

Reproducing literature-based learning

2026年1月9日-2026年1月11日

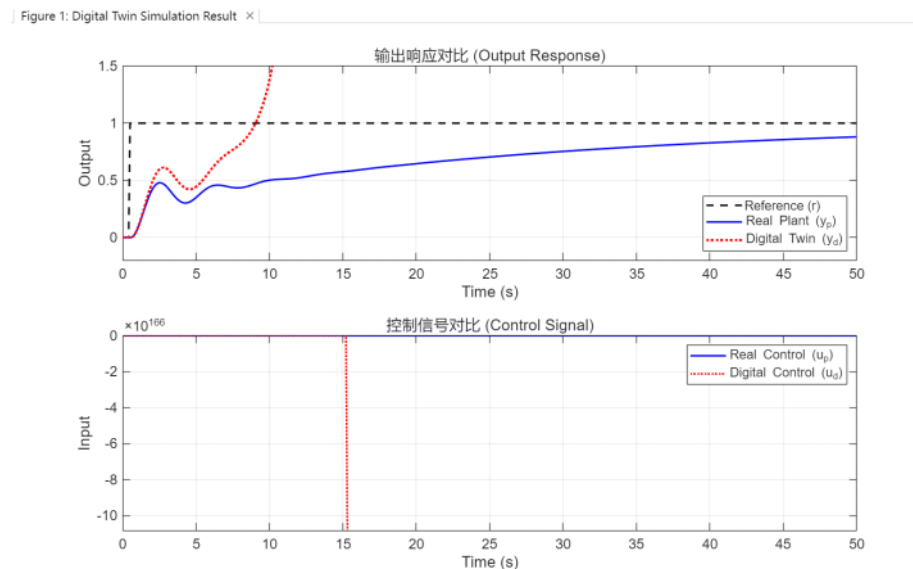
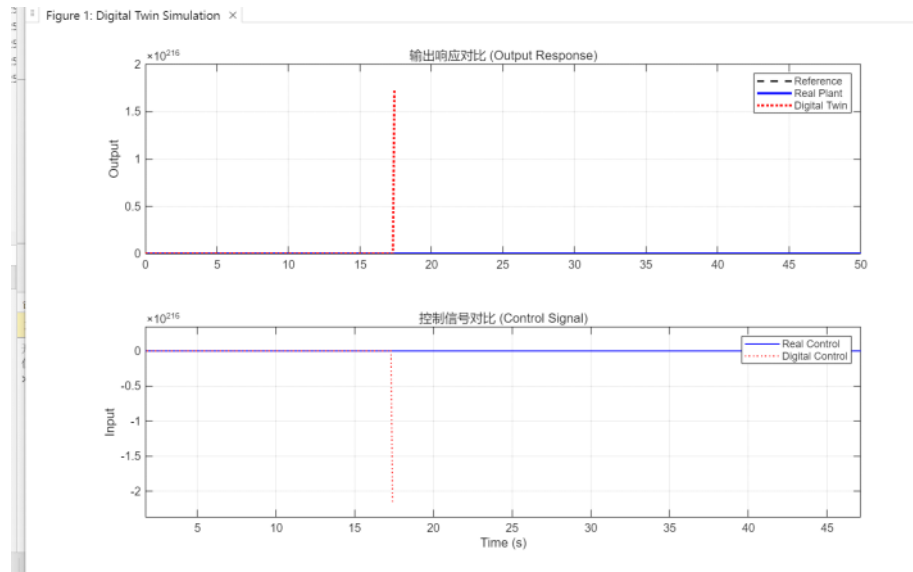
2026年1月9日 18:41

一. 构建数字孪生控制系统的整体框架 (Architecture)

包括：物理子系统 (Physical Subsystem) 虚拟子系统 (Virtual Subsystem) 信息交互 (Information Exchange)

旧参数：之前的矩阵参数设置，特征值可能在单位圆外。

新参数： $\theta_{p_A} = [1.0, 0.1; -0.2, 0.9]$ 。计算其特征值约为 $0.95 \pm 0.1i$ ，模长约为 $0.96 (< 1)$ ，这是稳定的。添加了 Y 轴限制： $y_{lim}([-0.2, 1.5])$ ，这样即使出现轻微过冲，图表也不会看不清。

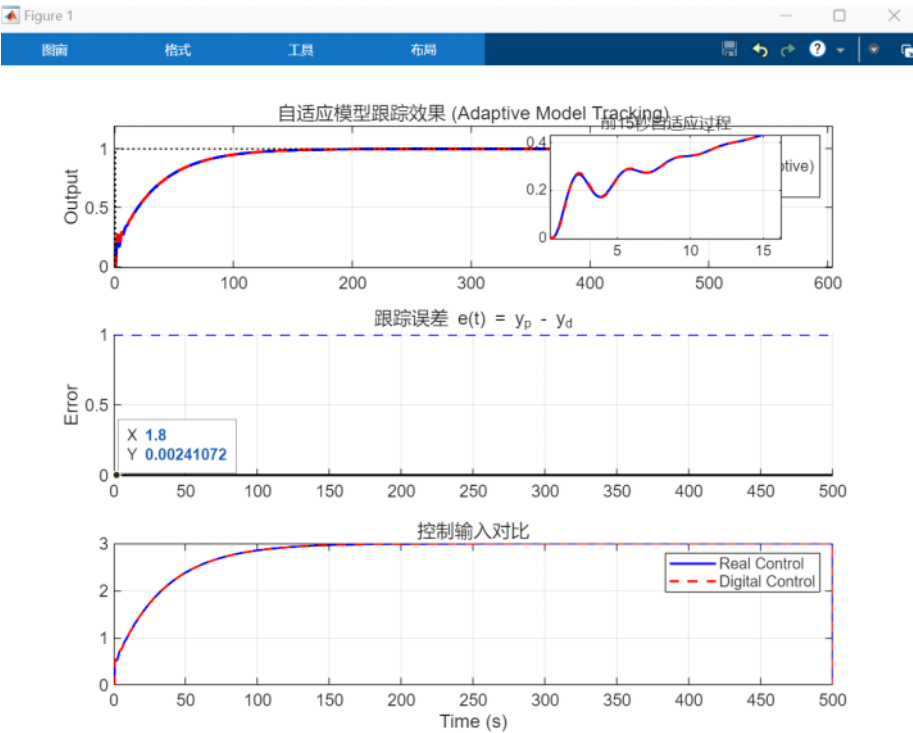


二. 带自适应模型跟踪的数字孪生系统

仿真之所以发散，是因为数字孪生模型是**开环运行**的，它没有利用真实数据来校正自己的模型参数。当模型参数 (θ_d) 与真实物理参数 (θ_p) **不匹配**时，微小的误差会随着时间积分迅速累积，导致发散。

