Available online at www.sciencedirect.com
SCIENCE DIRECT®

Mechatronics 15 (2005) 989–1004

MECHATRONICS

Neural-network-based observer for real-time tipover estimation

A. Meghdari [a,*], D. Naderi [b], M.R. Alam [c]

[a] Center of Excellence in Design, Robotics, and Automation, School of Mechanical Engineering, Sharif University of Technology, Tehran, Iran
[b] Department of Mechanical Engineering, Bu-Ali Sina University, Hamedan, Iran
[c] Department of Mechanical Engineering, Masachusetts Institute of Technology, Cambridge, MA, USA

Accepted 2 March 2005

Abstract

Tipover issue is an important problem in autonomous mobile manipulators. This issue is becoming more important in new generation of mobile robots where their size and weight are reduced, and they are designed to work on uneven terrains with higher speeds. Estimating the distance from the tipover stability margin would enhance the capability of the mobile manipulators in correcting their motion. For a valid estimation of the tipover margin one must take into account the full dynamic interaction between the vehicle and its manipulator. In mobile manipulators with several degrees of freedom a huge amount of time is needed to solve equations of motion while real-time tipover control needs a fast and on-line access to the data. The proposed approach is a neural-network-based algorithm that enables autonomous mobile manipulator to detect its instable situations. This method greatly reduces the observer calculation time and is fast enough to be used as an observer for real-time tipover control. Accuracy and effectiveness of the proposed method is shown by an example.

Keywords: Mobile manipulators; Dynamic interaction; Tipover stability; Neural network

* Corresponding author.
E-mail address: meghdari@sharif.edu (A. Meghdari).

0957-4158/$ - see front matter
doi:10.1016/j.mechatronics.2005.03.005

1. Introduction

A mobile manipulator is a multi-link arm placed on a moving vehicle. Examples of these kinds of robots are those used in construction, mining and forestry industries. From control point of view, they are either controlled by on-board operators or categorized as remotely operated vehicles (ROV). They are especially useful for exploration in unknown and dangerous regions, like space and underwater tasks. Tipover issue has always been a problem for mobile manipulators. Accidents like tipover of cranes or loaders is the usual headlines of local newspapers, and shows that tipover even threatens on-board operator type robots.

Stability against tipover for a big and heavy vehicle (compared to its manipulator) is not an important issue. However, nowadays industry and technology requires small size and lightweight mobile manipulators for less energy consumption, better maneuverability, and capability of working in limited work-spaces. Tipover stability indeed is a problem for these types of mobile robots. The manipulator, which is usually long compared to the vehicle dimensions, makes the robot more unstable. Hence, for a complete analysis, full dynamics interaction between a moving vehicle and its manipulating arm must be taken into account [1]. The importance of inertial loadings and their effects on machines instability is shown in [2]. There is a rich literature on the control of mobile manipulators and tracking problem considering disturbances and dynamic interaction between vehicle and manipulator (for example see [3–6], to name a few). However, the problem of stability against tipover due to dynamic interaction between the robot arm and the moving vehicle has gotten less attention so far. The few existing studies have focused on the obstacle avoidance and path planning of wheeled robots or legged systems in special cases.

Dubowsky and Vance [7] proposed a motion-planning algorithm for 'standing' mobile manipulators. In their paper, a mobile manipulator is a robot without a solid attachment to the floor and is standing on its outriggers (like a crane during operation). They used a numeric instability inequality to plan the optimum path that satisfies minimum time and avoids tipover. In [8] Papadopoulos and Rey defined "a new measure of tipover stability margin" based on the geometry of vehicle and the direction of resultant force on the mobile manipulator. Their method works very efficient for off-line path planning (i.e., when the path is predefined). We will discuss more about the existing tipover criteria in Section 3.

Nowadays, neural network is a strong tool for replacing complex nonlinear dynamics with a simple numeric database (for a comprehensive literature review the reader is referred to [9]). Neural network is used in different parts of a control procedure as an observer, controller and even an intelligent learner from the input data during performing task ([10–12]).

In this paper, we aim to develop a procedure for real-time tipover estimation in mobile manipulators. In Section 2, without loss of generality, we discuss a three degrees of freedom manipulator installed on a three-wheel vehicle. In Section 3, we review existing tipover stability criteria and propose our new formulation (ZETUF). ZETUF is a "tipover stability" criteria as well as "streerability stability" criteria. We continue discussing how we can solve the real-time estimation with this criterion and we end up using a neural-network observer to reduce the time needed for calculation. Section 4 is a brief introduction to neural networks and Section 5 is devoted to a numerical simulation of the procedure. We use two-dimensional slices of six-dimensional hyper-volume stability to present our results. The proposed tipover stability criteria and the neural-network observer, together, have made an efficient tool for our real-time estimation problem. The algorithm is almost independent of complexity of mobile manipulator and is an efficient method in real applications.

