NEUROINTERFACES FOR SEMI-AUTONOMOUS OBJECT MOVING SYSTEMS

Edson P. Ferreira[1], Marcelo M. Lamego[2] and Bernard Widrow[3]

[1]*DEL, Universidade Federal do ES, Brasil, edinho@ele.ufes.br*
[2,3]*E.E. Dept., Stanford University, USA, mmlamego@stanford.edu*

Abstract: This article presents a framework and shows how neural networks can be employed in the generation of human-machine interfaces (neurointerfaces) for real time object moving problems. In a great number of applications, due to technical and economic factors, full automation is not possible. In such cases, the human presence is essential and indeed, the system performance becomes highly dependent on human skills. Accordingly, an interface that modifies the problem, allowing unskilled human operators to perform the same task in a satisfactory way, becomes extremely useful. The basic concepts and the scope necessary for the problem formulation are built in a clear framework. The adaptive nonlinear inverse modeling approach is employed as the basic methodology for specification and design of neurointerfaces. A successful application of a neurointerface that helps an operator to back up a scaled model truck connected to a double-trailer configuration is presented.

Keywords: Neural Networks, Intelligent Interfaces, Adaptive Inverse Control, Semi-autonomous Systems.

1. INTRODUTION

For many tasks, productivity, safety, and liability conditions require a considerable degree of skill from human operators. In order to overcome lack of skill, special human-machine neurointerfaces (Widrow, et al., 1998a) may be adopted. The basic idea is to change the operational space through a neural network, allowing the human operator to interact with the process through less-specialized commands. Accordingly, the operator devotes his attention to solve a less complex problem, directly at the task level. The objective is to improve the productivity and safety levels of such tasks even in the case of unskilled operators.

Although in the literature they are treated as different systems, articulated vehicles and manipulator robots are both articulated mechanical chains. Providing them with autonomous or semi-autonomous motion leads to similar direct and inverse geometric control problems. There are also other systems, like construction cranes, that can be treated similarly. These systems may me called **object moving systems**. The dynamics of the task goals and the dynamics of the object moving system establish, in principle, the complexity of the global control problem: the relationship between the controlled space, configuration space, and the operational space. In this work, functions and their inverses are to be implemented by neural networks in a strategy to provide neurointerfaces for moving object problems. This paper aims at providing a framework for the problem and at showing how neural networks can be employed in the generation of human-machine interfaces. Due to the complexity of such problems, neural networks are becoming a natural choice. Their abilities to reproduce highly nonlinear behavior are described extensively in the literature.

This article is divided in 6 (six) sections. Section 2 builds a framework for the problem. Section 3 presents the basic ideas concerning neurointerfaces. In Section 4, the adaptive inverse modeling approach, a framework utilized for neurointerface design is described. A successful neurointerface application, helping a human driver to back up a scaled model truck connected to a double-trailer configuration, is presented in section 5. Section 6 presents conclusions.

2. OBJECT MOVING PROBLEMS

To establish and define a class of object moving problems in the physical world, one must first introduce the concepts of ***operational space, configuration space, and controlled space***. Roughly speaking, the operational space is the one where the task goals are defined. The configuration space represents the elementary independent degrees of freedom (linear and/or rotational) in the ***object moving system***. Because they are mechanical systems in a ***state space*** representation, the state variables can be chosen as the configuration variables plus their derivatives. Finally, the ***controlled space*** is a subspace of the configuration space, where the variables are effectively powered by another system (controller and drivers). In this section, these 3 spaces, and the concepts of ***direct and inverse models*** for the functions linking them, will be discussed.

The kind of system of interest here is called an ***object moving system***. As examples of these systems, one can cite construction cranes (Lamego and Rey, 1995), artificial arms (Ferreira, 1987), human arms, mobile robots, fork-lift-trucks (Espinosa, et al., 1998), truck and trailers (DeSantis, 1994 and Widrow, et al.,1998a), etc. Each one of these systems has its own organization and its own controlled space, or a different space where its elementary movements are generated to compose a compatible movement with those in the operational space.

The object moving system of specific interest to this paper is the truck-trailer-trailer configuration shown in figure 1. The angle of the front wheels is *(θ_1)*, and the angles of the 2 joints are truck-trailer *(θ_2)*, and trailer-trailer *(θ_3)*. Only the first configuration variable *(θ_1)* is driven by an external source. The configuration variables in the active joints, are called controlled variables.

