[6]Lee, A. Y., and Tsuha, W. S., "An Enhanced Projection and Assembly Model Reduction Methodology," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 1, 1994, pp. 69–75.

[7]Craig, R. R., Jr., *Structural Dynamics: An Introduction to Computer Methods*, John Wiley & Sons, New York, 1981.

[8]Lee, A. Y., and Tsuha, W. S., "A Component Modes Projection and Assembly Model Reduction Methodology for Articulated Flexible Structures," JPL Internal Document D-9364, Jet Propulsion Laboratory, California Institute of Technology, Dec. 1991.

Synthesis of Minimum-Time Feedback Laws for Dynamic Systems Using Neural Networks

Allan Y. Lee* and Padhraic Smyth†
Jet Propulsion Laboratory, Pasadena, California 91109

Introduction

GUIDANCE and control dynamical optimization problems in aeronautics, astronautics, and other fields have been routinely formulated and solved over the past 30–35 years. Of necessity, these complex optimization problems are often numerically solved. Past research efforts have produced several software packages (see, e.g., Refs. 1 and 2) which can be used to determine the open-loop solutions of dynamical optimization problems.

By an open-loop solution we mean that the control at any time instant is not explicitly determined by the states of the system at that instant. It is well-known that a system with an open-loop controller can be sensitive to noise and external disturbances. In contrast, feedback control, in which the control is a function of the instantaneous state of the system is generally robust with respect to such disturbances. Unfortunately, only rarely is it feasible to determine the feedback laws for nonlinear systems of any practical significance.

A novel approach of using neural networks to synthesize nonlinear feedback laws is explored in the present research. The details and efficacy of this approach will be demonstrated using a minimum-time orbit injection problem.

Open-Loop Solutions of Dynamical Optimization Problems

Consider the following dynamical optimization problem:

$$\min_{\vartheta(\tau),\,\pi} I = \phi_1(x_1, \pi) + \int_0^1 L(x, \vartheta, \pi, \tau)\, d\tau \tag{1}$$

$$\dot{x} = f(x, \vartheta, \pi, \tau), \qquad x(0) = \text{given} \tag{2}$$

$$\psi(x_1, \pi) = 0 \tag{3}$$

Here, $x(n \times 1)$, $\vartheta(m \times 1)$, and $\pi(p \times 1)$ are the state, control, and unknown parameter vectors, respectively. The independent variable (usually time) is τ. The scalar function L denotes a running cost while $f(n \times 1)$ is a vector function. The parameter vector π and the control vector $\vartheta(\tau)$ are to be optimally determined to minimize the cost functional I subject to the initial condition given in Eq. (2) and the terminal constraint $\psi(q \times 1)$ given in Eq. (3). The subscript 1 in Eqs. (1) and (3) denotes the condition of the variable concerned at the end time T. If T is free, we can define $\tau = t/T$ (which now varies from 0 to 1), and use it to transform the problem to one with a fixed end time.[2] The unknown end time T is now part of π.

The optimality conditions of this optimization problem are well-known, and are given in, for example, Refs. 2 and 3. These conditions are all that are needed to solve the dynamical optimization problem. However, unless the functions L and f are quite simple, analytical results are difficult to obtain and the optimization problem must be solved numerically, using, for example, the combined parameter and function optimization algorithm (CPFA).[2]

The solution obtained is only open-loop. A system with an open-loop controller can be sensitive to noise and external disturbances. In contrast, feedback control, in which the control is a function of both the state and time, $\vartheta = \vartheta(x, \tau)$, is generally robust with respect to such disturbances. For stationary systems where f and L are not explicit functions of time, and the end time is free, we have $\vartheta = \vartheta(x)$.[3] Unfortunately, only rarely is it feasible to analytically determine these feedback laws for nonlinear systems of any practical significance.

