Passive joint control of a snake robot by rolling motion

doi: 10.1007/s10015-020-00643-1

Noname manuscript No.
(will be inserted by the editor)

Passive Joint Control of a Snake Robot by Rolling Motion

Ryo Ariizumi · Kentaro Koshio · Motoyasu Tanaka · Fumitoshi Matsuno

Received: date / Accepted: date

Abstract Snake robots are capable of adapting to difficult situations, such as cluttered environments, using its many degrees of freedom. However, if one of the joints gets passive, it is generally very difficult to achieve ordinary performance. In this paper, control of a passive joint using rolling motion is considered, with the use of crawler gait in mind. Crawler gait is a state-of-the-art motion pattern for snake robots that is capable of moving on uneven terrain, but if there is a passive joint, the motion can be interrupted by freely moving part of the robot itself. As a key to solving this difficulty, this paper proposes to use the rolling motion, which has not been used in controlling a passive joint. A simplified model is proposed to consider the control and based on this, one simple controller is adopted. The validity of the idea of using rolling motion is tested by numerical simulations.

This work was supported by the ImPACT Program of the Council for Science, Technology and Innovation (Cabinet Office, Government of Japan).

R. Ariizumi
Department of Mechanical System Engineering, Graduate School of Engineering, Nagoya University, Nagoya 464-8603, Japan
Tel.: +81-52-789-2746
Fax: +81-52-789-2746
E-mail: ryo.ariizumi@mae.nagoya-u.ac.jp

K. Koshio and F. Matsuno
Department of Mechanical Engineering and Science, Graduate School of Engineering, Kyoto University, Kyoto 606-8501, Japan

M. Tanaka
Department of Mechanical and Intelligent Systems Engineering, Graduate School of Information Science and Engineering, The University of Electro-Communications, Tokyo 182-8585, Japan

Keywords Actuator malfunction · Snake robot · Passive joint

1 Introduction

Snake robots, which are composed of many serially connected actuators, are expected to be useful in a variety of difficult situations. They can move inside a pipe [1], can climb stairs or a ladder [2,3], and can run over uneven terrain [3–6], using its many degrees of freedom (DOFs). With this rich ability to negotiate with difficult environments, researchers and practitioners are now believe that a snake robot can be used in many tasks such as pipe inspection, urban search and rescue, and reconnaissance tasks. Therefore, many studies have been performed to design new snake robots, analyze its movements, and propose useful motions.

Another merit of having many DOFs, along with the ability to move in many different environments, is that the robot can be robust to actuator failures. However, there are only a few research on this point. When considering a planar snake robot and if lateral constraints are assumed, which is an often-used assumption for a planar snake robot, the existence of a passive joint makes no problem in many cases. Because in such cases, if there are enough number of working joints, then the movement of the passive joint is determined kinematically [7]. However, there are many occasions where the assumption of lateral constraints is not realistic.

Mehta et al. [8] considered the movement of snake robots when there exists a free joint or a locked joint, but they only tried predefined movements and did not propose any control to cancel the effect of a malfunctioning joint. Their study showed that the sidewinding locomotion, which is one of the most frequently

used 3-dimensional locomotion in snake robots, is robust to an actuator failure. However, restricting the movements to predefined ones such as the sidewinding locomotion severely limits the ability of the snake robots. In [9], they proposed a head trajectory tracking control method that can perform well even if there is a passive joint. However, the controller is only for a planar snake robot and cannot be used in 3-dimensional movements.

In this paper, we consider the control of a passive joint in a snake robot using a rolling movement. Especially, the case where the robot is moving by a crawler gait is assumed, though most of the results are applicable to any 3-dimensional locomotion. The crawler gait enables a robot to move in all directions flexibly and is especially suitable for moving on uneven terrain. In this gait, a part of a snake robot should be lifted and form a cantilever. If the passive joint is in this cantilever part, it will move downward because of the force of gravity, and this part can interrupt the locomotion. The purpose of the control is to prevent such undesired movements using the rolling motion. It is a novel idea to use the rolling movement to control a passive joint rather than to use it to design locomotion. To design the controller, we make a simplified model that concentrates on the expression of the cantilever part. A controller is constructed based on the linearization at the desired state and the validity of the controller is tested by numerical simulations.

This paper is organized as follows: In Section 2, we describe the snake robot that is considered in this paper and explain the problem to be solved. In Section 3, a simplified model is proposed and, based on the model, two controllers are given in Section 4: one is based on the linear approximation model and the other is based on the nonlinear model predictive control (NMPC) technique. Section 5 shows some simulation results and Section 6 concludes the paper.

This paper extends our previous work [10], in which we consider only one controller that depends on the linear approximation of the simplified model. In this paper, we additionally discuss the use of NMPC and show superiority over the other via simulation.

2 Problem Formulation

2.1 Snake robots

The structure of the snake robot that we consider in this paper is shown in Fig. 1. The robot has yaw and pitch joints connected alternately. Many real snake robots employ this structure because of its simplicity and its potential flexibility. Figure 2 shows one such snake robot.

