Proceedings of the 35th
Conference on Decision and Control
Kobe, Japan • December 1996

Model Predictive Control: Breaking through Constraints

Vesna Nevistić

Automatic Control Laboratory
Swiss Federal Institute of Technology (ETH)
CH-8092 Zürich, Switzerland
e-mail: vesna@aut.ee.ethz.ch

James A. Primbs

Control and Dynamical Systems
California Institute of Technology
Pasadena, California 91125
e-mail: jprimbs@hot.caltech.edu

Abstract

Because it naturally and explicitly handles constraints, particularly control input saturation, model predictive control (MPC) is a potentially powerful approach for nonlinear control design. However, nonconvexity of the nonlinear programs (NLP) involved in the MPC optimization makes the solution problematic. Extending the concept of solving the Hamilton–Jacobi–Bellman equation backwards (the so-called "converse HJB approach") to the constrained case provides a method to generate various classes of challenging nonlinear benchmark examples, where the true constrained optimal controller is known. Properties of MPC-based constrained techniques are then evaluated and implementation issues are explored by applying both nonlinear MPC and MPC with feedback linearization.

1 Introduction

Determination of the optimal feedback law for nonlinear optimal control problems require solutions of Hamilton-Jacobi-Bellman (HJB) partial differential equations. Difficulties in solving the HJB equation for high dimensional systems have precluded their use except in specific areas, and has motivated the study of alternative control techniques.

Euler-Lagrange methods, which are strictly local and thus computationally less formidable, have found extensive use in open loop optimization for many decades. While it has long been recognized that such methods could also be used for on-line feedback control using a receding horizon, improvements in computer power has made that a practical possibility. Receding horizon control (RHC), also called model predictive control (MPC), was first exploited in the process control industry where relatively slow sample times made extensive on-line intersample computation feasible. One of the most attractive features of MPC is that both multivariable input and output constraints can be handled naturally and explicitly by direct inclusion into the optimization problem. The most important, and fortuitously also the most easily handled, is hard constraints or saturation of the control input, which makes MPC very useful in a variety of applications.

On the other hand, the purely local nature of the underlying Euler-Lagrange treatment of optimal control remains one of the major challenges facing application of MPC to nonlinear systems, because the resulting nonlinear optimization problems rarely have exploitable convexity properties. For these reasons, an essential issue, both theoretical and practical, is whether the optimization can be successfully employed in MPC. Since in most cases, the optimal solution, even if there are no constraints, is unknown and uncomputable, it is not easy to evaluate the results obtained by using MPC.

However, by introducing the *converse optimal control problem*, which follows the basic idea of solving the HJB equation backwards (*"converse HJB method"*), various classes of nonlinear systems for which the optimal controller is known can be constructed and used as benchmarks for validating the performance of different control techniques. In contrast to the standard optimal control problem, the converse HJB method starts with an optimal value function, and for a given performance objective uses the HJB equation to characterize the class of dynamics for which the assumed value function corresponds to the solution of the optimal control problem. [1].

In this paper the converse HJB method is extended to the constrained case, where the *converse constrained optimal control problem ("converse constrained HJB method")* is stated, and some challenging benchmark examples, for which the constrained optimal controller is known, are generated. These examples are used to explore the properties of various control schemes, and, in particular, the efficiency and necessity of using constrained MPC-based control schemes is investigated.

2 Constrained Optimal Control Problem

Consider the following nonlinear system:

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t), \; f(0) = 0 \quad (1)$$

subject to input and state inequality constraints:

$$c(x, u) \leq 0 \quad (2)$$

with performance objective:

$$J(u) = \int_o^\infty (q(x(t)) + u^T(t)u(t)) \; dt \quad (3)$$

where $q : \mathbf{R}^n \to \mathbf{R}$ is C^1.

Def 1 ***Constrained Optimal Control Problem:*** *Find a state-feedback control law $u^* = \Phi(x)$ such that the performance objective (3) is minimized, subject to the nonlinear system dynamics (1) and inequality constraints (2).*

The constrained optimization problem (1)–(3) can be converted into an equivalent problem given by:

$$0 = \min_{u \in \mathbf{R}^m} \left\{ \frac{\partial V(x)}{\partial x}(f(x) + g(x)u) + u^T u + q(x) \right\} \quad (4)$$

$$\text{subject to: } c(x,u) \leq 0$$

where $V(x)$ denotes the *value (storage) function*, given by:

$$V(x) = \min_u \int_0^\infty (q(x(t)) + u^T(t)u(t)) \; dt \quad (5)$$

Equation (4) is referred to as the Hamilton-Jacobi-Bellman equation, which states a sufficient condition for optimality [2]. Note that (4) is not the most general form of the HJB. Various extensions including the case with disturbances are possible.

