国立大学法人 静岡大学
National University Corporation
Shizuoka University

Graduate School of
Science and Technology

PhD Thesis Presentation

Control of Non-Holonomic Driftless System with Unknown Sensorimotor Model by Jacobian Estimation

Francisco J. (Paco) ARJONILLA GARCÍA

Supervisor: Prof. Yuichi KOBAYASHI

10th of February, 2021

1.1 Motivation

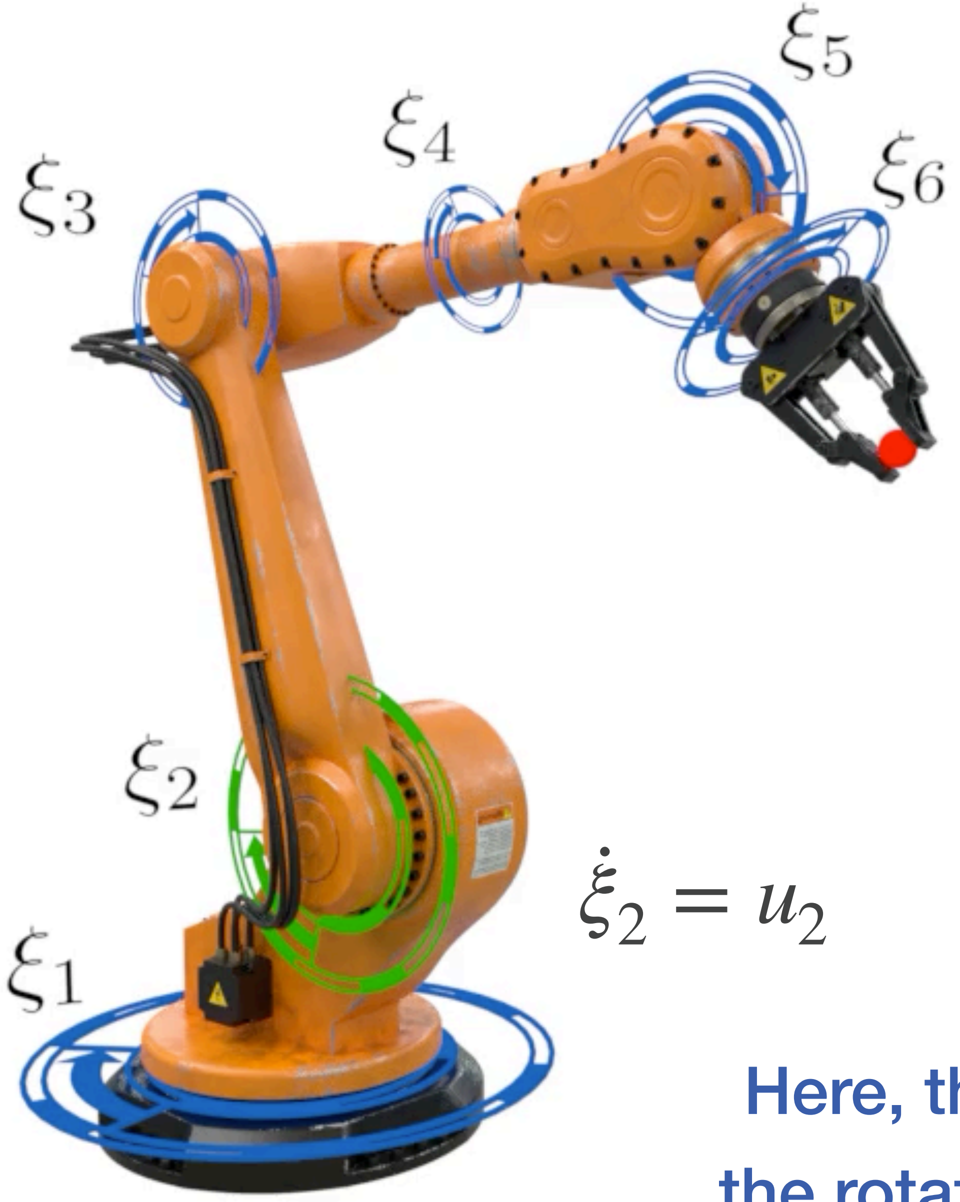
Unknown sensors
e.g. uncalibrated robots

Unknown kinematics
e.g. damaged robots

Unknown environment
e.g. relocation of external sensors

Can the process to obtain a controller under these circumstances be automated?

1.2 Joint coordinates



There are many ways of expressing the state of a system.

For example, the joint angles ξ_i in an industrial manipulator with six degrees of freedom.

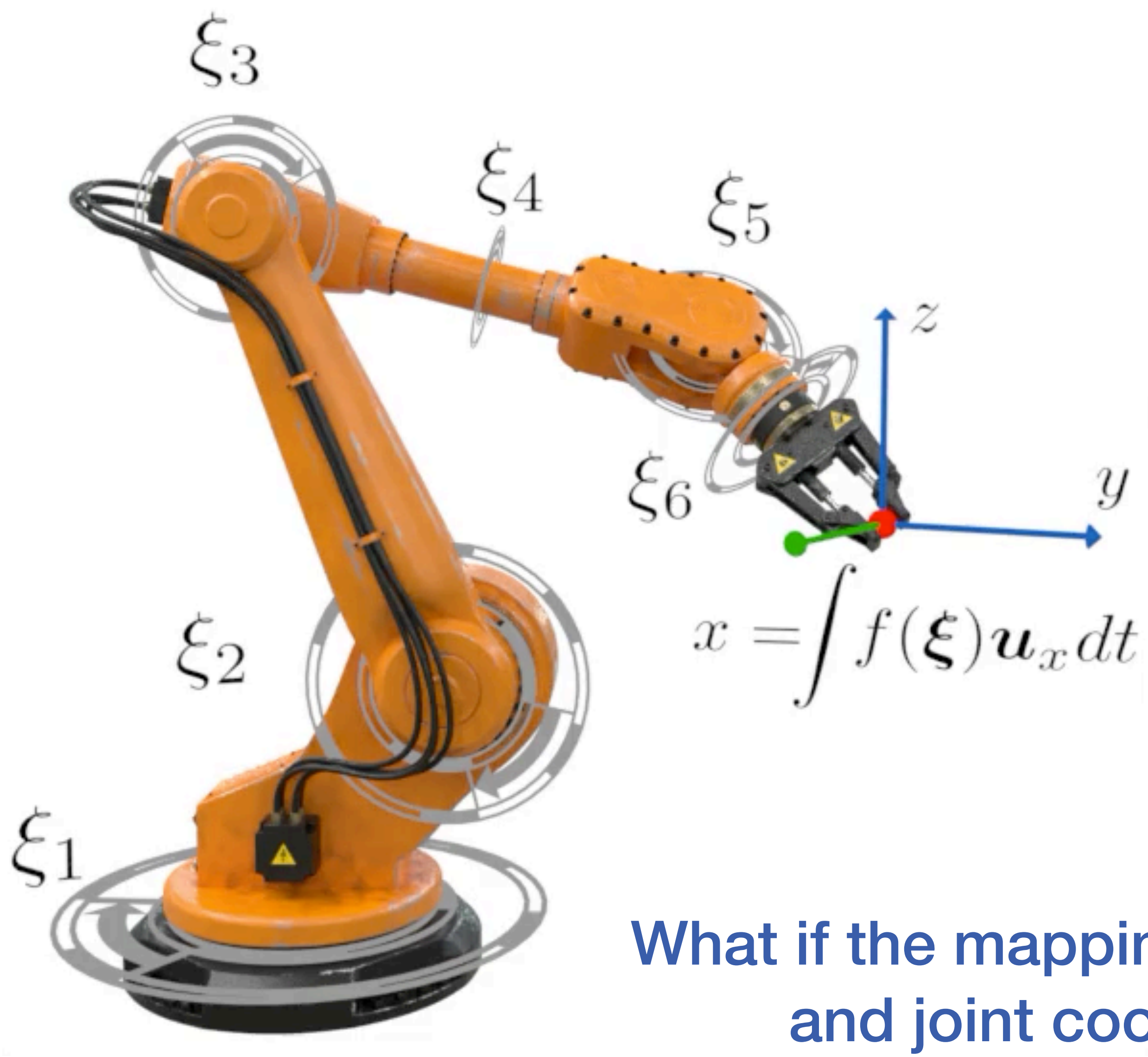
The values q_i chosen to represent the state are called generalized coordinates.

Here, the control input u_2 controls the rotational speed of joint $q_2 = \xi_2$

1.2 Cartesian and Euler coordinates

Another set of generalized coordinates is

$$(x, y, z, \theta_x, \theta_y, \theta_z).$$



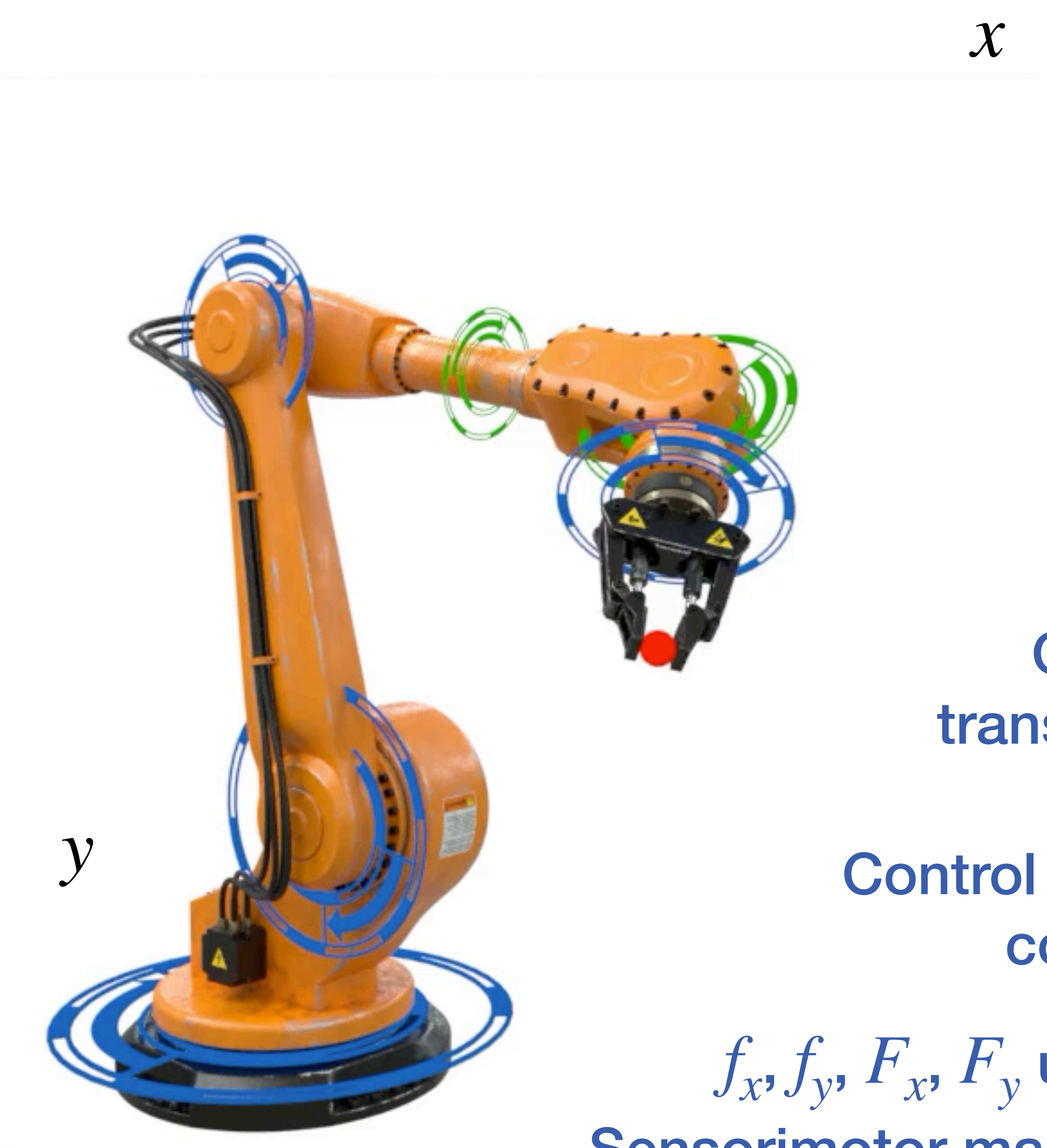
Movement is restricted here to cartesian axis x.

All joints ξ_i participate

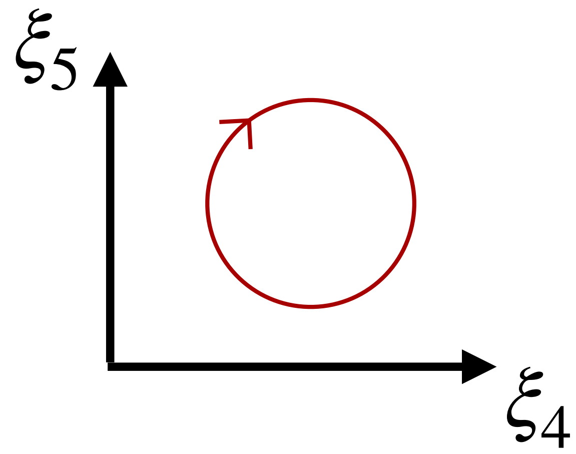
Converting between joint coordinates and these coordinates is called **forward** and **inverse kinematics**

What if the mapping from control coordinates and joint coordinates is unknown?

1.2 Camera coordinates



Here, 2 joint coordinates are controlled.



Coordinate transformation

$$x = f_x(\xi_4, \xi_5)$$
$$y = f_y(\xi_4, \xi_5)$$

Control of camera coordinates

$$\dot{x} = F_x(\xi_4, \xi_5)\boldsymbol{u}$$
$$\dot{y} = F_y(\xi_4, \xi_5)\boldsymbol{u}$$

f_x, f_y, F_x, F_y unknown:

Sensorimotor mapping problem

1.2 Related researches

Automatic sensorimotor mapping:

- Thomas Miller (1987) ***Sensor-Based** Control of Robotic Manipulators Using a **General Learning Algorithm**.* IEEE J. Robot. Autom., 3(2) pp.157-165.

- David Pierce and Benjamin J. Kuipers (1997) *Map Learning with **Uninterpreted Sensors and Effectors**.* Artificial Intelligence, 92(1-2) pp. 169-227.

