Control of Mobile Manipulator using the Dynamical Systems Approach

Lars-Peter Ellekilde Henrik I. Christensen

***Abstract*— The combination of a mobile platform and a manipulator, known as a mobile manipulator, provides a highly flexible system, which can be used in a wide range of applications, especially within the field of service robotics. One of the challenges with mobile manipulators is the construction of control systems, enabling the robot to operate safely in potentially dynamic environments. In this paper we will present work in which a mobile manipulator is controlled using the dynamical systems approach. The method presented is a two level approach in which competitive dynamics are used both for the overall coordination of the mobile platform and the manipulator as well as the lower level fusion of obstacle avoidance and target acquisition behaviors.**

I. INTRODUCTION

The majority of robotic research has in the last decades focused on either mobile platforms or manipulators, and there have been many impressive results within both areas. Today one of the new challenges is to combine the two areas, into systems, which are both highly mobile and have the ability to manipulate the environment. Especially within service robotics there will be an increased need for such systems. The demography of most western countries causes the number of old people in need of care to increase, while there will be less working to actually support them. This requires an increased automation of the service sector, for which robots able to operate safely in indoor and dynamic environments are essential.

The platform used in this work is shown in Figure 1, and consist of a Segway RMP200 with a Kuka Light Weight Robot. The result is a platform that has a relative small footprint and is highly maneuverable, making it well suited for moving around in an indoor environment. The Kuka Light Weight Robot has a fairly long reach and high payload compared to its own weight, making it ideal for mobile manipulation.

When controlling a mobile manipulator, there is a choice of whether to consider the system as one or two entities. In [1] and [2] they derive Jacobians for both the mobile platform and the manipulator and combine them into a single control system. The research reported in [3] and [4], on the other hand, considers them as separate entities when planning, but do include constraints, such as reachability and stability, between the two.

The control system we propose is based on the dynamical systems approach [5], [6]. It is divided into two levels, where we at the lower level consider the mobile platform and the manipulator as two separate entities, which are then combined in a safe manner at the upper level. The overall architecture is described in Section II, followed by the control of the mobile platform and the manipulator in Sections III and IV. In Section V we will show some experiments before concluding the paper in Section VI. However, first a summary of work related to the dynamical systems approach will be provided in Section I-A.

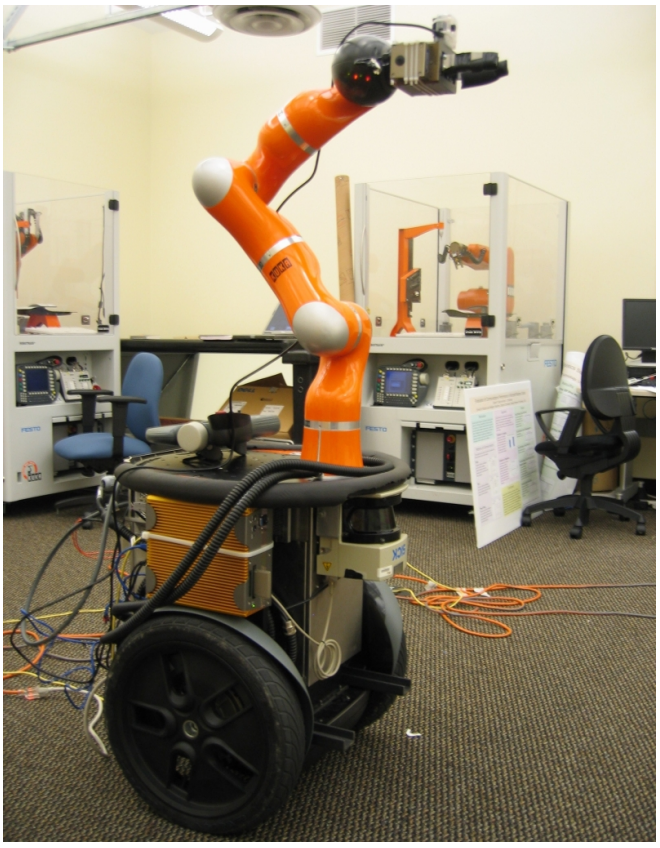


Fig. 1. Platform consisting of a Segway RMP200 and a Kuka Light Weight Robot.

A. Related Work

The dynamical systems approach [5], [6] provides a framework for controlling a robot through a set of behaviors, such as obstacle avoidance and target acquisition. Each behavior is generated through a set of attractors and repellors of a non-linear dynamical system. These are combined through simple addition of the vector fields to provide the overall behavior of the system. The dynamical systems approach relates to the more widely used potential field method [7], but has certain

L.-P. Ellekilde is with The Maersk Mc-Kinney Moller Institute, Faculty of Engineering, University of Southern Denmark, Campusvej 55, 5230 Odense M, Denmark, lpe@mmmi.sdu.dk

H. I. Christensen is with Center for Robotics and Intelligent Machines, Interactive Computing / College of Computing, Georgia Institute of Technology, 85 5th Street, Atlanta, GA, USA, hic@cc.gatech.edu

advantages. Where the behavior in the potential field method is the result of following gradients of the field, the behavior variables, such as heading direction and velocity, can be controlled directly using the dynamical systems approach.

