Collaborative Robots (Cobots) for Emergency Situations: a Snake Robot as a Team Member for Delivering First Aid in Emergency Situations

Syed Kumayl Raza Moosavi
National University of Sciences and Technology, Islamabad, Pakistan
smoosavi.msai21seecs@seecs.edu.pk

Filippo Sanfilippo
University of Agder, Grimstad, Norway
Kaunas University of Technology, Kaunas, Lithuania
filippo.sanfilippo@uia.no

Muhammad Hamza Zafar
University of Agder, Grimstad, Norway
muhammad.h.zafar@uia.no

Abstract

Cobots are robots that are built for human-robot collaboration (HRC) in a shared environment. In the aftermath of disasters, cobots can work with humans to reduce risk and increase the possibility of rescuing people. In this work, the collaboration between a snake robot, first responders and people to be rescued is considered. The possibility of delivering first aid to a victim is implemented. The snake robot receives (from first responders or another robot) the site planimetry, the location of the person to be rescued, and a aiding good to be delivered. The snake robot plans the path to reach the victim. By using its prehensile capabilities, the snake robot grasps the aiding object to be dispatched. Consequently, the snake robot reaches the delivering location and releases the item. To demonstrate the potential of the framework, several case studies are outlined concerning the execution of operations that combine locomotion and grasping.

Keywords: Cobots, Search and Rescue Operation, Human Robot Collaboration, Snake Robot, Path Planning, Disaster Scenarios

1. Introduction

Societies encounter unforeseeable crisis situations such as earthquakes, fires, floods, hazardous spills, hurricanes/typhoons, tsunamis, terrorist attacks, refugee crises, and more. These crises can arise from natural causes or human activities and result in substantial loss of life, injuries, displacement of people, and damage to property. In retrospect, humans have learned to adapt and manage such calamities on a global scale. However, due to an increase in the impact of the ever increase disasters, exacerbated partly due to the climate change, management of such disasters have become complex in this socio-ecological landscape. According to the latest data from insurer Munich Re (Munich Re, 2023), the average losses due to natural catastrophes over five years (2017-2021), adjusted to inflation, was approximately $270bn while the statistics for 2022 alone was over $270bn. Moreover, the impact of disasters are not always recorded in detail when disasters do occur.

Therefore, the need to manage such disaster situations is apparent. Therefore, it is increasingly important for disaster managers to assume an expanding role in safeguarding their communities through the formulation of effective management strategies. Emergency management processes are commonly categorized into distinct stages, although there is no universally agreed-upon model. On the aftermath of an event, disaster management typically involves four phases (Cova, 1999): to **mitigate** devastation of effective areas of potential disasters; have **preparedness** by incorporating trained personnel and shelter facilities; effective **response** during search and rescue (SAR) operations; **recovery** during the aftermath of the disaster. Conventional approaches, including field monitoring, physics-based models, expert surveys, and multi-criteria decision-making techniques, are utilized to identify hazards and risk factors. However, these methods often require extensive human effort and may be prone to generating false alarms. One of the technological tools that can be used in these dangerous environments without adding risk to the life of humans, who are in the process of the search and rescue operation, are robots. By integrating human skills with automation, a harmonious blend can be achieved, leveraging the adaptability of manual processes and the effectiveness and consistency of machines. This synthesis enables the realisation of benefits such as

URI: https://hdl.handle.net/10125/106428
978-0-9981331-7-1

design flexibility, varied product offerings, and system reconfiguration. One avenue to accomplish this is through the utilisation of collaborative robots, also known as cobots (Jeyson et al., 2022).

To enhance the definition of disaster recovery management strategy, researchers have proposed the integration of collaborative systems in a virtual space (Bertolino and Tanzi, 2019). An example of addressing this issue can be found in a recent study (Magid et al., 2019), where a novel framework and diverse control strategies were introduced for enhancing the collaborative performance of heterogeneous robotic teams in the context of sensing, monitoring, and mapping flood and landslide disaster zones. The research presents a foundation of virtual simulators that demonstrate various robot interaction protocols and system modeling concepts within the Gazebo environment of the Robot Operating System (ROS). The use of digital twins to enhance human-robot collaboration (HRC) in complex production systems was explored by Malik and Brem, 2021, presenting a case study, highlighting the potential advantages, and building blocks of digital twins in the field of collaborative robotics. In Sanfilippo, 2022, the author proposed a simulation case study for SAR operation combining modular robot, grasping, and locomotion capabilities. Thus, computer based virtual models of physical systems can be used to test and validate complex strategies and scenarios prior to their implementation in real world applications.

Building upon virtual models for disaster scenarios, a case study for a SAR simulation environment with HRC using snake robotics is presented in this study. The main contribution of this work includes the development of the control framework for HRC between a snake robot, first responders, and victims of a disaster. The proposed architecture distributes the control scheme of the HRC into four phases simulated in a virtual environment. In particular, the CoppeliaSim - formerly known as Vrep (Rohmer et al., 2013) is adopted as the virtual simulator. A model of a snake robot (Liljebäck et al., 2014) is used for locomotion on a dynamic path. The path is obtained at runtime by the first responders' UAV, which is deployed on top of a maze/disaster scenario. Regarding visual perception a camera is attached on the head of the snake for path following, while another camera is placed at the bottom of the UAV for path planning. To prove the efficacy of the design, three different maze case studies are presented which combine locomotion, grasping, dropping, path following and obstacle avoidance.

