Outdoor Navigation Strategy in Hazardous Environment

Pierre Pomiers, Antonella Semerano, Vincent Dupourqué, Meftah Ghrissi
Robosoft S.A.
Technopole d'Izarbel, 64200 Bidart, France.
Tel.: +33-5-59 41 53 60; Fax.: +33-5-59 41 53 79
E-mail: pierre@robosoft.fr

ABSTRACT
This article presents an approach for the control of mobile robots in difficult navigation conditions. After a general survey on the on-board operating system, ALBATROS^{TM}, trajectory tracking problems are invoked and discussed in relation with a specific application: the robotization of a 25 tons standard VOLVO wheel loader.

1. FOREWORD
This paper falls into four main sections. In section 2, the real time operating system ALBATROS^{TM} is introduced. The wheel loader application is discussed in section 3. Section 4 and 5 give a sight on a trajectory following approach dedicated to this mobile platform performing evolutions on uneven ground: a fuzzy logic based control supervised using a sliding manifold based procedure. Very a priori benefits and drawbacks about this approach are discussed and, finally, conclusions are drawn in section 6.

2. THE REAL TIME OPERATING SYSTEM
When an existing vehicle needs to be "robotized", some of its features must be improved in order to allow its automatic control. For instance, internal sensors, such as, position encoders on propulsive wheels, or gyro, as well as ultrasonic range-finders for obstacle detection and collision avoidance must be added to the system. At the software level, it means a specific interface modules, considering the number of actuators and sensors. When using ALBATROS, the software of the whole system is divided into two separate sets of programs.

System Programs: which is the resident part of ALBATROS generally delivered. This part is made of basic primitives allowing the users to access the lower levels of the system.

Application Programs: which depend on the application and are generally developed by the user in a high-level language on a host machine. The application programs communicate with system programs by sending parameterised commands.

3. THE WHEEL LOADER CONTROL APPLICATION
As part of the Anemona Esprit Project 27841, aiming to show vehicle autonomy capabilities in hazardous environment, a 25 tons standard Volvo wheel loader, provided with sensors needed for navigation and obstacle detection, is used as a platform for demonstration. See Fig. 1. The vehicle control system allows vehicle path trajectories to be generated and controlled from a remote control station. The platform has been tested in full scale outdoor real building site near Göteborg, Sweden.

C3.1 ***APPLICATION STRUCTURE***

Several sensors equip the L180C to retrieve information about its own kinematical state (position, steering and velocity). In order to deal with this data stream, the vehicle embeds a 68040 based computer. This part of the system is in charge of the basic operations like driving actuators, reading sensors and also communication tasks. All of the high level routines (like trajectory tracking, learning and human machine interface) are performed on an off-board Linux computer. This machine communicates with the on-board computer, via a wireless Ethernet link, to get information needed for the L180C remote control, such as tools states, velocity and position on the building site map. This part of the application is highly user oriented.

Figure 1 :The robotized wheel loader

C3.2 ***HARDWARE DEVICES***

Actuators: By default, on the machine, most of the devices are electrically commandable. So it is possible to interface directly the L180C main electric board with the calculator via an analog card. For the rest of commands (like forward and backward movement, bucket control), adaptations of additional actuators are needed.

Sensors: An encoder, directly linked to the transmission axle, returns wheel spin velocity. With regards to the different jack position, information is given by linear encoders fixed in parallel with the jacks themselves. Each sensor analog output is plugged into the calculator to handle the analog/digital conversions.

The Position Determination System : The PDS[1] is an embedded device that provides the estimation of the machine position in a three dimensional environment. It returns a set of six measurements: abscissa, ordinate, Z-coordinate and rotation angles around each of the three axis. This device (in fact, a laser transmitter-receiver) is fixed on the top of the machine. Some special reflecting targets are placed around the building site.

---

[1] developped by ARNEX Company

4. THE NON-LINEAR CONTROL STRATEGY: A FUZZY LOGIC BASED APPROACH [4]

In our particular case, the mathematical models the system is based on, do not recover from system uncertainties and, most of the time, can not integrate enough deep knowledge of the environment where the system is placed to provide a good navigation quality.