The relative low computational cost associated with the approach, makes it suitable for online control in dynamic environments, and allows it to be implemented even on fairly low-level platforms with limited computational capabilities [8]. The robustness to noisy sensors is shown in [9] and [10] where a combination of infrared sensors and microphones is used for obstacle avoidance and target acquisition. Despite being able to solve various tasks it is only a local method, for task and mission-level planning other methods (see e.g. [11]) should be applied.

A drawback of the approach in [5], [6] is the potential creation of spurious attractors when multiple behaviors are combined. To overcome this problem [12] introduces a weighting of the behaviors based on competitive dynamics. The influence of each behavior is controlled using an associated competitive advantage, which together with competitive interactions defined between the behaviors, controls the weights. This approach generalizes to an arbitrary number, n, behaviors, but with a $\mathcal{O}(n^2)$ worst-case complexity, if competitive interactions between all behaviors are needed.

Real-world indoor experiments using this competitive dynamics approach can be found in [13], [14]. In [13] only the heading direction of the vehicle is used, whereas in [14] both heading direction and velocity are controlled. [15] provides a brief discussion of strategies for the velocity behavior.

The dynamical systems approach has not only been used for planar mobile robots, but also for controlling the tool motion of a manipulator [16]. More complex dynamical systems using the Hopf Oscillator for generating limit cycles can also be used. [17] shows how limit cycles with different shapes can be constructed and used for both obstacle avoidance and trajectory generation. [18] uses the Hopf Oscillator to generate a timed trajectory, enabling a manipulator to catch a ball rolling down a table. The dynamical systems approach can not only be used for controlling the tool, but also to control the redundancy of a 7 degrees of freedom manipulator as demonstrated in [19].

II. Overall Architecture

The overall architecture of our system is illustrated in Figure 2. To control the mobile platform, in this case a Segway, two low level behaviors are use: One for target acquisition and one for obstacle avoidance. Using competitive dynamics these are fused together to provide the *Mobile* behavior, which specifies the desired motion of the mobile platform. Similarly we have target acquisition and obstacle avoidance behaviors for the manipulator fused together based on competitive dynamics, to give the *Manipulator Acquisition* behavior. When the target is not within reach, the manipulator should retract to a safe configuration, which is the purpose of the *Manipulator Retract* behavior. The last fusion combines the controls in a safe manner, such that the target acquisition and retract behaviors do not disturb one another and the mobile platform does not start moving towards a new target before the manipulator has been retracted.

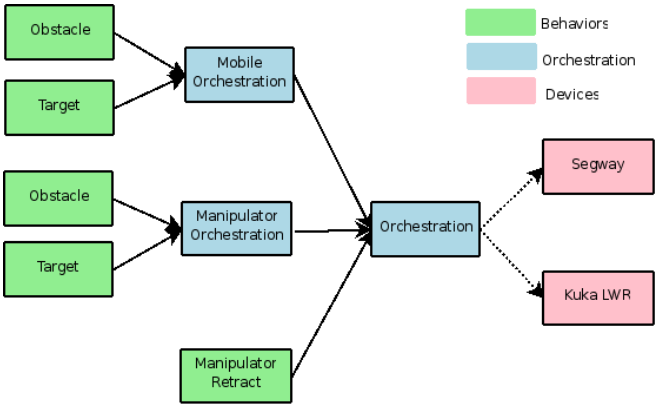


Fig. 2. Overall architecture of the control system

Using weights w^{mobile}, $w^{manip}_{acquisition}$ and $w^{manip}_{retract}$ to represent the influence of the *Mobile*, *Manipulator Acquisition* and *Manipulator Retract* behaviors, the control signals $\mathbf{u}^{mobile}$ and $\dot{\mathbf{q}}^{manip}$ for the mobile platform and the manipulator are given by

$$\mathbf{u}^{mobile} = w^{mobile} \begin{pmatrix} u_{left} \\ u_{right} \end{pmatrix} \quad (1)$$

$$\dot{\mathbf{q}}^{manip} = w^{manip}_{acquisition} \dot{\mathbf{q}}^{manip}_{acquisition} + w^{manip}_{retract} \dot{\mathbf{q}}^{manip}_{retract} \quad (2)$$

where (u_{left}, u_{right}) are control inputs to the left and right wheels of the platform as described in Section III, and $\dot{\mathbf{q}}^{manip}_{acquisition}$, $\dot{\mathbf{q}}^{manip}_{retract}$ are the manipulator joint velocities as described in Section IV.

A. Competitive Dynamics

The competitive dynamics approach used is based on [12], but with the additional parameter τ_b used to control the transition rate as in [14]. The dynamical system used is thus given by

$$\tau_b \dot{w}_b = \alpha_b (w_b - w_b^3) - \sum_{b' \neq b} \gamma_{b',b} w_{b'}^2 w_b + noise \quad (3)$$

in which α_b is the competitive advantage of behavior b and $\gamma_{b',b}$ is the competitive interaction of behavior b' upon b.

1) Mobile: The competitive advantages of the mobile platform should strengthen the behavior when far away from the target and reduce it when the target is reached. This is achieved through

$$\alpha^{mobile} = \tanh(k^{mobile}_{\alpha}(d_{tar} - d^{mobile}_{threshold})) \quad (4)$$

in which k^{mobile}_{α} determines how rapidly the advantage should change, d_{tar} is the distance to the target and $d^{mobile}_{threshold}$ specifies a minimum distance to the target required before the mobile platform should move.

