用神经网络进行系统辨识

Author: 章翰宇
ID: 3220104133

Abstract

如图所示二自由度机械臂模型（平面俯视图），q_1和q_2表示机械臂的两个关节角大小。请设计神经网络辨识方案，对该系统进行辨识（系统输入为τ_1，τ_2，输出为q_1，q_2）

1. 利用已知系统得到辨识所需的输入输出数据
2. 通过步骤 1 得到的数据来训练神经网络
3. 对比原系统与神经网络辨识得到的系统是否一致

Keywords: 智能控制，神经网络，系统辨识

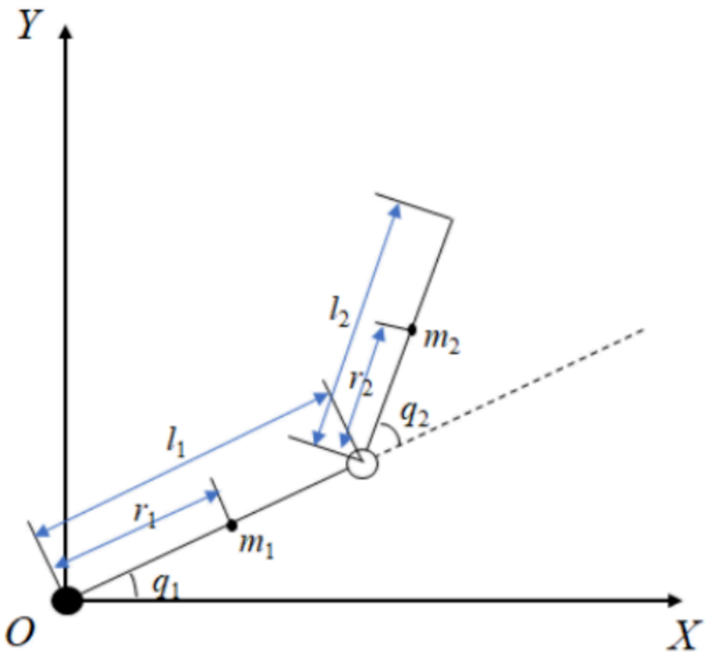


Figure 1: 物理动力系统示意图

物理建模

由于动力学方程为：

$$\begin{cases} m_{11}\ddot{q}_1 + m_{12}\ddot{q}_2 + c_{11}\dot{q}_1 + c_{12}\dot{q}_2 + g_1 = \tau_1 \\ m_{21}\ddot{q}_1 + m_{22}\ddot{q}_2 + c_{21}\dot{q}_1 + c_{22}\dot{q}_2 + g_2 = \tau_2 \end{cases}$$

有这些参数已知：

$$h_1 = m_1 r_1^2 + m_2 l_2^2 + I_1$$
$$h_2 = m_2 r_2^2 + I_2$$
$$h_3 = m_2 l_1 r_2$$
$$h_4 = m_1 r_1 + m_2 l_1$$
$$h_5 = m_2 r_2$$

为了便于在 Simulink 中仿真构造模型，干脆消去所有时变系数，整理成仅仅含有时不变系数 $h_1 \dots h_5$的如下两式

- 描述$\ddot{q}_1$的：

$$
\begin{aligned}
(h_1 h_2 - h_3^2 \cos^2 q_2)\ddot{q}_1 = (2h_2 h_3 \sin q_2)\dot{q}_1 \dot{q}_2 + h_2 h_3 \sin q_2 \dot{q}_2{}^2 + (h_3^2 \cos q_2 \sin q_2 + h_2 h_3 \sin q_2)\dot{q}_1{}^2 \\
-h_2 h_4 g \cos q_1 + h_3 h_5 g \cos q_2 \cos(q_1 + q_2) + h_2 \tau_1 - (h_2 + h_3 \cos q_2)\tau_2
\end{aligned} \tag{1}
$$

- 描述$\ddot{q}_2$的（上式基础上，由于 Simulink 连线可直接引$\ddot{q}_1$线，故保留$\ddot{q}_1$）：

$$
h_2 \ddot{q}_2 = \tau_2 - h_5 g \cos(q_1 + q_2) - h_3 \sin q_2 \dot{q}_1{}^2 - (h_2 + h_3 \cos q_2)\ddot{q}_1 \tag{2}
$$

