

Design and Evaluation of a Mobile Manipulation System with Learning-Based Slip Mitigation

Qabas Alkaissi, Alanoud Alsalem, Leen Almousa and Qais Jildeh
*Department of Computer Engineering, King Abdullah II School of Engineering
Princess Sumaya University for Technology (PSUT), Amman, Jordan*
{qab20210786, ala20210371, lee20210268, qai20210155}@std.psut.edu.jo

Abstract—This work presents the design, modeling, and evaluation of a mobile manipulation system that integrates navigation, manipulation, learning-based slip estimation, and vision-based localization within a unified control framework. A key challenge addressed is wheel slip, which violates the no-slip assumptions underlying classical mobile robot kinematics and leads to degraded odometry. To address this, a supervised learning-based slip estimation model is trained using simulation-generated data. This estimator is embedded within a real-time finite-state traction controller that dynamically adapts wheel commands. In simulation, the controller reduces slip duration compared to unmitigated control; navigation, manipulation, and vision-based localization are demonstrated on a physical robot.

Keywords— Mobile Robotics, Slip Estimation, Proprioceptive Sensing, Finite-State Control, Simulation-Based Learning

I. INTRODUCTION

Mobile manipulators operating in unstructured environments frequently experience wheel slip, violating classical no-slip kinematic assumptions and causing severe odometry degradation and control instability [1], [2]. Although external sensing modalities such as vision or inertial measurements can partially compensate for these effects, they introduce dependencies on sensor availability, environmental conditions, and additional system complexity. More recently, learning-based methods have demonstrated the feasibility of inferring slip from proprioceptive signals alone; however, integrating such noisy and uncertain estimates into real-time control loops without inducing oscillations or unstable behavior remains an open challenge.

This paper proposes a unified traction-aware control framework that combines simulation-based learning with a finite-state supervisory controller. A supervised learning model is trained to estimate wheel slip using only proprioceptive measurements and is embedded within a state-based traction controller that explicitly filters transient disturbances and enforces stable recovery behavior. The learning component provides continuous slip awareness, while the finite-state logic ensures control-oriented decision making suitable for real-time deployment. Simulation experiments show reduced slip duration and improved recovery compared to unmitigated control, and hardware experiments demonstrate navigation, manipulation, and vision-based localization on a physical robot.

- A control-oriented formulation of wheel slip that enables real-time estimation using only proprioceptive sensing.
- An automated simulation-based data generation pipeline for supervised slip learning under extreme traction conditions.
- A learning-based slip estimator integrated within a finite-state traction control framework for robust real-time deployment.
- A comprehensive system-level evaluation demonstrating improved stability and recovery behavior compared to non-mitigated controllers.

II. RELATED WORK

Wheel slip has long been recognized as a major source of error in mobile robot navigation, particularly in unstructured or low-traction environments. Early work by Iagnemma and Dubowsky [3] investigated the effects of wheel-terrain interaction on rough terrain and demonstrated how slip degrades odometry and control performance, motivating the need for explicit slip-aware control strategies. Similarly, Ojeda and Borenstein [2] analyzed odometry errors in overconstrained mobile robots and proposed techniques to reduce error accumulation caused by wheel slippage under non-ideal contact conditions. These studies highlight the limitations of purely kinematic models when traction assumptions are violated.

Several approaches have been proposed to mitigate slip through control and estimation. Balakrishna and Ghosal [4] modeled wheel-ground traction as a function of slip and showed that controllers ignoring slip track poorly. Ward and Iagnemma [1] proposed a dynamic-model-based estimator that detects wheel slip and immobilization on outdoor terrain. While these approaches provide valuable insight into traction dynamics, they often require detailed terrain models or assumptions that are difficult to satisfy in practice.

More recently, learning-based methods have been explored as a means to estimate slip using onboard sensing. Gonzalez et al. [5] showed that regression models trained on proprioceptive features can predict wheel slip. Such approaches are attractive due to their low sensing requirements and adaptability to complex, nonlinear dynamics. However, most learning-based slip estimation studies focus primarily on prediction accuracy and do not explicitly address how learned

models should be integrated into real-time control architectures to ensure stable behavior under noisy or transient conditions.

Hybrid control architectures combining continuous dynamics with discrete supervisory logic have been widely studied in robotics. Brooks [6] introduced layered control systems that emphasize robustness through behavioral decomposition, while Alur et al. [7] formalized hybrid automata as a framework for modeling and verifying systems that exhibit both continuous and discrete dynamics. Finite-state machines (FSMs) remain a practical and widely adopted tool for enforcing safety constraints and managing mode transitions in real-world robotic systems, particularly when operating under uncertainty.

