TRAJECTORY GENERATION FOR MOBILE MANIPULATORS USING A LEARNING METHOD

FOUDIL ABDESSEMED
Al Jouf University
Kingdom of Saudi Arabia
Al Jouf - Sakaka
Email:fodil_a@hotmail.com

Abstract **In this paper, the generalized trajectory of a mobile manipulator is obtained. The objective is to allow the end-effector track a given trajectory in a fixed world frame. The motion of the platform and that of the manipulator are coordinated by a pseudo neural network designed from the kinematics model of the system. A learning paradigm is used to produce the required reference variables for each of the mobile platform and the robot manipulator for an overall coordinate behavior. Simulation results are presented to show the effectiveness of the proposed scheme.**

***Keywords*:** Mobile Manipulator, Pseudo neural-network, Coordination, Learning rule.

1. INTRODUCTION

In recent years, path planning and motion control of mobile manipulators have emerged as a very significant one in the area of autonomous robotics. However, few solutions have been provided. A mobile manipulator system is a robotic manipulator mounted on mobile platform. This combination allows manipulation tasks over unlimited working space. However, since the platform and the manipulator have independent movement, a particular point in the workspace may be reached in multiple configurations, resulting in a system with redundancy. This can be helpful when it is desirable to perform tasks in a cluttered environment, or to optimally configure the system. Considerable effort is being devoted by researchers to investigate the capabilities of mobile platforms with onboard manipulators. Dubowsky and Tanner [1] associated a holonomic platform to a robotic manipulator. A framework is used to deal with motion planning and control to minimize a certain given criterion. Yamamoto and Yun [2] in their work considered the manipulator fixed during the platform motion, and the platform to be anchored during manipulator motion. They proposed a control algorithm for the platform so that the manipulator is always positioned at preferred configurations. Separate control of the end effector and the mobile plate form has been also considered. This approach proposed by Ficher *et al.*[3] is based on the natural separation of the two motion subsystems using a fuzzy based locomotion control strategy including criteria evaluating the desired system configurations. The problem of controlling the simultaneous motions of the mobile platform and the robot manipulator has been the area of research of several authors. Pin *et al* [4] designed the *FSP* method to optimally solve the inverse kinematics problem for redundant systems in the presence of applied constraints and behavioral criterion. In the work of Lee and Cho [5], we find a motion planning method for mobile manipulator to execute a multiple task which consists of a sequence of tasks. They formulate the motion planning problem as a global optimization problem and simultaneously obtain the motion trajectory set and commutation configurations. Zhao *et al* [6], have considered simultaneous motions of the base and the manipulator. They developed a genetic algorithm to solve the optimal sequence of base positions and manipulator configurations given a series of task specifications. Chen and Zalzala, have proposed a genetic algorithm approach to multi-criteria motion planning of mobile manipulator systems [7]. In developing their approach they have taken into consideration the dynamics and the non-holonomic constraints. Seradji [8], obtained the required mobile platform and robot manipulator motions by solving a set of differential kinematic equations resulted from the combination of the non-holonomic platform constraint, the desired end effector motion, and additional constraints specified by the user. Although from the computational point of view the Moore-Penrose pseudo-inverse as the generalized inverse is very attractive, the dynamic parameters of the manipulator are not taken into account. Based on the space formulation, Khatib *et al*. [9][10] developed a model for the mobile manipulator viewed as a combination of two subsystems. The mobile platform is considered as a macro-mechanism with coarse, slow dynamic response, and the arm is a fast and accurate mini-device. Their approach is based mainly to controlling redundant systems by obtaining the end-effector dynamic model by projecting the mechanism dynamics into the operational space, and a dynamically consistent force/torque relationship that provides decoupled control of motions in the null space associated with the redundant mechanism. The work of Nassal [11], emphasizes on the motion of a mobile two arm systems.

