Symbolic Tools for Variable Structure Control System Design: the Zero Dynamics

Harry G. Kwatny
Department of Mechanical Engineering & Mechanics
Drexel University
Philadelphia, PA 19104
harry.kwatny@coe.drexel.edu

Gilmer L. Blankenship
Department of Electrical Engineering & Institute for Systems Research
University of Maryland
College Park, MD 20742
gilmer@eng.umd.edu

Abstract: A new approach to the symbolic computation of the zero dynamics for affine systems is described. The method is far less computationally burdensome than alternative approaches and it has been successfully applied to systems of moderate complexity. An implementation of the algorithm in the *Mathematica* language is described and an example is provided.

1. Introduction

Over the past several years a number of investigators have developed the connection between variable structure control theory and smooth feedback linearization of affine nonlinear systems, e.g. [1, 2]. Both approaches to control system design share common elements including certain constructions like reduction to normal form. Also, in both cases, as in all variable structure methods, the zero dynamics play an essential role. In circumstances where the system must operate over a wide range of parameter values, such as described in [3], understanding the parametric dependence of the zero dynamics is essential to control system design.

Application of the theory to nonlinear systems beyond text book level require development of computer tools that implement the lengthy differential–algebraic calculations that accompany the theory. A number of investigators have considered the implementation of the required constructions in symbolic computation languages such as MACSYMA, *Mathematica* and Maple [4-6]. The symbolic calculation of the zero dynamics has not yet been adequately resolved — even for the simple case of well defined relative degree. To our knowledge, only de Jager has seriously considered the construction of the zero dynamics via computer algebra, and in [6] he concludes that such calculations are not yet viable for moderate scale problems. The approach described in [6] is based on the construction of the local normal form [7] which requires solving certain sets of nonlinear differential equations in order to identify the necessary coordinate transformations.

In this paper we describe an entirely different approach to the calculation of zero dynamics which is far less computationally demanding. The method is based on the characterization of zero dynamics given in [2]. We illustrate the calculations with an example that includes, as a special case, a problem described in [6].

2. Computation of Normal Forms

In the following we consider affine nonlinear dynamics of the form

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

$$y = h(x) \tag{1b}$$

First, we summarize the standard reduction to normal form [7] and give our characterization of zero dynamics, following [2, 3, 8]. Then we describe an approach to computing the zero dynamics.

Definition of the Normal Form

Denote the k<u>th</u> 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 (1b) we arrive at the following definitions for the list of integers r_i, the column vector $\alpha(x)$ and the matrix $\rho(x)$:

$$r_i := \inf\{k | L_{g_j}(L_f^{k-1}(h_i)) \neq 0 \quad \textit{for at least one } j\} \tag{2a}$$

$$\alpha_i(x) := L_f^{r_i}(h_i), \quad i = 1,..,m \tag{2b}$$

$$\rho_{ij}(x) := L_{g_j}(L_f^{r_i-1}(h_i)), \quad i,j = 1,..m \tag{2c}$$

Also define the vector $z_i \in \mathbb{R}^{r_i}, i = 1,\ldots,m$, as

$$z := \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_m \end{bmatrix}, \quad z_i \in \mathbb{R}^{r_i}, i = 1,\ldots,m \tag{3a}$$

where

$$z_i^k(x) = L_f^{k-1}(h_i),\ k = 1,..,r_i \ \ and \ \ i = 1,..,m \qquad (3b)$$

It is a straightforward calculation to verify that the variables z defined by (3) satisfy the relation

$$\dot{z} = Az + E[\alpha(x) + \rho(x)u] \qquad (4a)$$

$$y = Cz \qquad (4b)$$

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 non zero 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

$$A = diag(A_1,\ldots,A_m),\ A_i = \begin{bmatrix} 0 & I_{r_i-1} \\ 0 & 0 \end{bmatrix} \in \mathrm{R}^{r_i \times r_i} \qquad (5)$$

Equation (4) is the point of departure for the variable structure design as described in [2]. It constitutes a *regular form* in the sense of [9].

In the following analysis, we will make use of the following elementary, but important, lemma:

Lemma 1: Suppose that $\rho(x)$ has continuous first derivatives with $\det\{\rho(x)\} \neq 0$ on $\}_0 = \{x|z(x) = 0\}$. Then $\partial z(x)/\partial x$ is of maximum rank on the set $\}_0 = \{x|z(x) = 0\}$.

