# Quotient method for controlling the acrobot

S. S. Willson, Philippe Mullhaupt and Dominique Bonvin

***Abstract*— This paper describes a two-sweep control design method to stabilize the acrobot, an input-affine under-actuated system, at the upper equilibrium point. In the forward sweep, the system is successively reduced, one dimension at a time, until a two-dimensional system is obtained. At each step of the reduction process, a quotient is taken along one-dimensional integral manifolds of the input vector field. This decomposes the current manifold into classes of equivalence that constitute a quotient manifold of reduced dimension. The input to a given step becomes the representative of the previous-step equivalence class, and a new input vector field can be defined on the tangent of the quotient manifold. The representatives remain undefined throughout the forward sweep. During the backward sweep, the controller is designed recursively, starting with the two-dimensional system. At each step of the recursion, a well-chosen representative of the equivalence class ahead of the current level of recursion is chosen, so as to guarantee stability of the current step. Therefore, this stabilizes the global system once the backward sweep is complete. Although stability can only be guaranteed locally around the upper equilibrium point, the domain of attraction can be enlarged to include the lower-equilibrium point, thereby allowing a swing-up implementation. As a result, the controller does not require switching, which is illustrated in simulation. The controller has four tuning parameters, which helps shape the closed-loop behavior.**

## I. INTRODUCTION

Any system with fewer actuators than the number of configuration variables can be regarded as an under-actuated system. There are numerous control approaches for nonlinear systems ([1], [2], [3], [4]), but many of them cannot satisfactorily control under-actuated systems. The control of under-actuated nonlinear systems has been a challenge due to the complexity of the internal dynamics. In most cases, these systems are not feedback linearizable. In contrast, there is no difficulty with the control of linear under-actuated systems as long as they are controllable.

The acrobot is an under-actuated system [5]. Control in the neighborhood of the upper equilibrium point can be achieved by linearizing the model around that point. Methods such as Linear Quadratic Regulator (LQR) are adequate to control the resulting linear model, but the domain of attraction is often small. As a consequence, it is not possible to achieve swing-up using the linear model only.

The acrobot is not feedback linearizable [2]. Nevertheless, controllers have been designed based on pseudo–linearization [6] and approximate state-feedback linearization [7]. These methods try to achieve feedback linearization around a trajectory. Furthermore, swing-up through partial feedback linearization has been proposed [8], which leaves marginally stable zero-dynamics and hence is incapable of stabilizing the system. Energy-based swing-up methods have also been suggested for the acrobot [8], [9]. With all these methods, switching to a linear controller is needed when the system reaches the upright equilibrium position. Control of the acrobot has also been attempted using the interconnection and damping assignment (IDA) method [10].

For a similar under-actuated problem, namely the inverted pendulum, a global strategy that can achieve both swing-up and balancing has been suggested in [11]. The method relies on (i) changing the reference for the pendulum in order to pump the required energy for the swing-up, and (ii) using a singular perturbation approach to stabilize the cart. The control strategy is developed only for a neighborhood of the upright position, and switching of the reference value is used to reach that region from any given initial condition.

In the present work, a nonlinear controller will be designed to locally asymptotically stabilize the acrobot in the neighborhood of the upwards position. The domain of attraction will be tuned in order to include the lower equilibrium point so that the controller can achieve both swing-up and balancing. This is achieved by first constructing quotient manifolds to achieve approximate feedback form [12], which decomposes the system into subsystems of lower dimensions. Then, these small subsystems are stabilized one by one, thereby leading to the full controller. This method is based on feedback linearization [12], [13], but it differs from pseudo-linearization [6] and approximate linearization [7], where the approximation is done at the transformation stage to alleviate the hindrance arising from the Frobenius condition [2]. In the proposed method, the approximation is done at the controller design stage to adequately handle the fact that the system does not satisfy the condition for feedback linearization.

The next section introduces the acrobot model. Section III describes the forward and backward sweeps required for control design. Section IV presents simulation results for the controlled system, while concluding remarks are presented in the last section.

## II. ACROBOT DYNAMICS

The model of the acrobot is derived in [8]. It has two configuration variables, $q = [q_1, q_2]^T \in \mathbb{R}^2$, defined on the configuration space $Q = \mathbb{S}^1 \times \mathbb{S}^1$ and the input torque $u$. The equation of motion can be written in the general form

$$M(q)\ddot{q} + h(q, \dot{q}) + \phi(q) = \tau$$

Laboratoire d'Automatique, Station 9, Ecole Polytechnique Federale de Lausanne, CH-1015 Lausanne, Switzerland. willson.shibani@epfl.ch philippe.muellhaupt@epfl.ch dominique.bonvin@epfl.ch

