# Revisiting Reward Design and Evaluation for Robust Humanoid Standing and Walking

Bart van Marum, Aayam Shrestha, Helei Duan, Pranay Dugar, Jeremy Dao, Alan Fern

***Abstract*— A necessary capability for humanoid robots is the ability to stand and walk while rejecting natural disturbances. Recent progress has been made using sim-to-real reinforcement learning (RL) to train such locomotion controllers, with approaches differing mainly in their reward functions. However, prior works lack a clear method to systematically test new reward functions and compare controller performance through repeatable experiments. This limits our understanding of the trade-offs between approaches and hinders progress. To address this, we propose a low-cost, quantitative benchmarking method to evaluate and compare the real-world performance of standing and walking (SaW) controllers on metrics like command following, disturbance recovery, and energy efficiency. We also revisit reward function design and construct a minimally constraining reward function to train SaW controllers. We experimentally verify that our benchmarking framework can identify areas for improvement, which can be systematically addressed to enhance the policies. We also compare our new controller to state-of-the-art controllers on the Digit humanoid robot. The results provide clear quantitative trade-offs among the controllers and suggest directions for future improvements to the reward functions and expansion of the benchmarks.**

## I. Problem Statement and Related Work

We consider the problem of producing a controller for a bipedal humanoid robot that supports the following two commands: **1) Stand.** The robot should stop if moving and stand in place with two feet on the ground. **2) Walk.** The robot should walk at a specified velocity (direction and speed) and a specified heading with an important special case corresponding to rotating in place.

To be useful in practice, a *standing and walking (SaW)* controller must be able to reliably switch between different commands and reject physical disturbances, such as bumps or terrain features, that may occur in an application.

## II. Quantitative SaW Performance Benchmark

We propose a reproducible set of benchmarks for quantitatively assessing key aspects of a SaW controller in the real-world. These metrics quantify the disturbance rejection ability, accuracy in command following, and energy efficiency. The benchmark is intended to allow comparison of any SaW controller, regardless of the method a controller is based on.

### A. Disturbance Rejection

Figure 2 shows our impulse application device in the lab environment. The impulse applicator works by releasing a weight suspended by magnets, which is automatically disconnected after a preset duration. After applying a fixed duration impulse, the robot is freely able to recover.

All authors are with the Dynamic Robotics and Artificial Intelligence Laboratory. Oregon State University. Corvallis, Oregon, USA. Email: {vanmarub, shrestaa, duanh, dugarp, daoje, afern}@oregonstate.edu.

