CARE: A Collaborative Autonomous Robot–Assistive Device System for Everyday Transfers

Gavin Chen

Cornell University Department of Computer Science
Faculty Advisor: Professor Tapomayukh Bhattacharjee
Graduate Research Advisors: Yunting Yan, Ruolin Ye

***Abstract*—In this work, we present an assistive device and robot collaboration system designed to perform human transfers within the context of Activities of Daily Living (ADLs). Building upon our prior work on the Chariot-Kit project—which provides a modular customization framework for assistive devices such as powered wheelchairs, powered hospital beds, and powered patient lifts—we introduce a multi-model control architecture that addresses the three critical phases of sling-based transfer: sling strap tying, transfer execution, and sling strap untying. To avoid full soft-body modeling, we use a proxy-perception strategy that detects and localizes the rigid hanger supporting the strap, paired with a vision and state diffusion policy that executes the untie sequence. Devices are retrofitted with Arduino/Raspberry Pi interfaces and software safety interlocks, and we emphasize manual validation (wiring/timing checks, repeatable sequences) prior to on-device trials. Preliminary prototypes demonstrate reliable autonomous grasping and untying and coordinated device control, enabling progress toward bed-to-wheelchair transfers without manual handling. By combining Chariot-Kit's unified I/O with proxy perception and diffusion-policy execution, the system improves usability, reduces caregiver burden, and moves assistive robotics toward practical, scalable deployment.**

I. Introduction

Robotic systems for assisted living have demonstrated significant potential in supporting individuals with limited mobility, particularly in facilitating Activities of Daily Living (ADLs). Among these, bed-to-wheelchair transfers remain some of the most physically demanding and time-consuming tasks for both caregivers and care recipients [1]. While recent advances have introduced robotic systems to assist with human transfers [2], [3], to the best of our knowledge, a fully autonomous solution has yet to be realized. Existing frameworks still require human intervention during one or more phases of the transfer process.

In this work, we bridge this gap and present CARE, an assistive device and robot collaboration system that has the potential to automate the entire transfer cycle for Human Transfer processes. Building on prior progress in tying and bed→wheelchair coordination, our contribution is a practical approach to grasping deformable sling straps/ropes and autonomously untying the sling from the lift to complete the loop. We use a proxy-perception strategy that detects the rigid hanger supporting the strap and guides grasp/release, executed by a vision- and state-conditioned diffusion policy. Concretely, we instrument a hospital bed, a Hoyer lift, and a powered wheelchair with Arduino/Raspberry Pi control nodes that expose simple motion primitives—bed height/section angles, lift up/down, and chair translate/rotate—via relay/motor drivers and basic safety interlocks. A vision pipeline detects the rigid hanger that supports each sling strap and directs a robot-arm grasp, and a diffusion policy executes the untying sequence. A lightweight task planner coordinates the phases (tie → lift/translate → untie) by issuing discrete device commands, first verified on bench rigs (I/O and timing checks, repeatable sequences) and then exercised on integrated hardware for end-to-end and sectional trials.

II. Design

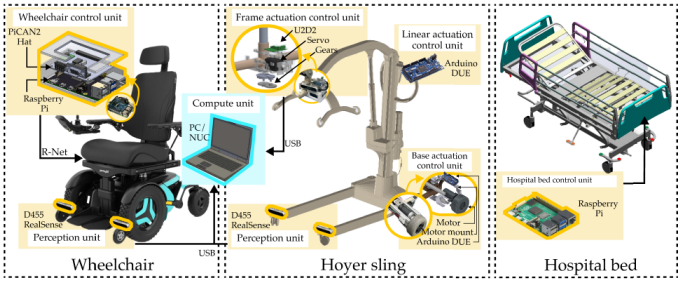


Fig. 1: System overview: instrumented bed, Hoyer lift, and wheelchair; hanger-focused perception; and a coordinator that sequences tie/transfer/untie.

