

## 537 Supplementary Materials

### 538 A Proof for Theorem 1

539 From  $V_i(x_i, p_i) \leq c_i(p_i)$  we know that the entering state  $x_i$  is within the maximum  $\epsilon$ -stable level set  
 540 of equilibrium point  $x^*$ , hence the entering state  $x_i$  is within the  $\epsilon$ -RoA of mode  $i$ . Next, we show  
 541 that the next entering state  $x_j = h_i(\bar{x}_i, u; p_i, p_j)$  is also within the  $\epsilon$ -RoA of mode  $j$ .

542 Since  $x_i$  is within  $\epsilon$ -RoA of mode  $i$ , we know  $\|\bar{x}_i - x^*\| \leq \epsilon$ . Then from  $\alpha_j \|x - x^*\| \leq V_j(x, p_j) \leq$   
 543  $\beta_j \|x - x^*\|$  we have

$$\begin{aligned}
 V_j(x_j, p_j) &= V_j(h_i(\bar{x}_i, u; p_i, p_j), p_j) \quad (\text{Definitions of the jump map, entering states and exiting states}) \\
 &\leq \beta_j \|h_i(\bar{x}_i, u; p_i, p_j)\| \quad (\text{Lyapunov bounding condition, and equilibrium at origin}) \\
 &\leq \beta_j \|h_i(x^*, u; p_i, p_j)\| + \beta_j K_i \|\bar{x}_i - x^*\| \quad (\text{Local Lipschitz condition for } h_i \text{ at } x^*) \\
 &\leq \beta_j \|h_i(x^*, u; p_i, p_j)\| + \beta_j K_i \epsilon \quad (\text{definition of } \epsilon\text{-RoA for mode } i) \\
 &\leq \frac{\beta_j}{\alpha_j} V_j(h(x_i^*, u; p_i, p_j), p_j) + \beta_j K_i \epsilon \quad (\text{Lyapunov function bounding condition}) \\
 &\leq \frac{\beta_j}{\alpha_j} \left( \frac{\alpha_j}{\beta_j} c_j(p_j) - \alpha_j K_i \epsilon \right) + \beta_j K_i \epsilon \quad (\text{The condition in Eq. 2}) \\
 &\leq c_j(p_j)
 \end{aligned} \tag{8}$$

544 therefore we derive that  $V_j(x_j, p_j) \leq c_j(p_j)$ , which means  $x_j$  is within the  $\epsilon$ -RoA for mode  $j$ . So  
 545 the whole hybrid system is  $\epsilon$ -stable according to Def. 3.

### 546 B Proof for Theorem 2

547 We consider the lower bound of  $\|p_i - p_k\|$  for every jump. We know that the Lyapunov value at the  
 548 entering state of mode  $k$  (denote the switching  $i \rightarrow k$ ) is:

$$\begin{aligned}
 V_k(h_i(x_i, u; p_i, p_k), p_k) &= V_k(h^+(x_i, p_i) + p_i - p_k, u; p_k) \quad (\text{definition of the special system}) \\
 &\leq \beta_k \|h^+(x_i, p_i) + p_i - p_k\| \quad (\text{Lyapunov bounding condition}) \\
 &\leq \beta_k \|h^+(x_i, p_i)\| + \beta_k \|p_i - p_k\| \quad (\text{Triangle inequality}) \\
 &\leq \beta_k \|h^+(x^*, p_i)\| + K_m \beta_k \|x_i - x^*\| + \beta_k \|p_i - p_k\| \quad (\text{Local Lipschitz condition}) \\
 &= K_m \beta_k \|x_i - x^*\| + \beta_k \|p_i - p_k\| \quad (\text{Since } h^+(x^*, p_i) = x^*) \\
 &\leq \beta_k K_m \epsilon + \beta_k \|p_i - p_k\| \quad (\text{Definition of } \epsilon\text{-RoA})
 \end{aligned} \tag{9}$$

549 If the optimization is feasible and the optimal  $p$  exists, then from the assumption we know that  
 550  $V_k(h^+(x_i, p_i) + p_i - p_k, p_k)$  for the optimal  $p$  must be no larger than  $c_k(p_k)$  (zero the first loss in  
 551 Eq. 7). We are going to show that  $V_k(h^+(x_i, p_i) + p_i - p_k, p_k)$  must strictly equal to  $c_k(p_k)$ . If  
 552 not, based on the continuity of the  $V_k$ , there must exist a  $\tilde{p}_k$  around  $p_k$  that can also zero the first  
 553 loss term in Eq. 7, and make  $\|\tilde{p}_k - p_j\| \leq \|p_k - p_j\|$  which brings contradiction. Thus we have  
 554  $V_k(h^+(x_i, p_i) + p_i - p_k, p_k) = c_k(p_k)$ , hence based on Eq. 9, we have:

$$\|p_i - p_k\| \geq \frac{c_k(p_k)}{\beta_k} - K_m \epsilon \tag{10}$$

555 For each jump, the step length is lower bounded as shown in Eq. 10. Thus we have the number of  
 556 jumps is:

$$N \leq \left\lceil \frac{\|p_j - p_i\|}{\min_m \frac{c_m}{\beta_m} - K_m \epsilon} \right\rceil \tag{11}$$

557 **C Details for the simulation environments**

558 **C.1 Car tracking control**

559 The goal here is to make sure the car can drive on the road under different road conditions.  
 560 Given a reference state  $(x, y, v, \psi)^T$  for a Dubins car model, the state of the car model is  
 561  $(x_e, y_e, \delta, v_e, \psi_e, \dot{\psi}_e, \beta)^T$ , where  $x_e, y_e$  represent the Cartesian error,  $\delta$  denotes the steering angle,  $v_e$   
 562 denotes the velocity error,  $\psi_e$  and  $\dot{\psi}_e$  are the heading angle error and angular velocity error, and  $\beta$  is  
 563 the slip angle. The dynamics are given by  $\dot{x} = f(x) + g(x)u$ , with

$$f(x) = \begin{pmatrix} v \cos(\psi_e) - v_{ref} + \omega_{ref} y_e \\ v \sin(\psi_e) - \omega_{ref} x_e \\ 0 \\ 0 \\ \dot{\psi}_e \\ -\frac{\mu m}{v I_x (l_r + l_f)} (l_f^2 C_{Sf} g l_r + l_r^2 C_{Sr} g l_f) (\dot{\psi}_e + \omega_{ref}) + \frac{\mu m}{I_x (l_r + l_f)} (l_r C_{Sr} g l_f - l_f C_{Sf} g l_r) \beta + \frac{\mu m}{I_x (l_r + l_f)} (l_f C_{Sf} g l_r) \delta \\ (\frac{\mu}{v^2 (l_r + l_f)} (C_{Sr} g l_f l_r - C_{Sf} g l_r l_f) - 1) (\dot{\psi}_e - \omega_{ref}) - \frac{\mu}{v (l_r + l_f)} (C_{Sr} g l_f C_{Sf} g l_r) \beta + \frac{\mu}{v (l_r + l_f)} (C_{Sf} g l_r) \delta \end{pmatrix} \quad (12)$$

564 and

$$g(x) = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \quad (13)$$

565 where  $u$  is the acceleration and the steering angle output,  $I_x, l_r, l_f, C_{Sf}, C_{Sr}, g$  are constant factors,  
 566 and  $\mu$  is the road friction factor. More details can be found in [2].

567 The road consists of multiple segments with different road conditions (different  $\mu$ ). Each segment  
 568 belongs to a system mode with the configuration of reference waypoint  $(X^E, Y^E)$ , reference velocity  
 569  $v^E$  and the friction factor  $\mu$ . Different combinations of friction factor and the velocity will give  
 570 different traction force for the vehicle. At junctions of the two segments, the mode switching causes  
 571 the system state jump because of the change of the reference waypoint.