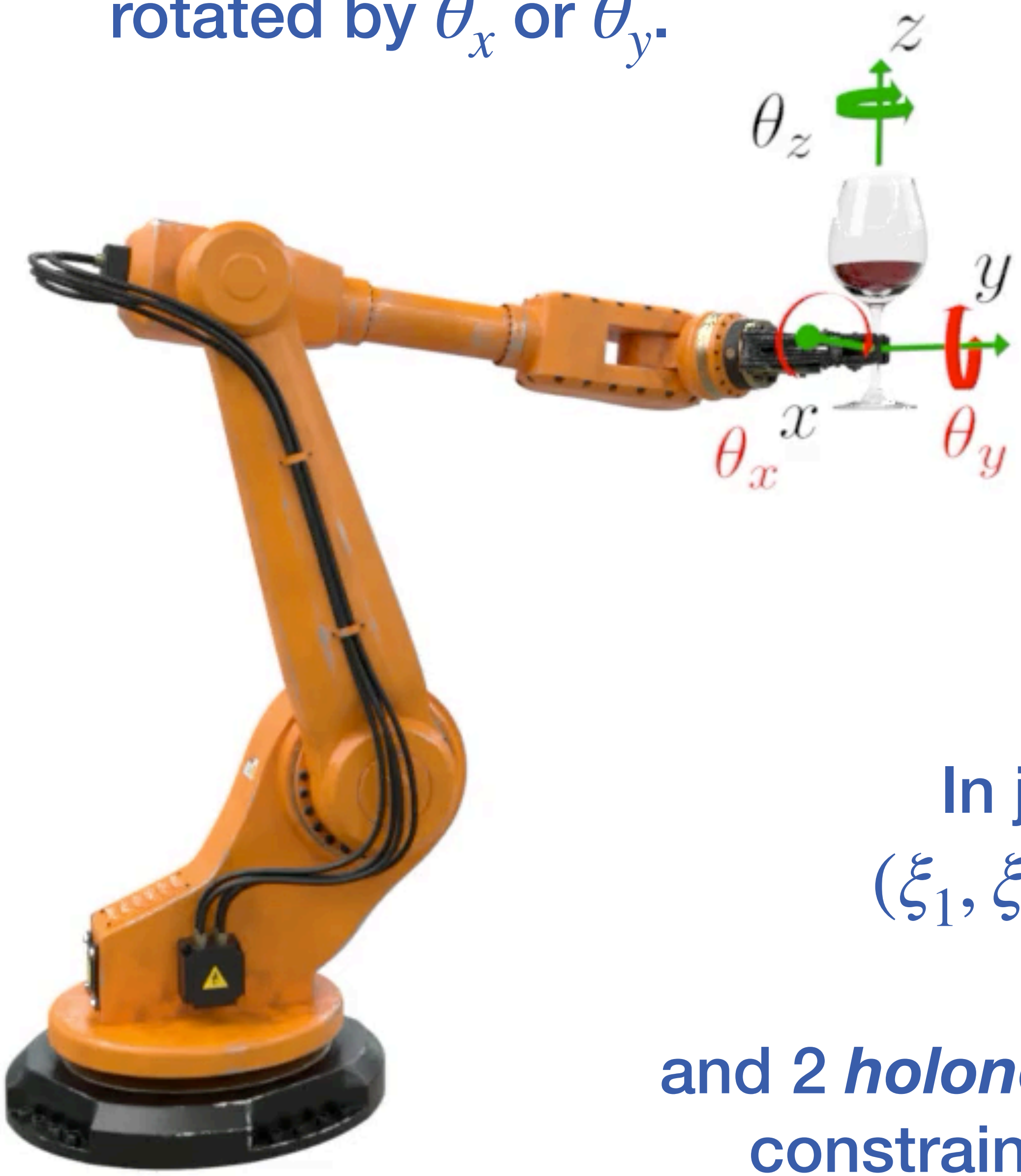
- Jonathan Mugan (2005) *Robot Learning: A Sampling of Methods*. Technical report

- David Navarro-Alarcon, Andrea Cherubini, and Xiang Li (2019) *On **Model Adaptation** for Sensorimotor Control of Robots.* In Proc. 38th Chinese Control Conf., pp. 2548-2552.

These methods are for **holonomic systems** only.

1.3 Constraints

The glass of wine must not be rotated by θ_x or θ_y.



Therefore, there are four degrees of freedom (DoF): (x, y, z, θ_z)

The states where θ_x or θ_y change are never reached.

In joint coordinates, $(\xi_1, \xi_2, \xi_3, \xi_4, \xi_5, \xi_6) =: \boldsymbol{\xi}$

and 2 ***holonomic*** constraints

$$f_{\theta_x}(\boldsymbol{\xi}) = 0$$

$$f_{\theta_y}(\boldsymbol{\xi}) = 0$$

1.3 Non-holonomic constraints

Object slips on rotation of one axis

All states are reachable: The system has six Degrees of Freedom (DoF)

But one control input is redundant: It can be removed

In joint coordinates,

$$(\xi_1, \xi_2, \xi_3, \xi_4, \xi_5, \xi_6) =: \boldsymbol{\xi}$$

But no reduction in number of DoF

<u>Non-holonomic constraint</u>

$C(q)\dot{q} = 0$ (Pfaffian form)

Control is difficult

1.3 Related researches

Control of non-holonomic systems with **known sensorimotor mapping**

- R.W. Brocket (1983) *Asymptotic stability and feedback stabilization.* In Differential Geometry Control Theory, pp.181-191.

- M. Galicki (2017) *The planning of optimal motions of non-holonomic systems.* Nonlinear dynamics, 90(3) pp. 2163-2184.

- A. Censi and R. M. Murray (2015) ***Bootstrapping** bilinear models of simple vehicles.* International Journal of Robotics Research, 34(8) pp. 1087-1113.

- I. Goral and K. Tchon (2017) *Lagrangian **Jacobian** motion planning: a parametric approach.* Journal of Intelligent Robotic Systems, 85 pp. 511-522.

These methods assume that **the Jacobian is known** beforehand.

1.4 Research question

- ✓ **Unknown sensor configuration problem:** The generalized coordinates corresponding to sensor observations are unknown.
- ✓ **Unknown sensorimotor mapping problem:** The kinematics of the system are unknown as well — The relation between control inputs and sensor values is unknown.

➡ How to enable control of **non-holonomic systems** with **unknown sensorimotor mapping**?

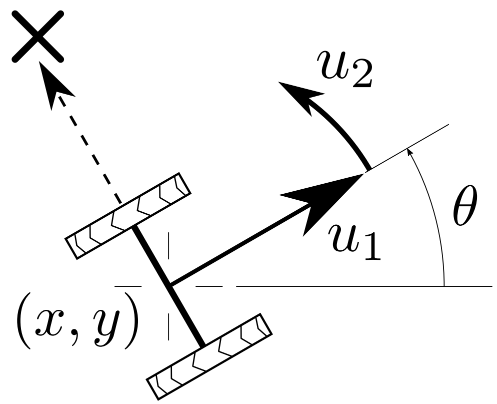
- ➡ Benefits:
  - ✦ Increment flexibility of controllers.
  - ✦ Simplify deployment of robotics.
  - ✦ Improve resilience in damaged robots.

1.4 Problem setting

- Consider the equations of a dynamic affine system:

State equation	$\dot{\boldsymbol{q}} = F(\boldsymbol{q})\boldsymbol{u}$	where	$\boldsymbol{u}$: Control inputs
			$\boldsymbol{q}$: Arbitrary coordinates
Output equation	$\boldsymbol{s} = H(\boldsymbol{q})$		$\boldsymbol{s}$: Sensor observations

- with non-holonomic constraints in Pfaffian form: $C(\boldsymbol{q})\dot{\boldsymbol{q}} = 0$
- with unknown kinematics $F(\boldsymbol{q}) = ?$
- with unknown sensor configuration $H(\boldsymbol{q}) = ?$
- and only sensor observation vector $\boldsymbol{s}$ is accessible
- Application to the unicycle



$x?$
$y?$
$\theta?$

1.4 Objective

- The objective is to **find a control law** φ such that

$$\boldsymbol{u} = \varphi(\boldsymbol{s})$$

realizes a desired sensor value of $\boldsymbol{s}^{(d)}$ under the conditions of **unknown sensorimotor mapping in a non-holonomic system**:

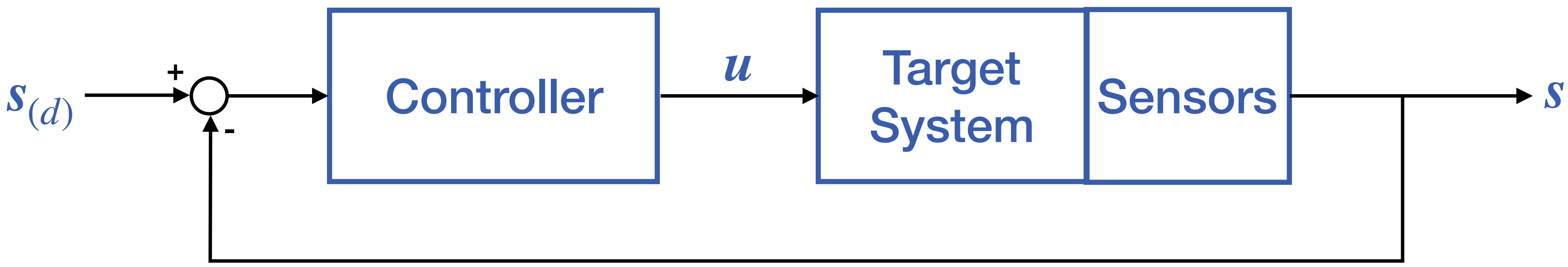
$$\dot{\boldsymbol{s}} = ?(\boldsymbol{u})$$

$\boldsymbol{u}$: Control inputs
$\boldsymbol{s}$: Sensor observations

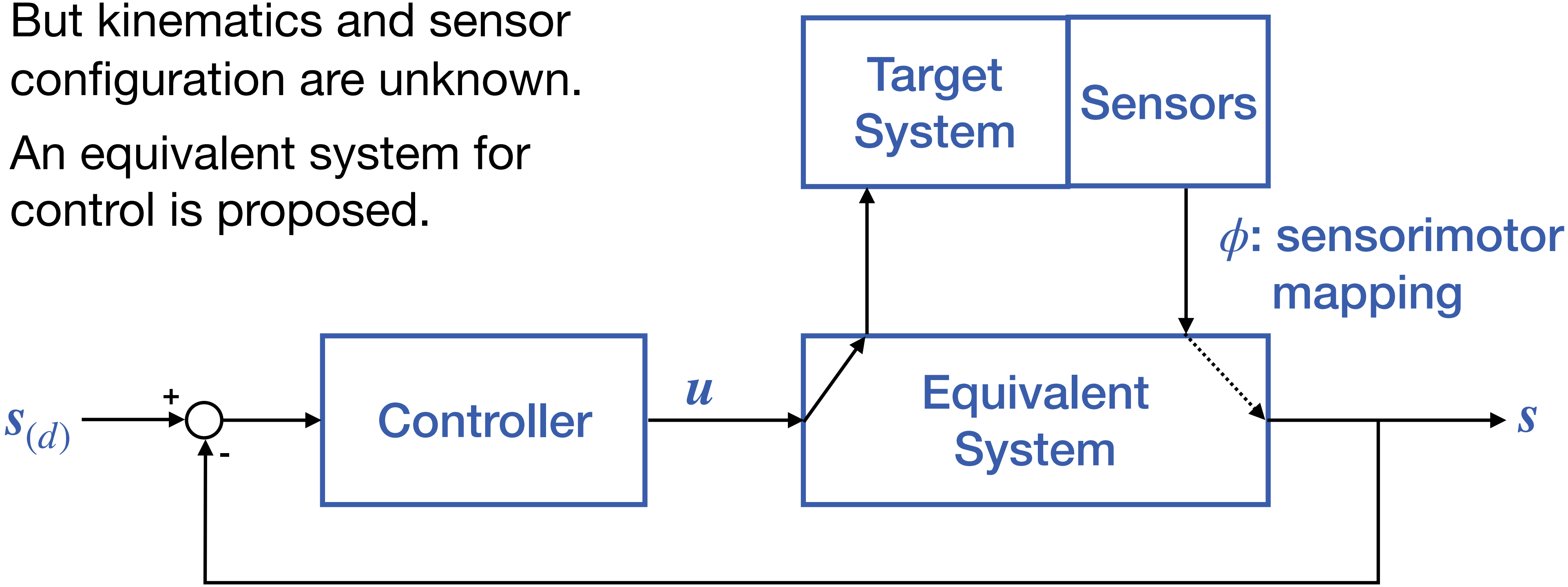
- The following assumptions hold:
  - The system has **one non-holonomic constraint.**
  - The control **input has two components:** $\boldsymbol{u} = [u_1 \quad u_2]^\top$
  - The unknown output function $\boldsymbol{s} = H(\boldsymbol{q})$ is **isomorphic.**
  - One of the inputs *rotates* the system around itself, *i.e.* the controlled subspace by the other input changes with the former.
- Use the unicycle as the target system.

2.1 Overview of the proposed approach

- Feedback controller with known kinematics and sensors:



- But kinematics and sensor configuration are unknown.
- An equivalent system for control is proposed.