We use the same base hardware layout as prior prototypes (Invacare Hospital Bed, Invacare Reliant-450 Patient Lift, and ROVI Powered Wheelchair) with task-specific updates for reliable strap handling and post-transfer untying. The system comprises perception (hanger/strap localization), device interfaces (discrete motion primitives per device), and a coordinator that sequences phases.

A. Perception

The strap we aim to grasp and manipulate is highly deformable, making it a classic soft-body perception problem. We therefore use *proxy-perception*: rather than modeling the



Fig. 2: Arm locating and grasping strap (no external vision aids).

strap directly, an RGB camera mounted on the robot arm detects and localizes an ArUco [4] marker affixed to the rigid hanger that supports the strap. From the hanger pose, we derive grasp/release targets for the end-effector. A Kinova arm is mounted to the wheelchair frame, equipped with a RealSense D345i to provide the local view needed for precise grasping and subsequent operations.

Tying (prior work): For the tie/attachment phase we adopt the CART-MPC method for coordinating the arm and sling bar [2].

Untying (this work): Our contribution is an untying stack that leverages the same proxy-perception targets and executes the release with a vision+state diffusion policy (details below).

B. Untie Policy Execution

For **untying**, a vision+state diffusion policy takes the current hanger pose, end-effector pose, and a short observation history, and outputs short-horizon waypoints plus open/close events. Preconditions include a valid hanger ID and a secured grasp. Termination is declared when the strap clears the hanger; otherwise, the policy re-plans within a bounded workspace around the hanger.

III. Sling Strap Grasping

Our goal is to reliably acquire the sling strap using proxy-perception and execute a short, safe approach with the Kinova arm. We detect the *hanger* (not the strap) with a fiducial and generate grasp targets; motion is planned with an RRT-based planner, with simple recoveries (base/arm adjustment) when reach limits are encountered.

A. Perception → Target Generation

Proxy-perception: An RGB camera on the arm detects an **ArUco** marker affixed to the rigid hanger. From the marker pose we compute the hanger frame H and a hook-region target.

Frames & transforms: We maintain fixed extrinsics between the camera and end-effector (T_C^E), and between the wheelchair and world (T_{chair}^W). Each detection yields:

$$T_H^C \;\Rightarrow\; T_H^E = T_C^E\, T_H^C, \quad {}^E p_{\text{grasp}} = {}^E p_H + \Delta_{\text{hook}}.$$

where Δ_{hook} is a known offset from the marker center to the desired hook contact.

Smoothing & gating: We apply a short EMA on T_H^C and require a minimum detection confidence/area; if confidence drops or the pose jumps beyond a bound, the coordinator pauses and requests a re-acquisition.

Pre-grasp & approach: We define a pre-grasp ${}^E p_{\text{pre}} = {}^E p_{\text{grasp}} + d\,\mathbf{n}$ along the hanger normal $\mathbf{n}$, then approach along $\mathbf{n}$ with the gripper open.

B. RRT Path Planning & Execution (Kinova)

Planning problem: Given current joint state q, plan a collision-free path to the pre-grasp pose, then to the grasp pose. We use an *RRT*-based planner in joint space with:

- joint and velocity limits from the Kinova model,
- self- and environment-collision checks (bed rails, hanger bar) with inflated safety margins near the hook

Execution sequence:

1) Plan $q \to q_{\text{pre}}$ (to ${}^E p_{\text{pre}}$); we recorded 3 poses to segment the arm control.
2) Plan $q_{\text{pre}} \to q_{\text{grasp}}$; execute a short, straight-line approach in task space for the last centimeters.
3) Close gripper; retreat a small distance along $-\mathbf{n}$; hands off to tying/untie phases.

Recovery behaviors: If IK/planning fails or the pose is outside the arm's workable envelope, we trigger a small wheelchair reposition (forward/back or lateral nudge) *or* adjust the commanded hanger/arm height, then re-plan. These recoveries mirror what we observed experimentally: nominal heights succeed readily; near reach limits, small base/arm adjustments resolve most cases.