572 **C.2 Pogobot navigation**

573 The state of the pogobot is  $s = (x, \dot{x}, y, \dot{y})^T$ , where  $x, y$  are the 2d coordinate of the pogobot head,  
 574 and the  $\dot{x}, \dot{y}$  are the corresponding velocities. The movement of a pogobot involves two phases. In  
 575 the flight phase, the pogobot follows a ballistic dynamics  $\dot{s} = f(s)$  where:

$$f(s) = \begin{pmatrix} \dot{x} \\ 0 \\ \dot{y} \\ -g \end{pmatrix} \quad (14)$$

576 here  $g$  is the gravity and the stance foot is determined by the pogobot pose  $\theta$  (which can be controlled  
 577 instantly, since we assume a mass-less leg). In the stance phase, together with the stance foot position  
 578  $(x_f, y_f)^T$ , the dynamics becomes

$$\dot{s} = \begin{bmatrix} \dot{x} \\ \frac{x-x_f}{\sqrt{(x-x_f)^2+(y-y_f)^2}} \left( k \left( \sqrt{(x-x_f)^2+(y-y_f)^2} - l_0 \right) + F \right) \\ \dot{y} \\ \frac{y-y_f}{\sqrt{(x-x_f)^2+(y-y_f)^2}} \left( k \left( \sqrt{(x-x_f)^2+(y-y_f)^2} - l_0 \right) + F \right) - g \end{bmatrix} \quad (15)$$

579 where  $l_0$  denotes the original length of the pogobot,  $k$  denotes the spring constant factor, and  $F$  and  $\theta$   
 580 are the control inputs (the stance force and the swing leg angle). Here we consider the apex-to-apex  
 581 control strategy. We first collect the simulation data to learn an apex-to-apex dynamic estimator.  
 582 Then we use this dynamic estimator to train our Lyapunov function and controllers (as well as the  
 583 RL-based methods).

584 **C.3 Bipedal walker locomotion**

585 The state of the Bipedal walker is  $s = (q, \dot{q})^T$  where  $q = (q_1, q_2)^T$ , and  $q_1$  is the angle between the  
 586 normal vector of the ground and the stance leg, and the  $q_2$  is the angle between the stance leg and  
 587 the swing leg.  $\dot{q}_1$  and  $\dot{q}_2$  are the corresponding angular velocities. Within each mode, the continuous  
 588 dynamics of the system follows the manipulator equation:

$$\dot{s} = \begin{bmatrix} \dot{q} \\ D_s^{-1}(q)[-C_s(q, \dot{q}) - G_s(q) + B_s(q)u] \end{bmatrix} \quad (16)$$

589 where  $D_s, C_s, G_s, B_s$  are functions of  $q$  (and  $\dot{q}$ ),  $u$  is the control input (torque in this case), and a  
 590 state jump will occur when  $q_2 + 2q_1 = 0$ . We recommend readers to [13] for more details.

591 **D Implementation of our approach**

592 **D.1 Car tracking control**

593 Before entering the  $i$ -th segment, we optimize for the configuration  $p_i$ , which is the waypoint  
 594  $W_i = (x_i^{ref}, y_i^{ref})$  at the junction between the  $i$ -th segment and the  $i + 1$ -th segment, and the  
 595 reference velocity  $v_i^{ref}$  to track on the  $i$ -th segment. And at the  $i$ -th segment, we use the environment  
 596 reference waypoint  $(x_{i+1}^E, y_{i+1}^E)$  and the reference velocity  $v_{i+1}^E$  for the next configuration. We make  
 597 sure: (1) the current entering state  $x_i$  is within the RoA of the current system under the configuration  
 598 of  $p_i = (W_i, v_i^{ref})$ . (2) the next entering state  $x_{i+1}$  is within the RoA of the system at segment  $i + 1$   
 599 with configuration  $p_{i+1}$ .

600 **D.2 Pogobot navigation**

601 Before entering the  $i$ -th segment, we optimize for the configuration  $p_i$ , which is the reference apex  
 602 state height and reference apex state horizontal velocity at the next cycle (during the  $i$ -th segment).  
 603 We can find the last apex state  $\tilde{x}_i$  before exiting the  $i$ -th segment using the dynamics estimator, and  
 604 make sure  $\tilde{x}_i$  is within the RoA for the  $i + 1$ -th segment under the reference apex state  $X_{i+1}^E$  given  
 605 from the environment.

606 **D.3 Bipedal walker locomotion**

607 In this case, due to the difficulty to synthesize a control Lyapunov function, for the low-level controller,  
 608 we directly use the QP controller derived from [13] and the corresponding Lyapuony function is  
 609 replaced by an RoA classifier, which outputs value  $< 0.9$  if the entering state is within the RoA, and  
 610 outputs value  $> 1.1$  for the rest case. During the planning, we use the differentiable planner but with  
 611 the loss Eq. 7 in to find the optimal configuration (in this case, the reference gait).

612 **E Implementation of baseline approaches**

613 **Reinforcement learning approaches:** We modify the RL implementation code from [https://github.com/RITCHIEHuang/DeepRL\\_Algorithms](https://github.com/RITCHIEHuang/DeepRL_Algorithms), created the RL environments for each ex-  
 614 periment, and train each RL method for 24 hours. For the car experiment, we use the reward as a  
 615 combination of Root Mean Square Error (RMSE) penalty with the reference state and the valid rate  
 616 of the trajectory segments. For the pogobot experiment, we use the RMSE, the collision rate with the  
 617 ceiling/floor, and the distance to the goal as the rewards/penalties. For the Bipedal walker, we use the  
 618 L2-distance from the current state to the reference state on the target gait as the penalty to guide the  
 619 controller to converge to the target gait.

620 **Model predictive control approaches:** We use the CasADi [3] to implement non-linear optimization  
 621 for each tasks. In each case, the system is simulated under some parameters (controls) and the cost  
 622 function is computed to optimize the control input. For the car experiment, the cost function is the  
 623 tracking error within the prediction horizon ( $T=20$ ). For the pogobot experiment, the cost function  
 624 is a penalty term with collisions and a tracking error term to the horizontal reference velocity. For  
 625 the bipedal walker experiment, the cost function is the L2-norm between the leg angle  $q_1$  after the  
 626 switching and the target gait leg angle  $q_1^{ref}$ .

628 **Linear quadratic control approaches:** At each segment, we require the car to track the segment  
 629 endpoint and the designed reference velocity on the current segment given the friction factors. We  
 630 compute the error dynamics, and synthesize the controller by solving the Algebraic Riccati Equation  
 631 similar as in [19].

632 **Control Lyapunov function approaches:** Here we jointly train a single NN Lyapunov function  
 633 for all the system modes with a NN controller (that can also take modes as inputs), with the same  
 634 amount of training time used as for our approach.

635 **CLF-QP approach:** We directly use the QP controller derived from [13] for the Bipedal walker  
 636 comparison.

637 **Hamilton Jacobian based approach:** We directly use the computed result (the value function)  
 638 from [13] for the comparison for the target gait with the leg angle  $q_1 = 0.13$  rad.

639 **F Success rate for Bipedal walker locomotion under different initial  
 640 conditions**

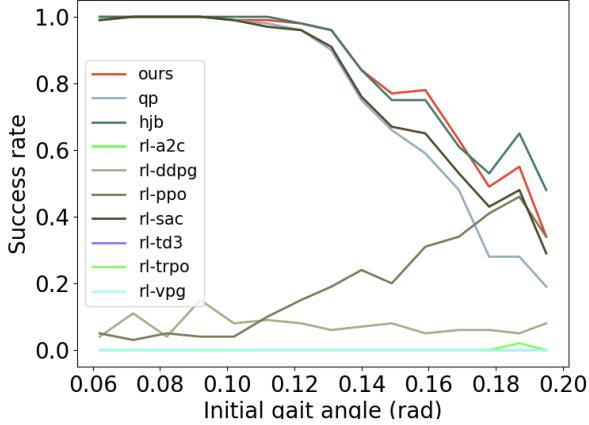


Figure 1: Walker success rate comparison under different initial states

641 For the bipedal walker experiment, we compare our approach with multiple RL-based approaches  
 642 (A2C, DDPG, PPO, SAC, TD3, TRPO, VPG) and classic methods (QP, HJB) under different initial  
 643 gait angles. As shown in Fig. 1, our approach can achieve similar-to-HJB performance, outperforming  
 644 all the RL-baselines and the QP baseline. The largest improvement (comparing to RL methods) is  
 645 from the "small initial angles". And our gain compared to QP-based methods is mostly from the  
 646 "large initial angles", which might be because the large deviation from the target gait angle makes the  
 647 linearization more inaccurate, hence the QP-based method cannot achieve good performance.

