Collaborative Block Stacking with Non-Prehensile Catching

Brendan Pousett[1], **Johan Ubbink**[1], **Wilm Decré**[1], **and Herman Bruyninckx**[1,2]

[1]Dept. of Mechanical Engineering, KU Leuven, and Flanders Make@KU Leuven, Leuven, Belgium
[2]Dept. of Mechanical Engineering, TU Eindhoven, Eindhoven, The Netherlands

{brendan.pousett, johan.ubbink, wilm.decre, herman.bruyninckx}@kuleuven.be

Abstract -

This paper presents a collaborative manipulation framework for automated stacking of large masonry blocks and other prefabricated elements on-site. Our approach utilizes a hoist for lifting the blocks and transporting them to the deposition site, combined with a ground-based manipulator for catching and precisely guiding the blocks into place. By observing skilled human teams, we focus on bimanual, non-prehensile manipulation as the key strategy for the catching and guidance task phases. We construct a dual arm robot suitable for contact-rich manipulation of high-inertia payloads, and develop an associated impedance controller for stabilizing block oscillations. We are able to consistently stack full-scale blocks weighing 150 kg, and demonstrate three distinct stacking modes (free, adjacent, atop) towards automated wall building. Furthermore, we demonstrate robustness to variance in hoist motion profiles with zero feedback and minimal coordination, highlighting the practicality of our system towards real applications. Videos of experiments are available at `https://youtu.be/k6tN2Kd9RBU`.

Keywords -

Construction robotics, On-site automation, Masonry, Non-prehensile manipulation

1 Introduction

In the near future, severe housing shortages combined with aging building infrastructure will require drastic productivity improvements in construction processes. Towards this goal, the industry is shifting to prefabricated components, which are produced in a dedicated facility and thus benefit from scale-enabled cost-reductions. Despite intensive research into off-site robotic production [1], on-site assembly remains a manual process [2]. Thus, automated on-site assembly represents a crucial step towards improving the productivity of the construction industry.

Concrete masonry units, hereafter referred to as "blocks", are an important building component for load-bearing superstructures in Western Europe. The manual assembly process is shown in Figure 1. Although simpler than other prefabricated components, the large block inertia and dimensions pose the same technical challenges as general automated on-site assembly. Blocks transported by cranes can oscillate due to wind, lifting dynamics, and other uncontrolled disturbances characteristic of the construction environment. These oscillations must be stabilized in the presence of dynamic obstacles and limited footholds. Human masons thus rely heavily on *non-prehensile manipulation* (applying forces without controlling all degrees of freedom) during the catching and guidance task phases.

Figure 1. The manual E150 block stacking process. A crane operator (left) loads the blocks from a pallet and controls transportation to the deposition site. The mason (right) prepares the site with mortar and precisely guides the block into its final location.

Motivated by these challenges, we present a solution for the automated stacking of large masonry elements. We use the popular Silka E150 as the reference element (dimensions of 0.897 m × 0.623 m × 0.150 m and mass of 151 kg) [3]. Inspired by Figure 1, we introduce an automated crane, referred to as the "hoist", and a dual-arm mobile manipulator, referred to as the "helper", to transport, catch, and guide blocks into pre-specified locations. Our system is shown in Figure 2. Sub-tasks such as mortar application, foundation preparation, and block placement planning remain out of scope.

1.1 Related Work

In contrast to our approach, several automated masonry solutions have already been developed (see [4] for a comparison). The Hadrian [5] and WLTR [6] represent two serial-arm masonry robots on the verge of market entry. Both systems demonstrate significantly faster building rates than humans. However, the maximum brick payloads currently supported by the Hadrian and WLTR are only 45 kg

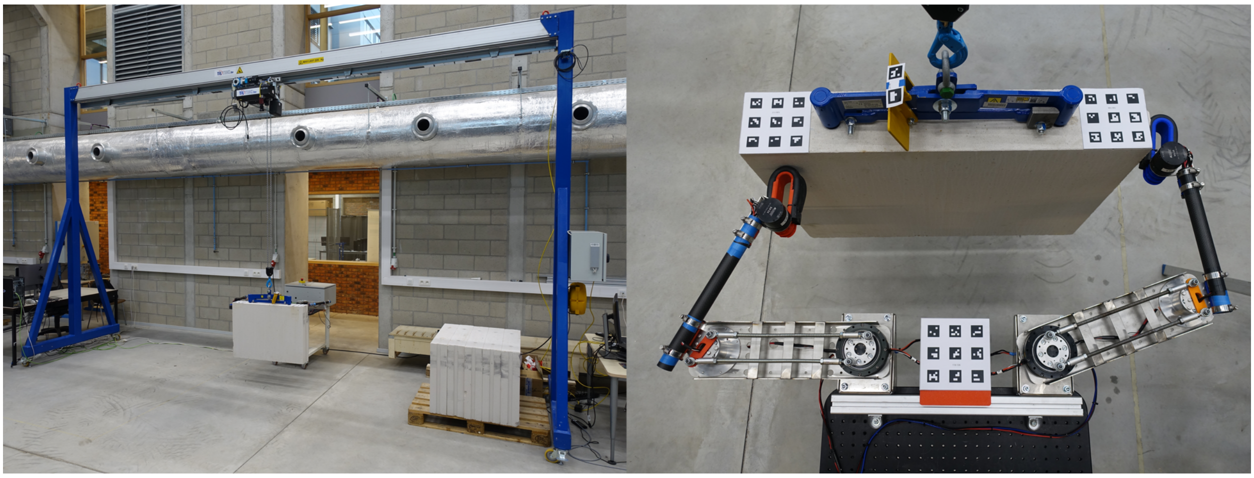

Figure 2. The automated "hoist-helper" system. Left: The hoist (blue/grey) consists of a linear gantry with both the translational and lifting axes possessing velocity control interfaces. The E150 blocks are shown on a pallet to the right and hanging from the block gripper in the middle. Right: The helper robot is a dual arm platform, where each arm possesses three degrees of freedom, shown with respect to an E150 for reference.

