International Journal of Mechanical Engineering and Robotics Research Vol. 11, No. 2, February 2022

Combining Grasping with Adaptive Path Following and Locomotion for Modular Snake Robots

Filippo Sanfilippo
Department of Engineering Sciences, University of Agder (UiA), Grimstad, Norway
Email: filippo.sanfilippo@uia.no

Abstract—**In this paper, a framework architecture that combines grasping with adaptive locomotion for modular snake robots is presented. The proposed framework allows for simulating a snake robot model with locomotion and prehensile capabilities in a virtual environment. The simulated robot can be equipped with different sensors. Tactile perception can be achieved by using contact sensors to retrieve forces, torques, contact positions and contact normals. A camera can be attached to the snake robot head for visual perception purposes. To demonstrate the potential of the proposed framework, a case study is outlined concerning the execution of operations that combine locomotion and grasping. Related simulation results are presented.**

Index Terms— **grasping, locomotion, snake robotics**

I. INTRODUCTION

In nature, snakes are capable of performing an astounding variety of tasks. They can locomote, swim, climb and even glide through the air in some species. One of the most interesting features is their ability to exploit and traverse various typologies of terrain, which allows them to adapt to different types of environments. Biological snakes can push against rocks, stones, branches, obstacles, or other environment irregularities. They can also exploit walls and surfaces of narrow passages or pipes for locomotion. Another significant feature that many natural snakes exhibit concern their prehensile capabilities [1], which enable them to wrap around and grasp objects. Snake robots imitating this wide array of actions could enable a variety of possible applications for use in demanding 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 (SAR) activities. The possibility of achieving versatile locomotion and grasping of objects with snake-like configurations is of critical interest for SAR missions, e.g., to enlarge passages around victims or while passing through an evadable but blocked area during exploratory navigation or transport [2]. These features may also be used to bring first aids or drugs to the trapped people. Even though seminal works can be found in the previous literature, a comprehensive control framework specifically designed for combining grasping and locomotion control of snake-like robots is still missing to the best of our knowledge.


(a) (b)

Figure 1. The idea of combining grasping (a) with adaptive path following and locomotion (b) for snake robots.

The main contribution of this work is the development of a control framework architecture that makes it possible to combine grasping with adaptive locomotion for modular snake robots. The proposed architecture enables researchers to build a model of a snake robot with locomotion and prehensile/grasping capabilities in a virtual world. In particular, *CoppeliaSim Edu* [3] is adopted as a simulation environment. *CoppeliaSim Edu* is a flexible and modular simulation platform that enables different control methods to be implemented with ease. The model of the snake robot is based on the *Mamba* robot [4], which is a modular, reconfigurable, and waterproof experimental platform. The underlying idea is shown in Fig. 1. A variety of sensors can be embedded on the virtual robot. It is possible to attain tactile perception by means of contact sensors to extract forces, torques, contact positions and contact normals. For visual perception purposes a camera may be tethered to the head of the snake robot. To prove the effectiveness of the proposed system, a case study is illustrated regarding the completion of interventions which combine locomotion and grasping and are applicable to SAR operations. Corresponding simulation and results are outlined and discussed.

The paper is organised as follows. A review of the related research work is described in Section II. The proposed framework architecture and simulation environment are presented in Section III. In Section IV,

---

Manuscript received July 21, 2021; revised November 1, 2021.

doi: 10.18178/ijmerr.11.2.59-65

related simulation results are outlined. Finally, conclusions and future work are discussed in Section V.

II. RELATED RESEARCH WORKS

Regrading grasping capabilities for snake-like robots, seminal studies exist in the previous literature. In [5], the conditions that a snake robot must meet to be able to grasp an object with a circular cross-section are discussed. The type of grasp considered is of the enveloping type, and the conditions to guarantee a grasp with form-closure using three contacts, the minimum number of contacts possible, are presented.

