An enhanced mobile manipulator

Arnaud Lelevé, Pierre Dauchez, Philippe Fraisse, François Pierrot

AN ENHANCED MOBILE MANIPULATOR

A. LELEVE, P. DAUCHEZ, P. FRAISSE, F. PIERROT
LIRMM - UMR 5506 CNRS / Université Montpellier II
161 rue Ada - 34392 Montpellier CEDEX 5 - France

ABSTRACT

Since 1995, we have been developing a terrestrial mobile manipulator that is able to operate on uneven terrain. It has been equipped with several sensors and various control modes and laws have been tested. This paper presents the hardware of our enhanced mobile manipulator and describes some tasks it is able to perform in automatic or teleoperated modes within a scenario of manipulation of an explosive charge.

KEYWORDS: mobile manipulation, teleoperation

INTRODUCTION

There are situations when firms or laboratories have to resort to remote manipulation. We can find such cases when dangerous objects have to be handled or/and when the environment is too aggressive for humans. A typical example is the handling of an explosive charge in building sites. One needs to carry the charge to the target place and then to carefully put it in a cavity. Such a manipulation is usually performed by a specialist who knows the danger it means. We propose a robotic solution that would avoid an accident while carrying the charge to the explosion area. We consider using a mobile manipulator which would carry the charge close to the target area (**stage 1**) by handling it at the end of a robotic arm. This arm would be controlled such as to avoid the charge to be shaken about. Meanwhile the vehicle would be teleoperated by an operator located in a safe area. This way he would just have to pilot his vehicle without paying attention to the charge automatically handled by the arm. Moreover, the operator could avoid obstacles and be able to drive in an unknown or dynamic environment thanks to several views from cameras located aboard the mobile manipulator. Once arrived near the target area, the operator would just have to leave the mobile manipulator moving by itself in order to place the vehicle and the arm in a position (**stage 2**) that would allow the operator to manually and remotely put the charge in its final place (**stage 3**). Then, the operator would just have to leave the area by teleoperating the vehicle as in stage 1.

Theses three stages involved in this typical **scenario** introduce three different ways of controlling the mobile manipulator:

- Stage 1 combines the basic teleoperation of a vehicle with some kind of force control of the arm in order to compensate for the inertial forces applied to the object to handle.
- Stage 2 involves an autonomous behavior of the mobile manipulator; it offline computes the best way to move the object close enough to its target place. This operation involves simultaneous movements of both the vehicle and the robotic arm.
- Stage 3 corresponds to a teleoperation case with force feedback.

In a first place, we will describe our enhanced mobile manipulator. We will then explain how it is possible to teleoperate it for an application such as in stages 1 and 3. Next, we will introduce the control law used to compensate for inertial effects in stage 1. Finally we will briefly present the global motion generation method used for stage 2.

DESCRIPTION OF THE EXPERIMENTAL SETUP

Our Mobile Manipulator

It is a terrestrial vehicle as pictured in Figure 1. The original vehicle is a 6 directive and propulsive wheel electric vehicle. We have modified the electric part in order to be able to electronically pilot it and we have added a *PUMA 560* robotic arm at the rear of it. This mobile manipulator is entirely autonomous concerning its power supply.

Figure 1: Picture of our mobile manipulator

Control hardware

The mobile manipulator is entirely controlled by a single processor: a *dSPACE* set based on a *Texas Instrument C40* DSP. Encoders on the steering wheel and on the left front wheel (we assume it does not skid) permit to control the vehicle. A force transducer set installed on the wrist of the *PUMA* (visible in figure 3) allows us to use force control schemes. A powerful PC laptop supervises the DSP set. As it is fitted with a radio *Ethernet* board for remote communication with an operator, it is as able to run autonomous scripts as being teleoperated. A differential GPS (not visible in figure1) gives the current position of the manipulator with an accuracy of about 5m.

Communication Structure for teleoperation

Communication between different parts of the mobile manipulator and the teleoperator is summed up in figure 2.