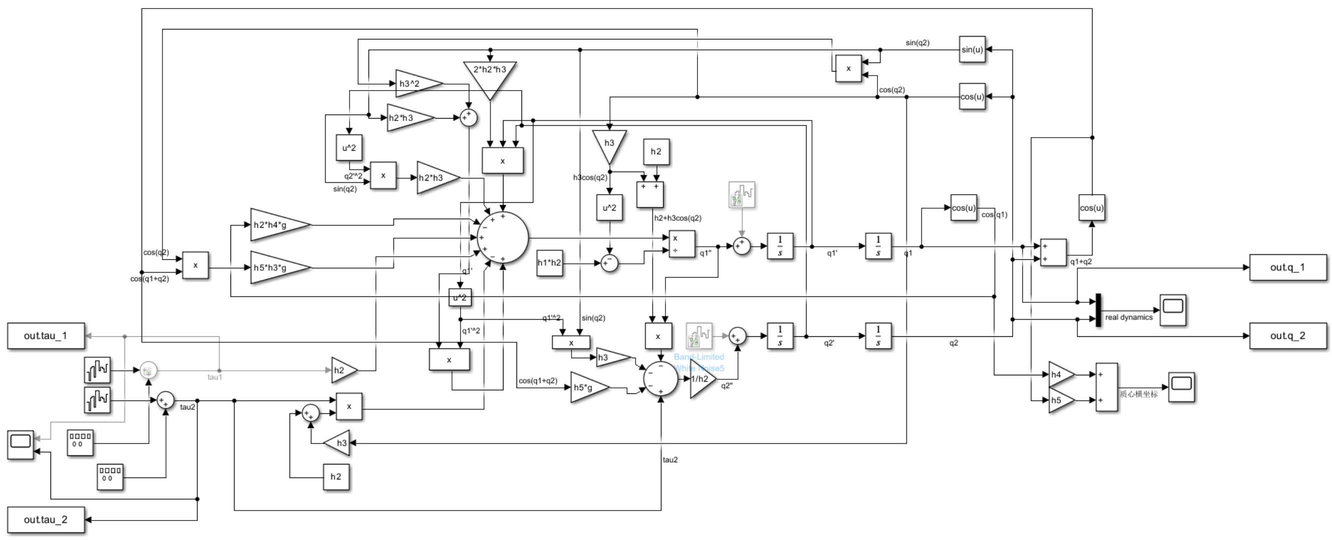
据此，建立如下的 Simulink 模型。解释一下该图：

核心是中间偏右的几个积分块，代表$\ddot{q}_1 \to \dot{q}_1 \to q_1$以及$\ddot{q}_2 \to \dot{q}_2 \to q_2$两路（先不用看两个 Guass 白噪声，这是后续用的）。然后，$\ddot{q}_1$根据 Equation 1 给出，把该式右边表达出然后除以系数$(h_1 h_2 - h_3^2 \cos^2 q_2)$即得到$\ddot{q}_1$；$\ddot{q}_2$由 Equation 2 得出，表达出右边再乘以$1/h_2$的增益即可。

左下角部分的两个白噪声那块部分，就代表τ_1, τ_2两个输入（当然也可以是别的形式，我这里为了训练，取白噪声叠加方波信号作为输入，取该输入的响应信号作为训练的数据集）

```
h1 = 0.0308;h2 = 0.0106;
h3 = 0.0095;h4 = 0.2086;
h5 = 0.0631;g = 9.8;
```

Simulink



数学分析

思路概述

使用 NARMA 模型。分析该系统的阶数，由第一性原理，该模型的动力学都来自牛顿第二定律，本质上是二阶模型。而且，系统的演化（写成状态方程就能很明显看出）仅与输入量的本体（不依赖于输入的导数乃至高阶导数）以及输出量的本体和一阶导有关，故假设该系统为Φ，可以把该连续系统演化关系写为：

$$y(t+\mathrm{d}t)=\Phi_{\mathrm{d}t}(y(t),\dot{y}(t),u(t))$$

其实写为状态方程应该用$u(t+\mathrm{d}t)$，但是由于预测需要物理可实现，因此取$u(t)$

将其离散化为：

$$y(k+1)=\varphi(y(k),y(k-1),u(k))$$

题目中想用一个神经网络f来近似这个系统，可以写出这个虚拟的动力系统方程（采取串并联结构）：

$$\hat{y}(k+1)=f(y(k),y(k-1),u(k))$$

训练标准，采取均方误差最小。定义损失泛函L如下：

$$L(f):=\sum_k e^2(k)=\sum_k \|\hat{y}(k)-y(k)\|^2$$

调参寻优即为泛函极值问题：求 $f^*=\arg\min L(f)$

具体表达

（注：按照状态方程实际应该$y(k+1)$对应$u(k+1)$，但是对于一个系统预测其演化出于实际考量，可以写为我这里的样子，更加方便）

将上一段的分析具体用本例的变量写出来，本系统的离散化为：

$$\begin{pmatrix} q_1(k+1) \\ q_2(k+1) \end{pmatrix}=\varphi\left(\begin{pmatrix} q_1(k) \\ q_2(k) \end{pmatrix},\begin{pmatrix} q_1(k-1) \\ q_2(k-1) \end{pmatrix},\tau_1(k),\tau_2(k)\right)$$

因此虚拟系统为：

$$\begin{pmatrix} \hat{q}_1(k+1) \\ \hat{q}_2(k+1) \end{pmatrix}=f\left(\begin{pmatrix} q_1(k) \\ q_2(k) \end{pmatrix},\begin{pmatrix} q_1(k-1) \\ q_2(k-1) \end{pmatrix},\tau_1(k),\tau_2(k)\right)$$

变形整理成能在 matlab 里处理的形式，即：

$$\begin{pmatrix} \hat{q}_1(k+1) \\ \hat{q}_2(k+1) \end{pmatrix}=f\left(\begin{pmatrix} q_1(k) \\ q_2(k) \\ q_1(k-1) \\ q_2(k-1) \\ \tau_1(k) \\ \tau_2(k) \end{pmatrix}\right) \tag{3}$$

