SnakeSIM: a Snake Robot Simulation Framework for Perception-Driven Obstacle-Aided Locomotion

Filippo Sanfilippo[1†], Øyvind Stavdahl[1] and Pål Liljebäck[1]

[1]Dept. of Eng. Cybernetics, NTNU – Norwegian University of Science and Technology, 7491 Trondheim, Norway
(Tel: +47-942-58-929; E-mail: filippo.sanfilippo@ntnu.no)

Abstract: Snake robot locomotion in cluttered environments where the snake robot utilises a sensory-perceptual system to perceive the surrounding operational environment for means of propulsion can be defined as *perception-driven obstacle-aided locomotion* (POAL). The development of POAL is challenging. Moreover, testing new control methods for POAL in a real setup environment is very difficult because potential collisions may damage both the robot and the surrounding environment. In this perspective, a realistic simulator framework may enable researchers to develop control algorithms for POAL more safely, rapidly and efficiently. This paper introduces *SnakeSIM*, a virtual rapid-prototyping framework that allows researchers for the design and simulation of control algorithms for POAL. To demonstrate the potential of *SnakeSIM*, a possible control approach for POAL is also considered as a case study.

Keywords: perception-driven obstacle-aided locomotion, snake robots, rapid-prototyping, ROS.

1. INTRODUCTION

Biological snakes may push against rocks, stones, branches, obstacles, or other environment irregularities in the terrain for locomotion, which allows them to be remarkably adaptable to different types of environments. Snake robots that can mimic this variety of behaviour could open up to a variety of possible applications for use in challenging real-life operations, such as explorations of earthquake-hit areas, pipe inspections for the oil and gas industry, fire-fighting operations, and search-and-rescue activities. Snake robot locomotion in cluttered environments where the snake robot utilises a sensory-perceptual system to exploit the surrounding operational space and identifies walls, obstacles, or other external objects for means of propulsion can be defined as *perception-driven obstacle-aided locomotion* (POAL) [1, 2]. The underlying idea is shown in Fig. 1. The snake robot exploits the environment for locomotion by using augmented information: potential push-points are chosen (shown as cylinders), while achievable contact reaction forces are illustrated by arrows.

The development of POAL is challenging. Furthermore, testing new control methods for POAL in a real setup environment is very difficult because potential collisions may damage both the robot and the surrounding environment. For these reasons, a realistic simulator framework may enable researchers to develop control algorithms for POAL in a realistic and safe simulation setup. Robotic simulators are normally used in the design and testing of control algorithms. Related to this, the Robot Operating System (ROS) [3] has emerged as a de facto standard for robot software architecture among the research community in recent years. The primary goal of ROS is to provide a common platform to make the design of capable robotic applications quicker and easier. Some of the features it provides include hardware abstraction, device drivers, message-passing and package management. In conjunction with ROS, Gazebo 3D simulator [4] can be adopted to efficiently simulate robots in complex indoor and outdoor environments. Gazebo also provides a robust physics engine, high-quality graphics, and convenient programmatic and graphical interfaces. In this perspective, ROS serves as the interface for the robot model, while Gazebo is used to simulate both the robot and its operational environment.

Even though, ROS and Gazebo provide advanced features for general robotic applications, the main drawback is that a comprehensive collection of tools, libraries, and conventions specifically designed for rapid-prototyping [5] POAL is still missing. The main contribution of this work is the development of a rapid-prototyping framework for POAL and the integration of this missing technology with ROS. This integrated framework will enable researchers to develop control algorithms for POAL more safely, rapidly and efficiently. To demonstrate the potential of *SnakeSIM*, a possible control approach for POAL is also considered as a case study in this paper.

† Filippo Sanfilippo is the presenter of this paper.

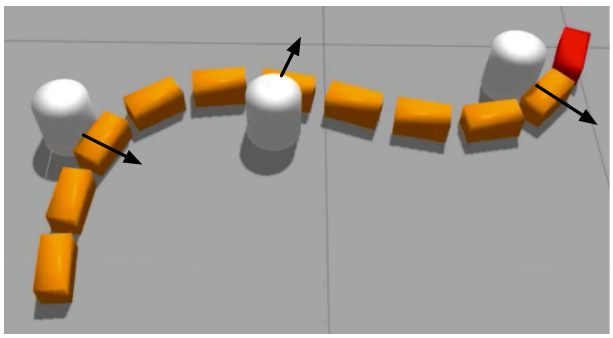


Fig. 1 The underlying idea of snake robot perception-driven obstacle-aided locomotion (POAL).

2. FRAMEWORK ARCHITECTURE

As highlighted in Fig. 2, the control framework is implemented in ROS [3], while Gazebo [4] is adopted to provide seamless simulations. In addition to ROS and

Gazebo, the RViz (ROS visualisation) [6] visualisation tool is adopted to visualise and monitor sensor information retrieved in real-time from the simulated scenario. In particular, the normal and tangent vectors to the snake robot at the contact points are visualised and continuously updated according to the simulated scenario.