C4.1 BASIC KINEMATICAL ANALYSIS

A preliminary study of the L180C kinematics is needed to make use of an appropriate controller. The L180C motion is a composition of a rotation (for the steering) and a linear motion (for forward and backward movement). That is what is depicted on Fig. 2. On this figure, it is shown that the variation of the steering angle implies a modification of the radius of the turning circle whose center is called C. So we assume that the L180C motion behaviour can be assimilated to an unicycle-type mobile robot[2] one.

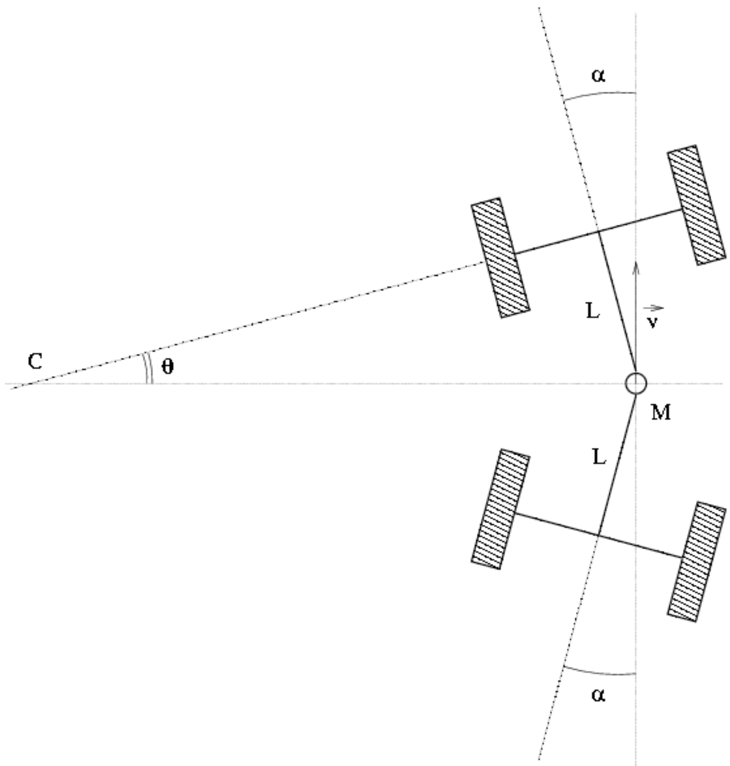

Figure 2: The kinematical scheme of the Volvo L180C.

C4.2 TRAJECTORY FOLLOWING

Let us refer to Fig. 3 to illustrate the trajectory following approach. Let M be the current state of the machine and M' the desired one. Moreover, let Γ be the real trajectory the wheel loader is following and Γ_d be the trajectory it is expected to follow. In fact, those trajectories are two series of time ordered states.

[2] a mobile robot with two actuated wheels placed on a common axle

The vehicle is controlled by an acceleration and steering velocity couple of commands that minimize the error between the current state and the desired one. Indeed, this command couple aims to minimize a set of three errors (i.e.: error in position, orientation and velocity).

C4.3 PRELIMIRARY CONSIDERATIONS ABOUT THE CONTROLLER

In order to compute its output command, the controller in charge of the wheel loader motion control takes as input the couple ($\alpha(t)$, $d(t)$) which represents the angular error and the distance error between the two point M and M', and also the couple ($\Delta\theta(t)$, $\Delta v(t)$) which represents the relative orientation error and velocity error between the real state and the desired state. This input vector has the following form:

$$err(t) = (\alpha(t), d(t), \Delta\theta(t), \Delta v(t))$$

A problem still subsists. We have to deal with very particular navigation conditions. Indeed, on such a building site landslides and skids forbid exact modelisation of the wheel loader motion. Hence, we have to integrate "uncertainty management" to well cope with these conditions.

