LARGE MOTION CONTROL OF MOBILE MANIPULATORS INCLUDING VEHICLE SUSPENSION CHARACTERISTICS

Norbert A. M. Hootsmans
Steven Dubowsky

Department of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, MA 02139

ABSTRACT

Conventional fixed-base controllers are shown not to perform well on mobile manipulators due to the dynamic interactions between a manipulator and its vehicle. An extended jacobian transpose control algorithm is developed to improve the performance of such manipulator systems. It is shown to perform well in the presence of modelling errors and the practical limitations imposed by the sensory information available for control in highly unstructured field environments.

I. INTRODUCTION

Robotic systems are being considered for a wide variety of applications outside their traditional factory uses, in such diverse tasks as remote maintenance, toxic waste cleanup, and fire-fighting [1-3]. These tasks require *mobile* robotic manipulators, i.e. manipulators carried by vehicles, see figure 1.

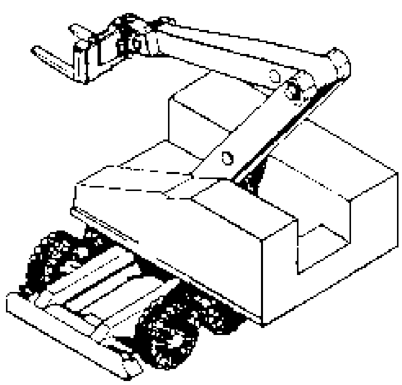

Fig. 1. A Mobile Manipulator System.

Unlike conventional industrial manipulators mounted on stationary bases, a mobile manipulator's motions interact dynamically with its vehicle, including its suspension, which degrades system performance, resulting in performance problems like excessive end-effector errors and poor system stability. Furthermore, in practice these systems will operate in highly unstructured environments, thus limiting the various sensing techniques available for their control, such as large motion endpoint sensing.

The above characteristics of mobile manipulators present challenging control problems; to date there has been relatively little research done to solve them. Significant work has been done in space robotics which considers the important dynamic interactions between a manipulator and its spacecraft [4,5]. However, this work does not consider the important effects of gravity and vehicle suspension/tire dynamic characteristics found in terrestrial applications. Most studies of *terrestrial* mobile robots treat them as simple vehicles without dynamics and without any manipulators and focus on map building of unknown environments and on motion planning algorithms [6]. A few studies include simple kinematic and dynamic models of a vehicle without manipulators, or consider only the kinematics of the vehicle and manipulator [7].

In the limited studies considering the dynamic interaction of a manipulator and its vehicle, the interactions have been simplified by assuming either a very massive vehicle or by using outriggers [2,8]. A study which did consider the control of manipulators on vehicles with suspension and tire compliance, using endpoint control, was limited to the acquisition phase of manipulator tasks and did not consider the control of large motions of the manipulator [9]. None of these studies considered the practical limitation on sensory information available for control, nor the effects of large changes in system parameters.

This study develops a control algorithm for manipulators mounted on a vehicle with significant compliance due to its suspension and tire/ground interaction, and applies to large manipulator motions when system characteristics are highly variable and when the accuracy and the availability of sensory information is limited. Several alternative control algorithms for mobile manipulators were developed and evaluated in this study, and an algorithm based on the jacobian transpose approach proved to be most promising. This algorithm is shown to have the potential to provide good large motion control. Furthermore, it relies only on practically available vehicle sensory information, rather than on difficult-to-obtain large motion endpoint sensing.

II. THE PROTOTYPE SYSTEM

A prototype system was designed in some detail as a base-line for the control algorithm development. Its properties were selected to be representative of a system that might be designed to perform human scale material handling tasks in a field environment [10]. The system consists of a three link 125 kg hydraulic manipulator with a 25 kg payload. It has a total of nine degrees of freedom (DOF), including the position and orientation of its 500 kg, four-wheel vehicle. The manipulator control bandwidth was chosen to be 4-10 Hz, depending on its configuration and payload. These bandwidths are well below the predicted first structural resonance of the manipulator, and the manipulator can thus be modeled as a rigid link system.

