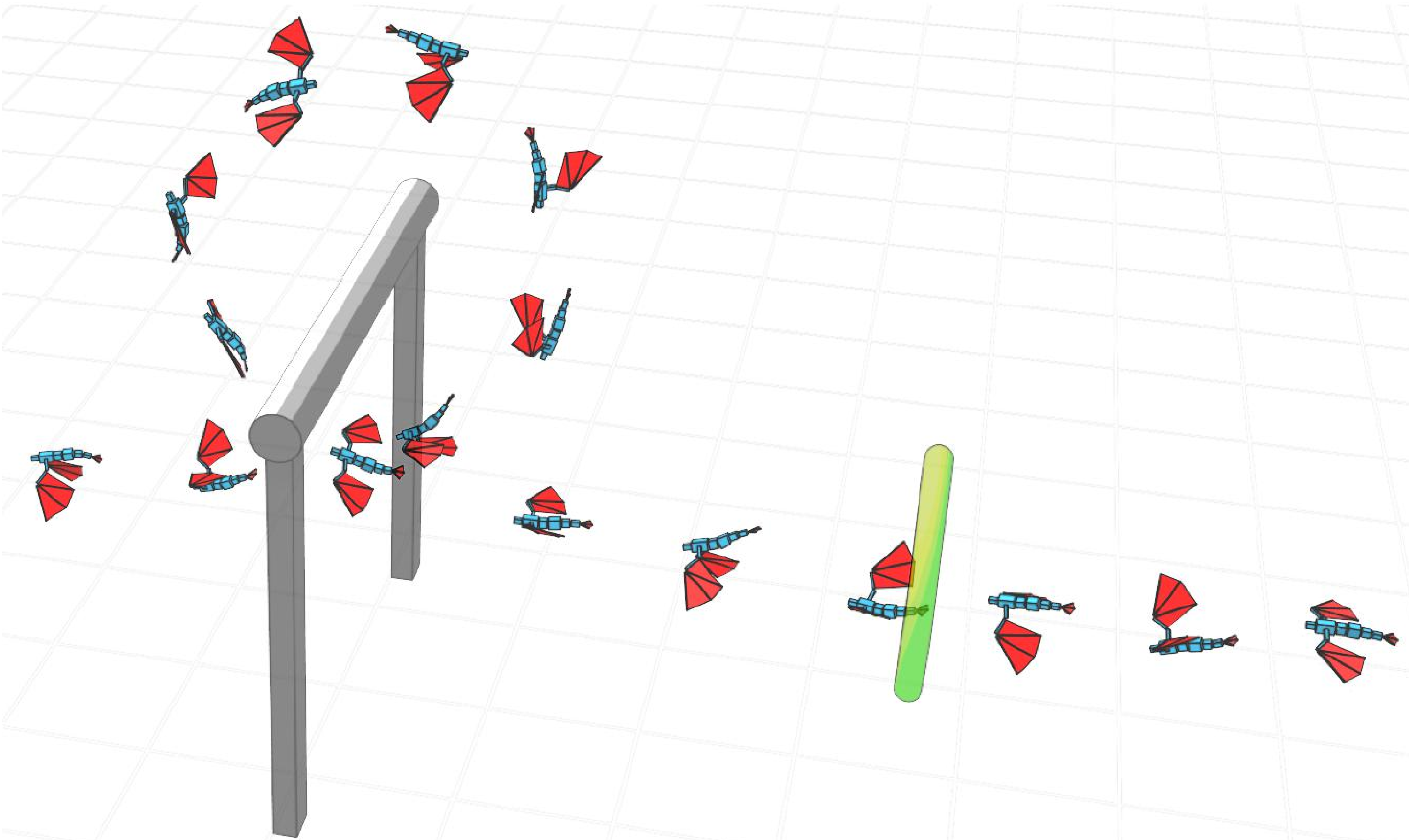
Aerobatics(特技飞行) Control of Flying Creatures via Self-Regulated Learning

JUNGDAM WON, Seoul National University, South Korea
JUNGNAM PARK, Seoul National University, South Korea
JEHEE LEE＊ , Seoul National University, South Korea

Target:

