

基于 GD-GEP 算法的航空发动机振动监控

单兵, 倪世宏, 谢川, 孙逢啸

(空军工程大学 航空航天工程学院, 西安 710038)

摘要: 针对传统基因表达式编程算法 (GEP) 在进行函数挖掘时易陷入局部最优以及收敛精度不高等问题, 提出了一种基于基因多样性的 GEP 算法, 该算法采用基于基因多样性的初始种群生成策略, 引入了种群更新策略, 使得种群基因多样性增加; 仿真试验表明, 该算法具有更好的全局搜索能力和更高的解精度; 基于某型飞机飞行数据记录器记录的飞行数据, 利用 GD-GEP 算法进行航空发动机模型辨识, 将辨识得到的模型用于发动机振动值监控, 真实的振动故障数据验证了该方法能够成功监测到振动故障。

关键词: 基因表达式编程; 飞行数据; 航空发动机; 模型辨识; 振动值监控

Vibration Monitoring of Aircraft Engine Based on GD-GEP

Shan Bing, Ni Shihong, Xie Chuan, Sun Fengxiao

(Aeronautics and Astronautics Engineering College, Air Force Engineering University, Xi'an 710038, China)

Abstract: In order to escape from local optimum and get a higher accuracy, GEP based on gene diversity is proposed, which using initial population producing strategy based on gene diversity, introducing the population update strategy, improved genes diversity of population. The simulation results show that GD-GEP has a better global search ability and higher accuracy. Using the flight data recorded by FDR of a certain type of aircraft, the model of aircraft engine based on GD-GEP is set up and used to monitor the vibration of engine, real malfunction data proved the method to be effective.

Key words: gene expression programming; flight data; aircraft engine; model identification; monitoring of vibration

0 引言

从大量飞行数据中提炼数学模型, 拟合相关变量之间的依赖关系, 是飞行数据深层次利用的方法之一。在基于飞行数据的发动机模型辨识中, 目前常用的方法有支持向量机^[1]和神经网络^[2], 这两种方法均取得了良好的模型精度, 但都不能给出一种显式的表达形式。基因表达式编程算法是融合了遗传算法和遗传编程的优势提出的一种新的进化算法, 独特的编码和解码方式使得它能够挖掘样本数据间的函数关系, 并给出一个显式的数学表达形式, 且比传统进化计算快 2~4 个数量级^[3]。本文利用基因表达式编程算法进行发动机的模型辨识, 进而利用模型进行振动值的监控。

标准 GEP 算法存在早熟和易陷入局部最优等不足, 从而使得如果按照传统的方式挖掘, 挖掘出的函数对样本数据的拟合精度不高。为此, 本文基于基因多样性对传统 GEP 算法进行了改进 (GD-GEP), GD-GEP (GEP based on gene diversity) 算法可提高进化过程中的种群基因多样性, 改善算法性能, 提高模型对样本的拟合精度。

1 GD-GEP 算法

1.1 标准 GEP 算法

基因表达式编程是融合遗传算法 (GA) 和遗传编程 (GP) 的优点而提出的一种新型进化算法, 具有编码简单、处理问题灵活、收敛速度快等优点, 在函数发现^[4-5]、时间序列预测^[6]、关联规则挖掘^[7]、支持向量机参数优化^[8]等方面均取

得了良好的应用。

在基因表达式编程算法中, 个体被编码成线性符号串的形式, 其基因由头部和尾部组成, 头部符号可以来自函数符号集和终结符号集, 尾部符号只能来自终结符号集, 其头部和尾部长度满足如式 (1) 的函数关系, 其中 h 为头部长度, t 为尾部长度, n 为基因中含有的所有函数符号的最大参数数目。

$$t = h \times (n - 1) + 1 \quad (1)$$

表达式树是 GEP 个体的表现形式, 可以由基因按照从上到下, 从左到右的顺序遍历得到。如式 (2) 所示一个基因 (加黑部分为尾部, S 代表正弦函数), 其对应的表现型 (表达式树) 如图 1 所示。