The mobile behavior has no ability to interact and suppress other behaviors, thus its competitive interactions are set to 0.

2) Manipulator Acquisition: This behavior should be strengthened when the mobile platform gets close to its target. The competitive advantage will thus be defined as

$$\alpha^{manip}_{acquisition} = -\tanh(k^{manip}_{\alpha}(d_{tar} - d^{manip}_{threshold})) \quad (5)$$

The activation distance $d^{manip}_{threshold}$ must be greater than $d^{mobile}_{threshold}$ to make sure the behavior is activated. This behavior has no direct interaction with the others, thus its interactions are set to 0.

3) Manipulator Retract: The retract behavior should be activated opposite the goal behavior, hence

$$\begin{aligned}\alpha^{manip}_{retract} &= -\alpha^{manip}_{acquisition} \\ &= \tanh(k^{manip}_{\alpha}(d_{tar} - d^{manip}_{threshold}))\end{aligned} \quad (6)$$

Except for a very small transition time this prevents the manipulators acquisition and retract behaviors from being active at the same time, thus we can set $\gamma_{retract,acquisition} = 0$. For the interaction between the retract and the mobile behaviors we wish retract to deactivate mobile when the manipulator is far away from its home configuration. The interaction is therefore defined as

$$\gamma_{retract,mobile} = \frac{1}{2}(1 + \tanh(k^{retract}_{\gamma}(\|\mathbf{q}_{current} - \mathbf{q}_{home}\| - \epsilon_q))) \quad (7)$$

in which $\mathbf{q}_{cur}$ and $\mathbf{q}_{home}$ are the manipulators current and home configurations, ϵ_q specifies a proximity distance around $\mathbf{q}_{home}$ and $k^{retract}_{\gamma}$ specifies how quickly the interaction changes.

III. Control of the Mobile Platform

The control of the mobile platform is constructed very similar to what is presented in [14], but with a few differences. First of all only the target acquisition and obstacle avoidance behaviors are used. The corridor following and wall avoidance are not included, but would be straight forward extensions. The second area in which this work differs is in how the density of obstacles is calculated. Details of this will be explained in section III-D.

For the control to actually be able to navigate through the environment, it is necessary with a method for localization. The approach we have used is based on dominating lines, and is described in Section III-E.

The control of the platform is encoded using the orientation, ϕ, and the velocity, v, which results in a system with control inputs $\mathbf{f}^{mobile} = \{\dot{\phi}, \dot{v}\}$. The values of $\mathbf{f}^{mobile}$ are made up of two parts, $\mathbf{f}^{mobile}_{tar}$ and $\mathbf{f}^{mobile}_{obs}$, which are combined as

$$\mathbf{f}^{mobile} = w^{mobile}_{tar}\mathbf{f}^{mobile}_{tar} + w^{mobile}_{obs}\mathbf{f}^{mobile}_{obs} \quad (8)$$

where the weights w^{mobile}_{tar} and w^{mobile}_{obs} are controlled using Eq. (3) with the competitive advantage and interactions described in section III-C.

As control input we need expressions for the left and right wheels of the mobile platform, denoted u_{left} and u_{right}, respectively. To obtain these $\dot{v}$ is integrated to get v, which together with the desired rotational velocity $\dot{\phi}$, the wheel diameter d_{wheel} and the distance between the wheels $d_{wheelbase}$ can be used to calculate the control inputs as

$$u_{left}(\dot{\phi}, v) = \frac{v}{\pi d_{wheel}} - \frac{\Delta}{2} \quad (9)$$

$$u_{right}(\dot{\phi}, v) = \frac{v}{\pi d_{wheel}} + \frac{\Delta}{2} \quad (10)$$

$$(11)$$

where Δ is the needed difference in wheel speed given by

$$\Delta = \frac{\dot{\phi} d_{wheelbase}}{d_{wheel}\pi} \quad (12)$$

A. Target Dynamics

The basic dynamics of this target behavior is

$$f^{mobile,\phi}_{tar}(\phi) = \lambda^{mobile,\phi}_{tar} sin(\psi_{tar} - \phi) \quad (13)$$

$$f^{mobile,v}_{tar}(v) = \lambda^{mobile,v}_{tar}(min(k^{mobile}_{tar} d_{tar}, v_{max}) - v) \quad (14)$$

in which $\lambda^{mobile,\phi}_{tar}$ and $\lambda^{mobile,v}_{tar}$ are the strengths of the attractors and ψ_{tar} is the direction to the target. The constant k^{mobile}_{tar} gives the relation between the distance to the target and the desired velocity. Finally v_{max} is the maximal velocity allowed for the mobile platform.

