Symbolic Computing Tools for Nonsmooth Dynamics and Control

C. Teolis
Techno-Sciences, Incorporated, 10001 Derekwood Lane, Suite 204
Lanham, MD 20706

H. G. Kwatny
Department of Mechanical Engineering & Mechanics, Drexel University
Philadelphia, PA 19104

M. Mattice
Advanced Drives and Weapon Stabilization Lab, U. S. Army ARDEC
Picatinny Arsenal, NJ 07806

Abstract In this paper we describe a set of symbolic computing tools for variable structure control system design. The software implements all aspects of a design approach for input-output linearizable systems. It is part of a comprehensive symbolic computing environment for nonlinear and adaptive control system design that has been under continuous development for several years. Current work is focused on plants with nondifferentiable nonlinearities. Some preliminary results are reported.

1 Introduction

The purpose of this paper is twofold. First, to describe a set of symbolic computing tools developed to assist in the design and implementation of variable structure control systems. The tools enable the efficient design of sliding surfaces and reaching controllers including the inclusion of 'smoothing' and 'moderating' functions and the assembly of C-source code for simulation and real time implementation. These functions extend the capabilities of the symbolic modeling and control design software described in [1] and elsewhere.

The second purpose is to introduce a new backstepping methodology for systems with uncertain nondifferentiable nonlinearities. The key innovations in our approach are (1) that the states are grouped depending on where an uncertainty enters the system and the robustification is attempted only where the uncertainty is identified, and (2) that the control designed at each step is a variable structure control.

The variable structure methods we have implemented are developed in [2-4]. These references deal with variable structure control system design for smooth affine systems that are feedback linearizable in the input-output sense. Such systems are of the form:

$$\begin{aligned} \dot{x} &= f(x) + G(x)u \\ y &= h(x) \end{aligned} \tag{1}$$

where *f, G, h* are sufficiently smooth and satisfy certain feedback linearizability conditions [5]. All of the basic functions needed for design including reduction to regular form (as described in [1]) and computation of the zero dynamics (as previously reported in [6]), as well as functions for designing sliding surfaces and switching controllers, have been integrated into a convenient *Mathematica* package[1]. Ongoing work is focused on extending these techniques to plants containing hard nonlinearities such as dead zone, backlash, hysteresis and coulomb friction. To do this has required extending *Mathematica's* facilities for working with nondifferentiable nonlinear functions. Our examples describe applications to friction compensation.

In Section 2 we summarize the methods and computations that we have implemented. We include some preliminary remarks concerning nonsmooth plant dynamics and we briefly discuss chattering reduction techniques. Section 3 describes and illustrates the symbolic computing tools. A very simple system with nonsmooth friction is used for illustrative purposes. The effects of control smoothing and moderation are illustrated. Section 4 describes ongoing work involving plants with nondifferentiable nonlinearities. A simple example is given which demonstrates the problems that can occur when applying methods designed for smooth system to those with nonsmooth nonlinearities by simply approximating the nonsmooth nonlinearities by smooth functions. A backstepping approach to the variable structure control design is shown to solve the problem in this simple case. In Section 5, the symbolic computing tools are used to design a friction compensating slewing controller for the US Army Apache Helicopter 30-mm chain gun. Simulation results are given that show the control robustness to parameter variation. Some concluding remarks are given in Section 6.

2 Variable Structure Control Design

There are 2 basic steps to designing a variable structure control. The first is the design of the sliding control or equivalently the sliding surface. The second is the design of the reaching or switching control. The system is typically reduced to normal, or regular, form before the design begins. Also, in order to avoid exciting higher order unmodeled dynamics, 'smoothing' and 'moderating' functions are used to reduce chattering. In this section, these methods are summarized. In addition, a simple example is given with nonsmooth friction and local asymptotic stability of the variable structure control is proven.

2.1 Normal Form

Denote the $k^{\underline{\text{th}}}$ Lie (directional) derivative of the scalar function $\phi(x)$ with respect to the vector field $f(x)$ by $L_f^k(\phi)$. Now, by successive differentiation of the outputs y in (1) we arrive at the following definitions for the list of integers r_i, the column vector $\alpha(x)$ and the matrix $\rho(x)$:

[1] More information can be found at the website: www.technosci.com.

$$r_i := \inf\{k \mid L_{g_j}(L_f^{k-1}(h_i)) \neq 0 \ \ for\ at\ least\ one\ j\}$$

$$\alpha_i(x) := L_f^{r_i}(h_i),\ i = 1,..,m$$

$$\rho_{ij}(x) := L_{g_j}(L_f^{r_i-1}(h_i)),\ i,\ j = 1,..m$$

Also define the partial state transform $x \to z \in R^r, \quad r = r_1 + \ldots + r_m \leq n$ as

$$z := \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_m \end{bmatrix},\ z_i \in \mathrm{R}^{r_i}, i = 1,...,m \tag{2a}$$

where

$$z_i^k(x) = L_f^{k-1}(h_i),\ k = 1,..,r_i \ \ and \ \ i = 1,..,m \tag{2b}$$

It is a straightforward calculation to verify that the variables z defined by (2) satisfy the relation

$$\dot{z} = Az + E[\alpha(x) + \rho(x)u]$$

$$y = Cz$$

where the only nonzero rows of E are the m rows $r_1, r_1+r_2, .., r$ and these form the identity I_m, the only nonzero columns of C are the columns $1, r_1+1, r_1+r_2+1, .., r-r_m+1$ and these form the identity I_m, and

$$A = diag(A_1,...,A_m), A_i = \begin{bmatrix} 0 & I_{r_i-1} \\ 0 & 0 \end{bmatrix} \in \mathrm{R}^{r_i \times r_i}$$

The variables z are referred to as the linearizable coordinates. The remaining part of the transform can be defined by arbitrarily choosing additional independent coordinates. The condition $\det\{\rho(x)\} \neq 0$ insures the existence of a local (around x_0) change of coordinates $x \to (\xi, z)$, $\xi \in R^{n-r}, z \in R^r$ such that

$$\dot{\xi} = F(\xi, z) \tag{3a}$$

$$\dot{z} = Az + E[\alpha(x(\xi, z)) + \rho(x(\xi, z))] \tag{3b}$$

$$y = Cz \tag{3c}$$

Equation (3) is frequently referred to as the *local normal form* of (1). It is common to refer to (3a) as the *internal dynamics* and (3b) as the *linearizable dynamics*. If z is set to zero in (3a) then we have a local representation of the zero dynamics.

Equation (3) is the point of departure for the variable structure design as described in [2]. It constitutes a *regular form* in the sense of [7].

2.2 Sliding

The reduction to this normal form is commonly associated as the first step in the process of feedback linearization. Here instead of feedback linearization, we construct a variable structure control law with switching surface of the form, $s(x)=Kz(x)$. We can prove that during sliding, the equivalent control is $u_{eq} = Kz$, so that we achieve feedback linearized behavior in the sliding phase (see, [2-4, 9]).

2.3 Reaching

The second step in VS control system design is the specification of the control functions $u_i^{\pm}$ such that the manifold s(x)=0 contains a stable submanifold which insures that sliding occurs. There are many ways of approaching the reaching design problem, Utkin [10]. We consider only one. Consider the positive definite quadratic form in s

$$V(x) = s^T Qs$$

A sliding mode exists on a submanifold of s(x)=0 which lies in a region of the state space on which the time rate of change V is negative. Upon differentiation we obtain

$$\frac{d}{dt}V = 2\dot{s}^T Qs = 2\left[KAz + \alpha\right]^T QKz + 2u^T \rho^T QKz$$

If the controls are bounded, $\left|u\right|_i \le \overline{U}_i > 0$ ($0 > U_{\min,i} \le u_i \le U_{\max,i} > 0$) then obviously, to minimize the time rate of change of V, we should choose

$$u_i = U_{\min,i}\ \text{step}(s_i^*) + U_{\max,i}\ \text{step}(-s_i^*),\ i = 1,\ldots,m,$$

$$s^*(x) = \rho^T(x)QKz(x)$$

Notice that if $U_{\min,i} = -U_{\max,i}$, the control reduces to

$$u_i = -U_{\max,i}\ \text{sign}(s_i^*)$$

In this case it follows that $\dot{V}$ is negative provided

$$\left|U_{\max}{}^T \rho^T QKz\right| > \left|\left[KAz + \alpha\right]^T QKz\right| \tag{4}$$

A useful sufficient condition is that

$$\left|\left(\rho(x)U_{\max}\right)_i\right| > \left|\left(KAz(x) + \alpha(x)\right)_i\right| \tag{5}$$

Condition (4) or (5) may be used to insure that the control bounds are of sufficient magnitude to guarantee sliding and to provide adequate reaching dynamics. This

rather simple approach to reaching design is satisfactory when a “bang-bang” control is acceptable.