$$+ / - aSabcbac \quad (2)$$

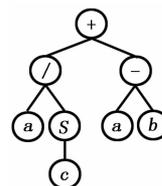


图 1 表达式 (2) 对应的表达式树

该表达式进而可解码为方程: $a/\sin(c) + (a - b)$ 。

1.2 基于基因多样性的初始种群生成策略

基因多样性是影响演化算法性能的一个重要因素, 多样性好的种群能通过遗传算子的作用, 发生组合效应, 产生更多的模式, 从而使得算法搜索到全局最优解的概率增大^[9]。标准 GEP 算法采用随机初始化种群的策略, 产生的种群基因多样性有限, 不利于进化。为了使种群中的基因模式最大化, 提高种群多样性, 引入了基因空间均匀分布思想, 并给出了该思想的一种实现方法。为了避免死基因的出现, 将基因分为头、身、尾三部分^[10]。

收稿日期: 2013-12-20; 修回日期: 2014-02-18。

基金项目: 航空科学基金 (20110896009); 陕西省自然科学基金青年项目 (2012JQ1019)。

作者简介: 单兵 (1989-), 男, 河南商丘人, 硕士研究生, 主要从事飞行数据智能处理方向的研究。

该算法的主要思想是: 给每个基因位一个可取符号集, 通过遍历该符号集, 使得每个基因位上的基因符号均匀分布, 从而提高基因的多样性。

具体实现思路为: 以单基因染色体为例, 设函数符号集为 $F = \{+, -, *\}$, 终结符号集为 $T = \{a, b\}$, 染色体头部长度 $h=1$, 身部长度 $s=3$, 尾部长度 $t=5$, 则头部可取符号集为 $F = \{+, -, *\}$, 身部可取符号集为 $S = \{+, -, *, a, b\}$, 尾部可取符号集为 $T = \{a, b\}$ 。对于染色体头部基因符, 从函数符号集 F 中取出, 以保证基因的有效性; 对于身部的任一个基因位, 将符号集 S 任意打乱排列顺序作为该基因位的可取符号集进行顺序遍历, 即种群中的第 i 个个体的该位基因取打乱排列顺序后的 S 中第 i 个基因符, 直到 S 中的基因符取遍, 完成一个循环。尾部基因符的获得同身部。多基因染色体实现方法类似。

表 1 给出了应用该方法产生的初始种群前 10 个个体的各基因位的基因符。

表 1 部分个体示意

	基因位								
	0	1	2	3	4	5	6	7	8
基因符号	+	*	-	b	a	b	b	b	a
	*	+	+	+	b	a	a	a	b
	-	a	b	a	b	a	a	b	a
	+	b	a	*	a	b	b	a	b
	*	-	*	-	a	b	a	a	a
	+	*	a	b	b	a	b	b	b
	+	a	*	*	b	a	b	a	a
	-	-	b	-	a	b	a	b	b
	-	b	+	a	a	b	b	b	a
	-	+	-	+	b	a	a	a	b

以身部的基因位 1 为例, 可以看出, 种群中的前 5 个个体中, 身部可取符号集 S 中的基因符分别出现一次, 接下来的 5 个个体中又分别出现一次。对于尾部的基因位 5, 前两个个体中尾部可取符号集 T 中的基因符分别出现一次, 接下来的 2 个个体中又分别出现一次。因此, 就整个种群而言, 各基因符在相应基因位等概率均匀出现, 从而实现了基因的均匀分布, 增加了种群中基因的多样性。

1.3 种群更新策略

在 GEP 进化初期, 种群多样性良好, 算法进化较快, 随着迭代步数的增加, 种群发生基因趋同效应, 多样性下降, 使得进化速度缓慢。为了增加进化过程中种群的多样性, 增加算法跳出局部最优的能力, 引入了种群更新策略, 其主要思想是: 如果算法连续 n 代未进化, 随机产生一部分个体注入到种群中^[11], 替换掉原种群中相同数量的适应度较低的个体, 该策略通过注入新的基因, 提高了种群中基因的多样性, 从而提高 GEP 算法的进化效率。

1.4 算法流程

GD-GEP 算法的流程图如图 2 所示。

从图 2 可以看出, GD-GEP 算法和标准 GEP 算法相比, 改变了种群初始化方法, 增加了种群更新的步骤。

1.5 算法性能测试

为了测试 GD-GEP 算法的性能, 在实验中对两个函数进行挖掘: 一元二次函数 $F_1 = \pi r^2$, 复杂三角函数 $F_2 = \frac{\sin(a)\cos(b)}{\sqrt{\exp^c}} + \tan(d - e)$ 。对每个函数的每个自变量在区间