2. Dynamic modeling of a mobile manipulator

The dynamic equations describing the motion of a mobile manipulator can be formulated in any convenient manner. Equations of motion are written for a manipulator (a collection of arms and joints) during operation, which is carried by a moving vehicle. Therefore, all dynamical interactions between manipulator and vehicle have taken into account. Without loss of generality we consider a three degrees-of-freedom (3 DOF) manipulator placed on a three-wheeled vehicle (see Fig. 1). Two rear wheels are co-axial and there is a single wheel (castor) at the front.

We use Luh et al. method [13], which is derived directly from Newton's law, to relate the motion of the chain of arms and joints together, and derive a global equation for the manipulator. The selection of coordinates is based on the so-called Denavit–Hartenberg method [14] that requires four coordinates for a three DOF mechanical arm (see Fig. 2).

The Denavit–Hartenberg parameters are given in Table 1.

We use external iterative relations and then internal iterative relations (as explained in [15]) to find the resultant force and torque applied to the vehicle from the manipulator. These resultant force and torque are functions of joint angles, joint angular velocities, joint angular accelerations, vehicle accelerations, vehicle angular velocities and angular accelerations. Therefore we replace the effect of the manipulator with one force and one torque; each of them has three components. The External iterative relations and internal iterative relations are written for three links in

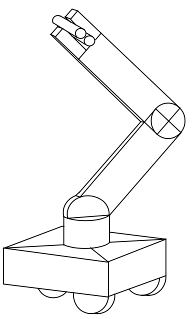

Fig. 1. Three degrees of freedom manipulator placed on a moving vehicle.

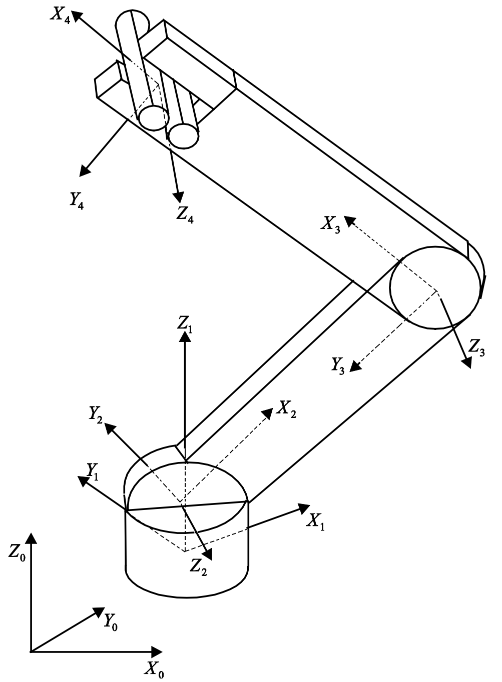

Fig. 2. Coordinates attached to manipulator joints.

Table 1
Denavit–Hartenberg parameters

I	a_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	$h/2$	θ_1
2	90	0	0	θ_2
3	0	$L1$	0	θ_3
4	0	$L2$	0	0

order to find the complete dynamics of the manipulator. Fig. 3 shows the free body diagram of the vehicle.

The manipulator is positioned above the center of gravity of the vehicle center of gravity, i.e., if the vehicle stands on a horizontal plane, the extension of gravity acceleration vector passes through the manipulator hinge. Fig. 4 shows notations use to describe the dynamics of the vehicle.

The following relations exist between manipulator's resultant force and moment and tire forces

Fz:

$$Fz_{w1} + Fz_{w2} + Fz_{w3} + fz - Mg = Ma_z \tag{1}$$

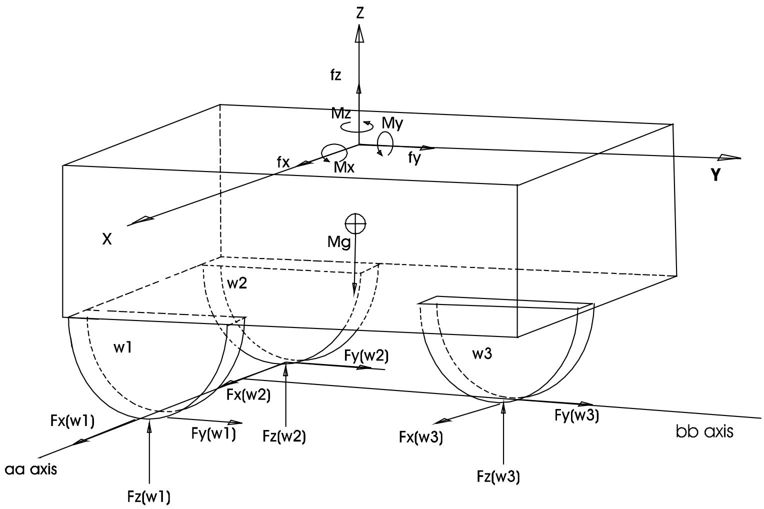

Fig. 3. Freebody diagram of the vehicle.