B. Obstacle Dynamics

Given a distance $d_{obs,i}$ and a direction ψ_i to the i'th obstacle, the dynamics of the obstacle avoidance are

$$f^{mobile,\phi}_{obs,i} = \lambda^{mobile,\phi}_{obs}(\phi - \psi_i) e^{-c^{mobile}_{obs} d_{obs,i}} e^{\frac{-(\phi-\psi_i)^2}{2\sigma_i^2}} \quad (15)$$

$$f^{mobile,v}_{obs,i} = \begin{cases} -\lambda^{mobile,v}_{obs}(v - v_{min}) & \text{for } v < v_{min} \\ 0 \quad \text{for } v_{min} \le v \le v_{max,i} \\ -\lambda^{mobile,v}_{obs}(v - v_{max,i}) & \text{for } v > v_{max,i} \end{cases} \quad (16)$$

where $v_{max,i} = \max(k_{obs} d_{obs,i}, v_{min})$.

The dynamics of ϕ consists of 3 elements: (i) The relative direction to the obstacle $(\phi - \psi_i)$, (ii) a scale $e^{-c^{mobile}_{obs} d_{obs,i}}$ in which c^{mobile}_{obs} determines the decay depending of the distance, $d_{obs,i}$, and (iii) a scale, $e^{\frac{-(\phi-\psi_i)^2}{2\sigma_i^2}}$, based on the direction to the obstacle and with $\sigma_i = \arcsin(\frac{1+D_s}{1+d_{obs,i}})$ ensuring the generation of an attractor between two obstacles if the robot can pass through while ensuring the safety distance D_s. See [14] for more details.

For $f^{mobile,v}_{obs,i}$ the expression adjusts the velocity towards $k_{obs} d_{obs,i}$, but ensures that a minimum velocity of v_{min} is kept.

To obtain the value of $\mathbf{f}^{mobile}_{obs}$ we sum over all obstacles

$$\mathbf{f}^{mobile}_{obs} = \begin{pmatrix} f^{mobile,\phi}_{obs} \\ f^{mobile,v}_{obs} \end{pmatrix} = \sum_i \begin{pmatrix} f^{mobile,\phi}_{obs,i} \\ f^{mobile,v}_{obs,i} \end{pmatrix} \quad (17)$$

C. Competitive Dynamics

The weights for the competitive dynamics are controlled by equation (3) as explained above. Below are the competitive advantages and interactions for the two behaviors.

1) Target: The competitive advantage is set to $\alpha_{tar}^{mobile} = 0.5$ whenever a target is present, otherwise $\alpha_{tar}^{mobile} = -0.5$.

The target behavior has the ability to interact with and suppress the obstacle avoidance behavior, when the ratio between the distance to the target and the closest object is sufficient to ensure the movement towards the target will be collision free. This is modeled as

$$\gamma_{tar,obs}^{mobile} = \frac{1}{2}(1 + \tanh(r_{gain}^{mobile}(\frac{d_{obs,min}}{d_{tar}} - r_{limit}^{mobile}))) \quad (18)$$

in which $d_{obs,min}$ is the distance to the closest obstacle, r_{gain}^{mobile} is a gain constant giving how quickly the behavior should interact and $r_{limit}^{mobile} \geq 1$ expresses the ratio between the distances to obstacle and target for which we will start to suppress obstacle avoidance.

2) Obstacle: The competitive advantage of the obstacle behavior is given by

$$\alpha_{obs}^{mobile} = \tanh(\frac{\rho^{mobile} - \rho_0^{mobile}}{\rho_0^{mobile}}) \quad (19)$$

In which ρ^{mobile} is the obstacle density as defined in Section III-D.

The interaction is defined as

$$\gamma_{obs,tar}^{mobile} = \frac{1}{2}(1 + \tanh(\frac{\rho - \rho_0}{\sigma_\rho}))(1 - \gamma_{tar,obs}^{mobile}) \quad (20)$$

The first part, $\frac{1}{2}(1 + \tanh(\frac{\rho^{mobile} - \rho_0^{mobile}}{\sigma_\rho}))$ suppresses the target behavior when the obstacle density exceeds the threshold ρ_0^{mobile}. The last part, $(1 - \gamma_{tar,obs}^{mobile})$, makes sure this only happens when the obstacle avoidance is not being suppressed due to $\gamma_{tar,obs}^{mobile}$.

D. Calculation of Obstacle Density

Given a set of distances, $d_{obs,i}$, between the mobile platform and obstacles the density, ρ, is calculated as

$$\rho = \max_i \frac{1}{d_{obs,i}} \quad (21)$$

This density function differs from [14] in which $\rho = \sum_i e^{-d_{obs,i}}$ is used. The main problem with this formulation is that we cannot distinguish between many objects relative far away and a single object close by. For example having 5 objects 2 meters away will give the same density as a single object 40 centimeters away. With the exponential function a single object in the scene can never cause ρ to exceed 1. The threshold for switching to the obstacle avoidance behavior will thus have to be less than 1, but given a scene with multiple obstacles the threshold of 1 will often be too low.

Furthermore it is found that using $\frac{1}{d_{obs,i}}$ instead of $e^{-d_{obs,i}}$ made tuning the parameters easier as we could now think of the density as the inverse of the distance. It also caused the density to grow very rapidly when getting close to an obstacle, thereby quickly forcing the behaviors to change.

E. Localization

For the localization we assume to have a set of domination lines in the environment, which we will be able to identify with a laser scanner. We can then use the work in [20], which through an Extended Kalman Filter (EKF) combines input from a laser scanner and odometry to provide an estimate of the platforms position and orientation. A short description of the approach follows, but a thorough description will be outside the scope of this paper.

