Deep Reinforcement Learning for Coordinated Payload Transport in Biped-Wheeled Robots

Dhruv K Mehta, Ajinkya Joglekar, Venkat Krovi

Abstract— **Coordinated payload transport via a fleet of modular wheeled mobile robots offers flexibility for handling larger loads in indoor and outdoor environments. Biped-wheeled robots have recently emerged as a viable architecture for an independent/stand-alone wheeled mobile robot. In this work, we explore the use of multiple biped-wheeled robots that can leverage their mobility and maneuvarability for enhanced spatial pose control and stabilization for various payload transport tasks. However, coordinated control of multiple articulated wheeled robots for path tracking of a payload presents significant (and potentially competing) challenges, including kinematic redundancy, stability concerns, relative motion between the payload and robots, and precise motion control to achieve effective coordination. To address these challenges, we propose a Deep Reinforcement Learning (DRL) framework to develop the motion-plans for the system. In particular, this approach generates the ego robot's body twist and the follower robot's relative twist with respect to the ego robot. By formulating the action space of the follower robot as a relative twist, our approach facilitates pairwise interactions between robots. Furthermore, we use only relative pose information and the errors as states for the DRL controller, thereby making it agnostic to initial conditions and avoiding explicit dependency on absolute pose. We validate our approach through simulations conducted in Isaac Sim and on hardware using Diablo biped-wheeled robots with zero-shot transfer, demonstrating effective payload path tracking across varying parameters.**

I. Introduction

In recent years, robotic systems have been deployed in increasingly diverse operational environments, ranging from structured warehouse settings to unstructured off-road terrains. These applications include transporting goods across warehouse floors, delivering materials on varying outdoor surfaces, and coordinating robots in challenging, obstacle-laden environments. The versatility required to navigate these different environments places new demands on robotic platforms, particularly for payload transport tasks where stability and adaptability are critical.

Although widely used for their agility, traditional monolithic wheeled mobile robots often struggle when handling payloads of varying sizes, especially on uneven surfaces or mixed environments. This leads to the need for modular robotic platforms capable of adapting to different operational contexts, reducing the dependency on specialized systems, and enabling flexible deployment. Rather than designing separate robots for each scenario, multi-robot coordination offers an efficient alternative, allowing for the dynamic sharing of tasks across multiple robots. In this milieu, biped-wheeled robots have emerged as a promising solution to these challenges due to their ability to operate both in on-road and off-road settings. The modular nature of biped-wheeled robots enhances their adaptability, providing stability for payload transport across diverse scenarios. Each biped-wheeled robot can provide compact, cost-effective compliance coupled with the ability to control 3D pose. The resulting modular composite system can be configured to enhance transport accuracy and efficiency for varying payload sizes and environmental conditions. This stands in contrast with previous works, which largely focused on rigid-axle wheeled robots with added compliance offered by active/passive serial- or parallel-chain structures [1], [2] to aid the coordinated payload transport.

To this end, this paper explores the implementation of a Deep Reinforcement Learning (DRL) framework designed to control the coordinated motion of multiple modular biped-wheeled robots, ensuring stable payload transport in on-road, off-road and warehouse scenarios. Our key contributions are as follows: (1) A novel approach to multi-robot coordination using biped-wheeled robots, allowing for modular deployment across diverse environments; (2) a single DRL agent outputs the policy regulating the ego robot's body twist and the follower robot's relative twist, ensuring adaptability in both structured and unstructured terrains; and (3) a framework that is agnostic to initial conditions and robot configurations.

II. Related Work

Cooperative payload transportation has been studied across various mobile robotic platforms, including differential drive robots, skid-steered systems, and articulated robots such as quadrupeds and rovers [3], [4]. These platforms typically employ fixed attachments for payload manipulation, combining pose control and obstacle avoidance strategies.

A comprehensive review of state-of-the-art cooperative transportation systems is demonstrated in [5], which analyzes approaches ranging from centralized and decentralized path planning to reinforcement learning-based techniques. For example, [1] employs an optimization-based path planning approach for a team of three skid-steered robots, each equipped with a UR5e robotic arm, to manage transportation and obstacle avoidance. Similarly, [2] utilizes an omnidirectional mobile base of the mecanum wheel paired with a 3-DoF parallel manipulator, featuring an independent control scheme for the base and manipulator. While these platforms achieve effective payload pose control, the added degrees of freedom introduce kinematic and dynamic redundancies, complicating control and increasing system cost, especially as payload sizes scale up.

