Available online at www.sciencedirect.com

ScienceDirect

Procedia Manufacturing 17 (2018) 976–984

Procedia MANUFACTURING

www.elsevier.com/locate/procedia

28th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2018), June 11-14, 2018, Columbus, OH, USA

Stochastic Search Methods for Mobile Manipulators☆

Amoako-Frimpong Samuel Yaw[a], Matthew Messina[a], Henry Medeiros[a,*], Jeremy Marvel[b], Roger Bostelman[b]

[a]*Department of Electrical and Computer Engineering, Marquette University, Milwaukee, WI 53233 USA.*
[b]*Intelligent Systems Division, National Institute of Standards and Technology, Gaithersburg, MD 20899*

Abstract

Mobile manipulators are a potential solution to the increasing need for additional flexibility and mobility in industrial applications. However, they tend to lack the accuracy and precision achieved by fixed manipulators, especially in scenarios where both the manipulator and the autonomous vehicle move simultaneously. This paper analyzes the problem of dynamically evaluating the positioning error of mobile manipulators. In particular, it investigates the use of Bayesian methods to predict the position of the end-effector in the presence of uncertainty propagated from the mobile platform. The precision of the mobile manipulator is evaluated through its ability to intercept retroreflective markers using a photoelectric sensor attached to the end-effector. Compared to a deterministic search approach, we observed improved robustness with comparable search times, thereby enabling effective calibration of the mobile manipulator.

Keywords: mobile manipulators, calibration, search algorithms

1. Introduction

Robotic manipulators have found a wide variety of applications, most prominently in manufacturing. Their versatility in handling several tasks allows them to be applied in a wide range of assembly and material handling applications. With their increased deployment, the need for flexibility to enable reassignment with minimal reconfiguration and motion in unstructured environments is high - especially in small industries [1]. The fixed manipulators currently deployed, however, are restricted by their limited workspaces and inability to assist in complex assembly tasks. Mobile manipulators have the potential to address these needs. With the base providing an unlimited workspace, and

☆ This work was supported by the National Institute of Standards and Technology (NIST) award 70NANB16H196. Commercial equipment are identified in this paper to foster understanding. This does not imply recommendation or endorsement by NIST, nor that the equipment identified are necessarily the best available for the purpose.

* Corresponding author
E-mail address: henry.medeiros@marquette.edu

10.1016/j.promfg.2018.10.106

enabling the repositioning of the arm to ensure increased manipulability, mobile manipulators are better suited to a number of manipulation applications. However, dynamic manipulation is a notoriously challenging task [2, 3, 4]. Although research on mobile manipulators started almost two decades ago, [5, 6, 7], most works focus on the control and stability of the base and arm [8, 7, 9, 10] using techniques traditionally applied to a variety of robotics problems [11, 12, 13, 14]. A comprehensive discussion of previous research on mobile manipulators, including topics such as robot navigation, path planning, and performance evaluation, can be found in [15, 16].

However, without external localization and feedback control, mobile manipulators tend to suffer from reduced precision, as the coupled dynamics experienced during the simultaneous motion of the manipulator and mobile platform reduces their overall precision [17]. Therefore, methods to calibrate mobile manipulators are needed [18]. Previous attempts at solving this problem have used deterministic approaches [19] such as a spiral search with the platform stationary to localize a set of retroreflective markers using a photoelectric laser sensor and hence recalibrate the overall pose of the manipulator. In most manufacturing applications, however, continuous motion operations can substantially decrease production times in comparison with stop-and-go approaches [20]. In applications involving the manufacture of large structures, such as in the aerospace and nautical industries, allowing the robotic platform to move would be substantially more effective than continuously moving the structure itself. Also, in some applications such as welding of large components, continuous motion can also improve the uniformity of the process, thereby increasing its quality [21]. Therefore, there is a need to extend mobile manipulator calibration methods to scenarios in which the motion of the base is not interrupted.

Stochastic search mechanisms have been shown to outperform spiral search methods in the presence of larger uncertainty levels, which tends to be the case while the base is in motion [22]. Therefore, this paper presents novel algorithms to improve search performance using stochastic methods that allow the pose of the manipulator to be estimated without the need to interrupt the motion of the base.