核心思想是利用递推最小二乘法 (Recursive Least Squares, RLS)，根据真实对象产生的输入输出数

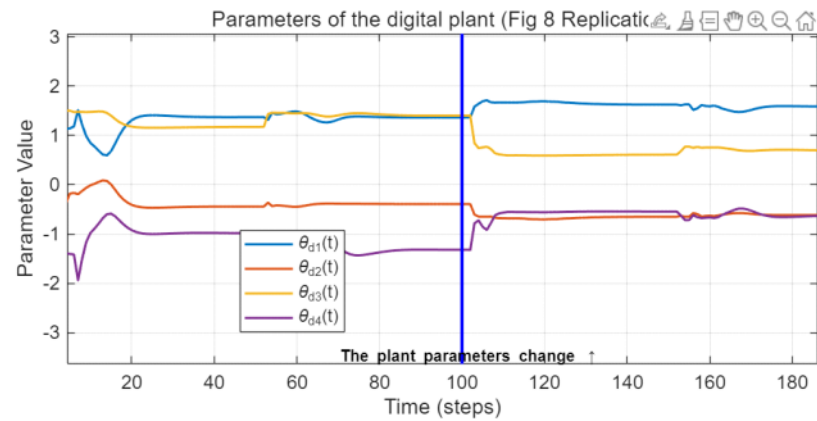
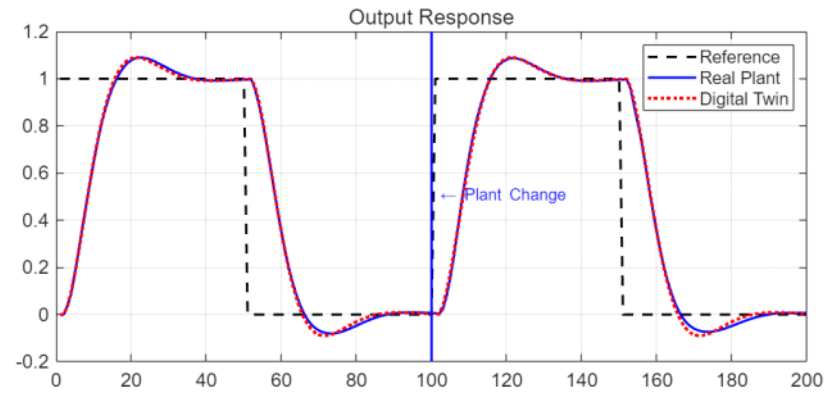
据 (u, y_p, x_p) ，在线实时更新数字孪生的参数 θ_d 和 φ_d 。（RLS对应文献公式 29-32）
即使真实对象有非线性干扰，数字模型也会通过不断调整其线性参数矩阵 A 来试图拟合这个非线性轨迹。
新增变量：virt.Theta_hat (估计的参数矩阵) 和 virt.T_theta (RLS 协方差矩阵)