and 18 kg, respectively, less than 30% of the E150 mass. Furthermore, the WLTR must be transported between wall locations, whereas the Hadrian relies on a boom truck to increase workspace. Both platforms highlight the fundamental payload, stiffness, and workspace constraints inherent to serial-arm robot topologies, and the associated challenges in scaling to larger blocks exceeding 100 kg.

Conversely, cable driven parallel robots are scalable to wide ranges of payloads over a large workspace. Despite the benefits, these systems introduce complex motion planning problems to guarantee collision-free operation on crowded worksites [7]. Furthermore, platforms often require auxiliary structures on-site (e.g. towers, pulleys, anchors)—a significant economic barrier to adoption.

Heterogeneous multi-robot collaboration is a promising framework for on-site construction automation [2]. In this paradigm, complementary robotic platforms take specific roles, allowing each to address a subset of task requirements more effectively than a monolithic system. In the collaborative assembly context, [8] demonstrated coordinated building of extra-terrestrial superstructures with a crane and mobile robot. More recently, [9] proposed a similar architecture for block stacking, introducing an "explicit reference governor" control approach in [10]. However, initial experiments used 30 kg blocks, only 20% of the E150 mass. Furthermore, [10] assumed the pre-existence of a rigid grasp between the block and arm, neglecting the catching and stabilization task phases.

The catching of large, high inertia objects remains a salient omission from current work, given that blocks arrive at the stacking site with non-zero velocity. While many authors demonstrate crane control algorithms to remove payload oscillations (i.e. [11]), common cranes such as [12] lack the prerequisite high-bandwidth velocity control interfaces and instead rely on "on/off" interfaces with fixed speeds. [13] demonstrated bimanual robotic catching of swinging objects of 4.2 kg, but emphasis was on instantly forming prehensile grasps, leading to high dynamic forces. In contrast, human masons clearly demonstrate the value of non-prehensile stabilization during the catching and guidance of swinging blocks. Despite active research since [14], non-prehensile manipulation has largely been ignored in both heavy object catching and collaborative block stacking due to the complexity of planning these operations in real-time.

1.2 Contributions and Notation

In this work, we demonstrate how bimanual, contact-rich, non-prehensile manipulation can be incorporated into a collaborative and robust placement strategy for massive blocks and similar cable-constrained payloads. With respect to [9]-[10], we present a "hoist-helper" architecture with the following novel contributions: (1) experimental results stacking full-size E150 blocks in free space, adjacent to, and atop other blocks, (2) a helper-focused impedance control strategy compatible with "on/off" hoist velocity control interfaces with minimal coordination, and (3) demonstration of non-prehensile block catching for transitioning between the transport and guidance task phases.

Regarding mathematical notation, bold quantities ($\mathbf{a}$) denote vectors. The first subscript ($\mathbf{a}_z$) denotes a vector label, and the second subscript ($\mathbf{a}_{z,B}$) denotes the reference frame, which is omitted for the world frame. The hat operator ($\hat{\mathbf{a}}$) extends a 2D vector to 3D by appending 0. Lastly, a subscript $s \in \{x, y, z\}$ following a vector in parentheses ($(\mathbf{a})_s$) extracts the scalar s-component.

2 System Overview

2.1 Hoist Design and Control

The hoist design was not a primary consideration in our work, provided that it represented a common block stacking crane [12]. We consider a rotational axis at the base (fixed), and use a linear gantry to model a translational axis and lifting axis (vertical in the nominal configuration). The linear gantry has 6 m of travel and a height of 3.3 m, as shown in Figure 2. A laser distance sensor measures the trolley position along the translational axis, denoted x_O. The system is further equipped with an encoder to measure the active length of the cable ℓ. Both x_O and ℓ are adjusted via velocity control, which we restrict to an "on/off" control law to match the interface on small construction cranes. Given a goal position x_G, the control law is represented as:

$$\dot{x}_O = V(\mathbf{x}) = \begin{cases} \text{sign}(x_G - x_O)v & \text{if } |x_G - x_O| \geq \delta \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where v is a fixed scalar velocity, and δ is the positional deadband (0.02 m). An identical control law is formulated for $\dot{\ell}$. As noted in Section 1.1, the control interface renders swing compensation infeasible along the x-axis, and the controllability is rank-deficient relative to y-axis disturbances, necessitating collaboration with the helper.

The hoist is also equipped with a downward-facing camera; example frames are shown in Figure 5. The camera provides a spatial resolution of 1.37 mm/pixel when the image plane is located at $Z_c = 3.3$ m from the camera frame (the hoist height). The frames are processed at 20 Hz to provide marker pose information of the block and helper base relative to the hoist. The pose information is broadcast via MQTT over a local WiFi network.