2. Proposed Approach

We address the problem of measuring the performance of a robotic manipulator mounted on a mobile robotic platform. More specifically, we are interested in investigating the scenario in which both the platform and the manipulator simultaneously move while the robot accesses test points in a known configuration. The test points consist of multiple retroreflective markers enclosed in collimator tubes rigidly attached to a calibration artifact. While the base moves in a certain direction and with a certain speed, both of which can vary in time and are known up to an error, a six degrees of freedom (DOF) manipulator mounted to the base must align its end-effector in all 6DOF with each marker in sequence until the entire artifact has been observed. A laser sensor is mounted to the end-effector to localize the markers. The alignment must be done as quickly and as accurately as possible considering the uncertainties in the motion of the robot. The collimators restrict the visibility of the markers, thereby enforcing not only positional but also angular alignment between the sensor at the end-effector and the fiducial. Figure 1(a) shows a diagram illustrating the cross-section of the retroreflective fiducial and collimator. Figure 1(b) shows a picture of our simulation environment illustrating the arrangement of the fiducials with respect to the mobile manipulator, and Figure 1(c) shows a picture of our proof-of-concept platform.

Fundamentally, this is a problem of estimating the relative pose of each marker with respect to the end-effector, with the added difficulty that the base of the manipulator is in motion. A principled solution based on probabilistic methods to estimate and compensate for the target motion can be employed to simultaneously estimate the position and the velocity of the manipulator and the base. Evidently, the error caused by the imperfect platform motion would accumulate and the accuracy of the system would deteriorate from marker to marker. A 6DOF heuristic search at each marker is therefore needed to compensate for that. In this work, we propose a stochastic approach that estimates the distribution of the positions of the markers and uses this distribution to search for the marker in the presence of motion error. We compare this method with a deterministic search mechanism based on a spiral pattern.

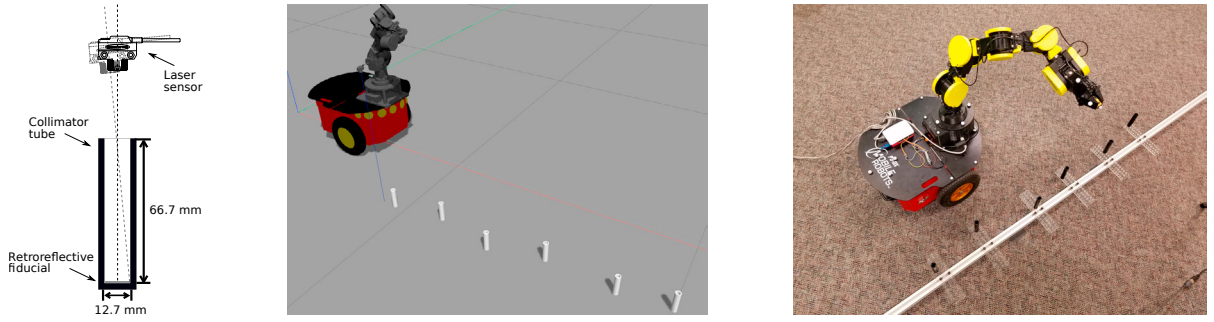


Figure 1: a) Cross-section of the retroreflective fiducial with collimator – the grayed out sensor position illustrates the maximum angle at which the fiducial can be observed. b) Simulation environment – the retroreflective fiducial markers are placed at the bottom of the cylindrical collimators. c) Proof-of-concept hardware platform and calibration artifact.

2.1. Fiducial Search Mechanism

In our proposed search algorithm, we initialize the base to move along a pre-defined path with linear velocity v_b and heading angle θ_b. A set of fiducial markers $\{f_i\}$, $i = 1, \ldots, n_m$ are positioned at coordinates $p_i = [x_i, y_i]$ on the ground plane, where x_i and y_i are the coordinates of the fiducial along the x and y axis of a previously defined coordinate frame. As the base moves, once a fiducial enters the range of the manipulator, the end-effector is moved to the expected position of the fiducial. That is, let r represent the maximum distance that can be reached by the manipulator at a certain orientation, and let $p_b(t) = [x_b(t), y_b(t)]$ represent the position of the base at time t. The manipulator is moved to the fiducial position once $\|p_b(t) - p_i\| \leq r$, where $\|\cdot\|$ represents the Euclidean norm. Since the fiducial markers are placed at the ground plane, the orientation of the end-effector is maintained constant (i.e., facing down). If the sensor cannot detect the presence of the fiducial, a search procedure is triggered. If the fiducial is intercepted, the position error, e_i, and search time, t_i, are recorded. This process is repeated until all the fiducials are either intercepted by the sensor or leave the range of the manipulator. For the sake of simplicity, in this work we assume the base velocity and heading angle are constant. We also assume all the fiducial markers can be reached by the manipulator as the base moves along this trajectory without the need to change its heading angle. Algorithm 1 summarizes the overall search procedure.