To mitigate the challenges mentioned above, recent advances in DRL offer promising solutions. DRL agents can mitigate the difficulties of redundancy resolution and control in cooperative transport tasks. For example, [6] employs a multi-agent DRL (MADRL) approach to facilitate communication between robots for coordinated task execution. However, real-time deployment of MADRL remains limited due to the exponential increase in the state and action space dimensions and the sensitivity of such systems to partial observability. Another DRL-based approach, described in [7], uses a mobile robot and robotic arm in the simulation, though this method can restrict the workspace for payload manipulation.

While most existing approaches rely on fixed designs, the emergence of articulated mobile robots, such as biped-wheeled robots, offers a novel alternative [8]. These robots leverage internal degrees of freedom to directly control the payload pose, avoiding the need for highly redundant systems. However, research on biped-wheeled robots has focused mainly on trajectory tracking with balance control [9], given the underactuated nature and non-linear dynamics that make motion planning and control particularly challenging [10]. However, with the current capabilities of biped-wheeled robots, efficient payload pose control remains feasible. To the best of our knowledge, coordinated payload transport using two biped-wheeled robots is not investigated in the existing literature.

III. Robot Kinematics

In this section, we discuss the biped wheeled robot Diablo and the twist based kinematic formulation that is used in both simulation and realtity. Diablo is a six degree of freedom robot consisiting of two hip motors, two waist motors and two motors for the wheels [11]. The robot is modeled in the Isaac Sim simulator as a body attached to a serial chain linkage as shown in Fig. 1.

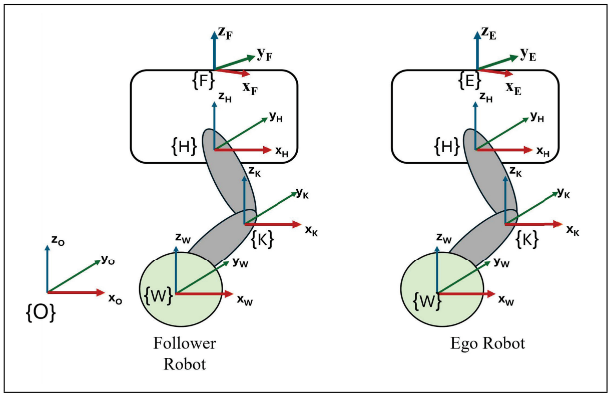

Fig. 1: Diablo Kinematics

To operate the biped robot at a pre-defined height in simulation, the length between the hip joint and wheel joint can be adjusted using homogeneous transforms as shown in the equation below:

$$^{H}T_{W} = [^{H}T_{K}][^{K}T_{W}] \tag{1}$$

where $\{K\}$ and $\{W\}$ represent knee and wheel joint frame of reference, respectively.

For motion control along a specified path, the configuration of each biped-wheeled robot is given by $(^{O}R_{E}, ^{O}\vec{p}_{E})$ and $(^{O}R_{F}, ^{O}\vec{p}_{F}) \in SE(2)$ [12]. Where, the ego robot's frame $\{E\}$ and the follower robot's frame $\{F\}$ are defined with respect to the inertial frame $\{O\}$. With the configuration defined in the planar space, the rotation and translation of the follower robot is obtained using the following equation,

$$^{E}T_{F} = [^{O}T_{E}]^{-1}[^{O}T_{F}] \tag{2}$$

Where $^{O}T_{E}$ and $^{O}T_{F}$ are the transformations from inertial frame to the ego and follower robots, respectively and $^{E}T_{F}$ is the transformation matrix from the ego robot to the follower robot. Looking from a top-down perspective of Fig. 1, in the planar space, each robot behaves like a differential drive robot and the twist $\xi \in \mathbb{R}^{6\times1}$ for the follower robot with respect to the ego robot can be written as:

$$\mathcal{V}_e = \begin{bmatrix} 0 \\ 0 \\ \omega_e \\ v_{x_e} \\ v_{y_e} \\ 0 \end{bmatrix} \tag{3}$$

where v_{x_e}, v_{y_e} and ω_e are the longitudinal, lateral and yaw velocities of the follower robot with respect to the ego robot that will be used to define a part of the DRL action space discussed later. Since the commands sent to the follower robot are ultimately the body twist commands in planar space, they can be obtained using the adjoint transformation given by,

$$\mathcal{V}_e = \begin{pmatrix} R_{ef} & o \\ [p_{ef}]R_{ef} & R_{ef} \end{pmatrix} \mathcal{V}_f \tag{4}$$

Here, R_{ef} represents the rotation matrix about the z-axis of the follower robot relative to the ego robot, and p_{ef} denotes the skew-symmetric matrix of the translation vector between the follower and ego robots. $\mathcal{V}_e$ is the twist of frame $\{F\}$ with respect to frame $\{E\}$ and $\mathcal{V}_f$ is the body twist of frame $\{F\}$.