This paper is organised as follows. A review of the related research work is given successively in Section 2. Then, the model of the snake robot is provided in Section 3. Subsequently, the proposed framework architecture is outlined in Section 4. Simulation results are presented in Section 5. Finally, conclusions and future works are discussed in Section 7.

2. Related Work

Snake robots have gained significant attention in the field of robotics due to their unique locomotion capabilities and potential applications in disaster scenarios. These robots, inspired by the flexibility and maneuverability of biological snakes, offer a great potential for navigating complex and challenging environments. In rough terrains, steered vehicles often provide non feasible solutions between waypoints due to the kinematic and dynamic limitations, specially on curvatures, even when vehicles can rotate and turn in place (Eskandarian et al., 2019). Snake robots exhibit a range of physical configurations and purposes, although their movement is often inspired by snakes. These robots can differ in terms of redundancy, wheel usage, and even their ability to operate in both land and water environments. Their slender, elongated bodies with thin cross-sections make them particularly well-suited for exploring narrow spaces or pipes. The distribution of mass and the presence of multiple ground contact points contribute to their stability, especially when compared to other robotic designs like wheeled or multipedal systems (Hopkins et al., 2009). Snake robots promise impressive adaptability to various terrains, primarily relying on the roughness of the ground or obstacles to gain sufficient traction and move forward without slipping (Webster et al., 2006). This adaptability and stability in different terrains make them robust to mechanical failure, enabling exploration in uncertain and challenging environments. In terms of gait patterns, snake movement can be categorized into four categories (Seeja et al., 2022): (a) lateral undulation; (b) concertina; (c) rectilinear progression; and (d) side winding.

There are two distinct approaches to snake robot locomotion based on the understanding of the environment: obstacle accommodation/exploitation locomotion and obstacle avoidance locomotion. In cluttered environments, snake robots exploit obstacles as an aid for propulsion purposes. This is known as "obstacle-aid Locomotion" (OAL) (Holden et al., 2014). Snakes utilize a strategy of pushing against unevenness or irregularities in the environment, creating bends in their body. This bending pattern is propagated from the head to the tail, allowing for smoother locomotion. However, this method is heavily reliant on the friction present in the environment, and collisions with obstacles can hinder further motion, potentially causing mechanical stress or damage to the equipment. The significance of environment perception, mapping, and representation cannot be overstated. In fact, our research group has introduced the term ”perception-driven obstacle-aided locomotion” (POAL) to underscore this concept. POAL refers to a locomotion approach where a snake robot leverages its sensory-perceptual system to utilize the surrounding operational space. It identifies walls, obstacles, or external objects as means of propulsion. Our group’s work on POAL has been documented in several publications (Sanfilippo, Azpiazu, Marafioti, Transeth, et al., 2017; Sanfilippo, Azpiazu, Marafioti, Transeth, et al., 2016; Sanfilippo, Stavdahl, et al., 2016). To facilitate the design and simulation of POAL, our research group has developed SnakeSIM, a virtual rapid-prototyping framework. SnakeSIM enables researchers to engage in the safer, faster, and more efficient design and simulation of POAL (Sanfilippo, Stavdahl, and Liljebäck, 2017, 2018). In terms of control, attaining POAL necessitates precisely identifying possible push-points and properly determining feasible contact response forces. Because of the lack of compliance, achieving this with typical rigidly-actuated robots is exceedingly difficult. To address this challenge, our research group has developed Serpens, a novel modular snake robot equipped with series elastic actuators (SEA). Serpens is notable for its low cost, open-source nature, and high compliance, making it suitable for various applications. We recently introduced Serpens in our research publications (Duivon et al., 2022; Sanfilippo et al., 2019). Regarding guidance, a biologically inspired steering controller was presented in (Rañó et al., 2018). With respect to navigation, a local path planning algorithm for snake robots was introduced in (Hanssen et al., 2020).

Various investigations in literature have concentrated on motion dealing with obstacle avoidance. In this context, the environment perception, mapping, and representation play a crucial role in the overall model. These elements are fundamental for the successful functioning and decision-making of the snake robot in its environment. An example of snake robot motion is the artificial potential field (APF) based locomotion, as described by Davy, 2002. This approach involves creating an artificial field around objects, and the robot’s motion is designed to avoid this force field. Another algorithm utilized in snake robot locomotion is the central pattern generator (CPG), mentioned by Nor and Ma, 2014. CPG enables the robot to navigate around obstacles or barriers by adjusting the turning of its body from its intended trajectory.