648 **G Visualization of learned RoA**

649 From Fig. 2 to Fig. 26, we show the visualization of the learned RoA in all three experiments under  
 650 different configurations.

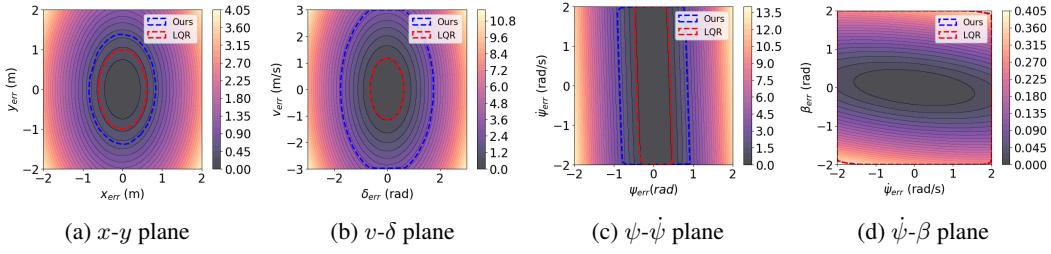


Figure 2: Car experiment (friction  $\mu = 1.0$ , reference speed  $v^E = 5m/s$ )

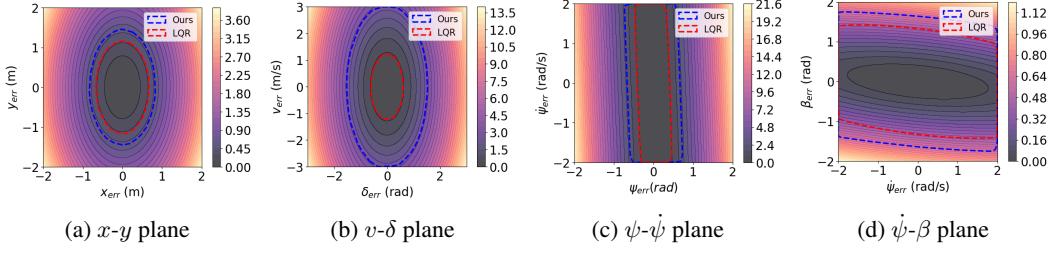


Figure 3: Car experiment (friction  $\mu = 0.1$ , reference speed  $v^E = 5m/s$ )

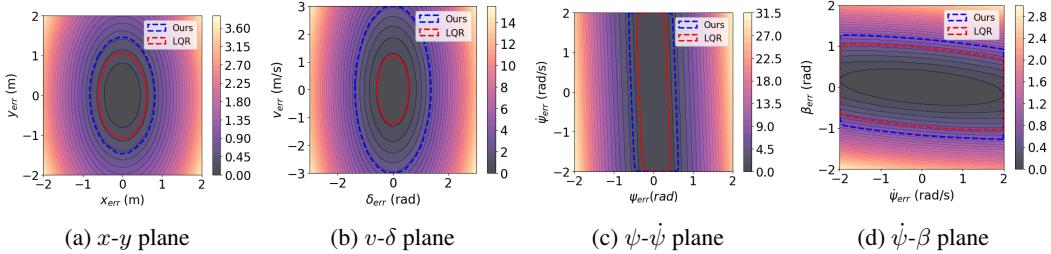


Figure 4: Car experiment (friction  $\mu = 1.0$ , reference speed  $v^E = 10m/s$ )

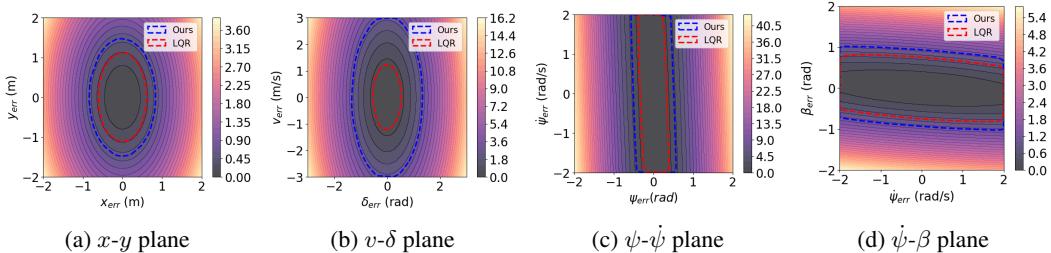


Figure 5: Car experiment (friction  $\mu = 0.1$ , reference speed  $v^E = 10m/s$ )

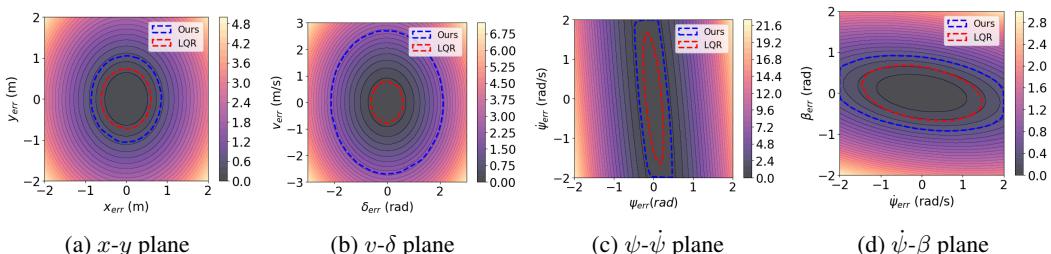


Figure 6: Car experiment (friction  $\mu = 1.0$ , reference speed  $v^E = 20m/s$ )

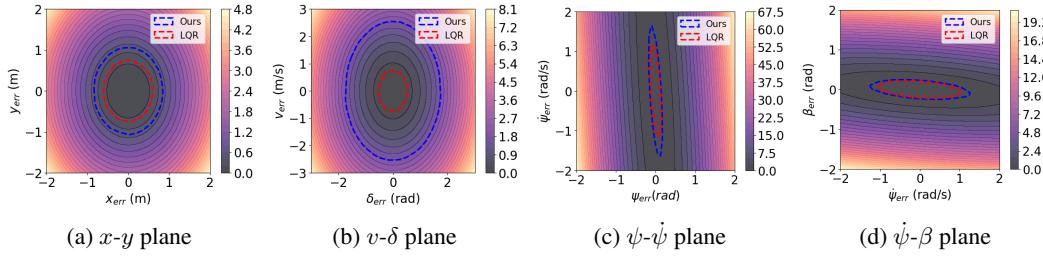


Figure 7: Car experiment (friction  $\mu = 0.1$ , reference speed  $v^E = 20m/s$ )

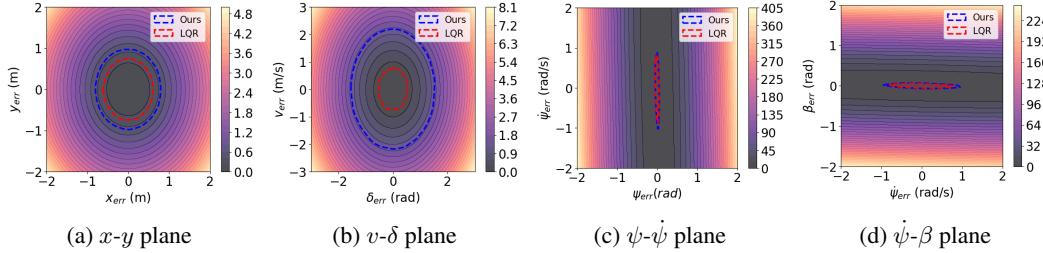


Figure 8: Car experiment (friction  $\mu = 1.0$ , reference speed  $v^E = 30m/s$ )