C. Relation to Tying and Untying

Tying (prior work). For the tie/attachment phase we adopt CART-MPC to coordinate the arm and sling bar [2].

Untying (this work). After a secured grasp, we pass the hanger and end-effector states to a vision+state diffusion policy that executes the release within a bounded workspace around the hanger.

IV. Untying Formulation and Policy

A. Problem Definition

We address the problem of untying straps secured to hooks on a patient lift during the post-transfer stage, in which the care recipient is seated in a wheelchair with the sling remaining underneath them, and all four sling straps are attached to the lift's hooks. A robotic arm is used to perform the untying operation, and we use a manikin in place of the care recipient. The task requires manipulation of deformable straps in a cluttered setting where the robot must remove each strap without damaging the sling or the manikin sitting in the wheelchair.

Our task is modeled in a continuous action space, with observations consisting of two RGB images and the robot's end-effector state. The image observations include o_t^{wrist}, captured from a wrist-mounted camera, and o_t^{bar}, captured from a

camera facing the hanger bar. The robot state is represented in end-effector space. The model generates executable actions $a_t = (p_t, q_t)$, where $p_t = (x_t, y_t, z_t) \in \mathbb{R}^3$ denotes the Cartesian position, and $q_t \in \mathbb{R}^4$ denotes the quaternion orientation of the end-effector such that $q_t = (w, x, y, z)$.

The objective is for the robot to execute a trajectory $\{a_1^i, \ldots, a_T^i\}$ over a time horizon T that transitions the system from an initial state, where the strap is attached, to a final state in which the straps are successfully untied:

$$s_T^i = \mathcal{T}(s_0^i, \pi_\theta^i), \quad \text{such that } \textsc{Untied}(s_T^i) = \texttt{True}.$$

B. Method

We formulate the untying task as a visuomotor control problem using a conditional denoising diffusion policy. Given the observation $o_t = \{o_t^{\text{wrist}}, o_t^{\text{bar}}, o_t^{\text{ee}}\}$ at timestep t, the goal is to generate an end-effector action $a_t = (p_t, q_t)$, where $p_t \in \mathbb{R}^3$ denotes the Cartesian position and $q_t \in \mathbb{R}^4$ denotes the quaternion orientation, as defined in Section IV-A.

1) Diffusion Policy Formulation: We adopt the Diffusion Policy framework [5], which formulates policy learning as a denoising diffusion probabilistic model (DDPM) over actions. Instead of directly regressing actions from observations, the policy learns to sample from the conditional distribution $\pi_\theta(a_t \mid o_t)$ by refining Gaussian noise into feasible actions via iterative denoising steps.

At inference time, an initial noisy action $a_t^{(K)} \sim \mathcal{N}(0, I)$ is sampled and denoised over K steps according to:

$$a_t^{(k-1)} = \alpha_k \left(a_t^{(k)} - \gamma_k \epsilon_\theta(o_t, a_t^{(k)}, k) \right) + \mathcal{N}(0, \sigma_k^2 I),$$

where ϵ_θ is the learned noise prediction network, and $\alpha_k, \gamma_k, \sigma_k$ are fixed parameters of the noise schedule. The final output $a_t^{(0)}$ is used as the action a_t.

2) Architecture and Conditioning: The policy network (Fig. 3) follows the UNet-based architecture described in [5], with FiLM (Feature-wise Linear Modulation) conditioning applied to the visual features extracted from both o_t^{wrist} and o_t^{bar}. Each image is encoded independently using a ResNet-18 backbone, and the resulting features are concatenated and used as conditioning input throughout the denoising process. The network is trained end-to-end using mean squared error loss between predicted and ground-truth noise.

