Path Planning for Image-based Control of Wheeled Mobile Manipulators

Moslem Kazemi, Kamal Gupta, Mehran Mehrandezh

***Abstract*— We address the problem of incorporating path planning with image-based control of a wheeled mobile manipulator (WMM) performing *visually-guided* tasks in complex environments. The WMM consists of a wheeled (non-holonomic) mobile platform and an on-board robotic arm equipped with a camera mounted at its end-effector. The visually-guided task is to move the WMM from an initial to a desired location while respecting image and physical constraints. We propose a *kinodynamic* planning approach that explores the camera *state space* for permissible trajectories by iteratively extending a search tree in this space and simultaneously tracking these trajectories in the WMM configuration space. We utilize weighted pseudo-inverse Jacobian solutions combined with a null space optimization technique to effectively coordinate the motion of the mobile platform and the arm. We also present the preliminary results obtained by executing the planned trajectories on a real WMM system via a decoupled control scheme where the on-board arm is servo controlled along the planned feature trajectories while the mobile platform is simultaneously controlled along its trajectory using a state feedback tracking method.**

I. Introduction

Wheeled mobile manipulators are major efforts to bring both mobility and manipulation capabilities to human environments. To move autonomously and accomplish tasks robustly in complex environments high-level global motion planning techniques should be closely integrated with sensor-based control of such systems. Many efforts have been devoted to both motion planning and sensor-based control with promising advances in each individual area over the past decades. However, the integration of planning and control, in particular for complex systems such as a WMM, remains a challenging topic, and also crucial towards fully autonomous and robust solutions.

Among various sensory inputs, vision has gained lots of attention and found many applications in robotic solutions due to its rich input providing a large amount of information with a high frequency. A great deal of research has been devoted to developing vision-based control strategies for robotics application, leading to an active area of research called *Visual Servoing* [1]. The main idea in visual servoing is to use vision feedback to control the motion of the robot for performing a task. In contrast to position-based visual servoing (PBVS), where the control is performed in the Cartesian space based on 3-D information retrieved from image, in image-based visual servoing (IBVS) techniques, the feedback is defined based on image features and the control loop is closed directly within the image. This results in a more robust control in presence of calibration and modeling errors, and hence adding to the popularity of IBVS.

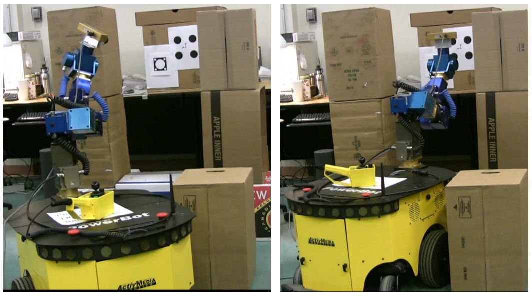

Fig. 1. SFU wheeled mobile manipulator system (a Powerbot mobile platform with an on-board 6-DOF Schunck robotic arm) reaches a desired location by tracking a target object.

In [2], through simple, yet, effective examples, Chaumette outlined the potential problems of stability and convergence of IBVS techniques: singularities in image Jacobian leading to an unstable behavior, and reaching local minima due to the existence of multiple camera poses yielding the same terminal image of the target. Moreover, in IBVS techniques there is no direct control over the image/camera/robot trajectories induced by the servoing loop in the image and physical spaces. Therefore, these trajectories might violate the image and/or physical constraints. The aforementioned challenges have recently motivated the researchers to incorporate path planning strategies into the visual servo loop.

The main idea of path planning for visual servoing is to plan and generate *feasible* image feature trajectories while accounting for the constraints, and then to servo the robot along the planned trajectories. A detailed review of existing path planning techniques for visual servoing is provided in [3]. A number of techniques aimed at interpolating a path *directly* in the image space between the initial and desired images without using any knowledge of camera calibration or target model (e.g., see [4], [5]). Potential fields have been employed in the context of visual servoing in face of constraints (e.g., field of view and joint limits [6], or obstacle avoidance behavior [7]). Some other techniques aimed at finding globally optimal paths with respect to various costs, for example distance from the image boundary, length of the path traversed by the robot, and energy expenditure. These approaches employ polynomial parametrization of camera paths (e.g., [8]), or utilize tools from optimal control (e.g., [9]). The convergence problems of potential field-

Moslem Kazemi is with the Robotics Institute, Carnegie Mellon University, PA, USA moslemk@cmu.edu, Kamal Gupta is with Simon Fraser University, BC, Canada kamal@sfu.ca, and Mehran Mehrandezh is with University of Regina, SK, Canada mehran.mehrandezh@uregina.ca.

based techniques on one hand and the expensive cost of the above optimization-based techniques on the other hand motivates the need for general and, yet global, path planning approaches such as randomized sampling-based techniques.

We devised a tree-based randomized path planning approach [10] to incorporate image and physical constraints for robust execution of servoing tasks. The technique proposed in [10] results in discrete feature trajectories which require to be properly time scaled so they can then be smoothly tracked at the execution stage. Later on we extended the work in [10] by incorporating dynamics of the camera at the planning stage providing a randomized kinodynamic planning framework for image-based control of robotic arms [11] in presence of image and physical constraints.