因此，看出待训练的神经网络的输入输出形状：输入为一个 6 维向量，输出为一个二维向量。

项目流程

今后只需要：

1. 在 Simulink 里把实际物理系统产生的序列数据$q_1(k), q_2(k), \tau_1(k), \tau_2(k)$导出来到工作空间
2. 将$q_1(k), q_2(k)$进行平移，得到$q_1(k-1), q_2(k-1), q_1(k+1), q_2(k+1)$这几个量，至此已经得到所有 8 个序列的数据
3. 根据 Equation 3 把这些数据拼成$6 \times N$的输入向量 input_data 以及$2 \times N$的 ground truth 标签向量 label_data
4. 启动 nftool 应用，导入 input_data 以及 label_data，根据该应用提示按按钮即可拟合
5. 将结果导出成 Simulink 模型，放回 Simulink 中测试，看看τ_1, τ_2变化后效果如何，若在各种输入下，神经网络输出和实际系统输出都很像，那么认为成功

为了方便操作，编写一个 data_preprocess.m 脚本，自动完成上述的 2,3 两个步骤。

```
%% data_preprocess.m
% 转置，得到行向量
q1 = out.q1';
q2 = out.q2';
% 使用向前平移一位的方法，构造 q1(k-1)以及 q2(k-2)
q1_shift = [0,q1(1:end-1)];
q2_shift = [0,q2(1:end-1)];
tau1 = out.tau1';
tau2 = out.tau2';

% 拼接成一个用于训练的输入数据
data_combine = [q1;q2;q1_shift;q2_shift;tau1;tau2];
% 去掉最后一列是为了和 label 对齐，因为最后一个时刻的 label 在未来，得不到
input_data = data_combine(:,1:end-1);

%ground truth 是下一个时刻的 q1 和 q2 两个角度，因此是后移一位：
q1_next = q1(2:end);
q2_next = q2(2:end);
% 拼接成一个用于训练的标签数据
label_data = [q1_next;q2_next];
```

神经网络训练

离散化说明

首先，把连续的物理系统和离散的神经网络分离清楚。我规定 Simulink 中导出的 To Workspace 模块的采样时间为 0.01 秒，即，我的项目中，神经网络根据“此时的输出”、“0.01 秒以前的输出”、“此时的输入”，来预测“0.01 秒以后的输出”。

训练过程

导入数据入 nftool 如图：

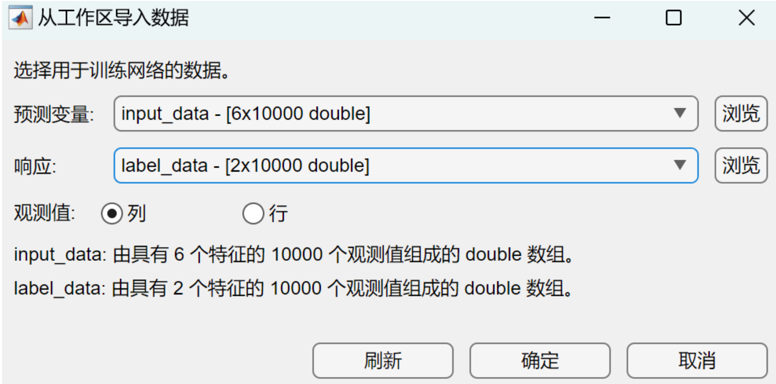


Figure 2: 数据导入

然后，默认 10 层网络，默认 70:15:15 的数据划分。点击训练按钮（用莱文贝格-马夸特法），训练结束后有以下可视化结果：

训练进度

单位	初始值	停止值	目标值
轮	0	325	1000
历时	-	00:00:12	-
性能	0.27	8.1e-09	0
梯度	0.749	9.92e-08	1e-07
Mu	0.001	1e-08	1e+10
验证检查	0	0	6

