



(12)发明专利申请

(10)申请公布号 CN 109376413 A

(43)申请公布日 2019.02.22

(21)申请号 201811168810.2

(22)申请日 2018.10.08

(71)申请人 四川大学

地址 610065 四川省成都市武侯区一环路
南一段24号

(72)发明人 苗强 王剑宇 刘慧宇 莫贞凌
曾小飞 张恒

(74)专利代理机构 成都虹桥专利事务所(普通
合伙) 51124

代理人 陈立志

(51)Int.Cl.

G06F 17/50(2006.01)

G06N 3/08(2006.01)

权利要求书1页 说明书14页 附图9页

(54)发明名称

基于数据驱动的在线神经网络故障诊断方法

(57)摘要

本发明涉及电液作动器故障诊断领域,公开了一种基于数据驱动的在线神经网络故障诊断方法,解决传统的作动器故障诊断中无法实时更新数据库的问题。本发明的要点为:分析电液作动器结构,搭建电液作动器的仿真模型;分析作动器失效形式,对电液作动器的仿真模型植入不同类型的故障,完成信号的采集;采集得到的数据,对其开展特征值提取和降维工作;构建标签故障数据矩阵,对其特征值开展离线训练和在线训练,从而获得在线神经网络;实时采集电液作动器工作数据,利用在线神经网络对采集到的数据进行故障诊断;故障诊断之后,对采集到的数据进行在线训练,从而更新在线神经网络。本发明适用于电液作动器故障诊断。



1. 基于数据驱动的在线神经网络故障诊断方法, 其特征在于,

步骤1: 分析电液作动器结构, 搭建电液作动器的仿真模型;

步骤2: 分析作动器失效形式, 对电液作动器的仿真模型植入不同类型的故障, 完成信号的采集;

步骤3: 分析步骤2采集得到的数据, 对其开展特征值提取和降维工作, 最大化表征故障特征;

步骤4: 构建标签故障数据矩阵, 对步骤3提取得到的特征值开展离线训练和在线训练, 从而获得在线神经网络; 所述在线训练的算法如下:

步骤41: 若已将k-1组训练样本进行离线训练, 并获得了最佳权值 w_{k-1} 和偏差 b_{k-1} , 则将权值 w_{k-1} 和偏差 b_{k-1} 作为第k组训练样本进行在线训练网络的初始权值和偏差;

步骤42: 计算初始权值和偏差的条件下在线神经网络的实际输出;

步骤43: 利用在线神经网络的实际输出和期望输出计算各神经元的输出误差, 并进一步计算累计误差能量; 神经元的输出误差计算公式如下:

$$e_{kp}(n) = \sum y_{kp}(n) \ln(d_{kp}(n)) + (1 - y_{kp}(n)) \ln(1 - d_{kp}(n))$$

累计误差能量计算公式如下:

$$E(n) = \frac{1}{2} \sum_{p=1}^P e_{kp}^2(n)$$

其中, n 为迭代次数, e_{kp} 为执行第k组训练样本时第p个神经元的输出误差值, d_{kp} 为执行第k组训练样本时第p个神经元的期望输出值, y_{kp} 为执行第k组训练样本时第p个神经元的实际输出值, $E(n)$ 为累计误差能量, P 为神经元个数;

步骤44: 根据梯度下降学习规则完成权值和偏差的更新, 经过多次迭代, 最终获得新的权值 w_k 和偏差 b_k , 并将获得新的权值 w_k 和偏差 b_k 作为下一组训练样本的初始权值和偏差, 重复步骤42-步骤44完成下一组训练样本的在线训练; 其中, 梯度下降学习规则按照以下公式定义局部梯度:

$$\delta_p^p(n) = a \cdot (y_{kp} - d_{kp})$$

其中, a 为求导参数量, δ_p^p 为P个神经元中第p个神经元的局部梯度;

步骤5: 实时采集电液作动器工作数据, 利用在线神经网络对采集到的数据进行故障诊断; 故障诊断之后, 按照步骤42-步骤44对采集到的数据进行在线训练, 从而更新在线神经网络。

2. 如权利要求1所述的基于数据驱动的在线神经网络故障诊断方法, 其特征在于, 步骤1所搭建的仿真模型包括Simulink模型和AMESim模型。

3. 如权利要求1所述的基于数据驱动的在线神经网络故障诊断方法, 其特征在于, 步骤2所采集的信号包括: 飞控指令信号、作动筒传感器信号、力马达线圈电流信号和直接驱动阀传感器信号。

4. 如权利要求1所述的基于数据驱动的在线神经网络故障诊断方法, 其特征在于, 步骤3对采集到的信号开展时域信号特征值的提取, 完成数据的降维融合。

5. 如权利要求4所述的基于数据驱动的在线神经网络故障诊断方法, 其特征在于, 步骤3所提取到特征指标包括均值、极大值差值、极小值差值、延迟时间以及相关系数。

基于数据驱动的在线神经网络故障诊断方法

技术领域

[0001] 本发明涉及电液作动器故障诊断领域,特别涉及基于数据驱动的在线神经网络故障诊断方法。

背景技术

[0002] 随着航空航天产业、机器人以及民用工业的迅速发展和功能需求的不断提高,控制系统的应用范围和复杂性越来越突出,同时设备的可靠性和安全运行也变得越来越重要。由于设备长期处于运行状态以及工作环境的复杂多变性,出现故障是不可避免的。一旦在未知情况下,相应的系统或者装备出现故障,将会带来巨大的财产损失和人员伤亡。因此需要实时地对设备开展健康状态的监测和故障诊断,当故障发生时,能及时地完成故障的诊断辨识,为进一步开展设备维修和保障设备系统的正常运行具有重要意义。电液作动器是一个集机械、电子、液压为一体的复杂设备,从普通的民用客机到太空飞船上都得到了广泛的应用。目前的电液作动器大多采用了冗余性设计,一定程度上保障了可靠性,但不可避免地存在结构复杂性,这样就使得它的故障存在多样性、突发性以及成因的复杂性。关键的,作动器是否正常运行,将会对飞行器正常飞行起决定性作用。因此,对作动器其开展状态监测和故障诊断是十分必要的。

[0003] 目前对于作动器的故障诊断方法主要分为两类:一类是基于模型的故障诊断方法,如卡尔曼滤波器、多重模型诊断方法、奇偶性方法、贝叶斯网络模型、RBF神经网络观测器、强跟踪多模型自适应估计器等。这类方法主要对作动器的正常输出做出预测,与实际故障状态下的输出信号相比产生残差,同时也通过模型产生相应的自适应阈值,通过将残差和阈值进行比较来判断系统是否产生故障。该方法的难点在于建立非线性系统模型,它需要通过构建复杂的数学关系式来模拟表达该模型。另一类是基于数据的故障诊断方法,如小波变换、支持向量机、BP神经网络、Elman网络、RBF网络、GRNN网络等,该方法需要对数据进行最为恰当的处理,才能实现故障的分类。基于数据驱动的故障诊断方法被广泛应用在作动器的故障诊断领域,其核心是基于不同故障状态下的数据建立起故障诊断的离线诊断库,从而实现故障的诊断分类。离线数据库需要获取大量的数据,周期时间长;其次无法完成对库进行更新,当新数据产生时需要已经训练好的模型进行初始化重新训练,因此无论是在时间成本还是在运算效率上都是不切实际的。随着互联网的发展,流式大数据日新月异,它具有实时性、易失性、无序性、无限性等特点。而通过在线训练的方式实时获取新的数据做出处理,并进行数据库的更新,为对进一步提高故障诊断的效率和精度具有重要意义。

发明内容

[0004] 本发明要解决的技术问题是:提供一种基于数据驱动的在线神经网络故障诊断方法,解决传统的作动器故障诊断中无法实时更新数据库的问题。

[0005] 为实现上述目的,本发明采用的技术方案是:基于数据驱动的在线神经网络故障

诊断方法,包括以下步骤:

[0006] 步骤1:分析电液作动器结构,搭建电液作动器的仿真模型;

[0007] 步骤2:分析作动器失效形式,对电液作动器的仿真模型植入不同类型的故障,完成信号的采集;

[0008] 步骤3:分析步骤2采集得到的数据,对其开展特征值提取和降维工作,最大化表征故障特征;

[0009] 步骤4:构建标签故障数据矩阵,对步骤3提取得到的特征值开展离线训练和在线训练,从而获得在线神经网络;所述在线训练的算法如下:

[0010] 步骤41:若已将k-1组训练样本进行离线训练,并获得了最佳权值 w_{k-1} 和偏差 b_{k-1} ,则将权值 w_{k-1} 和偏差 b_{k-1} 作为第k组训练样本进行在线训练网络的初始权值和偏差;

[0011] 步骤42:计算初始权值和偏差的条件下在线神经网络的实际输出;

[0012] 步骤43:利用在线神经网络的实际输出和期望输出计算各神经元的输出误差,并进一步计算累计误差能量;神经元的输出误差计算公式如下:

[0013]
$$e_{kp}(n) = \sum y_{kp}(n) \ln(d_{kp}(n)) + (1 - y_{kp}(n)) \ln(1 - d_{kp}(n))$$

[0014] 累计误差能量计算公式如下:

[0015]
$$E(n) = \frac{1}{2} \sum_{p=1}^P e_{kp}^2(n)$$

[0016] 其中,n为迭代次数, e_{kp} 为执行第k组训练样本时第p个神经元的输出误差值, d_{kp} 为执行第k组训练样本时第p个神经元的期望输出值, y_{kp} 为执行第k组训练样本时第p个神经元的实际输出值, $E(n)$ 为累计误差能量,P为神经元个数;

[0017] 步骤44:根据梯度下降学习规则完成权值和偏差的更新,经过多次迭代,最终获得新的权值 w_k 和偏差 b_k ,并将获得新的权值 w_k 和偏差 b_k 作为下一组训练样本的初始权值和偏差,重复步骤42-步骤44完成下一组训练样本的在线训练;其中,梯度下降学习规则按照以下公式定义局部梯度:

[0018]
$$\delta_p^P(n) = a \cdot (y_{kp} - d_{kp})$$

[0019] 其中,a为求导参数量, δ_p^P 为P个神经元中第p个神经元的局部梯度;

