Modeling and Path-following for a Snake Robot with Active Wheels

Boathy Murugendran, Aksel Andreas Transeth and Sigurd Aksnes Fjerdingen

Abstract— **Snake robots with active wheels provide interesting opportunities within many areas such as inspection and maintenance and search and rescue operations. The highly-articulated body of a snake robot combined with the advantages of wheeled locomotion makes it ideal for locomotion in, for example, pipes and other narrow or constricted structures. In this paper we present a mathematical model of the dynamics of a snake robot with active wheels together with a novel path-following approach for such robots. The path-following approach includes both how to find a desired turning angle for the snake robot head given a reference path, together with a module coordination strategy based on a n-trailer kinematic model. The path-following approach is tested and verified by simulations with the dynamic model. In addition, simulations suggest that the proposed approach results in reduced commanded joint and wheels shaft torques and, in most cases, a reduced path-following error compared to an implementation of a follow-the-leader algorithm.**

I. Introduction

Search and rescue operations in confined and narrow spaces or in the remains of collapsed and unstable buildings, in addition to inspection, maintenance, and repair operations of pipe structures in sewage and ventilation systems can be unreachable and dangerous to humans. Therefore alternatives such as employing robots need to be investigated.

Snake robots are especially advantageous for moving through narrow spaces and tight bends while carrying payloads required to perform the necessary operations. This is why locomotion of snake robots is a predominant research area when it comes to automated inspection of pipe structures and some search and rescue applications.

Wheeled snake robots have the advantage of the shape and build of a snake, while maintaining the direct motion of wheeled vehicles, thereby combining the best of both worlds. Multiple active wheels and joints allow the robot to continue forward propulsion even when several of the wheels are not in contact with the driving surface. This is a necessity when more complicated maneuvering than driving on flat surfaces is required, like traversing obstacles or climbing step-like structures.

Snake robots with active wheels and active joints are being increasingly studied in literature (see, e.g., [1], [2], [3], [4]). Scholl et al., for example, present an autonomous sewer inspection robot [2], and provide experimental results in [5] on the same robot demonstrating the ability of wheeled, multi-joint robots to maneuver in tight bends and traverse step-like obstacles. Hirose et al. present similar successes in maneuverability with a snake robot called ACM-R4 [1].

Several academic attempts for doing path-following with a snake robot with active wheels are based on a "follow-the-leader" (FTL) principle. FTL is based on that all links are set to follow a curve traced by the robot head. This approach is implemented in, e.g., [1] and [4].

The methods proposed for path-following have proved efficient for initial laboratory testing. However, there are still improvements that can be made both on the accuracy of path-following and also on the amount of joint and wheel torque required to realize the various motions. In addition, it is important to have a simulation environment in order to be able to test various path-following strategies and other aspects of locomotion without having to build and maintain an actual robot. Nevertheless, no mathematical model of the dynamics of a snake robot with active wheels has been found in literature. Such a model is important in order to be able to simulate and investigate factors such as the behavior of the robot and the forces and torques involved during locomotion.

In this paper we present a model of the dynamics of a snake robot with active wheels together with a novel path-following approach for such robots. The dynamic model is employed for simulations and is based on a non-minimal set of coordinates which is advantageous for numerical treatment of the system equations. Moreover, friction forces between the wheels of the robot and the ground surface is modeled as Coulomb friction and described within a framework of non-smooth dynamics and convex analysis. This allows us to describe "true" stick-slip transitions since the model can account for non-zero friction forces for zero velocities (i.e. during "stick-mode"). The path-following approach for wheeled snake robots presented in this paper is based on control strategies for n-trailer vehicles. Coordination of the many joints and wheel velocities of a snake robot is achieved in a passive-like manner which reduces the amount of necessary control torques. Simulations show that a 6-link snake robot is able to follow a set of various curves using the n-trailer approach. In addition, these results are compared with simulations of an implementation of a follow-the-leader approach. It is shown that the n-trailer-based path-following approach is able to follow a pre-defined path more closely for more complex paths in addition to requiring less total commanded torques.

This paper is organized as follows. Section II presents a dynamic model for simulations of a snake robot with active wheels. Section III details a controller for steering the first

B. Murugendran is with the Department of Engineering Cybernetics, Norwegian University of Science and Technology, NO-7491 Trondheim, Norway. He is also with the Department of Applied Cybernetics, SINTEF ICT, NO-7465 Trondheim, Norway. murugend@stud.ntnu.no

A. A. Transeth and S. A. Fjerdingen are with the Department of Applied Cybernetics, SINTEF ICT, NO-7465 Trondheim, Norway. {aksel.a.transeth, sigurd.fjerdingen}@sintef.no

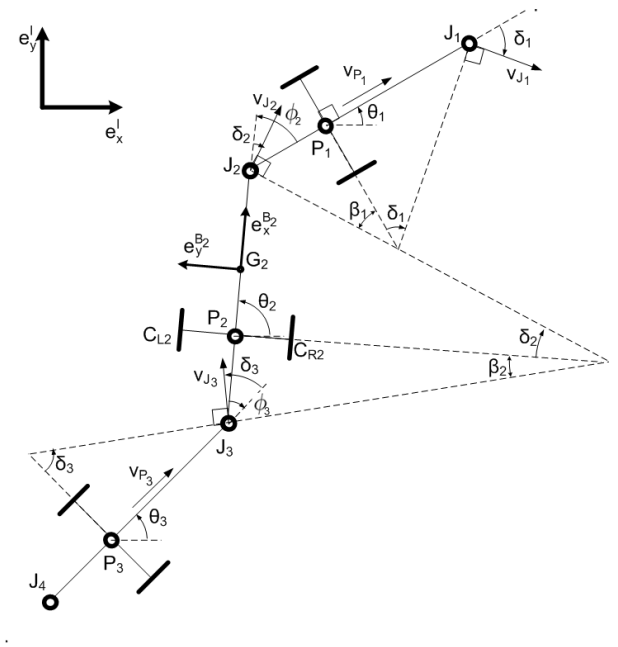

Fig. 1. Schematic of a general multi-joint robot/vehicle.

link of the robot along a desired path, while Section IV presents a new approach to path-following for the entire robot body. Section V summarizes a possible implementation of a follow-the-leader approach. Simulation results and discussions are presented in Section VI, while conclusions and further work are presented in Section VII.