Global vision-based localization has also been extensively studied as a complement to onboard sensing. Fiducial marker systems such as AprilTag, introduced by Olson [8], provide robust and computationally efficient pose estimation in controlled environments. Foundational work in multiple-view geometry by Hartley and Zisserman [9] establishes the mathematical basis for camera calibration and pose recovery, which underpins many modern vision-based localization pipelines. While vision systems can provide accurate absolute pose estimates, they are inherently sensitive to occlusion, lighting conditions, and field-of-view limitations.

In contrast to prior work, the approach presented in this paper focuses on integrating learning-based slip estimation with a structured finite-state control framework within a complete mobile manipulation system. Rather than relying on external sensing for slip detection, the proposed method uses simulation-trained models operating on proprioceptive data and embeds them into a supervisory controller that explicitly handles transient disturbances and recovery behavior. This combination of lightweight learning and FSM-based control enables robust traction management while maintaining compatibility with vision-assisted localization and manipulation tasks.

III. SYSTEM OVERVIEW

This section presents an overview of the mobile manipulation system, including the robot model, kinematic formulation, simulation environment, and sensing modalities used throughout this work. The system is designed to support integrated navigation, manipulation, slip estimation, and traction-aware control within a unified framework.

A. Robot Platform and Simulation Environment

The mobile robot consists of a wheeled mobile base equipped with a planar robotic arm for object handling tasks. The slip-estimation and traction-control components are modeled and evaluated in ROS–Gazebo and the physical platform uses mecanum wheels. The robot description is defined using the Unified Robot Description Format (URDF), specifying link geometries, inertial properties, joint constraints, and collision models.

The mobile base employs a four-wheel configuration that enables planar motion, while the manipulator arm is mounted

rigidly on the base and operates in a two-degree-of-freedom (2-DOF) planar workspace. Realistic inertial parameters are assigned to all links to suppress non-physical behavior and ensure stable simulation dynamics. Joint-level control is implemented using standard ROS control interfaces, allowing independent actuation of the mobile base and manipulator joints.

B. Manipulator Kinematic Modeling

The manipulator arm is modeled as a planar 2-DOF serial chain. Forward kinematics are used to compute the Cartesian position of the end-effector relative to the robot base frame. Given link lengths L_1 and L_2 , and joint angles θ_1 and θ_2 , the end-effector position (x, y) is expressed as

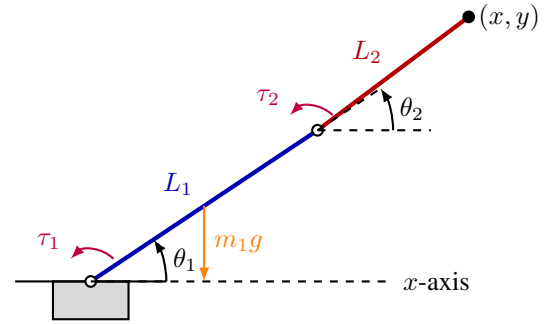


Fig. 1. 2-DOF Planar Arm Model showing kinematic parameters (L, θ) and dynamic parameters (m, τ).

$$\begin{aligned} x &= L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) \\ y &= L_1 \sin(\theta_1) + L_2 \sin(\theta_1 + \theta_2) \end{aligned}$$

Inverse kinematics are solved using a geometric approach based on the law of cosines, enabling the computation of joint configurations that reach a desired planar target. These kinematic models are used to generate reference joint trajectories for manipulation tasks and to ensure consistent interaction between the mobile base and the arm during operation.

C. Simulation-Based Dynamics and Control Interface

Robot dynamics, including gravity, friction, and contact interactions, are simulated using the Gazebo physics engine. Joint trajectory controllers regulate arm motion in position mode, while velocity commands issued to the mobile base are mapped to individual wheel velocities through the drive model. This separation enables coordinated yet independent control of locomotion and manipulation.

The simulation environment serves two primary roles: validating the stability of the integrated system under nominal conditions and enabling controlled manipulation of physical parameters such as friction and actuation limits. These capabilities are used to study slip behavior and generate training data under extreme traction conditions, as described in the following sections.

D. Sensing and State Information

The system utilizes a combination of proprioceptive and exteroceptive sensing. Proprioceptive measurements, including wheel joint velocities and commanded motion signals, are used for slip estimation and traction control. These signals remain available even under degraded perception and form the basis of the learning-based slip estimator introduced in Section IV.

Exteroceptive sensing is provided by a fixed overhead camera that supplies global pose estimates through fiducial marker detection. Vision-based localization is treated as an auxiliary sensing modality and is used to support navigation and evaluation, while core traction control relies exclusively on onboard proprioceptive information. This design choice ensures that slip mitigation remains functional even during vision occlusion or sensor dropout.