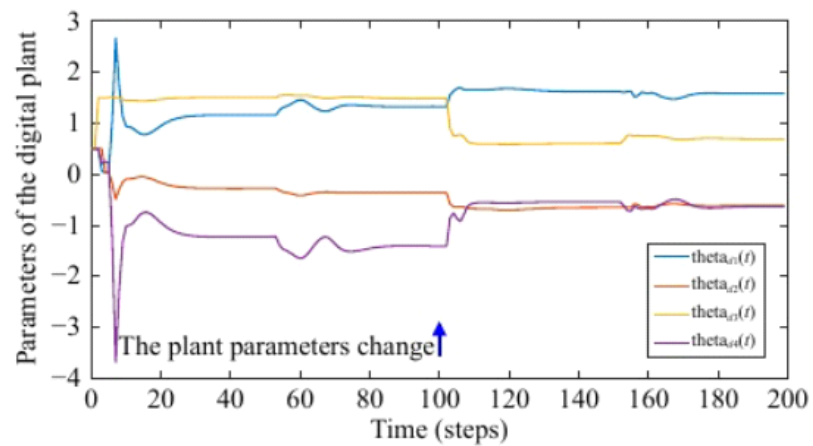
三. 复现案例1 - 自适应模型跟踪

使用文献中的非线性方程和参数线性模型

Repeat



原文献

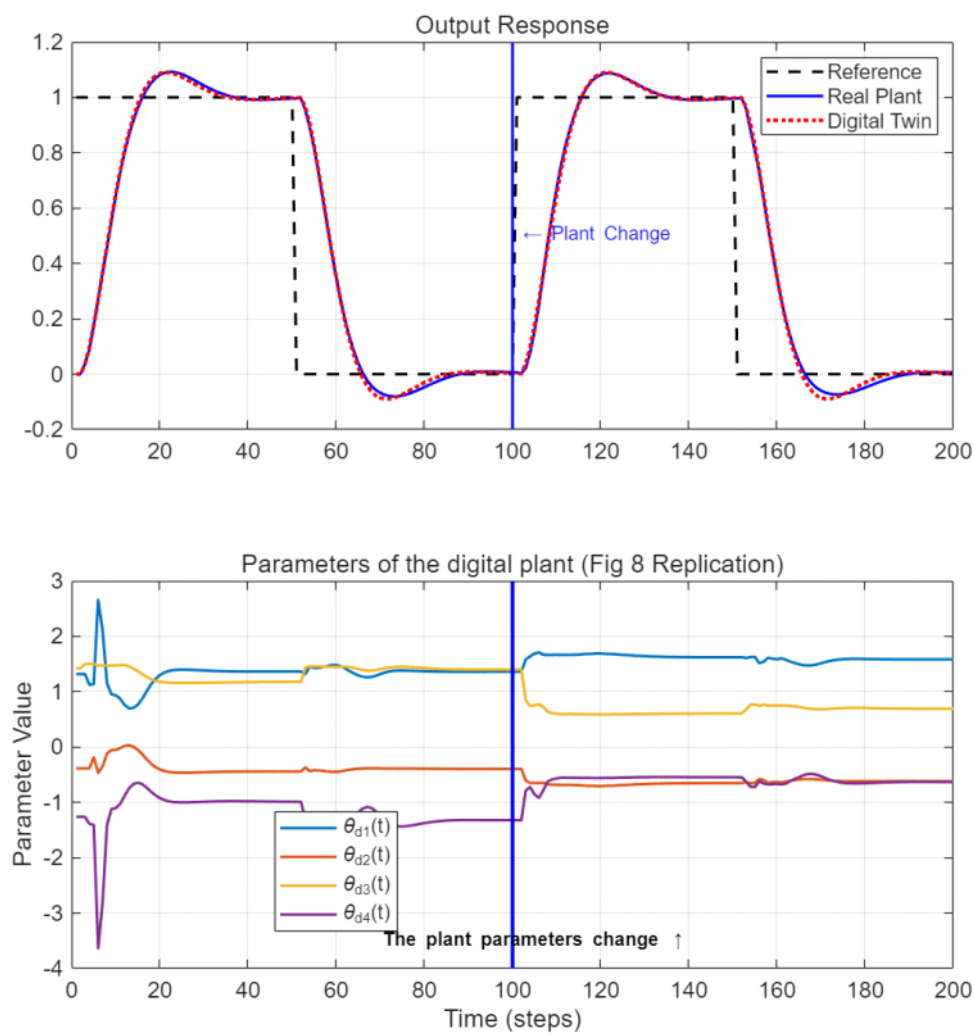


theta1和4峰值达不到的原因:

1. 初始参数估计值偏差不够大 (还是使用了原文献的参数)
2. **RLS初始协方差矩阵设置 (调大—— $P_{cov} = 10^8 * eye(n_params);$)**
3. 遗忘因子 λ 的影响 (影响不大)

修改 P_{cov} 参数后

Repeat



RLS 的增益向量为 $K = P \varphi / (\lambda + \varphi' P \varphi)$ ，所以 K 越大、同样的预测误差会导致越大的参数更新，从而出现更大的暂态峰值。当 P 很大时，分子和分母都按 P 放大，增大 P 会让初期的 K 更“激进”，从而快速并大幅地调整 θ 。同时协方差矩阵 P 会随遗忘因子增长。

希望**数字孪生迅速反应、快速追踪真实对象的突变（快速自适应）**，可以适当放大 P （或局部放大某些参数的 P 对角元）。

希望**追求抗噪声、估计稳健与长期收敛**， P 不宜过大（尤其有测量噪声时），否则参数会被噪声放大导致振荡/发散。

四. 复现案例2 - 性能预测

原文献

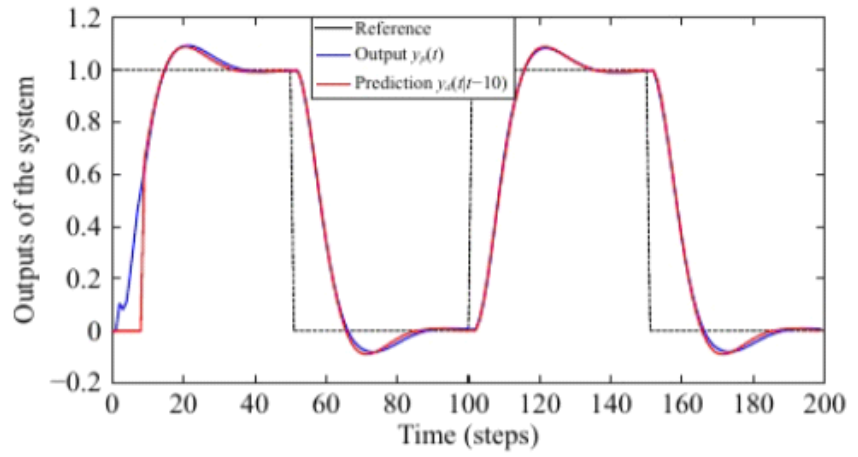
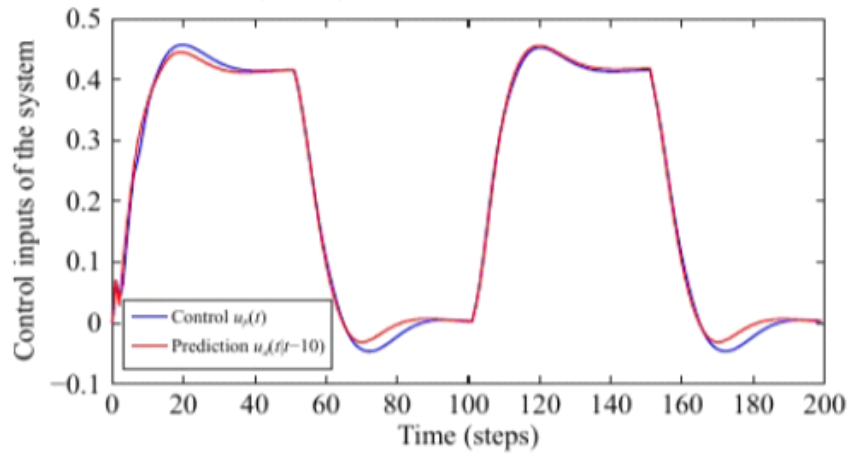