The vehicle's suspension was modeled as four identical passive quarter car suspensions resulting in combined vehicle/manipulator natural frequencies of approximately 1 Hz in the vertical direction and 2.5 and 10 Hz in the forward and sideward horizontal directions. The suspension was designed to give damping ratios in the 0.5 - 0.6 range. Figure 2 shows a schematic diagram of the manipulator and its vehicle, showing its kinematic variables.

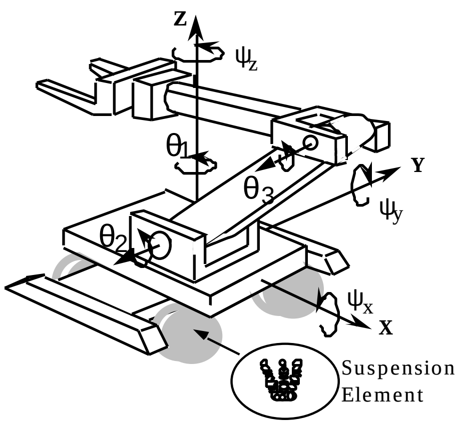

Fig. 2. System Model With Coordinates.

It was assumed that the system performance requirement would be to move a payload from rest to its final stationary position in 2 seconds. The maximum end-effector tracking error should be less than 20 mm, the steady state endpoint error less than 10 mm, and the settling time at the end of the motion less than 0.4 seconds. The settling time is defined to be the time after the end of the commanded motion required for the end-effector to stay within 0.5 mm of its steady state position. The performance of this system, using even a well designed conventional joint control algorithm of the type found in most industrial systems, proved to be quite unsatisfactory compared to the above specifications.

Consider the simple planar straight line trajectory, shown in figure 3. Link 1 is stationary for this straight line trajectory. The velocity profile is a simple sinusoid. The vehicle is assumed to be initially at rest with the manipulator in its initial position.A conventional proportional-derivative controller is applied to the manipulator. This yields a system with a 4-10 Hz bandwidth, depending on the system configuration, and a 1.0 to 2.0 damping ratio. When this controller was applied to the manipulator with its vehicle held stationary the maximum dynamic endpoint error was 8.5 mm, and the steady state endpoint error was 5.5 mm. However, when the vehicle is permitted to move, the manipulator takes more than a second to settle, and the errors are larger by an order of magnitude, as shown in figure 4.

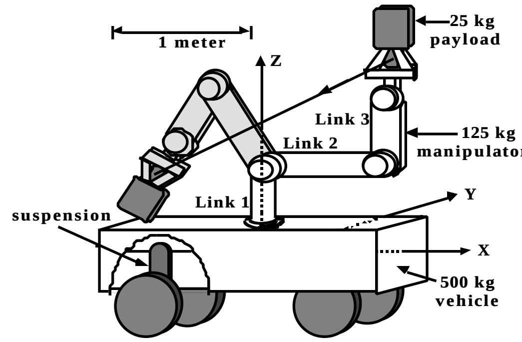

Fig. 3. Initial & Final Manipulator Positions

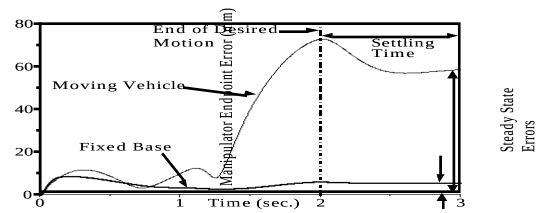

Fig. 4. Endpoint Error - Simple Joint Control.

The source of this error is the dynamic interaction of the manipulator and its vehicle, caused by both gravity and suspension effects. Figure 5 shows the vehicle pitch motion during the manipulator move. Clearly, the interaction of the manipulator and its vehicle significantly degrades system performance. A better control approach is required to achieve satisfactory large motion trajectory following.

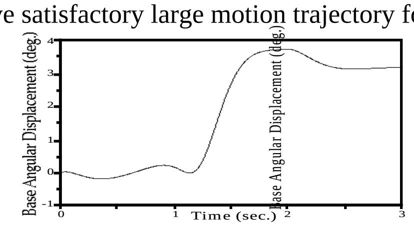

Fig. 5. Angular Vehicle Motion during Move.