Designing physics-based controller for flying creature to track trajectory



Basic Ideas

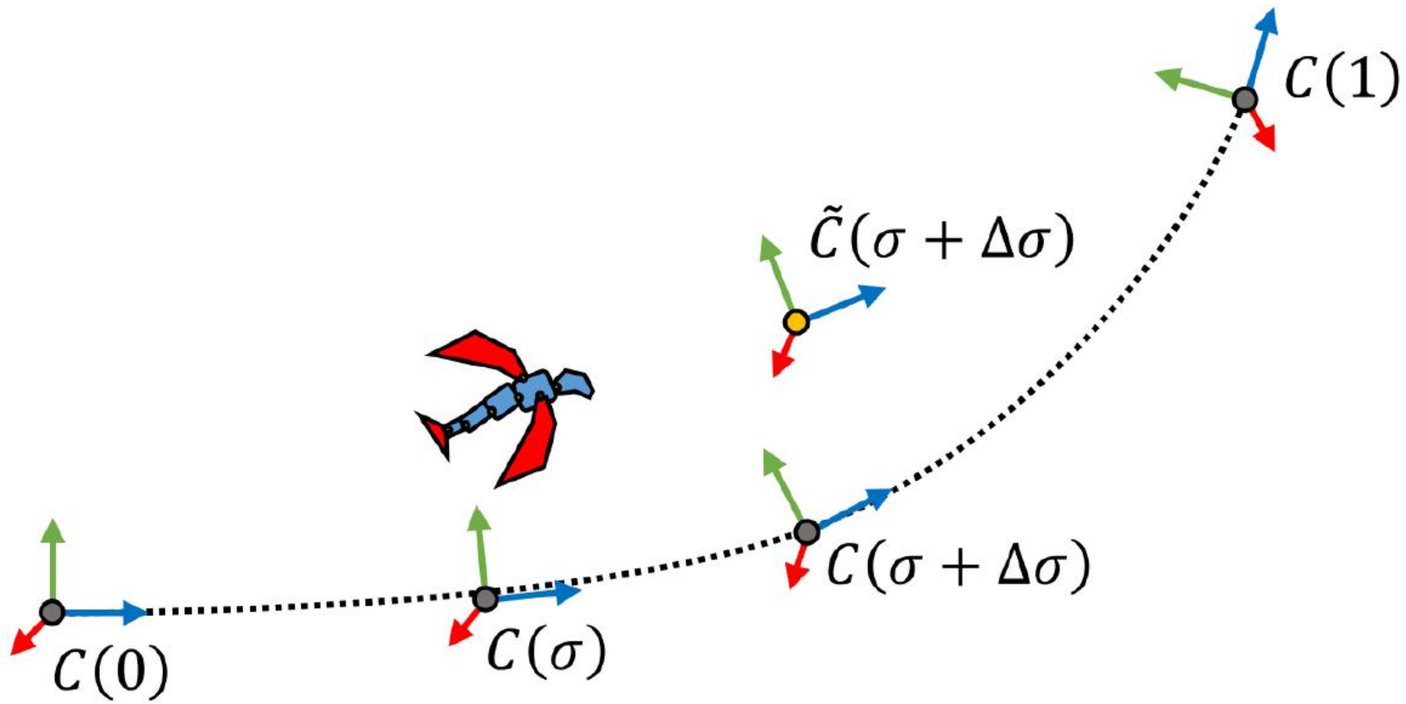
- Aerodynamics
- Reinforcement learning
- Discretization
- Neural network
- "Changeable rewards"