Proof: [7]. j

The Lemma is extremely important because it relates the invertibility of the decoupling matrix with the geometry of the set $\}_0$. With it, we can obtain several important results, one of which we state here.

Proposition 1: Suppose that $\rho(x)$ has continuous first derivatives with $\det\{\rho(x)\} \neq 0$ on $\}_0 = \{x|z(x) = 0\}$. Then $\}_0$ is a regular, n-r dimensional submanifold of R^n and any trajectory segment $x(t), t \in \mathrm{T}$, T an open interval of R^1, which satisfies $h(x(t)) = 0$ on T lies entirely in $\}_0$. Moreover, the control which obtains on T is

$$u_0(x) = -\rho^{-1}(x)\alpha(x) \qquad (6)$$

and every such trajectory segment with boundary condition $x(t_0) = x_0,\ t_0 \in \mathrm{T}$ satisfies

$$\dot{x} = f(x) - G(x)\rho^{-1}(x)\alpha(x), \quad z(x(t_0)) = 0 \qquad (7)$$

Proof: [2] j

Note that the manifold $\}_0$ defined by $z(x) = 0$ is invariant with respect to (7) so that any motion beginning in it remains therein. Indeed, (7) defines a flow on $\}_0$ with all trajectories satisfying $y(t) = h(x(t)) \equiv 0$. This justifies reference to (7) as the *zero output constrained dynamics* and to $\}_0$ as the *zero dynamics manifold*. We refer to (7) as a *global* representation of the zero dynamics. In some applications a global characterization of the zero dynamics is essential [2, 8].

The construction described above will be successful only if $\det\{\rho(x)\} \neq 0$ on $\}_0$. There does exist a remedy if it is not. Dynamic extension as described in [7] can provide a useful repair. More generally, conditions for decoupling via nonlinear dynamic feedback (of which the above construction augmented with dynamic extension is a special case) are given by Descusse and Moog [10] and Huijberts et al [11]. We will not consider that case here. It is assumed henceforth that the matrix $\rho(x)$ is nonsingular as needed. In this case, we can apply the feedback control law

$$u = -\rho^{-1}(x)[\alpha(x) - v] \qquad (8)$$

where v is a new control input. Thus, we have the linearized input-output model

$$\dot{z} = Az + Ev \qquad (9a)$$

$$y = Cz \qquad (9b)$$

Note that the control law (8) simultaneously linearizes the input-output relation and decouples some of the dynamics from the output.

It is not uncommon to refer to the variables z as the linearizable coordinates. The terminology of 'coordinates' is justified by the maximal rank condition in the following way. Let Z: $\mathrm{R}^n \rightarrow \mathrm{R}^r$ denote the map realized as the function $z(x)$. By virtue of the maximal rank assumption and the implicit function theorem we can choose local coordinates $(y_1,..,y_n)$ on R^n near any point x_0[}$_0$ such that $Z(y) = (y_1,..,y_r)$. In terms of these coordinates $\}_0$ is defined by y_1=0,..,y_r=0. As a matter of fact, the first r components correspond to the level sets $z(x)$=c which exist for all c in some neighborhood of the origin in R^r. The remaining components j := $(y_{r+1},..,y_n)$ provide local coordinates on $\}_0$. Thus, the above formal calculations make sense because the condition $\det\{\rho(x)\} \neq 0$ insures the existence of a local (around x_0) change of coordinates $x \rightarrow (j,z)$, j[R^{n-r}, z[R^r such that

$$\dot{\xi} = F(\xi, z) \qquad (10a)$$

$$\dot{z} = Az + E[\alpha(x(\xi,z)) + \rho(x(\xi,z))] \qquad (10b)$$

$$y\text{-}=Cz \qquad (10c)$$

Equation (10) is frequently referred to as the *local normal form* of (1). Also, in view of the above discussion and (9), it is common to refer to (10a) as the *internal dynamics* and (10b) as the *linearizable dynamics*. If z is set to zero in (10a) then we have a local representation of the zero dynamics.