2.1.1. Deterministic Search Points Generation

In the deterministic search approach, the fiducial search points are generated according to a spiral pattern, given by

$$x_k = R\phi_k \cos(\phi_k), \quad y_k = R\phi_k \sin(\phi_k) \tag{1}$$

where $s_k = [x_k, y_k]$ is the position of the next sample, $k = 1, ..., n_s$, n_s is the number of samples, R is the radius of the spiral, and ϕ_k is the constant interpoint angular interval.

2.1.2. Stochastic Search Points Generation

In our stochastic search method, we model the system using a simple recursive Bayesian estimator, specifically, a linear Kalman filter. The filter estimates the state of the end-effector, guiding it to the expected position of the marker. If interception fails, it samples the probability distribution of the position to find it. Once the fiducial is localized, the corresponding location error can be used to correct the position of the mobile platform. As Eq. 2 shows, we model the base motion using a simple 2D constant velocity model in which the target state $x(t)$ includes the position of the base $[x_b(t), y_b(t)]$ as well as its velocity $[\dot{x}_b(t), \dot{y}_b(t)]$. The state transition matrix $A(t)$ takes into consideration the time interval δ_t between two consecutive observations. The observation matrix $C(t)$ models the fact that only the fiducial

Algorithm 1 Fiducial search method.

Input: Fiducial coordinates $p_i = [x_i, y_i]$, $i = 1, \ldots, n_m$. Base velocity v_b and heading angle θ_b
Output: Position errors e_i and search times t_i of the $i = 1, \ldots, n_f$ intercepted fiducials

```
Set base velocity to v_b and heading angle to θ_b
n_f = 0, i = 1
while i ≤ n_m do
    if ||p_b(t) − p_i|| ≤ r then
        Move manipulator to p_i
        if Fiducial f_i is not intercepted then
            Generate sample points as described in Sections 2.1.1 and 2.1.2
            Compensate base velocity as described in Section 2.1.4
            Initiate fiducial search
            while ||p_b(t) − p_i|| ≤ r and f_i is not intercepted do
                Perform fiducial search
            end while
        end if
        if Fiducial f_i is intercepted then
            Compute error e_i and search time t_i
            n_f = n_f + 1
        end if
        i = i + 1
    end if
end while
```

positions are observed.

$$x(t) = \begin{bmatrix} x_b(t) \\ \dot{x}_b(t) \\ y_b(t) \\ \dot{y}_b(t) \end{bmatrix}, \quad A(t) = \begin{bmatrix} 1 & \delta_t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \delta_t \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad C(t) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{2}$$

The predicted target state $\hat{x}(t|t-1)$ and its corresponding covariance $\tilde{\Sigma}(t|t-1)$ are then computed using the standard Kalman filter equations. That is,

$$\hat{x}(t|t-1) = A(t-1)\hat{x}(t-1|t-1), \quad \tilde{\Sigma}(t|t-1) = A(t-1)\tilde{\Sigma}(t-1|t-1)A^T(t-1) + \Sigma_{ww}(t-1), \tag{3}$$

where $\Sigma_{ww}(t-1)$ is the process noise covariance. The estimated target state $\hat{x}(t|t)$ and its corresponding covariance $\tilde{\Sigma}(t|t)$ are given by

$$\hat{x}(t|t) = \hat{x}(t|t-1) + K(t)e(t), \quad \tilde{\Sigma}(t|t) = [I - K(t)C(t)]\tilde{\Sigma}(t|t-1), \tag{4}$$

