

Hybrid modelling and control of obstacle-aided snake robot locomotion

Pål Liljebäck, *Member, IEEE*, Kristin Y. Pettersen, *Senior Member, IEEE*, Øyvind Stavdahl, *Member, IEEE*, and Jan Tommy Gravdahl, *Senior Member, IEEE*

Abstract—A snake can traverse cluttered and irregular environments by using irregularities around its body as push-points to aid the propulsion. This characteristic feature of biological snake locomotion, denoted *obstacle-aided locomotion*, is investigated for snake robot locomotion purposes in this paper. The paper presents a *hybrid model* of the dynamics of a planar snake robot interacting with obstacles in its environment. Obstacle contact forces are calculated by formulating and solving a *linear complementarity problem (LCP)*. The *existence and uniqueness* properties of the state evolution of the hybrid model are investigated. The paper also presents a *hybrid control strategy* employing measured contact forces to maintain propulsion while simultaneously preventing the snake robot from being jammed between obstacles in its path. The simulation results validate the hybrid modelling approach and the effectiveness of the proposed control strategy.

Index Terms—Hybrid model, Linear complementarity problem, Underactuated Robots, Force and Tactile Sensing, Contact Modelling.

I. INTRODUCTION

INSPIRED by biological snake locomotion, snake robots carry the potential of meeting the growing need for robotic mobility in unknown and challenging environments. These mechanisms typically consist of serially connected joint modules capable of bending in one or more planes. The many degrees of freedom of snake robots make them difficult to control, but provide traversability in irregular environments that surpasses the mobility of the more conventional wheeled, tracked and legged forms of robotic mobility.

A unique feature of snake robot locomotion compared to other forms of robotic mobility is that irregularities on the ground are actually beneficial for the propulsion since they provide push-points for the robot. While *obstacle avoidance* is important for wheeled, tracked and legged robots, the goal of snake locomotion is rather *obstacle exploitation*. The term *obstacle-aided locomotion* was introduced by Transeth *et al.* [1] and captures the essence of this concept.

One of the earliest analytical studies of snake locomotion was given by Gray [2], who investigated the contact forces between a three-linked snake robot and a set of obstacles. Gray concluded that forward motion of a planar snake requires the existence of external forces acting in the normal direction of the snake body. Hirose [3] studied biological snakes and modelled the snake body as a continuous curve that could not

move sideways (no-slip conditions). A well-known result by Hirose is the formulation of the *serpenoid curve*, which is a mathematical description of *lateral undulation* (this is the most common form of snake locomotion). Hirose also proposed a strategy for *lateral inhibition* that modifies the shape of a snake robot based on contact force sensing along the snake body in order to avoid obstacles. Another interesting study of snake locomotion is given in [4], which studies the motion of biological snakes as they interact with pegs in order to push themselves forward.

Several mathematical models of snake robot dynamics have been proposed. Some models [5]–[9] assume that the body of the snake robot cannot move sideways (no-slip conditions) achieved by e.g. mounting passive wheels along the snake body). Such conditions are advantageous for realizing lateral undulation on a flat surface. Other models [10]–[14] allow sideways motion of the snake body, but assume that the ground friction normal to the body is higher than the friction tangential to the body. Such conditions are also advantageous for realizing lateral undulation on a flat surface since this gives the snake the tendency to glide forward rather than slip sideways.

The works in [1], [15], [16] present, to the authors' best knowledge, the only known models of snake robot dynamics that *also* include obstacle contact forces. In [15], a dynamic simulation software called *WorkingModel* is used to simulate a planar snake robot interacting with circular obstacles. Contact forces are calculated from a spring-damper approximation. A similar approach is employed in [16], where the simulation software *Open Dynamics Engine* (ODE) is used to model a snake robot interacting with various forms of obstacles. Neither [15] nor [16] present the equations of motion of the snake robot in an analytical form due to the use of general-purpose simulation software. The work in [1] employs the framework of nonsmooth dynamics to formulate a nonsmooth (hybrid) model of a snake robot interacting with obstacles. A *timestepping* method is used to simulate the hybrid dynamics of the robot. Timestepping methods discretize the system equations with a time step determined by a fixed error criterion and approximate trajectories of the hybrid system without tracking events (i.e. obstacle impacts). The resulting form of the model is therefore suitable for simulation purposes, but not for analysis and control design.

The majority of previous research on *control* of snake robots has focused on flat surface motion with preprogrammed motion patterns aimed at resembling gaits displayed by biological snakes. The works in [3], [15], [17] present, to the authors' best knowledge, the only known control strategies related to obstacle-aided snake locomotion (i.e. where the surface is no longer assumed to be flat). Along with these works, we should also mention the work in [18], which analyses how obstacles

Manuscript received January 26, 2010.

Affiliation of Pål Liljebäck is shared between the Dept. of Engineering Cybernetics at the Norwegian University of Science and Technology (NTNU), NO-7491 Trondheim, Norway, and SINTEF ICT, Dept. of Applied Cybernetics, N-7465 Trondheim, Norway. E-mail: Pal.Liljeback@sintef.no.

K. Y. Pettersen, Øyvind Stavdahl, and Jan Tommy Gravdahl are with the Dept. of Engineering Cybernetics at the Norwegian University of Science and Technology (NTNU), NO-7491 Trondheim, Norway. E-mail: {Kristin.Y.Pettersen, Oyvind.Stavdahl, Tommy.Gravdahl}@itk.ntnu.no.

around a snake robot affect its degrees of freedom. The work in [3] was described above. The work in [15] proposes an inverse dynamics approach by formulating and numerically solving an optimization problem in order to, for a given set of obstacle contacts, calculate the contact forces required to propel the snake in a desired direction. A strategy for calculating the actual torque inputs to the joints from the desired contacts was, however, not presented. A kinematic approach is proposed in [17], where a curve fitting procedure is used to determine the shape of the snake with respect to the obstacles. Subsequently, this shape is propagated backwards along the snake body under the assumption that this will push the robot forward.

A long-term goal of the work presented in this paper is to synthesize model-based controllers for obstacle-aided locomotion with provable stability properties. To facilitate such developments, the underlying mathematical model should have a simple and analytical form. As a step in this direction, this paper provides two main contributions. The first contribution is the development of a *hybrid model* of the dynamics of a planar snake robot interacting with obstacles in its environment. In contrast to the hybrid model presented in [1], which is based on a *timestepping* method that approximates trajectories of the hybrid system without tracking events, the hybrid model presented in this paper is based on *event-tracking* [19], where discrete events are tracked. The authors believe the model formulation with this approach is better suited for analysis and synthesis of model-based controllers. We model obstacle interaction by introducing a *unilateral velocity constraint* on each contacted link of the snake robot. This is a novel approach since conventional models of mechanical systems with unilateral constraints calculate constraint forces with respect to the normal direction of the *obstacles* [20]. With the approach described in this paper, the shape of the obstacles does not have to be considered explicitly as we instead calculate constraint forces with respect to the normal direction of the *contacted links*. This simplifies the equations of motion. As an extension to our preliminary work in [21], we show how the equations governing the obstacle contact forces on the snake robot can be formulated as a *linear complementarity problem* (LCP). This enables us to apply existing general results concerning existence and uniqueness of solutions to LCPs [22] to the model of the snake robot.

The second contribution of this paper is a *hybrid controller* for obstacle-aided locomotion aimed at resolving situations where the snake robot is jammed between obstacles. The concept of detecting and resolving snake robot jams has, to the authors' best knowledge, not been treated in previous literature, but is a genuine challenge during snake robot locomotion in cluttered environments. To our knowledge, this is the first published control strategy for a snake robot involving feedback and explicit use of measured contact forces to achieve propulsion. Note that the work in [3] also considers snake locomotion based on measured contact forces. However, the contact forces in [3] are employed to *avoid* obstacles, whereas the contact forces in this paper are employed to push the snake robot forward. The hybrid controller proposed in this paper is an extension of preliminary work by the authors in [21], where the hybrid nature of the controller was not explicitly considered. The paper presents simulation results that validate the hybrid modelling approach and the effectiveness of the proposed hybrid control strategy.

The paper is organized as follows. Section II presents

essential background material regarding hybrid modelling and control. Section III gives an overview of the main results of the paper. Section IV presents a 2D model of a snake robot without obstacle contact forces. Section V extends the 2D model in order to include obstacle contact forces. Section VI formulates the 2D model with obstacles as a hybrid dynamical system. Section VII presents the hybrid control strategy for the snake robot. Section VIII presents simulation results, and Section IX presents concluding remarks.

II. BACKGROUND MATERIAL

This section presents the framework employed in order to formulate the *hybrid model* of the snake robot in Section VI and the *hybrid controller* in Section VII. This section also presents the *linear complementarity problem*, which is central in the modelling of obstacle contact forces in Section V.

A. Modelling of hybrid dynamical systems

A *hybrid dynamical system* is a dynamical system that exhibits both continuous and discontinuous state evolution. A snake robot interacting with obstacles is a hybrid system since the impacts between the snake and the obstacles represent discrete events.

Several modelling frameworks for hybrid systems exist [19]. In this work, we have chosen to employ the modelling framework described in [23] since this framework captures a wide variety of hybrid phenomena, and it also facilitates stability analysis of hybrid systems.

In accordance with [23], a hybrid system has a state vector $x \in \mathbb{R}^n$ that can both *flow* (evolve continuously) and *jump* (evolve discontinuously). The data that determine the evolution of x is given by the four elements (C, F, D, G) , where C denotes the *flow set*, F denotes the *flow map*, D denotes the *jump set*, and G denotes the *jump map* of the hybrid system. Whenever the state x belongs to the *flow set*, it flows (or evolves continuously) according to F . During flows, the system acts as an ordinary continuous dynamical system. However, when x belongs to the *jump set*, it generally jumps according to G to a new value x^+ (superscript $+$ and $-$ denote 'the next value' and 'the previous value', respectively). Hence, the general form of a hybrid dynamical system is given by

$$\begin{aligned} \dot{x} &= F(x, u) & \text{for all } x \in C \\ x^+ &= G(x) & \text{for all } x \in D \end{aligned} \quad (1)$$

where we have also included a control input, $u \in \mathbb{R}^r$.

Existence and *uniqueness* of solutions is a very important issue when modelling hybrid systems. From a given initial state, x_0 , a hybrid system may have a single solution, several solutions, or no solution at all. For a general hybrid system, there are no easily verifiable necessary and sufficient conditions for existence and uniqueness of solutions. However, such conditions exist for special classes of hybrid systems, such as for *complementarity systems* (see Section II-C).

A hybrid system is simulated by letting the state vector, x , flow according to the flow map, $F(x, u)$, as long as $x \in C$. Whenever $x \in D$, a jump in the state vector, x^+ , is calculated according to the jump map, $G(x)$, and the simulation of the flow map is *restarted* from the new initial value, x^+ . This approach for simulating hybrid systems is called *event-tracking* [19] since the discrete events of the model are tracked.

B. Hybrid controllers for hybrid systems

Hereafter, we denote the system being controlled as the *plant*. If the controller that generates the control input, u , for the plant in (1) consists of an algorithm with discrete-valued states, then we denote this a *hybrid controller* [23]. A hybrid controller is a hybrid system with state $\eta \in \mathbb{R}^p$ (which can contain e.g. logic states, timers, and counters) that evolves as a function of both the controller state, η , and the plant state, x . The control input is generally calculated according to a function $u = \kappa(x, \eta)$. Sometimes hybrid controllers are used to control plants that are continuous-time systems (see e.g. examples in [23]). In this paper, however, we will consider a *hybrid controller* for a *hybrid plant*.

C. Complementarity systems

A hybrid system is called a *complementarity system* if the *flow* of the system states is constrained by a set of *complementarity conditions* [19]. A complementarity condition between two scalar variables requires that both variables are nonnegative and that their product is always zero, i.e. one variable is always zero. In mathematical terms, the complementarity condition between two scalar variables x and y can be written $x \geq 0 \wedge y \geq 0 \wedge xy = 0$. Two vectors $x \in \mathbb{R}^m$ and $y \in \mathbb{R}^m$ are said to be *complementary* if, for all i , the pair of variables (x_i, y_i) is subject to a complementarity condition. We will see in Section V that we can formulate complementarity conditions for the links of the snake robot that are in contact with an obstacle.

The constraint equations of a complementarity system can often be formulated as a *linear complementarity problem* (LCP). A LCP asks whether there exist two complementary vectors $x \in \mathbb{R}^m$ and $y \in \mathbb{R}^m$ such that

$$\begin{aligned} y &= a + Ax \\ x &\geq 0, y \geq 0, x^T y = 0 \end{aligned} \quad (2)$$

for a given vector $a \in \mathbb{R}^m$ and a matrix $A \in \mathbb{R}^{m \times m}$. The constraint equations of the snake robot are given in this form in Section V. The following result is proved in [22]:

Theorem 1: The LCP in (2) is uniquely solvable for all data vectors a if and only if A is a P -matrix.

