Simposio CEA de Robótica, Bioingeniería, Visión Artificial y Automática Marina 2025

Modeling an underground tunneling robot for pipeline construction

Colazo, Agustín[a,*], Menendez, Elisabeth[a], Martínez, Santiago[a], Balaguer, Carlos[a]

[a] RoboticsLab. Dpto. de Ingeniería de Sistemas y Automática, Universidad Carlos III de Madrid, C/Butarque, nº 15, 28911, Madrid, España.

Resumen

La robótica subterránea presenta retos complejos y sigue estando poco explorada en comparación con la robótica estacionaria y móvil de superficie. Mientras que los simuladores proporcionan un medio de bajo costo y seguro para estudiar el diseño mecánico y los algoritmos de control del movimiento, sus capacidades para estudiar sistemas subterráneos son más limitadas. En particular, a la hora de modelar interacciones complejas con el suelo o mecanismos de locomoción híbridos. Los simuladores carecen de herramientas para estudiar sistemas como los robots con locomoción de gusano, que anclan y liberan segmentos para moverse. Este artículo presenta el modelado del robot subterráneo ROBOSUB, diseñado para la construcción de tuberías, utilizando Simscape Multibody. El robot se ancla y libera de las paredes del túnel para desplazarse. En primer lugar, se describe el sistema robótico real y, a continuación, se analiza su implementación en Simscape Multibody. Finalmente, se valida el modelo mediante la ejecución de cuatro escenarios con diferentes señales de control, lo que da como resultado cuatro trayectorias distintas.

Palabras clave: Robótica de campo, Modelado y simulación, Sistemas robóticos autónomos, Control de movimiento

Abstract

Underground robotics presents complex challenges and remains underexplored compared to stationary and surface mobile robotics. While simulation software provides cost-effective and safe means to study mechanical design and motion control algorithms, its capabilities for underground systems are more limited. In particular, when modeling complex interactions with the soil or hybrid locomotion mechanisms. There is a lack of tools in simulation software to study systems such as robots with inchworm locomotion that attach and detach from surfaces to move. This paper introduces the Simscape Multibody model of the ROBOSUB robot, an underground robot designed for building pipelines. The robot anchors to and detaches from the tunnel walls to locomote. We first describe the real robotic system, then we discuss its implementation in Simscape Multibody. Finally, we validate the model by running four scenarios with different control inputs which result in four different trajectories.

Keywords: Field robotics, Modeling and simulation, Autonomous robotic systems, Motion control

1. Introduction

The usual approach to install underground utilities is open-cut excavation. This method involves digging a trench, installing the pipelines and backfilling. This process requires destroying any surface infrastructure along the desired path, leading to high environmental and economic impact, such as air pollution, traffic congestion, repair costs and noise. In contrast, trenchless technologies are rapidly growing in the construction sector because they enable pipeline installation without trenches, reducing the environmental impact compared to open-cut methods (Kumar et al., 2021). The BADGER robot, shown in Fig. 1, was the first of its kind underground autonomous robot for pipeline construction and is a clear example of a trenchless technology (Martinez et al., 2024).

Reliable simulation is a critical step for the analysis and design of robotic systems (Choi et al., 2021; Liu and Negrut, 2021). It enables safe and cost-effective development and testing of algorithms for control and planning. It is particularly valuable when physical testing is constrained by complexity, safety, time and cost. Simulations enable researchers and engineers to validate their concepts on simulated platforms before testing them on the real systems, thereby reducing the risk of damage, safety hazards and other potential failures. Robotic simulators are usually built for continuous dynamics and rigid contact, and inchworm locomotion with anchor-release cycles is not natively supported. Moreover, the discrete transitions between anchoring and releasing can pose challenges for physics

*Corresponding author: acolazo@pa.uc3m.es

solvers, potentially leading to instability or convergence issues during simulation.