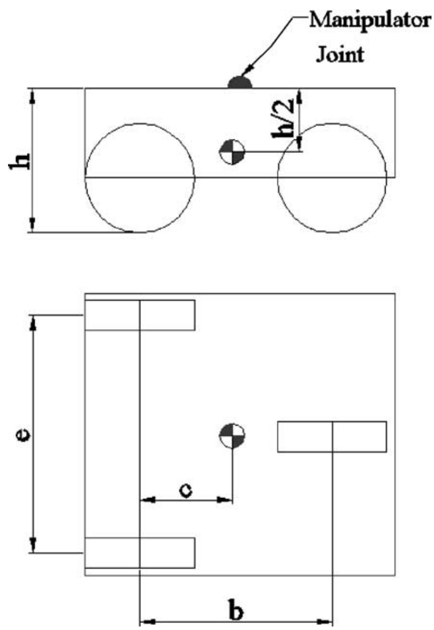

Fig. 4. Dimensions of the vehicle.

$M(aa)$:

$$Fz_{w3}b - M(g + a_z)c + \frac{1}{2}Ma_yh + f_zc - fyh + Mx = 0 \tag{2}$$

$M(bb)$:

$$\frac{1}{2}(Fz_{w2} - Fz_{w1})e + fxh + My - \frac{1}{2}Ma_xh = 0 \tag{3}$$

Given the resultant force and moment of the manipulator, these equations can be used to derive forces acting on the road by tires.

3. Tipover criterion

In this section we briefly review the existing criteria for tipover stability and propose our formulation. Different tipover stability criteria present a unique concept, which is the threshold of instability, in different languages. Therefore, we can interpret them as different formulations for one single fact. Force–angle criterion formulated by Rey et al. [16] states that the tipover occurs when the extension of the resultant force vector exceeds the area encompassed by the straight lines connecting adjacent tires. In a series of papers, Huagne et al. have used ZMP (zero moment point) as the tipover stability criteria. ZMP states that when tipover occurs, the point of zero moment is outside the projection of the vehicle image on the road (see [17–20]). In these papers, they used ZMP to satisfy different objective functions in a mobile manipulator path planning problem. Their approach is an off-line path planning method, which is based on this fact that the path of the mobile manipulator is known in advanced. The energy criterion is—as always—a formulation from a different point of view. Energy criterion says tipover occurs if the energy injected to the system reaches to the energy needed for the tipover at that configuration [2].

Having all these three criteria in mind, we propose another formulation based on the tire upward forces that not only measures tipover stability but also can be used as a measure for steerability stability. It has a more sensible physical interpretation and more direct effect on the vehicle situation. ZETUF (zero tire upward force) states that, for a 4 wheel vehicle with a convex geometry, tipover occurs if the upward forces exerting from the road to two adjacent tires of the vehicle reach zero. In this case, the mobile manipulator can tip on the other two tires; it is indeed the threshold of instability. ZETUF for three wheel vehicles requires only one of the upward forces to reach zero and it can be easily extended to vehicles with more than four wheels. Once we measure tire upward force for determining tipover stability, we can verify if vertical forces on tires are high enough for the traction needed to sustain steerability at that moment. This would show if the mobile manipulator is in the steerability stability zone.

When we tried to put this criterion to action, we understood that measuring force in real application is expensive and not accurate. Force sensors are usually sensitive to temperature and other environmental conditions. If we want to measure those environmental parameters and try to correct the output of the force sensors, it would become a costly collection of instrument for a robust and multi-purpose mobile manipulator. We eventually ended up using an observer by measuring easier-to-measure quantities like angles and velocities and determine the tire forces by solving mobile manipulators dynamical equations. Just by a portion of force-sensor price, we now have an observer combination with a very high accuracy.

The main idea is simple, to measure all angles, angular velocities and angular accelerations of joints and the vehicle (for the vehicle, a single gyro-accelerometer does provide us all required data), and put them in dynamic equations of mobile manipulator and solve for the tire upward force. These forces are simply solutions of Eqs. (1)–(3).

Since our main problem was "real-time" estimation of the stability of a mobile manipulator, we realized solving the set of algebraic equation even for a very simple configuration takes a huge amount of time. This calculation time increases exponentially when the degrees of freedom of the manipulator increase. We proposed to use a neural-network database instead of solving dynamic equations. To this end, we need to solve dynamic equations for all possible cases and save them to a numeric database. A neural network then, can be trained to interpolate (and extrapolate) data. Eventually this observer is attached to the autonomous mobile manipulator for an on-line tipover estimation. By increasing the number of degrees of freedom of the manipulator, the initial (off-line) training time of neural-network increases, but the time needed for getting data while working in the field is almost the same.

