Manipulability Optimization for Trajectory Generation

Luis Guilamo[1] James Kuffner[1,2] Koichi Nishiwaki[1] Satoshi Kagami[1]
[1]*The National Institute of Advanced Industrial Science and Technology*
Digital Human Research Center, Tokyo, Japan
[2]*The Robotics Institute, Carnegie Mellon University, USA*
{*luis.guilamo, j.kuffner, k.nishiwaki, s.kagami*}*@aist.go.jp*

***Abstract*— In this paper, we present an algorithm for manipulability based trajectory generation for any serial manipulator that has an inverse kinematic model that can obtain all solutions. Our strategy is a search-based approach that analyzes candidate configurations at discrete points along the workspace trajectory. Given such a model we prove the configuration space trajectories generated by our method are optimal within the limit of the discretization of the workspace trajectory.**

I. INTRODUCTION

Robot manipulators play an important role for a wide range of industrial and domestic applications. Many of these applications require interaction or contact with objects in the world. In such cases, highly compliant motion is desirable to react to the applied forces. One method for measuring the compliance or the ability to react "gracefully" to applied forces at a given configuration is *manipulability* [**?**]. This provides a quantitative measure of manipulating ability, where higher manipulability will lead to dextrous configurations staying clear of singularities.

This type of necessity can be seen greatly in the realm of humanoid robotics. To continue advancement for many applications within humanoid robots it has become essential that the robot be able to interact with the world, humans and objects alike. An example of this interaction where compliance is requisite can be seen in [**?**], where the robot performs navigation planning by clearing objects from its path. This type of problem motivates research to compute configuration-space trajectories with high compliance or high manipulability such that it can be in the best configuration possible along a trajectory to apply force control. In this paper we present a manipulability based global optimization algorithm for trajectory generation.

Traditionally, trajectory optimization, takes an existing trajectory obtained by some motion planning algorithm and optimizes it by the desired criteria. Our method differs from the average trajectory optimization approach in that we generate a configuration-space trajectory directly from a workspace trajectory input. Based on a manipulability optimization criteria our method reduces to a search problem of a mapping between the configuration-space and workspace. In Section IV we prove the global optimality of our methods criteria and that the provided algorithm is resolution complete. Figure 1 shows an image of a single configuration obtained an experiment run on the HRP2-DH with its manipulator following a straight-line trajectory at the boundary of the reachable workspace.

Fig. 1. HRP2-DH extending to the edge of its reachable workspace

Generated by the presented algorithm the robot is using its chest yaw and pitch angle to increase the manipulability at its end effector in a region of very low manipulability.

The importance of robot world interaction has been well noted throughout the robotics community and a number of trajectory optimizations currently work to enhance motions in the configuration-space. Such methods will be discussed next in Section II. The rest of the paper is organized as follows: Section III defines any problem assumptions; Section IV presents the algorithm for trajectory generation; Section V describes experimental results, and finally Section VI gives a summary and discussion.

II. RELATED WORK

Trajectory optimization and biasing has had a wide range of application for enhancing task specific world-robot interactions. In graphics there has been attempts to directly map the configuration-space to the workspace by using a database of realistic looking postures. In [**?**], [**?**], realistic postures were stored into a discretized workspace, these postures are adapted to positions along a workspace trajectory to generate the desired 3D character posture trajectory biased to have realistic looking postures.

Trajectory optimization methods in within robotics include the "bang-bang" approach to time-optimality [**?**], [**?**]. Low energy global optimization technique can be found in [**?**]. A Markovian Networks approach to industrial robot optimal trajectory generation with respect to distance measures and angular velocity and acceleration can found in [**?**].

More closely related to our approach, in the realm of compliant trajectory generation, is the *Operational Space Formulation* [**?**], [**?**]. The main idea here is to use a gradient descent approach with null space optimizations of the Jacobian to do prioritized posture control. Similarly, a prioritized method to full body compliant motion can be found in [**?**].

Similar to the aformentioned methods we present a method to enhance world-robot interaction. Many manipulator performance measures have been proposed such as in [**?**], [**?**] a dynamic measure for manipulator performance is considered. We focus on a manipulability based criteria that can provide trajectories over all better for handling compliant control at any point along the trajectory. Our approach differs from these trajectory optimization methods in that we build a configuration-space trajectory from scratch working completely in the workspace. We provide algorithm properties that yield optimaly generated trajectories which will be demonstrated in further detail in the following sections.