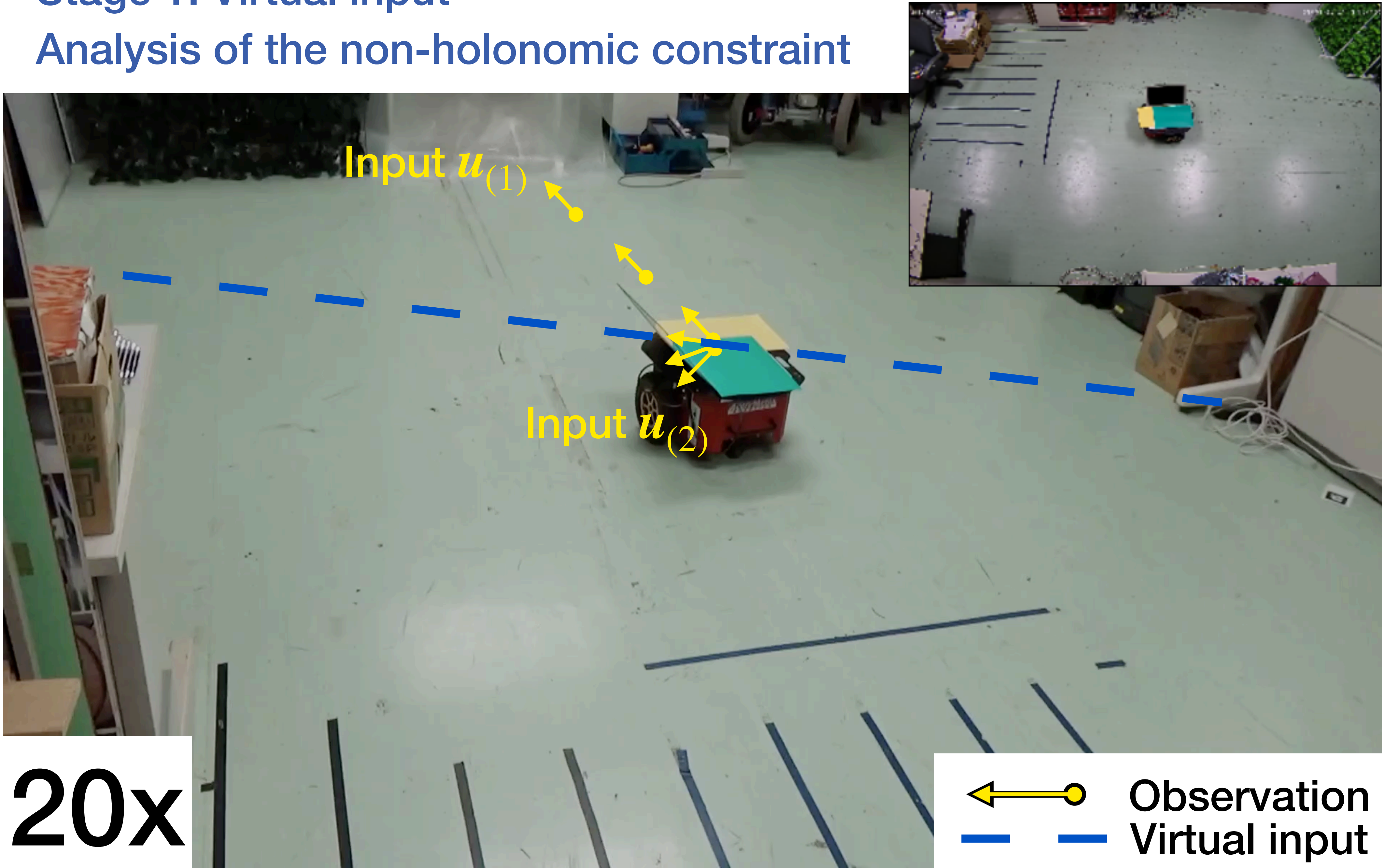


- Same control input, transformed sensor coordinates

2.1 Overview of the proposed approach

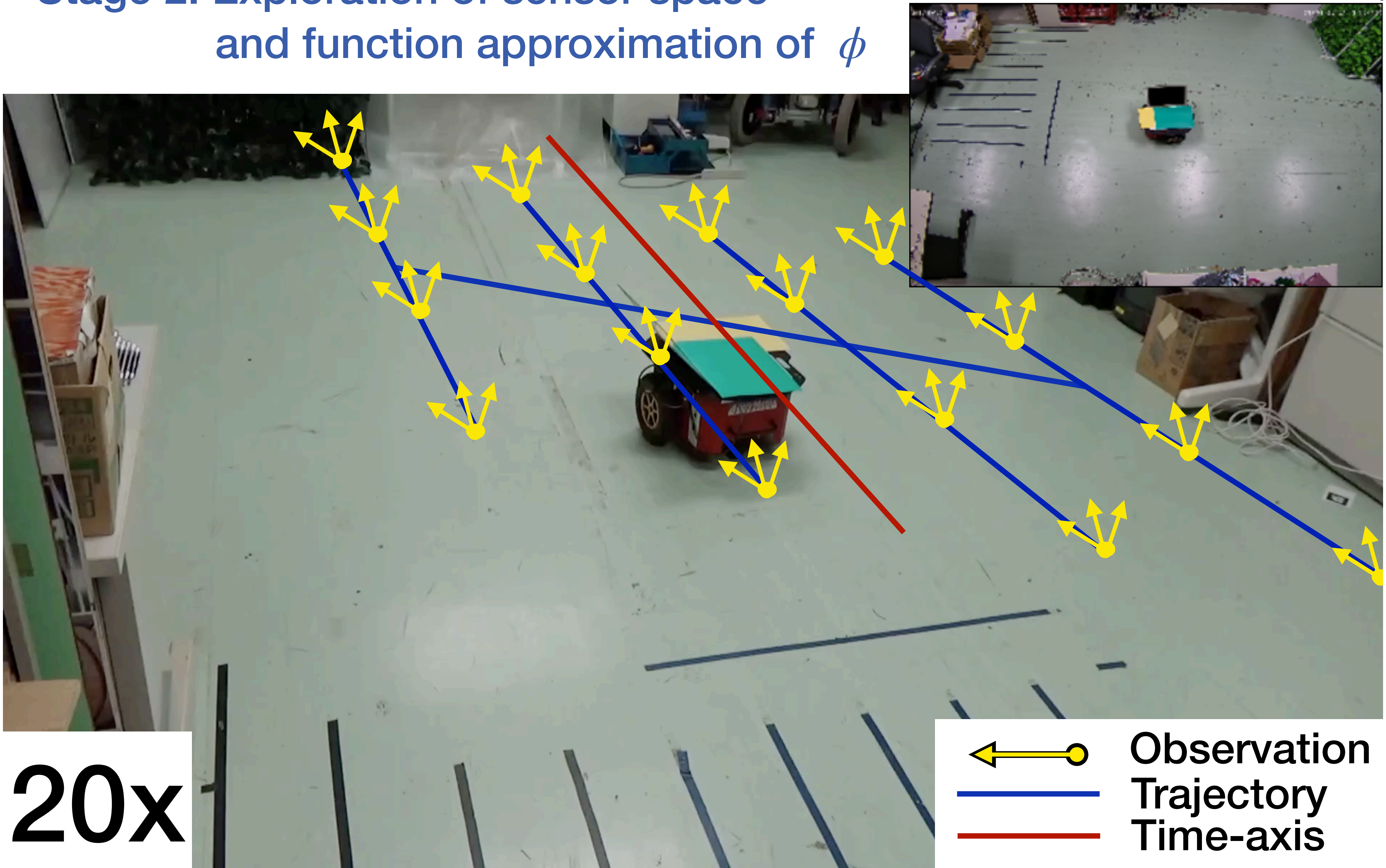
Stage 1: Virtual input

Analysis of the non-holonomic constraint



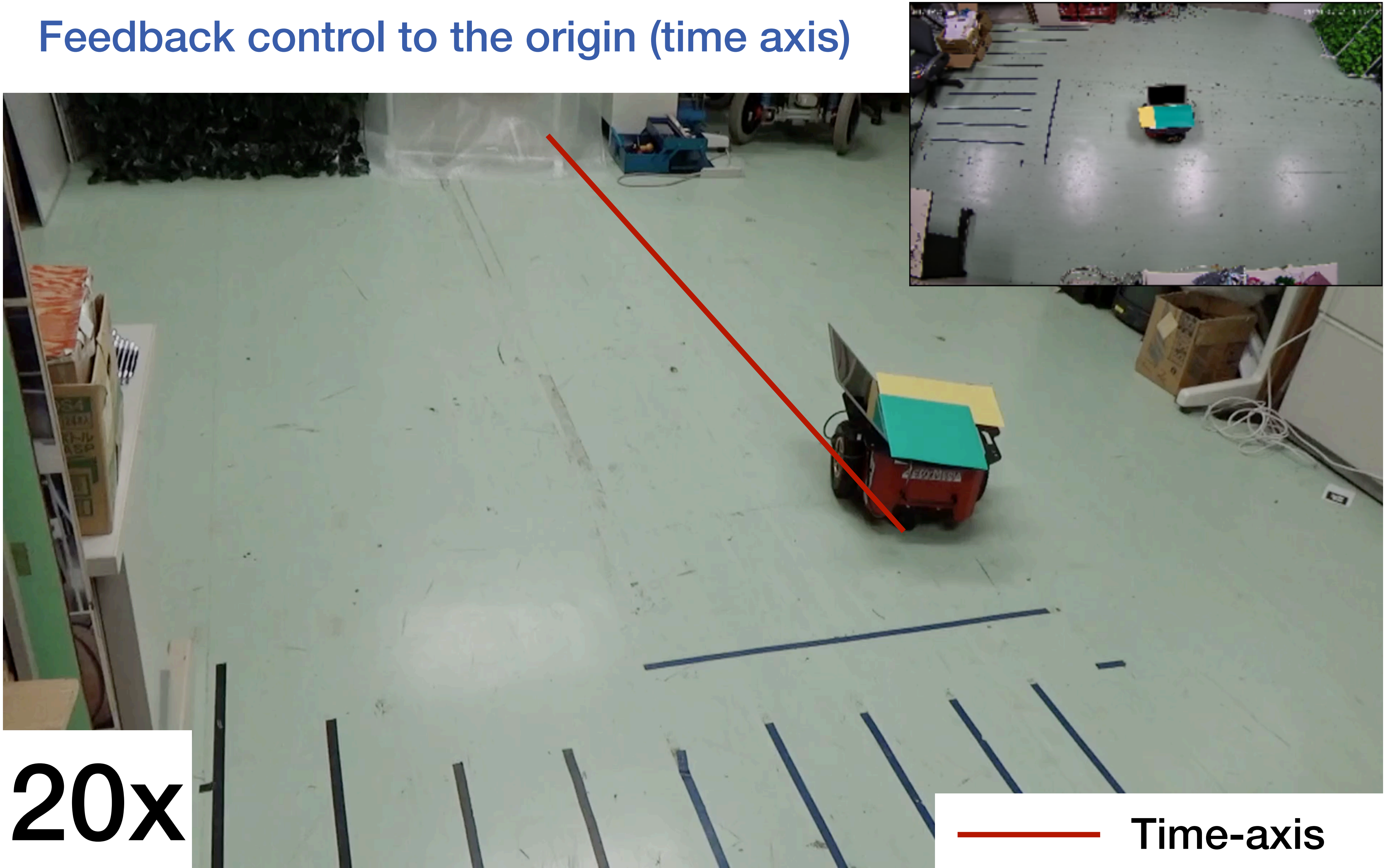
2.1 Overview of the proposed approach

Stage 2: Exploration of sensor space and function approximation of ϕ

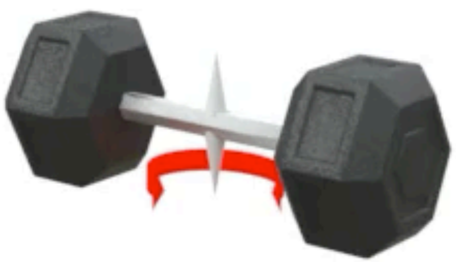


Evaluation

Feedback control to the origin (time axis)



2.2 Virtual input



The non-holonomic constraint defines a forbidden direction

We can construct a composite ***virtual input*** with a similar effect

With equivalent effect to moving along the constrained dimensions

Approximation to a holonomic system

2.2 Jacobians

Jacobian: Change of sensor (camera) coordinates with respect to control inputs

Jacobian elements

$$j_4 = \boldsymbol{u}_{(4)}\Delta t$$
$$j_5 = \boldsymbol{u}_{(5)}\Delta t$$

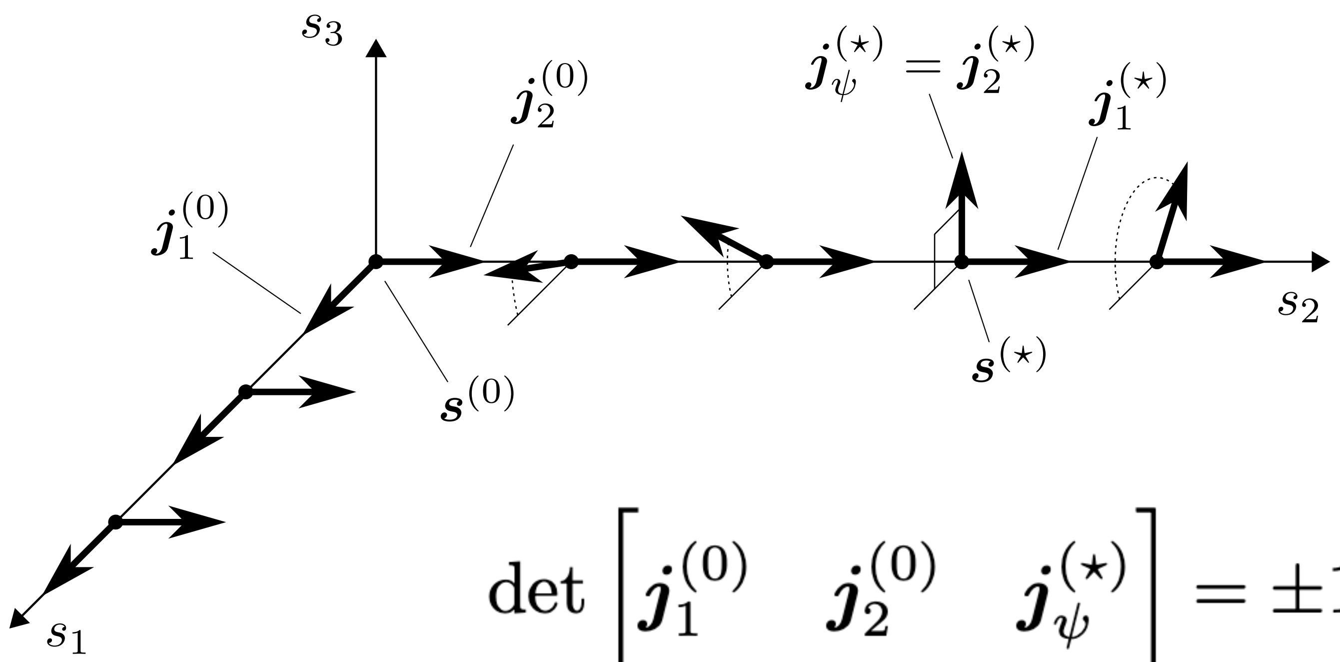
Jacobian matrix

$$J = \frac{\Delta z}{\boldsymbol{u}\Delta t}\bigg|_{n \times m}$$

J enables linear controllability

$$\dot{z} = J\boldsymbol{u}$$

2.2 Stage 1: Virtual input

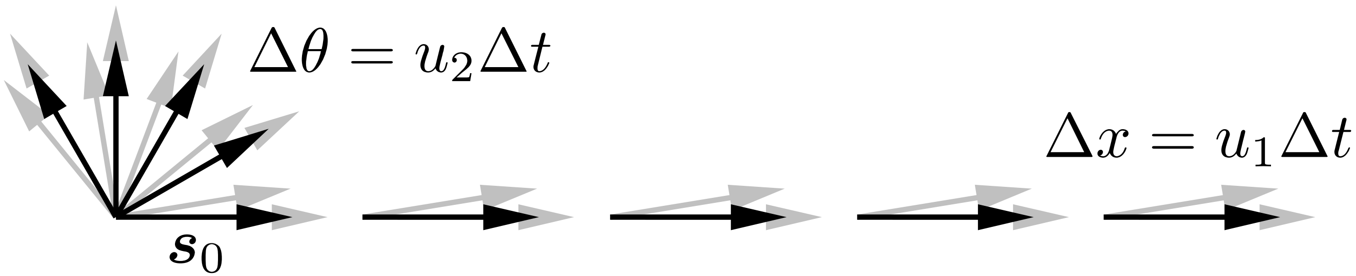


- Obtain the state $\boldsymbol{s}^{(\star)}$ that maximizes $\boldsymbol{j}_\psi^{(\star)}$ in the forbidden direction at $\boldsymbol{s}_0$.

