Integration of a Redundant Mobile Manipulator System: A Drink Serving Task

Damir Omrčen, Bojan Nemec, Andrej Kos, Leon Lahajnar, Jan Babič, Andrej Gams, Aleš Ude, Leon Žlajpah

"Jozef Stefan" Institute, Jamova 39, Ljubljana, Slovenia
E-mail: damir.omrcen@ijs.si

Abstract: *The paper shows an integration of a robot manipulator and a mobile platform into a single robot system called mobile manipulator. The mobile manipulator is used as a service robot. There are numerous possible applications with such system. We will show in the paper the integration of both systems into one and its use. To demonstrate the capability of the system we have realized an application, where the robot serves a drink to a human. Furthermore, the mobile manipulator uses robot vision to recognise the bottles and glasses.*

Keywords: *mobile manipulator, service robot, redundant system, robot vision*

I INTRODUCTION

Service robotics is one of the major research fields in robotics today. One of the most important properties of service robots are large workspace and their ability to move autonomously in unknown environment based on sensors information. We are interested especially in service robots which consist of a manipulator arm being mounted on a mobile platform, i.e. mobile manipulators. Mobile manipulators are often used as service robots [1, 2, 3, 4]. Mobility added by the mobile platform increases the robot's workspace and adds additional degrees of freedom (DOF) to the system. The additional DOF can make the system kinematically redundant. The redundancy combined with an appropriate control makes the system more autonomous and more versatile, e.g. a redundant robot can avoid obstacles in the workspace while executing a given task, or it can minimize torques or velocities in robot's joints etc.

However, the integration of two different robotics subsystems (manipulator and platform) opens new and yet unresolved control problems. Both subsystems usually have different dynamic properties and the control strategy.

This paper emphasises on the basics of the integration of two subsystems. In the paper we describe model developing, the control, the use of the vision system, the integration of the subsystems on the hardware and software level, etc. The developed strategy was used to accomplish a simple and evident example – a drink serving task.

The task of the robot is to pick a bottle, pour a drink to a glass and serve a glass with a drink to a human.

II MODEL AND CONTROL

In our work we have used the mobile manipulator as a service robot. The mobile manipulator is composed of a robot manipulator arm Mitsubishi Pa-10, which is mounted on a holonomic mobile platform Nomad XR4000. The system is equipped with a vision system that is capable of tracking simple objects. The service robot is shown in Figure 1.

2.1 Model

The mobile platform is holonomic. That means that it can move in all three DOF in plane (two positions and one rotation) independently. As a result the mobile platform can be modelled as two prismatic links plus a rotational joint.

The robot arm mounted on the platform has 7 DOF. Therefore, the whole system can be modelled as a serial chain with 10 DOF. The task of the mobile manipulator is to track the position and the orientation of the robot's end-effector, which requires 6 DOF. Therefore, the system is kinematically redundant with 4 DOF of redundancy.

To control the mobile manipulator we have to develop a kinematic model of the system. We developed the forward kinematics and system Jacobian using analytical approach.

2.2 Control

To control a redundant system we have used a simple velocity feed-forward controller:

$$\dot{\boldsymbol{q}}_{\mathrm{c}} = \mathbf{J}^{\#}(\mathrm{K}_{\mathrm{P}}(\boldsymbol{x}-\boldsymbol{r}_{\mathrm{x}}) + \mathrm{K}_{\mathrm{ff}}\dot{\boldsymbol{r}}_{\mathrm{x}}) + \mathbf{N}\dot{\boldsymbol{q}}_{\mathrm{N}}, \tag{1}$$

where $\dot{\boldsymbol{q}}_{\mathrm{c}}$ is the control velocity, $\mathbf{J}^{\#}$ and $\mathbf{N}$ are the Jacobian generalised inverse and null space matrix, respectively. K_{P} and K_{ff} are the

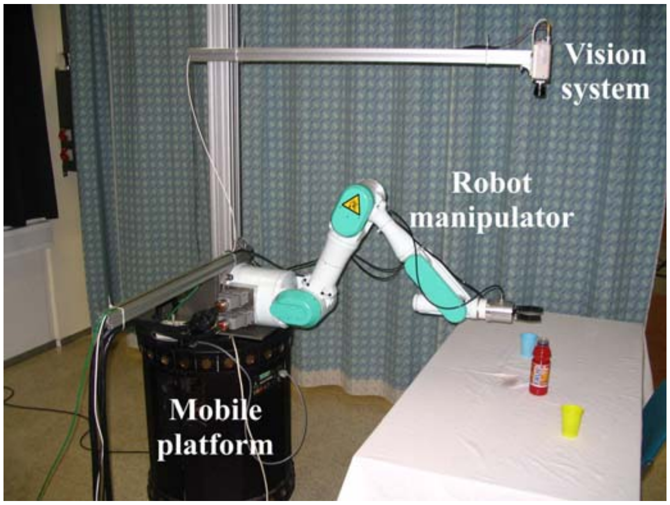

Figure 1
The service robot at work

controller gains. $\boldsymbol{x}$ and $\boldsymbol{r}_{\mathrm{x}}$ are the task space position/orientation and the reference in the task space. $\dot{\boldsymbol{q}}_{\mathrm{N}}$ is the desired null space velocity. The first part $\mathbf{J}^{\#}(\mathrm{K}_{\mathrm{P}}(\boldsymbol{x}-\boldsymbol{r}_{\mathrm{x}})+\mathrm{K}_{\mathrm{ff}}\dot{\boldsymbol{r}}_{\mathrm{x}})$ corresponds to task space controller and the second part $\mathbf{N}\dot{\boldsymbol{q}}_{\mathrm{N}}$ to the null space controller.

