# 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 for applications such as inspection and maintenance. A slender body combined with flexible movement makes it ideal for pipes and other narrow or constricted structures. In this paper we present a novel path-following algorithm for snake robots with active wheels. The algorithm is composed of two steps. First, the desired turning angle for the snake robot head is found based on its location with respect to a desired path. Then we employ a $n$-trailer kinematic model to find an energy efficient set of control inputs for the robot given a reference path for the snake robot head. Moreover, we extend a non-smooth model of a snake robot to include active wheels. The proposed snake robot system is simulated with the non-smooth model to test the viability of the control concept, and it is further compared to another common method used for similar robots.**

## I. Introduction

Search and rescue operations in confined and narrow spaces or in the remains of collapsed and unstable buildings, and inspection and maintenance of pipe structures in sewage and ventilation systems, can be unreachable and/or dangerous to humans. Our need to explore such areas have resulted in numerous robotic systems.

The snake-like structure is especially advantageous for moving through narrow spaces and tight bends while still carrying the payload required to fulfill it's task. This is why locomotion of snake-like robots is a predominant research area when it comes to automated inspection of pipe structures and some search and rescue applications.

Wheeled snake-like robots have the advantage of the shape and build of a snake, while maintaining the direct motion of vehicles, thereby combining the best of both worlds. Multiple active wheels allow the robot to continue forward propulsion also 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 steplike structures. This kind of complicated movements also require the use of active joints, eigher to lift certain parts of the robot off the ground, or to push the wheels into the internal walls of a vertical pipe, and thereby creating the friction necessary for propelling itself vertically.

Snake-like robots with active wheels and active joints are presented in [1], [2], [3] and [4]. Scholl et al. present an autonomous sewer inspection robot [3], and provide experimental results in [5] on the same robot demonstrating the the ability of wheeled, multijoint robots to maneuver in tight bends and traverse steplike obstacles. Hirose et al. present similar successes in maneuverabliity with the robot ACM-R4 [1]. Advances have also been made in using visual sensing in pipe structures, to adapt the heading and path of sewage inspection robots, and thus avoiding collisions and impacts with walls.

Many of these robots have to work in pipe structures that are distributed over large areas, making it impractical to have it connected by cable to some central power supply. It is therefore crucial that the robot uses its actuators as effective as possible. In a trailer system, any "unnatural" motion will dissipate energy. By "unnatural" we mean any motion of the wheels that is not strictly rolling, e.g. spinning of the wheels or any lateral movement of the wheels perpendicular to the rolling direction. It may therefore be natural to use the amount of slip on the wheels as a benchmark for energy consumption.

There exists a need to design a path-following algorithm that coordinates the movement of each module so as to minimize slip on the wheels, resulting in a smooth and flowing motion. This is not always accomplished by the common algorithms used in similar robots. Several academic attempts are based on the "follow-the-leader" principle, where it is assumed that all the modules follow the exact same path. The path would then be defined by the trace of the head of the robot. The subsequent modules imitate the actions of the head when they reach that point on the trajectory. This approach is implemented in [1] and [4]. This simple method has been effective in initial laboratory testing, but might not be the most energy efficient method, for reasons which we will discuss.

In [6], which also uses a "follow-the-leader" strategy, clothoids (as described in [7]) are used to construct trajectories, such that the wheel axes always are close to perpendicular to the tangent of the trajectory. This puts limitations on the possible trajectories or paths, e.g. that the curvature of the trajectory as a function of the curvelength along the trajectory must be continuous. And even then it cannot guarantee motion without lateral wheel slippage.

In this paper we present a path-following algorithm that coordinates the actuators on all modules in such a way that lateral and longitudinal wheel slippage and hence the energy consumption is reduced. To this end we develop a path-following algorithm based on frenet-frames so that the head

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 is with the Department of Applied Cybernetics, SINTEF ICT, NO-7465 Trondheim, Norway. aksel.a.transeth@sintef.no

S. A. Fjerdingen is with the Department of Applied Cybernetics, SINTEF ICT, NO-7465 Trondheim, Norway. sigurd.fjerdingen@sintef.no

of the snake robot can track an arbitrarly path. Then we use a kinematic model from a $n$-trailer problem to calculate all the control variables needed to follow a given path. Moreover, we extend a non-smooth model of a wheel-less snake robot presented in [8] to include active wheels. This model is then employed for simulation of the proposed path-following scheme in order to compare it to the previously published "follow-the-leader" approach.