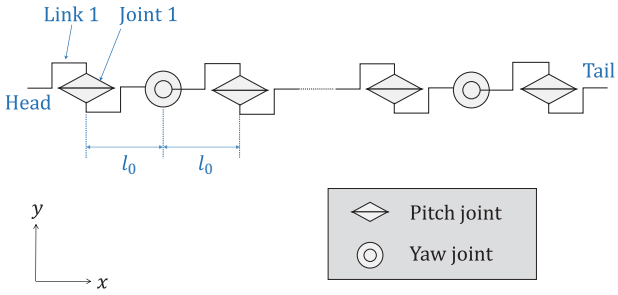


Fig. 1 Structure of the snake robot

Fig. 2 A snake robot

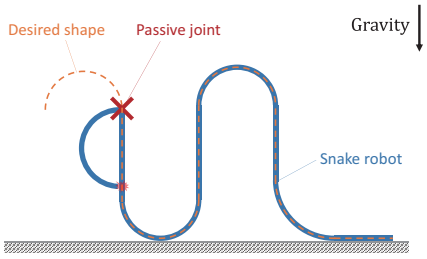
By this structure, it is well known that the rolling motion can be synthesized even without any roll joint [11].

The posture of the robot is described by a backbone curve [12], which expresses the reference shape of the robot and the reference orientation of joints. The joint angles are determined to approximately realize the postures defined by the backbone curve. In this paper, we mainly consider the backbone curve, rather than the snake robot itself to simplify the discussion.

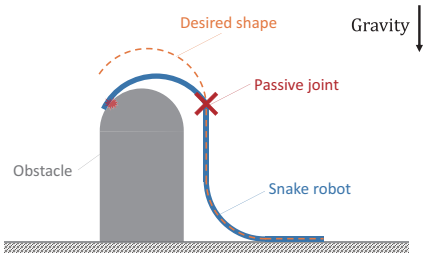
2.2 Problem description

The target situation is illustrated in Fig. 3. The robot has a passive joint shown by a cross mark and the joint is in the middle of a lifted part at the front part. In such cases, because of the gravitational force, the passive joint will move for the head part to fall, which will result in a collision with the robot itself or the environments. However, if the axis of the passive joint is vertical and if the robot is static, the gravity will not produce any torque to rotate the passive joint and the head part will not fall. Therefore, it is expected that, by controlling the axis direction, it is possible to control the movement of the passive joint. Note that, a joint can become passive amid of performing some task because of malfunctioning: this is the case that we have in mind and it is desired that even in such a case, the robot can continue moving.

Let the angle of the passive joint be $\bar{\theta}$, the unit vector in the direction of the axis of the passive joint be $\boldsymbol{e}_p$ and rotation angle of $\boldsymbol{e}_p$ around the snake body be



(a) Contact with its own body



(b) Collision with an obstacle

Fig. 3 Problems of the passive joint

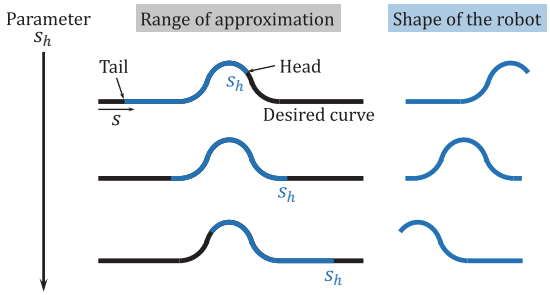


Fig. 4 Shift control of the snake robot

ψ_0. The origin of ψ_0 can be taken arbitrarily at the first step of the modeling. Also, assume that the snake robot moves on a flat surface using the crawler gait. Note that the changes in ψ_0 does not alter the body shape of the snake robot but only change its location in sideway.

In many cases, snake robots move using the shape shift, i.e., the shape of the head part will propagate to the rear part (Fig. 4) [13]. Let a parameter to show the progress of this shape shift be $s_h(t)$; the formal definition will be given in Section 3.1.

The problem to be considered in this paper is expressed as follows:

Problem *Assume that the reference shape is defined using a time-varying parameter $s_h(t)$. Find a control law $\ddot{\psi}_0 = u(\bar{\theta}, \dot{\bar{\theta}}, \psi_0, \dot{\psi}_0, s_h)$ that stabilizes $\bar{\theta}$ to the reference trajectory $\bar{\theta}^d(t, s_h)$.*

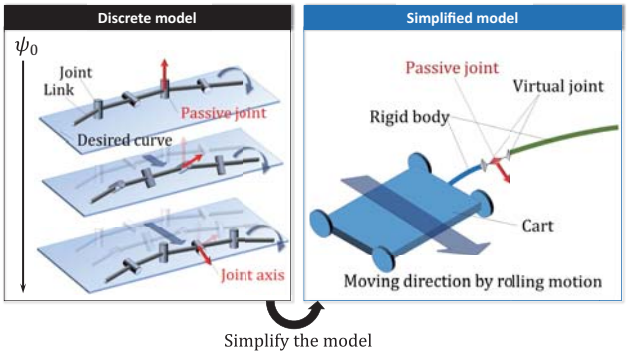


Fig. 5 Simplification of the model

In what follows, we use the following angle $\theta(t, s_h)$ instead of $\bar{\theta}$:

$$\theta(t, s_h) = \bar{\theta} - \bar{\theta}^d(t, s_h). \tag{1}$$

This makes the reference of θ be 0 for any t and s_h.

3 Simplified Model

3.1 Idea of Simplification

To solve the problem, we construct a simplified model. First, note that the movements of the parts other than the cantilever part have only a little importance: the sideways motion may affect θ via acceleration, but other effects can be negligible. Therefore, let this part of the body expressed as a cart that can move only in the sideways direction of the robot as shown in Figs. 5, 6. The robot will move in sideways by the rolling and this movement is expressed as a movement of the cart as in Fig. 5. Second, we only consider the passive joint and the virtual joints that control ψ_0, explicitly: other joints are not explicitly considered but are assumed to be controlled to realize the predefined reference shape of the crawler gait. Note that there are 2 virtual joints to have the robot's shape keep the reference shape: one is on the grounded part side of the passive joint and the other on the head side of the passive joint, and let them be denoted as joint p_1 and p_2 respectively. They move in the opposite direction with the same amount.