Despite many efforts on image-based control of robotic arms and mobile robots, there are very few works extending classical image-based control techniques to mobile manipulators. Wheeled mobile manipulators introduce two main challenges: non-holonomic kinematic constrains at the mobile base, and task redundancy. In [12] an inverse kinematic control is applied to servo control a mobile hand-eye system by deriving a Jacobian for the whole system as a kinematic chain. The work in [13] benefited from the task redundancy through a task sequencing strategy to avoid kinematic singularities during the servo control of a WMM. Due to the complexity of WMM systems accounting for image and physical constraints becomes even more challenging compared to robotic arms.

In this paper we extend our efforts on incorporating randomized path planning techniques with image-based control of robotic arms ([10] [11]) to wheeled mobile manipulators. We consider a visually-guided task [14], i.e., to move the robot from an initial to a desired location with respect to a stationary target while keeping the target in the field of view of the in-hand camera and avoiding occlusions/collisions (due to obstacles) without violating the kinematic constraints of the robot (Fig. 1).

The proposed planning approach explores the camera *state space* (i.e., a space of camera poses and velocities) for permissible trajectories by iteratively extending a search tree in this space and simultaneously tracking these trajectories in the WMM configuration space (C-space). We utilize weighted pseudo-inverse Jacobian solutions combined with a null space optimization technique to effectively coordinate the motion of the mobile platform and the arm. This forms one of the main contributions of this work comparing to our previous work in [11]. The proposed approach yields C^2-smooth camera trajectories by taking camera dynamics into account while accounting for a critical set of image and physical constraints.

This work also presents the preliminary results we obtained by executing the planned trajectories on a real WMM system (see Fig. 1) via a decoupled control scheme where the on-board arm is servo controlled along the planned feature trajectories while the mobile platform is simultaneously controlled along its trajectory in Cartesian space using a state feedback tracking method. The proposed decoupled control scheme is another contribution of this work.

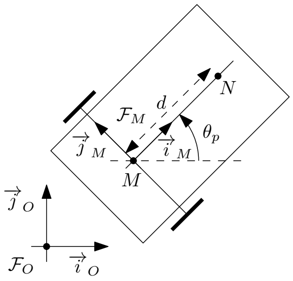

Fig. 2. Differential-drive mobile platform

II. Preliminaries

A. Mobile Platform Kinematics

We assume that the mobile platform moves on a planar surface (Fig. 2); and it has two actuated wheels which can roll independently with fixed headings. We make the rolling-without-slipping assumption for the wheels, i.e., the contact point of each wheel with the ground has zero velocity which introduces a non-holonomic constraint in the kinematic equations of the motion of the platform.

We follow the representation proposed by Samson and Ait-Abderrahim [15] in which the mobile platform position (x_p and y_p) is represented as the components of the vector $\overrightarrow{NO}$ in the basis of mobile frame $\mathcal{F}_M$ attached to the platform, i.e. $\overrightarrow{NO} = x_p \overrightarrow{i}_M + y_p \overrightarrow{j}_M$. The point N is located at a distance d from the wheels axis such that $\overrightarrow{MN} = d \overrightarrow{i}_M$ and point O is the origin of the fixed frame $\mathcal{F}_O$. The platform orientation θ_p is chosen as the angle between $\overrightarrow{i}_O$ and $\overrightarrow{i}_M$. Hence, the platform configuration is given as $\mathbf{q}_p = [x_p, y_p, \theta_p]^T$. The above representation is well adapted to the strategy we adopted from [15] to track the mobile platform along its planned trajectory in Cartesian space (see Section IV-B).

Considering the mobile platform as a kinematic system, we assume that the pseudo velocity $\mathbf{u}_p = [v_p \; \omega_p]^T$ where v_p is the linear velocity along $\overrightarrow{i}_M$ and $\omega_p = \dot{\theta}_p$, can be taken as the mobile platform control input.

The state equation of the mobile platform is then given as [15]

$$\dot{\mathbf{q}}_p = G_p(\mathbf{q}_p)\mathbf{u}_p \tag{1}$$

with

$$G_p(\mathbf{q}_p) = \begin{bmatrix} -1 & y_p \\ 0 & -(d + x_p) \\ 0 & 1 \end{bmatrix} \tag{2}$$

B. Robotic Arm Kinematics

The configuration space of the n-DOF robotic arm can be parameterized as $\mathbf{q}_a = [q_{a_1} \; q_{a_2} \; \dots \; q_{a_n}]^T$ where $q_{a_1}, q_{a_2}, \dots, q_{a_n}$ are the individual joint values. The arm is a kinematically unconstrained, i.e. holonomic, system. Hence, the joint velocities $\dot{\mathbf{q}}_a$ can be independently specified at any configuration. So, the arm control input vector $\mathbf{u}_a$ can be set as

$$\mathbf{u}_a = \dot{\mathbf{q}}_a \tag{3}$$

The pose of the arm's (or equivalently, WMM's) end-effector is represented as a 7-D vector

$$\boldsymbol{\xi}_a = [\mathbf{p}_\xi^T \ \mathbf{h}_\xi^T]^T \tag{4}$$

where $\mathbf{p}_\xi$ is a 3-D vector representing the end-effector position and $\mathbf{h}_\xi$ is a 4-D unit quaternion representing the end-effector orientation.