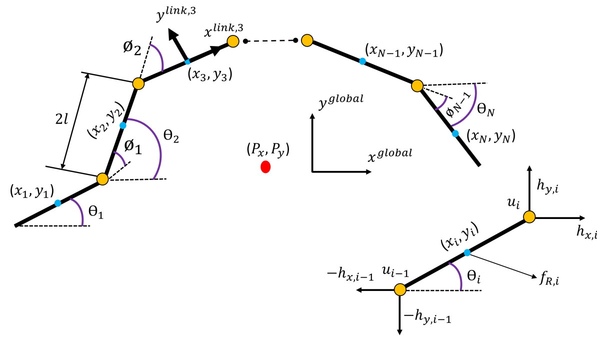

Figure 1: Kinematic and Force / Torque parameters of a snake robot

In the field of snake robotics, most of the past literature focuses on specific, static scenarios rather than the possibility of exploring dynamically changing and unpredictable scenarios. This represents a significant research gap. While numerous studies have investigated the locomotion and control mechanisms of snake robots in controlled environments, there is a lack of comprehensive research considering real-world situations where the environment and task requirements dynamically evolve. Such scenarios, which involve navigating through complex and unpredictable terrains, pose unique challenges that need to be addressed to enhance the adaptability and robustness of snake robots. Furthermore, the potential for collaboration between snake robots and other robots or humans is almost untapped. Investigating how snake robots can efficiently worked together and engage collaboratively with other entities opens up new opportunities for applications in fields such as SAR, exploration, and HRC. Bridging these research gaps will help to the advancement of snake robot capabilities and their practical deployment in real-world circumstances.

3. Modelling

To derive a kinematic model for the locomotion of the snake robot on a horizontal and flat surface, Liljebäck et al., 2013 proposed a linearisation of the model due to the many degrees of freedom and the dynamical couplings between links of the robot. A snake robot consists of N rigid links of length $2l$ joined by $N - 1$ joints. Each link is assumed to have the same mass m, thereby the center of mass of each rigid link is at the center point and the total mass of the snake comes out to be $N \times m$. The mathematical model is described in terms of the kinematic parameters illustrated in Figure 1.

The snake robot moves on a horizontal surface with

$N + 2$ degrees of freedom. The heading (orientation) of the snake robot is denoted by $\overline{\theta}$ and is defined as the average of link angles as described in equation (1).

$$\overline{\theta} = \frac{1}{N} \times \sum_{i=1}^{N} \theta_i \tag{1}$$

where link angles (θ_i) are defined as the angle that the link forms with the global x axis. On the other hand, joint angles, denoted as ϕ_i, is different such that it defines the difference between the link angles between each link i.e. $\theta_i - \theta_{i+1}$. The global position of the snake robot is given by:

$$\mathbf{p} = \begin{bmatrix} p_x \\ p_y \end{bmatrix} = \begin{bmatrix} \frac{1}{Nm} \sum_{i=1}^{N} m x_i \\ \frac{1}{Nm} \sum_{i=1}^{N} m y_i \end{bmatrix} \tag{2}$$

Following this, the forward velocity of the snake robot movement is defined as a component of the center of mass velocity ($\dot{p}$) and the current heading, i.e. as:

$$\bar{v}_t = \dot{p}_x \cos \overline{\theta} + \dot{p}_y \sin \overline{\theta} \tag{3}$$

Figure 1 shows the joint forces and the friction forces acted upon to a link i. Using the first principle of motion a dynamic model can be described for the whole snake robot in matrix form as:

$$\begin{aligned} m\ddot{\mathbf{X}} &= \mathbf{f}_{R,x} + \mathbf{D}^T \mathbf{h}_x \\ m\ddot{\mathbf{Y}} &= \mathbf{f}_{R,y} + \mathbf{D}^T \mathbf{h}_y \end{aligned} \tag{4}$$

where $\ddot{\mathbf{X}} = [\ddot{x_i}, \ddot{x_{i+1}} \ldots \ddot{x_N}]$, $\ddot{\mathbf{Y}} = [\ddot{y_i}, \ddot{y_{i+1}} \ldots \ddot{y_N}]$, $\mathbf{f}_{R,x}$, $\mathbf{f}_{R,y}$ are the ground friction forces. $\mathbf{h}_x = [h_{x,1} \ldots h_{N,1}]$ and $\mathbf{h}_y = [h_{y,1} \ldots h_{N,1}]$ are defined as the matrix for the joint constraint forces $h_{x,i}$ and $h_{y,i}$ respectively. The torque balance equation for the link i is given by:

$$\begin{aligned} J\ddot{\theta}_i = u_i - u_{i-1} - l \sin \theta_i \left(h_{x,i} + h_{x,i-1}\right) + \\ l \cos \theta_i \left(h_{y,i} + h_{y,i-1}\right), \end{aligned} \tag{5}$$

where u_i is defined as the torque forces exerted on the link from the next link in the chain of links of the snake robot. By using matrix form and introducing state variables the dynamic model of the snake robot can be compactly described in a state space form as

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\boldsymbol{\theta}} \\ \dot{\mathbf{p}} \\ \ddot{\boldsymbol{\theta}} \\ \ddot{\mathbf{p}} \end{bmatrix} = \mathbf{F}(\mathbf{x}, \mathbf{u}) \tag{6}$$

where elements of the $\mathbf{F}(\mathbf{x}, \mathbf{u})$ can be found in the mathematical breakdown provided by Liljebäck et al., 2013.

4. Framework Architecture

As introduced by Sanfilippo, 2022, the selected control framework is organized hierarchically, as shown in Figure 2. The input layer enables the robot to be guided by a human operator to achieve teleoperation or by other external systems (for example, an external planner) to reach higher levels of autonomy.