2.4 Chattering Reduction

The state trajectories of ideal sliding motions are continuous functions of time contained entirely within the sliding manifold. These trajectories correspond to the equivalent control $u_{eq}(t)$. However, the actual control signal, $u(t)$ – definable only for nonideal trajectories – is discontinuous as a consequence of the switching mechanism that generates it. Persistent switching or ‘chattering’ is undesirable in some applications. Several techniques have been proposed to reduce or eliminate chattering. These include: ‘regularization’ of the switch by replacing it with a continuous approximation; ‘extension’ of the dynamics by using additional integrators to separate an applied discontinuous pseudo-control from the actual plant inputs; and ‘moderation’ of the reaching control magnitude as errors become small.

Switch regularization entails replacing the ideal switching function, $\text{sign}(s(x))$, with a continuous function such as

$$\text{sat}\left(\frac{1}{\varepsilon}s(x)\right) \text{ or } \frac{s(x)}{\varepsilon + |s(x)|} \text{ or } \tanh\left(\frac{s(x)}{\varepsilon}\right)$$

This intuitive approach is employed by Young & Kwatny [11] and Slotine and Sastry [12, 13] and there are probably historical precedents. Regularization induces a boundary layer around the switching manifold whose size is $O(\varepsilon)$. The justification for this approach for linear systems is provided by the results in [14]. Some of those results have been extended to single input–single output systems nonlinear systems by Marino [9]. Switch regularization for nonlinear systems has been extensively discussed by Slotine and coworkers, e.g. [12, 13]. With nonlinear systems there are subtleties and regularization can result in an unstable system.

Dynamic extension is another effective approach to control input smoothing, Emelyanov et al [15]. A sliding mode is said to be of p-th order relative to an output y if the time derivatives $\dot{y}, \ddot{y}, \ldots, y^{(p-1)}$ are continuous in t but y^p is not. The following observation is a straightforward consequence of the regular form theorem: Suppose (1) is input-output linearizable with respect to the output $y = h(x)$ with vector relative degree $(r_1,...,r_m)$. Then the sliding mode corresponding to the variable structure control law is of order $p=\min(r_1,...,r_m)$ relative to the output y. We may modify the relative degree by augmenting the system with input dynamics as described. Hence, we can directly control the smoothness of the output vector y.

Control moderation involves design of the reaching control functions $u_i(x)$ such that $|u_i(x)| \rightarrow$ small as $|e(x)| \rightarrow 0$. For example,

$$u_i(x) = |e(x)| \,\text{sign}(s_i(x))$$

Control moderation was used by Young and Kwatny [11] and the significance of this approach for chattering reduction in the presence of parasitic dynamics was discussed by Kwatny and Siu [16].

2.4.1 Example 1: Simple Friction

The following is a simple rotor with friction and input torque:

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = -\phi_{fr}(x_2) + u$$

Suppose the input torque *u* is bounded, say, $u \in [-U, U]$. We can easily show that the controller $u = -U\,\mathrm{sgn}(cx_1 + x_2)$, $c > 0$ and *U* sufficiently large stabilizes the origin for all piecewise smooth friction functions with a discontinuity at the origin such that $\phi_{fr}(0)$ is bounded.

Consider a VS controller with

$$u(x) = \begin{cases} u^+(x) & s(x) > 0 \\ u^-(x) & s(x) < 0 \end{cases}, \quad s(x) = cx_1 + x_2,\ c > 0$$

Imposing the sliding condition $s(x) \equiv 0$ leads to

$$\dot{x}_1 = -cx_1,\ u_{eq} = -cx_2 + \phi_{fr}(x_2)$$

Now, we need to design the reaching control. Choose $V(x) = s^2(x)$ and compute

$$\dot{V} = 2(cx_1 + x_2)(cx_2 - \phi_{fr}(x_2) + u)$$

If *u* is bounded, say, $u \in [-U, U]$, choose $u = -U\,\mathrm{sgn}(cx_1 + x_2)$. Then

$$\dot{V} = 2\,\mathrm{abs}(cx_1 + x_2)\{\mathrm{sgn}(cx_1 + x_2)(cx_2 - \phi_{fr}(x_2)) - U\}$$

Certainly, $\dot{V} < 0$ if $\mathrm{abs}(cx_2 - \phi_{fr}(x_2)) < U$. It follows that so long as $U > \sup \mathrm{abs}(\phi_{fr}(0))$ there is a neighborhood of the origin *N* such that each trajectory beginning in *N* converges to the origin.

3 Computing Tools

We need to be able to reduce the system to normal form, compute an appropriate switching surface, assemble the switching control and insert smoothing and/or moderating functions as desired. Functions that we have implemented to do this are defined in Table 1 and Table 2.

3.1 Sliding Surface Computations

There are several methods for determining the sliding surface, $s(x) = Kz(x)$, once the system has been reduced to normal form. We have included a function `SlidingSurface` that implements two alternatives depending on the arguments provided. The function may be called via

```
{rho,s}=SlidingSurface[f,g,h,x,lam]
```

or

```
s=SlidingSurface[rho,vro,z,lam]
```

Function Name	Operation
VectorRelativeOrder	computes the relative degree vector
DecouplingMatrix	computes the decoupling matrix
IOLinearize	computes the linearizing control
NormalCoordinates	computes the partial state transformation,
LocalZeroDynamics	computes the local form of the zero dynamics
StructureAlgorithm	computes the parameters of an inverse system
DynamicExtension	applies dynamic extension as a remedy for singular decoupling matrix

Table 1. Nonlinear systems: Geometric Control

Function Name	Operation
SlidingSurface	generates the sliding (switching) surface for feedback linearizable nonlinear systems
SwitchingControl	computes the switching functions – allows the inclusion of smoothing and moderating functions
SmoothingFunctions	an option for SwitchingControl that introduces specified smoothing functions
ModeratingFunctions	an option for SwitchingControl that introduces specified moderating functions

Table 2. Nonlinear systems: Variable Structure Control

In the first case the data provided is the nonlinear system definition *f*, *g*, *h*, *x* and an *m*-vector lam which contains a list of desired exponential decay rates, one for each channel. The function returns the decoupling matrix rho and the switching surfaces s as functions of the state x. The matrix *K* is obtained by solving the appropriate Ricatti equation.

The second use of the function assumes that the input-output linearization has already been performed so that the decoupling matrix rho, the vector relative degree and the normal coordinate (partial) transformation $z(x)$ are known. In this case the dimension of each of the *m* switching surfaces is known so that it is possible to specify a complete set of eigenvalues for each surface. Thus, lam is a list of *m*-sublists containing the specified eigenvalues. Only the switching surfaces are returned. In this case *K* is obtained via pole placement.

3.2 Switching Control

The function `SwitchingControl[rho,s,bounds,Q,opts]` returns the variable structure control, where rho is the decoupling matrix, s is the vector of switching surfaces, 'bounds' is a list of controller bounds each in the form {lower bound, upper bound}, Q is an *m*x*m* positive definite matrix (a design parameter), and 'opts' are options that allow the inclusion of smoothing and/or moderating functions in the control.

Smoothing functions are specified by a rule of the form

```
SmoothingFunctions[x_]->{function1[x],...,functionm[x]}
```

Where *m* is the number of controls. Moderating functions are similarly specified by a rule

```
ModeratingFunctions->{function1[z],...,functionm[z]}
```

The smoothing function option replaces any pure switch sign by a smooth switch function as specified. The moderating function option multiplies the switch by the specified function. We give an example below.