The controller is based on the assumption that both subsystems are velocity controlled, in case that one subsystem enables torque control we propose combined control [4] to achieve better results.

2.3 Vision System

For a vision system we used a single camera, which was mounted on the platform and was perpendicular to the desk plane, as seen in the Figure 1.

The first step is to detect the objects (bottle and glasses) in the captured images. Secondly, we had to convert these positions into the global coordinate system in order to command the robot. Therefore a system of equations was required to convert pixel coordinates (u, v) into the platform coordinates (x, y) and then into the global coordinates.

This task was simplified due to the fact that the height of all the objects was constant, as they all resided on a desk. For this purpose a perspective mapping between two planes was used, which converts the image plane into a platform coordinate system (x, y) with a constant height z. This mapping can be written as a 3 × 3 matrix H, which is defined up to a scale factor s and thus has 8 DOF. As this mapping is invertible, the information from a single camera is sufficient for determining object's position on the workspace. For details regarding the calibration system see [5].

The mapping matrix H was determined only at the beginning of our experiment. The procedure was automated, as the robotic hand moved the calibration object across the surface, while our calibration program measured the data at five positions. We used one of the glasses as the calibration object. In our case five pairs of measurements were obtained: (u, v), (x, y), which were used to build the H matrix.

To detect and track the objects in the captured images it is required that the vision system can:

- acquire images for processing
- detect objects in a frame
- track objects in a non-engineered environment
- transmit the object's position data to other software via UDP/IP connection

To speed up the development and implementation of the visual system we used the previously developed software DBvision [7], which proved to be suitable for this task. The main part of this software is a probabilistic tracker that uses colour and shape to find and track objects in the scene. Each object is represented by a colour blob and its centre is being tracked.

Each object was represented by a distinct colour. The main problem when working with colours is the fact that an object's colour can change depending on lighting conditions. We tried to avoid this to certain extent by fixing the environment by closing the curtains. As the robot moved around it further affected the lighting conditions. We alleviated this problem by using hue and saturation only for tracking.

The second problem we had to address was the parallax problem. The perspective transformation, which was used, only works for a single plain. The taller objects such as bottles were above the surface and their positions did not transpose well into global coordinate system and therefore contained a significant error. We addressed this issue by performing two calibrations, one for the shorter objects (glasses) and one for taller objects (bottles). This way we achieved suitable accuracy.

2.4 Hardware and Software Integration

To control the robot system we have used one computer that runs in the real time. This computer communicates with all devices used in the system (platform, manipulator, vision) (see Figure 2).

To achieve real time ability we have used UDP/IP connection, which is preferred over TCP/IP, since it is connection-less and does not take any time for reply.

All devices have the possibility to communicate with the control computer using UDP/IP. The vision system sends objects position data with a frequency of 30 Hz. The platform and the manipulator send joint position, velocities and sonar readings and receive velocity commands from the control computer at 60 Hz.

The sample frequency is 60 Hz which is enough for slower systems like the one we have used. In the case of more dynamic system we propose higher sample frequency and also higher communication rate and faster vision system. Moreover, XPc Target (Matlab) or other real time operation system is preferred for higher sample frequencies.

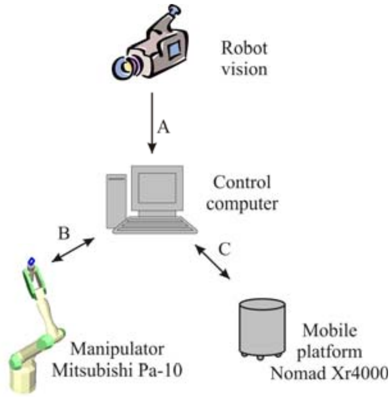

Figure 2
Hardware connection

III DRINK SERVING TASK

To demonstrate the successful integration of the mobile manipulator we have realized an application, where the robot serves a drink to a human.

This experiment integrates synchronised motion of both subsystems (platform and manipulator) and use of the vision system, force sensor, gripper etc.

Here, the robot should find a bottle with juice and glass using robot vision. Then it should pour the juice into the glass and offer the glass to a human.

To realize the drink serving task the robot should perform a series of basic actions. In this case the basic actions are:

- positioning of the end-effector
- griping the object
- releasing the object
- pouring
- force calibration
- force measurement
- finding an object
- ...