C. Mobile Manipulator Kinematics

The whole mobile manipulator configuration space can be parameterized in terms of the configurations of both the mobile platform and on-board robotic arm, i.e.

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_p \\ \mathbf{q}_a \end{bmatrix} \tag{5}$$

The kinematic model of the mobile manipulator sets the location of its end-effector (equivalently, the end-effector of the on-board robotic arm) as a function of its configuration, i.e.

$$\boldsymbol{\xi} = \boldsymbol{\xi}_a = \mathbf{F}(\mathbf{q}_p, \mathbf{q}_a) \tag{6}$$

By differentiating (6) with respect to time and using (2) we have

$$\begin{aligned} \dot{\boldsymbol{\xi}} &= \begin{bmatrix} \frac{\partial \mathbf{F}(\mathbf{q}_p,\mathbf{q}_a)}{\partial \mathbf{q}_p} G(\mathbf{q}_p) & \frac{\partial \mathbf{F}(\mathbf{q}_p,\mathbf{q}_a)}{\partial \mathbf{q}_a} \end{bmatrix} \begin{bmatrix} \mathbf{u}_p \\ \mathbf{u}_a \end{bmatrix} \\ &= \mathbf{J}(\mathbf{q}_p, \mathbf{q}_a)\mathbf{u} \end{aligned} \tag{7}$$

where $\mathbf{u} = [\mathbf{u}_p^T \ \mathbf{u}_a^T]^T$ denotes the mobile manipulator control input which is related to C-space (generalized) velocities as

$$\dot{\mathbf{q}} = \begin{bmatrix} G_p(\mathbf{q}_p) & \mathbf{0}_{3\times n} \\ \mathbf{0}_{n\times 2} & \mathbf{I}_{n\times n} \end{bmatrix} \begin{bmatrix} \mathbf{u}_p \\ \mathbf{u}_a \end{bmatrix} = G(\mathbf{q}_p)\mathbf{u} \tag{8}$$

and to the end-effector velocity screw through the mobile manipulator Jacobian $\mathbf{J}(\mathbf{q}_p, \mathbf{q}_a)$.

D. Modeling and Calibration Assumptions

We use the 2-D images of a number of 3-D target points $\mathcal{P}_j$ for $j = 1, \ldots, n$ as image features to represent the solution trajectories in the image. Hence, a point $\mathbf{s}$ along a feature trajectory is then represented as a vector $\mathbf{s} = [u_1 \ v_1 \ \ldots \ u_n \ v_n]^T$ where u_j and v_j are the image pixel coordinates of 3-D points $\mathcal{P}_j$ for $j = 1, \ldots, n$.

In the proposed approach we assume that the 3-D model of the target object (provided as the coordinates of a number of 3-D points $\mathcal{P}_j$ in the target frame which, without loosing generality, coincides with the fixed global coordinate frame $\mathcal{F}_O$) and camera intrinsic parameters are known *a priori*. The 3D model of the object is required to estimate the corresponding camera poses at the initial and desired configurations. Furthermore, these parameters are required to transform camera trajectories to feature trajectories in the image space. We also assume that obstacles in the workspace are known *a priori*.

III. Randomized Kinodynamic Planning with Visibility Constraints

The visually guided task is to move the WMM from a start to a goal configuration which are implicitly provided by the images captured from the known target using the in-hand camera. As we explain later, the image feedback from the target object is used to close the control loop for image-based tracking of the robotic arm along the planned feature trajectories, and also to localize the mobile platform with respect to the object given the known kinematics of the arm and its configuration. Therefore maintaining the visibility of the target throughout the trajectory is crucial for successful execution of the planned trajectories.

The aim of path planning for image-based control is to plan feature trajectories $\mathbf{s}(t)$ for $t \in [0, t_f]$ in the image space between initial and desired image features, $\mathbf{s}(0) = \mathbf{s}_i$ and $\mathbf{s}(t_f) = \mathbf{s}_d$, extracted from the images taken at the initial and final desired camera poses, respectively. The final time t_f is usually determined based on the time scaling strategy imposed by the underlying planning algorithm.

Our proposed planning approach is based on the ATACE (Alternate Task space and Configuration Space Exploration) framework, first presented in [16] for motion planning of robot manipulators with end effector constraints, and then adapted for path planning for visual servoing in [11] by us. It first plans the camera trajectory $\Gamma(t)$ for $t \in [0, t_f]$ which corresponds to a feasible robot trajectory $\mathbf{q}(t)$ for $t \in [0, t_f]$ in the WMM C-space between the start and goal configurations, $\mathbf{q}(0) = \mathbf{q}_i$ and $\mathbf{q}(t_f) = \mathbf{q}_d$, respectively. The planned camera trajectory $\Gamma(t)$ is then utilized to project the 3D target feature points into the image space and obtain the feature trajectories $\mathbf{s}(t)$ between the initial and desired image features in the image space. The details of the proposed planning approach has been explained in [11]. Here we provide a summary and explain the additional changes to extend the work in [11] to wheeled mobile manipulators.

ATACE explores the camera state space by extending an exploring tree as in Rapidly-exploring Random Tree (RRT) approach [17] in the camera state space and simultaneously tracking the tree local paths in the WMM C-space [11]. So, through this strategy the search in the camera state space is used to effectively guide the search in the WMM C-space.