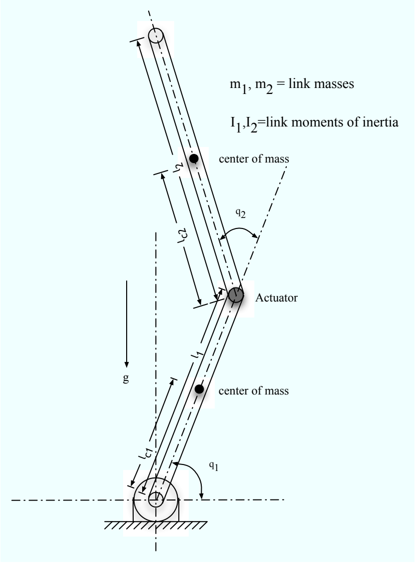


Fig. 1. The acrobot

where $M(q)$ denotes the positive-definite inertia matrix, vector $h(q, \dot{q})$ the centrifugal and Coriolis components, and $\phi(q)$ the gravitational component. The input vector $\tau = [0, u]^T$ represents the generalized forces. Defining the state vector as $\mathbf{x} = (x_1, x_2, x_3, x_4)^T = (q_1 - \pi/2, q_2, \dot{q_1}, \dot{q_2})^T$ [9], and representing the equations of motion as

$$\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})u \tag{1}$$

gives the two smooth vector fields $f(\mathbf{x})$ and $g(\mathbf{x})$:

$$f(\mathbf{x}) \quad = \quad \begin{pmatrix} x_3 \\ x_4 \\ f_3 \\ f_4 \end{pmatrix}, \quad g(\mathbf{x}) = \begin{pmatrix} 0 \\ 0 \\ g_3 \\ g_4 \end{pmatrix}$$

where

$$\begin{aligned}
f_3 \quad &= \quad \frac{1}{\gamma(x_2)}\{c_2c_3(x_3+x_4)^2\sin x_2 + c_2c_4\sin x_1 \\
&+ \quad c_3^2x_3^2\cos x_2\sin x_2 - c_3c_5g\sin(x_1+x_2)\cos x_2\} \\
f_4 \quad &= \quad \frac{1}{\gamma(x_2)}\{-(c_2+c_3\cos x_2)(2x_3x_4+x_4^2)c_3\sin x_2 \\
&- \quad c_2c_4g\sin x_1 - c_3c_4g\cos x_2\sin x_1 \\
&+ \quad c_1c_5g\sin(x_1+x_2) + c_3c_5g\cos x_2\sin(x_1+x_2) \\
&- \quad (c_1+c_2+2c_3\cos(x_2))x_3^2c_3\sin x_2\} \\
g_3 \quad &= \quad \frac{-(c_2+c_3\cos x_2)}{\gamma(x_2)} \\
g_4 \quad &= \quad \frac{c_1+c_2+2c_3\cos x_2}{\gamma(x_2)} \\
\gamma(x_2) \quad &= \quad c_1c_2 - c_3^2\cos^2 x_2.
\end{aligned}$$

The acrobot parameters are collected to give the following constants:

$$c_1 = m_1l_{c1}^2 + m_2l_1^2 + I_1 \quad c_2 = m_1l_{c2}^2 + I_2 \quad c_3 = m_2l_1l_{c2}$$
$$c_4 = m_1l_{c1} + m_2l_1 \qquad c_5 = m_2l_{c2}$$

The acrobot has four equilibrium points corresponding to $u = 0$, of which three are unstable, $(0,0,0,0)$, $(0,\pi,0,0)$ and $(\pi,\pi,0,0)$, and one is stable, $(\pi,0,0,0)$. The system is not feedback linearizable [2] since the distribution $\Delta =$span$\{g, [f,g], [f,[f,g]]\}$ is not involutive [9].

## III. CONTROLLER DESIGN

The controller design consists of two stages or sweeps. The first stage builds a strict feedback form for the equations of motion by successively constructing diffeomorphisms. The second stage iteratively constructs a controller for this feedback form.

### A. Forward Sweep

The forward sweep proceeds iteratively by producing a succession of systems, each one being one unit smaller in dimension than the previous one, until a smaller manageable system is obtained. These systems are then stabilized during the backward sweep.

To illustrate a reduction step, assume that it is applied when the system is described by $r$ states, that is,

$$\mathbf{x} = \begin{pmatrix} x_1 & x_2 & x_3 & \dots & x_r \end{pmatrix}^T$$

and it can be written as

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

One constructs the diffeomorphism $\mathbf{z} = \Phi(\mathbf{x})$ that cancels the first $r-1$ components of the transformed vector field $\frac{\partial\Phi}{\partial\mathbf{x}}g(\mathbf{x})$. The diffeomorphism is constructed by finding $r-1$ independent solutions $\phi_1(\mathbf{x})$, $\phi_2(\mathbf{x})$, ..., $\phi_{r-1}(\mathbf{x})$ of the following system (termed the Lagrange subsidiary system [14]):