$$\det \begin{bmatrix} \boldsymbol{j}_1^{(0)} & \boldsymbol{j}_2^{(0)} & \boldsymbol{j}_\psi^{(\star)} \end{bmatrix} = \pm 1$$

- The virtual input is the sequence $u_{(3)} \equiv \left(u_\psi \Delta t^{(\star)}, u_{\overline{\psi}} \Delta t_d, -u_\psi \Delta t^{(\star)} \right)$

- For the unicycle, in (x, y, θ) space, the sample points are:



2.3 Stage 2: Exploration of sensor space

- An equivalent linear system is considered with state equation:

$$\dot{z} = g_1(z)u_1 + g_2u_2 \qquad \text{such that} \qquad g_1(z) = \begin{bmatrix} 1 \\ 0 \\ z_2 \end{bmatrix} \qquad \text{and} \qquad g_2(z) = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

- This choice of generalized coordinates z is called chained form.
- Chained form is the preferred method for control of non-holonomic systems in the literature.
  (Murray & Sastry, 1991) (Jiang, 1999) (Lefeber et at. 2000, 2004) (Luo & Tsiopras, 2000)

- During sensor exploration, the same control input is applied to this system and to the target system.
- Sensor values s and equivalent state z are recorded in pairs.
- The coordinate transformation ϕ is calculated by function approximation, such that

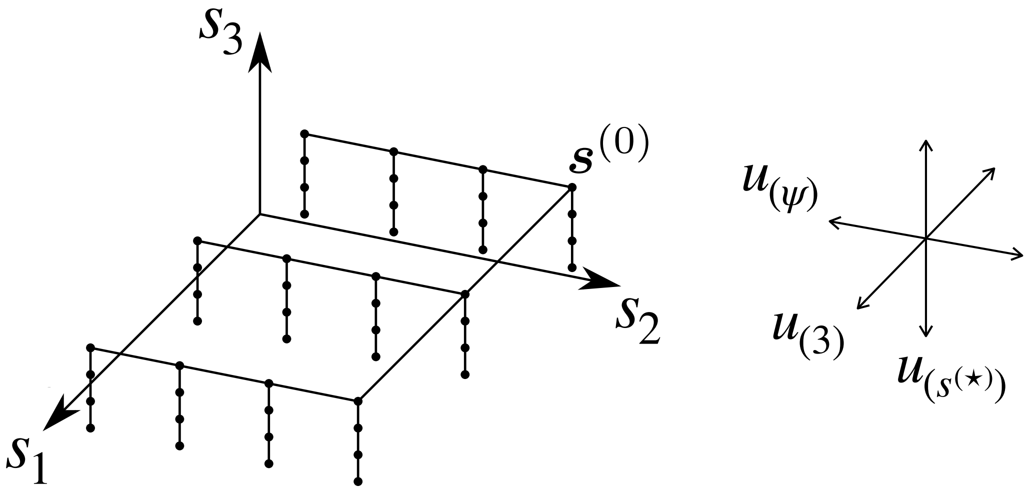
$$z \simeq \phi(s)$$

2.3 Fixed-pattern path

- The path followed (sequence of control inputs) is the same in the target system and in the equivalent system:

loop $\Delta t_1 : 0 \ldots \Delta T_1$
  apply input $\boldsymbol{u}_{(3)}$ for Δt_1;
  loop $\Delta t_2 : 0 \ldots \Delta T_2$
    apply input $\boldsymbol{u}_{(\psi)}$ for Δt_2;
    loop $\Delta t_3 : 0 \ldots \Delta T_3$
      apply input $\boldsymbol{u}_{(\boldsymbol{s}^{(\star)})}$ for Δt_3;
      dataset $\leftarrow$ dataset $\cup$ $(\boldsymbol{s}, (\boldsymbol{u}_{(3)}; \boldsymbol{u}_{(\psi)}; \boldsymbol{u}_{(\boldsymbol{s}^{(\star)})}))$;
      backtrack $\boldsymbol{u}_{(\boldsymbol{s}^{(\star)})}$;
    backtrack $\boldsymbol{u}_{(\psi)}$;
  backtrack $\boldsymbol{u}_{(3)}$;

- Example trajectory in chained form coordinates:



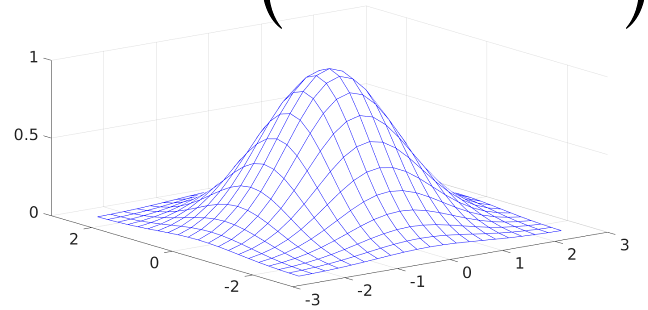
2.3 Mapping approximation

- The dataset $\{(s_i, z_i)\}$ is fed to Gaussian RBF function approximation to obtain the mapping ϕ from sensor space to chained form space.

- Radial Basis Functions (RBF) repeat one function Φ (the kernel) in several places $\boldsymbol{b}_k$ (the bases)

$$\phi(\boldsymbol{s}) = \sum_{k=1}^{P} \theta\Phi(\boldsymbol{s})$$

$$\Phi_k(\boldsymbol{s}) := \exp\left(-\frac{\left\|\boldsymbol{s}-\boldsymbol{b}_k\right\|^2}{2\sigma^2}\right)$$



- The approximation target is to minimize the error in $E = \Sigma(z_i - \Phi(\boldsymbol{s}_i))$

- Solution by Least Mean Squares with regularization term λ (One shot learning)

$$\boldsymbol{\theta} = (Q^\intercal Q - \lambda I)^{-1} Q^\intercal \boldsymbol{z}$$

$$Q(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_N) := \begin{bmatrix} \Phi_0(\boldsymbol{s}_1) & \cdots & \Phi_P(\boldsymbol{s}_1) \\ \vdots & \ddots & \vdots \\ \Phi_0(\boldsymbol{s}_N) & \cdots & \Phi_P(\boldsymbol{s}_N) \end{bmatrix}$$

- The resulting ϕ is the sensorimotor mapping to the equivalent system.

$$\dot{z} = \boldsymbol{g}_1(\phi(\boldsymbol{s}))u_1 + \boldsymbol{g}_2 u_2$$

2.4 Stage 3: Evaluation

- Evaluation of the method is performed by assessing controllability of the target system.
- Time-axis control was chosen due to its simplicity: It is like activating cruise control in a car. Only steering is left to the driver.
- Coordinates in chained form are renamed $(\tau, \zeta_2, \zeta_3) := (z_1, z_2, z_3)$

$$\frac{\tau}{dt} = u_1 \qquad\qquad \frac{d}{d\tau}\begin{bmatrix}\zeta_3\\ \zeta_2\end{bmatrix} = \begin{bmatrix}\zeta_2\\ 0\end{bmatrix} + \begin{bmatrix}0\\ 1\end{bmatrix} u_2$$

  - Time-control part
  - State-control part

- τ is the **time scale** of the state-control part.
- **Time axis** is the one dimensional subspace in chained form corresponding to $(0,0)$ in the state-control part.
- Strategy: **Fix** $u_1 = 1$ and **control** u_2 with feedback control.
- Switch the sign of u_1 if $\tau < 0$ to reach the origin in chained form.

2.5 Summary of the method

- ➡ Stage 1: Design a **virtual input**
  - ✓ By analyzing the Jacobian elements in the neighborhood of the initial position.
- ➡ Stage 2: **Approximate sensor observations** to a controllable **equivalent system**.
  - ✓ The path for sampling the sensor space follows a fixed-pattern.
  - ✓ Same path is followed in real and equivalent systems.
  - ✓ Sensor observations are mapped to coordinates in the equivalent system by function approximation (by Gaussian RBF).
- ➡ Evaluation: **Control the equivalent system** to the origin (time-axis)
  - ✓ The control inputs are applied to the target system.
  - ✓ Sensor observations are mapped to coordinates in chained form.

3.1 Simulation

- The method was evaluated in simulation of unicycle on Matlab
- Three instances of unknown sensor mapping

$$\dot{\boldsymbol{q}} = F(\boldsymbol{q})\boldsymbol{u}$$

$$\boldsymbol{s} = H(\boldsymbol{q})$$

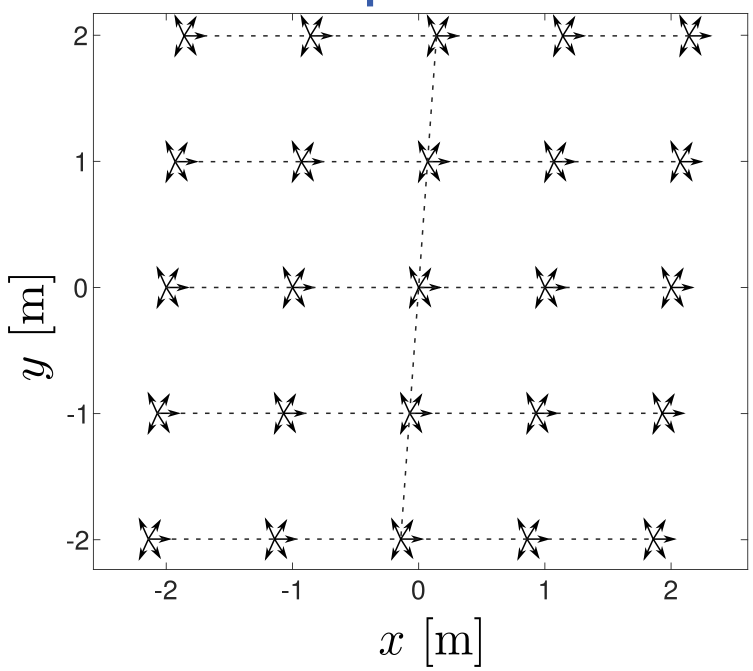
$$H_1(q) = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \qquad H_2(q) = \begin{bmatrix} \sinh(y) \\ e^x \\ \arctan(\theta) \end{bmatrix} \qquad H_3(q) = \begin{bmatrix} x + e^y \\ e^x - y \\ \theta^3 \end{bmatrix}$$

- In all cases, $\boldsymbol{s}_d = (0,0,0)$
  - The choice of $\boldsymbol{s}_d$ is standard in control engineering, because
  - any other $\boldsymbol{s}'_d$ is possible by a simple transformation $\boldsymbol{s}'_d = \boldsymbol{s} - \boldsymbol{s}_d$
- Virtual input stage: 9 Jacobian samples per axis
- Sensor sampling stage: 5 samples per axis (total 5^3)

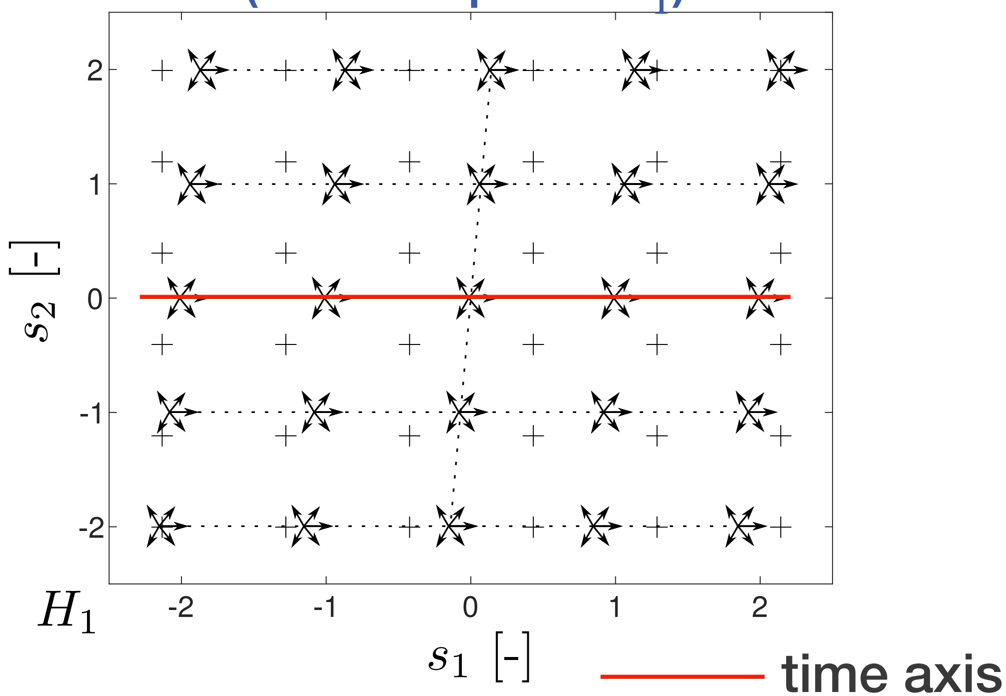
3.1 Simulation: H_1

- ➡ Stage 1: Virtual input $\boldsymbol{u}_{(3)}$ slightly off from ideal trajectory $x = 0$
- ➡ Stage 2: Observation points evenly distributed between kernels ($\sigma = 1.1970$) of function approximation
  - ✓ max $E_{\phi_3} = 0 \pm 0.014$m (Error of function approximation at sensor observations)

