

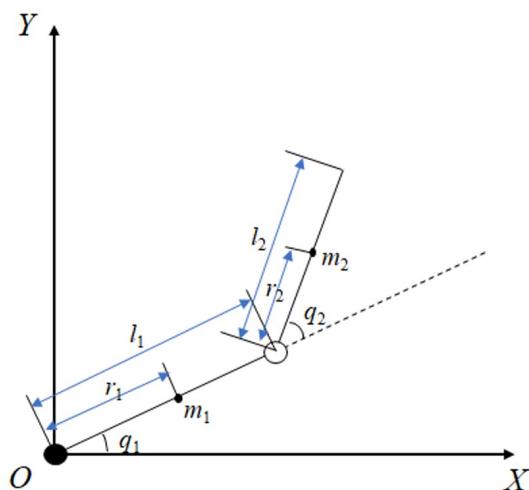
基于神经网络的二机械臂模型系统辨识

自动化(控制) 石景元 3220103824

1 背景介绍

神经网络辨识是一种利用人工神经网络对复杂系统进行建模和参数估计的方法，它通过学习系统的输入输出数据，构建一个能够近似系统行为的模型。这种方法特别适用于那些难以用传统物理模型描述的非线性、动态和复杂系统。在本报告中，我们将探讨如何应用神经网络辨识技术来识别和模拟一个二自由度机械臂的动力学行为，该机械臂的动力学方程由惯性矩阵、科氏力和向心力的结合矩阵以及重力势能矩阵共同决定。通过训练神经网络以匹配这些动力学参数，我们能够创建一个虚拟模型，用以预测机械臂在给定输入力矩下的角度响应，进而验证模型的准确性和可靠性。

在本实验中，我们采用了**两种**实现方式，分别是以时序状态和输入力矩为网络输入、以当前状态为网络输出，和以各阶差分和输入力矩为网络输入，以前向差分为网络输出。我们对上述两种方案各训练了一个神经网络，并比较了两种方案性能。



2 问题分析

二机械臂模型系统以输入力矩 $\tau = [\tau_1, \tau_2]^T$ 为输入，以 $q = [q_1, q_2]$ 为输出，我们令 $u = \tau = [\tau_1, \tau_2]^T$, $y = [q_1, q_2]^T$, $X = [q_1, q_2, \frac{dq_1}{dt}, \frac{dq_2}{dt}]^T$, 从而我们得到系统的状态空间模型

$$\begin{cases} \dot{X}(1) = X(3) \\ \dot{X}(2) = X(4) \\ \dot{X}_{3:4} = M^{-1}(X_{1:2})(\tau - G(X_{1:2}) - C(X_{1:2}, X_{3:4})X_{3:4}) \end{cases}$$

我们在matlab的 `system_transfer.m` 中实现了这个系统的物理模型，其中sfunction的 `mdlDerivatives` 函数如下

```
function sys=mdlDerivatives(t,x,u)
h1 = 0.0308;
h2 = 0.0106;
```

```

h3 = 0.0095;
h4 = 0.2086;
h5 = 0.0631;
g = 9.8;

q1 = x(1); q2 = x(2) ; dq1 = x(3); dq2 = x(4);tau = u;
m11 = h1 + h2 + 2*h3*cos(q2);
m12 = h2 + h3*cos(q2);
m21 = h2 + h3*cos(q2);
m22 = h2;
M = [m11 m12;
      m21 m22];

c11 = -h3* sin(q2)*dq2;
c12 = -h3* sin(q2)*(dq1+dq2);
c21 = h3*sin(q2)*dq1;
c22 = 0;
C = [c11 c12;
      c21 c22];

g1 = h4*g*cos(q1) +h5*g*cos(q1+q2);
g2 = h5*g*cos(q1+q2);
G = [g1;g2];

ddq = M\(\tau-C*[dq1;dq2]-G);
ddq1 = ddq(1);
ddq2 = ddq(2);

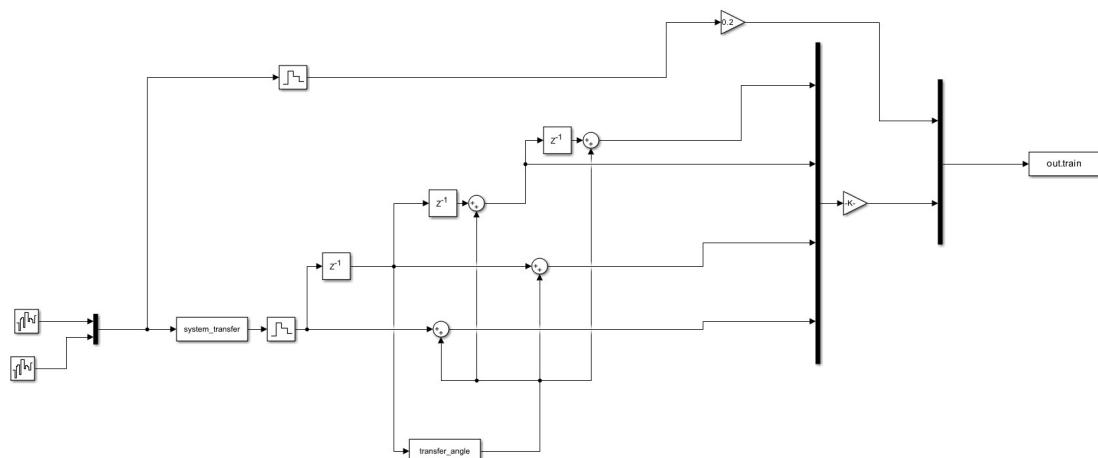
sys = [dq1;dq2;ddq1;ddq2];

```

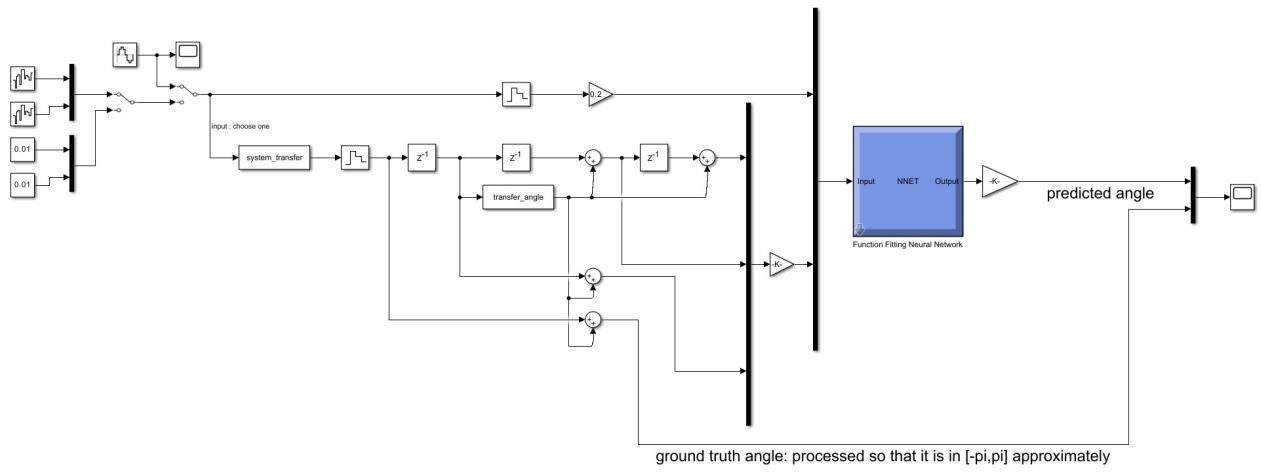
3 神经网络系统辨识建模: 时序状态作为网络输入

3.1 仿真simulink图

训练simulink仿真



测试simulink仿真



如图所示，图中左侧为系统输入，

3.2 数据预处理

主要是对数据的归一化。具体对 q 和 τ 的预处理情况如下：

对 q 的预处理：

- 确保 $q(k-1)$ 主值化，即 $q(k-1)$ 在 $[-\pi, \pi]$ 内， $q(k-2), q(k-3), q(k)$ 与 $q(k-1)$ 的差值保持不变，具体是通过 `transfer_angle` 计算 $q(k-1)$ 主值化需要加上的变化值 $\Delta q(k-1)$ ，然后其余时间步的 q 均加上这个值
- 对角度 q 归一化，即乘以 $1/\pi \approx 0.318$
- 输出的角度乘以 π ，使得角度在 $[-\pi, \pi]$ 内