4. Neural-network theory

Artificial neural networks are in fact a kind of black box. The simplest model of a neural network is shown in Fig. 5: a number of input units are connected to output units. Every connection has a weight; each output neuron has a threshold value. When patterns, combinations of activated and nonactivated input neurons, are fed to the input, these activities are fed through the weight function to each of the output neurons. When these weighted activities are summed in each output neuron and the threshold value is exceeded, then the output node emits a signal. Many configurations are possible for this sequence. A much-used model has an extra layer of

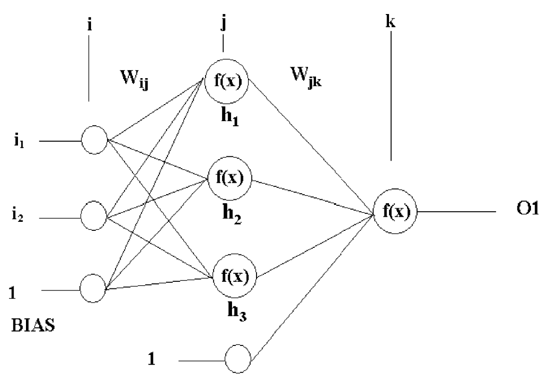

Fig. 5. Schematic representation of a three-layer artificial neural network.

neurons, called a hidden layer, between the input and output layers. The weights can be adapted automatically with different algorithms, the threshold function can have many shapes and the connection pattern of the neurons can be varied to a great extent. Each layer also has one bias input, as shown in Fig. 5, to accommodate nonzero offsets in the data. The observer data obtained from a numerical solution of dynamic equations were processed by NN which was trained with the back-propagation of errors learning algorithm. The structure of the network is comprised of three node layers: an input, a hidden and an output layer, represented by *i*, *h* and *o*. These symbols, respectively, indicate the number of nodes in the input layer, hidden layer and output layer. The input nodes transferred the weighted input signal to the nodes in the hidden layer, as same as from the hidden node to the output layer. A connection between the nodes of different layers was represented by a weight w_{ji}. During the training process, the correction of weight Δw_{ji} was defined as follows:

$$\Delta w_{ji}(n+1) = \eta\delta_j o_i + \alpha\Delta w_{ji}(n) \tag{4}$$

where δ_j is the error term, o_j is the output of node j, η is the learning rate, α is the momentum and n is the iteration number. The iteration would be finished when the error of prediction reached a user-defined minimum.

A nonlinear transformation and a suitable function was applied between the input and output of each node. The optimum values of η and α were calculated such that minimize the prediction error.

In the backpropagation method input vectors and the corresponding output vectors are used to train a network until it can approximate a function, to associate input vectors with specific output vectors. Networks with biases, a sigmoid layer, and a linear output layer are capable of approximating any function with a finite number of discontinuities [21]. Standard backpropagation method is a gradient descent algorithm. *Backpropagation* refers to the manner in which the gradient is computed for nonlinear multilayer networks. Properly trained backpropagation networks tend to give reasonable answers although the same inputs were not in the learning list. Typically, a new input will lead to an output similar to the correct output for input vectors used in training with the minimum defined error. This generalization property makes it possible to train a network on a representative set of input/target pairs and get good results without training the network on all possible input/output pairs.

The method of training used in this article is a modified backpropagation method, called Levenberg–Marquardt (LM) algorithm which is faster than other algorithms by a factor of 10–100 [22]. The main drawback of the Levenberg–Marquardt algorithm is its requirement to store many matrices, which can be quite large for certain problems.

5. An example

In this section we show how a neural-network observer can be used as a tipover estimator for real-time applications. We first determine the range of variation of variables, for example, the maximum and minimum of the speed of each joint. Then we solve the algebraic dynamical equation for a sufficient number of data-points. Sufficiency of number of points is determined by convergence of the neural network; if neural network converges to the desired efficiency, the number of selected points is enough and the way of selection is suitable, if not, either or both must be revised. Then the collection of points is used as an input, and neural-network structure is trained to estimate the output corresponded with each set of input data. After training process, it is usually important to compare the output of trained system with actual data; this is the validation step. Now mobile manipulator can be equipped by this compact fast observer. We also need to define a suitable stability norm that would be a function of the road condition, inclination of the road, possible noise in the path and the work zone, estimated error in the measuring system, training drift, and external unpredictable disturbances. Finally a control algorithm installed with this collection of data (which is in the form of a database) on the mobile manipulator can estimate stability situation and performs suitable actions to prevent tip-over (see [23] for example).