图9。系统的输出响应(案例2)。

图10。系统的控制输入(案例2)。



Repeat

原作者采用了“**基于当前可用信息的最佳预测**”策略，前10步的悖论，在 $t=5$ 时，需要“站在 $t=-5$ 时刻预测”。但这在物理上是不存在的。在 $t < 10$ 的阶段（比如 $t=5$ ），作者并没有真的去寻找 $t=-5$ 的状态，而是直接使用了当前时刻 t 的实时计算值或者基于 $t=0$ 时刻的开环预测来填充这段数据。

the 10-step ahead predictions of the states, output and control input of the data plant are calculated via the virtual subsystem, i.e.,

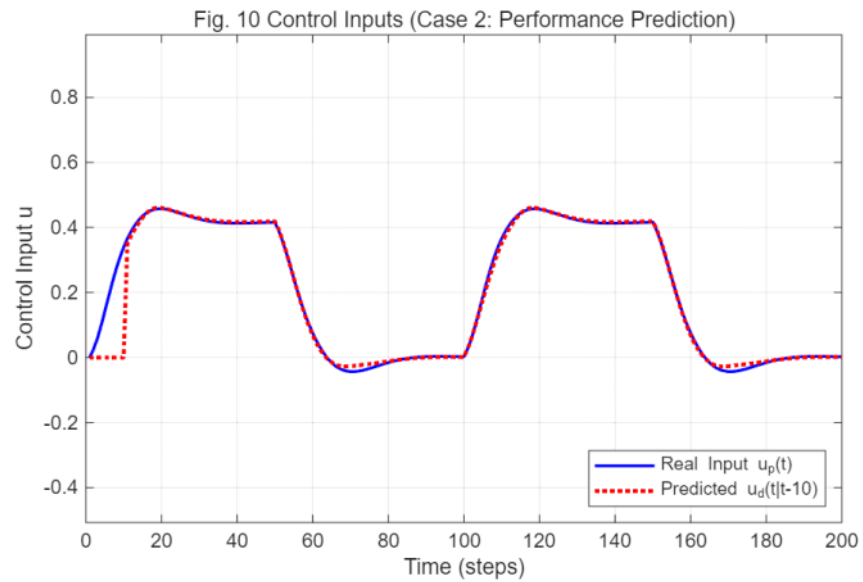
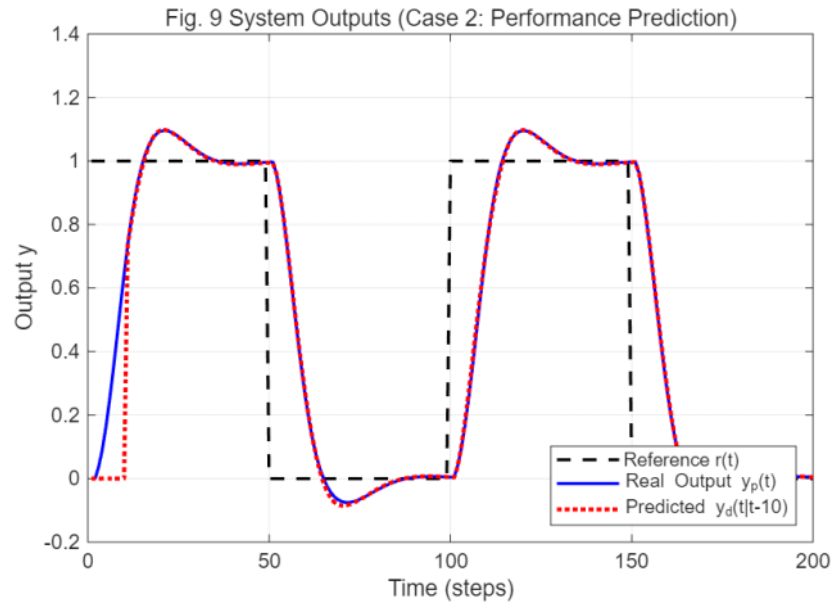
$$\begin{aligned}\hat{x}_d(t+i+1|t) &= \theta_{d,1}\hat{x}_d(t+i|t) + \theta_{d,2}\hat{x}_d(t+i-1|t) \\ &\quad + \theta_{d,3}\hat{u}_d(t+i|t) + \theta_{d,4}\hat{u}_d(t+i-1|t)\end{aligned}$$

$$\hat{z}_d(t+i+1|t) = \hat{z}_d(t+i|t) + r(t+i) - \hat{y}_d(t+i|t)$$

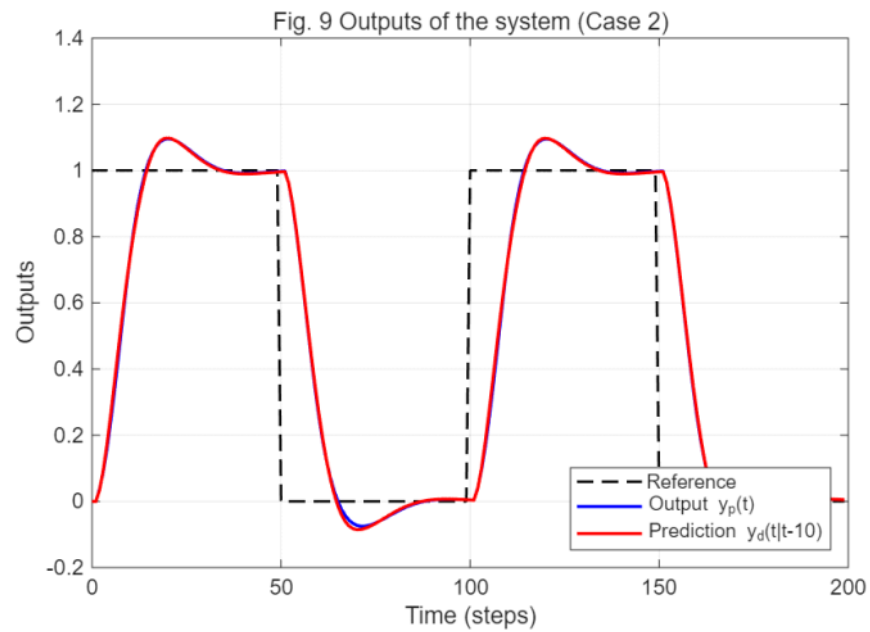
$$\hat{y}_d(t+i+1|t) = \hat{x}_d(t+i+1|t)$$