Some of the actions have to be done by the robot and some by the vision system, force sensor or gripper. In our case the robot does not "know" how to use the basic actions; therefore, the basic actions have to be programmed before the experiment into the following phases:

- force sensor calibration *(robot, force sensor)*
- find a bottle *(vision)*
- approach to the bottle *(robot)*
- grip the bottle *(gripper)*
- rise and weigh the bottle *(robot, force sensor)*
- find a glass *(vision)*
- approach to the glass *(robot)*
- pour juice *(robot, force sensor)*
- position the bottle to the deposit place *(robot)*
- release the bottle *(gripper)*
- approach to the glass *(robot)*
- grip the glass *(gripper)*
- offer juice to a human *(robot)*
- release the glass *(gripper)*

Some of the phases are composed of more than one basic action. For example, approach to the object is usually composed of a series of basic positioning actions, since the robot can not move directly to the gripping position because it could collide with the object.

In our case the pouring phase was the most demanding of all. There are various approaches to pour liquid from a bottle. We could pour the liquid by measure the time of pouring; another way is to use a mechanical device that stops the pouring when some specified amount of liquid is poured. The human uses visual information when pouring. In our case we have used bottle weight obtained from the wrist force sensor.

Before the pouring the weight of the bottle and the juice was measured. During the pouring the weight is constantly measured and when the difference between the initial and the measured mass achieves some specified value the pouring stops and the bottle returns into the vertical position. During pouring the flow of the liquid is changing due to the bottle shape. We have empirically defined the function of the pouring speed and the amount when the pouring stops.

During pouring the orientation of the force sensor is changing. As a result the weight of the gripper and the force sensor affect the measurement in various directions. It is hard to subtract this influence, since the weight and the measurement offsets of the force sensor are unknown. Therefore, we have to compensate the weight of the gripper and force sensor and force sensors offsets depending on the sensor orientation using pre-calibration (look first phase) [6].

3.1 Collision and Self Collision Avoidance

The robot that works in unstructured environments should avoid the obstacles in the environment.

The platform detects the obstacles near the platform using ultrasonic sensors mounted round the platform. These obstacles generate virtual repulsive velocity in the null space that moves the platform away from the obstacles without disturbing the task space. The manipulator does not detect the obstacles. We have only implemented the basic self-collision avoidance and avoiding collisions with the table.

The virtual velocities are applied in the null space in order that they do not disturb the task space (position of the end-effector). The measured and self-collision obstacles generate virtual null space velocity as:

$$\dot{\mathbf{q}}_{n_{xy}} = \begin{cases} -\sum_i w_i K_s \left(\frac{d_{soi}}{\|d_{0_i}\|} - 1\right) \frac{d_{0_i}}{\|d_{0_i}\|}, & \forall d_{0_i} < d_{soi}, \\ 0, & else \end{cases} \tag{2}$$

Here all obstacles *i* are taken into account, as long as they are inside the sphere of influence d_{soi}. The repulsive velocity is larger when an obstacle is closer to the robot.

Conclusion

We have shown an integration of different systems in a single working unit. The system can be used as a service robot.

The drink serving task is only one of the possible applications that was realised with the system. The main problem in all applications is that for every application the user must specify all the phases (the order of the basic actions) that will make the task successful. For such simple tasks as drink serving task there are more than 30 basic actions.

In the future work the robot will have to use cognition and reasoning and build the order of basic actions by itself.

Acknowledgement

This study was supported by the Slovenian Ministry of Higher Education, Science and Technology.

References

[1] O. Khatib: Mobile manipulation: The robotic assistant. Robotics and Autonomous Systems, 26:175-183, 1999

[2] L. Petersson, D. Austin, D. Kragic: High-level control of a mobile manipulator for door opening. In IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 3, 2333-2338, Takamatsu, Kagawa, Japan, oct 2000

[3] K. Lay et al.: MORPHA: Communication and Interaction with Intelligent, Anthropomorphic Robot Assistants. In Tagungsband Statustage Leitprojekte Mensch-Technik-Interaktion in der Wissensgesellschaft, Saarbrücken, Germany, October, 2001

[4] D. Omrčen, L. Žlajpah, B. Nemec: Autonomous motion of a mobile manipulator using a combined torque and velocity control. Robotica 22(6), 623-632, 2004

[5] D. C. Bentivegna, A. Ude, C. G. Atkeson, G. Cheng. Humanoid Robot Learning and Game Playing Using PC-Based Vision. Proc. IEEE/RSJ Intl. Conf. Inteligent Robots and Systems, pages 2449-2454, EPFL, Lausanne, Switzerland, October 2002

[6] OMRČEN, Damir, NEMEC, Bojan: Measuring knee movement using an industrial robot - gravity compensation for the automatic gripper. Stroniški vestnik 48(2), 87-97, 2002

[7] A. Ude, C. G. Atkeson: "Probabilistic Detection and Tracking at High Frame Rates Using Affine Warping", Proc. Int. Conf. Pattern Recognition, Quebec City, Canada, August 2002