We assume in this example that the mobile manipulator moves with constant velocity in *Y*-direction. Note that *Y*-direction is a component of body coordinate system and moving with a constant velocity in that direction does not mean that the robot would not change its direction of motion. We specifically assume that the robot moves in a circular path and is subjected to radial forces. Specification of our mobile manipulator is listed in Table 2.

As we discussed earlier, in order to set up a comprehensive database, we need to specify range of variation of different variables. Table 3 shows this range for this example.

The next step is to solve the system of algebraic equations and make the numeric database. Here we choose zero tire upward force as the stability criteria. If it is of interest to satisfy the steerability stability as well, a positive number instead of zero must be selected, which is the minimum required tire vertical force for steerability. For our mobile manipulator with a constant velocity in a circular path (in real application in a rotary), that allows six parameters to vary, the stability region is a six-dimensional hyper-volume. We consider the surface of this hyper-volume as the

Table 2
Vehicle and manipulator parameters

Vehicle mass	10 kg
Arm 1 mass	1 kg distributed in arm
Arm 2 mass	7 kg distributed in arm
$L1$	300 mm
$L2$	400 mm
b	400 mm
c	150 mm
e	300 mm
h	200 mm
Vehicle velocity	10 m/s
Vehicle acceleration	4 m/s^2

Table 3
Variables range

Variable	Range
Θ_1	225–275°
Ω_1	−1 rad/s to 1 rad/s
α_1	-1 rad/s^2 to 1 rad/s^2
Θ_2	50–90°
ω_2	−1 rad/s to 1 rad/s
α_2	-1 rad/s^2 to 1 rad/s^2

six-dimensional danger hyper-surface. Note that this hyper-volume is not necessarily simply-connected. A delicate method to show the stability region is to select slices of the stability volume. This is a common method in different branches of science, like stability of multi-dimensional stellar systems. Figs. 6 and 7 show two typical slices of stability volume. Two- and three-dimensional stability slices can be used as a guide for graphical path planning and give a good feeling about behavior of mobile manipulator in different range of variables.

Among different ways of setting up our primary database, we choose constant step variation. Although this method works very good and accurate for a systems with a few number of variables, for several degrees of freedom the overwhelming volume of data-points prevent processor to converge (due to RAM limitations also). For several degrees of freedom, the method of random selection is recommended. Increment steps for this example are listed in Table 4.

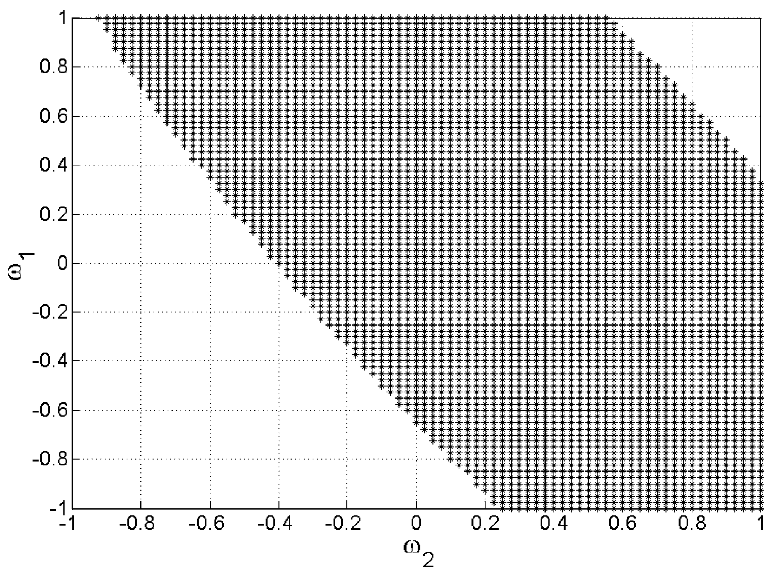

Fig. 6. Two-dimensional slice of six-dimensional stable region when $\theta_1 = 70°$, $\theta_2 = 60°$, α_1 and $\alpha_2 = 0$ rad/s^2.

