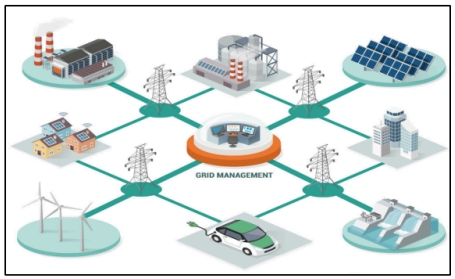
Safe Stabilization Control for Interconnected Virtual-Real Systems via Model-based Reinforcement Learning

Junkai Tan, Shuangsi Xue, Huan Li, Hui Cao, Dongyu Li

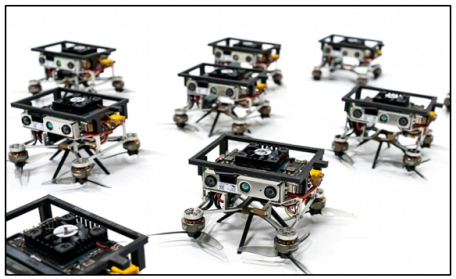
EE, Xi'an Jiaotong University.
CST, Beihang University.

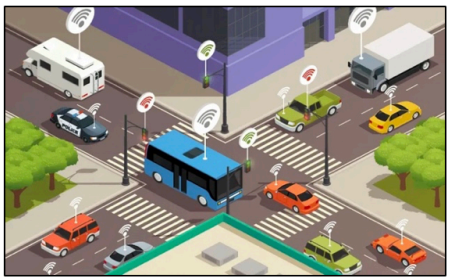
2024.7.8

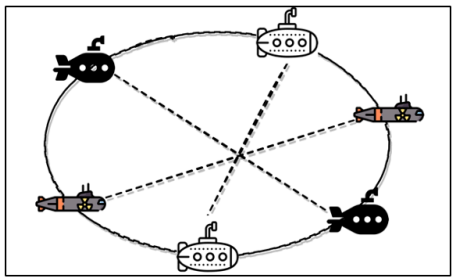
Background



Smart grid


Quadcopter swarm


Smart transportation


USV formation

Complex interconnected systems are widespread in today's social activities as well as in industrial production. These systems usually contain multiple controllers and sensors, and thus are highly nonlinear and coupled.

1. The original centralised control strategy is difficult to be further extended to complex systems with more interconnections.
2. Distributed control strategies are more convenient and effective for nonlinear interconnected systems than the traditional centralised control strategies.

Background

- Dynamic Programming (DP) is a method of forward recursion to future moments and backward solving to obtain the optimal policy. However, there is the problem of "dimensional disaster" of computational explosion.
- Adaptive Dynamic Programming (ADP) is an improvement of DP, which can effectively avoid the "dimension disaster" with the powerful fitting approximation ability of neural network.