2.2 Helper Design and Control

While the hoist specifications are derived from existing equipment, there is considerably more flexibility regarding the helper robot design. Let $\mathcal{P}$ be a plane parallel to the ground, coincident with the center of mass of the block. Given the natural decoupling between the hoist lifting axis and linear directions in $\mathcal{P}$, the helper was designed to compensate for disturbances in $\mathcal{P}$, as well as control the yaw angle about the vertical axis.

Our dual-arm helper robot is illustrated in Figure 2. Each arm has three co-axial revolute joints (shoulder, elbow, wrist) for end-effector positioning in a planar workspace with one degree of redundancy. The link dimensions (0.35 m, 0.3 m, 0.1 m) and mass (4.5 kg) are on the order of a human arm. We mount the arms on a wheeled cart and push them to different locations between stacking trials.

Design inspiration was taken from the legged robotics community, given the similarity of high-inertia, non-prehensile manipulation to footstep planning and loco-

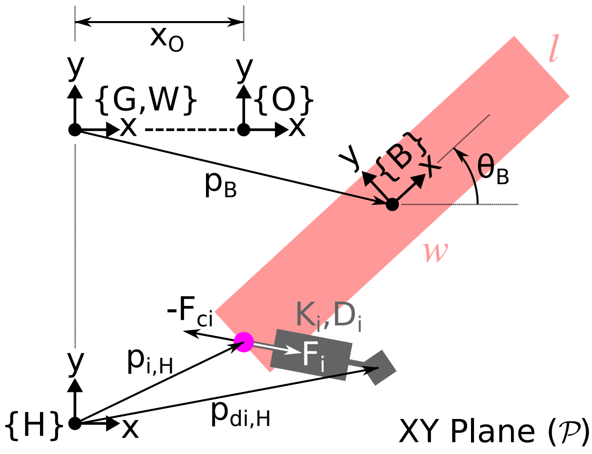

Figure 3. A top view of the system, showing the coordinate frames, including the hoist {O}, block {B}, world {W}, helper {H}, and goal {G} frames. An example helper end effector (pink) contacts a block (red). The spring-damper system (grey) represents the cartesian impedance control scheme, where setpoints are expressed in the helper frame.

motion control. Proprioceptive, low gear-ratio actuators mitigate impact severity and joint friction [15], and the actuators are placed at the base of the robot to minimize link inertia. Thus, both the acceleration and velocity-dependent backdrivability [16] of the system are optimized. While we don't present numerical metrics on the helper design, we refer interested readers to quadruped platforms with similar design intent [15].

Let $i \in \mathcal{I} = \{1, 2\}$ represent the set of end effector indices, where 1 = left and 2 = right. Each end-effector is represented by a point with state coordinates $\mathbf{x}_{i,H} = \begin{bmatrix}\mathbf{p} & \dot{\mathbf{p}}\end{bmatrix}_{i,H}$, expressed in the helper frame $\{H\}$, as shown in Figure 3. Each arm possesses a torque control interface, where torques are governed by a quasi-static cartesian impedance control law [17], as in:

$$\tau_i = \mathbf{J_i}^T \mathbf{F}_{i,H} = \mathbf{J_i}^T \left(\begin{bmatrix}\mathbf{K}_i & \mathbf{D}_i\end{bmatrix} (\mathbf{x}_{di,H} - \mathbf{x}_{i,H}) \right) \quad (2)$$

$\mathbf{K}_i, \mathbf{D}_i \in \mathbb{R}^{2\times2}$ are desired stiffness and damping matrices, respectively, $\mathbf{x}_{di,H}$ is the setpoint in the helper frame, and $\mathbf{J_i} \in \mathbb{R}^{3\times2}$ is the arm jacobian which maps joint velocities to linear end effector velocities.

The end effectors are covered with EPDM, which provides an extra layer of compliance to mitigate contact impulses during impact events. Furthermore, the EPDM/block combination possesses a high coefficient of static friction (μ_s) which prevents slipping, and yields contact forces close to the impedance force ($\mathbf{F}_{ci,H} \approx \mathbf{F}_{i,H}$). While detailed modelling of the contact forces is out of scope, development of a high-fidelity contact force model or sensing method is discussed in Section 5.

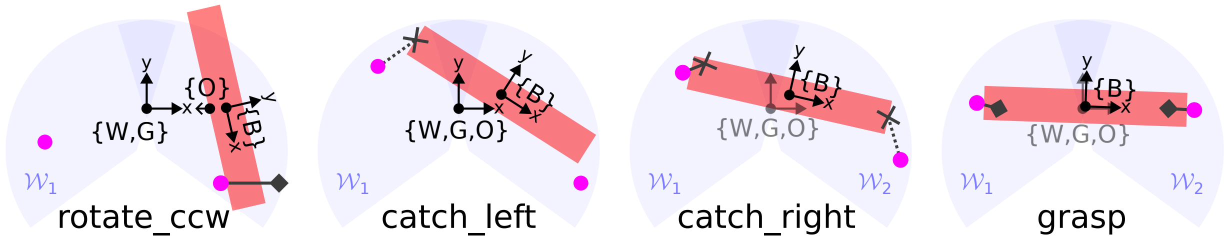

Figure 4. The control approach during a non-prehensile block catch. Each panel shows the setpoints (grey ◆ or ×), expected helper (pink circle) and block (red rectangle) positions, and frames for the indicated state. The states are ordered sequentially, such that time increases from left to right. The setpoints are constrained to a workspace (blue) such that $\mathbf{p}_{di} \in \mathcal{W}_i$. Setpoints expressed in the world frame $\{W\}$ are indicated with a ◆, whereas a × indicates expression in the block frame $\{B\}$. Solid lines connecting helper and setpoints indicate active contact, where dashed lines indicate motion towards the setpoint in free space via Bèzier spline interpolation.