IV. Problem Statement

This study introduces DRL framework to effectively coordinate two biped-wheeled robots (Diablo) to transport a shared payload along a specified path. The primary objective of the DRL agent is to minimize crosstrack and heading angle errors while ensuring the payload remains on the intended path.

Key challenges in this task include (1) Precise synchronization between the two robots to avoid misalignment that could destabilize the payload (2) Handling the dynamic interaction between the robots and the payload, particularly managing the effects of friction and its impact on payload movement (3) Ensure robust performance despite changes in the path or unexpected environmental disturbances.

Our proposed framework is implemented on a single workstation controlling two biped-wheeled robots. To test the robustness of the system, experiments are carried out using varying payload dimensions and weights on different paths. The following path types are used to challenge the system's adaptability: A variable curvature path with close waypoints and a constant curvature path with sparse waypoints to assess the system's performance when observation data are limited.

While the agent was trained in simulation to transport the payload over distances of up to 15 meters, the physical path tracking tests were constrained by hardware limitations. Space constraints restricted the available test area to 5m x 2m. Furthermore, in standing mode, the robot motors were prone to overheating, which limited the number of consecutive paths that could be evaluated

V. Methodology

This section outlines the process flow of our coordinated payload transport pipeline using biped-wheeled robots. The entire workflow, from simulation to real-world implementation, is illustrated in Fig. 2. Here, frame $\{B\}$ represents the payload frame of reference.

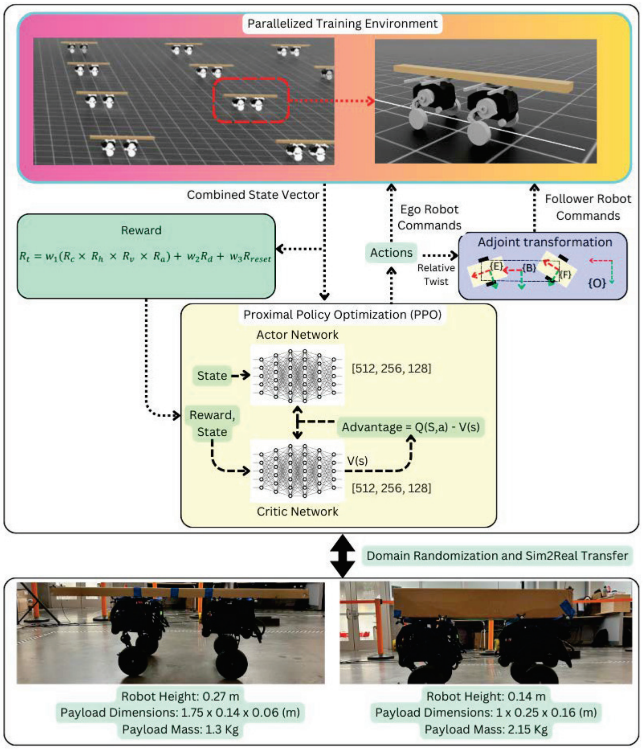
Parallelized Training Environment
Combined State Vector
Ego Robot Commands
Follower Robot Commands
Reward
$R_t = w_1(R_c \times R_h \times R_v \times R_a) + w_2 R_d + w_3 R_{reset}$
Actions
Relative Twist
Adjoint transformation
{E}
{B}
{F}
{O}
Proximal Policy Optimization (PPO)
Actor Network
State
[512, 256, 128]
Reward, State
Advantage = Q(S,a) - V(s)
V(s)
[512, 256, 128]
Critic Network
Domain Randomization and Sim2Real Transfer
Robot Height: 0.27 m
Payload Dimensions: 1.75 x 0.14 x 0.06 (m)
Payload Mass: 1.3 Kg
Robot Height: 0.14 m
Payload Dimensions: 1 x 0.25 x 0.16 (m)
Payload Mass: 2.15 Kg

Fig. 2: Combined Framework from Simulation to Reality

A. Parallelized Simulation Environment

Parallelized training has greatly improved the efficiency of DRL by leveraging deep neural networks as function approximators in sequential decision-making. Strategies like asynchronous methods allow multiple agents to explore the state space simultaneously, accelerating learning, especially in sparse reward environments where traditional approaches may face convergence challenges [13].