To solve the optimization problem (4), the first step is to perform the indicated minimization which leads to a control law of the form:

$$u^* = \psi(\frac{\partial V}{\partial x}, x) \quad (6)$$

The second step is to substitute (6) into equation (4), which becomes:

$$\frac{\partial V}{\partial x}(f(x) + g(x)\psi) + \psi^T\psi + q(x) = 0 \quad (7)$$

In the absence of constraints, the optimal control law (6) is given by:

$$u^* = -\frac{1}{2}g^T(x)\frac{\partial V^T(x)}{\partial x} \quad (8)$$

and the corresponding HJB equation is:

$$\frac{\partial V(x)}{\partial x}f(x) - \frac{1}{4}\frac{\partial V(x)}{\partial x}g(x)g^T(x)\frac{\partial V^T(x)}{\partial x} + q(x) = 0. \quad (9)$$

In the constrained case, the optimal control law (6) is obtained by solving the Kuhn-Tucker conditions [2] for the optimization problem (4):

$$2u^{*T} + \frac{\partial V(x)}{\partial x}g(x) + \lambda^T\frac{\partial c(x,u^*)}{\partial u} = 0 \quad (10)$$

$$\lambda^T c(x, u^*) = 0 \quad (11)$$

$$\lambda \geq 0 \quad (12)$$

Here, λ denotes the Lagrange multiplier.

Once the optimal control u^* is found, substitution in (7) for ψ results in the constrained version of the HJB equation.

The HJB equation is difficult to solve analytically, in particular there is no efficient algorithm available when the problem dimension is high. Thus reducing the optimal control problem to the HJB cannot be viewed as a general, practical method. In addition, the presence of constraints makes the optimal control problem even more difficult.

3 Converse Constrained Optimal Control Problem

The basic principle behind the *"converse problem (Co-HJB)"* [1] is to solve the HJB "backwards" to generate examples which may be highly nonlinear with possibly high state dimension, yet for which we know the optimal controller. Extending the original converse approach to the constrained case, the *"converse constrained optimal control problem"* is introduced and its solution is demonstrated for single input systems with input saturation constraints.

Def 2 ***Converse Constrained Optimal Control Problem (CoCHJB):*** *Given a performance objective (3) and a value function V, find the class of nonlinear systems (1) such that V corresponds to the solution of the constrained optimal control problem (4).*

In order for there to exist a solution to the CoCHJB, some restrictions on V and q are necessary. These admissibility conditions are derived in [4].

Solving the CoCHJB is equivalent to solving the HJB with known value function V. Since knowledge of V determines the optimal controller $u^* = \psi(\frac{\partial V}{\partial x}, x)$ through equations (10)–(12), the HJB (7) becomes an algebraic equation in f and g. In this way nonlinear dynamics and optimal controllers are produced. Although useless for design, the CoCHJB is great for constructing benchmark examples. It is easy to use, computational tractable, and can generate essentially all possible nonlinear optimal control problems, discrete or continuous. Since the optimal control is known, the CoCHJB provides an efficient procedure to validate the performance of different nonlinear control techniques in the presence of constraints.

3.1 Single Input Systems With Saturation Constraints

Consider a nonlinear system (1) with scalar input u, and saturation constraint:

$$|u| < \alpha \tag{13}$$

which is equivalent to:

$$c(x,u) = \begin{bmatrix} u-\alpha \\ -u-\alpha \end{bmatrix} \tag{14}$$

Then the constrained optimal u^*, according to (10)–(12) is characterized by:

$$2u^* + \frac{\partial V}{\partial x} g + \lambda_1 - \lambda_2 = 0 \tag{15}$$

$$\lambda_1(u^* - \alpha) + \lambda_2(-u^* - \alpha) = 0 \tag{16}$$

$$\lambda_1 \geq 0 \tag{17}$$

$$\lambda_2 \geq 0 \tag{18}$$

Solving the above equations for $u^* = \psi(\frac{\partial V}{\partial x}, x)$ leads to the following result:

The optimal controller for a single input system (1) under saturation (13) is given by:

$$u^* = sat_\alpha(-\frac{1}{2} g^T(x) \frac{\partial V^T(x)}{\partial x}), \tag{19}$$

where $sat_\alpha(\cdot)$ denotes the saturation operator:

$$sat_\alpha(z) = \begin{cases} \alpha & z > \alpha \\ z & -\alpha \leq z \leq \alpha \\ -\alpha & z < -\alpha \end{cases} \tag{20}$$

The corresponding HJB is given by:

$$\frac{\partial V}{\partial x}(f + gu^*) + u^{*T}u^* + q(x) = 0. \tag{21}$$

Equation (21), when thought of in terms of f and g, can be used to characterize all possible solutions of the CoCHJB. This is used later to generate illustrative examples for which the constrained optimal controller (19) is known.

4 Constrained Nonlinear MPC Formulation

An MPC algorithm is conventionally formulated in discrete time by solving an on–line open loop finite horizon optimal control problem at each sampling time k, respecting the following objective function $\mathcal{J}$:

$$\min_{\mathcal{U}_k} \left[\sum_{i=1}^{P} q(x_{k+i}) + \sum_{i=0}^{M-1} u_{k+i}^T u_{k+i} \right] \tag{22}$$

subject to:

$$c(x,u) \leq 0 \tag{23}$$

$$\dot{x} - f(x) - g(x)u = 0 \tag{24}$$

Here:

- x_{k+i} : predicted state vector at time $k+i$ based on the states x_k at time k, obtained by using prediction model (24)
- $\mathcal{U}_k$: control sequence u_{k+i}, $i = 0, ..., M-1$ computed by the optimization algorithm at time k; $u_{k+i} = 0$ for $i \geq M$; u_k is the control move to be implemented at time k.
- M, P : *input* (*control*) and *output* (*prediction*) horizon, respectively; $M \leq P$.

Since constraints are directly included in the optimization, MPC is considered as the only methodology which deals with both input and output constraints explicitly. Although the optimization problem solved at time k results in an optimal sequence of M present and future control moves, only the first control move, u_k, is implemented on the real plant over the time $[k, k+1]$. At time step $k+1$, the horizons M and P are shifted ahead by one step, and a new optimization problem with new initial condition x_{k+1}, is solved. This kind of implementation is known as the *receding* or *moving horizon* approach.
Determining an optimal open-loop control for a given initial state is a relatively simple task, and makes the MPC algorithms attractive in many situations. By repeatedly solving an on line open-loop optimization for the current state, and applying the minimizing control for a short time before repeating the procedure, MPC does not require the construction of diffeomorphic state-feedback transformations and avoids solving the HJB equation. Essentially, instead of being determined for each state, the value function is calculated for the sequence of states actually encountered.

Various MPC–based methods have been employed to solve the constrained control problem for nonlinear systems. Basically, three approaches exist:
1. Nonlinear MPC (NLMPC),
2. MPC in combination with feedback linearization,
3. Approximate MPC techniques (gain-scheduled linear MPC).

Since NLMPC and MPC+FL will be applied to the examples presented in the next section, some of the basic issues related to these methods will be briefly discussed.

4.1 NLMPC Technique

The standard nonlinear MPC (NLMPC) technique and its modifications use nonlinear models for prediction according to (22)-(24), which generally result in non-convex nonlinear programs (NLP), even if the cost function and constraint sets are convex. Therefore, finding a global optimum is a very difficult and computationally very demanding task. In other words, non-convexity makes the solution of NLMPC uncertain.