The core layer is the only layer required to perform the standard functions and capabilities of guidance, navigation, and control (GNC):

- Guidance: this concerns the process of identifying the desired course or trajectory for the snake robot to follow. It includes the decision-making process that specifies the robot's objectives and constraints. To generate commands that direct the robot along a desired path, the guidance system considers the robot's environment, mission requirements, and any other relevant parameters;

- Navigation: it involves the decision-making process regarding the optimal movement of the snake robot, including determining the appropriate location, timing, and method of locomotion. This decision-making process takes into account both external system commands and the sensory data gathered by the snake robot. The desired outcome of the navigation process is to generate a trajectory that includes path and velocity information for the robot to follow;

- Control: it serves as the central component of the presented control framework, offering researchers the flexibility to develop alternative control methods. The inputs to the control module include the desired trajectory and pertinent information obtained from the guidance level, such as perception data. The objective is to determine the necessary setpoints for the robot's actuators, enabling it to accurately track the desired trajectory.

In this work, a planner is added to the previously presented framework architecture (Sanfilippo, 2022). The planner considers four phases of HRC; 1) deployment (HRC): a first responder deploys the snake robot, a UAV, and a first aid item; 2) path planning and sharing (robot to robot collaboration): the UAV captures the planimetry of the area of interest and generates a path (e.g. shortest path) that the snake has to follow to

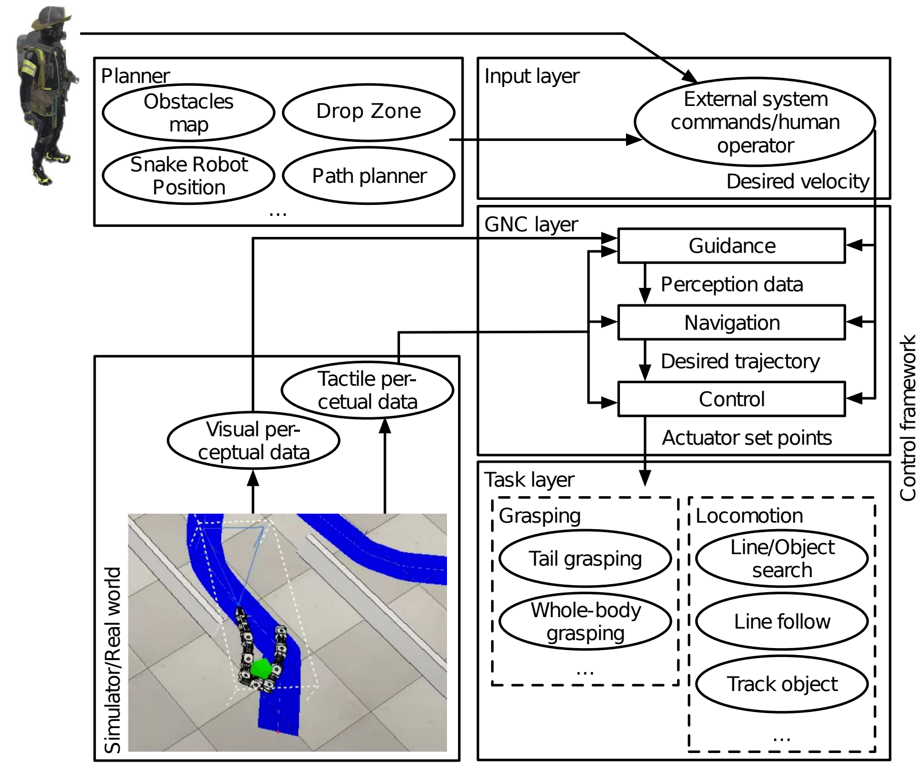

Figure 2: The proposed framework architecture.

reach the victim to be rescued. The path is shared by the UAV to the snake robot; 3) grasping and locomotion: the snake robot grasps the first aid item and locomotes to reach the victim to be rescued; 4) releasing first aid item and return: the snake robot delivers and drops the first aid item to the victim to be rescued. Successively, the snake robot returns to the first responder.

The task layer encompasses a variety of tasks that have clearly defined objectives to accomplish. The following tasks have been implemented:

- Line follow: in this task, the robot employs its visual sensor to track a designated line (Kelasidi et al., 2017). Through the utilisation of a proportional integral derivative (PID) controller, the snake robot computes the required adjustments to its locomotion parameters, ensuring that the line remains within the camera's field of view.

- Line search: in case the vision sensor loses the path following line, for example during sharp turns, this task performs the operation of rotating and exploring to re-adjust its position to get back on track of the path. The task makes the robot head rotate left and right while making small steps forwards or backwards to find the line.

- Object search: once the Emergency item is detected by the snake robot vision sensor, the line following control scheme is replaced by the object tracking and grasping scheme.

- Locate drop zone: once the drop zone near the victim's position is detected by the snake robot vision sensor, the line following control scheme is replaced by the object drop sequence.

- Track object: upon detection of an object with a specific color, this particular task employs a PID controller to compute the precise adjustments required for the snake robot's parameters. These adjustments are aimed at maintaining the object within the camera's field of view, thus ensuring continuous tracking.