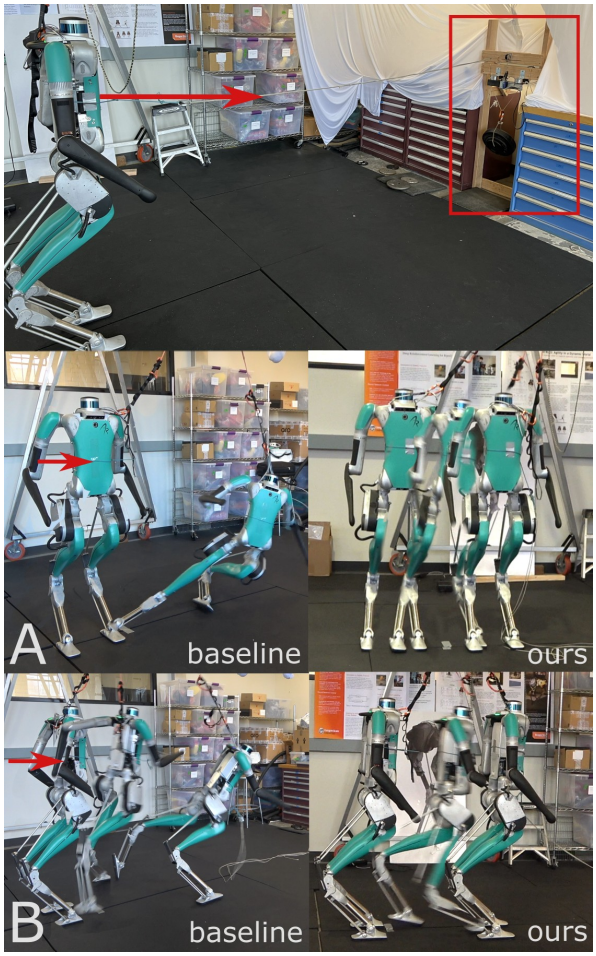


Fig. 1. We propose a set of metrics with an easy-to-setup testing fixture and provide quantitative results towards the controller performance in the real-world. Our proposed RL-based method produces a robust standing-and-walking controller for the humanoid robot Digit. The learned controller can handle a set of significant amount of disturbances, such as lateral push at 150N for 500ms shown in A and sagittal push at 200N for 500ms shown in B. The controller is able to walk, stand, and seamlessly transition between these two settings.

**Metric 1: Standing Fall Percentage.** For each direction and selected combinations of weight and duration, we compute the metric value over multiple trials. Each trial involves initializing the robot by issuing a standing command and then using the device connected in the appropriate direction to provide the specified impulse weight and duration. The metric value is the percentage of trials leading to success, where a trial is successful if the robot does not fall.

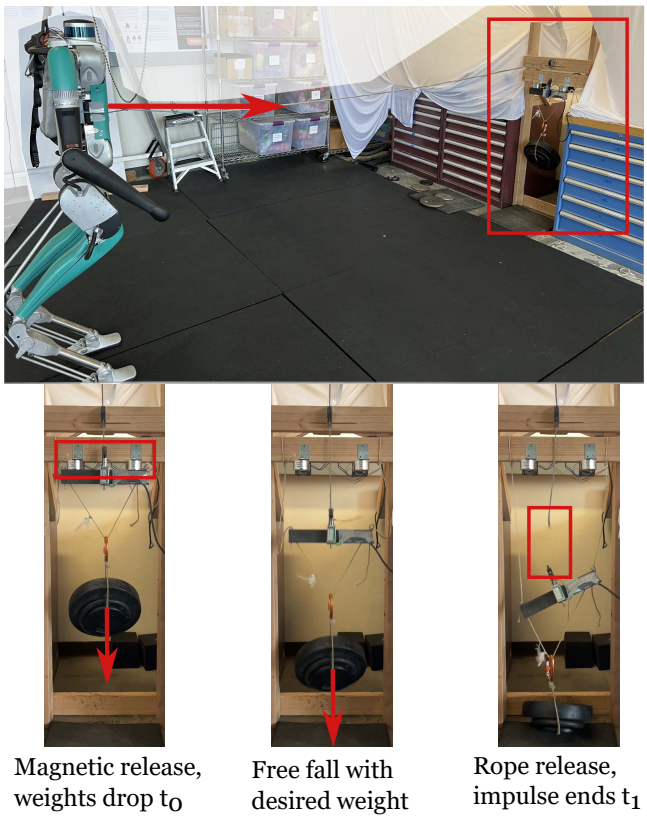


Fig. 2. An impulse is applied to the robot by means of a weight connected by a rope. Force $\boldsymbol{F}$ is regulated by adding and removing weight. Duration $\Delta t$ is regulated by a microcontroller that automatically disconnects the weight from the rope, after a set amount of time. The rope is always attached to Digit at the same height of 122 cm.

### B. Command Following

**Metrics 2 and 3: In-Place Rotation Accuracy.** For certain applications it is useful for a humanoid to be able to rotate its body to a particular orientation while remaining in place. To test this we conduct trials where the robot starts in a standing position in the middle of a 2ft diameter circle, which is considered the region of zero positional error. The robot is then given a command to rotate at angular velocity $\omega_z$ for $\Delta t$ seconds, which ideally should correspond to a commanded orientation change of $\theta_c = \omega_z \cdot \Delta t$. We compute two metrics at the end of the trial: 1) *angular error*, which is the difference between the commanded angular rotation and the actual rotation, and 2) *lateral drift*, which is measured as the distance of the furthest foot from the boundary of the starting circle.