III. CONTROL ALGORITHMS

A number of control approaches which have been applied to fixed-base manipulators were considered in this study for application to mobile manipulators, including the use of endpoint control feedback model-based methods, whether off-line or adaptive, in combination with a feedforward computed torque approach. Such approaches have been implemented in various forms including those using resolved rate control, impedance control, resolved acceleration control, and operational space control [11-14]. Some of these algorithms require a dynamic model of the system; all of them rely on an adequate formulation of manipulator kinematics in the form of a jacobian.

In this study a jacobian was formulated for mobile manipulators, and three control algorithms were developed and evaluated under trajectory control of such mobile manipulator systems. They are a jacobian transpose, a resolved rate, and a resolved acceleration algorithm for mobile manipulators.

1. The Mobile Manipulator Jacobian

A mobile manipulator on a suspended vehicle has a non-square jacobian, similar to those found for redundant manipulators. However, the extra degrees of freedom due to the mobile manipulator vehicle do not have control actuators, and hence one cannot simply resolve the redundancy. Instead, as discussed below, the Mobile Manipulator Jacobian (MMJ) can be partitioned into two sub-jacobians, $\mathbf{J}_v$ and $\mathbf{J}_m$, where $\mathbf{J}_v$ multiplies the vehicle velocities corresponding to the uncontrolled degrees of freedom, and $\mathbf{J}_m$ multiplies the manipulator joint velocities corresponding to the controlled degrees of freedom.

For an n DOF mobile manipulator on a 6 DOF vehicle, an equation is desired of the form:

$$\dot{\mathbf{x}}_e = \begin{Bmatrix} \dot{\mathbf{r}}_e \\ \omega_e \end{Bmatrix} = \mathbf{J}\,\dot{\mathbf{q}} = [\,\mathbf{J}_v \mid \mathbf{J}_m\,] \begin{Bmatrix} \dot{\mathbf{x}}_v \\ \dot{\mathbf{q}}_m \end{Bmatrix} \qquad (1)$$

where $\dot{\mathbf{x}}_e$ is the 6 x 1 dimensional column vector of end-effector linear and angular velocities, and where $\dot{\mathbf{r}}_e$ is the linear end-effector velocity and ω_e is the angular. The vector $\mathbf{q} = [\mathbf{r}_v^T, \theta_v^T, \mathbf{q}_m^T]^T = [x, y, z, \psi_x, \psi_y, \psi_z, q_1, \ldots, q_n]^T$ is the 6+n by 1 position vector, which includes the vehicle position and orientation, $\mathbf{r}_v$ and θ_v, as well as the manipulator joint displacements $\mathbf{q}_m$, see figure 2. $\dot{\mathbf{q}}_m$ is the vector of n manipulator joint velocities, and $\dot{\mathbf{x}}_v = [\dot{\mathbf{r}}_v^T, \omega_v^T]^T$, where $\dot{\mathbf{r}}_v$ is the vehicle linear velocity vector and ω_v is the angular. The MMJ sub-matrices $\mathbf{J}_v$ and $\mathbf{J}_m$ were derived as [10]:

$$\mathbf{J}_v(\theta_v, \mathbf{q}_m) = \begin{bmatrix} \mathbf{I} & -\mathbf{R}_v(\theta_v)\,[{}^V\mathbf{r}_e^V(\mathbf{q}_m)]^\times\,\mathbf{R}_v^T(\theta_v) \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \qquad (2)$$

$$\mathbf{J}_m(\theta_v, \mathbf{q}_m) = \begin{bmatrix} \mathbf{R}_v(\theta_v) & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_v(\theta_v) \end{bmatrix} \mathbf{J}_{fb}(\mathbf{q}_m) \qquad (3)$$

The matrix $\mathbf{J}_{fb}(\mathbf{q}_m)$ is the conventional 6 by n jacobian of an n DOF fixed-base manipulator. $\mathbf{I}$ is the 3 by 3 identity matrix and $\mathbf{R}_v(\theta_v)$ is the 3 by 3 vehicle rotation matrix, dependent only on the vehicle orientation, which relates the vehicle frame to the inertial frame. The term ${}^V\mathbf{r}_e^V(\mathbf{q}_m)$ is the 3 by 1 position vector of the manipulator end-effector with respect to the vehicle frame expressed in vehicle frame coordinates. $[\mathbf{a}]^\times$ is the cross product matrix for the vector $\mathbf{a}$, defined as:

$$[\mathbf{a}]^\times = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \qquad (4)$$