Regarding the various locomotion patterns, lateral undulation is the fastest and most commonly implemented locomotion gait for robotic snakes in literature [6]. This particular pattern can be realised through phase-shifted sinusoidal motion of each joint [7]. Even though the previous studies have provided researchers with a better understanding of snake robots dynamics, most of the past works on snake robot locomotion have essentially exclusively considered motion across smooth surfaces. However, many real-life environments are not smooth, but cluttered with obstacles and irregularities. Snake robot locomotion in a cluttered environment where the snake robot utilises walls or external objects, other than the flat ground, for means of propulsion can be defined as *obstacle-aided locomotion* (OAL) [8], [9]. In this perspective, the environment perception, mapping and representation is of fundamental importance for the model. To highlight even more this concept, the term *perception-driven obstacle-aided locomotion* (POAL) was introduced by our research group as locomotion 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 [10], [11]. Based on this approach, *SnakeSIM*, a virtual rapid-prototyping framework that allows researchers for the design and simulation of POAL more safely, rapidly and efficiently, was introduced by our research group [12]. From a control perspective, achieving POAL requires precisely identifying potential push-points and to accurately determine achievable contact reaction forces. Accomplishing this with traditional rigidly-actuated robots is extremely demanding because of the absence of compliance. To tackle this challenge, *Serpens*, a newly-designed low-cost, open-source and highly-compliant multi-purpose modular snake robot with series elastic actuator (SEA) was recently presented by our research group [13].

When considering the combination of grasping and locomotion for modular snake robots, limited work can be found in the previous literature. In [14], hyper-redundant locomotion concepts are applied to a grasping and manipulation scheme based on a *grasping wave*. In [2], a combined grasping and locomotion control approach of modular robots is presented. Firstly, different grasping modes are integrated based on a modular approach. Then manipulation capability of robotic arms and flexible locomotion of mobile robots are combined. Furthermore, the exploitation of a task priority-based approach is considered to manage the trade-off between these two functionalities. In [15], a 3-D lasso-type grasping scheme is proposed, where the snake robot grasps an object with any of its body links which are at close proximity to the object while undergoing its serpentine motion with the remaining links and dragging the grasped object.

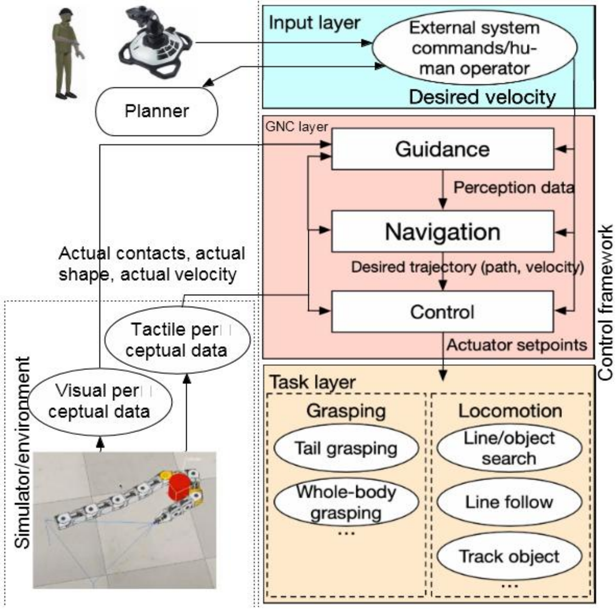

Figure 2. The proposed framework architecture.

Although the fundamental concepts are provided by these seminal works, an exhaustive control framework explicitly designed for combining grasping and locomotion control of snake-like robots is still missing to the best of our knowledge.

III. FRAMEWORK ARCHITECTURE AND SIMULATION ENVIRONMENT

A. Framework Architecture

The proposed control framework is hierarchically organised [13], as shown in Fig. 2. The input layer makes it possible to provide the robot with guidance either from a human operator to achieve teleoperation or other external systems (i.e., an external planner) to accomplish higher levels of autonomy.

The central layer is the layer that is strictly needed for achieving the standard functions and capabilities of guidance, navigation, and control (GNC) [16]:

- Guidance: this level is responsible for performing the functions of sensing, mapping and localisation. The snake robot's sensor data are used to produce a representation of the surrounding environment;
- Navigation: this level is responsible for decision making in terms of where, when and how the snake robot should ideally move [17]. External system commands and the snake robot's perception data represent the input to this level. The expected output from this level is the desired trajectory (e.g., path and velocity information);
- Control: this level is the core of the proposed control framework. It allows researchers for developing their own alternative control methods.

Each possible control method, however, must comply with the framework's given interfaces. The inputs to this level are the desired trajectory, as well as any relevant information from the above guidance level (perception data). The goal of the control level is to obtain the required setpoints for the robot actuators to follow the desired trajectory. This control action is based on the high-level information from the guidance level, but lower-level information like the actual position might be necessary depending on the actual algorithm employed in the control level.