II. Modeling for Simulation

In this section we present a non-smooth dynamic model of a snake robot with active wheels. The model parameters are based on the snake robot PIKo which has been developed at the Norwegian research organization SINTEF in Trondheim, Norway [6].

A mathematical model that includes the dynamics of a snake robot with active wheels is important in order to validate through simulation the path-following algorithm proposed later in this paper. The model of the dynamics extends a non-smooth model of a wheel-less snake robot presented in Transeth et al. [7] in that *active wheels* are added.

First a short system description and notation overview will be given. Then, the system equations will be described.

A. System Description and Notation

The planar snake robot modeled in this paper has n links connected by $n-1$ active rotational joints (the terms "link" and "module" will be employed interchangeably throughout this paper). Two wheels are mounted on the sides of each link (see Fig. 1) and the wheels are actuated by employing a torque around the shaft on which both wheels are mounted. For link i let L_i be the length of the link and L_{w_i} be the radii of each of the two wheels.

Let I denote an inertial frame. Let $B_i = \{G_i, \boldsymbol{e}_x^{B_i}, \boldsymbol{e}_y^{B_i}\}$ be a body-fixed frame with origin G_i which is the midpoint and center of gravity (CG) of link i and coordinate axis $\boldsymbol{e}_x^{B_i} \in \mathbb{R}^2$ pointing along link i toward link $i-1$. Let ${}_{B_i}\boldsymbol{r}_{AB} \in \mathbb{R}^2$ be a vector from a point A to a point B given in frame B_i. Let $\mathbf{R}_{B_i}^I \in \mathbb{R}^{2\times 2}$ be a rotation matrix such that ${}_I\boldsymbol{r} = \mathbf{R}_{B_i}^I {}_{B_i}\boldsymbol{r}$.

The position and orientation, and velocity, of each link are

$$\boldsymbol{q}_i = \begin{bmatrix} {}_I\boldsymbol{r}_{G_i} \\ \theta_i \\ \theta_{w_i} \end{bmatrix}, \text{ and } \boldsymbol{u}_i = \begin{bmatrix} {}_I\boldsymbol{v}_{G_i} \\ \omega_i \\ \omega_{w_i} \end{bmatrix}, \tag{1}$$

where ${}_I\boldsymbol{r}_{G_i} \in \mathbb{R}^2$ is the position of the CG of link i, $\theta_i \in S^1$ is the angle between the axes $\boldsymbol{e}_x^I$ and $\boldsymbol{e}_x^{B_i}$ and $\theta_{w_i} \in S^1$ is the wheel rotation angle. For $\boldsymbol{u}_i$ we have that ${}_I\boldsymbol{v}_{G_i} := {}_I\dot{\boldsymbol{r}}_{G_i}$, $\omega_i := \dot{\theta}_i$, and $\omega_{w_i} := \dot{\theta}_{w_i}$. All the positions and orientations, and velocities, are gathered in two vectors:

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{q}_1 \\ \boldsymbol{q}_2 \\ \vdots \\ \boldsymbol{q}_n \end{bmatrix}, \text{ and } \boldsymbol{u} = \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \\ \vdots \\ \boldsymbol{u}_n \end{bmatrix}. \tag{2}$$

B. Dynamics

The starting point for describing the dynamics of a snake robot with active wheels within the context of non-smooth dynamics and convex analysis is the *equality of measures* as introduced in [8]. This is found to be

$$\mathbf{M}\mathrm{d}\boldsymbol{u} - \mathrm{d}\boldsymbol{R} = \boldsymbol{\tau}_C \mathrm{d}t, \tag{3}$$

where $\mathbf{M} \in \mathbb{R}^{4n\times 4n}$ is the mass matrix, $\mathrm{d}\boldsymbol{u}$ is a differential measure (see [9], [10], [11], [7] for a more thorough treatment of non-smooth dynamics and differential measures), $\boldsymbol{\tau}_C \in \mathbb{R}^{4n}$ contains all the actuator torques, and $\mathrm{d}\boldsymbol{R} \in \mathbb{R}^{4n}$ accounts for the Coulomb friction forces between the wheels and the ground surface and the bilateral constraint forces in the joints. We will elaborate more thoroughly on $\mathrm{d}\boldsymbol{R}$ in the following. However, first the mass matrix and the vector of actuator torques will be presented.

The mass matrix is

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{M}_n \end{bmatrix} \in \mathbb{R}^{4n\times 4n}, \tag{4}$$

where $\mathbf{M}_i = \mathrm{diag}\left(\begin{bmatrix} m_i & m_i & \Theta_i & \Theta_{w_i} \end{bmatrix}\right) \in \mathbb{R}^{4\times 4}$, m_i and Θ_i are the mass and moment of inertia of link i, respectively, and Θ_{w_i} is the total moment of inertia of the two wheels connected to link i. Hence, the mass matrix is diagonal and constant which is advantageous for numerical treatment during simulations.

The snake robot is equipped with $n-1$ joint actuators and n wheel actuators which all are modeled as applied torques. Let τ_i be the torque applied to joint i (between link $i-1$ and i) and let τ_{w_i} be the torque applied to the wheel shaft connected to link i. Then the total torque applied to each link module is

$$\boldsymbol{\tau}_{C_{\text{link } i}} = \begin{bmatrix} 0 & 0 & \tau_{i+1} - \tau_i & \tau_{w_i} \end{bmatrix}^{\mathrm{T}} \tag{5}$$

for $i = 1, \ldots, n$ where $\tau_1 = \tau_{n+1} = 0$. We now find that the vector of all applied torques in (3) is

$$\boldsymbol{\tau}_C = \begin{bmatrix} \boldsymbol{\tau}_{C_{\text{link }1}}{}^{\mathrm{T}} & \boldsymbol{\tau}_{C_{\text{link }2}}{}^{\mathrm{T}} & \cdots & \boldsymbol{\tau}_{C_{\text{link }n}}{}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{6}$$

The force measure $\mathrm{d}\boldsymbol{R}$ accounts for tangential friction forces ("tangential" with respect to the ground surface) and bilateral constraint forces and is written

$$\mathrm{d}\boldsymbol{R} = \mathbf{W}_T \mathrm{d}\boldsymbol{P}_T + \mathbf{W}_J \mathrm{d}\boldsymbol{P}_J, \tag{7}$$

where $\mathrm{d}\boldsymbol{P}_T \in \mathbb{R}^{4n}$ is the tangential force measure (the friction) between the ground and the wheels and $\mathrm{d}\boldsymbol{P}_J \in \mathbb{R}^{2(n-1)}$ is the force measure due to the bilateral constraints in the joints. The matrices $\mathbf{W}_T \in \mathbb{R}^{4n\times 4n}$, $\mathbf{W}_J \in \mathbb{R}^{4n\times 2(n-1)}$ map the friction and bilateral constraint forces onto the non-minimal coordinates (2). In the following we give a short overview on how to find these matrices together with the force measures $\mathrm{d}\boldsymbol{P}_T$ and $\mathrm{d}\boldsymbol{P}_J$.