$$\frac{dx_1}{g_1} = \frac{dx_2}{g_2} = \dots = \frac{dx_r}{g_r}$$

To find all solutions, one selects an independent variable and rewrite the above system as a system of $r-1$ first-order ordinary differential equations in $r-1$ dependent variables. A set of $r-1$ independent solutions, $\phi_1(\mathbf{x})$, $\phi_2(\mathbf{x})$, ..., $\phi_{r-1}(\mathbf{x})$, is complemented with the $r$-th component $\psi(\mathbf{x})$ chosen such that

$$\frac{\partial\psi}{\partial\mathbf{x}}g(\mathbf{x}) \neq 0. \tag{2}$$

The diffeomorphism then becomes:

$$\Phi(\mathbf{x}) = \begin{pmatrix} \phi_1(\mathbf{x}) & \phi_2(\mathbf{x}) & \cdots & \phi_{r-1}(\mathbf{x}) & \psi(\mathbf{x}) \end{pmatrix}^T.$$

The consequence of these operations is that the one-dimensional integral manifold of the vector field $g(\mathbf{x})$ (a twisted one-dimensional curve in the $\mathbf{x}$-space) becomes a straight line in the $\mathbf{z}$-space aligned along the $z_r$-direction. Each line parallel to the $z_r$ axis corresponds to an equivalence class in the space described by the states $x_1$, $x_2$, $x_3$, ..., $x_r$. Each class corresponds to a one-dimensional integral manifold of the input vector field $g(\mathbf{x})$ and is described by the value of the $r-1$ states $z_1$, $z_2$, ..., $z_{r-1}$. The choice of a value of $z_r$ (defined as $z_r = \psi(x_1, x_2, x_3, \dots, x_r)$)

corresponds to the choice of a representative of the equivalence class associated with $z_1$, $z_2$, ..., $z_{r-1}$. This is similar to the classical construction of the quotient group $\mathbb{Z}_4 = \{\bar{0}, \bar{1}, \bar{2}, \bar{3}\}$, a finite group that is defined as the quotient of two infinite groups $\mathbb{Z}$ and $4\mathbb{Z}$ i.e $\mathbb{Z}_4 = \mathbb{Z}/4\mathbb{Z}$. Each element of $\mathbb{Z}_4$ stands for an associated infinite set, e.g. $\bar{1}$ stands for $\{\ldots, -7, -3, 1, 5, 9, \ldots\}$. The key idea is that one can choose the representative of $\bar{1}$ as either 1, 5 or $-7$ without affecting the value of $\bar{1}$. For the case of our dynamic system, the quotient elements are represented by $z_1$, $z_2$, ..., $z_{r-1}$, and a representative of the equivalence class associated with specific values of $z_1$, $z_2$, ..., $z_{r-1}$ is given by $z_r = \psi(x_1, x_2, x_3, \ldots, x_r)$. As in the previous example, where the choice of 1, 5 or $-7$ does not affect $\bar{1}$, there are many ways of representing $z_1$, $z_2$, ..., $z_{r-1}$ using $x_1$, $x_2$, $x_3$, ..., $x_r$. This choice is left open throughout the forward sweep, which will provide useful degrees of freedom in the backward (controller-design) sweep.

*Remark 1:* It is precisely the property that the choice of the representative does not affect the equivalence class which allows differing controller design until after the reduction stage is complete, i.e. without affecting the forward decomposition sweep.

The procedure is systematic for *feedback linearizable systems*, for which the following two properties are guaranteed at each step of the forward reduction [12]:

(i) Existence of a well-defined equivalence class, i.e. the choice of the representatives is independent of the forthcoming reduction steps.

(ii) Existence of a suitable input vector field along the tangent of the quotient manifold. This vector field is determined by grouping terms in the transformed vector field $\frac{\partial \Phi}{\partial \mathbf{x}} f \circ \Phi^{-1}(\mathbf{z})$, obtained after projection, so that it becomes the sum of a vector field independent of the equivalence class described by $z_r = \psi$ and a term that is the product of another vector field independant of $z_r$ times $z_r$ (i.e. the reduced system with state-space coordinates $z_1$, $z_2$, ..., $z_{r-1}$ is affine in the new input $z_r$). That is, it must be possible to express

$$\begin{aligned} Pr^{[r]} \quad & (\frac{\partial \Phi}{\partial \mathbf{x}} f \circ \Phi^{-1}(\mathbf{z})) \\ = \quad & \hat{f}(z_1, \ldots, z_{r-1}) + \hat{g}(z_1, \ldots, z_{r-1}) z_r \end{aligned}$$

Where $Pr^{[r]}$ is defined as.

*Definition 1:* The projection map of order $r$, $Pr^{[r]} : \mathbb{R}^r \to \mathbb{R}^{r-1}$, is defined as $Pr^{[r]}((a_1, a_2, ..., a_{r-1}, a_r)^T) = (a_1, a_2, ..., a_{r-1})^T$.