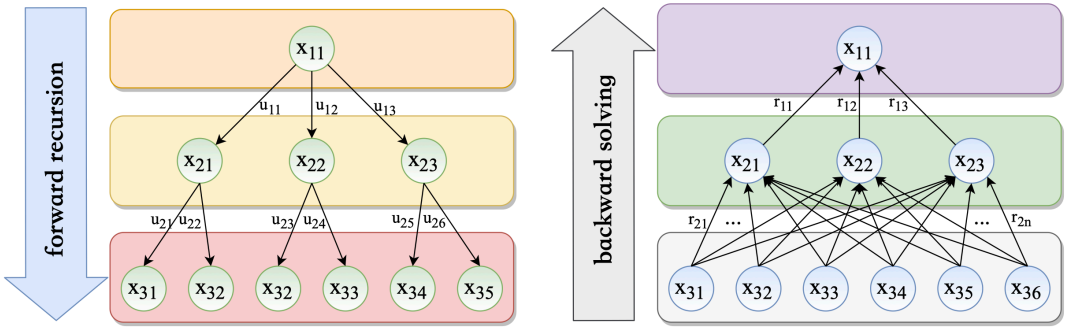


Fig: Process of DP

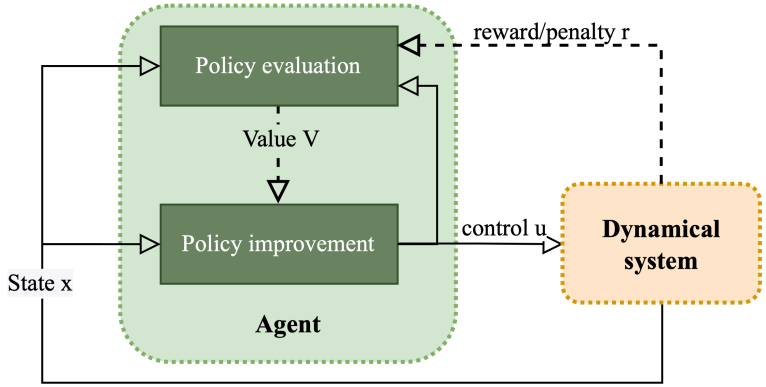


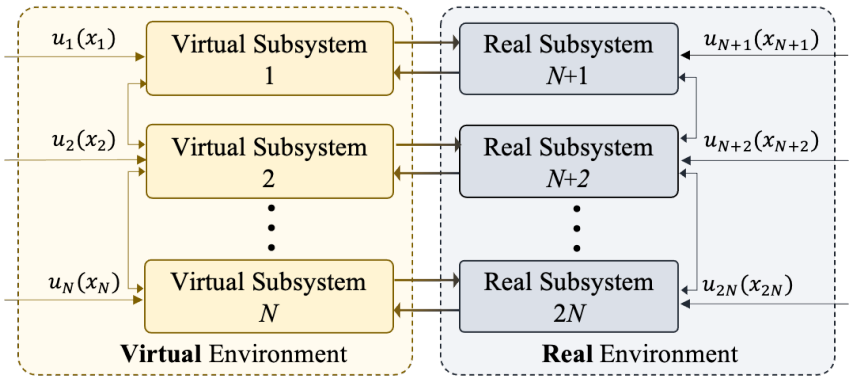
Fig: Scheme of ADP

Problem Statement: virtual-real system

Considering a novel type of interconnected systems, defined as virtual-real systems with $2N$ subsystems:

$$\dot{x}_i = f_i(x_i) + g_i(x_i)\left(u_i(x_i) - \mathbf{I}(x)\right), \quad i = 1, ..., 2N, \tag{1}$$

where $x_i \in \mathbb{R}^n$ are the state of each subsystem, $u_i(x_i) \in \mathbb{R}^n$ are the control input of the i th subsystem, $x = [x_1, \ldots, x_{2N}]$ is a vector of all the subsystem states, and $\mathbf{I}(x)$ is the interconnected term.



1. $[x_1, ..., x_N]$ as the virtual part;
2. $[x_{N+1}, ..., x_{2N}]$ as the real part.

Problem Statement: forward invariance

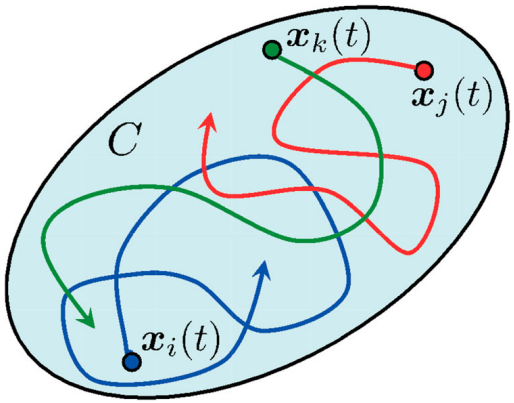
To design safe and efficient controller, give the definition of safety:

Definition 1: (Forward invariant)

For a pre-defined period $\mathcal{I}(x_0)$ and any $x_0 \in \mathbf{s}$, if system dynamic satisfies $x(t) \in \mathbf{s}$ for any $t \in \mathcal{I}(x_0)$. We define set s as a forward invariant set and can be separated as interior part and boundary part as

$$\mathbf{s} = \{x \in \mathbb{R}^n \mid h(x) \geq 0\},$$
$$\partial\mathbf{s} = \{x \in \mathbb{R}^n \mid h(x) = 0\},$$
$$\mathrm{Int}(\mathbf{s}) = \{x \in \mathbb{R}^n \mid h(x) > 0\},$$

where $h \in \mathbb{R}^n$ is the boundary function, which vanishes in the boundary of s.

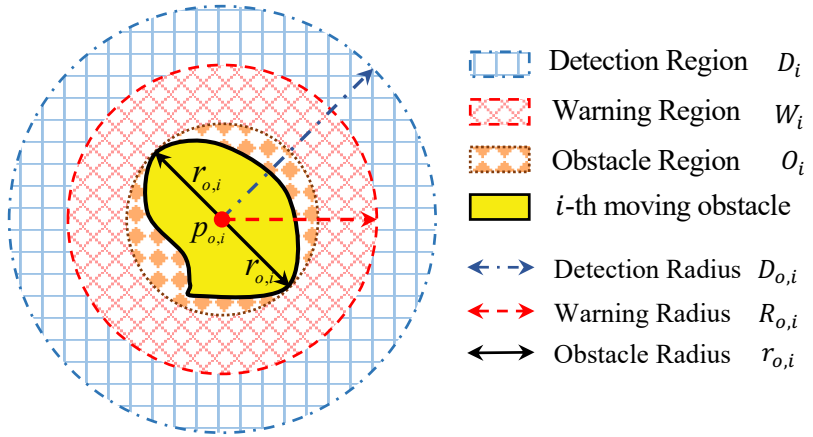


$x(t) \in \partial\mathbf{s},\ \forall t \in \mathcal{I}(x_0) \rightarrow$ safe operating region.