The proposed control framework is hierarchically organised, as shown in Fig. 2. The following abstraction levels are defined:

- Perception/Mapping: this level is responsible for achieving the functions of sensing, mapping and localisation. The snake robot's sensory-perceptual data are used to produce a representation of the surrounding environment. The output from this level matches the required input from the motion planning algorithm, which involves parsing or segmenting the resulting representations (e.g. point-clouds) into simplified and more manageable data (e.g. positions of the obstacles). This level is currently underdevelopment;
- Motion planning: this level is responsible for decision-making, path-planning and mission planning activities. External system commands (e.g. a joystick operated by a human operator or an external system) and the snake robot's perception/mapping data (e.g. obstacles relative location and properties) are used to provide an input to this level. The expected output from this level is the robot's path;
- High-level control: this level is the core of the control framework. This level enables researchers to develop their own alternative control method for POAL. Each possible control method must be compliant with the interface of the framework. Each alternative control method must combine force and torque information with positional data to satisfy simultaneous position and force trajectory constraints for mapping a desired parametrised path to obstacle contact forces, and these forces to control inputs for the joint actuators, given a desired robot velocity. The inputs for this level are the desired robot shape (which is the output from the level above), the desired robot velocity (which can be set by external system commands, e.g. a joystick or a mission control entity) and the actual contacts (which are given from the below level through the robot's tactile perceptual system). The expected output from this level consists of motor torques for the joint actuators to satisfy simultaneous position and force trajectory constraints.

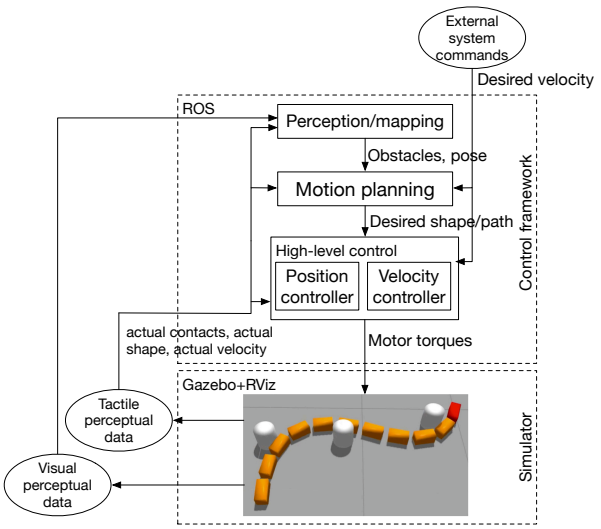


Fig. 2 The proposed framework architecture for *SnakeSIM*.

Some of the framework features are currently under developing. The implementation details of the framework will be presented in a future paper.

2.1. Simulated scenario

To carry out our experiment, a simulation scenario is built in Gazebo reproducing a cluttered environment. In particular, cylindrical objects or other shapes are placed in the scene and used as obstacles.

2.2. Snake robot model

To take full advantage of ROS, the snake robot model is implemented according to the Universal Robotic Description Format (URDF) [7]. The URDF is a specific file format used in ROS to describe all elements of a robot, such as links, joints, actuators and sensors. Simulated controllers are then adopted to actuate the joints of the snake robot. Simulated contact sensors are used to retrieve collisions with obstacles. Forces, torques, contact positions and contact normals can be retrieved.