- Pregrasping: in this task, it is assumed that the object to be grasped is positioned in front of the snake robot and is visible to the camera prior to initiating the execution. The snake robot begins by executing a continuous bending motion until the object to be grasped is no longer within the camera's field of view. Subsequently, a series of

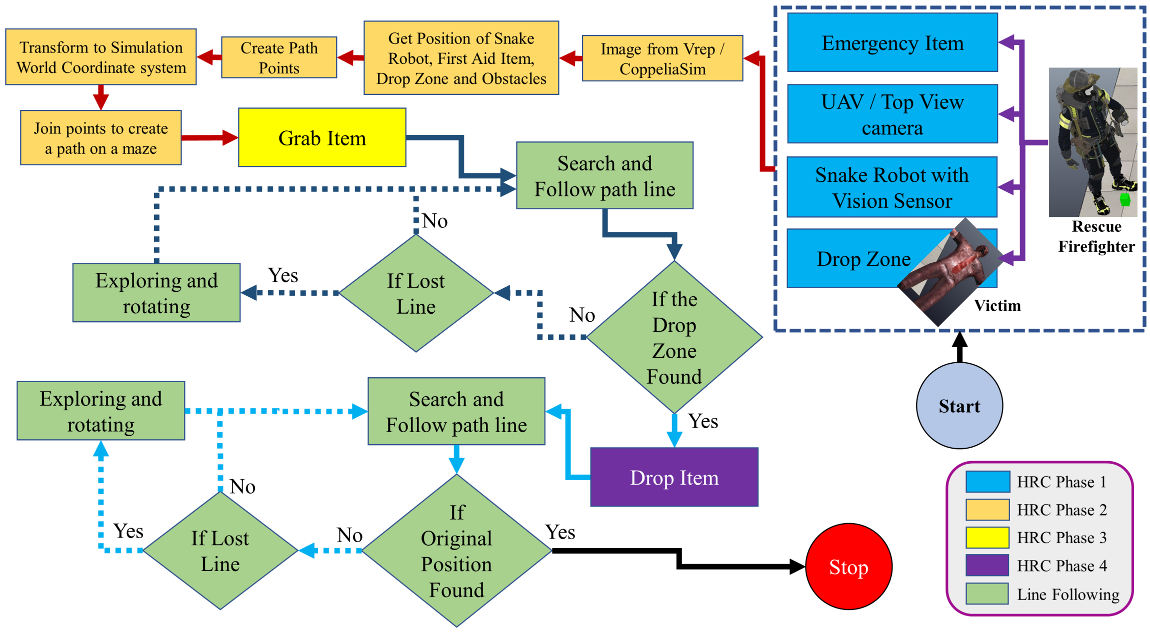

Figure 3: Simulation flowchart.

forward steps are performed by the snake robot, followed by a continuous bending motion of the head, aiming to detect a collision with the object to be grasped. This sequence is repeated until the object is determined to be in an optimal position for grasping. In the event that the object to be detected does not experience collision, the snake robot moves backwards until the object to grasp is back into the field of view of the vision sensor.

- Grasping: after the pregrasp task is completed, a series of bending maneuvers is then executed to transition the snake robot into the whole-body grasping posture. The snake robot adapts its shape to the object to grasp by determining minimum number of modules (n_{min}) needed to accomplish grasping according to equation (7).

$$n_{\min} = \frac{C_{obj}}{l_m} \tag{7}$$

where C_{obj} minimum circular length of object and l_m is the length of the module. To ensure the object remains correctly positioned throughout this sequence, torque sensing is implemented at the joint level. Once the bending procedure concludes, collision detection is employed to verify that the object is correctly positioned within the body of the snake. With the object grasped, the snake robot rotates itself until the path line is in front of the vision sensor, thereby, the line following locomotion begins.

- Drop zone tracking: this task ensures the drop zone is in front of the snake and within the camera's field of view via PID controller for precise adjustments.

- Dropping: the snake robot advances towards the drop zone, continuing until it is no longer visible within the camera's field of view. At this point, the snake robot will proceed to move forward for a predetermined duration, based on empirical observations. Following this, the snake robot will come to a stop and release the object by straightening its body, with the center section of the snake pushing the object onto the drop zone.

- Return to line: after the snake robot drops an item, it moves backwards for a predefined number of steps and then performs a sinusoidal rotatory movement to get back on track of the path line.

At the aftermath of a disaster event, first responders arrive at the scene to provide assistance. Due to obstructions from the fallout of the disastrous event, the victim is unreachable by the rescue firefighter. In the first phase of the HRC, the rescue responder deploys the snake robot and an emergency item (e.g. a water tank or an oxygen tank) in close proximity, while a UAV is launched on top of the area of interest, which is cluttered