1) Friction Forces: In this section we describe the term $\mathbf{W}_T \mathrm{d}\boldsymbol{P}_T$ in (7) which accounts for the friction forces between the wheels and the ground surface. In order to do this, we need to start by finding the relative velocities between the wheels and the ground surface.

Let $\gamma_{T_{Lxi}}$ and $\gamma_{T_{Lyi}}$ be the tangential relative velocities between the left wheel on link i and the ground surface along the $\boldsymbol{e}_x^I$- and $\boldsymbol{e}_y^I$-axis, respectively. Let C_{Li} be the contact point between the left wheel and the ground surface. Let ${}_I\boldsymbol{v}_{C_{Li}} := {}_I\dot{\boldsymbol{r}}_{C_{Li}}$. Now we find the relative velocity between the ground surface and the left wheel as

$$\boldsymbol{\gamma}_{T_{Li}} = \begin{bmatrix} \gamma_{T_{Lxi}} \\ \gamma_{T_{Lyi}} \end{bmatrix} = {}_I\boldsymbol{v}_{C_{Li}} + \mathbf{R}^I_{B_i} \begin{bmatrix} -\omega_{w_i} L_{w_i} \\ 0 \end{bmatrix}. \tag{8}$$

Since (8) is linear in $\boldsymbol{u}_i$ we can write it as $\boldsymbol{\gamma}_{T_{Li}} = \mathbf{W}_{T_{Li}}{}^{\mathrm{T}} \boldsymbol{u}_i$ where $\mathbf{W}_{T_{Li}} \in \mathbb{R}^{4\times 2}$ is a function of θ_i and can easily be found by employing ${}_I\boldsymbol{v}_{C_{Li}} = {}_I\boldsymbol{v}_{G_i} + \dot{\mathbf{R}}^I_{B_i}\, {}_{B_i}\boldsymbol{r}_{G_i C_{Li}}$. Similarly, $\boldsymbol{\gamma}_{T_{Ri}}$ and $\mathbf{W}_{T_{Ri}}$ (for the right wheel) are found by simply replacing subscript 'Li' with 'Ri' in (8).

We gather all the tangential relative velocities as $\boldsymbol{\gamma}_T = \begin{bmatrix} \boldsymbol{\gamma}_{T_{L1}}^{\mathrm{T}} & \cdots & \boldsymbol{\gamma}_{T_{Ln}}^{\mathrm{T}} & \boldsymbol{\gamma}_{T_{R1}}^{\mathrm{T}} & \cdots & \boldsymbol{\gamma}_{T_{Rn}}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$. Now $\mathbf{W}_T \in \mathbb{R}^{4n\times 4n}$ in (7) can be composed of $\mathbf{W}_{T_{Li}}$ and $\mathbf{W}_{T_{Ri}}$ $(i = 1, \ldots, n)$ such that

$$\boldsymbol{\gamma}_T = \mathbf{W}_T{}^{\mathrm{T}} \boldsymbol{u}. \tag{9}$$

The friction forces between the wheels and the ground surface, represented by $\mathrm{d}\boldsymbol{P}_T$, are modeled as Coulomb friction and found by employing a *set-valued* force law which relates the force measure to the tangential relative velocities $\boldsymbol{\gamma}_{T_{Li}}, \boldsymbol{\gamma}_{T_{Ri}}$. The force law ensures a true stick-slip nature and allows for non-zero friction forces in stick-phase. This is not possible with, e.g., regular sign-functions commonly used for describing Coulomb friction since such function will result in a friction force equal to zero in stick-phase (i.e. when sliding velocities are zero). See [7] for a detailed description of the force law.

2) Bilateral Constraint Forces: Each joint introduces two bilateral constraints in the model. These constraints keep the "gap" in the joint between adjacent links equal to zero. An expression for this "gap", a gap function, needs to be found in order to calculate relative velocities and, in turn, find the bilateral constraint forces.

To find the gap function, we need to relate the position of joint i between link $i-1$ and i to both adjacent links. By inspecting Fig. 1, we see that the position of joint i going via link $i-1$ and i is given by ${}_I\boldsymbol{r}_{J_{Ri-1}} = {}_I\boldsymbol{r}_{G_{i-1}} - \frac{1}{2} L_{i-1}\, {}_I\boldsymbol{e}_x^{B_{i-1}}$ and ${}_I\boldsymbol{r}_{J_{Fi}} = {}_I\boldsymbol{r}_{G_i} + \frac{1}{2} L_i\, {}_I\boldsymbol{e}_x^{B_i}$, respectively. Now the gap functions for the gaps in the joints are

$$g_{J_{\chi i}} = \left({}_I\boldsymbol{e}_\chi^I\right)^{\mathrm{T}} \left({}_I\boldsymbol{r}_{J_{Fi}} - {}_I\boldsymbol{r}_{J_{Ri-1}}\right), \tag{10}$$

for $\chi = x, y$ and $i = 2, \ldots, n$.