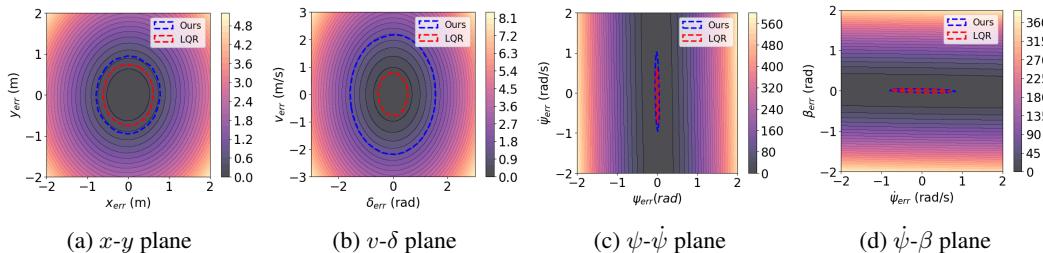


Figure 9: Car experiment (friction  $\mu = 0.1$ , reference speed  $v^E = 30m/s$ )

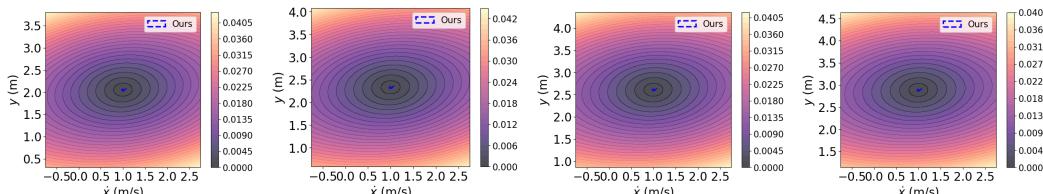


Figure 10: Pogobot experiment

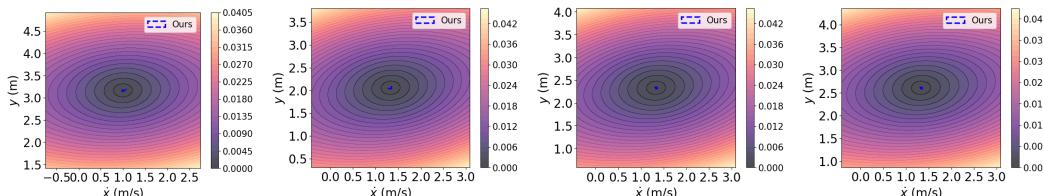
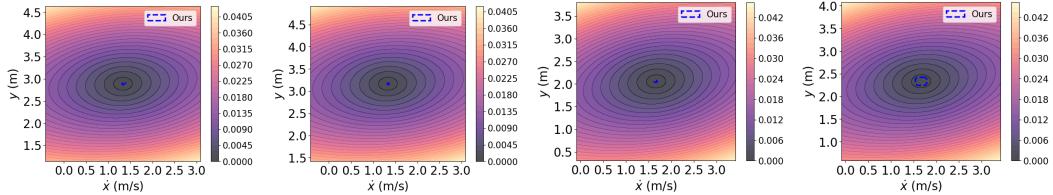
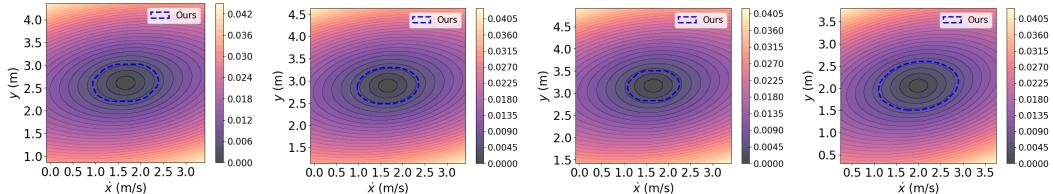


Figure 11: Pogobot experiment



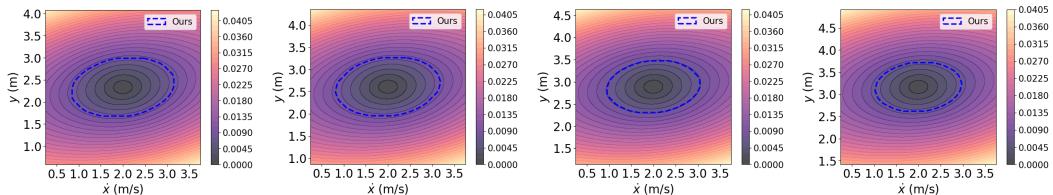
(a) apex velocity=1.3 m/s, apex height= 2.9 m      (b) apex velocity=1.3 m/s, apex height= 3.2 m      (c) apex velocity=1.7 m/s, apex height= 2.1 m      (d) apex velocity=1.7 m/s, apex height= 2.3 m

Figure 12: Pogobot experiment



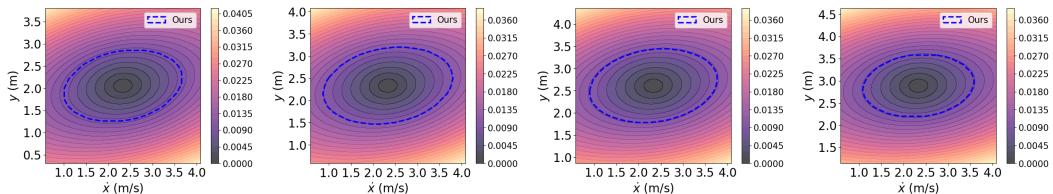
(a) apex velocity=1.7 m/s, apex height= 2.6 m      (b) apex velocity=1.7 m/s, apex height= 2.9 m      (c) apex velocity=1.7 m/s, apex height= 3.2 m      (d) apex velocity=2.0 m/s, apex height= 2.1 m

Figure 13: Pogobot experiment



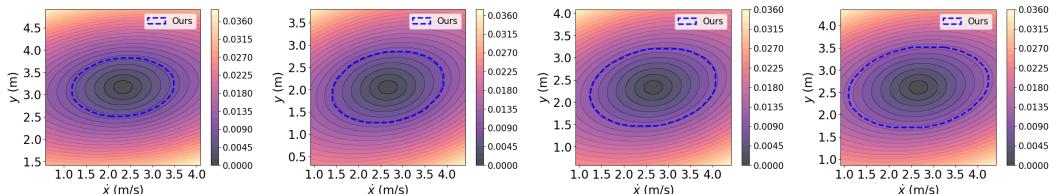
(a) apex velocity=2.0 m/s, apex height= 2.3 m      (b) apex velocity=2.0 m/s, apex height= 2.6 m      (c) apex velocity=2.0 m/s, apex height= 2.9 m      (d) apex velocity=2.0 m/s, apex height= 3.2 m

Figure 14: Pogobot experiment



(a) apex velocity=2.3 m/s, apex height= 2.1 m      (b) apex velocity=2.3 m/s, apex height= 2.3 m      (c) apex velocity=2.3 m/s, apex height= 2.6 m      (d) apex velocity=2.3 m/s, apex height= 2.9 m

Figure 15: Pogobot experiment



(a) apex velocity=2.3 m/s, apex height= 3.2 m      (b) apex velocity=2.7 m/s, apex height= 2.1 m      (c) apex velocity=2.7 m/s, apex height= 2.3 m      (d) apex velocity=2.7 m/s, apex height= 2.6 m