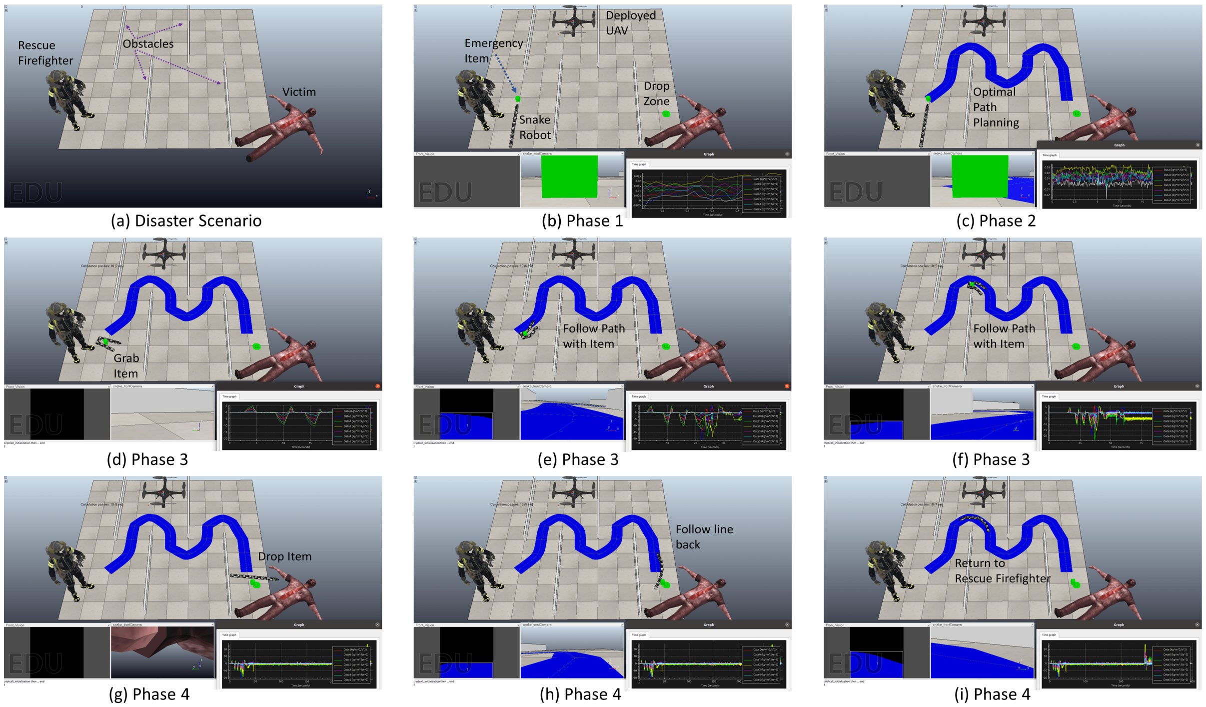

Figure 4: A sequence of successive screenshots for the selected four phase HRC system case study. The screenshots contain the simulated environment, raw and processed video streams of the snake vision sensor and robot joint's (7 in total) force / torque measurements.

with obstacles. Since the environment is dynamic and unknown to the snake robot, assistance from the rescue responders is required. The UAV is equipped with a vision sensor to create a map of the environment. The UAV is responsible for mapping out positions of key elements in the area of interest and creating an ideal path for the snake robot to reach the victim to be rescued and deliver the first aid item. The control framework for the mapping task procedure of the second phase is itemized as follows:

- Obstacles Map: this task uses the static image taken from the UAV to mark positions of the obstacles in the maze. It also marks the boundary regions creating a bound space of the map for the snake robot traversal.
- Snake robot position: this task compares the pre-defined template of a snake robot with every area in the map to determine the position and orientation of the robot. The line path points to be made start from head of the snake robot.
- Drop zone position: this task determines the position of the drop zone on the maze nearest to the victim. The line following path will end near this position.
- Path planner: once the requisite positions are determined. The UAV creates path points with small increments on the simulation environment. The path starts from the snake's position and ends near the drop zone. The control scheme involves a A* algorithm (Wang et al., 2015) with the obstacles as the heuristics of the algorithm for optimal path planning.
- Path creation: this task takes the path points and converts them according to the simulation world environment. These points are then embedded on the map maze. A line is then joined between each point to create the path on the maze which the snake robot will follow.

With this, the system enters the third phase of the HRC process. The snake robot grabs the item from the rescue firefighter and positions itself to get on track of the path line. The robot follows the path line to the position of the victim. When the robot's vision sensor detects the drop zone near the victim, the forth phase of the HRC system initiates. In this phase, the robot performs the tasks of dropping the object on the drop zone. After task completion the snake robot re-adjusts itself back onto the line and follows the path line back to the first responder, consequently, ending the simulation. The state machine flowchart of the four state HRC phase are shown in Figure 3.

5. Simulation Results

Due to the complexity of the environment, real-world development of control algorithms for snake robots can be challenging. Testing novel control approaches can potentially damage both the environment and the snake robot and can be time consuming. A realistic simulator is much more efficient for the development of control strategy. Coppelia Sim (Rohmer et al., 2013) is chosen as the simulation environment in this study because it is a flexible simulation framework that supports multiple operating systems. Each module can be controlled via embedded script, plugins, a remote application programming interface (API) client or a user-defined solution. Lua lightweight, multi-paradigm programming language (Ierusalimschy et al., 2012) - created in 1993 - is used within the Coppelia Sim simulator.

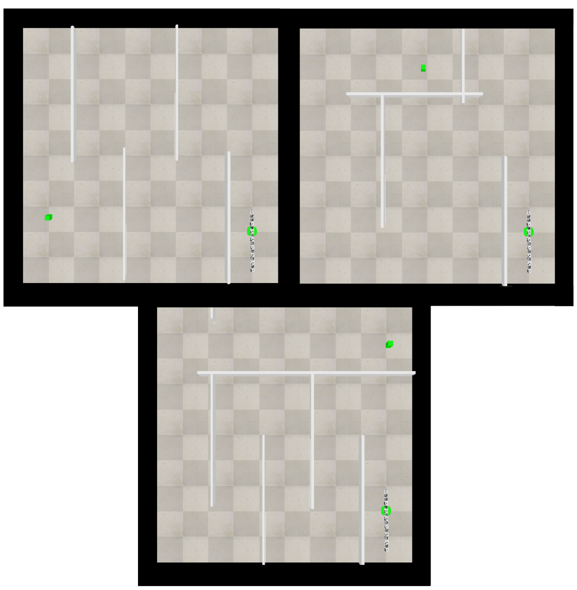

Figure 5: Map sample cases.