The task layer includes different tasks with well-defined objectives to fulfil. The following tasks are implemented:

- Following a line. This task uses the robot's vision sensor to track the line [18]. By using a proportional integral derivative (PID) controller, the necessary adjustments to the snake robot's locomotion parameters are calculated to keep the line in the field of view of the camera. The considered parameters for this task are listed in Table I, while the related events are listed in Table II;
- Searching for an object of a specified colour. This task is responsible for searching a certain object by using the vision sensor. Blob detection is applied to the captured images to find an object of a given colour. Everything else that does not match the specified colour is ignored. The considered parameters for this task are listed in Table III, while the related events are listed in Table IV;
- Tracking an object with a specified colour. Once an object of a specified colour is detected, this task uses a PID controller to calculate the necessary adjustments to the snake's parameters so that the object is kept within the camera's field of view. The considered parameters for this task are listed in Table V, while the related events are listed in Table VI. In this preliminary study, the velocity of the snake is kept constant;
- Pregrasping. This task assumes the object to be grasped to be in front of the snake robot and visible to the camera before starting the execution. The snake robot starts by executing a continuous bending motion until the object to be grasped disappears from the camera field of view. A sequence of forward steps is then executed by the snake robot, followed by a continuous bending motion of the head until a collision with the object to be grasped is detected. This is repeated until the object is considered to be in a good position for grasping. The considered parameters for this task are listed in Table VII, while the related events are listed in Table VIII;
- Grasping. This task requires the object to be grasped to be in the correct position before the grasping begins (i.e., object pregrasped). A series of bending manoeuvres are performed to transition the snake into the whole-body grasping posture. To make sure that the object is in the correct position while performing this sequence, torque sensing at the joint level is adopted. When the bending procedure is terminated, collision detection is applied to check that the object is positioned correctly inside the body of the snake. The considered parameters for this task are listed in Table IX, while the related events are listed in Table X;
- Dropping an object. This task assumes the drop zone to be just in front of the snake and within the camera field of view. The snake robot moves towards and over the drop zone until the former disappears from the camera field of view. Then the snake robot will simply continue forwards for an empirically predefined amount of time. Successively, the snake robot will stop and release the object. The considered parameters for this task are listed in Table XI, while the related events are listed in Table XII;
- Rotating snake robot. This task simply rotates the snake robot. The considered parameters for this task are listed in Table XIII, while the related events are listed in Table XIV;
- Exploring. This task makes the snake robot head move left and right while making small steps forwards or backwards. The considered parameters for this task are listed in Table XV.

TABLE I. PARAMETERS FOR THE TASK OF FOLLOWING A LINE

Parameters	Description
Line colour	The colour of the line to track.

TABLE II. EVENTS FOR THE TASK OF FOLLOWING A LINE

Event	Description
Line found	Sent when the line is visible.
Line lost	Sent when the line is no longer visible.
End of the line	Sent when the end of the line is reached.

TABLE III. PARAMETERS FOR THE TASK OF SEARCHING FOR AN OBJECT OF A SPECIFIED COLOUR

Parameters	Description
Object colour	The colour of the object to search for.

TABLE IV. EVENTS FOR THE TASK OF SEARCHING FOR AN OBJECT OF A SPECIFIED COLOUR

Event	Description
Object found	Sent when an object is detected.
Object lost	Sent when an object is no longer visible.

TABLE V. PARAMETERS FOR THE TASK OF TRACKING AN OBJECT WITH A SPECIFIED COLOUR

Parameters	Description
Object colour	The colour of the object to track.

TABLE VI. EVENTS FOR THE TASK OF TRACKING AN OBJECT WITH A SPECIFIED COLOUR

Event	Description
Object reached	Sent when the object is considered to be directly in front of the snake.
Object lost	Sent if the object is no longer visible.

TABLE VII. PARAMETERS FOR THE TASK OF PREGRASPING AN OBJECT WITH A SPECIFIED COLOUR

Parameters	Description
Object colour	The colour of the object to pregrasp.
Object handle	The handle of the object to pregrasp (from the simulation environment) to be used for collision detection.

TABLE VIII. EVENTS FOR THE TASK OF PREGRASPING AN OBJECT

Event	Description
Object pregrasped	Sent when the object is considered to be in a good position for grasping.
Object pregrasp lost	Sent when the object pregrasp is lost.