The camera tree extension in state space follows a kinodynamic extension by considering the dynamics of the camera as a rigid body. The local paths obtained as the result of this extension is projected into the image space to check for image constraints, i.e. field of view limits and occlusions of target object by other obstacles or itself. Given that the local camera path violates no image constraints, it is then tracked in the WMM C-space using a local planner to check for physical constraints, i.e. collision with obstacles and joint limits. Next, we explain the local approach employed for C-space tracking where weighted pseudo-inverse Jacobian solutions combined with a null space optimization technique is employed to effectively coordinate the motion of the mobile platform and the arm for tracking the camera local path in C-space.

A. Camera Trajectory Tracking in C-space

In order to track the camera local trajectories (i.e., camera RRT edges) in the robot C-space we employ an *inverse kinematic control* in which kinematic tracking control laws are designed using the differential kinematic model in (7). This step extends our work in [11] on robotic arms by coordinating the motion between the mobile platform and the arm through a weighted pseudo-inverse Jacobian strategy, and performing null-space optimization to utilize the task redundancy of the WMM.

Without loosing generality, we assume that the camera frame coincides with the robotic arm end-effector which is effectively the WMM end-effector. Now, given an end-effector trajectory $\boldsymbol{\xi}^*(t)$ (extracted from a camera local trajectory), the problem is to find control input $\mathbf{u}(t)$, such that

$$\dot{\boldsymbol{\xi}}^*(t) = \mathbf{J}\mathbf{u}(t) \tag{9}$$

as in (7), which asymptotically stabilizes the task error $\mathbf{e}(t) = \boldsymbol{\xi}(t) - \boldsymbol{\xi}^*(t)$. All the exact (least-squares) solutions which satisfy the end-effector task constraint (9) are given as

$$\mathbf{u}(t) = \mathbf{J}^+\dot{\boldsymbol{\xi}}^*(t) + (\mathbf{I} - \mathbf{J}^+\mathbf{J})\mathbf{z}(t) \tag{10}$$

in which $\mathbf{J}^+$ is the pseudo-inverse of $\mathbf{J}$, the term $\mathbf{I} - \mathbf{J}^+\mathbf{J}$ represents the orthogonal projection matrix in the null space of $\mathbf{J}$, and $\mathbf{z}(t)$ is an arbitrary input velocity vector. The second term of the solution is therefore a null space velocity which does not affect the task at the end-effector. This solution minimizes the Euclidean norm $\|\dot{\xi}^* - \mathbf{J}\mathbf{u}\|$.

Recalling $\dot{\mathbf{q}}(t) = G(\mathbf{q}_p)\mathbf{u}(t)$ from (1) we have

$$\dot{\mathbf{q}}(t) = G\mathbf{J}^+\dot{\boldsymbol{\xi}}^*(t) + G(\mathbf{I} - \mathbf{J}^+\mathbf{J})\mathbf{z}(t) \tag{11}$$

The corresponding C-space trajectory $\mathbf{q}^*(t)$ can then be reconstructed using a discrete time integration technique, e.g., an Euler forward integration as

$$\mathbf{q}^*(t + \Delta t) = \mathbf{q}^*(t) + \dot{\mathbf{q}}(t)\Delta t \tag{12}$$

where Δt is the time step. To avoid drifting due to the numerical integration errors, and hence to asymptotically stabilize the error $\mathbf{e}(t)$, a closed-loop inverse kinematic solution is chosen instead, i.e.

$$\dot{\mathbf{q}} = G\mathbf{J}^+\Big[\dot{\boldsymbol{\xi}}^* - K_\xi(\boldsymbol{\xi} - \boldsymbol{\xi}^*)\Big] + G(\mathbf{I} - \mathbf{J}^+\mathbf{J})\mathbf{z} \tag{13}$$

where K_ξ is a constant positive-definite gain matrix.

The C-space trajectory $\mathbf{q}^*(t)$ is then checked for collision between the WMM body and the obstacles, as well as joint limits of the arm. Given that $\mathbf{q}^*(t)$ is collision free and does not violate any joint limits, the camera trajectory $\boldsymbol{\xi}(t)$ is added to the tree along with its end-point as a new node.

Motion Coordination via Weighted Pseudo-Inverse Jacobian: The components of $\mathbf{u}(t)$ in (9) have different physical dimensions, i.e., it is composed of two linear velocity and one angular velocity components for the mobile platform and six angular velocity components of the arm joints. So, it would be necessary to evaluate the magnitude of the solutions obtained using (13) based on appropriate weighting of the components. For example, we prefer a weighting strategy in which the mobile platform has a relatively bigger contribution in translational motions comparing to the robotic arm. This can be done by obtaining weighted-norm solutions of (9) using a weighted pseudo-inverse Jacobian formulation [18]. Let

$$\mathbf{J}^\# = \mathbf{J}Q_0^{-1}$$

where Q_0 is the square root of a symmetric and positive definite weighting matrix Q, i.e., $Q = Q_0^TQ_0$. Then, all the Q-weighted-norm solutions of (9) are given as

$$\mathbf{u}(t) = Q_0^{-1}\mathbf{J}^{\#^+}\dot{\boldsymbol{\xi}}^*(t) + Q_0^{-1}(\mathbf{I} - \mathbf{J}^{\#^+}\mathbf{J}^\#)\mathbf{z}(t) \tag{14}$$