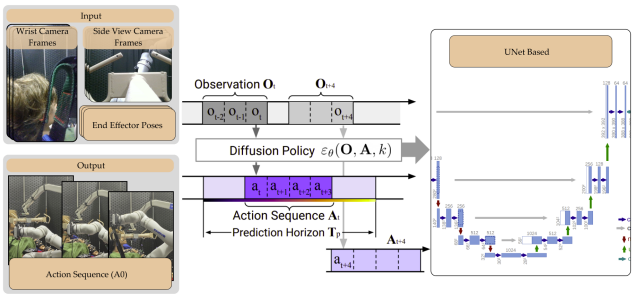


Fig. 3: Diffusion policy overview

3) Strap-Specific Policies: Because the visual and geometric configuration varies across straps, we train four separate policies $\pi_\theta^{(i)}$, one for each strap $i \in \{1, 2, 3, 4\}$. Each policy is trained independently using demonstrations for untying the corresponding strap. At inference time, the robot executes the four policies sequentially to untie all straps.

4) Closed-Loop Execution: At runtime, the policy generates actions in a closed-loop fashion: at each timestep t, a new observation o_t is used to condition the diffusion policy, which outputs a 6-DoF pose a_t. The robot executes a_t, then re-observes the scene to determine the next action. This allows the system to react dynamically to changes in strap configuration and contact conditions during the untying process.

V. Experiments

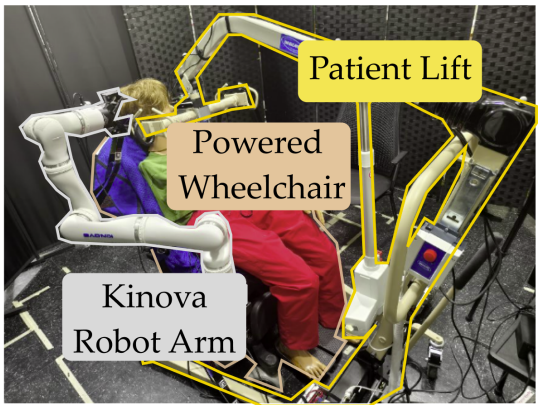


Fig. 4: Task setup.

Our experiment is conducted using the setup from our previous work [2], specifically employing a Kinova Gen3 7 DoFs robot arm [6] for the untying task, as shown in Fig. 4. We utilize two types of patient lifts, an Invacare Reliant 450 Battery-Powered Patient Lift and an Invacare 9805P Manual Patient Lift, as well as a ROVI A3 powered wheelchair and a manikin. The two patient lifts have different hanger bar designs (Fig. 5) and are used to train and test the learned policy model. Both the patient lifts and the wheelchair remain stationary during the experiment, with the robot arm being the only active agent. For perception, we employ three Intel RealSense D415 cameras: two mounted on each side of the hanger bar of the patient lift in use and one mounted on the robot's wrist.

Our system runs on an AMD Ryzen 9 5900X CPU (3.7 GHz base clock) and is equipped with an NVIDIA GeForce RTX 3070 GPU featuring 8 GB of GDDR6X memory and operates on Ubuntu 20.04 with ROS Noetic.

A. Strap-untying Dataset

We train strap-specific diffusion policies using human demonstrations collected through a custom tele-operation interface. Human operators perform the untying task using a joystick that controls the end-effector pose of the Kinova robot arm via a position controller. To begin data collection, the

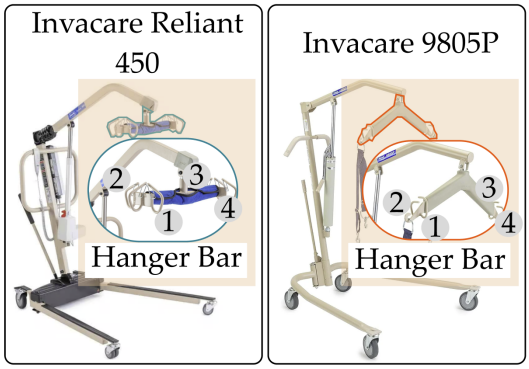


Fig. 5: *Patient Lifts*: On the left is the Reliant 450 Powered Lift (powered lift), and on the right is the 9805P Manual Lift (manual lift). The powered lift is used for both training dataset collection and experimental testing, while the manual lift is used for testing only. The numbering represents the predefined order in which the straps are untied.