The corresponding relative velocities for joint i along the $\boldsymbol{e}_x^I$- and $\boldsymbol{e}_y^I$-axis are defined as $\gamma_{J_{xi}} := \dot{g}_{J_{xi}}$ and $\gamma_{J_{yi}} := \dot{g}_{J_{yi}}$. Hence,

$$\gamma_{J_{\chi i}} = \boldsymbol{w}_{J_{\chi i}}^{\mathrm{T}} \begin{bmatrix} \boldsymbol{u}_{i-1} \\ \boldsymbol{u}_i \end{bmatrix}, \tag{11}$$

for $\chi = x, y$ and $i = 2, \ldots, n$, where $\boldsymbol{w}_{J_{xi}}^{\mathrm{T}} = \left[\left(-{}_I\boldsymbol{e}_x^I\right)^{\mathrm{T}}, \frac{-L_{i-1}}{2}\sin(\theta_{i-1}), 0, \left({}_I\boldsymbol{e}_x^I\right)^{\mathrm{T}}, \frac{-L_i}{2}\sin(\theta_i), 0\right]$, $\boldsymbol{w}_{J_{yi}}^{\mathrm{T}} = \left[\left(-{}_I\boldsymbol{e}_y^I\right)^{\mathrm{T}}, \frac{L_{i-1}}{2}\cos(\theta_{i-1}), 0, \left({}_I\boldsymbol{e}_y^I\right)^{\mathrm{T}}, \frac{L_i}{2}\cos(\theta_i), 0\right]$.

By defining $\boldsymbol{\gamma}_{J_i} = \begin{bmatrix} \gamma_{J_{xi}} & \gamma_{J_{yi}} \end{bmatrix}^{\mathrm{T}}$, we gather all the relative joint velocities as $\boldsymbol{\gamma}_J = \begin{bmatrix} \boldsymbol{\gamma}_{J_2}^{\mathrm{T}} & \cdots & \boldsymbol{\gamma}_{J_n}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$. Now $\mathbf{W}_J^{\mathrm{T}} \in \mathbb{R}^{4n\times 2(n-1)}$ introduced in (7) can be composed of $\mathbf{W}_{J_i} = \begin{bmatrix} \boldsymbol{w}_{J_{xi}} & \boldsymbol{w}_{J_{yi}} \end{bmatrix} \in \mathbb{R}^{6\times 2}$ $(i = 2, \ldots, n)$ such that

$$\boldsymbol{\gamma}_J = \mathbf{W}_J^{\mathrm{T}} \boldsymbol{u}. \tag{12}$$

The force measure $\mathrm{d}\boldsymbol{P}_J$ is obtained directly from an algebraic equation found from a numerical approximation of the equality of measures given in (3). See [7] for a detailed description of the procedure.

The mathematical model presented in this section is very suitable for numerical treatment and simulation. However, a representation with *minimal* coordinates is more appropriate for analytical considerations. To this end we develop a kinematic model with a minimal coordinate representation in the following section and use this to develop a path-following controller for snake robots with active wheels.

III. Path-Following using Frenet Frames

For path-following purposes we need to be able to measure the deviation of the robot from the path. The controller will regulate this deviation toward zero. In the path-following scheme we propose in this paper, we direct the robot toward the path by adjusting its heading.

In this section we define Frenet frames to obtain the deviation of the head of the robot with respects to the path. Then we propose a heading controller to eliminate this deviation, resulting in that the robot follows the path.

A. Schematic and Nomenclature

The notation presented in Section II also applies for the remainder of this paper. Hence, θ_i is the orientation of link i with respect to the $\boldsymbol{e}_x^I$-axis. In addition, let

$$
\begin{aligned}
P_i &: \text{center point of the wheel shaft of module } i,\\
(x_{P_i}, y_{P_i}) &: \text{coordinates of } P_i \text{ in the } \left({}_I\boldsymbol{e}_x^I, {}_I\boldsymbol{e}_y^I\right)\text{-plane},\\
v_{P_i} &: \text{speed of } P_i,\\
J_i &: \text{location of the front end of module } i,\\
v_{J_i} &: \text{speed of } J_i,\\
\delta_i &: \text{direction of motion of } J_i,\\
L_{PJ} &: \text{length of segment } P_iJ_i,\\
L_{JP} &: \text{length of segment } P_iJ_{i+1},\\
\phi_i &= \theta_i - \theta_{i-1}.
\end{aligned}
$$

Note that δ_1 is the direction of motion for the head, which is used as the turning angle for the robot. Moreover, δ_1 will be used as the control-variable for path-following. See Fig. 1 for more details.

B. Frenet Frames

We employ Frenet frames to find the deviation in position and heading of the robot head with respect to a desired path σ. To this end, a Frenet frame is assigned to every point along the path. Each Frenet frame has a tangent axis pointing tangent to and in the direction of the path, and a normal axis, anti-clockwise perpendicular to the tangent axis. A specific Frenet frame on the path can be specified by the arc length parameter s_σ (see Fig. 2), given by the curves natural parametrization.

When using Frenet frames for path-following, we need to choose the Frenet frame at the point given by s_{σ_1}, which is the point on the path closest to the robot head. In such a reference frame, the position and orientation of a module can be described by only two parameters. The offset distance z_1 describes the distance from the path to the robot head, while the offset angle $\tilde{\theta}_1 = \theta_1 - \theta_{\sigma 1}$ gives a measure of heading error. If $z_1 = 0$ and $\tilde{\theta}_1 = 0$, the head is following the path perfectly. If additionally, $z_i = 0$ and $\theta_i = 0$ for every subsequent robot module, $i \in \{2 \dots n\}$, then the robot is following the curve perfectly. Unfortunately, this is physically impossible for a snake robot on a general path. Therefore we will focus on following the path with the head module, and let the subsequent modules follow in a passive-like manner.

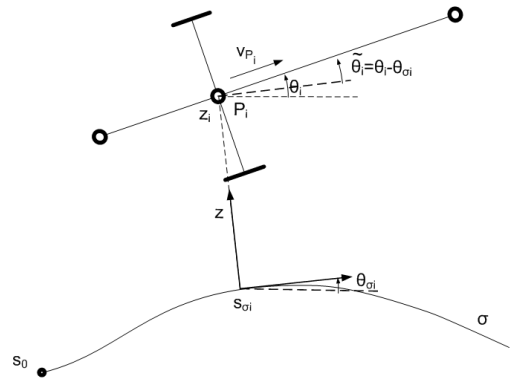

Fig. 2. Utilizing Frenet frames for path-following.

The Frenet variables $s_{\sigma 1}$, z_1 and $\tilde{\theta}_1$ define the position and orientation of the head of the robot with respect to the path. We have opted to numerically approximate these variables for the simulations presented in Section VI.