Correspondingly, (13) can be written as

$$\dot{\mathbf{q}} = GQ_0^{-1}\mathbf{J}^{\#^+}\Big[\dot{\boldsymbol{\xi}}^* - K_\xi(\boldsymbol{\xi} - \boldsymbol{\xi}^*)\Big] + GQ_0^{-1}(\mathbf{I} - \mathbf{J}^{\#^+}\mathbf{J}^\#)\mathbf{z} \tag{15}$$

The weights in Q can be either modified dynamically (on-line) based on some proper measures, or set fixed. We have chosen the latter approach in which we choose relatively smaller weights for the mobile platform linear velocities, which in turn favors the motion of the mobile platform over the arm's in translation movements.

Improving Manipulability via Null Space Optimization: The redundancy of the WMM with respect to the task at the end-effector (i.e., end-effector trajectory tracking) can be used to improve the value of (configuration-dependent) scalar performance criteria $H(\mathbf{q})$, such as distance from singular configurations, joint limits, and/or obstacles. This can be achieved by a gradient descent method in which the potential function $H(\mathbf{q})$ has its minimum value corresponding to the objective requirements. We can write the time variation of the $H(\mathbf{q}(t))$ as

$$\dot{H} = \nabla^T H(\mathbf{q})\dot{\mathbf{q}}, \tag{16}$$

and considering only the motion due to null space velocity term in (15), we have

$$\dot{H} = \nabla^T H(\mathbf{q})GQ_0^{-1}(\mathbf{I} - \mathbf{J}^{\#^+}\mathbf{J}^\#)\mathbf{z} \tag{17}$$

In order to decrease H, we need to ensure $\dot{H} \le 0$. One choice for $\mathbf{z}$ can then be

$$\mathbf{z} = -k_H\Big[\nabla^T H(\mathbf{q})GQ_0^{-1}(\mathbf{I} - \mathbf{J}^{\#^+}\mathbf{J}^\#)\Big]^T \tag{18}$$

where k_H is a positive scalar.

In our proposed planning framework we use the problem redundancy to increase the *manipulability* of the robotic arm only (not that of the whole WMM) through the above local optimization strategy. The rational behind choosing this objective (and not others mentioned above) goes back to our proposed decoupled control strategy in which the robotic arm is servo controlled along the feature trajectories using a closed-loop (image-based) inverse kinematic control; Clearly, maintaining good manipulability for the robotic arm will improve the performance of such controller which strongly relies on the Jacobian inverse. It is also noteworthy that through some trajectories the robotic arm manipulability may be poor whereas the whole WMM system keeps a good measure of manipulability, i.e., improving the latter does not necessary improves the former (see [19]).

In its general form the manipulability measure is defined as $\mu = \sqrt{\left|\mathbf{J}\mathbf{J}^T\right|}$. For a square Jacobian matrix, e.g., the case of 6-DOF robotic arm here, it is equivalent to the arm Jacobian determinant $\mu_a = \left|\mathbf{J}_a\right|$. Letting $H = -\mu_a(\mathbf{q}_a)$ in (18) will favor those null space motions leading to trajectories (or configurations) with good manipulability for the robotic arm. However, it must be noted that due to the local optimization nature of the above scheme, one cannot guarantee its singularity avoidance and its performance may degrade over long trajectories. Hence, the trajectory obtained through (15) needs to be checked for singularities.

IV. Decoupled Trajectory Tracking Scheme

The proposed path planning explained above yields a WMM trajectory $\mathbf{q}^*(t)$, which in turn can be decomposed into the corresponding mobile platform trajectory $\mathbf{q}_p^*(t)$, robotic arm trajectory $\mathbf{q}_a^*(t)$, camera trajectory $\boldsymbol{\xi}^*(t)$, and finally image features trajectories $\mathbf{s}^*(t)$ pertinent to the object features, i.e.,

$$\mathbf{q}^*(t) \rightarrow \mathbf{q}_p^*(t),\ \mathbf{q}_a^*(t),\ \boldsymbol{\xi}^*(t),\ \mathbf{s}^*(t) \qquad (19)$$

The planned trajectories could then be executed in different ways. An ideal and challenging way would be to design a controller that reconciles errors in both C-space and image space and is a future research topic. More simply, one could execute the WMM trajectory directly in the WMM C-space. However, deviation may occur due to uncertainties in the motion of the mobile platform. Using a feedback control strategy would help to reduce the mobile platform tracking error. Nonetheless, even small tracking errors of the mobile platform may result in large deviation of the camera at the end-effector, hence loosing the target in camera field of view. This suggests the need for closing the loop within the image space to account for potential tracking errors of the mobile platform.

Here we propose a decoupled feedback control strategy to move the WMM along the above desired trajectories; we utilize an image-based control technique to servo control the robotic arm along the desired feature trajectories $\mathbf{s}^*(t)$, while a state feedback control is employed to move the mobile platform along its desired trajectory $\mathbf{q}_p^*(t)$ in Cartesian space. These two controllers run simultaneously in two separate threads to move the whole WMM along the planned trajectories.