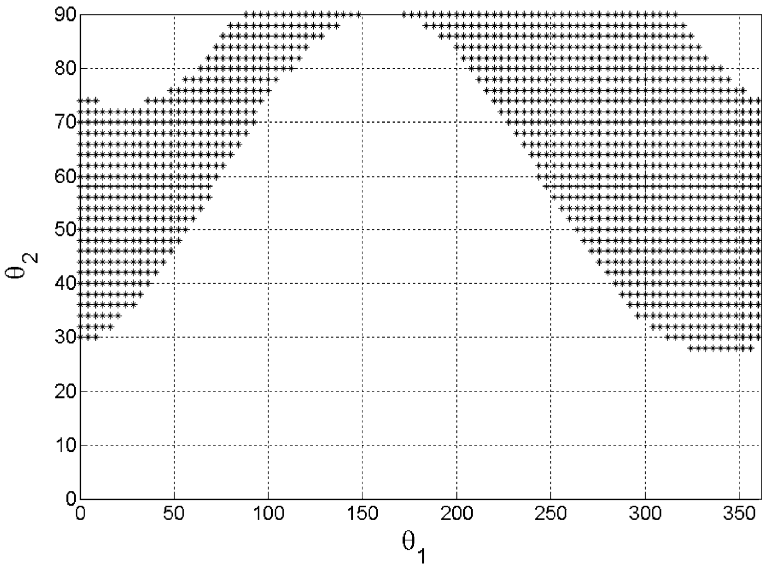

Fig. 7. Two-dimensional slice of six-dimensional stable region when ω_1, $\omega_2 = 1$ rad/s, $\alpha_1 = 0$ rad/s^2 and $\alpha_2 = -1$ rad/s^2.

Table 4
Variable increment for learning

Variable	Increment
Θ_1	8°
θ_2	8°
ω_1	0.5 rad/s
ω_2	0.5 rad/s
α_1	0.5 rad/s^2
α_2	0.5 rad/s^2

The database consists of 26,250 data-point, each of them consists of six inputs and three outputs (reaction forces on the tires). For training, the Levenberg–Marquardt (LM) algorithm has been used. The training has been done with MATLAB 6.1.

Network type: Feed-forward back-propagation.
Adaption learning function: Gradient Descent With Momentum (Learngdm).
Performance function: MSE (Mean Square Error).
Number of layers: 2.
Layer 1: 10 neurons, Transfer function Tansig.
Layer 2: 3 neurons, Transfer function Purelin.
Epochs: 600.
Final Performance: 3.

As we emphasized, testing the final neural-network database is a necessary part of this approach. We choose different slices from stability hyper-volume and compare it with original numeric data. Fig. 8 is an example of this type of comparison that shows at most six degrees error in the estimating of angular location. In Fig. 8 each black square indicates one degree error in angular location. This error can be decreased by increasing the efficiency of the neural network during training.

It is claimed that the proposed method, using neural network as an observer, decreases the volume of calculation resulting in real-time control of complex mobile manipulators. We will show here, how helpful it can be, even for a very simple mobile manipulator.

Consider again the previous setup: a mobile manipulator that its vehicle moves with a constant velocity in a circular path. Assume the angular acceleration of manipulator joints are given by

$$\alpha_1 = \sin(2t)$$
$$\alpha_2 = -0.1\sin(2t)$$

and initial conditions are (for manipulator joints)

$$\theta_1(t=0) = 260^\circ$$
$$\theta_2(t=0) = 55^\circ$$
$$\omega_1(t=0) = -0.6\ \text{rad/s}$$
$$\omega_2(t=0) = 0.1\ \text{rad/s}$$

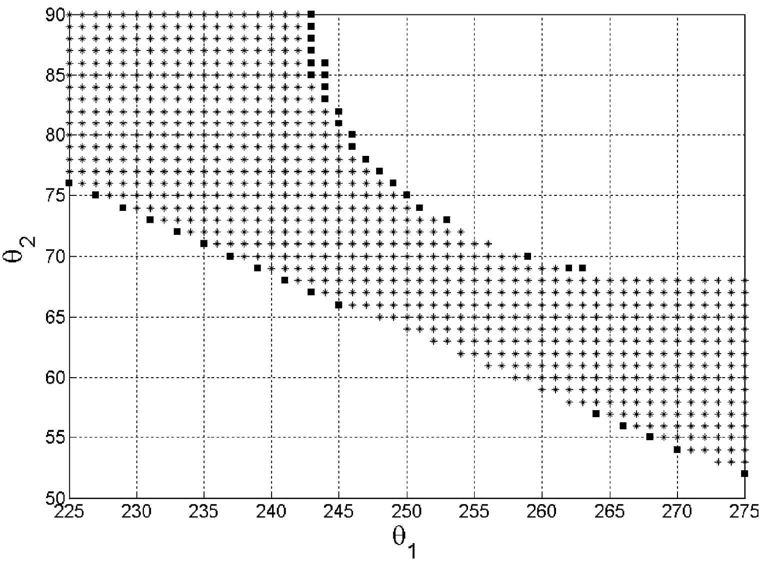

Fig. 8. The neural-network output comparison with actual data (each square is a 1-degree error).

We consider the motion of the mobile manipulator during 5.5 s. We use Euler solver for numerical simulation with the time step equal to 1/100 s. We want our algorithm to show the tipover situation during the motion. Fig. 9 shows the image of the path of the mobile manipulator on the θ_1–θ_2 plane. The thicker part of the graph shows the unstable region during which the tire upward force is less than zero. It is indeed not a physical possible situation, because it means that the road "pulls" tire(s). In real application the distance from threshold of tipover stability and the speed of decreasing of this distance is important and must be considered to design a suitable controller (see for example [23]). In the simplest case we can define the minimum of upward forces acting on tires as the tipover norm. This norm is shown in Fig. 10. It is to be noted that this graph can be discontinuous: for a while one tire gets the minimum value and after that another tire may get it.