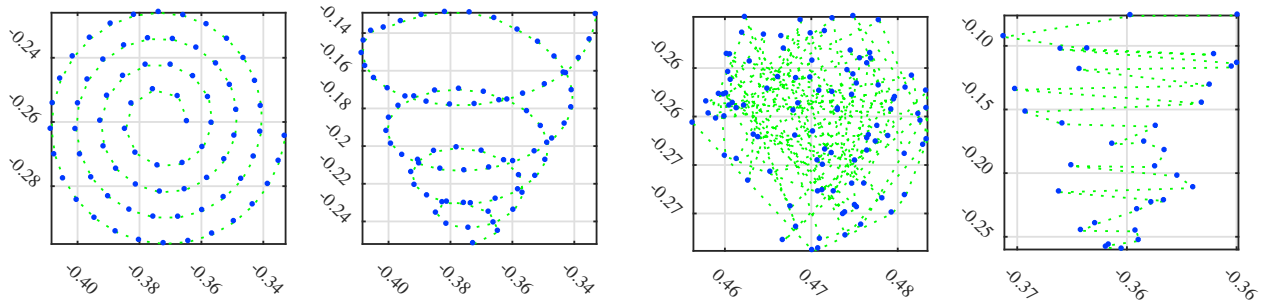
where $e(t)$ is the innovation and $K(t)$ is the Kalman gain, given by

$$e(t) = z(t) - C(t)\hat{x}(t|t-1), \quad \Sigma_{ee}(t) = C(t)\tilde{\Sigma}(t|t-1)C^T(t) + \Sigma_{vv}(t), \quad K(t) = \tilde{\Sigma}(t|t-1)C^T(t)\Sigma_{ee}^{-1}(t), \tag{5}$$

where $\Sigma_{ee}(t)$ is the innovation covariance. The observation $z(t)$ corresponds to the position of the previous fiducial with additive zero-mean Gaussian noise with covariance $\Sigma_{vv}(t)$, i.e., $z(t) = p_{i-1} + \eta(t)$ where $\eta(t) \sim \mathcal{N}(0, \Sigma_{vv}(t))$. The fiducial search points are then generated according to

$$s_k \sim \mathcal{N}(\hat{x}(t|t), \tilde{\Sigma}(t|t)), \tag{6}$$

where $\mathcal{N}(\mu, \Sigma)$ represents a normal distribution with mean μ and covariance Σ, and $s_k = [x_k, y_k]$, $k = 1, ..., n_s$, and n_s are the sample position and the number of samples as described in Section 2.1.1. The samples are then ordered according to their probability so that points which are more likely to contain the fiducial are searched first.



(a) Velocity-compensated spiral search.

(b) Velocity-compensated stochastic search.

Figure 2: Search point velocity compensation for a) spiral search and b) stochastic search. The figures represent the distances in centimeters from the base of the mobile manipulator to the end-effector in the x and y coordinate axes. The figures on the left correspond to the original sample points and the figures on the right show the points after velocity compensation.

2.1.3. Sample Discretization

Whereas the samples generated using the deterministic spiral pattern are intrinsically discrete, due to the fact that the normal distribution in Eq. 6 is continuous, in the stochastic sample generation mechanism, a large number of samples are very closely spaced, as Figure 3(a) illustrates. Therefore, searching over these samples would likely introduce substantial redundancy in the search process and therefore significantly increase the search time. To address this issue, we discretize the samples over a grid of resolution γ. That is, we divide the search space into a uniform grid over the interval $[x_{min}, x_{max}]$ and $[y_{min}, y_{max}]$ so that the distance between two neighboring points is $|x_j - x_{j-1}| \geq \gamma$ and $|y_j - y_{j-1}| \geq \gamma$.

Although grid-based Bayesian methods or sequential Monte Carlo approaches could be used to directly generate discrete distributions [23], these approaches require more complex modeling strategies and are computationally more intensive. As our experimental evaluation demonstrates, a simple discretized Kalman filter is able to effectively estimate the distribution of the positions of the fiducials.

2.1.4. Base Velocity Compensation

Since the base of the manipulator is in motion during the search, the search points generated as described above must be compensated for the base velocity. That is,

$$\hat{s}_k = s_k + v_b \delta_k, \tag{7}$$

where v_b is the base velocity and δ_k is the time elapsed as the manipulator moves between point $k-1$ and k. Figure 2 illustrates the sample points and the corresponding velocity-compensated points for both search mechanisms.