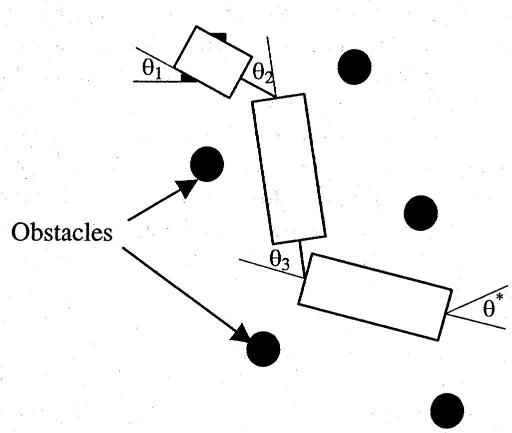

Figure 1: Truck-trailer System

Regarding the system of figure 1, the ***constraints*** are the jacknife configurations for θ_2 and θ_3, the limit for θ_1, and the location of nearby ***external obstacles***. The situation variables are defined for each different task goal. If the goal is going forward the task variable is the controlled variable θ_1. In this case the function between controlled and operational spaces is the identity function. If the goal is to back-up, the task variable is the angle θ^*. This is the angle between the desired direction of travel and the axis of the second trailer. In this case there is a nonlinear function linking variables θ^* and θ_1.

3. NEUROINTERFACES

For a human working with a semi-autonomous system, it is essential to perform all actions in the task space, in accord with the scheme illustrated in figure 2. Driving forward is a simple task because the person performs directly in the task space. To drive backwards, a coordination problem must be solved. To avoid having the human solving a coordination problem, we must provide an interface as shown in figure 2. Figure 2 shows the relationships among the elements defined previously for a semiautonomous object moving system.

To implement the interface of figure 2 we will use nonlinear inverse control techniques (Widrow and Walach, 1996), (Widrow, et al., 1998b), as a framework for the synthesis of neural network based solutions - ***neurointerfaces***. Aside from robots, other object moving systems, construction cranes for instance, are controllable with neurointerfaces (Lamego and Rey, 1995). At the end of this paper we will present a successful application, backing up a truck having the configuration of figure 1.

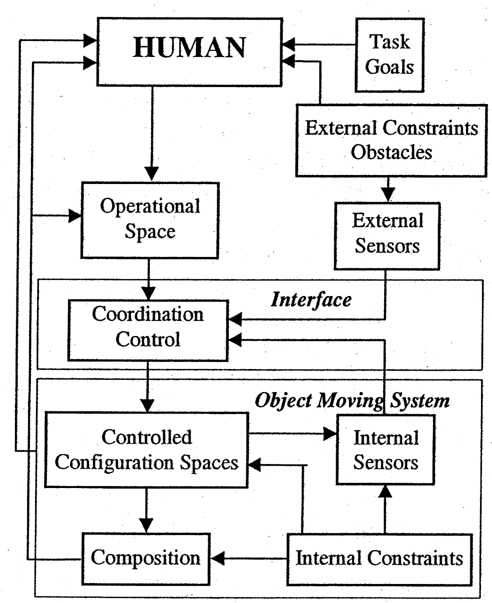

Figure 2: Semiautonomous Object Moving System.

For a system without an interface, an operator develops from his experience a set of causal rules that map standard behavior into control actions (cognitive model for the coordination), and this is exactly what we will try to reproduce with a neurointerface. While cases might exist in which the neurointerface provides just an approximation of those actions taken by an expert operator, the change of operational space made by the neurointerface, although it may not solve the problem completely, allows the human operator to interact with the system through less-specialized actions. Sometimes the operator can not observe directly the main variables of interest during operation. Instead, the operator controls the system by reasoning about a set of related variables that can be directly observed. The relationship between these observable variables and the main variables of interest are often not known precisely. A neurointerface applied to such a system may not be able to completely invert the system model due to the lack of information, and consequently may not solve the interface problem completely. Nevertheless, the productivity, safety and liability conditions may be far improved with its utilization.

