Reactive Control for Mobile Manipulation

Jonathan M. Cameron Douglas C. MacKenzie* Keith R. Ward
Ronald C. Arkin* Wayne J. Book

School of Mechanical Engineering *College of Computing
Georgia Institute of Technology
Atlanta, Georgia 30332 U.S.A.

Abstract

Research for executing large-scale motions of mobile manipulators is described. Mobile manipulators are mobile bases with an attached arm which function in an integrated manner. Motivation is given for moving the arm while the base is moving. This work applies reactive control concepts to achieve this type of motion. Tools for modeling integrated arm-vehicle kinematics and dynamics are discussed. Simulation results are presented.

1 Introduction

Mobile manipulation, the *integrated* control of a mobile platform and robotic arm, offers major advantages to the materials handling community. By treating this robotic configuration as a fully integrated system, a means to achieve a wide range of new capabilities for these systems can be provided. This includes the ability to function in only a partially modeled environment, the ability to carry out a diverse set of operational tasks relevant to materials handling, and the ability to complete these tasks in real-time. A *mobile manipulator*, in this context, refers to a fully integrated arm/vehicle system from a control viewpoint, *not* simply a mobile robot with an arm attached.

This paper covers one aspect of a large NSF funded project in mobile manipulation. A more complete description of the entire project appears in [6]. This research in intelligent transport systems incorporates novel aspects of autonomous cooperative subsystems with new mechanisms for free-path movement using multi-sensor systems for the control of material handling tasks such as object acquisition, storage and retrieval.

This paper is divided into the following sections: Section 1 introduces this project and some of the underlying concepts. Section 2 covers background material. Section 3 describes tools to assist in modeling integrated mobile manipulators. Section 4 presents some details involved in computing motions to move the manipulator towards the goal and away from obstacles. Section 5 presents results of simulations which incorporate the concepts outlined in the rest of the paper. Section 6 gives some conclusions and outlines future work.

2 Background

2.1 Arm Preshaping

There exists a body of psychological and neurophysiological evidence attesting to a preshaping phase prior to the picking up of an object (e.g., [2]). In order to make most efficient use of the mobile manipulator, it is wise to preshape the arm during the *ballistic* motion as the vehicle nears the work area. Our previous research [8] studies this problem in the context of delivering a mobile platform to a docking site.

In order to make the most efficient utilization of the time required for assembly, locomotion, and preliminary arm configuration, these motions should proceed in parallel, as opposed to sequential *move-vehicle* then *move-arm* operations. By decoupling the manipulation problem into ballistic and guarded (micro-manipulation) components, advantage can be taken of the latent time available during vehicle travel by using the *a priori* expectations available from both models of the workplace and the parts to be manipulated.

2.2 Reactive Robotic Systems

Reactive robotic control systems are characterized by a tight coupling between sensing and action, typ-

1050-4729/93 $3.00 © 1993 IEEE

ically without the use of any intervening global representations. Behaviors generally are the means for expressing these actions, while specialized perceptual strategies (variously referred to as action-oriented perception, selective perception, and task-dependent perception) are used to provide the requisite sensory information that is necessary for the execution of the behavioral action [3]. A growing number of these reactive systems exist, notable among them are the subsumption-based architectures [9, 12], REX [18], RAPs [15], and schema-based systems [5] among many others (e.g., [22, 1, 17]).

2.3 Schema-based Reactive Control

Motor schemas are the basic units of motor behavior in our control system [5]. Each schema acts as a concurrent and independent agent, contributing to the global success of the robot's task. Many of these behaviors have been encoded for a mobile robot:

- **move-to-goal** - move to a perceptually discernible goal
- **move-ahead** - move in a general compass direction
- **avoid-static-obstacle** - avoid a non-threatening object
- **escape and dodge** - avoid an actively threatening object
- **noise** - move randomly
- **move-up, move-down, maintain-altitude** - used for outdoor navigation over rough terrain.
- **docking** - move in a particular pattern towards a goal
- **avoid-past** - avoid where the robot's been recently
- **probe** - seek out new areas

Many of these schemas are described in [5, 4, 8, 7]. Each schema produces a velocity vector in a manner analogous to the potential fields method [19, 20]. The sum of these vectors determines the robot's motion.

Several of these behaviors are being used for the mobile manipulator. In particular **move-to-goal** and **avoid-static-obstacle** have been developed for use in part acquisition tasks. The details of their implementation appear in Section 4.