This paper is organized as follows. In Section II. we present a kinematic model for a general multijoint/articulated vehicle. Then in Section III. we describe how the aforementioned kinematic model can be used to steer a snake-like robot with multiple wheels. We go on to use the results of Section III. to design a path-following algorithm in Section IV.

## II. Kinematic model

This section presents a kinematic model which will serve as a basis for the proposed path-following algorithm presented in Section III. The model is based on the snake robot Piko which is currently being developed at the Norwegian research organization SINTEF in Trondheim, Norway.

### A. Schematic and nomenclature

$P_i =$ point on the center of the wheel axes of module $i$

$x_i =$ Cartesian x-coordinate of $P_i$ in global frame

$y_i =$ Cartesian y-coordinate of $P_i$ in global frame

$\theta_i =$ orientation of module $i$ with respect to an inertial frame

$v_i =$ speed of $P_i$

$Q_i =$ point defining the location of the front end of module $i$

$u_i =$ speed of point $Q_i$

$\delta_i =$ direction of motion for point $Q_i$

$\beta_i =$ $\phi_i + \delta_{i+1}$

$L =$ length of segment $P_iQ_i$

$M =$ length of segment $P_iQ_{i+1}$

$\phi_i =$ relative angle between modules $i$ and $i+1$, $\theta_{i+1} - \theta_i$

This notation may vary slightly from the dominant $n$-trailer literature.

### B. Model derivation

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

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 [9] and [10] for more detailed reading.

The equations of motion for a given module $i = 1, \ldots, n$, are given by

$$\dot{x}_i = v_i \cos\theta_i \tag{1}$$

$$\dot{y}_i = v_i \sin\theta_i \tag{2}$$

$$\dot{\theta}_i = \frac{v_i}{L}\tan\delta_i = \frac{u_i}{L}\sin\delta_i. \tag{3}$$

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

$$x_{i+1} = x_i - M\cos\theta_i - L\cos\theta_{i+1} \tag{4}$$

$$x_{i+1} = x_i - M\cos\theta_i - L\cos\theta_{i+1}, \tag{5}$$

and a nonholonomic constraint

$$\dot{x}_i \sin\theta_i - \dot{y}_i \cos\theta_i = 0, \tag{6}$$

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

Using (1) through (6) and the simple relation $\phi_i = \theta_{i+1} - \theta_i$ for the joint angles, we can derive an equation for the angular velocities of the joint angles

$$\dot{\phi}_i = -\frac{v_i}{L^2}\{L\sin\phi_i + (M\cos\phi_i + L)\tan\delta_i\}. \tag{7}$$

for $i = 1, \ldots, n-1$. We define a state vector

$$\mathbf{x} = \begin{bmatrix} x_1 & y_1 & \theta_1 & \phi_1 & \ldots & \phi_{n-1} \end{bmatrix}^T \tag{8}$$

which defines the robot's position and orientation in an inertial coordinate frame. (1)-(3) and (7) are the kinematic equations that describe the motion of the robot. Notice that solving these equations requires the knowledge of the variables $\delta_i$ and $v_i$, as well as the state vector, which we assume is known. The following relations can be derived from Fig. 1.

$$\delta_{i+1} = \beta_i - \phi_i \tag{9}$$

$$v_{i+1} = v_i \frac{\cos\delta_{i+1}}{\cos\beta_i} \tag{10}$$

where $\beta_i$ is given as follows

$$\tan\beta_i = -\frac{M}{L}\tan\delta_i. \tag{11}$$

Equations (9)-(11) allow us to iteratively calculate $\delta_i$ and $v_i$ as long as $\delta_1$ and $v_1$ are known. Hence the kinematics of the system can be completely defined by only two variables, $\delta_1$ and $v_1$. 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 nonslip condition we imposed upon the wheels.

## III. Controlling robot heading

In the previous section we saw how to set the various joint angles and link speeds based on the two variables $\delta_1$ and $v_1$. In this section we will see how to determine these to variables.

At this point we wish to again emphasize the difference between the $n$-trailer model and the snake-like robot PiKo.