Computing the Zero Dynamics

One approach to computing the local zero dynamics is to obtain the internal dynamics (10a) and then set z=0. An implementation of this calculation is described in [6]. We will describe an alternative. Note that the functions $z(x)$ can be directly computed using (3). Once they are obtained, we are in a position to compute the local form of the zero dynamics near any point x_0[}$_0$ in the following way. Without loss of generality assume $x_0 = 0$. Now, split $z(x)$ into its linear and nonlinear parts:

$$z(x) = Ax + N(x), \quad A = \frac{\partial z}{\partial x}(0) \tag{11}$$

We assume that $x_0 = 0$ is a regular point (*r* is nonsingular) so that A is of full rank. Let A^* denote a right inverse of A and define K such that its columns span $\ker A$. Define new coordinates v,w so that

$$x = A^* v + Kw \tag{12}$$

Then on the zero dynamics manifold, we have

$$v + N(A^* v + Kw) = 0 \tag{13}$$

Clearly, the Implicit Function Theorem guarantees the existence of a local solution to (13) $v^*(w)$, that is

$$v^*(w) + N(A^* v^*(w) + Kw) = 0 \tag{14}$$

on a neighborhood of $w = 0$, and $v^*(0) = 0$. Furthermore, $v^*(w)$ can be efficiently estimated because the mapping

$$v_{i+1} = -N(A^* v_i + Kw) \tag{15}$$

is a contraction. In fact, we have the following result.

Proposition 2: Suppose $z(x)$ is smooth, and A is of full rank. Then

(i) there exists a smooth function $v^*(w) = 0 + O(\|w\|)$,

(ii) if $v_i(w)$ satisfies $\|v^* - v_i\| = O(\|w\|^k)$, then $v_{i+1}(w)$, obtained via (15), satisfies

$$\|v^* - v_{i+1}\| = O(\|w\|^{2k}).$$

Proof: The first conclusion follows directly from the implicit function theorem and the fact that $v^*(w)$ is smooth. To prove the second, first subtract (15) from (14) to obtain

$$v^* - v_{i+1} = -N(A^* v^* + Kw) + N(A^* v_i + Kw) \tag{16}$$

Now, consider the function $N'(x) := \partial N(x)/\partial x$. Since N is smooth, with $\partial N(0)/\partial x = 0$, we have $N'(0) = 0$ and by continuity of the second derivative of N, we conclude that $\partial N'(x)/\partial x$ is bounded on a neighborhood of $x = 0$. Let L be such a bound on an appropriately defined neighborhood, \$, so that the usual arguments based on the Mean Value Theorem provide

$$\|N'(x) - N'(y)\| \le L\|x - y\|, \text{ for each } x, y \in \$ \tag{17}$$

Thus, we can write

$$N(x) - N(x + \delta x) = N'(x)\delta x + O(\|\delta x\|^2) \tag{18}$$

which in view of (17) gives

$$\|N(x) - N(x + \delta x)\| = O(\|\delta x\|^2), \text{ for } x, y = x + \delta x \in \$ \tag{19}$$

In order to apply this result to (16), take $x = A^* v^* + Kw$ and $\delta x = A^*(v^* - v_i)$. Then (16) and (19) yield

$$\|v^* - v_{i+1}\| = O(\|A^*(v^* - v_i)\|^2) = O(\|w\|^{2k}) \tag{20}$$

which is the desired conclusion. j

Recall the global form of the zero dynamics:

$$\dot{x} = f(x) - G(x)\rho^{-1}(x)\alpha(x)$$

which defines the zero dynamics flow everywhere on }$_0$. Near x_0 we simply project the flow onto the tangent space to }$_0$ at x_0. K has a left inverse K^* so that

$$\begin{gathered} \dot{w} = K^* f(x^*(w)) - K^* G(x^*(w))\rho^{-1}(x^*(w))\alpha(x^*(w)) \\ x^*(w) = A^* v^*(w) + Kw \end{gathered} \tag{21}$$

Assembly of the zero dynamics in the form of equation (21) via the constructions described above is typically more efficient than computing the internal dynamics (10a) because it is not necessary to solve the partial differential equations that define the transformations leading to (10a). On the other hand, if those equations can be (practically) solved the the internal dynamics may lead to a representation of the zero dynamics with a larger domain of validity.

We should also add that there are important engineering applications in which only a global characterization of the zero dynamics is useful, eg. [2, 8].

Computer Implementation

The control computations have been implemented in a *Mathematica* package which has evolved through a number of expansions and improvements [4, 5]. In its current form, the package is composed of three parts: a collection of basic geometric tools, a set of basic nonlinear control functions, and a set of advanced nonlinear control functions. Some of the available functions are summarized in Tables 1 and 2.