Trajectory in cartesian space



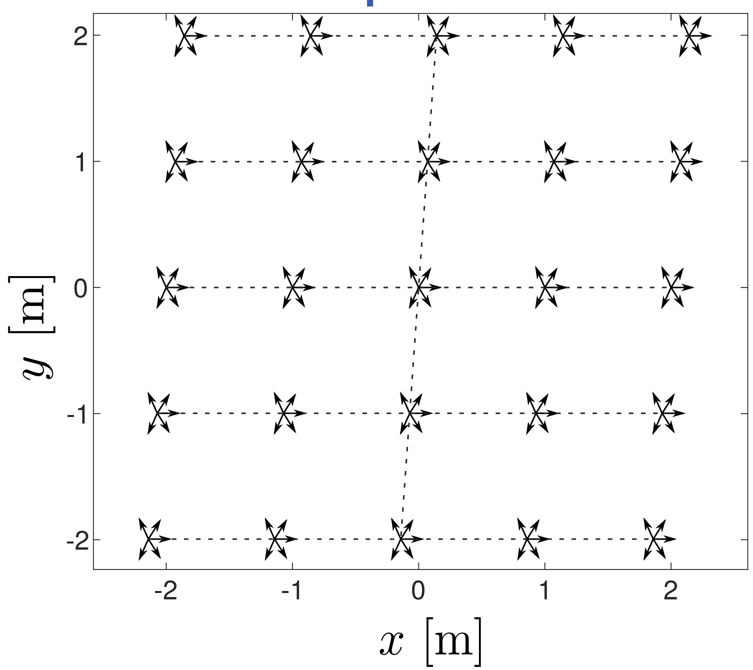
Bases, time-axis, observations (sensor space H_1)



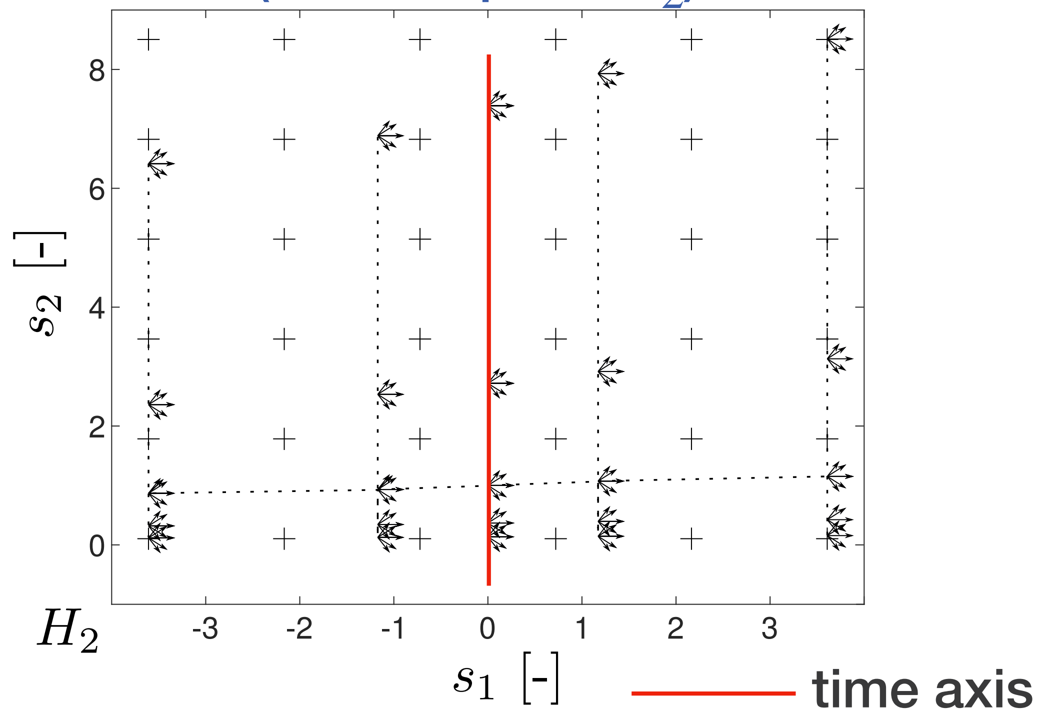
3.1 Simulation: H_2

- Stage 1: Same virtual input $\boldsymbol{u}_{(3)}$ as in H_1
- Stage 2: Some observations were concentrated between few kernels ($\sigma = 0.6643$)
  - ✓ max $E_{\phi_3} = 0 \pm 0.076$m (Error of function approximation) 550% increase with respect to H_1

Trajectory in cartesian space



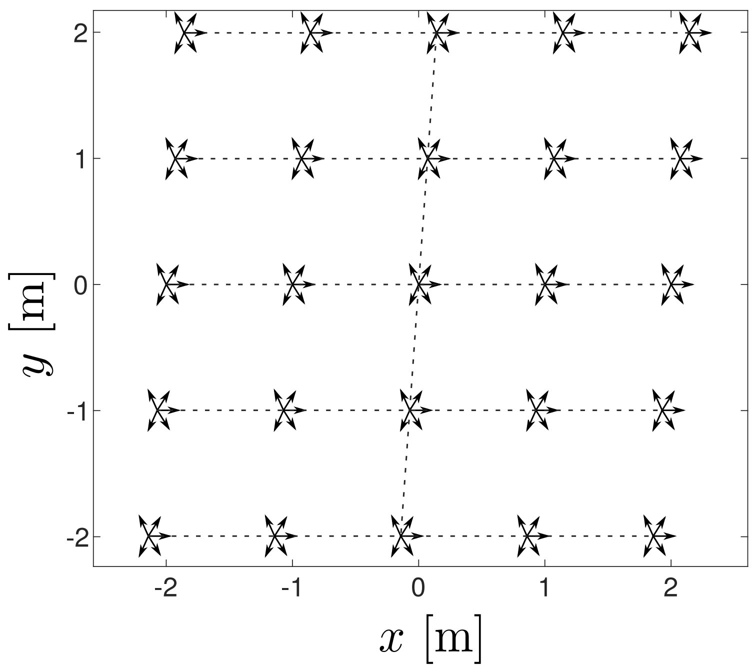
Bases, time-axis, observations (sensor space H_2)



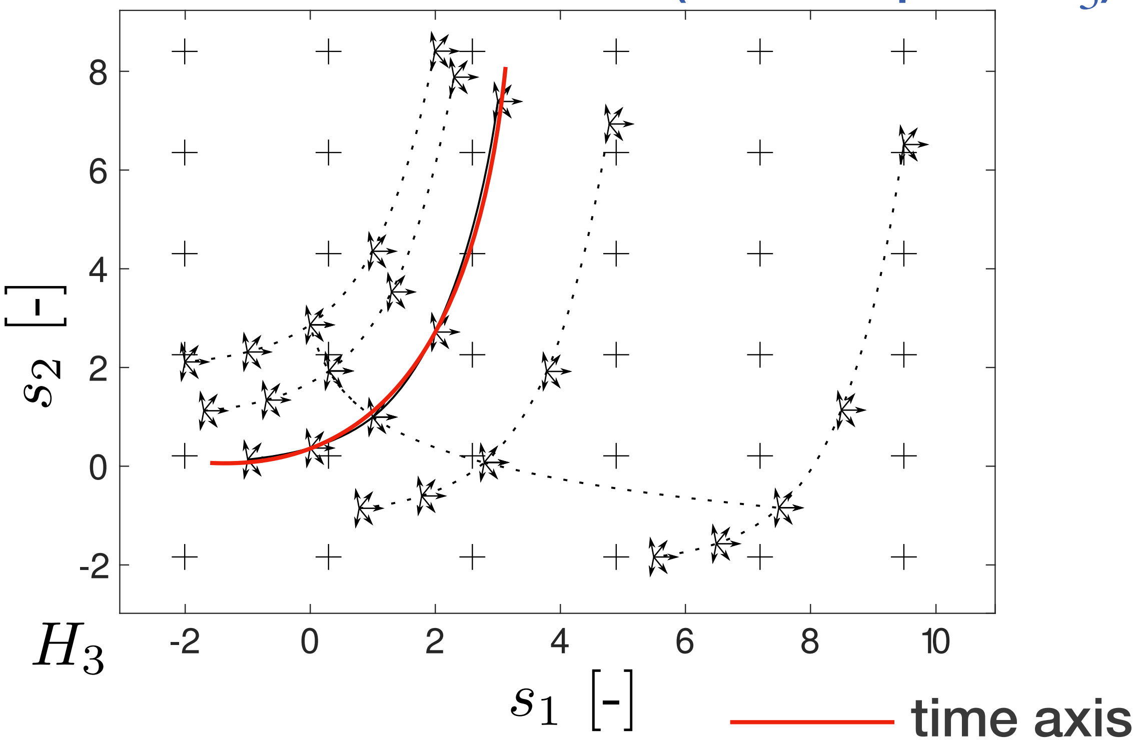
3.1 Simulation: H_3

- ➡ Stage 1: Same virtual input $\boldsymbol{u}_{(3)}$ as in H_1
- ➡ Strong deformations in sensor space with uneven distribution of observations between kernels ($\sigma = 3.075$) due to more complex mapping.
  - ✓ max $E_{\phi_3} = 0 \pm 0.236$m (Error of function approximation) 1701% increase with respect to H_1

Trajectory in cartesian space

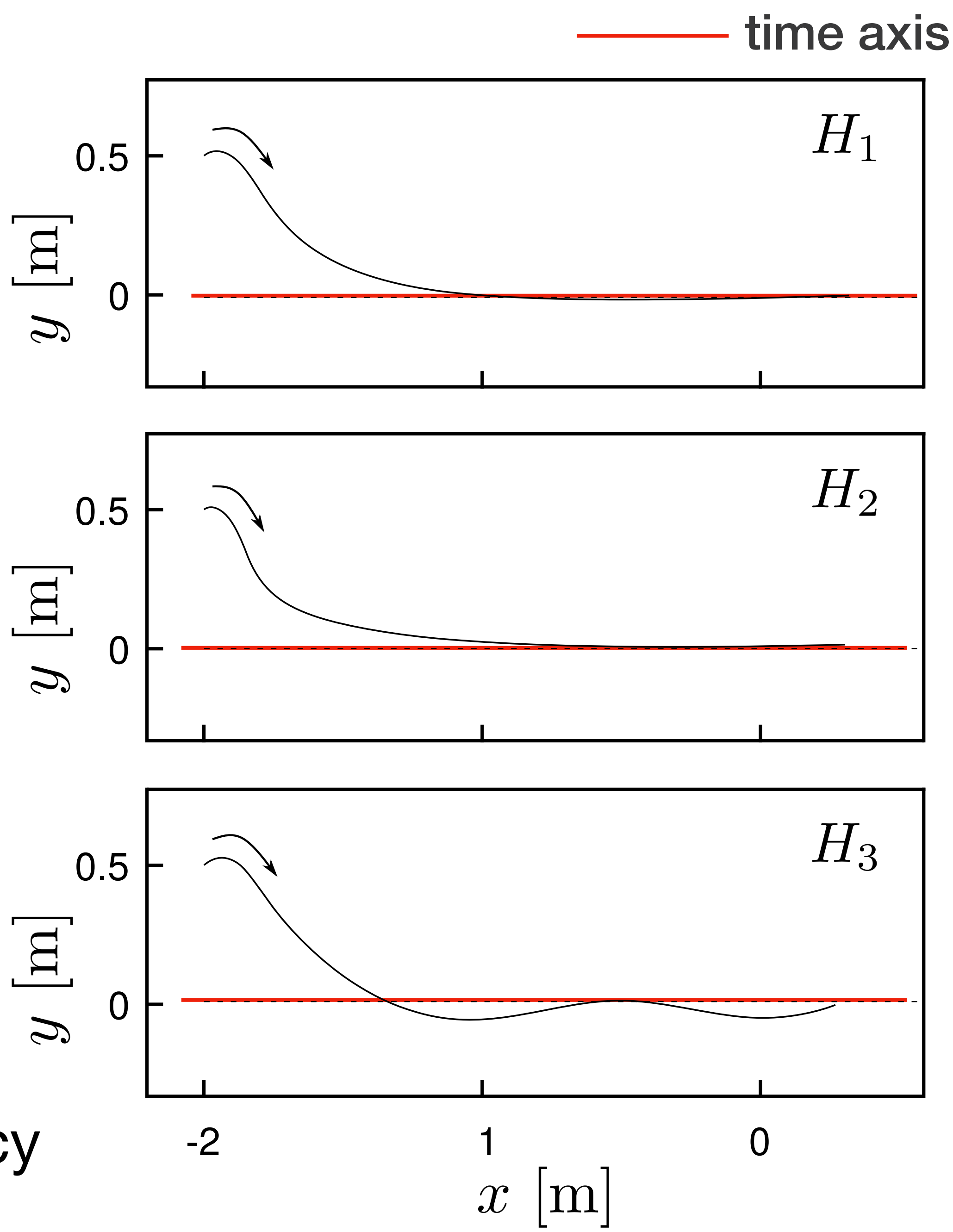


Bases, time-axis, observations (sensor space H_3)

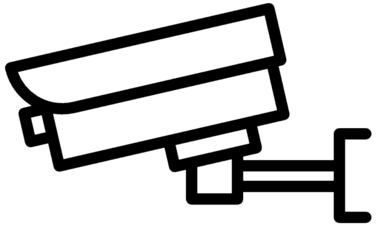


3.1 Feedback control: concluding remarks

- $(x_0, y_0, \theta_0) = (-2, 0.5, \pi/4)$
- Control poles $(-5, -5)$
- The system was successfully controlled in all cases: ϕ was good enough.
- Sensor space deformation → Reduced accuracy of ϕ → Reduced performance of feedback controller. (More kernels and observations may be required.)
- The performance of the method depends strongly on the accuracy of the approximated function