TABLE IX. PARAMETERS FOR THE TASK OF GRASPING AN OBJECT

Parameters	Description
Object handle	The handle of the object to grasp (from the simulation environment) to be used for collision detection.

TABLE X. EVENTS FOR THE TASK OF GRASPING AN OBJECT

	Description
Object grasped	Sent when the object is stably grasped.
Object grasp lost	Sent if the object to be grasped is lost and no longer positioned between the arms. This is detected when the torque readings at the joint level are below a predefined threshold.

TABLE XI. PARAMETERS FOR THE TASK OF DROPPING AN OBJECT OVER A DROP ZONE WITH A SPECIFIED COLOUR

Parameters	Description
Drop zone colour	The colour of the drop zone.

TABLE XII. EVENTS FOR THE TASK OF DROPPING AN OBJECT OVER THE DROP ZONE

Event	Description
Dropping done	Sent after the object has been dropped.

TABLE XIII. PARAMETERS FOR THE TASK OF ROTATING THE SNAKE ROBOT

Parameters	Description
Direction	The direction to rotate the snake robot towards to.
Timeout	Optional timeout for the rotation to be performed.

TABLE XIV. EVENTS FOR THE TASK OF ROTATING THE SNAKE ROBOT

Event	Description
Timeout	Sent if the timeout expires.

TABLE XV. PARAMETERS FOR THE TASK OF EXPLORING

Parameters	Description
Direction	The direction for the snake robot to move for towards to.

B. Simulation Environment

Due to the complex interaction between the snake robot and the surrounding environment, the development of control algorithms is considered to be challenging. Furthermore, testing new control methods in a real setup environment is very difficult because potential collisions may damage both the snake robot and the environment. This process may also be time consuming. In contrast, a realistic simulator framework may enable researchers to develop control algorithms in a practical, efficient and safe simulation setup. Robotic simulators are commonly used in the design and testing of control algorithms. Related to this, *CoppeliaSim Edu* [3] is a flexible control and simulation framework. *CoppeliaSim Edu* has support for multiple operating systems, and it is based on a distributed control architecture: each object/model can be individually controlled via an embedded script, a plugin, a Robot Operating System (ROS) [19] or BlueZero [3] node, a remote application programming interface (API) client, or a custom solution. This makes *CoppeliaSim Edu* ideal for single robot, as well as multi-robot applications. Controllers can be written in C/C++, Python, Java, Lua, Matlab or Octave. Compared to other existing simulation environments, such as the Gazebo 3D simulator [20], *CoppeliaSim Edu* is more intuitive and user-friendly. *CoppeliaSim Edu* is based on open-source software and is free for educational purposes. For these reasons, *CoppeliaSim Edu* is chosen as the simulation environment in this work.

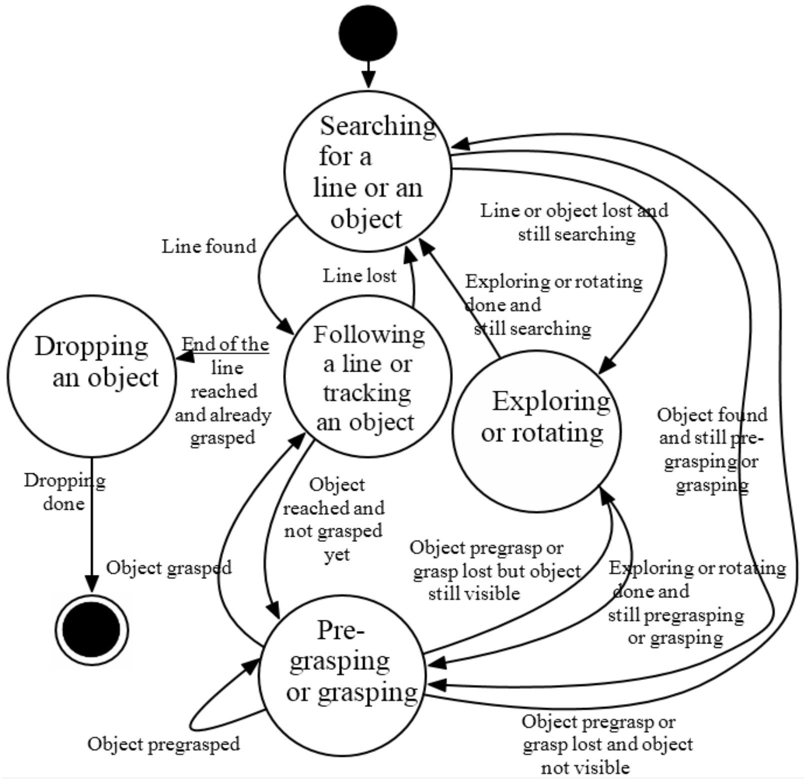

Figure 3. The autonomous planner for the selected case study.