IV. LEARNING-BASED SLIP ESTIMATION

This section presents the formulation, data generation process, and learning model used to estimate wheel slip from proprioceptive measurements. The objective is to infer traction loss in real time without reliance on external ground-truth sensors, enabling robust deployment under degraded perception conditions.

A. Slip Formulation

Wheel slip is defined as the discrepancy between wheel-induced motion and the true ground motion of the robot. Under ideal rolling conditions, the linear velocity of the robot is directly proportional to the wheel angular velocity. When traction is compromised, this relationship breaks down, resulting in divergence between commanded motion and actual displacement.

The classical slip ratio formulation expresses slip as a normalized quantity. In this work, slip is instead modeled as an absolute velocity divergence, which is more directly actionable for control purposes. Specifically, slip is defined as

$$\lambda = v_{\text{wheel}} - v_{\text{ground}}$$

where v_{wheel} denotes the wheel-induced linear velocity inferred from wheel angular velocities, and v_{ground} represents the true ground velocity of the robot. This formulation produces a signed, continuous signal that maps naturally to velocity modulation in traction control.

During the online deployment, v_{ground} is unobservable. Consequently, slip estimation is treated as a regression problem in which the unobservable slip state λ is inferred from observable proprioceptive signals.

B. Proprioceptive Feature Space

The learning model operates exclusively on onboard proprioceptive measurements. Wheel angular velocities are extracted directly from joint state feedback and mapped to a linear velocity estimate according to

$$v_{\text{wheel}} = \frac{r}{4} \sum_{i=1}^4 \dot{\theta}_i$$

where r is the wheel radius and $\dot{\theta}_i$ denotes the angular velocity of the i -th wheel. The commanded linear velocity v_{cmd} is obtained from the motion controller. The resulting feature vector is defined as

$$X = [v_{\text{cmd}}, v_{\text{wheel}}]^T$$

which captures both the intended motion and the observed wheel response. This minimal feature space was selected to reduce model complexity while preserving sensitivity to traction loss.

C. Simulation-Based Data Generation

Training data is generated entirely in simulation using an automated data collection pipeline. Each trial consists of issuing a velocity command, recording wheel kinematics, and logging the corresponding ground-truth state from the simulator. To ensure statistically independent samples, the simulation state is fully reset between trials, eliminating residual momentum and integrator effects.

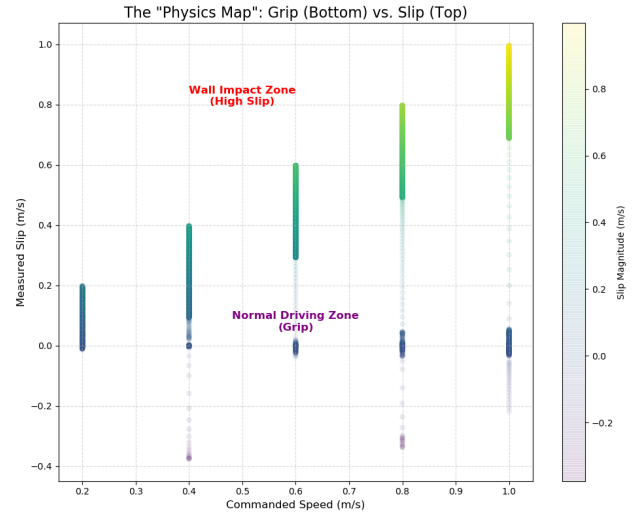


Fig. 2. Distribution of the training dataset ($N = 50,000$) spanning both high-slip and low-slip regimes.

To expose the learning model to the full spectrum of slip behavior, data are collected under two extreme operating regimes:

- 1) **High-slip conditions**, induced by commanding motion against immovable obstacles and by reducing friction coefficients to simulate ice-like terrain.
- 2) **Low-slip conditions**, corresponding to nominal motion on uniform terrain.

The final dataset consists of approximately 50,000 samples collected at a fixed simulation timestep. Ground-truth slip labels are computed offline using simulator-provided ground velocity and are never accessed during online inference.

D. Regression Model Selection

A Random Forest Regressor is used to estimate slip magnitude from the proprioceptive feature vector. Random Forests provide strong nonlinear modeling capability while

maintaining robustness to noise and limited sensitivity to hyperparameter tuning.

Linear regression was found to be insufficient due to the highly nonlinear and discontinuous nature of slip dynamics, particularly during transitions between traction and burnout regimes. Neural networks were considered but not adopted due to the low dimensionality of the input space and the increased risk of overfitting when trained exclusively on synthetic data.