[0020] 步骤5:实时采集电液作动器工作数据,利用在线神经网络对采集到的数据进行故障诊断;故障诊断之后,按照步骤42-步骤44对采集到的数据进行在线训练,从而更新在线神经网络。

[0021] 进一步的,步骤1所搭建的仿真模型可以包括Simulink模型和AMESim模型,通过Simulink和AMESim模型植入相应的故障,完成各类故障状态和正常工作状态下的数据采集。

[0022] 进一步的,步骤2所采集的信号可以包括:飞控指令信号、作动筒位移传感器信号、力马达线圈电流信号和直接驱动阀位移传感器信号。

[0023] 进一步的,步骤3可以对采集到的信号开展时域信号特征值的提取,完成数据的降维融合,以实现故障特征的最大化表征。

[0024] 进一步的,步骤3所提取到特征指标可以包括力马达电流信号、作动筒位移传感器和直接驱动阀传感器信号的均值、极大值差值、极小值差值、延迟时间以及相关系数等。

[0025] 本发明的有益效果是:本发明通过对作动器结构形式进行分析,构建出作动器的Simulink模型和AMESim模型。在对作动器展开失效分析的基础上,通过上述模型来植入相应的故障,并获取相应的输出数据。对获取的数据开展预处理工作,结合输出信号的特点运用故障特征提取方法,实现数据的降维和融合,最大化表征故障特征。接着构建故障特征标签矩阵,对神经网络开展离线和在线训练,验证了在线学习和诊断的可行性。本发明结合在线训练方法对不断获取的数据实时进行在线学习和故障诊断,并将上述算法转化成完整的Simulink模型进行封装,以便于实际工程的应用。

附图说明

- [0026] 图1是本发明的流程图;
- [0027] 图2是作动器的Simulink模型图;
- [0028] 图3是电液作动器的AMESim模型图;
- [0029] 图4a为DDV零偏超差故障+1时的直接驱动阀 (DDV) 位移传感器信号图;
- [0030] 图4b为DDV零偏超差故障+1.5时的直接驱动阀 (DDV) 位移传感器信号图;
- [0031] 图4c为DDV位移传感器精度超差故障5860时的直接驱动阀 (DDV) 位移传感器信号图;
- [0032] 图4d为DDV位移传感器精度超差故障5660时的直接驱动阀 (DDV) 位移传感器信号图;
- [0033] 图5a为DDV零偏超差故障+1时的力马达 (Mot) 线圈电流信号图;
- [0034] 图5b为DDV零偏超差故障+1.5时的力马达 (Mot) 线圈电流信号图;
- [0035] 图5c为DDV位移传感器精度超差故障5860时的力马达 (Mot) 线圈电流信号图;
- [0036] 图5d为DDV位移传感器精度超差故障5660时的力马达 (Mot) 线圈电流信号图;
- [0037] 图6a为DDV零偏超差故障+1时的作动筒 (Ram) 位移传感器信号图;
- [0038] 图6b为DDV零偏超差故障+1.5时的作动筒 (Ram) 位移传感器信号图;
- [0039] 图6c为DDV位移传感器精度超差故障5860时的作动筒 (Ram) 位移传感器信号图;
- [0040] 图6d为DDV位移传感器精度超差故障5660时的作动筒 (Ram) 位移传感器信号图;
- [0041] 图7是阶跃信号的特征指标示意图;
- [0042] 图8是神经网络结构示意图;
- [0043] 图9是sigmoid函数示意图;
- [0044] 图10是实施例封装的Simulink故障诊断模型图。

具体实施方式

[0045] 为实现在线神经网络的电液作动器故障诊断,本发明提供了一种基于数据驱动的在线神经网络故障诊断方法,包括以下步骤:

[0046] 步骤1:分析电液作动器结构,搭建电液作动器的仿真模型。

[0047] 步骤2:分析作动器失效形式,对电液作动器的仿真模型植入不同类型的故障,完成信号的采集。其中,步骤1所搭建的仿真模型可以包括Simulink模型和AMESim模型,通过Simulink和AMESim模型植入相应的故障,完成各类故障状态和正常工作状态下的数据采集;步骤2所采集的信号可以包括:飞控指令信号、作动筒传感器信号、力马达线圈电流信号

和直接驱动阀传感器信号。

[0048] 步骤3:分析步骤2采集得到的数据,对其开展特征值提取和降维工作,最大化表征故障特征。其中,步骤3可以对采集到的信号开展时域信号特征值的提取,完成数据的降维融合,以实现故障特征的最大化表征,从而所提取到特征指标可以包括:力马达电流信号、作动筒位移传感器和直接驱动阀传感器信号的均值(mean)、极大值差值(max)、极小值差值(min)、延迟时间(delay)、相关系数(corr)。

[0049] 步骤4:构建标签故障数据矩阵,对步骤3提取得到的特征值开展离线训练和在线训练,从而获得在线神经网络。

[0050] 步骤4中,离线神经网络算法如下:

[0051] 其中输入训练样本记为 $x_k = [x_{k1}, x_{k2}, \dots, x_{km}]$, k 为样本组数;输入层与隐藏层之间的权重记为 $w_{mi} = [w_{m1}, w_{m2}, \dots, w_{mi}]$,表示从输入层第 m 个神经元输出到隐层第 i 个神经元之间的权值,隐藏层与输出层之间的权重记为 $w_{ip} = [w_{i1}, w_{i2}, \dots, w_{ip}]$,表示从隐藏层第 i 个神经元输出到输出层第 p 个神经元之间的权值;偏差记为 $b_{jo} = [b_{j1}, b_{j2}, \dots, b_{jo}]$,表示从隐藏层开始的第 j 层神经元所对应的第 o 个神经元的偏差。 d_{kp} 表示期望输出值, y_{kp} 表示实际输出值,

传递函数为sigmoid函数 $\sigma(z) = \frac{1}{1+e^{-z}}$,如图9。

[0052] a) 定义输入层的输入数据为: $x_k = [x_{k1}, x_{k2}, \dots, x_{km}]$

[0053] b) 隐层的第 i 个神经元的输入为: $u_i^I = \sum_{m=1}^M w_{mi} x_{km} + b_{li}$

[0054] c) 隐层的第 i 个神经元的输出为: $v_i^I = \sigma(\sum_{m=1}^M w_{mi} x_{km} + b_{li})$

[0055] d) 输出层第 p 个神经元输入为: $u_p^P = \sum_{i=1}^I w_{ip} v_i^I + b_{2p}$

[0056] e) 输出层第 p 个神经元输出,即网络输出为: $y_{kp} = \sigma(\sum_{i=1}^I w_{ip} v_i^I + b_{2p})$

[0057] f) 输出层第 p 个神经元的输出误差cost函数为: $e_{kp}(n) = d_{kp}(n) - y_{kp}(n)$

[0058] g) 整个训练集上,输出层所有神经元的累计误差能量为: $E(n) = \frac{1}{2} \sum_{k=1}^K \sum_{p=1}^P e_{kp}^2(n)$

[0059] h) 采用梯度算法,定义 P 个神经元中第 p 个神经元的局部梯度,其中 a 为求导参数量,

[0060] $\delta_p^P(n) = -\frac{\partial E(n)}{\partial u_p^P(n)} = a \cdot \sigma'(z)$

[0061] i) 根据梯度下降学习规则, w_{ip} 的修正量为 $\Delta w_{ip}(n) = -\eta \frac{\partial E(n)}{\partial w_{ip}(n)} = \eta \cdot a \cdot \sigma'(z)$,权值的

调整为: $w_{ip}(n+1) = w_{ip}(n) + \Delta w_{ip}(n)$

[0062] j) 同样的对隐层与隐层之间,输入层与隐层之间的权值进行调整,对偏差也按照上述过程进行重复调整,这里不再赘述。当误差性能函数小于预先设定的某个值时,就可以完成神经网络结构的更新和构建。

[0063] 离线BP算法经常会出现收敛速度变慢的情况,其原因是学习率选择不当造成的,学习率选的太小,收敛太慢;学习率选得太大,则有可能导致矫枉过正造成发散。因此自适应的对在线学习的学习速率进行调整也是十分重要的,往往期望在初始训练阶段学习的快些,接近目标值时学习的慢些,这样才能达到良好的学习效果。因此对步骤f的输出误差 $e_{kp}(n) = d_{kp}(n) - y_{kp}(n)$ 进行相应的修正。由于传递函数为sigmoid函数,步骤h中定义的局部梯度 $\delta_p^p(n)$ 是关于传递函数sigmoid函数的导数,而权值更新的快慢程度是与该导数相关的。当sigmoid函数(如图9)的变量较大或较小时其接近于0,此时梯度变小,就可能会带来更新速率下降的问题。而采用cross-entropy函数则不会产生上述问题,相应的cross-entropy函数代替了上述步骤f中的cost函数。

[0064] $e_{kp}(n) = y_{kp}(n) \ln(d_{kp}(n)) + (1 - y_{kp}(n)) \ln(1 - d_{kp}(n))$

[0065] 在引入cross-entropy函数之后进行训练时,实际输出值与期望输出值差距较大,因此 δ_p^p 比较大,权值更新快;当实际输出值与目标值不断接近时, δ_p^p 也逐渐减小、权值更新不断变慢。这样在线训练可以达到较快的训练速度和较高的训练精度,并且可以实现期望训练速度在开始训练的时候更新快一些,后期训练的时候更新慢一些的功能,具有自适应的特点。步骤h的变化形式如下:

[0066] $\delta_p^p(n) = a \cdot (y_{kp} - d_{kp})$

[0067] 由于在线学习的样本是流数据形式的,往往样本是依次递增的。因此在线学习算法与离线学习算法的差别在于误差能量的定义形式。对于离线训练来说是计算批数据的误差能量,而对于在线训练的误差能量则需要定义为单个样本或小批样本的训练误差。因此基于在线训练算法仅需要第k组的数据和标签开展对k-1组数据下训练得到的神经网络开展训练,当训练满足相应的性能指标后便得到了k组数据下的最优权值 w^k 和偏差 b^k ,避免对神经网络开展重复训练,提高了计算的效率。其次新一组数据所对应的神经网络的权值和阈值都是基于前一组数据下训练的神经网络的权值阈值所进行训练的,因此使得权值和阈值的计算实现了递推的功能,这样就可以充分利用前一组训练的信息,提高算法的自适应能力。