对 τ 的预处理：

- 对输入乘以0.2，经观察此后数据基本落入 $[-1, 1]$ ，即完成归一化

3.3 训练细节

系统说明

输入信号	网络输入	网络输出
可以选择正弦、常数、白噪声输入	$q(k-1), q(k-2), q(k-3), \tau$	$q(k)$

参数设置

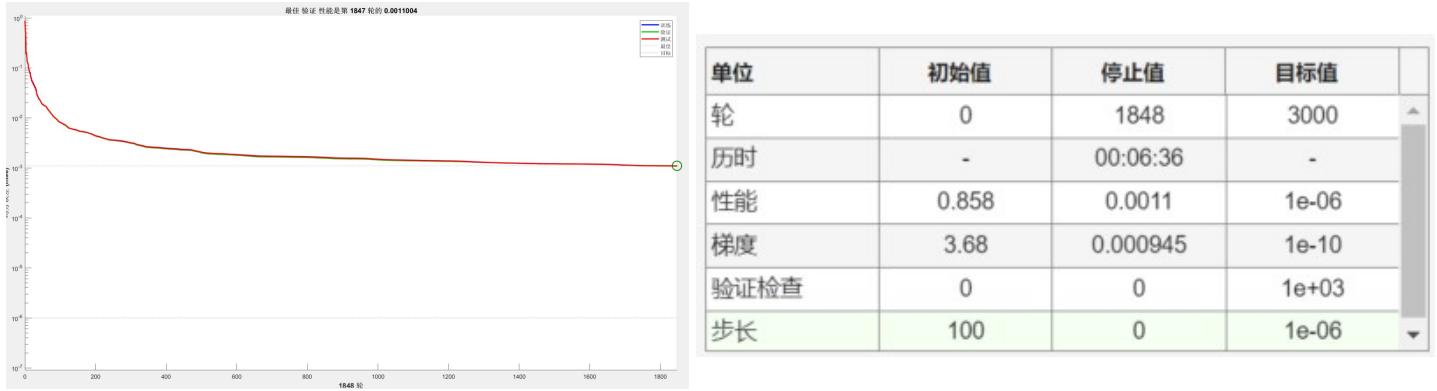
训练方式	网络结构	训练数据采样方式	输入信号
<code>trainFcn = 'traincgb'</code> (支持 GPU训练)	<code>hiddenLayerSize = [20,40,20,10];</code>	时长400s,采样间隔0.01s	功率0.004, 采样时间0.001, 系统采样时间0.01

- 注：采样间隔是采样时间的十倍，是为了充分保证数据的各个分量的随机性

训练结果

训练曲线

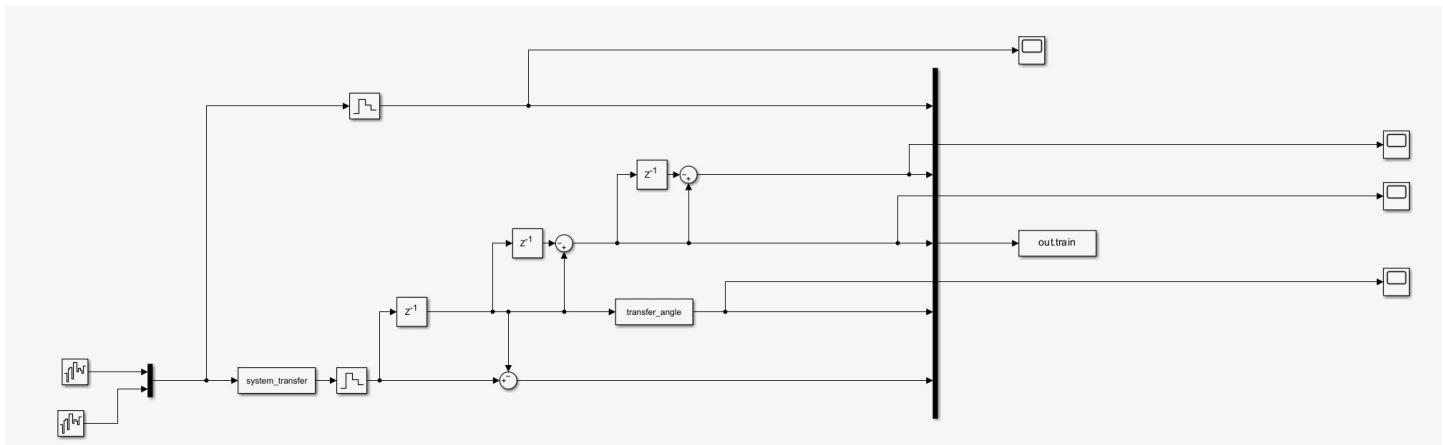
表格



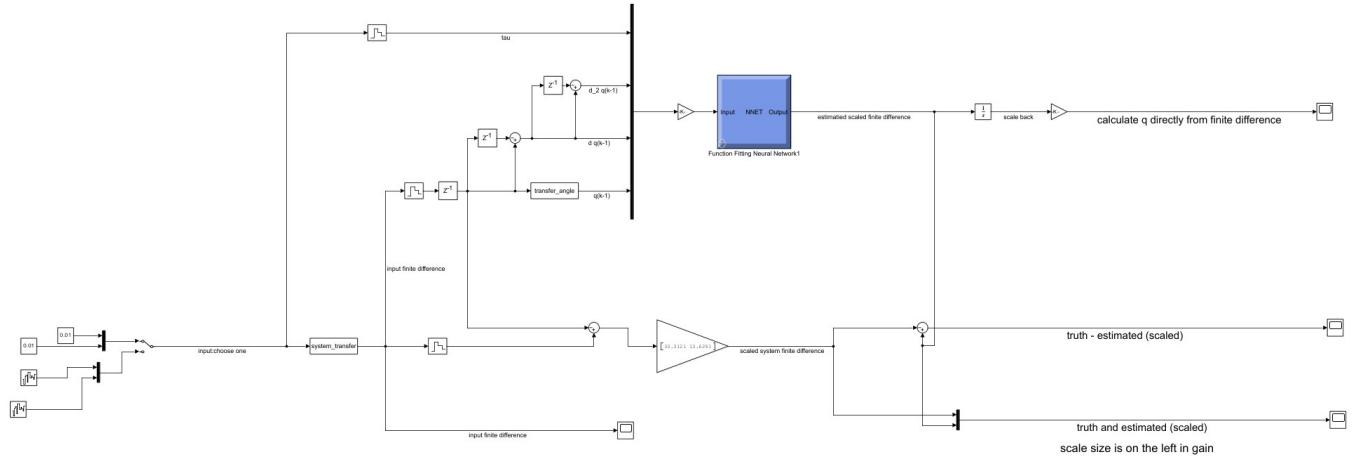
4 神经网络系统辨识建模: 各阶差分作为网络输入

4.1 仿真simulink图

训练simulink仿真



测试simulink仿真



其中，神经网络的输入自上而下分别为第k时间步的 $\tau(k)$, $\Delta_2 q(k-1)$, $\Delta q(k-1)$, $q(k-1)$,共8个维度；输出为 $\Delta q(k)$,共两个维度。