3 Task Execution Strategy

3.1 State Estimation

We define a control approach for the helper which demonstrates the value of non-prehensile manipulation to the block stacking application. Figure 3 provides a helpful overview of the frame notation used in the problem. We use the following state representation of the block-hoist system:

$$\mathbf{x} = \begin{bmatrix}\mathbf{p}_B^T & \theta_B & x_O & \dot{\mathbf{p}}_B^T & \dot{\theta}_B\end{bmatrix}^T \tag{3}$$

By only considering the planar state of the center of mass ($\begin{bmatrix}\mathbf{p}_B^T & \dot{\mathbf{p}}_B^T\end{bmatrix}^T$) and the yaw angle $\begin{bmatrix}\theta_B & \dot{\theta}_B\end{bmatrix}^T$, we neglect any roll and pitch oscillations of the block and ignore their effect on the dynamics. The state vector is estimated via an Extended Kalman Filter (EKF) running at 200 Hz, which receives measurements of marker poses on the block and helper, as well as x_O and cable length measurements (ℓ). The EKF state transition model considers the block-hoist system as a 2D pendulum with a quasi-static base ($\ddot{x}_O = 0$, as per Equation 1), assuming that the hoist chain stays in tension and fixed length. We use a constant velocity model for the yaw angle. We note that these state transition models assume no interaction with the helper robot. Although simple, this state estimation was sufficient for successful task execution. Improvements are discussed in Section 5.

3.2 Non-Prehensile Block Stabilization

Using the impedance control law introduced in 2, the full stabilization problem can be stated as:

$$\begin{aligned}&\text{Find } \mathbf{x}_{d1}, \mathbf{x}_{d2} \text{ such that } \mathbf{x} \longrightarrow \mathbf{x}_G,\\ &\text{given } \dot{x}_O \text{ follows 1 and } \mathbf{x}(0) = \mathbf{x}_{init}\end{aligned} \tag{4}$$

Since non-prehensile interaction planning is a complicated problem for general $\mathbf{x}_{init}$ and $\mathbf{x}_G$, we restrict our problem to a single initial-goal state pair, given as:

$$\begin{aligned}\mathbf{x}_{init} &= \begin{bmatrix}x_{init} & 0 & \frac{-\pi}{2} & x_{init} & \mathbf{0}_{1\times3}\end{bmatrix}^T\\ \mathbf{x}_G &= \mathbf{0}_{7\times1}\end{aligned} \tag{5}$$

Note that a prehensile grasp cannot be formed at $\mathbf{x}_{init}$ due to the limited workspace of the helper and large size of the block. Thus, the stabilization controller must rotate the block by $\frac{\pi}{2}$ radians while removing any kinetic energy, and transition to a prehensile grasp about $\mathbf{x}_G$.

To derive appropriate $\mathbf{x}_{di}$ setpoints, we note that the block accelerations are expressed as a function of the contact forces by the planar rigid-body dynamics equations:

$$\begin{bmatrix}\ddot{\mathbf{p}}_B\\ \ddot{\theta}_B\end{bmatrix} = \begin{bmatrix}m^{-1}[\sum_i \mathbf{F}_{ci} + T(\mathbf{p}_O - \mathbf{p}_B)/\ell]\\ I_{zz}^{-1}\sum_i((\hat{\mathbf{p}}_i - \hat{\mathbf{p}}_B) \times \hat{\mathbf{F}}_{ci})_z\end{bmatrix} \tag{6}$$

where m is the block mass, I_{zz} is the block moment of inertia about $\{B\}$, and T is the hoist cable tension. Equation 6 demonstrates that applying a unit normal contact force to the bottom edge of the block ($\mathbf{p}_{1,B} = [x, -l/2]^T$; $\mathbf{F}_{c1,B} = [0, 1]^T$) results in an angular acceleration proportional to the offset from the center of mass ($\ddot{\theta}_B = I_{zz}^{-1}x$).

The prehensile grasp relies on achieving force closure through dual-arm contact. Valid contact forces can only push, and must not slip according to a Coulomb friction cone approximation, such that:

$$\mathcal{F}_c = \{\mathbf{F}_c \mid \mathbf{F}_c \cdot \mathbf{n}_c < 0 \text{ and } |(\mathbf{F}_{c,B})_y| \leq \mu_s(\mathbf{F}_{c,B})_x\} \tag{7}$$

where $\mathbf{n}_c$ is the outward normal vector at the contact location.

