Automation in Construction 125 (2021) 103569

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

Robotic assembly of timber joints using reinforcement learning

Aleksandra Anna Apolinarska [a,*,1], Matteo Pacher [a,1], Hui Li [b,1], Nicholas Cote [b], Rafael Pastrana [a], Fabio Gramazio [a], Matthias Kohler [a]

[a] Gramazio Kohler Research, ETH, Zürich, Switzerland
[b] Autodesk Research, Autodesk Inc., San Francisco, United States of America

ARTICLE INFO

Keywords:
Robotic assembly
Reinforcement learning
Timber construction
Timber joints

ABSTRACT

In architectural construction, automated robotic assembly is challenging due to occurring tolerances, small series production and complex contact situations, especially in assembly of elements with form-closure such as timber structures with integral joints. This paper proposes to apply Reinforcement Learning to control robot movements in contact-rich and tolerance-prone assembly tasks and presents the first successful demonstration of this approach in the context of architectural construction. Exemplified by assembly of lap joints for custom timber frames, robot movements are guided by force/torque and pose data to insert a timber element in its mating counterpart(s). Using an adapted Ape-X DDPG algorithm, the control policy is trained entirely in simulation and successfully deployed in reality. The experiments show the policy can also generalize to situations in real world not seen in training, such as tolerances and shape variations. This caters to uncertainties occurring in construction processes and facilitates fabrication of differentiated, customized designs.

1. Introduction

1.1. Automated assembly in architectural construction

While automated assembly using robotic systems is commonplace in many manufacturing industries, it is still in its infancy in architectural construction. One of the reasons is that buildings are usually bespoke and the development of automation processes for each new design often requires substantial investment. However, automated assembly in architectural construction has been gaining momentum in recent years, especially in bricklaying [1–4] and timber prefabrication [5,6]. In the latter, matured CAD-CAM workflows and CNC technologies for parts manufacture provide favourable conditions for a full-stack digital chain, from design through fabrication of parts to assembly. So far, however, industrially available solutions for automated assembly in timber construction, such as automated framing stations [6,7], are scarce and focus on narrowly standardized and usually butt-jointed modular construction, which makes only limited use of the capabilities of the robotic and CNC tools. By contrast, research by [8–10] has shown that automated assembly can facilitate more resource-aware production and handle highly differentiated designs. The learning control methods presented in this work can expand assembly processes to highly varied, non-standard structures, including those with integral timber-to-timber joints (Fig. 1).

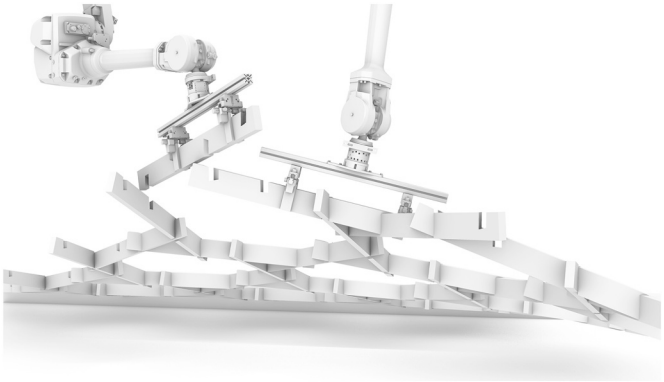

Fig. 1. Vision: robotic assembly of a free-form timber frame structure with lap joints.

* Corresponding author.
E-mail address: aapolina@ethz.ch (A.A. Apolinarska).
[1] Equal contribution

https://doi.org/10.1016/j.autcon.2021.103569
Received 27 May 2020; Received in revised form 4 January 2021; Accepted 18 January 2021
Available online 27 February 2021

1.2. Assembly of timber joints

Historic timber frame houses and structures were constructed using elaborate carpentry joints, which were laborious to cut and assemble. Modern European timber frame construction evolved to use predominantly simple butt joints and metal fasteners, but a certain amount of joints still entails form closure, e.g. to transfer shear force, for which a tight-fit contact between the parts is required. The mating notches are cut to 0 mm accuracy and usually assembled by hand; the force needed to insert such tight-fitting parts is exerted using a hammer or a clamp. The edges of the notch are often chamfered, or the sides tapered, to facilitate initiating the insertion. When multiple connections need to be joined simultaneously (multiple engagement), even small misalignment or lack of synchronization may lead to jamming and stick-slip effect. Tapered joints can alleviate this, but are often unfavourable structurally and can disengage more easily. Both the contact forces and accuracy form a challenging control problem for automated assembly.

1.3. Robotic control for assembly tasks

Assembly tasks are contact-rich and require interaction between the robot and its environment, which is often hard to predict [11]. In manufacturing industries, one of the classic approaches applied for repeatable and high precision tasks is to program robot's key configurations manually with a teach pendant, relying on robots' high repeatability. Methods that exploit passive mechanical compliance and insertion guiding features to overcome positional uncertainties can provide faster and more reliable responses for dynamic collision than sensors, but are usually engineered to a very specific task. Direct control methods for constrained motion control include active mechanical compliance and force control algorithms [11]. Most of these methods are based on explicit procedures, engineered to perform a specific task and require precise models of contact and friction dynamics [12]. In contrast to the aforementioned approaches, to handle products or environments with considerable uncertainty or variation, advanced force control methods are needed, such as adaptive or learning methods [13,14].

1.4. Reinforcement Learning for robotic assembly tasks

Reinforcement Learning (RL) is an area of machine learning in which agents generate actions within an environment based on observations and learn an optimal policy to maximize the expected total reward. With RL, a robotic system can autonomously acquire policies to manage contact-rich manipulation without prior knowledge of the environment or the dynamics model. Since these policies can generalize well to new scenarios and in reaction to real-time observations, recently [15–21] RL has become a popular technique [22–25] for control policies for robotic assembly tasks.

1.5. About this work

This paper presents – to our knowledge the first successful – application of Reinforcement Learning for assembly tasks using large industrial robots in the context of architectural construction. In particular, we demonstrate an autonomous execution of assembly tasks while being able to simultaneously adapt to uncertainties such as inaccuracies and geometric variance. To control the robot, a RL policy is trained entirely in simulation, based on force/torque (FT) and pose observations only. This challenge is explained in Section 2 and followed by an overview of related work in Section 3. The presented solution (see Section 4) uses a distributed DDPG algorithm [16], leverages human demonstrations [26], and applies domain randomization [21,27] to bridge the gap between simulation and reality. We test this approach in a series of physical experiments exemplified by lap-joint insertion tasks (Section 5), from which the conclusions are drawn in Section 6. Implementation details are provided in the Appendix.

