Optimization of Robot Configurations for Assistive Tasks

Ariel Kapusta and Charles C. Kemp
Healthcare Robotics Lab, Institute for Robotics and Intelligent Machines, Georgia Institute of Technology, Atlanta, Georgia, USA
Email: akapusta@gatech.edu, charlie.kemp@bme.gatech.edu

***Abstract*—Robots can provide assistance with activities of daily living (ADLs) to humans with motor impairments. Specialized robots, such as desktop robotic feeding systems, have been successful for specific assistive tasks when placed in fixed and designated positions with respect to the user. General-purpose mobile manipulators could act as a more versatile form of assistive technology, able to perform many tasks, but selecting a configuration for the robots from which to perform a task can be challenging due to the high number of degrees of freedom of the robots and the complexity of the tasks. As with the specialized, fixed robots, once in a good configuration, another system or the user can provide the fine control to perform the details of the task. In this short paper, we present Task-centric Optimization of robot Configurations (TOC), a method for selecting configurations for a PR2 and a robotic bed to allow the PR2 to provide effective assistance with ADLs. TOC builds upon previous work, Task-centric initial Configuration Selection (TCS), addressing some of the limitations of TCS. Notable alterations are selecting configurations from the continuous configuration space using a Covariance Matrix Adaptation Evolution Strategy (CMA-ES) optimization, introducing a joint-limit-weighted manipulability term, and changing the framework to move all optimization offline and using function approximation at run-time. To evaluate TOC, we created models of 13 activities of daily living (ADLs) and compared TOC's and TCS's performance with these 13 assistive tasks in a computer simulation of a PR2, a robotic bed, and a model of a human body. TOC performed as well or better than TCS in most of our tests against state estimation error. We also implemented TOC on a real PR2 and a real robotic bed and found that from the TOC-selected configuration the PR2 could reach all task-relevant goals on a mannequin on the bed.**

I. Introduction

Activities of daily living (ADLs), such as feeding and personal hygiene, are important for people, but these tasks can be challenging for those with motor impairments. Many specialized assistive devices can help people with motor impairments perform ADLs on their own. Specialized robots, such as desktop feeding devices, have been successful for a narrow range of assistive tasks when placed in fixed and designated positions with respect to the user. General-purpose mobile manipulators collaborating with robotic beds, wheelchairs, and the user, have the potential to provide assistance across a wide range of tasks, users, and environments. However, selecting a configuration for the robotic devices from which to perform the task can be challenging due to the high number of degrees of freedom of the robots.

Hawkins et al. [1] observed that some assistive tasks require that a mobile manipulator use multiple base positions, and that manually choosing those positions can be difficult. They presented a a human-in-the-loop system for the user to provide the fine control to perform the details of self-care tasks around the head once the robot is in a good configuration. Along those lines, the method we present in this work selects the 4-DoF configuration of a PR2 (X-Y base position, base orientation, and Z-axis height) and the 2-DoF configuration of a robotic bed (Z-axis height and head-rest angle), and leaves fine control of the PR2 arms to perform the task to some other system. In this work, each robot configuration consists of a PR2 configuration and a robotic bed configurations. Our system can select up to two robot configurations for a single task.

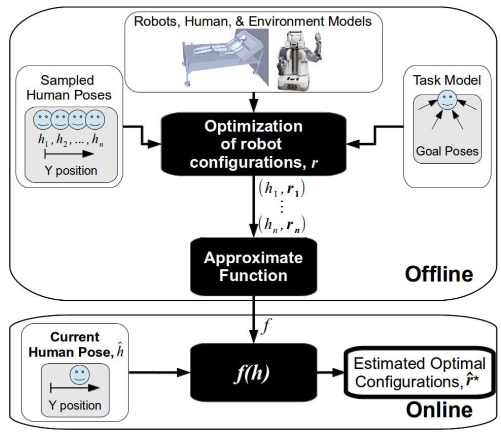

Fig. 1: The Framework used in TOC.

Selecting good configurations has been previously addressed in many ways. In our previous work, Kapusta et al. [2] presented Task-centric initial Configuration Selection (TCS). With a task-centric focus, TCS could use specifics of the problem, such as the specific geometries and a task model to aid the robot in finding collision-free solutions.