In our framework, the agent is trained entirely in simulation, utilizing the parallelization capabilities of NVIDIA's Isaac Lab [14]. As shown in Fig. 2, the parallelized training process enables efficient exploration and learning across multiple simulated environments simultaneously, accelerating the agent's ability to generalize and adapt to various scenarios.

B. Path Tracking

The agent determines the closest waypoint at each step based on the Euclidean distance between the payload and the path. This closest waypoint (denoted as i) is used to compute the crosstrack error, which is the lateral deviation of the payload from the path. The desired heading angle is computed with respect to the subsequent waypoint (denoted as i+1), ensuring a forward-looking perspective on the path. The heading angle error is then derived as the difference between the payload's current and desired heading calculated from the path geometry [15].

C. Deep Reinforcement Learning

Reinforcement Learning (RL) is a machine learning paradigm where an agent learns to maximize cumulative reward by interacting with an environment. While various DRL agents exist, they often rely on either policy optimization or action-value function approximation. Policy optimization provides stable trajectories, and action-value methods offer better sample efficiency. In this work, we implement Proximal Policy Optimization (PPO), an on-policy actor-critic algorithm known for its stable and high-performing policies. PPO uses a surrogate objective function to constrain policy updates, preventing destabilizing changes during training [16]. The actor network selects actions, while the critic network evaluates the state value to compute the advantage function $A(s_t, a_t)$, which guides policy updates. A clipping mechanism limits the magnitude of policy updates, ensuring stability and smooth policy evolution.

1) Observation and Action Space: DRL agent maps the inputs (observations) to actions. In order to avoid explicit dependency on absolute coordinates which can make the agent sensitive to initial conditions, we select the observations that are relative transforms between the robots and the object. This ensures that the payload and robots can be setup at any point in the path without a need for a precise starting point. Furthermore, the observation space provides sufficient information to the agent for achieving the coordinated payload transportation task. The observation space is given by,

$$\mathbf{O}_t = [\mathbf{d}_r \quad \mathbf{e}_c \quad \mathbf{e}_h \quad \mathbf{a}_t \quad \mathbf{r}_p] \tag{5}$$

Here, $\mathbf{d}_r$ denotes the distance between the robots, $\mathbf{e}_c$ represents the crosstrack error, $\mathbf{e}_h$ indicates the heading angle error, and $\mathbf{r}_p$ is an array of relative poses, including the payload's position relative to the ego robot and the follower robot's position relative to the payload and the overall observation space consists of 19 variables. To achieve precise coordination between the two robots using a single PPO agent, the action space is defined as follows:

$$\mathbf{a}_t = [\mathbf{v}_e \quad \omega_e \quad \mathbf{v}_{fx} \quad \mathbf{v}_{fy} \quad \omega_f] \tag{6}$$

where $\mathbf{v}_e$ and ω_e represent the linear and yaw velocity of the ego robot, respectively, and $\mathbf{v}_{fx}$, $\mathbf{v}_{fy}$, and ω_f denote the longitudinal, lateral, and yaw velocities of the follower robot relative to the ego robot. As discussed in Section III, the DRL agent outputs the relative twist and the adjoint transformation is computed to determine the body twist of the follower robot. This formulation enables pairwise interaction between the robots, where instead of an end-to-end approach—in which each robot operates independently without having knowledge of the other robot.

2) Reward Shaping: The core component of the DRL framework that affects the training quality and time is the reward formulation [17]. Dense reward functions have proven to better guide the DRL agent converge to an optimal policy as compared to sparse rewards since they provide continuous feedback at every step leading the agent to understand what better actions to take thereby accelerating the training process. For our framework, the reward function shown in Eq. 7 is formulated such that the crosstrack error is prioritized and it does not exceed the threshold of 0.5 m, while minimizing the heading angle error as well. The threshold was chosen considering the hardware limitations and the safety requirements within the available test area. For the reward function shown below, an exponential function was selected for key components—crosstrack error, heading angle error, and action deviations—to enhance sensitivity to deviations from the desired task by imposing significant penalties on undesired actions

$$\begin{aligned} R_t = W \cdot exp^{-w_1 e_c} \cdot exp^{-w_2 e_h} \cdot v \cdot exp^{-w_4 a} + \\ W_2 \cdot R_{\text{dist}} + W_3 \cdot R_{\text{reset}} \end{aligned} \tag{7}$$

where e_c is the crosstrack error, e_h is the heading error, v represents the velocity term, a accounts for preventing large difference between consecutive actions. R_{dist} is the reward associated with distance between robots, and R_{reset} is the reward related to reset conditions. R_{dist} and R_{reset} are modeled as sparse rewards shown below,.

$$R_{reset}, R_{dist} = \begin{cases} -1 & \text{if value} >= \text{threshold} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

The weights $[W, w_1, w_2, w_4, W_2, W_3]$ are carefully selected to prioritize minimizing crosstrack and heading angle errors, ensuring precise path tracking. Additionally, the weight associated with the distance component is designed to prevent robot collisions, promoting safer and more efficient payload transportation.