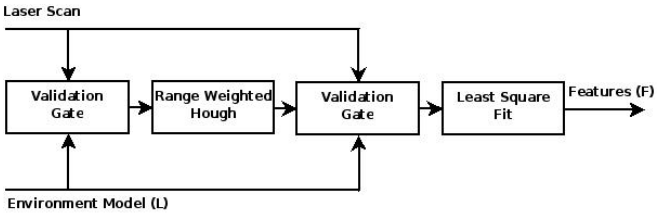


Fig. 3. Localization feature extraction procedure

The input from odometry is handled straight forward using the platform kinematics and with a covariance based on real world experiments. The update based on laser range data takes a model of the environment as a set of lines, $\mathcal{L} = l_1, \ldots, l_n$, and a scan, S. The feature extraction is organized, as illustrated in Figure 3, by first applying a relative wide validation gate to obtain all points which might be associated with a line l_i. Using a Range Weighted Hough Transform [21] a line is fitted, which is then used as input to a narrower validation gate. All points making it through this gate are then assumed to be on the line and are used to compute a least square fit [22], which provides the feature for the EKF. With respect to the covariance of the line, [20] reports little difference between using a computational expensive first order approximation or a simpler estimate, which is thus the one used in this work.

IV. Control of Manipulator

We will start by dividing the problem into two parts:

1) Determining the motion of the tool from the current position to the target while avoiding obstacles.
2) Inverse kinematics calculating joint velocities needed for the tool motion.

The second part is a well understood problem, which in this work is solved using the inverse kinematics strategy presented in [23]. This method incorporates both kinematics and dynamics limitations of the robot, such as joint position, velocity and acceleration limits. Furthermore this approach, based on quadratic optimization, has shown to be very robust with respect to singularities.

The motion of the tool is controlled using the manipulators *Target* and *Obstacle* behaviors, to which the weights w_{tar}^{manip} and w_{obs}^{manip} are associated. As input the inverse kinematics needs a 6D velocity screw $\nu = \{\dot{x}, \dot{y}, \dot{z}, \omega_x, \omega_y, \omega_z\}$, thus the behaviors must find a change $\mathbf{f}^{manip} = \{\ddot{x}, \ddot{y}, \ddot{z}, \dot{\omega}_x, \dot{\omega}_y, \dot{\omega}_z\}$, which can be integrated to give a desired tool velocity, ν_{des},

as

$$\nu_{des} = \nu_{cur} + dt(w_{tar}^{manip}\mathbf{f}_{tar}^{manip} + w_{obs}^{manip}(\mathbf{f}_{obs,dir}^{manip} + \mathbf{f}_{obs,dist}^{manip})) \quad (22)$$

where $\mathbf{f}_{tar}^{manip}$, $\mathbf{f}_{obs,dir}^{manip}$ and $\mathbf{f}_{obs,dist}^{manip}$ are the contributions from the target and obstacle avoidance behaviors.

A. Target Behavior

The inputs to the target behavior are the current and desired tool transformations $\mathbf{T}_{cur}$ and $\mathbf{T}_{des}$. From these we can compute a desired 6D velocity-screw ν_{tar}. To avoid requiring unrealistic fast motions ν_{tar} is scaled such that $\|\{\dot{x},\dot{y},\dot{z}\}\|_{tar} \leq v_{max}$ and $\|\{\omega_x,\omega_y,\omega_z\}\|_{tar} \leq \omega_{max}$, where v_{max} and ω_{max} here represent the maximal allowed linear and rotational velocities of the tool.

Calculating

$$\mathbf{f}_{tar}^{manip} = -\lambda_{tar}^{manip}(\nu_{cur} - \nu_{tar}) \quad (23)$$

we obtain a desired change to the current velocity.

B. Obstacle-Behavior

As input the obstacle avoidance behavior takes the current Cartesian velocity, $\mathbf{v} = \{\dot{x},\dot{y},\dot{z}\}$, and a set of closest obstacles as vectors, $\mathbf{n}_i \in \mathbb{R}^3$, giving direction and distance between tool and obstacle i. We now wish to compute a change to the Cartesian velocity based on the direction and distance to obstacles, denoted $\mathbf{f}_{obs,dir}^{manip}$ and $\mathbf{f}_{obs,dist}^{manip}$, respectively.

1) Dynamics for Direction: From the current velocity of the tool, $\mathbf{v}$, and the vector $\mathbf{n}_i$ we compute the angle θ_i, between the two as

$$\theta_i = \arcsin\left(\frac{\|\mathbf{v}\times\mathbf{n}_i\|}{\|\mathbf{v}\|\|\mathbf{n}_i\|}\right) \quad (24)$$

The size of the change in direction of the tool is then calculated as

$$\dot{\theta} = \lambda_{obs}^{manip,\theta}|\frac{\pi}{2} - \theta_i|e^{-c_{obs}^{manip}\|\mathbf{n}_i\|}e^{-\frac{\theta_i^2}{2\sigma_\theta^2}} \quad (25)$$

in which $\lambda_{obs}^{manip,\theta}$ is the strength of the repellor, c_{obs}^{manip} control the decay based the distance and σ_θ controls the relation with the angle to the obstacle.