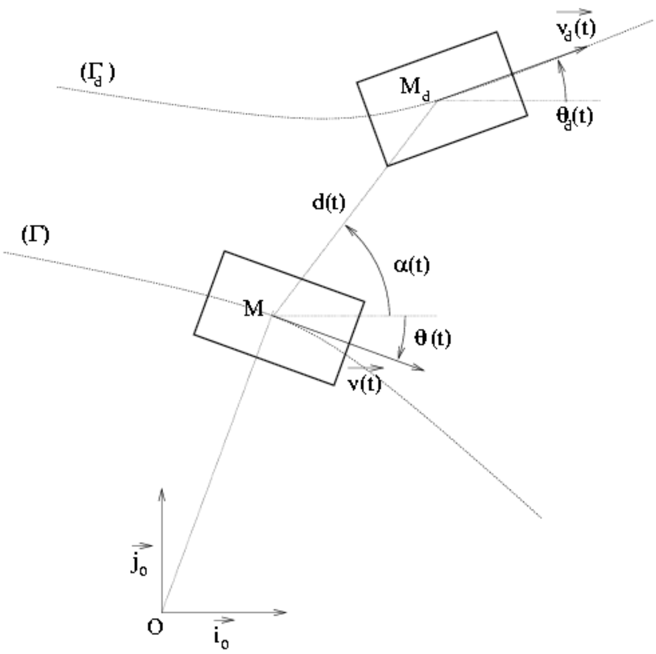

Figure 3: Current state M en desired state M_d of the vehicle at time t.

C4.4 THE FUZZY CONTROLLER SYNOPSIS

The fuzzy controller gets its input from the PDS server process. From those position estimations, it proceeds three steps using a given knowledge base: see Fig. 4.

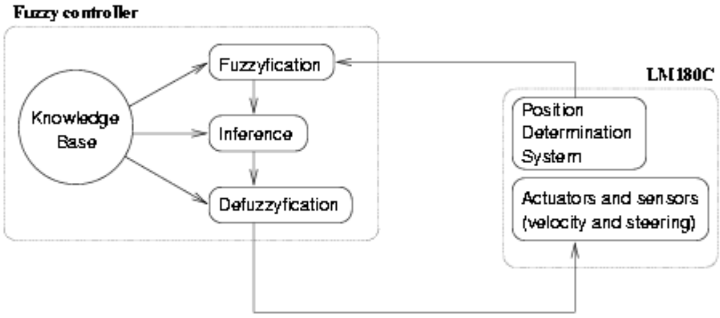

Figure 4: The fuzzy controller synopsis.

The interest of such an approach is twofold:

1. fuzzy sets and fuzzy inference permit approximate reasoning, which is particularly appropriate to the case of the wheel loader navigation conditions.
2. complete model of the controlled process is not necessary, a description of the way how it works in human like terms is used instead.

C4.5 THE FUZZY CONTROLLER IMPLEMENTATION

Let us illustrate the way the fuzzy controller works. In our application, we use three fuzzy sets labelled N, Z and P (respectively for Negative, Zero and Positive). Those sets are associated with $(d(t), \Delta v(t))$ for acceleration command computation and with $(\alpha(t)\}, \Delta\theta(t))$ for steering command computation. They are represented by trapezoid regions in the measurements versus member ship degree space. See Fig. 6

Acceleration command

$\Delta v(t)$ \ $d(t)$	N	Z	P
N	P	P	Z
Z	P	Z	N
P	Z	N	N

Steering command

$\Delta\theta(t)$ \ $\alpha(t)$	N	Z	P
N	Z	P	P
Z	N	Z	P
P	N	N	Z

Figure 5: The rules matrix.

