

# Supervised learning of mapping from sensor space to chained form for unknown non-holonomic driftless systems

*Francisco Jesús Arjonilla García*

Graduate School of Science and Technology, Shizuoka University, Shizuoka, Japan, and

*Yuichi Kobayashi*

Department of Mechanical Engineering, Faculty of Engineering, Shizuoka University, Shizuoka, Japan and  
Graduate School of Science and Technology, Shizuoka University, Shizuoka, Japan

## Abstract

**Purpose** – This study aims to propose an offline exploratory method that consists of two stages: first, the authors focus on completing the kinematics model of the system by analyzing the Jacobians in the vicinity of the starting point and deducing a virtual input to effectively navigate the system along the non-holonomic constraint. Second, the authors explore the sensorimotor space in a predetermined pattern and obtain an approximate mapping from sensor space to chained form that facilitates controllability.

**Design/methodology/approach** – In this paper, the authors tackle the controller acquisition problem of unknown sensorimotor model in non-holonomic driftless systems. This feature is interesting to simplify and speed up the process of setting up industrial mobile robots with feedback controllers.

**Findings** – The authors validate the approach for the test case of the unicycle by controlling the system with time-state control policy. The authors present simulated and experimental results that show the effectiveness of the proposed method, and a comparison with the proximal policy optimization algorithm.

**Originality/value** – This research indicates clearly that feedback control of non-holonomic systems with uncertain kinematics and unknown sensor configuration is possible.

**Keywords** Mapping acquisition, Jacobian, Non-holonomic system, Partially unknown kinematics, Sensorimotor mapping

**Paper type** Research paper

## 1. Introduction

For actual implementation of autonomous navigation of mobile robots, we face two problems in the real use. First, the mobile robot structure can be generally under non-holonomic constraint. Even though there exist mobile mechanisms such as omni-directional wheels that are free from non-holonomic constraints, they are complex and often their power efficiency is low. Among those non-holonomic driftless systems (Borisov *et al.*, 2016), the unicycle, which has a single non-holonomic constraint, is a typical example, which is considered in this research. Second, the structure of the robot system can be partially unknown. Implementation of mobile robots generally requires definition and management of sensor measurement based on global coordinate system, which requires parameters of sensor configurations. It may cause a preparation cost when such parameters are unknown. Regarding mobile robot kinematics, even when we know the structure of the mobile base, parameters such as wheel radius and wheel base can be

unknown. Thus, learning approach is expected to resolve this problem by covering both unknown sensor settings and partially unknown kinematics.

The stabilization problem of non-holonomic systems has been tackled often. Brockett (1983) suggested that there is no stabilizing control law in general for these systems. Rifford (2008) identified two obstructions, one global and one local, to the existence of stabilizing feedbacks. Discontinuous state feedback control laws have been proposed such as in Astolfi (1995), who suggested applying a single coordinate transformation to the non-holonomic system in chained form. Amar and Mohamed (2013) designed a controller based on kinematic polar coordinate transformations. D'Andrea-Novell *et al.* (1991) showed that stabilization of three-wheel mobile robots is possible with static state feedback using Lagrange

---

The current issue and full text archive of this journal is available on Emerald Insight at: <https://www.emerald.com/insight/0143-991X.htm>



Industrial Robot: the international journal of robotics research and application  
48/5 (2021) 710–719  
Emerald Publishing Limited [ISSN 0143-991X]  
[DOI 10.1108/IR-10-2020-0221]

---

© Francisco Jesús Arjonilla García, and Yuichi Kobayashi. Published by Emerald Publishing Limited. This article is published under the Creative Commons Attribution (CC BY 4.0) licence. Anyone may reproduce, distribute, translate and create derivative works of this article (for both commercial & non-commercial purposes), subject to full attribution to the original publication and authors. The full terms of this licence may be seen at <http://creativecommons.org/licenses/by/4.0/legalcode>

Received 6 October 2020  
Revised 11 November 2020  
16 December 2020  
18 January 2021  
Accepted 19 January 2021

formalisms and differential geometry. However, all these proposals assume that the robot kinematics, the sensor configuration and the environment are well known.

On the other hand, depending on the application, different sensors or image features may be used. For example, if Global Positioning System or ceiling camera is not available, it is not easy to obtain  $(x, y, \theta)$  coordinates in Cartesian space. Even in such case, it should be possible to navigate a robot to a destination by specifying *desired sensor value* as the target. This approach will widen applicability of mobile robots with less calibration effort. However, the model of a control law in such robot must still agree with the configuration of the actuators, the sensors and the environment at all times, otherwise the controller will not function correctly.

There have been many proposals to adapt the control law to the problem setting automatically (Kolmanovsky and McClamroch, 1995). For example, Graefe and Maryniak (1998) built a map from sensor-control Jacobians for controlling robot manipulators with calibration-free visual systems, but their approach did not consider robotic systems with non-holonomic constraints. Similarly, Navarro-Alarcon *et al.* (2019) computed adaptive navigation systems with unknown sensorimotor models. Kobayashi *et al.* (2013) used a method to approximate the Jacobian by gradient descent on non-overlapping sensor spaces and extrapolate the Jacobian mapping outside the sensing ranges to estimate an integrated sensor space. They demonstrated their method in a 2-degree-of-freedom (DoF) manipulator and in a non-holonomic mobile robot traveling along an infinite wall. Still, they neutralized the non-holonomic constraints of the unicycle by discarding the coordinate that was parallel to the reference wall. Miller (1987) used a general learning algorithm to learn the relation between control inputs and sensor outputs in a robot arm. Likewise, Kobayashi *et al.* (2019) also proposed estimating the Jacobian matrix for visual servoing with unknown kinematics and other system parameters by approximating the relation between actuators and sensors using a measurement given by mutual information. In contrast to previous works on stabilization of non-holonomic systems, these research studies do suggest solutions to modeling the system, although they are not sufficiently general enough for considering non-holonomic constraints.

More recently, reinforcement learning algorithms (Smart and Kaelbling, 2002) have tackled the problem of uninterpreted sensors and effectors by achieving controllability of non-holonomic systems with unknown sensorimotor mapping. The acquisition of a controller could be made more sample-efficient by considering the non-holonomicity of the system, rather than relying on a hand-crafted reward design, which is often required by reinforcement learning algorithms. In addition, in the case of a driftless system, sample collection can be made more efficient and even safer by a lattice-shaped pattern of exploratory motion.

In this paper, we address the problem of learning a sensorimotor mapping for a class of non-holonomic driftless systems with unknown kinematics and unknown sensor configuration to combine the applicability of adaptive controllers and non-holonomic controllers. For this purpose, we first formulate the problem, then we present the learning approach, and finally we show simulation and experimental

results of the method applied to the unicycle problem in a variety of sensor configurations. The main contribution of this paper is a method to automatically construct such mapping in a systematic way for a predefined region of expected controllability.