IV. SIMULATION RESULTS

To demonstrate the potential of the proposed framework, a case study is presented. The purpose of this case study is for the snake robot to follow a line in the simulated scenario and retrieve a set of objects at the end of the line. In particular, there are two distinctly coloured objects that are marked as targets (i.e., green and red). They must be retrieved in a specified sequence (i.e., first green then red). These target objects must be sequentially reached, grasped and transported back to the initial side of the line where they should be dropped at a drop zone which matches the colour of the object. The autonomous planner for this scenario uses a combination of tasks to successfully complete the use case. A state machine diagram is shown in Fig. 3 to depict how the planner switches between the various tasks to accomplish its objective. It starts by searching for the line or the object. If the line is found, it continues by following the line. If the line is lost, the snake robot will be recovering itself by going back to the line or object search. This continue until the object is found. Once the object is found, the pregrasping and grasping tasks are executed to grasp the object. Successively, the line is followed until the drop zone is found. Once the drop zone is found, the object is released, and the process is restarted to bring back the next object. A sequence of successive screenshots is shown in Fig. 4. In each screenshot, the simulated scenario is depicted, together with the raw video stream, processed video stream and the torque measurements at the joint level. A demo video is available on-line at https://youtu.be/Va5tR9HKhQ8.

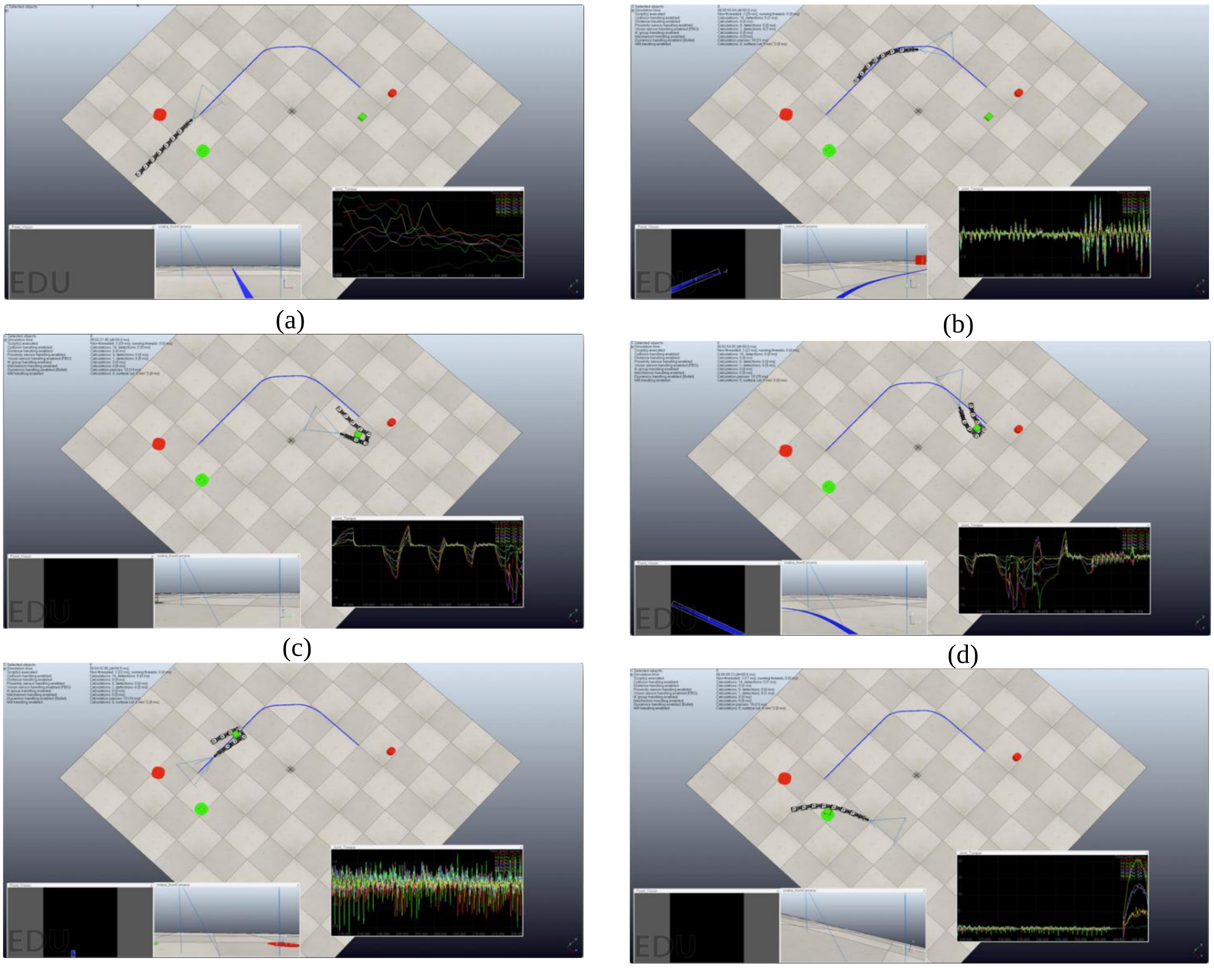
(a) (b) (c) (d) (e) (f)