To compute the zero dynamics, it is first necessary to compute the linearizing control and the partial transformation that defines the linearizable states. Appendix 1 includes a *Mathematica* program for computing the zero dynamics for the example problem. It illustrates the sequence of computations.

3. Example

In this section we present an example based on a simple ground vehicle similar to that described in [6]. Our intent is to illustrate the computation of the zero dynamics and also to highlight certain issues regarding zero dynamics, the understanding of which is central to the application of variable structure control — or any other form of decoupling control, for that matter.

Table 1

Some Geometric Tools

```
Grad::usage=
"Grad[f,varlist] computes the gradient of the
scalar function f with respect to the variables
varlist."

Jacob::usage=
"Jacob[flist,varlist] computes the Jacobian of
the functions flist with respect to the
variables varlist."

LieBracket::usage=
"LieBracket[f,g,varlist] computes the Lie
Bracket of the vector functions f,g with respect
to the variables varlist."

Ad::usage=
"Ad[f,g,varlist,n] computes the nth Adjoint
(iterated Lie Bracket) of the vector fields f,g
with respect to the variables varlist.
Ad[f,g,var,0]=g
Ad[f,g,var,n]=LieBracket[f,Ad[f,g,var,n-1],var]
Ad[f,g,var]=Ad[f,g,var,1] "

LieDerivative::usage=
"LieDerivative[f,h,x] computes the Lie
derivative of the real valued function h of the
vector x along the direction defined by the
vector field f:
  LDf[h](x) = < dh(x),f(x) >

LieDerivative[f,h,x,k] computes the k-th order
Lie derivative of the real valued function h of
the vector x along the direction defined by the
vector field f:
  LD^k f[h](x) = LD^(k-1) f[LDf[h]](x)
  LD^0 f[h](x) = h(x) "

Involutive::usage=
"Involutive[set,x] tests a set of vector fields
to determine if it is involutive. It reports the
result as a value True or False. The argument
`set' must be a list of vector fields, and x
must be a list of the variables."
```

The simple vehicle that we will consider is illustrated in the figure 1.

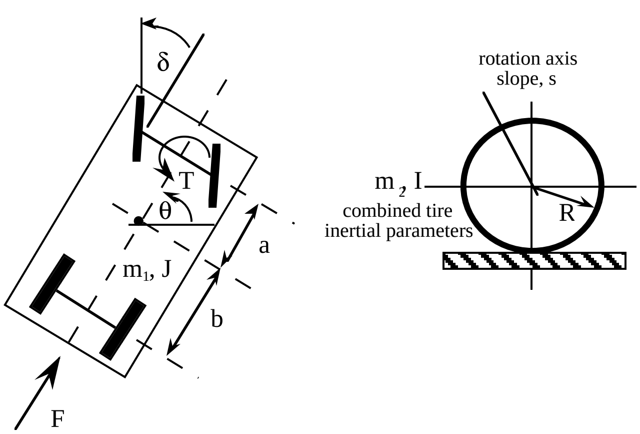

Figure 1. The essential parameters of the example are illustrated in this figure. The center of mass of the main body is located by the coordinates *x* and *y*. Its attitude of is *u*. The front wheels rotate an amount *d* about an axis of slope *s*, which is (*s*=0, results in a vertical axis), s is assumed small as are the tire inertial parameters.

Table 2

Basic Control Functions

```
ControllabilityMatrix::usage =
"ControllabilityMatrix[f,g,x] computes the set
of vectors
  Table[Ad[f,g[i],x,k],{k,0,Length[x]-1},
                        {i,1,Length[g]}]"

Controllable::usage=
"Controllable[f,g,x] tests the pair (f,g) to
determine if the system is locally controllable,
that is,does the controllabilty matrix have full
rank? It returns True or False."

FeedbackLinearizable::usage=
"FeedbackLinearizable[f,g,x] tests to see if the
pair (f,g) is exactly linearizable by means of
feedback and a change of coordinates. The pair
(f,g) must be controllable, and the set of
vector fields

  Table[Ad[f,g,x,k],{k,0,Length[x]-2},
                    {i,1,Length[g]}]

must be involutive. The function returns True or
False."

VectorRO::usage=
"VectorRO computes the vector relative order if
a MIMO system. VectorRO[f,g,h,x] where f,g,h can
be functions of x or defined explicitly as lists
of expressions in x"

DecouplingMatrix::usage=
"DecouplingMatrix[f,g,h,x,ro] computes the
decoupling matrix. f,g,h can be functions of x
or lists of expressions in x. x is a list and ro
is the vector relative degree. See VectorRO."

IOLinearize::usage=
"IOLinearize[f,g,h,x] computes a feedback
linearizing & decoupling control. It returns
  {DecoupMat, Nonlin, RelOrder, test1, control}
where,
  DecoupMat - decoupling matrix,
  Nonlin - vector of
    LieDerivative[f,h[[i]],x, RelOrder[[i]]],
  RelOrder - relative order,
  test1 - regions where the decoupling matrix is
          singular,
  control - decoupling control law."

NormalCoordinates::usage=
"NormalCoordinates[f,g,h,x,Vro] returns the
functions z(x) which define the linearizable
states."

LocalZeroDynamics::usage=
"LocalZeroDynamics[f,g,h,x,u0,z] returns a local
representation of  the zero dynamics about the
origin (x=0)   "
```