2. Problem statement

In this paper, we look at the final stage of an assembly routine – the insertion task – which is particularly challenging because of multiple inaccuracies and uncertainties that can occur and cannot be explicitly planned for. The preceding steps, including picking up the timber piece and bringing it close to its final location in a structure, are done as in [9] and are not discussed here further. The aforementioned uncertainties may stem from allowances in size and shape of the timber element, displacement of the already built structure or an inaccuracy of the robot that can amount to several millimetres, especially in large-scale robotic systems [28], and together they may result in a misalignment of the mating notches and hinder the insertion of the element. To solve this problem, we aim to learn a motion control policy using Reinforcement Learning. The controller should react to the contact forces and produce a smooth insertion movement without damaging the wooden elements. We intend to guide it in real-time using FT and pose feedback and an RL policy. Addressing non-standard, mass-customized structures, the controller should also be able to handle a variety of different element shapes and poses.

Since it is impractical, dangerous, costly and time-consuming to conduct sufficient training with real industrial robots and physical materials, we intend to train the policy purely in simulation. Physics engines are often used to simulate robotic manipulation tasks, negotiating different trade-offs between speed and accuracy [29]. To deploy that policy in reality, we need to bridge the gap between simulation and reality. Physical phenomena, such as friction or deformation, contribute substantially to contact forces during assembly and are very difficult to simulate. On the other hand, sensor noise and control delays do not occur in simulation but add to the perturbations an agent may experience in reality.

In this paper, our goal is to learn a motion-control policy that will output Cartesian velocities $[v_x, v_y, v_z, \omega_x, \omega_y, \omega_z]$ given FT observations $[f_x, f_y, f_z, t_x, t_y, t_z]$ and pose observations $[p_x, p_y, p_z, q_w, q_x, q_y, q_z]$ as input, and to evaluate the applicability of the policy to control an industrial robot for the assembly of timber frame structures.

3. Related work

3.1. Robotic assembly

In architectural context, notable examples of robotic assembly at real scale include non-standard timber frames [9] and trusses [30,31], however, in these projects connections are designed to accommodate inaccuracies and avoid concern for contact situations. In contrast, high accuracy and insertion force is indispensable to assemble timber plate structures with multiple through-tenons, for which [32] adds vibration at the end-effector.

On a smaller scale of desktop and collaborative robots, past research presented methods for autonomous acquisition of control policies for contact-rich assembly tasks such as peg-in-a-hole, toys or furniture. In [33], multiple linear-Gaussian controllers are used to train a non-linear policy, exemplified by assembly of various toys with a PR2 robot. To assemble an IKEA chair, [34] applies hybrid position/force control to insert dowels into holes or a part in multiple dowels.

3.2. Reinforcement Llearning for robotic assembly

Among various established RL methods, DDPG [15] is a model-free algorithm developed for problems with a low-dimensional observation space and a continuous action space, as needed for robotic control. Its follow-up Ape-X DDPG [16] is considered well suited to tackle complex assembly tasks thanks to a distributed architecture and a much higher

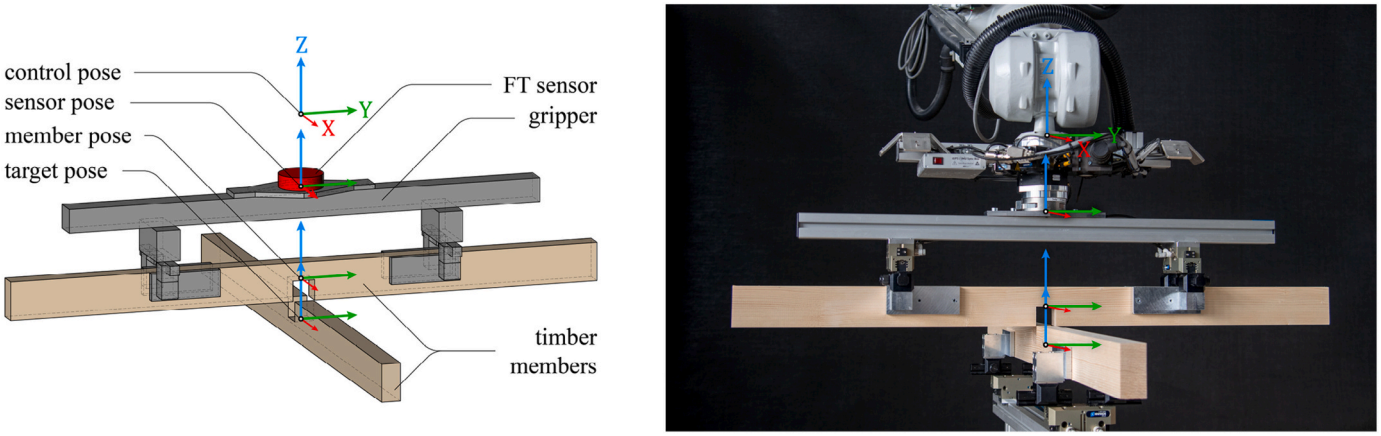

Fig. 2. Visualization of *control*, *sensor*, *member* and *target* poses with reference to the gripper, FT sensor and the timber members in simulation (left) and in reality (right). In a double-lap joint, the *member* and the *target* pose are at the average position of the notches. In the default configuration, all poses' *Z*-axes and origins are colinear.

training efficiency. Set to solve similar assembly tasks and also based on Ape-X DDPG, [23] focuses on a novel technique that allows RL algorithms to use experience replay samples from successful transitions generated by RL agents during training. In this paper, we focus on the simulation-to-reality transfer of the learned policy to solve timber joint assembly tasks and on extensive experimentation on industrial robots with varying geometries and conditions in the context of architectural construction.

Other notable RL-based approaches include, for example, a recurrent neural network with RL for peg-in-a-hole task using FT and pose observations [25]. To solve high-precision gear assembly tasks, [22] proposed model-based RL and operational-space force control. In [12], instead of using random exploration, geometric information in Computer-Aided Design (CAD) models is used to guide RL to solve various insertion tasks.

In terms of the choice of the action space, many approaches to robotic motion control involving RL rely on forward kinematics [12,22,24,26,35] and describe actions in joint space. Because our actions require a small volume, we can instead focus on Cartesian-space poses and learn a policy that is robot-agnostic.

3.3. Bridging simulation and reality

Research on bridging the gap between simulation and reality mainly focuses on two techniques. The first technique is incorporating experience from the *real*, or physical, world [36,37]. The other is Domain Randomization, with which discrepancies between the source and target domains are modeled as variability in the source domain. In dynamics, randomization has been used to develop controllers that are robust to uncertainty [38] and to transfer manipulation policies by randomizing physical parameters [21,39].