Aerodynamics

Trajectory: $C(\sigma) = (R(\sigma), p(\sigma), h(\sigma))$

$R(\sigma) \in SO(3)$ and $p(\sigma) \in R^3$

threshold: $d(R, p, \sigma^*) \; < \; h(\sigma^*)$



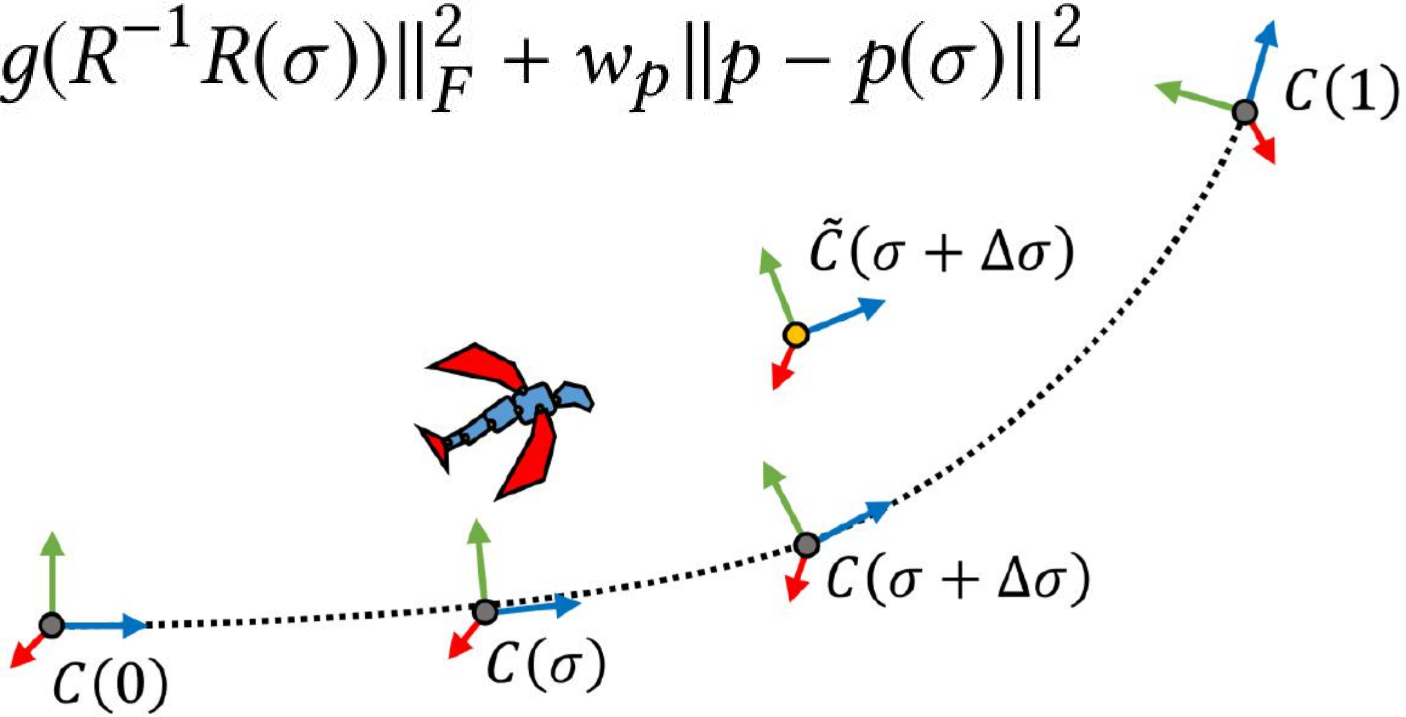
Aerodynamics

Trajectory: $C(\sigma) = (R(\sigma), p(\sigma), h(\sigma))$

$$R(\sigma) \in SO(3) \text{ and } p(\sigma) \in R^3$$

$$d(R, p, \sigma^*) < h(\sigma^*)$$

$$d(R, p, \sigma) = \|log(R^{-1}R(\sigma))\|_F^2 + w_p\|p - p(\sigma)\|^2$$



Aerodynamics

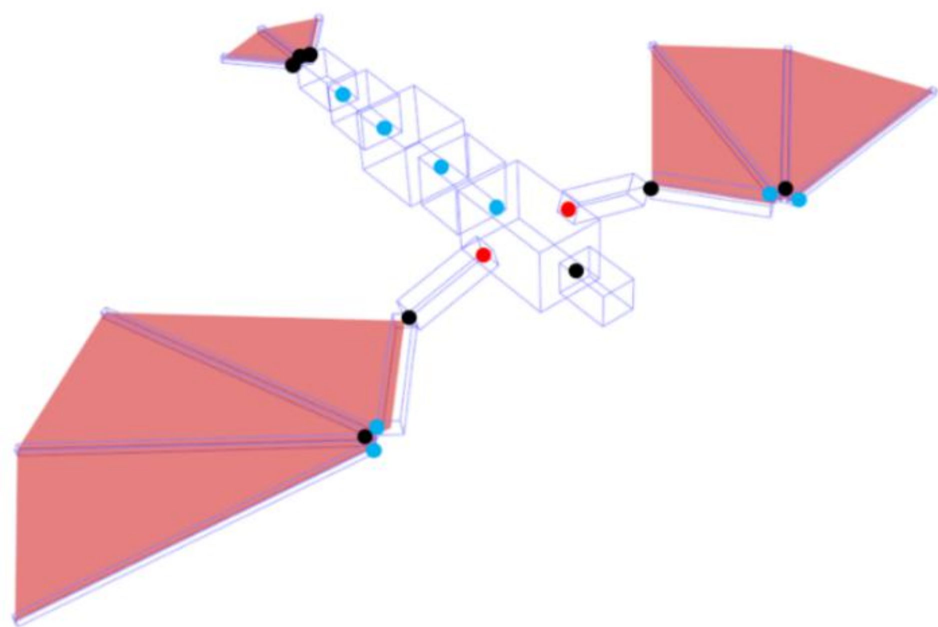
Dragon with bird-like articulated(用关节连接的) wings

