

# 基于神经网络算法的液体火箭发动机 实时故障检测方法研究

丁伟程, 吴建军, 刘洪刚

(国防科技大学航天与材料工程学院, 湖南 长沙 410073)

**摘 要:** 以某液体火箭发动机为研究对象, 将实时故障检测作为中心, 分析了神经网络算法的特点及其实现步骤, 利用 Lab Windows/CVI 与 MATLAB 混合编程的原理, 实现和改进了基于神经网络的发动机实时故障检测方法, 并用多次试车数据进行了检验。试车数据验证结果表明, 该方法能及时、准确、有效地检测发动机稳态过程的故障。研究结果对发展未来液体火箭发动机的箭载故障检测系统具有重要的参考价值。

**关键词:** 神经网络; 液体火箭发动机; 混合编程; 实时故障检测

**中图分类号:** V434

**文献标识码:** A

**文章编号:** (2005)05-0005-06

## Real-time fault detection method based on neural network algorithm for LRE

Ding Weicheng, Wu Jianjun, Liu Honggang

(Inst. of Aerospace and Material Engineering, National Univ. of Defense Technology, Changsha 410073, China)

**Abstract:** The real-time fault detection for a liquid-propellant rocket engine (LRE) is put on emphasis in this paper. The real-time fault detection based on the neural network algorithm is realized and ameliorated by means of analyzing the characteristic and realization steps of neural network (NN) algorithm, using the composite programming method with Lab Windows/CVI and MATLAB. The algorithm proposed in the paper has been tested and verified with a large amount of hot-fire test data of LRE. The results show that the method can detect the faults occurred in the steady-state process of LRE efficiently and accurately. The method presented can be valuable for the development of on-board fault detection system of LRE.

**Key words:** neural network; liquid-propellant rocket engine; composite programming; real-time fault detection

## 1 引言

在我国,目前液体火箭发动机试车中主要采用的仍然是针对几个重要参数的红线监控方法(Redline),这种方法故障覆盖面很有限,不能预示故障的发展,且极易因为传感器的故障而发生故障的误检测。其它基于数据统计的方法虽然简单可靠并已得到了广泛应用,但其准确程度非常依赖于试车数据的统计结果;基于信号分析的方法所利用的一般是振动和高频压力信号,不适合对缓变参数进行故障检测;基于数学模型的方法虽具有较强的检测能力,但计算量大且必须有准确的数学模型做基础,而对于液体火箭发动机,很难建立其准确的数学模型。而人工神经网络方法(Artificial Neural Networks, ANN)作为近年来飞速发展的一种全新的故障检测方法,因其能模仿人脑的思维,并具有联想、记忆和很强的数据处理能力,有着很好的应用前景<sup>[1]</sup>。而虚拟仪器编程语言 Lab Windows/CVI 和 MATLAB 作为两种功能强大的软件,分别在仪器软实现和数值计算领域发挥了重大作用。将两者有机地结合起来实时高效地处理液体火箭发动机基于神经网络的故障检测问题,实现“强强联合”是本文研究的重点。本文针对液体火箭发动机稳态工作过程,实现和改进了基于神经网络的实时故障检测算法。试车数据验证结果表明,该方法检测准确及时,效果较好。

## 2 神经网络算法分析及实现步骤

神经网络具有强大的逼近任意非线性函数的功能,在用于系统辨识时无需数学模型,只需向训练好的网络输入测量数据,再将网络输出结果与实际结果进行比较、分析、判断即可,准确度极高而且可以根据用户的不同需求对网络进行相应的修改和再训练,对实际问题的适应性极强。因此,应用神经网络实现液体火箭发动机的实时故障检测具有相当强的可行性和必要性。

神经网络用于系统辨识的实质就是选择一个适当的神经网络模型来逼近实际系统<sup>[2]</sup>,神经网络算法的实现具体分以下几个步骤:

万方数据

### 2.1 神经网络的训练

这是进行基于神经网络算法的实时故障检测的基础。为了得到理想的仿真输出,并以此作为是否发生故障的判断依据,就必须要有训练良好的网络。

神经网络的训练大致分为以下三个步骤:

#### (1) 生成神经网络结构并初始化

根据实际问题的需要,选取隐含层的层数,接着确定输入层、隐含层和输出层神经元的个数,选择各层神经元的传递函数和网络训练函数,然后对各层网络权值或阈值进行初始化<sup>[3]</sup>。

#### (2) 确定神经网络的学习规则

反映信息存储特征的权值一般是通过学习得到的。神经网络的学习方法可以分为有导师的学习(又叫有监督的学习)和无导师的学习(又叫无监督的学习)两类。本文采用的 BP(Back-Propagation)神经网络属于有导师的学习网络。

#### (3) 选取合理的样本数据进行训练

训练样本选取的合理与否将对整个神经网络的正确逼近程度产生很大的影响。本文在充分了解和掌握发动机工作特点的基础上,选取了最能代表其工作特性的正常数据来训练网络。当然,在开始训练之前应该对网络的学习速率、期望误差等参数进行适当设置。

可以看到,以上三个步骤中一些参数的选取和设置都是没有固定的模式可以遵循,也就无所谓最好的。只有通过不断地进行训练效果的对比才能得到比较符合实际情况、相对令人满意的网络结构。

### 2.2 神经网络的仿真

在经过了多次的反复训练后(每次都需进行网络参数的调整 and 比较),假定已经得到了一个相对较好、符合设计精度的神经网络,接下来应该将包括训练样本在内的所有试车数据作为测试样本进行神经网络的仿真,对比得到的结果和实际情况是否吻合。已经训练好的网络结构就代表了输入向量与输出向量之间的一种非线性关系。对于本文实现的神经网络算法而言,其实时性就体现在利用网络结构进行仿真时是否能达到要求的计算反应时间。这一点本文将在第 3 节中的实例验证时具体分析。

## 2.3 神经网络算法检测的判断准则

根据神经网络方法的“黑箱”特点,在经过多次很好的训练后,结构、权值和阈值已确定的神经网络辨识模型能正确地描述发动机正常工作过程中输出变量与输入变量之间的数据结构关系<sup>[8]</sup>。从理论上说,这种关系适用于发动机的整个工作过程。但是,于发动机启动过程和稳态过程工作特性差异很大,所以最好分阶段来对这两个过程进行训练和检测。

对于具有多个输入参数( $R$ 维向量)一个输出参数(1维向量)的神经网络而言,在发动机的正常工作过程(非训练过程)中,其期望输出 $\hat{y}$ (即神经网络运算后的仿真辨识输出)与实际测量值 $y$ 的误差应该在一个有限的误差带中,该输出参数的误差带就可以确定为故障检测的阈值范围,记作 $dm$ 。 $dm$ 可以通过对大量辨识误差的统计处理(如均值、方差)来确定。如果发动机工作出现异常,那么正常工作过程的神经网络辨识模型将不再有效,体现在模型输出值和实测值之间的误差上就是该误差值将超出阈值带 $dm$ 的范围<sup>[4]</sup>。

因此,可以由式(1)给出故障检测的决策标准,其中 $d$ 为实际误差值:

$$\begin{aligned} \text{若 } d > dm \text{ 则 该参数异常;} \\ \text{若 } d \leq dm \text{ 则 该参数正常。} \end{aligned} \quad (1)$$

为了降低检测过程中的误报警率,提高检测方法的鲁棒性,可引入持续性检验指标,即:当 $d = \min(\Delta y_1, \Delta y_2, \dots, \Delta y_k) > dm$ 时,才判断发生故障。其中的 $\Delta y_k = |\hat{y} - y|$ , $k$ 为持续检验时间的宽度。还可引入多参数原则来进一步降低误报警率,即针对发动机的多个重要监控参数都执行误差带的判断标准式(1),只有当 $n$ 个( $n$ 根据实际需要指定)参数的误差同时超出了各自误差阈值带时才发出报警。