4.2 数据预处理

主要是对数据的归一化。具体事项方式如下：

对神经网络输入的预处理：

- $q(k - 1)$ 先映射到主值空间，再映射到 $[-1, 1]$
- 其余输入维度方差归一化

对输出的预处理：

- 输出维度进行大致的方差归一化

4.3 训练细节

系统说明

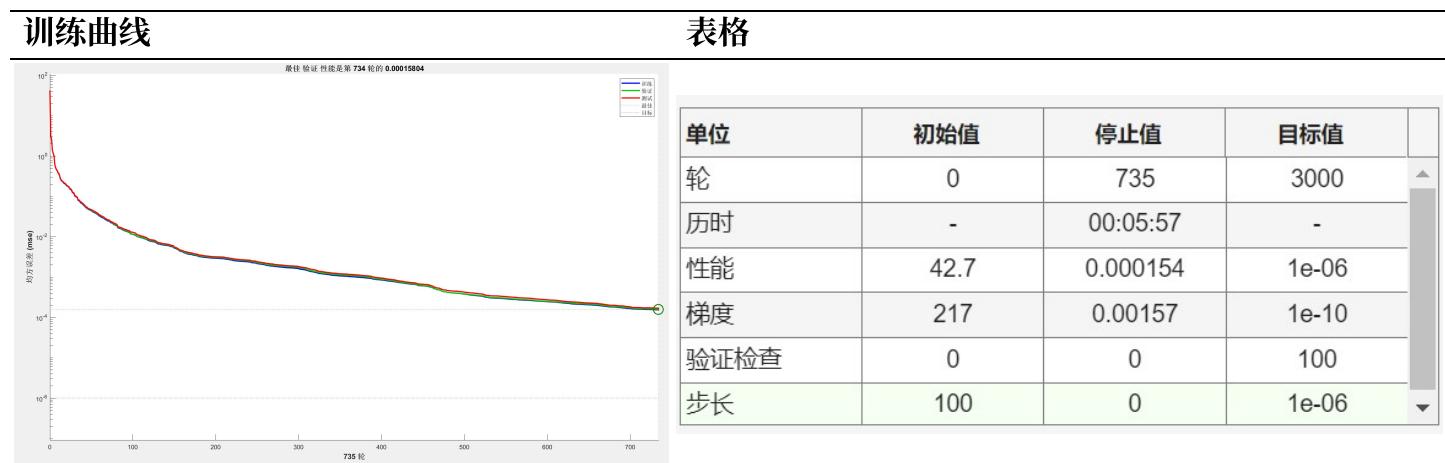
输入信号	网络输入	网络输出
可以选择常数、白噪声输入	$\tau(k), \Delta_2 q(k - 1), \Delta q(k - 1), q(k - 1)$	$\Delta q(k)$

参数设置

训练方式	网络结构	训练数据采样方式	输入信号
trainFcn = 'traincgb'(支持 GPU训练)	hiddenLayerSize = [20,40,60,20];	时长400s,采样间隔0.01s	功率0.004, 采样时间0.001, 系统采样时间0.001

