

# Development of a Lab-Scale Prototype of a Cable-Driven Parallel Robot (CDPR) for Automated Installation of Prefabricated Building Envelopes

Yifang Liu<sup>1</sup>, Nolan W. Hayes<sup>1</sup>, Balaji Selvakumar<sup>1</sup>, Diana Hun<sup>1</sup>, and Bryan P. Maldonado<sup>1</sup>

**Abstract**—This paper presents preliminary results on the design and implementation of cable-driven parallel robots (CDPRs) for automated installation of prefabricated building envelopes, aiming to address the challenges associated with traditional manual placement methods. Traditional installation techniques for prefabricated components, such as panelized systems, unitized curtain walls, or overclad retrofits, are labor-intensive, prone to misalignment, and often constrained by space limitations. To tackle these challenges, we developed a lab-scale CDPR prototype for autonomously installing envelope components. This paper focused on the mechanical design, system integration, and the control strategies for the robot. Experimental results validate the full integration of the robot and the effectiveness of the system’s low-level control strategy.

## I. INTRODUCTION

The increasing demand for automation in industries such as logistics, construction, and aerospace has driven the need for robotic systems capable of high-speed, precise, and large-scale manipulation. Among these, cable-driven parallel robots (CDPRs) have emerged as a promising solution due to their ability to transport heavy loads with precision over expansive workspaces. In the construction sector, automation is becoming increasingly critical, particularly for energy-efficient buildings assembled using panelized systems, such as overclad retrofits or curtain walls [1]. In particular, prefabricated retrofits have gained attention as an effective solution for improving the energy performance of existing structures. However, the installation of these prefabricated panels remains a significant challenge. Traditional methods rely on manual labor and crane-assisted placement, which are not only labor-intensive but also prone to misalignment, safety risks, and logistical difficulties—especially in dense urban environments with restricted space.

To address these challenges, prior efforts have introduced the real-time evaluation (RTE) [2], [3] systems to enhance manual installation accuracy by providing real-time feedback during panel hoisting and positioning. While this approach improves precision and reduces rework, it still relies heavily on human intervention. To further enhance efficiency and

Notice: This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). The US government retains and the publisher, by accepting the article for publication, acknowledges that the US government retains a non-exclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for US government purposes. DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (<https://www.energy.gov/doe-public-access-plan>).

<sup>1</sup>Buildings and Transportation Science Division, Oak Ridge National Laboratory, Oak Ridge, TN 37830 USA {liuy5, hayesnw, selvakumarb, hunde, maldonadobp}@ornl.gov



Fig. 1. Conceptual view of a CDPR for retrofitting a residential building using prefabricated panels in a densely populated urban area [4].

safety, this paper explores the use of CDPRs for automated panelized building retrofits [4]. We propose a CDPR system that autonomously lifts prefabricated panels and positions them accurately before construction workers fasten them onto the building facade, see Fig. 1.

This paper presents the design and development of a lab-scale CDPR prototype (Fig. 2) for prefabricated panel installation. We provide a comprehensive description of the system’s mechanical structure, as well as the selection of actuation, cabling, and sensing components. Additionally, we present the control design, including a description of the servo and torque controllers integrated in Robot Operation System (ROS). Experiments demonstrated that the system is fully integrated and the torque control operates effectively.

## II. PROTOTYPE DESIGN

### A. Frame and Moving Platform Design

To ensure the frame can withstand the maximum required force, we selected aluminum extrusions as the material for its construction. For real-world retrofitting applications, the robot’s width and height should match those of the retrofitted facade, while its depth can vary. In densely populated urban areas, the robot’s depth can be designed to match the width of the sidewalk in front of the building, minimizing disruption to traffic and pedestrians. The dimensions of our lab-scale CDPR model are 3.5 m in height, 2.7 m in width, and 1.5 m in depth. The end effector measures 0.67 m in height, 0.63 m in width, and 0.013 m in depth. Here, we assume that the panel size matches the dimensions of the end effector. The cables are equipped with hooks that attach to hoist rings positioned near the eight corners of the end effector.