To demonstrate the potential of the proposed framework for SAR operations, the case study presented offers simulation results for three different map cases, as shown in Figure 5. The purpose of the operation is for the snake robot to retrieve a distinctly coloured object (marked as a green cube) from a rescue firefighter and bring it to the victim in need (at the drop zone). The simulated autonomous planner of this operation in the simulation is divided into two main parts to accomplish its objective:

- A remote API - via Jupyter Notebook (Kluyver et al., 2016) - initializes the simulation, performs all the image recognition tasks of the UAV and determines the path points, using the A* algorithm, from the snake robot to the drop zone position. These path points are then converted to the simulation cartesian coordinate system and sent to the Coppelia Sim simulator. A child script in the simulator receives these points and draws a path onto the maze.

- The snake robot starts its locomotion (rectilinear progression) by first performing the pregrasping / grasping operation to get the emergency item and place the robot where the path line is in front of the snake head vision sensor. By avoiding obstacles, the snake robot traverses toward the drop zone following the path line. The snake robot then drops the object onto the drop zone, readjusts itself back to the path line and returns to the original position near the first responder, consequently ending the simulation.

A sequence of successive screenshots for the selected four phase HRC system case study and one of the maze is shown in Figure 4. The project open source repository with documentation and demo videos is available on-line at https://github.com/Dohvakiin793/Cobots-for-Emergency-Situations.git.

6. Discussion

A disaster scenario, man-made or natural, consists of many unknown variables and dangers. Human intervention becomes difficult if the risk of falling debris is a factor. In such situations a team of robots being deployed would be considered as an ideal scenario for SAR operations. Bio-inspired snake robots can traverse a variety of challenging terrains, such as narrow paths, uneven surfaces, as well as gravel, and debris, among others, where robots with different mobility systems may encounter significant difficulties. The snake like structure of the robot aids in locomotion into tight spaces. The collaborative synergy between snake robots and human responders amplifies the impact of SAR efforts, enabling the possibility to reach locations that otherwise might be unapproachable. Furthermore, the modular nature of snake robots allows for customisation and adaptability. Different modules can be attached or detached enhancing their versatility. This modularity aligns with collaborative efforts, as snake robots can be equipped with various sensors, cameras, or tools to aid in data collection, assessment, and interaction with the environment. Snake robots, while innovative and adaptable, come with inherent limitations. Energy efficiency poses a concern due to the energy-intensive movement and payload capacity is fairly limited. Furthermore, unique control interfaces, communication reliability, slower speeds, and ethical considerations must be addressed for effective collaboration. Balancing these drawbacks with their benefits necessitates ongoing research and tailored deployment strategies. Some of these disadvantages may potentially be compensated by collaborating with other robots with different mobility systems.

7. Conclusion and Future Work

The adoption of collaborative robots (cobots) by disaster managers heavily relies on their capabilities, reliability, and robustness during field deployments. The extent of autonomy exhibited by robotic systems not only influences the manpower needed for their operation but also determines the complexity and adaptability of the system. However, achieving full autonomy in real-world rescue scenarios is currently challenging and not readily applicable in practical situations. Nevertheless, there is a clear inclination towards incorporating semi-autonomous behaviors instead of relying solely on manual control. This approach aims to alleviate the cognitive burden on the operator, enabling them to multitask or operate multiple systems concurrently. However, it is crucial to involve humans in the decision-making loop to guide the robot's actions, particularly in tasks that involve dynamic changes throughout the mission, such as search and rescue operations during disaster scenarios. This paper proposes a human robot collaborative (HRC) environment for disaster scenarios. A simulation using the Coppelia Sim (Rohmer et al., 2013) interface is used, in which a snake robot is adopted to traverse through three different maze disaster scenarios cluttered with obstacles to grab an emergency item from the rescue firefighter, such as an oxygen tank or a water tank and provide it to the victim. The robot follows a predefined path set by a UAV using the A* algorithm deployed by the rescue firefighter. The control architecture framework considers four phases of a Human Robot Collaborative (HRC) system. Our research group aims to determine the efficacy of such a system in a physical environment for the future. Furthermore, shortest path algorithms can be better adjusted using metaheuristic algorithms for optimal path planning. Physical obstacles may also be replaced with synthetic fire (in mixed reality) for obstacle avoidance.

Acknowledgement

This work is supported by the Centre for Integrated Emergency Management (CIEM) and by the Biomechatronics and Collaborative Robotics research group, Top Research Centre Mechatronics (TRCM), University of Agder (UiA).

References

Bertolino, M., & Tanzi, T. J. (2019). Advanced robot 3d simulation interface for disaster management. *2019 Kleinheubach Conference*, 1–4.

Cova, T. J. (1999). Gis in emergency management. *Geographical information systems*, *2*(12), 845–858.

Davy, A. J. (2002). Establishment and manipulation of plant populations and communities in terrestrial systems. In *Handbook of ecological restoration* (pp. 223–241). Cambridge University Press.

Duivon, A., Kirsch, P., Mauboussin, B., Mougard, G., Woszczyk, J., & Sanfilippo, F. (2022). The redesigned serpens, a low-cost, highly compliant snake robot. *Robotics*, *11*(2), 42.