There are cases, however, where the neurointerface can be fully specified. The main variables of interest are either directly available or may be expressed as some function of the observed variables. In addition, the mapping between standard behavior and control actions can also be achieved by using knowledge of the functional relationship between the main variables of interest. This is the case, for instance, in backing a truck and trailer to a loading platform. Although, this constitutes a difficult task for all but the most skilled truck drivers, a neurointerface can be fully specified and the trailer truck operation exercise reduced to a much less complex problem. In this case, the neurointerface may be considered as a black box that takes commands from the driver (desired direction of the back part of the trailler) and provides the necessary actions (steer the wheels) in order to achieve such a goal. The truck speed and the angle between cab and trailer are sufficient information to obtain precise inverse modeling of the system. We should note that the driver was not eliminated in this problem. Nguyen and Widrow, (1990), proposed a neural network that provided full automation in backing a trailer truck to a loading dock and indeed, eliminating the presence of the driver. In the present work, the human action is essential. In fact, the driver is concerned with providing the desired spatial trajectory, free of obstacles and normally the shortest one. The truck-backing-up can be approximated by a kinematics inverse modeling problem. The dynamic effects that may occur during the operation are not significant, thus the coordination problem is simpler than a problem containing both kinematics and dynamic effects.

4. NEUROINTERFACES DESIGN – ADAPTIVE NONLINEAR INVERSE APPROACH.

Adaptive nonlinear inverse modeling has evolved from a similar approach for linear systems. Basically, the objective is to cancel the plant nonlinear dynamic effects by using a nonlinear device that can reproduce an approximated inverse of the plant. The term "approximated" is employed to emphasize that, in general, a nonlinear system does not possess an inverse. However, despite some pathological cases that might eventually exist, the methods of adaptive inverse modeling can often be applied to obtain acceptable inverse approximations of nonlinear systems. The specification of a neurointerface is based on the idea that a nonlinear plant can be approximated by a neural network model, here represented by the function $f : R^{p+q+2} \rightarrow R$, of the form

$$y_{k+1} = f(y_k, y_{k-1}, \cdots, y_{k-p}, u_k, u_{k-1}, \cdots, u_{k-q}, w_M) \qquad (1)$$
$$y, u \in R^m, w_M \in R^{\ell_M}$$

The variables u and y are, respectively, the plant input and plant output and w_M is the neural network weight vector. The first step in a neurointerface design is to obtain a neural network model for the plant as defined in equation (1) and then, use it to obtain a neural approximation for the plant inverse (neurointerface). Widrow , et al. (1998a), describes the nonlinear system identification procedure. The neural network uses as its inputs the current and previous values of the plant input and also previous values of its output. Its output represents an approximation of the plant output. The neural model can be trained with a set of input-output data either acquired from the real plant or obtained from the plant mathematical model (if available).

The Algorithm backpropagation-through-time (Werbos, 1990) may be used to adapt the weights of the neural network model. If the input of the neural network model does not include any connection to plant output (a feedforward neural network), the conventional backpropagation algorithm (Rumelhart, et al., 1986), may be employed.

Figure 3 illustrates the training of the neurointerface to compute an approximate inverse for the obtained plant neural model. The neurointerface computes a function $g : R^{P+Q+1} \rightarrow R$, of the form

$$u_k = g(u_{k-1}, u_{k-2}, \cdots, u_{k-Q}, r_k, r_{k-1}, \cdots, r_{k-P}, w_C) \qquad (2)$$
$$u, r \in R, \; w_C \in R^{\ell_C}$$

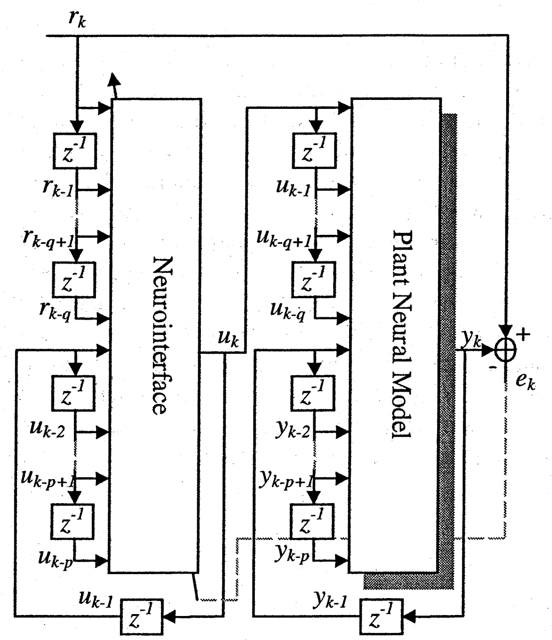

Figure 3: Adapting Neurointerfaces

Variables r and u are, respectively, the neurointerface input and output, and w_C is the neurointerface weight vector. During training, the signal r can be random noise with suitable choice of spectrum.