[0068] 综上,步骤4采用了基于自适应梯度下降的神经网络在线训练,其算法如下:

[0069] S1:假设已将k-1组训练样本进行离线训练,并获得了最佳权值 w_{k-1} 和偏差 b_{k-1} ,开展在线训练学习时,采集得到新的一组样本 x_k ,则可将权值 w_{k-1} 和偏差 b_{k-1} 作为第k组训练样本 x_k 进行在线训练网络的初始权值和偏差。

[0070] S2:计算初始权值 w_{k-1} 和偏差 b_{k-1} 的条件下在线神经网络的实际输出。这里可采用离线训练算法中步骤a-e来计算初始权值 w_{k-1} 和偏差 b_{k-1} 的条件下在线神经网络的实际输出。

[0071] S3:利用在线神经网络的实际输出和期望输出计算各神经元的输出误差,并进一步计算累计误差能量。这里通过定义新的步骤f,定义cross-entropy函数代替cost函数如下:

[0072] $e_{kp}(n) = \sum y_{kp}(n) \ln(d_{kp}(n)) + (1 - y_{kp}(n)) \ln(1 - d_{kp}(n))$

[0073] 同时定义新的步骤g,相比离线学习算法,在线学习不要记忆全部的训练样本,只需要对当前进行训练的样本开展误差性能的评估,因此它的累计误差能量为累计误差能量

计算公式如下：

$$[0074] \quad E(n) = \frac{1}{2} \sum_{p=1}^P e_{kp}^2(n)$$

[0075] 上述, n 为迭代次数, e_{kp} 为执行第 k 组训练样本时第 p 个神经元的输出误差值, d_{kp} 为执行第 k 组训练样本时第 p 个神经元的期望输出值, y_{kp} 为执行第 k 组训练样本时第 p 个神经元的实际输出值, $E(n)$ 为累计误差能量, P 为神经元个数。

[0076] S4: 根据梯度下降学习规则完成权值和偏差的更新, 可以通过步骤 $h-j$ 多次迭代, 最终获得新的权值 w_k 和偏差 b_k , 并将获得新的权值 w_k 和偏差 b_k 作为下一组训练样本的初始权值和偏差, 重复步骤 S2-步骤 S4 完成下一组训练样本的在线训练; 其中, 梯度下降学习规则按照以下公式定义局部梯度:

$$[0077] \quad \delta_p^p(n) = a \cdot (y_{kp} - d_{kp})$$

[0078] 其中, a 为求导参数量, δ_p^p 为 P 个神经元中第 p 个神经元的局部梯度

[0079] 步骤 5: 实时采集电液作动器工作数据, 利用在线神经网络对采集到的数据进行故障诊断; 故障诊断之后, 按照步骤 42-步骤 44 对采集到的数据进行在线训练, 从而更新在线神经网络。

[0080] 实施例

[0081] 以下通过实施例对本发明再做进一步说明。

[0082] 实施例提供了一种基于数据驱动的在线神经网络故障诊断方法, 其具体步骤如下:

[0083] 步骤 1: 对作动器结构和工作形式进行分析, 搭建作动器的 Simulink 模型和 AMEsim 模型, 电液作动器主要包括了信号处理系统, 液压系统和机械系统组成, 集合了它们共同的优势。电液伺服系统可以灵活地满足高的控制精度, 快速响应和信号处理等功能。信号处理部分主要是由 PID 控制系统完成操纵指令的处理; 液压部分通常是由伺服阀, 电磁阀, 模式转换阀等组成; 机械部分主要是由作动筒等各类执行部件组成。目前大多数的电液作动器采用了冗余控制系统, 如电气四余度和液压两余度的形式, 极大地保证了作动器工作的可靠性。本例在传统电液作动器的基础上, 采用了直接驱动阀式电液作动器 (DDV)。与传统的其他类型的电液余度作动器相比, 其独有的特点在于直接由力马达驱动主控阀芯, 可以大大降低静态泄露。当作动器正常工作时, 控制信号经 PID 控制系统实现信号的非线性增益变化, 传递给 DDV (直接驱动阀) 的力马达并驱动 DDV 阀芯运动, 并通过流量的变化推动主控阀阀芯运动, 分配的负载流量经过模态选择回中阀 (MSCV), 分别作用于双腔串联式作动筒的两腔, 推动作动筒运动。在 DDV 及作动筒上分别装有四余度的 LVDT, 可以完成内外回路的闭环控制。电气部分主要结构如图 2 所示, 液压部分主要结构如图 3 所示。

[0084] 步骤 2: 目前在基于理论模型的基础上, 结合工程实际, 对作动器开展了相应的故障诊断研究。电液作动器的故障失效形式通常可分为机械部件, 电气部件, 液压部件故障, 具体的来说主要有力马达失效, 各类阀的失效, 传感器失效, 泵的失效以及电子增益失效等。本专利重点研究对故障的诊断方法和应用, 最终选取了 6 类故障失效形式展开分析, 通过在 Simulink 和 AMEsim 模型中改变作动器各结构部分的参数, 分别模拟作动器的不同故障类型, 植入相应的故障类型, 如表 1 和图 3 所示。针对副翼作动器关键零部件的典型故障, 结

合副翼作动器理论模型和试验装置,梳理副翼作动器所涉及运行监测参数的数据类型,选择表征参数与故障相关性最大的监测数据,采集得到的信号包括4部分如表2所示。获取的同一故障不同程度的输出信号和正常输出信号如图4a-4d、5a-5d及6a-6d所示,以DDV零偏超差和DDV位移传感器位置精度超差两种故障为例,每类故障对应3类采集信号和相应的输入信号。

[0085] 表1 6种故障模式

[0086]

故障模式	数目	正常值	故障值
DDV 零偏超差	1	0	0.5-3
DDV 位移传感器位置精度超差	1	6660	5860-5060
作动筒位移传感器位置精度超差	1	0	0.5-3
作动筒零偏超差	1	1000	940-800
液压 1/2 压力下降 (左右液压系统)	2	28	26-15

[0087] 表2 4类采集信号

采集信号
飞控指令信号
作动筒传感器信号 (Ram)
力马达线圈电流信号 (Mot)
直接驱动阀传感器信号 (DDV)

[0089] 步骤3:对副翼作动器采集的四类数据信号开展特征值的提取。飞控系统的指令信号是随机信号,相应的输出信号也存在不确定性和不稳定性。本专利主要为了验证故障诊断方法的可行性,因此仅针对方波输入信号下的故障表征形式展开研究。大量的数据直接作为故障诊断的输入往往是不现实的,因为获取的数据量是十分庞大的,在计算效率和判别精度上会大打折扣。因此对数据进行降维显得十分重要,常见的降维方法主要包括了线性降维和非线性降维方法,如主成分分析 (PCA),线性判别分析 (LDA) 以及局部线性嵌入等。其次针对副翼作动器的输入输出信号,开展相应的特征提取也是一种十分重要的降维和故障表征方法。在采集得到的各类故障信号基础上,结合图4a-4d、5a-5d及6a-6d所示,对作动器的故障特征提取以时域信号特征提取为主。通常时域分析常用的特征参数如表3:峰峰值,均值,方差,高阶矩,波形指标,裕度指标,峭度指标,相关系数,除此之外,还包括了阶跃信号的超调量,稳态误差,峰值时间,63%的上升时间,86%的上升时间等,如图7所示。故障特征量的提取应满足以下3个条件:(1)对外界干扰不敏感,去除量纲化;(2)各故障类型的区别性和类内小的差距性;(3)特征量对故障敏感程度大,特征量尽可能多。针对步骤2采集得到的故障信号,以DDV的两类故障为主要研究对象,对其开展特征值的提取。最终在图4a-4d、5a-5d及6a-6d的基础上分别选择了直接驱动阀传感器信号、力马达和作动筒位移传感器信号的均值 (mean)、极大值差值 (max)、极小值差值 (min)、延迟时间 (delay)、相关系数

(corr) 作为相应故障的表征。提取的部分特征数据如表4-5所示。

[0090] 表3信号特征指标

[0091]

特征指标	计算公式	特征指标	计算公式
均值	$u_x = \frac{1}{n} \sum_{i=1}^n x_i$	均方根值	$\psi_x = \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}$
峭度因子	$K_v = \frac{\frac{1}{N} \sum_{i=1}^N (x_i - \mu_x)^4}{[\frac{1}{N} \sum_{i=1}^N (x_i - \mu_x)^2]^2}$	峰值因子	$C_f = \frac{X_p}{\psi_x}$
方差	$\sigma_x^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \mu_x)^2$	裕度因子	$CL_f = \frac{\max(x_i)}{\left \frac{1}{N} \sum_{i=1}^N \sqrt{ x_i } \right ^2}$
脉冲因子	$I_f = \frac{\max(x_i)}{\frac{1}{n} \sum_{i=1}^n x_i }$	波形因子	$S_f = \frac{\psi_x}{\frac{1}{n} \sum_{i=1}^n x_i }$
峰值	$X_p = \frac{1}{n} \sum_{i=1}^n X_{pj}$	相关系数	$\rho_{XY} = \frac{Cov(X, Y)}{\sqrt{D(X)}\sqrt{D(Y)}}$

[0092] 表4DDV零偏超差特征值

[0093]

Mot					DDV					Ram				
dela y	corr	mean	max	min	dela y	corr	mean	max	min	dela y	corr	mean	max	min
0.00 00	0.99 97	0.06 08	0.00 17	0.02 55	0.00 00	0.99 99	0.26 29	0.20 32	0.00 00	0.00 00	1.00 00	0.01 99	0.01 46	0.01 50
0.00 00	1.00 00	0.01 96	0.00 20	0.00 06	0.00 00	0.99 95	0.53 10	0.20 32	0.00 00	0.00 00	1.00 00	0.03 98	0.02 93	0.03 00
0.00 00	0.99 96	0.06 43	0.00 17	0.02 55	0.00 00	0.99 90	0.79 57	0.20 32	0.00 00	0.00 00	1.00 00	0.05 94	0.04 38	0.04 50
0.00 00	0.99 99	0.03 28	0.01 27	0.00 12	0.00 00	0.99 82	1.06 07	0.20 32	0.00 00	0.00 00	1.00 00	0.07 90	0.05 86	0.06 01
0.00 00	0.99 97	0.05 19	0.00 25	0.01 63	0.00 00	0.99 72	1.31 96	0.20 32	0.00 00	0.00 00	1.00 00	0.09 80	0.07 31	0.07 50
0.00 00	0.99 98	0.03 47	0.01 82	0.00 26	0.00 00	0.99 59	1.57 75	0.20 32	0.00 00	0.00 00	1.00 00	0.11 69	0.08 78	0.09 01
0.00 00	0.99 99	0.03 76	0.00 30	0.00 26	0.00 00	0.99 45	1.82 91	0.20 32	0.00 00	0.00 00	1.00 00	0.13 53	0.10 25	0.10 51