3 Modeling mobile manipulators

Modeling the kinematics and dynamics of a mobile manipulator can be a tedious and error-prone process. If done by hand, there is little certainty that the results are correct without extensive testing and validation. To circumvent some of these problems, we have developed a package with which the user can symbolically derive the kinematics and dynamics of a mobile manipulator (which consists of a mobile platform with a complicated arm) in an integrated fashion. The name of our package is `motion.m`. The results can be used symbolically for analytical purposes or can be used to automatically generate program code for use in simulations or control systems. Packages to derive kinematics and dynamics of manipulators exist [24, 27, 21, 23, 10, 25], but this package is unique because it is useful for symbolically modeling both kinematics and dynamics while treating the mobile base simply as another joint. This makes it possible to derive the kinematics and dynamics of the mobile manipulator in an integrated way.

The package `motion.m` can handle a manipulator composed of any number of limbs connected by joints of various types in a serial chain. Using a relatively simple mechanism description, the package can generate kinematic relationships and dynamic equations as desired. In this project, a Denning MRV-2 mobile platform with a CRS A251, 5 degree-of-freedom arm plus a gripper can be specified using the following mechanism description:

```
robot =
{{{x,y},{v,Fv}, {q1,T1}, 0,P1,m1,I1,Unicycle},
 {     0, 0,     {-q1,T2},d2,P2,m2,I2,Revolute},
 {     0, 0,      {q3,T3},d3,P3,m3,I3,Revolute},
 {-Pi/2, 0,{q4-Pi/2,T4}, 0,P4,m4,I4,Revolute},
 {     0,L4,      {q5,T5}, 0,P5,m5,I5,Revolute},
 {     0,L5,{q6+Pi/2,T6}, 0,P6,m6,I6,Revolute},
 { Pi/2, 0,       {q7,T7}, 0,P7,m7,I7,Revolute},
 {     0, 0,             0,L6,P8,m8,I8,Unactuated}};
```

This brief statement (plus a few simple definitions for some of the values used in it) describes the robot adequately for the package `motion.m` to construct the robot kinematics and dynamics. For instance, the transform from the global coordinates to the end effector can be constructed easily using the following Mathematica statement:

```
T[0,8] = Simplify[Transform[robot, 0, 8]];
```

The user can now use this result symbolically for analysis or can construct C code by using the Mathematica "Splice" command and appropriate constructs such as:

```
ee_world->x = <* T[0,8][[1,4]] *>;
```

Upon processing, useful C code is produced:

```
ee_world->x = x + L4*Cos(q3)*Sin(q4) +
              L5*Cos(q3)*Sin(q4 + q5) +
              L6*Cos(q3)*Sin(q4 + q5 + q6);
```

The modeling package treats many different kinds of joints in a unform way. This includes typical joints such as revolute or prismatic joints. But it also includes "joints" suitable for modeling wheeled platforms such as a Denning MRV-2 or a car. This approach produces kinematic and dynamic models which are automatically integrated.

4 Pseudo-force approach to reactive motion planning

One of the basic ideas in the implementation is to guide the motion of the arm and base by responding to artificial forces generated by the goal and obstacles. Two of the motor schemas have been transformed for mobile manipulation thus far: **move-to-goal** and **avoid-static-obstacle**. In the **move-to-goal** schema, the vehicle is "pulled" towards the goal by a pseudo-force acting on the end effector (or gripper). For the **avoid-static-obstacle** schema, pseudo-forces "repel" the mobile manipulator from each obstacle. These schemas also construct pseudo-torques to help turn the base or limbs in desired directions.

The following steps outline the computations involved in creating these pseudo-forces and pseudo-torques and generating motions from them.

1. Compute the position of the end effector. If the goal has been reached, stop.

2. The **move-to-goal** schema computes the pseudo-force "pulling" the end-effector towards the goal and "turning" the wheels towards the goal. This is described in Section 4-4.1.

   At the same time, the **avoid-static-obstacle** schemas compute the pseudo-forces and pseudo-torques on the mobile base, each joint, and each limb "repelling" them from nearby obstacles. This is described in Section 4-4.2.

3. At this point all of the "joints" (including the base and end effector) have pseudo-forces acting on them. Jacobians are then used to compute the corresponding pseudo joint torques. This is described in Section 4-4.3.

4. Desired joint speeds are computed using an artificial damping (or resistance) model for each joint:

$$\dot{q}_i = \tau_i / c_i \tag{1}$$

   where τ_i is the pseudo-torque on joint i, $\dot{q}_i$ is the desired speed for the joint and c_i is the artificial damping for the joint. Damping c_i is a function of the environment as described in Section 4-4.4.