$\dot{\theta}$ is then used to calculate a desired change in the direction of the tool as

$$\omega_i = \dot{\theta}\frac{\mathbf{v}\times\mathbf{n}_i}{\|\mathbf{v}\times\mathbf{n_i}\|} \quad (26)$$

Summing up the contributions from all obstacles we can calculate the change in motion of the tool based on direction to obstacles as

$$\mathbf{f}_{obs,dir}^{manip} = (\sum_i \omega_i)\times\mathbf{v} \quad (27)$$

2) Dynamics for Velocity: The dynamics of the velocity are controlled similar to Eq. (16). The contribution of obstacle i is

$$f_{obs,vel,i}^{manip} = \begin{cases} -\lambda_{obs}^{manip,\nu}(\|\mathbf{v}\| - v_{min}^{manip}) \\ \qquad\qquad \text{for } \|\mathbf{v}\| < v_{min}^{manip} \\ 0 \quad \text{for } v_{min}^{manip} \leq \|\mathbf{v}\| \leq v_{max,i} \\ -\lambda_{obs}^{manip,\nu}(\|\mathbf{v}\| - v_{max,i}) \\ \qquad\qquad \text{for } \|\mathbf{v}\| \geq v_{max,i} \end{cases} \quad (28)$$

with $v_{max,i} = \max(k_{obs}^{manip}\|\mathbf{n}_i\|, v_{min})$. Summing up over all obstacles the total contribution becomes

$$\mathbf{f}_{obs,vel}^{manip} = \sum_i f_{obs,vel,i}^{manip}\frac{\mathbf{v}}{\|\mathbf{v}\|} \quad (29)$$

C. Competitive Dynamics

1) Target Behavior: As for the mobile platform the competitive advantage of the target behavior is set to 0.5 when a target is present and -0.5 otherwise.

The competitive interaction of the target upon the obstacle behavior is again designed such that when the ratio between the distance to the target and to the nearest obstacle is greater then the threshold r_{limit}^{manip}, the obstacle avoidance is suppressed. This is accomplished by

$$\gamma_{tar,obs}^{manip} = \frac{1}{2}(1 + \tanh(r_{gain}^{manip}\left(\frac{\min_i(\|\mathbf{n}_i\|)}{d_{tar}^{tool}} - r_{limit}^{manip}\right))) \quad (30)$$

in which d_{tar}^{tool} is the distance between the tool and the target and r_{gain}^{mobile} is a gain factor specifying how quickly to change the value of $\gamma_{tar,obs}^{manip}$.

2) Obstacle Behavior: The competitive advantage of the obstacle behavior is the same as in Section III-C,

$$\alpha_{obs}^{manip} = \tanh(\frac{\rho^{manip} - \rho_0^{manip}}{\rho_0^{manip}}) \quad (31)$$

with the density calculated using Eq. (21), but with distances between obstacles and tool instead of obstacles and the mobile platform.

The competitive interaction is defineds as

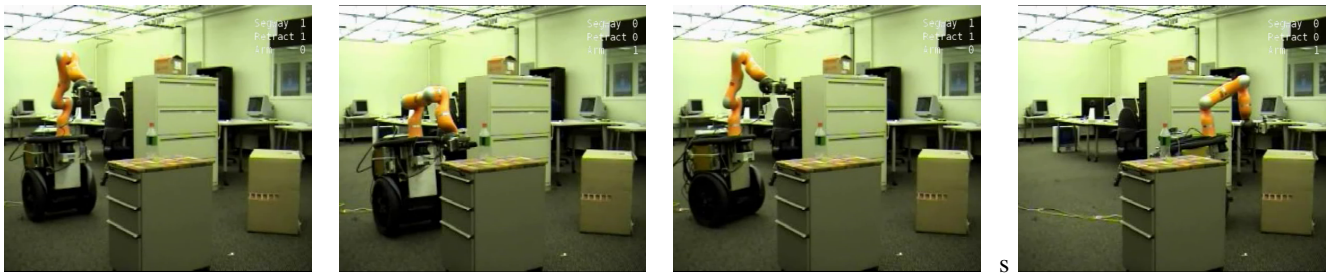
$$\gamma_{obs,tar}^{manip} = \frac{1}{2}(1 + \tanh\left(\frac{\rho^{manip} - \rho_0^{manip}}{\sigma_\rho}\right))(1 - \gamma_{tar,obs}^{manip}) \quad (32)$$

in which the $(1 - \gamma_{tar,obs}^{manip})$ term helps to deactivate the obstacle avoidance as the tool gets close to the target.

D. Retract Behavior

The retract behavior is operating directly in joint space. By defining $\Delta\mathbf{q} = \mathbf{q}_{home} - \mathbf{q}_{cur}$, where $\mathbf{q}_{home}$ is the home configuration to which it should retract, we can calculate the joint velocities as

$$\dot{\mathbf{q}}_{retract} = \lambda_{retract}^{manip}\frac{\Delta\mathbf{q}}{\|\Delta\mathbf{q}\|}\min\left(\frac{\dot{q}_{max}}{\lambda_{retract}^{manip}}, \|\Delta\mathbf{q}\|\right) \quad (33)$$



(a) Move to target (time = 0s) (b) Visual servoing (time = 28s) (c) Move to place position (time = 40s) (d) Place item (time = 72s)

Fig. 4. Experiment with the mobile manipulator

where $\dot{q}_{max}$ is the maximal velocity of the joints and $\lambda_{retract}^{manip}$ is strength of the attractor.