In this paper we present the implementation of the Simscape Multibody model for the ROBOSUB robot, the successor of the BADGER robot. The ROBOSUB robot features parallel mechanical structures and inchworm locomotion, which is challenging to simulate in currently available simulation software. Simscape Multibody provides advanced solvers that are well-suited to simulate parallel structures, making it a powerful tool to simulate and develop motion control algorithms for this robot (Boschetti and Sinico, 2024). Furthermore, it is integrated with the Simulink environment that enables rapid prototyping by employing block diagrams and physical modeling connections.

The main contribution of this paper is a kinematically accurate model of the ROBOSUB robot suitable for testing motion control strategies. A secondary contribution is the implementation of a clamp mechanism in Simscape Multibody that anchors and releases a body in 3D space, a component that is useful to model systems that incorporate anchoring mechanisms. The model presented in this work serves as a foundation for future motion control research.

Figure 1: Badger robot

2. Robot description

The main purpose of the ROBOSUB robot is the installation of pipelines for services like fiber optics. To achieve that goal, the robotic system is capable of sensing the environment, detecting obstacles such as rocks and civil infrastructure, autonomous navigation, obstacle evasion, underground mapping and leveraging GIS data to generate maps for planning before starting the operation (Colazo et al., 2024; Menendez et al., 2019).

The robot employs an inchworm gait for locomotion (Worrall et al., 2019; Vartholomeos et al., 2021). This type of locomotion is intermittent: the robot anchors one segment to the tunnel walls while moving the remaining segments forward. This enables the robot to bore through soil, propel forward and steer.

The bore head module has a rotating cutter head, as in Tunnel Boring Machines, that breaks the soil (Worrall et al., 2019; Vartholomeos et al., 2021). Then, the soil is extracted to the surface by a removal system that runs through the robot, from the head to the surface.

The robot has two service modules that anchor a segment of the robot to the tunnel walls (Colazo et al., 2024; Worrall et al., 2019; Vartholomeos et al., 2021). When one service module is attached, the other one detaches to enable the other segments of the robot to move. The service module has air chambers that inflate to generate pressure against the tunnel walls, effectively anchoring the robot to the walls, and deflate to release the segment. Each service module is composed by a steel cylinder of 775 mm and an air chamber 350 mm long.

There are two joint modules for directional and forward movement (Colazo et al., 2024; Worrall et al., 2019; Vartholomeos et al., 2021). The joint modules are composed by a propulsion module and a steering module. The propulsion module has three parallel legs that are welded between two plates. Each leg has an actuated prismatic joint and the simultaneous actuation of the prismatic joints achieves forward motion up to 100 mm. The steering module is a parallel mechanism with a bottom plate, three parallel legs and a top plate. The three legs are at the edges of the plate and equidistant between each other. Each leg is welded to the bottom plate, then it has an actuated prismatic joint followed by a passive revolute joint and finally a passive spherical joint that connects to the top plate. The coordinated action of the prismatic joints results in the steering of the top plate, where each prismatic joint can extend up to 50 mm.

Finally, a gait sequence was designed that combines the action of the service modules and the joint modules to propel and steer the robot.

3. Model Implementation

In this section, we present the model for the robot using Simscape Multibody and the Robotics Toolbox (Corke, 2017). The robot model mirrors the real robot, therefore, it has two propulsion modules, two steering modules, two service modules and a bore head module. Additionally, there are a gait controller, a clamp controller, a path following module and transform sensors.

Fig. 2 shows the Simscape Multibody model of the entire robot. Manipulators can be simulated with a fixed base, while mobile robots like quadrotors and wheeled robots have a movable base. In Simscape Multibody, to allow a virtual robot to move freely with respect to the world frame it needs to have a passive 6 degrees-of-freedom (DOF) joint between the robot and the world frame, otherwise, it would be fixed to the world frame. Needless to say, this joint does not exist in the real system. As their real counterpart, the service modules anchor the corresponding segment in 3D space, the propulsion modules push forward, and the steering modules change direction. The model's physical subsystems are connected in the following order: passive 6 DOF joint, tail service module, tail propulsion module, tail steering module, head service module, head propulsion module, head steering module and bore head module. There are three transform sensors to get the current pose of different parts of the robot. These are connected to the bore head, the head steering module and the tail steering module. The transform sensors send their output to the path following module, which sends its output to the gait controller. The gait controller sends signals to the clamp controller, the propulsion modules and the steering modules. Finally, the clamp controller sends control signals to the service modules.