To overcome this problem, Breakwell et al.[4] obtained a linear neighboring optimal feedback law (perturbation guidance law) via the minimization of the second variation of the performance index. The ranges over which these perturbation guidance laws are useful will depend on the system and might be small for nonlinear systems. In Ref. 5, open-loop optimal control data are used to train a feedforward multilayer neural network. The network is then used effectively to control a two-link robotic arm between its end conditions in minimum time. In this study, a similar approach is used to synthesize a feedback controller for a minimum-time orbit injection problem.

Artificial Neural Networks as Nonlinear Mapping Approximators

An artificial neural network (henceforth referred to as neural network) is a system which is composed of many simple and similar nonlinear processing elements (neurons). A typical neural network consists of an input layer, one or more hidden layers, and an output layer. The input layer works as a buffer and may include some preprocessing of the input data. The outputs of the hidden nodes are calculated from: $z_j^2 = F(\Sigma_{i=1}^{N_1} w_{ij}^1 z_i^1 + \theta_j^1)$, for $j = 1, \ldots, N_2$. Here, z_i^1 $(N_1 \times 1)$ are inputs to the network, z_j^2 $(N_2 \times 1)$ are outputs of the hidden layer, w_{ij}^1 $(N_1 \times N_2)$ are weights between the network inputs and the hidden layer outputs, and $\theta_j^1(N_2 \times 1)$ are internal offsets of the hidden layer. The function F is a monotone continuous function, usually a sigmoid function, or a hyperbolic tangent function. The sigmoid function $F(s) = 1/[1 + e^{-s}]$ is used here. The outputs of the network are calculated similarly.

To perform a desired mapping, the network has to adapt its weights through a training process. During this process, the network is presented with an input vector z_i^1, which causes its output to be z_k^3 (z_i^1). We then adjust the weights to cause the output to be something closer to the $d_k(z_i^1)$, the desired output. Back propagation algorithms are usually used to train multilayered neural networks. One version of these algorithms is described in Ref. 6.

In the present research, we train the network to approximate the optimal, nonlinear states-to-controls mapping. First, the open-loop solutions of an optimization problem are computed for multiple sets of end conditions at and around the nominal condition. Next, the computed time histories of the states and controls are then used to train a neural network with a prese-

Received July 10, 1993; revision received Sept. 8, 1993; accepted for publication Sept. 10, 1993.

*Member of Technical Staff, Guidance and Control Section, 4800 Oak Grove Drive.

†Member of Technical Staff, Communications Systems Research Section, 4800 Oak Grove Drive.

lected architecture. Finally, the trained network is used to output an approximate optimal control given the measured system states. It should be noted that other function approximation tools, such as radial basis functions are alternative candidates for achieving the desired mapping.

Minimum-Time Orbit Injection Problem[3]

Consider a particle of mass m, acted upon by a constant thrust of magnitude ma. If we use a coordinate frame where the y axis is opposite in direction to a constant gravitational force mg, then the velocity components (u, v) of the particle are governed by (cf., Fig. 1):

$$\dot{u} = a \cos \beta, \qquad \dot{v} = a \sin \beta - g, \qquad \dot{y} = -v \tag{4}$$

Here β is defined in Fig. 1, and $\dot{(\cdot)} \triangleq d/dt$. We wish to planar transfer the particle, initially at rest, to a path an altitude h away, and to arrive with a terminal velocity U parallel to the x axis in minimum time. Because the end time T of the problem is free, we use $\tau \triangleq t/T$ to transform Eq. (4) to have a unity end time.

The nominal end conditions of the state are $u(0) = v(0) = 0$, $y(0) = h$, and $u(T) = U$. Solving this problem for the case with $h = U = a = 1$, and $g = 1/3$ using CPFA, we obtain the optimal control and state time histories depicted in Fig. 2. The computed minimum time T is 2.2113 units, which closely agrees with that computed by the analytical formulae of Ref. 3 ($T_c = 2.2113$).

Consider now the synthesis of a feedback guidance law for trajectories whose end conditions (h, U) deviate from the nominal values (1, 1) by $\sqrt{\delta h^2 + \delta U^2} \leq 0.1$. To this end, we train a neural network using optimal time histories data generated with nine sets of end conditions. They include the nominal end conditions, and eight sets of end conditions that deviate from the nominal end condition by $(\delta h, \delta U)$, where $\sqrt{\delta h^2 + \delta U^2} = 0.2$.