In this short paper, we briefly present Task-centric Optimization of robot Configurations (TOC), which builds upon and addresses some of the limitations of TCS. Unlike TCS, TOC uses Covariance Matrix Adaptation Evolution Strategy (CMA-ES) to perform all optimization offline in continuous configuration space, and uses Joint-Limit-Weighted Kinematic Isotropy to penalize configurations near joint limits. TOC takes advantage of things it can model in advance, such as a geometric model of the human, models of a set of tasks, and models of the robots and environment. It optimizes the robot configuration for samples of the human's pose on the bed. It then approximates a function to estimate the optimal robot configuration bed given an observed human pose. At run-

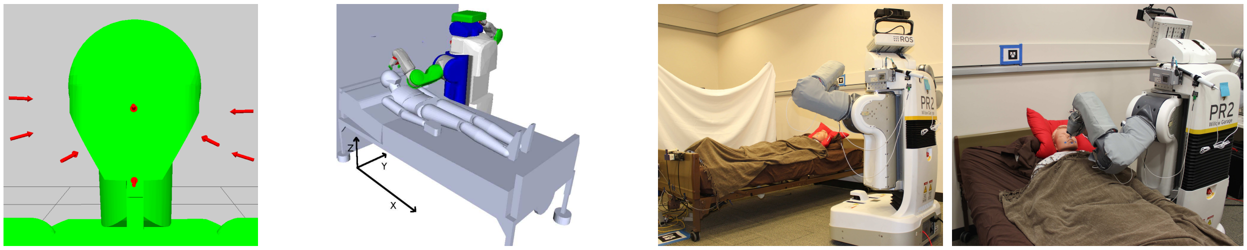

Fig. 2: From left to right: 1) goal poses for shaving task model; 2) optimized configuration of the PR2 and the robotic bed for wiping the mouth; 3&4) real robots before and after moving to optimized configuration for wiping the mouth. Note: the bed moved its height and angle.

time TOC applies the function. TOC's framework is shown in Figure 1 and described in Section II-A.

A. Related Work

Much prior research has investigated how to find good configurations for a mobile robot. A more thorough survey on related work can be found in our previous work [2]. Work continues to be done in the field, such as that by Dong and Trinkle [3], which uses a reachability map from Zacharias et al. [4], adjusted to allow extension to tools in the robot's end effector and desired orientations. Reachability maps are task generic and robot specific, facilitating application to new tasks. In contrast, our work is task specific and robot specific. TOC checks collisions, works in continuous task and configuration space, and returns a solution of up to two configurations.

Diankov et al. [5] presented BiSpace, a method that uses RRTs at run-time to find a path to a point in configuration space where the robot can achieve a set of goal poses. Our method performs little computation online, performing most computation beforehand, uses an optimization framework, and uses Task-centric manipulability to select configurations.

II. Method

A. Framework

Figure 1 shows the framework of TOC. TOC jointly optimizes two 6-DoF robot configurations, each of which consists of a 4-DoF configuration for the PR2 and a 2-DoF configuration for the robotic bed. The optimization is run for samples of the human's pose on the bed, h_i, given robot, human, and environments models. It interprets the optimization results to see if a single configuration is sufficient for the task, or if there is value in using two configurations. It then associates its choice, $\boldsymbol{r}_i$ with its respective h_i. These associations are used to approximate a function that is used at run-time to determine the estimated optimal configurations, $\hat{\boldsymbol{r}}^*$, given the observed human's pose on the bed, $\hat{h}$.

B. Implementation Details

1) Task Modeling: We created task models that are simple representations, a sparse set of end effector poses (Cartesian position and quaternion), that can allow a robot to efficiently make decisions about its ability to perform a task. TOC seeks one or two robot configurations from which the PR2 can not only reach the goal poses, but has high kinematic dexterity when reaching those poses, suggesting that it could also reach nearby poses relevant to the real task. We limited tasks to one-handed tasks and used only the robot's left arm in our evaluation. We modeled 13 tasks, listed in Table I. For example, Figure 2 shows the eight goal poses for the shaving task.