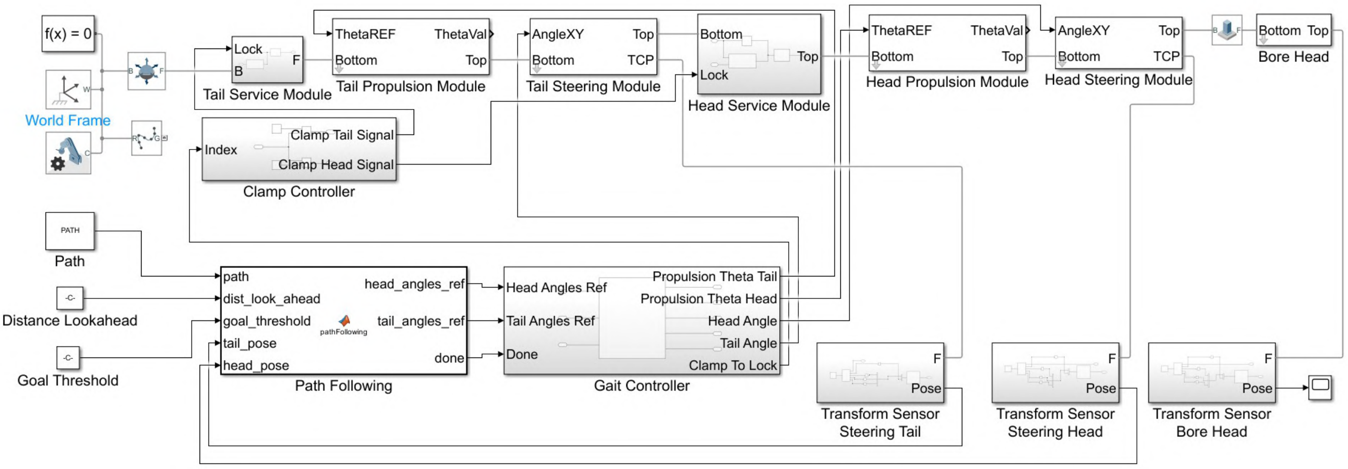

Figure 2: Robot Model

The purpose of the service modules is to anchor a segment of the robot in 3D space, simulating the effect of the air chambers attaching the robot against the tunnel walls. In the model implementation, only one of the service modules can be clamped at a given time while the other must be released. The clamp controller is responsible for commuting between both service modules. The clamp controller subsystem is simple (Fig. 3), it receives a signal and enables one of the clamps while disabling the other one. Both clamps cannot be activated at the same time. To keep the model simple and fast, we do not model the dynamics of inflation and deflation of the air chambers. The tail service module (Fig. 4) and the head service module (Fig. 5) are similar but not identical. Functionally, they are the same but due to how Simscape Multibody works the model has minor differences to simulate the anchoring and releasing. The tail service subsystem has one input, the lock signal, that goes into the clamp subsystem and a cylindrical solid representing the steel cylinder and air chamber. When the tail service subsystem is unlocked, it has unconstrained motion relative to the World Frame. Conversely, when locked, it remains fixed with respect to the World Frame. Similarly, the clamp of the head service module, when engaged, secures the module in a fixed pose relative to the World Frame. To enforce a rigid, zero-degree-of-freedom constraint between the tail modules and the head modules, the head service subsystem includes a parallel Weld Joint that connects to the tail steering module. This joint ensures that the two modules remain fixed at all times, as it prevents any relative motion between them. In the absence of this weld joint, the disengaged section would free fall as there would be no constraint mechanism keeping it fixed to the clamped section.