Eskandarian, A., Wu, C., & Sun, C. (2019). Research advances and challenges of autonomous and connected ground vehicles. *IEEE Transactions on Intelligent Transportation Systems*, *22*(2), 683–711.

Hanssen, K. G., Transeth, A. A., Sanfilippo, F., et al. (2020). Path planning for perception-driven obstacle-aided snake robot locomotion. *Proc. of the IEEE 16th International Workshop on Advanced Motion Control (AMC)*, 98–104.

Holden, C., Stavdahl, Ø., & Gravdahl, J. T. (2014). Optimal dynamic force mapping for obstacle-aided locomotion in 2d snake robots. *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 321–328.

Hopkins, J. K., Spranklin, B. W., & Gupta, S. K. (2009). A survey of snake-inspired robot designs. *Bioinspiration & Biomimetics*, *4*(2), 021001.

Ierusalimschy, R., Celes, W., & de Figueiredo, L. H. (2012). Lua: The programming language.

Jeyson, C. A., Luque, S., Talavera, J., et al. (2022). Analysis of the intuitive teleoperated system of the txrob multimodal robot. *International Journal of Advanced Computer Science and Applications*, *13*(10).

Kelasidi, E., Liljebäck, P., Pettersen, K. Y., & Gravdahl, J. T. (2017). Integral line-of-sight guidance for path following control of underwater snake robots: Theory and experiments. *IEEE Transactions on Robotics*, *33*(3), 610–628.

Kluyver, T., Ragan-Kelley, B., Pérez, F., et al. (2016). *Jupyter notebooks-a publishing format for reproducible computational workflows.* (Vol. 2016).

Liljebäck, P., Pettersen, K. Y., Stavdahl, Ø., & Gravdahl, J. T. (2013). *Snake robots: Modelling, mechatronics, and control*. Springer.

Liljebäck, P., Stavdahl, Ø., Pettersen, K. Y., & Gravdahl, J. T. (2014). Mamba - a waterproof snake robot with tactile sensing. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 294–301.

Magid, E., Pashkin, A., & Simakov, N. o. (2019). Artificial intelligence based framework for robotic search and rescue operations conducted jointly by international teams. *Proceedings of 14th International Conference on Electromechanics and Robotics "Zavalishin's Readings" ER (ZR) 2019, Kursk, Russia, 17-20 April 2019*, 15–26.

Malik, A. A., & Brem, A. (2021). Digital twins for collaborative robots: A case study in human-robot interaction. *Robotics and Computer-Integrated Manufacturing*, *68*, 102092.

Munich Re. (2023). *Natural disasters in 2022* [Accessed: 2023-05-29]. https://www.munichre.com/en/company/media-relations/media-information-and-corporate-news/media-information/2023/natural-disaster-figures-2022.html

Nor, N. M., & Ma, S. (2014). Cpg-based locomotion control of a snake-like robot for obstacle avoidance. *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 347–352.

Rañó, I., Eguíluz, A. G., & Sanfilippo, F. (2018). Bridging the gap between bio-inspired steering and locomotion: A braitenberg 3a snake robot. *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 1394–1399.

Rohmer, E., Singh, S. P. N., & Freese, M. (2013). V-rep: A versatile and scalable robot simulation framework. *IEEE International Conference on Intelligent Robots and Systems*, 1321–1326.

Sanfilippo, F. (2022). Combining grasping with adaptive path following and locomotion for modular snake robots. *International Journal of Mechanical Engineering and Robotics Research*, *11*(2), 59–65.

Sanfilippo, F., Azpiazu, J., Marafioti, G., Transeth, A. A., et al. (2017). Perception-driven obstacle-aided locomotion for snake robots: The state of the art, challenges and possibilities. *Applied Sciences*, *7*(4), 336.

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

Sanfilippo, F., Helgerud, E., Stadheim, P. A., & Aronsen, S. L. (2019). Serpens, a low-cost snake robot with series elastic torque-controlled actuators and a screw-less assembly mechanism. *Proc. of the IEEE 5th International Conference on Control, Automation and Robotics (ICCAR)*, 133–139.

Sanfilippo, F., Stavdahl, Ø., & Liljebäck, P. (2017). Snakesim: A ros-based rapid-prototyping framework for perception-driven obstacle-aided locomotion of snake robots. *Proc. of the IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 1226–1231.

Sanfilippo, F., Stavdahl, Ø., & Liljebäck, P. (2018). SnakeSIM: A ros-based control and simulation framework for perception-driven obstacle-aided locomotion of snake robots. *Artificial Life and Robotics*, *23*, 449–458.

Sanfilippo, F., Stavdahl, Ø., Marafioti, G., Transeth, A. A., & Liljebäck, P. (2016). Virtual functional segmentation of snake robots for perception-driven obstacle-aided locomotion? *Proc. of the IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 1845–1851.

Seeja, G., Selvakumar Arockia Doss, A., & Hency, V. B. (2022). A survey on snake robot locomotion. *IEEE Access*, *10*, 112100–112116.

Wang, C., Wang, L., Qin, J., Wu, Z., et al. (2015). Path planning of automated guided vehicles based on improved a-star algorithm. *2015 IEEE International Conference on Information and Automation*, 2071–2076.

Webster, R. J., Okamura, A. M., & Cowan, N. J. (2006). Toward active cannulas: Miniature snake-like surgical robots. *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2857–2863.