Let us compute the successive steps for the non-feedback-linearizable acrobot and see to what extent the Frobenius integrability condition [12] hinders meeting Properties (i) and (ii) described above. We will also explain how this hindrance can be overcome. The algorithm is initiated with $g(\mathbf{x}) = (0, 0, g_3, g_4)^T$, i.e. with $r = 4$. Property (i) holds since the system (1) has an equivalence class defined by a one-dimensional manifold whose tangent is $g(\mathbf{x})$. To determine the quotient manifold, the first step is to solve

$$\frac{\partial \phi_i}{\partial \mathbf{x}} g(\mathbf{x}) = 0 \quad \forall\, i = 1, 2, 3, \tag{3}$$

for example, using the Lagrange subsidiary equations

$$\frac{dx_1}{0} = \frac{dx_2}{0} = \frac{dx_3}{g_3} = \frac{dx_4}{g_4}.$$

This set of equations becomes a system of ordinary differential equations with, for example, the independent variable chosen as $x_4$, while $x_1$, $x_2$, and $x_3$ represent the dependent variables. Because the first two denominators are zero, the system

$$\begin{aligned} dx_1 &= 0 \\ dx_2 &= 0 \\ g_4 dx_3 &= g_3 dx_4 \end{aligned}$$

can be solved one equation at a time, which yields solutions that satisfies (3). Furthermore, let us choose $\psi(\mathbf{x}) = x_3$ to satisfy (2). It follows from

$$\Phi(\mathbf{x}) = \begin{pmatrix} \phi_1(\mathbf{x}) \\ \phi_2(\mathbf{x}) \\ \phi_3(\mathbf{x}) \\ \\ \psi(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} x_1 \\ x2 \\ (c_1 + c_2 + 2c_3 \cos x_2) x_3 \\ +(c_2 + c_3 \cos x_2) x_4 \\ x_3 \end{pmatrix} \tag{4}$$

that

$$\text{rank}\left( \frac{\partial \Phi(\mathbf{x})}{\partial \mathbf{x}} \right) = 4.$$

Now define $\mathbf{z} \triangleq \mathbf{\Phi}(\mathbf{x})$ with $\mathbf{z} \in \mathbb{R}^4$ so that (4) is a diffeomorphism with the inverse $\mathbf{x} = \mathbf{\Phi}^{-1}(\mathbf{z})$:

$$\begin{aligned} \mathbf{z} &= \mathbf{\Phi}(\mathbf{x}) \\ \dot{\mathbf{z}} &= \frac{\partial \mathbf{\Phi}(\mathbf{x})}{\partial \mathbf{x}} f(\mathbf{x})|_{\mathbf{x} = \mathbf{\Phi}^{-1}(\mathbf{z})} + \frac{\partial \mathbf{\Phi}(\mathbf{x})}{\partial \mathbf{x}} g(\mathbf{x})|_{\mathbf{x} = \mathbf{\Phi}^{-1}(\mathbf{z})}\, u \\ &= f_z(\mathbf{z}) + g_z(\mathbf{z}) u \end{aligned} \tag{5}$$

By construction of $\mathbf{\Phi}(\mathbf{x})$, the first 3 components of $g_z(\mathbf{z})$ are zero, i.e. $g_z(\mathbf{z}) = (0, 0, 0, *)^T$.

Application of the projection map of order 4 to System (5),

$$Pr^{[4]}(\dot{\mathbf{z}}) = Pr^{[4]}(f_z(\mathbf{z}) + g_z(\mathbf{z}) u),$$

removes the $4^{th}$ component of the state. Since $Pr^{[4]}(g_z(\mathbf{z})) = (0, 0, 0)$, we obtain the following system:

$$\begin{aligned} \begin{pmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \end{pmatrix} &= \begin{pmatrix} f_{z1}(\mathbf{z}) \\ f_{z2}(\mathbf{z}) \\ f_{z3}(\mathbf{z}) \end{pmatrix} \\ &= \begin{pmatrix} z_4 \\ \frac{-z_3 + (c_1 + c_2 + 2c_3 \cos z_2) z_4}{c_2 + c_3 \cos z_2} \\ g(c_4 \sin z_1 + c_5 \sin(z_1 + z_2)) \end{pmatrix} \end{aligned} \tag{6}$$

Here, $\hat{\mathbf{z}} \triangleq (z_1, z_2, z_3)^T \in \mathbb{R}^3$ defines a point on the quotient manifold, that is, a specific equivalence class. A representative of the equivalence class is given by a particular value of the $z_4$ coordinate. Hence, Property (i) is fulfilled.

Note that, by construction, the input vector field $g_z(\mathbf{z})$ only affects the time derivative of the representative coordinate $z_4$:

$$\dot{z}_4 = f_{z4}(\mathbf{z}) + g_{z4}(\mathbf{z})u \tag{7}$$

with $f_{z4}(\mathbf{z})$ and $g_{z4}(\mathbf{z})$ the $4^{th}$ elements of $f_z(\mathbf{z})$ and $g_z(\mathbf{z})$ defined in (5).