From Fig. 2, we note that $\beta(\tau)$ varies between ± 1.5 rad. Neural networks with a sigmoidal nonlinearity (whose output varies between 0 to 1) cannot approximate β over this range of variation. Hence, a modified control variable, $\bar{\beta} \triangleq \beta/4 + 0.5$, which varies between 0.1 and 0.9, is used instead. Next, we select a 3-6-1 architecture for the neural network because the system has three state and one control variables. A single hidden layer with six nodes (double that of the input layer) has been added between the input and output layers. A neural network training program[6] which uses a conjugate gradient-based algorithm is then used to train the 3-6-1 network.

The trained network can now be used to generate a control β given the current states (u, v, y) of the particle. The optimal control time history (via CPFA) for a test case with $(\delta h, \delta U) = (0.1/\sqrt{2}, 0.1/\sqrt{2})$, and that generated by the trained network, are compared in Fig. 3. The comparisons are excellent in spite of the relatively simple network architecture (the RMS error per sample is 0.0088).

It turns out that for the problem studied, an optimal feedback control law can be analytically derived: $\beta = \beta(\eta, \gamma)$.[7] Here, $\eta \triangleq (\sqrt{2ay}/U - u)$, and $\gamma \triangleq (v/U - u)$. Unfortunately, this nonlinear function $\beta(\eta, \gamma)$ is very complicated, and the determination of it involves the simultaneous solution of transcendental equations.[7] Again, we can use a neural network to approximate this unknown relation. Here, instead of using the time histories of $\bar{\beta}$ and (u, v, y) to train a 3-6-1 network, we use the time histories of $\bar{\beta}$ and (η, γ) to train a 2-4-1 network. A singularity occurs at the end time when $y = v = U - u = 0$. To overcome this difficulty, data points obtained at the end time are discarded from the training data. Optimal control time histories (via CPFA) and those predicted by the second trained network, for the same test cases, are compared in Fig. 3 (the RMS error per sample is 0.0047). The results obtained are slightly better than those found before.

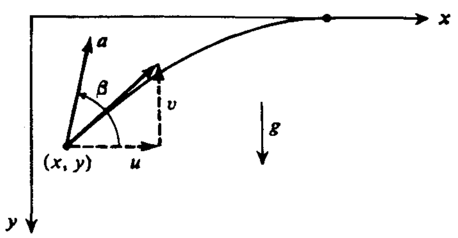

Fig. 1 Nomenclature for the minimum-time orbit injection problem.[3]

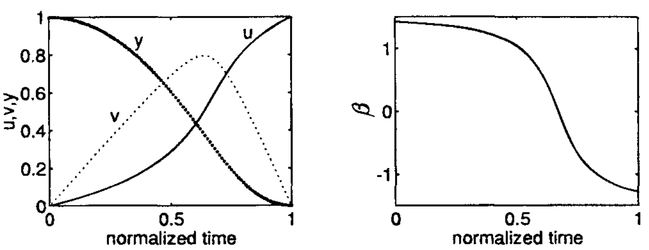

Fig. 2 Optimal control and state time histories of the minimum-time orbit injection problem (nominal case).

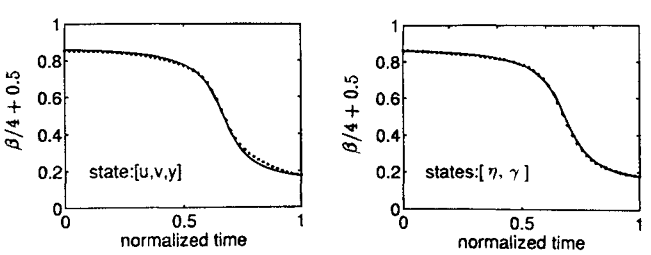

Fig. 3 Comparisons of control time histories: —— optimal results, . . . neural network approximation.