4. Approach

4.1. Setup

We simplify the overall assembly problem and focus only on the movements at the end-effector. We disregard the kinematic chain of the robot and assume that a valid joint configuration exists at all times since we focus on relatively small movements. Conveniently, we move at low speeds and may also ignore the dynamics of the system.

For simulation, we use a physics engine and we only consider objects that contribute to FT readings, i.e. a model of the gripper, of the sensor and of the timber members. We provide estimated inertial properties (mass and centre of mass) of all dynamic objects and include friction in the simulation. In reality, the robotic workcell is a 6-axis industrial robot with a custom-made FT sensor, an end-effector and a robot controller (see Appendix for details). In both simulation and reality, the FT sensor is mounted before the gripper, further referred to as *sensor pose*. In the following, we distinguish it from the *control pose*, *member pose* and *target pose* (Fig. 2).

4.2. Observations

The observation space is continuous, 13-dimensional, and consists of the position $[p_x, p_y, p_z]$ and orientation $[q_w, q_x, q_y, q_z]$ at the member pose as acquired from the robot and the contact force $[f_x, f_y, f_z]$ and contact torque $[t_x, t_y, t_z]$ at the sensor pose as acquired from the sensor.

4.3. Actions

The action space is continuous, 6-dimensional, and consists of the predicted linear velocity $[v_x, v_y, v_z]$ and angular velocity $[\omega_x, \omega_y, \omega_z]$ for each timestep at the control pose as acquired from the policy.

4.4. Human demonstration

We record one human demonstration of a given task conducted exclusively within the simulation environment, using a game controller to drive the end-effector [23] until the timber member and the timber assembly are successfully joined. The demonstrator is aware and in control of linear and angular velocities applied to the end-effector, but has no haptic feedback of contact forces during the recording of assembly.

4.5. Simulation-to-reality transfer

Within the simulation environment, we set gravity acceleration to 0 m/s^2 to isolate contact force and torque. In the real environment, gravity, sensor bias and drift are compensated using an approach similar to [40]. To reconcile FT sensing in simulation [29] with that in reality, we also calibrate the FT range in simulation and reality by performing the same task, and scale the FT observations by a computed factor.

To better reproduce the noise present in FT observations in reality, we apply domain randomization in the form of Gaussian noise to FT readings in simulation. Specifically, we apply correlated noise, sampled once per episode, and uncorrelated noise, sampled once per timestep (see Appendix for details). We experimented with adding randomization to pose observations and friction coefficients in simulation, however, results show no visible improvement in learning transfer.

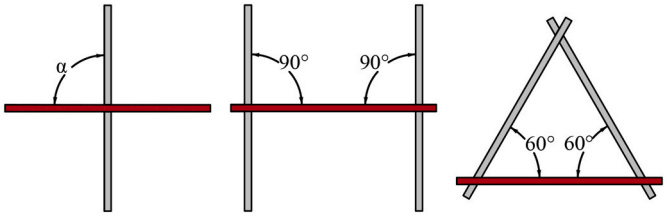

Fig. 3. Left to right: X-shaped single-lap, H- and Δ-shaped double-lap.

System delays in the real world are a critical factor in the simulation-to-reality gap [35]. Instead of adding randomization to the communication rate in simulation, we characterize the control delay profile of our physical robot [41], and use the maximum communication frequency that does not trigger the control delay on the real setup.

4.6. Algorithm

We use the Ape-X DDPG algorithm [16] and incorporate human demonstrations in its distributed architecture. In [16] the *actors* interact with their own instances of the simulation environment and accumulate the resulting experience in a shared experience replay memory, whereas the *learner* replays experience samples and updates the shared neural network, from which actors draw actions.

We record all transitions from one successful human demonstration. One transition is defined as $e_t = (s_t, a_t, r_t, s_{t+1})$, where s_t, a_t and r_t are observation (state), action and reward at time t, respectively. We use a linear reward function based on the linear and angular distance between the member pose and the target pose, adding a large positive reward (+100) if that distance is less than or equal to a distance threshold. The reward given at timestep t, defined as

$$r_t = \begin{cases} -|g - x_t|, & |g - x_t| > \varepsilon \\ -|g - x_t| + R, & |g - x_t| \leq \varepsilon \end{cases},$$

where x_t is the member pose at time t, g is the target pose, ε is a distance threshold, and R is the large positive reward, is added to the total reward of each episode. The negative distance at each timestep encourages the policy to reach the goal in fewer steps. No penalties for exceeding readable sensor range are added to the reward function.

The training consists of multiple iterations. In each training iteration, the agent makes several attempts (episodes) to complete the task. Training concludes when the average success rate of all the episodes within a training iteration reaches a predefined threshold or when the total number of training iterations is exceeded. An episode is considered successful when $|g - x| \leq \varepsilon$. In simulation, if this goal has not been reached within a user-defined number of steps, the episode is regarded as a failure. In reality, the number of steps is not limited, but the execution is aborted and the episode considered unsuccessful when the sensor or the robot's wrist torque limits are violated.

5. Experiments

5.1. Method

To test our approach, we deploy (roll out) the trained policy in reality on a robotic workcell and conduct a series of assembly tasks. We evaluate the robustness of the trained policy with respect to variations common in timber construction. Specifically, we vary the geometrical configurations, joint tightness, and the initial linear and angular offsets of each task. We also deploy the policy in simulation and compare these rollouts with those in reality to assess how successful the policy is at transferring from simulation to reality.