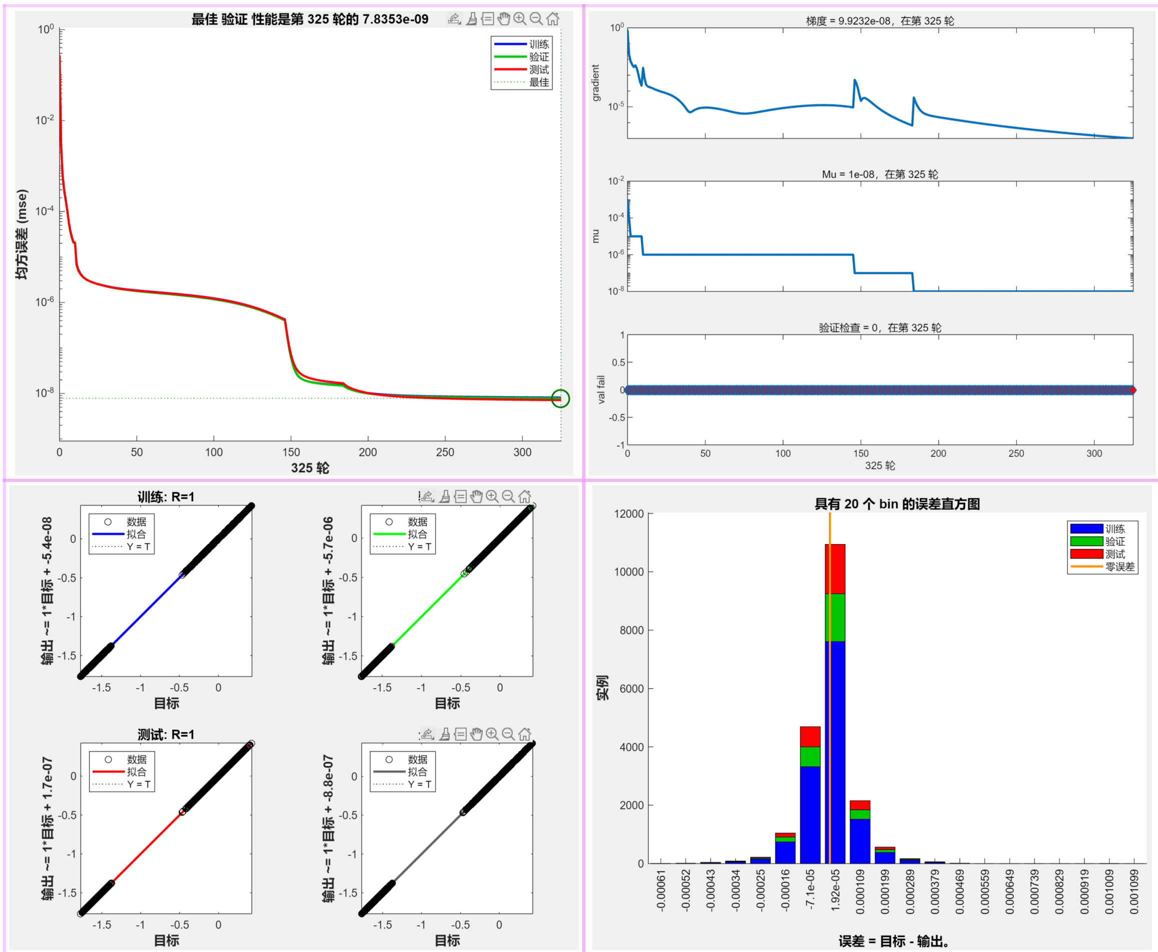


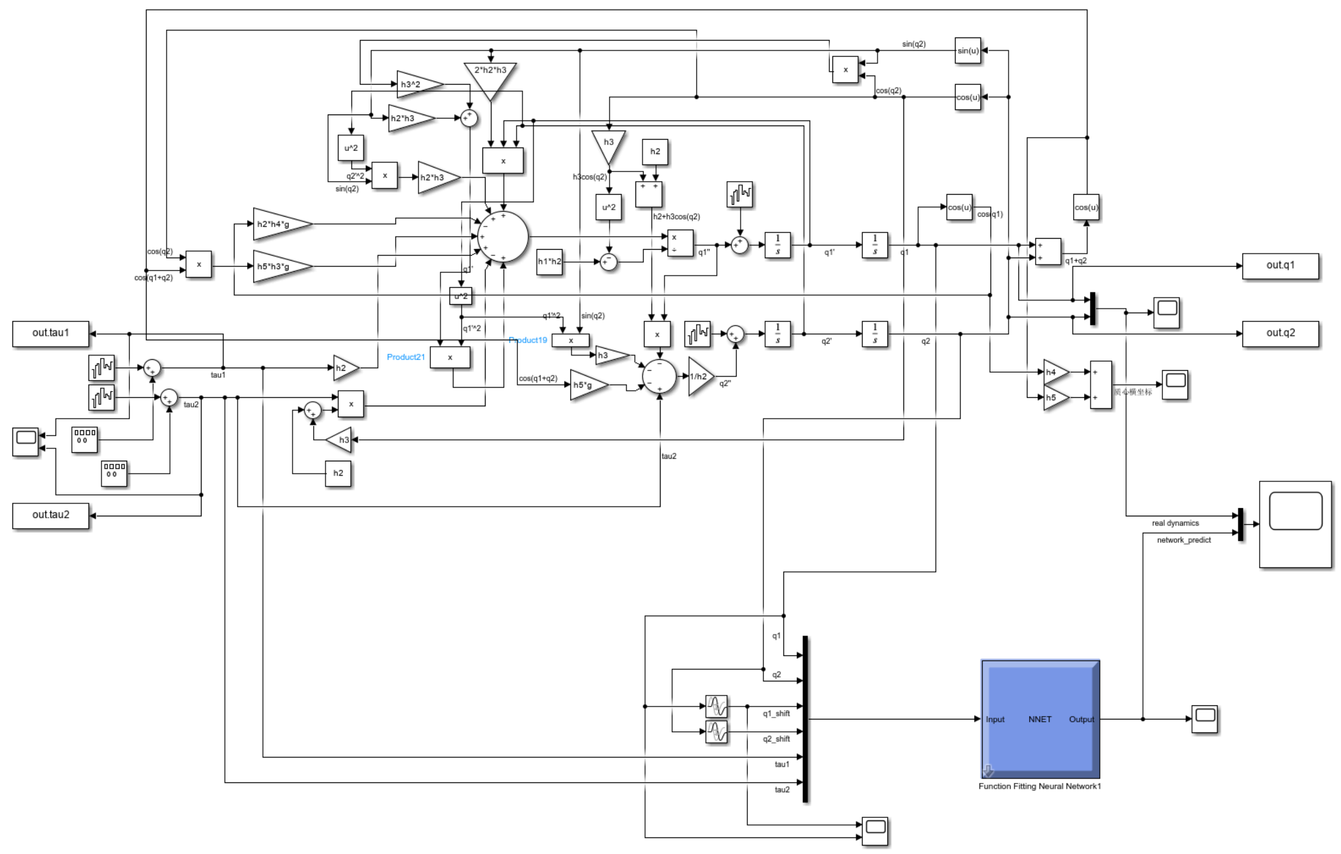
Table 1: 训练状态图

辨识效果

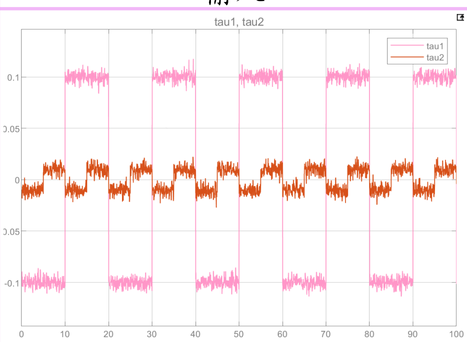
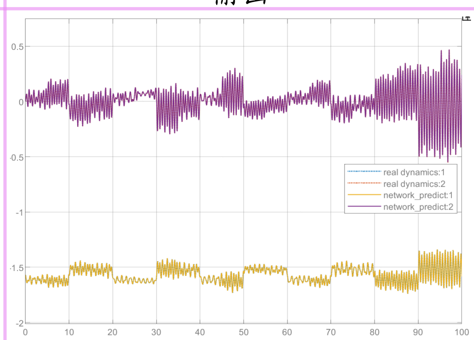
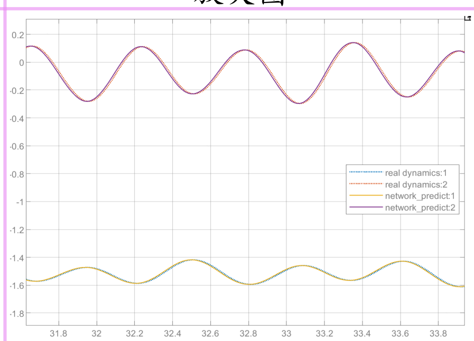
串并联系统

如下图，上方为真实系统，下方安装神经网络虚拟系统，该神经网络（蓝色方块）接受一个六维输入$\big(q_1(k)\ \ q_2(k)\ \ q_1(k{-}1)\ \ q_2(k{-}1)\ \ \tau_1(k)\ \ \tau_2(k)\big)^\top$，输出一个预测的$\big(\widehat{q}_1(k{+}1)\ \ \widehat{q}_2(k{+}1)\big)^\top$