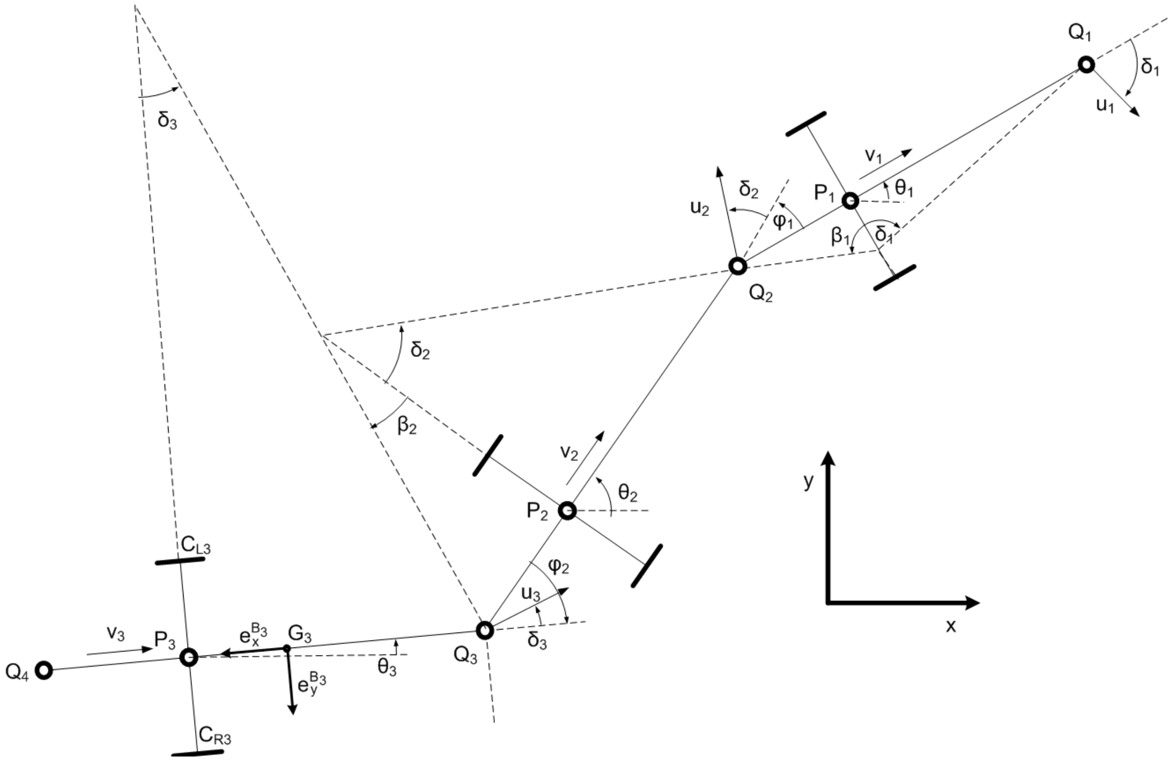

Fig. 1. Simple schematic of a general multijoint robot/vehicle

1) ***n*-trailer:** Only the first module, the truck, is active. The trucks speed $v_1$ and steering angle $\delta_1$ can be set. The following modules, the trailers, passively follow the truck so as to satisfy the rigid-body and nonholonomic constraints.
2) **PiKo:** All modules are active. The speed $v_i$ and the joint angles $\phi_i$ can be set. Even though the steering angle $\delta_1$ 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 $v_1$ and its virtual steering angle $\delta_1$, as an operator or a controller desires. The result from the previous section shows that given $v_1$ and $\delta_1$, the kinematics of the whole robot is defined. Consequently, we can use the kinematic equations found in the previous section to solve for the individual module speeds $v_i$ and joint angles $\phi_i$.

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 $v_i$ and $\phi_i$, 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.

## IV. Frenet frames and path-following

For path-following purposes we need to be able to measure the robot's deviation from the path. The controller will regulate this deviation towards zero. In the path-following scheme we propose in this paper, the steering angle $\delta_1$ will be used to steer the robots head towards the path. We will be assuming that the subsequent modules stay within an acceptable distance of the path. Frenet frames will be used to obtain the deviation of the head to the path.

### A. *Frenet frames*

The Frenet frame, as we will use it is a coordinate system defined on every point of a path $\gamma$. It has a tangent axis pointing tangent to and in the direction of the path, and a normal axis, perpendicular to the tangent axis pointing in the direction of positive curvature. A specific Frenet frame on the path can be specified by the arc length parameter $s_\gamma$. Fig. 2 illustrates this.