而对于具有多个输入参数( $R$ 维向量)多个输出参数( $R$ 维向量)的神经网络而言,可以定义一个综合检测指标 $S_{ee} = \sqrt{\sum_{i=1}^m (e_i^2 / y_i^2)}$ 来进行综合判断<sup>[1]</sup>。其中神经网络辨识误差向量定义为: $e = [e_1, e_2, \dots, e_m]^T \Delta [\hat{y}_1 - y_1, \hat{y}_2 - y_2, \dots, \hat{y}_m - y_m]^T$ ,神经网络期望输出向量定义为: $Y = [\hat{y}_1, \hat{y}_2, \dots, \hat{y}_m]^T$ 。阈值定义为 $S_m$ ,可以由发动机正常过程中对该综合检

测指标的统计处理(均值、方差等)获得。可以给出故障检测的决策标准<sup>[1]</sup>为:

$$\begin{aligned} \text{若 } S_{ee} = \sqrt{\sum_{i=1}^m (e_i^2 / y_i^2)} > S_m, \text{ 则发动机工作异常;} \\ \text{若 } S_{ee} = \sqrt{\sum_{i=1}^m (e_i^2 / y_i^2)} \leq S_m, \text{ 则发动机工作正常。} \end{aligned} \quad (2)$$

如果只是监控少数几个测量参数,并不能全面地反映发动机的工作特性和实际运行状态,对实时检测的效果会大打折扣,甚至导致检测结果的不及时、不准确。本文结合试车的具体要求,设计的是多输入多输出的神经网络,并采用了综合检测指标来进行实时检测。

## 3 稳态过程基于神经网络检测算法的实现与验证

对于相对启动过程而言简单平稳一些的稳态过程,利用多输入多输出的BP神经网络能比较全面地描述发动机的工作状态,很好地反映发动机的工作特性。

### 3.1 参数的选取和训练样本的确定

根据发动机稳态阶段的工作特点和实际试车的需要,选取氧泵出口压力(Pey)、推力室氧喷前压力(Py)、推力室压力(Pk)、氢泵出口压力(Per)、氢冷却套出口压力(Pel)、发生器氧喷前压力(Pfy)、发生器氢喷前压力(Pfr)、燃气发生器压力(Pf)、氧泵入口压力(Poy)、氢泵入口压力(Por)、氢涡轮泵转速(Nwr)、氧涡轮泵转速(Nwy)、发动机液氢流量(Gr)和发动机液氧流量(Gy)等14个参数组成输入向量,选取氧泵出口压力(Pey)、推力室氧喷前压力(Py)、推力室压力(Pk)、氢泵出口压力(Per)、氢冷却套出口压力(Pel)、发生器氧喷前压力(Pfy)、发生器氢喷前压力(Pfr)、燃气发生器压力(Pf)、氢涡轮泵转速(Nwr)、氧涡轮泵转速(Nwy)、发动机液氢流量(Gr)和发动机液氧流量(Gy)等12个参数组成输出向量。经过对已有的30次正常稳态过程试车数据的分析评价,试车TEST19-3稳态过程的干扰因素比较小,发动机工作参数的一致性很好,故选用了TEST19-3稳态过程的300个采样点进行神经网络的训练<sup>[1]</sup>。

### 3.2 BP 网络的训练

在定义了神经网络的训练样本（包括输入向量和输出向量）后，进行液体火箭发动机试车数据的转换和归一化等前期数据处理，然后将其导入构建好的神经网络<sup>[1,4]</sup>。接着对神经网络的权值和阈值进行初始化，开始训练网络。其中神经网络的各参数可以根据需要调整（本例中采用的是 14—23—12 的三层 BP 神经网络，隐含层和输出层传递函数分别用的是“Tansig”和“Purelin”，训练函数是泛化性较好的“Trainbr”，直至训练的效果较好（本例 epochs=500 步），达到设定的误差精度（本例 goal=10<sup>-6</sup>）样就得到了可用来进行实时辨识的网络。

### 3.3 BP 算法基于 LabWindows/CVI 与 MATLAB 混合编程的实现

虚拟仪器编程语言 Lab Windows/CVI 是美国 NI (National Instruments, 国家仪器) 公司开发的 32 位面向计算机测控领域的软件开发平台。它以 ANSI C 为核心，将 C 语言平台与数据采集、分析和表达等测控专业工具有机地结合起来，进一步加强了对数据库和网络的支持能力，更加适合分布式测控技术的发展要求<sup>[2,3]</sup>。