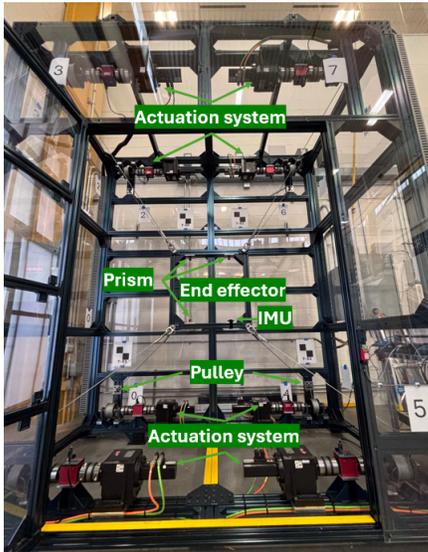


Fig. 2. Lab-scale CDPR prototype within safety enclosure.

The prototype also features reconfigurable anchor points, allowing cables to be arranged in either a standard or crossed layout by simply detaching and reattaching the hooks.

### B. Cable Guiding System

In CDPRs, electric motors control the cable lengths to adjust the pose of the end effector. To prevent random winding of the cable on the drum, most CDPR systems use a threaded winder to constrain lateral winding, a widely adopted solution in existing research [5], [6]. The drum selection process is guided by several key requirements. First, the drum must have sufficient capacity to wind the maximum cable length, which corresponds to the diagonal of the frame. Second, its load capacity must exceed the system's maximum load requirement. Third, the drum should be threaded, with groove dimensions that match the selected cable. Based on these criteria, we selected a drum originally designed for garage doors. The flat moment arm for a 4.8 mm radius cable is 6.91 cm, and the flat portion of the drum accommodates 12 cable revolutions, corresponding to a total cable length of 5.2 m. The choice of cables was determined through experimental characterization of two materials: steel and ultra-high molecular weight polyethylene. Winding and unwinding tests on the drum revealed that steel cables remained more consistently within the drum grooves. As a result, we selected an ultra-flexible galvanized steel rope with a diameter of 4.8 mm and a load capacity of 3559 N. To minimize cable wear, we incorporated swivel pulleys. However, this choice introduces some geometric errors, which will be further examined in future optimizations.

### C. Actuation System

The rotary actuation system in our CDPR model consists of servo-actuated winder, where cables are coiled onto cylindrical drums. The motor selection process considers both torque and speed requirements, with gearbox-assisted

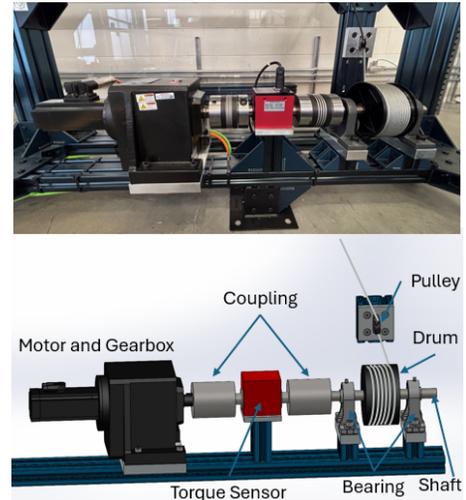


Fig. 3. CDPR actuation system for a single axis.

actuation enhancing performance. Torque sensors are integrated into the system to enable precise force control. The hardware configuration includes eight Nidec M753 servo drives connected to eight servomotors via EtherCAT. Torque measurements are captured using eight ATO Digital Rotary Torque Sensors, which interface with the Trio P378 Flexslice 8 Analogue Output module. A Trio P662 eight-axis motion controller manages the low-level control of the system. Stober servo motors and gearboxes provide reliable and efficient actuation, ensuring smooth and controlled cable manipulation. The actuation system is demonstrated in Fig. 3.