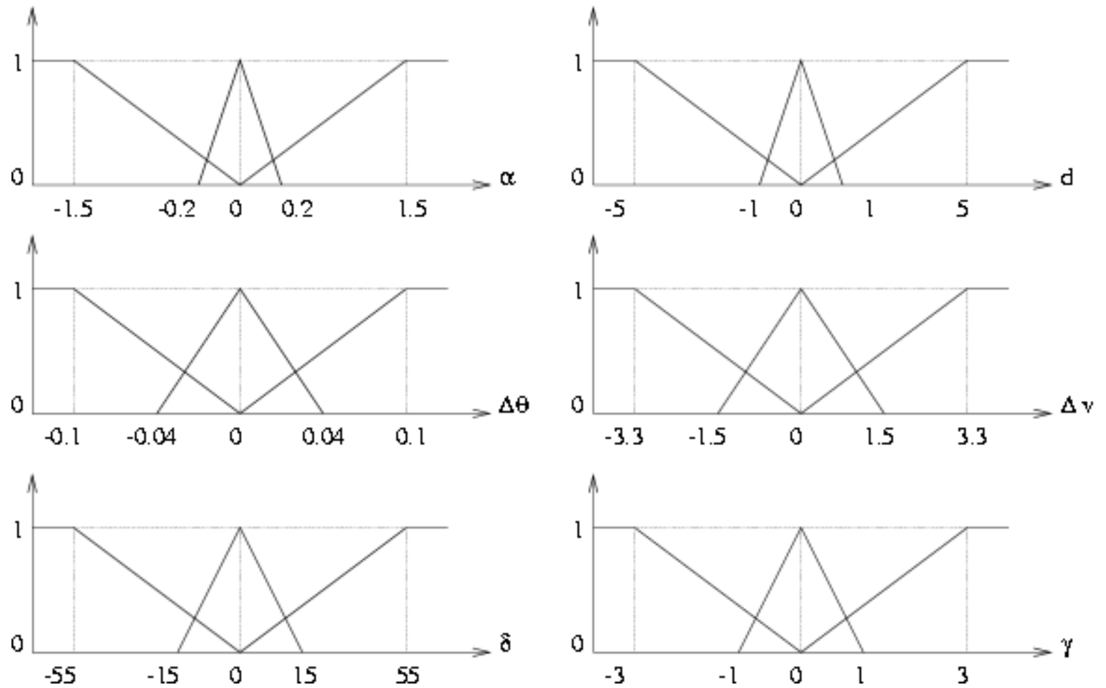

Figure 6: The fuzzy sets used for the wheel loader application. α is expressed in (rad), Δθ in (rad.s^{-1}), d in (m), Δv in (m.s^{-1}), the steering command δ in (rad) and the acceleration command γ in (m.s^{-1}).

The fuzzyfication step makes use of those fuzzy sets. Given a variable value the fuzzyfication module yields a (fuzzy set label, membership degree) couple. In our case fuzzyfication returns fuzzy values of d(t), Δv(t), α(t) and Δθ(t). In a second step, the inference engine gets the fuzzyfied sets corresponding to (d(t), Δv(t)) and α(t), Δθ(t)) couples. Then, by crossing their fuzzy values, the activation degree of each rule from the rules matrix (see Fig. 5) is computed using an "and" logic to combine the input variables.

In a third step, the activation degree of a given rule is then used for computing the result of the fuzzy inference for the rule considered. This result is a new fuzzy set. Finally, given all the fuzzy sets inferred by a given output variable (i.e.: acceleration or steering), the defuzzyfication module determines the actual command by using a method called "barycenter of the center of area".

C4.6 ***EXPERIMENTAL RESULTS***

For the following experiment, the vehicle velocity was fixed to a constant value. By this way we focused on the steering command as it is the command the most dependent on the ground conditions. In order to evaluate this first approach, the control scheme has been applied to the wheel loader in real environment. We ordered the machine to follow two different typical paths. Both paths were only designed by giving control point (i.e.: machine position and orientation). They are depicted as polygons in the following figures. The realised trajectories are shown on Fig. 7 and Fig. 8 and the computed commands are given on Fig. 9 and Fig. 10.

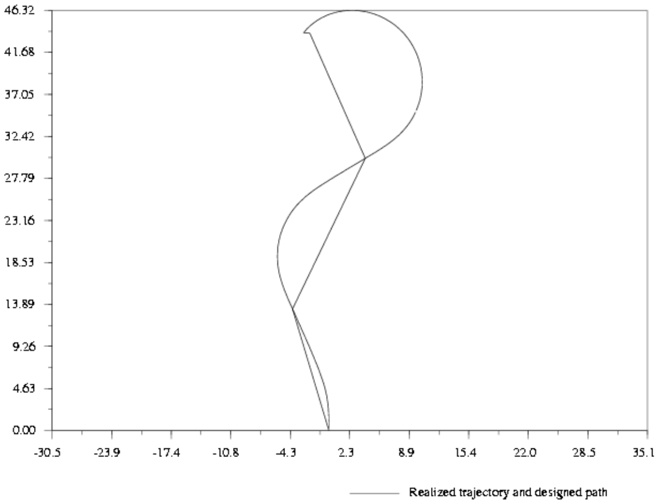

Figure 7: The first trajectory test.

