

Nonlinear controllers for non-integrable systems:
the Acrobot example

John Hauser*
Department of Electrical Engineering – Systems
University of Southern California
Los Angeles, CA 90089-0781
hauser@nyquist.usc.edu

Richard M. Murray†
Electronics Research Laboratory
University of California
Berkeley, CA 94720
murray@united.berkeley.edu

1 Introduction

Recent developments in the theory of geometric nonlinear control provide powerful methods for controller design for a large class of nonlinear systems. Many systems, however, do not satisfy the restrictive conditions necessary for either full state linearization [6, 5] or input-output linearization with internal stability [2]. In this paper, we present an approach to controller design based on finding a linearizable nonlinear system that well approximates the true system over a desirable region. We outline an *engineering* procedure for constructing the approximating nonlinear system given the true system. We demonstrate this approach by designing a nonlinear controller for a simple mechanical system patterned after a gymnast performing on a single parallel bar.

There has been considerable work in the area of system approximation including Jacobian linearization, pseudo-linearization [9, 10], approximation with a nonlinear system [8], and extended linearization [1]. Much of the work on system approximation has been directed toward analysis and the development of conditions that must be satisfied by the approximate systems rather than on the explicit construction of such approximations. Notable exceptions include the standard Jacobian approximation and the recent work of Krener using polynomial system approximations [7]. Wang and Rugh [10] also provide an approach for constructing configuration scheduled linear transformations to pseudo-linearize the system (note that this approach provides a *family* of approximations rather than a *single* system approximation). Rather than using polynomial systems or families of linear systems to approximate the given system, we approximate the given nonlinear system with a single nonlinear system that is full state linearizable.

We use as a guiding example the problem of controlling the *acrobot* (for acrobatic-robot) shown in Figure 1. The acrobot is a highly simplified model of a human gymnast performing on a single parallel bar. By swinging his legs (a rotation at the hip) the gymnast is able to bring himself into a completely inverted position with his feet pointing upwards and his center of mass above the bar. The acrobot consists of a simple two link manipulator operating in a vertical plane. The first joint (corresponding to the gymnast's hand sliding freely on the bar) is free to rotate. A motor is mounted at the second joint (between the links) to provide a torque input to the system (corresponding to the gymnast's ability to generate torques at the hip). Note that the acrobot is a good model for many under-actuated mechanical systems, such as unicycles and walking machines, where *balance* must be maintained while trying to accomplish the assigned task. A life size acrobot is currently being instrumented for experimentation at U.C. Berkeley.

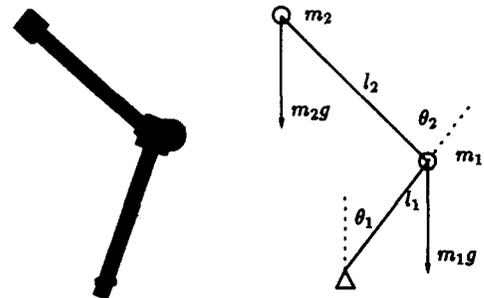


Figure 1: Acrobot: an acrobatic robot. Patterned after a gymnast on a parallel bar, the acrobot is only actuated at the middle (hip) joint; the first joint, corresponding to the gymnast's hands on the bar, is free to spin about its axis.

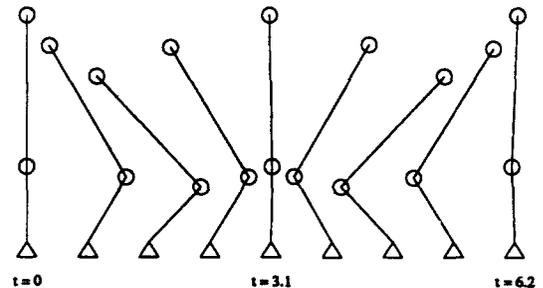


Figure 2: Motion of the acrobot along the manifold of inverted equilibrium positions.

The eventual goal in controlling this system is to precisely execute realistic gymnastic routines. Our modest initial goal is to understand and design controllers capable of system control in a neighborhood of the manifold of inverted equilibrium positions. That is, we would like to have the acrobot follow a smooth trajectory while inverted such as that shown in Figure 2.