The remainder of this paper is organized as follows. Problem definition, notation and essential knowledge are described in Section 2. Theoretical development of the method is described in Section 3. Simulation and experimental results are presented in Section 4 and discussion follows in Section 5.

## 2. Problem setting

Let  $u \in R^m$  be the control (input) vector and  $s \in R^n$ ,  $n > m$  be the sensor (output) vector of a dynamic driftless affine system with state and output equations

$$\begin{aligned}\dot{q} &= F(q)u \\ s &= H(q),\end{aligned}\quad (1)$$

where  $q \in R^n$  is the vector of generalized coordinates,  $\dot{q}$  is the vector of generalized velocities and  $H: R^n \rightarrow R^n$  is an isomorphic mapping of class  $C^1$ . The transformation  $H$  is arbitrary and has no units.

The problem tackled in this paper is to find a control law  $u = \varphi(s)$  that realizes a desired sensor value of  $s^{(d)}$  under the condition of unknown  $F, H$  (unknown kinematics and sensor configuration), with arbitrary  $q$ , and with non-holonomic constraints compatible with Pfaffian form, i.e.  $A(q)q = 0$  (Choset *et al.*, 2005). In other words, we can observe  $q$  but only through an uncalibrated sensor measurement. The problem is similar under redundant observations  $s' \in R^r$ ,  $r > n$ , in which case we consider  $s'' = H(q)$  with  $s'' \in R^n$  and  $H''$  an isomorphic mapping. We assume that the inputs can be driven independently and that the sensor signal is differentiable with respect to the input. The following sections present formal descriptions of the basic concepts.

### 2.1 Non-holonomic systems

Holonomic systems are those whose constraints obey the equation

$$f(q_1, \dots, q_n, t) = 0. \quad (2)$$

When the constraint cannot be expressed in the form of equation (2), then the system has non-holonomic constraints. Non-holonomic systems pose more difficulties than holonomic systems because the Lagrangian equations cannot be applied. A system with first-derivative non-holonomic constraints may be expressed with the equation

$$f(q_1, \dots, q_n, \dot{q}_1, \dots, \dot{q}_2, t) = 0. \quad (3)$$

Many of these systems are characterized by a smaller number of control inputs than DoFs, like in the case of the unicycle, the rolling wheel and the rolling sphere problems.

#### 2.1.1 Unicycle

In this paper, we rely on the unicycle for specifying the methods and results without loss of generality. The unicycle is a non-holonomic system with  $n = 3$  DoFs and  $m = 2$  control inputs. The state equation is

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \mathbf{u}, \quad (4)$$

where  $x$ ,  $y$  and  $\theta$  denote positions and orientation of the unicycle as depicted in Figure 1. The input vector  $\mathbf{u} = [u_1 \ u_2]^T$  is normally comprised of linear and rotational velocities, but in this research, it is not specified which component of  $\mathbf{u}$  corresponds to each of them. Here it should be noted that some variations of the unicycle, i.e. a car-like system with limited rotation, add an additional non-holonomic constraint

$$\sqrt{x^2 + y^2} \theta \geq R. \quad (5)$$

We do not deal with these constraints in this paper.

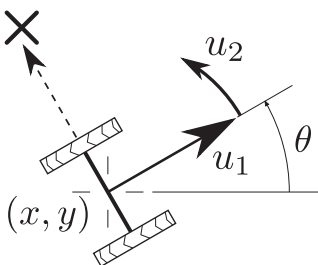
### 2.2 Approach

We propose an offline learning algorithm to obtain a mapping from sensor space to chained form (Jiang and Nijmeijer, 1999) as follows. First, we deduce a virtual input component  $u_3$  by Jacobian estimation and composed of a sequence of legal inputs such that the input vector becomes  $u_v = [u_1 \ u_2 \ u_3]^T$ , with the aim of overcoming the forbidden direction posed by the non-holonomic constraint at the initial state. Second, we explore the sensor space following a fixed trajectory considering the virtual input deduced earlier to obtain a mapping  $\phi$  from sensor space to chained form  $z = \phi(s)$ . Therefore,  $\phi$  is approximated by data collection rather than from a mathematical model, which is unknown. This method is advantageous because it allows skipping procedures of modeling, calibration and sensor mapping measurement. Finally, we validate the method by controlling the system with time-state control policy. There are many approaches to deal with controllability of systems in chained form (Luo and Tsiotras, 2000; Murray and Sastry, 1991; Jiang and Nijmeijer, 1999), but here we use time-state control, proposed in Sampei (1994) and Sampei et al. (1996) and later described in more detail with a similar control technique in Lefeber et al. (2000) and Lefeber et al. (2004), because it is relatively simple and easy to implement.

### 3. Estimation of sensorimotor mapping

We define two learning stages. The first stage tackles the problem of system controllability. In other words, the method

**Figure 1** The unicycle is a non-holonomic system with canonical generalized coordinates  $(x, y, \theta)$  as shown and non-holonomic constraint  $\dot{x}\sin\theta - \dot{y}\cos\theta = 0$



starts by learning the inputs required to explore the sensor space efficiently. Later, the second stage uses the results of the first stage to navigate in the sensor space and gather a data set comprised of sensor samples and the corresponding generalized coordinates in chained form assuming a well-defined trajectory. This data set is used to infer a mapping from sensor space to chained form. In Section 3.3, we describe the method to assess the accuracy of the approach.

### 3.1 Model learning

In the first stage, the controller learns to navigate efficiently through the sensor space. For that purpose, the control system needs to learn how to control the variation of each coordinate of the sensor signal independently. However, there are only two inputs but three coordinates in the sensor signal, thus only two dimensions are immediately controllable from the initial position. Here we show how to calculate a sequence of motions to travel along the forbidden direction while minimizing variations along the rest of the sensor space.

#### 3.1.1 Jacobian.

Let  $s^{(p)} \in R^3$  be the unitless sensor observation sampled at point  $p$ , where  $p$  is identified by sensor value, and let  $\mathbf{s}_{(\vartheta)}^{(p)}$  denote the time derivative of the sensor observation when input  $\mathbf{u}_{(\vartheta)}$  is applied, where  $\vartheta \in \{1, 2\}$ ,  $\mathbf{u}_{(1)} = [1 \ 0]^T$  and  $\mathbf{u}_{(2)} = [0 \ 1]^T$ .