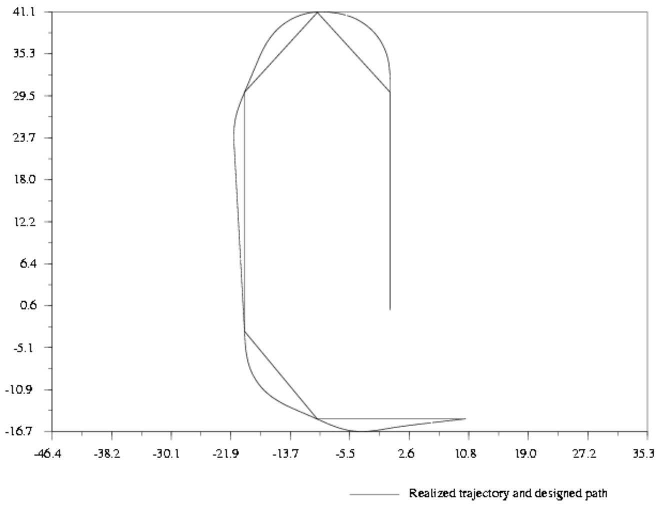

Figure 8: The second trajectory test.

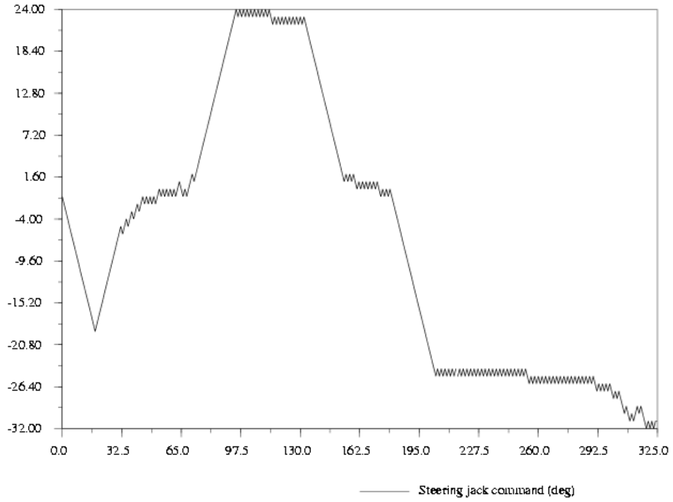

Figure 9: Command set generated for the first trajectory.

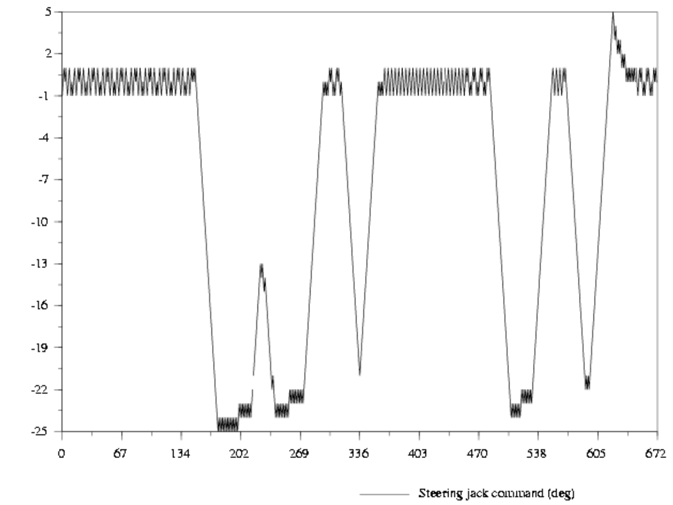

Figure 10: Command set generated for the second trajectory.