Note from equations (2) and (3) that the MMJ, $\mathbf{J}(\theta_v, \mathbf{q}_m) = [\mathbf{J}_v(\theta_v, \mathbf{q}_m) \mid \mathbf{J}_m(\theta_v, \mathbf{q}_m)]$, is a 6 by 6+n matrix dependent only on the manipulator joint displacements and the vehicle orientation.

2. Mobile Manipulator Dynamics

The equations of motion for the n DOF mobile manipulator can be written as [10]:

$$\mathbf{H}(\theta_v, \mathbf{q}_m)\,\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \begin{Bmatrix} \mathbf{0} \\ \mathbf{V}\,\dot{\mathbf{q}}_m \end{Bmatrix} + \mathbf{G}(\theta_v, \mathbf{q}_m) = \tau \qquad (5)$$

where the 6+n by 1 vector $\tau = [\mathbf{F}_{vs}^T(\mathbf{x}_v, \dot{\mathbf{x}}_v),\ \tau_m^T]^T$ consists of the forces and torques exerted on the vehicle by its suspension, $\mathbf{F}_{vs}$, and of the manipulator joint torques, τ_m. $\mathbf{H}$ is the 6+n by 6+n inertia matrix and is a function of the vehicle orientation θ_v; $\mathbf{C}$ is the 6+n by 1 nonlinear torque vector due to coriolis and centrifugal terms; $\mathbf{V}$ is the n by n manipulator viscous joint friction matrix, and $\mathbf{G} = [\mathbf{G}_v^T, \mathbf{G}_m^T]^T$ is the 6+n by 1 gravitational torque vector, consisting of the gravity forces and torques acting on the vehicle, $\mathbf{G}_v$, and the gravity torques acting on the n manipulator joints, $\mathbf{G}_m$.

3. The Jacobian Transpose Algorithm

The jacobian transpose control methods, such as operational space control [14] use the transpose jacobian to compute τ_m. For an n DOF manipulator on a 6 DOF vehicle this approach yields:

$$\tau = \begin{Bmatrix} \mathbf{F}_v \\ \tau_m \end{Bmatrix} = \mathbf{J}^T(\theta_v, \mathbf{q}_m)\,\mathbf{F}_e = \begin{bmatrix} \mathbf{J}_v^T(\theta_v, \mathbf{q}_m) \\ \mathbf{J}_m^T(\theta_v, \mathbf{q}_m) \end{bmatrix} \mathbf{F}_e \qquad (6)$$

where $\mathbf{F}_e$ is the vector of forces and torques reflected to the end-effector. By adding a gravity compensation torque, $\mathbf{G}_m$, to the term for τ_m in equation (6), in order to eliminate steady state error, and by applying equation (3), an extended jacobian transpose control algorithm, see figure 6, can be defined as:

$$\tau_m = \mathbf{J}_m^T(\theta_v, \mathbf{q}_m)\,\mathbf{F}_{edes} + \mathbf{G}_m(\theta_v, \mathbf{q}_m)$$

$$= \mathbf{J}_{fb}^T(\mathbf{q}_m) \begin{bmatrix} \mathbf{R}_v^T(\theta_v) & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_v^T(\theta_v) \end{bmatrix} \mathbf{F}_{edes} + \mathbf{G}(\theta_v, \mathbf{q}_m) \qquad (7)$$

where $\mathbf{F}_{edes}$ provides endpoint position, $\mathbf{x}_e$, and velocity, $\dot{\mathbf{x}}_e$, feedback by setting it as:

$$\mathbf{F}_{edes} = [\mathbf{K}_p]\,\{\mathbf{x}_{edes} - \mathbf{x}_e\} + [\mathbf{K}_d]\,\{\dot{\mathbf{x}}_{edes} - \dot{\mathbf{x}}_e\} \qquad (8)$$

Note that in general the 6 by 1 vector of vehicles forces and torques $\mathbf{F}_v$ in equation (6) consists of a combination of active control and passive suspension forces and torques acting on the vehicle. In this study it is assumed that the vehicle has no suspension actuators and hence $\mathbf{F}_v$ consists solely of the passive terms, $\mathbf{F}_{vs}$ in equation (5). The difference between the desired $\mathbf{F}_v$ and its actual value, $\mathbf{F}_{vs}$, is a disturbance to the system.