Because the plant is nonlinear, the neurointerface is trained in the configuration in which it will normally work. In the cascade of nerointerface and plant, the neurointerface comes first. Consequently, to compute the mean square error gradient with respect to the neurointerface weights, information concerning the plant must be available. This is the reason why the plant is replaced by its neural realization during the neurointerface adaptation procedure. The change in the neurointerface weights at each training step is in the negative direction of the gradient of the system mean square error $E(e_k{}^2)$. To find the gradient, the chain-rule expansion for ordered derivatives is employed

$$\frac{\partial \|e_k\|^2}{\partial w_C} = -2\frac{\partial^+ y_k}{\partial w_C} e_k \tag{3}$$

with

$$\frac{\partial^+ y_k}{\partial w_C} = \sum_{i=0}^{q}\left(\frac{\partial y_k}{\partial u_{k-i}}\right)\left(\frac{\partial^+ u_{k-i}}{\partial w_C}\right) + \sum_{i=1}^{p}\left(\frac{\partial y_k}{\partial y_{k-i}}\right)\left(\frac{\partial^+ y_{k-i}}{\partial w_C}\right) \tag{4}$$

and

$$\frac{\partial^+ u_k}{\partial w_C} = \left(\frac{\partial u_k}{\partial w_C}\right) + \sum_{i=1}^{Q}\left(\frac{\partial u_k}{\partial u_{k-i}}\right)\left(\frac{\partial^+ u_{k-i}}{\partial w_C}\right) \tag{5}$$

Each of the terms in equations (4) and (5) is either a Jacobian matrix, which may be calculated using the *dual-subroutine* (Werbos, 1992) of the backpropagation algorithm, or is a previously calculated value of $\partial^+ u_k / \partial w_C$ or $\partial^+ y_k / \partial w_C$. To be more specific, the first term in equation (5) is the partial derivative of the neurointerface's output with respect to its weights. This term is one of the Jacobian matrices of the neurointerface and may be calculated with the dual subroutine of the backpropagation algorithm. The second part of equation (5) is a summation. The first term of the summation is the partial derivative of the neurointerface's current output with respect to a previous output. However, since the neurointerface is externally recurrent, this previous output is also a current input. Therefore the first term of the summation is really just a partial derivative of the output of the neurointerface with respect to one of its inputs. By definition, this is a sub-matrix of the Jacobian matrix for the network, and may be computed using the dual-subroutine of the backpropagation algorithm. The second term of the summation in equation (5) is the ordered partial derivative of a previous output with respect to the weights of the neurointerface. This term has already been computed in a previous evaluation of equation (5), and need not be re-computed. A similar analysis may be performed to determine all of the terms required evaluating equation (4). After calculating these terms, the weights of the neurointerface may be adapted using the weight-update equation:

$$\Delta w_{C\,k} = 2\mu \frac{\partial^+ y_k}{\partial w_C} e_k \tag{6}$$

The neurointerface, once it is trained, is designed to operate in real time without any further adaptation. This implies that its adaptation procedure can be performed offline. The neurointerface does not cancel disturbances that may occur in the plant. It just changes the operational space through a recurrent neural network. Nonetheless, there are situations where the plant inversion supplied by the neurointerface is not adequate for providing reliable operating conditions. Disturbing effects may occur during the plant operation and may lead it to risky operating regions. To overcome the disturbing effects and provide more reliable operating conditions to the operator, adaptive linear control schemes may be adopted as shown by Widrow and Walach, 1996, or Widrow, et al., 1998b.

5. EXPERIMENTAL RESULTS

This section briefly presents experimental results of a neurointerface that reduces the difficulty of trailer truck backing operations. We will consider a real scaled truck with two trailers, as presented by Widrow, et al., 1998a, with the same configuration shown in figure 1. For this application we will consider the last configuration angle θ_3 as the variable to be controlled. The objective of steering while backing is to control θ_3, which will determine the radius of curvature of the backing trajectory.

The neurointerface may be considered as a black box that takes commands from the driver, in this case the desired direction of the back part of the last trailer, and provides the necessary control actions (steer the front wheels to ultimately control variable θ_3). In this implementation, the neurointerface works in closed loop. The adaptive linear control topology presented by Widrow, et al., 1998 is employed here. For this application, the neurointerface has as its inputs, the truck speed, the angle θ_2, the desired value of the angle between the first trailer and the second one (θ_3) and the previous value of the neurointerface's output (the front wheel steering angle, θ_1).

The neurointerface was designed following the steps described in section 3. Acquired data from the truck prototype was used to obtain the neural model. The obtained neural model was used for the training of the neurointerface.

An adaptive disturbance canceller was used (Widrow and Walach, 1996) to mitigate the effects of plant disturbance.