We focus on half-lap joints as a toy example for a simple and common joint in woodworking, in which two or more timber members overlap each other by half their depth at a mating notch. In our experiments we look at two scenarios: 1) two members with one joint (*single-lap*) and 2) three members with two joints (*double-lap*), as shown in Fig. 3. For each of the two scenarios, we train a single policy in simulation and then test that policy in reality, with varying task parameters to see if the policy is able to generalize. These parameters are: angle α between connecting timber elements, tolerance b between mating faces of the notch, and x_0 describing the offset applied to the initial pose of the timber member. The default values of the tasks parameters used in the training are $\alpha = 90°$, $b = 1$ mm and $x_0 = \mathbf{0}$. The relative position of the gripper to the

Task	α [°]	b [mm]	Linear offsets [x,y,z] in [mm]	Angular offsets [x,y,z] in [°]	Sim rollout success rate	Real rollout success rate
SINGLE	90°	0.5	0, 0, 0	0, 0, 0	9/10	0/3
		1	0, 0, 0	0, 0, 0	10/10	7/7
			2, 2, 0	0, 0, 0	10/10	3/3
			0, 0, 3	0, 0, 0	10/10	3/3
			0, 0, 5	0, 0, 0	10/10	3/3
			0, 0, 10	0, 0, 0	10/10	0/3
			0, 0, 0	0, 0, 1	9/10	5/5
			0, 0, 0	0, 0, 2	10/10	6/6
			0, 0, 0	0, 0, 3	4/10	0/3
			0, 0, 0	0, 0, 4	4/10	0/3
			0, 0, 0	0.5, 0.5, 0.5	5/10	3/3
			0, 0, 0	1, 1, 1	0/10	3/3
			0, 0, 0	2, 2, 2	0/10	2/3
			0, 0, 2	2, 2, 2	0/10	3/3
			1, 1, 1	2, 2, 2	0/10	0/3
			2, 2, 2	2, 2, 2	0/10	0/3
		2	0, 0, 0	0, 0, 0	10/10	15/15
	75°	1	0, 0, 0	0, 0, 0	10/10	6/7
	60°	1	0, 0, 0	0, 0, 0	10/10	2/3
	45°	1	0, 0, 0	0, 0, 0	2/10	0/3
DOUBLE	H 90°	1	0, 0, 0	0, 0, 0	6/10	10/10
	Δ 60°	1	0, 0, 0	0, 0, 0	5/10	0/21

Fig. 4. Overview of experiment parameters and results. The parameters used for training are highlighted.

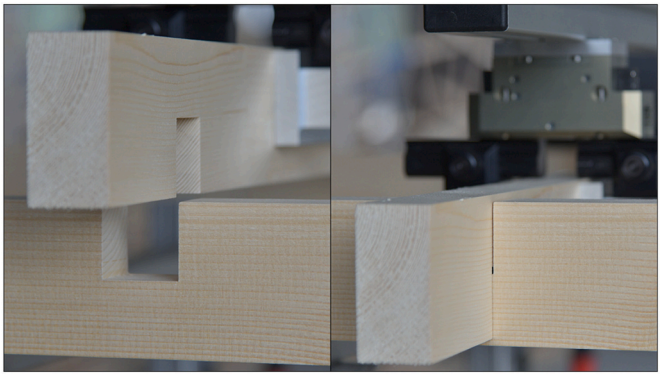

Fig. 5. Close-up of a 90° lap joint with a gap $b = 1$ mm.

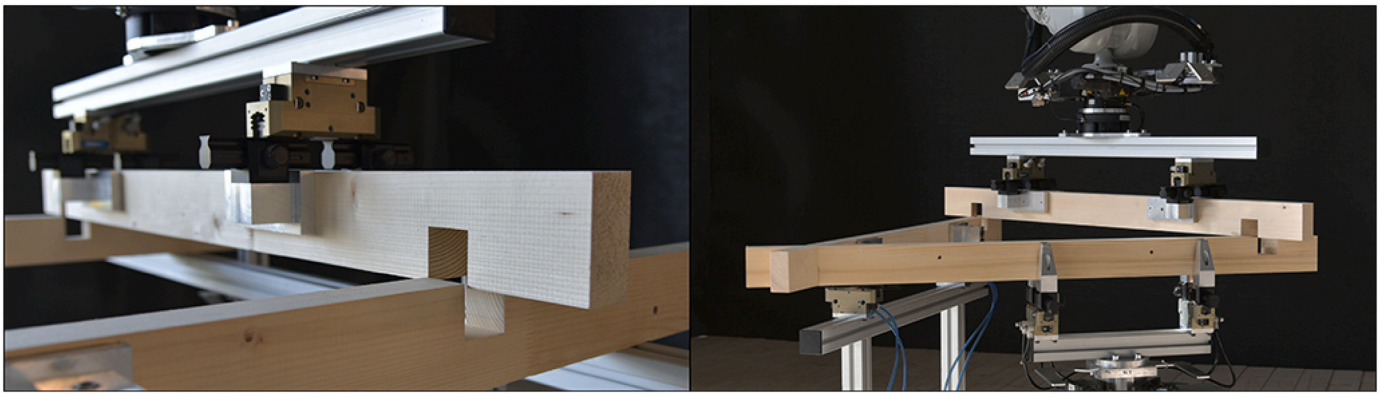

Fig. 6. Stills from an assembly of an H-shaped (left) and Δ-shaped (right) double-lap tasks by the real robot.

inserted member is constant in all experiments: above the notch in a single-lap joint and centred between two notches in the double-lap joint tasks. The overview of the experiments and results is given in Fig. 4. A video showing the execution of our experiments is available at https://vimeo.com/359005752/3a1513ed75.

5.2. Single-lap tasks

First, we test how the policy learned in simulation transfers to reality by rolling it out on a physical robot with the same default task parameters used during training (Fig. 5). It achieves a 100% success rate.

Next, we challenge the same single-lap policy (trained with $\alpha = 90°$) to assemble joints with α equal to 75°, 60°, and 45°. With the angle decreasing, the area of and, thus, the friction between mating faces is greater than in training. Additionally, FT readings and movement constraints will naturally differ with notch geometry. Both in the simulation and in reality rollouts, the policy performs robustly in the range 90° – 60°, which is a considerably high flexibility for the envisioned architectural applications.

Subsequently, we test if the policy is able to handle different levels of tight-fitness. As the gap b in the joint decreases the policy must produce finer movements. As expected, the policy easily succeeds if the gap is increased. When decreasing the gap from the default $b = 1$ mm to $b = 0.5$ mm, the policy still performs well in simulation but fails to do so in reality. These failures are caused by contact forces that violate the allowable range of our sensor, mostly torque on the X- and Y-axis.

Finally, to test if our policy is able to perform the insertion even if the elements are initially misaligned, which can arise from a combination of different unintended tolerances, we apply a combination of linear (≤ 10 mm) and angular ($\leq 4°$) offsets at the initial member pose. The policy proves to be reliable with up to 2 mm linear offsets in the X- and Y-direction, up to 5 mm linear offsets in the Z-direction, or 2° angular offsets on all axes. This range of adaptability may be promising for assemblies of larger scale, where offsets of this degree are common. Unfortunately, combinations of linear and angular offsets are largely unsuccessful and this would need to improve for a practical application in industry.