Because the shape shift is adopted, the entire reference trajectory for the robot can be specified by that of the head. Let the arc-length coordinate along the head trajectory be denoted as s. The head position parameter s_h is the reference position of the head expressed in this arc-length coordinate (Fig. 4). In this study, $s_h(t)$ is considered to be given as an exogenous input and for

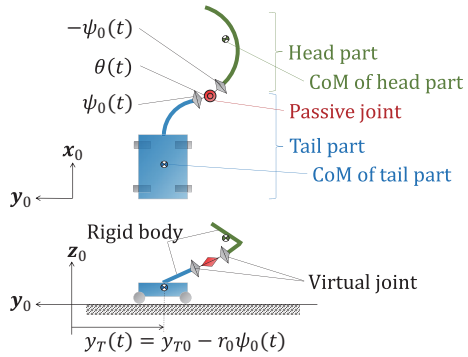


Fig. 6 Simplified model of the snake robot

simplicity do not take the dynamics related to s_h into consideration, i.e., we assume that the change of s_h is quasi-static.

3.2 Equation of Motion

Hereafter, the part from the head to the passive joint is called the *head* part and the rest is called the *tail* part. To calculate the Lagrangian, we first calculate the velocities of each part.

Let the homogeneous transformation matrix that relates coordinates Σ_i and Σ_j be denoted as ${}^iT_j$. Then, the position of the center of mass (COM) ${}^0\boldsymbol{p}_H$ of the head part expressed in Σ_0 satisfies

$$\begin{bmatrix} {}^0\boldsymbol{p}_H \\ 1 \end{bmatrix} = {}^0T_T\,{}^TT_1\,{}^1T_2\,{}^2T_3\,{}^3T_4\,{}^4T_H\boldsymbol{e}_4, \tag{2}$$

where $\boldsymbol{e}_4 = [0,\ 0,\ 0,\ 1]^T$ and ${}^0\boldsymbol{p}_H = [{}^0x_H,\ {}^0y_H,\ {}^0z_H]^T$. Similarly, the COM position ${}^0\boldsymbol{p}_T$ of the tail part expressed in Σ_0 satisfies

$$\begin{bmatrix} {}^0\boldsymbol{p}_T \\ 1 \end{bmatrix} = {}^0T_T\boldsymbol{e}_4. \tag{3}$$

The time-derivative of ${}^0\boldsymbol{p}_H$ is too complex to write down but can be calculated by a computer algebra system like Mathematica or Maple. Because of the assumption of the cart approximation, time-derivative of ${}^0\boldsymbol{p}_T$ can easily be calculated to give

$${}^0\dot{\boldsymbol{p}}_T = [r_0\dot{\psi}_0(t),\ 0,\ 0]^T. \tag{4}$$

Let 3 vectors ${}^2\boldsymbol{\omega}_{H,\psi_0}$, ${}^3\boldsymbol{\omega}_{H,\theta}$, ${}^4\boldsymbol{\omega}_{H,-\psi_0}$ be defined as

$${}^2\boldsymbol{\omega}_{H,\psi_0} = [\dot{\psi}_0(t),\ 0,\ 0]^T, \tag{5}$$
$${}^3\boldsymbol{\omega}_{H,\theta} = [0,\ 0,\ \dot{\theta}(t)]^T, \tag{6}$$
$${}^4\boldsymbol{\omega}_{H,-\psi_0} = [-\dot{\psi}_0(t),\ 0,\ 0]^T. \tag{7}$$

Then the angular velocity vector of the head part expressed in Σ_H is

$$\begin{aligned} {}^H\boldsymbol{\omega}_H &= {}^H\boldsymbol{\omega}_{H,\psi_0} + {}^H\boldsymbol{\omega}_{H,\theta} + {}^H\boldsymbol{\omega}_{H,-\psi_0} \\ &= {}^HR_4\,{}^4R_3\,{}^3R_2\,{}^2\boldsymbol{\omega}_{H,\psi_0} \\ &\quad + {}^HR_4\,{}^4R_3\,{}^3\boldsymbol{\omega}_{H,\theta} + {}^HR_4\,{}^4\boldsymbol{\omega}_{H,-\psi_0} \\ &= \begin{bmatrix} (\cos\theta(t)-1)\dot{\psi}_0(t) \\ -\cos\psi_0(t)\sin\theta(t)\dot{\psi}_0(t) - \sin\psi_0(t)\dot{\theta}(t) \\ -\sin\psi_0(t)\sin\theta(t)\dot{\psi}_0(t) - \cos\psi_0(t)\dot{\theta}(t) \end{bmatrix}, \end{aligned} \tag{8}$$

where ${}^iR_j$ is the rotation matrix that relates coordinates Σ_i and Σ_j.

The Lagrangian L is defined as

$$L = K_H + K_T - V_H - V_T, \tag{9}$$

where

$$K_H = \frac{1}{2}m_H\|{}^0\dot{\boldsymbol{p}}_H\|^2 + \frac{1}{2}{}^H\boldsymbol{\omega}_H^T\,{}^HI_H\,{}^H\boldsymbol{\omega}_H, \tag{10}$$
$$K_T = \frac{1}{2}m_T\|{}^0\dot{\boldsymbol{p}}_T\|^2, \tag{11}$$
$$V_H = m_H[0,\ 0,\ g]\,{}^0\boldsymbol{p}_H, \tag{12}$$
$$V_T = m_T[0,\ 0,\ g]\,{}^0\boldsymbol{p}_T, \tag{13}$$

and m_H and m_T are masses of the head part and the tail part, respectively. The inertia tensor ${}^HI_H$ of the head around its COM is determined by a shape and the posture of the head part. Assume that the reference shape and posture are realized exactly. Because these references are completely determined from the reference head trajectory, ${}^HI_H$ is a function of $s_h(t)$.