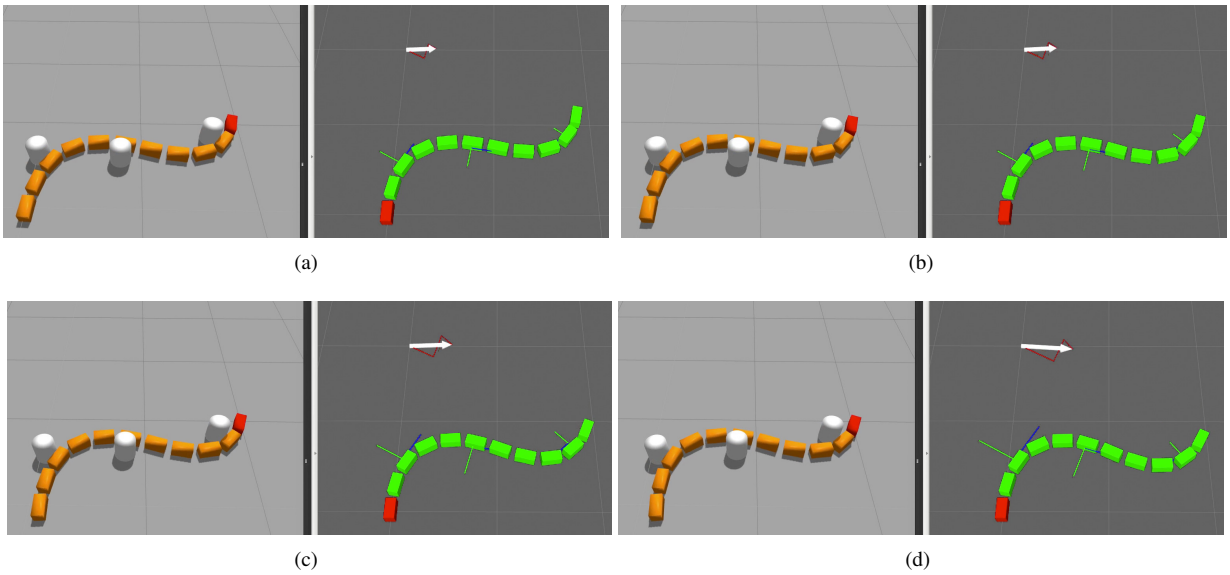
2.3. Snake robot sensors

Simulated contact sensors are adopted to retrieve bump contacts. In particular, the gazebo_ros_bumper_ controller is adopted [8]. Forces, torques, wrenches, contact positions and contact normals can be retrieved. Different other sensors can be also added, such as a depth camera, or an inertial measurement unit (IMU).

3. CASE STUDY AND SIMULATION RESULTS

As a case study, a novel control algorithm for POAL, which was previously presented by our research group [9], is implemented and tested using *SnakeSIM*. The aim of the selected control algorithm is to seek for a pragmatic approach to POAL by reducing the problem from a multi-dimensional formulation to only a two dimensional instance with a direction along the path and the other direction across the path. For further details, the reader is referred to [9].

A related simulation was carried out. For the simulation a desired propulsion speed, was indirectly set by imposing $f_s = 35N$. It should be noted that the joint 6 is controlled by the propulsion controller, while all the others joints are controlled in position (joints are numbered from left to right, starting from the joint 1). As shown in Fig. 3, a sequence of consecutive screenshots is taken from *SnakeSIM* (for each screenshot, the left snapshot is from Gazebo and the right snapshot is from RViz) demonstrating the effectiveness of the considered control algorithm for POAL. In particular, the link 6 is rotated



(a) (b) (c) (d)

Fig. 3 (a), (b), (c), (d) A sequence of consecutive screenshots taken from *SnakeSIM* (for each screenshot, the left snapshot is from Gazebo and the right snapshot is from RViz). Note that the action forces are shown in this sequence.

upwards, pushing on the obstacle to its left. The whole snake is then pushed slightly forward (links are numbered from left to right, starting from the link 1). It should be noted that the force polygon is also shown (red polygon) for each screenshot highlighting the resultant propulsion force (white arrow).

4. CONCLUSIONS AND FUTURE WORK

SnakeSIM, a virtual rapid-prototyping framework that allows for the design and simulation of control algorithms for perception-driven obstacle-aided locomotion (POAL) was presented in this paper. The framework proposed is integrated with ROS and enables researchers to develop control algorithms for POAL in a simulated environment with Gazebo. This integration makes the development of POAL algorithms more safe, rapid and efficient. Once the development phase is terminated, the designed control algorithms can be tested on a real prototype and continuously tuned with real sensor data. The integration with a real snake robot prototype, the Mamba snake robot [10], is currently undergoing.

The framework is built on open-source software. In the future, different control algorithms for POAL may be designed and tested. The framework may also be adopted as an educational tool.

REFERENCES

[1] F. Sanfilippo, J. Azpiazu, G. Marafioti, A. A. Transeth, Ø. Stavdahl, and P. Liljebäck, "A review on perception-driven obstacle-aided locomotion for snake robots," in *Proc. of the 14th International Conference on Control, Automation, Robotics and Vision (ICARCV), Phuket, Thailand*, pp. 1–7, 2016.

[2] F. Sanfilippo, J. Azpiazu, G. Marafioti, A. A. Transeth, Ø. Stavdahl, and P. Liljebäck, "Perception-driven obstacle-aided locomotion for snake robots: the state of the art, challenges and possibilities," *Applied Sciences*, vol. 7, no. 4, p. 336, 2017.

[3] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source robot operating system," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA), workshop on open source software*, vol. 3, p. 5, 2009.

[4] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3, pp. 2149–2154, 2004.

[5] J. Won, K. DeLaurentis, and C. Mavroidis, "Rapid prototyping of robotic systems," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, vol. 4, pp. 3077–3082, 2000.

[6] H. R. Kam, S.-H. Lee, T. Park, and C.-H. Kim, "Rviz: a toolkit for real domain data visualization," *Telecommunication Systems*, vol. 60, no. 2, pp. 337–345, 2015.

[7] Open Source Robotics Foundation, "Tutorial: Using a URDF in Gazebo," May 2016.

[8] Open Source Robotics Foundation, "Tutorial: Using Gazebo plugins with ROS," May 2016.

[9] F. Sanfilippo, Ø. Stavdahl, G. Marafioti, A. A. Transeth, and P. Liljebäck, "Virtual functional segmentation of snake robots for perception-driven obstacle-aided locomotion," in *Proc. of the IEEE Conference on Robotics and Biomimetics (ROBIO), Qingdao, China*, pp. 1845–1851, 2016.

[10] P. Liljebäck, Ø. Stavdahl, K. Pettersen, and J. Gravdahl, "Mamba - A waterproof snake robot with tactile sensing," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 294–301, Sept. 2014.