V. Experiments

The platform used in the experiments is shown in Figure 1 and consists of a Segway RMP200 and a Kuka Light Weight Robot equipped with a Schunk PG70 parallel gripper. The platform has a SICK LMS291 laser scanner for localization and obstacle avoidance and a tool mounted Unibrain Fire-i FireWire camera, used for aligning the gripper to the target. Unfortunately we did not have enough time to connect to the gripper and actually grasp the object. It thus only aligns and prepares to grasp, but never actually closes the gripper. As control framework we have used Microsoft Robotics Studio 1.5, which provided us with a tool for organizing the various inputs from sensors, outputs to actuators and ensuring concurrency of the different control algorithms.

The motion of the Segway and the majority of the manipulator movement are based on a specified Cartesian location of the target. However, once the target is within view of the tool-mounted camera, the guidance of the manipulator switches to rely on the visual input. Section V-A will explain details about the visual servoing approach, followed by Section V-B, which provides the test results.

A. Visual Servoing

For the final alignment of the gripper an eye-in-hand image based visual servoing approach is used. Feature extraction is done using the *SimpleVision* service in Microsoft Robotics Studio, which is able to identify colored blobs. In these experiments we are tracking a green marker as illustrated in Figure 5. We wish the orientation of the tool to be fixed, thus only the 3 degrees of freedom (dof) associated with the position should be influence by the servoing. Two of these dof are controlled using the location of the blob, which should be centered in the image. The last dof is controlled by the size of the blob.

B. Test Results

The task of the mobile manipulator is, as illustrated in Figure 4, to move a bottle from the table in the middle of the image to the box located to the far right. The weights associated with the *Mobile*, *Manipulator Retract* and *Manipulator Target* behaviors can be found in Figure 6.

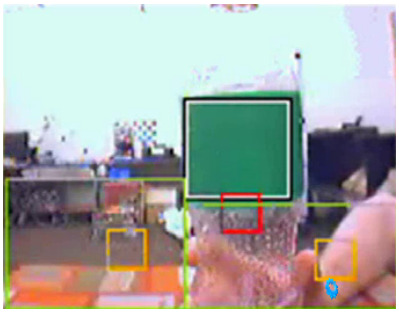


Fig. 5. Feature detection using Microsoft Robotics Studio's *SimpleVision* service. The black and white border marks the feature identified.

Initially both the *Manipulator Retract* and the *Mobile* behaviors are active causing the platform to move towards the target, while keeping the arm in its home configuration. After about 7 seconds the object gets within reach, thus the *Manipulator Retract* behavior deactivates and *Manipulator Acquisition* is activated. Shortly after the Segway behavior is also deactivated, to let the manipulator pick up the object without disturbances. However, the motion of the manipulator causes the Segway to drift, thus after a little while the mobile platform is reactivate until about time equals 20s, where it has again reached the desired position relative to the target. The visual servoing then aligns the gripper to the bottle as illustrated in Figure 4(b). After about 30 seconds the bottle should have been picked up by the gripper and a new target is given, causing the *Manipulator Retract* behavior to reactivate and *Manipulator Acquisition* to deactivate. At this point the mobile platform behavior is also activated, but is quickly suppressed while the arm is being retracted. Afterwards the control moves the platform to the desired location where the object is placed.

A video of this experiment can also be found associated with this paper.

VI. Conclusion

In this paper it has been presented how the dynamic systems approach can be applied to mobile manipulation. The contributions include a two level approach in which competitive dynamics are used both for the overall coordination of a mobile platform and a manipulator as well as the obstacle avoidance and target acquisition behaviors. The

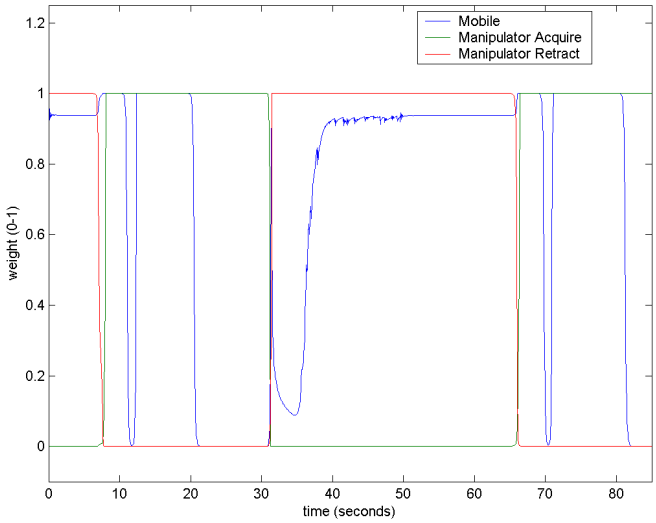


Fig. 6. Weights of the behaviors while operating

approach has been verified first in a simulated environment and secondly through real work experiments.