4.2 MPC+FL Technique

The MPC+FL technique (Fig.1) attempts to gain computational efficiency by feedback linearizing the plant and restating the MPC problem in the new linearized coordinates [3]. The use of a linearized prediction model may reduce the NLP in the NLMPC formulation to a quadratic program (QP) for the FL system, which can have dramatic effects on efficiency. The basic difficulty with the MPC+FL method is due to the fact that the original optimization problem for the nonlinear system subject to linear constraints on the input u, given by: $-\alpha \leq u \leq \alpha$, has been transformed into an optimization problem for a linear system subject to MPC constraints on the new input v, described by: $-\alpha \leq \xi(x) + \eta(x)v \leq \alpha$, which are *state dependent* and generally *nonlinear*. This new optimization problem is not necessarily easier to solve, unless the nonlinear constraints are convex. It is also important to note that the objective function used in MPC+FL is given in terms of the new control variable v:

$$\mathcal{J}_{fl} = \min_{\mathcal{V}_k} \left[\sum_{i=1}^{P} q(x_{k+i}) + \sum_{i=0}^{M-1} v_{k+i}^T R v_{k+i} \right] \qquad (25)$$

where it has been assumed that there were no changes in the state coordinates due to FL.

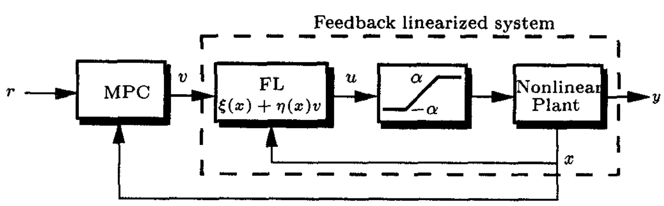

Figure 1: MPC+FL control structure

5 Examples

It is easy to construct examples which illustrate the MPC implementation issues previously described. In particular, simple (even 1-dimensional) examples can be generated where there are local optima with arbitrarily higher cost than the global optimum [4]. Also it is easy to construct examples where ignoring the possible effects of future disturbances in choosing the control action, as is common in MPC, leads to arbitrarily bad results. This is in contrast with the linear case, where the gap can be bounded. All these issues are analyzed in detail and some instructive examples are presented in [4].

In this section various results obtained by applying both NLMPC and MPC+FL to examples generating by solving the converse constrained optimal control problem are presented and evaluated by comparison with the known constrained optimal controller, as well as with other control techniques. As a benchmark, we consider the single input 2D oscillator subject to saturation constraints.

5.1 System description: 2-D Oscillators

Consider the following system:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \hat{f}(x) + \hat{g}(x)u, \end{cases} \qquad (26)$$

subject to the saturation constraint $|u| < \alpha$. Choosing $q(x) = x_2^2$, the performance objective is:

$$J(u) = \int_0^\infty (x_2^2 + u^2)dt. \qquad (27)$$

The solution to the CoCHJB for the 2D oscillator is derived according to the results described in Section 3.1. Assuming the known optimal value function V, equations (19) and (21) yield the constrained optimal control law for the 2D oscillator:

$$u^* = sat_\alpha(-\frac{1}{2}\hat{g}V_2) \qquad (28)$$

and corresponding dynamics:

$$\hat{f} = -\frac{(V_2\hat{g} + u^*)u^* + V_1 x_2 + q}{V_2} \qquad (29)$$

Here $V_1 = \frac{\partial V}{\partial x_1}$ and $V_2 = \frac{\partial V}{\partial x_2}$ have been introduced for notational convenience.

5.2 Simulation results

5.2.1 Example 1: Consider a system of the form (26) with input saturation constraint $|u| \leq \alpha$. Choosing the value function:

$$V = -2(1 + x_2)e^{-x_2} + x_1^2 + 2 \qquad (30)$$

we have $V_1 = 2x_1, \; V_2 = 2x_2 e^{-x_2}$.

Taking $\hat{g}$ to be

$$\hat{g} = e^{2x_1 + 2x_2}$$

determines $\hat{f}$ from equation (29):

$$\hat{f} = -\frac{(2x_2 e^{2x_1 + x_2} + u^*)u^* + 2x_1 x_2 + x_2^2}{2x_2 e^{-x_2}} \qquad (31)$$

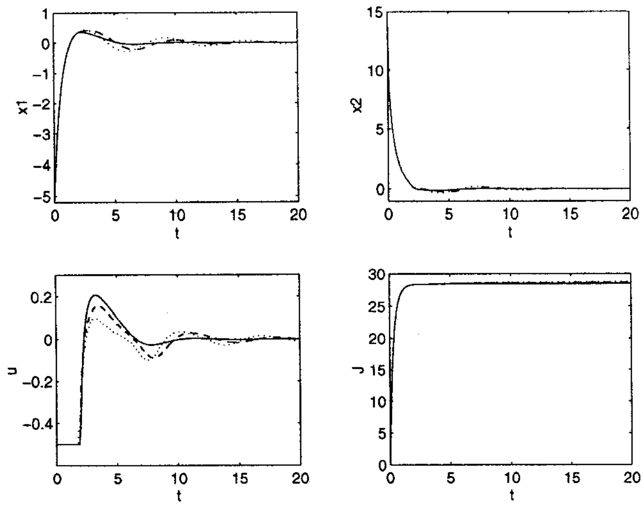

Figure 2: Comparison *Constrained Optimal Controller* (solid) vs. *Nonlinear MPC* (dashed: M=P=5, dot: M=P=3)