A prehensile grasp occurs when valid contact forces exist which produce zero block acceleration, such that for $i = 1, 2$:

$$\left\{\mathbf{F}_{ci} \in \mathcal{F}_c \;\middle|\; \begin{matrix}\sum_i \mathbf{F}_{ci} = T(\mathbf{p}_B - \mathbf{p}_O)/\ell\\ \sum_i((\hat{\mathbf{p}}_i - \hat{\mathbf{p}}_B) \times \hat{\mathbf{F}}_{ci})_z = 0\end{matrix}\right\} \neq \emptyset \tag{8}$$

We highlight the trivial prehensile grasp at $\mathbf{p}_B = \mathbf{p}_O$ and note that two antipodal contacts initiated on opposite ends of the block ($\mathbf{p}_{1,B} = [-w/2, 0]^T$; $\mathbf{p}_{2,B} = [w/2, 0]^T$) with valid contact forces can produce omnidirectional block accelerations according to equation 6.

The stabilization controller is formulated as a finite state machine with discrete states $s \in$ {rotate_ccw, catch_left, catch_right, grasp}. Each state cor-

Table 1. Setpoints of impedance controllers as executed in Figure 4 and Algorithm 1.

State (s)	$\mathbf{p}_{d1,\bullet}$ (Left)	$\mathbf{p}_{d2,\bullet}$ (Right)	Frame
rotate_ccw	—	$[x_{ws}, \frac{-w}{2} + \epsilon_1]$	{W}
catch_left	$[\frac{-w}{2}, 0]$	—	{B}
catch_right	$[\frac{-w}{2} + \epsilon_2, \epsilon_3]$	$[\frac{w}{2}, 0]$	{B}
grasp	$[\frac{-w}{2} + \epsilon_4, 0]$	$[\frac{w}{2} - \epsilon_4, 0]$	{W}

responds to setpoints of the cartesian impedance controllers of each arm, as defined in Table 1. The setpoints are defined with respect to the block width (w), and five additional "offset" parameters ($x_{ws}, \epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4$), used to produce valid contact forces and slow block rotations during the catch phase. The catch is illustrated in Figure 4.

The execution of the state machine is defined in Algorithm 1. The get_setpoints($s, \mathbf{x}$) function returns the world frame setpoints given the state s and block state $\mathbf{x}$. The helper initiates a non-prehensile push to start block rotation in the rotate_ccw state, before transitioning to the catch_left state when the left edge of the block comes in to the workspace. Upon making contact (checked with a kinematic observer in contact_detected($\mathbf{x}_i, \mathbf{x}$)), the helper transitions to a prehensile grasp via the catch_right and grasp states. We did not implement a failure state or a re-grasping state to be executed upon unsuccessful state transitions, but acknowledge that this would be valuable to increase the robustness of the system. A common observed failure mode occurred when the right arm failed to establish a stable grasp during the catch_right and grasp states, and is documented in the associated video. Furthermore, we note the future value in adding a visual servoing state after grasp to directly control the final positioning error ($||\delta \mathbf{p}_B||$), rather than relying on perfect relative alignment of the end effectors with the long axis of the block.

Setpoint and controller parameter tuning were performed both in simulation [18] and on the real setup by heuristic tuning based on observed failure modes. Example values are $x_{ws} = 0.6$ m, $\epsilon_1 = \epsilon_2 = \epsilon_3 = 0.1$ m, $\epsilon_4 = 0.08$ m, $\mathbf{K}_i = 60\mathbf{I}_{2\times2}$ N/m, and $\mathbf{D}_i = 10\mathbf{I}_{2\times2}$ Ns/m. We do not claim optimality of these values with respect to performance. However, we refer to Section 4 for a discussion on robustness of the catching skill with respect to variance in hoist speeds without parameter re-tuning.

Algorithm 1: The Block Stabilization Algorithm

```
x ← x_init  s ← rotate_ccw;
while s ≠ done do
    x, x_1, x_2 ← perception();
    switch s do
        case rotate_ccw do
            p_d1, p_d2 = get_setpoints(catch_left, x);
            if p_d1 ∈ W_1 then
                s ← catch_left;
        case catch_left do
            if contact_detected(x_1, x) then
                s ← catch_right;
        case catch_right do
            if contact_detected(x_2, x) then
                s ← grasp;
        case grasp do
            if (x − x_G)^2 ≤ res. then
                s ← done;
        otherwise do
            raise(error);
        end
    end
    p_d1, p_d2 ← get_setpoints(s, x);
    x_d1, x_d2 ← interpolate(s, p_d1, p_d2, x_1, x_2, x);
    τ_1 ← J^T [K_1  D_1] (x_d1,H − x_1,H);
    τ_2 ← J^T [K_2  D_2] (x_d2,H − x_2,H);
end
```

Table 2. Success rate, mean and standard deviation of final position error, across different hoist speeds.

v (m/s)	Trials	Success	$\mu_{\|\|\delta\mathbf{p}\|\|}$ (m)	$\sigma_{\|\|\delta\mathbf{p}\|\|}$ (m)
0.1375	10	10	0.048	0.008
0.1500	5	5	-	-
0.1625	5	4	-	-
0.1750	5	4	-	-
Total	**25**	**23**	—	—

4 Results

With respect to [9]-[10], we present the first results robustly catching and stabilizing full-scale masonry elements through hoist-helper collaboration. We demonstrate three distinct stacking modes which constitute useful primitives for complete wall building in Figure 5: free-space, adjacent, and atop stacking. Free space stabilization runs the controller described in Algorithm 1 for the initial and goal conditions in Equation 5. To demonstrate the accuracy and repeatability of this skill, we conducted 10 trials for a linear hoist speed setpoint of $v = 0.1375$ m/s and initial transport distance varying from $x_{init} = 1.5$ m to $x_{init} = 2.5$ m. The overlaid state data from the trials is plotted in Figure 6 and summarized in Table 2. While insufficiently accurate for wall building, the mean final positioning error of 4.8 cm is encouraging as an initial result. In addition to adding an aforementioned visual servoing capability, this error could be reduced further via contact exploitation. Human masons frequently use collisions with adjacent blocks to align the swinging element and provide a guide while dropping the block into place. We demonstrate this contact exploitation in Figure 5, row (c), where the tongue-in-groove feature facilitates precise relative alignment of the blocks.