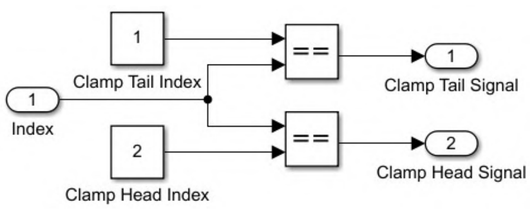

Figure 3: Clamp controller subsystem

The clamp subsystem (Fig. 6) was designed during this work for transitioning between anchoring a body in space and releasing it allowing free movement in a Simscape Multibody simulation. To the best of our knowledge, there is no other implementation in the literature that allows a body to commute between these two states for Simscape Multibody, usually simulated robots are either mobile bases or fixed bases. The Lock signal determines if the subsystem allows free movement or is fixed to a certain pose. The Transform Sensor measures the spatial relationship between the F frame and the World Frame, it is used to measure the pose of the body when it is moving freely (in released state). The measurements are multiplexed and passed through a delay subsystem and an Enabled Subsystem. Then, the signal is de-multiplexed and used to actuate a Bushing Joint (6 DOF). The result of this is that the Bushing Joint follows the F frame with a small delay. When the Lock signal is disabled, the Weld Joint is disengaged (it does not impose constraints on the system) so the F frame can move freely with respect to the Bushing Joint. Also, the Enabled Subsystem is activated, so the Bushing Joint inputs are updated. When the Lock signal is enabled, the Enabled Subsystem is deactivated and holds the most recent values. The Weld Joint is engaged, connecting the F frame to the output of the Bushing Joint, effectively holding the F frame to the current input value of the Bushing Joint.

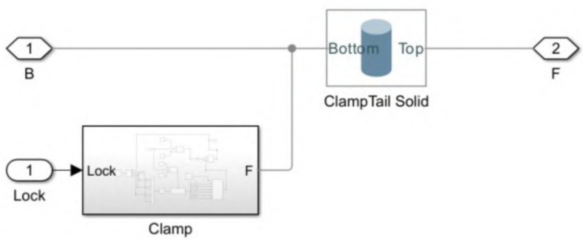

Figure 4: Tail service subsystem

The model of the propulsion module, shown in Fig. 7, follows a similar structure to other modules, featuring both an input and an output physical connection. In the real robot, this module has three parallel legs with prismatic actuators operating in unison. However, to improve simulation speed, it is modeled as a single open kinematic chain as it is kinematically equivalent to the three parallel legs when they actuate synchronously. The model has one leg welded to a plate and it has only one actuated joint. The Bottom Plate is attached to the top of the previous service module and it connects to the immovable part of the leg (Link0). Link0 connects to the prismatic joint which in turn connects to the Extender that moves up and down when the prismatic joint is actuated. Finally, the Extender and prismatic joint connect to the Top frame that connects to a steering module. The prismatic joint is controlled by position input and the force is automatically computed. Simscape Multibody does not support imposing force constraints when using actuators with automatically computed forces. As a result, the simulator may apply unrealistically large forces to achieve the reference position input and it results in quick unfeasible motions. To mitigate this, a feedback integrator is added, allowing the joint to gradually converge to the desired input and ensuring smoother joint motion.