Remark: It must be noted that successful execution of the planned WMM trajectory using the above decoupled control strategy relies on two assumptions: first, the robotic arm and the mobile platform both are precisely controlled along their respective trajectories, and second, the trajectory tracking threads are synchronized. The preliminary experiments presented in this work were obtained without a synchronized tracking strategy which is the subject of our future research.

A. Image-based control of the arm

Tracking of the desired feature trajectories follows our previous approach developed to control a stationary robotic arm with an eye-in-hand configuration [11]. We define the task tracking error $\mathbf{e}(t) = \mathbf{s}(t) - \mathbf{s}^*(t)$ where $\mathbf{s}(t)$ is the vector of current image features. The tracking control law can be obtained as

$$\dot{\boldsymbol{\xi}}^*(t) = \widehat{\mathbf{L}}^+(-\lambda\mathbf{e} + \dot{\mathbf{s}}^*(t)) \qquad (20)$$

where $\widehat{\mathbf{L}} = \mathbf{L}(\mathbf{s}^*(t), \mathbf{Z}^*(t))$ is a nominal model of the image interaction matrix which is created by stacking up the interaction matrices related to all the current desired image features and their 3D depth $\mathbf{Z}^*(t)$. The desired camera velocity screw $\dot{\boldsymbol{\xi}}^*(t)$ is achieved by applying appropriate joint velocities computed using the robotic arm Jacobian, i.e., $\dot{\mathbf{q}}_a^* = \mathbf{J}_a^{-1}\dot{\boldsymbol{\xi}}^*(t)$ which are sent to the joint-level velocity controllers of the robotic arm. The arm Jacobian, computed in the camera frame, is independent of the mobile platform pose, and is continuously updated using the current arm joint values $\mathbf{q}_a(t)$.

B. State feedback control of the mobile platform

We adapt the non-linear state feedback control law devised by [15] for stabilizing the non-holonomic mobile platform to its planned trajectory $\mathbf{q}_p^*(t) = [x_p^*(t)\ y_p^*(t)\ \theta_p^*(t)]^T$ in the Cartesian space. The linear and angular input velocities $(\omega_p(t),\ v_p(t))$ are calculated as [15]

$$\begin{aligned}\omega_p(t) &= \frac{k_3}{k_2}\tilde{\theta} - \frac{k_1}{k_2}h_2(\tilde{y} + d\tilde{\theta}) + \frac{k_6}{k_2}\tilde{x} &(21)\\ v_p(t) &= k_3k_5\tilde{x} + (2k_3k_4 + h_1 + k_6)\tilde{\theta} &(22)\\ &\quad +[(1-k_1)\tilde{y} - k_1d\tilde{\theta}](\omega_p(t) + \dot{\theta}_p^*(t))\end{aligned}$$

where

- $\tilde{\theta}_p(t) = \theta_p(t) - \theta_p^*(t)$, $\tilde{x}_p(t) = x_p(t) - x_p^*(t)$, and $\tilde{y}_p(t) = y_p(t) - y_p^*(t)$
- $h_1(\tilde{\theta}, t) = v_p^*(t)\frac{\cos\tilde{\theta} - 1}{\tilde{\theta}} + d\theta_p^*(t)\frac{\sin\tilde{\theta}}{\tilde{\theta}}$
- $h_2(\tilde{\theta}, t) = d\theta_p^*(t)\frac{\cos\tilde{\theta} - 1}{\tilde{\theta}} - v_p^*(t)\frac{\sin\tilde{\theta}}{\tilde{\theta}}$
- $v_p^*(t)$ is the desired linear velocity along the planned trajectory $\mathbf{q}_p^*(t)$
- $k_1, \ldots, k_6$ are constant gains

Mobile platform state estimation: As a property common to all such control laws, they require a fairly precise estimate of the mobile robot state (or pose) to ensure local stabilization to the desired trajectory. One can use existing localization techniques such as Monte Carlo localization to localize the base. This might require other types of sensors such as laser scanners to be incorporated, and hence increase the complexity of the overall system. Moreover such approaches are relatively slow, and require multiple scans of the scene to be able to provide a fairly precise estimate of the platform pose.

We implemented a state estimation strategy based on the image feedback from the known target object: we first localize the camera with respect to the target object using the iterative camera pose estimation technique in [20]. The estimated camera pose is then translated back to the arm base pose, i.e., the mobile platform pose, using the known arm kinematic model and its joint encoder data. The estimated pose is then used in the state feedback control law in (21). The above localization process runs in real-time on a separate thread while both the arm and the platform are controlled along their respective trajectories.

The decoupled feedback control scheme proposed above is our first effort to execute the planned trajectories on the mobile manipulator. Other strategies are worth to be compared. For example, one may control the robotic arm directly in the joint space (using, e.g., a simple resolved rate motion controller) along its desired trajectory without closing the loop in the image. However, through this approach the robotic arm is not constrained to track the desired feature trajectories in the image, and even small errors in positioning of the mobile platform will result in deviation of the robotic arm from the desired feature trajectories, and hence loosing the target object in the camera field of view and not being able to localize the platform through the image feedback. Closing the control loop in the image is an interesting advantage of the proposed control strategy which compensates for the possible errors in positioning of the mobile platform, and hence avoids deviation from the desired feature trajectories.