A P -matrix is a matrix whose *principal minors* are all positive. A principal minor of the matrix A is the determinant of a square submatrix of A consisting of the same set of rows and columns. A real *symmetrical* matrix is a P -matrix if and only if it is *positive definite*. For a real symmetrical matrix, one can therefore apply the standard criteria for positive definiteness in order to check if the matrix is a P -matrix. If the matrix is not symmetrical, one can e.g. apply the recursive algorithm in [24], which is $O(2^m)$, in order to check if the matrix is a P -matrix.

Several algorithms exist for solving the LCP in (2). A famous approach is the so-called *Lemke's algorithm* [22], which basically uses trial and error to find the non-zero elements of x and y , but with clever rules for changing the non-zero elements between trials.

III. SUMMARY OF MAIN RESULTS

In order to increase the readability of this paper, this section provides an overview of the main results.

- Using the notation and symbols presented in Section IV-A and Section IV-B, and the Coulomb ground friction

model presented in Section IV-C, the dynamics of a snake robot moving on a horizontal surface *without* obstacles is described by (28).

- Under Assumption 3 - Assumption 10 presented in Section V-A, the motion of the snake robot is constrained according to the velocity constraints in (38) when one or several snake robot links are in contact with obstacles.
- At time instants where the set of links in contact with obstacles is *unchanged*, the resulting constrained dynamics of the snake robot is described by (47) when the obstacles are frictionless, and by (52) when obstacle friction is present. Calculation of the constrained dynamics *without* obstacle friction requires us to solve the LCP in (48). By Proposition 16, this LCP always possesses a unique solution. Calculation of the constrained dynamics *with* obstacle friction requires us to solve the LCP in (53). By Proposition 18, there always exists a non-zero scalar $\mu_o^* > 0$ such that this LCP possesses a unique solution when the obstacle friction coefficient, μ_o , satisfies $\mu_o < \mu_o^*$. We are unable to derive an analytical expression for μ_o^* .
- At time instants where the set of links in contact with obstacles *changes*, the dynamics of the snake robot is described by (55), which requires us to solve the LCP in (58). By Proposition 21, this LCP always possesses a unique solution.
- The complete constrained dynamics of the snake robot is captured by the *hybrid model* formulated in (70).
- A *hybrid controller* for obstacle-aided locomotion is proposed in (92).
- The complete *closed-loop system* (hybrid plant and hybrid controller) is formulated in Section VII-G.

IV. MODEL OF A SNAKE ROBOT WITHOUT OBSTACLES

This section presents a *continuous* 2D model of the *dynamics* of a snake robot moving on a flat surface without obstacles. The model is based on [10], but whereas nonuniform ground friction was assumed in [10], we here employ a uniform ground friction model. The model is furthermore extended to include obstacle contact forces in Section V and subsequently fitted within the framework of a hybrid dynamical system in Section VI.

A. Notation and defined symbols

The snake robot consists of n links of length $2l$ interconnected by $n - 1$ joints. All n links have the same mass m and moment of inertia $J = \frac{1}{3}ml^2$. The total mass of the robot is thus nm . The mass of each link is uniformly distributed so that the link CM (center of mass) is located at its center point (at length l from the joint at each side). The mathematical symbols defined in order to represent the kinematics and dynamics of the snake robot are described in Table I and illustrated in Fig. 1 and Fig. 2.

Vectors are either expressed in the global coordinate system or in the local coordinate system of link i . This is indicated by superscript *global* or L_i , respectively. If otherwise is not specified, a vector with no superscript is expressed in the global coordinate system.

The following vectors and matrices are used in the subsequent sections:

C. Coulomb friction model

A Coulomb friction model is employed to model the ground friction forces. Each link is subjected to a ground friction force acting on the CM of the link and also a friction torque acting about the link CM. We define the friction force on link i as

$$f_{R,i} = \begin{cases} -\mu m g \frac{v_i}{|v_i|}, & |v_i| > 0 \\ 0_{2 \times 1} & |v_i| = 0 \end{cases} \quad (10)$$

where $\mu \in [0, 1]$ is the Coulomb friction coefficient, g is the gravitational acceleration constant, and $v_i = (\dot{x}_i, \dot{y}_i)$ is the velocity of link i . We now define the scalar value $\hat{v}_i \in \mathbb{R}$ given by

$$\hat{v}_i = \begin{cases} \frac{1}{|v_i|}, & |v_i| > 0 \\ 0 & |v_i| = 0 \end{cases} \quad (11)$$

and also the diagonal matrix $\Gamma = \text{diag}(\hat{v}_1, \hat{v}_2, \dots, \hat{v}_n) \in \mathbb{R}^{n \times n}$. This enables us to express the friction force on all links in matrix form as

$$f_R = \begin{bmatrix} f_{R,x} \\ f_{R,y} \end{bmatrix} = -\mu m g \begin{bmatrix} \Gamma \dot{x} \\ \Gamma \dot{y} \end{bmatrix} \in \mathbb{R}^{2n} \quad (12)$$

The friction torque about the CM of link i is a result of friction forces acting normal to the link during link rotation. The direction of the velocity of a point along the link with respect to the ground is actually also dependent on the translational velocity of the link. However, in order to simplify the friction model, the friction torque is modelled based on the link rotation only. The friction force df_i on an infinitesimal mass element dm of link i due to the link rotation $\dot{\theta}_i$ produces a friction torque $d\tau_{R,i}$ about the CM of the link given by

$$d\tau_{R,i} = s df_i = s \left(-\mu g \cdot \text{sgn}(s\dot{\theta}_i) \cdot dm \right) \quad (13)$$

where s is the distance from the CM of link i to the mass element dm . Using the relation $dm = \frac{m}{2l} ds$, we may write the total friction torque on link i as

$$\tau_{R,i} = \int_{-l}^l d\tau_{R,i} = -\frac{1}{2} \mu m g l \cdot \text{sgn}(\dot{\theta}_i) \quad (14)$$

The global frame friction torque on all links may be expressed in matrix form as

$$\tau_R = -\frac{1}{2} \mu m g l \cdot \text{sgn}(\dot{\theta}) \in \mathbb{R}^n \quad (15)$$

D. Equations of motion

This section presents the equations of motion of the snake robot in terms of the acceleration of the link angles, $\ddot{\theta}$, and the acceleration of the CM of the snake robot, \ddot{p} . The configuration of the snake robot is given by $q = (\theta, p) \in \mathbb{R}^{n+2}$ and describes all $n+2$ DOFs of the snake robot.

The forces and torques acting on link i are visualized in Fig. 2. The force balance for link i in global frame coordinates is given by

$$\begin{aligned} m\ddot{x}_i &= f_{R,x,i} + h_{x,i} - h_{x,i-1} \\ m\ddot{y}_i &= f_{R,y,i} + h_{y,i} - h_{y,i-1} \end{aligned} \quad (16)$$

The force balance equations for all links may be expressed in matrix form as

$$m\ddot{x} = f_{R,x} + D^T h_x, \quad m\ddot{y} = f_{R,y} + D^T h_y \quad (17)$$

The link accelerations may also be expressed by differentiating (6) twice with respect to time. This gives

$$D\ddot{x} = lA \left(C_\theta \dot{\theta}^2 + S_\theta \ddot{\theta} \right), \quad D\ddot{y} = lA \left(S_\theta \dot{\theta}^2 - C_\theta \ddot{\theta} \right) \quad (18)$$

where the square operator of $\dot{\theta}^2$ means that each element of $\dot{\theta}$ is squared ($\dot{\theta}^2 = \text{diag}(\dot{\theta})\dot{\theta}$). We obtain the CM acceleration by differentiating (5) twice with respect to time, inserting (17), and noting that the joint constraint forces, h_x and h_y , are eliminated when the link accelerations are summed. This gives

$$\begin{bmatrix} \ddot{p}_x \\ \ddot{p}_y \end{bmatrix} = \frac{1}{n} \begin{bmatrix} e^T \ddot{x} \\ e^T \ddot{y} \end{bmatrix} = \frac{1}{nm} \begin{bmatrix} e^T & 0_{1 \times n} \\ 0_{1 \times n} & e^T \end{bmatrix} f_R \quad (19)$$

This equation simply states, as would be expected, that the acceleration of the CM of a snake robot equals the sum of the external forces acting on the robot divided by its mass.

The torque balance for link i is given by

$$J\ddot{\theta}_i = u_i - u_{i-1} + \tau_{R,i} - l \sin \theta_i (h_{x,i} + h_{x,i-1}) + l \cos \theta_i (h_{y,i} + h_{y,i-1}) \quad (20)$$

Hence, the torque balance equations for all links may be expressed in matrix form as

$$J\ddot{\theta} = D^T u + \tau_R - lS_\theta A^T h_x + lC_\theta A^T h_y \quad (21)$$

What now remains is to remove the joint constraint forces from (21). Premultiplying (17) by D , solving for h_x and h_y , and also inserting (18), gives

$$\begin{aligned} h_x &= (DD^T)^{-1} \left(m l A \left(C_\theta \dot{\theta}^2 + S_\theta \ddot{\theta} \right) - D f_{R,x} \right) \\ h_y &= (DD^T)^{-1} \left(m l A \left(S_\theta \dot{\theta}^2 - C_\theta \ddot{\theta} \right) - D f_{R,y} \right) \end{aligned} \quad (22)$$

Inserting into (21) and solving for $\ddot{\theta}$ finally gives

$$M_\theta \ddot{\theta} = W \dot{\theta}^2 + lS_\theta N f_{R,x} - lC_\theta N f_{R,y} + \tau_R + D^T u \quad (23)$$

where

$$M_\theta = JI_{n \times n} + ml^2 S_\theta V S_\theta + ml^2 C_\theta V C_\theta \quad (24)$$

$$W = ml^2 C_\theta V S_\theta - ml^2 S_\theta V C_\theta \quad (25)$$

$$N = A^T (DD^T)^{-1} D \quad (26)$$

$$V = A^T (DD^T)^{-1} A \quad (27)$$

The equations describing the unconstrained dynamics (i.e. dynamics without obstacles) are in other words given by (19) and (23). These equations may be combined into

$$M(q) \ddot{q} = f_u(q, \dot{q}, u) \quad (28)$$

where subscript 'u' denotes *unconstrained* and

$$\begin{aligned} q &= \begin{bmatrix} \theta \\ p \end{bmatrix} \in \mathbb{R}^{n+2} \\ M(q) &= \begin{bmatrix} M_\theta & 0_{n \times 1} & 0_{n \times 1} \\ 0_{1 \times n} & nm & 0 \\ 0_{1 \times n} & 0 & nm \end{bmatrix} \in \mathbb{R}^{(n+2) \times (n+2)} \\ f_u(q, \dot{q}, u) &= \begin{bmatrix} lS_\theta N & -lC_\theta N \\ e^T & 0_{1 \times n} \\ 0_{1 \times n} & e^T \end{bmatrix} f_R + \begin{bmatrix} W \dot{\theta}^2 + \tau_R + D^T u \\ 0 \\ 0 \end{bmatrix} \end{aligned} \quad (29)$$

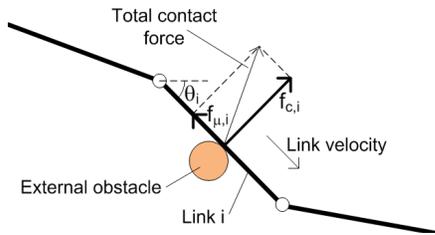


Fig. 3. The obstacle contact force on link i consisting of the normal direction constraint force, $f_{c,i}$, and the tangential direction friction force, $f_{\mu,i}$.

V. MODELLING OF OBSTACLE CONTACT FORCES

This section extends the model from Section IV in order to include contact forces from external obstacles in the environment around the snake robot. The contact model is based on several simplifying assumptions for control design purposes.

A. Overview of the contact modelling approach

The planar environment of the snake robot consists of an arbitrary number of external obstacles with *circular* shape. We consider circular obstacles to simplify the process of detecting overlap between a link and an obstacle, as described in Section V-B. The friction coefficient between the snake robot and any obstacle is μ_o . Furthermore, the shortest distance between the edges of any two obstacles is greater than the link length $2l$. This assumption prevents contact on both sides of a link.

The interaction between a snake robot link and an obstacle is modelled by introducing a *unilateral velocity constraint* for the link when it comes into contact with an obstacle. The constraint is *unilateral* (acts in one lateral direction only) since the constraint shall allow sideways motion of the link *away* from the obstacle, but prevent any sideways motion *towards* (and thereby into) the obstacle. Section V-C presents the equations describing the unilateral constraints for all contacted links.

As illustrated in Fig. 3, the obstacle contact force on link i consists of two orthogonal components. The first component is the *constraint force*, $f_{c,i} \in \mathbb{R}^2$, acting in the normal direction of link i and away from the obstacle (parallel to the local y axis of link i). The second component is the obstacle *friction force*, $f_{\mu,i} \in \mathbb{R}^2$, acting in the tangential direction of link i and in the opposite direction of the tangential link velocity (parallel to the local x axis of link i).

We assume that the contact force (i.e. the *constraint force* and the *friction force*) acts on the CM of a link only. Furthermore, we disregard any contact torque about the CM of the link. This simplifies the equations of motion considerably and does not have any significant influence on the overall motion of the snake when the length of the links is small.