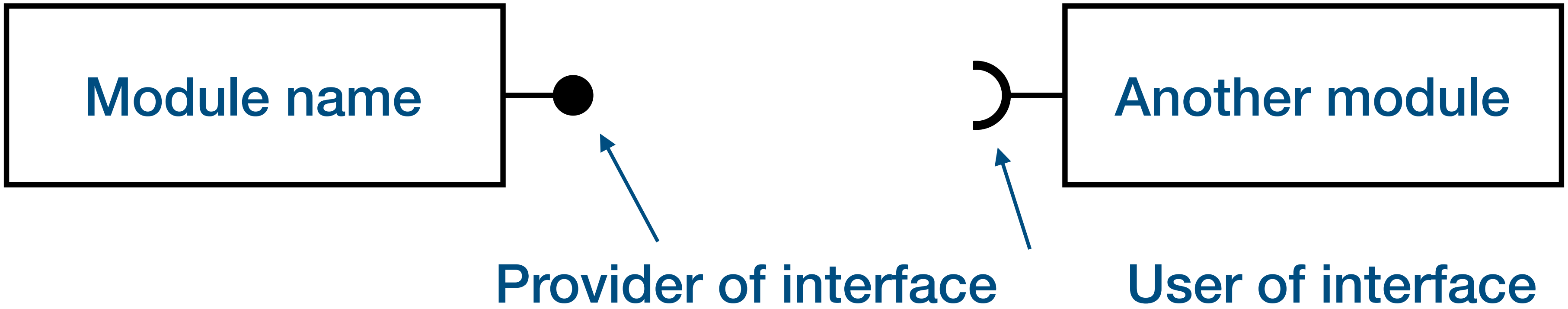


3.2 Experiment



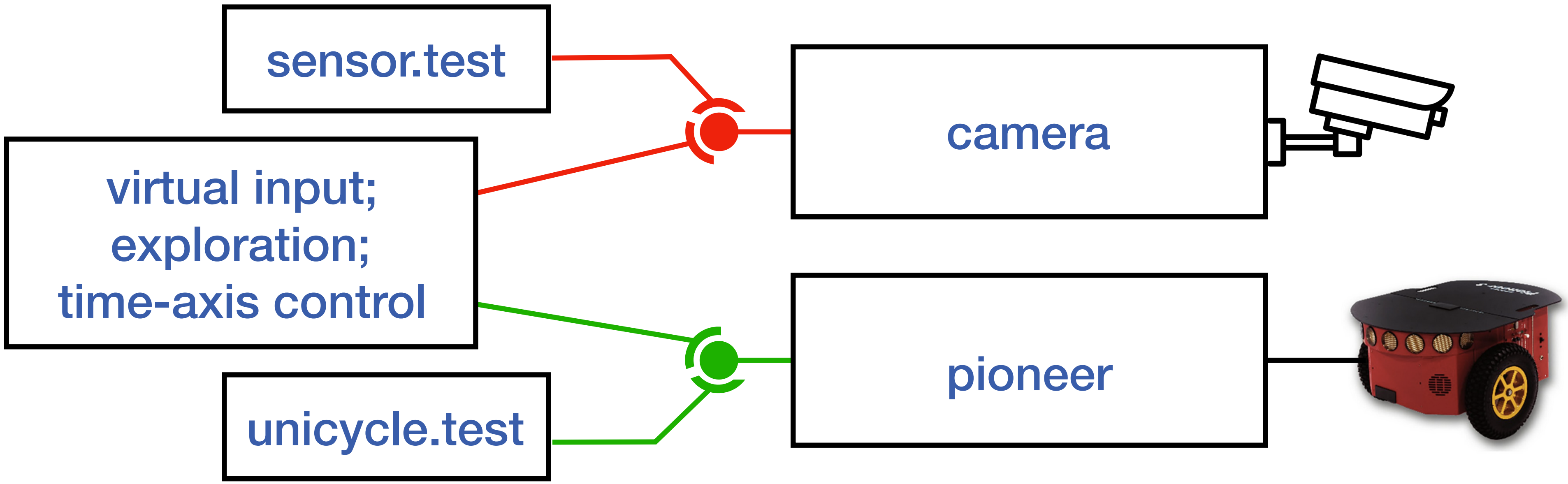
- Platform: Pioneer 3-DX
- Off-board 5K PTZ camera
- Onboard laptop
- CORBA Framework

- CORBA Component Model notation



3.2 Experiment setup

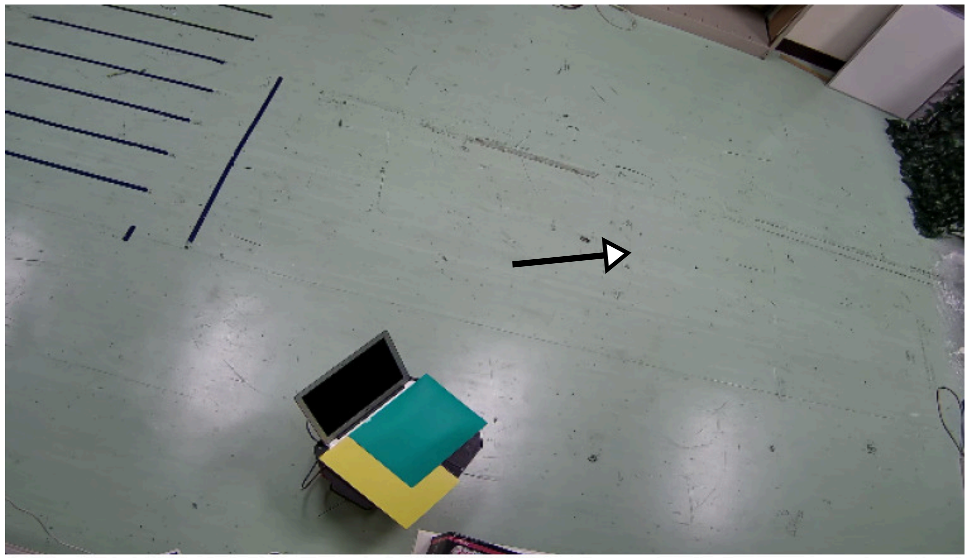
➡ CORBA layout during the experiment



➡ Starting point of stage 1 and 2 at approx. the center of the camera view (——▷)

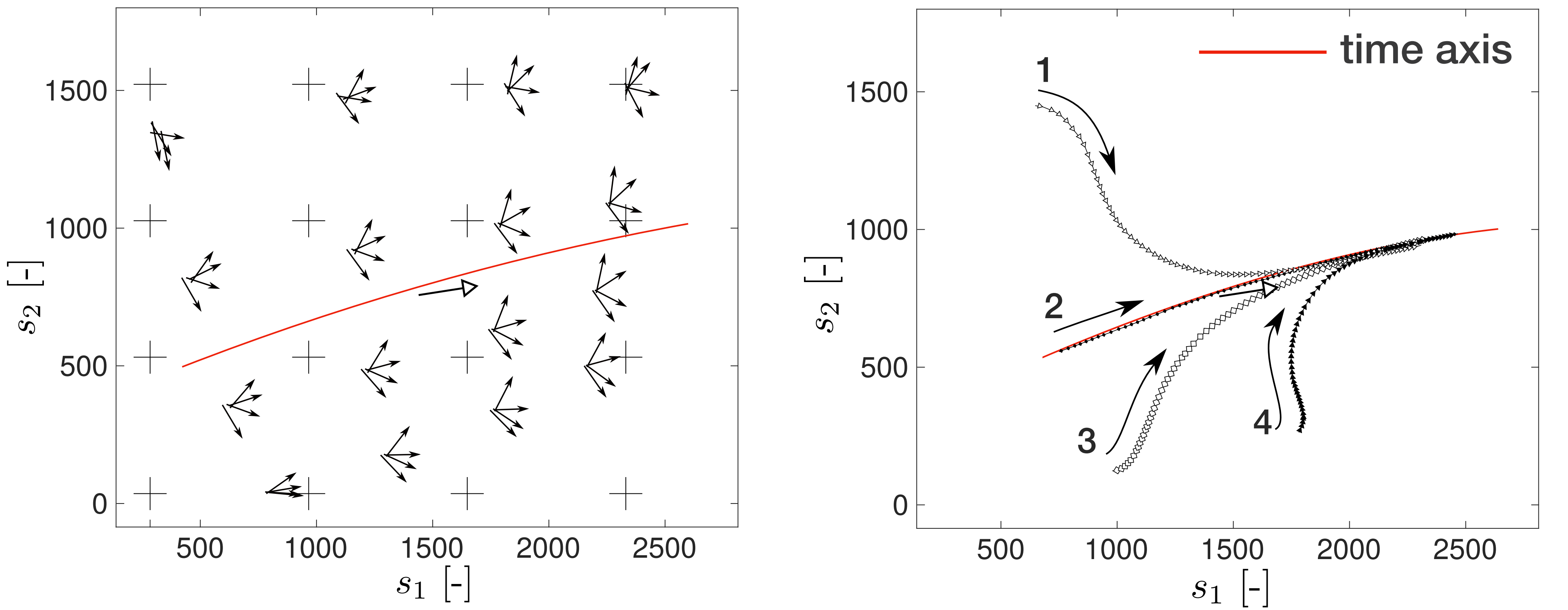
➡ 7 Jacobians per axis (Stage 1)

➡ 4^3 sensor observations (Stage



3.2 Experimental results

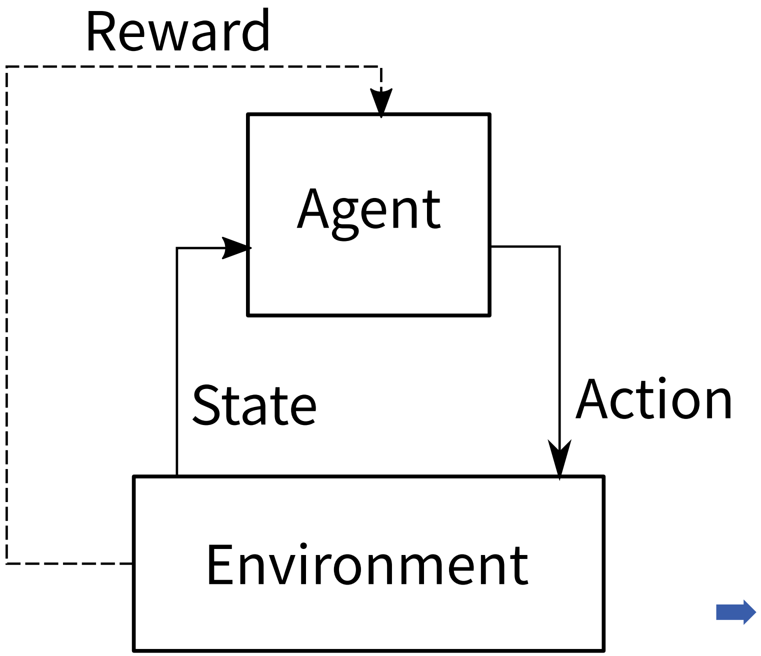
Exploration of sensor space | Evaluation



- ➡ Control to the origin was successful (4 starting points shown)
  - ✓ Despite sensor observation inaccuracies
  - ✓ Despite deviations by dead-reckoning
- ➡ As per time-axis control, control to the origin should be possible by switching sign of u_1.

3.3 Comparison with PPO - settings

- Proximal Policy Optimization (PPO) algorithm is a reinforcement learning (RL) method based on Actor-Critic agents.
  - ✓ Not designed explicitly for our problem setting.
  - ✓ RL methods can be adapted to any kind of control problem.



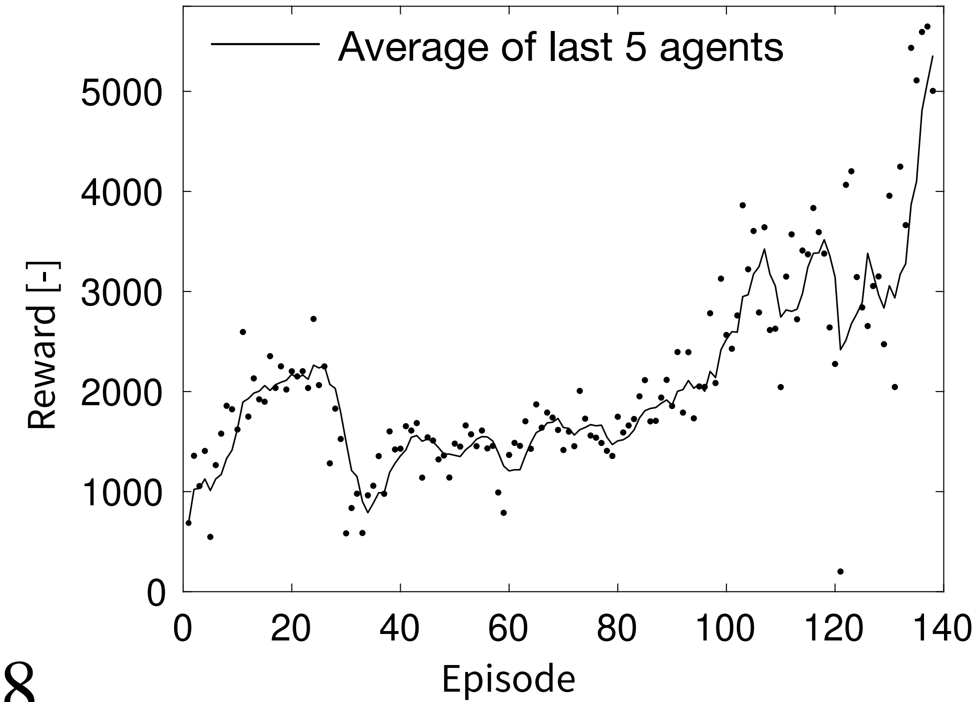
- Similar conditions to the Matlab environment.
  - ✓ Constant forward speed $u_1 = 1$
  - ✓ Discount factor $\gamma = 0.997$
  - ✓ $(x_0, y_0, \theta_0) = (-2{,}0.5{,}\pi/4)$

- Reward function $r_i = 100 \; || \; z \; ||^{-1}$

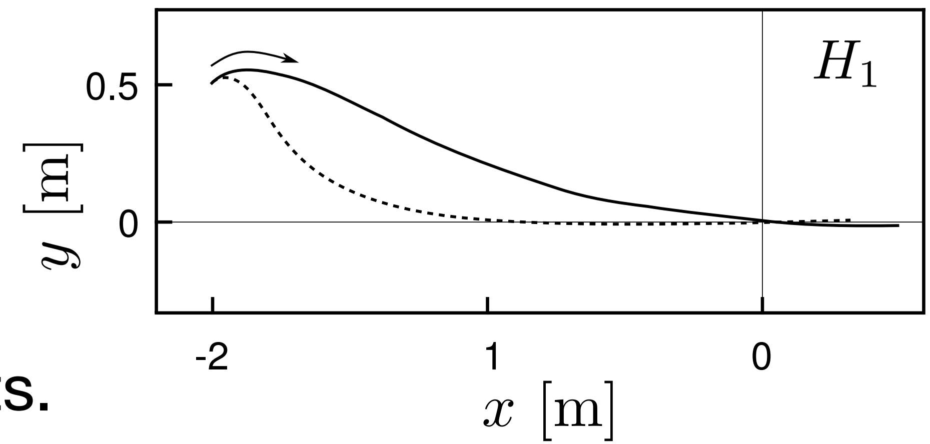
- Closer to origin: Bigger reward

3.3 Comparison with PPO - results

- Training
  - ✓ 138 episodes (agents)
  - ✓ 2926 sensor observations vs. 179 in the proposed method
  - ✓ Last 5 agents evaluated



- Evaluation
  - ✓ Distance to origin $d_{PPO} = -0.0783, \sigma_{d_{PPO}} = 0.0878$ vs. $d_{H_0} = 0.0114$ in the proposed method



- The proposed method is more efficient and more accurate.
- PPO is unfeasible in real environments.