[0094]

0.00 00	0.99 98	0.05 09	0.00 72	0.01 63	0.00 00	0.99 29	2.07 72	0.20 32	0.00 00	0.00 00	1.00 00	0.15 32	0.11 71	0.12 01
0.00 00	1.00 00	0.03 95	0.00 33	0.00 26	0.00 00	0.99 10	2.31 88	0.20 32	0.00 00	0.00 00	1.00 00	0.17 05	0.13 19	0.13 51
0.00 00	0.99 89	0.18 35	0.00 45	0.03 05	0.00 00	0.98 90	2.55 14	0.20 32	0.00 00	0.00 00	1.00 00	0.18 71	0.14 63	0.15 02
0.00 00	0.99 81	0.28 98	0.00 41	0.00 30	0.00 00	0.98 68	2.77 79	0.20 32	0.00 00	0.00 00	0.99 99	0.20 31	0.16 11	0.16 52
0.00 00	0.99 73	0.31 66	0.00 52	0.03 05	0.00 00	0.98 45	2.99 48	0.20 32	0.00 00	0.00 00	0.99 99	0.21 83	0.17 55	0.18 02
0.00 00	0.95 53	0.54 75	0.54 75	0.41 79	0.00 00	0.98 23	3.20 75	0.20 32	0.00 00	0.00 00	0.99 99	0.23 34	0.19 04	0.19 52
0.00 00	0.77 90	1.23 47	0.53 70	1.20 99	0.00 00	0.98 06	3.41 34	0.20 32	0.00 00	0.00 00	0.99 99	0.24 82	0.20 48	0.21 03
0.00 00	0.61 13	2.02 67	0.52 61	2.00 19	0.00 00	0.97 88	3.62 03	0.20 32	0.00 00	0.00 00	0.99 99	0.26 30	0.21 96	0.22 53
0.00 00	0.49 19	2.81 87	0.51 63	2.79 39	0.00 00	0.97 68	3.82 25	0.20 32	0.00 00	0.00 00	0.99 99	0.27 78	0.23 41	0.24 03
0.00 00	0.44 85	3.19 96	0.50 62	0.01 08	0.00 00	0.97 48	4.02 70	0.20 32	0.00 00	0.00 00	0.99 99	0.29 27	0.24 87	0.25 54
0.00 00	0.44 90	3.19 96	0.49 40	0.00 68	0.00 00	0.97 21	4.23 21	0.20 32	0.00 00	0.00 00	0.99 99	0.30 75	0.25 75	0.27 28
0.00 00	0.44 86	3.19 96	0.48 52	0.03 05	0.00 00	0.97 02	4.43 08	0.20 32	0.00 00	0.00 00	0.99 98	0.32 23	0.27 77	0.28 55
0.00 00	0.44 85	3.19 96	0.47 46	0.00 51	0.00 00	0.96 78	4.63 35	0.20 32	0.00 00	0.00 00	0.99 98	0.33 71	0.29 29	0.30 05

[0095] 表5DDV位移传感器位置精度超差特征值

[0096]

Mot					DDV					Ram				
delay	corr	mean	max	min	delay	corr	mean	max	min	delay	corr	mean	max	min
0.00 00	0.99 98	0.03 08	0.01 27	0.01 24	0.00 00	1.00 00	0.09 00	0.00 57	0.00 47	0.00 00	1.00 00	0.00 70	0.00 48	0.00 57
0.00 00	0.99 95	0.06 97	0.01 49	0.02 55	0.00 00	1.00 00	0.10 75	0.00 57	0.00 47	0.00 00	1.00 00	0.00 83	0.00 59	0.00 69
0.00 00	0.99 95	0.06 71	0.01 66	0.01 71	0.00 00	1.00 00	0.11 91	0.00 57	0.00 47	0.00 00	1.00 00	0.00 90	0.00 65	0.00 75
0.00 00	0.99 88	0.13 88	0.01 75	0.01 84	0.00 00	0.99 99	0.12 90	0.00 57	0.00 47	0.00 00	1.00 00	0.00 98	0.00 71	0.00 80
0.00 00	0.99 80	0.16 02	0.02 75	0.02 15	0.00 00	0.99 99	0.14 41	0.00 57	0.00 47	0.00 00	1.00 00	0.01 04	0.00 77	0.00 85
0.00	0.99	0.22	0.02	0.02	0.00	0.99	0.21	0.00	0.00	0.00	1.00	0.01	0.01	0.00

[0097]

00	68	23	09	55	00	99	95	57	47	00	00	50	32	90
0.00 00	0.99 95	0.06 58	0.01 82	0.04 08	0.00 00	1.00 00	0.06 20	0.00 57	0.00 47	0.00 00	1.00 00	0.00 48	0.00 25	0.00 38
0.00 00	0.98 15	0.36 00	0.02 43	0.36 00	0.00 00	0.99 99	0.39 88	0.00 57	0.00 47	0.00 00	1.00 00	0.02 87	0.02 78	0.00 86
0.00 00	0.97 74	0.36 07	0.02 56	0.36 07	0.00 00	0.99 99	0.40 41	0.00 57	0.00 47	0.00 00	1.00 00	0.02 90	0.02 81	0.00 79
0.00 00	0.97 45	0.36 36	0.02 84	0.36 36	0.00 00	0.99 99	0.40 92	0.00 57	0.00 47	0.00 00	1.00 00	0.02 91	0.02 82	0.00 67
0.00 00	0.97 45	0.36 07	0.02 96	0.36 07	0.00 00	0.99 99	0.41 31	0.00 57	0.00 47	0.00 00	1.00 00	0.02 92	0.02 84	0.00 35
0.00 00	0.97 12	0.35 72	0.03 05	0.35 72	0.00 00	0.99 99	0.41 59	0.00 57	0.00 47	0.00 00	1.00 00	0.02 93	0.02 85	0.00 49
0.00 00	0.99 95	0.06 13	0.03 19	0.03 05	0.00 00	1.00 00	0.09 81	0.00 57	0.00 47	0.00 00	1.00 00	0.00 76	0.00 54	0.00 63
0.00 00	0.99 23	0.33 79	0.02 21	0.02 30	0.00 00	0.99 99	0.35 90	0.00 57	0.00 47	0.00 00	1.00 00	0.02 59	0.02 40	0.00 93
0.00 00	0.98 62	0.35 74	0.02 34	0.35 74	0.00 00	0.99 99	0.39 11	0.00 57	0.00 47	0.00 00	1.00 00	0.02 82	0.02 71	0.00 92
1.00 00	0.99 97	0.06 18	0.02 17	0.02 63	1.00 00	1.00 00	0.01 56	0.00 67	0.00 43	1.98 00	1.00 00	0.00 07	0.00 04	0.00 06
1.00 00	0.99 97	0.06 45	0.00 63	0.03 11	1.00 00	1.00 00	0.02 12	0.00 67	0.00 43	1.98 00	1.00 00	0.00 17	0.00 12	0.00 13
1.00 00	0.99 98	0.04 86	0.00 43	0.00 78	1.00 00	1.00 00	0.03 15	0.00 67	0.00 43	1.98 00	1.00 00	0.00 24	0.00 13	0.00 16
1.00 00	0.64 95	3.18 59	0.51 67	0.04 40	1.06 00	0.88 83	9.73 85	0.00 67	0.00 43	1.98 00	0.96 28	2.50 52	1.97 16	0.01 92
1.00 00	0.64 97	3.19 16	0.51 87	0.06 92	1.04 00	0.89 28	9.73 85	0.00 67	0.00 43	1.98 00	0.96 61	2.77 38	2.64 98	0.01 72

[0098] 步骤4:针对副翼作动器开展的前期数据挖掘与知识获取的基础上,开展基于统计与人工智能的故障诊断研究。其中在人工智能的故障诊断研究中重点以神经网络开展故障诊断,基于神经网络的故障诊断方法被广泛应用于模式识别领域。其中应用最多是BP网络结构如图8所示,它是由输入层,输出层和隐藏层构成的,并通过相应的权值将它们连接起来。对其进行有监督的学习是通过将带标签的数据实际输出值与标签值进行对比,从而反向调整权重 w 和偏差 b ,最终将知识存储在相应的权重和偏差值当中,从而实现模式的识别,神经网络权重和偏差更新算法如下,本专利构造了一个3层的神经网络。传统的神经网络训练方式为离线学习方式,由于存在流数据,因此无法实时更新神经网络库,即使需要更新时,需要将之前训练的规则进行初始化重新学习。如果采用在线学习的方法对神经网络进行更新就可以实现相应故障的诊断,基于神经网络的在线学习主要实现对权值和偏差的更新,从而完成相应的神经网络诊断库的更新。

[0099] 离线神经网络算法如下:

[0100] 其中输入训练样本记为 $x_k = [x_{k1}, x_{k2}, \dots, x_{km}]$, k 为样本组数; 输入层与隐藏层之间的权重记为 $w_{mi} = [w_{m1}, w_{m2}, \dots, w_{mi}]$, 表示从输入层第 m 个神经元输出到隐层第 i 个神经元之间的权值, 隐藏层与输出层之间的权重记为 $w_{ip} = [w_{i1}, w_{i2}, \dots, w_{ip}]$, 表示从隐藏层第 i 个神经元输出到输出层第 p 个神经元之间的权值; 偏差记为 $b_{jo} = [b_{j1}, b_{j2}, \dots, b_{jo}]$, 表示从隐藏层开始的第 j 层神经元所对应的第 o 个神经元的偏差。 d_{kp} 表示期望输出值, y_{kp} 表示实际输出值,