## III. CONTROL DESIGN

### A. Drive Control

The low-level control of the cable-driven parallel robot is implemented using Motion Perfect software and the Trio eight-axis motion controller. This system ensures safe and precise operation by managing motor commands and enforcing key constraints. A dedicated table in the Trio controller's memory is used to store and exchange critical control parameters, allowing real-time read and write access. An infinite loop continuously reads motor states, encoder values, and torque sensor data, updating the table to provide real-time monitoring and feedback. This table also enables communication with the external PC for reading motor encoder positions and velocities, and for sending commands for motor control. To enhance safety, torque limits are defined to prevent excessive forces on the hardware components, such as pulleys, drums, and cables. Cable length constraints are also enforced to prevent the cables from winding or unwinding beyond safe limits. For interaction with higher-level control, the system requires data to be written into predefined table addresses. These low-level control strategies ensure smooth and reliable operation while safeguarding the system's mechanical and electrical components.

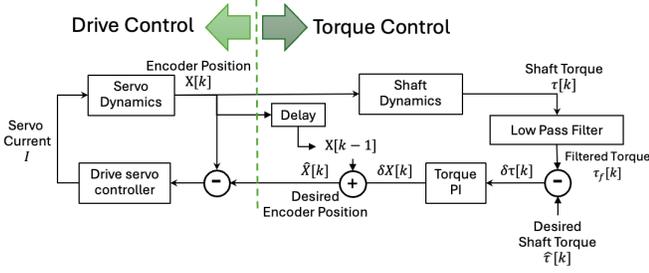


Fig. 4. Torque proportional-integral (PI) controller

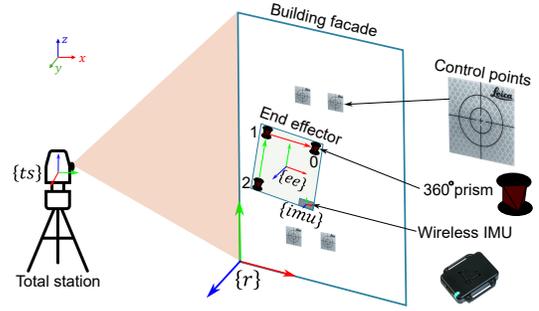


Fig. 5. Coordinate frame transformations.

## B. Torque Control

As illustrated in Fig 4, the torque PI control loop ensures accurate torque regulation by continuously adjusting the motor's position command based on torque feedback. The system measures the shaft torque  $\tau$  and applies a low-pass filter to obtain the filtered torque  $\tau_f$ . The filtered torque is then compared to the desired shaft torque  $\hat{\tau}$ , and the resulting error  $\delta\tau$  is processed by the torque PI controller. The controller generates a correction  $\delta\chi$ , which is added to the current encoder position, producing a modified encoder position command  $\hat{\chi}$ . This desired encoder position  $\hat{\chi}$  is then sent to the low-level drive controller to generate the necessary servo current  $I$  to drive the motor. By integrating torque control with drive control in this manner, the system effectively regulates torque while maintaining precise motor actuation, enabling smooth and stable operation of the CDPR.

## C. State Estimation

The states considered in this study include position, orientation, linear velocity, and angular velocity. It is important to note that not all states can be measured easily or precisely. For those states that cannot be directly observed or that cannot be measured with sufficient precision, algorithms like the Kalman filter can be employed to estimate the missing states. In construction, total stations are widely accessible and frequently used for highly precise angle, distance, and coordinate measurements. Given these considerations, in our application, we used a total station for position measurement. The orientation and angular velocity of the end effector are measured by an inertial measurement unit (IMU). We chose a wireless IMU to minimize the number of cables in the robot workspace, which is highly desirable on a real-world construction environment.

## D. Workflow

1) *Resection*: The objective of resection is to align the total station with the robot frame. The robot's geometry consists of the positions of eight proximal anchors, the dimension of the end effector, and the position of four known control points. The geometry of the robot is pre-measured and stored, making the digital model of the robot. Four Leica reflective tapes are attached to the robot's frame to represent the control points. The first step is to establish the transformation between the robot and the total station. We use the resection algorithm proposed in by Tang *et al.* [3].