Figure 16: Pogobot experiment

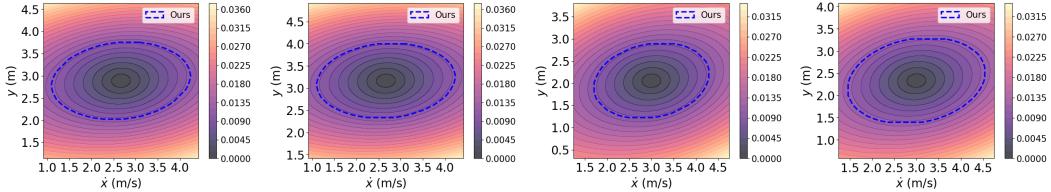


Figure 17: Pogobot experiment

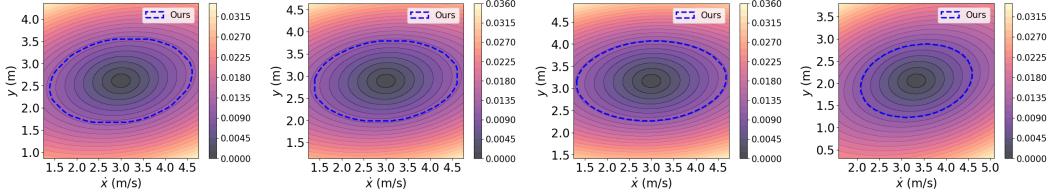


Figure 18: Pogobot experiment

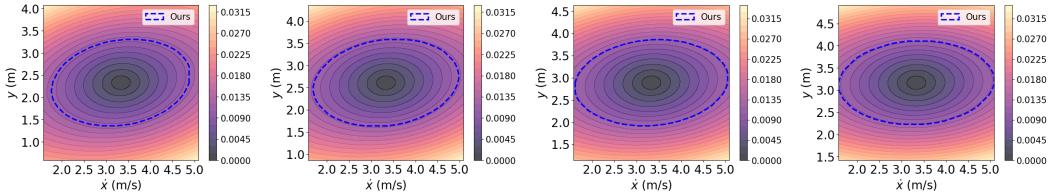


Figure 19: Pogobot experiment

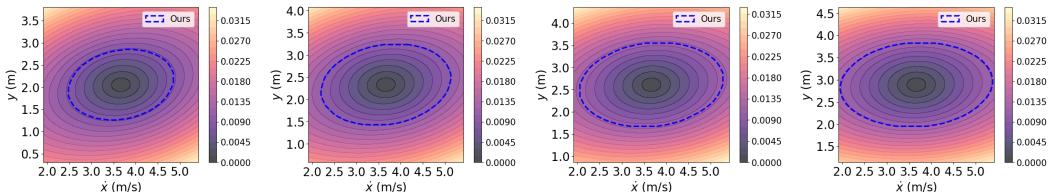


Figure 20: Pogobot experiment

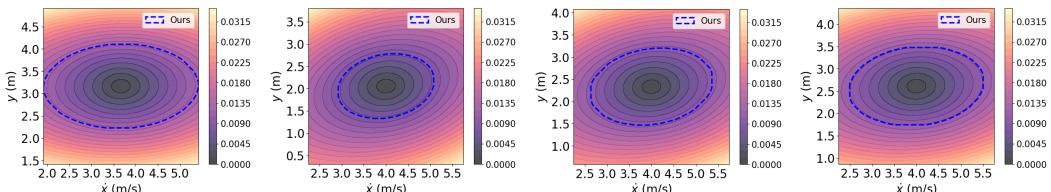


Figure 21: Pogobot experiment

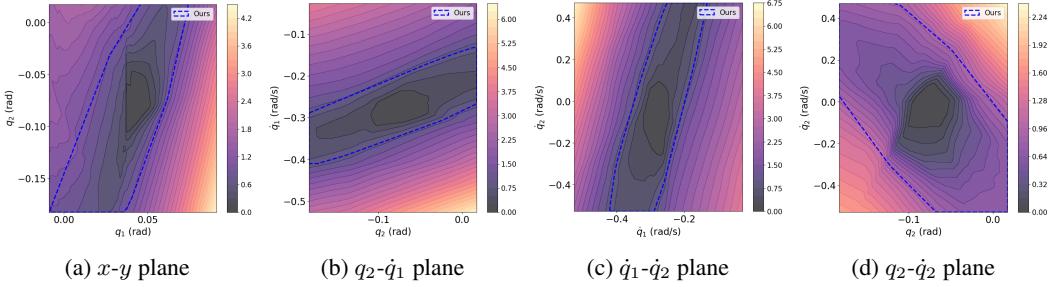


Figure 22: Bipedal walker experiment (reference gait  $q_1^{ref} = 0.05\text{rad}$ )

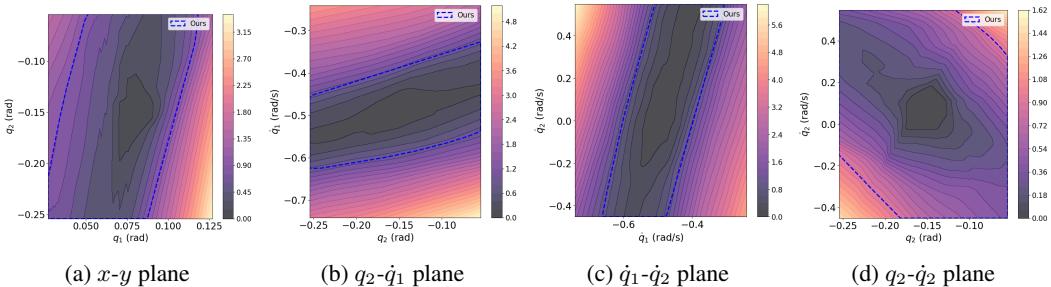


Figure 23: Bipedal walker experiment (reference gait  $q_1^{ref} = 0.08\text{rad}$ )

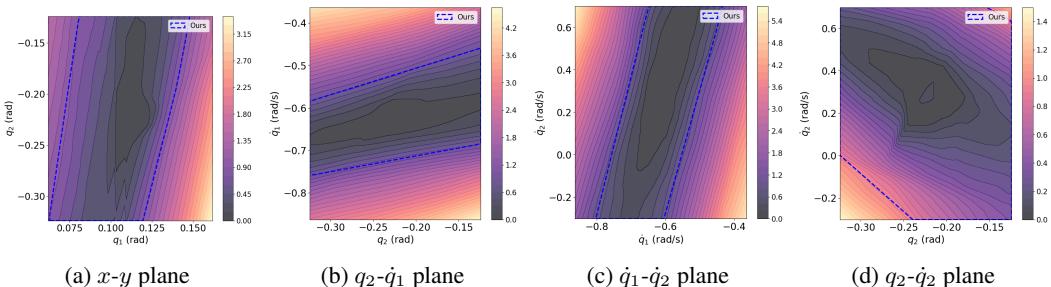


Figure 24: Bipedal walker experiment (reference gait  $q_1^{ref} = 0.10\text{rad}$ )

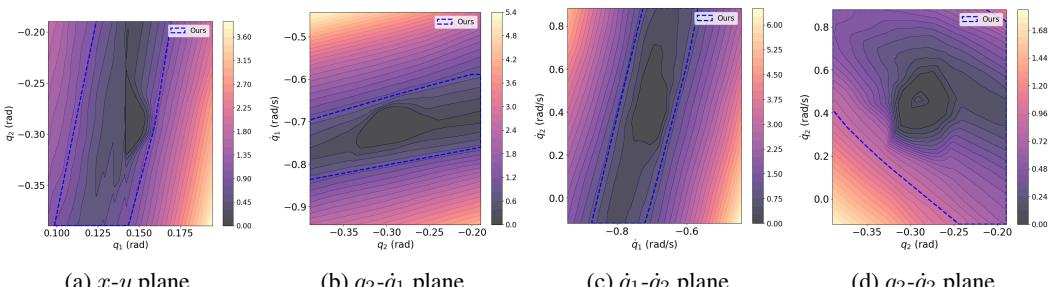


Figure 25: Bipedal walker experiment (reference gait  $q_1^{ref} = 0.13\text{rad}$ )