4. Conclusion

- A learning controller for a non-holonomic system with unknown sensorimotor mapping was developed:
    - First, a virtual input $u_{(3)}$ is deduced.
    - Then, the sensor space is explored in a fixed pattern.
    - A mapping from sensor space to chained form was approximated with the data obtained previously
- Simulation and experiment were successfully controlled to the origin by time-axis state control.
- The originality is in the problem tackled and in the method of virtual input; followed by fixed-pattern exploration.
- Limitations:
    - Region of exploration is manually fixed
    - Curse of dimensionality

4. References

- Amar, K. and Mohamed, S. (2013), "Stabilized feedback control of unicycle mobile robots. International Journal of Advanced Robotic Systems, Vol. 10 No. 2.
- Astolfi, A. (1995), "Exponential stabilization of a car-like vehicle", in 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-Novel, Bastin, G., and Campion, G. (1991), "Modelling and control of non-holonomic wheeled mobile robots", in IEEE International Conference on Robotics and Automation, Sacramento, California, 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", in IEEE International Symposium on Industrial Electronics, Pertoria, pp. 420–425.
- Harada, K. and Kobayashi, Y. (2017), "Estimation of structure and physical relations among multi-modal sensor variables for musculoskeletal robotic arm", in IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, Daegu, pp. 547–552.
- 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., 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.
- Smart, W. D., & Pack Kaelbling, L. (2002). Effective Reinforcement Learning for Mobile Robots.

4. References

- 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, 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, 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", in 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", in 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, pp. 260–269.
- Sampei, M. (1994), "A control strategy for a class of non-holonomic systems - time-state control form and its application", in IEEE Conference on Decision and Control, Florida, 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", in IEEE Conference on Decision and Control, Kobe, pp. 4745–4746.
- Zeng, J., Ju, R., Qin, L., Hu, Y., Yin, Q., & Hu, C. (2019). “Navigation in Unknown Dynamic Environments Based on Deep Reinforcement Learning. Sensors”

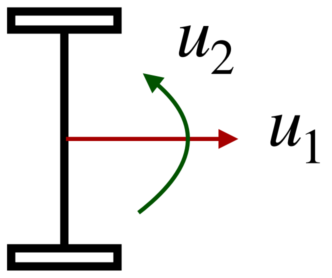
Control of Non-Holonomic Driftless System with Unknown Sensorimotor Model by Jacobian Estimation

Francisco J. (Paco) ARJONILLA GARCÍA

PhD Thesis Presentation

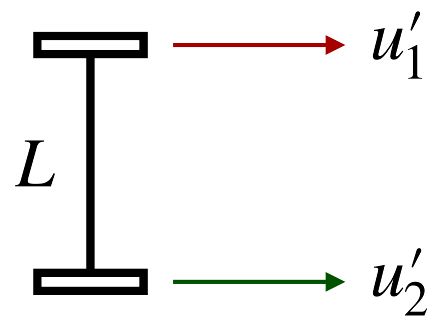
Thank you for your attention

5.1 Extension: Ratio between wheel radiuses



$$u_1 = \frac{u'_1 + u'_2}{2}$$

$$u_2 = \frac{u'_2 - u'_1}{L}$$



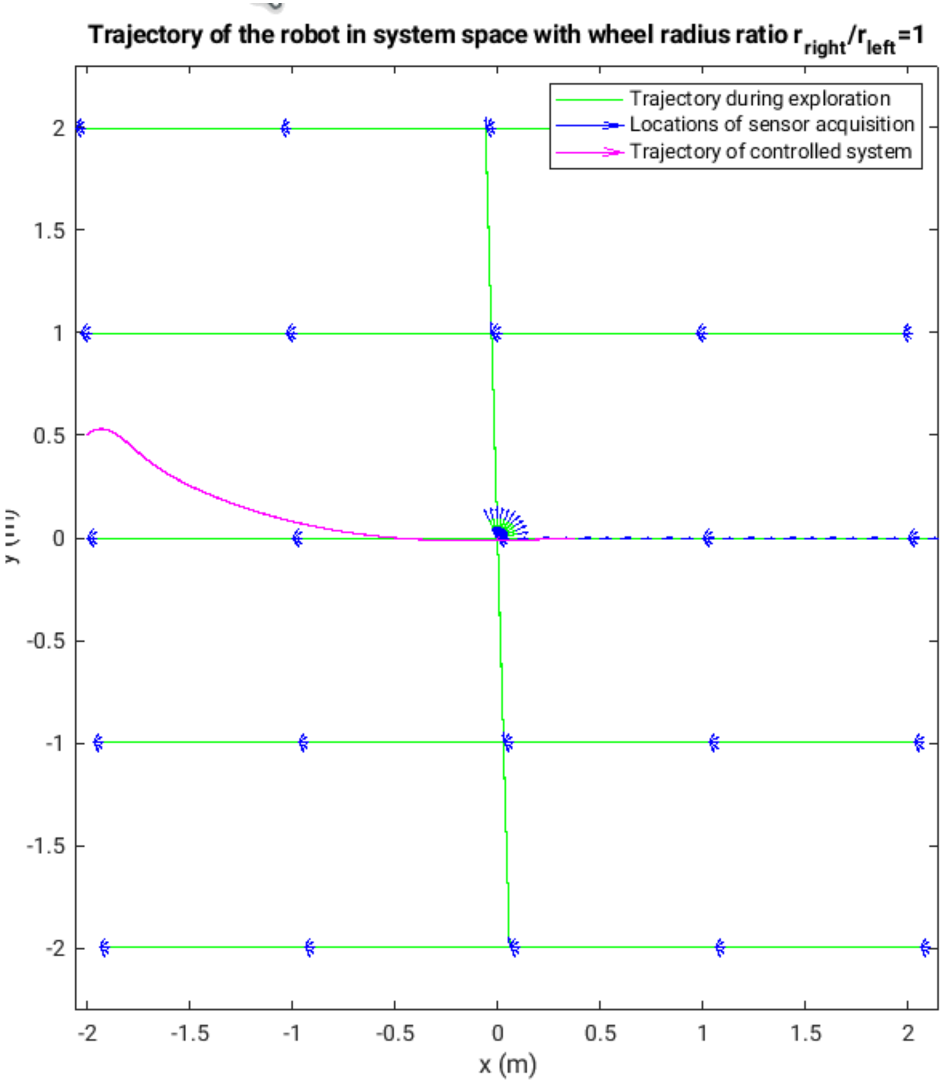
$$u'_1 = u_1 - \frac{L}{2} u_2$$

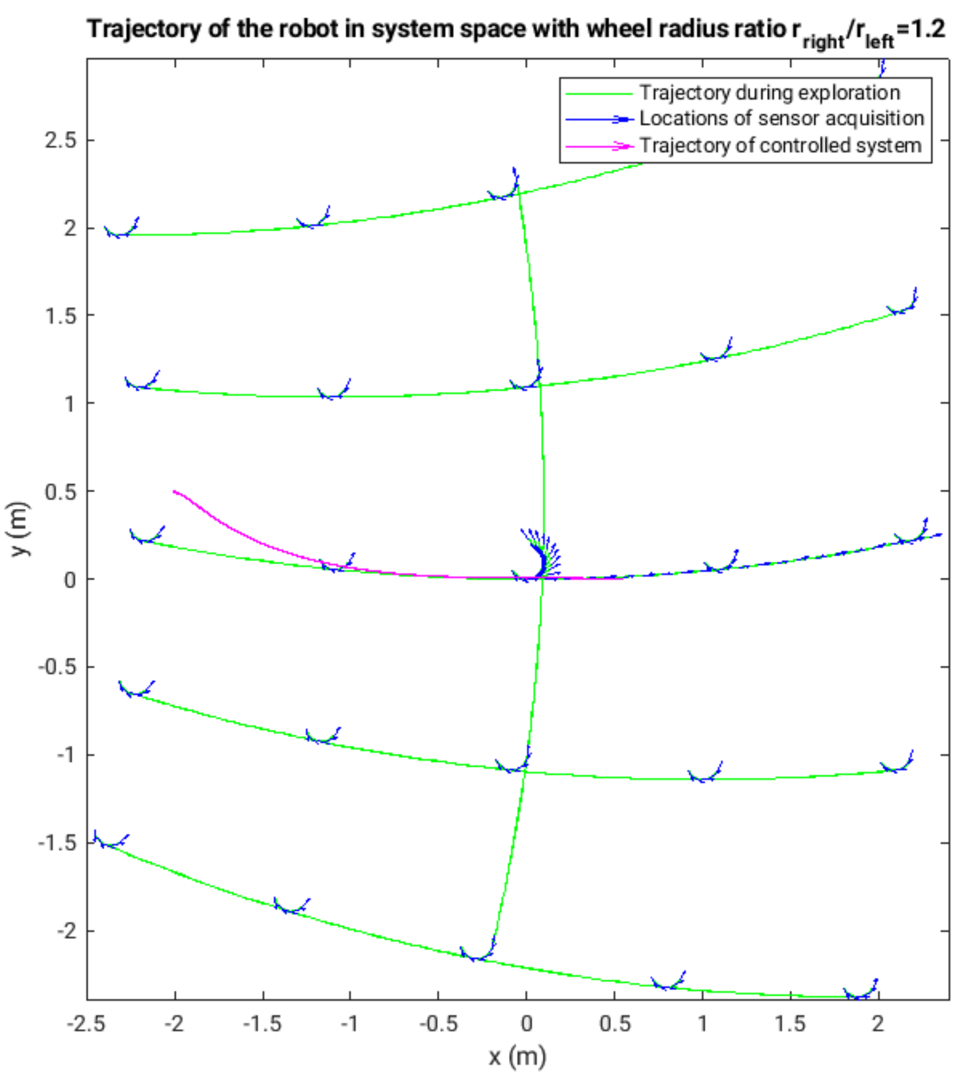
$$u'_2 = u_1 + \frac{L}{2} u_2$$

$$R = \frac{r_l}{r_r}$$

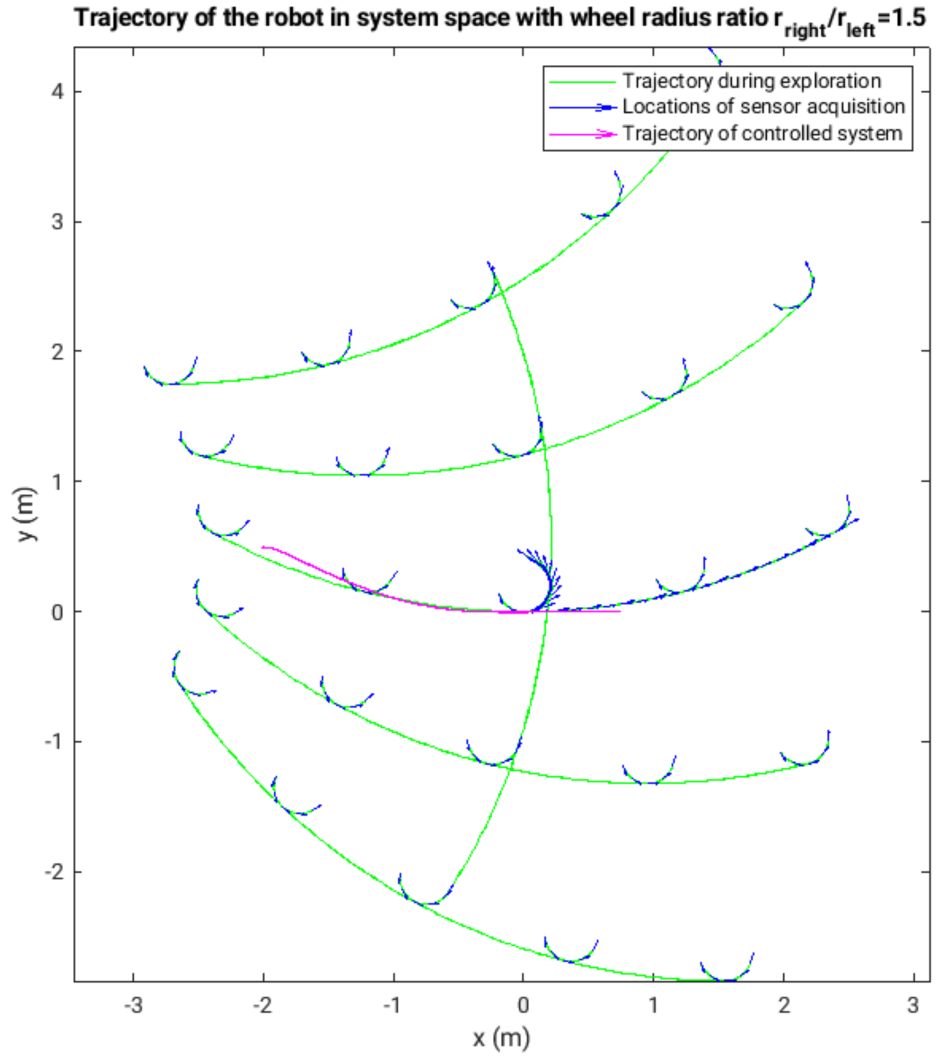
$$\dot{\boldsymbol{q}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{R+1}{2} \cos\theta \\ \frac{R+1}{2} \sin\theta \\ \frac{R-1}{2} \end{bmatrix} u_1 + \begin{bmatrix} \frac{R-1}{2} \cos\theta \\ \frac{R-1}{2} \sin\theta \\ \frac{R+1}{2} \end{bmatrix} u_2$$