III. Problem Formulation

As previously mentioned, the algorithm is required to search a sequence of candidate configurations at discrete points along a workspace trajectory. For some workspace trajectory $T_p = \{p_0, ..., p_n\}$ where n defines the number of discrete points along T_p and $p \in R^6$. We assume that for some manipulator A there exists some inverse kinematic model K defined as:

$$\{\forall p \in T_p \quad \exists \quad K(p_i)|K(p_i) = Q_i\}$$

where $Q_i = \{q_{i_0}, ..., q_{i_k}\}$ is the set of all configurations at point p_i where $0 \leq k \leq \infty$. Since at each point p_i there exist a candidate set of possible configurations Q we define the set of all possible configurations along T_p as T_Q where $T_{Q_i} = Q_i$.

A valid configuration-space trajectory within T_Q is a continuous configuration set T_{cont} where each $q_i \in T_{cont}$ has end effector at the corresponding position $p_i \in T_p$. T_{cont} is continuous if and only if for any configuration pair $\{q_i, q_{i+1}\} \in T_{cont}$ where $0 \leq i < n$ has angular configurations $\{\theta_i, \theta_{i+1}\}$ where $(\theta_i - \epsilon) < \theta_{i+1} < (\theta_i + \epsilon)$ where ϵ defines the largest angular displacement by any joint in A and $\theta_i = \{\theta_{i_0}, ..., \theta_{i_m}\}$ where m denotes the degrees of freedom (DOF) in A. Therefore, the set of all valid configuration-space trajectories, V is the set of all existing continuous trajectories within T_Q.

Lemma 1. $T_{opt} \in V$

Proof: Since V is defined as the set of all possible continuous configuration sets, if $T_{opt} \notin V$ then T_{opt} is not continuous and therefore is not a Valid trajectory. ■

Each T_{cont} has a related manipulability curve $M(T_{cont})$. The goal is to maximize the global minimum of the manipulability curve. Thus, following Lemma 1, the optimal manipulability based configuration-space trajectory reduces to the form:

$$\{ \quad T_{opt} = T_{cont_a} \in V|$$
$$M(T_{cont_a}).globalMin \geq M(T_{cont_b}).globalMin$$
$$\forall(b \neq a) \quad \}$$

IV. Trajectory Generation

Given some input workspace trajectory, T_p, we build a trajectory of all candidate configurations, T_Q. Since T_Q is the set of all possible configurations it may have an infinite number of elements, thus for this algorithm we consider a finite set $T_{Q_{calc}} \subseteq T_Q$. The algorithm reduces to a greedy search of $T_{Q_{calc}}$.

The first step is to obtain the discrete starting position p_i and starting configuration q_{start} defined as some $q_{i_j} \in T_{Q_{calc_i}}$. Since the goal is to maximize the global minima of the manipulability trajectory we choose the starting p_i with the T_{Q_i} that contains the configuration with the maximum possible value for a global minima. This configuration $q_{start} \in T_{Q_{calc_i}}$ can be defined as:

$$(\forall M(T_{Q_{calc_i}}).globalMax).globalMin$$

That is taking the configuration with maximum manipulability at each p_i and starting at the configuration with minimum manipulability contained within that set. To state more simply, the algorithm begins with the best possible global min that can exist for any trajectory derived from the existing $T_{Q_{calc}}$.

Given a starting configuration we want to build T_{opt} by growing a T_{cont} from q_{start} as described in Section III. By starting at configuration q_{start} we grow to configuration set $T_{Q_{calc_n}}$ and then from $q_{start} \rightarrow T_{Q_{calc_0}}$. Therefore, the starting configuration for the search process varies in an attempt to maintain the best possible global minima.

Since we are applying a manipulability based optimization we don't only want to grow a T_{cont} but constrain the transition from $q_{i_j} \rightarrow q_{i+1_j}$. The goal is to maximize the minima of the manipulability trajectory and therefore the continuous q_{i+1_j} with highest manipulability value within the set T_{Q_i} is considered and is inserted to the current growing trajectory if and only if $M(q_{i+1j}) \geq M(q_{start})$. Therefore, a q_{i+1_j} cannot be found with the presented criteria then a new q_{i_j} maintaining the same criteria with q_{i-1_j}. This search is iterated through $T_{Q_{calc}}$ until a T_{cont} is created.

