Appendix

I. Proofs

A. Proof of Proposition 1

Proof. The stability proof is composed of two parts: the stability of the nominal system, and the stability of the error system between the actual system (12) and the nominal system (8). First, we prove the stability of the nominal system (8). By solving the MPC problem (9)-(10), we can obtain the optimal control sequence

$$\{c^*_{t|t}, c^*_{t+1|t}, \dots, c^*_{t+H-1|t}\} \tag{24}$$

and the resulting optimal state trajectory

$$\{\hat{\mu}^*_{t|t}, \hat{\mu}^*_{t+1|t}, \dots, \hat{\mu}^*_{t+H-1|t}, \hat{\mu}^*_{t+H|t}\} \tag{25}$$

at instant t. By appending the control signal that is produced by the feedback controller $K\hat{\mu}^*_{t+H|t}$ to (24), a suboptimal solution at the next time step $t+1$ is given by

$$\{c^*_{t|t}, c^*_{t+1|t}, \dots, c^*_{t+H-1|t}, K\hat{\mu}^*_{t+H|t}\} \tag{26}$$

and

$$\{\hat{\mu}^*_{t|t}, \hat{\mu}^*_{t+1|t}, \dots, \hat{\mu}^*_{t+H-1|t}, \hat{\mu}^*_{t+H|t}, A_K\hat{\mu}^*_{t+H|t}\}, \tag{27}$$

where $A_K := A + BK$ denotes the closed-loop transition matrix. Based on this suboptimal solution, we can prove that the optimal value function $V^*(\hat{\mu}_t)$ decreases along the trajectory. Since (26) and (27) are suboptimal, one has

$$V(\hat{\mu}_{t+1}) = \sum_{k=1}^{H-1} q(\hat{\mu}^*_{t+k|t}, c^*_{t+k|t}) + q(\hat{\mu}^*_{t+H|t}, K\hat{\mu}^*_{t+H|t}) \tag{28}$$

$$+p(A_K\hat{\mu}^*_{t+H|t}) \tag{29}$$

$$=V^*(\hat{\mu}_t) + q(\hat{\mu}^*_{t+H|t}, K\hat{\mu}^*_{t+H|t}) + p(A_K\hat{\mu}^*_{t+H|t}) \tag{30}$$

$$- q(\hat{\mu}^*_{t|t}, c^*_{t|t}) - p(\hat{\mu}^*_{t+H|t}), \tag{31}$$

where $q(\hat{\mu}_t, c_t) = \|C\hat{\mu}_t\|^2_Q + \|c_t\|^2_R$ denotes the stage cost and $p(\hat{\mu}_t) = \|C\hat{\mu}_t\|^2_P$ denotes the terminal cost. Note that K is a stabilizing state feedback controller as designed by (15) and (16), thus it follows that

$$q(\hat{\mu}^*_{t+H|t}, K\hat{\mu}^*_{t+H|t}) + p(A_K\hat{\mu}^*_{t+H|t}) - p(\hat{\mu}^*_{t+H|t}) \leq 0 \tag{32}$$

Take into account the optimal value function $V(\hat{\mu}_{t+1}) < V^*(\hat{\mu}_{t+1})$, thus

$$V^*(\hat{\mu}_{t+1}) - V^*(\hat{\mu}_t) \leq -q(\hat{\mu}^*_{t|t}, c^*_{t|t}), \tag{33}$$

and the optimal value function $V^*(\cdot)$ is a valid Lyapunov function. Therefore, the expectation of the nominal state $\mathbb{E}\hat{x}_t = C\hat{\mu}_t$ converges to zero as $t \to \infty$; that is, the nominal state is mean square stable.

Second, let's consider the dynamics of the error system $e_t := \mu_t - \hat{\mu}_t$, which is defined by the difference between the (12) and the (8). By substituting the controller (13) into the error system, it follows that

$$e_{t+1} = (A + BK)e_t + g(w_t). \tag{34}$$

Iterate the dynamics of the error system (34) from the initial time instance 1 to t with $e_{t+1} = A_K g(w_t) + A_K^2 g(w_{t-1}) + \cdots + A_K^t g(w_1) + A_K^t e_1$. According to Algorithm 1, the initial instance e_1 equals zero, as $\hat{\mu}_1 = \mu_\theta(x_1)$. Then, the L_2 norm of the error state is given by

$$\|e_{t+1}\| = \|A_K g(w_t) + A_K^2 g(w_{t-1}) + \cdots + A_K^t g(w_1)\| \tag{35}$$

$$\leq \|A_K g(w_t)\| + \|A_K^2 g(w_{t-1})\| + \cdots + \|A_K^t g(w_1)\| \tag{36}$$

$$\leq \beta\|g(w_t)\| + \beta^2\|g(w_{t-1})\| + \cdots + \beta^t\|g(w_1)\|. \tag{37}$$