The velocity reward encourages the agent to move forward, but it is balanced by penalizing excessive crosstrack and heading angle errors, as these factors are multiplied together to maintain alignment. Additionally, during training, it was observed that significant variations in actions often destabilize the payload, slowing down the learning process. To mitigate this, a penalty is applied to discourage significant deviations between consecutive actions, promoting smoother control.

3) Hyperparameter Tuning: PPO's hyperparameters are critical to achieve higher rewards and an optimal stable policy. In our implementation, the policy clipping ratio (ϵ) is 0.2, while the discount factor (γ) is 0.99, and the Generalized Advantage Estimate (GAE) parameter (λ) is 0.95. Additionally, the entropy coefficient is set to 0.001 and the learning rate is adaptive with an initialization of $3 \cdot exp^{-4}$

D. Combined Workflow

In Fig. 2 we present the complete framework that addresses the challenge of coordinated payload transport using biped-wheeled robots. Due to the time-consuming and costly nature of hardware deployment, we leverage NVIDIA's Isaac Lab framework to implement and validate the system in a simulated environment. This allows for efficient testing and development before transitioning to physical platforms.

Training is conducted in a parallelized simulation environment with 4096 instances. The agent is trained on the variable curvature path, designed to develop the agent's ability to transport the payload laterally in both directions. The simulation operates at 50 Hz, closely matching real-time conditions, with both observations and actions synchronized to this frequency. The simulator provides observations that are fed into the DRL agent, and these observations are also used to define the reward function. The reward function guides the learning process of the critic network within the PPO agent. In our setup, the PPO agent outputs two sets of actions: (1) the body twist of the ego robot in the ego frame $\{E\}$, and (2) the relative twist of the follower robot with respect to the ego frame in frame $\{E\}$. Using the adjoint transformation, the relative twist of the follower robot in frame $\{E\}$ is converted to the body twist in frame $\{F\}$, which is then applied to control the follower robot's motion.

However, the agent trained in simulation under ideal conditions generally cannot be directly transferred on the actual robot because of a large simulation to reality gap. The critical elements that affect the transferability are difference in robot's system parameters, overfitting to inital conditions, noise in observations and actions due to the network characteristics and the friction parameters. In order to facilitate a zero-shot Sim2Real transfer, domain randomization can significantly improve the training quality [18]. Our approach not only uses domain randomization techniques, but also implements random reset conditions for the ego robot, follower robot, and the payload. The following are the key parameters that have been implemented for the domain randomization approach: We vary the static and dynamic friction parameters of the payload in the range of 0.4 to 0.7, along with variation to its mass from 1 kg to 2 kg, respectively. Furthermore, the initialization conditions of the ego and follower robot along with the payload are randomized between -0.5 m and 0.5 m in the x and y direction. The observations and actions are subjected to noise at each step with μ = 0.0 and σ = 0.05.

The above menioned framework seamlessly transfers the trained agent from simulation to reality. The hardware setup and results demonstrating our framework are highlighted in the next subsection. Furthermore, to mimic an unstructured environment and evaluate the agent's robustness against disturbances, speed bumps are introduced where the robots' internal controller manages the roll angle while the DRL agent compensates for payload disturbances. This compensation is critical since the robots' internal controllers do not directly stabilize the payload.

E. Experimental Setup

To deploy the coordinated payload transport setup on hardware and validate its performance, an indoor localization system is required to provide ground truth data for the robots and the box. This is achieved using an OptiTrack motion capture system, which consists of 12 infrared cameras capable of measuring object poses within a bounded environment with an accuracy of 0.2 mm as shown in Fig. 3 [19]. Both Diablos and the payload are modeled as rigid bodies using four markers each, and the pose is estimated based on the geometric centroid of each object. Using this pose data, relative measurements defined in Eq. 5 are computed. The estimated poses, processed by a machine running the Motive software, are transmitted to a second machine on a shared network, where our RL agent undergoes a forward pass with the computed states and publishes action commands to the robots via a ROS network. This completes the feedback loop of estimation, computation, and execution which is implemented at 50Hz.