For a set of links that are already in contact with obstacles, we will show in Section V-D how the problem of calculating the resulting obstacle contact forces can be formulated as a LCP. LCPs were introduced in Section II-C. By solving the LCP, we calculate the forces on the CM of the contacted links that are needed to satisfy the unilateral velocity constraints imposed on each contacted link. These constraint forces are then added to the equations of motion in (28) in order to cancel out the applied forces acting against the constraints. This represents the *continuous contact dynamics* of the snake robot.

When a link (that was previously not in contact with an obstacle) comes into contact with an obstacle, an *impact* occurs. We assume all impacts to be completely *inelastic*, meaning that the normal direction velocity of the link is completely absorbed during the impact. Furthermore, we assume all impacts to be *instantaneous*. During an impact, the contacted link is subjected to an *impulsive constraint force* in the normal direction of the link which instantaneously changes the normal direction link velocity in order to prevent it from continuing into the obstacle. This represents the *discrete impact dynamics* of the snake robot. The configuration, q , of the snake robot before and after the impact is assumed to be unaltered. We also assume that the obstacle friction forces cannot display impulsive behaviour, which means that we will disregard obstacle friction forces during an impact. Similar to the calculation of constraint forces, the problem of calculating the impulsive contact forces during an impact can be formulated as a LCP, as shown in Section V-E. It will be seen in Section VI that an impact triggers a *jump* in the state of the hybrid snake robot model.

The above description of the contact force model is now summarized with the following set of assumptions:

Assumption 3: All obstacles have a circular shape.

Assumption 4: The distance between the edges of any two obstacles is greater than the link length $2l$.

Assumption 5: The friction coefficient between the snake robot and any obstacle is μ_o .

Assumption 6: An obstacle contact force acts on the CM of a link only. The link length is small so that contact torques about the link CM are negligible.

Assumption 7: Impacts between the snake robot and the obstacles are completely inelastic.

Assumption 8: All impacts are instantaneous in time and all impact forces are impulsive.

Assumption 9: During an impact, the configuration of the snake robot, $q = (\theta, p)$, remains unaltered while the velocity, $\dot{q} = (\dot{\theta}, \dot{p})$, will generally experience a jump.

Assumption 10: Obstacle friction forces are negligible during an impact.

Remark 11: The common approach when modelling mechanical systems with unilateral constraints is to calculate the direction of a constraint force with respect to the normal direction of the *constraint surface* [20], i.e. the normal direction of the obstacles in this case. With the approach described in this paper, the shape of the obstacles does not have to be considered explicitly as we instead calculate the constraint forces with respect to the normal direction of the *contacted links*. This simplifies the equations of motion. Note that these two approaches produce similar constraint directions when the end point of a link is not in contact with an obstacle. To verify this, consider a snake robot link in contact with a circular obstacle. Since the link is tangent to the obstacle, the normal direction of the link and the obstacle must be equal. When the end point of a link, i.e. a joint, is in contact with an obstacle, however, both links attached to the joint are in contact. The approach in this paper will then produce a normal constraint force on both links attached to the joint.

Remark 12: A consequence of modelling obstacle contact by a unilateral force on the contacted link, is that there is nothing preventing the foremost link (the head) of the snake robot from penetrating an obstacle head-on along its tangential direction. A consequence of Assumption 6 is that a link in

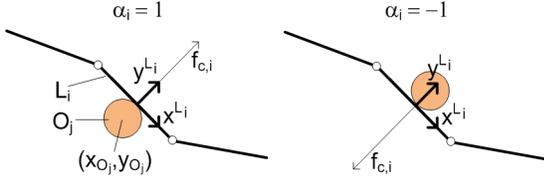


Fig. 4. The value of the contact parameter of link i is $\alpha_i = 1$ when the constraint force points along the positive link y axis and $\alpha_i = -1$ when the constraint force points along the negative link y axis. We set $\alpha_i = 0$ when there is no contact.

theory can rotate ‘into’ an obstacle while its CM has zero normal direction velocity. These two consequences are results of the goal of keeping the mathematical model as simple as possible, but are not critical in practice. Head-on collisions with the head of the snake robot can be avoided through the control strategy of the robot e.g. by assuming that the head is equipped with distance sensors. Link rotations ‘into’ an obstacle may only occur to a very small extent during obstacle-aided locomotion since this is mostly a forward gliding type of motion.

B. Collision detection

The environment around the snake robot consists of k circular obstacles indexed by $j \in \{1, \dots, k\}$. The coordinates of the center of obstacle j is denoted by (x_{O_j}, y_{O_j}) . The set O_j of points occupied by obstacle j is given by

$$O_j = \left\{ (x, y) \mid (x - x_{O_j})^2 + (y - y_{O_j})^2 \leq R_{O_j}^2 \right\} \quad (30)$$

where R_{O_j} is the radius of obstacle j . The set L_i of points occupied by link i (see Fig. 1) is given by

$$L_i = \{(x, y) \mid x = x_i + s \cos \theta_i, y = y_i + s \sin \theta_i, s \in [-l, l]\} \quad (31)$$

A collision between link i of the snake robot and obstacle j occurs whenever $L_i \cap O_j \neq \emptyset$, where \emptyset denotes an empty set. There is no collision if $L_i \cap O_j = \emptyset$.

We now introduce a vector of *contact parameters*, $\alpha \in \mathbb{R}^n$. The contact parameter of link i , denoted by $\alpha_i \in \{-1, 0, 1\}$, is a discrete state value that determines if the link is in contact with an obstacle and also on which side of the link there is contact. As shown in Fig. 4, $\alpha_i = 1$ when the obstacle constraint force points along the positive link y axis and $\alpha_i = -1$ when the obstacle constraint force points along the negative link y axis. We set $\alpha_i = 0$ when link i is not in contact with an obstacle. Whenever link i impacts an obstacle, the contact parameter, α_i , is updated according to $\alpha_i = \tilde{\alpha}(i)$, where

$$\tilde{\alpha}(i) = -\text{sgn} \left([0, 1] \left(R_{L_i}^{\text{global}} \right)^T \left(\min_{j \in \{1, \dots, k\}} r_{L_i, O_j} \right) \right) \quad (32)$$

and $r_{L_i, O_j} \in \mathbb{R}^2$ is the vector from link i to obstacle j . In (32), the global frame vector from link i to the *closest* obstacle is first found by use of the min operator, which is assumed to find the vector with the smallest *Euclidean norm*. Subsequently, this vector is transformed to the frame of link i using the rotation matrix $R_{L_i}^{\text{global}}$. Finally, the y component of this vector is extracted. The sign of the y component determines on which side of the link the obstacle is located.

C. Unilateral constraint equations

This section derives the unilateral velocity constraints for the links of the snake robot that are in contact with an obstacle. With reference to Fig. 4, the unilateral velocity constraint imposed on link i during contact with an obstacle may be compactly expressed as

$$\alpha_i v_{n,i} \geq 0 \quad (33)$$

where $v_{n,i}$ is the normal direction velocity of link i , i.e. the velocity of the CM of link i in the direction of the local link y axis. This constraint prevents sideways link motion towards (and thereby into) the obstacle. Using (4), the velocity constraint is given in the global frame as

$$\alpha_i (-\dot{x}_i \sin \theta_i + \dot{y}_i \cos \theta_i) \geq 0 \quad (34)$$

Hence, the unilateral velocity constraints for all links may be expressed in matrix form as

$$\text{diag}(\alpha) (-S_\theta \dot{x} + C_\theta \dot{y}) \geq 0 \quad (35)$$

where $\alpha = [\alpha_1 \ \dots \ \alpha_n]^T \in \mathbb{R}^n$. By inserting (9) into (35) and rearranging we get

$$\text{diag}(\alpha) C(q) \dot{q} \geq 0 \quad (36)$$

where $C(q) \in \mathbb{R}^{n \times (n+2)}$ is given by

$$C(q) = [-l(S_\theta N^T S_\theta + C_\theta N^T C_\theta) \quad -\sin \theta \quad \cos \theta] \quad (37)$$

We denote the number of contacted links by $m \in \{0, 1, \dots, n\}$. In order to easily select the velocity constraints from (36) that correspond to contacted links, we define a *selection matrix* $S_c(\alpha) \in \mathbb{R}^{m \times n}$, which simply contains the m rows from the matrix $\text{diag}(\alpha) \in \mathbb{R}^{n \times n}$ that contain a nonzero element. This enables us to write the velocity constraints for all links that are in contact with an obstacle as

$$\bar{C}(q, \alpha) \dot{q} \geq 0 \quad (38)$$

where $\bar{C}(q, \alpha) = S_c(\alpha) C(q) \in \mathbb{R}^{m \times (n+2)}$.

The calculation of obstacle contact forces in Section V-D requires the time derivative of the matrix $\bar{C}(q, \alpha)$, which is given by

$$\dot{\bar{C}}(q, \alpha) = S_c(\alpha) \left[l(S_\theta \tilde{N} C_\theta - C_\theta \tilde{N} S_\theta) \quad -C_\theta \dot{\theta} \quad -S_\theta \dot{\theta} \right] \quad (39)$$

where $\tilde{N} = \text{diag}(\dot{\theta}) N^T - N^T \text{diag}(\dot{\theta})$. Note that this derivative is only valid over intervals where the set of contacted links remains constant. The derivative of $\bar{C}(q, \alpha)$ is not defined for time instants where an element of α is changed.

The following rank property of the constraint matrix \bar{C} is important in order to uniquely determine the contact forces acting on the snake robot (see Section V-D1).

Property 13: The matrix \bar{C} has full rank ($\text{rank}(\bar{C}) = m$) for all (q, α) .

Remark 14: Due to the complexity of the elements in \bar{C} , it is difficult to present a purely mathematical proof that Property 13 holds. However, we can argue from a physical perspective that this property must hold. Assume that $\text{rank}(\bar{C}) < m$. This implies linear dependence between some of the m rows of \bar{C} , i.e. there must exist a row of \bar{C} , denoted \bar{C}_i , such that

$$\bar{C}_i = \sum_{j \in \{1, \dots, m\} \setminus \{i\}} k_j \bar{C}_j \quad (40)$$

where $k_j \in \mathbb{R}$. The scalar $\bar{C}_i \dot{q}$ is the magnitude of the normal direction velocity of link i , denoted by $|v_{n,i}|$. Multiplying (40) by \dot{q} therefore gives

$$|v_{n,i}| = \bar{C}_i \dot{q} = \sum_{j \in \{1, \dots, m\} \setminus \{i\}} k_j |v_{n,j}| \quad (41)$$

which states that the normal direction velocity of link i can be written as a linear combination of the normal direction velocities of all other contacted links of the snake robot. From a physical perspective, such a dependence could never occur unless all links are parallel since the couplings between the link velocities are given in terms of both normal *and* tangential link velocities. In particular, (6) implies that the velocity of link i can be written in terms of the velocities of link $i-1$ and link $i+1$. Unless link $i-1$, i , and $i+1$ are parallel, this is a relationship involving both the normal *and* tangential velocities of link $i-1$ and link $i+1$. This contradicts (41) since the relationship in (41) only contains normal direction velocities. This leaves the case of parallel links ($\theta_1 = \theta_2 = \dots = \theta_n$) as the only way for (41) to be true. A straightforward calculation of \bar{C} in e.g. *Matlab Symbolic Toolbox* shows that \bar{C} always has full rank when the link angles are equal. We therefore conclude that \bar{C} never drops rank.

This subsection can be summarized as follows. At any given time instant the snake robot is in contact with m obstacles. The interaction between the robot and these m obstacles is modelled by imposing the unilateral velocity constraints in (38) on the m contacted links.

D. Continuous contact dynamics

We will now use the unilateral velocity constraints in (38) to derive the resulting equations of motion of the snake robot. We assume that the m contact points between the links and the obstacles have *already* been established, i.e. we consider the *continuous contact dynamics* of the snake robot over a time interval where the set of contacted links remains constant. The *discrete impact dynamics* occurring when a non-contacted link comes into contact with an obstacle is treated in Section V-E. We first consider the frictionless case in Section V-D1, followed by contact forces *with* friction in Section V-D2.

1) *Contact dynamics without obstacle friction*: Let us first assume that the m velocity constraints of the snake robot in (38) are *bilateral*, i.e. that they are given by

$$\bar{C} \dot{q} = 0 \quad (42)$$

These are called *Pfaffian* constraints and are modelled by adding a term to the equations of motion in (28) as follows [25]:

$$M \ddot{q} = f_u + \bar{C}^T \lambda \quad (43)$$

The term $\bar{C}^T \lambda$ ensures compliance with the imposed velocity constraints, and $\lambda \in \mathbb{R}^m$ is a vector of scalars known as *Lagrange multipliers* [25]. The Lagrange multipliers are important because multiplier λ_j equals the magnitude of the constraint force that ensures compliance with the j th constraint [25]. This means that if the j th constraint in (42) corresponds to the velocity constraint on link i , then λ_j equals the magnitude of the constraint force $f_{c,i}$ acting on link i .