The derivation of the equation of motion can be performed using a computer algebra system. Although the detail is too lengthy to show here (it will consume a few pages), the equation of motion has the following structure:

$$m(\theta,\psi_0,s_h)\ddot{\theta} + h(\theta,\psi_0,\dot{\theta},\dot{\psi}_0,s_h) + p(\theta,\psi_0,s_h)\ddot{\psi} = 0. \tag{14}$$

Note that, in the above equation, we have omitted the dependencies on time t for simplicity. However, all θ, ψ_0, s_h, and their derivatives depend on t.

3.3 State-Space Representation

Let the state vector $\boldsymbol{x}$ be defined as

$$\boldsymbol{x} = [\theta,\ \psi_0,\ \dot{\theta},\ \dot{\psi}_0]^T, \tag{15}$$

then, the state equation of the system is given in the form of an input affine system:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})u, \tag{16}$$

$$f(\boldsymbol{x}) = \begin{bmatrix} x_3 \\ x_4 \\ a(\boldsymbol{x}) \\ 0 \end{bmatrix},\ g(\boldsymbol{x}) = \begin{bmatrix} 0 \\ 0 \\ b(\boldsymbol{x}) \\ 1 \end{bmatrix},$$

$$a(\boldsymbol{x}) = -\frac{h(\boldsymbol{x})}{m(\boldsymbol{x})},\ b(\boldsymbol{x}) = -\frac{p(\boldsymbol{x})}{m(\boldsymbol{x})},$$

where $u = \ddot{\psi}_0$.

The equilibrium point of (23) satisfies

$$h(\theta, \psi_0, \dot{\theta}, \dot{\psi}_0, s_h) = 0, \tag{17}$$

$$\dot{\theta} = 0, \tag{18}$$

$$\dot{\psi}_0 = 0, \tag{19}$$

$$u = 0. \tag{20}$$

Because of the definition of θ, for the reference to be the equilibrium, $\theta = 0$ must hold. Using these conditions, the reference of ψ_0 is derived to be

$$\psi_0(s_h) = -\mathrm{atan}\left(\frac{{}^T r_{31}^4 y_H - {}^T r_{32}^4 x_H}{{}^T r_{31}^4 z_H - {}^T r_{33}^4 x_H}\right) \tag{21}$$

To have the origin be equilibrium, the following transformation is applied:

$$\bar{\boldsymbol{x}} = [x_1(t),\ x_2(t) - x_{d,2}(s_h(t)),\ x_3(t),\ x_4(t)]^T. \tag{22}$$

Then, the state equation becomes

$$\dot{\bar{\boldsymbol{x}}} = \bar{\boldsymbol{f}}(\bar{\boldsymbol{x}}, s_h(t)) + \bar{\boldsymbol{g}}(\bar{\boldsymbol{x}}(t), s_h(t))u(t). \tag{23}$$

4 Linearization and Controller

As the simplified model is still a nonlinear one, it is not easy to design a controller. Another difficulty is that the system is underactuated: we want to drive all of the four states to the origin, but there is only a single input. One straightforward idea is to linearize the model around each operating point and construct a controller based on it.

4.1 Linearization around Operating Point

The linearized system of (23) can be written as

$$\dot{\bar{\boldsymbol{x}}} = \bar{A}(s_h(t))\bar{\boldsymbol{x}}(t) + \bar{B}(s_h(t))u(t), \tag{24}$$

where

$$\bar{A} = \frac{\partial \bar{f}}{\partial \bar{\boldsymbol{x}}}(\bar{\boldsymbol{x}}_d, s_h(t)) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \bar{a}_{31}(s_h(t)) & \bar{a}_{32}(s_h(t)) & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{25}$$

$$\bar{B} = \bar{\boldsymbol{g}}(\bar{\boldsymbol{x}}_d, s_h(t)) = \begin{bmatrix} 0 \\ 0 \\ \bar{b}_3(s_h(t)) \\ 1 \end{bmatrix}. \tag{26}$$

As this is a time-varying system, it is well-known that the eigenvalues of the system matrix cannot be linked with the stability of the system. Therefore, controller design based on pole-assignment cannot be applied. Furthermore, as the system is not necessarily periodic in the time interval to be considered, results on periodic systems are also not applicable. One possible choice for the control of such a system will be to employ robust control techniques. However, in this problem, we empirically confirmed that H_∞ control will not give satisfactory results, because of the too large variation of the system. In this paper, we consider the use of the nonstationary linear quadratic control (NSLQC) instead.