V. Preliminary Experiments and Results

We implemented our proposed kinodynamic planning approach and feedback control schemes on a wheeled mobile manipulator system which consists a Powerbot mobile platform with an on-board 6-DOF Schunck robotic arm with a Bumblebee2 camera mounted at its end-effector (see Fig. 5). Both the arm and the mobile platform are equipped with low-level velocity controllers which facilitates the implementation of our proposed image and state feedback trajectory tracking controllers.

We used the ViSP visual servoing platform [21] to implement the proposed image-based trajectory tracking for the arm. We used the camera pose estimation using virtual visual servoing [20] provided by ViSP for the real-time pose estimation of the mobile platform as we explained in previous section.

Here we present the results of our first experiments on the WMM system to perform a visually guided task: moving from an initial to a desired location while avoiding image and physical constraints. The initial pose of the mobile platform is estimated using the initial view, and the final configuration of the WMM is implicitly defined by the desired view at the four coplanar dots (Fig. 3) on the target object. Please note that the final configuration of the WMM falls out as the result of the planning which may vary on different runs but will yield the same desired view which is the objective of the task.

As shown in Fig. 4, the desired location of the camera is located in between the long obstacles on the sides of the target dots. Although the setup of this experiment may seem simple it is indeed a hard planning problem. To reach the desired location, the WMM system needs to move around the obstacle in front of it while keeping the feature in view and making sure that the long obstacles do not occlude its field of view. Planning in such a constrained space is very challenging. Our proposed kinodynamic planner succeeds to find a trajectory in about 200 seconds. The planned camera tree and its final trajectory are shown in Fig.4. The planned camera trajectory was then projected into the image space to obtain the feature trajectories as shown in Fig. 3(left).

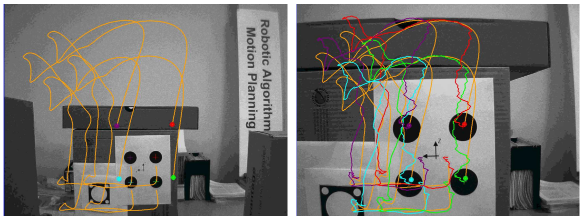

Fig. 3. Left: initial view and the desired feature trajectories, right: desired final view, and the followed feature trajectories

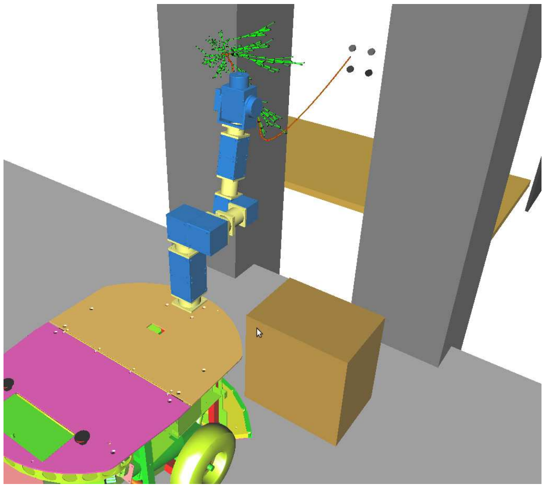

Fig. 4. Planning environment visualized in OpenRAVE: camera tree (in green) and the planned trajectory (in red)

Figure 5 shows the snapshots of the WMM tracking the planned trajectories using the proposed decoupled control scheme. As shown the WMM manages to move around the obstacle at the front while keeping the features in the field of view and avoiding occlusion/collision due to obstacles. The followed feature trajectories are shown in Fig. 3(right). The video of this experiment is available on-line at `http://youtu.be/m6HCwEctxj0`

VI. Discussions

The experiment presented above is our first effort on running the planned trajectories using our proposed decoupled feedback control scheme. Through this experiment we had a few observations which lead to our future research and developments:

- Developing a synchronization strategy between the motion of the mobile platform and the arm is very crucial: we observed that due to the lag of the mobile platform in responding to velocity control inputs (up to 5 cm/degrees in position/orientation) the arm deviates

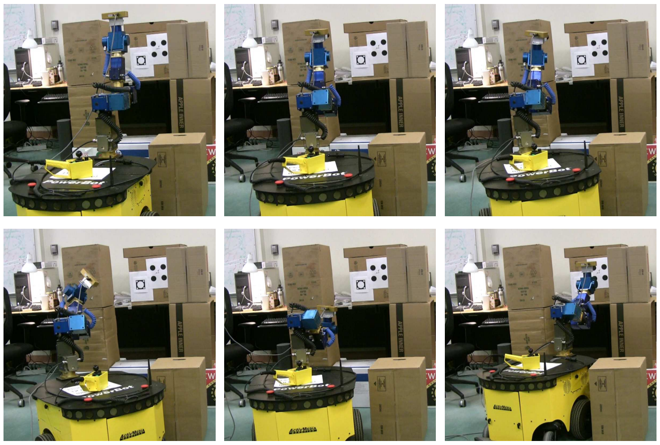

Fig. 5. (From left to right) snapshots of the wheeled mobile manipulator system following the planned trajectories to reach the desired location while avoiding collision/occlusion due to obstacles and keeping the target in the field of view.

from its desired trajectory (up to 20 degrees for some joints) to compensate for the platform tracking error while achieving its feature trajectory tracking task in image space.