Problem Statement: unsafe obstacle modeling

To construct safe region, unsafe obstacle are defined:

1. Detection region $\mathcal{D}_i$: Triggering execution of safe strategy.
2. Warning region $\mathcal{A}_i$: Buffer for possible unsafe collisions.
3. Obstacle region $\mathcal{O}_i$: Colliding with the obstacle in this region.



Design function $s_i(x)$ to characterize the regions of obstacle $x_{o,i}$:

$$s_i(x) = \begin{cases} 0, & d_{o,i} > D_{o,i}, \\ l_1 + l_1 \cos(\pi \frac{d_{o,i}^2 - R_{o,i}^2}{D_{o,i}^2 - R_i^2}), & R_{o,i} < d_{o,i} \leqslant D_{o,i}, \\ l_2 + l_3 \cos(\pi \frac{d_{o,i}^2 - r_{o,i}^2}{R_{o,i}^2 - r_{o,i}^2}), & r_{o,i} < d_{o,i} \leqslant R_{o,i}, \\ 1, & d_{o,i} \leqslant r_{o,i}, \end{cases} \quad \begin{cases} l_2 + l_3 = 1, \\ l_2 - l_3 = 2l_1 \end{cases}$$

Problem Statement: barrier function

Definition 2: (barrier function)

A continuous function $b(x)$ is called a barrier function with the following properties:

1) $b(x)$ isn't infinite when $x(t) \in \mathrm{Int}(s)$, that is, $|b(x)| < \infty$.
2) As state x approaches the boundary of forward invariant set, function $b(x)$ goes to infinity, expressed as $\lim_{z \to \partial s} b(x) = \infty$.
3) The equilibrium value of $b(x)$ vanishes, that is, $b(0) = 0$.

Then consider the barrier function in the following form:

$$b(x) = \frac{K_B s(x)}{h(x)} \tag{2}$$

where $h(x) = \sum_{i \in \mathcal{M}} h_i(x)$ and $s(x) = \sum_{i \in \mathcal{M}} s_i(x)$ are continuously differentiable smooth functions, K_B is positive constants.

Main Results: Optimal Controller Design

Considering $2N$ isolated subsystems corresponding to system (1):

$$\dot{x}_i(t) = f_i\left(x_i(t)\right) + g_i\left(x_i(t)\right) u_i\left(x_i(t)\right), \quad i = 1, 2, \ldots, 2N \tag{3}$$

To deal with the optimal control problem, the design objective is to find the controller $u_i\left(x_i\right)$ that minimizes the cost function as:

$$J_i(x_i, u(\cdot)) = \int_0^\infty Q_i^2(x_i(\tau)) + u_i^T\left(x_i(\tau)\right) R_i u_i(x_i(\tau)) d\tau \tag{4}$$

where $Q_i\left(x_i\right)$ is a positive definite function with $q_i\left(x_i\right) \leq Q_i\left(x_i\right)$. For any set of control policies $\mu_i \in \Phi_i(\Omega_i)$, $i = 1, \ldots, 2N$ that can achieve the aforementioned objective, if the associated cost function $\hat{J}_i(x_i, \mu(\cdot))$ is continuously differentiable, which given as:

$$\hat{J}_i(x_i, \mu(\cdot)) = \int_0^\infty Q_i^2(x_i(\tau)) + \mu_i^T\left(x_i(\tau)\right) R_i \mu_i(x_i(\tau)) d\tau. \tag{5}$$

Main Results: Optimal Controller Design

Then the Hamiltonian function for each isolated subsystem is obtained by taking derivatives on both sides of (5) as:

$$\begin{aligned} H_i(x_i, \mu_i, \nabla \hat{J}_i(x_i, \mu_i)) =& Q_i^2(x_i) + \mu_i^T(x_i) R_i \mu_i(x_i) \\ &+ (\nabla \hat{J}_i(x_i))^T (f_i(x_i) + g_i(x_i)\mu_i(x_i)) \end{aligned} \tag{6}$$

where $i = 1, \ldots, 2N$. The optimal cost function for each subsystem can be expressed as:

$$V_i = \min_{\mu_i \in \Phi_i(\Omega_i)} \int_0^\infty \{Q_i^2(x_i(\tau)) + \mu_i^T(x_i(\tau)) R_i \times \mu_i(x_i(\tau))\} d\tau, \tag{7}$$

where $i = 1, \ldots, 2N$ and $J_i^*(x_i)$ satisfies the HJB equation $0 = \min_{\mu_i \in \Psi_i(\Omega_i)} H_i(x_i, \mu_i, \nabla V_i)$, with $V_i = \partial V_i / \partial x_i$.