The offline process for adaptation of the disturbance canceler is started at every 500 samples, sampling period of 30 ms. The data acquired in this interval (15 sec.) are used as the training set. Experimental results are shown in figure 5. They correspond to sequences of data acquired from the model truck and trailers moving backwards. From this figure, one can see that the angle θ_3 was able to track the commanded angle. It was possible to precisely steer the truck and two trailers traveling backwards at quite high speed. Direct human control of the truck and two trailers going backwards without the neurointerface was impossible.

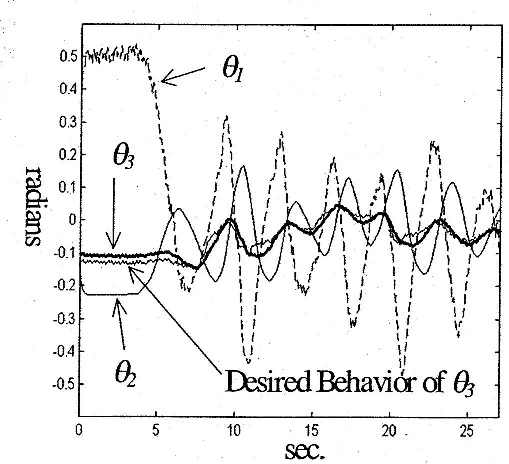

Figure 5: Experimental results.

6. CONCLUSIONS AND FURTHER WORK

This article presents a new framework for object moving problems and a new approach for generation of human-machine interfaces through neural networks (neurointerfaces). The adaptive nonlinear inverse modeling approach is employed as the basic methodology for specification and design of neurointerfaces. The neurointerface is able to cancel most of the nonlinear effect the plant may have and, indeed, it can be used in combination with adaptive linear control schemes for the control of nonlinear plants. A successful application, a neurointerface that helps an operator to back up a scaled model truck connected to a double-trailer configuration, is also presented. The results lead one to conclude that full utilization of neurointerfaces for real time applications will be very useful. With further work, the degree of autonomy of the truck will be improved adding sensors to the system. For the time being the operator must verify all the constraints due to obstacles. This methodology will be extended for more complex systems, like robots working with real tasks.

REFERENCES

DeSantis, R. M. (1994). Path-tracking for a Tractor-Trailer-like Robot. *The International Journal of Robotics Research.* **Vol. 13**, pp 533-544., December.

Espinosa, F., R. Garcia, M. Mazo, E. Lopez and R. Mateos (1998). Modelling and Simulation of the Kinematic and Dynamic Behavior of a Fork-Lift Truck. *3th IFAC Symposium of Intelligent Autonomous Syst.*, pp 75-79. Madrid.

Ferreira, E. P. (1987). *Robotica Industrial*, Ed. Kapelusz - EBAI, Buenos Aires.

Lamego, M. M and J. P. Rey (1995). The Interval Based Control Technique: Controlling Physical Systems Through Imprecise Models. *Proceedings of the IEEE Industrial Applications Society*, Annual Meeting, pp. 1822–1827, Florida, USA, October.

Nguyen, D., and B. Widrow (1990). Neural Networks for Self-learning Control Systems. *IEEE Control Systems Magazine*, pp. 18–23,April.

Rumelhart, D. E., G. E. Hinton and R. J. Williams (1986). Learning internal representations by error propagation. *Parallel Distributed Processing*. (D. E. Rumelhart and J. L McClelland editors), **volume 1**, chapter 8. MIT Press, Cambridge MA.

Werbos, P. (1990). Backpropagation through time: What it does and how to do it. *Proceedings of the IEEE*, **Vol. 78(10),** pp. 1545–1680.

Werbos, P.(1992). Neurocontrol and supervised learning. *Handbook of Intelligent Control: Neural, Fuzzy and Adaptive Approaches*. (D. White and D. Sofge editors), chapter 3. Van Nostrand Reinhold, New York.

Widrow, B. and E. Walach (1996). Adaptive Inverse Control. Prentice Hall PTR, Upper Saddle River, NJ.

Widrow, B., E. P. Ferreira and M. .M. Lamego (1998a). Neurointerfaces for Human-Machine Real Time Interaction. *5th International Federation of Automatic Control Workshop on Algorithms and Architectures for Real-Time Control*, pp 131-136, April.

Widrow, B.,. E. P. Ferreira and M. .M. Lamego (1998b). Adaptive Inverse Control Based on Nonlinear Adaptive Filtering. *5th International Federation of Automatic Control Workshop on Algorithms and Architectures for Real-Time Control*, pp 247-252, April.