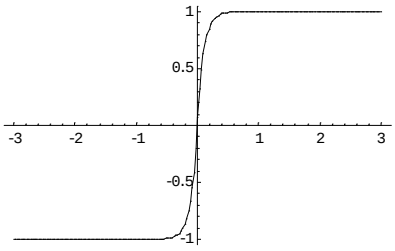
3.2.1 Example 1 Continued.

We will apply some of the above computations to Example 1. For illustrative purposes the friction function is taken to be

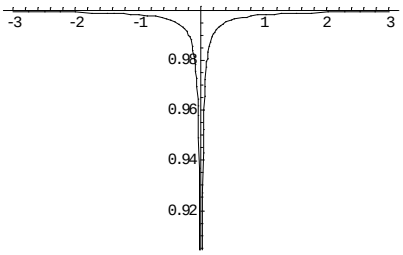
$$\phi_{fr} = \operatorname{sign}\omega\,.$$

```
{rho2, s2} = SlidingSurface[f, g, h, {theta, omega}, {2}]
Computing Decoupling Matrix
Computing linearizing/decoupling control
{{{1}}, {8.03066 omega + 16.1844 theta}}
```

Now, we compute the switching control using various combinations of smoothing and moderating functions. The particular functions chosen for this example are shown below in Figure 1. Results can change significantly when other functions are used or when the parameters of the functions are varied.



$$1 - e^{-Abs(x)/.1}$$



$$\frac{Abs(x)}{.005 + Abs(x)}$$

$$Abs(x) \leftarrow Abs(\theta) + Abs(\omega)/10$$

Smoothing Function **Moderating Function**

Figure 1. Smoothing and moderating functions used in the example.

We specify the control bounds as ± 5 and $Q = 1$. The following computation yields the four controls.

```
In[29]:= vsc1 = SwitchingControl[rho2, s2, ctrlbnds, Q]
         vsc2 = SwitchingControl[rho2, s2,
           ctrlbnds, Q, SmoothingFunctions[x_] -> {(1 - Exp[-Abs[x / .1]])}]
         vsc3 = SwitchingControl[rho2, s2, ctrlbnds, Q, ModeratingFunctions ->
             {(Abs[theta] + Abs[omega] / 10) / (.002 + Abs[theta] + Abs[omega] / 10)}]
         vsc4 = SwitchingControl[rho2, s2, ctrlbnds, Q, ModeratingFunctions ->
             {(Abs[theta] + Abs[omega] / 10) / (.002 + Abs[theta] + Abs[omega] / 10)},
          SmoothingFunctions[x_] -> {(1 - Exp[-Abs[x / .1]])}]

Out[29]= {5 Sign[-8.03066 omega - 16.1844 theta]}

Out[30]= {5 Sign[-8.03066 omega - 16.1844 theta] -
           5 E^(-10. Abs[-8.03066 omega - 16.1844 theta]) Sign[-8.03066 omega - 16.1844 theta]}

Out[31]= {Abs[omega] Sign[-8.03066 omega - 16.1844 theta] /
           (2 (0.002 + Abs[omega]/10 + Abs[theta])) +
          5 Abs[theta] Sign[-8.03066 omega - 16.1844 theta] /
           (0.002 + Abs[omega]/10 + Abs[theta])}

Out[32]= {Abs[omega] Sign[-8.03066 omega - 16.1844 theta] /
           (2 (0.002 + Abs[omega]/10 + Abs[theta])) -
          E^(-10. Abs[-8.03066 omega - 16.1844 theta]) Abs[omega] Sign[-8.03066 omega - 16.1844 theta] /
           (2 (0.002 + Abs[omega]/10 + Abs[theta])) +
          5 Abs[theta] Sign[-8.03066 omega - 16.1844 theta] /
           (0.002 + Abs[omega]/10 + Abs[theta]) -
          5 E^(-10. Abs[-8.03066 omega - 16.1844 theta]) Abs[theta] Sign[-8.03066 omega - 16.1844 theta] /
           (0.002 + Abs[omega]/10 + Abs[theta])}
```

Notice that the controllers do not depend on the specific parameters of the friction function. Figure 2 compares the closed loop performance of the first three controllers.

4 Nonsmooth Plants

Many important systems contain so-called ‘hard’ or ‘nonsmooth’ nonlinearities such as dead zone, backlash, hysteresis and coulomb friction. These nonlinearities can have a profound influence on the performance of a control system. While there exist standard models for these frequently neglected (often considered parasitic) effects, the parameters associated with them are almost always highly uncertain. Approaches to control system design that directly address hard nonlinearities must account for that uncertainty. Several alternatives have been suggested including a variety of adaptive [17, 18] and variable structure control methods.

Both adaptive and variable structure control designs are simple and effective if the system is input-output feedback linearizable and minimum phase [1, 2, 4, 19, 20]. When this is the case, the first step in design is to reduce the system the regular form described above. The basic reduction process applies to affine systems that are sufficiently smooth so that functions can be differentiated an appropriate number of times.

In this case we are interested in a more general class of models than given by (1):

$$\dot{x} = f(x) + G(x)\varphi(u)$$
$$y = h(x) \tag{1a}$$

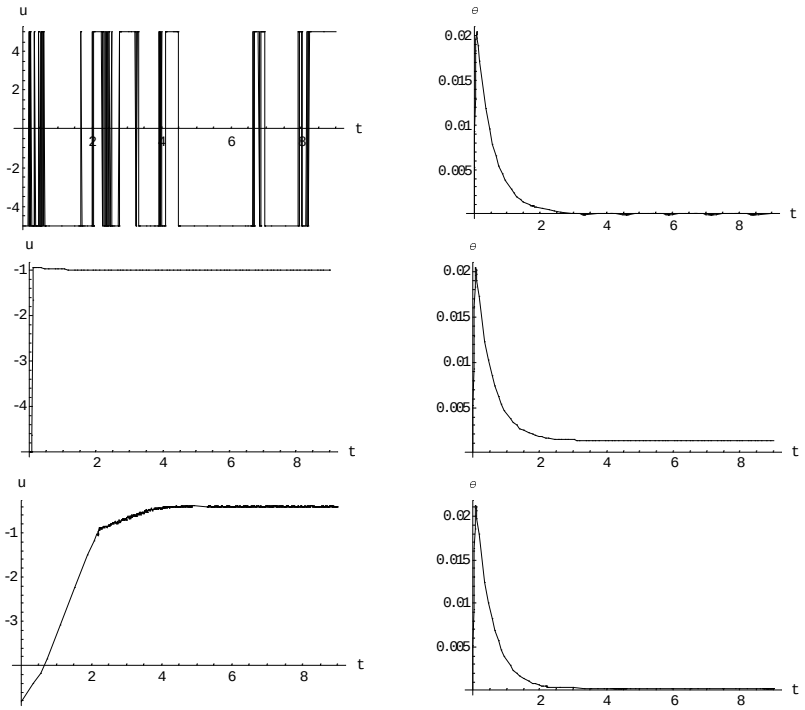


Figure 2. This figure compares, top to bottom, pure switching control, switching with smoothing and switching with moderating. From left to right: control, *u*, and position, *θ*. Chattering is virtually eliminated with either smoothing or moderating. However, smoothing leaves significantly larger steady state error because the effective gain is now bounded.

where $\varphi : R^m \to R^m$ is an invertible mapping, and *f*, *G*, *h* are only piecewise smooth functions.

Our primary interest has been applications to various pointing control systems associated with relatively small (Apache helicopter) to very large (Abrams tank) weapons. In these cases friction is a very significant issue and, depending on the drive system, backlash may also be important.

4.1 Controller Design with Nonsmooth Plants

One approach to dealing with nonsmooth nonlinearities is to approximate the nonsmooth function by a smooth one. In particular, we might consider replacing a piecewise smooth function $f(x)$ by a smooth ε-approximation $\hat{f}(x,\varepsilon)$ such that

$\lim_{\varepsilon \to 0} \hat{f}(x,\varepsilon) \to f(x)$. Then the design proceeds using the approximate system with ε sufficiently small. It is important to realize that there is no *a priori* assurance that the resulting control system when applied to the original nonsmooth plant will produce closed loop behavior close to that designed for the approximate smooth plant. There are many examples in which any smooth approximation to nonsmooth nonlinear dynamics produces qualitatively different behavior.