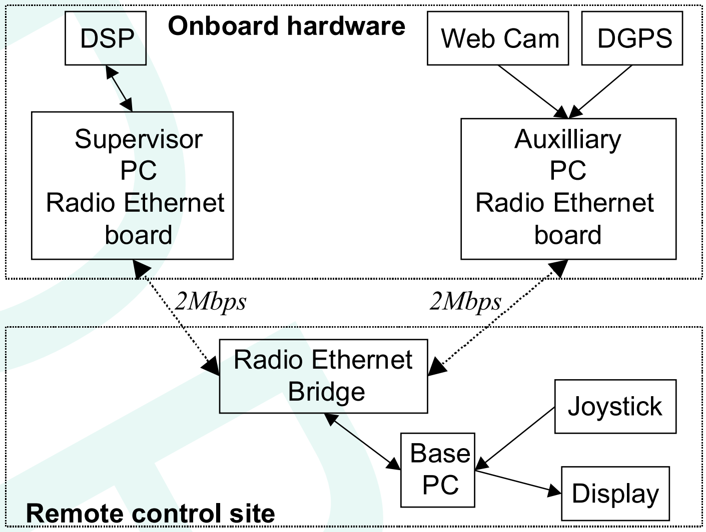

Figure 2: Communication structure for teleoperation

TELEOPERATION (STAGES 1 AND 3 OF THE SCENARIO)

In stage 1, the operator remotely pilots the vehicle while the robotic arm is autonomous. In stage 3, the operator controls the arm movements in order to carefully put the charge in its final place. Both stages use a same enhanced teleoperation structure as described below.

Enhanced Teleoperation Structure

To make it clear, we call the operator station (remote control site) *Base.* This station includes a powerful PC computer fitted with a joystick and a 19” screen. It is linked to the mobile manipulator through our laboratory local *Ethernet* network and through our local radio network. It discusses with the *supervisory PC* concerning the control of the mobile manipulator and with the *auxiliary PC* concerning video and *DGPS* feedback.

Currently, we only have one camera (webcam) that we have installed on the arm so that we can use it in every stage. In stage 1, the camera is orientated frontward in order to watch the path in front of the vehicle while the arm compensates for inertial effects on the charge it carries. When the arm is not carrying any object, the operator can turn the camera by simply moving the arm. In stage 3, the operator sees the movements of the arm as if he was “sitting” on it. In a first place, we have used already-made webcam software that could transmit through *Internet* the images from the camera to a common web browser running on the *Base* station. We are working on a software of ours that will allow us to simultaneously send several webcam pictures to the *Base.*

In stage 1, the operator uses the joystick to control the vehicle speed (axis Y) and steering (axis X). The problems that occur when teleoperating such systems are essentially due to communication limitations. In our case, we use a computer network which is also used by other people. As our mobile manipulator is located nearer than a few kilometers far from the *Base*, propagation delays are not prominent. But the network media still make performance worse because of the way data is transmitted (packetisation) and of the fact they are shared by a lot of simultaneous users.

In practice, the short distance network time delays vary according a stochastic *Poisson* law. These delays corrupt periodical signals by making their period vary from a sample to the other. Moreover, these delays are not symmetrical: mean value of data going from *Base* to the mobile manipulator may not be the same as the one in the opposite direction. It is so necessary to compensate for these variations of delays in both directions. To do so, we have developed a *Delay Variation Compensator* that acts as a FIFO file and makes the signal find back its initial sampling period. The method is described more precisely and tested in [1]. A similar method was used in [2]. The drawback of this method is to increase the mean delay by the time samples stored into the file. In case of long distance teleoperation, this is not very important because there is a delay that prevents for directly teleoperating the mobile manipulator. On the other hand, in our case of short distance teleoperation, we have to add a teleoperation control based on a prediction so that the operator does not perceive time delays. This method is described in [3] with simulation results and its structure is showed in figure 3.

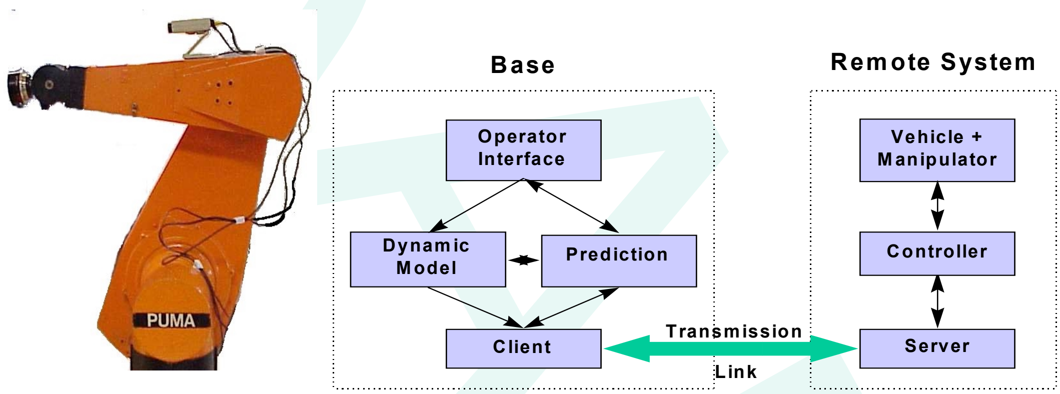

Figure 3: PUMA arm and enhanced teleoperation structure

Grasping and force-oriented teleoperation

In stage 3, the operator controls the robotic arm in Cartesian or joint space. A force transducer allows a force feedback. As our joystick does not include force feedback for the moment, we are juste able to display the forces and moments on the screen.