Taking the expectation over the random noise w_t, and using the fact that the random noise signal is independently distributed at different time instances, it follows that

$$\mathbb{E}\|e_{t+1}\| \leq \beta\mathbb{E}_{w_t}\|g(w_t)\| + \beta^2\mathbb{E}_{w_{t-1}}\|g(w_{t-1})\| + \dots \tag{38}$$

$$+ \beta^t\mathbb{E}_{w_1}\|g(w_1)\|. \tag{39}$$

According to Assumption (1), we can further infer that

$$\begin{aligned}\mathbb{E}\|e_{t+1}\| \leq & \beta L\mathbb{E}_{w_t}\|w_t\| + \beta^2 L\mathbb{E}_{w_{t-1}}\|w_{t-1}\| + \dots && (40)\\ & + \beta^t L\mathbb{E}_{w_1}\|w_1\| && (41)\\ \leq & \beta Lb + \beta^2 Lb + \dots + \beta^t Lb && (42)\\ = & \frac{(\beta - \beta^t)Lb}{1-\beta}, && (43)\end{aligned}$$

where the second inequality is a direct result of Assumption 2. As $t \to \infty$, the expectation of the error state norm is bounded by $\frac{\beta Lb}{1-\beta}$. The state of the original system is given by $\mathbb{E}x_t = C(\hat{\mu}_t + \mathbb{E}e_t)$, thus the effect of the error state upon the original state is bounded by $\frac{\beta\sigma Lb}{1-\beta}$, where $\sigma := \|C\|$. Because the nominal state is mean square stable and the error between the actual and nominal state is bounded, the system (4) is proven to be uniformly ultimately bounded. □

B. Proof of Proposition 2

Proof. As the nominal system remains the same as in Proposition 1, the proof for mean square stability of the nominal system is identical as well. We focus on proving the uniform ultimate boundedness of the error system.

In presence of the approximation residuals, the dynamic of the error system $e_t := \mu_t - \hat{\mu}_t$is given as follows,

$$e_{t+1} = (A + BK)e_t + g(w_t) + \epsilon_t \tag{44}$$

Iterating the above equation (34) from the initial time instance 1 to t, one has

$$\begin{aligned}e_{t+1} = & A_K^t e_1 + \underbrace{A_K\epsilon_t + A_K^2\epsilon_{t-1} + \dots + A_K^t\epsilon_1}_{\sum_1^t A_K^k\epsilon_k}\\ & + \underbrace{A_K g(w_t) + A_K^2 g(w_{t-1}) + \dots + A_K^t g(w_1)}_{\sum_1^t A_K^k g(w_k)}.\end{aligned} \tag{45}$$

According to Algorithm 1, the initial instance e_1 equals zero, as $\hat{\mu}_1 = \mu_\theta(x_1)$. Then the L_2 norm of the error state is given by

$$\begin{aligned}\|e_{t+1}\| &= \|\sum_1^t A_K^k\epsilon_k + \sum_1^t A_K^k g(w_k)\| && (46)\\ &\leq \sum_1^t \|A_K^k\epsilon_k\| + \sum_1^t \|A_K^k g(w_k)\| && (47)\\ &\leq \sum_1^t \beta^k\|\epsilon_k\| + \sum_1^t \beta^k\|g(w_k)\|. && (48)\end{aligned}$$

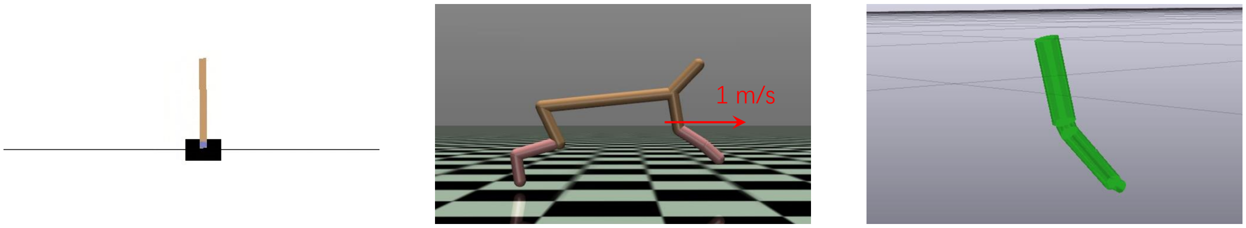
Taking the expectation over the random noise w_t, it follows that