The ensemble structure of the Random Forest provides implicit regularization and enables stable predictions under noisy measurements, making it well suited for real-time deployment within a control loop.

E. Model Evaluation

Model performance is evaluated using an 80/20 train–test split. The trained model achieves an R^2 score of approximately 0.95 on held-out data, indicating strong alignment between predicted and ground-truth slip values across diverse operating conditions.

Initial experiments revealed a bias toward false-positive slip detection on high-traction terrain due to overrepresentation of slip-heavy scenarios in the dataset. This bias was mitigated by augmenting the training data with additional nominal driving samples, resulting in improved baseline stability without sacrificing sensitivity to genuine slip events.

V. TRACTION CONTROL VIA FINITE-STATE MACHINE

This section describes the real-time traction control architecture used to regulate wheel commands based on the learned slip estimates. Rather than relying on continuous control laws alone, the proposed approach employs a finite-state machine (FSM) to enforce robust and interpretable behavior under slip-prone conditions.

A. Control Architecture Overview

The traction controller operates in a closed-loop configuration, receiving real-time slip estimates from the learning-based estimator described in Section IV. The estimated slip signal is used to supervise velocity commands issued to the mobile base, enabling dynamic adaptation to changing traction conditions.

A finite-state machine is adopted to manage mode transitions between nominal operation and slip recovery. This design choice provides explicit handling of transient disturbances, prevents overreaction to noisy estimates, and enforces safety constraints in a structured manner. FSM-based control is particularly suitable for systems that exhibit hybrid dynamics, combining continuous motion with discrete mode switches.

B. State Definitions

1) *Normal Traction* (S_0): In this state, the robot tracks the commanded velocity v_{cmd} without modification. Slip estimates remain below a predefined safety threshold, indicating sufficient traction.

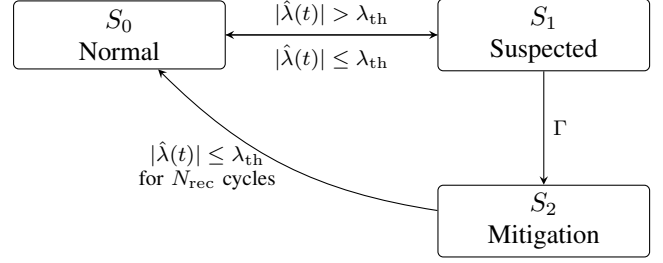


Fig. 3. Finite-state traction controller supervised by the estimated slip magnitude $\hat{\lambda}(t)$.

2) *Suspected Slip* (S_1): This transitional state acts as a temporal filter. When the estimated slip exceeds the threshold for a single control cycle, the controller enters S_1 but does not yet intervene. This prevents spurious state transitions caused by sensor noise or short-lived disturbances.

3) *Slip Mitigation* (S_2): If slip violations persist over multiple consecutive cycles, the controller enters the mitigation state. In this state, the output velocity is actively reduced to suppress wheel spin and recover traction.

C. Slip Thresholding and State Transitions

Let $\hat{\lambda}(t)$ denote the estimated slip at time t , expressed as a velocity divergence in meters per second. A safety threshold λ_{th} defines the maximum allowable slip magnitude under stable operation. The parameter Γ acts as a temporal hysteresis, ensuring that only sustained slip events trigger active intervention. The resulting finite-state machine is illustrated in Fig. 3, showing the state transitions governed by the estimated slip magnitude and temporal hysteresis.

D. Velocity Modulation Strategy

When the controller enters the slip mitigation state (S_2), the commanded output velocity is progressively reduced to suppress wheel spin and recover traction. Rather than applying an abrupt stop or hard saturation, the controller scales down the output velocity by a fixed mitigation factor α at each control cycle. This gradual reduction prevents impulsive command changes that could destabilize the system or induce oscillatory behavior.

Once the estimated slip magnitude falls below the safety threshold ($|\hat{\lambda}| \leq \lambda_{\text{th}}$), the controller transitions back to normal operation and restores the velocity smoothly toward the original commanded value. Velocity recovery is governed by a recovery gain β , which limits the rate of re-acceleration and prevents immediate re-entry into the slip regime.

E. Real-Time Execution and Stability Considerations

The FSM-based traction controller operates at a fixed control frequency and introduces negligible computational overhead. The learned slip estimator executes within the same loop, producing predictions with latency well below the control cycle duration.

The resulting closed-loop behavior exhibits a bounded limit cycle near the slip threshold, reflecting the controller

dynamically estimating and respecting the maximum available traction. Rather than enforcing overly conservative velocity limits, the FSM allows the system to operate near the traction boundary while preventing sustained wheel spin or loss of control.