传递函数为sigmoid函数 $\sigma(z) = \frac{1}{1+e^{-z}}$, 如图9。

[0101] a) 定义输入层的输入数据为: $x_k = [x_{k1}, x_{k2}, \dots, x_{km}]$

[0102] b) 隐层的第 i 个神经元的输入为: $u_i^I = \sum_{m=1}^M w_{mi} x_{km} + b_{li}$

[0103] c) 隐层的第 i 个神经元的输出为: $v_i^I = \sigma(\sum_{m=1}^M w_{mi} x_{km} + b_{li})$

[0104] d) 输出层第 p 个神经元输入为: $u_p^P = \sum_{i=1}^I w_{ip} v_i^I + b_{2p}$

[0105] e) 输出层第 p 个神经元输出, 即网络输出为: $y_{kp} = \sigma(\sum_{i=1}^I w_{ip} v_i^I + b_{2p})$

[0106] f) 输出层第 p 个神经元的输出误差cost函数为: $e_{kp}(n) = d_{kp}(n) - y_{kp}(n)$

[0107] g) 整个训练集上, 输出层所有神经元的累计误差能量为: $E(n) = \frac{1}{2} \sum_{k=1}^K \sum_{p=1}^P e_{kp}^2(n)$

[0108] h) 采用梯度算法, 定义局部梯度, 其中 a 为求导参数量, $\delta_p^P(n) = -\frac{\partial E(n)}{\partial u_p^P(n)} = a \cdot \sigma'(z)$

[0109] i) 根据梯度下降学习规则, w_{ip} 的修正量为 $\Delta \omega_{ip}(n) = -\eta \frac{\partial E(n)}{\partial \omega_{ip}(n)} = \eta \cdot a \cdot \sigma'(z)$, 权值

的调整

[0110] 为: $\omega_{ip}(n+1) = \omega_{ip}(n) + \Delta \omega_{ip}(n)$

[0111] j) 同样的对隐层与隐层之间, 输入层与隐层之间的权值进行调整, 对偏差也按照上述过

[0112] 程进行重复调整, 这里不再赘述。当误差性能函数小于预先设定的某个值时, 就可以完

[0113] 成神经网络结构的更新和构建。

[0114] 离线BP算法经常会出现收敛速度变慢的情况, 其原因是学习率选择不当造成的, 学习率选的太小, 收敛太慢; 学习率选得太大, 则有可能导致矫枉过正造成发散。因此自适应的对在线学习的学习速率进行调整也是十分重要的, 往往期望在初始训练阶段学习的快些, 接近目标值时学习的慢些, 这样才能达到良好的学习效果。因此对步骤e的输出误差 $e_{kp}(n) = d_{kp}(n) - y_{kp}(n)$ 进行相应的修正。由于传递函数为sigmoid函数, 步骤g中定义的局部梯度 $\delta_p^P(n)$ 是关于传递函数sigmoid函数的导数, 而权值更新的快慢程度是与该导数相关的。当sigmoid函数(如图9)的变量较大或较小时其接近于0, 此时梯度变小, 就可能会带来更新速率下降的问题。而采用cross-entropy函数则不会产生上述问题, 相应的cross-entropy

函数代替了上述步骤f中的cost函数。

$$[0115] \quad e_{kp}(n) = y_{kp}(n) \ln(d_{kp}(n)) + (1 - y_{kp}(n)) \ln(1 - d_{kp}(n))$$

[0116] 在引入cross-entropy函数之后进行训练时,实际输出值与期望输出值差距较大,因此 δ_p^P 比较大,权值更新快;当实际输出值与目标值不断接近时, δ_p^P 也逐渐减小、权值更新不断变慢。这样在线训练可以达到较快的训练速度和较高的训练精度,并且可以实现期望训练速度在开始训练的时候更新快一些,后期训练的时候更新慢一些的功能,具有自适应的特点。步骤h的变化形式如下:

$$[0117] \quad \delta_p^P(n) = a \cdot (y_{kp} - d_{kp})$$

[0118] 由于在线学习的样本是流数据形式的,往往样本是依次递增的。因此在线学习算法与离线学习算法的差别在于误差能量的定义形式。对于离线训练来说是计算批数据的误差能量,而对于在线训练的误差能量则需要定义为单个样本或小批样本的训练误差。因此基于在线训练算法仅需要第k组的数据和标签开展对k-1组数据下训练得到的神经网络开展训练,当训练满足相应的性能指标后便得到了k组数据下的最优权值 w^k 和偏差 b^k ,避免对神经网络开展重复训练,提高了计算的效率。其次新一组数据所对应的神经网络的权值和阈值都是基于前一组数据下训练的神经网络的权值阈值所进行训练的,因此使得权值和阈值的计算实现了递推的功能,这样就可以充分利用前一组训练的信息,提高算法的自适应能力。

[0119] 综上,步骤4采用了基于自适应梯度下降的神经网络在线训练,其算法如下:

[0120] S1:若已将k-1组训练样本进行离线训练,并获得了最佳权值 w_{k-1} 和偏差 b_{k-1} ,开展在线训练学习时,采集得到新的一组样本 x_k ,则可将权值 w_{k-1} 和偏差 b_{k-1} 作为第k组训练样本 x_k 进行在线训练网络的初始权值和偏差;

[0121] S2:计算初始权值 w_{k-1} 和偏差 b_{k-1} 的条件下在线神经网络的实际输出。这里可采用离线训练算法中步骤a-e来计算初始权值 w_{k-1} 和偏差 b_{k-1} 的条件下在线神经网络的实际输出。

[0122] S3:利用在线神经网络的实际输出和期望输出计算各神经元的输出误差,并进一步计算累计误差能量。这里通过定义新的步骤f,定义cross-entropy函数代替cost函数如下:

$$[0123] \quad e_{kp}(n) = \sum y_{kp}(n) \ln(d_{kp}(n)) + (1 - y_{kp}(n)) \ln(1 - d_{kp}(n))$$

[0124] 同时定义新的步骤g,相比离线学习算法,在线学习不要记忆全部的训练样本,只需要对当前进行训练的样本开展误差性能的评估,因此它的累计误差能量为累计误差能量计算公式如下:

$$[0125] \quad E(n) = \frac{1}{2} \sum_{p=1}^P e_{kp}^2(n)$$

[0126] 上述,n为迭代次数, e_{kp} 为执行第k组训练样本时第p个神经元的输出误差值, d_{kp} 为执行第k组训练样本时第p个神经元的期望输出值, y_{kp} 为执行第k组训练样本时第p个神经元的实际输出值,E(n)为累计误差能量,P为神经元个数。

[0127] S4:根据梯度下降学习规则完成权值和偏差的更新,可以通过步骤h-j多次迭代,最终获得新的权值 w_k 和偏差 b_k ,并将获得新的权值 w_k 和偏差 b_k 作为下一组训练样本的初始

权值和偏差,重复步骤S2-步骤S4完成下一组训练样本的在线训练;其中,梯度下降学习规则按照以下公式定义局部梯度:

$$[0128] \quad \delta_p^p(n) = a \cdot (y_{kp} - d_{kp})$$

[0129] 其中,a为求导参数量, δ_p^p 为P个神经元中第p个神经元的局部梯度。

[0130] 本专利为检验在线学习的可行性,针对直接驱动阀DDV的两类故障进行故障诊断。每类故障共获取了100组数据,对于离线训练方式而言,我们选取了75%的数据作为训练数据,剩余的25%的数据作为试验数据。而对于在线训练方式而言,首先以25%的数据作为离线训练,然后分别以25%的数据开展增量训练,最终按照75%的数据作为在线学习的训练数据,其余的25%作为试验数据。由于神经网络的初始输入权值和偏差具有随机性,因此我们对神经网络进行了100次的循环训练,从而获得平均训练时间以及训练精度作为对比。最终从表6可以看出,两类故障的识别率都达到了100%,但是在线学习方式所消耗的时间要远小于离线学习方式下所消耗的时间,可以保存之前训练好的规则并完成实时数据库的更新,从而证明了增量学习算法的可行性。

[0131] 表6训练方式

[0132]

训练方式	准确率	平均训练时间
离线训练	100%	7.4s
在线训练	100%	2.9985s

[0133] 步骤5:基于上述在线学习算法验证成功的基础上,可将Matlab中编写的数据预处理程序和在线神经网络网络编译成Simulink模型,以便于工程实际的应用。由于输入信号是实时的,故障特征提取方式是基于正常信号基础之上的,因此需要将图1的正常模型植入到用于故障诊断的Simulink模型中,以便实时产生正常数据,将实际获得的故障数据与正常数据进行融合比对,并进行特征的提取降维。将获取的标签特征矩阵传递到神经网络中进行识别和在线训练,从而实现故障的在线识别。

[0134] 步骤6:基于上述各类模型与算法的基础上,将它们转化为Simulink模型进行封装,以便于实际工程的应用,具有重要意义。如图10所示,输入信号为控制指令,通过作动器的故障模型可以获得3种采集信号,同时将正常工作模型植入在Simulink模型中,采集得到作动器的3种正常输出信号;接着运用信号采集模块,将上述信号按照一定的顺序完成组合;进一步地通过3个信号特征提取模块,分别获取力马达线圈、DDV位移传感器、作动器位移传感器的五种特征值(均值(mean)、极大值差值(max)、极小值差值(min)、延迟时间(delay)、相关系数(corr));进一步将获取的特征值通过神经网络在线训练及故障诊断模块完成训练和识别;最后通过故障识别模块,实现故障类型的可视化。