Dynamics state:

$$s_d = (q, \dot{q})$$

$$q = (q_1, \cdots, q_D)$$

$$\dot{q} = (\dot{q}_1, \cdots, \dot{q}_D)$$



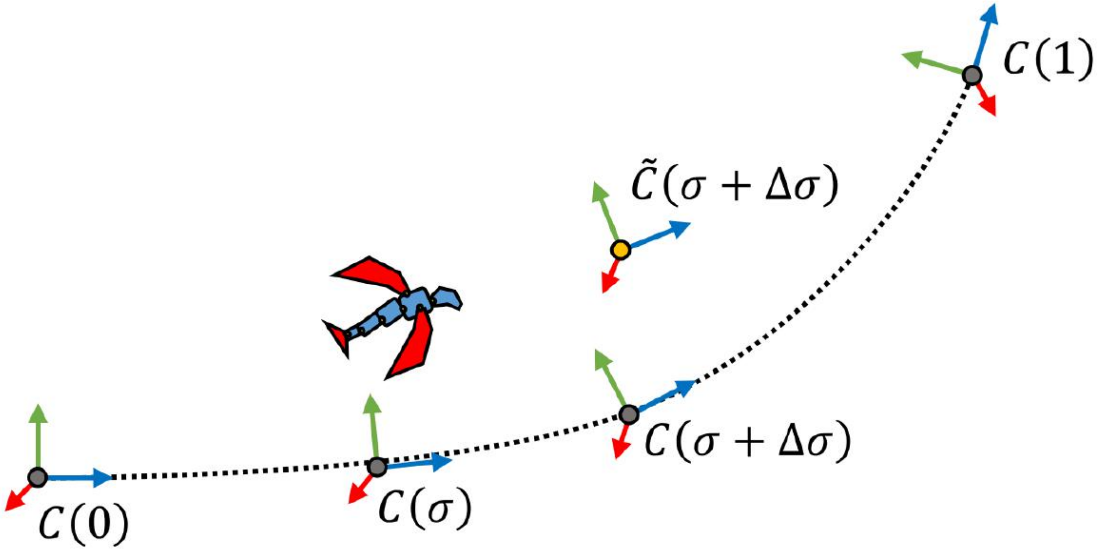
I think this should satisfy some constrains.

Aerodynamics

Dragon with bird-like articulated(用关节连接的) wings

Sensory state:

$$s_S = \big(C(\sigma), C(\sigma+\epsilon), \cdots, C(\sigma+w\epsilon)\big)$$

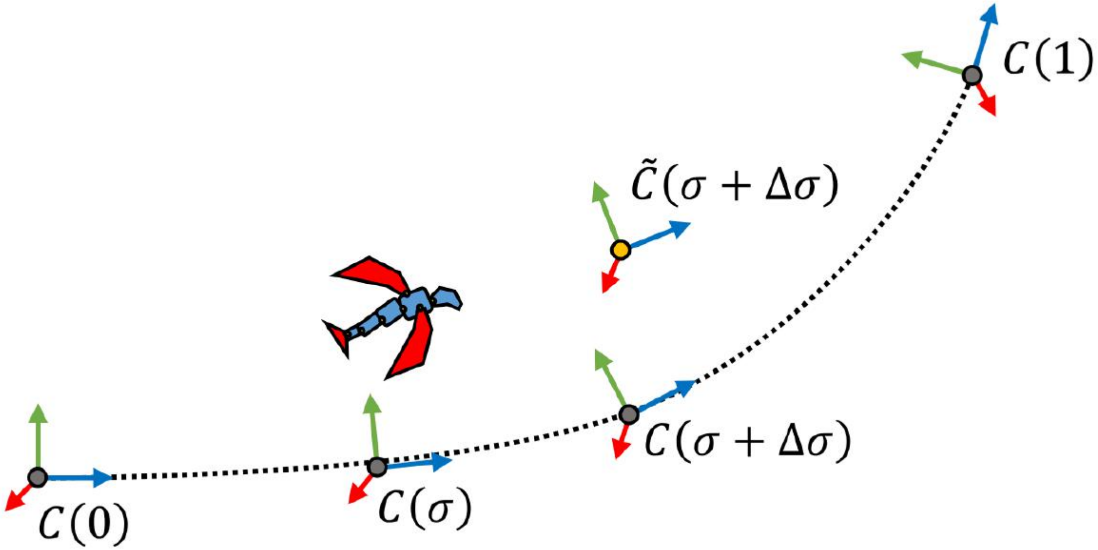


Aerodynamics

Dragon with bird-like articulated(用关节连接的) wings

Sensory state:

$$s_S = \big(C(\sigma), C(\sigma + \epsilon), \cdots, C(\sigma + w\epsilon)\big)$$



Controller: Reinforcement learning

Algorithm 1 DRL Algorithm

$Q|_{\theta_Q}$: state-action value network
$\pi|_{\theta_\pi}$: policy network
B : experience replay memory
repeat
  $s_0 \leftarrow$ random initial state
  for $i = 1, \cdots, T$ **do**
    $a_i \leftarrow \pi(s_{i-1})$
    if $\mathrm{unif}(0,1) \leq \rho$ **then**
      $a_i \leftarrow a_i + \mathcal{N}(\mathbf{0}, \Sigma)$
    $s_i \leftarrow \mathrm{StepForward}(s_{i-1}, a_i)$
    $r_i \leftarrow \mathcal{R}(s_{i-1}, a_i, s_i)$
    $e_i \leftarrow (s_{i-1}, a_i, r_i, s_i)$
    Store e_i in B
  $X_Q, Y_Q \leftarrow \emptyset$
  $X_\pi, Y_\pi \leftarrow \emptyset$
  for $i = 1, \cdots, N$ **do**
    Sample an experience tuple $e = (s, a, r, s')$ from B
    $y \leftarrow r + \gamma Q(s', \pi(s'|\theta_\pi)|\theta_Q)$
    $X_Q \leftarrow X_Q \cup \{(s, a)\}$
    $Y_Q \leftarrow Y_Q \cup \{y\}$
    if $y - Q(s, \pi(s|\theta_\pi)|\theta_Q) > 0$ **then**
      $X_\pi \leftarrow X_\pi \cup \{s\}$
      $Y_\pi \leftarrow Y_\pi \cup \{a\}$
  Update Q by (X_Q, Y_Q)
  Update π by (X_π, Y_π)
until no improvement on the policy