When using Frenet frames for path-tracking, we need to choose the Frenet frame at the point given by $s_{\gamma i}$, which is the point on the path closest to module $i$. In such a reference frame the position and orientation of a module can be described by only two parameters. The offset distance $z_i$ describes the distance from the path to module $i$, while the offset angle $\tilde{\theta}_i = \theta_i - \theta_{\gamma i}$ gives a measure of the error in

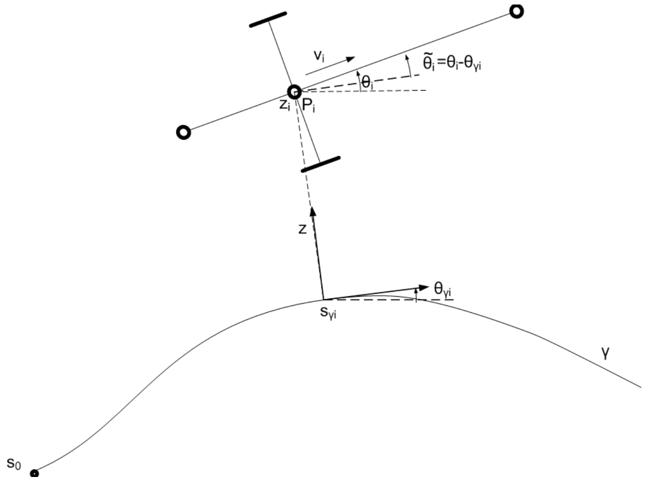

Fig. 2. Utilizing Frenet frames for path-tracking

orientation. If we have $z_i$ and $\theta_i$ equal zero for every module, then we are tracking the curve perfectly. Unfortunately, this is physically impossible for a general path (when the non-holonomic constraints are enforced). Therefore we will focus on tracking the path with the head module, and let the subsequent modules follow in a passive-like manner.

The Frenet parameters $s_{\gamma i}$, $z_i$ and $\tilde{\theta}_i$ define module $i$'s position and orientation with respects to the path. They can either be approximated numerically by optimization as we have done for the simulations, or be estimated using a kinematic model defined with respect to the changing Frenet frame. Such a model is presented in [11]. This model can also be used to design more effective nonlinear controllers, but this will not be necessary for the purposes of this discussion.

### B. A simple controller

We choose a controller of the simple form

$$\delta_1 = \delta_f - K(z_c + v_1 T \sin\tilde{\theta}_1) \tag{12}$$

where

$$z_c = \begin{cases} z_1 & \text{if } |z_1| \leq v_1 T \\ -v_1 T & \text{if } z_1 < -v_1 T \\ v_1 T & \text{if } z_1 > v_1 T. \end{cases} \tag{13}$$

Using (13) we limit the $z$-term from becoming too overpowering when $|z_1|$ is large. $z_c$ is $z_1$ clamped so as to not cause a breakdown of the path-following functionality when the robots head module is far from the curve. $z_1$ is the current offset from the path while $v_1 T \sin\tilde{\theta}_1$ 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 agressively we want the robot to close in on the path.

The feedforward term $\delta_f$ is the steering angle that would give perfect tracking if the robot was following on the path in the ideal case. This term can be calculated from the differential equation for the orientation angle $\theta_1$, (3), restated as follows:

$$\dot{\theta}_1 = \frac{v_1}{L}\tan\delta_f \tag{14}$$

$$\tan\delta_f = \frac{L\dot{\theta}_1}{v_1} = L\frac{d\theta_1/dt}{ds/dt} = L\frac{d\theta_1}{ds} \tag{15}$$

$$\tan\delta_f = L\kappa_{\gamma 1}. \tag{16}$$

As expected the $\delta_f$ is dependent on the curvature of the path at point, $s_{\gamma 1}$.

## V. Mathematical Model for Simulation

A mathematical model that includes the dynamics of a snake robot with active wheels is important in order to validate the proposed path-following algorithm. The model will also be used to compare the path-following algorithm proposed in this paper with a "follow-the-leader" approach previously published for snake robots. The model of the dynamics extends a non-smooth model of a wheel-less snake robot presented in Transeth et al. [8] in that *active wheels* are added.

A complete description of a non-smooth model of a snake robot with active wheels is too comprehensive to be included in this paper. Therefore we refer the reader to [8] for a thorough description of a non-smooth model of a snake robot *without* wheels. Moreover in this paper we describe the necessary changes needed to *introduce active wheels* to the model in [8].