Figure 4. A sequence of successive screenshots for the selected case study. In each screenshot, the simulated scenario is depicted, together with the processed video stream, the raw video stream and the torque measurements at the joint level.

V. CONCLUSION AND FUTURE WORK

A framework architecture that combines grasping with adaptive locomotion for modular snake robots was introduced in this paper. The framework enables the simulation of a snake robot model in a virtual scenario. The simulated robot is based on the *Mamba* robot [4] and it can be equipped with different sensors. Tactile perception can be achieved through the use of contact sensors to retrieve forces, torques, contact positions and contact normals. For visual perception purposes, a camera can be mounted on the head of the snake robot. The framework is based on opensource software. In particular, *CoppeliaSim Edu* [3], which is free for educational purposes, is adopted. To validate the proposed framework, a case study was illustrated regarding the execution of operations that combine locomotion and grasping. Related simulation results were presented. A demo video is available on-line at https://youtu.be/Va5tR9HKhQ8. As future work, the proposed approach needs to be validated with physical experiments. A more accurate study of the detection and perception errors need to be considered. Our research group is considering the integration of the proposed framework with *Serpens*, a newly-designed low-cost, opensource and highly-compliant multi-purpose modular snake robot with series elastic actuator (SEA) [13]. The possibility of implementing a more sophisticated sliding mode control method for trajectory tracking will be considered.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

FS conducted the research, analysed the data, and wrote the paper.

ACKNOWLEDGMENT

This work is supported by the Top Research Centre Mechatronics (TRCM), University of Agder (UiA). The authors acknowledge the contribution of the student Per Myren.

REFERENCES

[1] H. B. Lillywhite, *How Snakes Work: Structure, Function and Behavior of the World'S Snakes*. Oxford University Press, 2014.

[2] G. Salvietti, H. Zhang, J. Gonzalez-Gomez, D. Prattichizzo, and J. Zhang, "Task priority grasping and locomotion control of modular robot," in *Proc. of the IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2009, pp. 1069–1074.

[3] E. Rohmer, S. P. Singh, and M. Freese, “Coppeliasim (formerly vrep): A versatile and scalable robot simulation framework,” in *Proc. Int. Conf. Intell. Robots Syst.*, 2013, pp. 1321–1326.

[4] P. Liljeback, Ø. Stavdahl, K. Y. Pettersen, and J. T. Gravdahl, “Mamba-a waterproof snake robot with tactile sensing,” in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 294–301.

[5] F. Reyes and S. Ma, “On planar grasping with snake robots: Formclosure with enveloping grasps,” in *Proc. 2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*. IEEE, 2014, pp. 556–561.