2.1.5. Iterative Trajectory Planning

A direct solution of Eq. 7 requires knowledge of the exact value of δ_k, which is not available a priori since changing the position of the sample s_k affects the travel time itself. To resolve this problem, we propose an iterative trajectory planning mechanism. In our proposed method, we initially plan a trajectory using the velocity-compensated samples as waypoints. Once an initial trajectory is planned, it is possible to use the values of δ_k generated by the motion planner as it computes the trajectory of the manipulator to refine the trajectory. These values should represent a more accurate estimate of the actual time elapsed as the robot moves from one point to the other. As we perform additional iterations of this procedure, the compensated points converge to the actual positions of the samples. Algorithm 2 summarizes this procedure.

Figure 3(b) illustrates the iterative planning procedure for the spiral search method. In the figure, the different colors correspond to the trajectories generated at each iteration. It can be observed in the figure, particularly in regions where the radius of the spiral is large, that the trajectories gradually converge to a final path (near the center of the trajectories). A similar behavior is observed for the stochastic search method.

Algorithm 2 Iterative search planning.

Input: Initial samples s_k, number of samples n_s, number of iterations n_i
Output: Updated samples $\hat{s}_k$

1: Compute initial search times δ_k^0 using waypoint-planned trajectory for all $k = 1, ..., n_s$
2: **for** $j = 1$ to n_i **do**
3: Find the sample points $\hat{s}_k$ using Eq. 7 with $\delta_k = \delta_k^{j-1}$
4: Update δ_k^j using the times given by the motion planning algorithm
5: **end for**

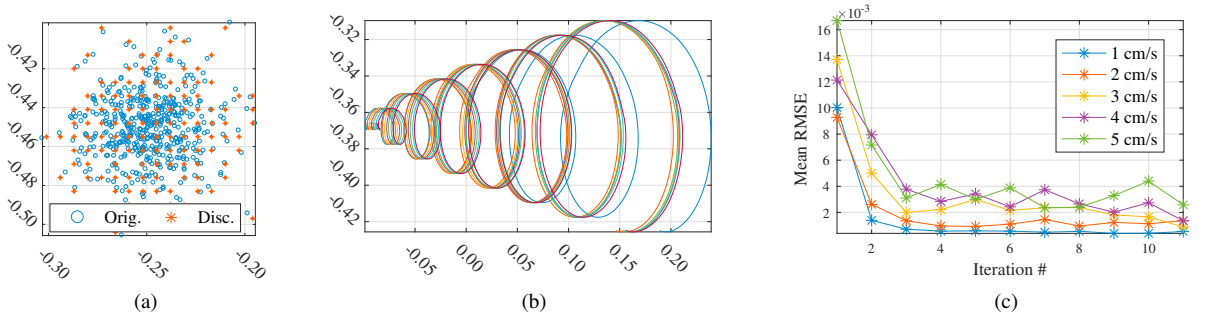


Figure 3: a) Illustration of the sample discretization process. b) Sample iterations of the planning algorithm. c) Convergence of the iterative planning algorithm for different base speeds.

Table 1: Fiducial arrangement.

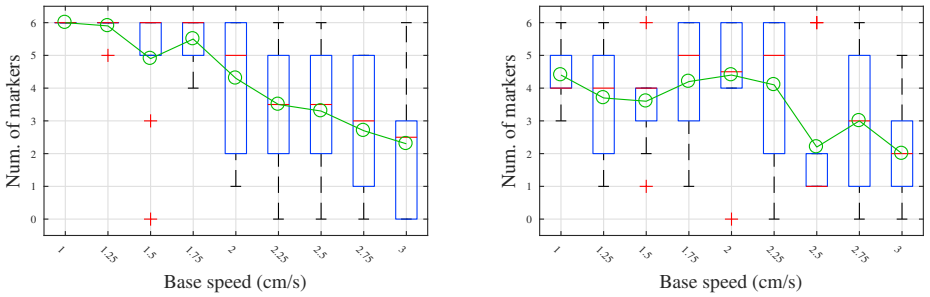
f_i	1	2	3	4	5	6
x_i (m)	0.94	1.14	1.37	1.57	1.82	2.00
y_i (m)	0.34	0.32	0.37	0.32	0.34	0.32