First we use the total station to measure the position of the four control points in the total station  $\{ts\}$  frame, then the translation and rotation between the measured control points and their positions in the digital model are calculated using Kabsch algorithm [7]. The rotation is used to calculate the angle (azimuth) for the total station. Note that only the yaw angle is needed since the total station is set to level before each use, which results in the  $x$ - and  $y$ -axis always forming a horizontal plane within machine tolerance. As a result, the  $y$ -axis of the total station is perpendicular to the building facade. The outcome of the resection process establishes the transformation between the  $\{ts\}$  and the robot  $\{r\}$  frames.

2) *Establishing the Transformations*: After getting the transformation between the  $\{ts\}$  and  $\{r\}$ , we need to estimate the initial transformation between the end effector  $\{ee\}$  frame and the  $\{ts\}$  (or  $\{r\}$ ) frame. Here we assume the transformation from  $\{ee\}$  to the IMU  $\{imu\}$  frame is fixed, demoted as  $T_{\{ee\} \rightarrow \{imu\}}$ . To define the body frame of the end effector, three Leica 360-degree mini prisms are attached to it. We use the total station to measure the position of these three prisms. As illustrated in Fig. 5, the vector formed between prisms 0 and 1 defines the body's  $x$ -axis, and the vector formed between prisms 1 and 2 defines the body's  $z$ -axis. The  $y$  axis is then determined by the cross product of the  $x$ -axis and  $z$ -axis vectors. By calculating these relationships, we can obtain the transformation  $T_{\{ts\} \rightarrow \{ee\}}$ . Consequently, all frame transformations are established.

3) *Panel Position Tracking*: Once the frame transformations are established, the total station promptly switches to target tracking mode, locking onto prism 0. The tracking process continuously provides precise position estimates for the end effector. Meanwhile, the orientation of the end effector is estimated using the IMU.

4) *Trajectory Following*: A predefined trajectory—such as a torque trajectory, end effector state trajectory, or servo angle trajectory—can be sent to the robot for execution. To make sure the robot follows this trajectory, feedback control strategies are commonly employed.

5) *Final position and orientation measurement*: After executing the trajectory, we perform precise orientation measurements using the total station by cycling through three prisms to compute the final transformation matrix  $T_{\{r\} \rightarrow \{ee\}}$ .

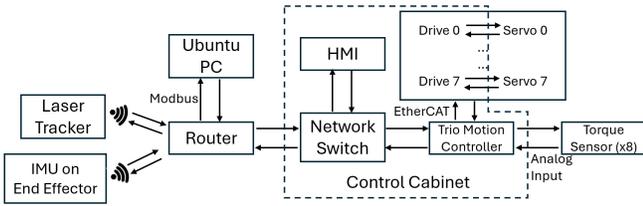


Fig. 6. CDPR network architecture.

### E. System Implementation

The system integrates multiple components for motion control and real-time state monitoring. As shown in Fig. 6, the central communication hub is a router, along with:

- The total station and IMU that provide real-time position and orientation measurement, respectively. These sensors communicate wirelessly with the router.
- Inside the control cabinet, the network switch facilitates communication between the human-machine interface (HMI), Ubuntu PC with ROS, and the Trio Motion Controller using Ethernet-based protocols.
- The Ubuntu PC communicates with the control cabinet using Modbus protocol.
- The Trio Motion Controller communicates with all 8 servo drives using EtherCAT.
- The Motion Controller receives analog inputs from the 8 torque sensors to monitor the applied forces.

For data processing, communication, and feedback control strategies, Python and ROS Noetic are employed. The system supports two operational modes: manual and automatic. In manual mode, each axis can be jogged directly through the HMI. Conversely, when the robot transitions to auto mode—controlled by Python and ROS—manual mode is automatically disabled to ensure safe and efficient operation.

## IV. PERFORMANCE EVALUATION

### A. Resection Accuracy

After performing resection, the transformation between the total station and the robot is established. The control points can be measured in two ways: First, we define the control points in the robot frame as  $\mathbf{cp}_{robot}$ . Then, we can convert  $\mathbf{cp}_{robot}$  into the  $\{ts\}$  frame, denoted as  $\mathbf{cp}_{ts}$ . Additionally, we can measure these control points directly using the total station, represented as  $\tilde{\mathbf{cp}}_{ts}$ . The mean errors between  $\mathbf{cp}_{ts}$  and  $\tilde{\mathbf{cp}}_{ts}$  are calculated as  $[-0.5, 0.1, 0.05]$  mm, and the root mean square error is 0.6 mm.