Those trajectories have been designed aiming to make each control points hardly reachable considering the mechanical limits of the machine (i.e.: the maximal steering angle). Indeed, on Fig. 7 and Fig. 8, we can remark that the machine encountered difficulties to execute the required steering commands: this is characterised by the motion overtaking at each control points of the path. Moreover, let us note the chattering phenomenon that appears on the output commands due to the difficult navigation condition. The fuzzy controller is forced to generate oscillatory discontinuous command to compensate the vehicle skids. See Fig. 9 and Fig.10. This discontinuity is undesirable in practice, since it involves high control, activity and may additionally excite high-frequency dynamics neglected in the course of modelling. Aiming to improve the fuzzy controller efficiency, a supervision method has been implemented and it is discussed in the following section.

5. FUZZY CONTROL IMPROVEMENT: A SLIDING MANIFOLD BASED APPROACH [7] [8] [9]

Those first poor results are the consequence of several perturbations. With the current system we have two main problems to deal with. On one hand, the commands are transmitted to the on-board computer via the wireless Ethernet link that leads to some significant transmission relays and on the other hand, the vehicle steering jack actuators appear to be integrator with high time constant. Hence, we are in presence of a relayed, delayed and highly non-linear system. Let us describe such a system. See Eq. 2 and Eq. 3.

Let X be the current state, X_d the desired state, $\hat{X}$ the estimated state (where $\hat{X} = \psi\left(\overline{X}\right)$, and ψ the estimator), $\overline{X}$ the observed state, $\tilde{X}$ the tracking error vector $\left(\tilde{X} = \hat{X} - X_d\right)$, δ the steering command, h the time relay constant, T the time constante of the steering actuators, v_s the instantaneous forward and backward steering jack speed motion and F the function that provide the system control.

$$\overline{X}(t) = \begin{pmatrix} \alpha(t-h) \\ \dot{\alpha}(t-h) \\ \Delta\theta(t-h) \\ \Delta\dot{\theta}(t-h) \end{pmatrix} \tag{1}$$

$$\dot{X} = A(\hat{X})X + B\delta \tag{2}$$

$$\dot{\delta}(t) = \min\left(\operatorname{sign}(F(\tilde{X}))v_s, \frac{\partial}{\partial t}F(\tilde{X})\right) \tag{3}$$

Sliding Manifold based supervision, besides being capable to cope with non-linear model and measurement uncertainties, promises to be sufficiently robust in respect to system perturbations and initial conditions. In a very broad sense, the functioning principle behind the Sliding Manifold Control approach is as follows.

Consider the non-linear system described above. Pick up a well behaved function S of the tracking error according to Eq. 4.

$$S(X,t) = \left(\frac{\partial}{\partial t} + \lambda\right)^{n-1} \tilde{X} \tag{4}$$

where λ is a strictly positive constant and $\tilde{X}$ the tracking error vector defined above. Now let us select a feedback control law u that keeps the state variable X on S(t) despite the presence of model imprecision and disturbances. Such a well behaved function is what it is called the sliding manifold. Roughly speaking, the sliding manifold constitute the set of states X that provide a null error $\tilde{X}$. The control law u is the solution of Eq. 5.

$$\varepsilon\dot{u} = dS \tag{5}$$

where S = S(z,t) denotes the sliding manifold which must be appropriately defined; dS denotes the differential of S :

$$dS(X,u) = \frac{\partial S(X,t)}{\partial t} + \lambda \frac{\partial S(X,t)}{\partial (X)} (A(\hat{X})X + Bu) \quad (6)$$

and ε includes information on the time delay that the tracking error will decay toward zero.
The rising question is how do we in practice apply this principle. In our case, we assume that the control law provided by the fuzzy controller is satisfactory. This assumption is true provided that the control points of the designed path are reachable from a kinematical point of view. Two steps are paramount.
First of all, the control law has to verify the sliding condition (see Eq. 7) where Φ is a fixed value.

$$\forall t \geq 0, |S(t)| \leq \Phi \quad (7)$$

Eq. 8 shows one of the possible solution of Eq. 7, where ξ is the defined tolerance along the trajectory.