5.3. Double-lap tasks

Although a single-lap is a good initial test scenario, in timber framing

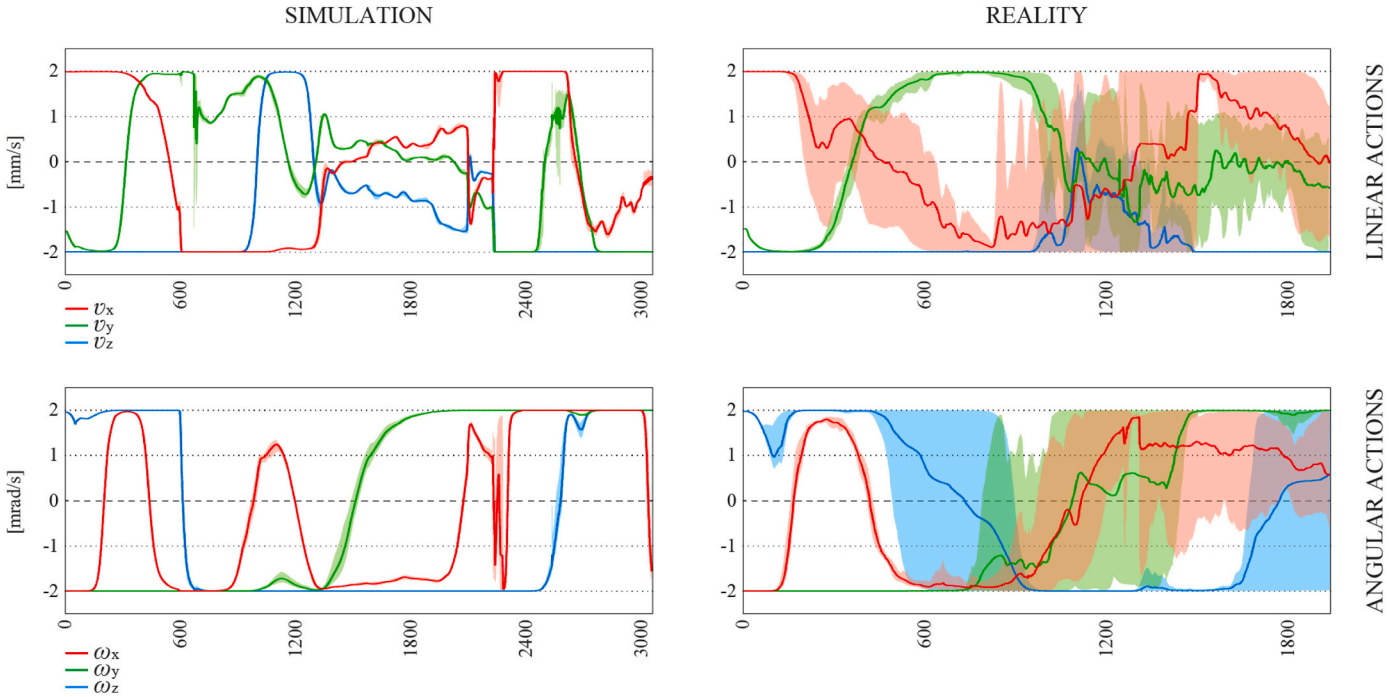

Fig. 7. Plots of linear and angular actions from rollouts in simulation and reality, from all conducted single-lap tasks where $\alpha = 90°$, $b = 1$ mm, and offsets are zero. All these rollouts were successful, as summarized in Fig. 4. The solid line represents the average, and the shaded area the bounds, of the values.

it is much more likely for a member to engage with more than one other member simultaneously. In order to consider such connections, next we attempt to join a timber member with two other members in H- and Δ-shaped configurations (Fig. 6). We train a policy using an H-shaped design with right angles, and deploy it for both the H- and Δ-shaped specimen. We are able to successfully and consistently assemble H-shaped joints, but the policy fails to generalize to Δ-shaped joints. Failures are caused by violating the FT sensor limits, specifically, by generating excessive torque about the Z-axis; the joints, displaced from center, form a significant lever arm during assembly. Also, if the two joints become unevenly engaged, for example due to rotation around the X-axis, they may jam.

5.4. Simulation vs. reality

On the success rates of rollouts in simulation compared with those in reality, approximately more than half of all tasks have similar outcomes, as shown in the last two columns of Fig. 4.

We observe that the tasks where success rates are significantly lower in reality are those with difficult parameters, i.e. largely different from the ones used in training, such as when $b = 0.5$, $\alpha = 45°$, with large x_0 or with substantially different joint geometry. We suspect it is because reality is less forgiving than simulation in terms of friction and contact forces. More accurate modeling of friction and contact forces in simulation may be desirable in the future.

We compare observations and actions of a rollout in simulation vs. reality in Fig. 7. There is a significantly higher variance in reality's action space, which we attribute to the policy's response to the large amount of noise FT observations captured by the sensor. While observations in reality differ substantially from those in simulation, actions output from the policy are appropriate responses. This shows the successful sim-to-real transfer and the capacity of the learned policy to generalize.

6. Conclusion and future work

This paper presented how RL can be applied to robotic construction to assemble timber structures using industrial robots. Exemplified by lap-joint assembly tasks, the experiments demonstrate that control policies trained in simulation can successfully transfer to reality and can overcome geometric inaccuracies during assembly.

In particular, the flexibility of the policy in single-lap assembly tasks exceeded our expectations. The fact that it can overcome positional tolerances suggests that during training the neural network learns to base actions more on force torque observation than on pose observation. In the future, pose observation could be removed to base actions purely on contact forces. In contrast, the ability to handle multi-engagement assembly tasks requires further work towards the reduction of torques arising from the lever-arm situations. For this, instead of a single FT sensor mounted before the gripper, a system of distributed sensors at the fingertips [42] could give a more meaningful haptic feedback from the robot. In terms of tolerances, our ability to assemble tight-fitting joints (with less than 1 mm gap) was considerably compromised by limitations of our experimental physical setup, i.a. the range of the FT sensor. In future work, penalties for actions that result in violation of sensor limits could also be included in the reward formulation during training, e.g. by Constrained Policy Optimization [43]. Furthermore, acceleration-based actions could be investigated to enable greater velocities and smoothness in robot control. It would be also interesting to explore success criteria other than distance or incorporate other sensing routines in order to ensure that tasks are properly assembled. Looking towards application in practice, future work could include a wider range of joint types, joint configurations and element geometries. Also, relative position between the member pose and the sensor pose could be added as a parameter to the training to add flexibility to gripping positions, or the policy could learn a wider range of experienced forces to accommodate for different positions.