Main Results: Optimal Controller Design

The optimal control policy for the $2N$ subsystems can then be derived as follows:

$$u_i^*(x_i) = \arg\min_{\mu_i \in \Psi_i(\Omega_i)} H_i(x_i, \mu_i, \nabla V_i) = -\frac{1}{2} R_i^{-1} g_i^T(x_i) \nabla V_i.$$

To establish a stable control scheme for the interconnected system (1), here we present the following lemma:

lemma 1:

Considering the isolated subsystems in (3), the following feedback control law could ensure asymptotically stablity of $2N$ subsystems

$$\hat{u}_i = \beta_i u_i^*(x_i) = -\frac{1}{2} \beta_i R_i^{-1} g_i^T(x_i) \nabla \hat{J}_i(x_i), \tag{8}$$

where $i = 1, \ldots, 2N$ and β_i are positive constants satisfies $\beta_i \geq \frac{1}{2}$.

Safe-Guarding Control Design

The safety-guarding term of the real part is introduced as the safety controller:

$$u_b^*(x_i) = -\alpha_i g_i^T(x_i)\nabla b^T(x_i),\ i = N+1,\ldots,2N, \tag{9}$$

where α_i is the selected control gain and $b(x)$ is the barrier function as we defined in (2).
The feedback control for the real part are rewritten into:

$$u_i = \beta_i u_i^*(x_i) + \gamma_i u_b^*(x_i),\ i = N+1,\ldots,2N, \tag{10}$$

where γ_i are positive constants set by users. By choosing the appropriate γ_i and β_i, the designed controller is sufficient to make the interconnected system achieve asymptotic stability. Due to our safety-guaranteed term, the real part of the system can operate within set $\mathbb{s}$, that is, always in a safe region.

Main Results: Concurrent Learning

To obtain the analytical solution of control policies u_i and value functions V_i, we utilize a single critic network for the approximation of value function V_i, which in the form of

$$V_i = W_i^T \phi(x) + \epsilon_i(x). \quad (11)$$

The estimated approximation of the ideal value function V_i is defined as

$$\hat{V}_i = \hat{W}_i^T \phi(x). \quad (12)$$

The approximation of control u_i is implemented through the single network method as:

$$u_i = -\frac{1}{2} R_i^{-1} g_i^T (\nabla \phi_i^T(x) W_i \nabla \epsilon_i^T(x)). \quad (13)$$

With the estimated value's gradient using the weights W_i in (12), the actual controller is:

$$\hat{u}_i = -\frac{1}{2} R_i^{-1} g_i^T \nabla \phi_i^T(x) \hat{W}_i. \quad (14)$$

Main Results: Concurrent Learning

Based on (13), and (14), we can define the error of approximating the Hamilton-Jacobi equation as

$$\begin{aligned} H_i(x, u_i, W_i) =& u_i^T R_i u_i + Q_i^2 + [W_i^T \nabla \phi_i + (\nabla \epsilon_i)^T](f_i + g_i u_i), \\ =& -\nabla \epsilon_i^T (f_i + g_i u_i), \end{aligned} \tag{15}$$

$$\begin{aligned} H_i(x, \hat{u}_i, \hat{W}_i) =& \hat{u}_i^T R_i \hat{u}_i + Q_i^2 + (\hat{W}_i^T \nabla \phi_i)(f_i + g_i \hat{u}_i) \\ =& e_i. \end{aligned} \tag{16}$$

Then we can obtain the adaptation law based on least squares for the estimated critic network weight $\hat{W}_i$ as follows.

$$\dot{\hat{W}}_i = -a_i \frac{\partial E_i}{\partial \hat{W}_i} = -a_i \frac{\omega_i e_i}{\left(1 + \omega_i^T \omega_i\right)^2} - a_i \sum_{k=1}^{M} \frac{\omega_i^k e_i^k}{\left(1 + (\omega_i^k)^T \omega_i^k\right)^2}, \tag{17}$$

where a_i is the learning gain for each subsystem, determine the convergence speed of each single network weight W_i of the subsystem.