As a matter of fact, a naïve application of the above approach for designing variable structure controllers, i.e., reduction to normal form, smooth ε-approximation of the nonsmooth friction, then variable structure control design (sliding and reaching control), will almost certainly fail. We will give a simple explanation below. As an alternative, we will use a backstepping approach, introduced in [21] for adaptive control design and adapted for recursive Lyapunov design in [22]. Now, let us consider the following simple example which highlights the essential issues.

4.1.1 Example 2: Sandwiched Friction

Suppose we reduce the system

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = -\phi_{fr}(x_2) + x_3$$
$$\dot{x}_3 = u$$

to normal form. Let us write the friction model in the form of a nominal plus an uncertain part: $\phi_{fr}(x_2) = \phi_{fr0}(x_2) + \delta\phi_{fr}(x_2)$, where $\phi_{fr0}(x_2)$ is smooth. For example, $\phi_{fr0}(x_2) = \tanh(x_2 / \varepsilon), \varepsilon > 0$ and $\delta\phi_{fr}(x_2) = \text{sign}(x_2) - \tanh(x_2 / \varepsilon)$.

Then we have the coordinate transform

$$z_1 = x_1$$
$$z_2 = x_2$$
$$z_3 = -\phi_{fr}(x_2) + x_3$$

which yields the transformed system

$$\dot{z}_1 = z_2$$
$$\dot{z}_2 = z_3$$
$$\dot{z}_3 = -\phi'_{fr0}(z_2) + \delta\phi'_{fr}(z_2) + u$$

Thus, any error in the friction function produces an uncertainty that depends on the derivative $\delta\phi'_{fr}(z_2)$. Obviously, if the friction function is nondifferentiable, this will produce an unbounded (although matched) uncertainty. The variable structure control, which has bounded control authority, cannot be made robust to this type of unbounded uncertainty. See Figure 5 for simulation results.

Let us instead base the normal form reduction on the smooth nominal system. Then we have the coordinate transform

$$z_1 = x_1$$
$$z_2 = x_2$$
$$z_3 = -\phi_{fr0}(x_2) + x_3$$

which yields the transformed system

$$\dot{z}_1 = z_2$$
$$\dot{z}_2 = z_3 + \delta\phi_{fr}(z_2)$$
$$\dot{z}_3 = -\phi'_{fr0}(z_2) + u$$

Now we have a bounded, although not matched, uncertainty. It is precisely because the uncertainty is unmatched that we use a backstepping approach. Before proceeding with this example we describe the backstepping process.

4.2 The VS Backstep Procedure

We give a brief description of the backstepping procedure we propose for SISO VS control system design in the presence of uncertain nonsmooth nonlinearities. Technical details and stability proofs will be given elsewhere. The key innovations in our approach for nonsmooth plants are (1) that the states are grouped depending on where an uncertainty enters the system and the robustification is attempted only where the uncertainty is identified, and (2) that the control designed at each step is a variable structure control.

Consider a SISO nonlinear system in the (multi-state back-stepping) form:

$$\begin{aligned} x_i^{(n_i)} &= x_{i+1} + \Delta_i(x,t), \quad i = 1,\ldots,p-1 \\ x_p^{(n_p)} &= \alpha(x) + \rho(x)u + \Delta_p(x,t) \\ y &= x_1 \end{aligned} \tag{6}$$

We assume that the (possibly nonsmooth) uncertainties $\Delta_i(x,t)$ are bounded by smooth, non-negative functions $\varepsilon_i(x)$, i.e.,

$$0 \le \left|\Delta_i(x,t)\right| \le \varepsilon_i(x)$$

Such a model might arise by reduction of a smooth nominal system to regular from and applying the transformation to the uncertain system.

At each of *p*-1 stages we design a 'pseudo-control' v_i. The k^{th} control is obtained by designing a stabilizing smoothed VS controller for a system in the form

$$\begin{aligned} y_i^{(n_i)} &= y_{i+1}, \quad i = 1,\ldots,k-2 \\ y_{k-1}^{(n_{k-1})} &= x_k, \\ x_k^{(n_k)} &= v_k \\ y_k &= x_k - v_{k-1} \end{aligned}$$

To design the control v_k we first reduce the system to normal form by successive differentiation:

$$y_k^{(n_k)} = v_k - L_f^{n_k}(x_k - v_{k-1})$$

Thus, we identify the evolution equation in the new coordinate y_k that will replace x_k. Notice that the zero dynamics of this system are

$$y_i^{(n_i)} = y_{i+1}, \quad i = 1,\ldots,k-2$$
$$y_{k-1}^{(n_{k-1})} = v_{k-1}$$

Now, we design a VS stabilizing controller, $v_k(y_k,\ldots,y_k^{(n_k)})$ such that $y_k(t) \to 0$ as $t \to \infty$. For each $k < p$ we smooth the controller so that the process can be continued. Working in this way through the *p* stages, and redefining the states ($x \to y$) at each stage we arrive at the final set of dynamical equations. Notice the triangular structure.

$$\begin{aligned} y_i^{(n_i)} &= y_{i+1} + v_i(y_i,\ldots y_i^{(n_i)}) \quad i = 1,\ldots,p-1 \\ y_p^{(n_p)} &= \alpha + \rho u(y_p,\ldots y_p^{(n_p)}) \end{aligned} \tag{7}$$

This structure, upon which a stability analysis is based, is illustrated in Figure 3. The basic idea is roughly as follows. A VS controller is designed for system *p*, (7), via methods described above. The system is stable if and only if the zero dynamics,

$$y_i^{(n_i)} = y_{i+1} + v_i(y_i,\ldots y_i^{(n_i)}) \quad i = 1,\ldots,p-1, \tag{8}$$

are stable. But, v_{p-1} is itself a (smoothed) VS control so that (8) is stable if its zero dynamics:

$$y_i^{(n_i)} = y_{i+1} + v_i(y_i,\ldots y_i^{(n_i)}) \quad i = 1,\ldots,p-2$$

are stable. The argument proceeds in this way. There are subtleties because of the smoothing. And we must also establish the robust stability properties.

4.2.1 Example 2 continued

Since the example system is already in multi-state back stepping form (6) no transformation is necessary. We break the system into two parts, treating x_3 as a temporary control and ignoring the uncertainty:

Step 1 Design a *smoothed* VS control, $v(x_1, x_2)$, for:

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = -\phi_{fr}(x_2) + v$$
$$y = x_1$$

Then, we design a VS control for the composite nominal system with modified output equation.

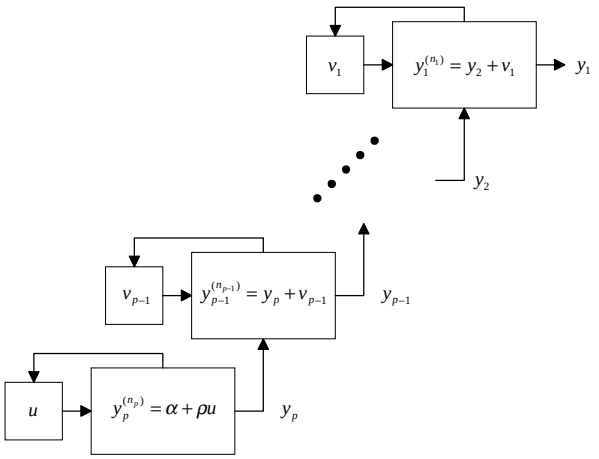


Figure 3. The triangular structure of the closed loop dynamics achieved with the multistate backstep control design.