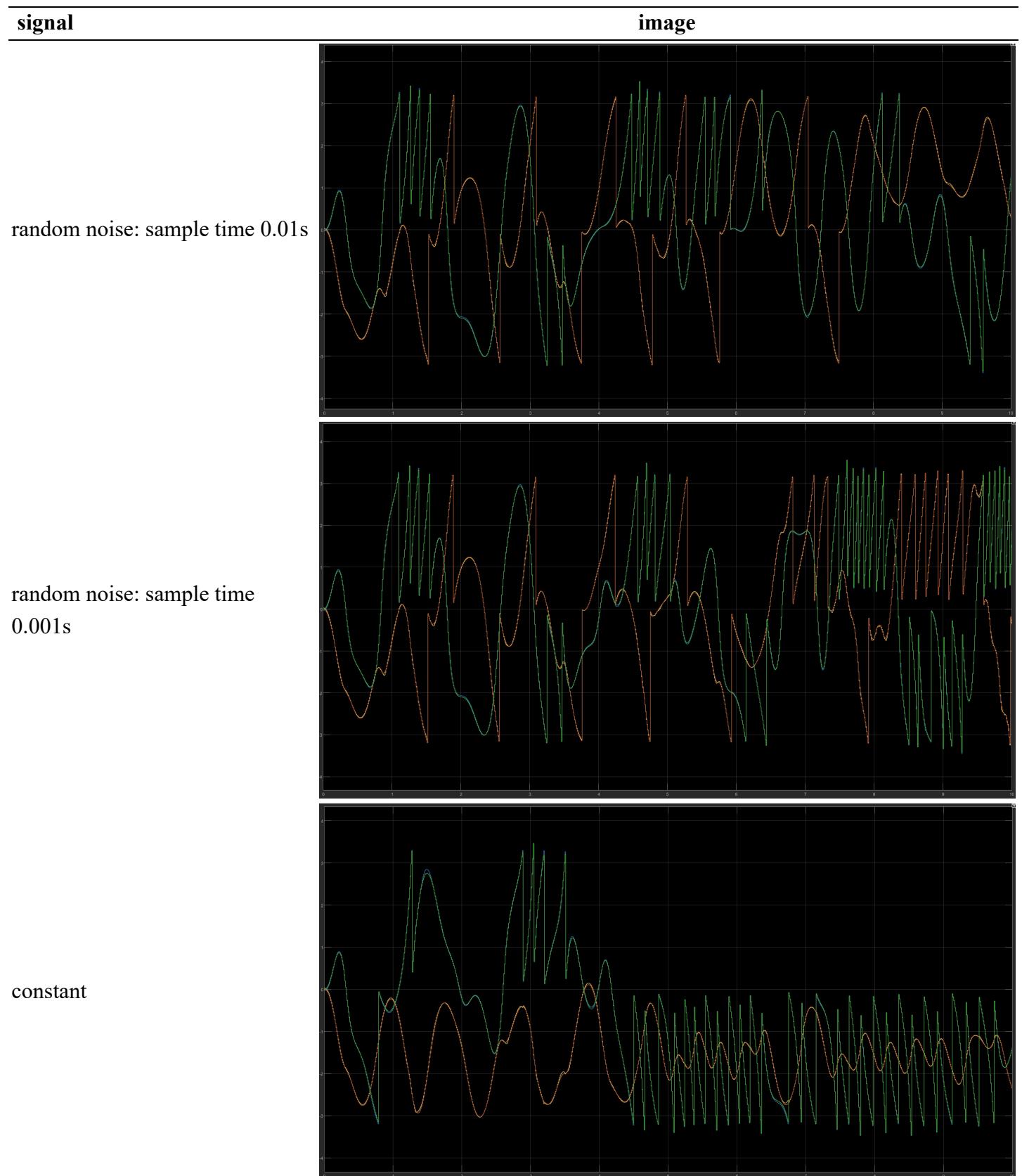
- 注：采样间隔是采样时间的十倍，是为了充分保证数据的各个分量的随机性

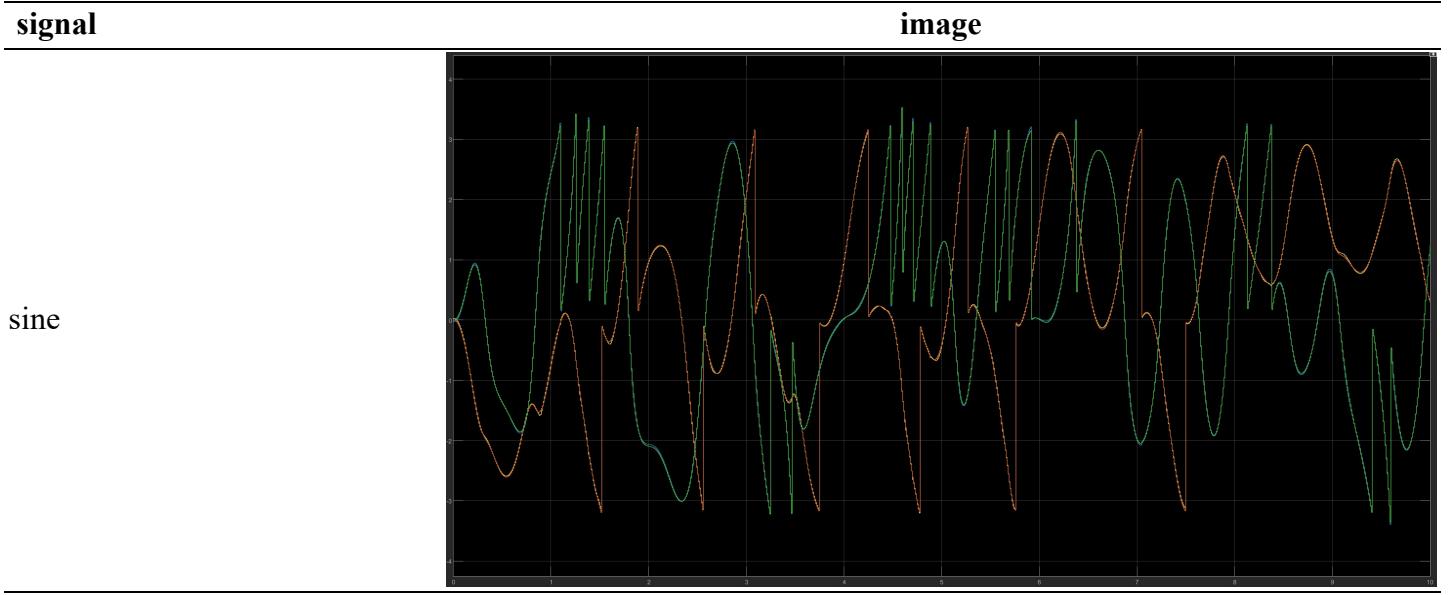
训练结果



5 仿真结果(对比基于时序的预测和基于各阶差分的预测)

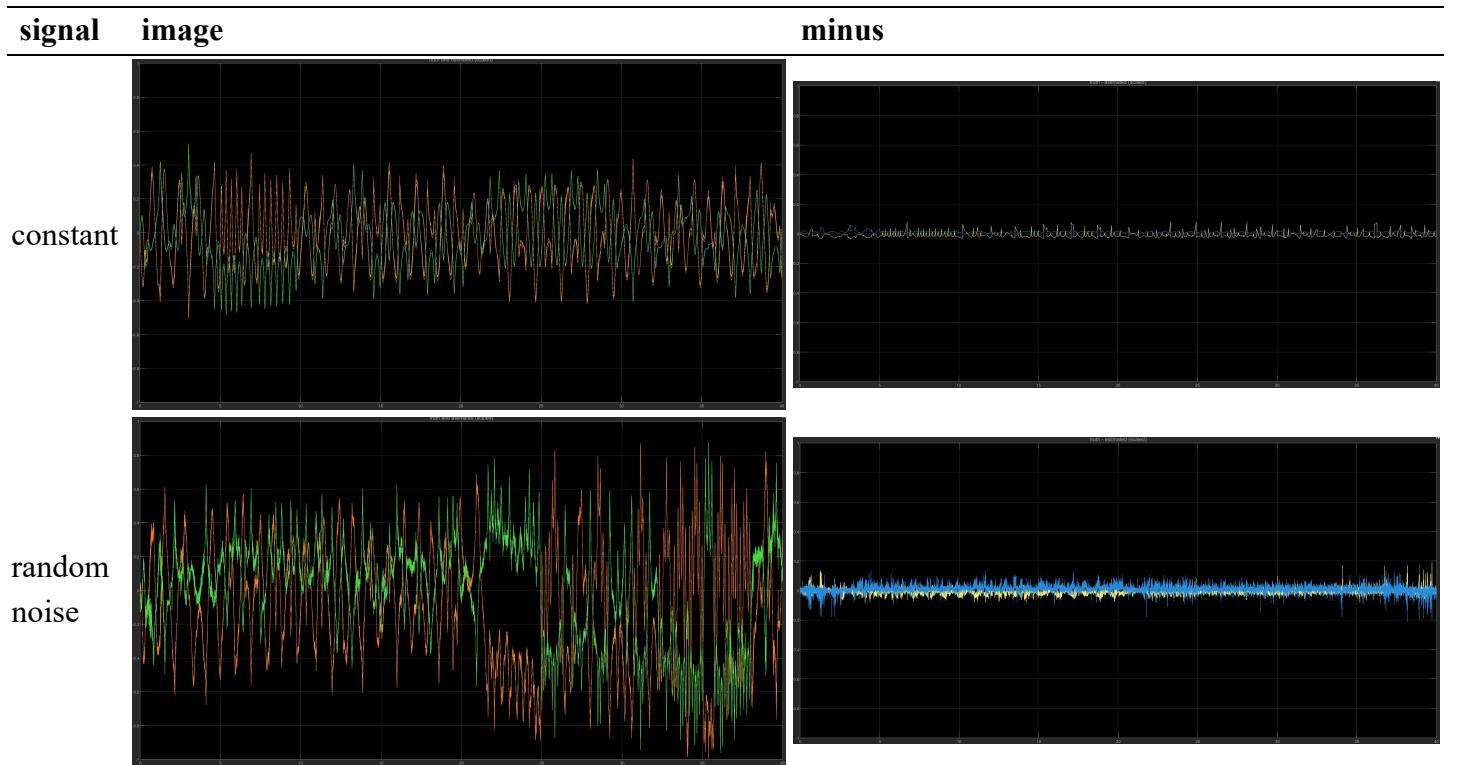
5.1 基于时序状态的神经网络





上图的角度以弧度为单位，都经过了主值化，可以从上图看出，神经网络对于系统辨识效果很好，对于串并联测试能够有效预测系统输出。

5.2 基于各阶差分的神经网络



从上图可以看出，尽管已经对输入输出数据进行了归一化或方差归一化以利于训练，但是事实上当常数值输入时，系统预测得到的噪声已经很明显了，而当随机噪声输入时，系统预测的输出存在较大误差，最大可以达到20%左右。尽管如此，这并不能说明基于各阶差分的神经网络系统辨识效果差很多，图像上展示的时被放大的在0.001s内的真实差分和预测得到的差分，两者本身大小不大，缩小至原值后真实误差绝对值不超过0.02，经过0.01s的误差累积不超过0.2。而我们发现，对于基于时序信号预测的系统，其对于随机噪声、三角函数和常值输入的误差的最大值均已超过了0.06，有时则会超过0.15；这说明事实上，两者的预测误差在同一个数量级，但粗估下基于时序的神经网络预测效果更好，理由如下：

- 基于时序的神经网络更容易对角度进行归一化，而基于差分的神经网络对于差分的归一化受差分随机性的影响较大，要同时考虑差分分布的范围和方差，对于不同时长这个值是不相同的，这给数据归一化带来了很大的麻烦。
- 基于时序网络在进行归一化后，相同性质的数据仍处在同一量级，更有利于利用相关量之间的信息，学习到目标值的映射关系。

5.3 并联测试方式讨论

事实上，我们尝试过并联测试方案，但是最后并没有成功。我们发现，尽管在串并联下每一个预测和真实值的差距都很小，但是10s内0.001s每次的时间步迭代产生的误差已经不可忽视。因此，上述方法并不能成功解决并联误差累积的问题