**Metric 4: Velocity Accuracy.** We consider a simple velocity tracking test using only basic measurements. Specifically, we issue a constant velocity command $v$ for duration $\Delta t$, which should ideally produce a net translation of $d_c = v \cdot \Delta t$. For our current procedure, each trial of the experiments starts the duration clock when the robot is in a standing position and then after $\Delta t$ seconds the standing command is issued. The distance traveled is then manually measured and averaged across trials. By comparing the actual distance traveled $d_r$ to the commanded distance $d_c$, we can quantify velocity control performance without specialized equipment.

TABLE I
REWARD TERMS

| Reward Term | Definition | Weighting |
|---|---|---|
| $x, y$ velocity | $\begin{cases} e^{-5\cdot(v_{xy}-c_{xy})} & \text{if } c_s \\ e^{-5\cdot(v_{xy}-c_{xy})^2} & \text{else} \end{cases}$ | 0.15, 0.15 |
| Yaw orient. | $e^{-300\cdot qd(\mathbf{q}_{yaw},\mathbf{c}_{yaw})}$ | 0.1 |
| Roll, pitch orient. | $e^{-30\cdot qd(\mathbf{q}_{rp},\mathbf{c}_{rp})}$ | 0.2 |
| Feet contact | $\begin{cases} 1 & \text{if } c_s \\ 1 & \text{if } n_{c,t^*} = 1 \text{ for any } t^* \in [t-0.2, t] \\ 0 & \text{else} \end{cases}$ | 0.1 |
| Base height | $e^{-20\cdot\lvert p_z - c_h\rvert}$ | 0.05 |
| Feet airtime | $\begin{cases} 1 & \text{if } c_s \\ \sum_{f\in(l,r)}(t_{air,f}-0.4)*\mathbb{1}_{td,f} & \text{else} \end{cases}$ | 1.0† |
| Feet orientation | $\begin{cases} e^{-\sum\lvert\mathbf{r}_{\text{feet, rp}}-\mathbf{c}_{feet,\text{rp}}\rvert} & \text{if } \lvert c_{yaw}\rvert > 0 \\ e^{-\sum\lvert\mathbf{r}_{\text{feet, rpy}}-\mathbf{c}_{feet,\text{rpy}}\rvert} & \text{else} \end{cases}$ | 0.05 |
| Feet position | $\begin{cases} e^{-3\cdot\lvert\mathbf{p}_{feet}-\mathbf{c}_{feet}\rvert} & \text{if } c_s \\ 1 & \text{else} \end{cases}$ | 0.05 |
| Arm | $e^{-3\cdot\lvert\lvert\boldsymbol{\theta}_{\text{arm}}-\mathbf{c}_{\text{arm}}\rvert\rvert}$ | 0.03 |
| Base acceleration | $e^{-0.01\cdot\sum\lvert\mathbf{b}_{xyz}\rvert}$ | 0.1 |
| Action difference | $e^{-0.02\cdot\sum\lvert\mathbf{a}_t-\mathbf{a}_{t-1}\rvert}$ | 0.02 |
| Torque | $e^{-0.02\cdot\frac{1}{N}\sum\lvert\mathbf{t}_{\text{motor}}\rvert/\mathbf{t}_{\max}}$ | 0.02 |

$c$ = a command; $c_s$ = standing command; $q$ = a quaternion; $p$ = a position; $\boldsymbol{b}$ = base acceleration; $qd(\cdot)$ = quaternion distance function; $n_c$ = number of feet in contact with ground; $\mathbb{1}_{td}$ = boolean variable indicating a touchdown in the current timestep; † = note that the feet airtime reward is the only sparse reward, therefore the weight is significantly higher than other terms.

### C. Energy Efficiency

**Metric 5: Energy Efficiency.** More efficient gaits directly extend operational runtime by conserving battery power. Additionally, a more efficient gait reduces mechanical wear from torque and impacts, prolonging hardware lifespan.