5. The desired joint speeds can then be commanded directly or used as inputs to a dynamic model to compute the actual joint torques required to produce the desired joint speeds. For instance, additional torques will be necessary to hold the arm up while resisting gravity. In our current work, the motion of the mobile manipulator has been simulated and the desired joint speeds have been used directly.

Figure 1 shows the schemas and the computation processes involved.

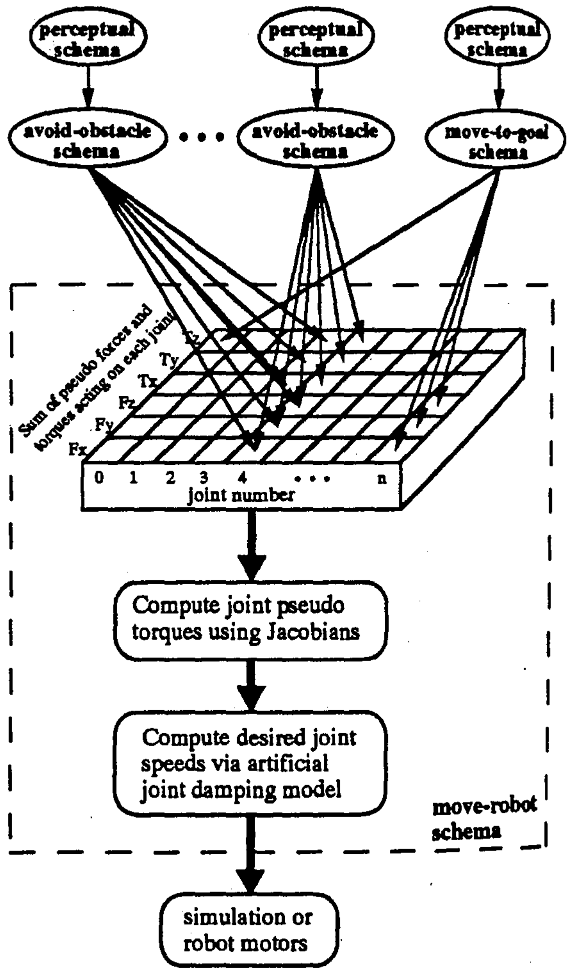

Figure 1: Reactive control schemas

4.1 Goal attraction pseudo-forces

The **move-to-goal** schema generates a pseudo-force "pulling" the end effector towards the goal.

$$f = f_c \hat{\mathbf{u}}_{g/ee} \tag{2}$$

where $\hat{\mathbf{u}}_{g/ee}$ is a unit vector pointing from the end effector towards the goal and f_c is a constant gain representing the magnitude of the pseudo-force pulling the end effector towards the goal.

An additional pseudo-torque is also applied to the wheels of the robot to steer them towards the goal. This torque is proportional to the angle between the wheels and a line towards the goal. Similar torques are applied to steer the wheels away from obstacles. These steering torques are all summed together before desired wheel steering speeds are computed.

4.2 Obstacle repulsion pseudo-forces

The **avoid-static-obstacle** schema generates "repelling" pseudo-forces and pseudo-torques which have two components: joint repulsion and limb repulsion.

The joint repulsion is a force repelling each joint directly away from each obstacle that is closer than a certain threshold distance. This can be imagined as a spherical repulsion field centered about each joint which pushes the joint away from obstacles.

Joint repulsion is not enough because it is possible for an obstacle to strike the middle of a limb (away from both adjoining joints) unless the threshold distance is made large. Therefore, limb repulsion is added to take care of this possibility. This is accomplished by introducing a pseudo-force and pseudo-torque on the corresponding joint to move and turn the limb away from an obstacle that is closer than some threshold. This can be imagined as a cylindrical repulsion field centered about the axis of the limb which pushes and twists the limb away from obstacles.

Currently the outputs of each motor schema are combined linearly to give total artificial forces and torques acting on each joint. Then the **move-robot** schema does a centralized computation to determine desired joint motions. In the future, when multiple parallel processors are available, some of the collected computations in **move-robot** will be distributed into the computations for each **move-to-goal** and **avoid-static-obstacle** schema and the linear combination of results will happen at a later stage.

After examining the literature in this field, it appears that this work is unique because it is apparently the first application of reactive control concepts to mobile manipulation for integrated arm-base motion. Although Connell does apply reactive control concepts to mobile manipulation using a subsumption architecture [11], he inhibits the simultaneous movement of the arm and the base—a standard subsumption architecture principle.