C. Heading Controller

Let δ_f be a turning angle which will give perfect path-following if the robot is on the curve. We choose a slightly modified PI-controller of the form

$$\delta_1 = \delta_f - K(z_c + v_{P_1} T \sin\tilde{\theta}_1) \tag{13}$$

where

$$z_c = \begin{cases} \hat{z} & \text{if } |\hat{z}| \leq v_{P_1} T \\ -v_{P_1} T & \text{if } \hat{z} < -v_{P_1} T \\ v_{P_1} T & \text{if } \hat{z} > v_{P_1} T, \end{cases} \tag{14}$$

and

$$\hat{z} = z_1 + K_i \int z_1 \, dt. \tag{15}$$

See Fig. 1 for an illustration of δ_1. The position offset between the head of the robot and the path is given by z_1. This is found simply by using the distance along the z-axis of the corresponding Frenet frame, as seen in Fig. 2. In addition, $v_{P_1} T \sin\tilde{\theta}_1$ is an estimate of the change in offset from the path caused by the orientation offset $\tilde{\theta}_1$ after a time T. Tuning T enables us to determine the relative importance of both of these offsets, and thereby determining how aggressively we want the robot to close in on the path.

By saturating $\hat{z}$ as in (14) we limit the $\hat{z}$-term from becoming too overpowering when $|\hat{z}|$ is large, e.g., when z_1 is large and the robot is far from the curve. z_c is $\hat{z}$ clamped so as to prevent a breakdown of the path-following functionality when the robots head module is far from the curve, or the integral action is large. In such situations we want the robot moving perpendicular toward the path. The clamping function ensures this.

The feed-forward term δ_f is the turn angle that would give perfect path-following if the robot was following on the path in the ideal case. This term can be found by considering the motion of J_1 caused by the change of θ_1:

$$L_{PJ}\dot{\theta}_1 = v_{P_1} \tan\delta_f \tag{16}$$

$$\tan\delta_f = \frac{L_{PJ}\dot{\theta}_1}{v_{P_1}} = L_{PJ}\frac{d\theta_1}{ds} = \kappa_{\sigma_1} L_{PJ}, \tag{17}$$

where $v_{P_1} = ds/dt$, and $\kappa_{\sigma_1} = d\theta_1/ds$. As expected δ_f is dependent on the curvature of the path at point $s_{\sigma 1}$.

The control law presented gives us the desired turn angle δ_1 for path-following. However, how do we coordinate the snake robot joints so that the snake-robot turns in the desired direction? This is elaborated on in Sections IV and V, where two different strategies are presented.

IV. Module Coordination by n-trailer Method

Assume that the speed v_{P_1} and the turning angle δ_1 are given. Now, how do we achieve this desired motion with the inputs available for a snake robot with active wheels and joints? As the head module of the robot cannot turn on its own accord, it relies on the coordinated efforts of all the modules to achieve these goals. In the following subsections we present the n-trailer model and derive a strategy for coordinating the inputs of the snake-robot to achieve the desired motion.

A. Kinematic n-trailer Model

The n-trailer problem is the problem of steering a truck pulling n linked trailers behind it. Physically it is similar to a wheeled multi-joint robot (schematic in Fig. 1), but with some important differences. The n-trailer system only has one active module, the truck, while all the trailers are passive. In contrast, both the joints and the wheels of each module are active in PIKo.

The n-trailer model differs from the dynamic model presented in section II in two main points. First of all, the n-trailer model is purely kinematic, not including any reference to inertial tensors, forces or acceleration. Secondly, it introduces non-holonomic constraints on the robot wheels, as elaborated on below. The n-trailer model therefore provides a simplified system more suited for analysis. Further, the non-holonomic constraints on the wheels free us from having to take into account friction forces when designing a strategy for coordinating the robot modules.

Although we will use the exact same kinematic model as in the n-trailer system, it will be utilized very differently. In this paper we will only be giving a brief summary of the model derivation and refer to [12] and [13] for more detailed reading.

The velocities for a link $i = 1, \ldots, n$, are

$$\dot{x}_{P_i} = v_{P_i} \cos\theta_i, \quad \dot{y}_{P_i} = v_{P_i} \sin\theta_i \tag{18}$$

$$\dot{\theta}_i = \frac{v_{P_i}}{L_{PJ}} \tan\delta_i = \frac{v_{J_i}}{L_{PJ}} \sin\delta_i. \tag{19}$$

In addition, the motion of each module is constrained by rigid-body constraints

$$x_{P_{i+1}} = x_{P_i} - L_{JP}\cos\theta_i - L_{PJ}\cos\theta_{i+1} \tag{20}$$

$$y_{P_{i+1}} = y_{P_i} - L_{JP}\sin\theta_i - L_{PJ}\sin\theta_{i+1}, \tag{21}$$

and a non-holonomic constraint

$$\dot{x}_{P_i}\sin\theta_i - \dot{y}_{P_i}\cos\theta_i = 0, \tag{22}$$

which we get by placing a condition of no lateral slip on the wheels.

Using (18) through (22) and the simple relation $\phi_i = \theta_i - \theta_{i-1}$ for the joint angles, we can derive an equation for the angular velocities of the joint angles

$$\dot{\phi}_i = -\frac{v_{P_{i-1}}}{L_{PJ}} \left\{ \sin\phi_i + \left(\frac{L_{JP}}{L_{PJ}} \cos\phi_i + 1 \right) \tan\delta_{i-1} \right\}, \tag{23}$$

for $i = 2, \ldots, n$. We define a state vector

$$\mathbf{x} = \begin{bmatrix} x_{P_1} & y_{P_1} & \theta_1 & \phi_2 & \ldots & \phi_n \end{bmatrix}^T \tag{24}$$

which defines the position and orientation of the robot in an inertial coordinate frame. Equations (18)-(19) and (23) now describe the motion of the robot. Notice that solving these equations requires the knowledge of the variables δ_i and v_{P_i}, as well as the state vector (24), which we assume is known. The following relations can be derived from Fig. 1:

$$\delta_{i+1} = \beta_i - \phi_{i+1} \tag{25}$$

$$v_{P_{i+1}} = v_{P_i} \frac{\cos\delta_{i+1}}{\cos\beta_i}, \tag{26}$$

where β_i is given as follows

$$\tan\beta_i = -\frac{L_{JP}}{L_{PJ}} \tan\delta_i. \tag{27}$$

Now (25)-(27) allow us to iteratively calculate δ_i and v_{P_i} as long as δ_1 and v_{P_1} are known. Hence the kinematics of the system can be completely defined by only two variables, δ_1 and v_{P_1}. This is an important result that we will make use of in the next section. Note that it holds only as a direct result of the no-slip condition we imposed upon the wheels in (22).