Safe RL tracking control: Stability Theory

Theorem1: Asymptotic stability

NN weights are asymptotically stable as following conditions are met:

$$\begin{cases} \overline{g}_i \overline{\phi}_j < 0 \\ \rho < 0 \\ \beta_i \left(\frac{p+1}{2} - 2\lambda_{\min}(\Gamma_k) \right) < 0 \end{cases} \tag{18}$$

where $\rho = \sum_{i=1}^{N} \left[\beta_i \frac{p+1}{2} \overline{\varepsilon}_i^2 - \left(\overline{\omega}_i \overline{\phi}_i + \overline{\epsilon}_i \right) \sum_{j=1}^{N} \left(\frac{1}{2} G_j \overline{\phi}_i \|\hat{\omega}_j\| - g_i \overline{\epsilon}_i \right) \right]$

Proof: Set the Lyapunov function

$$V_L = \sum_{i=1}^{N} (V_i + V_{\omega,i}) \tag{19}$$

where $V_{\omega,i} = \frac{1}{2} \tilde{\omega}_i^{\mathrm{T}} \tilde{\omega}_i$

Safe RL tracking control: Stability Theory

According to the given assumptions, the following inequality holds:

$$\dot{V}_i \leq -r_i - \left(\overline{\omega}_i \overline{\phi}_i + \overline{\epsilon}_i\right) \sum_{j=1}^{N} \left(\frac{1}{2} G_j \overline{\phi}_i \|\hat{\omega}_j\| - g_i \overline{\epsilon}_i \right) \tag{20}$$

$$\dot{V}_{\omega,i} \leq \beta_i \left[\frac{p+1}{2} - 2\lambda_{\min}\left(\Gamma_k\right) \right] \|\tilde{\omega}_i\|^2 + \beta_i \frac{p+1}{2} \overline{\varepsilon}^2_{\mathrm{hmax},i} \tag{21}$$

$$\begin{aligned} \dot{V} \leq & -\sum_{i=1}^{N} r_i + \rho \\ & + \sum_{i=1}^{N} \left[\overline{g}_i \overline{\phi}_i + \beta_i \left(\frac{p+1}{2} - 2\lambda_{\min}(\Gamma_k) \right) \right] \|\tilde{\omega}_i\|^2 \end{aligned} \tag{22}$$

So $\dot{V}_L \leq 0$ holds, i.e., the asymptotic stability of the weights is proved

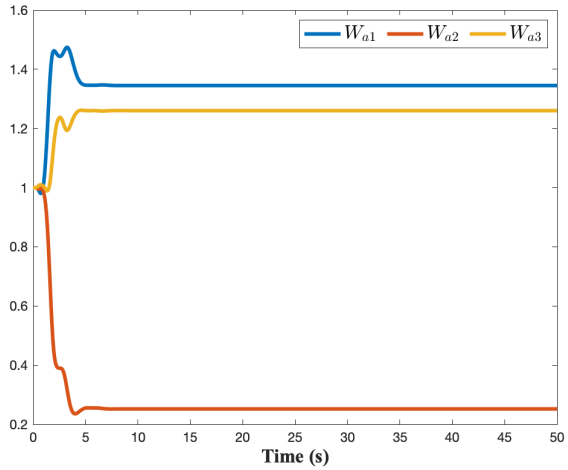
Simulation Verification

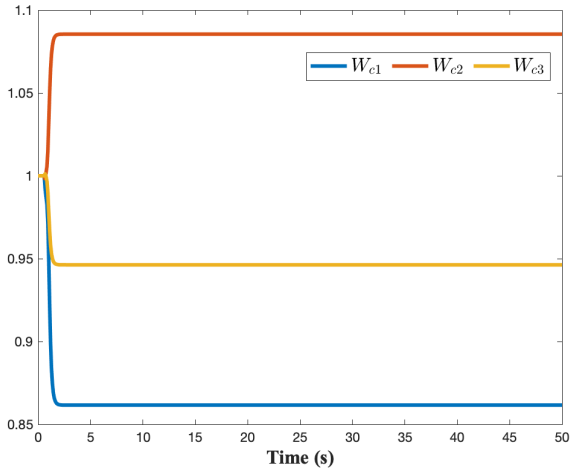
The system for numerical simulation is seleted as follow:

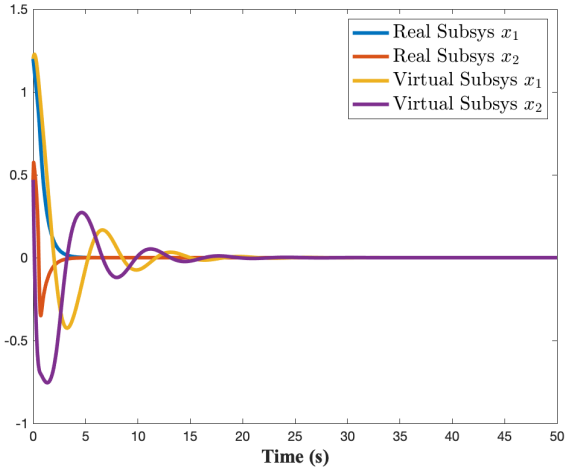
$$
\begin{aligned}
\dot{x}_1 = & \begin{bmatrix} -x_{11} + x_{12} \\ -0.5(x_{11} + x_{12}) - 0.5x_{12}\left(\cos\left(2x_{11}\right) + 2\right)^2 \end{bmatrix} \\
& + \begin{bmatrix} 0 \\ \cos\left(2x_{11}\right) + 2 \end{bmatrix} \left(u_1 + (x_{11} + x_{22}) \sin x_{12}^2 \cos\left(0.5x_{21}\right)\right), \\
\dot{x}_2 = & \begin{bmatrix} x_{22} \\ -x_{21} - 0.5x_{22} + 0.5x_{21}^2 x_{22} \end{bmatrix} \\
& + \begin{bmatrix} 0 \\ x_{21} \end{bmatrix} \left(u_2 + 0.5\left(x_{12} + x_{22}\right) \cos\left(e^{x_{21}^2}\right)\right),
\end{aligned}
\tag{23}
$$