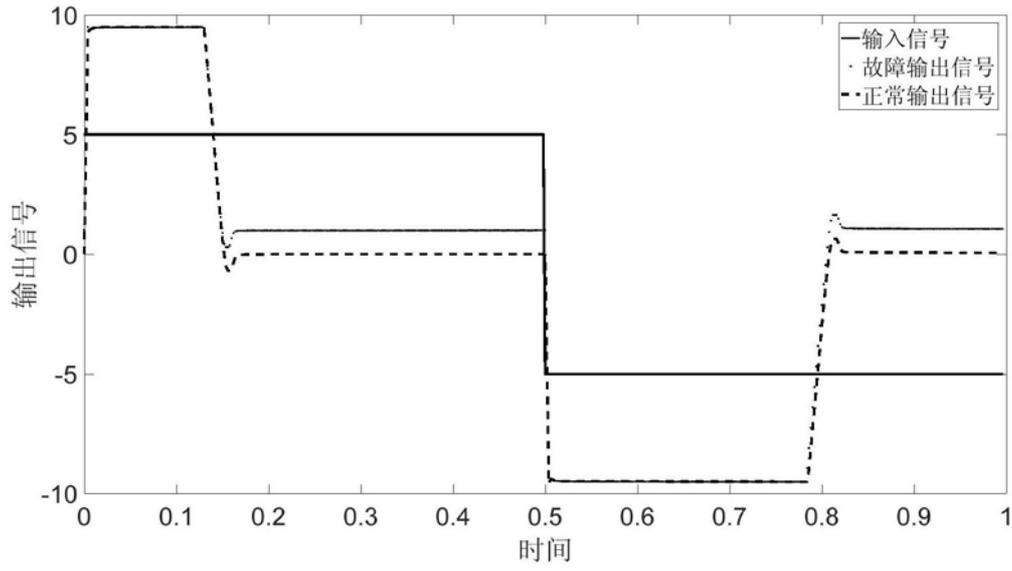


图4a

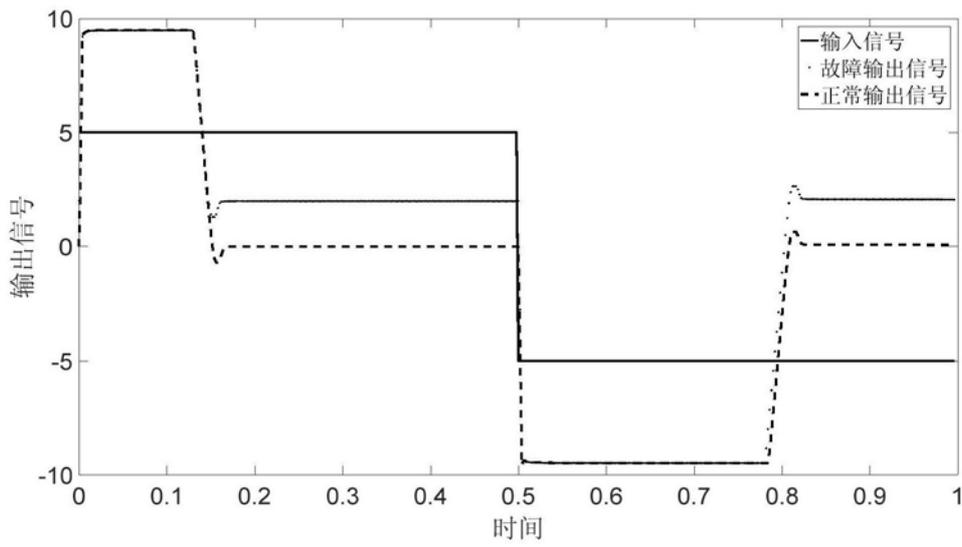


图4b

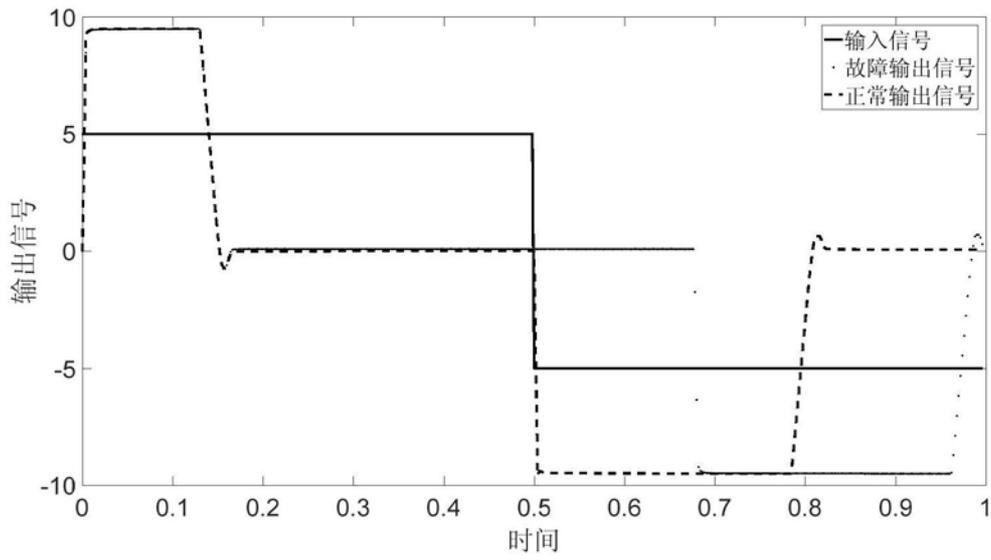


图4c

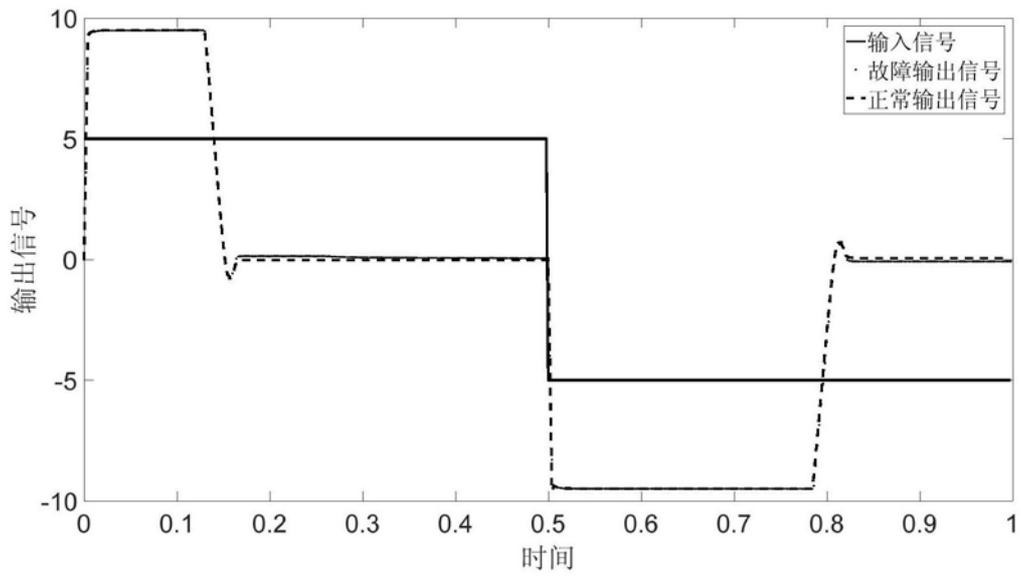


图4d

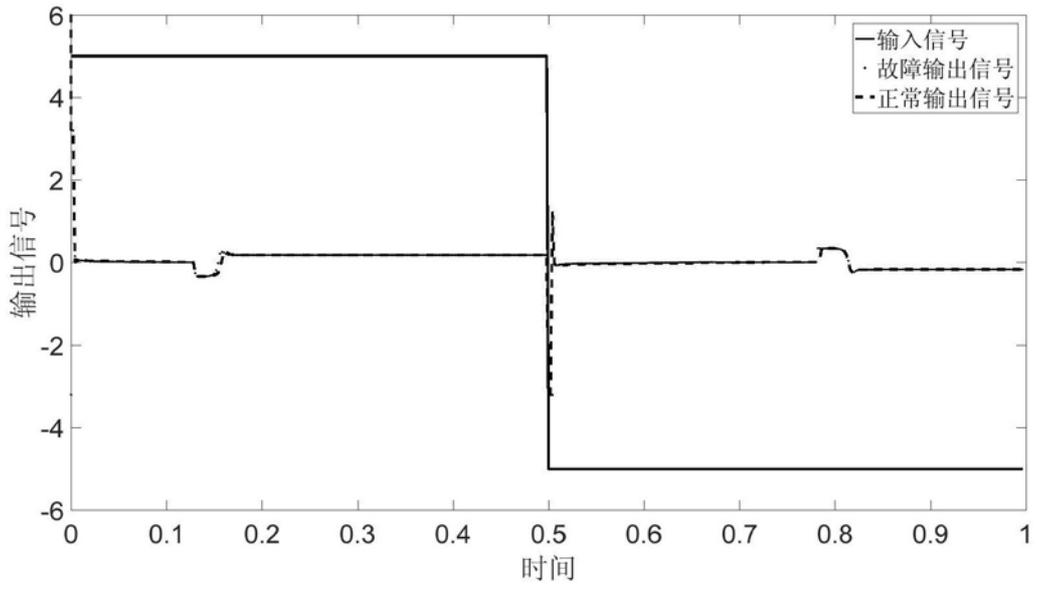


图5a