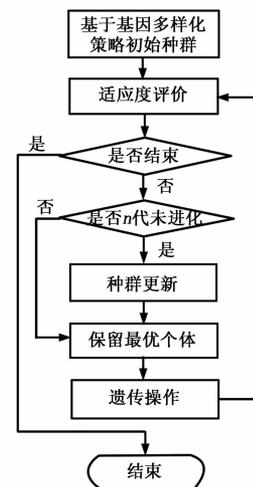


图 2 GD-GEP 算法流程图

$[-50, 50]$ 分别随机产生 100 个数据作为参数集, 然后分别求得两个函数在参数集上对应的目标值, 构成 100 组训练样本, 对每个函数重复 50 次挖掘实验, 参数设置如表 2^[12]。取结果的平均值作为最后的实验结果。结果如表 3 所示。

表 2 参数设置

函数	最大进化代数	函数符集	终结符集	基因个数	头部长度	连接函数	变异率	插串率	交叉率	适应度函数
F_1	1 000	+ - * /	a	3	6	+	0.044	0.1	0.3	复相关系数函数
F_2	5 000	+ - * / QESCT	abcde	3	6	+	0.044	0.1	0.3	复相关系数函数

表 3 性能对比

	函数	平均适应度	平均进化代数
标准 GEP	F_1	0.995 9	420
	F_2	0.875 2	2 907
GD-GEP	F_1	0.996 1	393
	F_2	0.913 6	2 410

从表 3 可以看出, GD-GEP 算法在对函数 F_1 、 F_2 进行挖掘时, 无论是对样本数据的拟合精度还是收敛速度均得到了提高。GD-GEP 算法对函数 F_1 的拟合精度提高有限, 这是因为传统 GEP 算法已经能够很好地拟合 F_1 函数, 提升空间有限。

2 航空发动机加力起飞过程模型辨识及振动值监控

在飞机部件出现故障之前, 可通过监控其对应的飞行参数来判断该部件是否有故障或故障趋势^[13]。利用飞行参数对振动值进行监控, 首先要建立基于飞行数据的发动机振动模型。

某型飞机起飞时发动机一般都是处于加力状态, 所以提取起飞阶段加力稳态时的数据作为研究对象。

2.1 数据提取

加力起飞稳定时满足如下条件:

$$|dPYD| < 1 \quad \& \quad Fop = 1 \quad \& \quad LGf = 1 \quad \& \quad Vcbc < 240 \quad (3)$$

式中, $dPYD$ 是油门杆变化量, Fop 是加力信号, LGf 为前起落

架放下标志, V_{cbc} 为飞机速度。通常振动值的稳定相对于发动机稳定要滞后 3~5 s, 所以提取数据时要求在上式连续 10 s 成立的前提下, 取后 5 s 数据, 提取的参数有: 振动值 V_{ib} 、高、低压转子转速 N_2, N_1 、高、低压压气机进口导流叶片转角 $HAPIC, BHA$ 和喷口指示值 PC 。

2.2 航空发动机加力起飞过程描述

某型飞机加力起飞过程的振动模型可用如下关系式来描述:

$$V_{ib} = f(N_1, N_2, BHA, HAPIC, PC) \quad (4)$$

式中, f 为描述发动机加力起飞过程中各参数间变化关系的非线性函数。基于上式的发动机振动模型, 既涵盖了由转子不平衡等因素引起的振动情况 (由 N_1, N_2 加以体现), 又涵盖了燃气——空气通道引起的发动机振动情况 (由 $HAPIC, BHA, PC$ 加以体现)。

2.3 模型辨识及仿真验证

以某型 30# 飞机右发为例, 取 30 个架次的数据作为实验数据, 前 10 个架次的数据作为训练样本, 用来建立模型, 剩余 20 架次的数据作为测试样本, 用来验证模型的推广能力。式 (5) 为用 GD-GEP 算法辨识得到的发动机加力起飞过程振动模型:

$$V_{ib} = \sin(N_1 - \cos(N_2) + 10.0007) - \sin\left(\frac{N_1}{BHA}\right) + 0.5571 * \cos(\sin(HAPIC)) - \sin(PC) + 18.8355 \quad (5)$$