4.2 Nonstationary Linear Quadratic Control

Suppose that $s_h(t)$ is given explicitly as a function of t, which is satisfied, e.g., if we employ the crawler gait. Then, the system (24), which is a linear parameter-varying system, can be seen as a linear time-variant system:

$$\dot{\bar{\boldsymbol{x}}} = A(t)\bar{\boldsymbol{x}}(t) + B(t)u(t). \tag{27}$$

For the linear time-varying system, we consider to use NSLQC. In other words, we consider the following optimal control problem:

$$\begin{aligned} &\underset{u}{\text{minimize}}\ J[u] \\ &\text{subject to (27)}, \end{aligned} \tag{28}$$

where the cost J is the functional of input u defined as follows:

$$\begin{aligned} J[u] =& \bar{\boldsymbol{x}}^T(t_f) S_f \bar{\boldsymbol{x}}(t_f) \\ &+ \int_{t_0}^{t_f} \{\bar{\boldsymbol{x}}^T(t) Q(t) \bar{\boldsymbol{x}}(t) + r(t)u^2(t)\} dt. \end{aligned} \tag{29}$$

The weight matrix $Q(t)$ is a positive semi-definite matrix, S_f is a positive definite one, and the weight $r(t)$ is

a positive scalar. In this study, we set $r(t) = 1,\ \forall\, t \geq 0$. The optimal control is derived as

$$u(t) = -K(t)\bar{\boldsymbol{x}}(t),\ K(t) = B^T(t)S(t), \tag{30}$$

where $S(t)$ is the solution of the Riccati differential equation:

$$\begin{aligned}\dot{S}(t) =& - S(t)A(t) - A^T(t)S(t) \\ &+ S(t)B(t)B^T(t)S(t) - Q(t),\end{aligned} \tag{31}$$

$$S(t_f) = S_f. \tag{32}$$

A sufficient condition for the above control to be valid is the uniform controllability of the time-varying system [14]:

Definition 1 A linear time-varying system (27) is said to be uniformly controllable if for any bounded $\bar{\boldsymbol{x}}_0$, $\bar{\boldsymbol{x}}_f$, and any time instant t, there exists a $\Delta > 0$ and a control u defined for the time interval $[t-\Delta, t]$, such that u drives the system from $\bar{\boldsymbol{x}}(t-\Delta) = \bar{\boldsymbol{x}}_0$ to $\bar{\boldsymbol{x}}(t) = \bar{\boldsymbol{x}}_f$.

Note that the uniform controllability of our system depends on the body shape. Although it is difficult to check if the system (24) is uniformly controllable or not in general, we will discuss it numerically in Section 5 for the body shape of the crawler gait.

4.3 Nonlinear Model Predictive Control

Another possible choice of the controller than NSLQC based on the linearization around the operating point is to use NMPC. By this method, linearization is no more necessary and the nonlinear model (23) can be used directly.

Let the cost function J at time t with time-horizon T be defined as

$$J = \int_0^T [\tilde{\boldsymbol{x}}(\tau,t)^T Q(t+\tau)\tilde{\boldsymbol{x}}(\tau,t) + r(t)u^{*2}(\tau,t)]d\tau, \tag{33}$$

$$\tilde{\boldsymbol{x}}(\tau,t) = \boldsymbol{x}^*(\tau,t) - \boldsymbol{x}_d(s_h(t+\tau)). \tag{34}$$

In this method, at every time t, the following optimization problem is considered:

$$\underset{u^*}{\text{minimize}}\, J \tag{35}$$

subject to

$$\begin{aligned}\frac{d}{d\tau}\boldsymbol{x}^*(\tau,t) =& \boldsymbol{f}(\boldsymbol{x}^*(\tau,t), s_h(t+\tau)) \\ &+ \boldsymbol{g}(\boldsymbol{x}^*(\tau,t), s_h(t+\tau))u^*(\tau,t),\end{aligned} \tag{36}$$

$$\boldsymbol{x}^*(0,t) = \boldsymbol{x}(t). \tag{37}$$

The vector $\boldsymbol{x}^*(\tau,t)$ is the prediction of the state at time $t+\tau$, which is calculated using the model of the system. This effectively means that, at each time t, the controller solves the optimal control problem for the time interval $[t, t+T]$ based on the model. The solution of the optimization problem at $\tau = 0$ is adopted as the control input at time t, i.e., the actual input $u(t)$ is set to be $u(t) = u^*(0,t)$.

The solution of the above mentioned optimization problem is given as the solution of the following Euler-Lagrange equations:

$$\begin{aligned}&\frac{d}{d\tau}\boldsymbol{\lambda}^*(\tau,t) \\ &\quad = -\left(\frac{\partial H}{\partial \boldsymbol{x}}\right)^T (\boldsymbol{x}^*(\tau,t), u^*(\tau,t), \boldsymbol{\lambda}^*(\tau,t), s_h(t+\tau)),\end{aligned} \tag{38}$$

$$\boldsymbol{\lambda}^*(\tau,t) = \boldsymbol{0}, \tag{39}$$

$$\frac{\partial H}{\partial u}(\boldsymbol{x}^*(\tau,t), u^*(\tau,t), \boldsymbol{\lambda}^*(\tau,t), s_h(t+\tau)) = 0 \tag{40}$$

with (36) and (37). Here, $\boldsymbol{\lambda}^*(\tau,t)$ is the adjoint variable vector and H is the Hamiltonian:

$$H = J + \lambda^{*T} \tag{41}$$

5 Simulations

5.1 Environment and Parameter Settings

To show the validity of the idea of using a rolling motion for the control purpose, we tested the controller by simulations. The parameters of the snake robot that we assumed in the simulations are shown in Table 1. The 5th link was assumed to be passive. The reference shape of the robot is shown in Fig. 7, with the parameters shown in Table 2.