6 对比实验

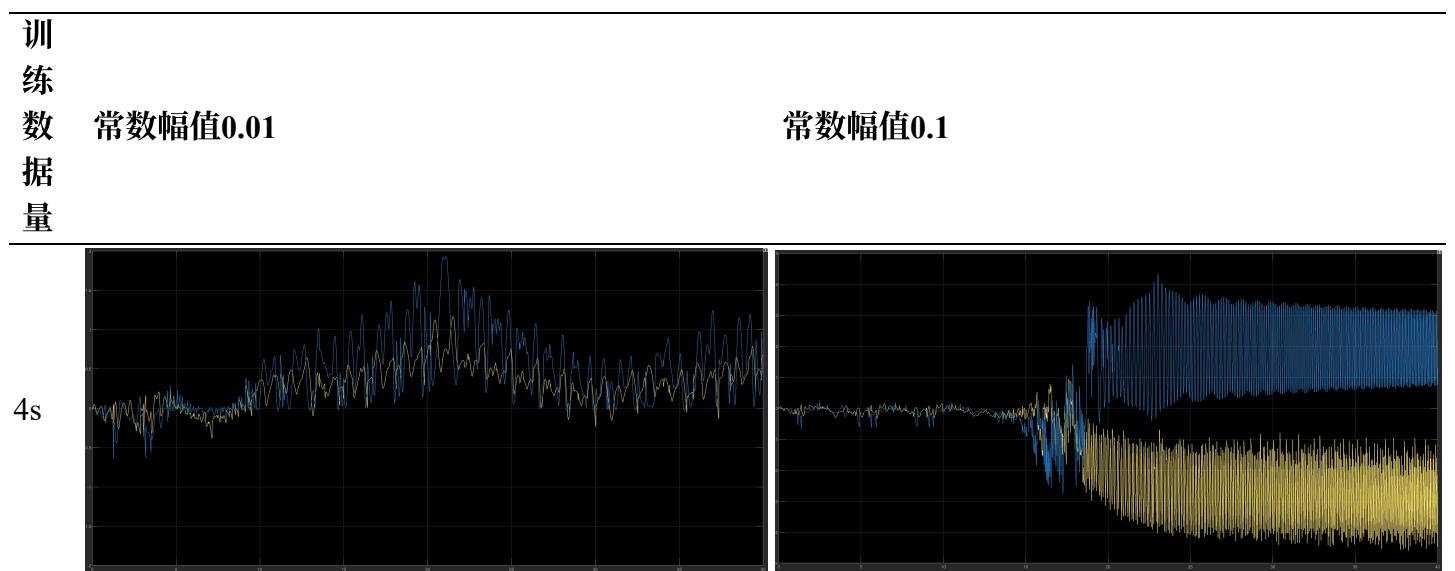
根据老师的要求，在这一部分中，我们着重比较系统对不同的数据量、神经网络规模的神经网络，预测值和真实值的差别。根据我们上述分析，基于时序的神经网络效果相对较好，而且对于不同输入数据规模不需要调整归一化系数，因此更加有利于分析。因此我们选择基于时序的神经网络完成任务，并分数据量和神经网络规模进行分析：