The time needed for calculation is (using 750 MHz CPU)

Neural network	Direct solving
3.9 s	10.2 s

This example shows that using neural network as an observer would reduce the time of calculation by a factor of 2.6. For 5.5 s simulation indeed a 750 MHz CPU cannot estimate the stability situation online while the proposed algorithm enable it to do that easily. Although the number of seconds decreases with faster machines, the time ratio of these two way of calculations is almost the same. This ratio is a function of degrees of freedom of mobile manipulator and the number of free

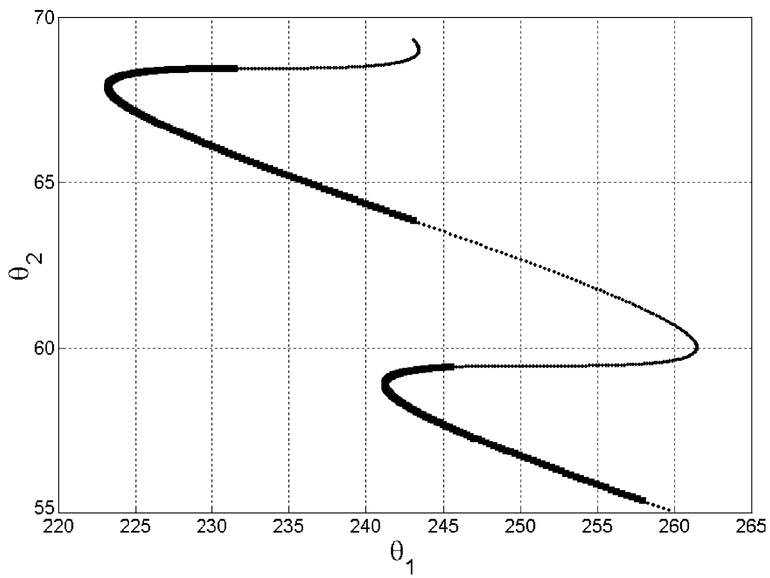

Fig. 9. Image of path in θ_1–θ_2 plane.

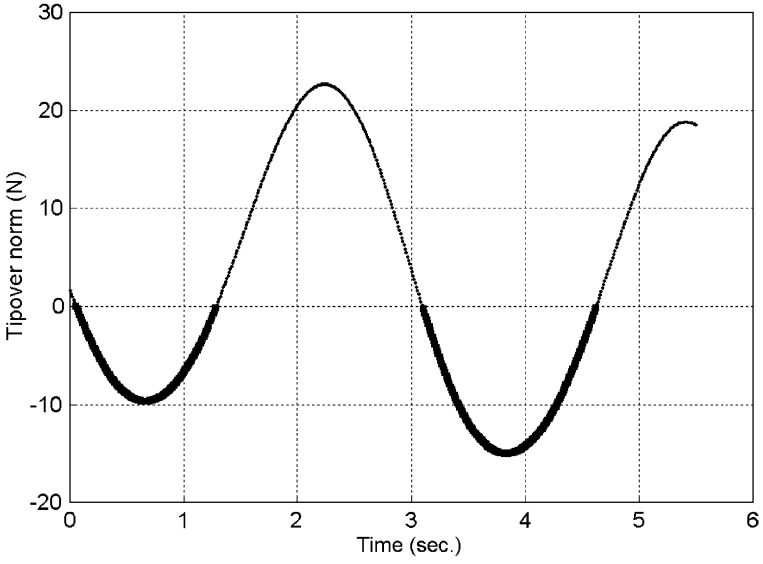

Fig. 10. Tipover norm versus time.

parameters exist in the model. Higher degree of freedom mobile manipulator results in a higher coefficient. We again emphasize that the importance of this algorithm emerges for a complicated mobile manipulator. For such robots, although the learning time increases exponentially, the time needed for taking data from our observer is almost the same. Optimizing the type and shape of the neural network may also modify the time ratio.

6. Conclusion

Tipover is an important danger that menaces mobile manipulators, especially autonomous ones. In these types of robots, the dynamic interaction between manipulator and vehicle plays an important role in the determination of the stability. In this paper we first compare different stability criteria and propose a new formulation that includes both stability and steerability. Our criteria—named ZETUF—is based on tires upward forces. While applying our algorithm, we found direct measurement of force an expensive and inaccurate task. Therefore we proposed to replace tire forces data by an observer. This observer solves equations of motion of the mobile manipulator, and also computes tire forces based on the measured data. The next problem is to deal with the huge volume of computation which is problematic in our real-time estimation objective. To overcome this problem, we use another neural-network database. Finally, by a numerical simulation we showed that this neural-network observer reduces the time of computation considerably. The method is applicable to a wide range of mobile manipulators such as those classified in [24].