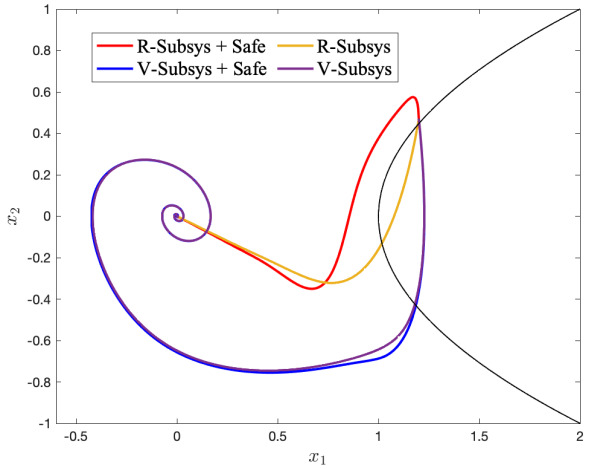
$x_1 = [x_{11}\ x_{12}]^T \in \mathbb{R}^2$ $\rightarrow$ the virtual subsystem state,
$x_2 = [x_{21}\ x_{22}]^T \in \mathbb{R}^2$ $\rightarrow$ real subsystem state,
u_1 $\rightarrow$ the virtual part control input, u_2 $\rightarrow$ the real part control input.

Simulation Verification









Hardware Experiment: Experiment Setup

The detailed experiment setup is listed as follows:

1. Operation platform: Rflysim motion capture OptiTrack.
2. Virtual part: Droneyee-X150 quadrotor.
3. Real part: 4-wheel drive car.
4. Control frequency & form : 30Hz, velocity command.

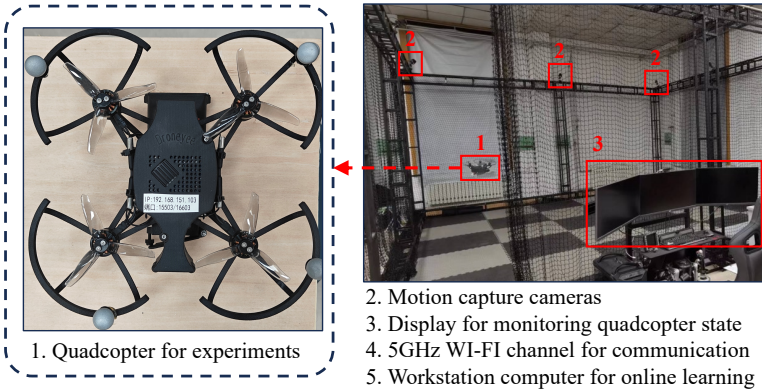


Fig 1: Droneyee-X150 quadrotor

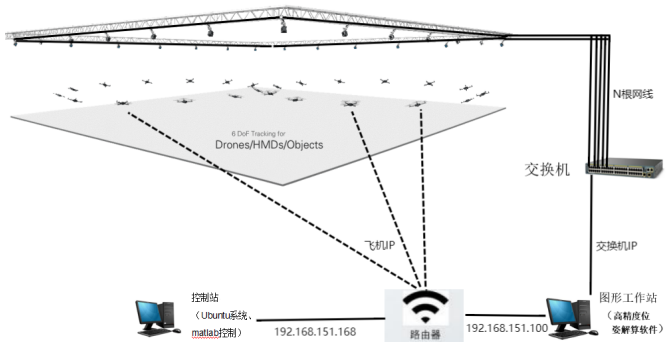


Fig 2: Motion capture OptiTrack

Hardware Experiment: Experiment Results

The learning process of the NN weight is presented in fig3. The control input to the quadrotor(virtual part) and car(real part) is showed in fig4.