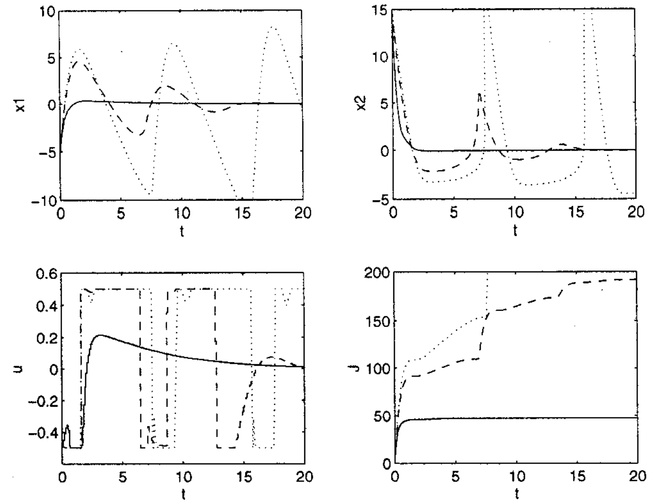

Figure 3: Comparison *FL* (dot) vs. *MPC+FL* (solid: R=0.1, dashed: R=1)

where u^* is given by:

$$u^* = sat_\alpha(-e^{2x_1+x_2}x_2) \tag{32}$$

The simulation results which follow have been obtained for the initial condition $x_0 = [-5.15\ \ 15]$ and saturation $\alpha = 0.5$.

Figure 2 shows a comparison between the constrained optimal nonlinear controller (32) and NLMPC, implemented for two different horizon lengths. State responses x_1 and x_2 are given in the top figures, and control action u and cost J in the bottom figures. The performance obtained by NLMPC is nearly identical to that of the constrained optimal controller u^*. This is not surprising since NLMPC includes the constraint directly in its formulation, and deviations from the optimal performance are most likely attributable to the finite horizon lengths or possible failures in the on-line optimization. Also as expected, the larger the MPC horizon used, the nearer the cost is to the optimal (see also Table 1). However, the price for such near optimal performance is the large computational burden associated with NLMPC. Therefore, an important question is whether the expensive computations are justified and how well other less demanding control techniques will fair when applied to this system.

An obvious alternative choice is a controller designed by feedback linearization (FL). An FL controller designed to have the same closed loop dynamics as the locally optimal LQR solution for the system linearized at the origin is given by: $u_{fl} = -(1/g)(f + x_1 + x_2)$. Note that this design does not take the constraint into account. For the same initial condition and saturation level as was used previously, the FL controller applied to the system with saturation results in instability, as shown in Fig. 3 (dotted line). The unstable performance is due to the fact that linearity in the FL controlled closed loop is not preserved during saturation. More specifically, the FL controller fails to take advantage of its full control potential at the time the system is sensitive to the control action, i.e. when $\hat{g}$ is large. Later, saturation and small $\hat{g}$ conspire to diminish the attempts of any control action. In contrast to the FL controller, the optimal controller recognizes that saturation will limit its ability to influence the dynamics in certain portions of the state space, and responds accordingly by saturating fully when in the appropriate regions. In particular, this can be seen at the initial condition where $\hat{g} = e^{19.7}$, and the difference between $u_{fl} = -0.4317$ and $u^* = -0.5$ has a tremendous effect on the closed loop dynamics ($\hat{f} + \hat{g}u_{fl} = -9.85$ vs. $\hat{f} + \hat{g}u^* = -2.45 \times 10^7$).