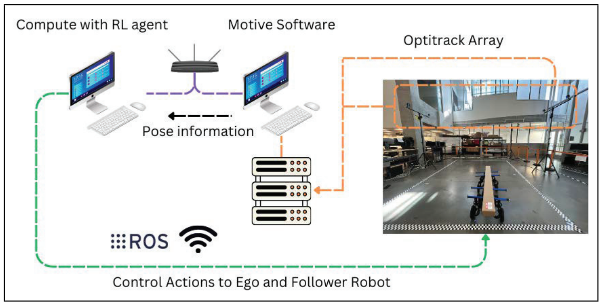

Fig. 3: Experimental Setup for Hardware Validation

VI. Results and Discussion

In this section, we provide a detailed analysis of the zero-shot Sim2Real transferability of our proposed approach, along with its robustness in handling unexpected variations in operating conditions. We illustrate these aspects using three distinct test scenarios. In the first scenario (S1), the payload used in the simulation matches that of the hardware. The second scenario (S2) introduces changes in the payload's dimensions and weight, providing an additional layer of complexity. Payload details are given in Fig. 2. Lastly, the third scenario evaluates the system's resilience to external disturbances, such as ground-based perturbations like speed bumps. Scenarios S1 and S2 involve paths with varying parameters that were not part of the training set, making these critical for assessing the generalizability of the agent's performance. The third scenario evaluates the agent's ability to secure and transport the payload without being dropped during straight-line motion, even under external disturbances. The agent's task is transporting the payload while minimizing crosstrack error (CTE) and heading angle error (HE) relative to the user-defined path. It is important to note that the reward function prioritizes minimizing CTE more than HE. Fig. 4 and Fig. 5 show the box plots for these error metrics across simulation and hardware, while Fig. 6 and Fig. 7 show that the agent successfully coordinates the two robots to track the payload along a reference path within the desired error threshold as mentioned in Section V-C.2.

Both scenarios S1 and S2 as illustrated in Fig. 4 and Fig. 5 indicate that the crosstrack errors consistently remain within the 0.5 m threshold. The data distribution shows comparable performance between simulation and real-world tests, highlighting the agent's zero-shot transfer capability for crosstrack error. However, for the heading angle error, the discrepancy is indicative of challenges in modeling the varying friction between robots and the payload. As such, the payload moves synchronously with the robots in simulation, whereas in reality, due to variable friction and payload dynamics, relative motion between the robots and the payload occurs, causing a larger error as compared to the simulation results. Despite these challenges, the agent effectively minimizes both crosstrack and heading angle errors in all test scenarios, showcasing the robustness and adaptability of our approach across simulation and hardware.

The path tracking performance is further assessed using the RMSE data presented in Tables I and II. In these tables, max height corresponds to the robot's leg length of 0.27 meters from the wheel to the hip joint, while mid height corresponds to a leg length of 0.14 meters. The RMSE data shows that the heading angle errors are the lowest for the constant curvature path. This is because the constant curvature path features gradual turns, which are easier to navigate than the more abrupt changes in the variable curvature path. On the other hand, crosstrack errors are higher for the constant curvature path. This is expected due to the constant curvature path's sparse waypoints, spaced 0.1 meters apart, compared to the variable curvature path's denser waypoints, spaced 0.005 meters apart. The sparse waypoints lead to less frequent updates, causing the robot to deviate from the path and primarily minimize heading angle errors relative to the nearest waypoint. Additionally, for runs involving bumps as shown in the supplementary video which are unseen disturbances, the crosstrack errors increase. Despite this, the agent effectively minimizes heading angle errors and maintains the payload's stability. Notably, the success rate for preventing the payload from dropping across all hardware runs is 100%.

VII. Conclusion and Future Work

In this study, we developed a DRL framework and integrated relative twist-based kinematics to implement an ego

TABLE I: CTE and HE RMSE values - Flat Surface

	S1				S2			
	Sim		Hardware		Sim		Hardware	
	CTE	HE	CTE	HE	CTE	HE	CTE	HE
Max height variable curvature	0.0402	0.101	0.144	0.1113	0.2413	0.0571	0.1482	0.2207
Max height constant curvature	0.2568	0.1018	0.2842	0.063	0.2531	0.0979	0.3102	0.1781
Mid height variable curvature	0.2098	0.0771	0.1013	0.1501	0.1197	0.0601	0.1948	0.2367
Mid height constant curvature	0.2795	0.0841	0.316	0.0918	0.2323	0.0698	0.3113	0.1293