3. Experimental Results

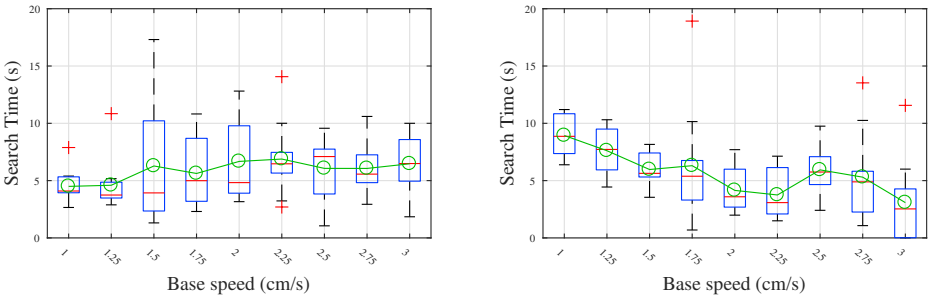
We evaluate our proposed stochastic search mechanism and the spiral search method in a simulation environment as well as on a real proof-of-concept hardware platform. Our proof-of-concept platform consists of a Robai Cyton Epsilon 1500 7 DOF manipulator rigidly attached to a Pioneer 3DX robot (Figure 1), and our simulations were carried out using the Gazebo robot simulator. For the simulated experiments as well as the real world evaluations, each experiment consisted of positioning the base at the origin of the global coordinate system, setting it in motion with velocity v_b, waiting for it to search for all the fiducials and then returning it to its initial position. Our evaluation takes into consideration two main performance metrics: 1) the number of fiducials intercepted per experiment, n_f, and 2) the search time per fiducial, t_i. The experiments were carried out for base speeds v_b ranging from 1 cm/s to 3 cm/s in increments of 0.25 cm/s for the simulated experiments and 0.5cm/s for the real-world experiments. The heading angle θ_b was set to zero in all the experiments. We used $n_m = 6$ fiducials with collimators of internal diameter of 12.7 mm and height of 66.7 mm arranged as indicated in Table 1. The parameters of the stochastic search algorithm were set to γ=14mm, $\Sigma_{ww} = 10^{-4} \times \text{diag}(3, 2, 3, 2)$, $\Sigma_{vv} = 10^{-4} \times (1.8, 1.8)$, where diag$(\cdot)$ represents a diagonal matrix. The spiral search parameters were R=4cm and $\phi_k = \frac{\pi}{5}$, which translates to a constant interpoint spacing of approximately 12mm. All the search performance results reported in this section are based on 10 repetitions of each experiment.

3.1. Iterative Planning

To determine the number of iterations needed for the planning method described in Section 2.1.5 to converge to a stable trajectory, we used our simulation system to evaluate the root mean squared error (RMSE) between each pair of consecutive iterations as a function of the number of iterations. Figure 3(c) shows the average results over ten runs



(a) Average number of fiducials intercepted: left) stochastic search, right) spiral search.



(b) Average search times: left) stochastic search, right) spiral search.

Figure 4: Performance evaluation using the simulated environment. The green lines with circular markers indicate average values, red bars show the medians, blue boxes represent the region between the first and third quartiles of the distribution, dotted lines show the minimum and maximum values of the distributions. Red crosses represent points which were considered outliers and hence lie outside of the distribution.

of the experiment for several base velocities. As the figure indicates, the method converges after approximately four iterations, and the number of iterations needed for convergence is relatively independent of the base velocity.

3.2. Search Performance

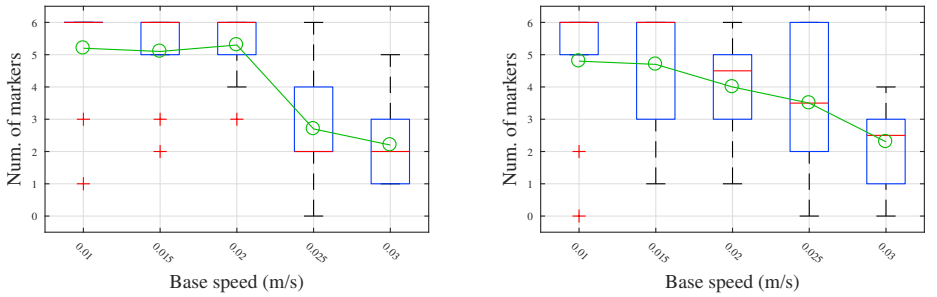
This section analyzes the search performance of both search methods in our simulation environment. In these experiments, to emulate robot positioning error, we disturb its initial position with additive Gaussian noise so that it is within a 1cm radius of its exact position 99% of the time. As demonstrated in the previous section, the iterative planning algorithm converges for less than five iterations for all the base speeds under consideration. Therefore, the results for the remaining of our experiments were obtained for $n_i = 5$ planning iterations.