The sensor-control Jacobian  $\mathcal{J}$  is defined as a measure of the variability of the sensor signals with respect to the inputs in matrix form:

$$\mathcal{J} := \frac{\partial \mathbf{s}}{\partial \mathbf{u}} \Big|_{n \times m} = \begin{bmatrix} \frac{\partial s}{\partial u_1} & \frac{\partial s}{\partial u_2} \\ \frac{\partial s_2}{\partial u_1} & \frac{\partial s_2}{\partial u_2} \\ \frac{\partial s_3}{\partial u_1} & \frac{\partial s_3}{\partial u_2} \\ \frac{\partial u_1}{\partial u_1} & \frac{\partial u_2}{\partial u_2} \end{bmatrix} = [\mathbf{j}_1 \ \mathbf{j}_2] \quad (6)$$

At the initial state  $s^{(0)}$ , the sensor-control Jacobian  $\mathcal{J}^{(0)} = \mathcal{J}(s = s^{(0)})$  of the system indicates the variation of the sensor signal with respect to each input  $u_{(1)}$  and  $u_{(2)}$ . The state equation in sensor space at any point  $p$  and input  $u_{(\vartheta)}$  is

$$\mathbf{s}_{(\vartheta)}^{(p)} = \mathbf{g}(\mathbf{s}^{(p)})\mathbf{u}_{(\vartheta)} = \mathbf{g}_1(\mathbf{s}^{(p)})\mathbf{u}_{(\vartheta),1} + \mathbf{g}_2(\mathbf{s}^{(p)})\mathbf{u}_{(\vartheta),2}. \quad (7)$$

Considering control input  $\mathbf{u} = \tau_1 \mathbf{u}_{(1)} = [\tau_1 \ 0]^T$ , then  $\mathbf{s}_{(1)}^{(p)} = \mathbf{g}_1(\mathbf{s}^{(p)})\tau_1$  so

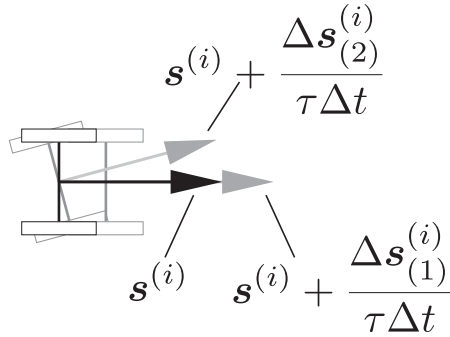
$$\begin{aligned} \frac{\partial \mathbf{s}_{(1)}^{(p)}}{\partial \mathbf{u}} &= \frac{\partial}{\partial \mathbf{u}} \mathbf{g}_1(\mathbf{s}^{(p)})u_1 + \frac{\partial}{\partial \mathbf{u}} \mathbf{g}_2(\mathbf{s}^{(p)})u_2 = \mathbf{g}_1(\mathbf{s}^{(p)}) \frac{\partial u_1}{\partial u_1} = \frac{\mathbf{s}^{(p)}}{\tau_1} \\ &\approx \frac{\Delta \mathbf{s}^{(p)}}{\tau_1 \Delta t}. \end{aligned} \quad (8)$$

Similarly for  $\mathbf{u} = \tau_2 \mathbf{u}_{(2)} = [0 \ \tau_2]^T$  and substituting for  $\mathbf{s}_{(2)}^{(p)}$ , we arrive at

$$\mathcal{J}^{(p)} = \begin{bmatrix} \mathbf{j}_1^{(p)} & \mathbf{j}_2^{(p)} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{s}^{(p)}}{\partial u_1} & \frac{\partial \mathbf{s}^{(p)}}{\partial u_2} \end{bmatrix} \approx \begin{bmatrix} \frac{\Delta \mathbf{s}_1^{(p)}}{\tau_1 \Delta t} & \frac{\Delta \mathbf{s}_2^{(p)}}{\tau_2 \Delta t} \end{bmatrix}. \quad (9)$$

See Figure 2 for a depiction of equation (9). Hence, the Jacobian is easily obtained by measuring the change in sensor values before and after applying a constant input  $u_{(\vartheta)}$  to the system, sequentially for every input. After obtaining each

**Figure 2** The Jacobian is obtained from subtracting the sensor observation at the target state  $s^{(i)}$  from the observations after applying inputs  $u_{(1)}$  and  $u_{(2)}$  for a small amount of time



Jacobian element, the system backtracks its movements to return to  $s^{(i)}$ . In the case of the unicycle, there is one inaccessible dimension, so we need to find a state  $s^{(*)}$ , whose Jacobian  $\mathcal{J}^{(*)}$  contains an element  $j_{\psi}^{(*)}$  with  $\psi \in \{1,2\}$  such that

$$\det \begin{bmatrix} j_1^{(0)} & j_2^{(0)} & j_{\psi}^{(*)} \end{bmatrix} = \pm 1, \tag{10}$$

or at least as close to  $\pm 1$  as possible, assuming that all Jacobian elements are normalized (Figure 3). We explore the vicinity of  $s^{(0)}$  in search for  $s^{(*)}$  by applying a control policy such that an input  $u_{(s^{(p)})} = u_{(1)}$  or  $u_{(s^{(p)})} = u_{(2)}$  is driven for a fixed amount of time  $\Delta t_{(s^{(p)})}$ . Then the Jacobian  $\mathcal{J}^{(p)}$  is obtained at the resulting state  $s^{(p)}$ , and the system is taken back to the initial state by applying  $-u_{(s^{(p)})}$  for the same amount of time. The process is repeated in an exponential search for  $\Delta t_{(s^{(*)})}$  and  $u_{(s^{(*)})}$  that optimizes equation (10):

$$(s^{(*)}, \psi) = \arg \max_{s^{(p)}, i \in \{1,2\}} |\det \begin{bmatrix} j_1^{(0)} & j_2^{(0)} & j_i^{(p)} \end{bmatrix}|. \tag{11}$$

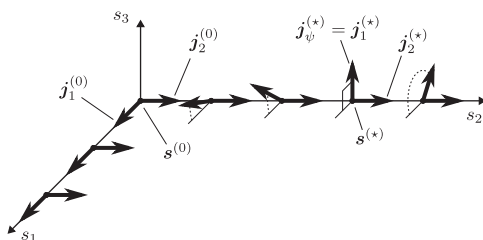
3.1.2 Virtual input.

As shown above, the input  $u_{(s^{(*)})}$  applied for  $\Delta t_{(s^{(*)})}$  will reach state  $s^{(*)}$ , which is a state where applying input  $u_{(\psi)}$  will maximize movement in the direction forbidden by the non-holonomic constraint at  $s^{(0)}$ . If  $u_{(\psi)}$  for  $\Delta t$  is then followed by input  $-u_{(s^{(*)})}$  for  $\Delta t_{(s^{(*)})}$ , the resulting state is

$$s = s^{(0)} + j_{\psi}^{(*)} u_{(\psi)} \Delta t, \tag{12}$$

that is, the system effectively travels along the forbidden direction at  $s^{(0)}$ . Therefore, we have deduced a sequence of inputs whose end result is equivalent to a virtual input the direction of the non-holonomic constraint:

**Figure 3** The Jacobian element  $j_{\psi}^{(*)}$  orthogonal to  $j_1^{(0)}$  and  $j_2^{(0)}$  in sensor coordinates is found by exploration of the sensor space



$$u_{(3)} \Delta t \iff u_{(*)} \Delta t_{(*)}; u_{(\psi)} \Delta t; -u_{(*)} \Delta t_{(*)}. \tag{13}$$

We will use inputs  $u_{(1)}$ ,  $u_{(2)}$  and virtual input  $u_{(3)}$  to navigate the sensor space freely in the next stage. Conceptually, it is similar to a holonomic system with an additional input.