Concluding Remarks

In this research, neural network-based feedback laws for dynamic systems are synthesized using the computed optimal time histories of the state and control variables. The efficacy of the proposed approach has been successfully demonstrated on a minimum-time orbit injection problem. If the methodology is found to scale up well to real-life problems, with many state and control variables, it would have significant potential for a variety of guidance and control problems.

Acknowledgments

The research described in this paper was conducted at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with NASA, with the second author supported in part by ARPA under grant number N00014-92-J-1860. We wish to thank F. Hadaegh and G. Man for their interest and encouragement.

References

[1] Goh, C. J., and Teo, K. L., "Control Parameterization: A Unified Approach to Optimal Control Problems with General Constraints," *Automatica*, Vol. 24, No. 1, 1988.

[2] Lee, A. Y., and Bryson, A. E., Jr., "Neighboring Extremals of Dynamic Optimization Problems with Parameter Variations," *Optimal Control Applications and Methods*, Vol. 10, 1989, pp. 39–52.

[3] Bryson, A. E., Jr., and Ho, Y. C., *Applied Optimal Control*, Hemisphere, Washington DC, 1975.

[4]Breakwell, J. V., Speyer, J. L., and Bryson, A. E., Jr., "Optimization and Control of Nonlinear Systems Using the Second Variation," *SIAM Journal of Control*, Series A, Vol. 1, No. 2, 1963, pp. 193–217.

[5]Goh, C. J., Edwards, N. J., and Zomaya, A. Y., "Feedback Control of Minimum-time Optimal Control Problems Using Neural Networks," *Optimal Control Applications and Methods*, Vol. 14, March 1993, pp. 1–16.

[6]Barnard, E., and Cole, R., "A Neural Network Training Program Based On Conjugate Gradient Optimization," Oregon Graduate Centre TR No. CSE 89-014, OR, 1989.

[7]Winfield, D. H., and Bryson, A. E., "Nonlinear Feedback Solution for Minimum-time Injection Into Circular Orbit With Constant Thrust Acceleration Magnitude," NASA CR 81474, July 1966.

Tracking Mobile Vehicles Using a Non-Markovian Maneuver Model

D. D. Sworder*
University of California, San Diego, La Jolla, California 92093
R. Vojak†
INRIA-Rocquencourt, 78153 Le Chesnay Cedex, France

Introduction

MEASUREMENTS of the motion paths of agile targets show motion at roughly constant speed for extended intervals. An osculating path is created by turning at different rates and at different times. A tractable model of such (planar) motion is given by the forth-order stochastic differential equation

$$\mathrm{d}\begin{pmatrix} X \\ Y \\ V_x \\ V_y \end{pmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\omega_\psi \\ 0 & 0 & \omega_\psi & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ V_x \\ V_y \end{pmatrix} \mathrm{d}t + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \mathrm{d}\begin{pmatrix} w_x \\ w_y \end{pmatrix} \tag{1}$$

where $\{X, Y\}$ are position coordinates, and $\{V_x, V_y\}$ are associated velocities. The target is subject to two types of acceleration: a wide band omnidirectional acceleration represented by a Brownian motion $\{w_x, w_y\}$ with intensity W ($\mathrm{d}w\, \mathrm{d}w' = W\,\mathrm{d}t$), and a maneuver acceleration represented by the turn rate process $\{\omega_\psi\}$. The former influences both the speed and the direction of the target, and fits well within the linear-Gauss-Markov (LGM) modeling framework. The latter is troublesome because the sample paths of the maneuver process are better described by assuming that they are piecewise constant, taking on values from a fixed set of turn rates; $\omega_\psi \in \{a_1, \ldots, a_K\}$ (see Ref. 1 and the references therein). Treating the maneuver as an additive disturbance neglects the strict coupling between the direction of the acceleration and the velocity vector.