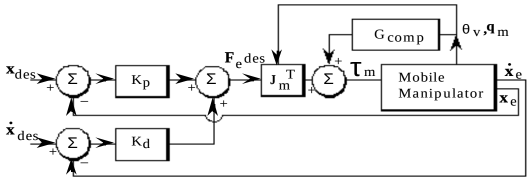

Fig. 6. Jacobian Transpose Control Diagram.

As long as the fixed-base jacobian matrix is of full rank, the controller can compensate for vehicle disturbance forces and torques, and the manipulator endpoint will track the desired trajectory. Tracking errors will result if the jacobian matrix is singular, although the algorithm will not fail computationally.

4. The Resolved Rate Algorithm

In a similar manner a resolved rate algorithm [11] with a conventional proportional derivative joint controller and gravity compensation torque to eliminate steady state gravitational offsets was written as [10]:

$$\tau_m = [\mathbf{K}_p]\, \mathbf{J}_m^{-1} \{\mathbf{x}_{edes} - \mathbf{x}_e\} + [\mathbf{K}_d]\{\dot{\mathbf{q}}_{mdes} - \dot{\mathbf{q}}_m\} + \mathbf{G} \quad (9)$$

where

$$\dot{\mathbf{q}}_{mdes} = \mathbf{J}_{fb}^{-1} \left\{ \begin{bmatrix} \mathbf{R}_v^T & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_v^T \end{bmatrix} \{\dot{\mathbf{x}}_{edes} - \dot{\mathbf{x}}_v\} + \begin{bmatrix} [\mathbf{r}_e^v]^{\times} \mathbf{R}_v^T \\ \mathbf{0} \end{bmatrix} \omega_v \right\} \quad (10)$$

5. The Resolved Acceleration Algorithm

If the kinematic and dynamic parameters of the system are known, the resolved acceleration control method can be extended to mobile manipulators, such as has been done for free-flying space systems [13,15]. The resolved acceleration algorithm for terrestrial mobile manipulators yields the desired joint torques as [10]:

$$\tau_m = [\,\mathbf{J}(\theta_v,\mathbf{q}_m)\,(\mathbf{H}(\theta_v,\mathbf{q}_m)^{-1})_m\,]^{-1} \Big\{ \mathbf{a}_{des} - \dot{\mathbf{J}}(\theta_v,\mathbf{q}_m)\,\dot{\mathbf{q}} + \mathbf{J}(\theta_v,\mathbf{q}_m)\mathbf{H}(\theta_v,\mathbf{q}_m)^{-1} \left\{ \mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) - \begin{Bmatrix} \mathbf{F}_v(\mathbf{x}_v,\dot{\mathbf{x}}_v) \\ -\mathbf{V}\,\dot{\mathbf{q}}_m \end{Bmatrix} \right\} \Big\} \quad (11)$$

where $(\mathbf{H}^{-1})_m$ represents the last n columns of $\mathbf{H}^{-1}$, and where the desired end-effector acceleration vector $\mathbf{a}_{des}$ is a combination of feed-forward acceleration, and endpoint position, $\mathbf{x}_e$, and velocity, $\dot{\mathbf{x}}_e$, feedback, such as:

$$\mathbf{a}_{des} = \ddot{\mathbf{x}}_{edes} + [\mathbf{K}_p]\{\mathbf{x}_{edes} - \mathbf{x}_e\} + [\mathbf{K}_d]\{\dot{\mathbf{x}}_{edes} - \dot{\mathbf{x}}_e\} \quad (12)$$

IV. CONTROL ALGORITHM SELECTION

The three algorithms presented in the previous section rely on endpoint as well as vehicle orientation sensing ($\mathbf{x}_e$, $\dot{\mathbf{x}}_e$, θ_v), in addition to conventional joint sensing ($\mathbf{q}_m$, $\dot{\mathbf{q}}_m$). Furthermore, the resolved acceleration algorithm requires sensing or estimation of the vehicle's position and veloci-ty. While all three algorithms require knowledge of the system's kinematic and the manipulator's mass properties, only the resolved acceleration algorithm requires knowledge of the inertia properties of the manipulator and of the vehicle and suspension's dynamic properties. This makes the resolved acceleration algorithm more complex, computationally expensive, and sensitive to modelling errors.