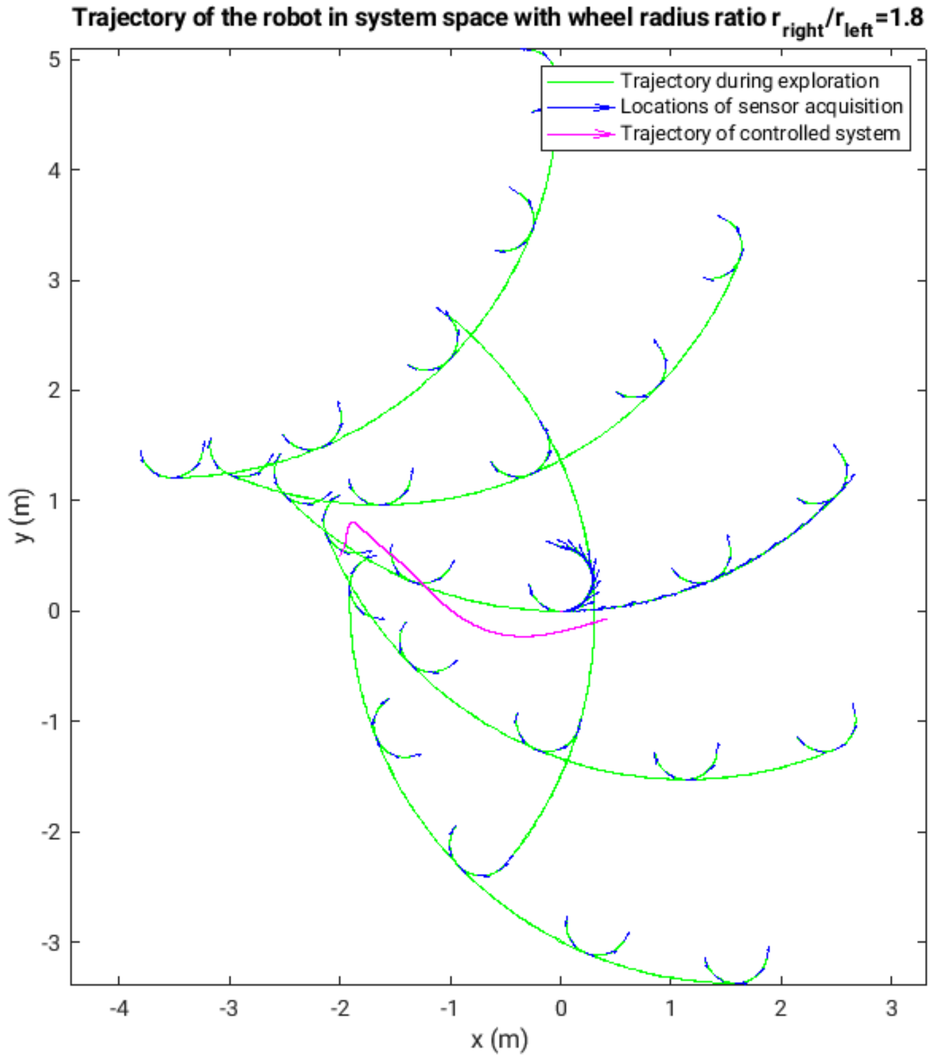
5.1 Extension: Ratio between wheel radiuses



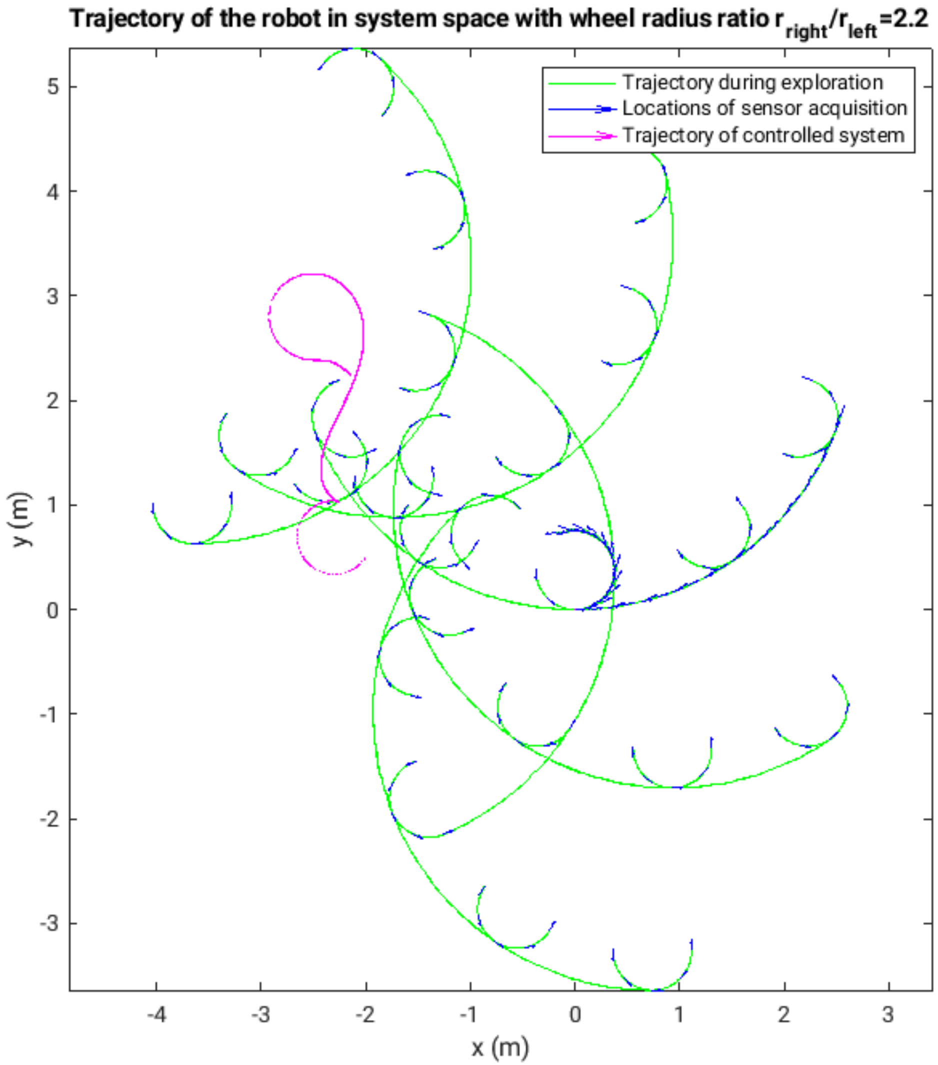


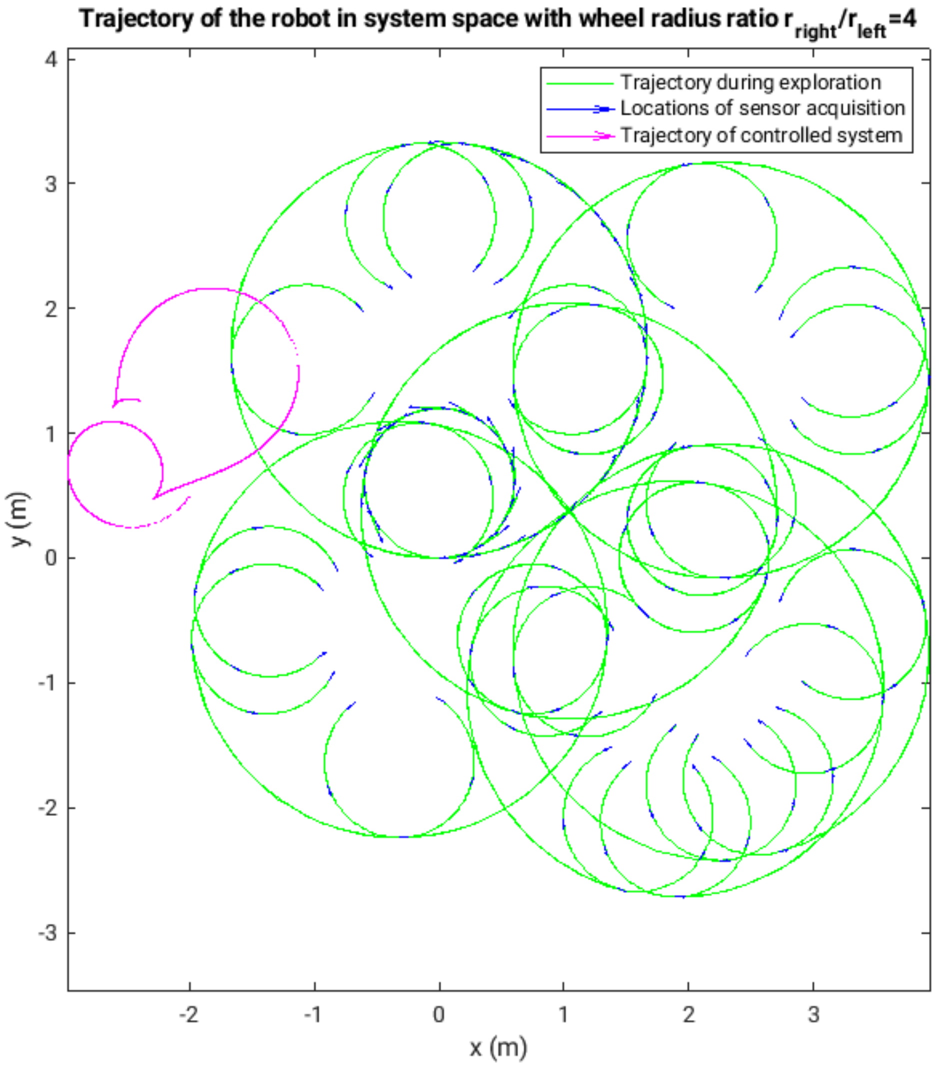
5.1 Extension: Ratio between wheel radiuses



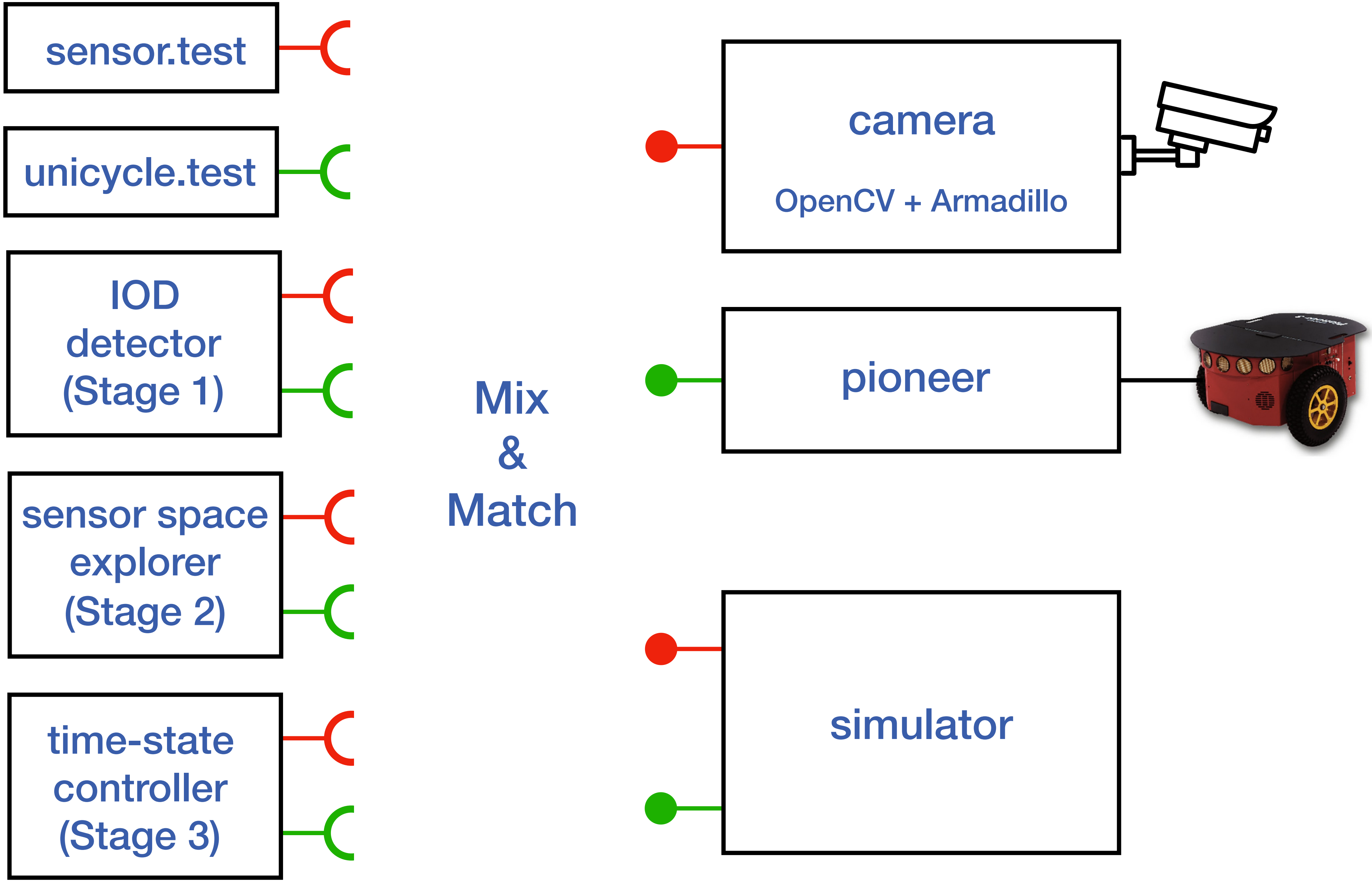


5.1 Extension: Ratio between wheel radiuses





5.2 CORBA modules



5.3 Beacon detection algorithm