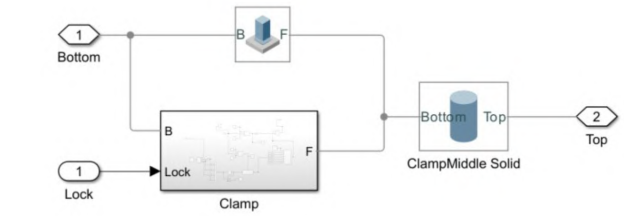

Figure 5: Head service subsystem

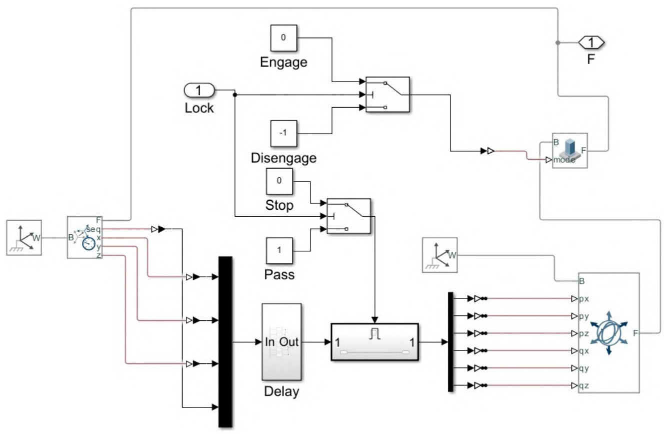

Figure 6: Clamp subsystem

The steering module subsystem (Fig. 8) is a kinematically accurate model of the three-legged steering module of the real system and is used for directional movement. The input frame connects to a solid cylinder that represents the bottom plate and to three parallel legs. The output frame of each leg connects to the Top Plate (imposing constraints between the legs) which connects to the output frame of the subsystem. Each leg receives the ThetaAct signal to actuate the prismatic joints and outputs the ThetaVal signal with the sensed prismatic joint value. The inverse kinematics controller receives the ThetaVal signal from each leg and the reference pose, and outputs the ThetaAct signal to control all the legs. Fig. 9 shows the expanded subsystem for a leg. Base-to-Leg is a Rigid Transform that translates each leg with the vector $(R * cos(\alpha), R * sin(\alpha), 0)$ from the input frame where α is 30°, 150° and 270° respectively. Each leg has an actuated prismatic joint that takes a position signal as input, a passive revolute joint and a passive spherical joint like the real module.

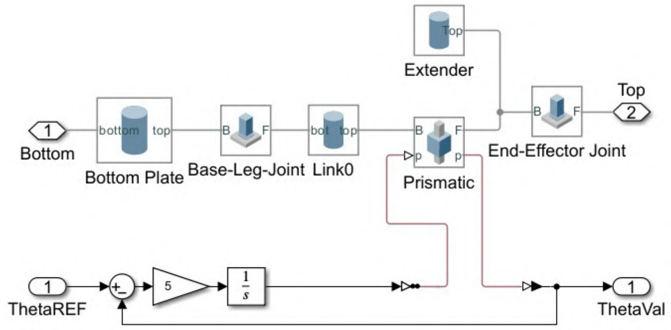

Figure 7: Propulsion module subsystem

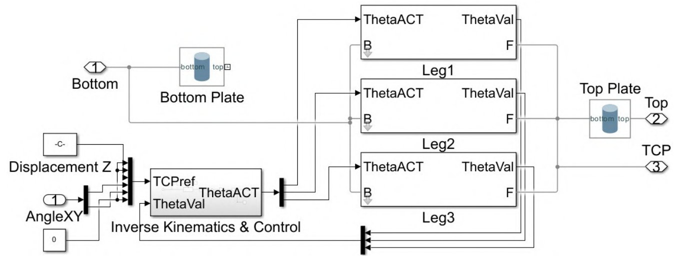

Figure 8: Steering module subsystem

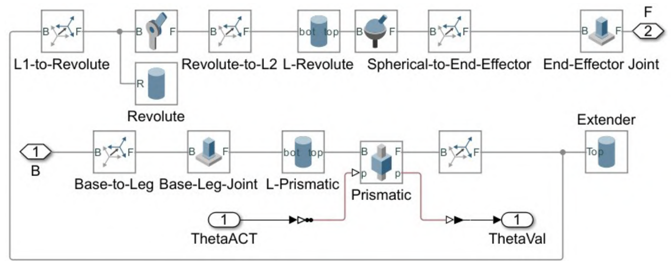

Figure 9: Leg subsystem

The final physical segment of the robot is the bore head module, represented by two solid bodies (Fig. 10). The base is modeled as a solid cylinder, while the cutter head is modeled as a cone using a revolved solid block. The rotation of the cutter head was not simulated but could be incorporated using a revolute joint. The interaction between the soil and the cutter head was not analyzed, as this is a complex process requiring careful modeling and analysis with methods such as finite element analysis.