If the presented manipulability threshold criteria fails a new q_{start} needs to be chosen. This configuration is chosen to be the $q \in T_{Q_{calc}}$ where:

$$M(q_{start}) - M(q) \leq M(q_{start}) - M(q_{ij})$$
$$\forall q_{ij} \in T_{Q_{calc}} \neq q_{start}$$

With this new q_{start} the algorithm iterates till it finds a T_{cont} as before. This T_{cont} is defined as $T_{opt,calc}$. It is important to note that through the definitions of these sets, the generated trajectory $T_{opt,calc}$ take on the following properties:

Trajectory	$T_{Q_{calc}}$ size	Num. Positions	Branching Factor	Calc. Time
Circular	59,561	50	~4	1.5min
Line	33,902	20	~6	20sec

TABLE I

PERFORMANCE RESULTS FOR CIRCULAR AND LINE TRAJECTORY.

Lemma 2. If only a single continuous configuration-space trajectory exist it will be represented by $T_{opt,calc}$

Proof:

If the magnitude of $V = 1$, the existing continuous trajectory T_{only} must have a configuration $q_{min} \in T_Q$ such that $M(q_{min}) = M(T_{only}).globalMin$. Since the algorithm works to generate a T_{cont} starting with q_{start} being the maximum possible global minima within T_Q then $M(q_{min}) \leq M(q_{start})$. By definition, all $q_i \in T_Q$ where $M(q_i) \leq M(q_{start})$ are checked as a possible global minima before failing to generate a trajectory. Therefore, for some $\exists q_i = q_{min} \Rightarrow T_{cont} = T_{only}$. ■

Lemma 3. The trajectory found, $T_{opt,calc}$ with manipulability trajectory $M(T_{opt,calc})$ has the maximum global minimum of all $T_{cont} \in V$

Proof:

If there exist some continuous configuration set T_{missed} such that $M(T_{missed}).globalMin > M(T_{opt,calc}).globalMin$ then $q_{min} \in T_{missed}$ where $M(q_{min} = M(T_{missed}).globalMin$ must have value $M(q_{min}) > M(q_{start}$ since by definition $\forall q_i \in T_Q$ where $M(q_i) \leq M(q_{start})$ are checked. However, by definition q_{start} has the maximum possible manipulability of all global min therefore there does not exist any such $q_m in$ and thus follows no T_{missed}. ■

Theorem 1. For some input workspace trajectory $T_p = \{p_0, ..., p_n\}$ as $n \rightarrow \infty$ and as $Q_{calc} \rightarrow Q$, the calculated optimal trajectory $T_{opt,calc} \rightarrow T_{opt}$.

Proof:

As the number of input configurations $T_{Qcalc} \rightarrow T_Q$ and $T_p = \{p_0, ..., p_n\}$ as $n \rightarrow \infty$ the number of considered configurations and valid trajectories become the actual complete sets. Therefore, by Lemma 3 $T_{opt,calc} = T_{opt}$. ■

V. EXPERIMENTS

To test the quality of our method presented in Section IV, we generated highly manipulable trajectories on the Digital Human version of the Humanoid Robot Platform 2 (HRP2-DH). For the following experiments we utilized 9 degrees of freedom of the HRP2-DH incorporating the chest pitch and yaw with the right arm, from the shoulder to the wrist with the wrist being considered as the end-effector position. Experemental specifics can be found in Table I which will be further described in the following paragraphs.

One of many experiments that were run, was to have the robot manipulator follow a circular trajectory oriented vertical in the workspace on a parallel plane in front of the robot. This trajectory can be found in Figure 2. The hand orientation was maintained at a 90 degrees horizontal relative to the waist with a continuity delta defined at two degrees. The algorithm defined in Section IV was used to generate a manipulability optimized configuration-space trajectory. The optimized manipulability curve can be found in Figure 3. Images of four configuration along the circular trajectory applied to the robot can be seen in Figure 6. The top row of images are a frontal view of the HRP2-DH while the bottom row gives the same configurations from a side view better showing the use of

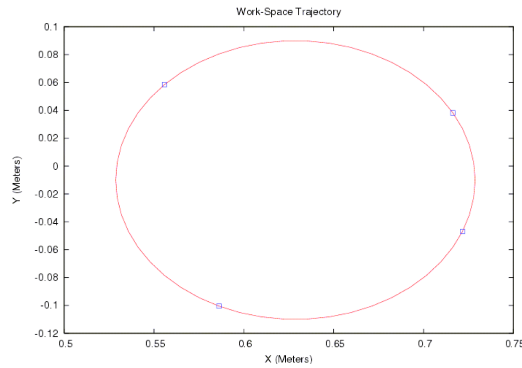

Fig. 2. Circular workspace trajectory followed by HRP2-DH.