In the adjacent stacking scenario, the free-space stabi-

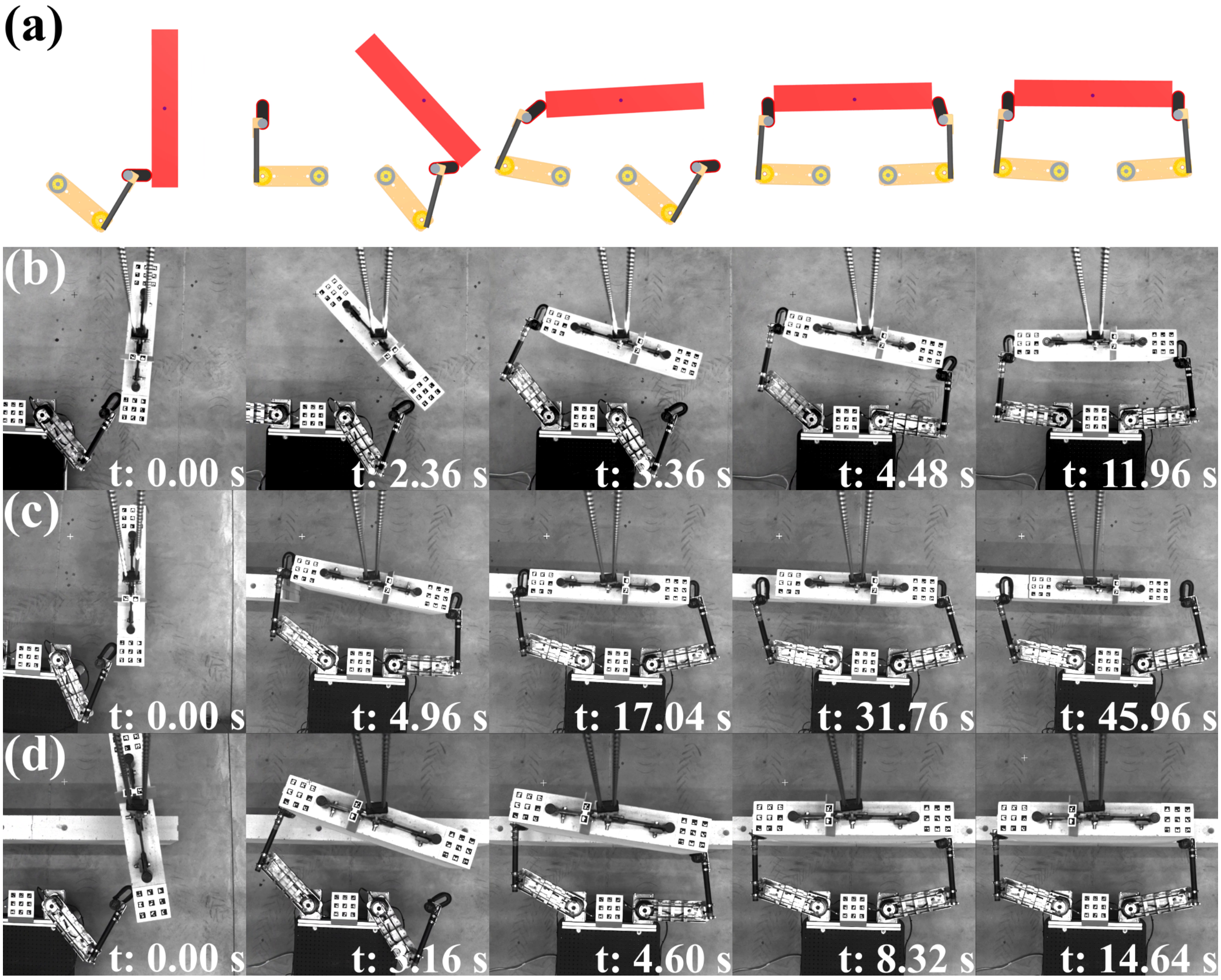

Figure 5. Image frames from simulation (a) and the hoist perception system (b-d) during stacking experiments. The execution times are included in the insets for the real experiments. The non-prehensile catching and stabilization of block oscillations is demonstrated in rows (a, b). Row (c) demonstrates a catch plus contact-based alignment with an adjacent block with a tongue-in-groove feature—a common action during automated wall building. The slower execution speed is due to manual coordination of the alignment and dropping phases. Row (d) demonstrates a catch and drop atop an existing wall, where a non-prehensile grasp is maintained during the dropping phase to facilitate alignment with the wall edges. The full video of all actions is included via the link in the abstract.

lization algorithm is executed to a goal state offset along the x-axis from a stationary block on the ground. The helper pushes the suspended block until contact with the stationary block and alignment of the tongue-in-groove feature, before releasing the block completely during the drop phase. While a useful demonstration, the tongue-in-groove alignment robustness was limited by open loop execution. We refer to Section 5 for a discussion of perception improvements to facilitate robust relative alignment.