Step 2 Design a VS control, u for

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = -\phi_{fr}(x_2) + x_3$$
$$\dot{x}_3 = u$$
$$y = x_3 - v(x_1, x_2)$$

Now, we will implement these calculatons. The Mathematica code is shown below where $[x_1, x_2, x_3] \rightarrow [\text{theta}, \text{omega}, \text{uu}]$. Using the previously described tools we have for Step 1:

```
f1 = {omega, -Tanh[omega / .02]};
g1 = {0, 1};
h1 = {theta};
{rho1, s1} = SlidingSurface[f1, g1, h1, {theta, omega}, {2}]
ctrlbnds = {{-5, 5}};
Q = {{1}};
vsc0 = SwitchingControl[rho1, s1, ctrlbnds, Q,
  SmoothingFunctions[x_] -> {Tanh[x / .01]}]
```

```
Out[4]= {-5 Tanh[100. omega + 423.607 theta]}
```

and Step 2:

```
f = {omega, -Tanh[omega / .02] + uu, 0};
g = {0, 0, 1};
h = {uu - vsc0[[1]]};
{rho2, s2} = SlidingSurface[f, g, h, {theta, omega, uu}, {20}]
ctrlbnds = {{-5, 5}};
Q = {{1}};
vsc1 = SwitchingControl[rho2, s2, ctrlbnds, Q]
```

```
Out[19]= {5 Sign[-uu - 5 Tanh[100. omega + 423.607 theta]]}
```

Simulation results obtained with this controller are illustrated by the trajectory in Figure 4. For comparison purposes, Figure 5 illustrates the failure of the non-backstepping controller to eliminate the position output error – as anticipated.

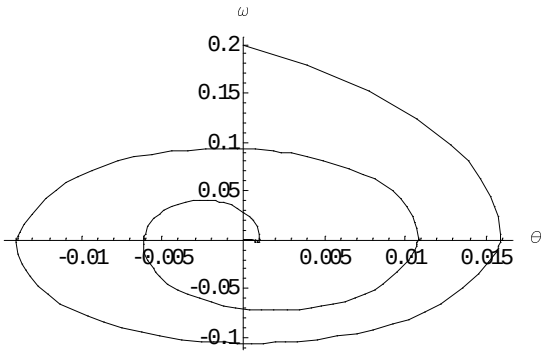


Figure 4. The projection of a state trajectory on the $\omega - \theta$ plane illustrates asymptotic convergence.

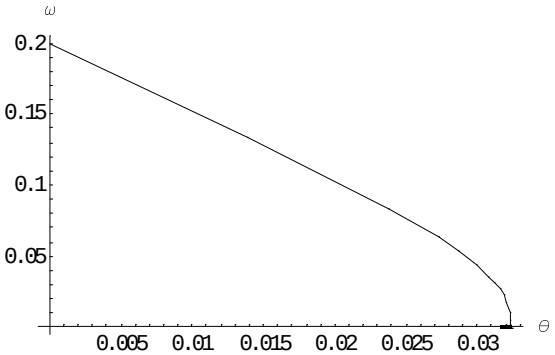


Figure 5. A similar projection using the conventional (non-backstep) design illustrates how the trajectory "sticks" because of the large matched uncertainty.

5 Apache Helicopter 30-mm Chain Gun

As a test bed problem for the variable structure control design tools, we used the US Army Apache Helicopter 30-mm chain gun. The goal of the control design is to increase the pointing and tracking performance of the gun by including friction compensation.

Figure 6. Apache 30mm Chain Gun Test Bed ADAWS Lab, Picatinny Arsenal

The control will be tested at the Apache 30mm Chain Gun Test Bed ADAWS Lab, Picatinny Arsenal Figure 6. The test bed gun is driven by a direct drive electric motor that is simply modeled as an input torque. The friction in the motor is dominant, so this is a simple friction control problem, not sandwiched friction problem.

5.1 Dynamic Model of Apache Gun

A four-degree of freedom (DOF) model of the apache gun system was developed. A schematic of the multibody-flex model appears in Figure 7. The model consists of a rigid turret with flexible forks that connect to a rigid gun which has a flexible barrel attached to it. A rigid blast suppressor is attached to the muzzle end of the barrel. A two channel bending actuator, developed by TSi to increase pointing accuracy, is mounted to the flexible barrel. The actuator can deliver two pairs of torques to produce muzzle angular deflection in both azimuth and elevation. The bending actuator will not be used for this study.

The gun system model was developed using the Mathematica package *ProPac* [23]. Figure 7 shows the three bodies into which the gun system is broken for modeling. Each body has a local reference frame that is located at the inboard joint.

The reference frames and their associated degrees of freedom are also shown in Figure 7.

The model was developed for designing and testing slewing controls. In order to keep the model dimension to a minimum; motions not related to slewing or slewing disturbances are not modeled. For example, the elevation of the gun is assumed a fixed value, thus eliminating a potential degree of freedom. In addition, any bending of the forks such as that which might be caused by a firing disturbance is not modeled.

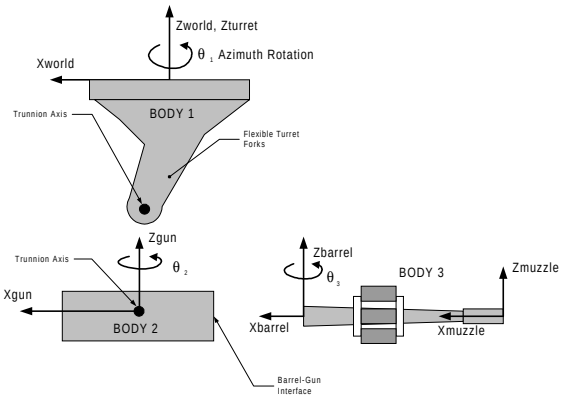


Figure 7. Body coordinate reference frames and degrees of freedom.

As depicted in Figure 7, the turret has one degree of freedom, θ_1, about the azimuth axis. The flexibility of the forks is modeled by a torsional spring with one degree of freedom, θ_2. The majority of the vibration energy in the system comes from the approximately 12 Hz flexure mode of the forks in the azimuth direction. The barrel-gun interface is allowed one degree of freedom, θ_3. This joint allows for the unintentional motion at the gun-barrel interface due to clearances in the fitting. The barrel-gun interface joint was added to the model during the model validation process to improve the matching of the transfer functions.

The flexible barrel is modeled using a reduced order FEM modal model and includes the first lateral bending modes in the azimuth plane. The reasoning behind including only the first mode is that for a cantilevered beam, 75% of the energy is in the first mode. The complete non-linear system model of the apache gun contains 4 degrees of freedom which are $[\theta_1, \theta_2, \theta_3, x_1,]^T$, where x_1 is the barrel modal coordinate.

For the slewing control model, there is one control input to the system: a torque, $\tau_{tur,}$ generated by the turret motor and applied to the turret about the Z_{tur} axis. There are two outputs measurements from the system: turret azimuth, θ_{tur}, and muzzle azimuth acceleration, a_{maz}, both muzzle variables are measured with respect to the world coordinate frame.

5.1.1 Modal Analysis

Under the assumption that the fork deflections are small, i.e., θ_2 is small, then the non-linear model can be linearized about θ_1 to generate a representative linear state space model. The state space equations take the following form

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

where $u = [\tau_{tur}]$ is the control input vector, $y = [\theta_{tur} \quad a_{maz}]^T$ is the system measurement (output) vector, and $x = [q \quad \dot{q}]^T$ is the system state vector, and $q = [\theta_1 \quad \theta_2 \quad \theta_3 \quad x_1]^T$ is the coordinate vector.

Mode Number	***Frequency [Hz]***	***Mode Description***
1	0	Rigid body rotation about the turret azimuth axis
2	12.6	Fork azimuth flexure. Motion at gun/barrel interface in phase with gun motion. Barrel flexure negligible.
3	54.3	System moves as three bodies with the first and last moving out of phase with the second. In addition, the last body, the barrel, flexes in phase with the turret and gun/barrel interface motion.
4	322.0	Primarily barrel flexure in azimuth plane. Interface motion and barrel flexure out of phase. Other motion negligible.