We now argue that (43) also represent the equations of motion of the snake robot when the velocity constraints are *unilateral*, as in (38). This is quite obvious since the

influence of a *unilateral* constraint on the snake robot when this constraint is active (i.e. when the unilateral constraint is preventing sideways motion of a link) must necessarily be identical to the influence that the corresponding *bilateral* constraint would have. In other words, the nature of the constraint (i.e. unilateral or bilateral) is not apparent when the constraint is active since it is only active in one direction at a time. The only difference between the bilateral and unilateral case concerns the calculation of the constraint forces given by λ . In the bilateral case, one may calculate λ directly by differentiating (42) with respect to time, inserting (43), and solving for λ . In the unilateral case, however, the constraint forces must be calculated so that they comply with the directional requirements of the constraints, i.e. we require that $\lambda \geq 0$. This means that the constraint forces can only point *away* from the obstacles.

We will now handle this directional requirement by employing the theory of *linear complementarity problems* (LCPs) introduced in Section II-C. This approach is based on the work in [26]. An important observation is that the normal direction velocity of a contacted link and the corresponding constraint force are subjected to a *complementarity condition*. If the normal direction velocity is non-zero (i.e. the link is moving away from the obstacle), then the corresponding constraint force must be zero. Likewise, the normal direction velocity must be zero if the corresponding constraint force is non-zero. This complementarity condition also applies to the normal direction *acceleration* of a contacted link and the corresponding constraint force.

From the above discussion, the equations of motion of the snake robot that include *unilateral* constraint forces from *frictionless* obstacles are given by

$$M \ddot{q} = f_u + \bar{C}^T \lambda \quad (44)$$

$$\bar{C} \dot{q} \geq 0, \quad \lambda \geq 0, \quad \lambda^T \bar{C} \dot{q} = 0 \quad (45)$$

The vector $\bar{C} \dot{q} \in \mathbb{R}^m$ contains the normal direction velocity of each contacted link in the direction *away* from each obstacle. The normal direction *acceleration* of each contacted link in the direction *away* from each obstacle, denoted by $\bar{a}_n \in \mathbb{R}^m$, is given by

$$\bar{a}_n = \frac{d}{dt} (\bar{C} \dot{q}) = \bar{C} \ddot{q} + \dot{\bar{C}} \dot{q} \geq 0 \quad (46)$$

By solving (44) for \ddot{q} and inserting into (46), we finally arrive at the following model of the *continuous contact dynamics* of the snake robot with *frictionless* obstacles:

$$M \ddot{q} = f_u + \bar{C}^T \lambda \quad (47)$$

$$\begin{aligned} \bar{a}_n &= \bar{C} M^{-1} f_u + \dot{\bar{C}} \dot{q} + \bar{C} M^{-1} \bar{C}^T \lambda \\ \bar{a}_n &\geq 0, \quad \lambda \geq 0, \quad \lambda^T \bar{a}_n = 0 \end{aligned} \quad (48)$$

Equation (48) is in the form of the general LCP given in (2) with $A = \bar{C} M^{-1} \bar{C}^T$ and $a = \bar{C} M^{-1} f_u + \dot{\bar{C}} \dot{q}$. In order to calculate the dynamics of the snake robot at any given time instant, this LCP must be solved for the unknowns \bar{a}_n and λ subject to the complementarity conditions. The calculated λ gives the constraint forces from the obstacles and is plugged into (47) in order to calculate \ddot{q} .

In order to determine the existence and uniqueness properties of the LCP in (48), we will need the following result, which is proved in [27] (Proposition 8.1.2, item *xiii*):

Proposition 15: Let $M \in \mathbb{R}^{n \times n}$ be a *symmetrical* and *positive definite* matrix ($M > 0$), and $C \in \mathbb{R}^{m \times n}$ be a matrix of full rank ($\text{rank}(C) = m$). Then $CMC^T > 0$.

We may now state the following result concerning the existence and uniqueness properties of the LCP in (48):

Proposition 16: The LCP in (48) always possesses a unique solution (\bar{a}_n, λ) .

Proof: From Theorem 1, the proof is complete if we can show that $A = \bar{C}M^{-1}\bar{C}^T$ is a P -matrix. Since \bar{C} has full rank (by Property 13) and $M = M^T > 0$ (the inertia matrix is always symmetrical and positive definite), we have from Proposition 15 that $A = \bar{C}M^{-1}\bar{C}^T > 0$. Since A is symmetrical and positive definite, it must also be a P -matrix. This completes the proof. ■

Remark 17: The LCP in (48) can be regarded as the problem of, at a given time instant, determining which obstacle contacts that will persist into the next time instant, and which will not. A link contact will persist onto the next time instant if the corresponding value of λ is non-zero. If, however, the value of \bar{a}_n for a link contact is non-zero, then the link will detach from the obstacle.

2) *Contact dynamics with obstacle friction:* A Coulomb friction model is employed in order to describe the gliding friction force between the links and the obstacles. In accordance with Fig. 3, we define the obstacle friction forces on all the links as

$$f_\mu = \begin{bmatrix} f_{\mu,x} \\ f_{\mu,y} \end{bmatrix} = -\mu_o \begin{bmatrix} C_\theta \\ S_\theta \end{bmatrix} \text{diag}(\text{sgn}(v_t)) |f_c| \quad (49)$$

where $\mu_o \in [0, 1]$ is the Coulomb friction coefficient of the obstacles, $f_{\mu,x} \in \mathbb{R}^n$ and $f_{\mu,y} \in \mathbb{R}^n$ contain the obstacle friction forces on the links in the global x and y direction, respectively, $|f_c| \in \mathbb{R}^n$ contains the magnitude of the constraint force on each link, and $v_t \in \mathbb{R}^n$ contains the tangential link velocities in the local x direction of each link. The mapping between the friction forces and the link accelerations, \ddot{q} , is identical to the mapping between ground friction forces, f_R , and \ddot{q} given in (29) since f_R and f_μ both act on the CM of the links. By using the easily verifiable relation $|f_c| = S_c(\alpha)^T \lambda$, where $S_c(\alpha)$ is the selection matrix introduced in (38) and λ is the vector of *Lagrange multipliers* introduced in (43), we may write the link accelerations due to the obstacle friction forces, temporarily denoted \ddot{q}_{f_μ} , as

$$\ddot{q}_{f_\mu} = -\mu_o \Lambda \lambda \quad (50)$$

where $\Lambda \in \mathbb{R}^{(n+2) \times m}$ is given by

$$\Lambda(q, \dot{q}, \alpha) = \begin{bmatrix} lS_\theta N & -lC_\theta N \\ e^T & 0_{1 \times n} \\ 0_{1 \times n} & e^T \end{bmatrix} \begin{bmatrix} C_\theta \\ S_\theta \end{bmatrix} \text{diag}(\text{sgn}(v_t)) S_c(\alpha)^T \quad (51)$$

By adding (50) to the equations of motion in (47) and following the exact same approach that led to the LCP in (48), we get the following equations describing the *continuous contact dynamics* that include obstacle friction forces:

$$M\ddot{q} = f_u + (\bar{C}^T - \mu_o \Lambda) \lambda \quad (52)$$

$$\begin{aligned} \bar{a}_n &= \bar{C}M^{-1}f_u + \dot{\bar{C}}\dot{q} + \bar{C}M^{-1}(\bar{C}^T - \mu_o \Lambda) \lambda \\ \bar{a}_n &\geq 0, \quad \lambda \geq 0, \quad \lambda^T \bar{a}_n = 0 \end{aligned} \quad (53)$$

We again identify (53) as a LCP of the general form given in (2) with $A = \bar{C}M^{-1}(\bar{C}^T - \mu_o \Lambda)$ and $a = \bar{C}M^{-1}f_u + \dot{\bar{C}}\dot{q}$. When obstacle friction is present, we can no longer guarantee existence and uniqueness of the solution to the LCP in (53) since it is no longer evident that the matrix A is a P -matrix (see Section II-C). This existence and uniqueness issue is a general and well-known problem for acceleration LCPs that include Coulomb friction [20], [28]–[31]. Unfortunately, we are unable to provide an analytical upper bound of μ_o due to the complexity of determining if A is a P -matrix (A is not symmetrical, which complicates the P -matrix check). However, we can still state the following result:

Proposition 18: For a given (q, \dot{q}, α) , there exists a $\mu_o^* > 0$ such that the LCP in (53) possesses a unique solution (\bar{a}_n, λ) for $\mu_o \in [0, \mu_o^*]$.

Proof: Recall from Section II-C that $A = \bar{C}M^{-1}(\bar{C}^T - \mu_o \Lambda)$ is a P -matrix if all principal minors of A are positive. We know from Proposition 16 that the LCP in (53) *always* possesses a unique solution for $\mu_o = 0$ since A is a P -matrix in this case. All principal minors of A must therefore be positive for $\mu_o = 0$. Assume now that we increase μ_o until a principal minor of A becomes zero, and denote the corresponding value of the friction coefficient by $\mu_o^* > 0$. It is then evident that the P -matrix property of A must be preserved for $\mu_o < \mu_o^*$, i.e. existence and uniqueness of the solution to the LCP in (53) must hold for $\mu_o < \mu_o^*$. This completes the proof. ■

Remark 19: During our numerical treatments of the LCP in (53) so far, we have not yet encountered a single instance where A has failed to be a P -matrix. The authors therefore conjecture that μ_o must have an unrealistically high value in order for A to no longer be a P -matrix, and that the LCP in (53) will always be uniquely solvable during our simulations of the snake robot.

Remark 20: The matrix A being a P -matrix is only a *sufficient* condition for existence and uniqueness of the solution to a LCP. The LCP in (53) may therefore possess a unique solution for a given (q, \dot{q}, α) even if A is not a P -matrix [22].

E. Discrete impact dynamics

An *inelastic impact* occurs when a link comes into contact with an obstacle. By Assumption 8, the impact is instantaneous in time and the resulting impact forces are impulsive, resulting in a discrete *jump* in the velocity of the snake robot. We model the impact as [20]

$$M(q^+) \dot{q}^+ - M(q^-) \dot{q}^- = F_{\text{impulse}} \quad (54)$$

where $F_{\text{impulse}} \in \mathbb{R}^{n+2}$ denotes the generalized impulsive impact forces and q^- , \dot{q}^- , q^+ , and \dot{q}^+ denote the generalized coordinates and velocities immediately *before* and *after* the impact, respectively. This superscript notation is commonly used when modelling hybrid systems. By Assumption 9, the configuration of the snake robot is unaltered during an impact ($q^+ = q^-$). This means that $M(q^-) = M(q^+) = M(q)$. By Assumption 10, the impact forces are frictionless. Following the same argumentation that led to the expression in (47), we can now rewrite (54) as

$$M(q) (\dot{q}^+ - \dot{q}^-) = \bar{C}^T(q, \alpha^+) \lambda \quad (55)$$

where $\lambda \in \mathbb{R}^m$ is a vector of *impulsive constraint forces*. Note that the constraint matrix, \bar{C} , which is defined in (38),

depends on the contact parameter vector *after* the impact, i.e. α^+ . This is because the contact parameter of the impacted link is zero immediately *before* the impact (i.e. if link i impacts an obstacle, then $\alpha_i^- = 0$). In order to include this link in the impact dynamics, we must calculate \bar{C} based on the value of α_i *after* the impact, i.e. $\alpha_i^+ = \tilde{\alpha}(i)$, where $\tilde{\alpha}(i)$ was derived in (32). Also note that $\bar{C}(q, \alpha^-) \in \mathbb{R}^{(m-1) \times (n+2)}$, while $\bar{C}(q, \alpha^+) \in \mathbb{R}^{m \times (n+2)}$.

We will now calculate the impulsive constraint forces, λ , and the post-impact velocity, \dot{q}^+ , by following the approach presented in [19]. The post-impact velocity and the impulsive constraint forces are subjected to the same complementarity conditions as given in (45). We therefore have

$$\bar{C}(q, \alpha^+) \dot{q}^+ \geq 0, \quad \lambda \geq 0, \quad \lambda^T \bar{C}(q, \alpha^+) \dot{q}^+ = 0 \quad (56)$$

Solving (55) for \dot{q}^+ and premultiplying by $\bar{C}(q, \alpha^+)$ gives

$$\bar{C} \dot{q}^+ = \bar{C} \dot{q}^- + \bar{C} M^{-1} \bar{C}^T \lambda \quad (57)$$

Denoting the normal direction velocities of each of the contacted links (in the direction *away* from each obstacle) by the vector $\bar{v}_n \in \mathbb{R}^m$, we may combine (56) and (57) into the following LCP describing the impact dynamics of the snake robot:

$$\begin{aligned} \bar{v}_n^+ &= \bar{v}_n^- + \bar{C} M^{-1} \bar{C}^T \lambda \\ \bar{v}_n^+ &\geq 0, \quad \lambda \geq 0, \quad \lambda^T \bar{v}_n^+ = 0 \end{aligned} \quad (58)$$

The LCP in (58) is in the general form of the LCP given in (2) with $A = \bar{C} M^{-1} \bar{C}^T$ and $a = \bar{v}_n^-$, and must be solved for the unknowns \bar{v}_n^+ and λ . Subsequently, the post-impact velocity is found by solving (55) for \dot{q}^+ and inserting the calculated λ . The following result concerns the existence and uniqueness properties of the LCP in (58):

Proposition 21: The LCP in (58) always possesses a unique solution (\bar{v}_n^+, λ) .