An image-enhanced tracking problem has been explored in numerous references using a dual-path architecture (e.g., Ref. 2) in which one path maps the image processor outputs into a maneuver estimate. The other path maps range-bearing data into an estimate of target location, with the image-information used to adjust the gains and biases in the position path. In a series of studies, it has been shown that the image-enhanced algorithms promise performance superior to that attainable from a tracker using range-bearing measurements exclusively. Image algorithms studied ranged in complexity from simply adding the image-based maneuver acceleration estimate to a conventional extended Kalman filter (EKF) (Ref. 3) to adding mean acceleration and augmenting W with image-based "pseudonoise," (called $\mathrm{EKF_P}$ in Ref. 4) and, finally, to one which used several unconventional error moments to adapt the gains of a Kalman filter (called $\mathrm{EKF_A}$ in Ref. 5). The order in which the trackers are listed is indicative of both the tracking performance and the complexity of the algorithm.

Each of the referenced algorithms was derived on the basis of a Markov maneuver model. The sojourn times of a Markov process are exponentially distributed, and there is a strong likelihood of short modal lifetimes. Paths of agile targets do not have this macroscale behavior. There is little evasive purpose for maneuvers which are short with respect to the response dynamics of the vehicle; the duration of turns are unpredictable but, clearly, not exponentially distributed. A better representation of the maneuver sojourn time is provided by a γ density; a two parameter family $\gamma(t;R, \Lambda)$, in which R controls shape and Λ controls time scale: $\gamma(t;R, \Lambda) = I_{[t>0]}\Lambda[\Gamma(R)]^{-1}(\Lambda t)^{R-1} \exp(-\Lambda t)$; $R, \Lambda > 0$. The mean of a gamma distributed random variable is $\nu = R/\Lambda$, and because it has a more intuitive connotation, ν will be used instead of Λ to indicate the time scale. Figure 1 shows three γ densities with $\nu = 1$. The maneuver sojourns described by the first ($R = 1$) would generate the Markov model used in the indicated references. The second ($R = 2$) dramatically reduces the frequency of very short lifetimes, and the third curve ($R = 5$) is sharply peaked, with more quasipredictable intervals. The densities with $R > 1$ mirror more closely the turn intervals for an agile target. This paper compares non-Markovian ($R > 1$) tracking algorithms with simpler Markov ($R = 1$) versions on a coast-turn-cost path. It has been shown that the performance of $\mathrm{EKF_P}$ improves with a non-Markov maneuver model (Ref. 6). Unfortunately, the $R > 1$ algorithms are more complicated to implement, and this is particularly true for the $\mathrm{EKF_A}$ tracker. It is shown here that a simple (Markov) $\mathrm{EKF_A}$ is superior to the $\mathrm{EKF_P}$ even when the latter uses a nonMarkov acceleration model. An $\mathrm{EKF_A}$ for $R > 1$ has improved response, but not to a degree commensurate with the increase in algorithmic complexity. This relative improvement is found even when as the realized sojourn times differ from their mean values. This suggests that the simplest EKF is a suitable for applications that are not well described by a Markov model.

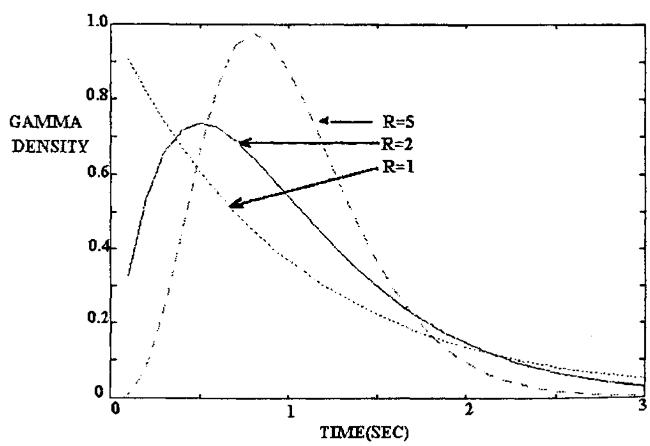

Fig. 1 Gamma densities for different values of *R*.

Received May 3, 1993; revisision received July 5, 1993; accepted for publication July 13, 1993.

*Professor, Department of Electrical and Computer Engineering.
†Research Associate.