By combining learning-based slip estimation with explicit state supervision, the proposed controller achieves robust traction management under noisy measurements, low-friction conditions, and transient disturbances.

VI. VISION-BASED LOCALIZATION AND NAVIGATION

The mobile manipulation system employs a global, vision-based localization pipeline to provide absolute pose estimates for navigation and system-level evaluation. A fixed overhead camera acts as a global observer, decoupling localization from onboard sensing and compensating for odometry degradation caused by wheel slip and contact disturbances.

Prior to operation, intrinsic camera calibration is performed using a planar checkerboard to estimate the camera matrix and distortion coefficients. Online undistortion is applied to each frame to ensure geometric consistency before further processing. Because the robot operates on a planar workspace, pixel-to-world mapping is modeled using a planar homography, calibrated through four reference points corresponding to known world-frame coordinates.

Robot pose estimation is achieved using a fiducial marker (AprilTag) rigidly mounted on the robot's top surface. The vision system detects the tag in the image frame and recovers its pose using geometric constraints derived from the known tag dimensions. The detected tag center is mapped to the world frame using the calibrated homography, yielding the robot's planar position (x, y) . Robot orientation (yaw) is computed from the relative geometry of adjacent tag corners in the world frame.

The vision system runs on a central laptop and transmits pose estimates to the onboard controller over a lightweight UDP interface. Pose messages include timestamps and validity flags, allowing the onboard controller to reject stale or invalid measurements. When pose information becomes unavailable due to occlusion or field-of-view limitations, the navigation controller transitions into safe fallback modes, ensuring stable operation despite perception outages.

Navigation is implemented using a goal-driven control loop. The operator selects target waypoints via a click-to-goal interface, which are converted from pixel coordinates to world-frame positions using the homography. The onboard controller computes heading and distance errors relative to the current pose and issues velocity commands to the mobile base. During manipulation actions, navigation is explicitly paused through safety interlocks to prevent destabilizing interactions between locomotion and arm motion.

VII. EXPERIMENTAL SETUP

Slip-estimator training data and the slip-mitigation comparison were generated in a ROS–Gazebo simulation using a simplified four-wheel skid-steer model of the base. The navigation, manipulation, and vision-based localization

experiments were run on the physical robot, a mecanum-wheeled mobile base with a 2-DOF planar arm, in a 3×3 m indoor workspace observed by a single fixed overhead camera.

The robot model was defined using URDF and included realistic joint constraints and collision geometries. The mobile base was actuated through a velocity-controlled drive interface, while the manipulator arm was controlled using joint trajectory controllers with tuned PID gains. Vision processing, calibration, operator interaction, and logging were executed on a central laptop, while low-level control, safety arbitration, and arm actuation were handled by an onboard controller.

Experimental data were logged at 30 Hz and included synchronized pose telemetry, system events, and annotated video streams. Camera calibration quality was assessed using the root-mean-square reprojection error computed over all checkerboard samples. Navigation experiments consisted of sequential multi-waypoint missions interleaved with manipulation tasks, allowing evaluation of both continuous motion and discrete state transitions.

Slip mitigation experiments were conducted under intentionally degraded traction conditions by reducing ground friction coefficients and increasing wheel actuation limits. Identical scenarios were executed with and without slip mitigation to enable direct comparison between controllers.

VIII. RESULTS AND ANALYSIS

A. Localization Accuracy

Initially, the camera calibration performance was assessed. The root-mean-square reprojection error (RMSE_px) quantifies the average discrepancy, in pixel units, between the observed checkerboard corner locations and their corresponding projections computed using the estimated camera intrinsic parameters and distortion model. As such, RMSE_px serves as a direct measure of how accurately the calibrated camera model represents the true imaging geometry.

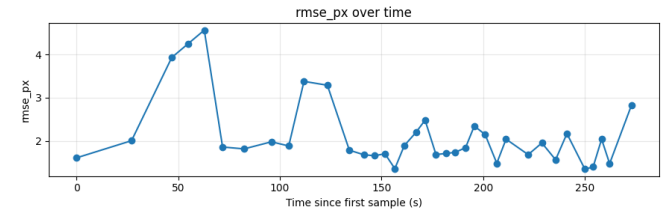


Fig. 4. RMSE Reprojection Error

Across the calibration dataset, the RMSE_px values remain predominantly within the range of approximately 1.5–2.5 pixels, with a median value of 1.87 pixels. This indicates that, for the majority of calibration images, the projected corner locations deviate from the detected corners by fewer than two pixels on average, which is generally considered acceptable for camera calibration in robotics and computer vision applications.