The image of the input vector field $g_z(\mathbf{z})$ on the tangent of the quotient manifold is zero. Note that (6) is affine in $z_4$ even though $f_3$ and $f_4$ contain quadratic terms in $x_3$ and $x_4$. That is, the diffeomorphism has eliminated the quadratic terms in $f_z(\mathbf{z})$, which was possible because the distribution $\Delta = \text{span}\{g, [f, g]\}$ is involutive ( [12] [13]). Hence, Property (ii) is also fulfilled, and the quotient dynamics read:

$$\dot{\hat{\mathbf{z}}} = \hat{f}_z(\hat{\mathbf{z}}) + \hat{g}_z(\hat{\mathbf{z}})z_4 \tag{8}$$

with

$$\begin{aligned} \hat{f}_z(\hat{\mathbf{z}}) &= \begin{pmatrix} 0 \\ \frac{-z_3}{c_2+c_3\cos z_2} \\ g(c_4 \sin z_1 + c_5 \sin(z_1+z_2)) \end{pmatrix} \\ \hat{g}_z(\hat{\mathbf{z}}) &= \begin{pmatrix} 1 \\ \frac{(c_1+c_2+2c_3\cos z_2)}{c_2+c_3\cos z_2} \\ 0 \end{pmatrix}, \end{aligned}$$

and the coordinate $z_4$ acting as the new input.

To obtain the next quotient manifold, the Lagrange subsidiary system for $\hat{g}_z(\hat{\mathbf{z}})$ is considered:

$$\frac{dz_1}{1} = \frac{(c_2 + c_3 \cos z_2)dz_2}{(c_1 + c_2 + 2c_3 \cos z_2)} = \frac{dz_3}{0},$$

which yields the following set of differential equations to be solved:

$$\begin{aligned} dz_1 &= \frac{(c_2 + c_3 \cos z_2)dz_2}{(c_1 + c_2 + 2c_3 \cos z_2)} \\ dz_3 &= 0. \end{aligned}$$

Solving these equations gives the first two functions of the next diffeomorphism, while the third function is chosen to satisfy (2):

$$\mathbf{\Phi}_z(\hat{\mathbf{z}}) = \begin{pmatrix} z_1 - \zeta(z_2) \\ z_3 \\ z_2 \end{pmatrix}$$

where

$$\zeta(z_2) = \int \frac{(c_2 + c_3 \cos z_2)}{(c_1 + c_2 + 2c_3 \cos z_2)} dz_2.$$

Let $\mathbf{y} = (y_1, y_2, y_3)^T \triangleq \mathbf{\Phi}_z(\hat{\mathbf{z}})$, which gives:

$$\hat{\mathbf{z}} = \mathbf{\Phi}_z^{-1}(\mathbf{y}) = \begin{pmatrix} y_1 + \zeta(y_3) \\ y_3 \\ y_2 \end{pmatrix}.$$

Applying the projection map gives

$$\begin{aligned} \dot{y}_1 &= \frac{y_2}{c_1 + c_2 + 2c_3 \cos y_3} && (9) \\ \dot{y}_2 &= g(c_4 \sin(y_1 + \zeta(y_3)) \\ &+ c_5 \sin(y_1 + \zeta(y_3) + y_3)), && (10) \end{aligned}$$

where $\hat{\mathbf{y}} = (y_1, y_2)^T$ defines a point on the quotient manifold and $y_3$ is a representative of the equivalence class. Property (i) holds since the system (8) has an equivalence class defined by a one-dimensional manifold whose tangent is $\hat{g}_z(\hat{\mathbf{z}})$.

Since equations (9) and (10) are not affine in $y_3$, Property (ii) is not satisfied.

The dynamics inside the equivalence classes determine the rate at which the representative $y_3$ changes. This rate depends on the coordinate $y_2$ and $y_3$ of the point on the quotient manifold:

$$\dot{y}_3 = \frac{y_2}{c_2 + c_3 \cos y_3} + \frac{(c_1 + c_2 + 2c_3 \cos y_3)}{(c_2 + c_3 \cos y_3)} z_4. \tag{11}$$

Note that equations (9) and (10) are similar to those for a free fall under gravity, where the gravitational acceleration term is manipulated using the input $y_3$. The variables $y_1$ and $y_2$ are expressed using the original state space as

$$\begin{aligned} y_1 &= q_1 - \zeta(q_2) \\ y_2 &= (c_1 + c_2 + 2c_3 \cos(q_2))\dot{q}_1 + (c_2 + c_3 \cos(q_2))\dot{q}_2 \end{aligned}$$

Equations (9), (10), (11) and (7) represent an alternate feedback form of the acrobot equations. This transformation is similar to that suggested in [15] to derive a cascade normal form for the acrobot. In fact, (9) and (10) are the same equations, while (11) and (7) are different. Note that it is possible to arrive at the cascade normal form using the quotient method by appropriately using the degree of freedom at each step. However, the quotient method is more general and does not exploit any specific property of under-actuated mechanical systems as opposed to the method suggested in [15].