human expert positions themselves at a designated spot near the patient lift, wheelchair, and robot arm to maintain full visibility of the setup while avoiding occlusion of the camera's view. Each trial starts with all straps securely tied to the sling hooks and a manikin seated in the wheelchair. The hanger bar remains static, allowing passive pitch at the pivot. Prior to each trial, the robot arm grasps a strap with its gripper. The operator then uses the joystick to manipulate the end effector and untie the strap from the hook. At every timestep, we record synchronized observations, including RGB images from both wrist-mounted and bar-mounted cameras, as well as proprioceptive data from the robot at 10 Hz. After each trial, the expert resets the environment to its initial configuration. Once data for one strap is complete, they proceed to the next, repeating the process for all four straps. In total, we collected 60 trials for each strap and 240 trials in total.

B. Model Training

We train one diffusion policy per strap using the human demonstration data collected. Each demonstration consists of synchronized RGB observations along with the corresponding proprioceptive data during untying. The policy is trained to model the conditional distribution $\pi_\theta(a_t \mid o_t)$, where a_t is the end-effector pose and o_t is the combined visual observation at timestep t.

Following the Diffusion Policy framework [5], we formulate policy learning as a conditional denoising diffusion process over the action space. During training, we perturb the ground-truth actions with random Gaussian noise. The model learns to predict this noise using a mean squared error loss. The diffusion model consists of a UNet-based denoising network conditioned on visual features extracted by a ResNet-18 encoder. Observations from both camera views are processed independently and fused before conditioning the denoising network using Feature-wise Linear Modulation (FiLM).

We train each model using the Adam optimizer with a fixed learning rate of 1.0^{-4}, on 1 A100 GPU on Nvidia Saturn Cloud. Policies converge within 50k training iterations for all straps. At inference time, the model generates actions by iteratively refining noise through 8 denoising steps [5].

C. Preliminary Evaluation

In this section, we evaluate the performance of our diffusion policy model on the task of autonomous strap untying using two patient lift setups, as shown in Sec. V. We assess performance based on the success rate, defined as the proportion of trials in which the strap was successfully untied from the hook. We conduct 10 test trials for each strap on each device, and the testing sequence follows Fig. 5.

1) Results: Table I summarizes the performance across all trials. Our policy achieves a success rate of 77.5% on average across the four straps, with slightly lower performance on strap 2. For the manual lift, the policy for untying straps 3 and 4 would lead to a collision with the lift; therefore, these straps were not tested.

TABLE I: Strap Untying Performance Across Straps and Setups

Strap	**Powered Lift Success Rate (%)**	**Manual Lift Success Rate)**
Strap 1	0.9	0
Strap 2	0.6	0
Strap 3	0.8	Collision
Strap 4	0.8	Collision
Average	0.78	0

2) Discussion: The results on the powered patient lift demonstrate that the diffusion policy model is capable of learning visual-action associations for the untying task. Notably, the success rate for strap 2 is lower compared to the other straps. We argue that this may be due to limitations in the training data, which the human expert demonstrations prioritized task completion speed over safety and task quality. Although these demonstrations successfully completed the untying, the robot's end effector often moved close to the patient lift, and the strap was not displaced far enough from the hanger bar hook. This emphasizes the importance of demonstration quality in determining the inference performance of the diffusion policy.

Additionally, for the powered lift, we observed that the robot exhibited excessive compliance when fully extended. External forces or perturbations on the end effector led to oscillations in the robot arm. While these oscillations did not lead to instability, they may have contributed to task failure.

Comparing straps 1 and 2 across the powered and manual lifts, we note that policy failures may be attributed to both insufficient training data and differences in the hook design of the hanger bar. Nevertheless, the policy produced an upward untying motion on the manual lift (Fig. 6), which aligns with the strategy used by the human demonstrator. However, the robot continued this upward motion until stalling.

For straps 3 and 4, failures were due to collisions, which may be attributed to limitations in the end-effector space controller. A null space controller could mitigate this issue. More broadly, these failures suggest that incorporating environmental perception is necessary for effective collision avoidance during manipulation.