2 System Dynamics

The dynamics of the acrobot have the general form

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) + G(\theta) = \begin{pmatrix} 0 \\ \tau \end{pmatrix} \quad (1)$$

where $\theta = (\theta_1, \theta_2)$ are the joint angles measured relative to the vertical and the first link, respectively, M is the (uniformly positive definite) inertia tensor, C contains the Coriolis and centrifugal forces, G contains the effects of gravity, and τ is the torque applied between the first and second links. For

*Fred O'Green Assistant Professor of Engineering
†Research supported in part by an IBM Manufacturing fellowship

simplicity of presentation, we will consider an acrobot with specific (and somewhat unrealistic) parameters. Modeling the acrobot as in Figure 1 with $m_1 = m_2 = 8$, $l_1 = 1/2$, $l_2 = 1$, and $g = 10$, the quantities in (1) are given by

$$\begin{aligned} M(\theta) &= \begin{bmatrix} 12 + 8 \cos \theta_2 & 8 + 4 \cos \theta_2 \\ 8 + 4 \cos \theta_2 & 8 \end{bmatrix} \\ C(\theta, \dot{\theta}) &= \begin{bmatrix} -4\dot{\theta}_2(2\dot{\theta}_1 + \dot{\theta}_2) \sin \theta_2 \\ 4\dot{\theta}_1^2 \sin \theta_2 \end{bmatrix} \\ G(\theta) &= \begin{bmatrix} -80(\sin \theta_1 + \sin(\theta_1 + \theta_2)) \\ -80 \sin(\theta_1 + \theta_2) \end{bmatrix} \end{aligned} \quad (2)$$

Note that, due to the presence of rotary joints, these dynamics are highly nonlinear and contain important trigonometric terms.

As noted above, we are interested in controlling the acrobot in the neighborhood of the manifold of inverted equilibrium points. This manifold can be parameterized by θ_2 . Physically, the acrobot will be in equilibrium whenever the center of mass of the system is directly above the first joint. Indeed, if we choose $x_1 = \theta_2$ and $x_2 = \theta_1 + \theta_2/2$ and set $x_3 = \dot{x}_1$, $x_4 = \dot{x}_2$, the equilibrium manifold will consist of points of the form $(x_1, 0, 0, 0)$ with $x_1 \in (-\pi, \pi)$. The fact that this balancing set of coordinates is a linear transformation of the natural coordinates is due specifically to the special choice of parameters given above—almost any other choice will require a complicated nonlinear transformation.

Using the balancing coordinates and defining $u := \tau$, we may write (1) in the form

$$\dot{x} = f(x) + g(x)u. \quad (3)$$

As a tedious calculation verifies, the linearized system is completely controllable at any equilibrium point for $x_1 = \theta_2 \in (-\pi, \pi)$ and can therefore be controlled locally using a linear state feedback controller. By using nonlinear control techniques we attempt to improve on the performance of the linear compensator.

3 Approximate Integration of the System

Given the objective of controlling the acrobot in the neighborhood of the equilibrium manifold, we consider first the possibility of exactly linearizing the system from input to state [6, 5]. Since the acrobot is controllable through its linearization at inverted equilibrium points with $\theta_2 \in (-\pi, \pi)$, the system can be (locally) linearized if and only if the distribution

$$\Delta = \text{span} \{ g \quad ad_1 g \quad ad_1^2 g \} \quad (4)$$

can be completely integrated. The expression $ad_1^2 g$ is the iterated Lie bracket $[f, \dots [f, g]]$. We know by the Frobenius Theorem that this cannot be accomplished since, for the acrobot, Δ is not involutive. Failing this, we take an engineering approach and try to approximately integrate Δ . That is, we seek vector fields \tilde{f} and \tilde{g} , close to f and g , and a function \tilde{h} such that

$$\frac{\partial \tilde{h}}{\partial x} \begin{bmatrix} \tilde{g} \quad ad_1 \tilde{g} \quad ad_1^2 \tilde{g} \end{bmatrix} = 0. \quad (5)$$

As is usual in approximation, we require that the approximate vector fields agree with the original vector fields up to higher order terms in x . Furthermore, we require that this higher order property be satisfied around each point on the equilibrium

manifold. With this additional requirement, we expect the eventual closed loop system to behave in a reasonable manner in a neighborhood of the equilibrium manifold rather than only in a neighborhood of a single equilibrium point.

Note that a function $\tilde{h}(\cdot)$ satisfying (5) can be thought of as an output that effectively gives the system a relative degree of four. More importantly, the functions \tilde{h} , $L_f \tilde{h}$, $L_f^2 \tilde{h}$, $L_f^3 \tilde{h}$ define a coordinate change that, along with a state feedback, will put the true system into an approximate Brunovsky canonical form (i.e., a chain of integrators). This is, perhaps, the simplest way to express the constraint that the system follow the given differential equation. By forcing the system to follow a desired output, we can effectively steer the system through the state space.