However, a small number of calibration images exhibit higher RMSE_px values, exceeding 4 pixels. These outliers

correspond to frames where corner detection accuracy is likely degraded, potentially due to motion blur, suboptimal lighting conditions, partial visibility of the checkerboard, or extreme viewing angles. The presence of these outliers inflates the upper tail of the error distribution but does not reflect the typical calibration performance.

1) *Information Logging*: The performance of the hybrid Global Vision and Mobile Manipulation system was evaluated in a controlled $3m \times 3m$ workspace. The experimental data was logged at a frequency of 30Hz using the central control laptop. The dataset consists of:

- **Pose Telemetry (pose.csv)**: Containing timestamped coordinates (x, y, θ) , yaw headings, and validity flags from the computer vision system.
- **System Events (events.csv)**: A log of high-level state transitions, including goal reception and manipulator activation.

B. Trajectory Tracking and Navigation

The robot successfully completed a multi-waypoint mission lasting 231.05 seconds, traversing a total distance of 22.57 meters. The recorded trajectory closely followed the commanded waypoints, with smooth arc transitions and no evidence of oscillatory or unstable behavior. The absence of sharp discontinuities at turning points confirms appropriate tuning of the motion controller relative to the robot's inertia and velocity limits.

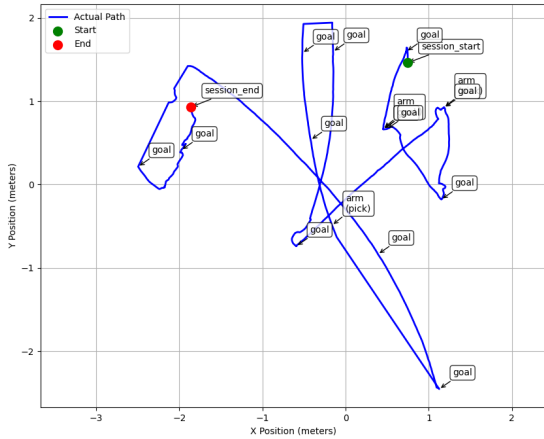


Fig. 5. The blue trace indicates the actual path recorded by the global vision system, while red markers denote the sequential target waypoints.

C. Dynamic Response and Convergence

The step-response characteristics of the controller are analyzed in Figure 6. The vertical spikes represent the instant a new goal was issued by the operator (Human-in-the-Loop clicks), causing an immediate jump in the position error.

The position tracking error is defined as

$$e(t) = \sqrt{(x(t) - x_g)^2 + (y(t) - y_g)^2},$$

- **Settling Time**: The system consistently demonstrated a smooth exponential decay in error, typically converging

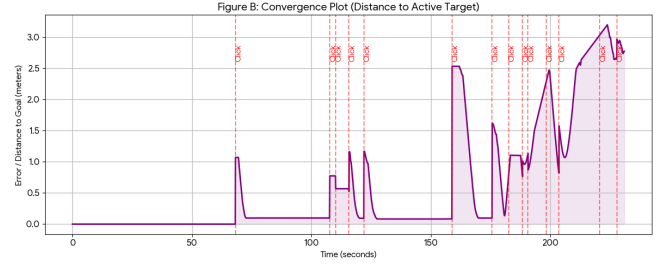


Fig. 6. System step response and convergence analysis. Vertical spikes indicate the reception of new goal commands, followed by exponential decay in position error.

to the target threshold within 10-15 seconds, depending on travel distance.

- **Velocity Profile**: To ensure stability against network latency (33ms) and image processing delays, the linear velocity was saturated at 0.10 m/s. This design choice prevented overshoot and ensured the Manipulator Arm remained stable during transit.

D. Operational Efficiency

Figure 7 details the temporal sequence of the mission, replacing standard duty-cycle metrics with a state-based timeline.

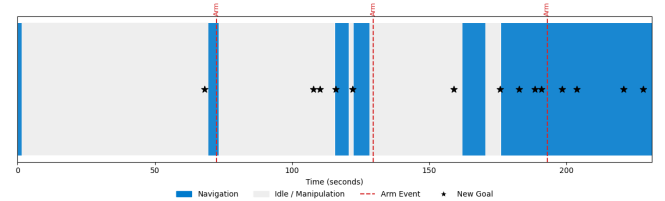


Fig. 7. Operational timeline showing state transitions. The correlation between 'Stationary' phases (Grey) and 'Arm Events' (Red lines) confirms that idle time was utilized for manipulation tasks.