而 MATLAB 推出的针对神经网络的功能强大的工具箱，用 MATLAB 特有的语言构造出一系列典型的神经网络的激活函数、修正网络权值和阈值的规则以及网络设计、学习、训练的整套过程，使得神经网络的设计者可以根据自己的需要去调用神经网络工具箱中相应的子程序，从而大大提高了运用神经网络的效率和解题质量<sup>[4,5]</sup>。

因此，本文使用 LabWindows/CVI 自身带有的丰富的仪器面板控件及用户界面库函数<sup>[3,4]</sup>，并以 MATLAB 引擎函数的形式来将两者有机地结合起来，实现了 LabWindows/CVI 与 MATLAB 的“混合编程”<sup>[8]</sup>，神经网络实时故障检测算法的主界面如图 1 所示。

### 3.4 BP 算法实时性的实现

算法的实时性是最重要也是最令人关注的。神经网络的训练可以在试车之前就完成，是一个离线的过程，而真正需要实时运算的就是将待测数据代入到训练好的神经网络中进行仿真的过程。因此，仿真所需要的时间长短关系到能否实现神经网络算法的实时性。

为了达到实时性的要求，首先需要对测试样本进行一些处理：在仿真之前，运用 MATLAB 中最大最小型函数或零均值和标准偏差型函数将测试样本基于训练样本的范围进行归一化<sup>[7,8]</sup>，这样得到的输入向量和输出向量处在[-1,1]内或化为零均值和偏差为 1 的标准向量，大大降低了仿真时的计算复杂程度。

同时，为了减小 MATLAB 的实时计算量，将每个时刻某个参数的数据采样点作为一个 1×1 的矩阵来处理。那么，仿真的实质就是将大小为 1×1 的  $n$  个矩阵 ( $n$  为输入向量中包含测量参数的个数，即输入向量的维数) 组成一个  $n \times 1$  的新矩阵后，送到 MATLAB 中已经训练好的神经网络结构中得到网络的仿真输出结果矩阵 ( $m \times 1$ ,  $m$  为输出向量中包含测量参数的个数，即输出向量的维数)，然后将该仿真结果矩阵中的元素与同一时刻原来的实际输出矩阵中的对应元素比较，得到一个大小同样为  $m \times 1$  的误差矩阵。该误差矩阵就是进行实时故障检测的判断矩阵。利用训练好的网络进行实时仿真计算的步骤如图 2 所示。

对 MATLAB 而言，处理如此大小的矩阵是非常迅速的，这样就能保证计算每个时刻点的试车数据时系统的时耗尽可能的少<sup>[8]</sup>。最后，再将误差矩阵里的每一个元素用 postmnmx 或 poststd 函数还原成最初网络训练时输出向量所使用的单位，得到与原始试车数据范围一致的判断误差值<sup>[2,9]</sup>。实践证明，运用该改进方法可以使试车数据的两