$$\begin{aligned} & X_d(t) - (\xi + Ae^{-\beta t}) \leq X(t) \leq X_d(t) - (\xi + Ae^{-\beta t}) \\ & \Leftrightarrow |X(t) - X_d(t)| \leq \xi + Ae^{-\beta t} \end{aligned} \quad (8)$$

We can remark that such a method allow large starting point errors and will focus the command so to make the system reach a goal point in the fixed tolerance circle around the desired position. Moreover, such a supervised fuzzy controller will not generate extrem command variations as the control law will be suitably smoothed to achieve an optimal trade-off between control bandwidth and tracking precision. The supervised control process is now robust in respect of high frequency unmodeled dynamics.

C5.1 ***EXPERIMENTAL RESULTS***

To evaluate the benefits of this method, we have experimented it in the conditions proposed in section 4.6. The new results are shown on Fig. 11, 12, 13, 14. These new results testify to the efficiency of this approach. Regarding the tracking precision we obtain, the generated commands are much more adapted to the mechanical behaviour of the wheel loader.

6. CONCLUSIONS

The present work discusses results on a new control strategy implemented on a specific wheel loader. Control strategy is evaluated in respect to trajectory tracking performance and robustness. No one strategy is absolutely better than the others. Convenience of use is entirely application dependent. In this paper, some very a priori benefits and constraints associated to each of this control strategy are presented from the designer control point of view.

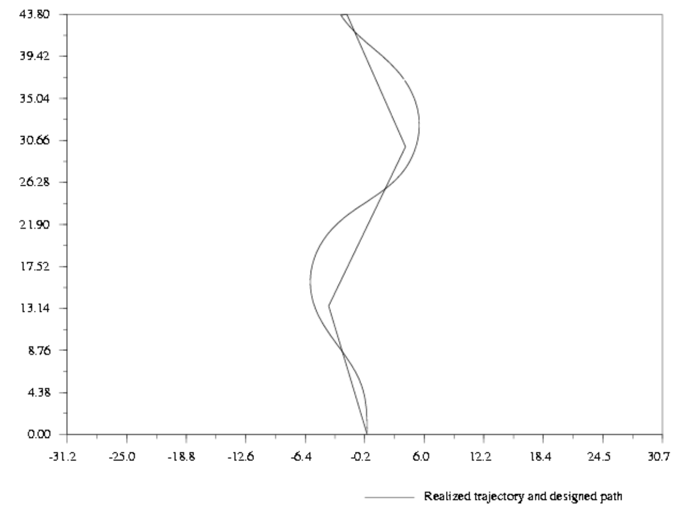

Figure 11: The new first trajectory test.

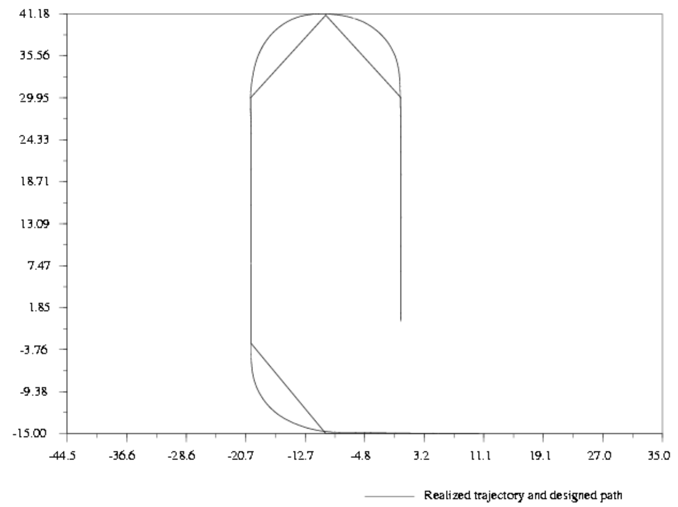

Figure 12: The new second trajectory test.