- **State Switching**: The system exhibits a clear alternating behavior between Navigation Phases (Blue) and Manipulation Phases (Grey).
- **Idle Utilization**: The correlation between the 'Idle' periods and the 'Arm Event' markers confirms that the majority of stationary time was effectively utilized for robotic arm operations, rather than being caused by computational delays or path planning failures.

E. Vision System Robustness

A critical analysis of the sensor robustness is shown in Figure 8.

Vision availability is defined as

$$A = \frac{1}{K} \sum_{k=1}^K 1[\text{valid}_k = 1] \times 100\%.$$

A pose estimate is marked as valid if a fiducial marker is successfully detected and a consistent homography mapping can be computed; otherwise, the estimate is flagged invalid and handled by the FSM safety logic.

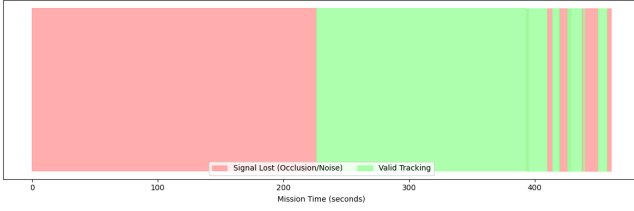


Fig. 8. Vision system availability map. Green regions indicate valid tracking, while red regions represent signal loss due to occlusion, handled by the FSM safety logic.

F. Slip Mitigation Performance

Figure 9 compares slip dynamics under identical conditions with and without traction control. In the absence of mitigation, the system exhibits slip events characterized by divergence between wheel-induced and ground velocities. In contrast, the proposed ML-based FSM controller actively regulates slip, rapidly suppressing large transients and maintaining operation within a bounded limit cycle near the traction threshold. This behavior reflects the controller dynamically estimating and respecting the maximum available friction rather than enforcing velocity limits.

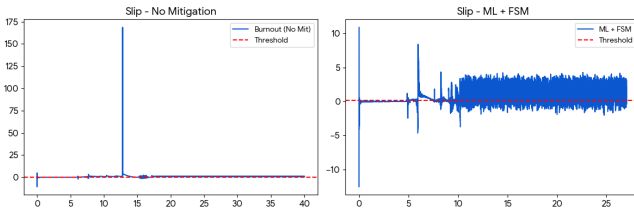


Fig. 9. Slip Dynamics in Gazebo ($\hat{\lambda}$). (a) The unmitigated controller ($\lambda_{th} = 0.2$). (b) The proposed ML+FSM controller.

IX. DISCUSSION

The experimental results demonstrate that combining learning-based slip estimation with structured finite-state control yields robust traction management without reliance on external sensing. The system operates safely even when global vision is unavailable for extended periods, highlighting the advantage of grounding traction control in proprioceptive measurements.

The vision system, while not continuously available, provides valuable absolute pose information for navigation and evaluation. Its intermittent nature underscores the importance of decoupling core stability mechanisms from perception-dependent components. The FSM architecture plays a critical role in filtering transient disturbances, enforcing safety constraints, and coordinating navigation with manipulation.

The slip controller does not enforce conservative velocity limits; instead, it adapts dynamically to the maximum available traction, resulting in efficient yet stable behavior. This limit-cycle behavior reflects a practical compromise between performance and safety, particularly in environments with uncertain or varying surface properties.

X. CONCLUSION

This work presented a complete mobile manipulation system integrating analytical modeling, simulation-based learning, structured control, and vision-based localization within a unified framework. A key contribution is the learning-based slip estimation module, trained in simulation and deployed using onboard proprioceptive sensing. When embedded within a finite-state traction controller, the learned model enables real-time slip mitigation without reliance on external ground-truth measurements.

Experimental evaluation demonstrated stable navigation, smooth convergence to waypoints, effective utilization of idle time for manipulation, and robust operation despite significant perception outages. Compared to unmitigated approaches, the proposed controller reduced slip duration and improved recovery behavior.

Overall, this project illustrates how lightweight learning models, when combined with principled control logic, can address fundamental challenges in mobile robotics. The architecture presented provides a scalable foundation for future extensions, including real-world hardware deployment, terrain-aware adaptation, and tighter integration between perception, learning, and control.