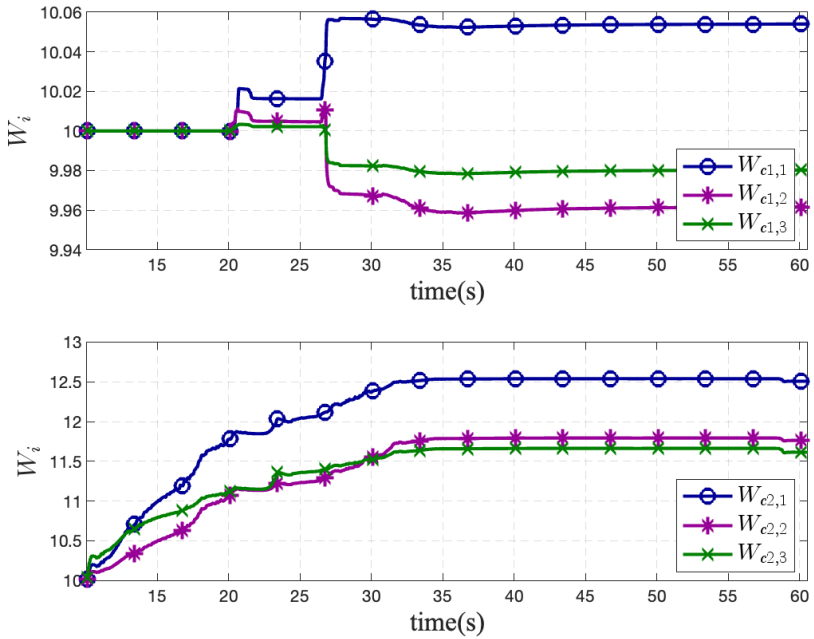


Fig 3: NN weight learning process

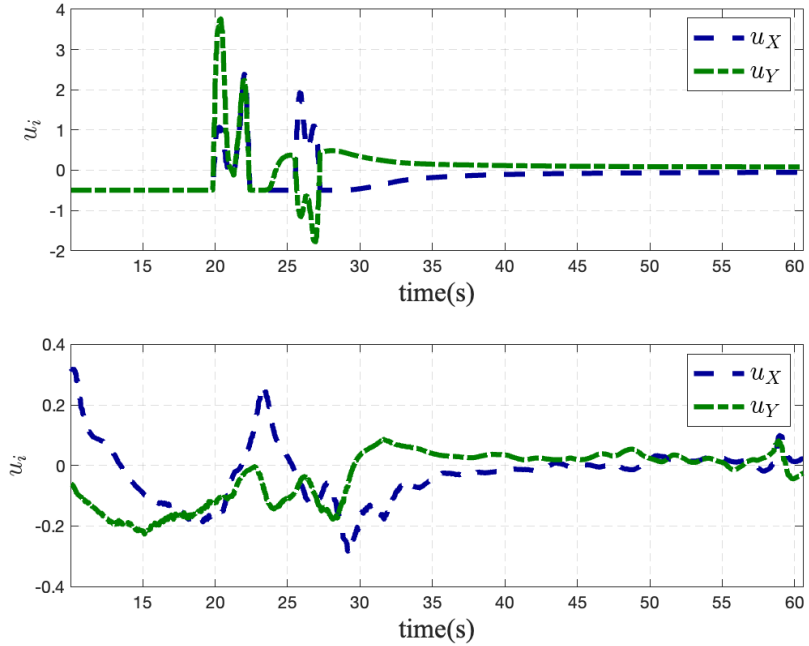


Fig 4: Control input

Experiment Results

The distance to the unsafe region and the value of function $s(x)$ is shown in fig5. The position of quadrotor(virtual part) and the 4WD car(real part) is shown in fig6.