MANIPULATION OF FRAGILE OBJECTS (STAGE 1 OF THE SCENARIO)

In stage 1 of our scenario, the teleoperator pilots the vehicle while he arm has to compensate for any disturbance due to the vehicle's motion that could affect the manipulated object. Therefore it is necessary that the arm acts as a spring-damper system. In other words, some kind of impedance control of the arm must be designed. The one we have chosen and tested is briefly described here.

The impedance controller we have implemented is a classical one where Cartesian position and velocity gains are regarded as stiffness and damping matrices respectively.

The behavior of the robot is impedance-like if we implement a dynamic decoupling. The dynamic model of the manipulator is defined by :

$$F_c - F_{ext} = \Lambda(q).\ddot{X} + \mu(q;\dot{q}) + p(q) + \Gamma_f(q;\dot{q}) \quad (1)$$

where F_c is the (6×1) force control vector, F_{ext} is the external force vector, $\Lambda(q)$ is the inertia matrix, μ represents the Coriolis and centrifugal effects, p is the gravitational force vector and Γ_f the friction effects. The control vector F_c* which allows dynamic decoupling is such that :

$$F_c^* = \overline{\Lambda}.M_d^{-1}.(B_d.\dot{\varepsilon}_x + K_d.\varepsilon_x) \quad (2)$$

where ε_x is the tracking error vector defined as: $\varepsilon_x = X_d - X$; X_d is the (6×1) desired position/orientation vector, X is the actual position/orientation vector, M_d is the desired mass matrix, B_d and K_d are the gain matrices which define the impedance (respectively damping and stiffness matrix). These assumptions are verified if and only if $\dot{\varepsilon}_x$ and ε_x tend towards zero. The behavior of the robot is then given by the following equation:

$$-F_{ext} = M_d.\ddot{X} - B_d.\dot{\varepsilon}_x - K_d.\varepsilon_x \quad (3)$$

For our experiments, we have modified this basic control scheme by adding an external force control loop to obtain a position/force control scheme (as shown in figure 4). The force loop control law is an integral gain which modifies the desired trajectory (equation 4).

$$X_d^* = X_d + \Psi.K_f.S_f.\int_0^t (F_d - F).dt \quad (4)$$

The force vector F, the selection matrix S_f and the desired force vector F_d are defined in the task frame. The matrix Ψ allows the transformation between the task and the reference frame.

We have also tested two other controllers: the first one was a robust controller based on a sliding mode approach; the other one was an evolution of the impedance controller described above, with the introduction of an external acceleration loop. Experiments proved that both were less efficient that the “simple” impedance controller [5].

AUTOMATIC MOTION GENERATION (STAGE 2 OF THE SCENARIO)

Because of the redundancy and above all the non-holonomy or our mobile manipulator, it seems difficult to teleoperate it for the “fine” motion required in stage 2 of our scenario. Therefore we have developed a method that automatically generates (off-line, i.e. before operating the actual robot) the simultaneous motion of both the vehicle and the arm in order to go from an initial position/orientation of the end-effector to a final one. This final posture is assumed to be determined in the vehicle reference frame thanks to some sensors mounted on the robot (for instance a vision system analyzing the target location).

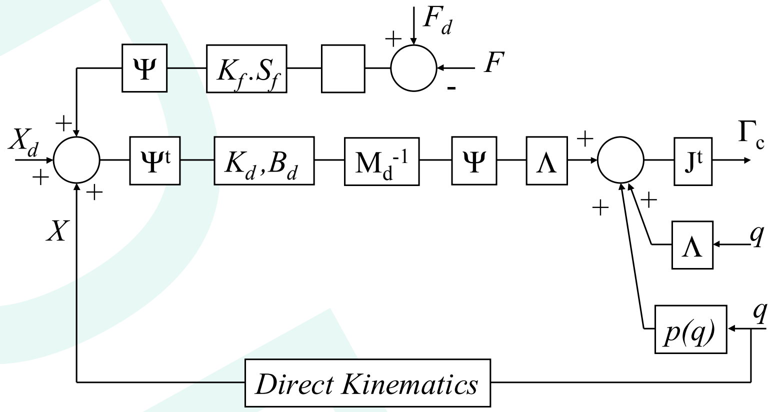

Figure 4: Impedance control scheme

It is however to be noted that this posture does not have to be determined extremely accurately. As a matter of fact, it is enough for the end-effector to reach the target with a positioning error that one can compensate with the teleoperated maneuver of stage 3.