3.2 Mapping of sensor space to chained form

The state of a non-holonomic system depends on the history of the control inputs as a result of the non-integrable constraints. Therefore, navigation in the sensor space requires tracking the input history and applying it to the kinematics of the system to obtain a consistent state. However, in this problem, we cannot rely on the kinematics of the system. Here we propose to circumvent the kinematics problem by returning the state to its initial position by backtracking the sequence of inputs applied to reach each sampled state. This method requires that there are no significant deviations in the trajectory of the system when backtracking compared to the outward trajectory.

3.2.1 Chained form

Chained form is a canonical formulation that obeys the formula (shown here for the two-input case)

$$\dot{z} = G(z)u = g_1(z)u_1 + g_2(z)u_2, \tag{14}$$

where

$$g_1(z) = \begin{bmatrix} 1 \\ 0 \\ z_2 \end{bmatrix} \text{ and } g_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}. \tag{15}$$

Now, let  $\phi = [\phi_1 \ \phi_2 \ \phi_3]^T$  be the mapping of sensor coordinates to chained form coordinates, denoted by

$$z^{(i)} = \phi(s^{(i)}), \tag{16}$$

where  $i$  indicates some state. Knowing from the definition of Jacobian that

$$\dot{s}^{(i)} = \mathcal{J}^{(i)}u \tag{17}$$

and deriving equation (16) with respect to time assuming that  $u$  is constant,

$$z_i = \begin{bmatrix} \frac{d\phi_1(s)}{ds} \Big|_{s^{(i)}} s^{(i)} \\ \frac{d\phi_2(s)}{ds} \Big|_{s^{(i)}} s^{(i)} \\ \frac{d\phi_3(s)}{ds} \Big|_{s^{(i)}} s^{(i)} \end{bmatrix} u = \frac{d\phi(s)}{ds} \Big|_{s^{(i)}} \mathcal{J}^{(i)}u. \tag{18}$$

From the definition of chained form equation (14), the following holds:

$$\dot{z}^{(i)} = G(z^{(i)})u = G(\phi(s^{(i)}))u. \tag{19}$$

Equating equation (18) to equation (19) and removing  $u$ , we arrive at

$$G(\phi(s^{(i)})) = \frac{d\phi(s)}{ds} \Big|_{s^{(i)}} \mathcal{J}^{(i)}, \tag{20}$$

where it can be seen that  $G$  has two terms: the first one  $\frac{d\phi}{ds}$  is the Jacobian of the mapping from sensor space to chained form with respect to the sensor observations, and the second one  $\mathcal{J}^{(i)}$



is the same Jacobian as in the first stage, although sampled at different coordinates. Actually, we can sample  $G$  directly, bypassing the need to calculate the two terms separately.

### 3.2.2 Exploration of sensor space

In this section, we describe the method used to obtain pairs  $(s^{(i)}, z^{(i)})$  of corresponding coordinates in sensor space and chained form space. The sensor space sampling procedure involves controlling the system with a fixed sequence of inputs to reach each vertex in a grid in virtual space. The coordinates of each vertex are  $(c_1 u_{(1)} \Delta t_c, c_2 u_{(2)} \Delta t_c, c_3 u_{(3)} \Delta t_c)$  where  $c_1, c_2, c_3 \in Z$  are the indexes for the grid coordinates and  $\Delta t_c$  is a fixed parameter.

The fixed sequence of inputs must abide to the following rules on account of previously mentioned limitations:

- Every input must be backtracked in reverse order.
- $u_{(s^{(*)})}$  must always be applied in last place to prevent traversing along the subspace of virtual input  $u_{(3)}$  inadvertently.
- The use of the virtual input  $u_{(3)}$  should be minimized to reduce cumulative position errors.

The algorithm used herein is shown in Algorithm 1. The sensor samples  $s^{(i)}$  are read from the sensor observations while the virtual states  $z^{(i)}$  are calculated internally based on the input history. At each point of the grid, the virtual coordinate  $z^{(i)}$  is recorded together with the sensor observation  $s^{(i)}$  at that point. The resulting pair is incorporated into the data set for training the approximated mapping.

### 3.2.3 Function approximation.

The last step toward obtaining  $\phi$  consists in processing the data set obtained earlier by making use of radial basis functions with Gaussian kernels (Gaussian RBF), but other supervised-learning techniques such as neural networks should also be valid.

A Gaussian kernel takes the form

$$\varphi_j(\mathbf{s}) = \exp\left(-\frac{\|\mathbf{s} - \mathbf{b}_j\|^2}{2\sigma^2}\right). \quad (21)$$

Gaussian RBF is a linear combination of Gaussian kernels distributed in the target region of the approximation. Each location is set by  $\mathbf{b}_j$  and denoted a base. Thus, for one output variable,

$$\phi(\mathbf{s}) = \sum_{j=1}^{N_B} w_j \varphi_j(\mathbf{s}), \quad (22)$$

where  $N_B$  is the total number of kernels and  $\mathbf{w} = [w_1 \ \dots \ w_{N_B}]^T$  are the unknown linear coefficients, or weights. In this research, we specified the number of kernels and the approximation started by distributing the kernels in an orthogonal grid covering all sensor samples in the data set. The weights are calculated by least squares as specified in Kondor (2004). The loss function is

$$L(\mathbf{z}, \phi(\mathbf{s})) = \frac{1}{2} \|\mathbf{z} - \phi(\mathbf{s})\|^2. \quad (23)$$

Taking  $N$  as the number of points in the data set, we define

$$C(\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_{N-1}) = \begin{bmatrix} \varphi_1(\mathbf{s}_0) & \dots & \varphi_P(\mathbf{s}_0) \\ \vdots & \ddots & \vdots \\ \varphi_1(\mathbf{s}_{N-1}) & \dots & \varphi_P(\mathbf{s}_{N-1}) \end{bmatrix} \quad (24)$$

and the solution with regularization term  $\lambda$  is

$$\mathbf{w} = (C^T C - \lambda I)^{-1} C^T \mathbf{z},$$

which replaced in equation (22) gives us, at last,  $\phi$ .

### 3.3 Assessment

Evaluation of the estimated mapping from sensor space to chained form space is performed by placing the system at any point in the sampled region of the sensor space and controlling it to the origin. Here we apply time-axis control although alternative controllers may be equally valid. Non-holonomic systems often have non-linearities that linear controllers cannot deal with. Overcoming these difficulties is out of the scope of this research. For simplification, we assume that the starting rotation of the assessment of the controller is approximately parallel to the starting rotation of the learning stages.