4.3 Jacobians

Transforming the pseudo-forces and pseudo-torques acting on each joint into corresponding joint torques is a tedious process if done by hand. The relationship between static forces and the corresponding joint torques is defined by the Jacobian matrix [13, p. 179].

$$\{\tau\} = [J]^T \{f\} \qquad (3)$$

where τ is the n-vector of joint torques and f is the 6-vector of forces and torques acting on the end effector (or joint). The pseudo-forces and torques for each joint can be converted to corresponding joint drive torques using appropriate Jacobians. In order to implement this, different Jacobians have to be constructed for each joint. We use the modeling package `motion.m` to simplify this process.

The effect of this approach is to prescribe more motion to the degrees of freedom which have more effect on reaching the goal. Note that scaling of joint variables is an important step and that mixed units (angles and distances) mean that a change of units will affect the results.

Jacobians have been used extensively in robot control [13]. Hootsmans and Dubowsky use Jacobians for controlling a mobile manipulator based on a desired trajectory [14]. In the approach described in our paper there is no predefined trajectory—the motion is constructed using pseudo-forces and torques generated by attraction towards the goal and repulsion away from obstacles. This work is the first use of Jacobians to convert pseudo-forces and pseudo-torques (from schemas or potential fields) into corresponding joint pseudo-torques for use in mobile manipulation.

4.4 Damping functions

The artificial joint damping (or resistance) model relates the joint pseudo-torques to the desired joint speeds. Since damping is the natural physical relationship between forces and speeds (when no springs are involved), damping is an appropriate analogy. Constant and equal damping in each joint gives the mobile manipulator an overall behavior pattern. This pattern is marked by equal use of all the joints so that the robot translates toward the goal and reaches out as it moves. If the goal is not reached before the arm is fully extended, the robot continues to translate towards the goal with the arm extended. This "Frankenstein walk" needs to be avoided in cluttered areas or when lifting a heavy object. The most natural movements are generated when the arm only begins moving as the vehicle nears the goal, and finishes preshaping just as the goal is reached and the base stops. Our approach is to vary the damping coefficient of each joint separately as a function of the task and environment.

In some of our simulations, the artificial joint dampings are calculated as a function of the distance from the end effector to the goal. In general, larger damping values reduce the motion of a joint while small damping values increase its motion. In these simulations, the damping of steering and forward drive of the base increase as the robot gets close to the goal, while the remaining joint dampings decrease as the robot approaches to the goal. When the mobile manipulator is far away from the goal, the steering and forward drive dominate the motion to move the robot towards the goal. As the goal is approached, the damping of the base steering and drive increases to slow the robot down while the damping on the arm joints decreases causing the arm to preshape without becoming fully extended. A plot of example damping functions of this type is shown in Figure 2. The terms in the plot

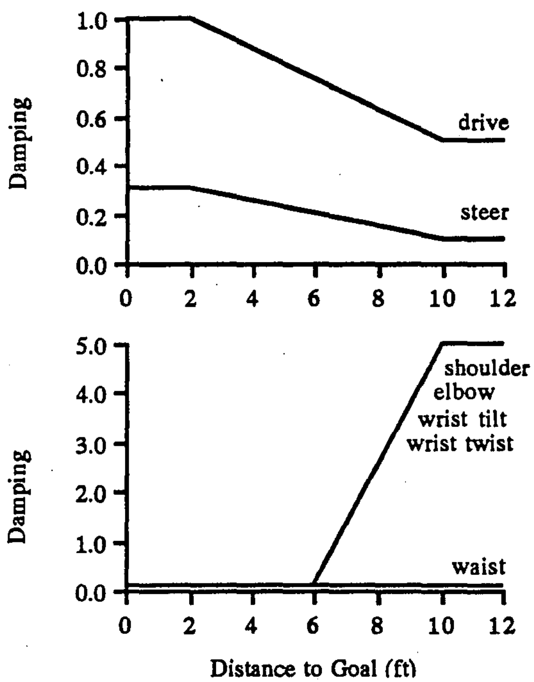

Figure 2: Typical artificial damping functions

have the following meaning: *steer* is the direction the wheels are pointing, *drive* corresponds to spinning the wheels to move the base, *waist* is the angle of the base of the arm with respect to the top of the mobile base, *shoulder* means the tilting of the upper arm with respect to the base of the arm, *elbow* is the bending of between the upper arm and the lower arm, *wrist tilt* is the bending of the gripper with respect to the lower arm, and *wrist twist* is the twisting of the gripper with respect to the lower arm.