为了验证模型的精度及推广能力, 分别用模型对训练样本和测试样本进行拟合和预测。设单架次中振动值 V_{ib} 的实测数据为 $V_{ib}_i^t, t = 1, 2, \dots, 5$, 模型输出数据为 $\overline{V_{ib}_i^t}, t = 1, 2, \dots, 5$, 则定义每个架次模型的输出误差为:

$$E(V_{ib}_i) = \frac{1}{5} \sum_{t=1}^5 |V_{ib}_i^t - \overline{V_{ib}_i^t}|, i = 1, 2, \dots, 30 \quad (6)$$

式中, i 为架次序号。

以 $E(V_{ib})$ 作为指标验证模型质量, 与支持向量机所建模型进行对比, 结果如图 3 所示。

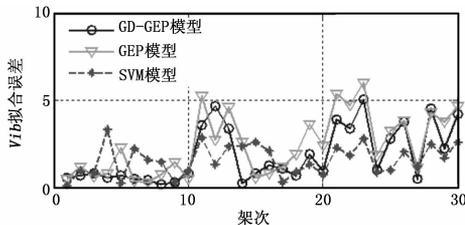


图 3 基于 GD-GEP、GEP 和 SVM 的模型对各参数的拟合比较

图 3 中 ‘o’ 线是基于 GD-GEP 算法辨识得到的模型对样本的拟合误差统计曲线, ‘v’ 线是基于标准 GEP 模型的误差曲线, ‘*’ 线是基于 SVM 模型的误差曲线, 前 10 个架次为 3 个模型对训练样本的拟合, 后 20 个架次为 3 个模型对测试样本的预测。从图 3 中可以看出, 基于 GD-GEP 辨识的模型, 无论对训练样本的拟合能力还是对测试样本的测试能力, 均能够达到与支持向量机模型相当的水平。可见, 在工程上, 将 GEP 方法用于航空发动机模型辨识是可行、有效的。

2.4 基于 GD-GEP 模型的振动值监控

将某型 30# 飞机右发 45 个架次的加力起飞阶段数据代入 2.3 节辨识的模型中, 把模型中振动值 V_{ib} 的输出误差按飞行架次的先后顺序排列成一组序列。其中图 4 为加力起飞阶段模型 V_{ib} 的拟合误差。

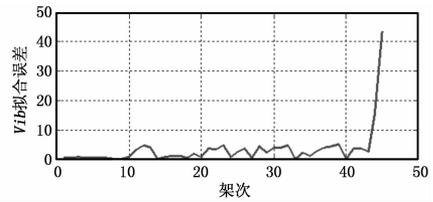


图 4 加力起飞阶段模型 V_{ib} 拟合误差

45 个样本中, 前 10 个是训练样本, 第 45 个架次为真实故障样本。将 1~40 架次数据作为无故障检测样本, 考察 40 个无故障检测样本, 用“峰度、偏度法”^[1] 检验偏差数据近似服从 $N(\mu, \sigma^2)$ 分布, 在置信概率为 99.87% 时, 把单侧置信区间的偏差上限 $\mu + 3\sigma$ 作为偏差监控门限, 30# 机右发加力起飞过程模型的输出数据与实测数据的偏差统计及监控门限见表 4。

表 4 加力起飞过程模型偏差统计及监控门限

模型	μ	σ	$\mu + 3\sigma$
起飞阶段模型	2.153 9	1.737 1	7.365 1

依照表 4 中的偏差监控门限, 共有 44、45 两个架次的模型输出偏差超过监控门限, 可以提前一个架次发现振动异常。

3 结束语

对标准 GEP 算法进行了改进, 使得算法更容易跳出局部最优解, 对样本的拟合精度提高。给定一组真实的飞行数据, 将改进后的 GEP 算法应用到航空发动机加力起飞过程振动模型的辨识中, 取得了比较好的模型精度。将模型用于航空发动机振动值的监控中, 实验结果表明了该方法的有效性。