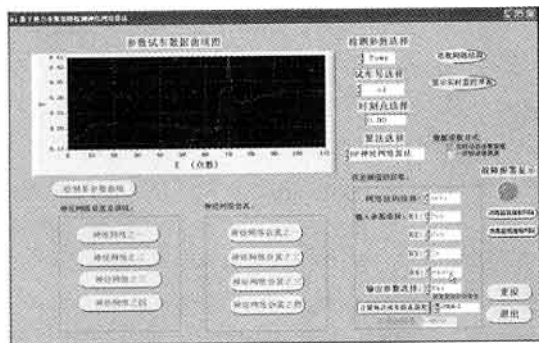


图 1 基于神经网络的实时故障检测算法的主界面

Fig.1 Interface of fault detection based on neural network  
万方数据

个采样点之间的仿真时间间隔控制在 10 毫秒左右,很好地达到了实时性的要求。

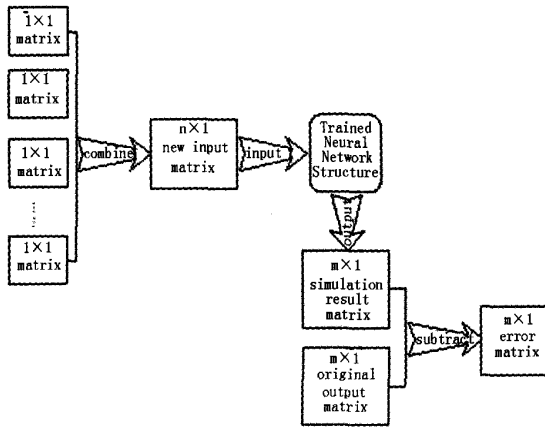


图 2 利用训练好的网络进行实时仿真计算的步骤示意图

Fig.2 Sketch map of real-time simulative operation based on trained neural network structure

3.5 试车数据仿真及阈值的确定

将待检测的试车数据作为网络的输入向量就可以进行实时仿真,得到关于 12 个输出监测参数的综合检测指标  $S_{ee} = \sqrt{\sum_{i=1}^m (e_i^2/y_i^2)}$  的实时值并与其阈值  $S_m$  进行比较,利用式(2)建立的检测策略可以判断该次试车的稳态过程是否正常。阈值  $S_m$  采用大量正常试车代入训练好的 BP 网络得到的综合检测指标结果的均值加上其 3~6 倍的标准差来确定<sup>[1]</sup>,大量的试车数据统计表明,在发动机稳态工作期间,大部分测量数据服从正态分布,因此,本例由正态分布的 99.73% 置信区间确定均值加上标准差的 3 倍作为检测阈值,效果较好。

仿真的同时可以将辨识误差曲线实时地画出。以上所述各步均可通过调用 MATLAB 引擎库函数来实现<sup>[4,5]</sup>。

表 1 BP 算法与红线关机算法的检测结果比较

Tab.1 Contrast of detection results between BP and redline

试验编号	红线检测时间/s	BP 检测时间/s
TEST10-1	43.21	42.80
TEST10-2	43.21	42.80
TEST15-5	275.80	275.50
TEST20-0	30.23	30.20

万方数据

共计用 4 次发动机故障数据对 BP 检测算法进行了验证,与红线关机的比较结果如表 1 所示。

图 3 为 BP 算法对 TEST10-1 试车数据的检测结果。图中曲线分别为稳态阶段综合检测指标  $S_{ee} = \sqrt{\sum_{i=1}^m (e_i^2/y_i^2)}$  的实时曲线和阈值线,由报警结果显示:算法在 42.80 秒报警。图 4 为 BP 算法对 TEST15-5 试车数据的检测结果,由报警结果显示:算法在 275.50 秒报警。图 5 为 BP 算法对 TEST20-0 试车数据的检测结果,由报警结果显示:算法在 30.20 秒报警。图 6 为 BP 算法对正常试车 TEST15-0 的实时检测曲线。