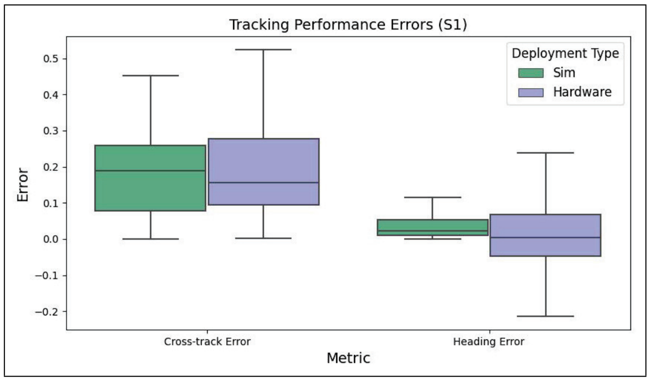

Fig. 4: Box Plot for Testing with Payload 1

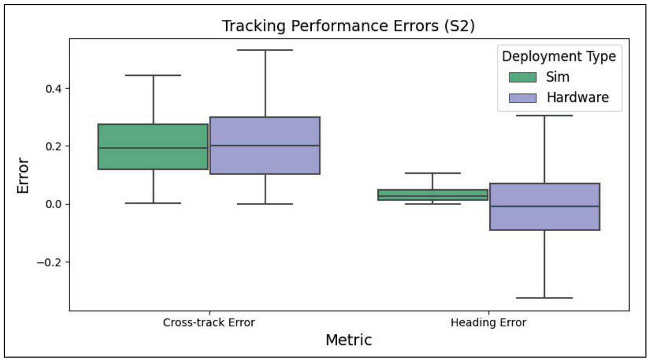

Fig. 5: Box Plot for Testing with Payload 2

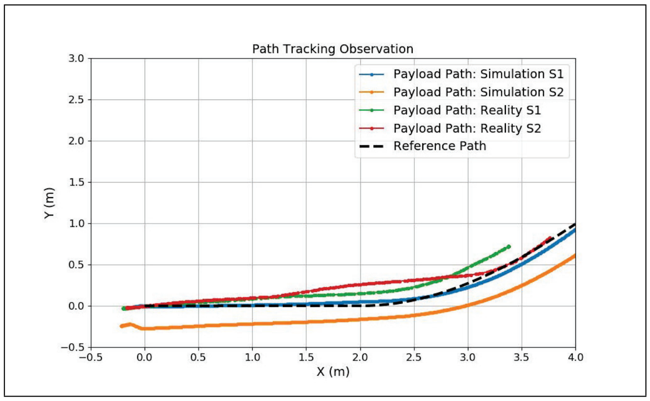

Fig. 6: Payload Tracking for Variable Curvature Path

follower control strategy for biped-wheeled robots. In this framework, the ego robot receives body twist actions directly, while the follower robot's relative twist is determined with respect to the leader to achieve coordinated path tracking of a payload across various previously unseen paths. The trained DRL agent demonstrated effective zero-shot transfer capabilities, successfully minimizing crosstrack and heading angle errors within the desired threshold. Future research will focus on extending the framework to handle longer paths on rough terrain as well with lower error thresholds, incorporating real-time height control for payload stabilization and adding minimalist design to the platform for preventing unwanted relative motion between the payload and the robots.

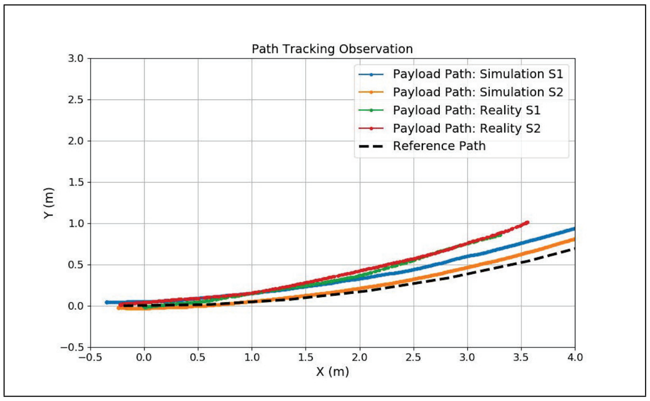

Fig. 7: Payload Tracking for Constant Curvature Path

TABLE II: CTE and HE RMSE values - Bumpy Surface

	Single bump		Dual bumps	
	CTE	HE	CTE	HE
Straight Line - Single Bump	0.3463	0.1823	0.3997	0.1427
Straight Line - Dual Bumps	0.3137	0.1169	0.5205	0.2802

ACKNOWLEDGMENT

This work was supported in part by the US National Science Foundation under NSF IIS-1925500 and NSF CNS-1939058

References

[1] F. Kennel-Maushart and S. Coros, "Payload-aware trajectory optimisation for non-holonomic mobile multi-robot manipulation with tip-over avoidance," *IEEE Robotics and Automation Letters*, vol. 9, no. 9, pp. 7669–7676, 2024.