The manipulator controller uses the desired end-effector position and the current platform position to compute the frame of the end-effector with respect to the robot base. This frame is mapped to a vector of joint angles that are evaluated by the mobile platform which uses a cost function for that purpose. The cost function are mapped to a gradient that serves as a vector error signal to the coordinated motion controller. But, when one focuses on the different articles appeared in this area, he can classify these different schemes into global or local approaches. This depends on whether we want to compute the optimal configuration with respect to the whole path, e.g. [12], or with respect to the transition of tasks, e.g. [13]. However, although a lot of papers have dealt with this topic [14][15][16], it is still considered as new. The main contribution here is the organization of the kinematic model of the mobile manipulator into a sort of graph of operation chosen to be called a pseudo-neural network structure. In this paper, the motion of the platform and the motion of the manipulator are coordinated by this pseudo neural network that works like a supervisor. This network provides reference output values of the desired motion to the mobile manipulator system. We have chosen to apply the Levenberg-Marquardt back-propagation learning rule to generate the appropriate parameter weighting vector. Once determined, this vector is used to compute inputs to the mobile platform and to the manipulator so that the end-effector trajectory, specified in a fixed world frame, is tracked with minimum error.

2 ANALYSIS OF THE MOBILE MANIPULATOR SYSTEM

In this section we present the analysis of the mechanical system made up of the non-holonomic platform upon which is mounted a robot manipulator with 3 rotational degrees of freedom. The joint coordinates of the manipulator are $q_m = (q_{m1,}\ q_{m2},\ q_{m3})^T$ (thus *n=3*). Therefore the generalized coordinates of the mechanical system are $q = (q_{1,}\ q_2,\ \ldots\ q_6)^T = (x_A,\ y_A,\ z_A,\ q_{m1},\ q_{m2},\ q_{m3})^T$. Hence, the generalized space dimension of the mechanical system is equal to $\lambda=6$. Now, for a given mechanical configuration system $\boldsymbol{q}$, its structures imposes to its end effector η position and orientation constraints. Since only the end effector position is considered in this study, the number of constraints is reduced to 3. Thus the degree of freedom is equal to $\sigma=3$. This is the number of coordinates needed to configure locally the end effector in the configuration space, i.e. $\boldsymbol{X}=(x_E,\ y_E,\ z_E)$. On the other hand, we observe that the system is non-holonomic, and taking into account the constraint of non-holonomy of the mobile platform, we can deduce the degree of mobility of the system and which is equal to $(\sigma-1)$. We deduce in this case the order of redundancy and which is equal to $(\lambda-\sigma-1) = 2$. This redundancy helps in increasing the manipulator dexterity, the arm is prevented from singular configurations, and the system avoids obstacles while completing the given task. Unfortunately, control of such mechanisms becomes much harder. Following the *D-H* parameterization, the outputs of the network are given by equations (1), which designates the Cartesian coordinates of the task variable *E*, with respect to the world frame{***W***}. In a closed form this can be written as $\boldsymbol{X_E(t)=F(q(t))}$; where $\boldsymbol{F}$ represents the direct kinematics mapping from the joint space to the task space.

$$
\begin{aligned}
x_{E/W} &= x_{A/W} + \cos(\theta).\ \left[l_2\cos(q_{m2}) + l_3\cos(q_{m2}+q_{m3})\right] \\
y_{E/W} &= y_{A/W} + \sin(\theta)\ .\left[l_2\cos(q_{m2}) + l_3\cos(q_{m2}+q_{m3})\right] \\
z_{E/W} &= z_{A/W} + l_1 - l_2\sin(q_{m2}) - l_3\sin(q_{m2}+q_{m3})
\end{aligned} \tag{1}
$$

such that,

$$\theta = q_{m1} + \varphi \tag{2}$$

Where φ is the heading angle of the mobile platform, and l_1, l_2 and l_3 are the lengths of the three segments composing the manipulator arm. $x_{A/W}$, $y_{A/W}$ and $z_{A/W}$ are the coordinates of the point ***A*** located in the front of the mobile platform with respect to the world coordinates. In the sequel, we consider $z_{A/W}$ equals zero for simplicity. The goal is to find the generalized trajectory $\boldsymbol{q}(t)$ for a given task space trajectory $\boldsymbol{X_E}(t)$ such that $\boldsymbol{F(q(t)) = X_E(t)}$ is satisfied.