In the reference head trajectory of the crawler gait, the origin of the arc-length coordinate s is defined to be the switching point at which the trajectory is divided into the grounded and the lifted parts, as shown in Fig. 8. In the setting we used in the simulations, the passive joint reaches the origin when the head position

Table 1 Specification of the snake robot

Symbol	Description	Value
n	Number of links	37
r_0	Radius of a link	0.028 m
m_0	Mass of a link	0.16 kg
l_0	Length of a link	0.07 m

Table 2 Parameter of Crawler-Gait

Symbol	Description	Value
h_c	Height	0.25 m
w_c	Width	0.2 m
d_c	Distance between circular arcs	0.2 m

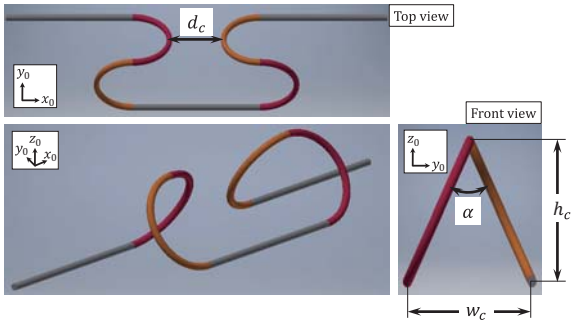


Fig. 7 Configuration of Crawler Gait

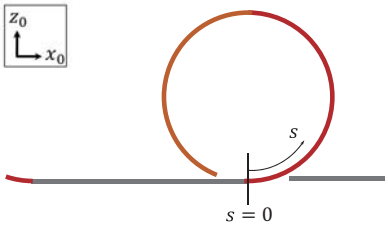


Fig. 8 Definition of $s = 0$

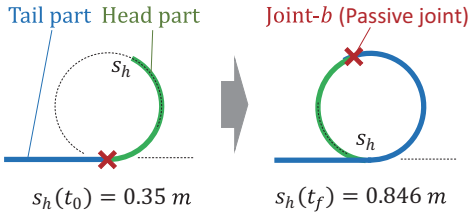


Fig. 9 Variation range of parameter $s_h(t)$

parameter is $s_h = 0.35$ m, and the head re-touches the ground with $s_h = 0.846$ m (Fig. 9). Because we assume that the passive joint is in the lifted part that is forming a cantilever-like structure, the time-interval of the simulations was set to be $t \in [t_0, t_f]$, where $s_h(t_0) = 0.35$ m and $s_h(t_f) = 0.846$ m. Note that, if $s_h \notin [0.35, 0.846]$, the passive joint looses its degrees of freedom because of the kinematic constraints, which eliminates the necessity of a careful control.

5.2 Uniform Controllability of the Linearized System

Before performing the simulations, the uniform controllability of the linearized system was checked. A criterion to check the uniform controllability was given in [15].

Theorem 1 ([15]) *A linear system* $(A(t), B(t))$ *is uniformly controllable if the controllability matrix* $Q_C(t)$ *defined below is full-rank for all* $t \geq 0$*:*

$$Q_C(t) = [\boldsymbol{p}_0(t),\ \ldots,\ \boldsymbol{p}_{n-1}(t)], \tag{42}$$

$$\boldsymbol{p}_{k+1}(t) = -A(t)\boldsymbol{p}_k(t) + \dot{\boldsymbol{p}}_k(t), \tag{43}$$

$$\boldsymbol{p}_0(t) = B(t). \tag{44}$$

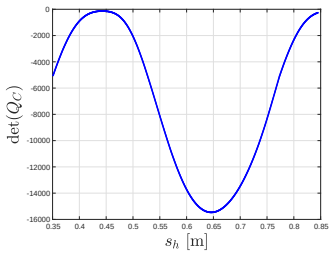


Fig. 10 The determinant of the controllability matrix Q_C

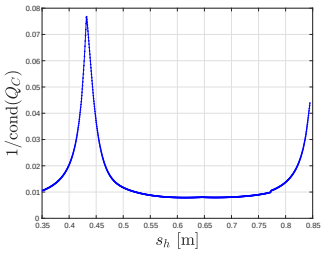


Fig. 11 The reciprocal of the condition number of Q_C

It is difficult to check this property for all possible configurations. Therefore, in this test, we check the property for the reference configuration sequence of the crawler gait. The determinant and the reciprocal of the condition number of the controllability matrix Q_C are shown in Figs. 10, 11. It can be seen that the determinant is less than 0 for all s_h, which shows that the linearized system is uniformly controllable. Although the determinant gets close to 0 around $s_h = 0.44$ m and $s_h = 0.85$ m, the reciprocal of the condition number is rather large, which implies that the matrix Q_C is not close to singular even around those points.

5.3 Results of Nonstationary Linear Quadratic Control

In this simulation, we define S_f as the solution of the Riccati algebraic equation

$$\begin{aligned} &-S_f A(t_f) - A^T(t_f)S(t_f) \\ &\quad + S_f B(t_f)B^T(t_f)S_f - Q(t_f) = O. \end{aligned} \tag{45}$$

For the weight matrix Q, we tested the following 2 cases:

1. $Q = \mathrm{diag}\ (1.0 \times 10^7, 1.0, 1.0, 1.0)$,
2. $Q = \mathrm{diag}\ (1.0 \times 10^8, 1.0, 1.0, 1.0)$.

This setting implies that we put much more weight on θ than other state variables, i.e., $\dot{\theta}$, ψ_0, and $\dot{\psi}_0$.

The results are shown in Fig. 12. The dash-dotted lines refer to the reference trajectory, dashed lines to the obtained trajectories in the case 1 ($Q = \mathrm{diag}\ (1.0 \times 10^7, 1.0, 1.0, 1.0)$), and solid lines to those in the case 2 ($Q = \mathrm{diag}\ (1.0 \times 10^8, 1.0, 1.0, 1.0)$).

It can be seen that the realized trajectories of θ closely follow the reference, which implies the validity of the controller and of the idea of using rolling motion to control a passive joint. Although there are relatively large tracking error in $\dot{\psi}_0$, it does not matter as long as the tracking of θ is achieved, because ψ_0 does not affect the shape of the robot. As expected, the tracking error can be reduced by increasing the weight for θ.