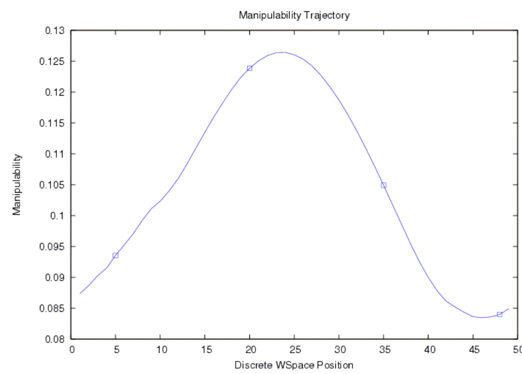

Fig. 3. Manipulability plot along the generated circular trajectory.

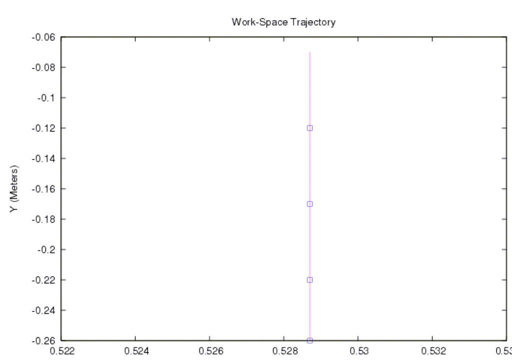

Fig. 4. Straight-line workspace trajectory followed by HRP2-DH.

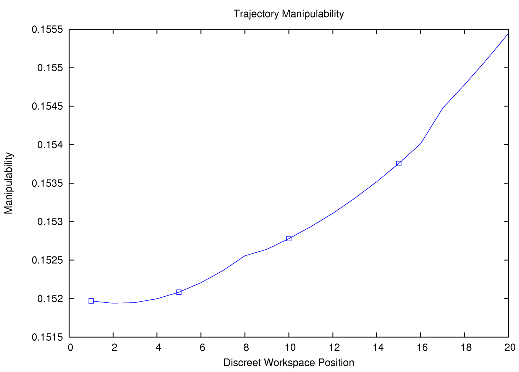

Fig. 5. Manipulability plot along the generated straight-line trajectory.

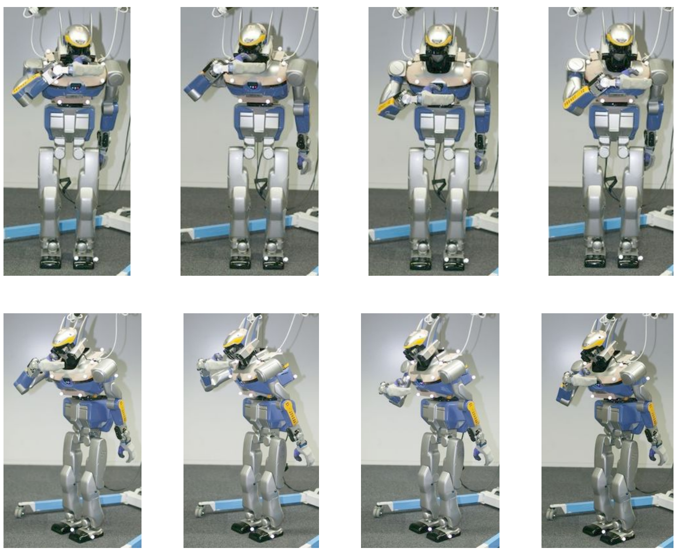

Fig. 6. Front view *(top)* and side view *(bottom)* of the generated circular workspace trajectory.

the chest pitch and yaw angles. These trajectories were also developed in simulation, with corresponding images found in Figure 7. The purpose of this experiment was to demonstrate the robots use of the chest yaw and pitch to maintain the end-effector in a high region of manipulability in front of the chest. These can be better seen in the aerial view of the simulation results Figure 8(bottom).

For this experiment $306,306$ configurations were considered at each discrete position. Of all configurations considered $59,561$ valid sample configurations were entered into $T_{Q_{calc}}$ and taken into account for 50 discrete workspace positions along a $20cm$ diameter circular path. At each of these discrete points there existed anywhere from ~$800 - 2000$ configurations. When generating the trajectory $9,407$ new starting configurations were attempted and $36,596$ branches were made (~4 branches per new starting configuration) before realizing the optimal trajectory within the given data set with the defined delta for continuous angular displacement. It took ~1.5min to generate this trajectory.