REFERENCES

- [1] C. C. Ward and K. Iagnemma, "A dynamic model-based wheel slip detector for mobile robots on outdoor terrain," *IEEE Trans. Robot.*, vol. 24, no. 4, pp. 821–831, 2008.
- [2] L. Ojeda and J. Borenstein, "Methods for the reduction of odometry errors in over-constrained mobile robots," *Auton. Robots*, vol. 16, no. 3, pp. 273–286, 2004.
- [3] K. Iagnemma and S. Dubowsky, *Mobile Robots in Rough Terrain: Estimation, Motion Planning, and Control with Application to Planetary Rovers*, Springer Tracts in Advanced Robotics, vol. 12. Berlin, Germany: Springer, 2004.
- [4] R. Balakrishna and A. Ghosal, "Modeling of slip for wheeled mobile robots," *IEEE Trans. Robot. Autom.*, vol. 11, no. 1, pp. 126–132, 1995.
- [5] R. Gonzalez, M. Fiacchini, and K. Iagnemma, "Slippage prediction for off-road mobile robots via machine learning regression and proprioceptive sensing," *Robot. Auton. Syst.*, vol. 105, pp. 85–93, 2018.
- [6] R. A. Brooks, "A robust layered control system for a mobile robot," *IEEE J. Robot. Autom.*, vol. RA-2, no. 1, pp. 14–23, 1986.
- [7] R. Alur et al., "The algorithmic analysis of hybrid systems," *Theor. Comput. Sci.*, vol. 138, no. 1, pp. 3–34, 1995.
- [8] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2011, pp. 3400–3407.
- [9] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2004.

APPENDIX I

COORDINATE FRAMES AND GEOMETRIC CONVENTIONS

This appendix summarizes the coordinate frames and geometric assumptions used throughout the vision-based localization and navigation pipeline.

- **World Frame $\{W\}$:** A fixed, planar reference frame attached to the workspace floor. All navigation goals and robot positions are expressed in this frame using metric units.
- **Camera Frame $\{C\}$:** A fixed frame attached to the overhead camera, with the Z_C axis aligned with the optical axis of the camera.

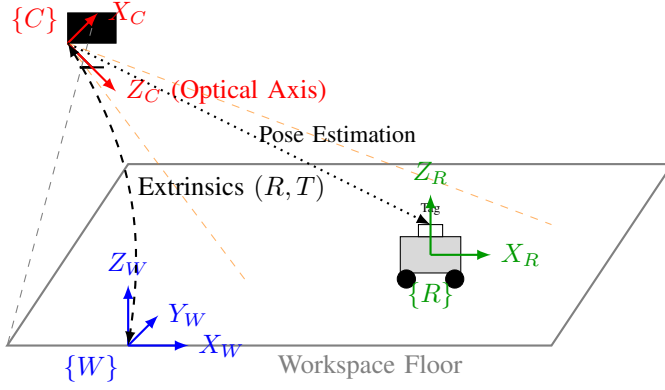


Fig. 10. System Architecture: Global Vision Localization. The camera $\{C\}$ observes the tag on the robot $\{R\}$ to determine its position in the world frame $\{W\}$.

- **Robot Frame $\{R\}$:** A body-fixed frame attached to the mobile robot, with the X_R axis aligned with the robot's forward direction and the yaw angle ψ describing orientation relative to $\{W\}$.

APPENDIX II TRACTION CONTROL ALGORITHM

Algorithm 1 ML-Based Traction Control with Noise Filtering

```

1: Initialize:
2:  $v_{\text{ref}} \leftarrow 0$ ,  $v_{\text{out}} \leftarrow 0$ ,  $\text{count} \leftarrow 0$ 
3: while active do
4:   measure  $v_{\text{wheel}}$ 
5:    $\hat{\lambda} \leftarrow \text{model}(v_{\text{out}}, v_{\text{wheel}})$ 
6:   if  $|\hat{\lambda}| > \lambda_{\text{th}}$  then
7:      $\text{count} \leftarrow \text{count} + 1$ 
8:   else
9:      $\text{count} \leftarrow 0$ 
10:  end if
11:  if  $\text{count} \geq 3$  then
12:     $v_{\text{out}} \leftarrow v_{\text{out}}(1 - \alpha)$ 
13:  else
14:     $v_{\text{out}} \leftarrow v_{\text{out}} + \beta(v_{\text{ref}} - v_{\text{out}})$ 
15:  end if
16:  publish  $v_{\text{out}}$ 
17: end while

```

APPENDIX III CONTROL PARAMETERS

Table I summarizes key control parameters used in the FSM-based traction controller.

TABLE I
SLIP DETECTION AND MITIGATION PARAMETERS.

Symbol	Parameter Name	Value	Description
$\hat{\lambda}$	Estimated Slip	Output	Real-time prediction (m/s)
λ_{th}	Safety Threshold	0.20 m/s	Compromised traction limit
Γ	Filter Window	3 frames	Consecutive violations before mitigation
α	Mitigation Factor	0.10	Slip velocity decay rate
β	Acceleration Gain	0.05	Gain during velocity ramp-up