In the atop stacking mode, the helper runs the same free-space stabilization algorithm. In contrast to the first two modes, the helper maintains a prehensile grasp during the block dropping phase as the end effectors remain within the block height bounds.

The most fundamental development to the stacking modes in Figure 5 is the catch transition from a non-prehensile to a prehensile grasp. A prominent feature of our catching approach is the lack of coordination with the hoist. Although the on/off velocity profile excites block oscillations along the x-axis, it is representative of common crane control interfaces, and very easy to reason about from a stabilization perspective. To demonstrate robustness to slight variations in the hoist velocity, we tested the stabilization action over varying hoist speeds without parameter re-tuning. As summarized in Table 2, we report a catch rate of 92% across 25 trials with speeds between $v = 0.1375$ m/s and $v = 0.175$ m/s. We refer to Section 5 for a more thorough discussion about robustness.

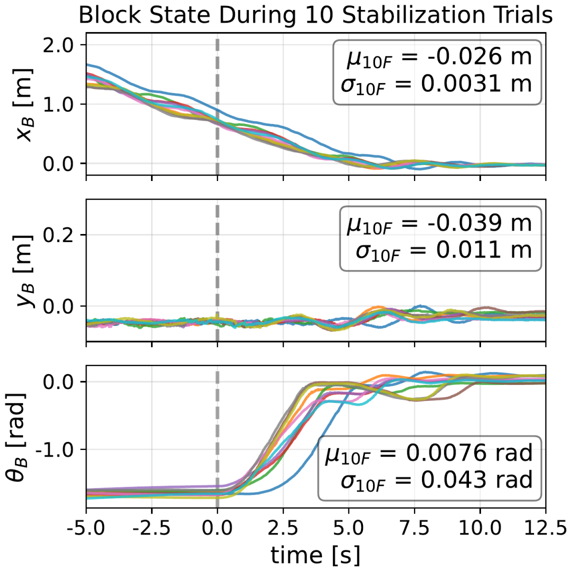

Figure 6. Block state values ($[\mathbf{p}_B^T, \theta_B]$) for 10 stabilization trials when $v = 0.1375$ m/s. The moment of initial contact between the helper and block is set to $t = 0$ s. The mean and standard deviation of the final data point across all 10 trials is given in the inset.

5 Conclusion and Future Work

The results demonstrate the value of bimanual, contact-rich, non-prehensile interactions to collaborative robotic manipulation of heavy constrained payloads. Specifically, the general strategy of simultaneous kinetic energy removal and object re-orientation has many applications in construction industry, where large and irregularly shaped prefabricated components are assembled on-site. Such applications exceed the payload and workspace capabilities of serial robot manipulators and necessitate a hoist-helper approach. While we encourage further hoist development, we advocate for drastic hardware, control, and state estimation improvements on the helper subsystem.

Regarding hardware, we advocate for mobile, dual-arm manipulation platforms which are lightweight, backdrivable, and have moderate payload specifications. A larger workspace would also enable easier planning of collision-free motions between contact points, and increase the number of reachable contacts. Grippers could be redesigned to facilitate a richer set of configurable grasping modes, such as switching from single-face pushing to corner grasping. Human masons frequently make these transitions to gradually transition to more prehensile grasping modes.

Regarding control, the main limitations are generalization and robustness. We did not address contact-rich planning in this paper by only considering a single initial-goal state pair, thus requiring one sequence of setpoints to stabilize the block. One method to generalize our approach would be to formulate multiple state machine controllers with different contact sequences and characterize regions of attraction for each one, where robust convergence to the goal state is guaranteed. This approach is similar to [19], where a planner selected the appropriate controller at runtime based on observed state. However, the high dimensionality of the state space and numerous potential goal states would make sufficient coverage difficult. Furthermore, unmodelled disturbances (such as wind) make robust characterization of regions of attraction challenging.

Alternative approaches to generalization include simultaneously planning the contact sequences and forces online, reformulating Equation 4 as a constrained optimization problem. This problem is similar to locomotion, where model-predictive control [20] and reinforcement learning [21] have furnished robust and performant controllers.

We propose state estimation improvements, both to the state transition model and sensing modalities. As noted in Section 3.1, the dynamic model of the block-hoist system does not include contact interactions. Contact forces could be estimated either through the combination of helper system identification and a momentum observer, or via direct tactile sensing. Furthermore, a real system would need to estimate block poses without markers. While the hoist camera provided a useful bird's-eye view of the worksite, we observed linear and angular deflections of the mounting location during block transport. Given the height of 3.3 m, the perception has a sensitivity of 5.8 cm linear error per degree of deflection. Therefore, we advocate for moving block perception on-board the helper to reduce latency (by removing the MQTT broker) and improve accuracy. The block size represents useful and reliable prior knowledge to be exploited given the addition of depth information. Additionally, fiducials placed on the hoist trolley and block gripper could facilitate accurate and easy tracking from the helper, as the gripper pose relative to the block is reliable and fixed. Due to the potential improvements to the perception system, we refrained from a detailed analysis and optimization of our estimator covariances in this prototype.