[2] M. Elwin, B. Strong, R. Freeman, and K. Lynch, "Human-multirobot collaborative mobile manipulation: The omnid mocobots," *IEEE Robotics and Automation Letters*, vol. 8, no. 1, pp. 376–383, Jan. 2023, publisher Copyright: © 2016 IEEE.

[3] A. Trebi-Ollennu, H. Das Nayar, H. Aghazarian, A. Ganino, P. Pirjanian, B. Kennedy, T. Huntsberger, and P. Schenker, "Mars rover pair cooperatively transporting a long payload," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 3, 2002, pp. 3136–3141 vol.3.

[4] J. Kim, R. T. Fawcett, V. R. Kamidi, A. D. Ames, and K. A. Hamed, "Layered control for cooperative locomotion of two quadrupedal robots: Centralized and distributed approaches," *IEEE Transactions on Robotics*, vol. 39, no. 6, pp. 4728–4748, 2023.

[5] E. Tuci, M. H. M. Alkilabi, and O. Akanyeti, "Cooperative object transport in multi-robot systems: A review of the state-of-the-art," *Frontiers in Robotics and AI*, vol. 5, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:44118313

[6] K. Shibata, T. Jimbo, and T. Matsubara, "Deep reinforcement learning of event-triggered communication and control for multi-agent cooperative transport," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 8671–8677.

[7] Y. Wang and C. de Silva, "Cooperative transportation by multiple robots with machine learning," in *2006 IEEE International Conference on Evolutionary Computation*, 2006, pp. 3050–3056.

[8] D. Mehta, K. C. Kosaraju, and V. N. Krovi, "Actively articulated wheeled architectures for autonomous ground vehicles - opportunities and challenges," *SAE Technical Paper Series*, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:258047076

[9] V. Klemm, A. Morra, L. Gulich, D. Mannhart, D. Rohr, M. Kamel, Y. de Viragh, and R. Siegwart, "Lqr-assisted whole-body control of a wheeled bipedal robot with kinematic loops," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3745–3752, 2020.

[10] Z. Cui, Y. Xin, S. Liu, X. Rong, and Y. Li, "Modeling and control of a wheeled biped robot," *Micromachines*, vol. 13, no. 5, 2022. [Online]. Available: https://www.mdpi.com/2072-666X/13/5/747

[11] T. Guo, J. Liu, H. Liang, Y. Zhang, W. Chen, X. Xia, M. Wang, and Z. Wang, "Design and dynamic analysis of jumping wheel-legged robot in complex terrain environment," *Frontiers in Neurorobotics*, vol. 16, p. 1066714, 2022.

[12] M. Abou-Samah, C. P. Tang, R. Bhatt, and V. N. Krovi, "A kinematically compatible framework for cooperative payload transport by nonholonomic mobile manipulators," *Autonomous Robots*, vol. 21, pp. 227–242, 2006. [Online]. Available: https://api.semanticscholar.org/CorpusID:7603997

[13] Q. Yin, T. Yu, S. Shen, J. Yang, M. Zhao, W. Ni, K. Huang, B. Liang, and L. Wang, "Distributed deep reinforcement learning: A survey and a multi-player multi-agent learning toolbox," *Machine Intelligence Research*, vol. 21, no. 3, pp. 411–430, 2024.

[14] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar, A. Mandlekar, B. Babich, G. State, M. Hutter, and A. Garg, "Orbit: A unified simulation framework for interactive robot learning environments," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3740–3747, 2023.

[15] D. Mehta, A. Salvi, and V. Krovi, "Rough terrain path tracking of an ackermann steered platform using hybrid deep reinforcement learning," in *2024 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, 2024, pp. 685–690.

[16] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[17] A. Raman, A. Salvi, M. Schmid, and V. Krovi, "Reinforcement learning control of a reconfigurable planar cable driven parallel manipulator," in *2023 IEEE International Conference on Robotics and Automation (ICRA2023)*, 2023.

[18] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 23–30, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:2413610

[19] A. Joglekar, S. Sutavani, C. Samak, T. Samak, K. C. Kosaraju, J. Smereka, D. Gorsich, U. Vaidya, and V. Krovi, "Data-driven modeling and experimental validation of autonomous vehicles using koopman operator: Distribution a: Approved for public release; distribution unlimited. opsec# 7248," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 9442–9447.