- The performance of the mobile platform state estimation strategy degrades due to shaky movements of the platform in particular at the beginning of the trajectory and during the turns. Moreover, for a better estimation accuracy the image features should be spread in the image which requires the camera to be close to the target. In the above experiment the camera was initially located about 150 cm away from the target. For distance above this threshold, or when the platform shakes during the tracking we had to use the on-board odometry to maintain an estimate of the platform state. One interesting direction is to combine the odometry with the image-based state estimation for a better performance.
- Finally, although the error dynamics and stability of each individual controller used in our decoupled control scheme has been studied previously ([15] [22]), the overall stability and performance of the proposed decoupled control scheme requires further studies.

VII. Acknowledgments

This research was funded, in part, by NSERC discovery grants to Kamal Gupta and Mehran Mehrandezh.

References

[1] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control," *IEEE Trans. Robot. Autom.*, vol. 12, no. 5, pp. 651–70, 1996.

[2] F. Chaumette, "Potential problems of stability and convergence in image-based and position-based visual servoing," in *Confluence of Vision and Control*, D. Kriegman, G. Hager, and A. Morse, Eds. Springer Berlin / Heidelberg, 1998, pp. 66–78.

[3] M. Kazemi, K. Gupta, and M. Mehrandezh, "Path planning for visual servoing: A review and issues," in *Visual Servoing via Advanced Numerical Methods*, G. Chesi and K. Hashimoto, Eds. Springer, Mrach 2010, ch. 11, pp. 189–208.

[4] K. Hosoda, K. Sakamoto, and M. Asada, "Trajectory generation for obstacle avoidance of uncalibrated stereo visual servoing without 3d reconstruction," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 1995, pp. 29–34.

[5] V. Kyrki, D. Kragic, and H. Christensen, "New shortest-path approaches to visual servoing," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2004, pp. 349–54.

[6] Y. Mezouar and F. Chaumette, "Path planning for robust image-based control," *IEEE Trans. Robot. Autom.*, vol. 18, no. 4, pp. 534–49, 2002.

[7] L. Deng, F. Janabi-Sharifi, and W. Wilson, "Hybrid motion control and planning strategies for visual servoing," *IEEE Trans. Ind. Electron.*, vol. 52, no. 4, pp. 1024–40, 2005.

[8] G. Chesi and Y. Hung, "Global path-planning for constrained and optimal visual servoing," *IEEE Trans. Robot.*, vol. 23, no. 5, pp. 1050–60, 2007.

[9] H. Zhang and J. Ostrowski, "Visual motion planning for mobile robots," *IEEE Trans. Robot. Autom.*, vol. 18, no. 2, pp. 199–208, 2002.

[10] M. Kazemi, K. Gupta, and M. Mehrandezh, "Global path planning for robust visual servoing in complex environments," in *IEEE Int. Conf. on Robot. and Automat.*, May 2009, pp. 326 – 332.

[11] M. Kazemi, M. Mehrandezh, and K. Gupta, "Kinodynamic planning for visual servoing," in *IEEE Int'l Conf. on Robotics and Automation*, 2011, pp. 2478 –2484.

[12] R. Pissard-Gibollet and P. Rives, "Applying visual servoing techniques to control a mobile hand-eye system," in *IEEE Int'l Conf. on Robotics and Automation*, 1995, pp. 166 –171.

[13] A. De luca, G. Oriolo, and P. r. Giordano, "Image-based visual servoing schemes for nonholonomic mobile manipulators," *Robotica*, vol. 25, pp. 131–145, March 2007.

[14] D. Kragic, L. Petersson, and H. Christensen, "Visually guided manipulation tasks," *Robotics and Autonomous Systems*, vol. 40, no. 2-3, pp. 193 – 203, 2002.

[15] C. Samson and K. Ait-Abderrahim, "Feedback control of a nonholonomic wheeled cart in cartesian space," in *IEEE Int'l Conf. on Robotics and Automation*, 1991, pp. 1136 –1141.

[16] Z. Yao and K. Gupta, "Path planning with general end-effector constraints," *Robot. Auton. Syst.*, vol. 55, no. 4, pp. 316–27, 2007.

[17] S. LaValle and J. Kuffner, "Randomized kinodynamic planning," *Int. J. Robot. Res.*, vol. 20, no. 5, pp. 378–400, 2001.

[18] Y. Nakamura, *Advanced Robotics: Redundancy and Optimization*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1990.

[19] B. Bayle, J.-Y. Fourquet, and M. Renaud, "Manipulability of wheeled mobile manipulators: application to motion generation," *Int. J. Robot. Res.*, vol. 22, no. 7-8, pp. 565 – 81, 2003.

[20] E. Marchand and F. Chaumette, "Virtual visual servoing: a framework for real-time augmented reality," in *Proc. EUROGRAPHICS Conf.*, vol. 21, no. 3, Saarebrcken, Germany, 2002.

[21] E. Marchand, F. Spindler, and F. Chaumette, "ViSP for visual servoing: a generic software platform with a wide class of robot control skills," *IEEE Robot. Autom. Mag.*, vol. 12, no. 4, pp. 40 –52, Dec. 2005.

[22] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Trans. Robot. Autom.*, vol. 8, no. 3, pp. 313–26, 1992.