#### 3.3.1 State space control of time-axis form.

The time-state control strategy involves transforming the state equation of a non-holonomic system into two independently controlled state equations (Sampei, 1994): the time control part and the state control part. The transformation involves a change in coordinates  $\mathbf{z} \alpha[\tau \ \xi]^T \in \mathbf{R}^3$ . The state equation of the time control part

$$\tau = h(\tau, \xi) u_1 \quad (26)$$

consists of a single generalized coordinate  $\tau \in \mathbf{R}$  controlled by the input component  $u_1$ . Typically, control of  $\tau$  is constant, i.e.  $u_1 = 1$ . The state control part can then be represented by

$$\frac{d\xi}{d\tau} = f_0(\xi) + f_1(\xi) u_2. \quad (27)$$

Equation (27) sees the time variable replaced with  $\tau$ , thereby equation (26) controls the time scale of equation (27). Control of  $\tau$  to the origin is achieved by alternating positive and negative values of  $u_2$  until  $\xi = 0$  and then making  $\tau = 0$  with  $u_1$ . The advantage of time-state control form is that in many cases, the state control part can be designed as if there were no non-holonomic constraints in the state equation. Application of non-linear transformation to  $q$  from some non-linear system to time-state control form with generalized coordinates  $(\tau, \xi)$  enables linear feedback control on the system.

The time-state part of a 3-DoF state is

$$\tau = \frac{dz_1}{dt} = u_1 \quad (28)$$

where we have set  $h(\tau, \xi) = 1$ , and its control-state part is

$$\frac{d}{dz_1} \begin{pmatrix} z_3 \\ z_2 \end{pmatrix} = A \begin{pmatrix} z_3 \\ z_2 \end{pmatrix} + B u_2, \quad (29)$$

where

$$A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \quad (30)$$

We now show the control law for assessing controllability of the target system. System (28) is driven by a constant input  $u_1 = 1$  and system (29) is controllable by state-space control with one input  $u_2 = f(z_2, z_3)$ , which is calculated as follows. The controllability matrix of equation (29) is (Dominguez et al., 2006)

$$Q = (B \mid AB) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad (31)$$

which is controllable as its rank is two. Under feedback stabilization with parameters  $K = (k_1 \ k_2)$ , the feedback controlled matrix  $A_f$  becomes

$$A_f = A + BK = \begin{pmatrix} 0 & 1 \\ k_1 & k_2 \end{pmatrix}. \quad (32)$$

Given control poles  $p_1$  and  $p_2$ , the characteristic polynomial is

$$P_f(s) = \left[ \frac{1}{(s - p_1)(s - p_2)} \right]^{-1} = s^2 - (p_1 + p_2)s + p_1p_2, \quad (33)$$

so  $k_1 = p_1 + p_2$  and  $k_2 = p_1 p_2$ . Thus, the control input with poles  $p_1$  and  $p_2$  is

$$u_2 = (k_1 \ k_2) \begin{pmatrix} z_3 \\ z_2 \end{pmatrix} = -p_1 p_2 z_3 + (p_1 + p_2) z_2. \quad (34)$$

Consequently, by controlling  $u_2$  with equation (34), we stabilize the system close to the time-axis indicated by equation (28).

#### 4. Implementation and results

We tested and validated our approach under simulated conditions and experimentally on a real robot.

##### 4.1 Simulation

We used the canonical state equation for the unicycle

$$q = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} u \quad (35)$$

$$s = H(q). \quad (36)$$

with  $u_1$  indicating linear speed and  $u_2$  rotational speed. These two equations are hidden from the learning and control algorithm: only  $s$  may be sampled and only  $u$  may be modified arbitrarily. Here, we show the results for the following three variations of  $H$ , which have been designed so that the mapping is isomorphic in the region of interest:

$$H_1(q) = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}, \quad H_2(q) = \begin{bmatrix} \sinh(y) \\ e^x \\ \arctan(\theta) \end{bmatrix}, \quad (37)$$

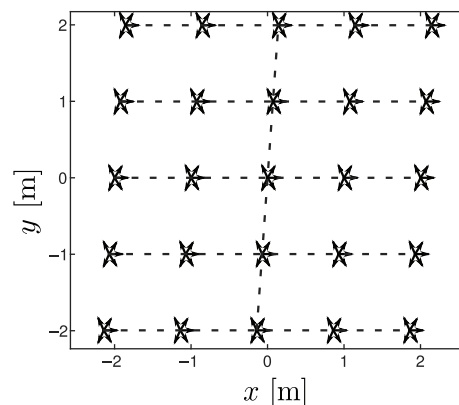
$$H_3(q) = \begin{bmatrix} x + e^y \\ e^x - y \\ \theta^3 \end{bmatrix}.$$

In all three cases, the parameters of the simulation were set as follows: the initial state was  $q_0 = [0 \ 0 \ 0]^T$ , 9 samples per axis with a separation of 0.25 units in the first stage, 5 samples per

axis (total of  $5^3$  samples) in the range  $[-2, 2]$  for constructing the data set, approximated by Gaussian RBF with  $5^3$  kernels, standard deviation 1.5 multiplied by the minimum distance between kernels and regularization term  $\lambda = 0.5$  in the second stage. The linear controller for assessment had poles  $(-5, 5)$ , starting position  $(x, y, \theta) = (-2, 0.5, \pi/4)$  and running time of 2.5 s. Standard deviation for the Gaussian kernels for each case were  $\sigma_1 = 1.1970$ ,  $\sigma_2 = 0.6643$  and  $\sigma_3 = 3.0745$ , respectively. The trajectory, observations and Gaussian kernel locations in sensor space of the sensor space mapping stage are shown in Figures 4 and 5.