Proof: The proof is identical to the proof of Proposition 16. ■

This subsection is now summarized. The *discrete impact dynamics* of the snake robot when link i impacts an obstacle and the state immediately before the impact is $(q^-, \dot{q}^-, \alpha^-)$, is given by

$$\begin{aligned} \alpha_j^+ &= \begin{cases} \tilde{\alpha}(i), & j = i \\ \alpha_j^-, & j \neq i \end{cases} \\ q^+ &= q^- \\ \dot{q}^+ &= \dot{q}^- + M^{-1}(q) \bar{C}^T(q, \alpha^+) \lambda \end{aligned} \quad (59)$$

where $j \in \{1, \dots, n\}$, $\tilde{\alpha}(i)$ is given by (32), and λ is calculated from the LCP in (58).

VI. HYBRID MODEL OF A SNAKE ROBOT WITH OBSTACLES

In this section, we employ the framework of *hybrid dynamical systems*, described in Section II-A, in order to encapsulate the *continuous* and the *discrete* dynamics of the snake robot into a single *hybrid* model, or a *hybrid plant*. We will denote the hybrid model as a *plant* to distinguish it from the hybrid *controller* presented in Section VII. The *jump set* (D_p), *jump map* (G_p), *flow set* (C_p), and *flow map* (F_p) of the plant are presented in the first four subsections, respectively, followed by a summary of the hybrid plant in the last subsection. We define the state vector of the hybrid plant as

$$x = (q, \dot{q}, \alpha) \in \mathbb{R}^{3n+4} \quad (60)$$

A. Jump set

A *jump* in the state vector x of the plant occurs when a link *impacts* an obstacle (jump in \dot{q} and α) or when a link *detaches* from an obstacle (jump in α). By employing the notation from Section V-B, the *jump set* corresponding to an *impact* between link i and an obstacle may be expressed as

$$D_{L_i}^{\text{Impact}} = \{x | L_i \cap O_j \neq \emptyset, j \in \{1, \dots, k\}, \tilde{\alpha}(i) C_i \dot{q} < 0\} \quad (61)$$

where $\tilde{\alpha}(i)$ is given by (32) and C_i denotes the i th row of the matrix $C(q)$ in (37). We use $\tilde{\alpha}(i)$ instead of α_i in (61) because $\alpha_i = 0$ before the impact has taken place. The *jump set* corresponding to link i *detaching* from an obstacle may be expressed as

$$D_{L_i}^{\text{Detach}} = \{x | L_i \cap O_j = \emptyset, j \in \{1, \dots, k\}, \alpha_i \neq 0\} \quad (62)$$

The jump sets comprising the *impacts* and the *detachments* of all the links, respectively, are given by

$$D^{\text{Impact}} = \bigcup_{i \in \{1, \dots, n\}} D_{L_i}^{\text{Impact}}, \quad D^{\text{Detach}} = \bigcup_{i \in \{1, \dots, n\}} D_{L_i}^{\text{Detach}} \quad (63)$$

The complete jump set of the hybrid plant may now be compactly expressed as

$$D_p = D^{\text{Impact}} \cup D^{\text{Detach}} \quad (64)$$

B. Jump map

The *jump map* corresponding to the *impact* between link i and an obstacle is presented in (59). The *jump map* corresponding to link i *detaching* from an obstacle involves simply setting α_i to zero. We may therefore express the complete jump map of the plant as

$$x^+ = G_p(x) = (q^+, \dot{q}^+, \alpha^+) \quad \text{for all } x \in D_p \quad (65)$$

where

$$\begin{aligned} q^+ &= q^- \\ \alpha_i^+ &= \begin{cases} \tilde{\alpha}(i) & \text{when } x \in D_{L_i}^{\text{Impact}} \\ 0 & \text{when } x \in D_{L_i}^{\text{Detach}} \\ \alpha_i^- & \text{when } x \notin (D_{L_i}^{\text{Impact}} \cup D_{L_i}^{\text{Detach}}) \end{cases} \\ \dot{q}^+ &= \begin{cases} \dot{q}^- + M^{-1}(q) \bar{C}^T(q, \alpha^+) \lambda & \text{when } x \in D^{\text{Impact}} \\ \dot{q}^- & \text{when } x \notin D^{\text{Impact}} \end{cases} \end{aligned} \quad (66)$$

The value of $\tilde{\alpha}(i)$ is given by (32) and λ is calculated from the LCP in (58).

C. Flow set

We define the *flow set* of the plant so that the state vector x always flows as long as the *jump set* is empty. The flow set is therefore simply given as

$$C_p = \{x | x \notin D_p\} \quad (67)$$

D. Flow map

The *flow map* of q is simply \dot{q} and the *flow map* of \dot{q} is given by (52). The contact vector, α , remains unchanged between jumps of x . The *flow map* of α is therefore the zero vector. The complete flow map of the plant is given by

$$\dot{x} = F_p(x, u) = (\dot{q}, \ddot{q}, 0_{n \times 1}) \quad \text{for all } x \in C_p \quad (68)$$

where

$$\ddot{q} = M^{-1} \left(f_u + \left(\bar{C}^T - \mu_o \Lambda \right) \lambda \right) \quad (69)$$

and λ is calculated from the LCP in (53). A control input, u , for the plant is proposed in Section VII.

E. Summary of the complete hybrid plant

In accordance with Section II-A, the complete model of the plant is given as

$$\begin{aligned} \dot{x} &= F_p(x, u) & \text{for all } x \in C_p \\ x^+ &= G_p(x) & \text{for all } x \in D_p \end{aligned} \quad (70)$$

The following result summarizes the existence and uniqueness properties of the plant.

Proposition 22: Given a control input, u , the evolution of the state x of the plant in (70) from any initial state can always be uniquely determined when the obstacles are frictionless ($\mu_o = 0$). With obstacle friction, there exists a $\mu_o^* > 0$ such that existence and uniqueness of the evolution of x is guaranteed for $\mu_o \in [0, \mu_o^*)$, but not guaranteed for $\mu_o \geq \mu_o^*$.

Proof: From (67), the *flow* and *jump set* are mutually exclusive, so we can always uniquely determine whether x should flow or jump. By Proposition 21, the *jump map* of x is always unique. By Proposition 16, the *flow map* of x is always unique with frictionless obstacles. By Proposition 18, the *flow map* of x is always unique for some $\mu_o = \mu_o^* > 0$. This completes the proof. ■

VII. HYBRID CONTROL OF OBSTACLE-AIDED LOCOMOTION

In this section, we propose a *hybrid controller* for obstacle-aided locomotion. The purpose of the controller is to propel the snake robot forward along the global x axis while detecting and resolving situations where the robot is *jammed* between obstacles.

A. Overview of the control strategy

A major challenge during obstacle-aided locomotion is to prevent the snake robot from being *jammed* between the obstacles. In a jammed situation, the *propulsive* components (i.e. the force components in the desired direction of motion) of the contact forces from the obstacles are too small to overcome the friction forces on the robot, and hence the forward motion stops. In this jammed situation, the obstacle contact forces will also prevent a number of the snake robot joints from moving to their reference angle.

The control strategy proposed in the following is a *hybrid controller* consisting of a *leader-follower* scheme and a *jam resolution* scheme, and also a supervisory mechanism for switching between these two schemes, denoted the *jam detection* scheme. The *leader-follower* scheme is carried out as long as the robot is able to move without being jammed between the obstacles. If the *jam detection* scheme detects

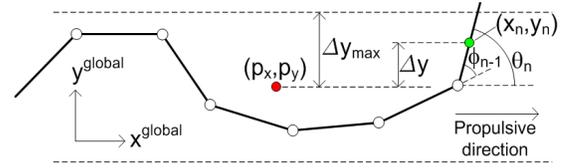


Fig. 5. Control of the head of the snake robot.

a jam, the *jam resolution* scheme is carried out in order to effectively ‘unlock’ the jammed joints.

The hybrid controller produces the torque input vector, $u \in \mathbb{R}^{n-1}$, for the $n-1$ joints of the robot. We assume that the *link angles*, $\theta \in \mathbb{R}^n$, the *link angle velocities*, $\dot{\theta} \in \mathbb{R}^n$, and the *constraint forces*, $f_c \in \mathbb{R}^n$, are measured.

We will first present the various schemes of the hybrid controller *without* considering the hybrid nature of the controller. In Section VII-F, the complete hybrid controller is summarized and formulated in terms of the hybrid modelling framework described in Section II-B.

B. The leader-follower scheme

The leader-follower scheme is motivated by the fact that each part of a biological snake conducting lateral undulation follows the path traced out by the head [2]. We therefore choose the head joint angle (the foremost joint), ϕ_{n-1} , as the reference angle for all subsequent joints. The preferred direction of motion for the snake is defined to be along the global positive x axis.

In order to achieve the sinusoidal motion characteristic of lateral undulation, we alternate between moving the head in the leftward and rightward direction with respect to the global positive x axis. This may be achieved by choosing the reference angle for the head link, $\theta_{n,\text{ref}}$, equal to a suitable positive constant θ_{left} when the head should move leftward and a negative constant θ_{right} when the head should move rightward. The criterion for switching between these two reference directions is defined to be the instant when the distance, Δy , between the position of the head along the global y axis, y_n , (perpendicular to the direction of locomotion) and the y axis coordinate of the CM of the snake robot, p_y , becomes greater than some predefined amplitude, Δy_{max} . This criterion is illustrated in Fig. 5. The distance, Δy , is easily calculated as a function of measured state values only. In previous work by the authors [21], it is shown that Δy is given by

$$\Delta y = y_n - p_y = \sum_{j=1}^{n-1} \frac{l}{n} j (\sin \theta_j + \sin \theta_{j+1}) \quad (71)$$

The reference angle of the head link is in other words set according to the rule

$$\begin{aligned} \text{Leftward motion: } \theta_{n,\text{ref}} &= \theta_{\text{left}} \text{ until } \Delta y > \Delta y_{\text{max}} \\ \text{Rightward motion: } \theta_{n,\text{ref}} &= \theta_{\text{right}} \text{ until } \Delta y < -\Delta y_{\text{max}} \end{aligned} \quad (72)$$

which means that $\theta_{n,\text{ref}} \in \{\theta_{\text{left}}, \theta_{\text{right}}\}$. To obtain this *link angle*, the head *joint angle*, ϕ_{n-1} , is controlled according to the reference

$$\phi_{n-1,\text{ref}} = \theta_{n-1} - \theta_{n,\text{ref}} \quad (73)$$

Note that $\phi_{n-1,\text{ref}}$ will experience a jump each time $\theta_{n,\text{ref}}$ switches. However, the actuator torque applied at joint $n-1$ will still be bounded since the derivative of $\phi_{n-1,\text{ref}}$ with respect to time is not included in the joint controller presented below in Section VII-E.

In order to generate a leader-follower based control reference to the remaining links, the head joint angle, ϕ_{n-1} , is propagated backwards along the snake body at a constant predefined propagation velocity, v_{ref} , and used as the reference angle for all subsequent joints. For a given choice of v_{ref} , the time offset, Δt , between two consecutive joints with intermediate distance $2l$ is found as $\Delta t = \frac{2l}{v_{\text{ref}}}$.

To summarize, the reference angles for all the joints of the snake robot in this leader-follower scheme are

$$\begin{aligned} \phi_{n-1,\text{ref}}(t) &= \theta_{n-1}(t) - \theta_{n,\text{ref}} \\ \phi_{i,\text{ref}}(t) &= \phi_{n-1}(t - (n-i-1)\Delta t) \quad \forall i = 1..n-2 \end{aligned} \quad (74)$$

The design parameters θ_{left} , θ_{right} , Δy_{max} , and v_{ref} were introduced in order to calculate these reference angles. Note that the implementation of (74) requires a buffer which keeps track of the history of the head joint angle, $\phi_{n-1}(t)$.

C. The jam detection scheme

A single joint of the snake robot is defined to be *jammed* if the deviation between the joint angle and its reference angle exceeds a certain limit, $\Delta\phi_{\text{max}}$. It is reasonable to assume that a jam of a single joint will resolve by itself. However, two jammed joints could be caused by a situation where the contact forces cause the jammed joints to act ‘against’ each other. This situation may not always resolve by itself. The entire snake robot is therefore defined to be *jammed* if two or more joints are jammed. If the robot is jammed over a continuous period longer than $t_{\text{jam,max}}$, the leader-follower scheme is stopped in order to carry out the jam resolution scheme. We let the robot execute jam resolution for a predefined amount of time, $t_{\text{resolution,max}}$, since it is difficult to come up with a specific criterion for when a jam has been resolved. Subsequently, the leader-follower scheme continues.

In summary, the design parameters $\Delta\phi_{\text{max}}$, $t_{\text{jam,max}}$, and $t_{\text{resolution,max}}$ determine the switching between the *leader-follower* scheme and the *jam resolution* scheme.

D. The jam resolution scheme

The idea behind the jam resolution scheme is to *rotate the links affected by contact forces so that the propulsive component of each contact force increases*. By *propulsive component*, we mean the force component in the desired direction of motion. In a jammed situation, the propulsive components of the contact forces from the obstacles are too small to overcome the friction forces on the robot. Rotating the contacted links (and thereby the direction of the contact forces) to increase the total propulsive contact force should therefore resolve the jammed situation.

