I. Developing the Intelligent Controllers

The process of developing the intelligent controller is presented in Fig. 1.

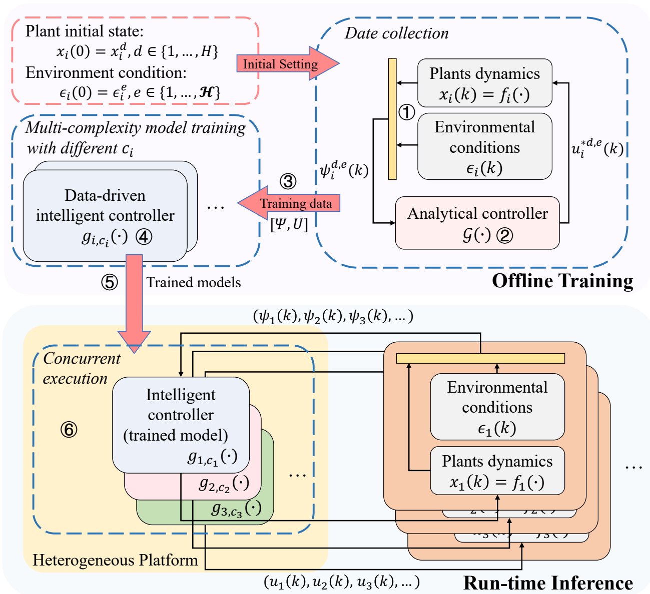

Fig. 1: Data-driven intelligent controller.

A. Data Collection

The foremost step to derive the intelligent control with DNN models $g_{i,c_i}(\cdot)$ is to collect the data offline. As shown in Fig. 1, the **first step** ① is to construct the plant dynamics $f_i(\cdot)$ according to the knowledge of physical plant dynamics as described in Sec. III-B of raw paper and environmental conditions ϵ_i, such as obstacles and noises. Therefore, we define augmented states $\psi_i = [x_i^\top, \epsilon_i^\top]^\top$, which considers the states of plants and the environment simultaneously. The **second step** ② is to derive the proper analytical controller $u_i^*(k) = \mathcal{G}_i(\psi_i(k))$. For the **third step** ③, we sample the tuples of $[\psi_i(k), u_i^*(k)]$ by traversing $x_i(k)$ under different environmental conditions ϵ_i with $\mathcal{G}_i(\cdot)$ in the loop, in order to cover comprehensive operating scenarios.

To be more specific, the process of data collection is as follows: 1) Generate $\mathcal{H}$ ($\mathcal{H} \in \mathbb{N}$) environmental conditions $\epsilon_i \in [\epsilon_i^1, \ldots, \epsilon_i^{\mathcal{H}}]$, and H ($H \in \mathbb{N}$) random initial plant states $x_i(0) \in [x_i^1, \ldots, x_i^H]$ to be sampled. 2) Run the simulations of the control systems initiated from initial states $x_i(0)$ for sampling interval T_s at a fixed control rate $1/T_i$. For each simulation interval with initial plant states $x_i(0) = x_i^d$, $d \in \{1, 2, \ldots, H\}$, and environmental condition $\epsilon_i(0) = \epsilon_i^e$, $e \in \{1, 2, \ldots, \mathcal{H}\}$, in the kth control period, we collect the augmented states $\psi_i^{d,e}(k)$ and the corresponding control command $u_i^{*d,e}(k)$ generated by $\mathcal{G}_i(\cdot)$. Thus, for each simulation interval T_s, we collect T_s/T_i pairs of labeled data $[\psi_i^{d,e}(k), u_i^{*d,e}(k)], k \in \{1, 2, 3, \ldots, T_s/T_i\}$. T_s is determined by the time constant of the control system. As a result, by traversing $\mathcal{H}$ environmental conditions and H initial plant states, we can get a training dataset $[\Psi, U]$ with $\mathcal{H}HT_s/T_i$ pairs of data to train the intelligent controller.

B. Model Training and Deployment

After collecting training data, for the **fourth step** ④, we train the data-driven model $g_{i,c_i}(\psi_i)$ with different complexity c_i to optimally regress $[\Psi, U]$ of $\mathcal{G}_i(\cdot)$. As we explored in Sec. III-C of raw paper, it is interesting to find that the relationship among the complexity and accuracy of $g_{i,c_i}(\psi_i)$, and its corresponding control performance are intertwined with physical plant dynamics, environmental conditions, and task execution on the heterogeneous platform. The selection of the complexity of $g_{i,c_i}(\psi_i)$ will be introduced in Sec. V of raw paper by comprehensively considering the above factors.

For the **fifth step** ⑤, we deploy $g_{i,c_i}(\psi_i)$ onto the heterogeneous platform, which raises the *critical question* of how to map the limited heterogeneous resource with the intelligent control task execution model, which we will address in Sec. IV-B of raw paper. For the **sixth step** ⑥, it comes to enable the runtime intelligent controllers (DNN inference tasks) on the heterogeneous platform, which raises another **critical question** of how to deal with the concurrent executions of multiple intelligent controllers efficiently to shorten the response time of certain intelligent controllers that is critical for the overall control performance. This question will be answered in Sec. V of raw paper. The intelligent controller receives $\psi_i(k)$ at time k and generate the control commands $g_{i,c_i}(\psi_i(k))$ by executing the kth job of control task τ_i, which computes $g_{i,c_i}(\psi_i(k))$.