References

[1] Yammamoto Y, Yun X. Effect of the dynamic interaction on coordinated control of mobile manipulator. IEEE Trans Robot Automat 1996;12(5):816–24.
[2] Ghasempoor A, Sepehri, N. A measure of machine stability for moving base manipulators. In: Proceedings of the 1995 IEEE international conference on robotics and automation.
[3] Dubowsky S, Tanner AB. A study of the dynamics and control of mobile manipulators subjected to vehicle disturbances. In: Proceedings of the IV international symposium on robotics research, Santa Cruz, CA, 1987.
[4] Hootsmans NAM, Dubowsky S. The control of mobile manipulators including vehicle dynamic characteristics. In: Proceedings of the IV topical meeting on robotics and remote systems, Albuquerque, NM, February 24–28, 1991.
[5] Meghdari A, Durali M, Naderi D. Dynamic interaction between the manipulator and vehicle of a mobile manipulator. J Intell Robot Syst 2000;28(3):277–90.
[6] Naderi D, Meghdari A, Durali M. Dynamic modeling and analysis of a two DOF mobile manipulator. Robotica Int J 2001;19:177–85.
[7] Dubowsky S, Vance EE. Planning mobile manipulator motions considering vehicle dynamic stability constraints. In: Proceedings of the 1989 IEEE international conference on robotics and automation, Scottsdale, AZ, May 14–19, 1989.
[8] Papadopolous E, Rey D. A new measure of tipover stability margin for mobile manipulators. In: IEEE international conference on robotics and automation, 1996. p. 3111–6.
[9] Sheng L, Goldenberg AA. Neural-network control of mobile manipulators. IEEE Trans Neural Networks 2001;12(5).
[10] Jin Y et al. Stable manipulator trajectory control using neural networks. In: Omid X, editor. Neural systems for robotics. New York: Academic; 1997. p. 117–51.
[11] Fierro R, Lewis FL. Control of a nonholonomic mobile robot using neural networks. Proc IEEE Trans Robot Automat 1998;9:589–600.
[12] Lewis FL, Jagannathan S, Yesildirek A. Neural network control of robot manipulators and nonlinear systems. London, UK: Taylor and Francis; 1999.
[13] Luh JYS, Walker MW, Paul RP. On-line computational scheme for mechanical manipulator. Trans ASME J Dynam Syst Measure Control 1980.
[14] Denavit J, Hartenberg RS. A kinematic notation for lower-pair mechanisms based on matrices. J Appl Mech 1955:215–21.
[15] Craig JJ. Introduction to robotics: mechanics and control. 2nd ed. Addison-Wesley; 1989.
[16] Ray DA, Papadopoulos EG. On-line automatic tipover preventation for mobile manipulators. IEEE 1997.
[17] Huang Q, Sugano S. Manipulator motion planning for stabilizing a mobile manipulator. Intelligent robots and systems '95. 'Human robot interaction and cooperative robots'. In: Proceedings of the 1995 IEEE/RSJ international conference, vol. 3, 1995. p. 467–72.
[18] Huang Q, Sugano S. Motion planning of stabilization and cooperation of a mobile manipulator. Intelligent robots and systems '96, IROS '96. In: Proceedings of the 1996 IEEE/RSJ international conference, vol. 2, 1996. p. 568–75.
[19] Huang Q, Sugano S, Tanie K. Stability compensation of a mobile manipulator by manipulator motion: feasibility and planning. Intelligent robots and systems, 1997, IROS '97. In: Proceedings of the 1997 IEEE/RSJ international conference, vol. 3, 1997. p. 1285–92.
[20] Huang Q, Sugano S, Tanie K. Motion planning for a mobile manipulator considering stability and task constraint robotics and automation, 1998. In: Proceedings of the 1998 IEEE international conference, vol. 3, 1998. p. 2192–8.
[21] Demuth H, Beale M. Neural network toolbox for use with MATLAB. MATLAB 5.3.1 on line manual.
[22] Hagan MT, Menhaj M. Training feedforward networks with the Marquardt algorithm. IEEE Trans Neural Networks 1994;5(6):989–93.

[23] Meghdari A, Naderi D, Alam MR. Tipover stability estimation for autonomous mobile manipulators using neural networks. In: CD-Rom proceedings of 2004 Japan–USA symposium on flexible automation (JUSFA), Denver CO, USA.
[24] Campion G, Bastin G, D'Andrea-Novel B. Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. In: 1993 IEEE International conference on robotics and automation, Atlanta, GA, vol. 1, May 1993. p. 462–9.