相比于支持向量机“黑箱”式的模型, GEP 模型的显式表达形式更容易理解, 但其物理概念仍然不够明确, 模型的复杂度有待降低, 如何挖掘出更加简洁、能够根据物理实际正确反应各参数随动关系的模型是下一步的研究方向。

参考文献:

- [1] 倪世宏, 史忠科, 沙孟春. 一种基于支持向量机回归的发动机振动监控方法 [J]. 空军工程大学学报 (自然科学版), 2006, 7 (5): 1-3.
- [2] 姜涛, 李应红. 基于动态 RBF 网络的发动机起动过程模型辨识 [J]. 航空动力学报, 2002, 17 (3): 381-384.
- [3] Ferreira C. Gene Expression Programming: A New Adaptive Algorithm For Solving Problems [J]. Complex System, 2001, 13 (2): 87-129.
- [4] 向勇, 唐常杰, 朱明放, 等. 内嵌基因表达式编程及其在函数发现中的应用 [J]. 电子科技大学学报, 2011, 40 (1): 116-121.
- [5] Ferreira C. Function finding and the creation of numerical constants in gene expression programming [C]. Advance in Soft Computing Engineering Design and Manufacturing, 2003: 257-266.
- [6] Zuo Jie, Tang Chang Jie, Li Chuan, et al. Time series prediction based on Gene Expression Programming [J]. Lecture Notes in Computer Science, 2004, 3129 (1): 55-64.
- [7] 黄钢, 杨捷, 李德华, 等. 基于 GEP 与小生境的关联规则挖掘的研究 [J]. 计算机应用研究, 2009, 26 (1): 56-58.

```

TIM_CKD_DIV1;
    TIM_TimeBaseStructure.TIM_CounterMode = TIM_Counter-
Mode_Up;
    TIM_TimeBaseInit(ENCODER_TIMER, &TIM_TimeBase-
Structure);
    TIM_EncoderInterfaceConfig(ENCODER_TIMER, TIM_Enco-
derMode_TI12,
    TIM_ICPolarity_Rising, TIM_ICPolarity_Rising);
    TIM_ICStructInit(&TIM_ICInitStructure);
    TIM_ICInitStructure.TIM_ICFilter = ICx_FILTER;
    TIM_ICInit(ENCODER_TIMER, &TIM_ICInitStructure);
    TIM_ClearFlag(ENCODER_TIMER, TIM_FLAG_Update);
    TIM_ITConfig(ENCODER_TIMER, TIM_IT_Update, ENABLE);
    TIM1->CNT = COUNTER_RESET;
    TIM_Cmd(ENCODER_TIMER, ENABLE);
}
void TIM1_UP_IRQHandler(void)
{
    u8 temp=0;
    TIM_ClearFlag(ENCODER_TIMER, TIM_FLAG_Update);
}
s16 ENC_Get_Electrical_Angle(void)
{
    static u16 lastCount = 0;
    u16 curCount = ENCODER_TIMER->CNT;
    s32 dAngle = curCount - lastCount;
    if(dAngle >= 2000){
        dAngle -= (4 * ENCODER_PPR);
    }else if(dAngle < -2000){
        dAngle += (4 * ENCODER_PPR);
    }
    lastCount = curCount;
    return (s16)dAngle;
}

```

2.2 数据存储与通信软件模块

系统数据设定与参数保存用 SST25VF08 芯片构建了 1M 字节的数据存储区, 存储芯片与 CPU 直接采用了 SPI 的数据接口; 同时, 采用了 SP3232 构建了与上位触摸屏的 RS-232 串口通信接口, 串口通信协议采用了 MODBUS-RTU 模式, 同时主 CPU 上, 为了保存定时器对电机和编码器接口的实时相应, 在串口通信上采用了 DMA 模式, 数据通信交互不需要 CPU 的参与。下面列出部分 DMA 设定相应代码^[6]:

```

define SRC_USART2_DR 0x40004404
u8 USART2_DMA_Buf1[UART_DATA_LENGTH];
void USART_DMAToBuf1(void)
{
    DMA_InitTypeDef DMA_InitStructure;