The presented work addresses just a selected part of an overall robot-based assembly process and is only a small step towards automated timber construction. We address two challenges of automation in architectural construction which are essentially distinct from other manufacturing industries: the prevalence of uncertainties and small series. First, the control policy is able to handle tolerances, inaccuracies and deformations that naturally occur when building with elements made of wood at this scale. Second, by being able to cater to different shapes, a wider range of element geometries can be handled without the need to be reprogrammed for every new design. These traits are indispensable for automated timber construction to gain ground in the near future.

Supplementary data to this article can be found online at https://doi.org/10.1016/j.autcon.2021.103569.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

The project was a collaboration between Autodesk Research, Autodesk Inc. and Gramazio Kohler Research, ETH Zürich. The funding was provided by Autodesk. Real-world tasks were conducted at the Robotic Fabrication Laboratory of ETH Zürich [44]. We would like to thank Hannes Mayer and Erin Bradner for fostering this collaboration; Jieliang Luo for implementing the initial environment and algorithm; Lukas Stadelmann for implementing the EGM interfaces; BOTA Systems for providing the custom FT sensor; Klajd Lika and Ilias Patsiaouras for their generous assistance with the sensor.

Appendix A

A.1. Software

Simulations were conducted using PyBullet [45]. In simulation, the friction noise coefficient is set to 0.1.

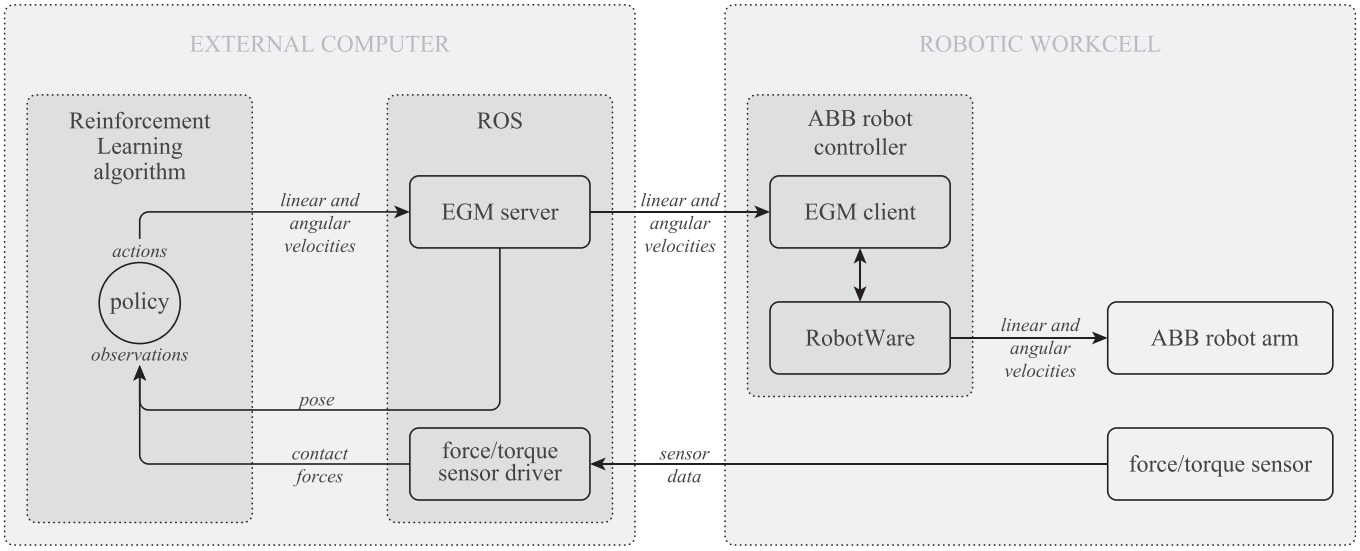

Fig. A.1. Architecture of the software-hardware interfaces.

The RL algorithm is based on the RLlib [46] implementation and default parameters. For training, the minimum iteration time is 60 s, the maximum number of steps taken per episode is 6000, and the minimum success rate is 90%. Maximum linear velocity is 0.002 m/s, and maximum angular velocity is 0.002 rad/s. Domain randomization in the form of Gaussian noise $\mathcal{N}(0, \sigma^2)$ is added to FT observations, where σ is [2 N, 2 N, 2 N, 0.2 Nm, 0.2 Nm, 0.2 Nm] as correlated noise, and [0.5 N, 0.5 N, 0.5 N, 0.05 Nm, 0.05 Nm, 0.05 Nm] as uncorrelated noise. The distance threshold, used to define the exit condition, is ε= 0.005 m. Other hyperparameters may be found in the code, available at https://github.com/AutodeskRoboticsLab/RLRoboticAssembly.

A.2. Physical setup

We use ABB IRB4600–40/2.55 robot mounted to a large, overhead gantry. To control the robot with the RL policy, we implemented following interfaces (see Fig. A.1). We use ROS and roslibpy [47] to communicate with the robot and the sensor at a rate of 50 Hz. For real-time control, we use ABB Externally Guided Motion (EGM) [48], which has control delay of 10–20 ms [48,p.328]. We use a prototypical BOTA Systems FT sensor [49]. The nominal calibrated range of the FT sensor is [3 kN, 3 kN, 6 kN, 100 Nm, 100 Nm, 77 Nm]. The resolution of an axis is equal to 1/10000 of its range. The accuracy of an axis is equal to 1% of its range.

Timber members are planed spruce slats, 40 × 80 × 1200 mm and 1.6 kg. These dimensions are small for construction but practical for our tasks. Notches are 40+b × 40 × 20 mm and CNC-milled; overall dimensional accuracy is sub-millimeter. As the material wore off during experiments, members were regularly replaced to avoid biasing success rates.

References

[1] J. Andres, T. Bock, F. Gebhart, W. Steck, First results of the development of the masonry robot system ROCCO: A fault tolerant assembly tool, in: D.A. Chamberlain (Ed.), Automation and Robotics in Construction Xi, Elsevier, Oxford, 1994, pp. 87–93. ISBN: 978-0-444-82044-0. https://doi.org/10.1016/B978-0-444-82044-0.50016-3.

[2] T. Bonwetsch, Robotically Assembled Brickwork. Manipulating Assembly Processes of Discrete Elements, Ph.D. thesis, ETH Zurich, 2015, https://doi.org/10.3929/ethz-a-010602028.

[3] Construction Robotics, Sam100 (semi-automated mason), https://www.construction-robotics.com/sam100, Accessed: 26/03/2020 (2018).

[4] FastBrick Robotics, Hadrian X, https://www.fbr.com.au/view/hadrian-x, Accessed: 26/03/2020 (2016).