3. TRAJECTOTY GENERATION OF THE MOBILE MANIPULATOR

To provide a solution to the mobile manipulation motion, we have arranged the direct geometric model equations into a sort of graph of operation made up of three layers (Fig. 1). Each layer has a number of transfer functions each of which is found defined by the expressions given by equations (3). This neural network is the kernel of our proposal. This design will facilitate the implementation of the back propagation algorithm as a learning rule to adapt the weights so that the output values of the neural network come close to the desired reference values describing the task space trajectory. In this case, the two mechanical structures are considered as a unique entity. The accomplishment of the task is the result of the permanent movement of the two structures for which the success is based on the satisfaction of the tracking error.

With regard to the kinematic equations (1), let us define the following functions:

$$
\begin{aligned}
&f_{11}(\theta,\ w_{11}) = x_{11} = \cos(w_{11}\theta)\\
&f_{12}(q_{m2},\ w_{22}) = x_{12} = \cos(w_{22}q_{m2})\\
&f_{13}(x_{12}) = x_{13} = \cos^{-1}(x_{12})\\
&f_{14}(x_{13}, q_{m3},\ w_{33}) = x_{14} = (x_{13} + w_{33}q_{m3})\\
&f_{21}(x_{11}) = x_{21} = \sin(\cos^{-1}(x_{11}))\\
&f_{22}(x_{12},\ x_{14}) = x_{22} = l_3\cos(x_{14}) + l_2 x_{12}\\
&f_{23}(x_{13}) = x_{23} = l_2\sin(x_{13})\\
&f_{24}(x_{14}) = x_{24} = l_3\sin(x_{14})\\
&f_{31}(x_{11},\ x_{22}) = x_{31} = x_{11}.x_{22} + b_1 x_A\\
&f_{32}(x_{21},\ x_{22}) = x_{32} = x_{21}.x_{22} + b_2 y_A\\
&f_{33}(x_{23},\ x_{24}) = x_{33} = l_1 - x_{23} - x_{24}
\end{aligned} \quad (3)
$$

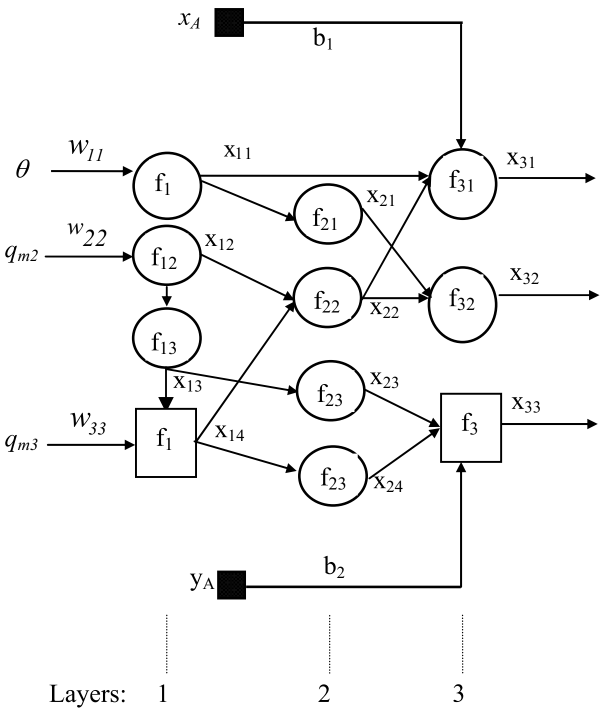

Fig. 1. The adaptive graph of operations

For convenience we define x_{31}, x_{32}, x_{33} as the outputs of the network, and which designates the Cartesian coordinates of the task variable $\boldsymbol{E}$: $x_{E/W}$, $y_{E/W}$ and $z_{E/W}$. Let **q**(*k*) be the input vector, such that $q^T(k) = [\theta, q_{m2}, q_{m3}, x_A, y_A]$ and X_E^m the measured output vector such that $X_E^m(k) = [x_e^m, y_e^m]^T$, and the weighting vector $\boldsymbol{W}^T(k) = [w_{11},\ w_{22},\ w_{33}]$. If we define the tracking error as being the sum of the squared errors given by equation (4), then the objective is to guaranty: $E_p \to 0$ *as* $k \to \infty$, (*k* designates the running time.)

