads to a high profile of manipulability.

Fig. 13 illustrates the difference between the final configurations in scenario 2 in two cases: I)- without relaxation nor manipulability optimization, II)- with relaxation and manipulability optimization, showing that the configuration of the robot has been changed to a more convenient one regarding the manipulability, while having the same endeffector pose.



Fig. 11. Normalized manipulability evolution in scenario 2 while relaxing the constraints



Fig. 12. Reference and computed trajectories in scenario 2 while relaxing the constraints

#### V. CONCLUSION AND FUTURE WORK

In this paper, we presented a new method to integrate the manipulability index maximization into an optimizationbased IK solver. When considering the first-order approximation of the manipulability index, our algorithm does not require heavy calculations and is real-time compatible, it can be combined with other constraints without leading to infeasibility. Moreover, an implementation of our method while considering the trajectory relaxation has proven to give excellent results in maximizing the manipulability index with a reasonable deviation in the desired trajectory of the endeffector.

However, it should be noted that with this approach the optimal value of coefficient  $\alpha$ , which allows the maximization of manipulability index, varies depending on the trajectory. For repetitive tasks the optimal value of  $\alpha$  can be determined offline, however our experiments with the Baxter robot pointed out that the manipulability index has been improved whatever the value of  $\alpha$  in the interval  $[0, \frac{250}{T}]$ .

Future work will focus on integrating other securityrelated constraints such as the avoidance of moving obstacles, taking into account human awareness and dealing with the presence of a human beside the robot. Moreover, we are





(a) Without relaxation nor manipulability optimization

(b) With relaxation and manipulability optimization



also interested in optimizing the calculation time to ensure the real-time property as well as testing our algorithm in different scenarios with the real Baxter robot.

#### ACKNOWLEDGMENT

This research is supported by a discovery grant (Prof. Wael Suleiman) from the Natural Sciences and Engineering Research Council of Canada (NSERC).

#### REFERENCES

- J. Krüger, T. K. Lien, and A. Verl, "Cooperation of human and machines in assembly lines," *CIRP Annals - Manufacturing Technology*, vol. 58, no. 2, pp. 628–646, 2009.
- [2] A. Albu-Schäffer, S. Haddadin, C. Ott, A. Stemmer, T. Wimböck, and G. Hirzinger, "The DLR lightweight robot: design and control concepts for robots in human environments," *Industrial Robot: An International Journal*, vol. 34, no. 5, pp. 376–385, 2007.
- [3] D. Kulić and E. Croft, "Pre-collision safety strategies for human-robot interaction," Autonomous Robots, vol. 22, no. 2, pp. 149–164, 2007.
- [4] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *The International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.
- [5] L. E. Kavraki, P. Švestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.
- [6] S. Murray, W. Floyd-Jones, Y. Qi, G. Konidaris, and D. J. Sorin, "The microarchitecture of a real-time robot motion planning accelerator," in 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), Oct 2016, pp. 1–12.
- [7] a. Escande, N. Mansard, and P.-B. Wieber, "Hierarchical quadratic programming: Fast online humanoid-robot motion generation," *The International Journal of Robotics Research*, vol. 33, no. 7, pp. 1006– 1028, 2014.
- [8] W. Suleiman, F. Kanehiro, and E. Yoshida, "Infeasibility-free inverse kinematics method," 2015 IEEE/SICE International Symposium on System Integration, SII 2015, no. 1, pp. 307–312, 2016.
- [9] T. Yoshikawa, "Manipulability of Robotic Mechanisms," *The International Journal of Robotics Research*, vol. 4, no. 2, pp. 3–9, jun 1985.
- [10] W. Suleiman, "On inverse kinematics with inequality constraints: new insights into minimum jerk trajectory generation," *Advanced Robotics*, vol. 30, no. 17-18, pp. 1164–1172, 2016.
- [11] B. Faverjon and P. Tournassoud, "A local based approach for path planning of manipulators with a high number of degrees of freedom," *IEEE International Conference on Robotics and Automation*, pp. 1152–1159, 1987.
- [12] F. Kanehiro, W. Suleiman, K. Miura, M. Morisawa, and E. Yoshida, "Feasible pattern generation method for humanoid robots," in 2009 9th IEEE-RAS International Conference on Humanoid Robots. IEEE, dec 2009, pp. 542–548.
- [13] H. Ferreau, C. Kirches, A. Potschka, H. Bock, and M. Diehl, "qpOASES: A parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, no. 4, pp. 327–363, 2014.