### B. Torque Control

We used a model-based approach to generate a feasible torque trajectory from a lower position to a higher position within the robot frame. First, we activated auto mode and started the Modbus server ROS node. The torque control methods outlined in Section III-B were then employed. The commands sent to the servos from the torque controller correspond to the specified desired angles. As shown in Figure 7, the results depict the measured torques alongside the desired torque across all eight axes.

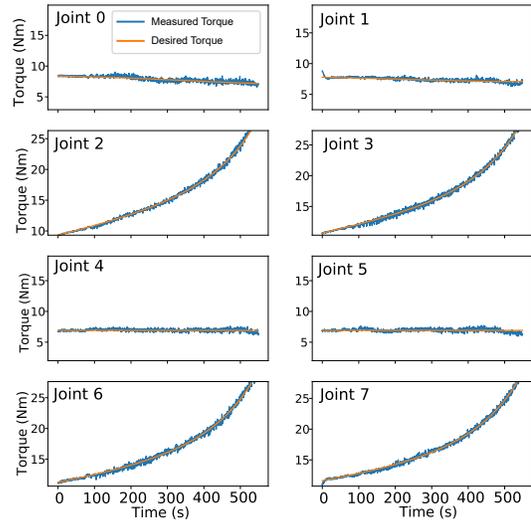


Fig. 7. Torque control results.

## V. CONCLUSION AND FUTURE WORK

This paper presented the preliminary design, development, and validation of a lab-scale cable-driven parallel robot (CDPR) prototype for automated installation of prefabricated building envelopes. Experimental results demonstrated that the prototype can successfully follow pre-planned force trajectories using torque control. Future work will incorporate position feedback to enable accurate position tracking to improve the system’s autonomy for real-world deployment.

### ACKNOWLEDGMENT

This research was supported by the DOE Office of Energy Efficiency and Renewable Energy (EERE), Building Technologies Office (BTO), under the guidance of Sven Mumme, and used resources at the Building Technologies Research and Integration Center (BTRIC), a DOE-EERE User Facility at Oak Ridge National Laboratory (ORNL). Research also sponsored by the Laboratory Directed Research and Development Program of ORNL, managed by UT-Battelle, LLC, for the U. S. Department of Energy.

### REFERENCES

- [1] K. Iturralde, M. Feucht *et al.*, “Cable-driven parallel robot for curtain wall module installation,” *Automation in Construction*, vol. 138, p. 104235, 2022.
- [2] N. W. Hayes, B. P. Maldonado *et al.*, “Optimization of prefabricated component installation using areal-time evaluator (rte) connection locating system,” in *Proceedings of the 35th International Symposium on Automation and Robotics in Construction (ISARC)*, 2024.
- [3] M. Tang, N. W. Hayes *et al.*, “Component pose reconstruction using a single robotic total station for panelized building envelopes,” Lille, France, pp. 105–112, June 2024.
- [4] Y. Liu, R. Zhang *et al.*, “Cable-driven parallel robot (cdpr) for panelized envelope retrofits: Feasible workspace analysis,” in *Proceedings of the International Symposium on Automation and Robotics in Construction*, vol. 41, 2024, pp. 1081–1088.
- [5] J.-B. Izard, M. Gouttefarde *et al.*, “Integration of a parallel cable-driven robot on an existing building façade,” in *Cable-driven parallel robots*. Springer, 2012, pp. 149–164.
- [6] N. Pedemonte, T. Rasheed *et al.*, “Fastkit: A mobile cable-driven parallel robot for logistics,” *Advances in robotics research: from lab to market: ECHORD++: robotic science supporting innovation*, pp. 141–163, 2020.
- [7] J. Lawrence, J. Bernal, and C. Witzgall, “A purely algebraic justification of the kabsch-umeyama algorithm,” *Journal of research of the National Institute of Standards and Technology*, vol. 124, p. 1, 2019.