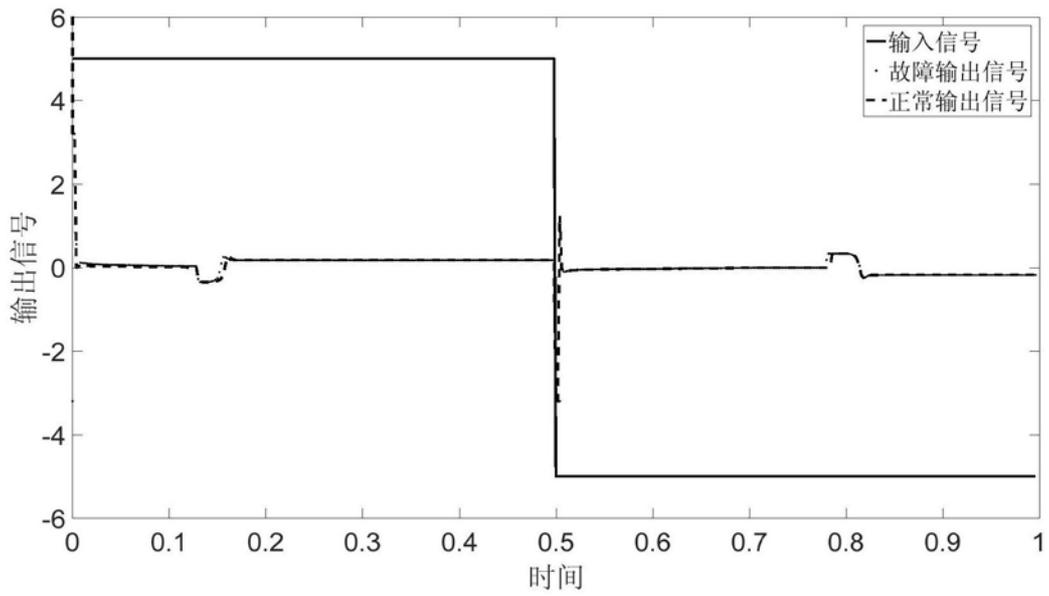


图5b

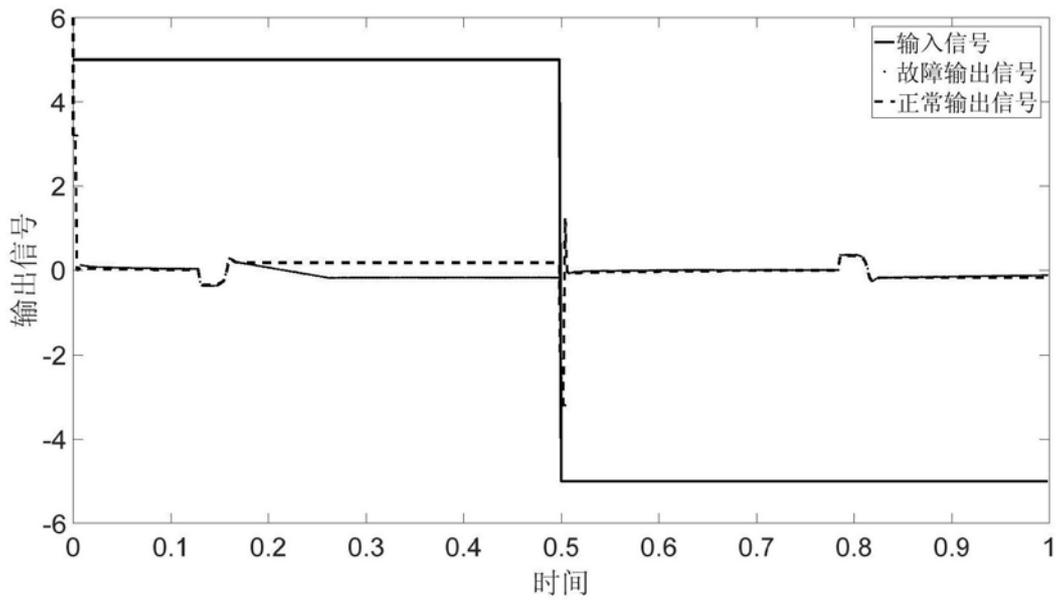


图5c

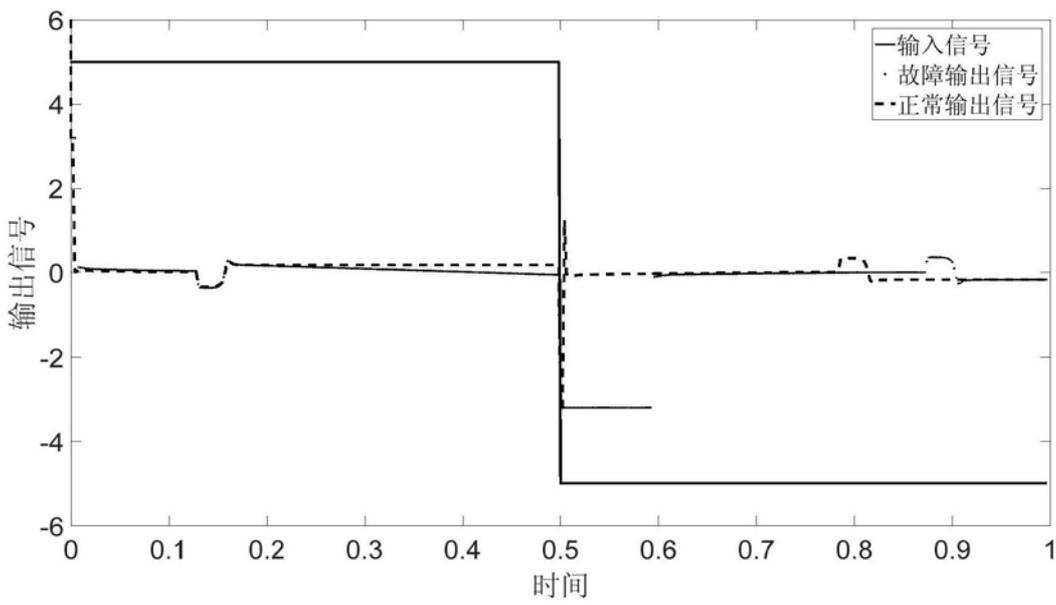


图5d

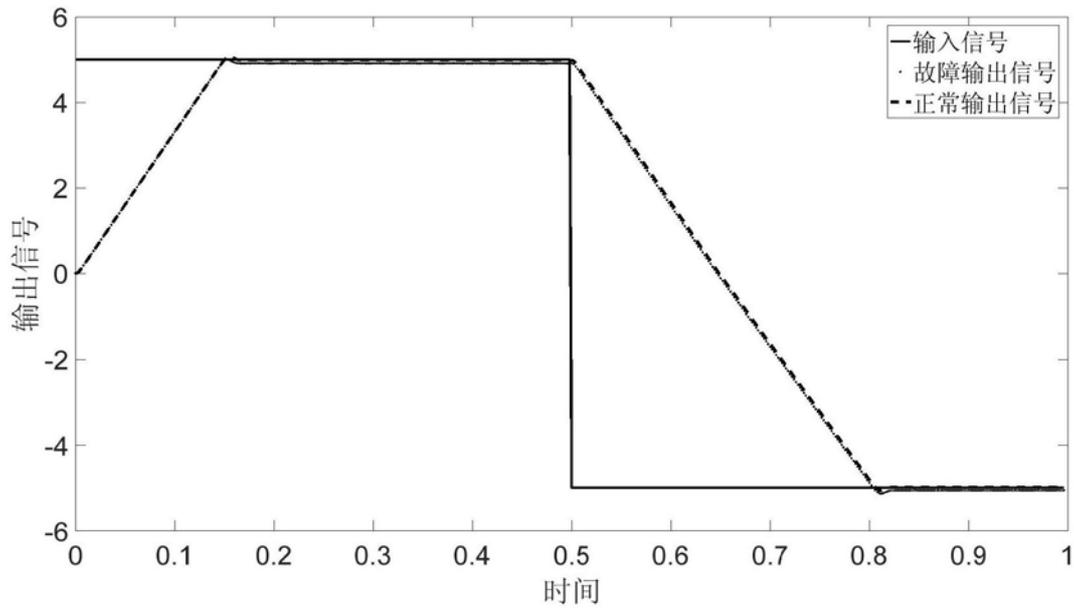


图6a

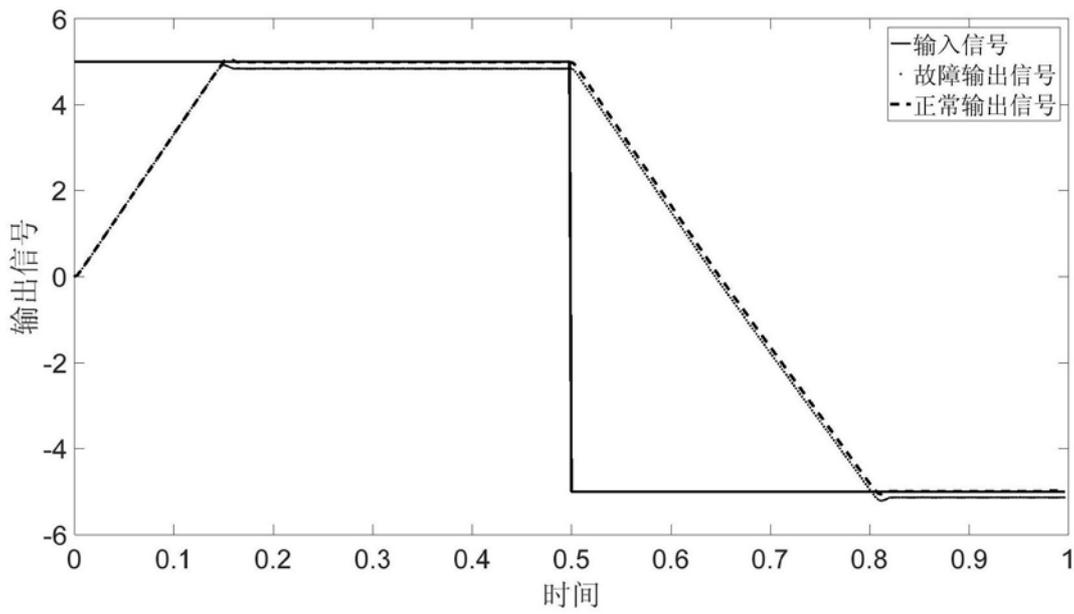


图6b

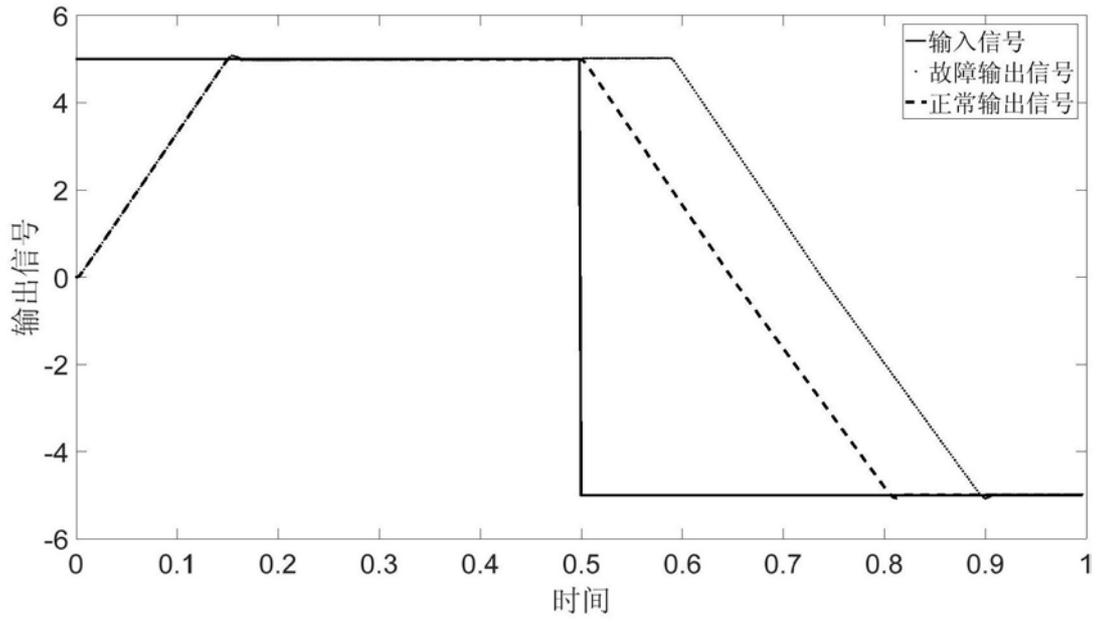