According to [12] the singularities of the general n-trailer system occur at $\phi_i = \pm\frac{\pi}{2}$ and $\delta_i = \pm\frac{\pi}{2}$. For PIKo and most wheeled snake robots the physical limitations constrain the relative orientation angle within $-\frac{\pi}{2} < \phi_i < \frac{\pi}{2}$. If additionally $|\delta_1| < \frac{\pi}{2}$ is guaranteed, the system will remain non-singular. In effect this puts a limit on the minimum turning radius of the robot.

B. Module Coordination Strategy

Before we proceed we wish to again emphasize the difference between the n-trailer model and the snake robot PIKo.

1) ***n*-trailer:** Only the first module, the truck, is active. The speed v_{P_1} and steering angle δ_1 of the truck can be set. The remaining modules, the trailers, passively follow the truck so as to satisfy the rigid-body and non-holonomic constraints (20)-(22).
2) **PIKo:** All modules are active. The speed v_{P_i} and the joint angles ϕ_i can be set. Even though the steering angle δ_1 is defined for the first module, it cannot steer on its own accord, as it just like all the other modules does not have an additional wheel axle for steering (contrary to the truck of the n-trailer system).

The problem is then, how can we make the head of PIKo turn and head in the desired direction without active steering in the head module?

Assume temporarily that we can consider the head of PIKo as a virtual truck, with a virtual steering wheel. This allows us to set its speed v_{P_1} and its virtual steering angle δ_1, as an operator or a controller desires. The result from the previous section shows that given v_{P_1} and δ_1, the kinematics of the whole robot is defined. Consequently, we can use the kinematic equations (23) and (25)-(27) found in the previous

section to solve for the individual module speeds v_{P_i} and joint angles ϕ_i.

Now an assumption of symmetry is made. We assume that if we set the individual module speeds and joint angles according to the calculated values of v_{P_i} and ϕ_i, PIKo will move as if it actually was being pulled by the virtual truck. Instead of the steering being the result of a steerable axle in the truck, it will emerge as a result of the collective actions of all modules. As a bonus, since the trailers of the n-trailer system behave in a passive manner, our control scheme should result in an equivalent "passive" behavior in PIKo. This means that in the ideal case, the aforementioned slip of the wheels will be eliminated.

The presented strategy therefore consists of two parts. First we use the n-trailer equations of the previous section to calculate v_i and ϕ_i, given desired speed v_{P_1} and turning angle δ_1 (possibly given by the heading controller of Section III-C). Then we achieve the desired motion by applying the calculated inputs to the robot.

V. Module Coordination by Follow-the-Leader

An intuitive and often employed alternative to the module coordination approach presented in Section IV is a "follow-the-leader" (FTL) approach. FTL approaches have already been used in several designs for motion through bends [1],[6]. We therefore consider it an important benchmark against the proposed algorithm. It utilizes a very basic principle for control: All modules should repeat the pattern of the first module – the leader – at the exact same spatial position as the leader module. This can either be done by adjusting the speed of the wheels to match the curvature shift [1], or by measuring the distance traveled by the wheels to set correct curvatures [6]. In either case slip will introduce inaccurate movements. Formally, this latter approach to FTL may be stated as

$$\phi_{r_i}(d = d + L_i) = \phi_{r_{i-1}}(d), \qquad \text{for } i = 3, \ldots, n, \quad (28)$$

where ϕ_{r_i} is the reference angle of joint i and d is the curve traced by the longitudinal direction of the robot links.

VI. Simulation Results and Discussion

In this this section we present simulation results for path-following using the n-trailer approach presented in Section IV. In addition, we compare the results to simulations of an implementation of the "follow-the-leader" (FTL) algorithm elaborated on in Section V. The dynamic model presented in Section II are employed for the simulations.

A. Simulation Parameters and Implementation Description

The model parameters used for simulation are based on the snake robot PIKo which is developed at the Norwegian research institution SINTEF in Trondheim, Norway [6]. We employ $n = 6$ links in the simulations and for each link i we have that $L_i = 0.122$ m, $L_{w_i} = 0.065$ m, ${}_I\boldsymbol{r}_{G_iC_{Ri}} = [0.018 \ -0.041]^{\mathrm{T}}$ m, ${}_I\boldsymbol{r}_{G_iC_{Li}} = [0.018 \, 0.041]^{\mathrm{T}}$ m, $L_{PJ} = 0.079$ m, $L_{JP} = 0.043$, $m_i = 1.2$ kg, $\Theta_i = \Theta_{w_i} = 0.002$ kg m^2. The joint and wheel shaft torques are controlled by PD-controllers. The heading controller parameters are $K = 1.5$, $K_i = 0.04$, $T = 2.0$ s. Simulations are performed with a time step of $\Delta t = 2.5 \times 10^{-4}$ s. However, the head controller only calculates new joint angle and wheels velocity references every 0.025 s in order to make the simulations more similar to a control system on an actual snake robot. The reference forward speed is $v_{P_1} = 0.5$ m/s. The numerical algorithm used for simulation is called the time-stepping method (see, e.g., [11], [7]).

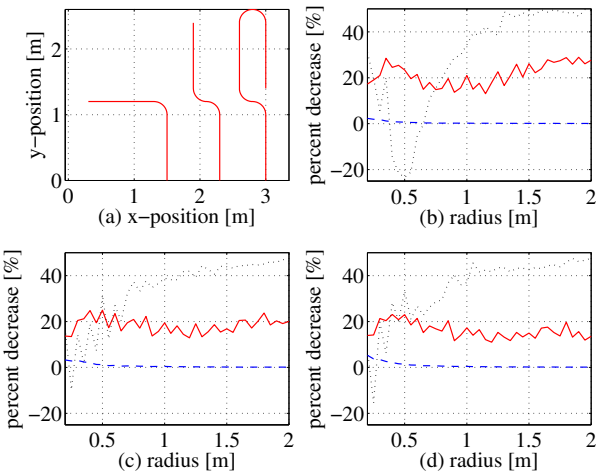

Fig. 3. Simulation results for three different paths with various curvatures. (a) Shape of paths (illustrated with turning radius of 0.2 m). (b)-(d) Decrease in total commanded torque (red solid line), decrease in total path-following error (black dotted line), and decrease in total distance covered (blue dashed line) by employing the n-trailer approach compared to the FTL approach. Figures (b)-(d) correspond to the left-most path, the middle path, and the right-most path illustrated in (a), respectively, and simulations were run for turning radii from 0.2 to 2 m for each path.