The measured constraint force on link i is a scalar measurement denoted by $\hat{f}_{c,i} \in \mathbb{R}$, which represents the magnitude of the constraint force vector, $f_{c,i} \in \mathbb{R}^2$. The propulsive component, $\hat{f}_{\text{prop},i} \in \mathbb{R}$, of the constraint force on link i is illustrated in Fig. 6 and is defined as the force component in the desired direction of motion (the global positive x axis). The propulsive force is given by

$$\hat{f}_{\text{prop},i} = -\hat{f}_{c,i} \sin \theta_i \quad (75)$$

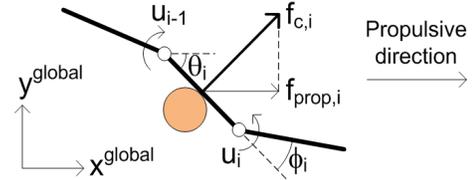


Fig. 6. The propulsive component, $f_{\text{prop},i}$, of the constraint force on link i .

The change of the propulsive force due to a change of the link angle is found by differentiating (75) with respect to θ_i :

$$\frac{\partial \hat{f}_{\text{prop},i}}{\partial \theta_i} = -\hat{f}_{c,i} \cos \theta_i \quad (76)$$

We see that a change of a constraint force near perpendicular to the direction of motion has greater effect on the propulsive force than a similar change of a constraint force near parallel to the direction of motion. During jam resolution, we should therefore prioritize to rotate links with a high propulsive force gradient with respect to the link angle. This suggests the following expression for the desired change of the link angle during jam resolution:

$$\Delta \theta_{i,\text{ref}} = k_\theta \frac{\partial \hat{f}_{\text{prop},i}}{\partial \theta_i} = -k_\theta \hat{f}_{c,i} \cos \theta_i \quad (77)$$

where k_θ is the gain of the jam resolution controller. We choose to only change the angle of link i while leaving the angle of link $i-1$ and $i+1$ unchanged. This means that $\Delta \theta_{i-1,\text{ref}} = \Delta \theta_{i+1,\text{ref}} = 0$. From the relation $\phi_i = \theta_i - \theta_{i+1}$, we may now write the desired change of the joint angles as

$$\begin{aligned} \Delta \phi_{i-1,\text{ref}} &= \Delta \theta_{i-1,\text{ref}} - \Delta \theta_{i,\text{ref}} = k_\theta \hat{f}_{c,i} \cos \theta_i \\ \Delta \phi_{i,\text{ref}} &= \Delta \theta_{i,\text{ref}} - \Delta \theta_{i+1,\text{ref}} = -k_\theta \hat{f}_{c,i} \cos \theta_i \end{aligned} \quad (78)$$

We now explain two important controller design choices. First of all, during jam resolution, we leave the head joint angle, ϕ_{n-1} , unchanged to maintain a smooth head angle. We thereby avoid that any jam resolution motion of the head link propagates backwards to all other links once the leader-follower scheme resumes. This would create undesirable shapes of the snake robot. Secondly, we use the constraint forces that were measured *at the instant the jam resolution scheme was initiated* as feedback so that the force measurements used in the feedback loop are *constant* during jam resolution. This ensures a steady rotation of the contacted links in accordance with the contact forces that produced the jam. If the force measurements had been updated *during* jam resolution, then jam resolution would very quickly be aborted for most of the contacted links because the link rotation generally causes the links to detach from the obstacles. We denote the measured constraint forces on all links *at the instant the jam resolution scheme was initiated* by $f_{\text{jam}} \in \mathbb{R}^n$.

From the above discussion, the reference angles for all the joints of the snake robot in the jam resolution state may now be summarized as

$$\begin{aligned} \phi_{n-1,\text{ref}} &= \phi_{n-1} \\ \phi_{i,\text{ref}} &= \phi_i + k_\theta (-f_{\text{jam},i} \cos \theta_i + f_{\text{jam},i+1} \cos \theta_{i+1}) \end{aligned} \quad (79)$$

where $i \in \{1, \dots, n-2\}$ and k_θ is a design parameter.

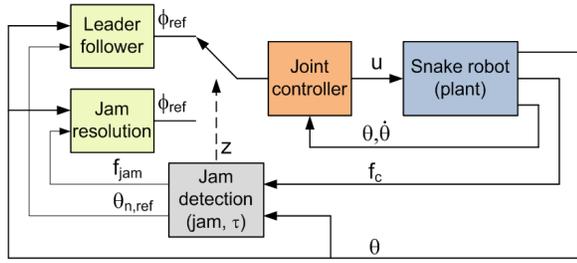


Fig. 7. The block diagram of the closed-loop system (hybrid plant and hybrid controller).

E. Low-level joint angle controller

A standard PD-controller is used to calculate the joint actuator torques, $u \in \mathbb{R}^{n-1}$, from the joint reference angles, $\phi_{\text{ref}} \in \mathbb{R}^{n-1}$. We limit the reference angles to the range $[-\phi_{\text{max}}, \phi_{\text{max}}]$ corresponding to the maximum deflection of the joints. The saturated reference angle of joint i is given by

$$\bar{\phi}_{i,\text{ref}} = \max(\min(\phi_{i,\text{ref}}, \phi_{\text{max}}), -\phi_{\text{max}}) \quad (80)$$

We thereby define the PD-controller of joint i as

$$u_i = k_P (\bar{\phi}_{i,\text{ref}} - \phi_i) - k_D \dot{\phi}_i \quad (81)$$

where k_P and k_D are the gains of the PD-controller. A velocity reference is not included in (81) since the transitions between the two schemes of the proposed control strategy produce steps in the reference angles, which would lead to large and undesirable velocity references.

F. The complete hybrid controller

We now provide a formal and precise specification of the complete *hybrid controller* of the snake robot in terms of the modelling framework of hybrid controllers described in Section II-B. We begin by defining the *state vector* of the hybrid controller as

$$\eta = (\theta_{n,\text{ref}}, f_{\text{jam}}, \text{jam}, \tau, z) \quad (82)$$

where $\theta_{n,\text{ref}} \in \{\theta_{\text{left}}, \theta_{\text{right}}\}$ is the current *reference angle* of the head link, $f_{\text{jam}} \in \mathbb{R}^n$ denotes the *measured constraint forces* on all links at the instant the jam resolution scheme is initiated, $\text{jam} \in \{0, 1\}$ is a *boolean variable* indicating if the robot is currently *jammed* ($\text{jam} = 1$ indicates a jam), $\tau \in \mathbb{R}_{\geq 0}$ is a *timer variable*, and finally $z \in \{0, 1\}$ is a *switching variable* that decides if the robot should execute the *leader-follower* scheme ($z = 0$) or the *jam resolution* scheme ($z = 1$). The block diagram of the closed-loop system is illustrated in Fig. 7. The state vector η is maintained inside the *jam detection* block.

The following subsections define the *jump set* (D_c), *jump map* (G_c), *flow set* (C_c), and *flow map* (F_c) of the hybrid controller, where subscript c is used to distinguish these sets from the corresponding sets of the hybrid plant in Section VI.

1) *Jump set*: A *jump* in the state vector η of the controller occurs either when the *direction* of the head link should change, when the jam state changes, or when the *jam resolution* scheme is initiated or stopped (controlled by switching the value of z).

In direct accordance with Section VII-B, the direction of the head link should change when the state of the plant and the controller, (x, η) , belongs to the jump set

$$D_{\text{dir}} = \{(x, \eta) | z = 0, \theta_{n,\text{ref}} = \theta_{\text{left}}, \Delta y > \Delta y_{\text{max}}\} \cup \{(x, \eta) | z = 0, \theta_{n,\text{ref}} = \theta_{\text{right}}, \Delta y < -\Delta y_{\text{max}}\} \quad (83)$$

In order to determine the jam state, we define the following index set corresponding to pairs of jammed joints:

$$I_{\text{jam}} = \{(i, j) | i \neq j, |\phi_i - \phi_{i,\text{ref}}| > \Delta \phi_{\text{max}}, |\phi_j - \phi_{j,\text{ref}}| > \Delta \phi_{\text{max}}\} \quad (84)$$

We consider *pairs* of jammed joints since the robot is defined to be jammed when two or more joints are jammed, i.e. when $I_{\text{jam}} \neq \emptyset$. In accordance with Section VII-C, the jump set of the jam state is given by

$$D_{\text{jam}} = \{(x, \eta) | z = 0, \text{jam} = 0, I_{\text{jam}} \neq \emptyset\} \cup \{(x, \eta) | z = 0, \text{jam} = 1, I_{\text{jam}} = \emptyset\} \quad (85)$$

In accordance with Section VII-C, the switching variable z should change when (x, η) belongs to the jump set

$$D_{\text{res}} = \{(x, \eta) | z = 0, \text{jam} = 1, \tau > t_{\text{jam,max}}\} \cup \{(x, \eta) | z = 1, \tau > t_{\text{resolution,max}}\} \quad (86)$$

The complete *jump set* of the hybrid controller may now be compactly expressed as

$$D_c = D_{\text{dir}} \cup D_{\text{jam}} \cup D_{\text{res}} \quad (87)$$

2) *Jump map*: From the description in Section VII-B, Section VII-C, and Section VII-D, we can directly state the *jump map* of the hybrid controller as

$$\eta^+ = G_c(x, \eta) = (\theta_{n,\text{ref}}^+, f_{\text{jam}}^+, \text{jam}^+, \tau^+, z^+) \quad (88)$$

where

$$\theta_{n,\text{ref}}^+ = \begin{cases} \theta_{\text{left}} & \text{when } (x, \eta) \in D_{\text{dir}}, \Delta y < -\Delta y_{\text{max}} \\ \theta_{\text{right}} & \text{when } (x, \eta) \in D_{\text{dir}}, \Delta y > \Delta y_{\text{max}} \\ \theta_{n,\text{ref}}^- & \text{otherwise} \end{cases}$$

$$f_{\text{jam}}^+ = f_c$$

$$\text{jam}^+ = \begin{cases} 1 & \text{when } (x, \eta) \in D_{\text{jam}}, \text{jam} = 0 \\ 0 & \text{when } (x, \eta) \in D_{\text{jam}}, \text{jam} = 1 \\ & \text{or } (x, \eta) \in D_{\text{res}} \\ \text{jam}^- & \text{otherwise} \end{cases} \quad (89)$$

$$\tau^+ = \begin{cases} 0 & \text{when } (x, \eta) \in D_{\text{jam}} \cup D_{\text{res}} \\ \tau^- & \text{otherwise} \end{cases}$$

$$z^+ = \begin{cases} 1 & \text{when } (x, \eta) \in D_{\text{res}}, z = 0 \\ 0 & \text{otherwise} \end{cases}$$

3) *Flow set*: We define the *flow set* of the hybrid controller so that the state vector η always flows as long as the *jump set* is empty. The *flow set* is therefore simply given as

$$C_c = \{(x, \eta) | (x, \eta) \notin D_c\} \quad (90)$$

4) *Flow map*: The only variable in the state vector η that should change between jumps is the timer variable, τ . Since the time derivative of τ is 1, the *flow map* is given by

$$\dot{\eta} = F_c(x, \eta) = \begin{pmatrix} \dot{\theta}_{n,\text{ref}} \\ \dot{f}_{\text{jam}} \\ \dot{\text{jam}} \\ \dot{\tau} \\ \dot{z} \end{pmatrix} = (0, 0_{n \times 1}, 0, 1, 0) \quad (91)$$

5) *Calculation of the control input for the plant:* The joint torques, u , of the snake robot are calculated as

$$u = \begin{cases} \kappa_{\text{nojam}}(x, \eta) & \text{when } z = 0 \\ \kappa_{\text{jam}}(x, \eta) & \text{when } z = 1 \end{cases} \quad (92)$$

where $\kappa_{\text{nojam}} : C_c \rightarrow \mathbb{R}^{n-1}$ is given by (74) and (81), and $\kappa_{\text{jam}} : C_c \rightarrow \mathbb{R}^{n-1}$ is given by (79) and (81).

G. Summary of the closed-loop system

The *closed-loop system* is a hybrid system with *state* (x, η) and *data* (C, F, D, G) , where x is the state of the hybrid plant and η is the state of the hybrid controller. The closed-loop system flows as long as neither x nor η should jump. In other words, (x, η) flows as long as both $x \in C_p$ and $(x, \eta) \in C_c$, and jumps when $x \in D_p$ or when $(x, \eta) \in D_c$. The data of the closed-loop system is therefore given by

$$C = \{(x, \eta) | x \in C_p, (x, \eta) \in C_c\} \quad (93)$$

$$F(x, \eta) = \begin{bmatrix} \dot{x} \\ \dot{\eta} \end{bmatrix} = \begin{bmatrix} F_p(x, \kappa(x, \eta)) \\ F_c(x, \eta) \end{bmatrix}, (x, \eta) \in C \quad (94)$$

$$D = \{(x, \eta) | x \in D_p \vee (x, \eta) \in D_c\} \quad (95)$$

$$G(x, \eta) = \begin{bmatrix} x^+ \\ \eta^+ \end{bmatrix}, (x, \eta) \in D \quad (96)$$

where

$$x^+ = \begin{cases} G_p(x) & \text{when } x \in D_p \\ x^- & \text{otherwise} \end{cases} \quad (97)$$

$$\eta^+ = \begin{cases} G_c(x, \eta) & \text{when } (x, \eta) \in D_c \\ \eta^- & \text{otherwise} \end{cases} \quad (98)$$

By design, the evolution of η can always be uniquely determined. The control input, u , to the plant is therefore always well-defined. We can therefore conclude that Proposition 22 also applies to the closed-loop system.