The method proceeds as follows. The initial position of the vehicle reference frame **Mv** defines a world reference frame **F**. Let $^{F}\mathbf{D}_{\mathbf{v(init)}}$ be the displacement between **F** and the initial **Mv**. As shown in figure 5, the non-holonomic motion between two successive vehicle reference frames $\mathbf{Mv}_{(t)}$ and $\mathbf{Mv}_{(t+dt)}$ can be represented by the displacement $^{v}\mathbf{D}_{\mathbf{v}(t)}$, which can be written as a function of the curvature S (the inverse of the radius of curvature R) and v (the velocity of the vehicle). At any time t, the position of the end-effector in the vehicle reference frame $\mathbf{Mv}_{(t)}$ can be represented (figure 5) by the displacement $^{v}\mathbf{D}_{e(t)}$ between $\mathbf{Mv}_{(t)}$ and $\mathbf{Me}_{(t)}$, **Me** being the end-effector frame. This displacement can be written as a function of the arm joint parameters θ_1, θ_2, ..., θ_6 at time t. So, at any time t+dt, the position of $\mathbf{Me}_{(t+dt)}$ depends on the previous position of the vehicle $\mathbf{Mv}_{(t)}$ and the current joint values of the arm $\theta=[\theta_1(t+dt), \theta_2(t+dt), ..., \theta_6(t+dt)]^T$.

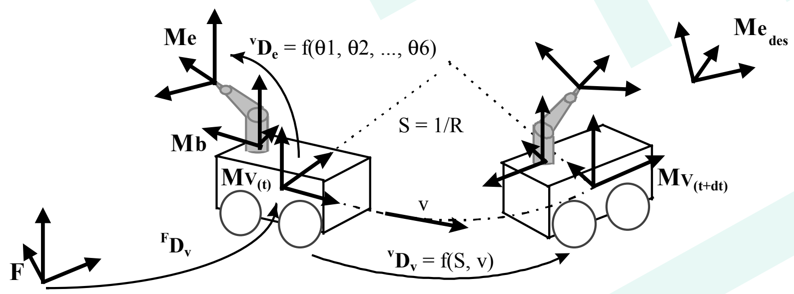

Figure 5: Considered variables

Therefore, after n displacements, the position of the end-effector in the world reference frame is given by the displacement $^{F}\mathbf{D}_{\mathbf{e(tn)}}$ which is the composition of $^{F}\mathbf{D}_{\mathbf{v(init)}}$, $^{v}\mathbf{D}_{\mathbf{v(ti)}}$ for i=1..n, and $^{v}\mathbf{D}_{\mathbf{e(tn)}}$ as shown in figure 6. So, $^{F}\mathbf{D}_{\mathbf{e(tn)}}$ is written as a function of every joint parameter $\{(Si)_{i=1..n},(vi)_{i=1..n},(\theta i)_{i=1..6}\}$. $(Si, vi)_{i=1..n}$ are respectively the n curvatures and the n velocities of the vehicle involved in the n displacements $({}^{V}\mathbf{D}_{v(ti)})_{i=1..n}$, and $(\theta i)_{i=1..6}$ are the six joint parameters of the arm involved in the last displacement ${}^{V}\mathbf{D}_{e(tn)}$.

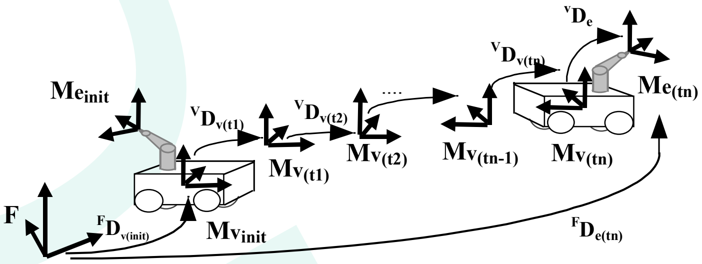

Figure 6: Considered displacements

The desired position of the end-effector in the world reference frame **F** is represented by the displacement ${}^{F}\mathbf{D}_{e(goal)}$. Then, we have to compute the 2n+6 joint variables $\{(Si)_{i=1..n},(vi)_{i=1..n},(\theta i)_{i=1..6}\}$ that make the feasible displacement ${}^{F}\mathbf{D}_{e(tn)}$ equal to the desired displacement ${}^{F}\mathbf{D}_{e(goal)}$. The algorithm proceeds a follows. First of all, the number n of displacements must be fixed a priori. It can be increased if the global displacement does not allow the mobile manipulator to come close enough to the goal (recall that everything is computed off-line, i.e. the result can be checked before sending the commands to the actual robot). n is initialized as the ratio of the distance to be covered to the product of a desired average velocity v_m of the vehicle (determined by the capabilities of the vehicle) by the sampling period dt: $n = \frac{\| D\varepsilon \|}{v_m dt}$. Then figure 7 shows how this method is implemented.