B. Simulation Results and Discussion

In this section simulation results with both the n-trailer and the FTL approach are presented. The results are discussed and compared with respect to issues such as path-following error and total commanded torque. A range of different paths with increasing complexity are simulated and the results are discussed.

Fig. 3 (a) depicts the shape of three different types of paths used for simulations of path-following with both the n-trailer and the FTL approach. Simulations were carried out for radius of curvature $r_k = 0.2, 0.25, \ldots, 2.0$ m (this gives a total of 37 simulations for each combination of path type and path-following approach). Fig. 3 (b)-(d) show the decrease in total path-following error, decrease in total commanded torque (for both wheels and joints), and decrease in total distance covered gained by employing the n-trailer approach compared to the FTL approach for each curve radius. The total path-following error was calculated as the sum of distances from the desired path for each link during each simulation. We see from Fig. 3 (b)-(d) that there is an average decrease of ca. 20 % in total commanded torque gained by employing the n-trailer approach compared to the

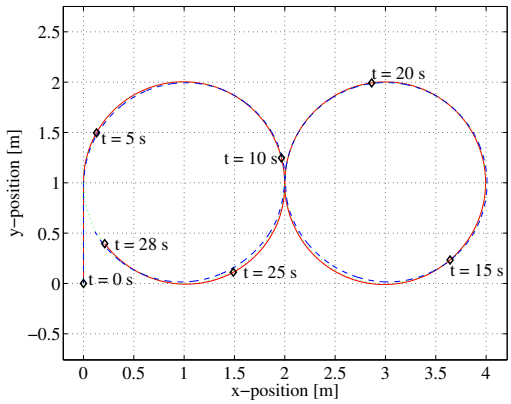

Fig. 4. Position of center of gravity of link 1 in the $(\boldsymbol{e}_x^I, \boldsymbol{e}_y^I)$-plane during path-following with the n-trailer (red solid line) and the FTL (blue dashed line) approach. The green dotted line marks the path to be followed.

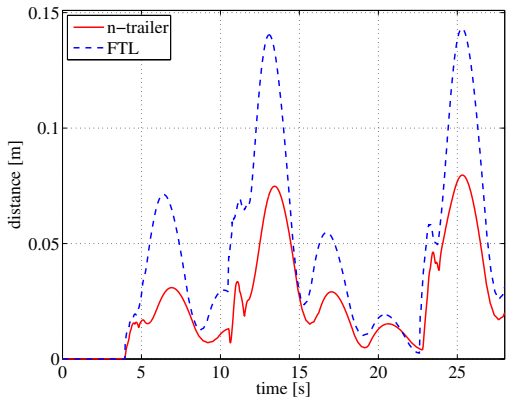

Fig. 5. Sum of absolute value of distance from each link to path during path-following of 8-shaped curve with n-trailer (red solid line) and FTL (blue dashed line) approach.

FTL approach. Moreover, this added benefit seems almost independent of path curvature. This is an important issue since snake robots will need to operate untethered in order to minimize the chance of getting stuck. Hence, it is important to keep the energy consumption low in order to maximize the possible period of operation. For the simple one-turn path depicted to the left in Fig. 3 (a), we see that the path-following error is decreased by employing the n-trailer approach for paths with radius of curvature less than 0.35 and larger than 0.65 m. Reduction in path-following error is also obtained when employing the n-trailer approach for the more complex paths in Fig. 3 (a) except when the radius of curvature is equal to 0.25 m. For larger radii of curvature (e.g., $r_k > 0.75$ m) we see that there is a 40-50 % decrease in path-following error gained by employing the n-trailer approach. Except for very small radii of curvature the total distance covered with the two appraoches was almost identical. The head position (i.e., link 1) was used as a reference for finding the total distance covered.

We see from Fig. 3 that the n-trailer approach yields in general a better result for path-following than FTL. However, for some radii FTL results in more accurate path-following. This is also the case for simulations of motion along a 5 m straight line. However, the maximum deviation from the path was only ca. 35 μm larger for the n-trailer approach compared to FTL. Hence, this difference is negligible and would probably not have been detected in a live experiment. For comparison we note that for the simulations shown in Fig. 3 the maximum deviation from the path was up to 2.5 cm more for FTL than the n-trailer approach.

Fig. 4 depicts the position of the head (link 1) of the snake robot while following an 8-shaped curve. Both the n-trailer and the FTL approach are employed respectively. We see from the figure that the snake robot is able to follow the curve better when the n-trailer approach is employed compared to the FTL approach. Since Fig. 4 only depicts the position of the first link of the snake robot in a xy-plot, we also include Fig. 5 which depicts the sum of the absolute value of the path-following error for all links. Hence, the plot shows the sum of the absolute values of the distances (i.e. the Frenet parameters) z_i between link i and the path for $i = 1, \ldots, n$. We see from Fig. 5 that the total distance between the snake robot links and the path is in general smaller when employing the n-trailer approach compared to FTL.

Table I presents the reduction in total path-following error, total distance covered, and total commanded torque gained by employing the n-trailer approach presented in this paper compared to the FTL approach during path-following of the 8-shaped curve. The total path-following is found and compared simply by calculating the area between the time-axis and the path-following error values in Fig. 5. The total distance covered is found by calculating the distance covered during 28 seconds of link 1. The total commanded torque is found in a similar manner by calculating the total amount of commanded torque used in the joints and on the wheel shafts for guiding the robot through a path.

The illustrations of the simulation results together with Table I show that both the total commanded torque and the total path-following error are reduced, sometimes significantly, when the n-trailer approach is employed for path-following of a snake robot with active wheels compared to when FTL is employed. Moreover, we see that there is a slight reduction in distance covered by link 1 when the n-trailer approach is used. However, this percentage is much smaller than those which describe the reduction in path-following error and commanded torque.

Note that the heading controller presented in Section III-C is the same for both the n-trailer and the FTL approach.

TABLE I

REDUCTION IN PATH-FOLLOWING ERROR, DISTANCE COVERED AND COMMANDED TORQUE FOR 8-SHAPED CURVE.