$$\begin{aligned}\mathbb{E}\|e_{t+1}\| &\leq \sum_1^t \beta^k\|\epsilon_k\| + \sum_1^t \beta^k\|g(w_k)\| && (49)\\ &\leq \sum_1^t \beta^k\gamma + \sum_1^t \beta^k Lb && (50)\\ &= \frac{(\beta-\beta^t)(Lb+\gamma)}{1-\beta}, && (51)\end{aligned}$$

where the second inequality is a direct result of Assumption 2 and Assumption 3. As $t \to \infty$, the expectation of the error state norm is bounded by $\frac{\beta Lb+\gamma}{1-\beta}$. Because the nominal state is mean square stable and the error between the actual and nominal state is bounded by a constant $\frac{\beta(Lb+\gamma)}{1-\beta}$, the system (4) is proven to be uniformly ultimately bounded. □

II. Experimental Setup

The experimental evaluation occured in OpenAI Gym or the Drake simulator . A snapshot of the adopted environments in this paper can be found in Figure 20.



(a) Cartpole

(b) HalfCheetah

(c) SoPrA

Fig. 20. Snapshot of the environments. The CartPole and HaflCheetah are simulated in OpenAI gym. The SoPrA soft robotic arm is visualized using Drake.

A. CartPole - Inverted Pendulum on a Cart

We developed a modified version of CartPole with a continuous action space instead of a discrete action space. The system contains a horizontally moving cart and has an inverted pendulum attached to it. The cart is fully actuated while the inverted pendulum is unactuated. In this experiment, the controller is expected to maintain the pendulum in its upright, vertical orientation. The action is the horizontal force applied upon the cart ($a \in [-20, 20]$). $x_{\text{threshold}}$ and $\theta_{\text{threshold}}$ represents the maximum of position and angle, respectively, $x_{\text{threshold}} = 10$ and $\theta_{\text{threshold}} = 20°$. The episode ends if $|\theta| > \theta_{\text{threshold}}$ and the episodes end in advance. The episodes for control evaluation are of length 250. For robustness evaluation in Section IV-E, we apply an impulsive disturbance force F on the cart every 20 steps, of which the magnitude ranges from 80 to 150 and the direction is opposite to the direction of control input.

B. HalfCheetah - Two-legged Running Robot

HalfCheetah is a legged robot locomotion task adapted from OpenAI Gym. The task is to control a two-legged simulated robot to run at the speed of 1 m/s. The control input is the torque applied on each joint, ranging from -1 to 1. The episodes for control evaluation are of length 200.

To achieve dynamic locomotion, a reference signal is first produced for the DeSKO and DKO controllers. In our case, we trained a model-free RL agent using DDPG [29] to run forward at the desired speed and record its state trajectory as the reference signal. Nonetheless, this reference signal is suboptimal and could be improved by using model-based planning methods. In the meantime, SAC is trained directly with the reward to run forward at $1m/s$ without the need for a reference signal.

C. SoPrA - Soft Continuum Robotic Arm

SoPrA is a pneumatic two-segment soft continuum robotic arm [28], built and simulated with the Drake simulation [31]. The pose of the SoPrA arm is described by just two configuration variables ϕ and θ per segment, which are the relative angle of the plane of bending and the curvature. In order to eliminate a singularity in the representation, the following parameterization is adopted

$$\theta_x := \theta \cos(\phi) \tag{52}$$

$$\theta_y := \theta \sin(\phi) \tag{53}$$

and the pose vector $q = [\theta_{x,1}, \theta_{y,1}, \theta_{x,2}, \theta_{y,2}]$, where the subscripts indicate the indexes of the segments. The state is composed of the pose and its derivative, i.e., $x = [q, \dot{q}]$. The controller adjusts the pressure in the six air chambers of SoPrA. Each segment contains three air chambers. Further modeling details can be found in [28]. The episodes for control evaluation are of length 250.

D. Synthetic Biology Gene Regulatory Networks

The gene regulatory networks (GRNs) considered here are in the nano-scale and their physical properties are vastly different compared to the other examples. Particularly to note is that GRNs can exhibit interesting oscillatory behavior.