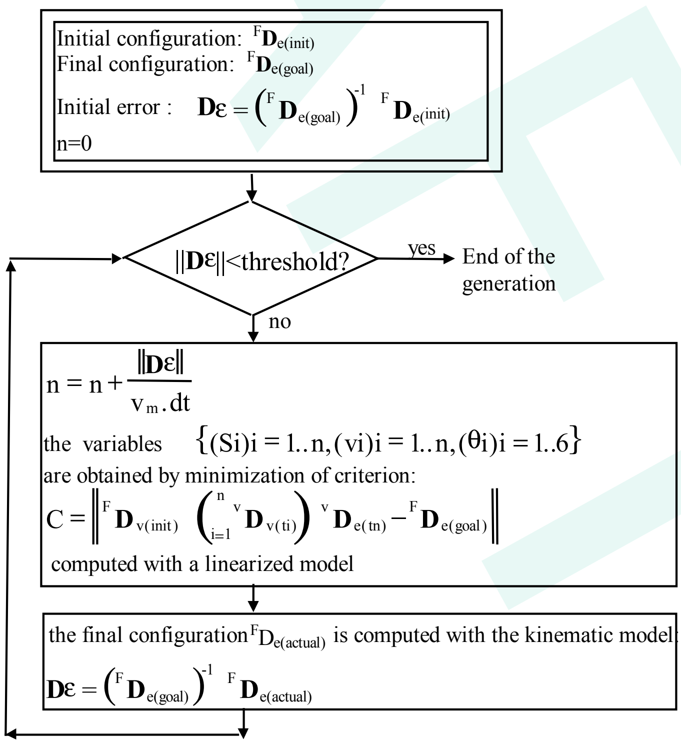

Figure 7: Implemented algorithm («o» denotes the composition of two transformations)

CONCLUSION

This paper was an overview of the current state of our mobile manipulator and of the manipulation tasks it is now able to perform. Real tests have been successfully carried out for all the methods we have proposed in this paper. However, we have not run yet a global experimentation that would correspond to the scenario described in the introduction. In some sense, we have built many pieces of a jigsaw but they are not assembled yet. To do so, we need to be able to switch from a working mode to another without electrically stopping the robot (for instance from force-control to motion generation). This is what we are currently studying. We look forward to running a complete complex task in the near future.

REFERENCES [1]

1. A. Lelevé, P. Fraisse, A. Crosnier, P. Dauchez, F. Pierrot, *"Towards Virtual Control of Mobile Manipulators"*, Proc. of the 3rd World Automation Congress (WAC'98), Anchorage, USA, 1998, pp. ISORA 049.1-ISORA 049.6 (CD-ROM).

2. K. Kosuge, H. Murayama, K. Takeo, *"Bilateral Feedback Control of Telemanipulators via Computer Network"*, Proc. of the Intl. Conf. on Intelligent Robots and Systems (IROS'96), pp. 1380-1385.

3. A. Lelevé, P. Fraisse, P. Dauchez, F. Pierrot, *"Modeling and Simulation of Robotic Tasks Teleoperated through the Internet",* Proc. of the Intl. Conf. on Advanced Intelligent Mechatronics (AIM'99), Atlanta, USA, 1999, pp. 299-304 (CD-ROM).

4. A. Lelevé, P. Fraisse, P. Dauchez, F. Pierrot, *"Teleoperation through the Internet : Experimental Results with a Complex Manipulator"*, Proc. of the Intl. Conf. on Robotics (ISR'99), Tokyo, Japan, 1999, pp. 63-70.

5. P. Fraisse, P. Dauchez, F. Pierrot, L. Cellier*, "Mobile manipulation of a fragile object"*, Proc. of the 4th Intl. Symp. on Experimental Robotics (ISER'95), Stanford, USA, 1995, pp. 290-295.

[1] Because this paper exclusively presents methods we have developed at LIRMM, almost all references correspond to our work on mobile manipulators. These references, in turn, refer to other works.