Constructing artificial joint damping functions is an area of continuing research. More information about the environment and goal location will be incorporated into the damping functions to generate more natural movements. The basis for the selection of damping functions depends on the criteria the system is to be optimized under (e.g., speed, stability, load, etc.). Primate performance studies in reaching can also provide the basis for generating suitable damping functions that emulate natural motion.

5 Simulation Results

To test and validate the ideas presented earlier, a simulation system has been developed as a first step to implementing these ideas with real hardware. A more detailed description of the simulation system is given in [6].

5.1 Hardware being simulated

This simulation is based on hardware being assembled in our laboratory. The mobile manipulator is being constructed from a Denning MRV-2 mobile robot and a CRS A251 industrial robot arm. Figure 3 shows a photo of the arm installed on the base. The MRV-2 is two feet tall with a diameter of 27 inches. It uses a three wheel synchronized steering system to allow motion without turning the vehicle body. The maximum steer velocity is 75 degrees per second. The maximum movement velocity is 4 feet per second. The A251 arm weighs 37 pounds without the controller and provides a 2.2 pound nominal payload capacity. It has a 26 inch reach while moving at 18 inches per second in linear motion. The arm pick and place cycle is 2 seconds.

A Sun workstation functions as the mobile manipulator controller, providing an integrated view of the arm and base as a mobile manipulator. Treating the two as an integrated mechanism is central to this research.

The sensor platform that the vehicle will carry is still evolving. Proprioceptive sensors include position encoders on the wheels and joints, and a force/torque sensor on the wrist. A ring of 24 ultrasonic sensors around the base provides obstacle location information. A camera will be mounted on the wrist to accommodate active visual processes which will be used for obstacle detection and localization (see [6] for more

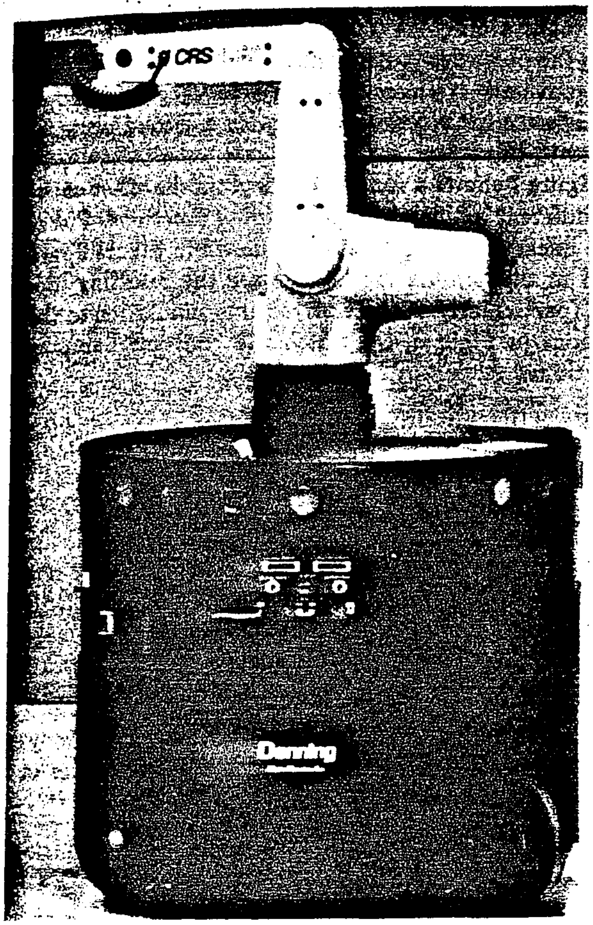

Figure 3: Integrated mobile manipulator

details). The use of infrared (IR) sensors around the arm for obstacle detection is being evaluated.

5.2 Simulations

Simulation results have been generated to evaluate the **move-to-goal** and **avoid-static-obstacle** behaviors. Three of these simulation runs are shown in Figures 4-6. In each case, the vehicle starts in the upper left corner with the arm raised and moves towards the goal in the lower right. The robot is shown in several poses it moves through as it traverses to the goal. Notice that there is a small cube on the top edge of base of the robot which indicates the direction the wheels are pointing.