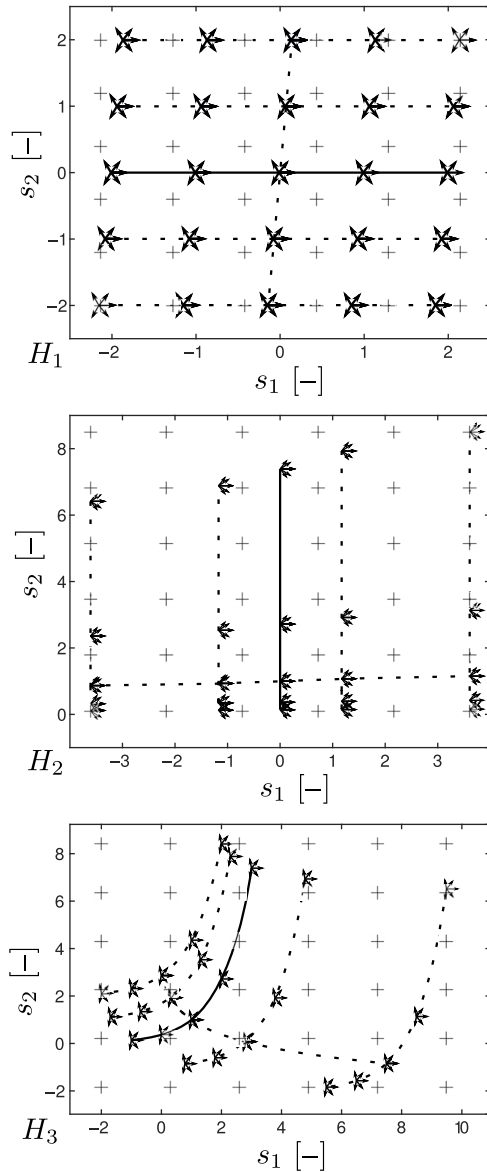
In the three sensor configurations, the system was successfully controlled to the time axis. We did not add a back and forth control to  $u_1$  to control the whole system to the origin because it was irrelevant for the purposes of this paper. Errors in the  $z_2$ -axis of the transformation function  $\phi$ , corresponding to  $y$  in  $(x, y, \theta)$  space, along the time-axis in chained space were negligible ( $\phi_2(\tau) = 0 \pm 10^{-13}$  for the three cases). In contrast, errors in the  $z_3$ -axis, corresponding to  $\theta$  in  $(x, y, \theta)$  space, were significant:  $\phi_3(\tau) = 0 \pm 0.0138$  for  $H_1$ ,  $\phi_3(\tau) = 0 \pm 0.0759$  for  $H_2$  and  $\phi_3(\tau) = 0 \pm 0.2356$  for  $H_3$ . The inaccuracies in the approximation of sensor observations to chained space are perceived as perturbations by the control law, and are appropriately corrected. Indeed, these inaccuracies resulted in small deviations of the controlled trajectory as shown in Figure 6. Positional errors derived from inaccuracies in the actuators were negligible as expected in a simulated environment. In the case of  $H_1$ , the sensor mapping is the identity, thus the trajectory in sensor space matches the trajectory in  $(x, y, \theta)$  space. The sampled observations are evenly distributed across the sensor space and the approximation of  $\phi$  is good. With respect to  $H_2$ , the Gaussian kernels cannot approximate accurately all the sampled observations in the region  $s_2 \in (0, 1)$  (Figure 5 –  $H_2$ ), which derives in a slight deviation of the trajectory as shown at the left of Figure 6 –  $H_2$ . In the case of  $H_3$ , the oscillation in the trajectory cannot be explained by the control poles because they are real values. Rather, the inaccuracies are better explained by the high concentration of sampled observations compared to the number of kernels near the initial state as shown in Figure 5 –  $H_3$ .

Figure 4 The trajectory of the simulated system during sensor space sampling (stage 2) in  $(x, y, \theta)$  coordinates is the same for  $H_1, H_2$  and  $H_3$



Note: Arrows indicate sampled states

**Figure 5** Dotted line: trajectory of the simulated system during sensor space sampling (stage 2) in sensor coordinates for  $H_1$  (top),  $H_2$  (middle) and  $H_3$  (bottom). Solid line: trajectory of the robot along the line taken as time axis

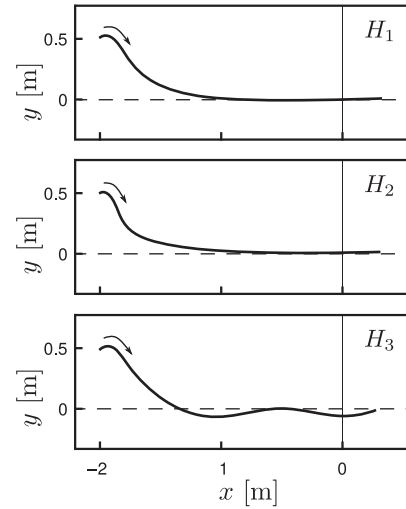


**Notes:** Arrows: sampled observations with their rotations indicating units of  $s_2$  in radians. Crosses: location of Gaussian kernel centers

#### 4.2 Mobile robot

We tested our approach experimentally on a real robot (Figure 7). We used the mobile robot model Pioneer 3-DX [1], which features two feedback-controlled wheels with a high resolution encoder and a swivel caster for balance. We used a 5K PTZ camera fixed on the ceiling and connected to an image processing workstation. The camera images were processed with OpenCV and Armadillo libraries and involved image segmentation by colors, noise removal and identification of beacon characteristics. The beacons were installed on the robot

**Figure 6** Trajectory of the simulated system after learning of sensorimotor mapping for the three sensor configuration transformation functions  $H_1$ ,  $H_2$  and  $H_3$



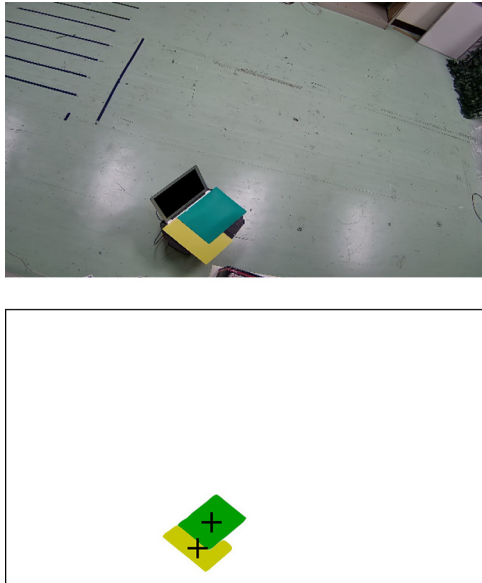
**Note:** The dotted line indicates the time axis  $\tau = z_1$

**Figure 7** The Pioneer 3-DX robot



as shown in Figure 8. The sensor outputs were the  $(x, y)$  pixel coordinates of the centroid of one beacon and the angle between the line connecting both beacons and  $y = 0$ . The CORBA [2] implementation by omniORB [3] was used to connect all off-board and on-board components. Inputs to the robot were linear and rotational speed, as in the simulation. The parameters of the sampling controllers were similar to the simulation but with reduced number of samples: six samples per axis with a separation of 0.3 units in the first stage, four samples per axis (total of  $4^3$  samples) separated by 0.667 units for constructing the data set, approximated by Gaussian RBF with  $4^3$  kernels, standard deviation of 0.45 and regularization

**Figure 8** The experimental setup as seen by the camera and image processing output



**Note:** Centroids of beacons 1 (green) and 2 (yellow) are indicated with a cross

term  $\lambda = 0.5$  in the second stage. The linear controller for assessment had poles  $(-5, -5)$ .

The mobile robot was successfully controlled after exploration (Figure 9) of the sensor space. The output of the first stage was  $\mathbf{u}_{(s)} = \mathbf{u}_{(2)} \Delta t_{(s)} = 1.5$  s. Figure 10 shows four trajectories starting at different points in the left and bottom sides of the figure converging toward the approximate position of the time axis. As in the simulations, we did not add a back and forth control to the linear velocity input. The imperfections in the sample observations in the second stage may be attributed to perspective deformation, lens aberrations, signal noise, image processing lag, partial occlusion of beacons and cumulative positional errors.