The following model incorporates two simplifications; $m_2 = 0$, $s \ll 1$, so that only first order terms in s are included.

$$\begin{bmatrix} \dot{\theta} \\ \dot{x} \\ \dot{y} \\ \dot{\delta} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \omega_\theta \\ v_x \\ v_y \\ \omega_\delta \end{bmatrix} \tag{22a}$$

$$\begin{bmatrix} I_{zz}+J_{zz} & 0 & 0 & I_{zz} \\ 0 & m_1 & 0 & 0 \\ 0 & 0 & m_1 & 0 \\ I_{zz} & 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \dot{\omega}_\theta \\ \dot{v}_x \\ \dot{v}_y \\ \dot{\omega}_\delta \end{bmatrix} + \begin{bmatrix} 0 \\ m_1 v_y \omega_\theta \\ -m_1 v_x \omega_\theta \\ 0 \end{bmatrix} + \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} = 0 \tag{22b}$$

The functions f_i are given in the appendix. These equations were derived using a *Mathematica* package for multibody dynamics [12].

Our goal is to consider the problem of steering the vehicle along a path of constant radius, and at constant speed V_d. There are several ways of formulating this problem. One common approach is to replace the constant radius condition by the requirement that the angular velocity ω_θ is a constant, say ω_d. This leads to a constant curvature path of radius, with $R = V_d / \omega_d$. Thus, we introduce two output relations

$$\begin{aligned} y_1 &= v_x^2 + v_y^2 - V_d^2 \\ y_2 &= \omega_\theta - \omega_d \end{aligned} \tag{23}$$

We are interested in the zero dynamics relative to these two outputs and the two controls T, F. Notice that in this formulation it is not necessary to retain the kinematic equations which define the vehicle location and orientation in the plane — i.e., θ, x, y. Thus, the system equations include (22b) and only the last equation of (22a).

We first compute the zero dynamics for the case of motion along a straight path, $\omega_d = 0, R = \infty$. The vector relative degree is found to be [1, 1]. Therefore, the zero dynamics involve three first order differential equations in the zero dynamics 'state' variables w_1, w_2, w_3. Up to fourth order terms, these equations are:

```
w1dot={w2},

w2dot={(kappa*(2*a + R*s)*w1)/(2*Izz)
     -(kappa*(a + R*s)*w1^3)/(2*Izz)
     +(kappa*(-2*a + 2*b - R*s)*w3)/(2*Izz*Vd)
     +(kappa*(-2*a + 2*b - R*s)*w3^3)/(12*Izz*Vd^3)
     + w1^2*((kappa*(a + R*s)*w3)/(2*Izz*Vd))
     +w2*((a*kappa*R*s)/(2*Izz*Vd)
     +(a*kappa*R*s*w1*w3)/(2*Izz*Vd^2)
     -(a*kappa*R*s*w3^2)/(4*Izz*Vd^3)
     +w1^2*(-(a*kappa*R*s)/(2*Izz*Vd))},

w3dot={(kappa*w1)/m1 - (kappa*w1^3)/(2*m1)
     -(2*kappa*w3)/(m1*Vd)-(kappa*w3^3)/(3*m1*Vd^3)
     +w1^2*((kappa*w3)/(2*m1*Vd))
     +w2*((kappa*R*s)/(2*m1*Vd)
     +(kappa*R*s*w1*w3)/(2*m1*Vd^2)
     - (kappa*R*s*w3^2)/(4*m1*Vd^3)
     - (kappa*R*s*w3^4)/(16*m1*Vd^5)
     +w1^2*(-(kappa*R*s)/(2*m1*Vd)))}
```