$$E_p = \sum_{k=1}^{N(3)} (x_{3k} - r_k)^2 = \left[(x_{31} - r_1)^2 + (x_{32} - r_2)^2 + (x_{33} - r_3)^2\right] \quad (4)$$

Where $r_1 = x_e^d,\ r_2 = y_e^d,\ r_3 = z_e^d$ are the desired operational coordinates.

$x_{31} = x_e^m,\ x_{32} = y_e^m,\ x_{33} = z_e^m$ are the measured operational coordinates. We define:

$X_{e/W}^d = (x_e^d, y_e^d, z_e^d)^T$ and $X_{e/W}^m = (x_e^m, y_e^m, z_e^m)^T$.

If $\boldsymbol{W(k)}$ is the weight in the input layer, then the effect of adjusting it to the error E_p is determined by the ordered derivatives $\partial^+ E_p / \partial W(k)$ [17].

The main part of this design concerns how to apply the back-propagation learning rule to generate the appropriate parameter-weighting vector $\boldsymbol{W(k)}$. Once determined, the weights are used to update the input vector $\boldsymbol{q}$. The elements of this vector will serve as input references to a well-designed controller. The unknown weights are determined using the back propagation algorithm [18], in such a way the sum of the squared errors of equation (4) is minimized.

4. BACK-PROPAGATION LEARNING RULE

4.1. Output Layer

The error signal for the *j-th* output node can be calculated directly:

$$\varepsilon_{3,i} = \frac{\partial^+ E_p}{\partial x_{3,i}} = \frac{\partial E_p}{\partial x_{3,i}} \quad (5)$$

4.2. Internal Layers

The error signals of these internal nodes at the *j-th* position are calculated using the following equation

$$\underbrace{\varepsilon_{l,i} = \frac{\partial^+ E_p}{\partial x_{l,i}}}_{\text{Error Signal of layer } l} = \sum_{m=1}^{N(l+1)} \underbrace{\frac{\partial^+ E_p}{\partial x_{l+1,m}}}_{\text{Error Signal of layer } l+1} \times \frac{\partial f_{l+1,m}}{\partial x_{l,i}} \quad (6)$$

4.3. Input Layer

The first layer contains four neurons arranged in the way presented by figure 3. The general form for the error signal is given by the following equation:

$$\varepsilon_{1,i} = \sum_{m=1}^{4} \varepsilon_{2,m} \frac{\partial f_{2,m}}{\partial x_{1,i}} \qquad (7)$$

4.4. Weight adjustment

To adjust the weights we make use of the following known equation:

$$w_{ij}^{l}(k+1) = w_{ij}^{l}(k) - \mu \frac{\partial E_p}{\partial w_{ij}^{l}(k)}\bigg|_{w_{ij}^{l}(k)} \qquad (8)$$

Nevertheless, the steepest descent algorithm is slower for on-line applications. For that reason, we have used the Levenberg-Marquardt algorithm, which has proved to be an effective way to accelerate the convergence rate [19][20]. Its principal advantage is that it uses information about the first and second derivatives and does not need to invert the Hessian matrix.