Table 3: Description of mode shapes and frequencies of chain gun model

An eigenvalue decomposition of the *A* matrix gives the linearized approximation of the system modal frequencies. The system mode frequencies and mode shapes are described in Table 3. The lowest frequency mode is a rigid body mode corresponding to turret azimuth angular displacement. The next lowest frequency mode is predicted at 12.6 Hz and corresponds to fork azimuth flexure. This mode is characterized by angular displacements of the rigid turret and the gun/barrel assembly that are 180° out of phase with each other. The predicted frequency of this mode agrees with observations made of the production gun system frequency, which was determined to be around 10 Hz[2]. This is probably the most problematic mode in the system, since the firing rate of the gun is approximately 10 Hz.

[2] The slight increase in frequency of this mode may be due to differences in the testbed system from the real system.

The remaining system modes are associated with barrel flexure. The third mode excites the turret and barrel to move out of phase with the gun. In addition, the barrel flexes in phase with the turret and gun/barrel interface motion. The fourth mode is azimuth barrel flexure mode. In this mode, the flexure of the barrel is out of phase with motion of the barrel/gun interface.

5.2 Friction Modeling

The goal of this control design is to increase the pointing and tracking performance of the apache gun testbed at ARDEC by including friction compensation. Experimentalists have observed several characteristic properties of friction. These properties can be broken into two categories: static and dynamic. The static characteristics of friction, including the stiction force, the kinetic force, the viscous force, and the Stribeck effect, are functions of steady state velocity. The dynamic phenomena include pre-sliding displacement, varying breakaway force, and frictional lag. Many empirical friction models have been developed which attempt to capture specific parts of observed friction behavior.

With all the models available one must decide which friction model should be used in the friction compensating control. It is unclear whether complicated friction models improve control performance. One problem is the difficulty in obtaining *good* parameter estimates. In experiments we have performed to obtain parameter estimates, we found it particularly difficult to estimate the dynamic friction parameters [24]. The problem is complicated by the fact that the parameters may vary considerably based on such factors as temperature, lubricant condition, and material wear [25, 26]. Moreover, the various friction models in the literature represent many empirical features; however, realistically the friction present in any physical system may be different from that described in the model. We have chosen to use a simple static friction model.

Some sort of control is needed which is robust to the inaccuracy in parameter measurement, variation of parameters, and model inaccuracy. In the simulation results below, we demonstrate that the variable structure control achieves the first two. Testing is scheduled to determine the control performance on the test bed gun system.

5.2.1 Friction Experiments

The tests performed to determine static friction parameter estimates are described below. It is assumed that friction enters only at joint 1 and is a function of the angular velocity of the turret.

The steady state friction parameters including static, Coulomb, viscous, and Stribeck friction terms (F_S, F_C, F_V, v_{str}) were estimated. In the estimation problem, nonlinear constrained optimization methods from the Matlab Optimization Toolbox [27] were applied. The objective function used was the error between the observed and predicted value of friction. Being non-convex with respect to some of the optimization parameters, the objective function may have local minima. It is thus important to start from a reasonable initial guess for the parameters. This is not too difficult for the steady state parameters.

To estimate the steady state friction parameters a friction versus velocity[3] map is constructed. To construct the friction versus velocity map, several constant velocity experiments were run with reference velocities ranging from -0.03 radians/second to +0.05 radians/second. A closed-loop PI velocity control law was implemented for the tests. The velocity for feedback control was estimated from a low pass filtered derivative of the motor encoder output. Average steady state velocity and friction force[4] were computed from the time histories of each experiment to produce the data points 'o' in Figure 8. The zero velocity data point was obtained by computing the average of the break away force from experiments where the driving force was a linearly increasing input force.

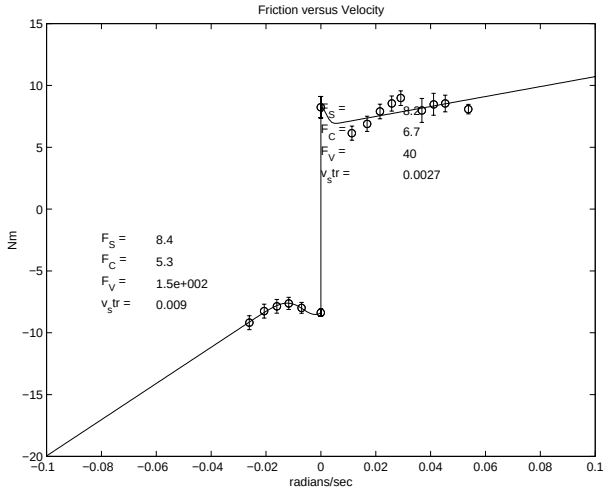


Figure 8. Friction versus Velocity

The parameter estimates were obtained by minimizing over $\hat{F}_S, \hat{F}_C, \hat{F}_V, \hat{v}_{str}$ the cost function

$$\sum_{i=1}^{n} \left\{ F_{SS}(v_i) - \hat{F}_{SS}(v_i) \right\}^2$$

where $F_{SS}(v_i)$ are the friction data obtained during constant velocity motion and

$$\hat{F}_{SS}(v_i) = \begin{cases} \left[\hat{F}_C + (\hat{F}_S - \hat{F}_C) \exp\left(-\left(v_i / \hat{v}_{str} \right)^2 \right) \right] \mathrm{sgn}(v_i) + \hat{F}_V v_i & v_i \neq 0 \\ \min(\hat{F}_S, F) & v_i = 0 \end{cases}.$$

---

[3] The measured velocity data is from the motor encoder. In these low constant velocity motions the system is assumed rigid.

[4] The friction force is approximately equal to the negative of the input torque in constant velocity motion.

The parameters were all constrained to be greater than or equal to zero, with the additional constraints that $0.00001 \le \hat{v}_{str} \le 0.01$, and $\hat{F}_S \ge \hat{F}_C$. The optimization was done using the Matlab constrained optimization function Constr with tolerances of 1.0e-5 and the maximum number of iteration set to 1000.

The steady state friction parameter estimates are given in Table 4. The estimates for static, F_S, and coulomb, F_C, friction parameters were obtained by averaging the values for positive (clockwise) and negative (counterclockwise) motion. The viscous, F_V, and Stribeck, v_{str}, friction parameters were taken directly from the positive motion data since it seems that estimates from the negative motion data may be incorrect due to insufficient measurements at high enough velocity.

5.2.2 Simulation

Simulations of the nonlinear apache gun model were run using a simple PID control with anti-windup to demonstrate the effects of friction on simple control strategies. The Simulink block diagram of the closed loop system is displayed in Figure 9.

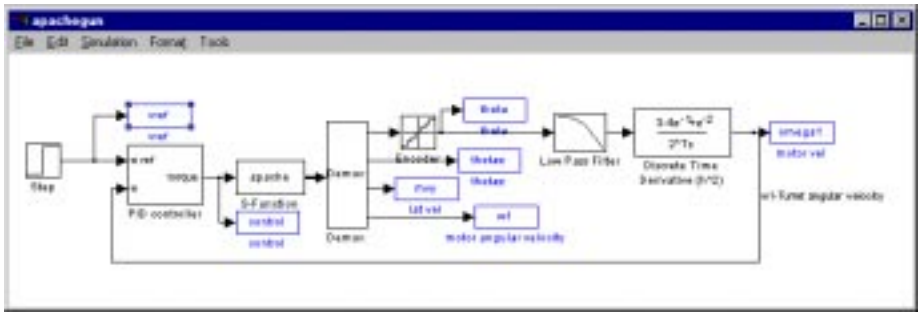


Figure 9. Simulink Apache gun simulation model

Parameter	Value
Static friction, F_S	8.3 N m
Coulomb friction, F_C	6.0 N m
Viscous friction, F_V	40 N m s/rad
Stribeck velocity, v_{str}	0.003 rad/s

Table 4: Friction Parameters

In addition to friction, there are other nonlinearities in the system that must be accounted for if one hopes to obtain realistic simulation results. For example, care must be taken to model the nonlinearities imposed by actuator saturation as well as

measurement quantization. Accounting for these factors can alter significantly the simulation results. The PID controller output torque is saturation limited. The saturation values are ± 162.5 Nm (32.5 Nm/V*5 V). In addition, a quantizer with quantization interval $2\pi/2^{20}$ is included to model the encoder output. A low pass filter is added before the derivative block to smooth the encoder output. Care must be taken when adding a filter to a feedback loop. The phase delay of the filter could cause the closed loop system to be come unstable.