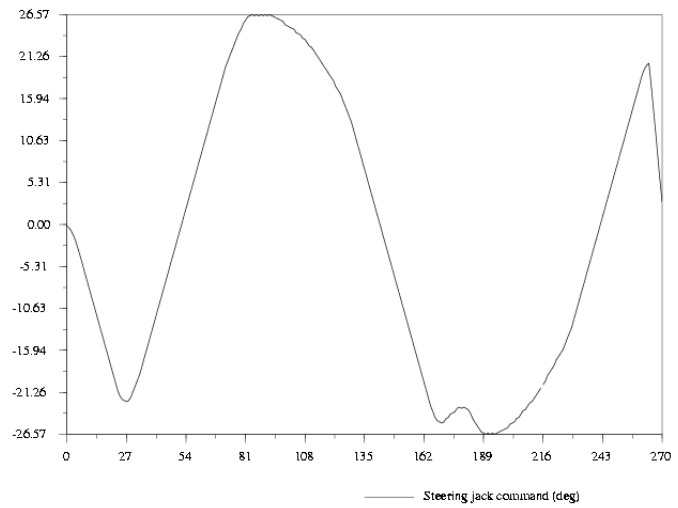

Figure 13: New command set generated for the first trajectory.

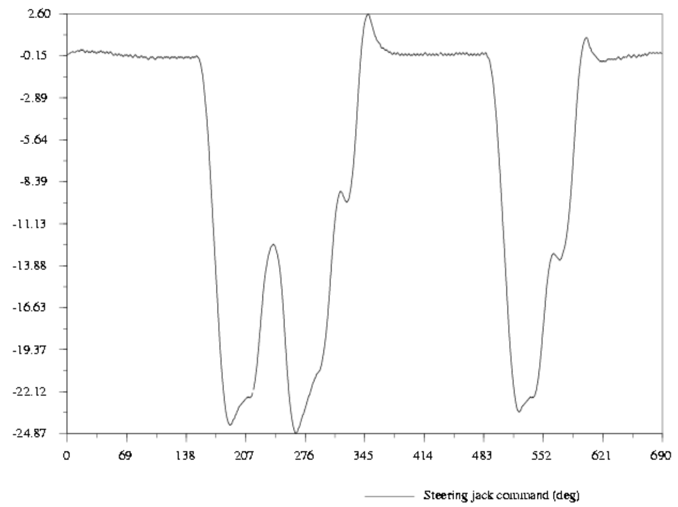

Figure 14: New command set generated for the second trajectory.

REFERENCES

[1] R.K. Miller, Mobile Robots, SEAI Technical Publications P.O. Box 50 Madison GA 30650, 1987,

[2] V. Dupourqué, Designing Intelligent and Autonomous Vehicles from the ROBUTER Concept, Mechatronics \& Robotics IOS Press ESPRIT CIM-Europe, 1991,

[3] U.M. Nassal, R. Junge, Fuzzy Control for Mobile Manipulation, Proceeding of the 1996 IEEE International Conference on Robotics and Automation,

[4] J.-Y. Jeon, J.-H. Kim, K. Koh, A Fuzzy Controller for a Car-Like Vehicle, INRIA Research report n° 3200 -- June 25, 1997,

[5] P. Garnier T. Fraichard, High-Precision Control of X-Y Table Using Experimental Evolutionary Programming-Based Scheme, Proceeding of the 1996 IEEE International Conference on Robotics and Automation,

[6] T. Fraichard, C. Laugier, Smooth Trajectory Planning For Car-Like Vehicle in a Structured World, INRIA Research report n° 1448 -- June 1991,

[7] A. Cavallo, G. De Maria, P. Nistri, A Sliding Manifold Approach to the Feedback Control of Rigid Robots, International Journal of Robust Non-linear Control, Vol. 6, 501-516, 1996,

[8] A. Cavallo, G. De Maria, P. Nistri, Some Control Problems Solved via Sliding manifold Approach, Differential Equations and Dynamical Systems, Volume 1, Number 4, October 1993, pp. 295 - 310.

[9] D. Gabay, Minimizing a Differentiable Function over a Differential Manifold, INRIA Research report n° 1448 -- June 1991,

© Copyright ROBOSOFT S.A. 1999