6.1 数据量

系统的采样频率是0.01s，因此我们直接考虑仿真物理系统的仿真时间t作为变量，t到数据量的转换公式为 $N = 100t(s)$

我们固定神经网络 `hiddenLayerSize = [20, 40, 20, 10];`，下面是我们的对比结果：

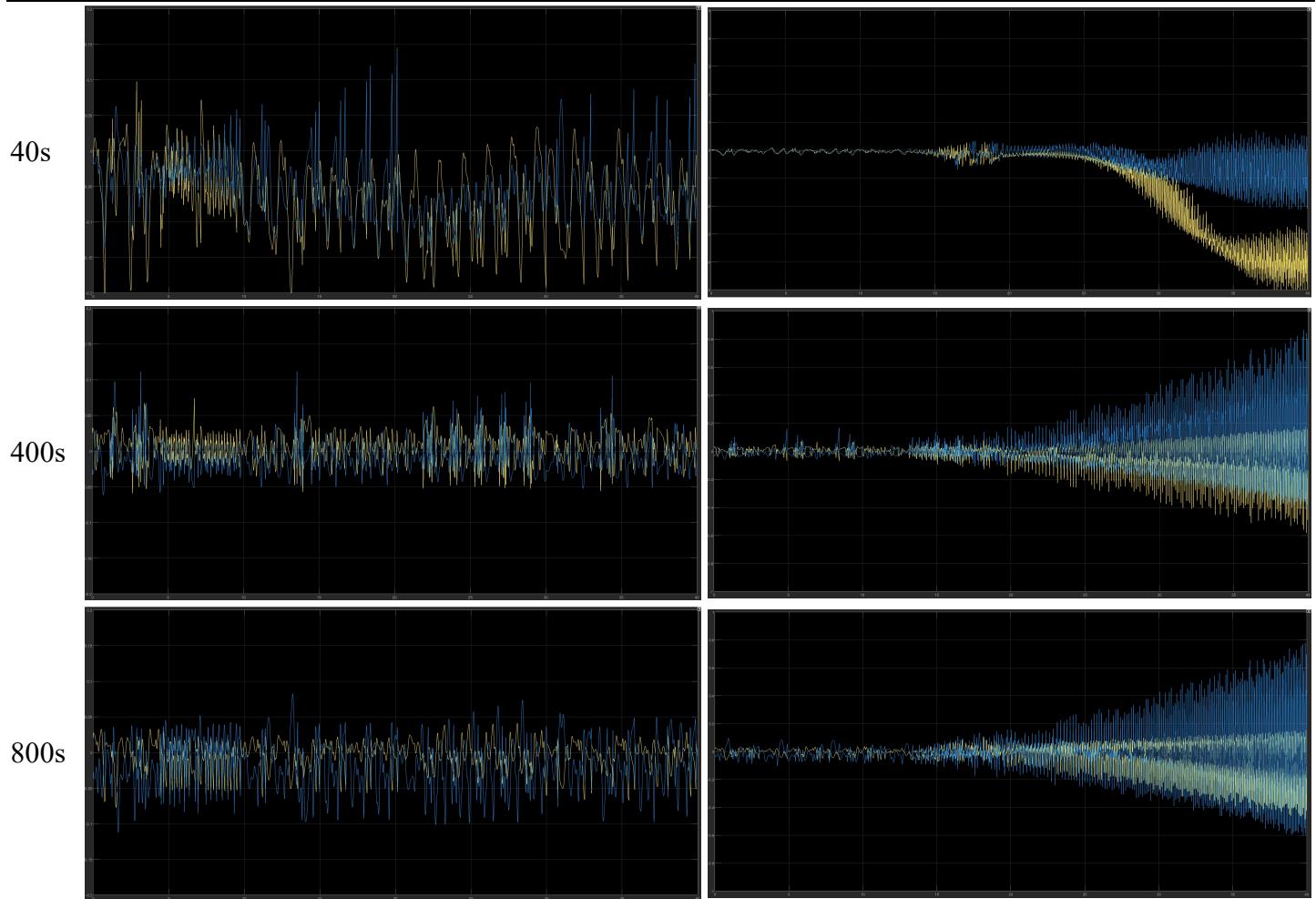
常数输入



训
练
数
据
量

常数幅值0.01

常数幅值0.1

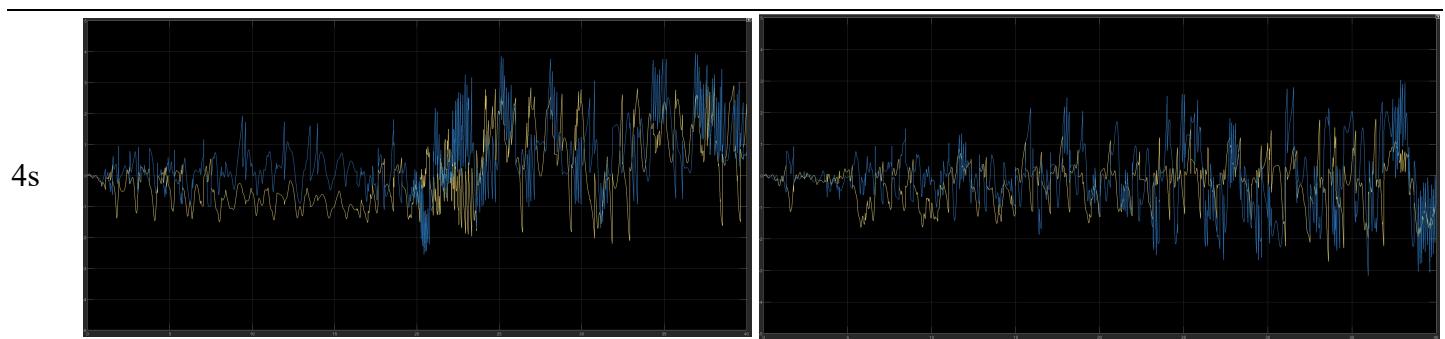


正弦输入 角速度2rad/s

训
练
数
据
量

振幅0.2

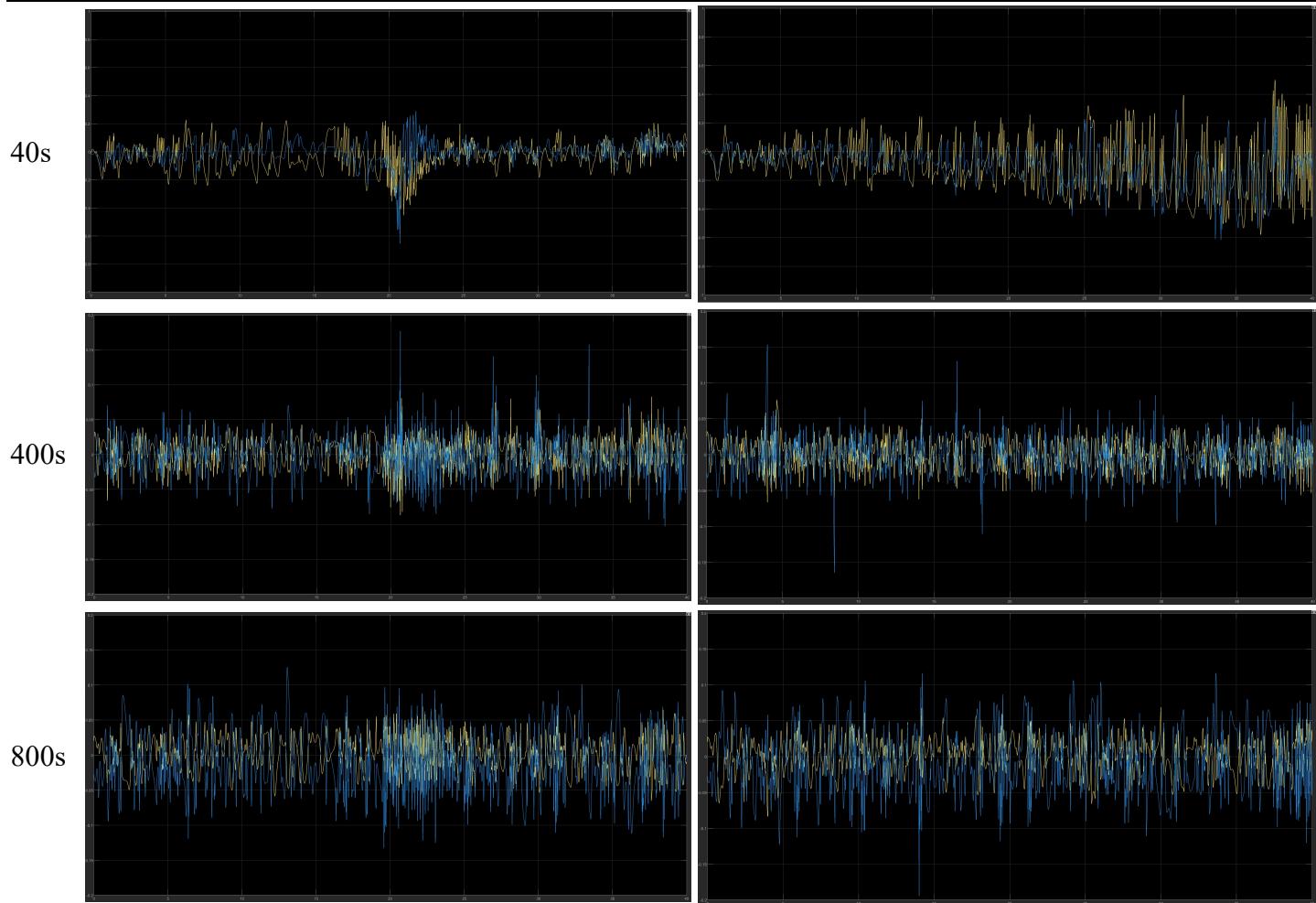
振幅0.02



训练
数据
量

振幅0.2

振幅0.02

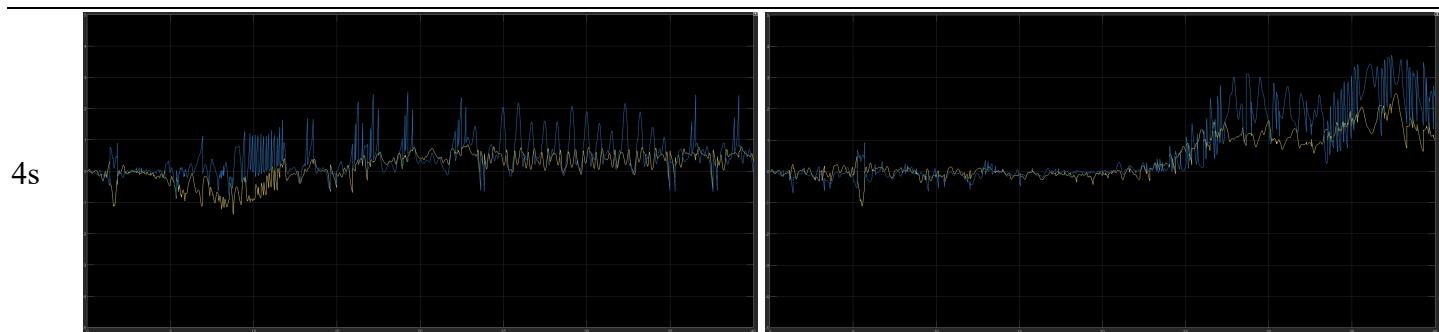


幅值0.02

训练
数据
量

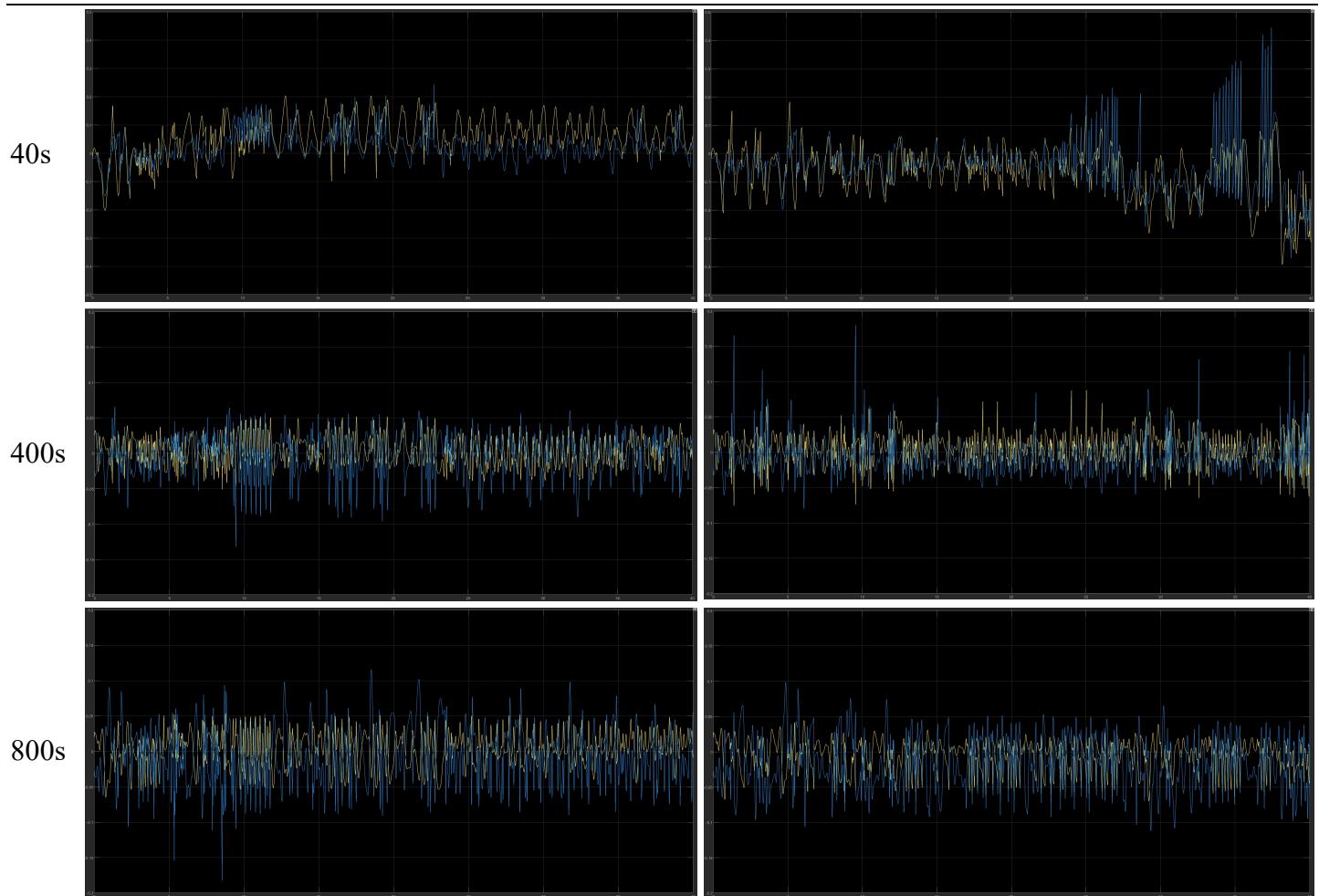
角速度4rad/s

角速度1rad/s



角速度4rad/s

角速度1rad/s



可以得出一下结论:

- 对于同一输入幅值，接受的训练数据越多，系统的预测误差基本上越小