The empirical system was implemented using Microsoft Robotics Studio 1.5 (MSRS). The system was initially simulated and parameter tuning was performed using the simulator. The physics based simulator is ideal for parameter turning. Overall MSRS was a useful environment for implementation of the platform. Control was implemented at 20Hz, but due to the non real-time behavior of Windows XP there were occasional outliers.

REFERENCES

[1] H. Seraji, A Unified Approach to Motion Control of Mobile Manipulators, *The International Journal of Robotics Research*, Vol. 17, No. 2, 1998, pp. 107-118.

[2] E. Papadopoulos, J. Poulakakis, Planning and Model-Based Control for Mobile Manipulators, *Proceedings of the IROS'00* , 2000, pp. 1810-1815.

[3] Q. Huang, K. Tanie, S. Sugano, Coordinated Motion Planning for a Mobile Manipulator Considering Stability and Manipulation, *Thee International Journal of Robotics Research*, Vol. 19, No. 8, 2000, pp. 732-742.

[4] D.H. Shin, B.S. Hamenr, S. Singh, M. Hwangbo, Motion Planning for a Mobile Manipulator with Imprecise Locomotion, *Proceddings of the IROS'03*, 2003, 847-853.

[5] G. Schöner, M. Dose, A dynamical systems approach to task-level system integration used to plan and control autonomous vehicle motion, *Robotics and Autonomous Systems*, Vol. 10, 1992, pp. 253-267.

[6] G. Schöner, M. Dose, C. Engels, Dynamics of behavior: theory and applications for autonomous robot architecture. *Robotics and Autonomous Systems*, Vol. 16, 1995, pp. 213-245.

[7] O. Khatib, Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. *The International Journal of Robotics Research*, Vol. 5, No. 1, 1986, pp. 90-98.

[8] E. Bicho, G. Schöner, The dynamic approach to autonomous robotics demonstrated on a low-level vehicle platform. *Robotics and Autonomous Systems*, Vol. 21, 1997, pp. 23-35.

[9] E. Bicho, P. Mallet, G. Schöner, Using Attractor Dynamics to Control Autonomous Vehicle Motion. *Proceedings of the IECON'98*, Vol. 2, 1998, pp. 1176-1181.

[10] E. Bicho, P. Mallet, G. Schöner, Target Representation on an Autonomous Vehicle with Low-Level Sensors *The International Journal of Robotics Research*, Vol. 19, No. 5, 2000, pp. 424-447.

[11] H. Choset, K.M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L.E. Kavraki, S. Thrun, *Principles of Robot Motion* . The MIT Press, 2005.

[12] E.W. Large, H.I. Christensen, R. Bajcsy, Scaling the Dynamic Approach to Path Planning and Control: Competition among Behavioral Constraints. *The International Journal of Robotics Research*, Vol. 18, No. 1, pp. 37-58.

[13] P. Althaus, H.I. Christensen, F. Hoffmann, Using the Dynamical System Approach to Navigate in Realistic Real-World Environments. *Proceedings of IROS'01* , Vol. 2, 2001, pp. 1023-1029.

[14] P. Althaus, *Indoor Navigation for Mobile Robots: Control and Representations*, Ph.d. Dissertation, Royal Institute of Technology (KTH), Stockholm, Sweden, 2003.

[15] S. Goldenstein, E. Large, D. Metaxas, Non-linear dynamical system approach to behavior modeling, *The Visual Computer*, Vol. 15, 1999, pp. 349-364.

[16] I. Iossifidic, G. Schöner, Autonomous reaching and obstacle avoidance with the anthropomorphic arm of a robotics assistant using the attractor dynamics approach, *Proceedings of ICRA'04*, 2004, pp. 4295-4300.

[17] L.-P. Ellekilde, J.W. Perram, Tool Center Trajectory Planning for Industrial Robot Manipulators Using Dynamical Systems, *The International Journal of Robotics Research*, Vol. 24, No. 5, 2005, pp. 385-396.

[18] C. Santos, M. Ferreira, Ball Catching by a Puma Arm: a Nonlinear Dynamical Systems Approach, *Proceedings of IROS'06*, 2006, pp. 916-921

[19] I. Iossifidic, G. Schöner, Dynamical Systems Approach for the Autonomous Avoidance of Obstacles and Joint-limits for an Redundant Robot Arm. *Proceedings of the IROS'06*, 2006, pp. 580-585.

[20] P. Jensfelt, H.I. Christensen, Pose tracking using laser scanning and minimalistic environment models, *IEEE Transactions on Robotisc and Automation*, Vol. 17, No. 2, 2001, pp. 138-147.

[21] J. Forsberg, P. Åhman, . Wernersson, The Hough transform inside the feedback loop of a mobile robot, *Proceedings of ICRA*, Vol 1, 1993, pp. 791-798.

[22] K.O. Arras, R.Y. Siegwart, Feature Extraction and scene interpredation for map-based nagivation and map building, *Proceedings of SPIE, Mobile Robotics XII*, Vol. 3210, 1997, pp. 42-53.

[23] L.-P. Ellekilde, P. Favrholt, M. Paulin, H.G. Petersen, Robust control for high-speed visual servoing applications, *International Journal of Advanced Robotic Systems* , Vol. 4, No. 3, 2007, pp. 272-292.