### A. Notation

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

### B. How to Include Active Wheels

In order to add active wheels to the non-smooth snake robot model, we first need to find the relative tangential velocity between the ground and the two wheels on each link ("tangential" with respect to ground surface and not the link). This velocity is later employed in conjunction to finding the friction forces between the wheels and the ground surface.

Let $\boldsymbol{r}_{C_{Qi}} = \boldsymbol{r}_{G_i} + \boldsymbol{r}_{G_i C_{Qi}}$ denote the position of the center of the wheel on the right ($C_{Qi} = C_{Ri}$) and left ($C_{Qi} = C_{Li}$) side of link $i$ (see link 3 in Fig. 1). Let ${}_I\boldsymbol{v}_{C_{Qi}} := \frac{d}{dt}\,{}_I\boldsymbol{r}_{C_{Qi}}$. Now the relative sliding velocity between the ground and the part of the wheel in contact with the ground is

$$\boldsymbol{\gamma}'_{T C_{Qi}} = {}_I\boldsymbol{v}_{C_{Qi}} + \mathbf{R}_{B_i}^I \begin{bmatrix} -\omega_{w_i} L_{w_i} \\ 0 \end{bmatrix} \tag{17}$$

for $Qi = Li, Ri$ where $\omega_{w_i}$ and $L_{w_i}$ are the rotational speed and radius of the two identical wheels on link $i$, respectively. Note that the rotational velcotiy $\omega_{w_i}$ is controlled directly in the model for simplicity. $\|\boldsymbol{\gamma}'_{T C_{Qi}}\| \neq 0$ indicates that a wheel on link $i$ is slipping.

In order to adapt to the form in [8] we define $\boldsymbol{\gamma}'_{T_{CQi}} = \mathbf{R}^I_{B_i}\,\boldsymbol{\gamma}_{T_{CQi}}$ and find that

$$\boldsymbol{\gamma}_{T_{CQi}} = \mathbf{W}_{T_{CQi}}{}^{\mathrm{T}}\boldsymbol{u}_i + \tilde{\mathbf{W}}_{T_{CQi}} \tag{18}$$

where

$$\mathbf{W}_{T_{CQi}}{}^{\mathrm{T}} = \begin{bmatrix} \mathbf{I}_{2\times 2} \\ (\mathbf{R}^I_{B_i}\mathbf{D}_{B_i}\boldsymbol{r}_{G_iC_{Qi}})^{\mathrm{T}} \end{bmatrix} \mathbf{R}^I_{B_i} \tag{19}$$

$$\mathbf{D} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix},\ \tilde{\mathbf{W}}_{T_{CQi}} = \begin{bmatrix} -\omega_{w_i}L_{w_i} \\ 0 \end{bmatrix} \tag{20}$$

The tangential relative velocities $\boldsymbol{\gamma}_{T_{CQi}}$ for $i = 1, \ldots, n\ Qi = Ri, Li$ found from 18 can now be replaced by the existing tangential relative velocties in the model in [8] to obtain a complete non-smooth model of a snake robot with active wheels.

## VI. Simulation results

In this section we first present the "follow-the-leader" approach which the path-following algorithm presented in this paper is compared against. Then the simulation parameters are given. Finally the simulation results are presented.

### A. Simlation parameters

The model parameteres are based on the snake robot PiKo which is being developed at the Norwegian research institution SINTEF in Trondheim, Norway. For each link $i$ we have that $M_i = 0.045$ m, $L_i = 0.13$ m, $L_{w_i} = 0.065$ m, ${}_I\boldsymbol{r}_{G_iC_{Ri}} = [0.043\ 0.4]^{\mathrm{T}}$ m, ${}_I\boldsymbol{r}_{G_iC_{Li}} = [0.043\ -0.4]^{\mathrm{T}}$ m. Each link weighs 1.2 kg.

### B. Comparison of Path-following Schemes

In this section we run simulations with the dynamical model of PiKo where the snake robot is set to follow a path. Fig. 3 depicts the position of the centre of gravity of link 1 in PiKo with the previously published "follow-the-leader" path-following algorithm and the path-following algorithm proposed in this paper. The rotational velocties of all wheels are assummed to be equal. Both approaches consists of two similar sub-problems. First the first link needs to follow the path correctly. Then the other links need to follow the first link in a suitable manner. Comparing the energy usage from the two algorithms, we get a 99% energy usage of the proposed algorithm compared to "follow-the-leader". Total energy usage is computed by summing the absolute values of the friction forces acting on each link of the robot. The result is closer to "follow-the-leader" than expected. However, there is an indication of an energy advantage for the links involved.