## III. SAW TRAINING AND REWARD DESIGN

### A. Reward Design

**Basic Command Following.** The first three essential components in Table I measure how well the current robot velocities and orientation match the commands. We found that training with just these components results in a hopping locomotion behavior, where the robot moves by jumping with both feet.

**Single Foot Contact.** To address hopping we have found multiple approaches that can individually be added to the above three reward terms to learn to walk instead. We found that the most reliable and unconstrained way to produce walking instead of hopping is via the single foot contact reward, which also does not require tuning.

For non-standing commands, the single foot contact component provides a reward of 1 at each time step where only one foot is in contact with the ground. To allow for some overlap in the stance and swing phases, we add a grace period of 0.2 seconds. This means that if single contact occurred at least once in the last 0.2 seconds, the reward is granted, otherwise the reward is 0.

For the standing command, this reward component is a constant of 1, giving no preference for foot contact. Intuitively, we might expect standing to involve rewarding double foot contact. However, this is problematic since it

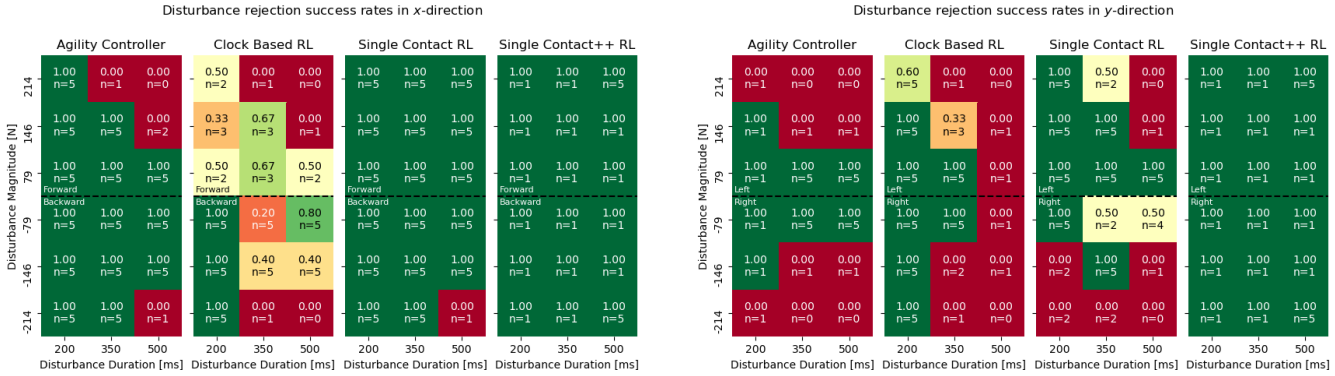


Fig. 3. Disturbance rejection success rates for various humanoid SaW controllers in the $x$-direction (left) and $y$-direction (right). Results show that our Single Contact reward function outperforms competing alternatives. * *Results for Single Contact++ are incomplete due to the robot being damaged in unrelated experiments, noting that experiments for the largest forces were completed before attempting to fill in the rest of the table.*

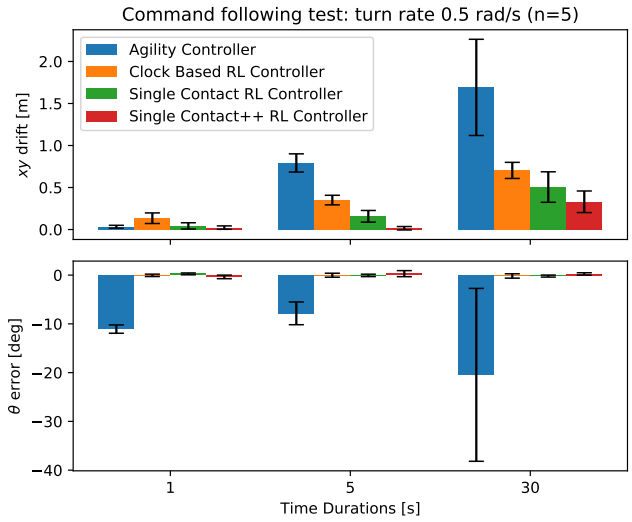


Fig. 4. Command following accuracy for turning in place. Error bars are standard deviation. Also note that the 30 seconds drift results for Agility Controller were in some cases helped by the robot tether. It is safe to assume results without tether would have been closer to the upper end of the error bar.