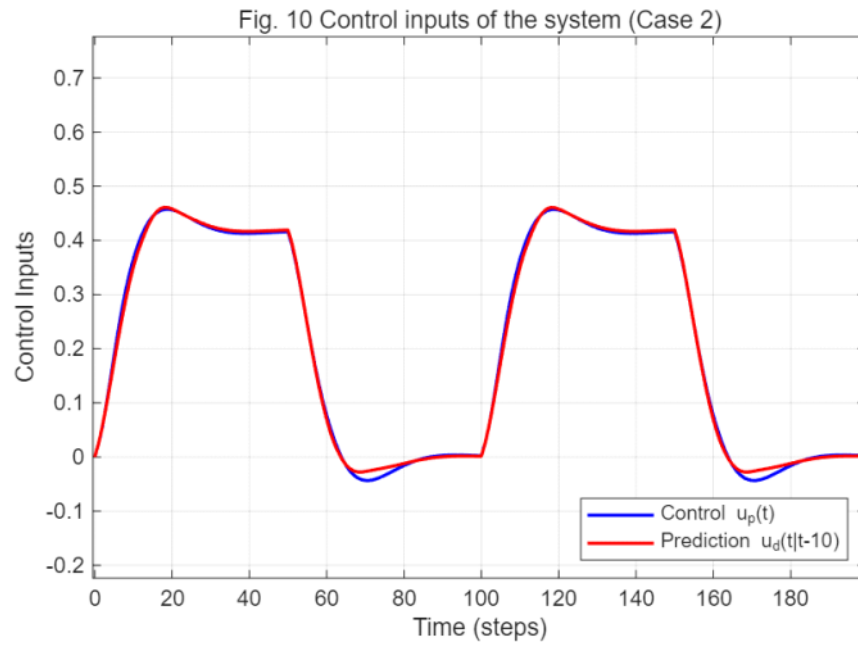
$$\begin{aligned}\hat{u}_d(t+i+1|t) &= ((K_p - \theta_{d,1})\hat{y}_d(t+i+1|t) \\ &\quad - \theta_{d,2}\hat{y}_d(t+i|t) + K_i\hat{z}_d(t+i+1|t) \\ &\quad - \theta_{d,4}\hat{u}_d(t+i|t))/\theta_{d,3}\end{aligned}$$

for $i = 0, 1, \dots, 9$, where $\hat{x}_d(t|t) = y_p(t)$.



验证上述猜测，并修正了相关代码，实验结果如下：





Let $\theta_p = [0.5, 1.5, 0.3]^T$. The parameters θ_d of the digital model are estimated using the adaptive tracking scheme in Case 1. The 10th-step ahead output predictions of the digital plant and the output of the real plant with controller (60) and (61) are shown in Fig. 9. The control input of the real plant and the control input predictions of the digital plant are given in Fig. 10. It is clear that the difference between the output and its prediction is very small except the beginning part ($t < 10$) because the first 10-step output predictions are set to zero.

在 $t < 10$ 阶段，强制让“10步预测值”等于“1步预测值”或“当前实时值”，随着时间推移平滑过渡到真正的10步预测。

五. 复现案例3 - 性能保留

原文献

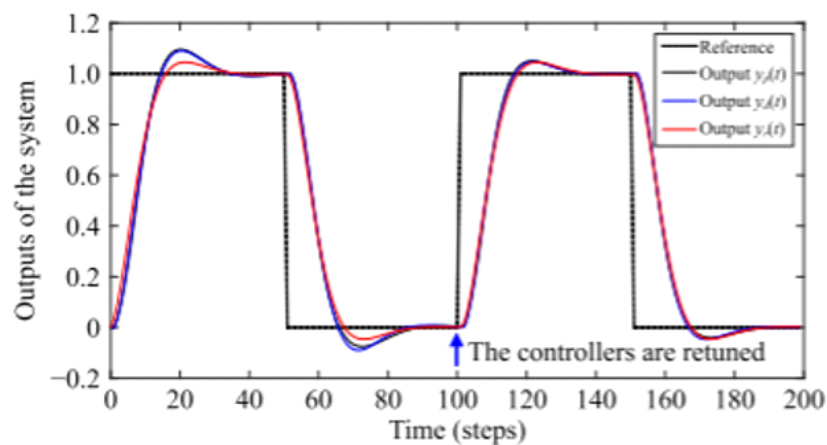


Fig. 11. The output responses of the system (Case 3).