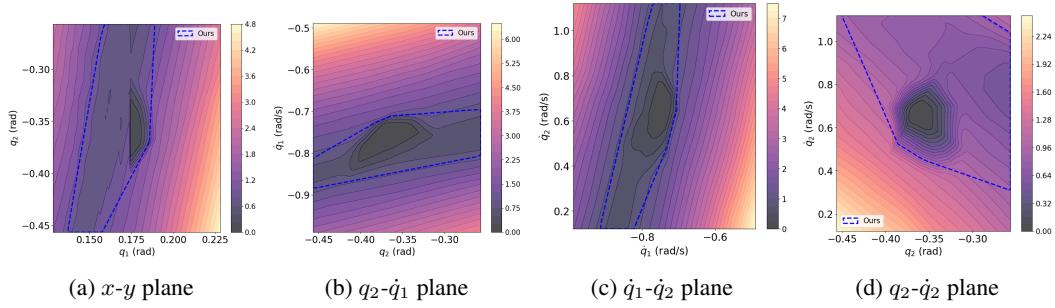


Figure 26: Bipedal walker experiment (reference gait  $q_1^{ref}=0.18\text{rad}$ )

## 651 H Visualization for the simulations

652 From Fig. 27 to Fig. 62, we show the simulation result visualization for all three experiments under  
653 different configurations.

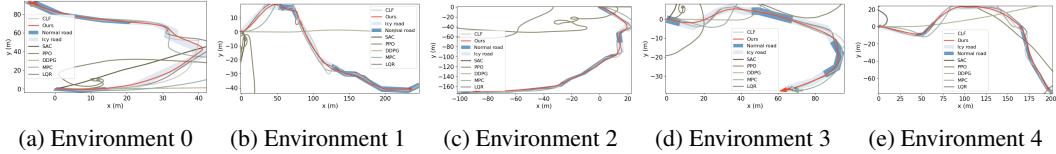


Figure 27: Car simulation comparisons (environment 0 ~ 4)

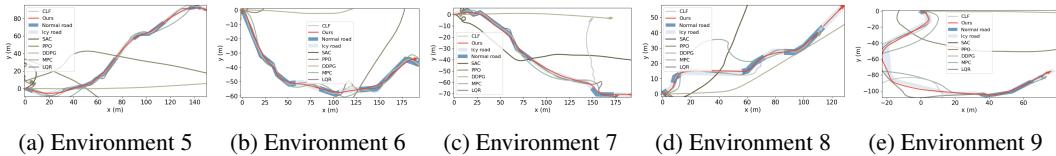


Figure 28: Car simulation comparisons (environment 5 ~ 9)

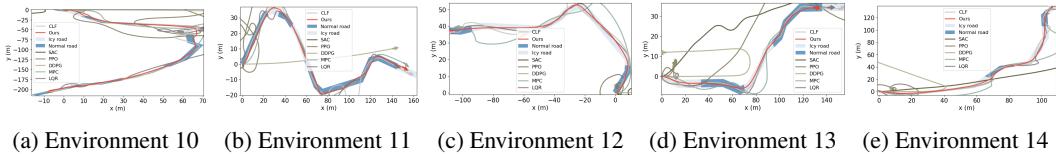


Figure 29: Car simulation comparisons (environment 10 ~ 14)

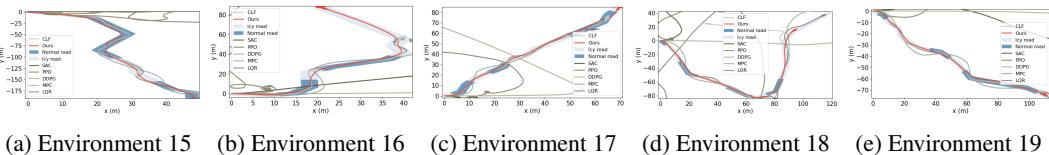


Figure 30: Car simulation comparisons (environment 15 ~ 19)

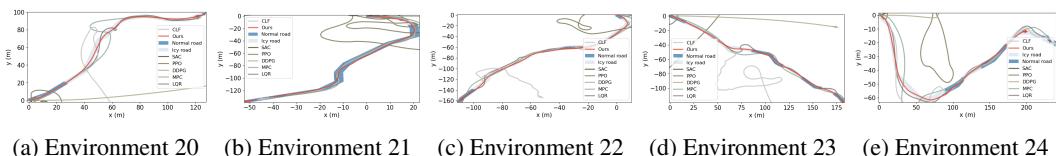


Figure 31: Car simulation comparisons (environment 20 ~ 24)

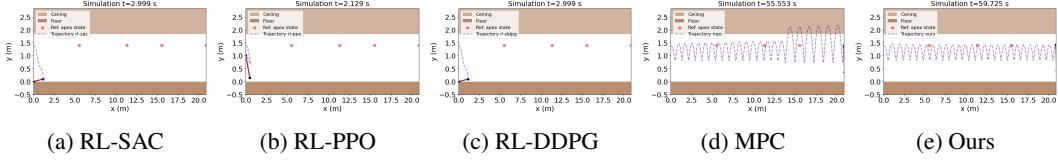


Figure 32: Pogobot simulation comparisons (environment 0)

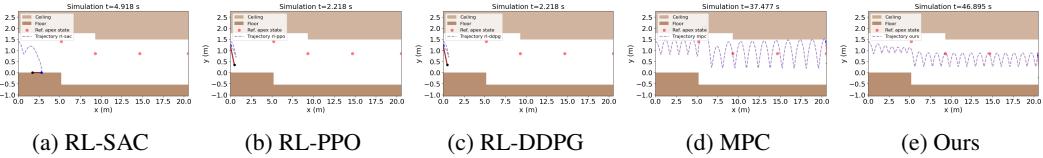


Figure 33: Pogobot simulation comparisons (environment 1)

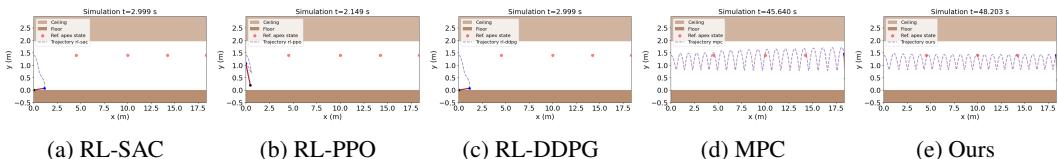


Figure 34: Pogobot simulation comparisons (environment 2)

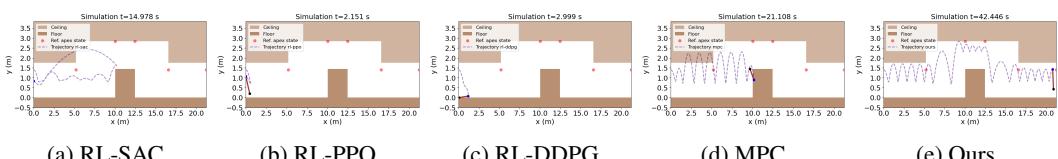


Figure 35: Pogobot simulation comparisons (environment 3)

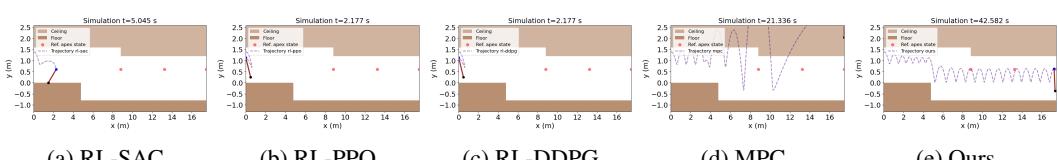


Figure 36: Pogobot simulation comparisons (environment 4)