Simulink



更改噪音的生成种子，并且修改输入形状如下：

输入	输出	放大图
		

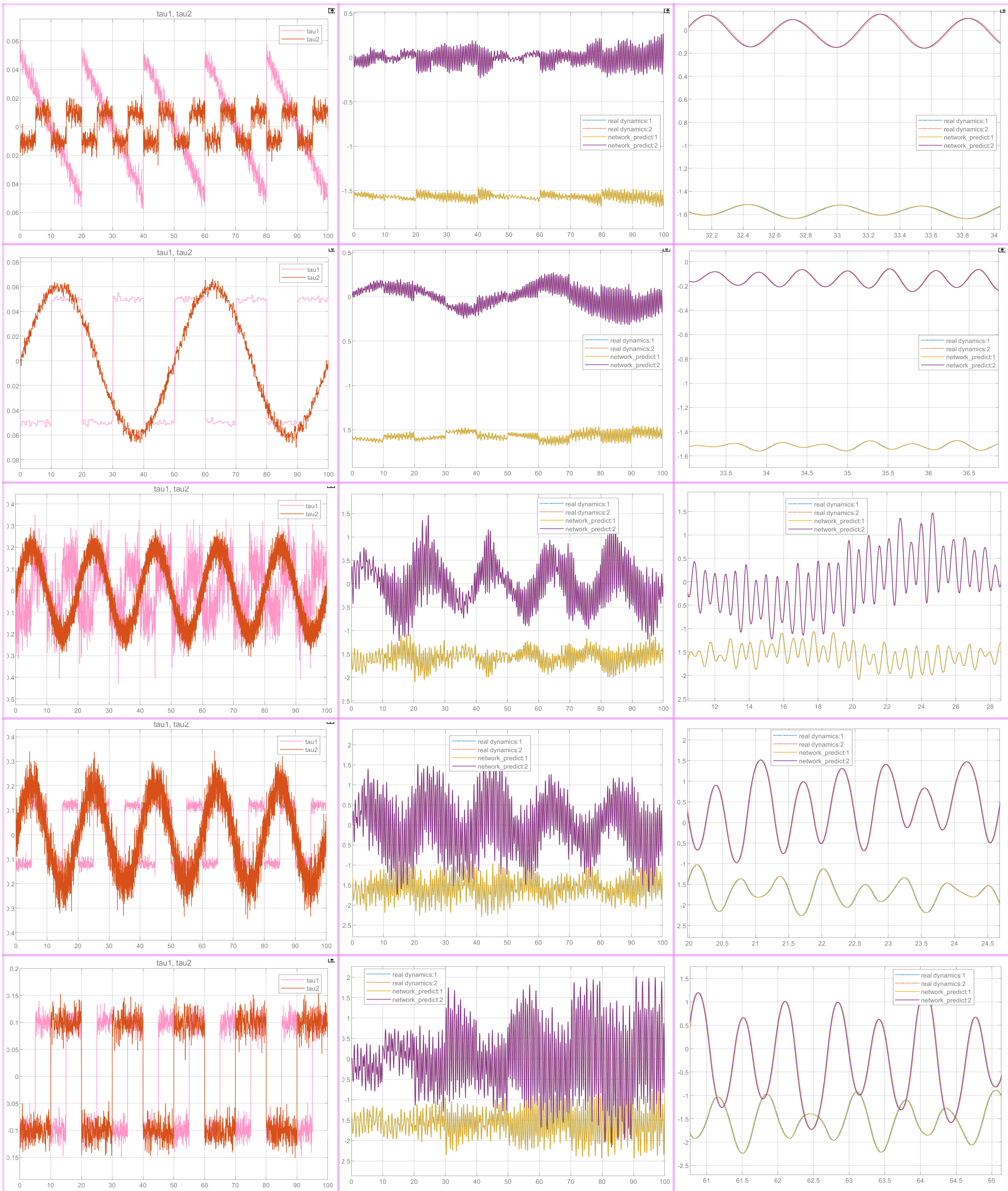


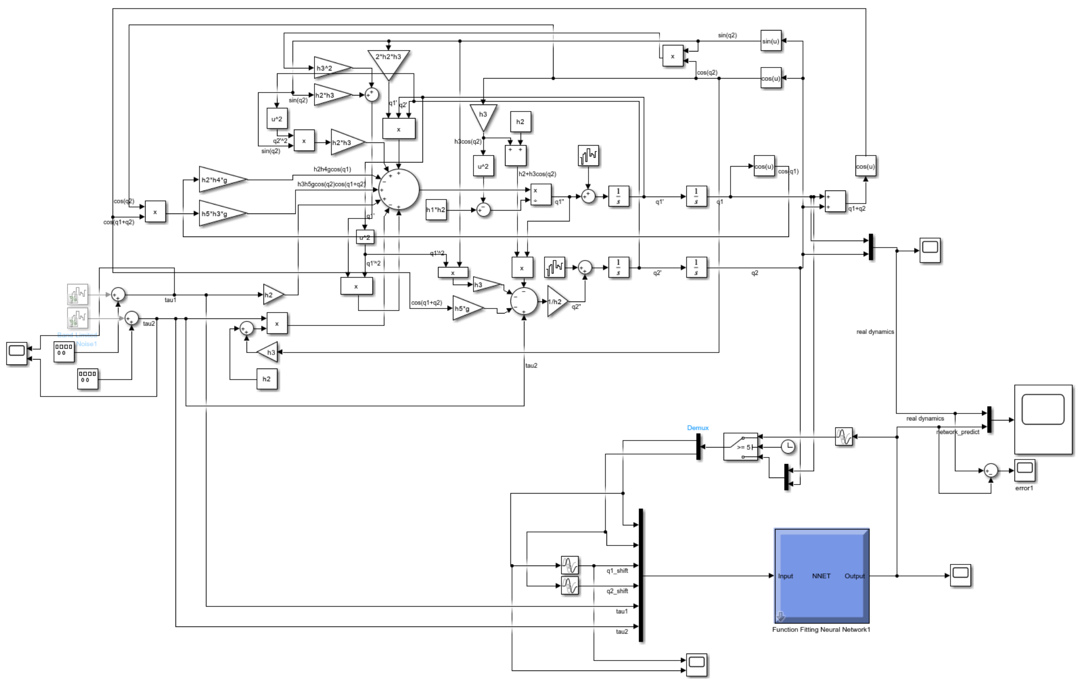
Table 3: 串并联结构网络拟合图

可见，在各种情况中，输出几乎都完全重合。说明神经网络的拟合效果还可以。但是，当本身的噪音选取过大或者恰好输入导致结果发散的话，那么预测误差也会发散。

并联模型

我尝试用一下并联模型，搭建模型如下：唯一的修改就是右下角部分，q_1, q_2这两路的来源，现在改为前 5 秒来自真实系统，后 5 秒来自神经网络自己的输出。这是模拟一开始学习，后来神经网络与原系统脱离的情景。（前 5 秒必须来自原始系统的原因是神经网络初始的输出是很奇怪的，修改了$q_1(0)$并不能让神经网络意识到这一点，所以先得跟踪一会）