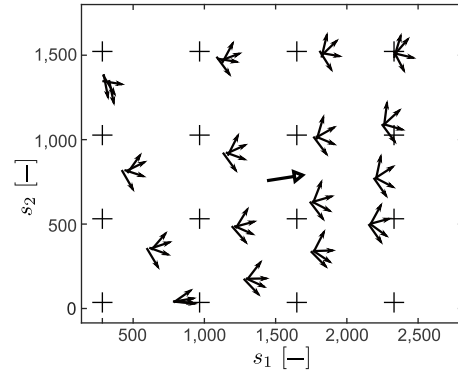
### 4.3 Comparison to proximal policy optimization

We compared the proposed approach to proximal policy optimization (PPO), which is a class of reinforcement learning algorithm (Schulman et al., 2017), in a problem setting similar to the proposed one. The desired sensor value  $\mathbf{s}^{(d)} = H([0 \ 0 \ 0]^T)$  was implicitly defined in the reward function

$$R := \frac{100}{\mathbf{s} - \mathbf{s}^{(d)}}. \quad (38)$$

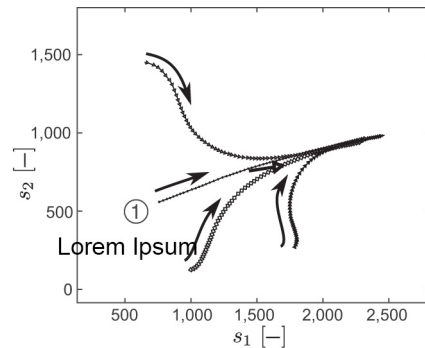
The unknown output equation was the same as in the first simulated environment  $H(q) = H_1(q) = q$ , discount factor  $\gamma = 0.997$ , sample time  $T_s = 0.1$  s and initial state for each training episode  $(x_0, y_0, \theta_0) = (-2, 0.5, \pi/4) + r/10$  where  $r$  is a vector of standard normal distributed random values. The linear speed control input of the simulated system was fixed at 1, while rotation speed control input was controlled by the PPO algorithm. This way, the control inputs used in the assessment

**Figure 9** Sampled points in camera coordinates for the data set



**Notes:** Note that the camera y-axis is inverted. The angle of the arrows indicates the value of  $s_3$  in radians. The empty-filled arrow indicates coordinates of the initial position for stages 1 and 2. The crosses indicate the location of Gaussian kernel centerst

**Figure 10** The feedback-controlled trajectories in sensor space starting from four different points show convergence at the time-axis



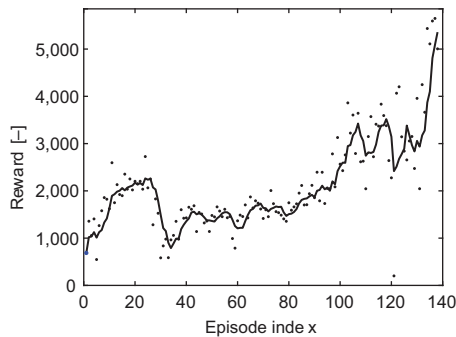
**Notes:** Arrows indicate starting point and direction of travel. The empty-filled arrow indicates coordinates of the initial position for stages 1 and 2. The trajectory marked with ①;1 matches approximately the time axis (28)

of the proposed method, which relies in time-axis control, and in the PPO controller matched more closely.

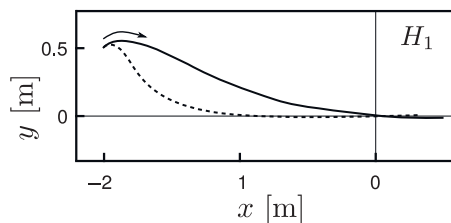
After 138 episodes, training was stopped with an average reward of 5,358 units over the past 5 agents (Figure 11) and a total number of sensor observations of 2,926. The agents from episode 134 to episode 138 were selected to control a unicycle from  $(x_0, y_0, \theta_0) = (-2, 0.5, \pi/4)$ . Figure 12 shows the trajectory of the agent at step 136. The average closest distance to the origin by the last five PPO agents was  $\mu_{PPO} = -0.0783$  m (against  $\mu_{H_1} = -0.0114$  m in our method) units with standard deviation  $\sigma_{PPO} = 0.0878$ . Compared to the proposed method, PPO required more samples to arrive at a controller (2,926 against  $5^3 = 125$  in the proposed method), and yet PPO was only trained for controlling the system from  $(x_0, y_0, \theta_0) = (-2,$



**Figure 11** Evolution of the reward in PPO training for each agent (dots) and for the average of the last five agents (line)



**Figure 12** Trajectory of a trained PPO agent (solid line) and the proposed method (dotted line) for the sensor configuration  $H_1$



0.5,  $\pi/4$ ). Moreover, PPO required that the system was repositioned on the starting state at the beginning of each episode, while the proposed method only requires to be placed at the desired state once and is controllable in all of the region of sensor exploration. The proposed method was shown to be safer and efficient in the sense that it can avoid unexpected exploration in the process of sample collection.

## 5. Conclusion

In this paper, we have proposed a method to learn the sensorimotor mapping of an unknown non-holonomic driftless system and unknown sensor configuration with the purpose of system controllability in a predefined target region. The proposed method consists of two stages. First, we explored the vicinity of the system at the initial state to maximize maneuverability of the system with respect to the sensor signal. Second, we explored the sensor space to construct a mapping from sensor space to chained form. We carried out some simulations and real experiments to show that the trained controller is capable of controlling the system after exploration of the sensor space, therefore validating the method. The results show that the accuracy of the approximation of the mapping from sensor space to chained form and the repeatability of the movements of the robot play a significant role in the performance of the method. Finally, the results were compared against the PPO algorithm, showing that the proposed method requires fewer observations and is safer to deploy in the target environment.

The most important limitations that we have identified are, first, that the controllability region is bounded to the sampled region of the sensor space, although this limitation is not specific to our

method but to function approximation by radial basis functions in general. Second, that learning is performed offline because we could not rely on assumptions that would have enabled online learning because of the generality of the problem requirements (i.e. we do not know the kinematics of the system). Third, the sensor space sampling stage is affected by the curse of dimensionality; hence, this method is not suitable to systems whose state space has a high number of dimensions.

We expect to make further improvements in the future by dropping some of the assumptions. Specifically, this method does not support non-holonomic systems with  $\mathbf{j}_1^{(0)}$  not orthogonal to  $\mathbf{j}_2^{(0)}$ , such as the unicycle system with independently controlled wheels. To overcome this problem, an additional stage prior to Jacobian learning should search for the combination of inputs that maximizes the orthogonality between  $\mathbf{j}_1^{(0)}$  and  $\mathbf{j}_2^{(0)}$ . Furthermore, it seems reasonable to remove backtracking by controlling the system to the origin using linear control but it is not clear yet under which conditions it is possible. More research is needed in these areas to increase the scope of applicability.