VIII. SIMULATION RESULTS

This section investigates the hybrid model and controller summarized in Section VII-G through simulations. The simulator was implemented in *Matlab R2008b* on a laptop running *Windows XP*. The continuous dynamics of the model was calculated with the *ode45* solver in Matlab with a relative and absolute error tolerance of 10^{-3} .

A. Comparison of the model with previous experimental results and simulation results

In order to investigate the validity of the hybrid model proposed in this paper, we have compared simulation results from the hybrid model (70) with experimental and simulation results presented in [1]. The work in [1], which was described in the introduction of this paper, presents experimental results of obstacle-aided locomotion and also simulation results from a model developed based on the framework of nonsmooth dynamics. The joints of both the physical and the simulated snake robot in [1] were controlled according to the *serpenoid curve* motion introduced by Hirose [3]:

$$\phi_{i,\text{ref}} = \alpha \sin(\omega t + (i-1)\delta) \quad (99)$$

with $i \in \{1, \dots, n-1\}$, $\alpha = 40^\circ$, $\omega = 80^\circ/\text{s}$, and $\delta = 50^\circ$. Furthermore, the parameters characterizing the snake robot

in [1] were $n = 11$, $l = 0.061$ m, $m = 0.682$ kg, and $J = 0.0013$ kgm². Obstacles of diameter 25 cm were placed in accordance with the expected motion of the robot (see [1] for details). We implemented the model (70) in accordance with the above presented parameters and employed the PD controller in (81) to control the joints of the robot according to (99) with $k_P = 20$ and $k_D = 5$.

The experimental and simulation results from [1] are given in Fig. 8 and Fig. 9. The corresponding simulation results from the model presented in this paper is given in Fig. 10 and Fig. 11. We see that there is a close resemblance between the results from [1] and the simulation results from the model (70). The y direction amplitude in Fig. 9 and Fig. 11 are slightly different because the model in [1] also includes the width of each link, while the model in this paper assumes the width to be infinitesimal. In summary, the simulation result supports the conjecture that, despite its simplifying assumptions, the model proposed in this paper captures the essential part of the dynamics of a snake robot interacting with obstacles.

B. Simulation of the hybrid controller

1) *Simulation parameters:* The parameters characterizing the simulated snake robot were $n = 10$, $l = 0.07$ m, $m = 1$ kg, and $J = 0.0016$ kgm². These parameters characterize a physical snake robot recently developed by the authors.

The various controller parameters were set to $\theta_{\text{left}} = 50^\circ$, $\theta_{\text{right}} = -50^\circ$, $\Delta y_{\text{max}} = 0.14$ m, $v_{\text{ref}} = 0.2$ m/s, $\Delta \phi_{\text{max}} = 20^\circ$, $t_{\text{jam,max}} = 0.5$ s, $t_{\text{resolution,max}} = 0.5$ s, $k_\theta = 0.05$, $\phi_{\text{max}} = 50^\circ$, $k_P = 20$, and $k_D = 5$. These parameters were found based on both physical insight and through trial and error.

The ground friction coefficient was set to $\mu = 0.3$ and the obstacle friction coefficient was set to $\mu_o = 0.2$. Two different obstacle environments were considered, denoted the *structured* and *unstructured* obstacle environment, respectively. In the *structured* environment, the obstacles were chosen to be three rows (parallel to the x axis) of circular objects with equal radius $R_{O_j} = 10$ cm, $j \in \{1, \dots, k\}$. The center distance between two obstacles in a row and the distance between two rows were 25 cm. The middle row was displaced with respect to the other two rows by 12.5 cm along the x axis. In the *unstructured* environment, obstacles of varying radius were placed in a random fashion around the snake robot.

The initial link angles and position of the snake robot were set to $\theta_0 = [7, -32, -57, -46, -8, 33, 53, 45, 12, -23]^T$ [deg] and $p_0 = [0, 0]^T$, respectively. The initial shape was more or less randomly chosen in order to give the snake an initial curl around the obstacles without intersecting them.

In order to ensure a unique solution to the LCP in (53), we verified at each timestep of a simulation that the matrix A of the LCP in (53) was a P -matrix by employing the P -matrix test algorithm presented in [24].

2) *Attempting a serpenoid curve in a structured obstacle environment:* We begin by illustrating the need for feedback of obstacle contact forces by repeating the open-loop serpenoid curve motion presented in Section VIII-A when the obstacles are *not* placed in accordance with the expected evolution of the shape and position of the snake robot. We chose the gait pattern parameters in (99) as $\alpha = 40^\circ$, $\omega = 40^\circ/\text{s}$, and $\delta = 40^\circ$. The initial ($t = 0$ s) and final ($t = 20$ s) shape and position of the snake robot are shown at the top of Fig. 12, and the trace of the head is indicated with a dotted line. We see that

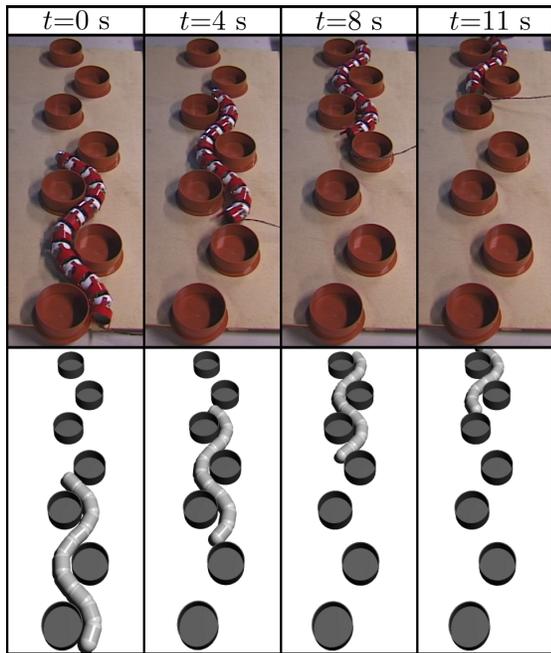


Fig. 8. A reprint of Fig. 15 from [1], which shows experimental results (top) and simulation results (bottom) of serpenoid curve motion in an environment with obstacles.

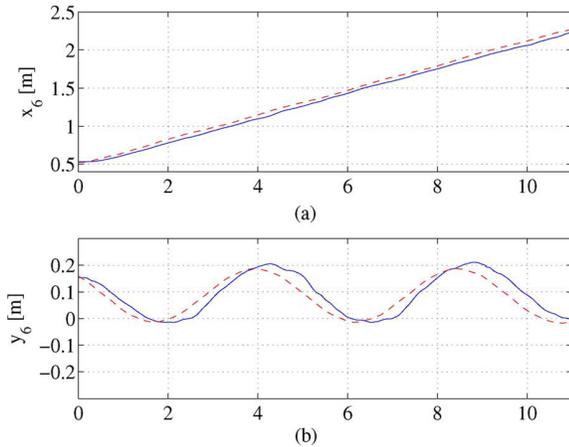


Fig. 9. A reprint of Fig. 14 from [1], which shows the simulated (dashed line) and experimentally measured (solid line) position of link 6 during serpenoid curve motion in an environment with obstacles.

the robot was only able to crawl about 0.5 m in 20 s. Since there was no adaptation of the motion to the environment, the obstacles prevented the snake robot from assuming the serpenoid curve. This simulation illustrates that a snake robot is generally unable to locomote in a cluttered environment when the joint motion is preprogrammed.

3) *Hybrid controller in a structured and unstructured obstacle environment*: The next simulation shows the effectiveness of the *hybrid controller* proposed in this paper when the snake robot moves for 30 s in a *structured* and *unstructured* obstacle environment, respectively. The initial ($t = 0$ s) and final ($t = 30$ s) shape and position of the snake robot in these two environments are shown in the middle and at the bottom of Fig. 12, respectively, while a plot of the x direction velocity of the snake, \dot{p}_x , is shown in Fig. 13. Vertical dashed lines in

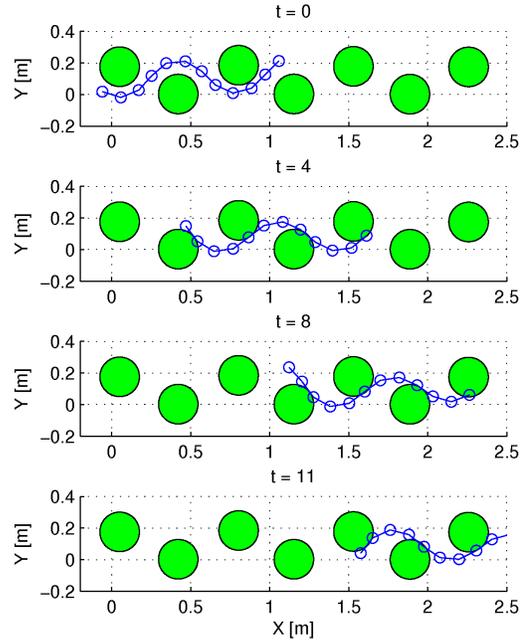


Fig. 10. Simulation of serpenoid curve motion in an environment with obstacles based on the hybrid model presented in this paper. The simulated scenario is similar to the scenario shown in Fig. 8.

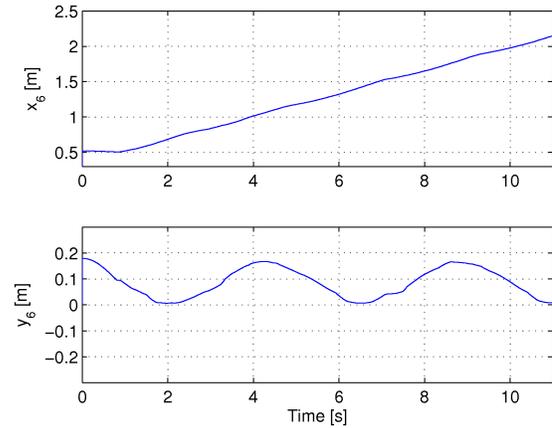


Fig. 11. The simulated position of link 6 during serpenoid curve motion in an environment with obstacles based on the hybrid model presented in this paper.

Fig. 13 indicate time instants where the *jam resolution* scheme is initiated. After 30 s, the snake robot has managed to crawl about 2.5 m along the global x axis in both environments. Fig. 13 shows that the velocity in both environments varies around 10 cm/s. The jam resolution scheme was initiated six and eight times, respectively, in the structured and unstructured environment, and all the jams were successfully resolved by the proposed algorithm.

In order to give an idea of the forces involved in obstacle-aided locomotion, we provide a plot of the measured constraint forces on the center link (link 5) of the snake robot, $\hat{f}_{c,5}$, at the top of Fig. 14. The torque input applied to joint 5, u_5 , is plotted at the bottom of this figure. We see that constraint forces above 200 N occur during the motion, and that the

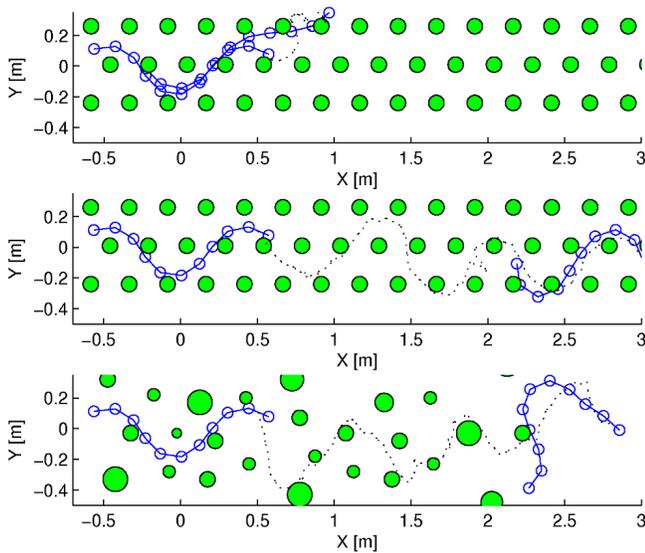


Fig. 12. The initial and final shape and position, and the trace of the head of the snake robot. Top: Serpentine curve (jam resolution disabled). Middle: Jam resolution in the structured obstacle environment. Bottom: Jam resolution in the unstructured obstacle environment.