In this example, we consider a classical dynamical system in systems/synthetic biology which we use to illustrate the reference tracking task at hand. The GRN is a synthetic three-gene regulatory network where the dynamics of mRNAs and proteins follow an oscillatory behavior. A discrete-time mathematical description of the GRN, which includes both transcription and translation dynamics, is given by the following set of discrete-time equations:

$$\begin{aligned}
x_1(t+1) &= x_1(t) + dt \cdot \left[-\gamma_1 x_1(t) + \frac{a_1}{K_1 + x_6^2(t)} + u_1\right] + \xi_1(t),\\
x_2(t+1) &= x_2(t) + dt \cdot \left[-\gamma_2 x_2(t) + \frac{a_2}{K_2 + x_4^2(t)} + u_2\right] + \xi_2(t),\\
x_3(t+1) &= x_3(t) + dt \cdot \left[-\gamma_3 x_3(t) + \frac{a_3}{K_3 + x_5^2(t)} + u_3\right] + \xi_3(t),\\
x_4(t+1) &= x_4(t) + dt \cdot \left[-c_1 x_4(t) + \beta_1 x_1(t)\right] + \xi_4(t),\\
x_5(t+1) &= x_5(t) + dt \cdot \left[-c_2 x_5(k) + \beta_2 x_2(t)\right] + \xi_5(t),\\
x_6(t+1) &= x_6(t) + dt \cdot \left[-c_3 x_6(t) + \beta_3 x_3(t)\right] + \xi_6(t).
\end{aligned} \tag{54}$$

Here, x_1, x_2, x_3 (resp. x_4, x_5, x_6) denote the concentrations of the mRNA transcripts (resp. proteins) of genes 1, 2, and 3, respectively. ξ_i, $\forall i$ are i.i.d. uniform noise ranging from $[-\delta, \delta]$, i.e., $\xi_i \sim \mathcal{U}(-\delta, \delta)$. During training, $\delta = 0$ and for evaluation δ is set to 0.5 and 1 respectively in Section IV-E. a_1, a_2, a_3 denote the maximum promoter strength for their corresponding gene, $\gamma_1, \gamma_2, \gamma_3$ denote the mRNA degradation rates, c_1, c_2, c_3 denote the protein degradation rates, $\beta_1, \beta_2, \beta_3$ denote the protein production rates, and K_1, K_2, K_3 are the dissociation constants. The set of equations in Eq.(54) corresponds to a topology where gene 1 is repressed by gene 2, gene 2 is repressed by gene 3, and gene 3 is repressed by gene 1. dt is the discretization time step.

In practice, only the protein concentrations are observed and given as readouts, for instance via fluorescent markers (e.g., green fluorescent protein, GFP or red fluorescent protein, mCherry). The control scheme u_i will be implemented by light control signals which can induce the expression of genes through the activation of their photo-sensitive promoters. To simplify the system dynamics and as it is usually done for the GRN model, we consider the corresponding parameters of the mRNA and protein dynamics for different genes to be equal. More background on mathematical modeling and control of synthetic biology gene regulatory networks can be referred to [32]. In this example, the parameters are as follows:

$$\forall i: \;\; K_i = 1, a_i = 1.6, \gamma_i = 0.16, \beta_i = 0.16, c_i = 0.06, dt = 1$$

In Figure 21, a single snapshot of the state temporal evolution without ξ is depicted. We uniformly initialized between 0 to 5, i.e., $x_i(0) \sim \mathcal{U}(0, 5)$, persistent oscillatory behavior is also exhibiting similar to the snapshot in Figure 21.

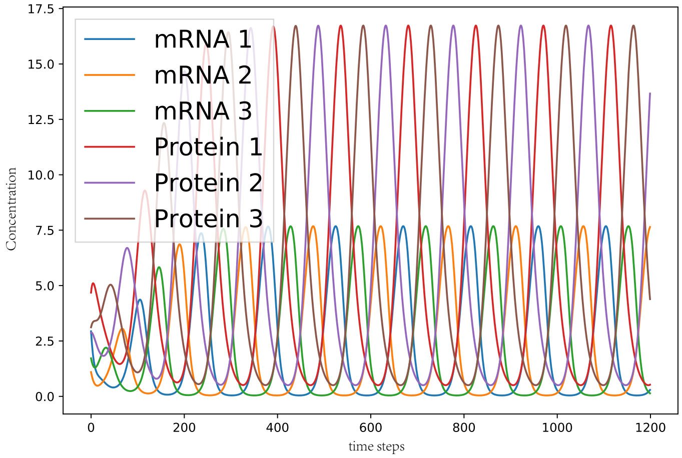


Fig. 21. A snapshot of the natural oscillatory behavior of a GRN system consisting of 3 genes. The oscillations have a period of approximately 150 arbitrary time units. The task is to control the concentration of Protein 1 to track a set-point reference signal. The X-axis denotes time and Y-axis denotes the value/concentration of each state.