To find a suitable output function \tilde{h} , we first note that every system approximation satisfying the higher order property (uniformly on the equilibrium manifold) must be tangent to the Jacobian approximation at each equilibrium point. If $x_2 = x_3 = x_4 = 0$ on the equilibrium manifold, then the system matrices of the Jacobian linearization (parameterized by x_1) are given by

$$\begin{aligned} A(x_1) &= Df(x) + u_e(x_1)Dg(x)|_{x=(x_1, 0, \dots, 0)} \\ b(x_1) &= g(x)|_{x=(x_1, 0, 0, 0)} \end{aligned} \quad (6)$$

where $u_e(x_1)$ is such that $f(x) + g(x)u_e(x_1) = 0$ for each equilibrium point $x = (x_1, 0, 0, 0)$. Candidate functions $\tilde{h}(x)$ must therefore satisfy the linearized version of (5), namely,

$$c(x_1) \begin{bmatrix} b(x_1) \quad A(x_1)b(x_1) \quad A(x_1)^2 b(x_1) \end{bmatrix} = 0 \quad (7)$$

where $c(x_1) = \frac{\partial \tilde{h}}{\partial x}(x)|_{x=(x_1, 0, 0, 0)}$. Note that only the direction of $c(x_1)$ is fixed, the magnitude can be chosen at will and can have a significant affect on the resulting approximation.

For the acrobot with the parameters given, $c(x_1)$ is given by

$$c(x_1) = \begin{bmatrix} 1 & 6 + 4 \cos x_1 & 0 & 0 \end{bmatrix}. \quad (8)$$

Therefore, one of many possible output functions for this system is given by

$$\tilde{h}(x) = x_1 + (6 + 4 \cos x_1)x_2 \quad (9)$$

It is interesting to note that any function $\tilde{h}(x)$ with differential $c(x)$ satisfying (7) and having full rank at each point on the equilibrium manifold actually parameterizes the equilibrium manifold [3]. That is, $\tilde{h}(x^1) \neq \tilde{h}(x^2)$ for equilibrium points x^1 , x^2 where the system is controllable.

With $\tilde{h}(\cdot)$ in hand, we take the approach introduced in [4] and proceed to approximately input-output linearize the acrobot system using this output function. Basically, one constructs a nonlinear change of coordinates $\Phi(x)$ so that the system expressed in the new coordinates $\xi = \Phi(x)$ looks like a chain of integrators perturbed by higher order terms in x and u . The nonlinear system approximation is then found by neglecting the higher order terms and expressing the resulting system in the original coordinates.

In this case, we are careful to make this approximation uniform with respect to the equilibrium manifold. Since we have initially transformed the system so that each point on the equilibrium manifold is such that $x_2 = x_3 = x_4 = 0$, we can easily discard terms that are uniformly higher order.

The procedure is constructive in that each $\phi_i(\cdot)$ is selected to match $L_f \phi_{i-1}$ to higher order terms. That is,

$$\phi_{i+1} = L_f \phi_i - \phi_i^i(x) \quad (10)$$

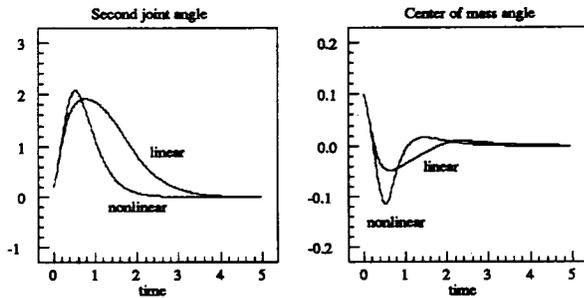


Figure 3: Simulation results for acrobot: regulation to a point

where $\psi_i^1(x)$ is higher order. The choice of $\psi_i^1(\cdot)$ is restricted to higher order terms but is otherwise up to the designer. It is at this point where the designer's *engineering* intuition comes in to play. The choice of the neglected terms does not change the nature of the local (linear) approximation but can greatly affect the performance and stability in the large of the eventual closed loop system. Specifically, we set $\xi_1 = \phi_1(x) = \tilde{h}(x)$. Then, along the system trajectories, we have (defining $\phi_i(\cdot)$ recursively)

$$\begin{aligned}\dot{\xi}_1 &= \underbrace{x_3 + x_4(6 + 4 \cos x_1)}_{\xi_2 = \phi_2(x)} + \underbrace{(-4x_2x_3 \sin x_1)}_{\psi_1^1(x)} \\ \dot{\xi}_2 &= \underbrace{40(\sin(\frac{x_1}{2} + x_2) - \sin(\frac{x_1}{2} - x_2))}_{\xi_3 = \phi_3(x)} \\ \dot{\xi}_3 &= \underbrace{20 \left[(x_3 + 2x_4) \cos(\frac{x_1}{2} + x_2) - (x_3 - 2x_4) \cos(\frac{x_1}{2} - x_2) \right]}_{\xi_4 = \phi_4(x)} \\ \dot{\xi}_4 &= d(x) + a(x)u\end{aligned}\quad (11)$$

where $a(x)$ is nonzero for each $x = (x_1, 0, 0, 0)$ with $x_1 \in (-\pi, \pi)$.