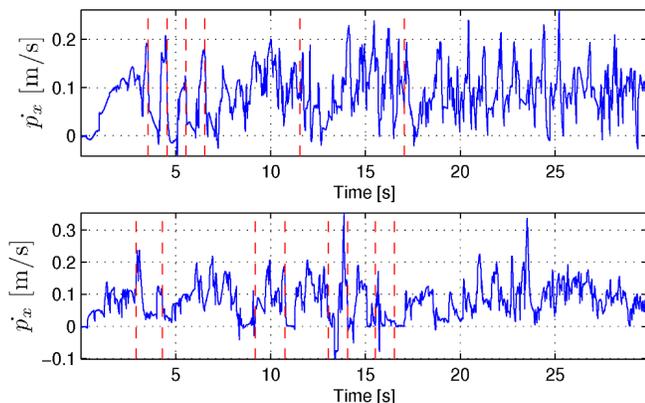


Fig. 13. The global x direction velocity of the snake robot in the structured (top) and unstructured (bottom) obstacle environment. Vertical dashed lines indicate when the *jam resolution* scheme is initiated.

applied joint torque is sometimes as high as 10 Nm.

To clearly illustrate the effect of the jam resolution scheme, a plot of the snake robot before (dashed) and after (solid) a jam resolution is shown in Fig. 15. The figure shows the jam occurring in the structured obstacle environment at time $t = 17.05$ s and ending at time $t = 17.55$ s, which is caused by contact forces acting on links 4 and 8. The jam is resolved by rotating link 4 clockwise and link 8 counterclockwise, thereby increasing the propulsive components of the two constraint forces enough to overcome the friction forces from the ground and the obstacles.

IX. CONCLUSIONS AND FUTURE WORK

This paper has presented a *hybrid model* of the dynamics of a planar snake robot interacting with obstacles in its environment where obstacle contact forces are calculated by formulating and solving a *linear complementarity problem* (LCP). A *hybrid controller* was proposed for enabling the

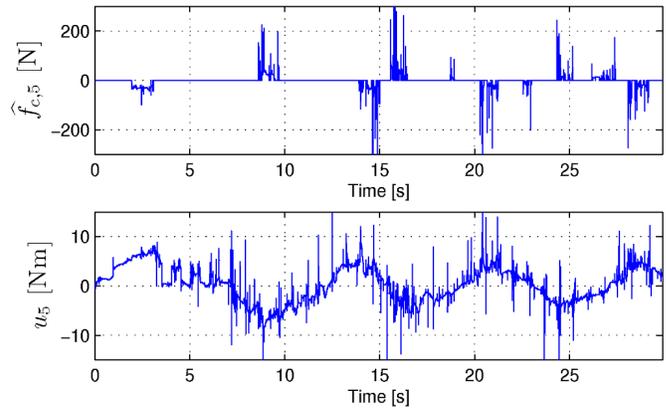


Fig. 14. The obstacle constraint forces on link 5 (top) and the torques applied to joint 5 (bottom) in the structured obstacle environment.

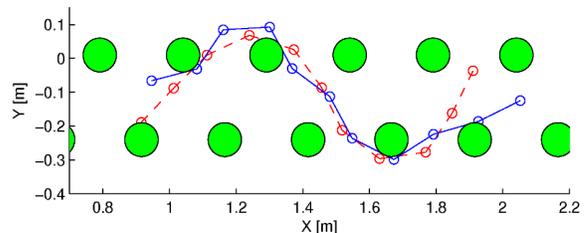


Fig. 15. The shape of the snake robot before (dashed) and after (solid) an instance of the jam resolution scheme.

snake robot to propel itself forward by using obstacles along its path as push-points (denoted *obstacle-aided locomotion*) while preventing the obstacles from jamming the motion.

The *existence* and *uniqueness* properties of the state evolution of the hybrid plant (the snake robot) were considered. In particular, it was shown that the evolution of the state x of the plant from any initial state can always be uniquely determined when the obstacles are frictionless ($\mu_o = 0$). With obstacle friction, we showed that there exists a $\mu_o^* > 0$ such that existence and uniqueness of the evolution of x is guaranteed for $\mu_o \in [0, \mu_o^*)$, but not guaranteed for $\mu_o \geq \mu_o^*$.

The hybrid controller was designed so that the evolution of its state η can always be uniquely determined. We could therefore conclude that the existence and uniqueness properties of the plant also applies to the closed-loop system.

The simulation results validated the hybrid modelling approach and illustrated how the proposed jam detection and resolution scheme can help to maintain the propulsion of a snake robot in a cluttered environment.

In future work, the authors will validate the presented simulation results through experiments with a physical snake robot and also investigate the stability properties of the proposed hybrid controller.

REFERENCES

- [1] A. A. Transeth, R. I. Leine, C. Glocker, K. Y. Pettersen, and P. Liljebäck, "Snake robot obstacle aided locomotion: Modeling, simulations, and experiments," *IEEE Trans. Robot.*, vol. 24, no. 1, pp. 88–104, February 2008.
- [2] J. Gray, "The mechanism of locomotion in snakes," *J. Exp. Biol.*, vol. 23, no. 2, pp. 101–120, 1946.
- [3] S. Hirose, *Biologically Inspired Robots: Snake-Like Locomotors and Manipulators*. Oxford: Oxford University Press, 1993.

- [4] B. Moon and C. Gans, "Kinematics, muscular activity and propulsion in gopher snakes," *Journal of Experimental Biology*, vol. 201, pp. 2669–2684, 1998.
- [5] J. Ostrowski and J. Burdick, "The geometric mechanics of undulatory robotic locomotion," *Int. J. Robot. Res.*, vol. 17, no. 7, pp. 683–701, 1998.
- [6] P. Prautsch and T. Mita, "Control and analysis of the gait of snake robots," in *Proc. IEEE Int. Conf. Control Applications*, Kohala Coast, HI USA, 1999, pp. 502–507.
- [7] H. Date, M. Sampei, and S. Nakaura, "Control of a snake robot in consideration of constraint force," in *Proc. IEEE Int. Conf. on Control Applications*, 2001, pp. 966–971.
- [8] S. Ma, "Analysis of creeping locomotion of a snake-like robot," *Adv. Robotics*, vol. 15, no. 2, pp. 205–224, 2001.
- [9] F. Matsuno and H. Sato, "Trajectory tracking control of snake robots based on dynamic model," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2005, pp. 3029–3034.
- [10] M. Saito, M. Fukaya, and T. Iwasaki, "Serpentine locomotion with robotic snakes," *IEEE Contr. Syst. Mag.*, vol. 22, no. 1, pp. 64–81, February 2002.
- [11] I. Grabec, "Control of a creeping snake-like robot," in *Proc. 7th Int. Workshop on Advanced Motion Control*, July 2002, pp. 526–513.
- [12] K. McIsaac and J. Ostrowski, "Motion planning for anguilliform locomotion," *IEEE Trans. Robot. Autom.*, vol. 19, no. 4, pp. 637–625, August 2003.
- [13] G. P. Hicks, "Modeling and control of a snake-like serial-link structure," Ph.D. dissertation, North Carolina State University, 2003.
- [14] P. Liljebäck, Ø. Stavadahl, and K. Y. Pettersen, "Modular pneumatic snake robot: 3D modelling, implementation and control," in *Proc. 16th IFAC World Congress*, July 2005.
- [15] Z. Bayraktaroglu and P. Blazevic, "Understanding snakelike locomotion through a novel push-point approach," *J. Dyn. Syst. - Trans. ASME*, vol. 127, no. 1, pp. 146–152, March 2005.
- [16] I. Tanev, T. Ray, and A. Buller, "Automated evolutionary design, robustness, and adaptation of sidewinding locomotion of a simulated snake-like robot," *IEEE Trans. on Robotics*, vol. 21, no. 4, pp. 632–645, August 2005.
- [17] Z. Y. Bayraktaroglu, "Snake-like locomotion: Experimentations with a biologically inspired wheel-less snake robot," *Mechanism and Machine Theory*, vol. 44, no. 3, pp. 591–602, 2008.
- [18] Y. Shan and Y. Koren, "Design and motion planning of a mechanical snake," *IEEE Trans. Syst. Man Cyb.*, vol. 23, no. 4, pp. 1091–1100, July-August 1993.
- [19] A. J. v. d. Schaft and J. M. Schumacher, *An Introduction to Hybrid Dynamical System*, M. Thoma and M. Morari, Eds. Springer, 2000.
- [20] B. Brogliato, *Nonsmooth Mechanics*, 2nd ed. London: Springer, 1999.
- [21] P. Liljebäck, K. Y. Pettersen, and Ø. Stavadahl, "Modelling and control of obstacle-aided snake robot locomotion based on jam resolution," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2009, pp. 3807–3814.
- [22] R. W. Cottle, J. S. Pang, and R. E. Stone, *The linear complementarity problem*. Academic Press, 1992.
- [23] R. Goebel, R. Sanfelice, and A. Teel, "Hybrid dynamical systems," *IEEE Control Systems Magazine*, vol. 29, no. 2, pp. 28–93, April 2009.
- [24] M. J. Tsatsomeros and L. Li, "A recursive test for P-matrices," *Bit Numerical Mathematics*, vol. 40, pp. 410–414, 2000.
- [25] H. Goldstein, C. Poole, and J. Safko, *Classical Mechanics - Third Edition*. Addison Wesley, 2002.
- [26] P. Lötstedt, "Mechanical systems of rigid bodies subject to unilateral constraints," *SIAM J. Appl. Math.*, vol. 42, no. 2, pp. 281–296, April 1982.
- [27] D. S. Bernstein, *Matrix Mathematics: Theory, Facts, and Formulas (Second Edition)*, 2nd ed. Princeton University Press, 2009.
- [28] P. Lötstedt, "Coulomb friction in two-dimensional rigid body systems," *Zeitschrift fr Angewandte Mathematik und Mechanik*, vol. 61, pp. 605–615, 1981.
- [29] J. Trinkle, J. S. Pang, S. Sudarsky, and G. Lo, "On dynamic multi-rigid-body contact problems with coulomb friction," *Zeitschrift fr Angewandte Mathematik und Mechanik*, vol. 77, pp. 267–280, 1997.
- [30] M. T. Mason and Y. Wang, "On the inconsistency of rigid-body frictional planar mechanics," in *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 1, Apr 1988, pp. 524–528.
- [31] P. Song, P. Kraus, V. Kumar, and P. Dupont, "Analysis of rigid body dynamic models for simulation of systems with frictional contacts," *Journal of Applied Mechanics*, vol. 68, pp. 118–128, 2000.



Pål Liljebäck (M'08) received the MSc degree from the Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Trondheim, Norway, in 2004. He is currently a PhD candidate at this department at NTNU, and simultaneously works as a research scientist at SINTEF ICT, Department of Applied Cybernetics, Trondheim, Norway, which is a Norwegian research organization. His research interests include modelling and control of dynamical systems, and design and implementation of mechatronic systems.



Kristin Y. Pettersen (S'93–M'98–SM'04) received her MSc and PhD degree in Electrical Engineering at the Norwegian University of Science and Technology (NTNU), Trondheim, Norway, in 1992 and 1996 respectively. In 1996 she became Associate Professor, and in 2002 Professor, at the Department of Engineering Cybernetics, NTNU. In 1999 she was a Visiting Fellow at the Department of Mechanical and Aerospace Engineering, Princeton University, Princeton, NJ and in 2008 Visiting Professor at Section for Automation and Control, University of Aalborg, Denmark. She has published more than 100 conference and journal papers. In 2006 she received the IEEE Transactions on Control Systems Technology Outstanding Paper Award. She is a senior member of IEEE and Associate Editor of the IEEE Transactions on Control Systems Technology. She furthermore holds several board positions in industrial and research companies. Her research interests include nonlinear control of mechanical systems with applications to robotics, satellites, AUVs and ships.



Øyvind Stavadahl (M'09) was born in Lillehammer, Norway, 1967. He received the MSc and PhD from the Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Trondheim, Norway, in 1994 and 2002, respectively. He became a Research Scientist in 1998 and in 2004 a Senior Scientist at SINTEF ICT, Department of Applied Cybernetics, where he currently is a Scientific Advisor. In 2004 he became a Postdoctoral Fellow, in 2007 Lecturer and in 2008 Associate Professor at the Department of Engineering Cybernetics, NTNU. In 2005 he was a Visiting Fellow at the Institute of Biomedical engineering, University of New Brunswick, Fredericton, NB, Canada. Dr. Stavadahl is a Board Member of the Norwegian Association of Automatic Control. His research interests include rehabilitation engineering, human movement analysis and biomimetic robotics.



Tommy Gravdahl (S'94–M'98–SM'04) graduated siving (1994) and dr.ing (1998) in engineering cybernetics at the Norwegian University of Science and Technology (NTNU), Trondheim, Norway. He was appointed Associate Professor (2001) and Professor (2005) at the Department of Engineering Cybernetics, NTNU. In the academic year 2007/08 he was Visiting Professor with The Centre for Complex Dynamic Systems and Control (CDSC), The University of Newcastle, Australia. He has published more than 75 papers at conferences and in journals and is author of *Compressor Surge and Rotating Stall: Modeling and Control* (London, Springer, 1999), co-author of *Modeling and Simulation for Automatic Control* (Trondheim, Marine Cybernetics, 2002) and co-editor of *Group Coordination and Cooperative Control* (Berlin, Springer, 2006). His current research interests include mathematical modeling and nonlinear control in general, modeling and control of turbomachinery and control of spacecraft, robots and nanopositioning devices. Professor Gravdahl received the IEEE Transactions on Control Systems Technology Outstanding Paper Award in 2000 and is member of the editorial board of *Simulation Modeling Practice and Theory*.