III. Hyperparameters

The data set is split into the training set composed of 40000 data points, and the validation set composed of 4000 data points. Each data point contains the current state and action input, and the resulting state, i.e. $\{x_t, u_t, x_{t+1}\}$. In the simulation experiments, the data is collected by randomly sampling actions from a uniform distribution over the action space. In the SoPrA experiment, we introduced two patterns of random input generation: the sinusoidal input and the polyline input. In the sinusoidal input pattern,

$$a_t = \alpha \left[\sin\omega_1 t, \sin\omega_1 t + \frac{2\pi}{3}, \sin\omega_1 t + \frac{4\pi}{3}, \sin\omega_2 t, \sin\omega_2 t + \frac{2\pi}{3}, \sin\omega_2 t + \frac{4\pi}{3},\right]^{\mathrm{T}} + \beta\mathbf{1}$$

, where $\omega_{1,2}$, α and β are randomized every 1000 steps to generate sinusoidal input with different magnitude and frequency. In the polyline input pattern, the input $a_t = p_c + \frac{p_t - p_c}{T}$, where $p_c \in \mathbb{R}^6$ denotes the initial pressure, $p_t \in \mathbb{R}^6$ denotes a randomly generated target pressure and T is the randomly generated time interval. The DeSKO model is trained with stochastic gradient descent. In our implementation, the ADAM solver is used for optimization. At each step during training, a batch of 256 data points is sampled from the training set and used for the model update.

TABLE IV
TRAINING HYPERPARAMETERS OF DESKO

Hyperparameters	GRN	GRN (Observation Noise)	SoPrA Simulation	SoPrA Simulation (Observation Noise)	Real-world SoPrA	Others
Size of data set	40000	40000	40000	40000	200000	40000
Batch Size	256					
Learning rate	5×10^{-4}	5×10^{-4}	1×10^{-4}	1×10^{-4}	1×10^{-3}	1×10^{-3}
Prediction horzion H	16	16	16	16	30	16
Structure of $\mu_\theta(\cdot)$	(256,128,64)					
Structure of $\sigma_\theta(\cdot)$	(256,128,64)					
Activation function	ELU					
Dimension of observables	20					
Entropy threshold $\mathcal{H}$	-100	-60	-40	-60	-20	-20

TABLE V
CONTROL HYPERPARAMETERS OF DESKO

Hyperparameters	Cartpole	Cartpole (Observation Noise)	Cartpole (Process Noise)	GRN	GRN (Observation Noise)	GRN (Process Noise)
Prediction Horizon	16	16	30	30	40	30
Control Horizon	6	6	6	10	10	10
Weight matrix Q[1]	$[1, 0.1, 1, 0.01]$	$[1., 0.1, 10, 0.01]$	$[0.01, 0, 2, 0.01]$	$[0, 0, 0, 1, 0, 0]$	$[0, 0, 0, 0.1, 0, 0]$	$[0, 0, 0, 1, 0, 0]$
Weight matrix R[2]	$[0.1]$	$[0.1]$	$[2]$	$[0.1, 0.1, 0.1]$	$[0.01, 0.01, 0.01]$	$[0.1, 0.1, 0.1]$

[1,2] The weight matrices are diagonal matrices of the shown vector.

TABLE VI
CONTROL HYPERPARAMETERS OF DESKO

Hyperparameters	Halfcheetah	Halfcheetah(Observation Noise)	SoPrA Simulation	SoPrA Simulation (Observation Noise)
Prediction Horizon	16	16	30	40
Control Horizon	6	6	10	10
Weight matrix Q[1]	[2, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 2, 1, 0.1, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0.1]	[2, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 2, 1, 0.1, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0.1]	[1, 1, 1, 1, 0, 0, 0, 0]	[4, 6, 4, 4, 0, 0, 0, 0]
Weight matrix R[2]	[0.01,0.01,0.01,0.01,0.01,0.01]	[0.01,0.01,0.01,0.01,0.01,0.01]	[0.1,0.1,0.1,0.1,0.1,0.1]	[0.1,0.1,0.1,0.1,0.1,0.1]

[1,2] The weight matrices are diagonal matrices of the shown vector.

TABLE VII
CONTROL HYPERPARAMETERS OF DESKO

Hyperparameters	Real-world SoPrA
Prediction Horizon	36
Control Horizon	6
Weight matrix for MPC Q[1]	$[0, 0, 0, 0, 0, 0, 2.9, 2.9, 2.9, 0.1, 0.1, 0.1]$
Weight matrix for MPC R[2]	$[0.2, 0.2, 0.2, 0.2, 0.2, 0.2]$
Weight matrix for integral control Q_I[3]	$[0, 0, 0, 0, 0, 0, 0.05, 0.05, 0.05, 0, 0, 0]$

[1-3] The weight matrices are diagonal matrices of the shown vector.