Assuming, ideally, that all the sensing and modelling requirements are met, one can compare the performance of these algorithms in simulation, as in figure 7. For this comparison, the control gains of the three algorithms were chosen to provide a similar performance when applied to a fixed-base manipulator without gravity compensation.

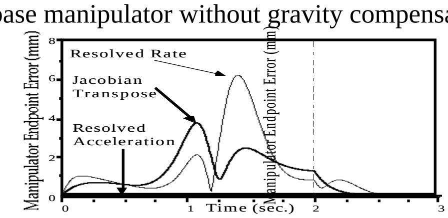

Fig. 7. Endpoint Errors for Three Controllers.

It can be seen from figure 7 that the resolved rate algorithm had a maximum endpoint error of 6.2 mm and settled in 0.3 seconds. Thus it satisfies the performance requirements and is a significant improvement over the conventional PD approach shown in figure 4 which had a 73 mm maximum endpoint error and took over a second to settle. The jacobian transpose algorithm performed even better. With a 3.7 mm maximum endpoint error, it settled in 0.12 seconds. As expected, the resolved acceleration algorithm which uses a complete knowledge of the system performs ideally with effectively zero error.

As shown in figure 7, the performance of the jacobian transpose algorithm is slightly better than that of the resolved rate algorithm. In addition, by virtue of its nature requiring a transpose as opposed to an inverse jacobian operation, the jacobian transpose algorithm is more robust in the presence of singularities, where the resolved rate algorithm fails. Hence, the jacobian transpose algorithm was considered preferable to the resolved rate algorithm.

The fundamental difference between the jacobian transpose algorithm and the resolved acceleration algorithm is the reliance of the latter on the complete dynamic model of the system. However, in practice modelling errors cannot be avoided due to variations in vehicle and manipulator loads, tire and ground characteristics and other factors. In such cases, the use of the resolved acceleration algorithm can result in substantial transient and steady state endpoint errors, as shown in figure 8.

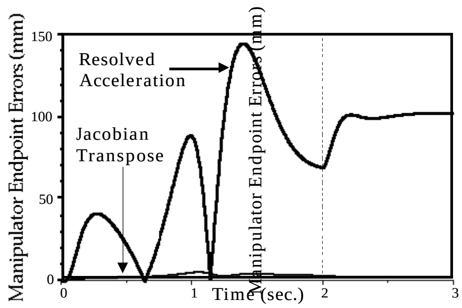

Fig. 8. Endpoint Error with Modelling Errors.

Figure 8 compares the performance of the jacobian transpose and the resolved acceleration algorithms under the assumption of a 10 % error in system masses, a 20 % error in system inertia's, a 25 % error in suspension characteristics, and an unmodelled 25 kg payload. The required endpoint and vehicle motion sensing is assumed to be available. Clearly, the resolved acceleration algorithm results in unsatisfactory errors, with a maximum error of 145 mm, a steady state error of 102 mm, and a settling time of 0.7 seconds. A good adaptive controller can overcome these modelling errors [16]. However, this adds substantially to the complexity of an already complex approach. The far simpler jacobian transpose control algorithm, on the other hand, deteriorates only by about 2-3 mm, still well within the performance specs.

V. JACOBIAN TRANSPOSE ALGORITHM WITH LIMITED SENSING

1. The Problem

Since the jacobian transpose algorithm also has a good performance in the case of substantial modelling errors, it was chosen for further development. The results demonstrated above use both manipulator endpoint as well as vehicle orientation sensory data. The question as to the algorithm's performance with limited sensory information is presented in this section.