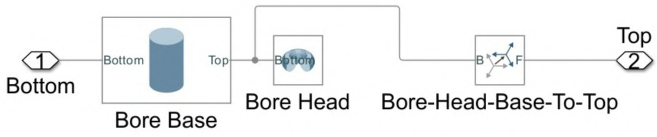

Figure 10: Bore head subsystem

Three Transform Sensor subsystems (Fig. 11) are used to measure the pose of the steering modules and the bore head module relative to the world frame. Each Transform Sensor block takes the world frame as the base frame and frame F as the target frame. The output consists of the XYZ translation and a rotation sequence. To avoid algebraic loops during simulation, all signals pass through a delay subsystem. The delay subsystem passes all signals through a first-order transfer function with a small time constant to minimize the effect on the dynamics.

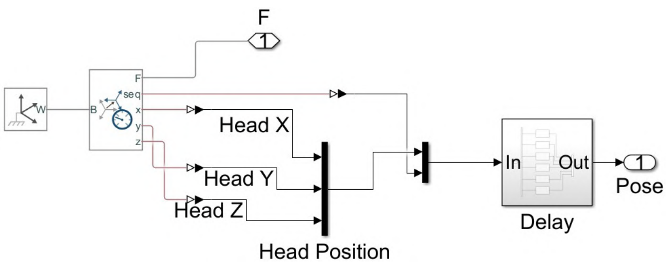

Figure 11: Transform Sensor subsystem

The measured poses are sent to the path following function, which computes the desired steering joint angles required to follow the reference path. Once the goal is reached, the path following module sets the done signal to true. The reference angles are sent to the gait controller, which executes the gait sequence by signaling to the clamp controller which service module to lock, and actuating the propulsion and steering modules. When the done signal is received, the gait controller ceases all commands to the clamp controller, the propulsion modules and the steering modules. A detailed discussion of these functions is beyond the scope of this paper.

4. Results

The complete robot model, including the service modules, propulsion modules, steering modules and the bore head, is shown in Fig. 12. We validate the Simscape Multibody model through four simulations. In these experiments, we provide constant angle references to the gait controller for the head and tail steering modules. Based on the current gait stage, the gait controller sends control signals to the clamp controller, the propulsion modules and the steering modules. The combined action of these subsystems results in the robot moving forward and steering depending on the inputs received.

We execute four different scenarios, each running for 500 simulated seconds. The first scenario consists of no steering, only forward motion by the propulsion modules. In the second, third and fourth scenarios, the gait controller receives constant reference angles for the head and the tail steering modules. The reference values and execution time for each scenario are summarized in Table 1. The rotation sequence used is XYZ, the value for Z is kept at 0° for all scenarios. The resulting trajectories for all four scenarios are depicted in Fig. 13.

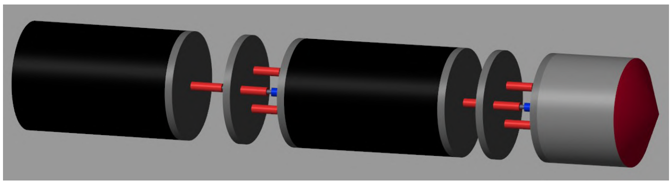

Figure 12: Robot model

Table 1: Scenarios

Scenarios	*RotX* [deg]	*RotY* [deg]	*Execution Time* [*s*]
1	0	0	40.093
2	1	0	384.110
3	0	1	176.969
4	1	1	325.668