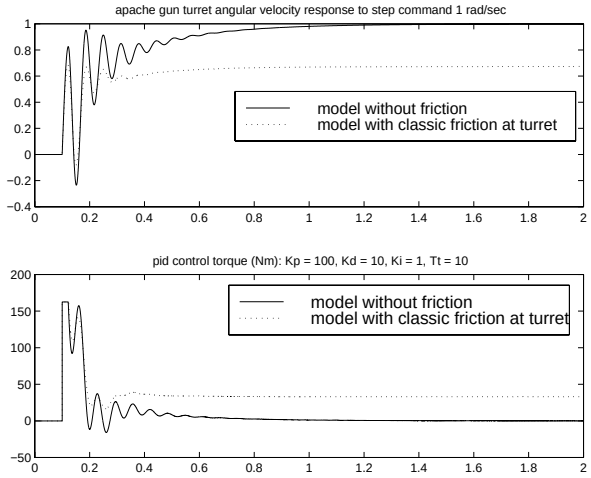


Figure 10. Step response with PID control with and without friction

In the simulations, friction is modeled using the classical model with the values obtained during the friction experiments described above, see Table 4. The friction enters the system at the first joint and is a function of the turret angular velocity, ω_1.

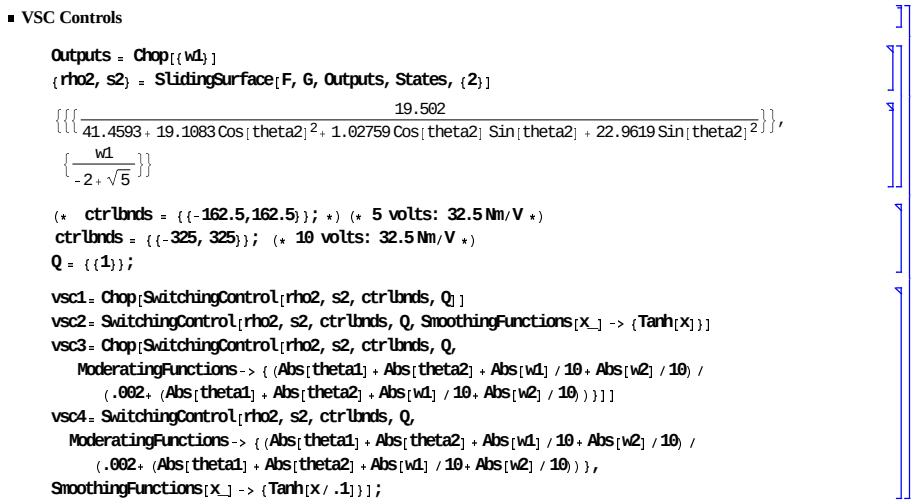
For comparison purposes, simulations were run both with and without friction. A step velocity command of 1 rad/sec at 0.1 sec was input to the PID controller. The simulation results are shown in Figure 10. It is clear from the simulation results that accounting for friction in the gun model simulation has significantly deteriorated the PID controlled tracking performance. The system with friction has a large steady state error from the commanded velocity.

5.3 Variable Structure Control

For control design purposes we used a rigid two-body model of the apache gun. The flexibility in the barrel and the movement at the gun barrel interface were ignored. The friction enters the system at the first joint and is a function of the turret angular velocity ω_1, as described in above.

Once the control design model was generated, the control was simple to obtain. The Mathematica/*ProPac* code for the control design is shown in Figure 11. A simple VSC control was generated. In addition, controls with smoothing using the

function $\tanh(x)$, and moderation using the function $\dfrac{|\omega_1|/10+|\omega_2|/10}{0.02+|\omega_1|/10+|\omega_2|/10}$, and both moderation and smoothing were generated.



```
VSC Controls

Outputs = Chop[{w1}]
{rho2, s2} = SlidingSurface[F, G, Outputs, States, {2}]
{{{19.502 / (41.4593 + 19.1083 Cos[theta2]^2 + 1.02759 Cos[theta2] Sin[theta2] + 22.9619 Sin[theta2]^2)}},
 {{w1 / (-2 + Sqrt[5])}}}

(*  ctrlbnds = {{-162.5,162.5}}; *) (* 5 volts: 32.5 Nm/V *)
ctrlbnds = {{-325, 325}}; (* 10 volts: 32.5 Nm/V *)
Q = {{1}};

vsc1 = Chop[SwitchingControl[rho2, s2, ctrlbnds, Q]]
vsc2 = SwitchingControl[rho2, s2, ctrlbnds, Q, SmoothingFunctions[x_] -> {Tanh[x]}]
vsc3 = Chop[SwitchingControl[rho2, s2, ctrlbnds, Q,
   ModeratingFunctions -> {(Abs[theta1] + Abs[theta2] + Abs[w1] / 10 + Abs[w2] / 10) /
      (.002 + (Abs[theta1] + Abs[theta2] + Abs[w1] / 10 + Abs[w2] / 10))}]]
vsc4 = SwitchingControl[rho2, s2, ctrlbnds, Q,
  ModeratingFunctions -> {(Abs[theta1] + Abs[theta2] + Abs[w1] / 10 + Abs[w2] / 10) /
     (.002 + (Abs[theta1] + Abs[theta2] + Abs[w1] / 10 + Abs[w2] / 10))},
SmoothingFunctions[x_] -> {Tanh[x / .1]}];
```

Figure 11. Mathematica code for VSC design

Simulations were run from Simulink testing control of the 3-body apache model against each of the controls. The goal of each of the controls is to drive the angular velocity ω_1 to zero. The results are shown below. It is clear from the simulation that for this command, the VSC with smoothing performs best, i.e., induces the least vibration in the system.

6 Conclusions

In this paper we have described a set of symbolic computing tools that enable efficient design and implementation of variable structure control systems. The functionality includes reduction to regular form, computation of zero dynamics, design of sliding modes, assembly of the switching controller, the addition of smoothing and moderating functions and assembly of C-code for real-time implementation. The toolbox in its present form applies to smooth, (partially) feedback linearizable systems.

In developing the toolbox for variable structure control, we have had to tackle several fundamental issues involved in symbolic computing with nonsmooth functions in general, i.e., whether control or modeling related. Ongoing work is focused on extending the design method as well as the software to plants with nondifferentiable nonlinearities other than friction. We have provided a simple example above that illustrates some of the issues involved in the control design and explains why we have adopted a backstepping approach to plants of this type. Our results to date suggest that this formulation can be very effective for systems

involving uncertain nonlinear friction. The backstepping approach to handling robust control is not new, however in our approach we put a new twist on it. The key innovations in our approach for nonsmooth plants are (1) that the states are grouped depending on where an uncertainty enters the system and the robustification is attempted only where the uncertainty is identified, and (2) that the control designed at each step is a smoothed variable structure control.

An additional example was given which demonstrates the power of the computing tools to easily tackle real industrial control problems with simple nonsmooth uncertainties. The variable structure control designed for the apache gun system is scheduled to be tested using the real-time C-code implementation.