$$x_{k+1} = x_k - \left[J^T J + \mu I\right]^{-1} J^T e \qquad (9)$$

Where, I is the identity matrix. The reference states of the plant at time $k+1$ are functions of the states and of the computed weights at time k. The functional relationship can be expressed symbolically as

$$\boldsymbol{q}(k+1)=\psi(\boldsymbol{q}(k), \boldsymbol{W}(k+1)) \qquad (10)$$

V. SIMULATION RESULTS

Simulation examples are carried out in order to evaluate the developed approach. It is desirable to move the end effector from its initial position *P1 (1, 1, 0.2)* to its final position *P2 (5,5,0.5)*, by tracking instantaneously a linear specified trajectory of the end effector generated by a uniform Cartesian movement. The pseudo neural network learns the desired values presented and adjusts the weights appropriately in order to output to the system the corresponding reference state variables. The results of the simulation are shown in figures 2 to 5 and indicate how successfully the Cartesian coordinates of the end-effector track their corresponding reference values very closely. We notice that the small departures from the reference trajectories are due to the cumulated tolerable errors from the learning process. The learning algorithm was run by using a learning rate μ*=0.05* for a laps of time not exceeding real time control. All the weights have been initialized to the value of one. At each step, the learning rate is updated depending on the behavior obtained. If the overall error is improved, then the learning rate is increased by the value $\mu=\mu*\mu\_inc$; otherwise, it is deceased by the value $\mu=\mu*\mu\_dec$, knowing that μ_inc and μ_dec take the values of *1.05*, and *0.95* respectively, and figure 6 clearly shows the trajectories of the end effector and the mobile platform in the xyz space. Figure 7 depicts a *3D* perspective of the simulation environment. The proposed neural network design with back propagation training rule is significantly simpler than the use of the pseudo inverses especially when there is a need to avoid obstacles that are on the desired path of the end effector.

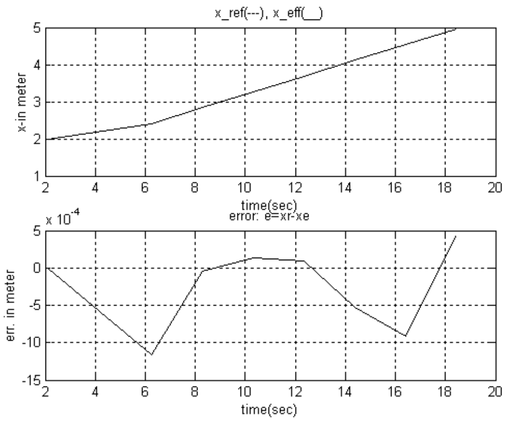

Fig. 2. Desired and measured x-trajectory plots and the error resulted.

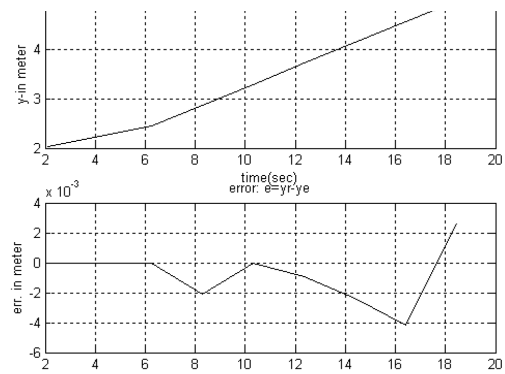

Fig. 3. Desired and measured y-trajectory plots and the error resulted.

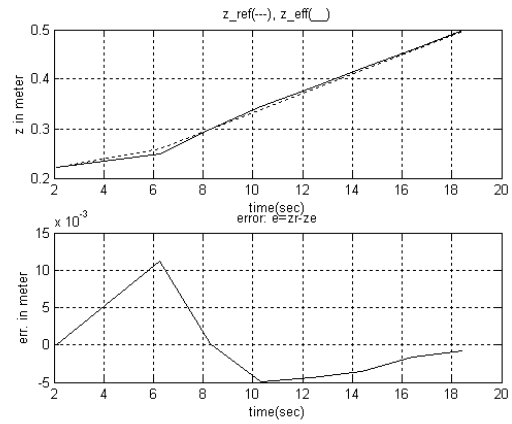

Fig. 4. Desired and measured z-trajectory plots and the error resulted.