[6] P. Liljeback, K. Y. Pettersen, Ø. Stavdahl, and J. T. Gravdahl, “A review on modelling, implementation, and control of snake robots,” *Robotics and Autonomous Systems*, vol. 60, no. 1, pp. 29–40, Jan. 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889011001618

[7] D. L. Hu, J. Nirody, T. Scott, and M. J. Shelley, “The mechanics of slithering locomotion,” in *Proc. of the National Academy of Sciences*, vol. 106, no. 25, pp. 10081–10085, Jun. 2009. [Online]. Available: http://www.pnas.org/content/106/25/10081

[8] A. Transeth, R. Leine, C. Glocker, K. Pettersen, and P. Liljeback, “Snake robot obstacle-aided locomotion: Modeling, simulations, and experiments,” *IEEE Transactions on Robotics*, vol. 24, no. 1, pp. 88–104, feb 2008.

[9] C. Holden, Ø. Stavdahl, and J. T. Gravdahl, “Optimal dynamic force mapping for obstacle-aided locomotion in 2d snake robots,” in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Chicago, Illinois, United States*, 2014, pp. 321–328.

[10] F. Sanfilippo, J. Azpiazu, G. Marafioti, A. A. Transeth, Ø. Stavdahl, and P. Liljeback, “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*, 2016, pp. 1–7.

[11] F. Sanfilippo, J. Azpiazu, G. Marafioti, A. A. Transeth, Ø. Stavdahl, and P. Liljeback, “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.

[12] F. Sanfilippo, Ø. Stavdahl, and P. Liljeback, “SnakeSIM: a ROS-based control and simulation framework for perception-driven obstacle-aided locomotion of snake robots,” *Artificial Life and Robotics*, vol. 23, no. 4, pp. 449–458, 2018.

[13] F. Sanfilippo, E. Helgerud, P. A. Stadheim, and S. L. Aronsen, “Serpens: A highly compliant low-cost ROS-based snake robot with series elastic actuators, stereoscopic vision and a screw-less assembly mechanism,” *Applied Sciences*, vol. 9, no. 3, p. 396, 2019.

[14] G. S. Chirikjian and J. W. Burdick, “Kinematics of hyper-redundant robot locomotion with applications to grasping,” 1991.

[15] B. Atakan, A. Erkmen, and I. Erkmen, “3-d grasping during serpentine motion with a snake-like robot,” Ph.D. dissertation, Citeseer, 2005.

[16] F. Kendoul, “Towards a unified framework for UAS Autonomy and Technology Readiness Assessment (ATRA),” *Autonomous Control Systems and Vehicles*, ser. Intelligent Systems, Control and Automation: Science and Engineering, K. Nonami, M. Kartidjo, K.-J. Yoon, and A. Budiyono, Eds. Springer Japan, 2013, no. 65, pp. 55–71, [Online]. Available: http://link.springer.com/chapter/10.1007/978-4-431-54276-6 4

[17] I. Rañó, A. G. Eguíluz, and F. Sanfilippo, “Bridging the gap between bio-inspired steering and locomotion: A braitenberg 3a snake robot,” in *Proc. of the 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. IEEE, 2018, pp. 1394–1399.

[18] E. Kelasidi, P. Liljeback, K. Y. Pettersen, and J. T. Gravdahl, “Integral line-of-sight guidance for path following control of underwater snake robots: Theory and experiments,” *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 610–628, 2017.

[19] 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, no. 3.2, 2009, p. 5.

[20] 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, 2004, pp. 2149–2154.

Filippo Sanfilippo holds a PhD degree in Engineering Cybernetics. He is an Associate Professor at the Dept. of Engineering Sciences, Faculty of Engineering and Science, University of Agder (UiA), Grimstad, Norway. He is also appointed as a Professor II at the Dept. of Mechanical, Electronic and Chemical Engineering, Faculty of Technology, Art and Design, Oslo Metropolitan University (OsloMet), Oslo, Norway. He is responsible for the research theme on Collaborative Robots (CoBots) at the Priority Research Centre Mechatronics, UiA. He is also a member of the Centre for Integrated Emergency Mangement (CIEM), UiA. His research focuses on robotics, wearable haptics and safe human-robot interaction. He carries experience in participating to European research programs and various national projects from the Research Council of Norway (RCN). He is an IEEE Senior Member. He is currently the Chair of the IEEE Norway Section. He is also the Chair of the IEEE Robotics and Automation, Control Systems and Intelligent Transportation Systems Joint Chapter. He is also the treasurer of the Norsk Forening for Kunstig Intelligens (NAIS), the Norwegian Association for Artificial Intelligence. He has authored and co-authored several technical papers in various journals and conferences. He is a reviewer for several international conferences and journals.