We can test the stability of the equilibrium point $w = 0$, by examining the linearized zero dynamics:

$$\dot{w} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{\kappa(2a+Rs)}{2I_{zz}} & \frac{a\kappa Rs}{2I_{zz}V_d} & \frac{2\kappa(b-a)-\kappa Rs)}{2I_{zz}V_d} \\ \frac{\kappa}{m_1} & \frac{\kappa Rs}{2m_1V_d} & \frac{-2\kappa}{m_1V_d} \end{bmatrix} w \tag{24}$$

The eigenvalues are readily obtained but they are lengthy functions of the parameters. Some insight is obtained, however, by examining the special case, $a = b$ and $s = 0$, in which case the eigenvalues simplify to:

$$\lambda_1 = -\frac{2\kappa}{m_1 V_d}, \lambda_{2,3} = \pm\frac{\sqrt{a\kappa}}{\sqrt{I_{zz}}} \tag{25}$$

Hence, we see that the zero dynamics are unstable. Because the eigenvalues vary smoothly as a function of parameters, this situation will be true for *a-b* and *s* small, but not necessarily zero. Furthermore, since I_{zz} is small, $\lambda_{2,3}$ are a pair of 'parasitic' zeros, one of which is far into the right half plane, the other to the left. Such circumstances occur frequently and need not fatally obviate the use of variable structure or feedback linearizing control [13, 14]. The physical explanation in the present situation is a weak control force introduced by the small inertial cross–coupling produced by the off diagonal I_{zz} terms in the inertia matrix.

4. Conclusions

In this paper, we have described a method for symbolic computation of the local form of the zero dynamics for affine systems with well defined vector relative degree. The method is based on the characterization of zero dynamics given in [2]. It is efficient because it avoids computing the transformation relations which reduce the system to local normal form. An implementation in *Mathematica* has been described and an example of its application given.

Understanding zero dynamics behavior is essential in variable structure control system design and any other design method which directly or indirectly exploits decoupling. In applications, it is necessary to investigate parametric influences on the zero dynamics and the zero dynamics often constrain the envelope of operation of nonlinear feedback systems.

5. References

1. Marino, R., "High Gain Feedback Non–Linear Control Systems," *International Joural of Control*, 1985, 42(6): pp. 1369–1385.
2. Kwatny, H.G. and H. Kim, "Variable Structure Regulation of Partially Linearizable Dynamics," *Systems & Control Letters*, 1990, 15: pp. 67–80.
3. Kwatny, H.G. and J. Berg, *Variable Structure Regulation of Power Plant Drum Level*, in *Systems and Control Theory for Power Systems*, 1993, Springer–Verlag: New York.
4. Akhrif, O. and G.L. Blankenship, "Computer Algebra for Analysis and Design of Nonlinear Control Systems," in *Proceedings American Control Conference*, 1987, Minneapolis, MN: IEEE, pp. 547–554.
5. Blankenship, G.L., R. Ghanadan, H.G. Kwatny, and V. Polyakov, "Modeling and Design Tools for Control of Multibody Systems," in *Proceedings ASME Winter Annual Meeting: High Performance Computing in Vehicle Systems*, 1993, New Orleans: pp.
6. de Jager, B., "The Use of Symbolic Computation in Nonlinear Control: Is it Viable?," in *Proceedings IEEE*

Conference on Decision and Control, 1993, San Antonio, TX: IEEE, pp. 276–281.

7. Isidori, A., *Nonlinear Control Systems*. 1989, NY: Springer-Verlag.

8. 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.

9. Luk'yanov, A.G. and V.I. Utkin, “Methods of Reducing Equations of Dynamic Systems to Regular Form,” *Avtomatica i Telemechanika*, 1981, (4): pp. 5–13.

10. Descusse, J. and C.H. Moog, “Decoupling with Dynamic Compensation for Strong Invertible Affine Non–Linear Systems,” *International Journal of Control*, 1985, 42(6): pp. 1387–1398.

11. Huijberts, H.J.C., H. Nijmeijer and L.L.M.v.d. Wegen, “Minimality of Dynamics Input–Output Decoupling for Nonlinear Systems,” *Systems and Control Letters*, 1992, 18(6): pp. 435–443.