Algorithm 2 Step forward with self-regulation

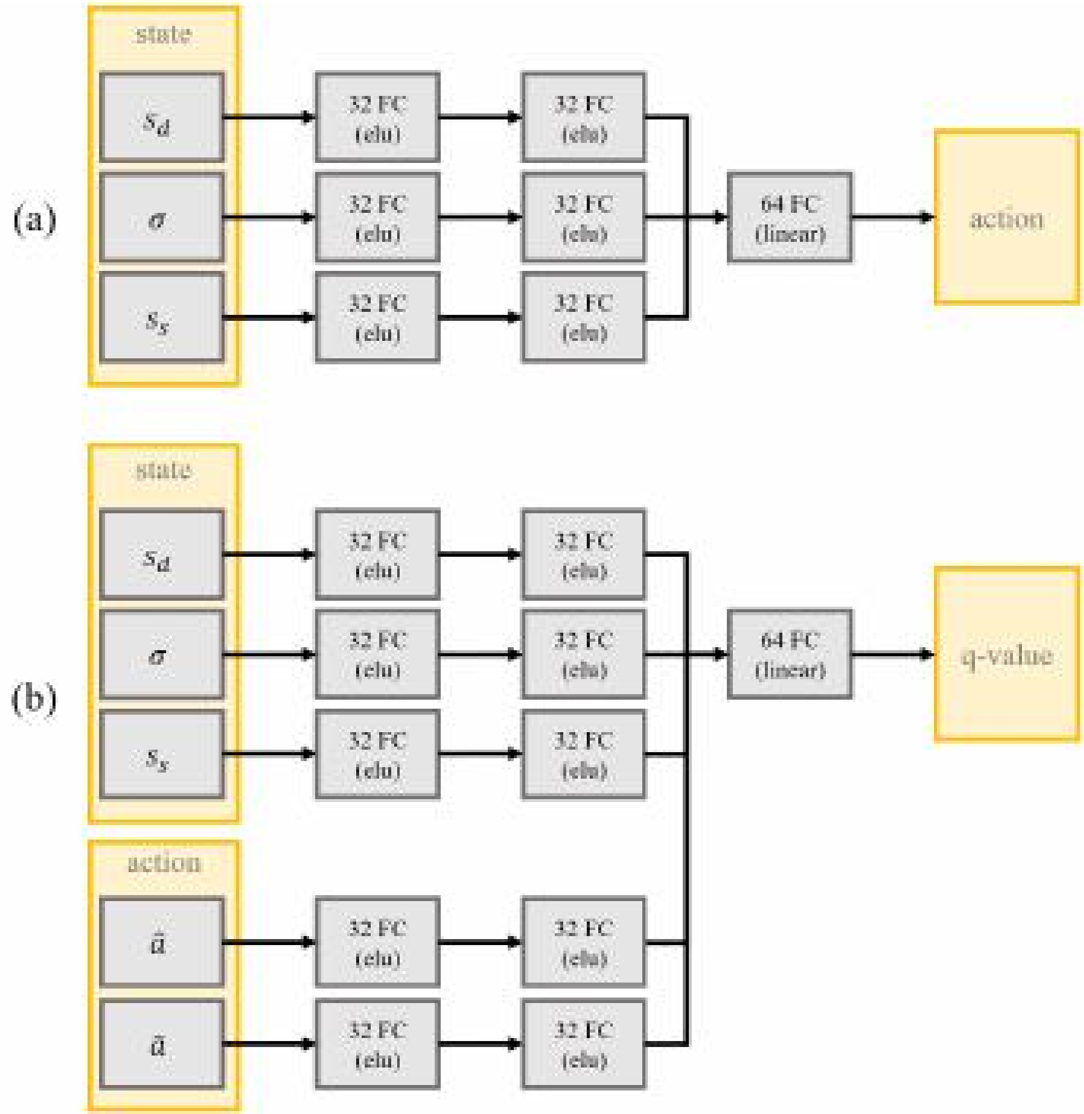
s : the current state
$a = (\hat{a}, \tilde{a})$: the action determined by the current policy
$\tilde{a} = (\Delta\sigma, \Delta R, \Delta p, \Delta h)$: a self-regulation part of the action

procedure StepForwardWithSRL(s, a)
  $\sigma \leftarrow \sigma + \Delta\sigma$
  $\tilde{R} \leftarrow R(\sigma)\Delta R$
  $\tilde{p} \leftarrow p(\sigma) + R(\sigma)\Delta p$
  $\tilde{h} \leftarrow h(\sigma) + \Delta h$
  $s' \leftarrow$ Dynamic simulation with $\hat{a}$
  $r \leftarrow$ Compute $\mathcal{R}(s, a, s')$ with progress σ and target $(\tilde{R}, \tilde{p}, \tilde{d})$