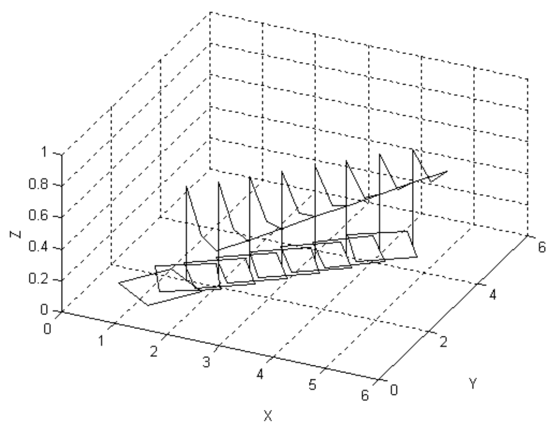

Fig. 7. A 3D perspective of the simulation environment

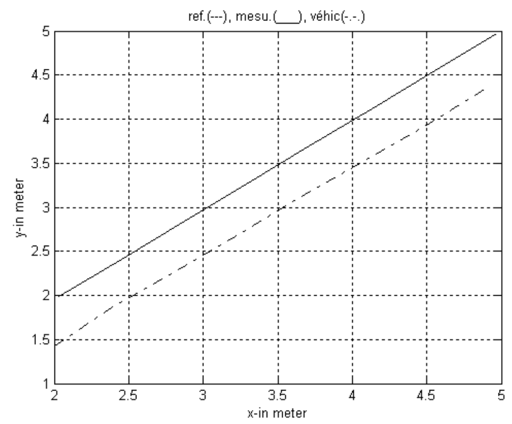

Fig. 5. X-Y Plots of the end-effector and the mobile platform trajectories.

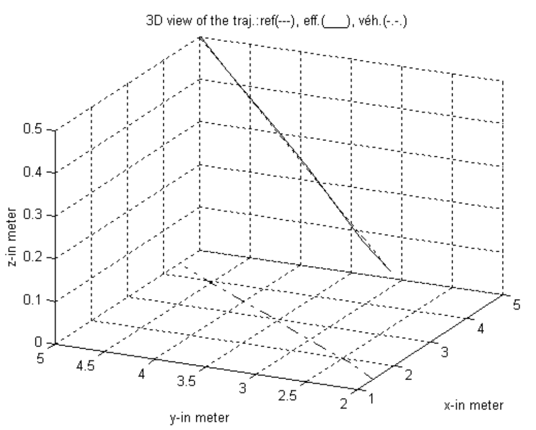

Fig. 6. X-Y-Z plots of the end-effector and the mobile platform trajectories

5. CONCLUSION

In this paper we propose a new scheme to motion control, designed to perform mobile manipulation tasks. It combines the motion of the robot manipulator with that of the mobile platform to execute an end effector tracking trajectory task. A pseusdo neural network assures the mapping from the operational space to the generalized coordinate space. This latter has been implemented in order to supply the low level controllers of the system with the appropriate generalized reference values when the operational coordinates of the end point of the manipulator are given. The results obtained by simulation, show that the proposed method performed very well. This controller offers a general solution to a class of mobile manipulators executing a task in the operational space in an unknown environment.

REFERENCES

[1]. S.Dubowsky and A.B. Tanner, A study of the dynamics and control of mobile manipulator subjected to vehicle disturbances,.*Robotics Research: 4th International Symposium*, 1988, pp 111-117.

[2]. Y.Yamamoto and X.Yun: Coordination locomotion and manipulation of a mobile manipulator, *IEEE Transactions on Automatic Control*, *39*(6),1994 1326-1332

[3]. C.Fischer, M.Buss and G.Shmidt: Soft control of an effector path for a mobile manipulator, WAC'96.

[4]. F. G. Pin, K. A. Morgansen, F. A. Tulloch, C. J. Hacker, and K. B. Gower Motion Planning for Mobile Manipulator with a Non-holonomic Constraints Using the FSP (Full Space Parametrization Method), *Jour. of Robotic Systems. 13*(11), 1996, 723-736.