Figure 4 shows the manipulator moving towards a goal. The top view shows the path the vehicle traversed while the lower view highlights the preshaping of the arm. Figure 5 shows the same run, but with an obstacle added. The base is repelled by the obstacle

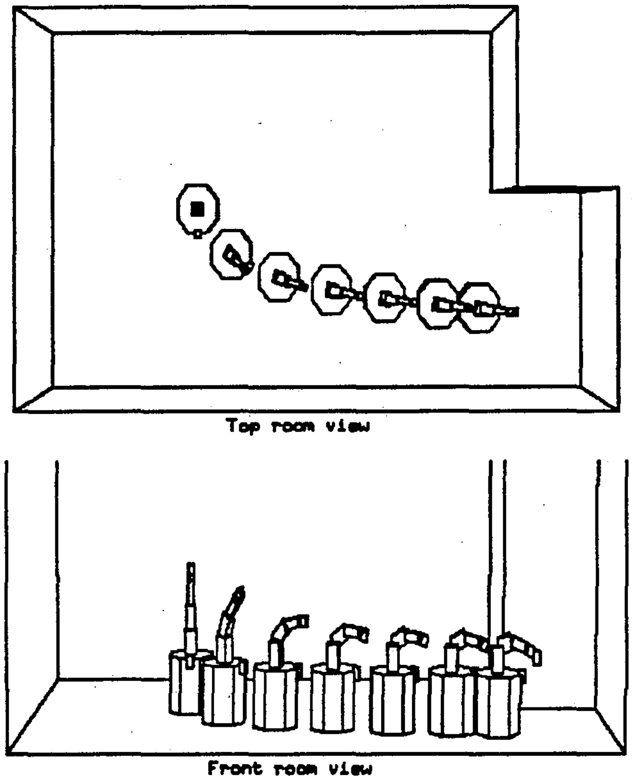

Figure 4: Simulation 1: Movement with no obstacles

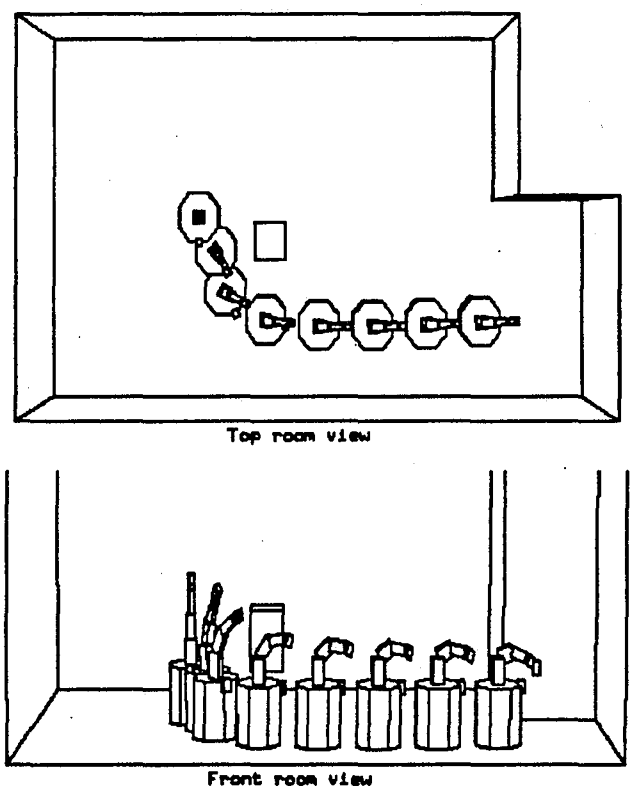

Figure 5: Simulation 2: Movement with one obstacle

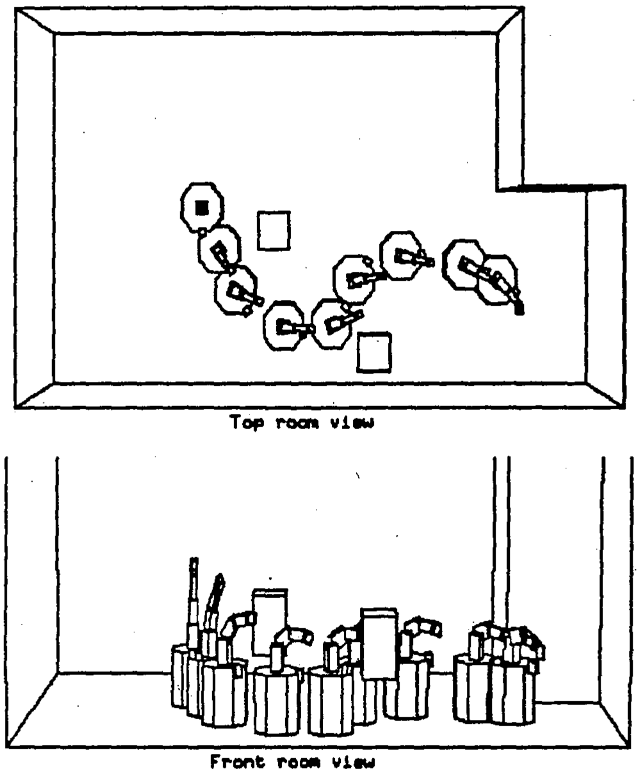

Figure 6: Simulation 3: Movement with two Obstacles

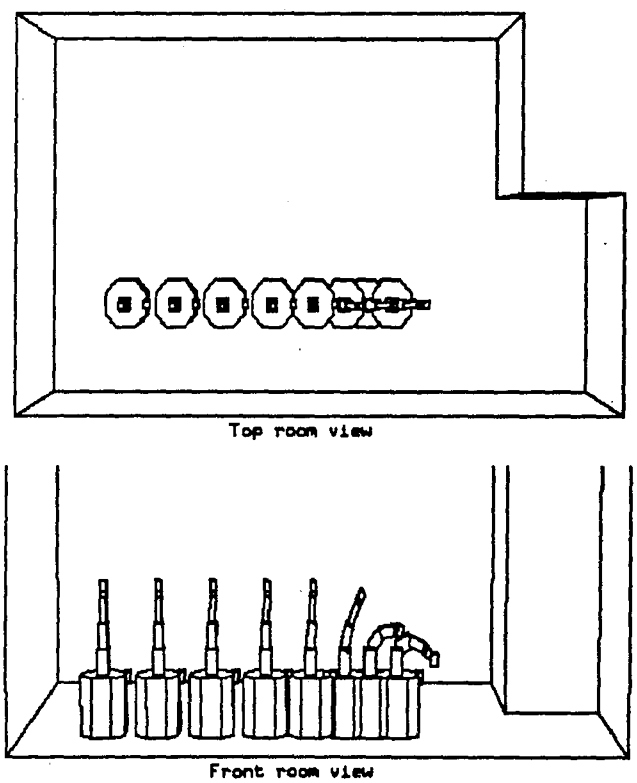

Figure 7: Simulation 4: Movement with distance-dependent artificial damping functions

and takes a different path. The arm also is repelled from the obstacle. Notice that it doesn't turn towards the goal until it clears the obstacle. Figure 6 shows this effect a little more clearly with a second obstacle added to the scene.

Figure 7 shows a run which displays the effect of using different artificial joint damping functions. Piecewise linear damping functions (shown in Figure 2) are used in this simulation. Compare the resulting motion (particularly of the arm) in this simulation to the motions in Figures 4-6 which use constant damping functions.

6 Conclusions and future work

The integrated combination of mobility and manipulation in dynamic environments is a promising area for robotics research. The possibility of both a large workspace and precise, high bandwidth motion make the combination highly attractive. This work makes the following contributions:

- Illustrates techniques that can function in dynamic environments, deal with uncertainties in sensor input, and utilize existing knowledge.
- Describes tools for modeling integrated vehicle-arm kinematics and dynamics.