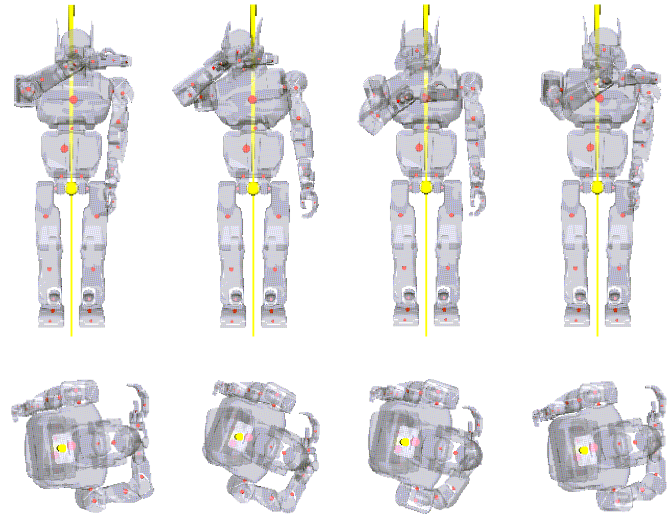

Fig. 7. Front view *(top)* and top view *(bottom)* of the simulation results for the circular workspace trajectory.

Another experiment was done to demonstrate that this trajectory generation method can obtain trajectories along the bounding box of the manipulator reachable workspace. This is a difficult region to obtain trajectories do to the increasing regions containing singular solutions as well as there being very few configurations when the manipulator is fully extended. A straight line trajectory in the workspace is shown in Figure 4. The corresponding manipulability curve is shown in Figure 5. It is important to note the perturbations in the manipulability curve. Taking notice of the scaling on the graph, this is a magnified view to demonstrate that the curve is not actually perfectly smooth which is purely do to the sampling resolution. This also, has a direct correlation to the the smoothness in the configuration-space trajectory. There exist small vibrations due to the defined threshold for continuity. However, due to the fineness of our sampling resolution the shown perturbations was unnoticed during experimentation with an actual maximum manipulability delta of ~$1/1000$.

For this straight line trajectory experiment, the same number of $306,306$ configurations were considered at each discrete position. Of all configurations considered $33,902$ were validated and inserted to $T_{Q_{calc}}$ and considered when analyzing configurations along the 20 discrete positions along the $20cm$ in length straight line path. Generating the optimal trajectory took just under $4sec$ with a branching factor of ~6.

VI. Discussion

In Section III we consider only manipulators that have an inverse kinematic model that can return all solutions such as the arm of the HRP2-DH robot. However, since in actual implementation only a finite set of configurations can be considered there are more general methods that can calculate a subset of configurations that would suffice for our $T_{Q_{calc}}$ to generate $T_{opt,calc}$ which is the optimal trajectory within the calculated sampling resolution. One such method is pre-

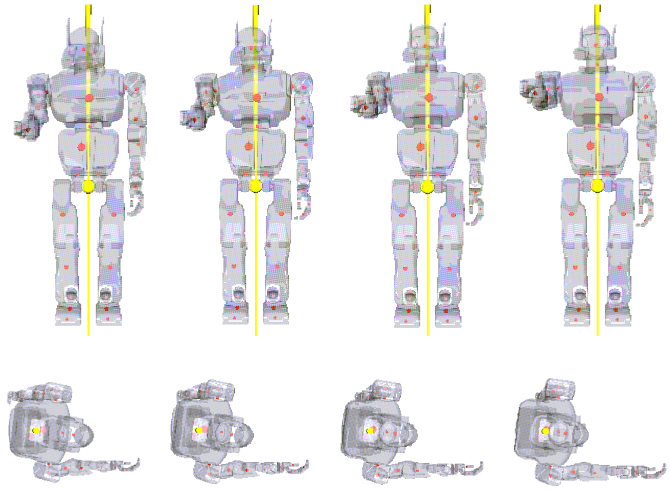

Fig. 8. Front view *(top)* and top view *(bottom)* of the simulation results for the linear workspace trajectory.

sented in [?] where a memory approach is taken to mapping the configuration-space of any general manipulator to its workspace. Another very general and more adaptive approach is a learning based method that can be found in [?]. Utilizing such methods would lead to manipulability optimization for manipulators with partial known or even unknown inverse kinematic models. However, it is important when utilizing such methods to define a more strict guidlines for calculating $T_{Q_{calc}}/subsetT_Q$ since the algorithm is resolution complete.