Quantity	Reduction
Path-following error	44.7 %
Distance covered	0.4 %
Commanded torque	19.3 %

Hence, there could be a possibility of that a different heading control strategy or different parameters for the heading controller could prove to be advantageous for FTL (or, for that matter, the n-trailer approach). However, a large number of heading controller parameteres have been tested and it was found that the best performance for both the FTL and the n-trailer approach was obtained with an identical set of parameters. Hence, there has not been a focus on tuning the heading controller parameters in order to obtain a better performance of the n-trailer approach compared to the FTL approach. Moreover, the heading controller is designed as a stand-alone module and not specifically for the n-trailer approach. The authors acknowledge that there could exist an improved implementation of the FTL approach. However, FTL implementations have mainly been described with text and not equations (see, e.g., [1]). Therefore, we have chosen the FTL implementation described in (28) to be used for comparing with to the n-trailer approach described in this paper and, thus, leave the task of finding an optimal FTL implementation as a subject for further research.

From a practical point of view it is indeed expected that an FTL approach will require more commanded torque than the n-trailer approach. This is because all consecutive links try to follow the exact path traced by the robot head which requires a significant joint control effort. This differs from the result of the n-trailer approach where the consecutive links follow the head in a passive-like manner much like the same way as a trailer on a car follows the car. What is perhaps more surprising from a practical point of view is that the simulations suggest that the path-following error is in general lower for the n-trailer approach than the FTL approach. However, the simulations show that the n-trailer approach gives an improvement of more than 40 % with respect to path-following error for larger radii of curvatures.

C. Discussion regarding extension to 3-D

This paper deals with planar motion even though locomotion in a real environment will be an inherently 3-D experience. Nevertheless, many "3-D environments" consist of several close-to planar surfaces such as floors or the wall of a pipe. Hence, the techniques presented in this paper can be used for locomotion in such environments and extensions to these algorithms can be developed in order to move a snake robot up or down between surfaces. A simple strategy for such an extension could be to use FTL for vertical motion while extending the n-trailer approach to account for vertical module motion during path-following. The non-smooth model for simulation presented in this paper can be extended to describe 3-D motion. This can be achieved by, e.g., employing techniques developed for a simulation model of a 3-D snake robot without wheels in [14].

VII. Conclusions and Future Work

In this paper we have presented a dynamic model of a snake robot with active wheels together with a novel path-following approach. Moreover, simulations have shown that a 6-link snake robot with active wheels is indeed able to follow a pre-defined path using the n-trailer path-following approach. In addition, this approach has been compared with the commonly used "follow-the-leader" approach with favorable results.

Simulations suggest that the n-trailer approach possess, in most cases, better path-following abilities achieved with, in every case tested, the use of a reduced amount of total commanded torque on the wheel shafts and the joints of approximately 20 %. This is favorable as it decreases both wear and tear of the robot and the amount of energy needed to control the robot along its path. These improvements are accredited the cooperative nature in which the modules achieve the control goals. Low energy consumption is vital for snake robots in order to be able to operate cordlessly on prolonged missions.

Future work will consist of implementing the developed path-following algorithm on the snake robot PIKo developed at SINTEF ICT Applied Cybernetics. First the method will be tested with equipment available to measure the position and orientation of each link quite accurately. Then the path-following algorithm will be extended to account for odometry errors and measurement noise which arise in live experiments.

References

[1] H. Yamada and S. Hirose, "Development of practical 3-dimensional active cord mechanism ACM-R4," *Journal of Robotics and Mechatronics*, vol. 18, no. 3, pp. 1–7, 2006.

[2] K.-U. Scholl, V. Kepplin, K. Berns, and R. Dillman, "An articulated service robot for autonomous sewer inspection tasks," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 1999, pp. 1075–1080.

[3] W. Ilg, K. Berns, S. Cordes, M. Eberl, and R. Dillmann, "A wheeled multijoint robot for autonomous sewer inspection," in *Proc. IEEE/RSJ Int.l Conf. on Intelligent Robots and Systems*, 1997, pp. 1687–1692.

[4] M. Kolesnik and H. Streich, "Visual orientation and motion control of makro: adaptation to the sewer environment," in *Proc. 7th Int. Conf. on Simulation of Adaptive Behavior on from animals to animats*, 2002, pp. 62–69.

[5] K.-U. Scholl, V. Kepplin, K. Berns, and R. Dillmann, "Controlling a multijoint robot for autonomous sewer inspection," in *Proc. IEEE Int. Conf. on Robotics and Automation*, April 2000, pp. 1701–1706.

[6] S. A. Fjerdingen, A. A. Transeth, and P. Liljebäck, "A snake-like robot for internal inspection of complex pipe structures (PIKo)," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, October 2009.

[7] A. A. Transeth, R. I. Leine, Ch. 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.

[8] J. J. Moreau, "Unilateral contact and dry friction in finite freedom dynamics," in *Non-Smooth Mechanics and Applications*, ser. CISM Courses and Lectures, J. J. Moreau and P. D. Panagiotopoulos, Eds. Wien: Springer Verlag, 1988, vol. 302, pp. 1–82.

[9] B. Brogliato, *Nonsmooth Mechanics*, 2nd ed. London: Springer, 1999.

[10] Ch. Glocker, *Set-Valued Force Laws, Dynamics of Non-Smooth Systems*, ser. Lecture Notes in Applied Mechanics. Berlin: Springer-Verlag, 2001, vol. 1.

[11] R. I. Leine and H. Nijmeijer, *Dynamics and Bifurcations of Non-Smooth Mechanical Systems*, ser. Lecture Notes in Applied and Computational Mechanics. Berlin: Springer Verlag, 2004, vol. 18.

[12] C. Altafini, "Some properties of the general n-trailer," *International Journal of Control*, vol. Vol.74:4, pp. 409–424, 2001.

[13] P. Bolzern, R. M. DeSantis, A. Locatelli, and S. Togno, "Dynamic model of a two-trailer articulated vehicle subject to nonholonomic constraints," *Robotica*, vol. Vol.14, pp. 445–450, 1996.

[14] A. A. Transeth, R. I. Leine, Ch. Glocker, and K. Y. Pettersen, "3-D snake robot motion: Non-smooth modeling, simulations, and experiments," *IEEE Trans. on Robotics*, vol. 24, no. 2, pp. 361–376, April 2008.