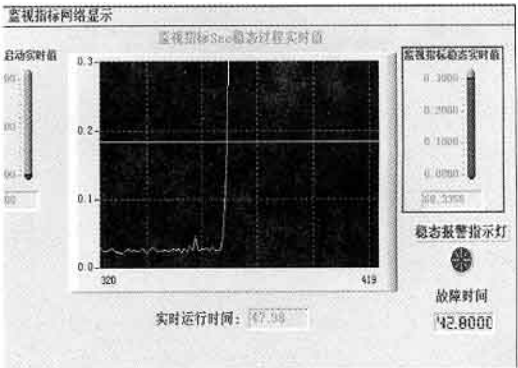


图 3 BP 算法对 TEST10-1 试车稳态过程的检测结果曲线图

Fig.3 BP detection curve of TEST10-1 steady-state process

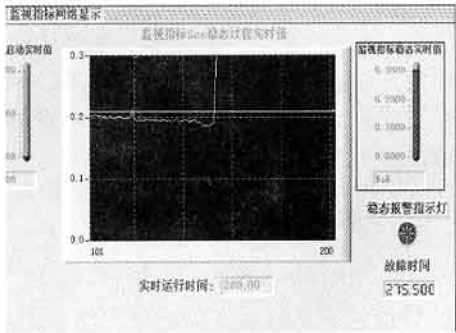


图 4 BP 算法对 TEST15-5 试车稳态过程的检测结果曲线图

Fig.4 BP detection curve of TEST15-5 steady-state process

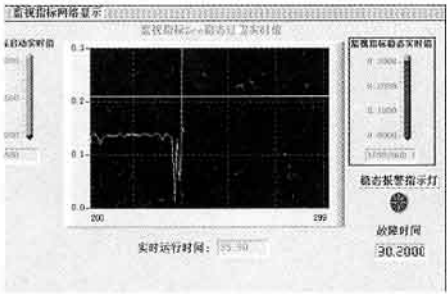


图 5 BP 算法对 TEST20-0 试车稳态过程的检测结果曲线图

Fig.5 BP detection curve of TEST20-0 steady-state process

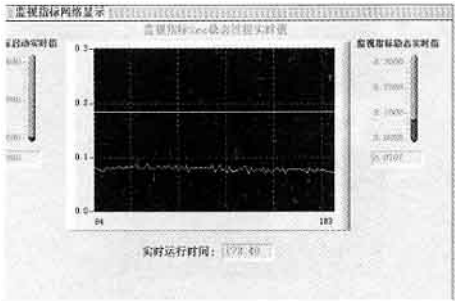


图 6 BP 算法对 TEST15-0 试车稳态过程的检测结果曲线图

Fig.6 BP detection curve of TEST15-0 steady-state process

4 结 论

本文通过发动机多次试车数据对所构建的 BP 神经网络的验证结果表明：基于神经网络的实时故障检测方法能及时准确地检测出试车故障且比红线系统的报警时间有不小的提前，没有出现误报警或漏报警的现象，充分体现了其可行性和优越性。在实际检测过程中，该算法很好地达到了实时性的要求（每两个实时数据点之间的仿真辨识时间只需 10 毫秒左右），是具有很高推广价值的一种测控领域里的新方法。同理，基于 RBF

（Radial Basis Function 径向基函数）神经网络进行液体火箭发动机实时故障检测的方法也得到了实现和改进。由神经网络逼近任意非线性函数的强大功能以及用于系统辨识时无需数学模型的极强适应性分析可知，BP 和 RBF 算法同样可以适应于发动机启动和关机阶段的实时故障检测，现有试车数据的验证结果证明了其可行性。

总之，基于神经网络的液体火箭发动机实时故障检测方法为进一步实现现场级故障检测与报警系统奠定了坚实的基础，并对发展未来液体火箭发动机箭载故障检测系统有重要参考价值。

参考文献：

[1] 黄敏超. 液体火箭发动机故障的神经网络诊断研究[D]. 国防科技大学博士学位论文, 1998.

[2] 刘君华主编. 虚拟仪器编程语言 Lab Windows/CVI 教程[M]. 北京：电子工业出版社, 2001.

[3] 张毅刚, 乔立岩, 等. 虚拟仪器软件开发环境 Lab Windows/CVI 6.0 编程指南[M]. 北京：机械工业出版社, 2002.

[4] 闻新, 周露, 等. MATLAB 神经网络仿真与应用[M]. 北京：科学出版社, 2003.

[5] 许东, 吴铮. 基于 MATLAB6.X 的系统分析与设计——神经网络(Version 2)[M]. 西安：西安电子科技大学出版社, 2002.

[6] 杨成, 申丽然, 等. MATLAB 引擎的 VC++6.0 接口调用及其面向对象编程[J]. 应用科技, 2001, (11).

[7] Martin T Hagan, Howard B Demuth, Mark H Beale. Neural network design[M]. PWS Publishing Company, 1996.

[8] The Math Works Inc. Neural Network Toolbox User's Guide(Version 4)[M]. 2001.

[9] 吴建军. 液体火箭发动机故障检测与诊断研究[D]. 国防科技大学博士学位论文, 1995.

（编辑：陈红霞）