2) A Measure of Kinematic Dexterity: We use two measures to estimate how well the PR2 will be able to perform the task from a configuration: task-centric reachability (TC-reachability) and task-centric manipulability (TC-manipulability). These differ from common terms.

TC-reachability, P_R, is the percent of goal poses to which the robot can find an IK solution from robot configurations, $\boldsymbol{r}_k$, and can be found defined in [2].

TC-manipulability, P_M, is related to the average kinematic dexterity of the arm when reaching the goal poses. Its previous definition can be found in [2].

Hammond III and Shimada [6] used a torque-weighted global isotropy index to estimate the dexterity of a robotic arm given joint torques and torque limits. We have similarly modified kinematic isotropy, used previously in TC-manipulability, replacing it with joint-limit-weighted kinematic isotropy (JLWKI) by scaling with an nxn diagonal joint-limit-weighting matrix $\boldsymbol{T}$ defined as:

$$\boldsymbol{T}(q_j, q_{min}, q_{max}) = \begin{bmatrix} t_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & t_n \end{bmatrix}$$

where each element t_i in $\boldsymbol{T}$ is defined as

$$t_i = 1 - 0.5^{\left|\frac{1}{2}(q_{max,i} - q_{min,i}) - q_{j,i} - q_{min,i}\right|}$$

q_j is the arm's full joint configuration; q_i is the configuration for joint i; q_{max} and q_{min} are the arm's full joint max and min limits; $q_{max,i}$ and $q_{min,i}$ are the max and min limits for joint i.

We then compute JLWKI as:

$$JLWKI(q_j) = \frac{\sqrt[a]{\det(J(q_j)\boldsymbol{T}(q_j, q_{min}, q_{max})J(q_j)^T)}}{(\frac{1}{a})\text{trace}(J(q_j)\boldsymbol{T}(q_j, q_{min}, q_{max})J(q_j)^T)}$$

where a is the order of the robot arm's workspace (6 in the case of our 7-DoF arm).

3) The Optimization: Tan et al. [7] used CMA-ES to design a controller for articulated bodies moving in a hydrodynamic environment, inspiring our use of CMA-ES. We used CMA-ES (from https://pypi.python.org/pypi/cma) to optimize the robot configurations, $\boldsymbol{r}_i$, given the task, robot, human, and environment models, c, and a pose of the human on the bed,

TABLE I: Evaluation of performance of TOC vs TCS with error introduced in 1000 Monte Carlo simulations. Values in bold are statistically significant ($p < 0.001$ in Wilcoxon Rank Sum tests).

Task	**TOC**: mean (std)	**TCS**: mean (std)
Shaving	99.6.% (2.1)	99.9% (1.0)
Bathing	**86.7% (2.9)**	76.9% (6.8)
Wiping Mouth	95.3% (10.)	**100.% (0.0)**
Feeding	99.9% (1.7)	99.8.% (2.2)
Scratching left upper arm	100.% (0.0)	100.% (0.0)
Scratching right upper arm	100.% (0.0)	100.% (0.0)
Scratching left forearm	100.% (0.0)	100.% (0.0)
Scratching right forearm	100.% (0.0)	100.% (1.6)
Scratching left thigh	99.6% (4.2)	100.% (0.0)
Scratching right thigh	100.% (0.0)	99.9% (2.2)
Scratching left knee	99.6% (3.3)	99.4% (4.4)
Scratching right knee	99.8% (3.0)	99.7% (3.3)
Scratching chest	**99.5% (6.3)**	72.4% (38.0)

h_i, using the objective function shown in equation 1. The optimization was run for each task for samples of the human's pose on the bed.

$$\underset{\boldsymbol{r}_i}{\arg\min} \quad -\alpha P_R(\boldsymbol{r}_i, h_i, c) - \beta P_M(\boldsymbol{r}_i, h_i, c) \tag{1}$$

The optimization simultaneously optimizes two robot configurations. Any configuration where the PR2 base would collide with the robotic bed if shifted 2cm in either X or Y direction was considered to be in collision. We used a heuristic when both P_R and P_M are zero that pushes the objective function toward configurations that may have non-zero P_R and P_M. All values from the heuristic are larger than 0. We used a value of 10 for α and values of 1 for β. TOC interprets the results of the optimization to see if using one or two configurations works best for the task.