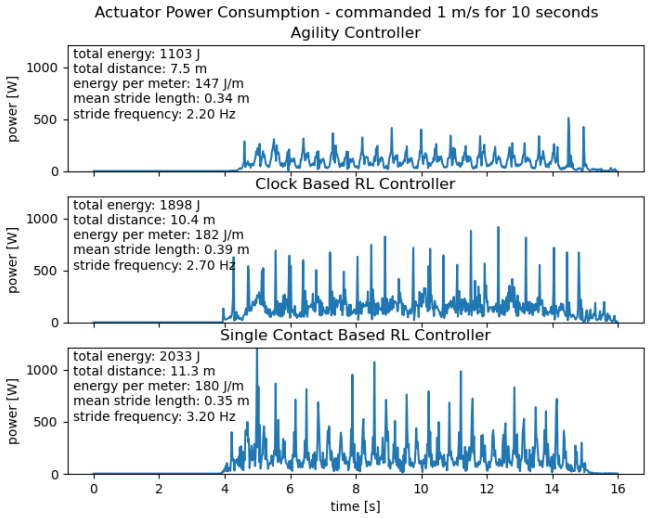


Fig. 5. Power consumption for a commanded run of 1 m/s for 10 seconds. Policies start in standing mode, and end in standing mode. * *Note that results for Single Contact++ RL Controller are missing due to an unrelated experiment damaging the robot close to submission.*

penalizes the recovery steps needed to reject disturbances, as that requires breaking ground contact of at least one of the feet. Additionally, when transitioning from walking to standing, requiring double foot contact will cause a policy to opt for the closest stance position rather than the most stable one. Thus, to learn standing while avoiding these problems we opt to implicitly reward standing utilizing existing reward terms. It turns out that most reward terms will be greater when a policy stands still with both feet on the ground, than when it steps in place or only stands on one foot.

**No Clocks.** While prior work that uses clock-based reward signals (e.g. [1]) does allow for standing, the nontrivial question of what to do with the required clock inputs during standing and transitions remains a challenge. Additionally, the clock framework incentivizes low foot velocities in standing mode, which directly impedes disturbance rejection capabilities. Rather, the above reward function does not require reference clocks, trajectories or signals of any sort to learn walking, and allows a policy to control such parameters internally. Without such signals we eliminate the problem of having to engineer the transitions between standing and walking modes. Additionally for disturbance rejection, without any references to attain to, the policy is free to move feet in any way it seems fit to stay upright.

## IV. Evaluation Results

We use our proposed benchmarking procedure to evaluate and compare three SaW controllers for the Digit V3 humanoid robot manufactured by Agility Robots: 1) *Single Contact RL.* trained using our minimally-constrained SaW reward function from Table I, 2) *Clock Based RL.* trained using a state-of-the-art clock-based [1] reward function, and 3) *Agility Controller* the manufacturer-provided controller. The benchmarks reveal unexpected failure modes in the learning-based controllers, which guided targeted improvements, ultimately resulting in an enhanced controller that successfully handles all tested disturbances, called the *Single Contact++ RL.* The results can be found in Figures 3, 4 and 5.

## REFERENCES

[1] Jonah Siekmann et al. “Sim-to-Real Learning of All Common Bipedal Gaits via Periodic Reward Composition”. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. Xi’an, China: IEEE Press, 2021, pp. 7309–7315. DOI: 10.1109/ICRA48506.2021.9561814. URL: https://doi.org/10.1109/ICRA48506.2021.9561814.