5.4 Results of Nonlinear Model Predictive Control

In this simulation, the weight matrix $Q = \mathrm{diag}(1.0 \times 10^5, 2.0 \times 10^5, 5.0 \times 10^3, 1.0)$ was used.

We employed the C/GMRES method [16] to solve the two-point boundary value problem (36)–(40). In the C/GMRES method, the time-horizon T is defined to be a time-varying one as $T(t) = T_f(1 - e^{-\alpha t})$, where we used $T_f = 1$ s and $\alpha = 1$.

The results are shown in Fig. 13. The dash-dotted lines refer to the reference trajectory and the solid lines to the obtained trajectories. The dashed lines show the result by NSLQC with the same weight matrix Q. It can be seen that NMPC achieves less errors in θ and ψ_0 with smaller input.

The calculation time required to solve the NMPC problem was 0.53 s for the entire time duration (10 s) and therefore, it can be used in real time. Because the NSLQC requires more calculation to solve the Riccati differential equation without clear improvements in the performance over NMPC, we believe NMPC is the better choice.

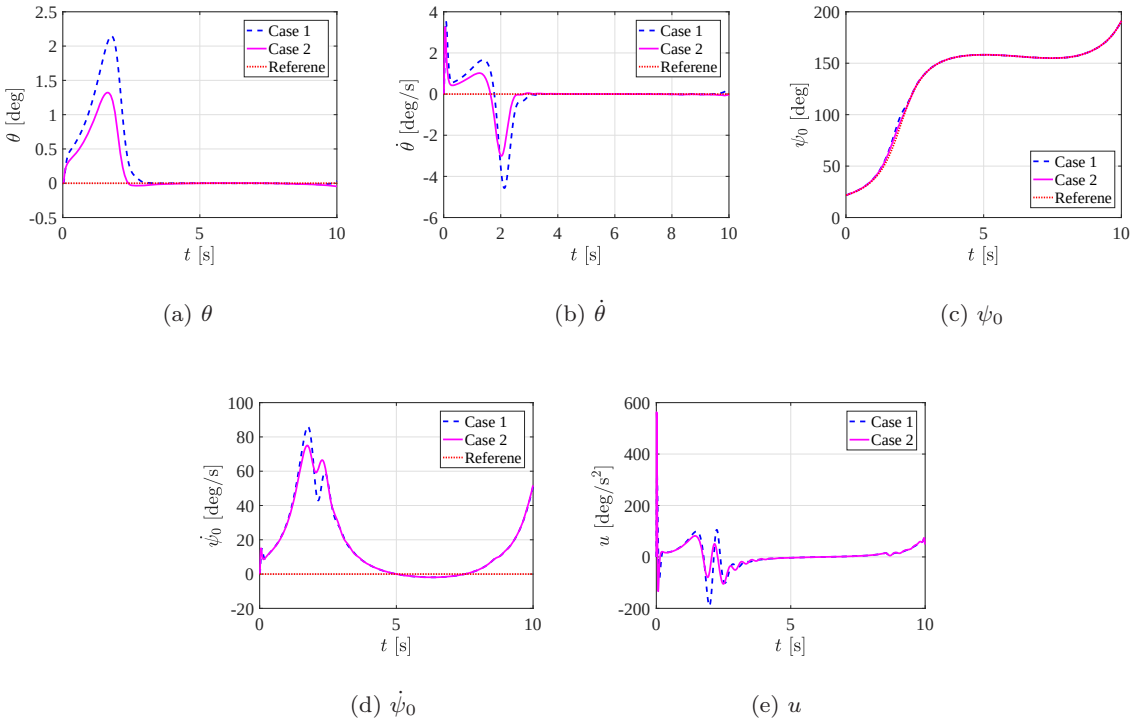
6 Conclusion

In this paper, we proposed to use rolling motion to control a passive joint in a snake robot, with a case of actuator malfunctioning in mind. A simplified model for the controller design is proposed and it is numerically confirmed that the linearization of the model is uniformly controllable. NSLQC and NMPC methods are employed to show the validity of the idea.

Although the validity of the idea of using rolling for control purpose was shown, the effectiveness of the controller to a real robot is still to be tested. We are planning to test the controller in more realistic simulations and a real robot. Another remaining problem is in the assumption that the passive joint moves smoothly. As one of the target situation will be the case where a joint is malfunctioning, this assumption can be problematic: if the malfunction is caused by mechanical problems rather than electronic ones, it is often the case that there is a non-negligible stick-slip like effect. Dealing with such more practical problems will be one of our future works.