- Combines kinematics and dynamics based concepts (such as Jacobians) with reactive control approaches in order to move toward a practical level of complexity and performance.
- Shows initial simulation results.

Future research efforts will add planning capabilities to the system. Given a task to be achieved, the planner will select the type of damping functions to be used as well as their parameterization. This planner will function as a pilot, configuring the underlying reactive execution layer and then letting it run. The planner will then monitor the progress of the vehicle, reconfigure parameters or schemas as necessary, and terminate execution in case of failure.

Acknowledgements

This research is supported by the National Science Foundation under grants IRI-9100149 and IRI-9113747, and the CIMS program and the Material Handling Research Center of the Georgia Institute of Technology.

References

[1] T. Anderson and M. Donath, "Animal Behavior as a paradigm for developing robot autonomy", in *Designing Autonomous Agents*, (P. Maes, ed.), MIT Press, Cambridge, MA, pp. 145, 1990.

[2] M. Arbib, T. Iberall, and D. Lyons, "Coordinated Control Programs for Movements of the Hand", *Exp. Brain Res. Suppl.*, 10, pp. 111-129, 1985.

[3] R.C. Arkin, "The Impact of Cybernetics on the Design of a Mobile Robot System: A Case Study", *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 20, No. 6, Nov/Dec 1990, pp. 1245-1257.

[4] R.C. Arkin, "Behavior-based Robot Navigation in Extended Domains", to appear in *J. of Adaptive Behavior*, Vol. 1, No. 2., 1992.

[5] R.C. Arkin, "Motor Schema Based Mobile Robot Navigation", *International J. of Robotics Research*, 8, 4, pp. 99, (1989a).