In practice, a mobile manipulator often operates in highly unstructured environments. Consequently, vehicle sensing may be limited due to a lack of good environmental reference points. In addition, although it may still be possible to perform endpoint sensing in the close vicinity of the target [9], imagine the difficulty in achieving accurate 6-dimensional endpoint sensing along the entire trajectory of large motions away from this target.

Without vehicle orientation and endpoint sensory information the performance of a jacobian transpose controller on a moving vehicle is similar to that of the simple joint controller shown in figure 4. A significant part of the resulting endpoint error is caused by static gravitational effects. One could use a model of the system's mass properties and suspension characteristics to compensate for this error, estimating the quasi-static vehicle orientation. The result of this approach is shown in figure 9, where the maximum transient error is reduced from 70 to 21 mm. The settling time is reduced from 0.9 sec. to 0.8 sec. Still, the result does not meet the performance specifications. Furthermore, it requires a good knowledge of the payload and manipulator weights. Errors in these quantities will cause the performance to degrade further.

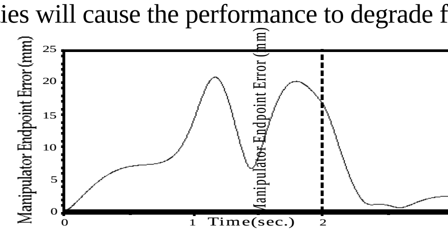

Fig. 9. Endpoint Error with Estimation.

2. Using Vehicle Sensors to Replace Endpoint Sensing

Our studies show that the effectiveness of the jacobian transpose methods and other algorithms such as resolved acceleration comes from the use of endpoint sensing. As discussed above, this is very difficult to obtain in practice. Here the use of sensors to measure the vehicle's motion and to use this information to replace endpoint sensing was investigated. This proved to be an effective approach.

Although complete vehicle motion sensing may not be feasible, limited vehicle sensing is. Pitch and roll sensing can be performed using inexpensive inclinometers that exploit gravity, although yaw motion is difficult to measure. Note that the pitch and roll motion corresponds to the vehicle motions which cause the most significant errors. Similarly, it is relatively easy to measure the heave (z) motion of the vehicle, since the ground surface provides a reference plane for ultrasonic range sensing. However, there are often no similarly convenient references for sensing lateral motions. Fortunately, a vehicle's suspension will typically be least stiff in the heave direction, to isolate rough ground vibrations during travel, and thus this motion will be substantially larger than lateral motions of the vehicle. Hence, the use of vehicle sensing to replace endpoint sensing was limited to easily measured data.

The jacobian transpose control algorithm yields good transient and steady state endpoint errors as well as a good settling time of 0.13 seconds in simulation, if one uses only the relatively easily measured vehicle motions (z, ψ_x, and ψ_y), and assumes that non-measurable vehicle variables are zero. In fact, as figure 10 shows, the performance approaches the ideal performance of the jacobian transpose control algorithm with complete endpoint and vehicle sensing. With the addition of modelling errors, endpoint errors increased on the order of those shown in figure 8 for the jacobian transpose control with endpoint sensing, approximately 3 mm.

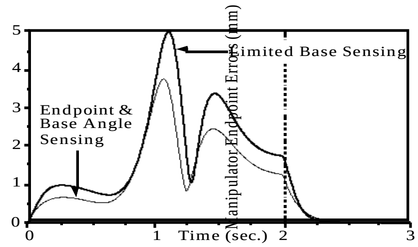

Fig. 10. Error with & without Endpt. Sensing.

VI. CONCLUSIONS

Conventional fixed-base joint controllers do not perform well on terrestrial mobile manipulators due to vehicle motions produced by gravity effects and dynamic interactions between a manipulator and its suspended vehicle. Here an extended jacobian transpose control algorithm is presented, along with two other algorithms, that compensates for these effects. In the past such controllers have always relied on endpoint sensing in their implementations, and the jacobian transpose control algorithm performs well with such endpoint sensing even in the presence of modelling errors. However, in practice, in remote unstructured environments, endpoint sensing is not possible for large motion trajectories. Without vehicle motion sensing the jacobian transpose algorithm does not perform sufficiently well, even when the vehicle orientation is estimated using a static gravity model. But by using limited, readily available vehicle sensors, such as inclinometers and ultrasonic sensors, endpoint sensing can be replaced and the performance of the jacobian transpose control algorithm approaches that achieved by endpoint sensing. The results strongly suggest that other methods such as resolved acceleration would benefit from this approach.