$$a = (\hat{a}, \tilde{a}) \quad \hat{a} = (\hat{q}, \tau)$$

$$\tilde{a} = (\Delta\sigma, \Delta R, \Delta p, \Delta h)$$

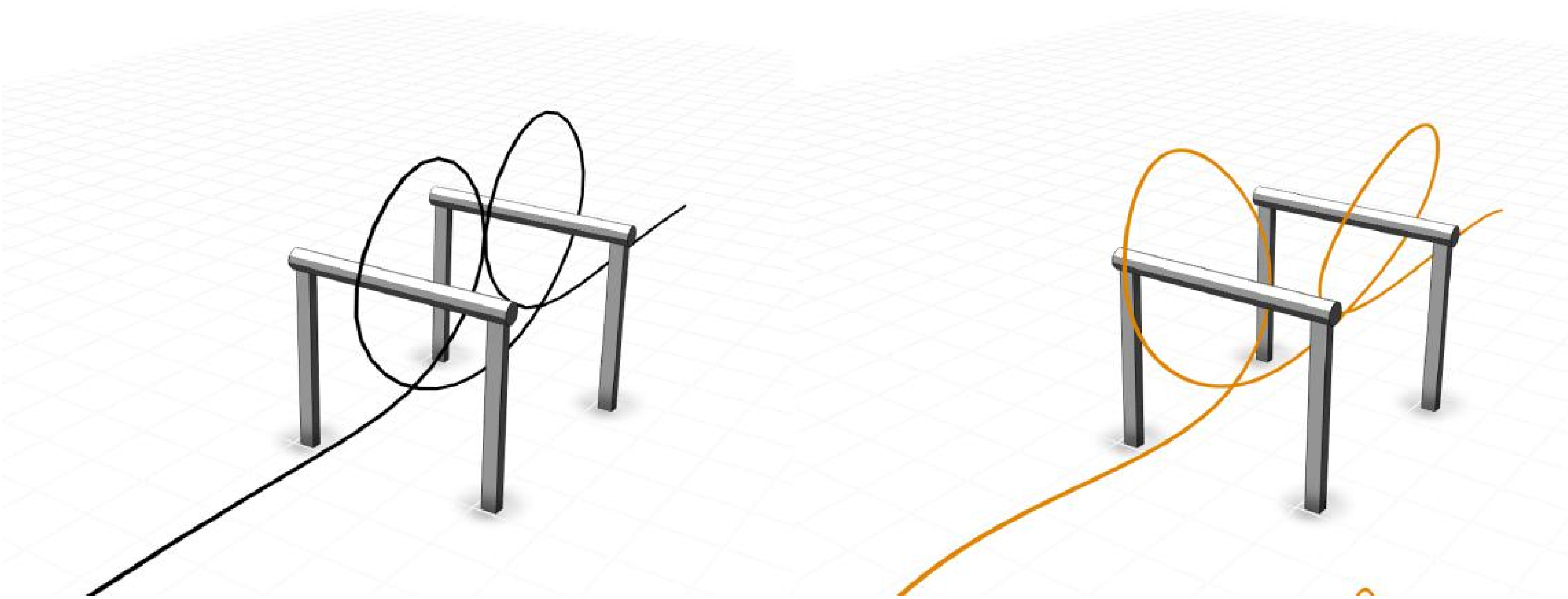
$$\tilde{R}(\tilde{\sigma}) = R(\tilde{\sigma})\Delta R,$$
$$\tilde{p}(\tilde{\sigma}) = p(\tilde{\sigma}) + R(\tilde{\sigma})\Delta p,$$
$$\tilde{h}(\tilde{\sigma}) = h(\tilde{\sigma}) + \Delta h,$$



(a)
state
s_d
σ
s_s
32 FC (elu)
64 FC (linear)
action
(b)
state
s_d
σ
s_s
action
$\bar{a}$
$\tilde{a}$
32 FC (elu)
64 FC (linear)
q-value

Table 1. Simulation and learning parameters

Simulation time step	0.001
Control time step	≈ 0.2
Policy learning rate (π)	0.0001
Value learning rate (Q)	0.001
Discount factor (γ)	0.95
Exploration probability (ρ)	0.5
Exploration noise (Σ)	$0.05I$
Maximum time horizon (sec)	50
Action range (normalized)	± 10
State range (normalized)	± 10
w_p	0.005
w_h	0.001
$\bar{h}$	20.0
d_{max}	3.0
W	0.02



Algorithm	X-turn	Y-turn	XY-turn	Double X-turn	Ribbon	Z-turn	Zigzag	Infinite X-turn	Combination
Default	2304.2 (12137)	1815.6 (10401)	1644.9 (13428)	14201 (79039)	8905.4 (42555)	2180.2 (11043)	1046.2 (4348.9)	36182 (107998)	48869 (250762)
Closest	**28.193*** (**132.4**)	162.10* (**461.81**)	274.48* (1266.9)	**35.891*** (**145.72**)	146.46* (739.98)	152.68 (1846.5)	175.46* (609.79)	942.81 (5653.4)	9050.0 (54705)
SRL	30.235* (177.93)	**115.89*** (516.43)	**114.77*** (**531.96**)	39.18* (232.25)	**131.47*** (**484.56**)	**67.479*** (**228.965**)	**137.70*** (**500.29**)	**136.82*** (**1456.8**)	**264.82*** (**988.96**)

Basic Ideas

- Aerodynamics for physics simulation
- Reinforcement learning for controller policy
- Discretization, Neural network for continuous, high dimesion value function & policy function
- "Changeable rewards" for unrealizable trajectory