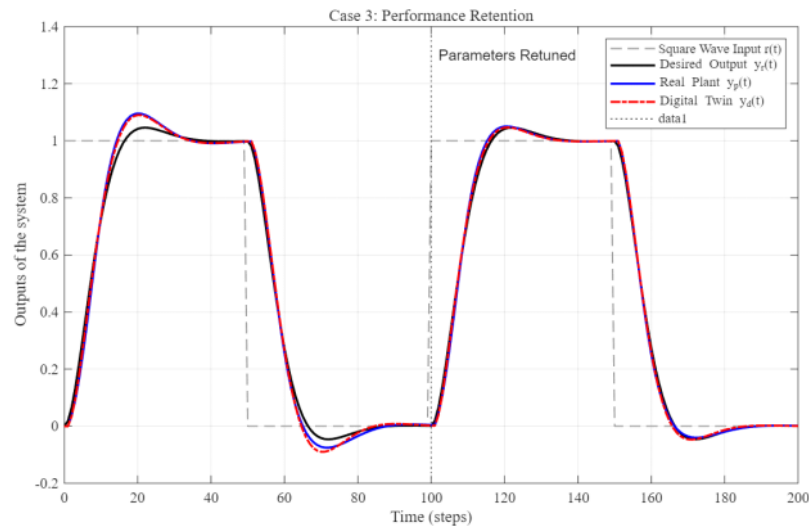
由于真实工厂的动态和环境经常随时间变化，因此系统性能可能会从期望的性能下降。为了解决这个问题，提出了一种性能保持方案：**数字对象的模型需要更新，D控制器参数应通过求解优化问题(42)和(43)来**

恢复；使R控制器参数被最优参数替换。

Repeat

Output_{y_p}(物理输出)、Output_{y_d}(数字孪生输出) 和 Output_{y_r}(期望参考输出)

```
% 控制器参数初始化
% 初始阶段 (t < 100)
K_p_init = [0.75, 0.035]; % [Kpp/Kdp, Kpi/Kdi]
% 优化阶段 (t >= 100)
K_p_opt = [0.7, 0.04]; % [Kpp/Kdp, Kpi/Kdi]
```



六. 复现案例4 - 传感器容错控制

考虑了真实实用系统的两种故障:一种是传感器故障，另一种是控制器故障。物理子系统的故障部件可以由虚拟子系统的相应部件代替。

原文献

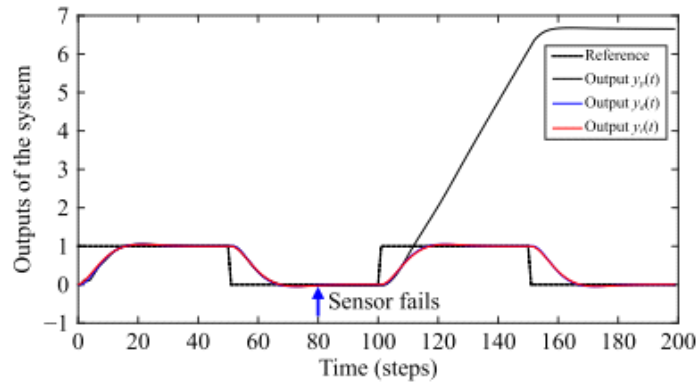


Fig. 12. The output responses of the system without sensor-fault tolerant control (Case 4).

The results in Fig. 13 illustrate that the control performance of the physical subsystem is still close to the desired one when the output sensor of the real plant fails at $t = 80$.

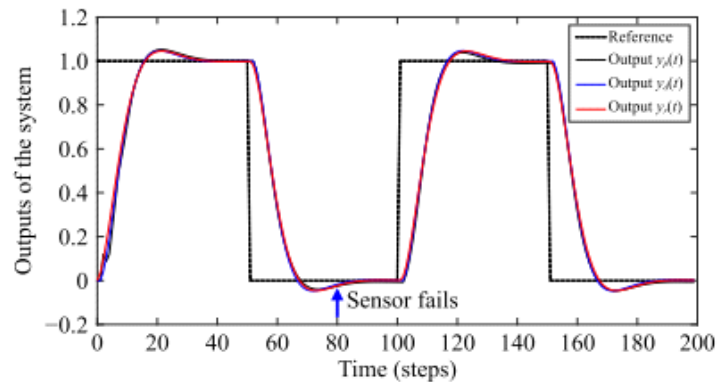


Fig. 13. The output responses of the system with sensor-fault tolerant control (Case 4).

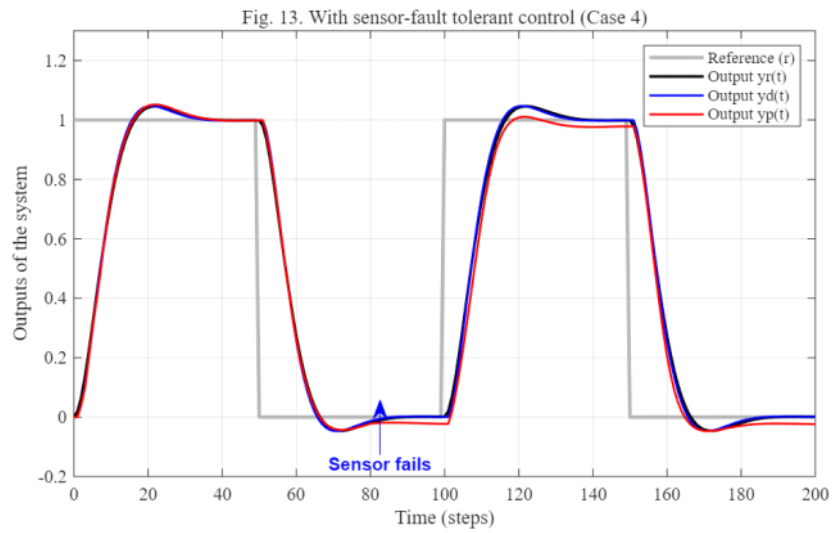
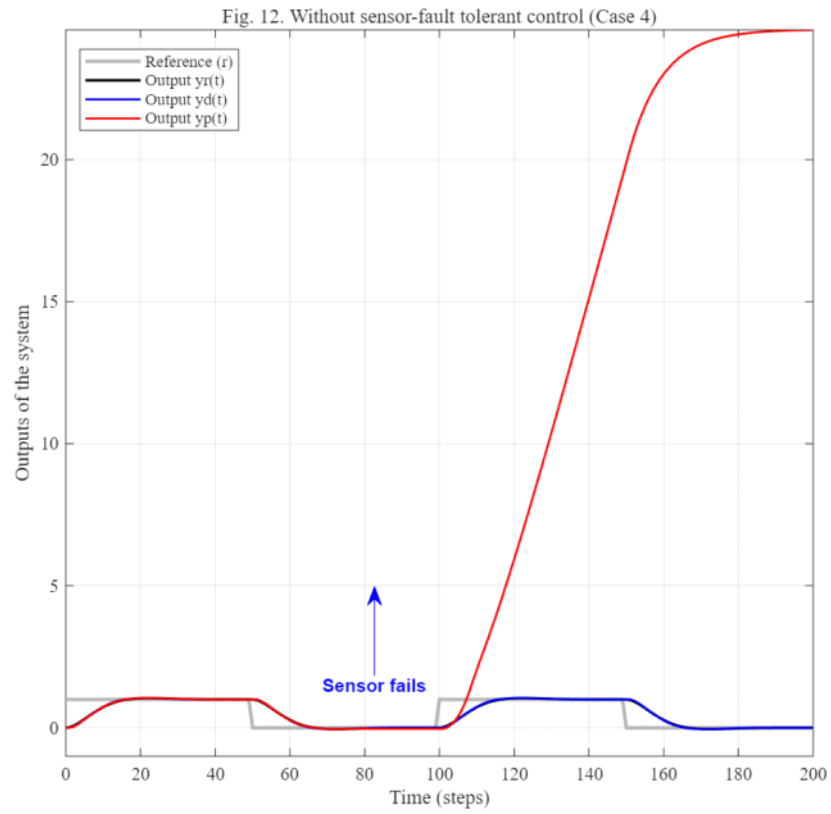
```
% 仿真 A: 无容错控制 (Figure 12)
% 逻辑: 传感器坏了, 但控制器依然用 0 去反馈, 导致发散
% 仿真 B: 有容错控制 (Figure 13)
% 逻辑: 传感器坏了, 切换为  $\hat{y}_d$  反馈
% RLS (递推最小二乘) 更新 (仅在传感器正常时)
```