Simulink



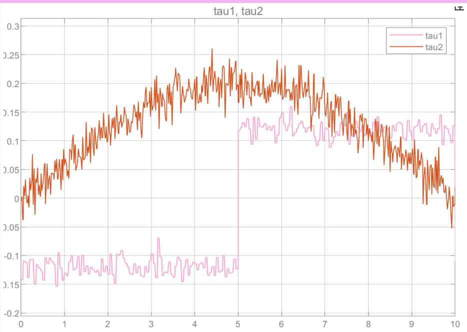
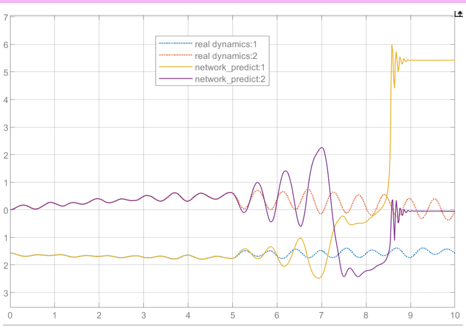
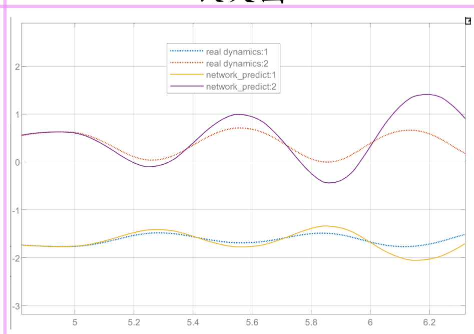
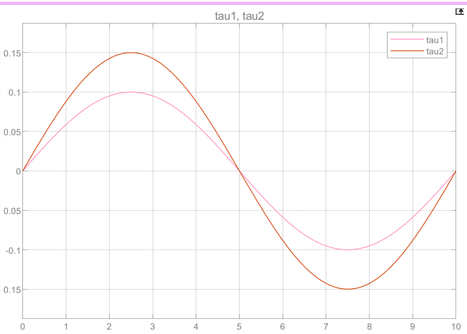
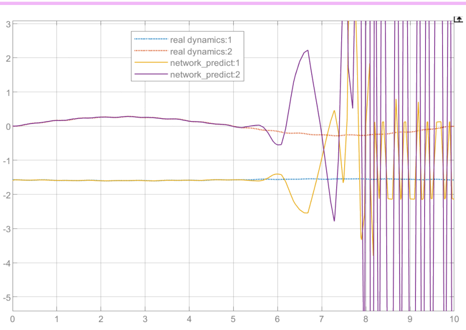
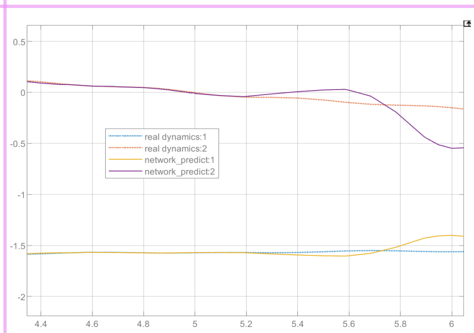
输入	输出	放大图
		
		

Table 4: 并联结构网络拟合图

可见，这个系统的非线性性非常强，随着几轮迭代，输出越差越大。这类似于蝴蝶效应。

加入噪声的数据训练出的网络

假设考虑到实际情形，从机械臂的传感器得来的角度是有噪音的（体现在两个角加速度那一项受到额外的扰动），具体表现就是打开之前流程图里$\ddot{q}_1$和$\ddot{q}_2$处的两个白噪声，这样得到的数据再进行训练。之所以情况不同，就是原本神经网络学的是一个力学函数关系，但现在不是一个完美的函数关系，会有一些噪点，网络可能学坏。