Figure 4(a) shows the average number of intercepted fiducials as a function of the base speed for both methods. As the figure indicates, the stochastic search method intercepts an average of 4.27 fiducials whereas the spiral search approach intercepts 3.55 fiducials. The figures also indicate that the number of interceptions decreases as the base speed increases. This is caused by the fact that at higher speeds the manipulator has to cover a larger area while searching for the fiducials and hence fewer points can be reached by the manipulator after base speed compensation.

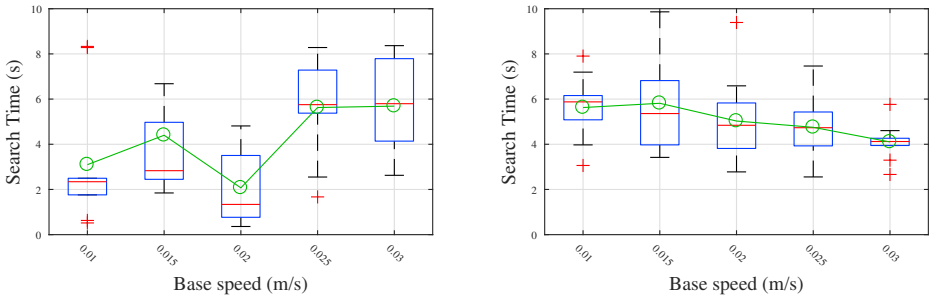
Figure 4(b) shows the average search times for the two approaches. Both methods show similar search times: 5.90 seconds on average for stochastic search and 5.67 for spiral search. However, at higher velocities, the planning algorithms have difficulty generating velocity-compensated samples for the spiral search approach, causing the search to be truncated and hence generating artificially lower search times. The stochastic search method is less affected by these planning difficulties as indicated by the fact that its search time does not decrease with the base speed.

3.3. Real World Experiments

In our experiments with the proof-of-concept platform, we used an optical tracking system consisting of six Optitrack Flex 13 cameras to manually align the position and orientation of the robot with the origin of the global



(a) Average number of fiducials intercepted: left) stochastic search, right) spiral search.



(b) Average search times: left) stochastic search, right) spiral search.

Figure 5: Performance evaluation using the real world proof-of-concept platform. See the caption of Figure 4 for a description of the elements of the graph.

coordinate system. Figures 5(a) and (b) show the average number of intercepted fiducials and the average search times in our real-world experiments. In these experiments, the stochastic approach intercepted 4.10 fiducials on average versus 3.86 from the spiral search method. The average search times were 4.18 and 5.06 seconds, respectively.

4. Conclusions

In this paper, we address the problem of calibrating mobile manipulators. We propose two dynamic search mechanisms to locate fiducial markers from a mobile platform: a spiral-based deterministic method and a stochastic mechanism based on Kalman filters. Our methods allow the manipulator to search for fiducial markers in a calibration artifact while the platform is in motion, thereby eliminating the need for the base to remain stationary at each calibration location. Our simulated experiments demonstrate that the stochastic search algorithm outperforms the spiral search method in terms of its ability to intercept markers without increasing the search times. We additionally validated our approach using a proof-of-concept hardware platform. We are currently investigating the possibility of extending our search mechanisms to 6DOF. Since performing a deterministic search in higher-dimensional spaces is a non-trivial problem, we expect the stochastic approach to show further performance improvements in these scenarios.