由于输入数据量的增加，输入数据的分布更接近真实分布，因此学习到的映射关系更加符合真实的映射关系。

- 对于不同的输入幅值，如果输入幅值相对于训练数据有较大的偏差，神经网络无法做很好的预测。说明实际上该简单的神经网络只能学习原映射关系的一部分，泛化性并不强。这和其它的应用场景基本相近。

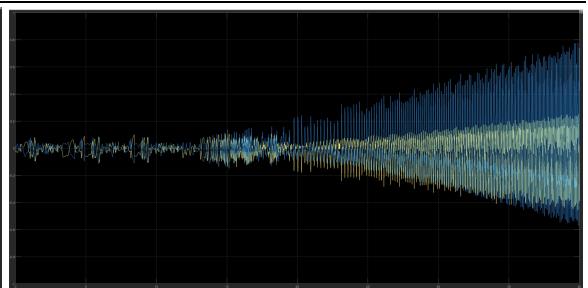
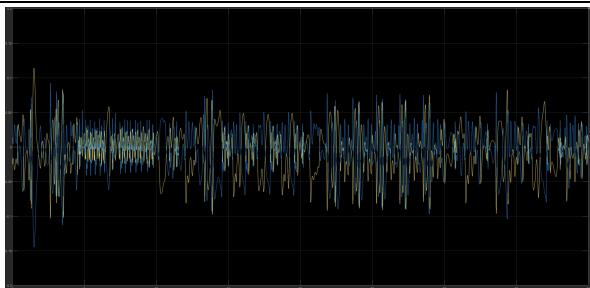
6.2 神经网络结构

常数输入

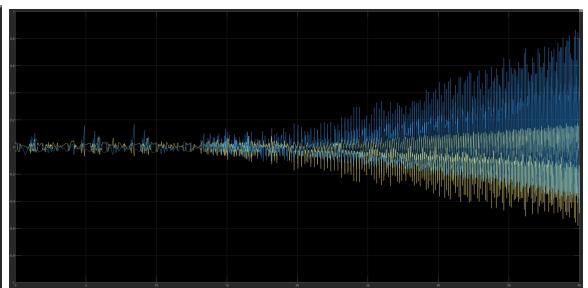
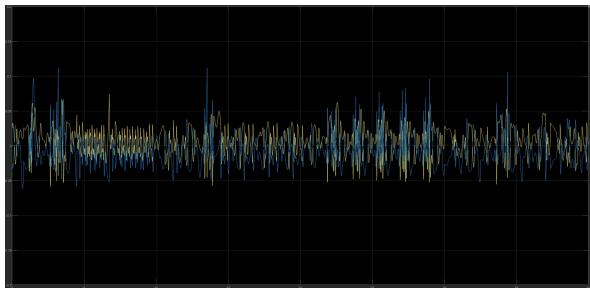
hiddenLayerSize 常数值0.01

常数值0.1

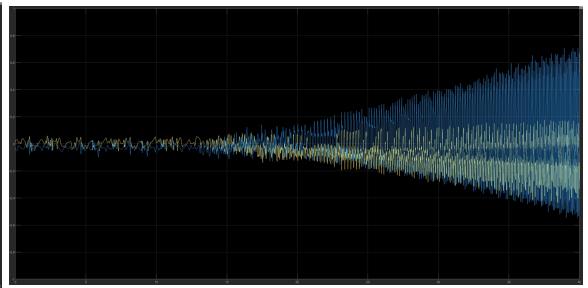
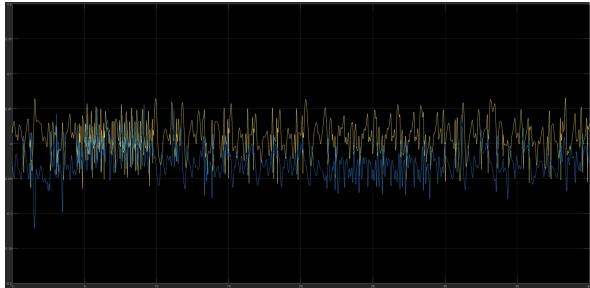
[20,10]



[20,40,20,10]



[20,40,60,40,20,10]

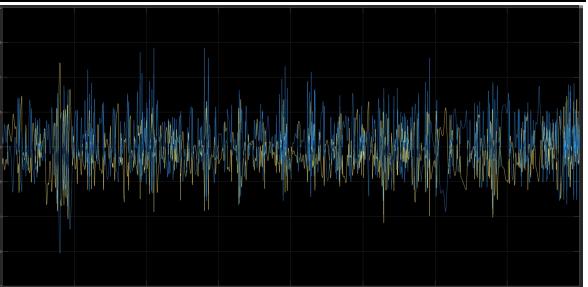
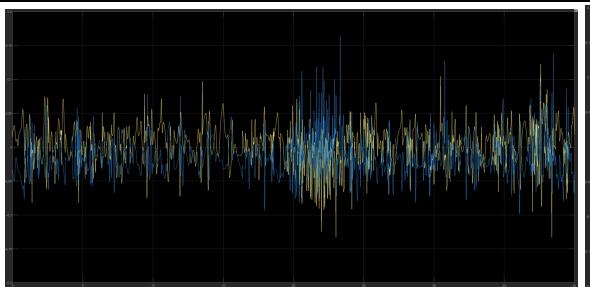