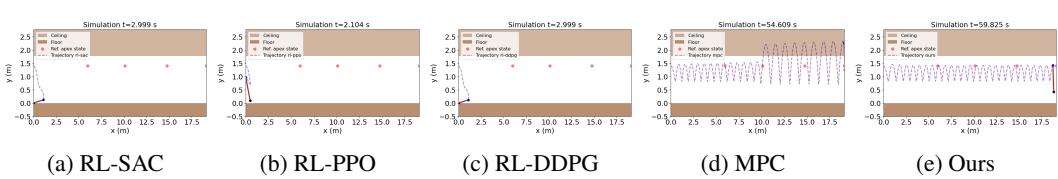


Figure 37: Pogobot simulation comparisons (environment 5)

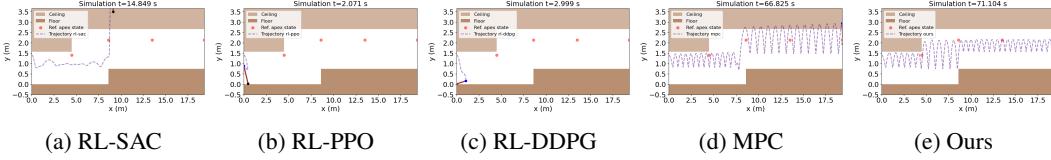


Figure 38: Pogobot simulation comparisons (environment 6)

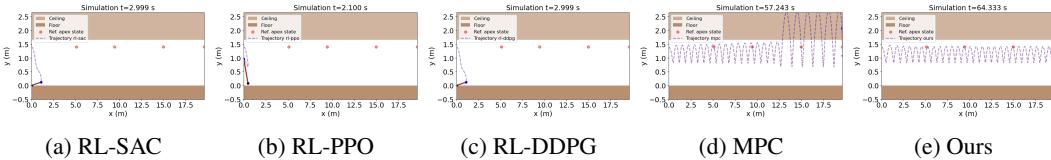


Figure 39: Pogobot simulation comparisons (environment 7)

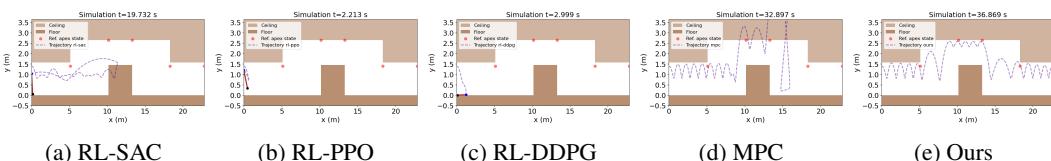


Figure 40: Pogobot simulation comparisons (environment 8)

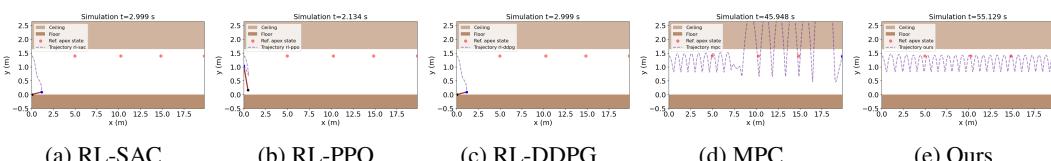


Figure 41: Pogobot simulation comparisons (environment 9)

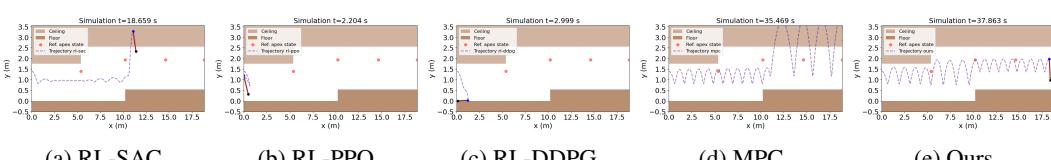


Figure 42: Pogobot simulation comparisons (environment 10)

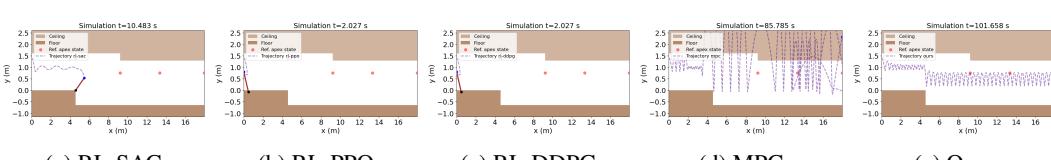


Figure 43: Pogobot simulation comparisons (environment 11)

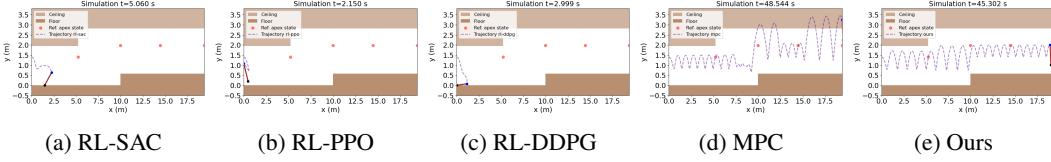


Figure 44: Pogobot simulation comparisons (environment 12)

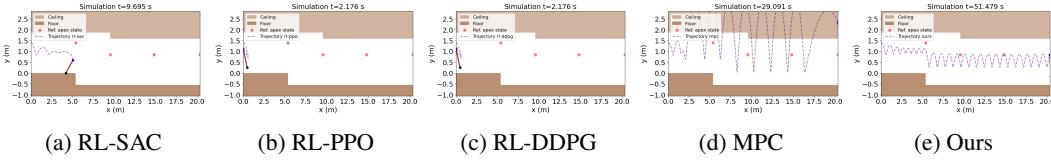


Figure 45: Pogobot simulation comparisons (environment 13)

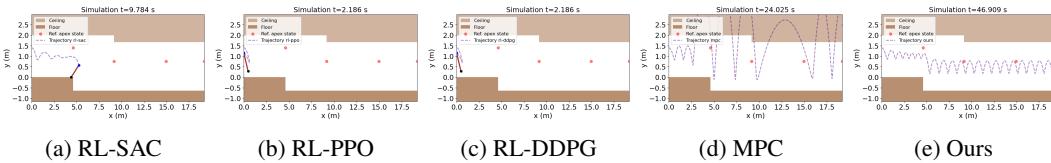


Figure 46: Pogobot simulation comparisons (environment 14)

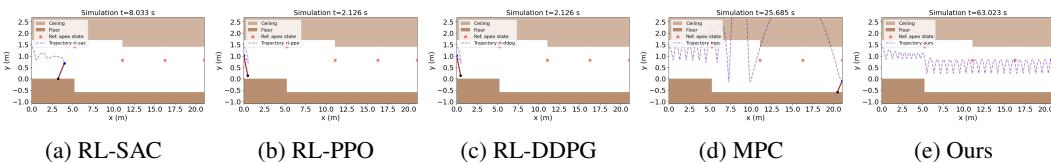


Figure 47: Pogobot simulation comparisons (environment 15)

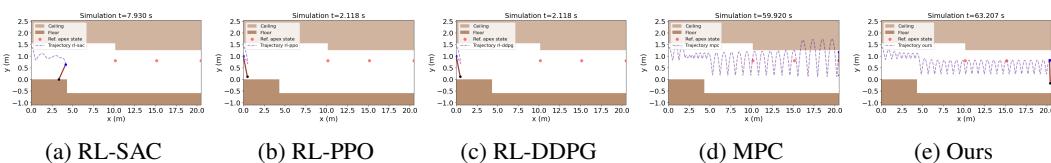


Figure 48: Pogobot simulation comparisons (environment 16)

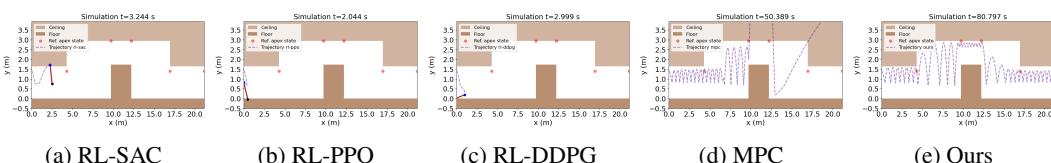


Figure 49: Pogobot simulation comparisons (environment 17)