[5] K. Orlowski, Automated manufacturing for timber-based panelised wall systems, Autom. Constr. 109 (2020) 102988. ISSN: 0926-5805, https://doi.org/10.1016/j.autcon.2019.102988.

[6] Weinmann, Frame work stations and combi-wall systems FRAMETEQ F-300/500/700 M-300/500, https://www.homag.com/fileadmin/product/house construction/brochures/weinmann-frame-work-stations-FRAMETEQ-en.pdf, Accessed: 26/03/2020 (2020).

[7] Homag, Robots in timber work, https://www.homag.com/en/news-events/news/article/robots-in-timber-work, Accessed: 05/12/2019 (2018).

[8] A.A. Apolinarska, Complex timber structures from simple elements: Computational design of novel bar structures for robotic fabrication and assembly, Ph.D. thesis, ETH Zurich, 2018, https://doi.org/10.3929/ETHZ-B-000266723.

[9] A. Thoma, A. Adel, M. Helmreich, T. Wehrle, F. Gramazio, M. Kohler, Robotic fabrication of bespoke timber frame modules, in: J. Willmann, P. Block, M. Hutter, K. Byrne, T. Schork (Eds.), Robotic Fabrication in Architecture, Art and Design 2018 (ROBARCH 2018), Springer International Publishing, Cham, 2019, pp. 447–458. ISBN: 978-3-319-92294-2. https://doi.org/10.1007/978-3-319-92294-2_34.

[10] S. Parascho, T. Kohlhammer, S. Coros, F. Gramazio, M. Kohler, Computational design of robotically assembled spatial structures. A sequence based method for the generation and evaluation of structures fabricated with cooperating robots, in: L. Hesselgren, A. Kilian, S. Malek, K.-G. Olsson, O. Sorkine-Hornung, C. Williams (Eds.), Advances in Architectural Geometry 2018 (AAG 2018), Klein Publishing GmbH, Vienna, 2018, pp. 112–139 (ISBN: 978-3-903015-13-5).

[11] M. Vukobratovic, A. Tuneski, Contact control concepts in manipulation robotics – an overview, IEEE Trans. Ind. Electron. 41 (1) (1994) 12–24. ISSN: 1557-9948, https://doi.org/10.1109/41.281603.

[12] G. Thomas, M. Chien, A. Tamar, J.A. Ojea, P. Abbeel, Learning Robotic Assembly from CAD, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), 2018, pp. 3524–3531. ISSN: 2577-087X, https://doi.org/10.1109/ICRA.2018.8460696.

[13] D. Whitney, Historical perspective and state of the art in robot force control, in: Proceedings. 1985 IEEE International Conference on Robotics and Automation vol. 2, 1985, pp. 262–268, https://doi.org/10.1109/ROBOT.1985.1087266.

[14] G. Zeng, A. Hemami, An overview of robot force control, Robotica 15 (5) (1997) pp. 473–482, Cambridge University Press, New York, USA, ISSN: 0263-5747. doi: https://doi.org/10.1017/S026357479700057X.

[15] T.P. Lillicrap, J.J. Hunt, A.E. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, Continuous Control with Deep Reinforcement Learning, arXiv e-printsAccessed: 13/09/2019, 2019.

[16] D. Horgan, J. Quan, G. Barth-Maron, M. Hessel, Distributed prioritized experience replay, in: 6th International Conference on Learning Representations (ICRL 2018), Vancouver, Canada, 2018. https://iclr.cc/Conferences/2018/Schedule?showEvent=134 (Accessed: 13/09/2019).

[17] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal Policy Optimization Algorithms, arXiv e-printsAccessed: 01/05/2020, 2017.

[18] W.H. Montgomery, S. Levine, Guided policy search via approximate mirror descent, in: D.D. Lee, M. Sugiyama, U.V. Luxburg, I. Guyon, R. Garnett (Eds.), Advances in Neural Information Processing Systems 29, 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Curran Associates, Inc., NY, USA, 2016, pp. 4008–4016 (ISBN: 978-1-5108-3881-9).

[19] G. Barth-Maron, M. W. Hoffman, D. Budden, W. Dabney, D. Horgan, D. Tb, A. Muldal, N. Heess, T. Lillicrap, Distributed distributional deterministic policy gradients, in: 6th International Conference on Learning Representations (ICRL 2018), Vancouver, Canada, 2018, https://iclr.cc/Conferences/2018/Schedule?showEvent=25, Accessed: 26/08/2020.

[20] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, O. Pieter Abbeel, W. Zaremba, Hindsight experience replay, in: I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems 30, 31th Annual Conference on Neural Information Processing Systems (NIPS 2017), Curran Associates, Inc., NY, USA, 2017, pp. 5048–5058 (ISBN: 978-1-5108-6096-4).

[21] M. Andrychowicz, B. Baker, M. Chociej, R. Zefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, J. Schneider, S. Sidor, J. Tobin, P. Welinder, L. Weng, W. Zaremba, Learning dexterous in-hand manipulation, The International Journal of Robotics Research 39 (1) (2020) 3–20. ISSN: 0278–3649, Publisher: SAGE Publications Ltd STM, https://doi.org/10.1177/0278364919887447.

[22] J. Luo, E. Solowjow, C. Wen, J.A. Ojea, A.M. Agogino, A. Tamar, P. Abbeel, Reinforcement learning on variable impedance controller for high-precision robotic assembly, in: 2019 International Conference on Robotics and Automation (ICRA), 2019, pp. 3080–3087. ISSN: 2577-087X, https://doi.org/10.1109/ICRA.2019.8793506.

[23] J. Luo, H. Li, Dynamic experience replay, in: L.P. Kaelbling, D. Kragic, K. Sugiura (Eds.), Proceedings of the Conference on Robot Learning (CoRL), Vol. 100 of Proceedings of Machine Learning Research, PMLR, Osaka, Japan, 2020, pp. 1191–1200, in: http://proceedings.mlr.press/v100/luo20a.html (Accessed: 17/05/2020).

[24] Y. Fan, J. Luo, M. Tomizuka, A learning framework for high precision industrial assembly, in: 2019 International Conference on Robotics and Automation (ICRA), 2019, pp. 811–817. ISSN: 2577-087X, https://doi.org/10.1109/ICRA.2019.8793659.

[25] T. Inoue, G. De Magistris, A. Munawar, T. Yokoya, R. Tachibana, Deep reinforcement learning for high precision assembly tasks, in: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2017, pp. 819–825. ISSN: 2153-0866, https://doi.org/10.1109/IROS.2017.8202244.