In order to avoid the difficulties due to saturation, MPC is applied to the FL system according to the MPC+FL formulation described in Section 4.2. Since the constraint is now explicitly respected, the FL system is protected from instability. Despite the fact that the objective (25) minimized in MPC+FL is not the same as the original objective (22) used in NLMPC, satisfactory performance is achieved (Figure 3). To evaluate the MPC+FL results, the cost measured by the original objective (22) is used, as shown in Figure 3 and given in Table 1. Simulation results clearly indicate that the choice of control weight R in the MPC+FL objective (25) can be crucial, i.e. careful tuning and an appropriate choice of R can lead to performance close to the optimal. In this example, decreasing the weight R makes the performance better.

Table 1

Controller	Cost
OPTIMAL	28.5225
NLMPC ($M = P = 5$)	28.6208
NLMPC ($M = P = 3$)	28.7281
MPC+FL ($M = P = 5, R = 0.1$)	47.2107
MPC+FL ($M = P = 5, R = 1$)	191.8618
FL	unstable

This example clearly shows the necessity of using MPC-based techniques (NLMPC and MPC+FL) for such types of systems. In addition, due to its computational efficiency, MPC+FL may in many cases be the preferable strategy.

5.2.2 Example 2: In this example, it is shown that saturating a controller designed by feedback linearization may actually make its performance be closer to the optimal.

Using $V = x_1^2 + x_2^2$ as the value function, and choosing $\hat{g} = e^{x_2}$, the following optimal controller and dynamics are obtained:

$$u^* = sat_\alpha(-x_2 e^{x_2}) \tag{33}$$

$$\hat{f} = -\frac{(2x_2 e^{x_2} + u^*)u^* + 2x_1 x_2 + x_2^2}{2x_2} \tag{34}$$

In contrast to the FL controller in Example 1, in this example the saturation constraint actually prevents the FL controller from taking the "wrong" action. The unsaturated FL controller differs most greatly from the optimal control by expending large control effort when the optimal control uses only a small amount (recall for Example 1 that the optimal control was saturated when the FL controller was not). The constraint protects the FL controller against making this mistake by restricting any attempt to use an undue amount of control effort.

In this example the performance obtained using the MPC+FL scheme is actually worse than those obtained by just using the stabilized FL control. This is not completely unexpected considering the fact that MPC+FL does not use the original performance objective.

The following table summarizes the results from Example 2 obtained for $\alpha = 0.2$ and initial condition $x = [5 \quad 0]$, with $M = 2,\ P = 20$:

Table 2

Controller	Cost
OPTIMAL	25.00
NLMPC	25.17
FL	25.0098
MPC+FL	27.4182

Based on the discussion above, we can conclude that for systems which behave similar to the one explored here, applying computationally expensive MPC algorithms is unnecessary, even in the presence of constraints. More details about this and other examples investigated can be found in [4].

6 Conclusion

Various examples for which the constrained optimal controller is known were generated using the constrained converse strategy, and used to test and validate the performance of different control techniques in the presence of constraints. MPC, which explicitly handles constraints, worked well on these examples with cost close to optimal. Practical implementation of MPC is dominated by the computational burden both for finding the control action online and for evaluating the scheme off-line by simulation. For a class of FL nonlinear systems, MPC+FL strategy appears to be attractive due to its relative computational efficiency.

The examples described here are somewhat extreme and not necessarily representative of practical applications, yet they fulfill the aim of creating low order nonlinear problems that illustrate specific phenomena, and help to deepen our understanding of nonlinear control.

7 Acknowledgment

This work was performed during the first author's visit to Caltech. Both authors would like to thank John Doyle for his extensive and useful criticism of this entire paper. Even this section couldn't have been written without his help. The research was supported by NSF, AFOSR, and Rockwell.

References

[1] John C. Doyle, "Nonlinear games: examples and counterexamples", in *Proceedings of the 35th IEEE Conference on Decision and Control*, 1996.

[2] A. E. Bryson and Y. Ho. *Applied Optimal Control*, Hemisphere Publ. Corp., Washington D.C., 1975.

[3] V. Nevistić and M. Morari, "Constrained control of feedback-linearizable systems", In *Proceedings of the European Control Conference*, pp. 1726–1731, 1995.

[4] V. Nevistić and J. Primbs, "Constrained control of nonlinear systems: A converse hjb approach", see *http://hot.caltech.edu/~doyle, also to be published as CDS technical memo*, California Institute of Technology, Pasadena, CA 91125, 1996.