4) Approximate Function: Offline, TOC approximates a function that estimates the optimal configurations, $\hat{\boldsymbol{r}}^*$, given an estimated pose of the human on the bed, h. At run-time, TOC applies this function to the observed human pose, $\hat{h}$. For this paper we used 1-nearest neighbor as the function, f.

III. Evaluation

Figure 2 shows the simulation environment with a PR2 and with a human on a configurable bed. We put a wall behind the bed to simulate how beds are often positioned in rooms. We compared the performance of TOC with TCS (as implemented by Kapusta et al. [2] in the same environment) in Monte Carlo simulations with introduced error in the human's global X and Y positions (translating around on the bed). The two systems selected robot configurations for the task given the human is positioned in the center of the bed. We evaluated how many goal poses could be reached with the human position error introduced. We did this analysis for all 13 modeled tasks. The error introduced was normally distributed around 0 with a standard deviation of 2.5 cm in the global X direction and 5 cm in the global Y direction.

The results are shown in Table I. TOC has comparable or better performance in most tasks compared to TCS.

We also implemented TOC on a real PR2 and a real robotic bed. The robotic bed is based on that presented by Grice et al. [8]. We had the PR2 attempt the wiping mouth task for a mannequin on the bed using TOC to select the configuration. We manually checked that the PR2 could touch points all around the mouth. Figure 2 shows the environment when TOC starts and the configuration the PR2 and robotic bed moved to, selected by TOC for the task, as well as the PR2 reaching the task area.

IV. Discussion and Conclusion

TOC incorporates improvements from TCS:

- A reworked framework that separates the method into an offline optimization and an approximated function at runtime.
- Use of CMA-ES to search for configurations in continuous configuration space instead of a brute force search over discretized space.
- Use of Joint-Limit-Weighted Kinematic Isotropy (JL-WKI) to mitigate problems due to joint limits.

TOC has equal or better performance than TCS in most tasks in our evaluation. We have implemented TOC on a real robot and demonstrated the feasibility of using it to select configurations for a real PR2 and a real robotic bed.

***Acknowledgment**: We thank Yash Chitalia for his work on the robotic bed. This work was supported in part by the National Institute on Disability, Independent Living, and Rehabilitation Research (NIDILRR), grant 90RE5016-01-00 via RERC TechSAge, and by NSF Awards IIS-1514258 and IIS-1150157.*

References

[1] Kelsey P Hawkins, Phillip M Grice, Tiffany L Chen, et al. Assistive mobile manipulation for self-care tasks around the head. In *Computational Intelligence in Robotic Rehabilitation and Assistive Technologies (CIR2AT), 2014 IEEE Symposium on*, pages 16–25. IEEE, 2014.

[2] Ariel Kapusta, Daehyung Park, and Charles C Kemp. Task-centric selection of robot and environment initial configurations for assistive tasks. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 1480–1487. IEEE, 2015.

[3] Jun Dong and Jeffrey C Trinkle. Orientation-based reachability map for robot base placement. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 1488–1493. IEEE, 2015.

[4] Franziska Zacharias, Christoph Borst, and Gerd Hirzinger. Capturing robot workspace structure: representing robot capabilities. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 3229–3236. Ieee, 2007.

[5] Rosen Diankov, Nathan Ratliff, Dave Ferguson, Siddhartha Srinivasa, and James Kuffner. Bispace planning: Concurrent multi-space exploration. *Proceedings of Robotics: Science and Systems IV*, 63, 2008.

[6] Frank L Hammond III and Kenji Shimada. Improvement of kinematically redundant manipulator design and placement using torque-weighted isotropy measures. In *Advanced Robotics, 2009. ICAR 2009. International Conference on*, pages 1–8. IEEE, 2009.

[7] Jie Tan, Yuting Gu, Greg Turk, and C Karen Liu. Articulated swimming creatures. In *ACM Transactions on Graphics (TOG)*, volume 30, page 58. ACM, 2011.

[8] Phillip Grice, Yash Chitalia, Megan Rich, et al. Autobed: Open hardware for accessible web-based control of an electric bed. RESNA, 2016.