Furthermore, another area that we are working to enhance is our definition of continuity. Currently as presented in Section IV we are only considering a strict angular delta between adjacent configurations. However, we have been working to improve this definition by incorporating secondary optimization criteria such as minimum jerk optimizations [?] to provide extremely smooth high manipulable trajectories as direct output from the system. This methodology for enhancing c-space continuity also opens possibilities for incorporating other task specific secondary optimizations.

In Section V values are given for branching factor and actual calculation time. It is important to note that there exist a strong correlation between the two values. In addition, calculation time and branching factor is greatly effected by two main inputs: the defined threshold for continuity and resolution of sampled data. Decreasing the sample threshold will decrease the branching factor by making a more strict branching condition, however, in our database setup this can greatly increase the calculation time. This is do to each new configuration is linearly searched in order by manipulability values.

Secondly, increasing the sampling resolution will increase the branching factor and potentially decrease the calculation time. By having a more resolute sample set there will exist a greater number of T_{cont} therefore increasing the number of valid transitions that do not maintain the optimization conditions. Although, the branching factor is increased with a great number of valid transitions the number of new starting configurations to be checked will be reduced therefore reducing the calculation time.

Given such a relationship, it is important to balance the the defined threshold for continuity to the sampling resolution. This is not only important for computation time, but using $threshold : resolution$ ratios towards either extreme (small threshold : corse resoltion; large threshold : fine resolution) will greatly alter the shape of the trajectory within the configuration-space.

We have also taken notice to this relationship to be a fundamental property with the data organization and linear search method. We are currently looking into more specialized search methods to make a directly proportional relationship between branching factor and calculation time which would lead to the development of an extremely efficient and on-line manipulability based trajectory optimization system.

References

[1] T. Yoshikawa, "Manipulability of robotic mechanisms," in *Int. J. Robot. Res.*, vol. 4, Apr. 1985, pp. 3–9.

[2] M. Stilman and J. Kuffner, "Navigation among movable obstacles: Real-time reasoning in complex environments," in *Proceedings of the 2004 IEEE International Conference on Humanoid Robotics*, Dec. 2004.

[3] Y. Aydin and M. Nakajima, *Database guided computer animation of human grasping using forwar and inverse kinematics*. Computers and Graphics, 1999, vol. 23.

[4] Y. Aydin, H. Takahashi, and M. Nakajima, *Database guided realistic grasping posture generation for virtual actors*. Picture Coding Symposium, Germany, 1997.

[5] J. E. Bobrow, S. Dubowsky, and J. S. Gibson, "Time-optimal control of robotic manipulators along specified paths," in *International Journal of Robotics Research*, vol. 4, 1985.

[6] K. Shin and N. McKay, "Minimum-time control of robot manipulators with geometric path constraints," in *IEEE Trans. Autom. Control*, vol. 6, 1985.

[7] D. Wales, "Global optimization by basin-hopping and the lowest energy structures of lennard-jones clusters containing up to 110 atoms," in *IEEE Intl. Conf. Sys., Man, Cybern.*, Mar. 1998.

[8] J. Puzicha, N. Goerke, and R. Eckmiller, "Optimal trajectory generation for industrial robot by markovian networks," in *Int. Conf. on Neural Information Proccessing Systems*, 1998.

[9] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," in *International Journal of Robotics Research*, vol. 3, 1987.

[10] ——, "Object manipulation in a multi-effector robot system," in *Robotics Research*, vol. 4, 1988.

[11] L. Sentis and O. Khatib, "Control of multiple task primitives through multi-level prioritization," in *International Journal of Humanoid Robotics*, Aug. 2005.

[12] S. Harmeyer and A. Bowling, "Dynamic performance as a criterion for redundant manipulator control," in *IROS*, vol. 3, 2004.

[13] A. Bowling and O. Khatib, "The dynamic capability equations: A new tool for analyzing robotic manipulator performance," in *IEEE Transactions on Robotics*, vol. 21, 2005.

[14] L. Guilamo, J. Kuffner, K. Nishiwaki, and S. Kagami, "Efficient prioritized inverse kinematic solutions for redundant manipulators," in *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, 2005.

[15] A. D'Souza, S. Vijayakumar, and S. Schaal, "Learning inverse kinematics," in *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, 2001.

[16] T. Flash and N. Hogan, "The coordination of arm movements: an experimentally confirmed mathematical model," in *J. Neurosci*, vol. 5, 1985.