VII. ACKNOWLEDGEMENTS

The support of this work by DARPA (US Army Human Eng. Lab. and Oak Ridge National Lab. as agents - Martin Marietta Subcontract No. 19X-5570C) is acknowledged.

VIII. REFERENCES

[1] Goldsmith, S., “It’s a Dirty Job, but Something’s Gotta Do It,” *Business Week*, pp. 92-97, Aug. 20, 1990.

[2] Dubowsky, S., and Tanner, A. B., “A Study of the Dynamics and Control of Mobile Manipulators Subjected to Vehicle Disturbances,” Proc. IV Int. Symp. of Robotics Research, Santa Cruz, CA, 1987.

[3] Blume, C., Gremminger, K., Messemer, G., Smidt, D., and Wadle, M., “EMIR - A Combination Of Manipulator and Robot for New Out-Door Applications in Unstructured Environments,” Proc. IEEE Int. Conf. on Robotics and Automation, Scottsdale, AZ, May 1989.

[4] Vafa, Z., and Dubowsky, S., "The Kinematics and Dynamics of Space Manipulators: The Virtual Manipulator Approach," *The Int. Jour. of Robotics Research.* Vol 9, No. 4, August 1990, pp. 3-21.

[5] Papadopoulos, E., and Dubowsky, S., “On the Nature of Control Algorithms for Space Manipulators,” Proc. IEEE Int. Conf. on Robotics and Automation, Cincinnati, OH, May 1990.

[6] Jacobs, P., and Canny, J., “Planning Smooth Paths for Mobile Robots,” Proc. IEEE Int. Conf. on Robotics and Automation, Scottsdale, AZ, May 1989.

[7] Sekiguchi, M., Nagata, S., and Asakawa, K. “Behavior Control for a Mobile Robot by Multi-Hierarchical Neural Network,” Proc. IEEE Int. Conf. on Robotics and Automation, Scottsdale, AZ, May 1989.

[8] Dubowsky, S., and Vance, E.E., "Planning Mobile Manipulator Motions Considering Vehicle Dynamic Stability Constraints," Proc. IEEE Int. Conf. on Robotics and Automation, Scottsdale, AZ, May 1989.

[9] West, H., Hootsmans, N., Dubowsky, S., Stelman, N., "Experimental Simulation of Manipulator Base Compliance," Proc. First Int. Symp. on Experimental Robotics, Montreal, Canada, June 1989.

[10] Hootsmans, N.A.M., “The Control of Manipulators on Mobile Vehicles,” Sc.D. Thesis, Dept. of Mechanical Engineering, MIT, Cambridge, MA, 1991.

[11] Whitney, D.E., “Resolved Motion Rate Control of Manipulators and Human Prostheses,” *IEEE Trans. on Man-Machine Systems,* Vol. MMS-10 No.2, June 1969.

[12] Hogan, N., “Impedance Control: An Approach to Manipulation: Part II - Implementation,” *ASME Jour. of Dynamic Systems, Measurement and Control*, Vol. 107, pp 8-16, March 1985.

[13] Luh, J.Y.S., Walker, M.W., and Paul, R.P.C., "Resolved-Acceleration Control of Mechanical Manipulators", *IEEE Trans. on Automatic Control*, Vol AC-25, No.23, pp. 468-474, June 1980.

[14] Khatib, O, “A Unified Approach for Motion and Force Control of Robot Manipulators: The Operational Space Formulation,” *IEEE Jour. of Robotics and Automation*, Vol. RA-3, No. 1, pp. 43-53, Feb., 1987.

[15] Alexander, H.L., Cannon, R.H. Jr, “Experiments on the Control of a Satellite Manipulator,” Proc. American Control Conference, Seattle, WA, June 1987.

[16] Slotine, J.-J., and Li, W., “On the Adaptive Control of Robot Manipulators,” *Int. Jour. of Robotics Research*, Vol. 6, No. 3, Fall 1987.