References

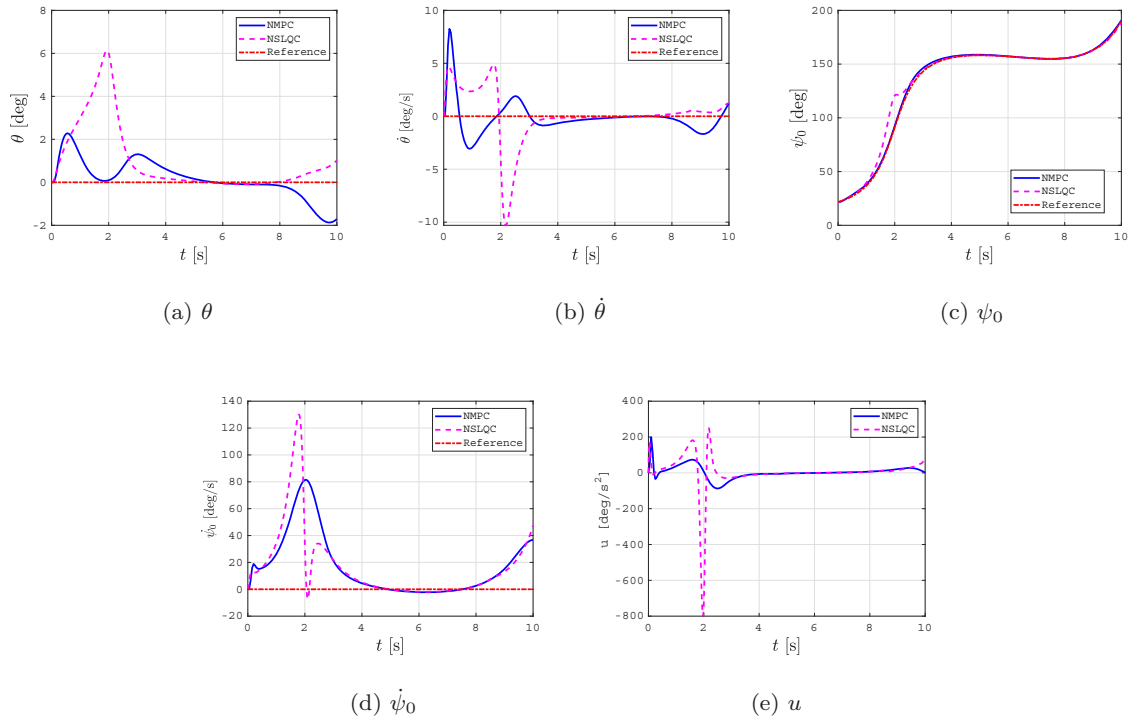
1. Baba T, Kameyama Y, Kamegawa T, Gofuku A (2010) A snake robot propelling inside of a pipe with helical rolling motion. Proc SICE Annual Conference: 2319–2325
2. Tanaka M, Tanaka K (2015) Control of a Snake Robot for Ascending and Descending Steps. IEEE Trans Robot 31(2): 511–520
3. Takemori T, Tanaka M, Matsuno F (2018) Gait Design for a Snake Robot by Connecting Curve Segments and Experimental Demonstration. IEEE Trans Robot 34(5):1384–1391
4. Liljebäck P, Pettersen KY, Stavdahl Ø, Gravdahl JT (2010) Hybrid Modeling and Control of Obstacle-Aided Snake Robot Locomotion. IEEE Trans Robot 26(5): 781–799
5. Travers M, Gong C, Choset H (2015) Shape-Constrained Whole-Body Adaptivity. Proc IEEE Int Symposium on Safety, Security, and Rescue Robotics: 1–6
6. Travers M, Whitman J, Schiebel P, Goldman D, Choset H (2016) Shape-Based Compliance in Locomotion. Proc Robotics: Science and Systems
7. Matsuno F, Mogi K (2000) Redundancy controllable system and control of snake robots based on kinematic model. Proc IEEE Conf Decision and Control: 4791–4796
8. Mehta V, Brennan S, Gandhi F (2008) Experimentally Verified Optimal Serpentine Gait and Hyperredundancy of a Rigid-Link Snake Robot. IEEE Trans Robot 24(2): 348–360
9. Ariizumi R, Takahashi R, Tanaka M, Asai T (2019) Head Trajectory Tracking Control of a Snake Robot and its Robustness Under Actuator Failure. IEEE Trans Control System Technol 27(6): 2589–2597
10. Ariizumi R, Koshio K, Tanaka M, Matsuno F (2019) Control of a Passive Joint in a Snake Robot Using Rolling Motion. Proc The 3rd Int Symposium on Swarm Behavior and Bio-Inspired Robotics: 161–166
11. Tesch M, Lipkin K, Brown I, Hatton R, Peck A, Rembisz J, Choset H (2009) Parametrized and Scripted Gaits for Modular Snake Robots. Adv Robot 23(9): 1131-1158
12. Yamada H, Hirose S (2008) Study of Active Cord Mechanism—Approximations to Continuous Curves of a Multi-joint Body—. Journal of the Robotics Society of Japan 26(1): 110–120 (in Japanese).
13. Klaassen B, Paap KL (1999) GMD-SNAKE2: A snake-like robot driven by wheels and a method for motion control. Proc IEEE Int Conf Robot Automat: 3014–3019
14. Callier F, Desoer CA (1992) Linear System Theory. Springer-Verlag, Hong Kong
15. Silverman LM, Meadows HE (1967) Controllability and Observability in Time-Variable Linear Systems. J SIAM Control 5(1): 64–73
16. Ohtsuka T (2004) A continuation/GMRES method for fast computation of nonlinear receding horizon control. Automatica 40: 563-574



(a) θ (b) $\dot{\theta}$ (c) ψ_0

(d) $\dot{\psi}_0$ (e) u

Fig. 12 Results of the trajectory tracking control by NSLQC with 2 different weight matrices. Case 1: $Q = \text{diag}\ (1.0 \times 10^7, 1.0, 1.0, 1.0)$ and Case 2: $Q = \text{diag}\ (1.0 \times 10^8, 1.0, 1.0, 1.0)$



(a) θ (b) $\dot{\theta}$ (c) ψ_0

(d) $\dot{\psi}_0$ (e) u

Fig. 13 Results of the trajectory tracking control by NMPC and NSLQC using the same weight matrix $Q = \text{diag}(1.0 \times 10^5, 2.0 \times 10^5, 5.0 \times 10^3, 1.0)$