## *B. Backward Sweep*

This section develops a controller for the acrobot based on the feedback form obtained in the preceding section. For this purpose, the following lemma will be used recursively.

*Lemma 1:* Consider the system

$$\begin{aligned} \dot{\mathbf{z}} &= f(\mathbf{z}, y) \\ \dot{y} &= p(y) \end{aligned} \tag{12}$$

and assume that $\dot{y} = p(y)$ has an asymptotically stable equilibrium point at $y = 0$. If $\dot{\mathbf{z}} = f(\mathbf{z}, 0)$ has an asymptotically stable equilibrium at $\mathbf{z} = 0$, then System (12) has an asymptotically stable equilibrium at $(z, y) = (0, 0)$.
This result is provided in Appendix B.2 of [2]. The following corollary of this lemma will help design the controller.

*Corollary 1:* Consider the system

$$\begin{aligned} \dot{\mathbf{x}} &= f_x(\mathbf{x}, \xi) && (13) \\ \dot{\xi} &= f_\xi(\mathbf{x}, \xi) + g(\mathbf{x}, \xi)u && (14) \end{aligned}$$

where $\mathbf{x} \in \mathbb{R}^{n-1}$, $\xi \in \mathbb{R}$, $u \in \mathbb{R}$, $f_x : \mathbb{R}^n \to \mathbb{R}^{n-1}$, $f_\xi : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}$ with the equilibrium point $(\mathbf{x}, \xi) = (0, 0)$, and $g(\mathbf{x}, \xi) \neq 0$ in the domain of operation. If there exists $\xi_d(\mathbf{x})$, with $\xi_d(0) = 0$, that asymptotically stabilizes System (13), then

$$u = \frac{\frac{\partial \xi_d}{\partial x} f_x(\mathbf{x}, \xi) - f_\xi(\mathbf{x}, \xi) + k(\xi_d(\mathbf{x}) - \xi)}{g(\mathbf{x}, \xi)}, \tag{15}$$

where $k$ is any arbitrary positive constant, locally asymptotically stabilizes both (13) and (14).

*Proof:* Let the error variable be

$$e = \xi_d(\mathbf{x}) - \xi. \tag{16}$$

Substituting (16) and (15) into (13) and (14) gives the new set of equations

$$\begin{aligned} \dot{\mathbf{x}} &= f_x(\mathbf{x}, \xi_d(\mathbf{x}) - e) && (17) \\ \dot{e} &= -ke && (18) \end{aligned}$$

Using Lemma 1, it can be shown that the above system is locally asymptotically stable since

- $e = 0$ is an asymptotically stable equilibrium of (18),
- for $e = 0$, by assumption, $\dot{\mathbf{x}} = f_x(\mathbf{x}, \xi_d(\mathbf{x}))$ is also asymptocically stable at $\mathbf{x} = 0$.

Hence, the system described by (17) and (18) is locally asymptotically stable at $(\mathbf{x}, e) = (0, 0)$, which implies that the system is locally asymptotically stable at $(\mathbf{x}, \xi) = (0, 0)$. ■

This proof suggests that locally stabilizing the $e$-dynamics achieves local stability of (13) and (14). This method is similar to cascade design [16], and global stability can be achieved under certain conditions as outlined in [16]. This approach is used recursively to obtain the controller for the full system.

For System (9), it is desired to have a function $y_{2d}(y_1)$ that, when subsituted for $y_2$, gives asymptotically stable $y_1$-dynamics. The presence of $y_3$ in the denominator of (9) is immaterial. The denominator is always positive since it represents the first component of the mass matrix $M(q)$. Hence, $y_{2d} \triangleq -k_1 y_1$, with $k_1$ any positive gain, stabilizes $y_1$. The error dynamics read:

$$\begin{aligned} e_1 &= y_2 - y_{2d} \\ &= y_2 + k_1 y_1. \end{aligned}$$

Differentiating the error dynamics with respect to time gives

$$\dot{e}_1 = \dot{y}_2 + k_1 \dot{y}_1,$$

To obtain the function $y_{3d}(y_1, y_2)$ that makes $e_1 \to 0$ as $t \to \infty$, the dynamics given by (18) are used. In particular, $\dot{e}_1 = -k_2 e_1$, and substituting from (9) and (10) gives:

$$\begin{aligned} -k_2 e_1 = \; & \frac{y_2}{c_1 + c_2 + 2c_3 \cos y_{3d}} \\ & + k_1[g(c_4 \sin(y_1 + \zeta(y_{3d})) \\ & + c_5 \sin(y_1 + \zeta(y_{3d}) + y_{3d}))], \end{aligned}$$

where $k_2$ is any positive gain. Since it is not possible to obtain a closed-form solution for $y_{3d}$, the equation is linearized around the equilibrium point, which gives the following solution for $y_{3d}$:

$$\begin{aligned} y_{3d} = \; & \frac{(c_1 + c_2 + 2c_3)(c_4 g + c_5 g + k_1 k_2)}{(c_2 c_4 + c_3(c_4 - c_5) - c_1 c_5)g} y_1 \\ + \; & \frac{k_1 + (c_1 + c_2 + 2c_3)k_2}{(c_2 c_4 + c_3(c_4 - c_5) - c_1 c_5)g} y_2 \end{aligned}$$