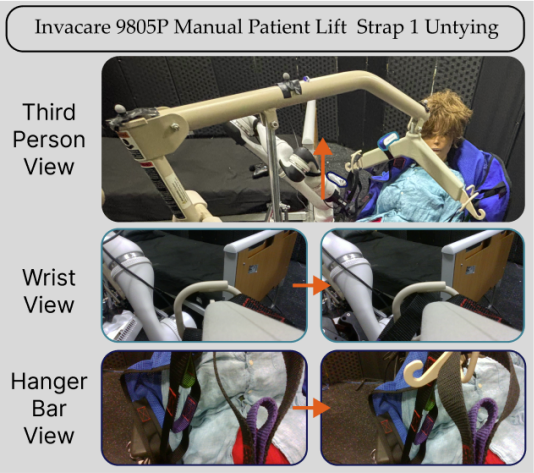


Fig. 6: Untying Strap 1 with the manual patient lift. The arrow shows the action direction of the robot end effector and camera views before and after the action.

VI. Testing

A. Test Design

We conducted preliminary tests to assess whether the subsystems meet minimum performance for autonomous strap handling and post-transfer untying. Tests were staged from components to integration, and outcomes were logged as success, partial (cases where despite the movements being executed correctly, due to environment instability or affect, it was not possible to complete the procedure), or failure with notes on failed trials.

1: Base mobility (wheelchair). We exercised waypoint-following and stop behavior in our laboratory environment. Prior to the JSM control-block change, runs exhibited intermittent–frequent stalling (electronics audibly engaged, but no motion). On the R-Net bus, the joystick module (JSM) heartbeat continuously publishes the last commanded speed to keep the motors at that speed; when the stick is neutral it repeatedly asserts zero-velocity. Our Arduino-injected velocity commands therefore raced with the JSM heartbeat, and the neutral frames routinely overwrote our commands, triggering brakes and stalls. To obtain a reliable control path, we inserted a small control block at the JSM harness: in autonomous mode it isolates/blocks the JSM heartbeat so only the planner's stream reaches the motor controller; otherwise, it passes the joystick through unchanged. After isolation, velocity tracking and stopping became consistent and repeatable.

2: Control interface consistency. Using the Xbox-compatibility shim, we replayed constant-velocity and step-input scripts to verify deterministic command acceptance and stable planner-to-base latency across repeated trials.

3: Manipulator grasp feasibility. With proxy-perception of the hanger, we tested Kinova end-effector grasps across a grid of hanger heights/offsets. Failures were dominated by reachability (hanger set below the arm's workable envelope); repositioning the wheelchair resolved most cases, confirming the need for base repositioning as a recovery action.

4: Integrated nav + grasp. We combined navigation with grasp: the wheelchair repositioned to a target, the arm acquired the hanger, and the system attempted a strap grasp. We observed a decent number of preliminary successes in this integrated setting; full end-to-end transfers still require some more tests to evaluate their reliability and versatility.

B. Test Procedure

1) **Setup and calibration:** Initialize all device nodes; begin pose trackers; verify camera feed; check extrinsics between camera, end-effector, and wheelchair frame; run a quick test on the hanger target to make sure that the camera should correctly respond to the marker.
2) **Component trials:**
   a) *Mobility:* Execute waypoint tracks and stop tests; record stalls/latency before and after the JSM-block change.
   b) *Interface:* Replay constant-velocity and step-input scripts via the Xbox-compatibility shim; log command acceptance and timing.
   c) *Grasping:* Sweep hanger height/offset; attempt grasp; record reachability and grasp outcome.
3) **Integrated Trials (Navigation and Grasping):** Navigate to a predefined pose, sweep in a small radius to locate the hanger, plan movement, and attempt a strap grasp. If perception confidence drops or reachability fails, trigger base repositioning and retry.
4) **Logging Data:** For each run, log timestamps, commands, acknowledgments, poses, and outcome class. A trial passes if all phase transitions occur within budget and the final outcome is a secured strap grasp.