## Notes

- 1 <https://cyberbotics.com/doc/guide/pioneer-3dx>
- 2 <https://www.corba.org>
- 3 <http://omniorb.sourceforge.net>

## References

- Amar, K. and Mohamed, S. (2013), “Stabilized feedback control of unicycle mobile robots”, *International Journal of Advanced Robotic Systems*, Vol. 10 No. 4.
- Astolfi, A. (1995), “Exponential stabilization of a car-like vehicle”, *IEEE International Conference on Robotics and Automatics, Nagoya*, pp. 1391-1396.
- Borisov, A.V., Mamaev, I.S. and Bizyaev, I.A. (2016), “Historical and critical review of the development of nonholonomic mechanics: the classical period”, *Regular and Chaotic Dynamics*, Vol. 21 No. 4, pp. 455-476.
- Brockett, R. (1983), “Asymptotic stability and feedback stabilization”, *Differential Geometric Control Theory, Birkhauser*, pp. 181-191.
- Choset, H.M., Hutchinson, S., Lynch, K.M. and Kantor, G. (2005), *Principles of Robot Motion: theory, Algorithms, and Implementation*, MIT press, Cambridge.
- D’Andrea-Novell, B., Bastin, G. and Campion, G. (1991), “Modelling and control of non-holonomic wheeled mobile robots”, *IEEE International Conference on Robotics and Automation, Sacramento, CA*, pp. 1130-1135.
- Dominguez, S., Campoy, P., Sebastian, J.M. and Jimenez, A. (2006), *Control en el Espacio de Estado*, Prentice Hall, Madrid.
- Graefe, V. and Maryniak, A. (1998), “The sensor-control Jacobian as a basis for controlling calibration-free robots”, *IEEE International Symposium on Industrial Electronics, Portoria*, pp. 420-425.
- Jiang, Z.P. and Nijmeijer, H. (1999), “A recursive technique for tracking control of nonholonomic systems in chained form”,

- IEEE Transactions on Automatic Control*, Vol. 44 No. 2, pp. 265-279.
- Kobayashi, Y., Harada, K. and Takagi, K. (2019), "Automatic controller generation based on dependency network of multimodal sensor variables for musculoskeletal robotic arm", *Robotics and Autonomous Systems*, Vol. 118, pp. 55-65.
- Kobayashi, Y., Kurita, E. and Gouko, M. (2013), "Integration of multiple sensor spaces with limited sensing range and redundancy", *International Journal of Robotics and Automation*, Vol. 28 No. 1.
- Kolmanovsky, I. and Mcclamroch, N.H. (1995), "Developments in nonholonomic control problems", *IEEE Control Systems*, Vol. 15 No. 6, pp. 20-36.
- Kondor, R. (2004), "Regression by linear combination of basis functions", unpublished manuscript.
- Lefeber, E., Robertsson, A. and Nijmeijer, H. (2000), "Linear controllers for exponential tracking of systems in chained form", *International Journal of Robust and Nonlinear Control*, Vol. 10 No. 4, pp. 243-263.
- Lefeber, E., Robertsson, A. and Nijmeijer, H. (2004), "Linear controllers for tracking chained-form systems", *Lecture Notes in Control and Information Sciences*, Springer, London, pp. 183-199.
- Luo, J. and Tsiotras, P. (2000), "Control design for chained-form systems with bounded inputs", *Systems & Control Letters*, Vol. 39 No. 2, pp. 123-131.
- Miller, W.T. (1987), "Sensor-based control of robotic manipulators using a general learning algorithm", *IEEE Journal on Robotics and Automation*, Vol. 3 No. 2, pp. 157-165.
- Murray, R.M. and Sastry, S.S. (1991), "Steering nonholonomic systems in chained form", *IEEE Conference on Decision and Control, Brighton*, pp. 1121-1126.
- Navarro-Alarcon, D., Cherubini, A. and Li, X. (2019), "On model adaptation for sensorimotor control of robots", *IEEE Chinese Control Conference, Guangzhou*, pp. 2548-2552.
- Rifford, L. (2008), "Stabilization problem for nonholonomic control systems", *Geometric Control and Nonsmooth Analysis, Series on Advances in Mathematics for Applied Sciences*, World Scientific, pp. 260-269, doi: [10.1142/9789812776075\\_0015](https://doi.org/10.1142/9789812776075_0015), available at: [https://www.worldscientific.com/doi/abs/10.1142/9789812776075\\_0015](https://www.worldscientific.com/doi/abs/10.1142/9789812776075_0015)

- Sampei, M. (1994), "A control strategy for a class of nonholonomic systems - time-state control form and its application", *IEEE Conference on Decision and Control, FL*, pp. 1120-1121.
- Sampei, M., Kiyota, H., Koga, M. and Suzuki, M. (1996), "Necessary and sufficient conditions for transformation of nonholonomic system into time-state control form", *IEEE Conference on Decision and Control, Kobe*, pp. 4745-4746.
- Schulman, J., Wolski, F., Dhariwal, P., Radford, A. and Klimov, O. (2017), "Proximal policy optimization algorithms", arXiv:1707.06347.
- Smart, W.D. and Kaelbling, L.P. (2002), "Effective reinforcement learning for mobile robots", *IEEE International Conference on Robotics and Automation*, Vol. 4, pp. 3404-3410.

### Further reading

- Masuda, N. and Ushio, T. (2017), "Control of nonholonomic vehicle system using hierarchical deep reinforcement learning", *International Symposium on Nonlinear Theory and Its Applications*, pp. 26-29.

### Algorithms

Algorithm 1 Pseudo code for sampling the sensor space.  $u_{(\psi)}$  and  $u_{(s^*)}$  are the corresponding inputs obtained in Section 3.1.

```

loop  $\Delta t_1 : 0 \dots \Delta T_1$ 
  apply input  $u_{(3)}$  for  $\Delta t_1$ ;
  loop  $\Delta t_2 : 0 \dots \Delta T_2$ 
    apply input  $u_{(\psi)}$  for  $\Delta t_2$ ;
    loop  $\Delta t_3 : 0 \dots \Delta T_3$ 
      apply input  $u_{(s^*)}$  for  $\Delta t_3$ ;
      dataset  $\rightarrow$  dataset  $\cup (s, (u_{(3)}; u_{(\psi)}; u_{(s^*)}))$ 
      backtrack  $u_{(s^*)}$ ;
      backtrack  $u_{(\psi)}$ ;
      backtrack  $u_{(3)}$ ;

```

### Corresponding author

Francisco Jesús Arjonilla García can be contacted at: [arjonilla.francisco@shizuoka.ac.jp](mailto:arjonilla.francisco@shizuoka.ac.jp)