The nonlinear system approximation is then given by neglecting $\psi_1^1(x)$ in (11). In x coordinates, this corresponds to subtracting $\Delta f(x) = [D\Phi(x)]^{-1}(\psi_1^1(x), 0, 0, 0)^T$ from $f(x)$ which is, in fact, a very complicated expression. This points out the difficulty of modifying the f and g vector fields as given to obtain an integrable distribution. The above approach is, to the contrary, quite straightforward and yields the desired approximation.

4 Simulation Results

To illustrate the benefits of the approach outlined here, we present simulation results for the acrobot using linear and nonlinear controllers. The linear controller was constructed by linearizing the system about $\theta_1 = \theta_2 = 0$. Both controllers were designed so that the closed loop poles are all at -3.5 .

For regulation to an equilibrium point, the system performance is similar for the two controllers. The region of attraction is not noticeably different though the linear system converges somewhat more slowly. This is due to the fact that the linear controller sees a reduced effective gain at system configurations away from the nominal operating point. In contrast, the nonlinear controller provides *instantaneous gain scheduling* at each position near the equilibrium manifold. This phenomenon is clearly shown in figure 3 where the initial position was given by $\theta_1 = 0$, $\theta_2 = .2$ and regulation to $\theta_1 = \theta_2 = 0$ was desired.

A more striking difference in controller performance is apparent when we attempt to track a trajectory. As evident from

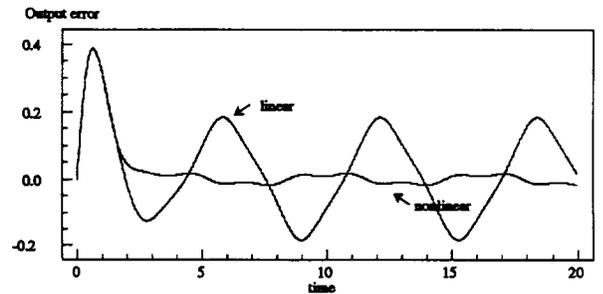


Figure 4: Simulation results for acrobot: tracking a trajectory

Figure 4, the nonlinear controller had significantly better output tracking capability. A large part of this linear controller error results because a strictly linear controller cannot calculate the input necessary to hold the nonlinear system at more than one operating point (this requires a nonlinear function or table lookup). The nonlinear controller, however, directly provides the instantaneous *nonlinear trim* needed at each different system configuration along the equilibrium manifold.

Note that Figure 2 simply contains a sequence of configurations from the nonlinear controller simulations. Computer graphics animations of these motions have been produced to effectively demonstrate this system.

References

- [1] W. T. Baumann and W. J. Rugh. Feedback control of nonlinear systems by extended linearization. *IEEE Transactions on Automatic Control*, 31:40-46, 1986.
- [2] Christopher I. Byrnes and Alberto Isidori. Local stabilization of minimum-phase nonlinear systems. *Systems and Control Letters*, 11:9-17, 1988.
- [3] John Hauser. Nonlinear control via uniform nonlinear system approximation. to appear.
- [4] John Hauser, Shankar Sastry, and Petar Kokotović. Nonlinear control via approximate input-output linearization: the ball and beam example. In *28th Conference on Decision and Control*, pages 1987-1993, Tampa, FL, 1989.
- [5] L. R. Hunt, Renjeng Su, and George Meyer. Global transformations of nonlinear systems. *IEEE Transactions on Automatic Control*, AC-28:24-31, 1983.
- [6] Bronislaw Jakubczyk and Witold Respondek. On linearization of control systems. *Bulletin de L'Academie Polonaise des Sciences, Série des sciences mathématiques*, XXVIII:517-522, 1980.
- [7] A. J. Krener, S. Karahan, M. Hubbard, and R. Frezza. Higher order linear approximations to nonlinear control systems. In *26th Conference on Decision and Control*, pages 519-523, 1987.
- [8] Arthur J. Krener. Approximate linearization by state feedback and coordinate change. *Systems and Control Letters*, 5:181-185, 1984.
- [9] C. Reboulet and C. Champetier. A new method for linearizing non-linear systems: the pseudolinearization. *International Journal of Control*, 40:631-638, 1984.
- [10] Jianliang Wang and Wilson J. Rugh. On the pseudolinearization problem for nonlinear systems. *Systems and Control Letters*, 12:161-167, 1989.