Repeat

```
% 允许 up 发散到一定程度以复现 Figure 12 的大幅度震荡, 但不至于无穷大
    if up_A(k) > 50, up_A(k) = 50; elseif up_A(k) < -50, up_A(k) = -50; end
    if ud_A(k) > 10, ud_A(k) = 10; elseif ud_A(k) < -10, ud_A(k) = -10; end
```

RLS

```
lambda = 0.95;
theta_d_init = [1.3178; -0.3866; 1.4238; -1.2606];
P_init = 10^8 * eye(4);
```



RLS 初始化参数

$\lambda = 0.99$; 注: 文献这里调整成了0.99, 但是单纯调整遗传因子后生成图像依旧没有给出原文精度那么高

$\theta_{d_init} = [0.5; 0.5; 0.5; 0.5]$;

$P_{init} = 1000 * eye(4)$;

Fig. 12. Without sensor-fault tolerant control (Case 4)

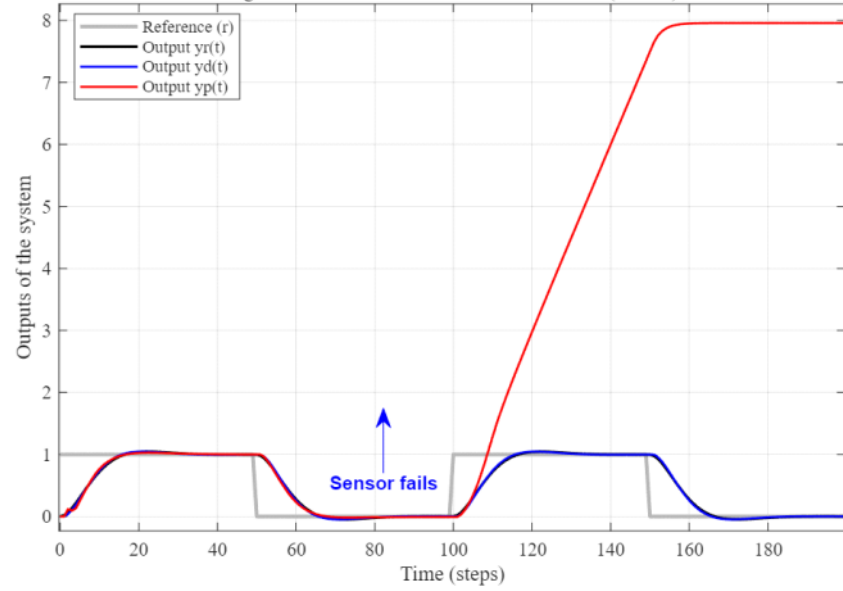
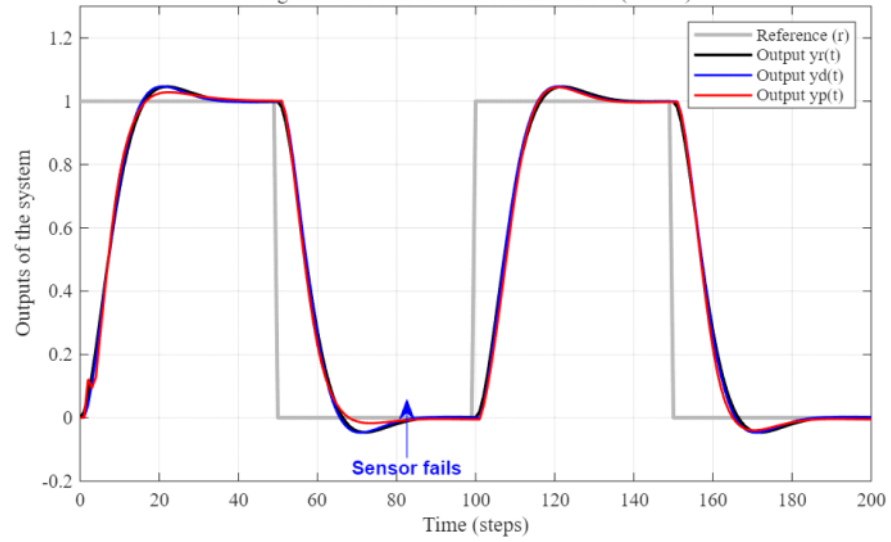