We see from Fig. 3 that the proposed algorithm tracks the ideal curve significantly better than the "follow-the-leader" algorithm; the dotted curve is almost entirely overlapping the ideal solid curve. This result is promising for being able to use the path-tracking algorithm in combination with the $n$-trailer following algorithm to gain a better, smoother path-following.

## VII. Conclusions and Future Work

We have developed a path-following algorithm for a snake robot with active wheels in this paper. Moreover, we have extended a non-smooth model of a snake robot to include active wheels which are directly controlled by setting reference rotational velocities. In addition, we have shown by simulation that less friction forces between the ground surface and the snake robot arise compared to a "follow-the-leader" path-following approach. Also, the approach presented in this paper track a path better.

Future work will consist of implementing the path-following algorithm on the snake robot PiKo when it is finished. 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 must 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] W. Ilg, K. Berns, S. Cordes, M. Eberl, and R. Dillmann, "A wheeled multijoint robot for autonomous sewer inspection," in *Proc. IEEE International Conference on Intelligent Robots and Systems*, 1997, pp. 1687–1692.

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

[4] M. Kolesnik and H. Streich, "Visual orientation and motion control of makro: adaptation to the sewer environment," in *Proceedings of the seventh international conference 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 *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, April 2000, pp. 1701–1706.

[6] K. Paap, F. Kirchner, and B. Klaassen, "Motion control scheme for a snake-like robot," in *Proc. IEEE Int. Symp. Computational Intelligence in Robotics and Automation*, 1999, pp. 59–63.

[7] B. Klaassen and K. Paap, "Gmd-snake2: a snake-like robot driven by wheels and a method for motion control," in *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 4, 1999, pp. 3014–3019.

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

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

[10] 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.

[11] P. Bolzern, R. DeSantis, and A. Locatelli, "An input-output linearization approach to the control of an n-body articulated vehicle," *Journal of Dynamic Systems, Measurement, and Control*, vol. Vol. 123, pp. 309–316, September 2001.

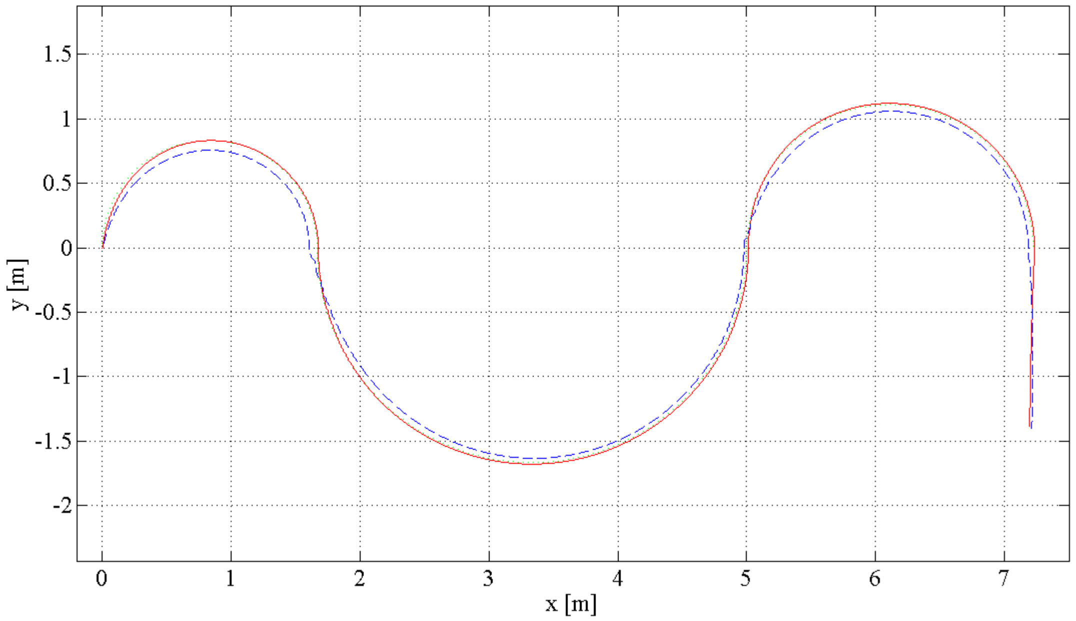

Fig. 3. XY plot of tracking of the proposed algorithm (dotted) and follow-the-leader (dashed) compared to the ideal curve (solid).