[6] R.C. Arkin, editor, "Integrated Control for Mobile Manipulation for Intelligent Materials Handling", *Intelligent Material Handling Using Mobile Robots—Collected Papers from the Mobile Robot Laboratory*, Georgia Tech Material Handling Research Center Report OP-92-19, 1992.

[7] R.C. Arkin and W.C. Carter, "Active Avoidance: Escape and Dodging Behaviors for Reactive Control", *Proc. Applications of Artificial Intelligence X*, Orlando, FL, pp. 676-689, 1992.

[8] R.C. Arkin and R.R. Murphy, "Autonomous Navigation in a Manufacturing Environment", *IEEE Trans. on Robotics and Auto.*, 6, 4, pp. 445, 1990.

[9] R. Brooks, "A Robust Layered Control System for a Mobile Robot", *IEEE Jour. of Robotics and Auto.*, RA-2, 1, pp. 14, 1986.

[10] G. Cesareo, F. Nicolò, "DYMIR: A Code for Generating Dynamic Model of Robots," *Proceedings of the 1st Intl. IEEE Conf. Robotics*, R. Paul, Ed., Atlanta, GA, pp. 115-120, March 1984.

[11] J.H. Connell, "A Behavior-Based Arm Controller," *IEEE Trans. on Robotics and Auto.*, Vol. 5, Num. 6, Dec. 1989, pp. 784-791.

[12] J.H. Connell, *Minimalist Mobile Robotics: A Colony-style Architecture for an Artificial Creature*, Academic Press, 1990.

[13] J.J. Craig, *Introduction to Robotics: Mechanics and Control*, 2nd Ed., Addison-Wesley Pub. Co., Reading, Mass., 1989.

[14] N.A.M. Hootsmans, S. Dubowsky, "Large motion control of mobile manipulators including vehicle suspension characteristics," *IEEE Conf. on Robotics and Auto.*, pp. 2336-2341, 1991.

[15] R.J. Firby, "Adaptive Execution in Complex Dynamic Worlds", Ph.D. Dissertation, Yale University, YALEU/CSD/RR #672, January, 1989.

[16] J.D. Foley, A. van Dam, S.K. Feiner, J.F. Hughes, *Computer Graphics: Principles and Practice*, Second Edition, Addison-Wesley Publishing Company, Reading, Mass., 1990.

[17] E. Gat, "Robust, Task-Directed, Reactive Control of Autonomous Mobile Robots", *Ph.D. Dissertation*, Virginia Polytechnic Inst. and State Univ., 1991.

[18] L. Kaelbling and S. Rosenschein, "Action and planning in embedded agents", in *Designing Autonomous Agents*, pp. 35, (P. Maes, ed.), MIT Press, Cambridge, MA, 1990.

[19] O. Khatib, "Real-time Obstacle Avoidance for Manipulators and Mobile Robots", *Proc. IEEE Int. Conf. Robotics and Auto.*, pp. 500, 1985.

[20] B. Krogh, "A Generalized Potential Field Approach to Obstacle Avoidance Control", *SME - RI Technical Paper* MS84-484, 1984.

[21] J.J. Murray, C.P. Neuman, "ARM: An Algebraic Robot Dynamic Modeling Program," *IEEE Int. Conf. on Robotics*, Atlanta, GA, pp. 103-114, 1984.

[22] D. Payton, "An Architecture for Reflexive Autonomous Vehicle Control", *IEEE Conf. on Robotics and Auto.*, pp. 1838, 1986.

[23] R.W. Toogood, "Efficient Robot Inverse and Direct Dynamics Algorithms Using Micro-computer Based Symbolic Generation," *IEEE Conf. on Robotics and Auto.*, pp. 1827-1832, 1989.

[24] E. Tunstel, N.Vira, "Mechanization of Manipulator Kinematic Equations via Macsyma," *Proceedings, Int. Computers in Engr. Conf. and Exhib. (ASME)*, pp. 649-655, 1989.

[25] J. Wittenburg, U. Wolz, "MESA VERDE: A Symbolic Program for Nonlinear Articulated-Rigid-Body Dynamics," Presented at the ASME Design Engr. Div. Conf. and Exhib. on Mech. Vib. and Noise, 1985, *ASME 85-DET-151*.

[26] Wolfram Research, Inc., *Mathematica*, Version 2.0, Wolfram Research, Inc., Champaign, Illinois, 1991.

[27] S. Yin, J. Yuh, "A User-Friendly PC Program for Symbolic Robot Dynamic Equations: ARDEG," *Proceedings, Int. Computers in Engr. Conf. and Exhib. (ASME)*, pp. 643-648, 1989.