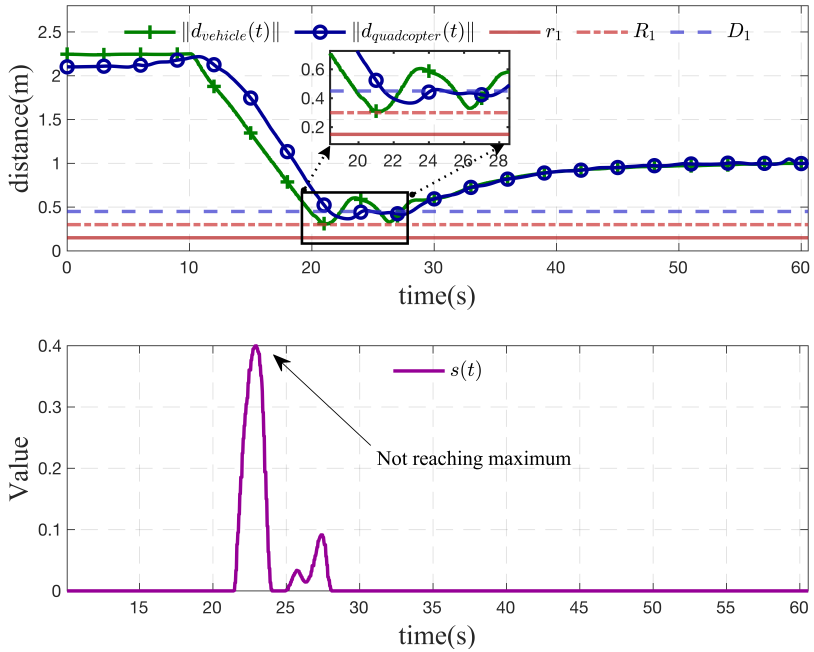


Fig 5

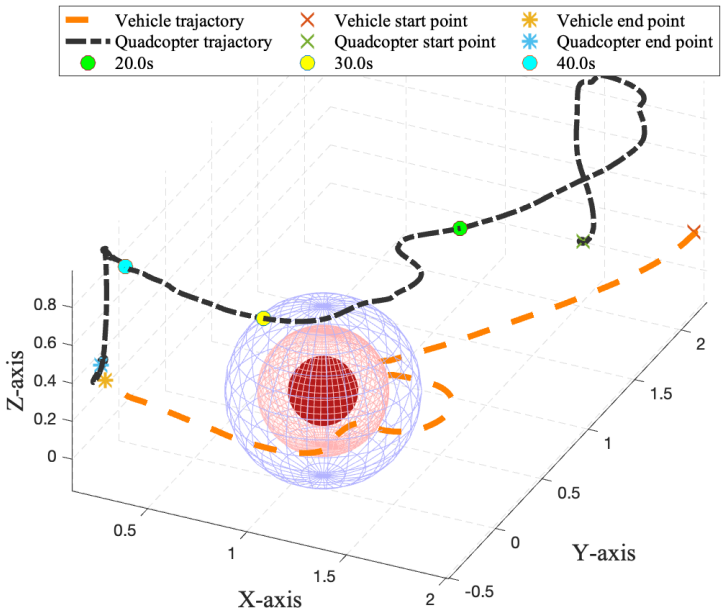


Fig 6

Experiment Results

The detailed process of safe control quadrotor(virtual part) and the 4WD car(real part) is illustrated in fig 7.

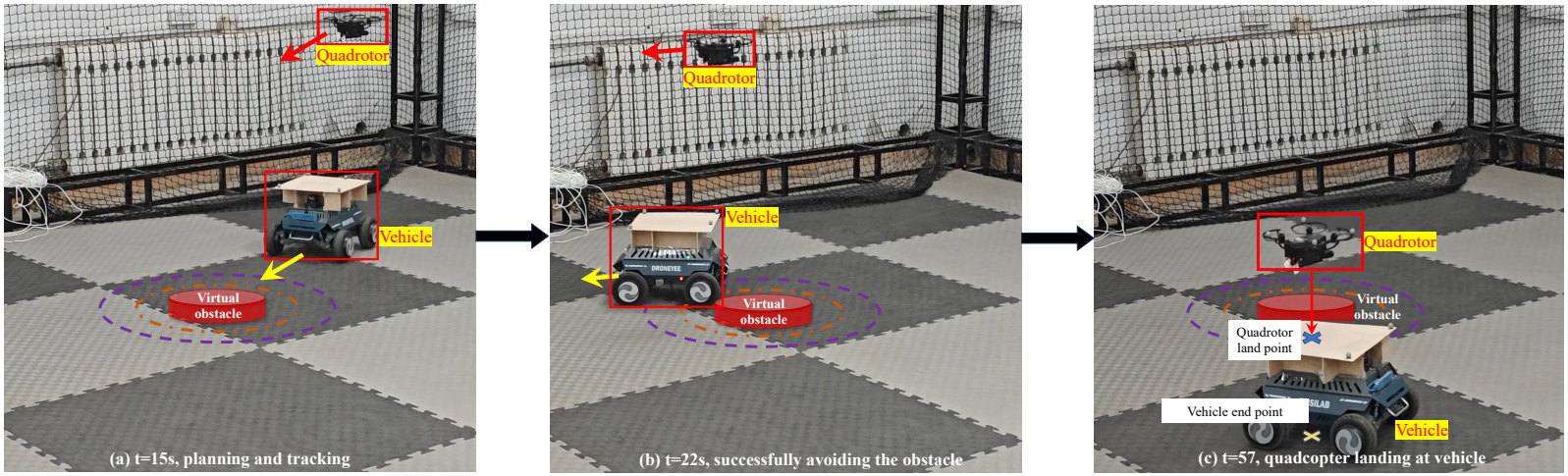


Fig 7: Snapshots of safe control quadrotor and car

Thanks!

Thanks for your listening