```

```

    RCC_AHBPeriphClockCmd(RCC_AHBPeriph_DMA1, ENA-
BLE);
    DMA_DeInit(DMA1_Channel6);
    DMA_InitStructure.DMA_PeripheralBaseAddr = (u32)SRC_
USART2_DR;
    DMA_InitStructure.DMA_MemoryBaseAddr = (u32)USART2
_DMA_Buf1;
    DMA_InitStructure.DMA_DIR = DMA_DIR_PeripheralSRC;
    DMA_InitStructure.DMA_BufferSize = UART_DATA_
LENGTH;
    DMA_InitStructure.DMA_PeripheralInc = DMA_Periph-
eralInc_Disable;
    DMA_InitStructure.DMA_MemoryInc = DMA_
MemoryInc_Enable;
    DMA_InitStructure.DMA_PeripheralDataSize = DMA_Periph-
eralDataSize_Byte;
    DMA_InitStructure.DMA_MemoryDataSize =
DMA_PeripheralDataSize_Byte;
    DMA_InitStructure.DMA_Mode = DMA_Mode_Circular;
    DMA_InitStructure.DMA_Priority = DMA_Priority_High;
    DMA_InitStructure.DMA_M2M = DMA_M2M_Disable;
    DMA_Init(DMA1_Channel6, &DMA_InitStructure);
    USART_DMACmd(USART2, USART_DMAREq_Rx, ENABLE);
    DMA_Cmd(DMA1_Channel6, ENABLE);
}

```

3 结论

本系统主要根据双轴切片机床数控系统的需求, 构建了系统硬件方案与软件实现方法, 解决了以下几个问题: (1) 利用 CPU 自带定时器, 解决了伺服电机的加减速控制和编码器反馈问题; (2) 采用了 DMA 的方式, 实现了 MODBUS RTU 协议, 兼容了工业常用 PLC 与触摸方案; (3) 结合了实际的应用背景, 针对性进行了硬件平台和专用软件的开发。通过测试表明, 本系统提供的设计方案是可行的。

参考文献:

- [1] 王永红, 徐 炜, 郝立平. STM32 系列 ARM Cortex-M3 微控制器原理与实践 [M]. 北京: 北京航空航天大学出版社, 2008.
- [2] 鲍文昕. RS485/MODBUS 数据采集网络在自动测报系统的应用研究 [J]. 水利科技, 2006, (2): 42-44.
- [3] 王昌杰, 熊帮新. 一种传统车床控制系统改进方法的研究 [J]. 长江大学学报(自然科学版)理工卷, 2008, (2): 45-46.
- [4] 涂 焯, 彭 瑜, 周怡颀, 等. Modbus 通信协议的紫蜂无线传输网 [J]. Process Automation Instrumentation, 2007, (7): 10-13.
- [5] 钟汉如. 基于 RS485 的 PC 与 ARM 间文件传输的实现 [J]. 微计算机信息, 2008, (5): 144-146.
- [6] 高 琴, 刘淑聪, 彭宏伟. 步进电机控制系统的设计及应用 [J]. 制造业自动化, 2012, (1): 150-152.
- [7] 邹 岩, 杨志义, 张凯龙. CUDA 并行程序的内存访问优化技术研究 [J]. 计算机测量与控制, 2009, (12): 2504-2506.

(上接第 1452 页)

- [8] 钱晓山, 阳春华. 基于 GEP 的最小二乘支持向量机模型参数选择 [J]. 智能系统学报, 2012, 7 (3): 1-5.
- [9] 胡建军, 唐常杰, 段 磊, 等. 基因表达式编程初始种群的多样化策略 [J]. 计算机学报, 2007, 30 (2): 305-310.
- [10] 陈春香, 胡光道, 李悦乔. 适用于函数发现的遗传算法设计开发

- [J]. 计算机应用, 2006, 26 (12): 194-196.
- [11] 胡建军, 唐常杰, 彭 京, 等. 快速跳出局部最优的 VPS-GEP 算法 [J]. 四川大学学报(工程科学版), 2007, 39 (1): 128-133.
- [12] 左 劼. 基因表达式编程核心技术研究 [D]. 成都: 四川大学, 2004.
- [13] 杨 洲, 景 博, 张 劼, 等. 飞机故障预测与健康应用管理模式研究 [J]. 计算机测量与控制, 2011, 19 (9): 2061-2063.