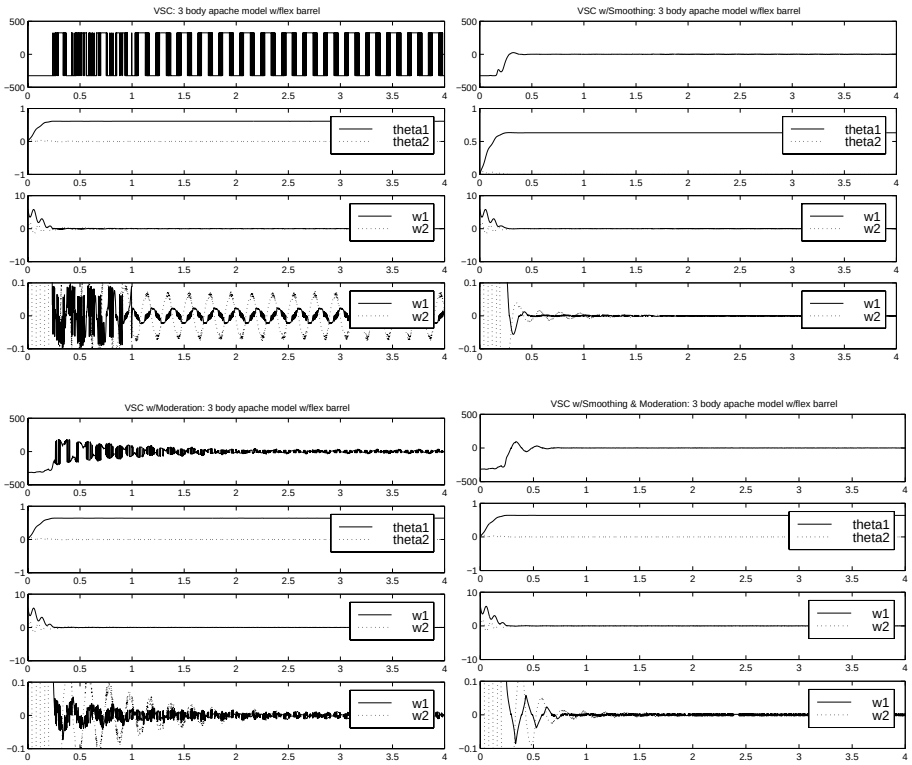


Figure 12. Apache VSC simulations

In order to address the issue of robustness to parameter variation, simulations were performed over a large range of friction parameters. Of course, with less friction the control performed better, i.e., less induced vibration in the system. In Figure 13, simulation results are shown for the case when the friction parameters are increased from their nominal values, see Table 4, to approximately double the friction, $F_v = 45$ Nm, $F_C = 12$ Nm, $F_S = 17$ Nms / rad, $v_S = 0.003$ rad / sec and then to approximately eight times the friction, $F_v = 45$ Nm, $F_C = 48$ Nm, $F_S = 64$ Nms / rad, $v_S = 0.003$ rad / s.

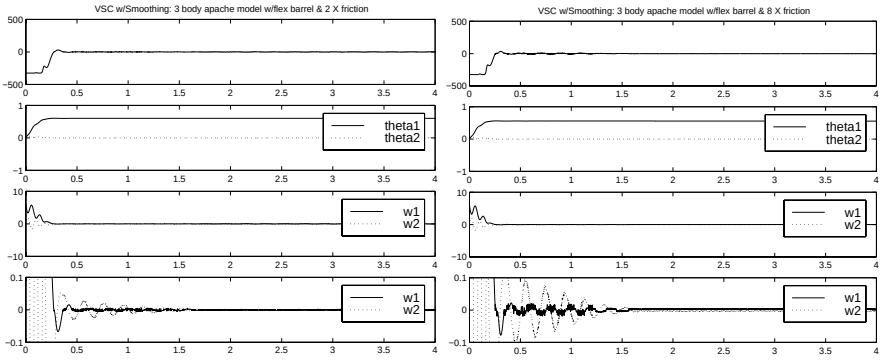


Figure 13. Robustness to varying friction parameters.

Acknowledgement: Supported, in part, by U. S. Army ARDEC, Contract No.: DAAE30-96-C-0063

7 References

1. Blankenship, G.L., *et al.*, *Integrated tools for Modeling and Design of Controlled Nonlinear Systems.* IEEE Control Systems, 1995. **15**(2): p. 65-79.
2. Kwatny, H.G. and H. Kim, *Variable Structure Regulation of Partially Linearizable Dynamics.* Systems & Control Letters, 1990. **15**: p. 67–80.
3. Kwatny, H.G., *Variable Structure Control of AC Drives*, in *Variable Structure Control for Robotics and Aerospace Applications*, K.D. Young, Editor. 1993, Elsevier: Amsterdam.
4. Kwatny, H.G. and J. Berg, *Variable Structure Regulation of Power Plant Drum Level*, in *Systems and Control Theory for Power Systems*, J. Chow, R.J. Thomas, and P.V. Kokotovic, Editors. 1995, Springer–Verlag: New York. p. 205-234.
5. Isidori, A., *Nonlinear Control Systems*. 1989, NY: Springer-Verlag.
6. Kwatny, H.G. and G.L. Blankenship. *Symbolic Tools for Variable Structure Control System Design: The Zero Dynamics*. in *IFAC Symposium on Robust Control via Variable Structure and Lyapunov Techniques.* 1994. Benevento, Italy.
7. Luk'yanov, A.G. and V.I. Utkin, *Methods of Reducing Equations of Dynamic Systems to Regular Form.* Avtomatica i Telemechanika, 1981(4): p. 5–13.
8. Isidori, A., *Nonlinear Control Systems*. 3 ed. 1995, London: Springer-Verlag.
9. Marino, R., *High Gain Feedback Non–Linear Control Systems.* International Joural of Control, 1985. **42**(6): p. 1369–1385.
10. Utkin, V.I., *Sliding Modes and Their Application*. 1974 (in Russian) 1978 (in English), Moscow: MIR.
11. Young, K.D. and H.G. Kwatny, *Variable Structure Servomechanism and its Application to Overspeed Protection Control.* Automatica, 1982. **18**(4): p. 385-400.

12. Slotine, J.J. and S.S. Sastry, *Tracking Control of Non–Linear Systems Using Sliding Surfaces, With Application to Robot Manipulators.* International Journal of Control, 1983. **38**(2): p. 465–492.
13. Slotine, J.J.E., *Sliding Controller Design for Non–Linear Control Systems.* International Journal of Control, 1984. **40**(2): p. 421–434.
14. Young, K.D., P.V. Kokotovic, and V.I. Utkin, *Singular Perturbation Analysis of High Gain Feedback Systems.* IEEE Transactions on Automatic Control, 1977. **AC–22**(6): p. 931–938.
15. Emelyanov, S.V., S.K. Korovin, and L.V. Levantovsky, *A Drift Algorithm in Control of Uncertain Processes.* Problems of Control and Information Theory, 1986. **15**(6): p. 425–438.
16. Kwatny, H.G. and T.L. Siu. *Chattering in Variable Structure Feedback Systems*. in *10th IFAC World Congress*. 1987. Munich.
17. Friedland, B., *Advanced Control System Design*. 1996, Englewood Cliffs: Prentice hall.
18. Tao, G. and P.V. Kokotovic, *Adaptive Control of Systems with Actuator and Sensor Nonlinearities*. 1996, New York: John Wiley and Sons, Inc.
19. Bennett, W.H., *et al.*, *Nonlinear and Adaptive control of Flexible Space Structures.* Transactions ASME, Journal of Dynamic Systems, Measurement and Control, 1993. **115**(1): p. 86–94.
20. Bennett, W.H., H.G. Kwatny, and M.J. Baek, *Nonlinear Dynamics and Control of Articulated Flexible Spacecraft: Application to SSF/MRMS.* AIAA Journal on Guidance, Control and Dynamics, 1994. **17**(1): p. 38–47.
21. Kanellakapoulos, I., P.V. Kokotovic, and A.S. Morse, *Systematic design of Adaptive Controllers for Feedback Linearizable Systems.* IEEE Transactions on Automatic Control, 1991. **AC–36**(11): p. 1241–1253.
22. Freeman, R.A. and P.V. Kokotovic, *Design of 'Softer' Robust Nonlinear Control Laws.* Automatica, 1993. **29**(6): p. 1425-1437.
23. Kwatny, H.G. and C. LaVigna, *TSi Dynamics User's Guide*, . 1994, Techno–Sciences, Inc.: Lanham, MD.
24. Teolis, C., *Contract Summary Report: Adaptive Control of Systems with Friction and Backlash*, . 1997, Techno-Sciences, Inc.: Lanham.
25. Armstrong-Helouvry, B., *Control of Machines with Friction*. 1991, Boston: Kluwer Academic Publishers.
26. Armstrong-Helouvry, B., P. Dupont, and C.C.d. Wit, *A survey of models, analysis tools and compensation methods for the control of machines with friction.* Automatica, 1994. **30**(7): p. 1083-1138.
27. Branch, M.A. and A. Grace, *Optimization Toolbox*. 1996, Natick, MA: The Mathworks, Inc.