图6c

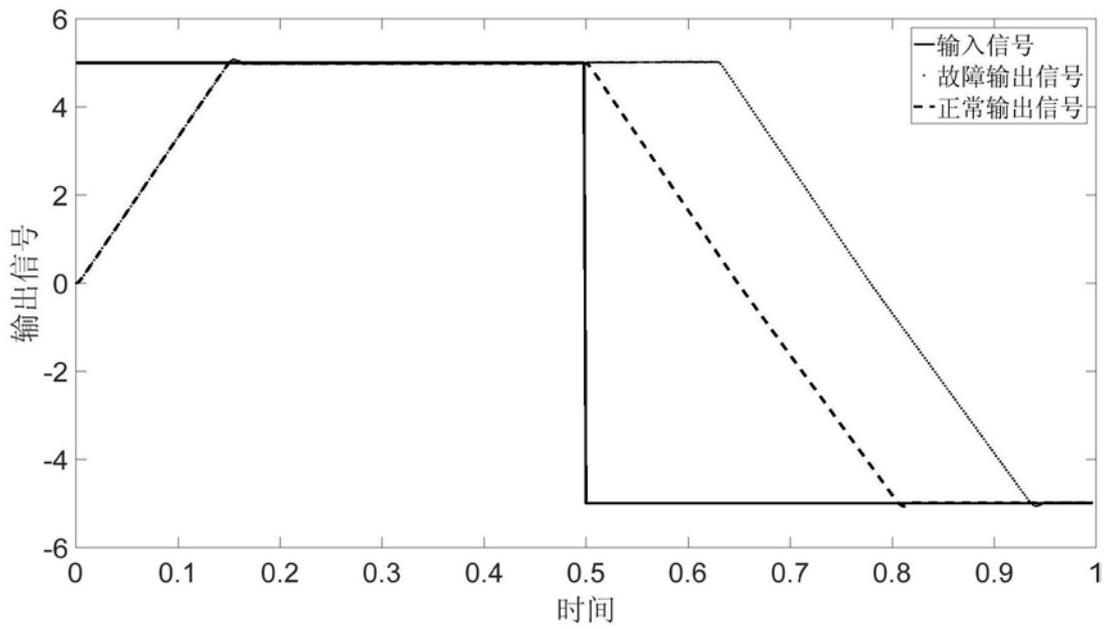


图6d

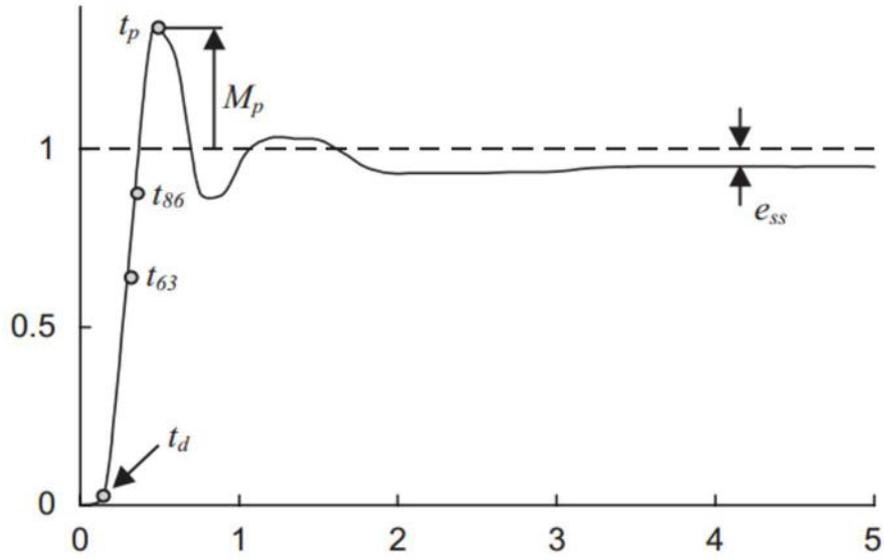


图7

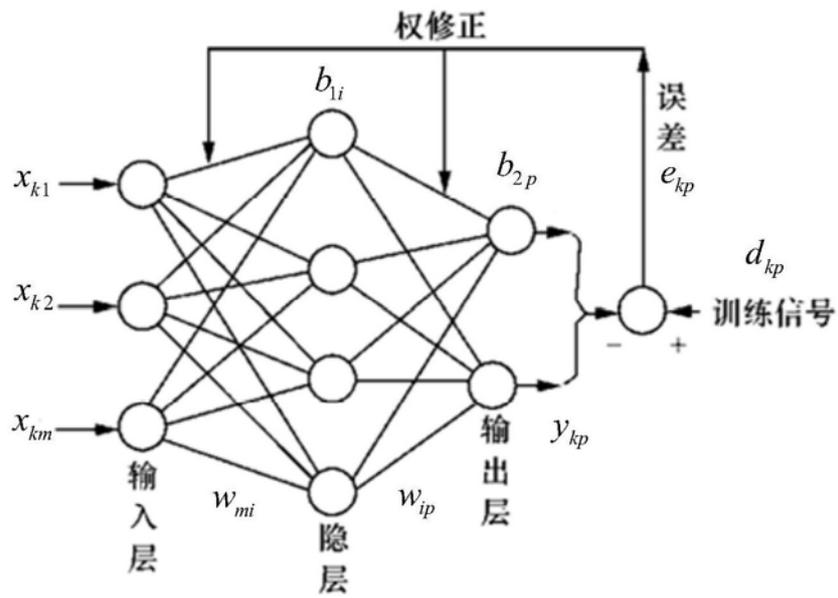


图8

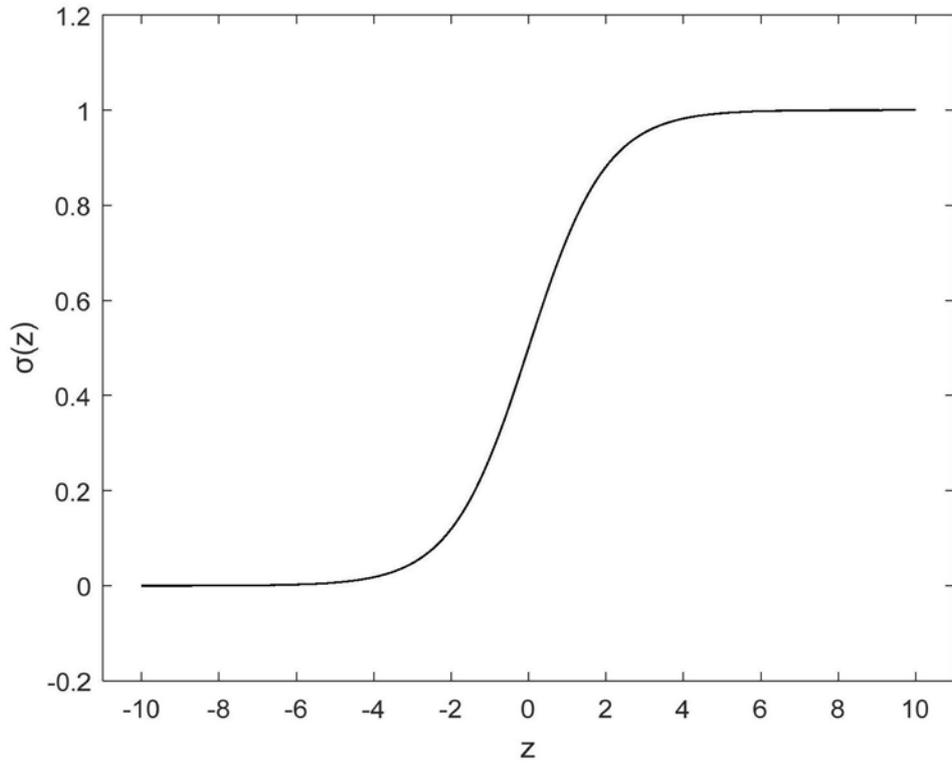


图9

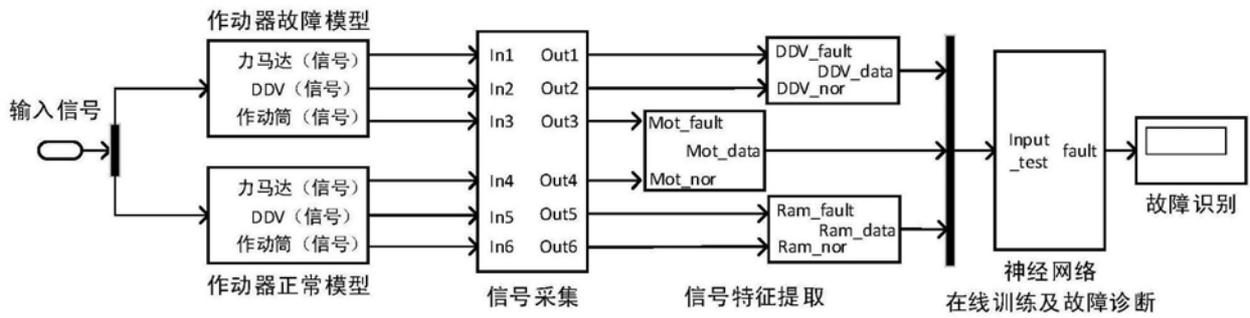


图10