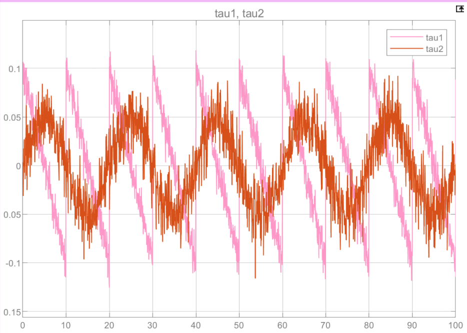
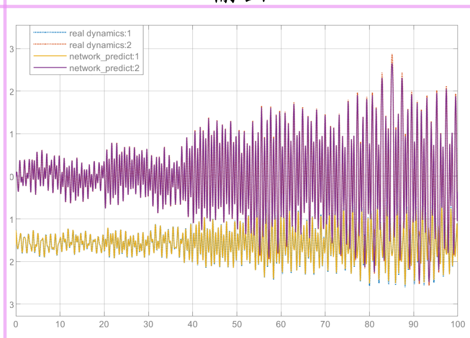
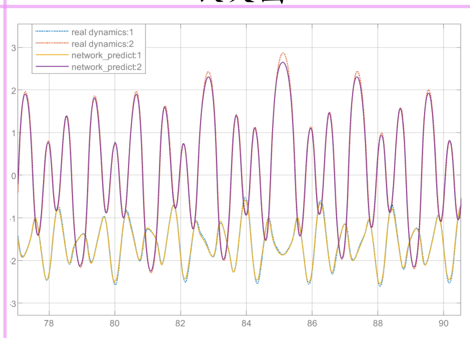
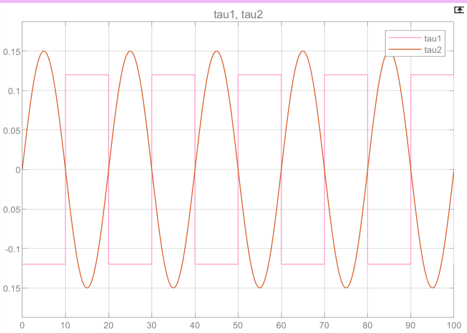
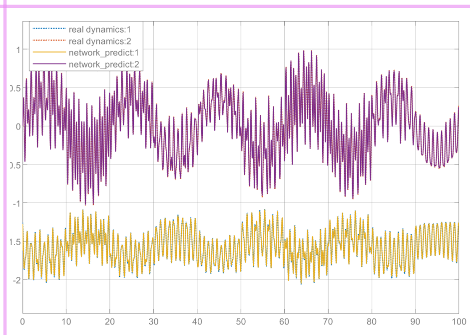
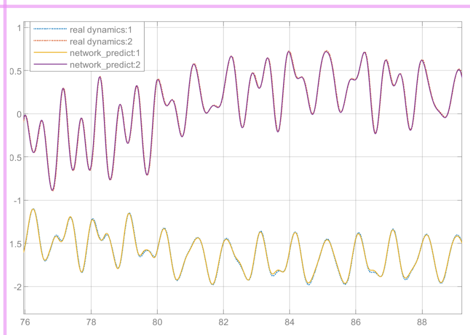
输入	输出	放大图
		
		

Table 5: 串并联结构，用带噪音的数据训练的网络拟合图

可以看到，效果稍微差一些，仔细看放大图能发现明显的偏离。在上表第二种输入下，对比旧网络与现在这个网络拟合表现如下。用含噪音的数据训练出来效果略差一些（也可能是运气问题）。

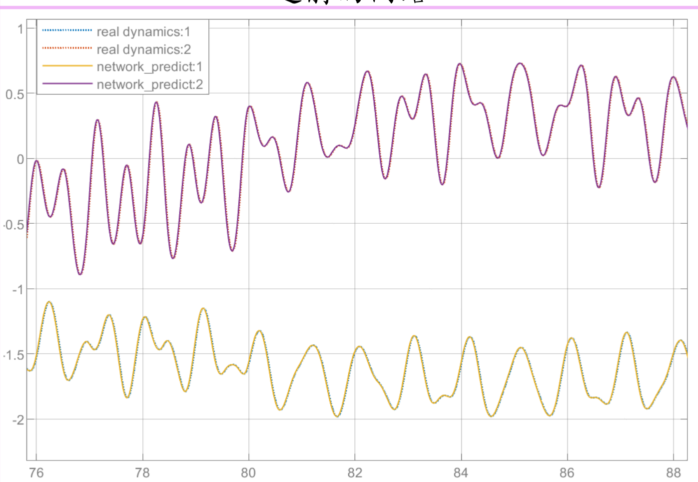
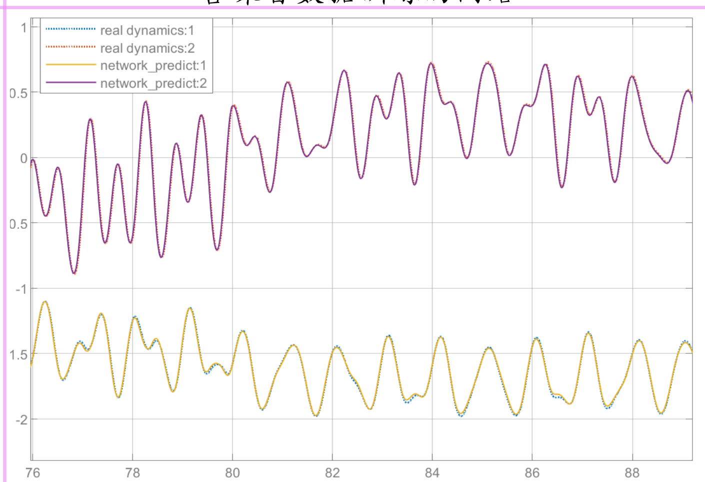
之前的网络	含噪音数据训练的网络
	

Table 6: 用带噪音与不带噪音的数据训练的两个网络对比

代码附件与运行说明

> 使用的 Matlab 版本为 2023b

附件含有：
> levitate_model.slx 仿真模型
> levitate_param.m 含有参数
> levitate_controller.fis 模糊控制器
> levitate_controller2.fis (输出论域偏移的模糊控制器)
> levitate_controller3.fis (模糊 PID 尝试，但参数调不好)

测试时，请先运行 levitate_param.m 文件（使得 workspace 中有参数的数据）然后即可运行 slx 文件。slx 文件中含有两组，一组是普通 PID（对照组，说明无稳态工作点较难调），一组是模糊控制。

如有疑问，请访问 https://github.com/Maythics/Control_simulink.git