TABLE I

PARAMETERS USED FOR SIMULATION

| $l_1$ | $l_2$ | $m_1$ | $m_2$ | $I_1$ | $I_2$ |
|---|---|---|---|---|---|
| $1m$ | $2m$ | $1kg$ | $2kg$ | $0.083kgm^2$ | $0.667kgm^2$ |

Designing $y_{3d}$ through linearization considerably reduces the domain of attraction. Nevertheless, only a locally stabilizing controller is required, which is obtained through this linearization.

Next, consider the system described by (9), (10) and (11). A stabilizing $y_{3d}$ is known for (9) and (10), which allows constructing $e_2 = y_3 - y_{3d}$ and by assigning $\dot{e}_2 = -k_3 e_2$ (using (15) ), gives $z_{4d}$ that will stabilize the system.

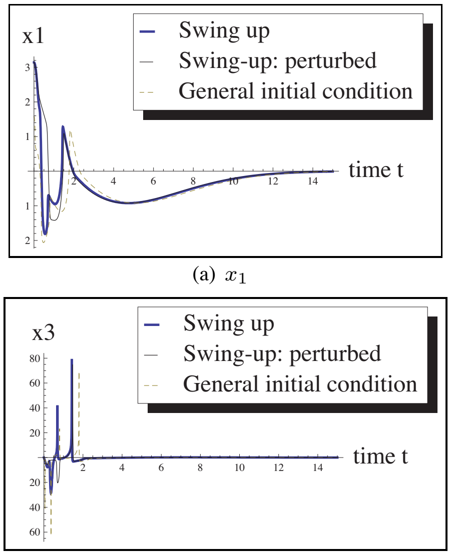
Similarly, when the full system described by (9), (10), (11) and (7) is considered, the knowledge of $z_{4d}$ allows constructing $e_3 = z_4 - z_{4d}$ and assigning $\dot{e}_3 = -k_4 e_3$ gives $u$ that locally asymptotically stabilizes the system around $(0, 0, 0, 0)$. The stable open-loop equilibrium point $(\pi, 0, 0, 0)$ is no longer an equilibrium point of the closed-loop system. Due to the presence of $\tan(x_2/2)$ in $\zeta(x_2)$, the other two equilibrium points become singularity points, where $\zeta(x_2)$ is no longer defined. This represents an obstacle for obtaining a large domain of attraction. However, this does not prevent implementing the swing-up, as shown through simulation in the next section.

## IV. NUMERICAL SIMULATION

This section illustrates the controller design of the previous section. The parameters used in this simulation are given in Table I [9].

To determine the controller gains $k_1, k_2, k_3, k_4$, the closed-loop system is linearized around $(0, 0, 0, 0)$ and the gains determined so that all four eigenvalues are at $-1$, which results in $k_1 = 10, k_2 = 1, k_3 = 1, k_4 = 1$. Three sets of simulations are presented. The first one corresponds to the swing-up of the acrobot, i.e. with initial conditions corresponding to the lower equilibrium point $(\pi, 0, 0, 0)$. The second simulation illustrates the swing-up from the perturbed lower equilibrium point $(\pi, 0.1, -0.2, -0.3)$. The last simulation shows stabilization from the arbitrary initial condition $(\pi/2, \pi/2, 1, 1)$. The state trajectories are represented in Fig. 2 and Fig. 3, from which the following observations can be made:

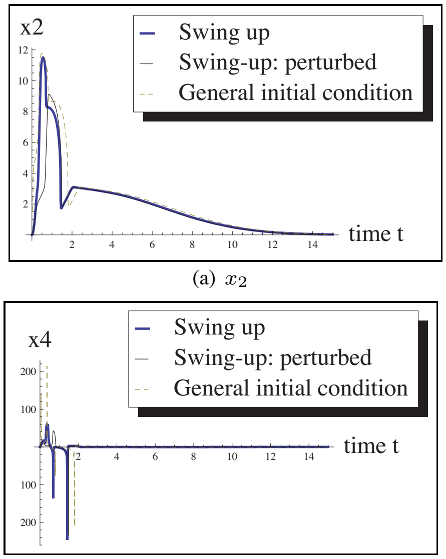
1) The trajectories first come together before converging to the origin. Indeed, since the controller is designed to stabilize the error dynamics, as soon as the errors come close to zero, the behavior is the same irrespective of the initial conditions.
2) The angular velocities $x_3$ and $x_4$ reach very large values in all three cases, which calls for smaller controller gains.
3) Since sharp changes in angular velocities require large torques, the controller gains require tuning before practical implementation.