[26] M. Vecerk, T. Hester, J. Scholz, F. Wang, O. Pietquin, B. Piot, N. Heess, T. Rothörl, T. Lampe, M.A. Riedmiller, Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards, arXiv e-printsAccessed: 18/08/2020, 2017.

[27] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, P. Abbeel, Domain randomization for transferring deep neural networks from simulation to the real world, in: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2017, pp. 23–30. ISSN: 2153-0866, https://doi.org/10.1109/IROS.2017.8202133.

[28] L. Stadelmann, T. Sandy, A. Thoma, J. Buchli, End-effector pose correction for versatile large-scale multi-robotic systems, IEEE Robotics and Automation Letters 4 (2) (2019) 546–553. ISSN: 2377-3766, https://doi.org/10.1109/LRA.2019.2891499.

[29] J. Collins, D. Howard, J. Leitner, Quantifying the reality gap in robotic manipulation tasks, in: 2019 International Conference on Robotics and Automation (ICRA), 2019, pp. 6706–6712. ISSN: 2577-087X, https://doi.org/10.1109/ICRA.2019.8793591.

[30] M. Krammer, Individual Serialism Through the Use of Robotics in the Production of Large-Scale Building Components, in: D. Reinhardt, R. Saunders, J. Burry (Eds.), Robotic Fabrication in Architecture, Art and Design 2016, Springer International Publishing, Cham, 2016, ISBN 978-3-319-26378-6, pp. 460–467, https://doi.org/10.1007/978-3-319-26378-6_38.

[31] A.A. Apolinarska, R. Bärtschi, R. Furrer, F. Gramazio, M. Kohler, Mastering the Sequential Roof - Computational methods for integrating design, structural analysis and robotic fabrication, in: S. Adriaenssens, F. Gramazio, K. Matthias, A. Menges, M. Pauly (Eds.), Advances in Architectural Geometry (AAG 2016), vdf Hochschulverlag AG ETH Zürich, Zürich, 2016, pp. 240–258. ISBN: 978-3-7281-3778-4, https://doi.org/10.3218/3778-4_17.

[32] C. Robeller, Y. Weinand, V. Helm, A. Thoma, F. Gramazio, M. Kohler, Robotic Integral Attachment, in: Rethinking Design and Construction (FABRICATE 2017), vol. 3, UCL Press, 2017, pp. 92–97, ISBN: 978-1-78735-001-4. https://doi.org/10.3929/ethz-b-000229209.

[33] S. Levine, N. Wagener, P. Abbeel, Learning contact-rich manipulation skills with guided policy search, in: 2015 IEEE International Conference on Robotics and Automation (ICRA), 2015, pp. 156–163, ISSN: 1050-4729. https://doi.org/10.1109/ICRA.2015.7138994.

[34] F. Suárez-Ruiz, X. Zhou, Q.-C. Pham, Can robots assemble an IKEA chair? Science Robotics 3 (17) (2018) https://doi.org/10.1126/scirobotics.aat6385. Section: Focus, pp. eaat6385, ISSN: 2470–9476.

[35] A. Rupam Mahmood, D. Korenkevych, B.J. Komer, J. Bergstra, Setting up a reinforcement learning task with a real-world robot, in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018, pp. 4635–4640. ISSN: 2153-0866, https://doi.org/10.1109/IROS.2018.8593894.

[36] Y. Chebotar, A. Handa, V. Makoviychuk, M. Macklin, J. Issac, N. Ratliff, D. Fox, Closing the Sim-to-Real Loop: Adapting Simulation Randomization with Real World Experience, in: 2019 International Conference on Robotics and Automation (ICRA), 2019, pp. 8973–8979. ISSN: 2577-087X, https://doi.org/10.1109/ICRA.2019.8793789.

[37] A. Marco, F. Berkenkamp, P. Hennig, A.P. Schoellig, A. Krause, S. Schaal, S. Trimpe, Virtual vs. real: Trading off simulations and physical experiments in reinforcement learning with Bayesian optimization, in: 2017 IEEE International Conference on Robotics and Automation (ICRA), 2017, pp. 1557–1563. https://doi.org/10.1109/ICRA.2017.7989186.

[38] I. Mordatch, K. Lowrey, E. Todorov, Ensemble-CIO: Full-body dynamic motion planning that transfers to physical humanoids, in: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015, pp. 5307–5314. doi: https://doi.org/10.1109/IROS.2015.7354126.

[39] X.B. Peng, M. Andrychowicz, W. Zaremba, P. Abbeel, Sim-to-real transfer of robotic control with dynamics randomization, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), 2018, pp. 3803–3810. ISSN: 2577-087X, https://doi.org/10.1109/ICRA.2018.8460528.

[40] S. Vougioukas, Bias Estimation and Gravity Compensation for Force-Torque Sensors, in: 6th WSEAS International Multiconference on Circuits, Systems, Communications and Computers (CSCC 2002), WSEAS Press, Rethymno, Crete, Greece, 2002, pp. 8091–8094 (ISBN: 960-8052-63-7).

[41] E. Schuitema, L. Buşoniu, R. Babuka, P. Jonker, Control delay in reinforcement learning for real-time dynamic systems: A memoryless approach, in: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010, pp. 3226–3231. ISSN: 2153-0866, https://doi.org/10.1109/IROS.2010.5650345.

[42] OnRobot, RG2-FT – smart robot gripper with in-built force/torque and proximity sensor, https://onrobot.com/en/products/rg2-ft-gripper, (Accessed: 08/04/2020).

[43] G. Dulac-Arnold, D. Mankowitz, T. Hester, Challenges of real-world reinforcement learning, in: Reinforcement Learning for Real Life (RL4RealLife) Workshop, Long Beach, California, USA, 2019. https://openreview.net/pdf?id=S1xtR52NjN (Accessed: 18/08/2020).

[44] Robotic Fabrication Laboratory, ETH Zürich. http://ita.arch.ethz.ch/archteclab/rfl.html (Accessed: 08/04/2020).

[45] E. Coumans, Y. Bai, Pybullet, a Python module for physics simulation for games, robotics and machine learning. http://pybullet.org.

[46] Ray Project, RLlib, https://github.com/ray-project/ray, Accessed: 18/08/2019 (2016).

[47] G. Casas, M. Lüdtke, Roslibpy: Python ROS Bridge Library. http://github.com/gramaziokohler/roslibpy (Accessed: 04/02/2018).

[48] ABB Application Manual - Controller Software irc5. https://us.v-cdn.net/5020483/uploads/editor/7y/p0a51v5nb8kg.pdf. Accessed: 08/04/2020.

[49] BOTA Systems, www.botasys.com, Accessed: 08/04/2020 (2019).