References

[1] H. Christensen, et al., A roadmap for us robotics: From internet to robotics (2016).
[2] B. Hamner, S. Koterba, J. Shi, R. Simmons, S. Singh, Mobile robotic dynamic tracking for assembly tasks, in: Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on, 2009, pp. 2489–2495.
[3] G. N. DeSouza, A. C. Kak, A subsumptive, hierarchical, and distributed vision-based architecture for smart robotics, IEEE Trans. on Systems, Man, and Cybernetics, Part B (Cybernetics) 34 (5) (2004) 1988–2002.
[4] Y. Yoon, A. Kosaka, A. C. Kak, A new kalman-filter-based framework for fast and accurate visual tracking of rigid objects, IEEE Trans. on Robotics 24 (5) (2008) 1238–1251.

[5] D. Katz, E. Horrell, Y. Yang, B. Burns, T. Buckley, A. Grishkan, V. Zhylkovskyy, O. Brock, E. Learned-Miller, The UMass mobile manipulator UMan: An experimental platform for autonomous mobile manipulation, in: Workshop on Manipulation in Human Environments at Robotics: Science and Systems, 2006.
[6] Y. Yang, O. Brock, Elastic roadmaps—motion generation for autonomous mobile manipulation, Autonomous Robots 28 (1) (2009) 113.
[7] Q. Huang, K. Tanie, S. Sugano, Stability compensation of a mobile manipulator by manipulatorpaper motion: feasibility and planning, Advanced Robotics 13 (1) (1998) 25–40.
[8] B. Hamner, S. Koterba, J. Shi, R. Simmons, S. Singh, An autonomous mobile manipulator for assembly tasks, Autonomous Robots 28 (1) (2009) 131.
[9] M. Mashali, R. Alqasemi, R. Dubey, Mobile manipulator dual-trajectory tracking using control variables introduced to end-effector task vector, in: 2016 World Automation Congress (WAC), 2016, pp. 1–6.
[10] S. Lin, A. A. Goldenberg, Neural-network control of mobile manipulators, IEEE Trans. on Neural Networks 12 (5) (2001) 1121–1133.
[11] M. Vukobratovic, B. A. Borovac, Zero-moment point - thirty five years of its life, International Journal of Humanoid Robotics 01 (01) (2004) 157–173.
[12] Y. Nakamura, H. Hanafusa, Inverse kinematic solutions with singularity robustness for robot manipulator control, ASME, Trans., Journal of Dynamic Systems, Measurement, and Control 108 (1986) 163–171.
[13] J. B. Rosen, The gradient projection method for nonlinear programming. part i. linear constraints, Journal of the Society for Industrial and Applied Mathematics 8 (1) (1960) 181–217.
[14] D. E. Whitney, Resolved motion rate control of manipulators and human prostheses, IEEE Trans. on man-machine systems 10 (2) (1969) 47–53.
[15] M. Shneier, R. Bostelman, Literature review of mobile robots for manufacturing, Tech. rep., NIST Internal report (2014).
[16] Z. Chebab, J. Fauroux, N. Bouton, Y. Mezouar, L. Sabourin, Autonomous collaborative mobile manipulators: State of the art, in: Symposium on Theory of Machines and Mechanisms/UMTS2015/TrISToMM, 2015.
[17] B. Hamner, S. Koterba, J. Shi, R. Simmons, S. Singh, Mobile robotic dynamic tracking for assembly tasks, in: 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2009, pp. 2489–2495.
[18] M. Hvilshoej, S. Boegh, O. Madsen, M. Kristiansen, Calibration techniques for industrial mobile manipulators: Theoretical configurations and best practices, in: ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics), 2010.
[19] R. Bostelman, T. Hong, J. Marvel, Performance measurement of mobile manipulators, in: SPIE Sensing Technology+ Applications, International Society for Optics and Photonics, 2015, pp. 94980E–94980E.
[20] E. Fetters-Walp, Fine line (2010).
URL http://www.boeing.com/news/frontiers/archive/2010/april/i_ca01.pdf
[21] J. Sprovieri, New technology for robotic welding, Assembly Magazine.
URL https://www.assemblymag.com/articles/93555-new-technology-for-robotic-welding
[22] K. V. Wyk, M. Culleton, J. Falco, K. Kelly, Comparative peg-in-hole testing of a force-based manipulation controlled robotic hand, IEEE Trans. on Robotics PP (99) (2018) 1–8. doi:10.1109/TRO.2018.2791591.
[23] M. S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking, IEEE Transactions on Signal Processing 50 (2) (2002) 174–188. doi:10.1109/78.978374.