(a) $x_1$

(b) $x_3 = \dot{x}_1$

Fig. 2. Position and velocity of the first link for different initial conditions



(a) $x_2$

(b) $x_4 = \dot{x}_2$

Fig. 3. Position and velocity of the second link for different initial conditions

## V. CONCLUSIONS AND FUTURE WORK

A nonlinear controller design capable of swinging-up and balancing the acrobot at its upright position has been presented. Although the controller is local, the domain of attraction has been enlarged to achieve swing-up from its lower equilibrium point. Hence, a single controller achieves both swing-up and balancing as opposed to most previous attempts, where two different control strategies are required (one for swing-up and another for balancing). Locally stabilizing (balancing) behaviour of the controller is validated using Lyapunov's first method and is guaranteed for all set of positive controller gains. The domain of attraction can be enlarged by appropriate choice of these gains. Although the swing-up capability has been validated in simulation for a particular set of gains, finding an analytical proof requires further work.

The method is straightforward to implement for feedback linearizable systems. For non-feedback-linearizable systems, however, direct implementation is not possible since the design requires the distribution $\Delta = \text{span}\{g, [f, g]\}$ to be involutive at every step of the reduction process. Hence, each non-feedback-linearizable system needs to be investigated individually. This was undertaken here for the acrobot, for which a local linerization of the two-dimensional quotient dynamics led to a nonlinear stabilizing controller. The pendubot and the inverted pendulum are more difficult to deal with since the distribution $\Delta = \text{span}\{g, [f, g]\}$ is not involutive at the first step already. On the other hand, the systematic nature of the method makes it likely to be turned into a numerical technique which can be used for a larger class of systems. Finally, the problem of input saturation has not been addressed in this paper. Further work could consider adapting the quotient method to handle the input saturation problem.

## VI. ACKNOWLEDGMENTS

Supported by the Swiss National Science foundation grant FN 200021-11753/1

## REFERENCES

[1] H. K. Khalil. *Nonlinear Systems*. Prentice-Hall, Englewood Cliffs, N.J., 3rd edition, 2002.

[2] A. Isidori. *Nonlinear Control Systems*. Springer Verlag, Berlin, Heidelberg, New York, second edition, 1989.

[3] H. Nijmeijer and A. van der Schaft. *Nonlinear Dynamical Control Systems*. Springer Verlag, New York, 1990.

[4] M. Krstic, P. V. Kokotovic, and I. Kanellakopoulos. *Nonlinear and Adaptive Control Design*. John Wiley & Sons, Inc., New York, NY, USA, 1995.

[5] R. M. Murray and J. Hauser. Nonlinear controllers for non-integrable systems: the acrobot example. *Proceedings of American Control Conference*, pages 669–671, 1990.

[6] S. A Bortoff and M. W. Spong. Pseudolinearization of the acrobot using spline functions. *Proceedings of IEEE Conference on Decision and Control*, 1(8):593–598, 1992.

[7] S. A Bortoff. Approximate state-feedback linearization using spline. *Automatica*, 33(8):1449–1458, 1997.

[8] M. W. Spong. The swing up control problem for the acrobot. *Control System Magazine, IEEE*, 15(1):49–55, 1995.

[9] A Mahindrakar and R. Banavar. The swing up of acrobot based on a simple pendulum strategy. *International Journal of Control*, 78(6):424–429, 2005.

[10] A Mahindrakar, A. Astolfi, R. Ortega, and G. Viola. Further constructive results on interconnection and damping assignment control of mechanical systems: The acrobot example. *American Control Conference*, pages 5584–5589, 2006.

[11] B. Srinivasan, P. Huguenin, and D. Bonvin. Global stabilization of an inverted pendulum - Control strategy and experimental validation. *Automatica*, 45:265–269, 2009.

[12] Ph. Mullhaupt. Quotient submanifolds for static feedback linearization. *Systems & Control Letters*, 55:549–557, 2006.

[13] S. S. Willson, Ph. Mullhaupt, and D. Bonvin. Controller design using successive invariant manifold decomposition. *Under preparation*.

[14] A. R. Forsyth. *A Treatise on Differential Equations*. MacMillan & Company, Ltd., London and New York, sixth edition, 1929.

[15] R. Olfati-Saber. Cascade normal forms for underactuated mechanical systems. *Proceedings of Conference on Decision and Control*, pages 2162–2167, 2000.

[16] R. Sepulchre, M. Jankovic, and P. Kokotovic. *Constructive Nonlinear Control*. Springer Verlag, Berlin, Heidelberg, New York, first edition, 1997.