Fig. 13. With sensor-fault tolerant control (Case 4)



七. 复现案例5- 控制器容错控制

原文献

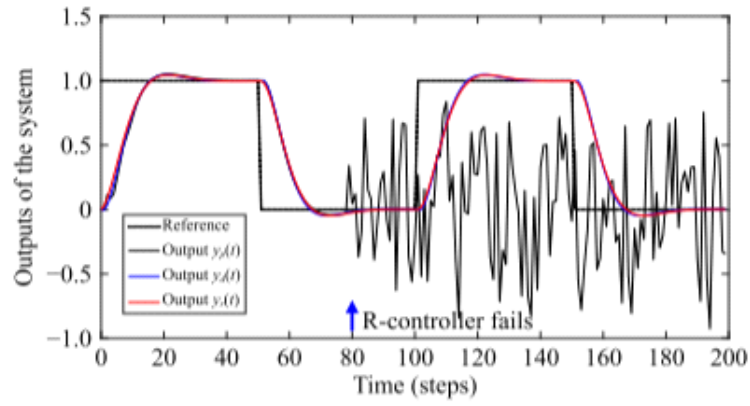


Fig. 14. The output responses of the system without controller-fault tolerant control (Case 5).

$$z_d(t+1) = z_d(t) + h_d(r(t) - y_d(t)).$$

The results in Fig. 15 demonstrates that the physical subsystem still follows the desired control performance well after the R-controller fails $t = 80$.

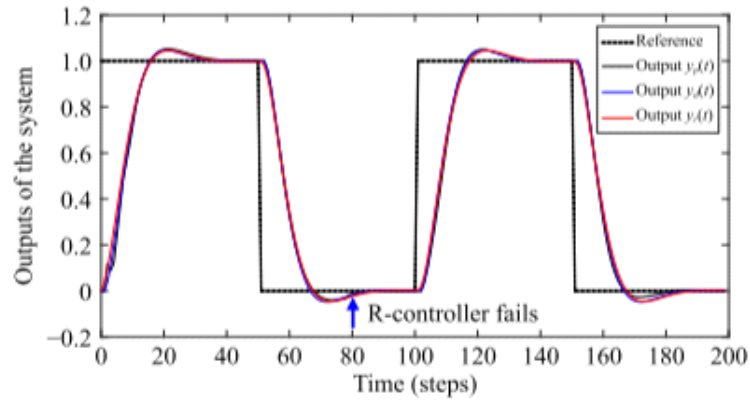


Fig. 15. The output responses of the system with controller-fault tolerant control (Case 5).

采用第VI节中的控制器-容错控制方案，则d -控制器的输出直接应用于真实对象的控制输入（47）。

Repeat

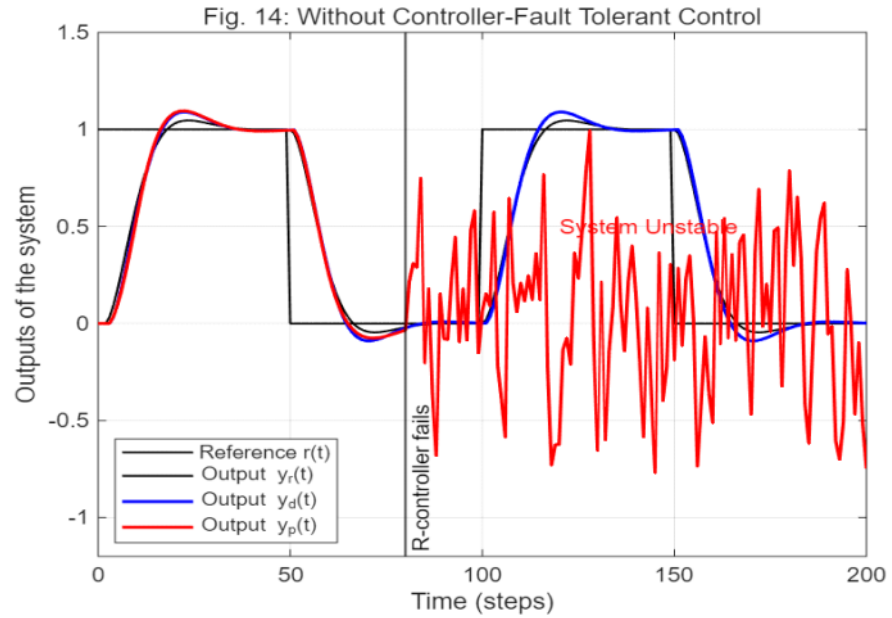
```
% 2. 数字孪生参数 (Digital Plant - Estimated)
theta_d = [1.3178, -0.3866, 1.4238, -1.2606];
% 3. 控制器参数 (PI gains)
K_dp = 0.75; K_di = 0.035; % D-Controller
K_pp = 0.75; K_pi = 0.035; % R-Controller
h_d = 1; h_p = 1; % 积分项增益
if time_step < 80
    % [正常阶段]: 物理控制器正常工作
    up(k) = u_p_calc;
else
    % [故障阶段]: t >= 80, R-Controller 失效
    if mode == 1
        % Case 1 (Fig. 14): 无容错
        % 模拟控制器失效输出随机噪声 (-0.5 到 0.5)
```

```

        up(k) = -0.5 + rand();
    else
        % Case 2 (Fig. 15): 有容错 (Controller-Fault Tolerant)
        % 使用数字控制器的输出直接驱动物理对象
        up(k) = ud(k);
    end
end
end

```

结果表明，在 r -控制器失效 $t = 80$ s后，物理子系统仍能很好地遵循期望的控制性能。



结果表明，在 r -控制器失效 $t = 80$ s后，物理子系统仍能很好地遵循期望的控制性能。