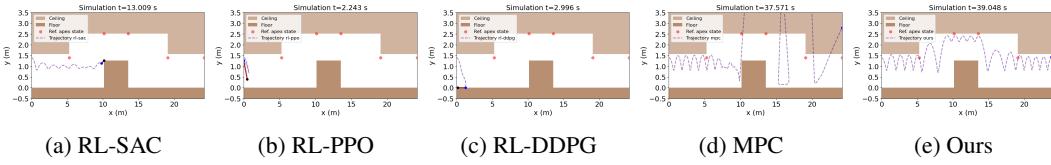


Figure 50: Pogobot simulation comparisons (environment 18)

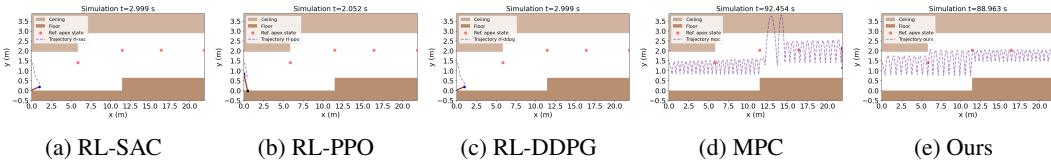


Figure 51: Pogobot simulation comparisons (environment 19)

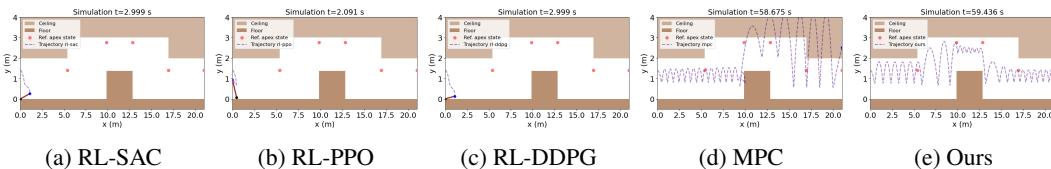


Figure 52: Pogobot simulation comparisons (environment 20)

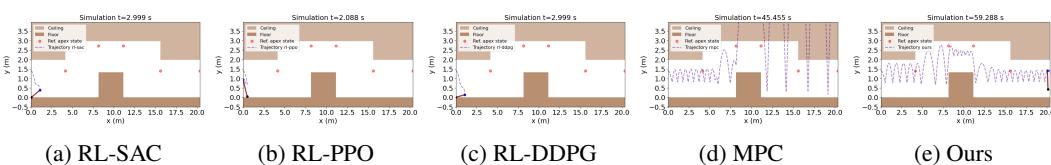


Figure 53: Pogobot simulation comparisons (environment 21)

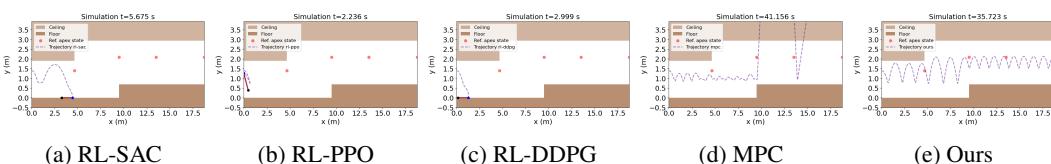


Figure 54: Pogobot simulation comparisons (environment 22)

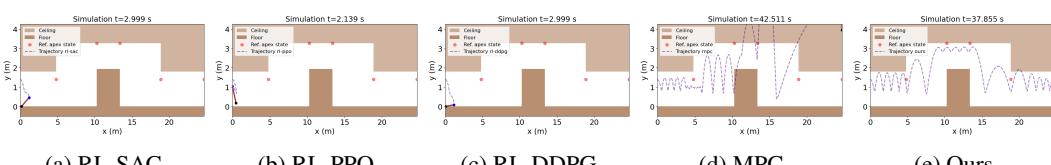


Figure 55: Pogobot simulation comparisons (environment 23)

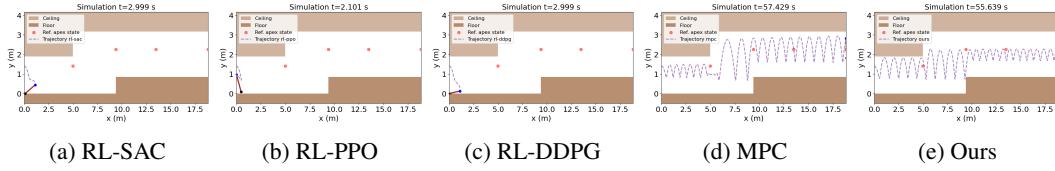


Figure 56: Pogobot simulation comparisons (environment 24)

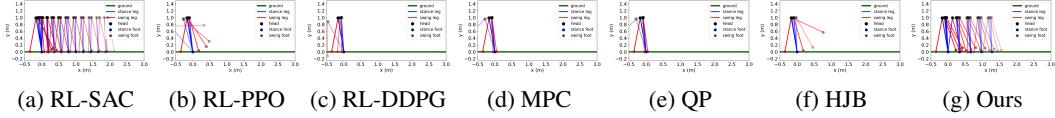


Figure 57: Bipedal walker simulation comparisons (same target angle)

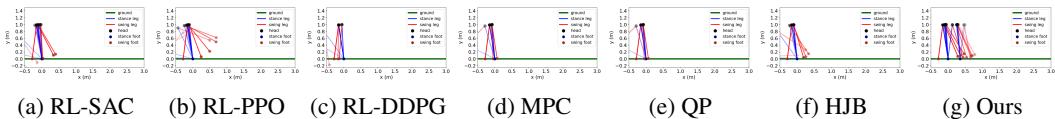


Figure 58: Bipedal walker simulation comparisons (same target angle)

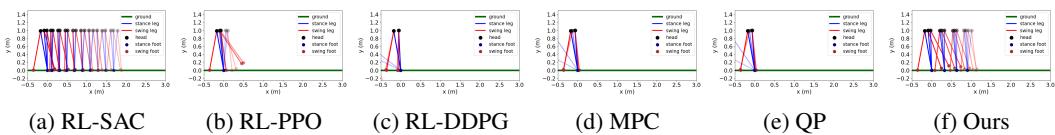


Figure 59: Bipedal walker simulation comparisons (different target angles)

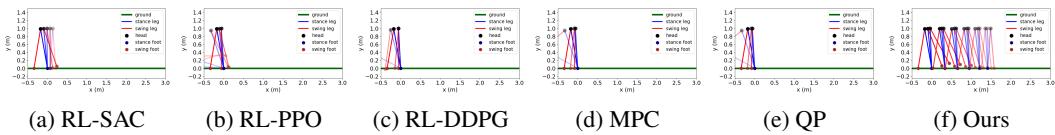


Figure 60: Bipedal walker simulation comparisons (different target angles)

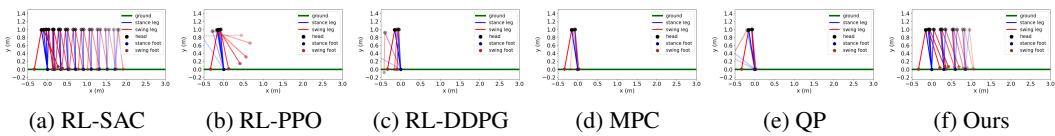


Figure 61: Bipedal walker simulation comparisons (different target angles)

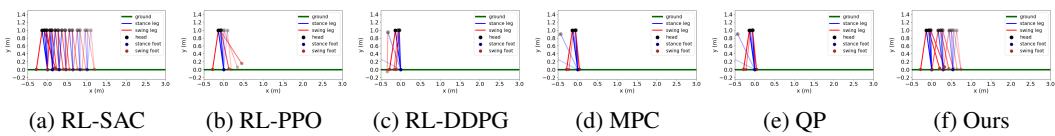


Figure 62: Bipedal walker simulation comparisons (different target angles)