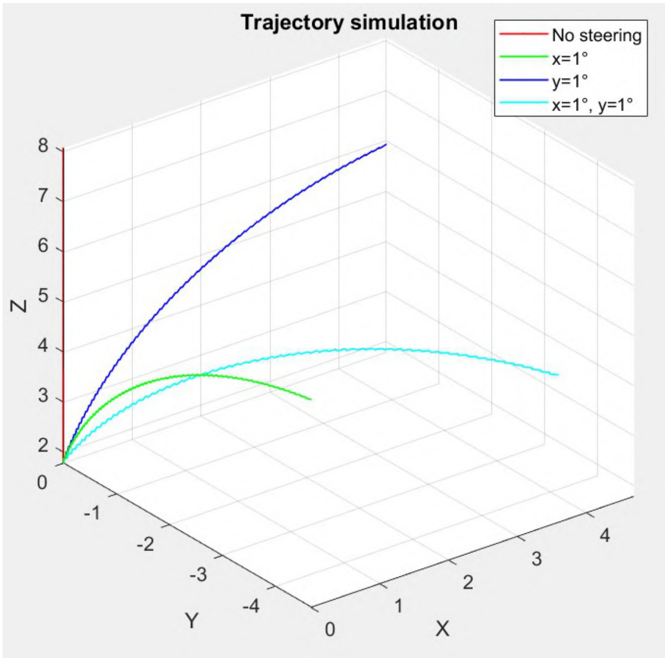

Figure 13: Trajectories for scenarios 1 (red), 2 (green), 3 (blue) and 4 (cyan)

Figures 14, 15 and 16 show the measured rotations for the tail steering module and the head steering module in scenarios 2, 3 and 4 respectively. To achieve steering, both mechanisms cyclically rotate to the target angles and then reset, following a periodic control pattern. In scenario 2, the modules rotate only about the X-axis; in scenario 3, only about the Y-axis; and in scenario 4, they rotate about both axes. The rotations are applied following an XYZ Euler angle sequence, allowing pitch and yaw steering.

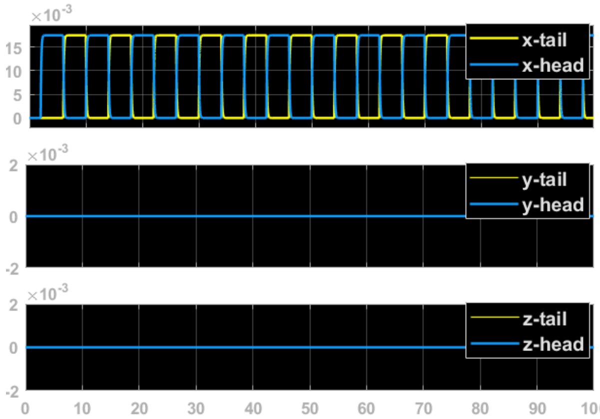

Figure 14: Scenario 2: Rotation XYZ (in radians) for head and tail steering modules

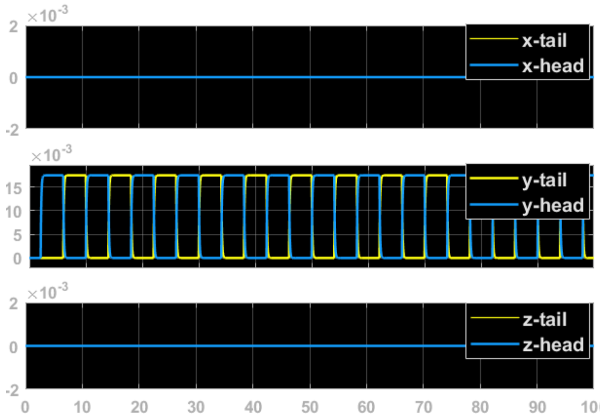

Figure 15: Scenario 3: Rotation XYZ (in radians) for head and tail steering modules

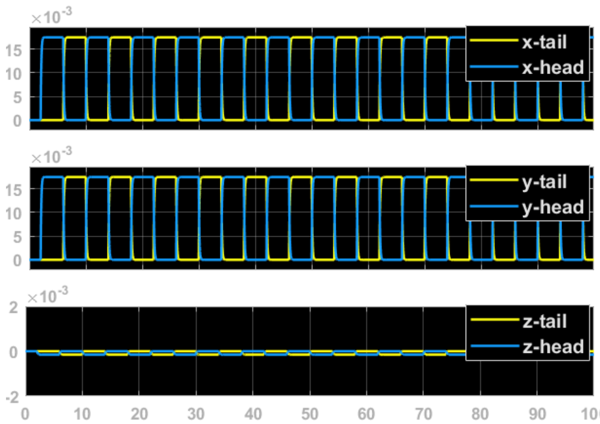

Figure 16: Scenario 4: Rotation XYZ (in radians) for head and tail steering modules