Status: Current results reflect preliminary successes at the nav + grasp level. We still require a larger number of full end-to-end runs (tie $\rightarrow$ lift/translate $\rightarrow$ untie) to report quantitative success rates, timing distributions, and robustness under lighting/occlusion variants.

VII. Results

*Unless otherwise noted, we count **partial** outcomes as **successes** when reporting success rates.*

A. Wheelchair

We evaluated the wheelchair's ability to execute planner commands before and after isolating the joystick module (JSM) heartbeat with a control block. Before the change, stalls were frequent (22/40 successes including partials; 55%); after isolation, behavior became deterministic with no hard failures (40/40 including partials; 100%).

B. Grasping

We measured grasp outcomes across three height bands relative to a nominal setup. Non-successes at the extremes were typically recoverable by small base/arm adjustments (e.g., moving the wheelchair forward or shifting arm height); aggregated success across bands was 70.5%.

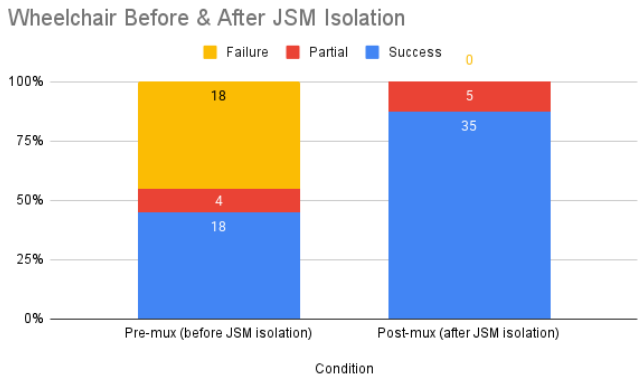


Fig. 7: Successful wheelchair executions before vs. after JSM isolation (partials counted as successes).

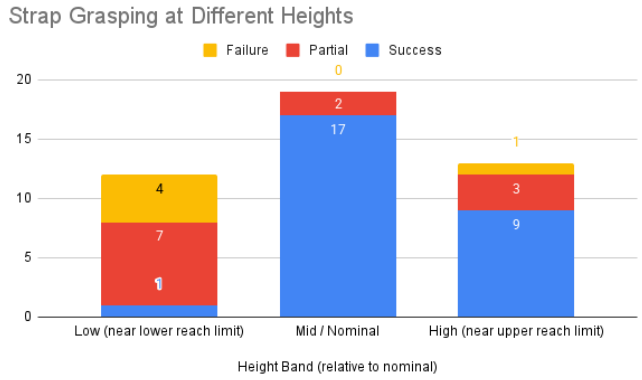


Fig. 8: Successful strap grasping at different set height poses.

C. Full Integration

We combined navigation with grasp in early integrated trials. Full end-to-end transfer (tie to lift/translate to untie) testing is ongoing; current integrated outcomes are 3/8 full successes, 3/8 partial, and 2/8 failures.

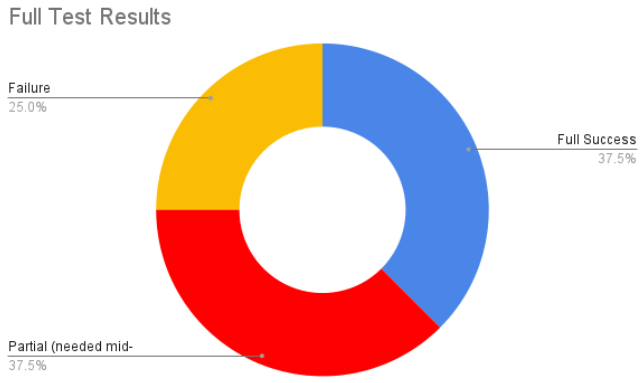


Fig. 9: Successful full end-to-end tests (preliminary).