[5]. J. K. Lee, and H..S. Cho Mobile Manipulator Motion Planning for Multiple Tasks Using Global Optimization Approach, *Jour. of Intell. and Robotic Systems*, 18, 1997.169-190,

[6]. M. Zhao, N. Ansari, and E. S. H. Hou Mobile Manipulator Path Planning by a Genetic Algorithm, Jour. of Robotic Systems, *11*(3), 1997, 143-153.

[7]. M.W.Chen and A.M.S. Zalzala, Dynamic modeling and genetic-based trajectory generation for non-holonomic mobile manipulators, *Control Eng. Practice, 5*(1), 1997, 39-48.

[8]. H. Seraji, Configuration control of rover-mounted manipulators, *IEEE Int.Conf. on Robotics and Automation,* 1995, 2261-2266,.

[9]. O. Khatib, Mobile Robotic Manipulation, *Symposium of the VDI/VDE society for Measurement and Automation Intelligent robot Control,* Langen, Germany, Nov. 1993, 51-66.

[10]. O. Khatib, K. Yokoi, K.Chang, D. Ruspini, R. Holmberg, and A. Casal, Vehicle/arm Coordination and Multiple Mobile Manipulator Decentralised Cooperation, *Proc. IROS96,* 546-553

[11]. U.M. Nassal, Motion coordination and reactive control of autonomous multi manipulators systems, *Jour. of Robotic Systems, 13*(11), 1996, 737-754.

[12]. F.G. Pin, K.A. Morgansen, F.A. Tulloch, C.J. Hacker, and K.B. Gower, Motion planning for mobile manipulator with non-holonomic constraint using the FSP method, *Jour.of Robotic Systems 13*(11), 1996, 723-736.

[13]. K. Nagatani and S. Yuta, Door-opening behavior of autonomous mobile manipulator by sequence of action primitives, *Jour.of Robotic Systems 13*(11), 1996, 709-721

[14]. M. Renaud et P. Dauchez, ″ Modélisation et commande des manipulateurs mobiles à roues", *JNRR'99-Monpellier* 1999, 179-192,

[12]. Y. Kanayama, A. Nilpour and C.A. Lelm, A Locomotion Control Method for Autonomous Vehicle, *IEEE Internat. Conf. Robotics and Automation*, 1988, pp.1315-1317.

[13]. I.J. Cox, Blanche - An Experiment Guidance and Navigation of an Autonomous Robot Vehicle, *IEEE Trans. Robot. Automat.*, 7, 1991, 193-204.

[14]. C. Samsung and K.A. Abderrahim, Mobile robot control Part.I: Feedback control of a nonholonomic wheeled cart in cartesian space, *Technical report, INRIA,* France, 1990

[15] O. Brock and O. Khatib, ″ Real time replanning in high-dimensional configuration spaces using sets of homotopic paths″, In Proc. of the Int. Conf. on Robotics and Automation, vol. 1, pp550-555, 2000.

[16] O. Brock and O. Khatib, Sriram Viji, ″ Task-Consistent Obstacle Avoidance and Motion Behavior for Mobile Manipulator ″, In Proc. of the IEEE Int. Conf. on Robotics and Automation, 2002.

[17]. P.Werbos, Beyond Regression: New tools for Prediction and Analysis in the Behavioral Sciences, *Ph.D. Dissertation*, Harvard Univ., 1974.

[18]. D.E.Rumelhart, G.E.Hinton, and R.J.Williams, Learning internal representations by Error Propagation, in Parallel Distributed Processing: Exploration, in the Microstructure of Cognition, *MA: MIT Press,* Vol.1, D.E.Rumelhart and James L. McClelland, Eds, Cambridge*,* Chap.8, pp318-362, 1986.

[19]. K.Levenberg, A method for the solution of certain nonlinear problems in least squares, Qty. Appl. Math., v.2, pp164-168, 1944.

[20]. D. W. Marquardt, An Algorithm for the estimation of non-linear parameters, SIAM J., v.11, 431-441, 1963.