5. Conclusion

This paper presents in great detail the implementation of the 3D model of the ROBOSUB robot, an underground robot with inchworm locomotion based on anchor-release mechanisms, and its implementation in Simscape Multibody. The Simscape Multibody model captures the kinematics of the robot and its anchoring mechanism to achieve locomotion. The scenarios tested in this work demonstrate that the developed model produces different trajectories in response to the different control inputs.

Future works should improve the current model and use it to test new motion control schemes. While we captured the kinematics of the robotic platform, there is room for improving the dynamical model by modeling its internal forces and the interactions between the robot, the soil and the tunnel walls. The model enables the study and optimization of gait sequences, as well as the development of new motion control strategies.

Acknowledgments

This work is supported by the grant-funded project *ROBOSUB: "Robots inteligentes subterráneos para la transición ecológica y digital del subsuelo urbano"* (TED2021-129420B-I00), funded by MCIN/AEI/10.13039/501100011033 and the European Union's *"NextGeneration EU/PRTR"*; and by the project *iRoboCity2030-CM, Robótica inteligente para ciudades sostenibles* (TEC-2024/TEC-62), funded by Programas de Actividades I+D en tecnologías of the Community of Madrid.

References

Boschetti, G., Sinico, T., 2024. Designing digital twins of robots using simscape multibody. Robotics 13 (4).
URL: `https://www.mdpi.com/2218-6581/13/4/62`
DOI: `10.3390/robotics13040062`

Choi, H., Crump, C., Duriez, C., Elmquist, A., Hager, G., Han, D., Hearl, F., Hodgins, J., Jain, A., Leve, F., et al., 2021. On the use of simulation in robotics: Opportunities, challenges, and suggestions for moving forward. Proceedings of the National Academy of Sciences 118 (1), e1907856118.

Colazo, A., Sanz, G., Martinez, S., Balaguer, C., 07 2024. Application for occupancy grid generation for underground robot. Jornadas de Automática, 45.
DOI: `10.17979/ja-cea.2024.45.10953`

Corke, P., 01 2017. Robotics, Vision and Control. Vol. 118.
DOI: `10.1007/978-3-319-54413-7`

Kumar, K., Chandramouli, K., Sree, J., Chaitanya, N., Gowreswari, B., Chandramouli, K., Sree, N., Chaitanya, B., Gowreswari, Ijmtst, E., 07 2021. A review on trenchless technology. International Journal for Modern Trends in Science and Technology 7, 202–207.
DOI: `10.46501/IJMTST0707033`

Liu, C., Negrut, D., 05 2021. The role of physics-based simulators in robotics. Annual Review of Control, Robotics, and Autonomous Systems 4.
DOI: `10.1146/annurev-control-072220-093055`

Martinez, S., et al., 2024. BADGER: Intelligent Robotic System for Underground Construction. John Wiley & Sons, Ltd, Ch. 10, pp. 205–220.
URL: `https://onlinelibrary.wiley.com/doi/abs/10.1002/9781394162871.ch10`
DOI: `https://doi.org/10.1002/9781394162871.ch10`

Menendez, E., et al., 2019. uSLAM implementation for autonomous underground robot. In: 2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI). pp. 237–241.
DOI: `10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00083`

Vartholomeos, P., Marantos, P., Karras, G., Menendez, E., Rodriguez, M., Martinez, S., Balaguer, C., 2021. Modeling, gait sequence design, and control architecture of BADGER underground robot. IEEE Robotics and Automation Letters 6 (2), 1160–1167.
DOI: `10.1109/LRA.2021.3056068`

Worrall, K., Houston, C., Cebecauer, M., Flessa, T., Mcgookin, E., Thomson, D., Harkness, P., 2019. Design and implementation of a control system for a tunneling robot. In: 2019 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI). pp. 242–247.
DOI: `10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00084`