VIII. Discussion

A. Review of Expected Findings

We expected two things to matter most for reliability: (1) stabilizing the wheelchair control path by removing bus contention with the joystick module (JSM), and (2) making strap handling practical by combining proxy-perception of the hanger with base/arm adjustments when reach limits are hit. We also anticipated that integrated navigation and grasp would surface residual timing and reachability issues before full end-to-end transfers.

B. Comparison of Actual Results

Wheelchair control: Results match expectations: isolating the JSM heartbeat converted an unreliable interface (22/40 including partials; 55%) into a deterministic one (40/40 including partials; 100%) with no hard failures (Fig. 7); occasional partials were all due to inconsistencies in environmental factors.

Strap grasping: Outcomes depended strongly on viable ranges of motion. (Fig. 8). At nominal heights the success rate was high (85%); near lower/upper reach limits, non-successes were typically resolved by small chair repositioning or arm-height adjustments. Aggregated success across bands was 70.5%, this suggests that we should allow for base movement in some scenarios so as to guarantee successes.

Integrated navigation and grasping: Early integrated trials yielded 3/8 full successes, 3/8 partials, and 2/8 failures (Fig. 9). While device I/O is stable, additional tuning and data are needed for consistent one-shot execution and tighter phase coordination.

C. Conclusions from Preliminary Data

- After adding the control block, base motion is reliable with stable timing, enabling higher-level policies to assume consistent actuation.
- Reachability dominates failure modes; integrating automatic reposition interwoven with error handling should lift success beyond the current 70.5%.
- Navigation to grasping works in a meaningful fraction of runs; full tie to lift/translate to untie requires much more trials and tuning.

IX. Conclusion

A. Summary

These preliminary results prove that stabilizing the wheelchair control path and handling straps via proxy-perception are extremely effective directions for closing the loop. Isolating the JSM heartbeat transformed base control from an intermittent 55% success pre-blocking, counting partials as successes, to deterministic 100% success post-blocking, with no hard failures. Grasping success depended on range — 85% at nominal height vs. lower rates near reach limits—yet was typically recoverable with small base/arm adjustments, yielding 70.5% overall. Early integrated navigation to grasping runs achieved 3/8 full successes with 3/8 partials

and 2/8 failures. Together, the data support our direction towards bus isolation for reliable actuation and proxy-perception plus repositioning for practical deformable-object handling.

B. Outlook and Future Research

Next steps focus on scale, safety, and full-sequence robustness. We will (1) expand integrated trials across hook types, heights, and lighting to report success probabilities, timing distributions; (2) encode autonomous recovery (base/arm reposition) derived from observed fixes; (3) broaden safety mechanisms with tighter software interlocks and a physical hardware caregiver override; and finally, (4) extend beyond the current navigation to grasping segment to a full tie→lift/translate→untie cycle. Other areas that require improvement could include improved perception under occlusion or fault-injection testing of the control path.

REFERENCES

[1] H. R. Kulich, L. Wei, T. M. Crytzer, R. A. Cooper, and A. M. Koontz, "Preliminary evaluation of an automated robotic transfer assist device in the home setting," *Disability and Rehabilitation: Assistive Technology*, vol. 18, no. 5, pp. 511–518, 2021, pMCID: PMC10759809. Author manuscript in PMC (2024-01-02). [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC10759809/

[2] "Cart-mpc: Coordinating assistive devices for robot-assisted transferring with multi-agent mpc," https://arxiv.org/abs/2501.11149, arXiv:2501.11149 [cs.RO], v1, Jan. 2025.

[3] R. Bolli Jr, P. Bonato, and H. Asada, "Handle anywhere: A mobile robot arm for providing bodily support to elderly persons," *arXiv preprint arXiv:2209.15169*, 2022.

[4] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.

[5] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," https://arxiv.org/abs/2303.04137, arXiv:2303.04137 [cs.RO], v5, Mar. 2024.

[6] "Kinova modular robot arms for service robotics applications," https://www.researchgate.net/publication/322410820_Kinova_Modular_Robot_Arms_for_Service_Robotics_Applications, white paper, accessed Aug. 2025.