Lastly, we address the safety of the proposed approach. As we automate an existing building process, we transfer the inherent risks (falling, collisions with payload) from the human mason to the helper robot. However, additional safety concerns are introduced to other workers on the construction site. During block transport, there is no natural emergency stop mechanism, as stopping transport may increase swinging and dropping the block may be unsafe to workers. Thus, we must be sufficiently confident that the system will execute the task safely before the block is picked up. In other words, the automated system must be at least as robust as human masons, able to dynamically plan safe mitigation strategies (i.e. moving a grasped block to free space), or determine if the task is unsafe to begin with. The robustness also encompasses the final stack quality, as falling blocks pose risks to workers. Although, these barriers are significant given the high uncertainty and many disturbances present on construction sites, we note the tight coupling between safety and robustness and encourage further development towards a reliable system.

This work was funded by Flanders Make as a CTO action. Thank you to Erard de Laminne, Raphaël Furnémont, Stijn Helsen, and Taner Incirci for support with the hoist, vision system, and communication architecture.

References

[1] Mi Pan, Y. Yang, Z. Zheng, and W. Pan. Artificial Intelligence and Robotics for Prefabricated and Modular Construction: A Systematic Literature Review. *Journal of Construction Engineering and Management*, 2022. doi:10.1061/(ASCE)CO.1943-7862.0002324.

[2] Nathan Melenbrink, Justin Werfel, and Achim Menges. On-site autonomous construction robots: Towards unsupervised building. *Automation in Construction*, 2020. doi:10.1016/j.autcon.2020.103312.

[3] Silka Calcium Silicate Elements, . URL `https://www.xella.co.uk/en_GB/Silka`.

[4] Eric Brehm. Robots for masonry construction – Status quo and thoughts for the German market. *Mauerwerk*, 2019. doi:10.1002/dama.201900004.

[5] Hadrian® | Automated Construction Robot from FBR, . URL `https://www.fbr.com.au/view/hadrian`.

[6] WLTR Masonry Robot – Precision Bricklaying and Construction Efficiency, . URL `https://www.green.build/`.

[7] Tobias Bruckmann and Roland Boumann. Simulation and optimization of automated masonry construction using cable robots. *Advanced Engineering Informatics*, 2021. doi:10.1016/j.aei.2021.101388.

[8] B. Sellner et al. Coordinated multiagent teams and sliding autonomy for large-scale assembly. *Proceedings of the IEEE*, 2006. doi:10.1109/JPROC.2006.876966.

[9] M. Ambrosino, F. Boucher, M. Pierre, and E. Garone. Full-Scale Prototype for Bricklaying Activity. In *Proceedings of the 40th ISARC*, 2023. doi:10.22260/ISARC2023/0013.

[10] M. Ambrosino, F. Boucher, P. Mengeot, and E. Garone. A trajectory-based explicit reference governor for the laying activity with heavy prefabricated elements. *Construction Robotics*, 2023. doi:10.1007/s41693-023-00091-w.

[11] Milan Vukov et al. Experimental validation of nonlinear MPC on an overhead crane using automatic code generation. In *2012 American Control Conference (ACC)*, 2012. doi:10.1109/ACC.2012.6315390.

[12] Böcker MK 300 / 400 - The powerful and efficient mini crane, . URL `https://boecker.de/en/products/mini-cranes/details/produkt-2`.

[13] Lei Yan et al. Impact-aware bimanual catching of large-momentum objects. *IEEE Transactions on Robotics*, 2024. doi:10.1109/TRO.2024.3381551.

[14] Matthew T. Mason. Mechanics and planning of manipulator pushing operations. *The International Journal of Robotics Research*, 1986. doi:10.1177/027836498600500303.

[15] Patrick M. Wensing et al. Proprioceptive actuator design in the mit cheetah: Impact mitigation and high-bandwidth physical interaction for dynamic legged robots. *IEEE Transactions on Robotics*, 33(3), 2017. doi:10.1109/TRO.2016.2640183.

[16] W. T. Townsend and J. K. Salisbury. Mechanical Design for Whole-Arm Manipulation. In *Robots and Biological Systems*, Berlin, Heidelberg, 1993. Springer. doi:10.1007/978-3-642-58069-7_9.

[17] Christian Ott. *Cartesian Impedance Control of Redundant and Flexible-Joint Robots*, volume 49 of *Springer Tracts in Advanced Robotics*. Springer, 2008. doi:10.1007/978-3-540-69255-3.

[18] Russ Tedrake and the Drake Development Team. Drake: Model-based design and verification for robotics, 2019. URL `https://drake.mit.edu`.

[19] R. R. Burridge, A. A. Rizzi, and D. E. Koditschek. Sequential Composition of Dynamically Dexterous Robot Behaviors. *The International Journal of Robotics Research*, 18(6), 1999. doi:10.1177/02783649922066385.

[20] Gijeong Kim et al. Contact-Implicit Model Predictive Control: Controlling Diverse Quadruped Motions Without Pre-Planned Contact Modes or Trajectories. *The International Journal of Robotics Research*, 44 (3), 2025. doi:10.1177/02783649241273645.

[21] Takahiro Miki et al. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, 7(62), 2022. doi:10.1126/scirobotics.abk2822.