正弦输入 角速度2rad/s

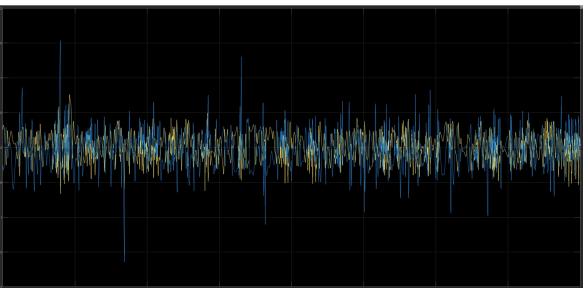
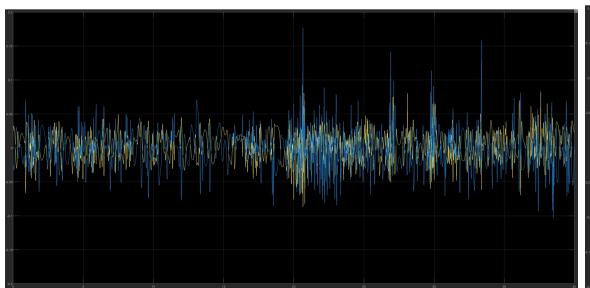
hiddenLayerSize 振幅0.2

振幅0.02

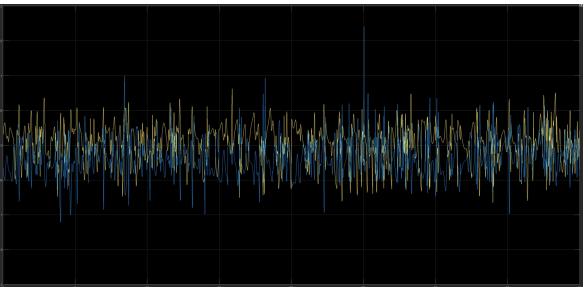
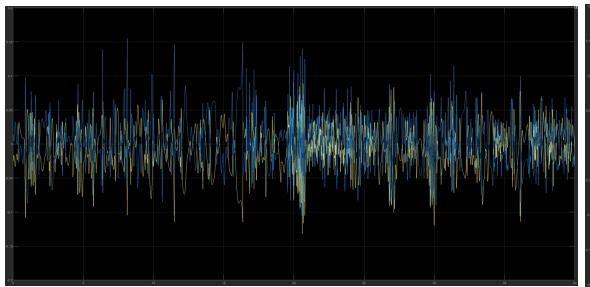
[20,10]



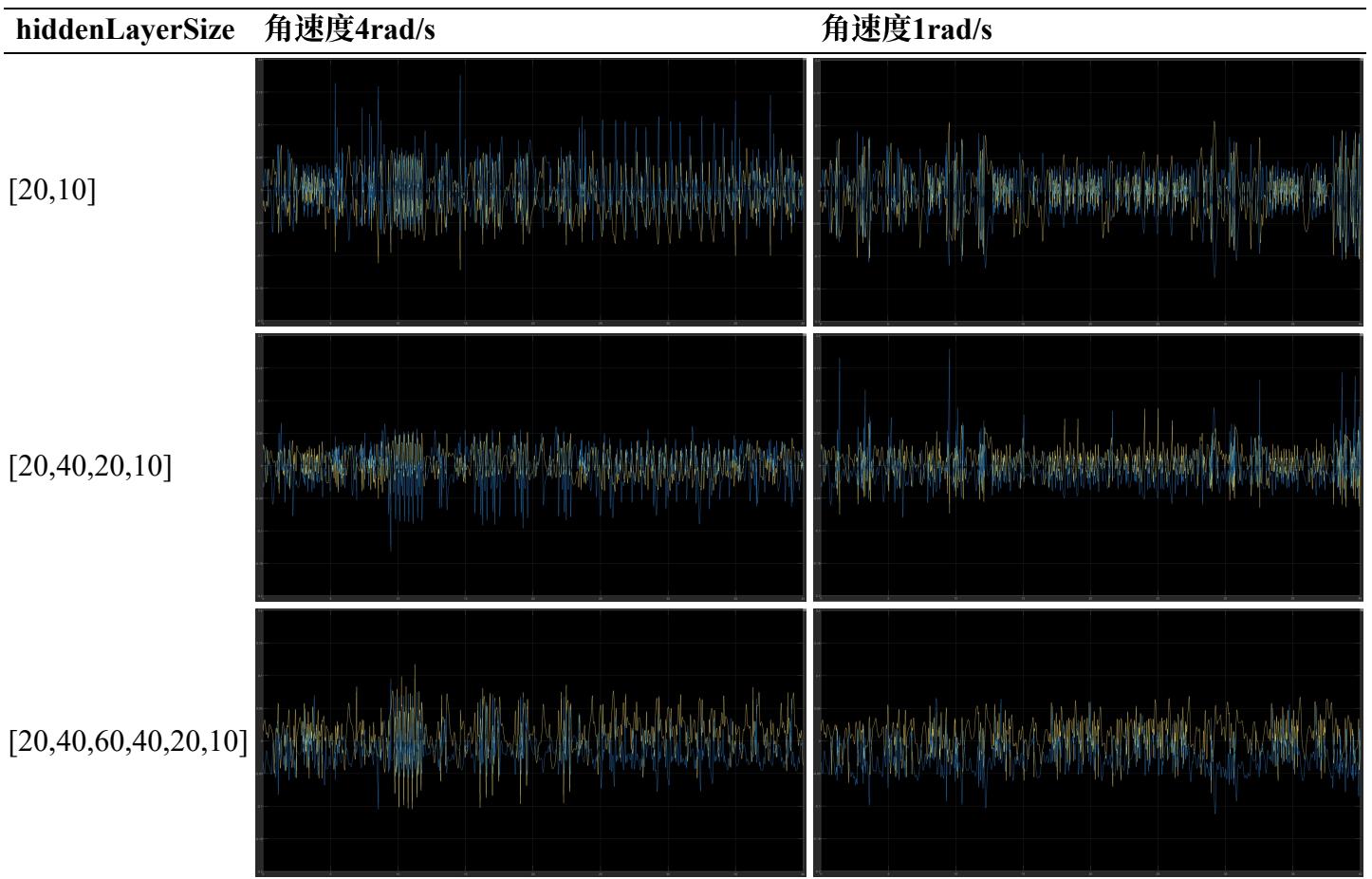
[20,40,20,10]



[20,40,60,40,20,10]



幅值0.02



从上述的实验结构表明，网络结构规模的增加对神经网络的拟合能力有一定的作用，但事实上并不明显。现总结规律如下：

- 从[20,10]到[20,40,20,10]的变化一定程度上可以减小预测误差
- 从[20,40,20,10]到[20,40,60,40,20,10]在部分情况下可以减小预测误差，但在部分情况下改善不明显或者略微变差
- 振幅增加或者频率增加会使得预测效果变差

上述现象可以从以下角度解释：

- 优化神经网络的过程即为优化 f_θ 的过程，而复杂的神经网络其 f 的空间更大，拟合能力更强
- 简单神经网络的训练容易陷入局部最优，因此盲目增加神经网络的复杂度不一定能起到更好的效果
- 简单的MLP的拟合能力有限(函数空间有限)，上述待辨识系统本身关系复杂(含有输入的乘法)，给学习带来挑战。因此MLP只能学习原映射域子空间的映射关系，泛化能力不强。

7 总结

在本实验报告中，我们对比了基于时序和差分的神经网络的预测效果，并对比了对于时序神经网络，不同的训练数据规模、网络结构对于训练效果的作用；并比较了不同种类的输入所产生的预测误差。我们发现，神经网络需要更多的训练数据，适当的网络结构复杂度，但是盲目增加网络复杂度并不利于最后的结果。同时我们发现，简单的MLP只能学习原映射关系子映射域的映射关系，并不具有较强

的泛化性。在训练的过程中，我们发现数据归一化的重要性不容小觑，min-max归一化和方差归一化都具有可取性，需要根据实际情况选择合适的归一化方法。