12. LaVigna, C., H.G. Kwatny and G.L. Blankenship,“Flexible Multibody Dynamical Analysis System,” Techno–Sciences, Inc., 1993, Phase I Final Report, Contract No. DAAE07–93–CR022.

13. Hauser, J., S. Sastry and P. Kokotovic, “Nonlinear Control Via Approximate Input-Output Linearization: The Ball and Beam Example,” *IEEE Transactions on Automatic Control*, 1992, AC-37(3): pp. 392-398.

14. Lane, S.H. and R.F. Stengel, “Flight Control Design Using Non-linear Inverse Dynamics,” *Automatica*, 1988, 24(4): pp. 471-483.

Appendix 1: Program to Generate Zero Dynamics for Example problem

```
<<NCBasics`  (* Nonlinear Control Package *)
<<GeoTools`  (* Diff Geometry Tools *)
(* compute relative degree *)
ro=VectorRO[f,g,h,var];
(* compute feedback ,linearizing/decoupling
control *)
{R1,R2,R3,R4,u}=IOLinearize[f,g,h,var];
(* compute linearizable coordinates *)
z=NormalCoordinates[f,g,h,var,ro]/.{wd->0};
(* shift origin to point of interest *)
{f,g,h,u,z}={f,g,h,u,z}/.{x2->x2+Vd};
(* compute zero dynamics *)
u0=u/.{v1->0,v2->0};
f0=LocalZeroDynamics[f,g,h,var,u0,z];
(* linearize zero dynamics and determine
stability of origin *)
Anu=Jacob[f0,{w1,w2,w3}]/.{w1->0,w2->0,w3->0,b-
>a+nu};
Eigenvalues[Anu/.{nu->0,s->0}]
```

Appendix 2: Equations Used in Example Problem

```
                      vy - b wth
f1 = -(b kappa ArcTan[----------])
                          vx

                    vy Cos[delta] + a wth Cos[delta] - vx Sin[delta]
  + a kappa ArcTan[ ------------------------------------------------] Cos[delta]
                    vx Cos[delta] + vy Sin[delta] + a wth Sin[delta]

                     vy Cos[delta] + a wth Cos[delta] - vx Sin[delta]
  +(kappa s(R ArcTan[------------------------------------------------](-1 + 2 Cos[delta])
                     vx Cos[delta] + vy Sin[delta] + a wth Sin[delta]


  + (a R Cos[delta] (2 vx wth - vx wdel Cos[delta] - vx wth Cos[delta] - vy wdel Sin[delta]
                                                     2                   2     2                  2    2
     - vy wth Sin[delta] - a wdel wth Sin[delta] - a wth  Sin[delta])) / (vx  + vy  + 2 a vy wth + a  wth ))) / 2,


                        vy Cos[delta] + a wth Cos[delta] - vx Sin[delta]
f2 = -F - kappa ArcTan[ ------------------------------------------------] Sin[delta]
                        vx Cos[delta] + vy Sin[delta] + a wth Sin[delta]

    + (kappa R s Sin[delta](-2 vx wth + vx wdel Cos[delta] + vx wth Cos[delta]
                                                                 2                    2     2                    2    2
    + vy wdel Sin[delta] + vy wth Sin[delta] + a wdel wth Sin[delta] + a wth  Sin[delta]))/(2 (vx  + vy  + 2 a vy wth + a  wth )),


                  vy - b wth
f3 = kappa ArcTan[----------]
                      vx

                  vy Cos[delta] + a wth Cos[delta] - vx Sin[delta]
   + kappa ArcTan[------------------------------------------------] Cos[delta]
                  vx Cos[delta] + vy Sin[delta] + a wth Sin[delta]

   + (kappa R s Cos[delta] (2 vx wth - vx wdel Cos[delta] - vx wth Cos[delta]
                                                                                2     2                    2    2
   - vy wdel Sin[delta] - vy wth Sin[delta] - a wdel wth Sin[delta] - a wth  Sin[delta]))/ (2 (vx  + vy  + 2 a vy wth + a  wth )),


                            vy Cos[delta] + a wth Cos[delta] - vx Sin[delta]
f4 = -T - (kappa R s ArcTan[ ------------------------------------------------]) / 2}
                            vx Cos[delta] + vy Sin[delta] + a wth Sin[delta]
```