

DOI:10.19651/j.cnki.emt.2107277

信号时序逻辑方法识别电机运行状态研究

覃 姜 肖 兵

(华南理工大学 自动化科学与工程学院 广州 510641)

摘要: 矢量控制是目前永磁同步电机(PMSM)的主流控制方法,针对其计算复杂,依赖电机参数辨识等优点,提出采用信号时序逻辑(STL)的形式化方法对电机运行状态进行识别,从而可以通过控制电机驱动电路的脉宽调制(PWM)实现最大转矩电流比控制(MTPA)。采集与驱动电路直流母线串联的 shunt 电阻在电机同种工况不同运行状态下的电压数据,采用 5 折交叉验证,基于决策树学习 STL 公式,最后使用 STL 公式判断电机运行状态正常,欠调制,或过调制。定义了一阶基元和二阶基元分别作为决策树节点,学习过程使用粒子群优化(PSO),并采用了不同不纯度度量作为优化时的损失函数。实验结果表明,使用一阶基元的 STL 对电机状态识别准确率可达 98.78%,程序运行耗时 0.150 9 s;使用二阶基元识别准确率可达 95.06%,程序运行时间 2.397 9 s,对基于 STL 实现电机控制算法具有重要意义。

关键词: 信号时序逻辑;决策树;电机状态识别;不纯度度量;异常检测

中图分类号: TP391.4;TB663 **文献标识码:** A **国家标准学科分类代码:** 520.20

Research on motor states recognition using signal temporal logic

Qin Jiang Xiao Bing

(School of Automation Science and Engineering, South China University of Technology, Guangzhou 510641, China)

Abstract: Vector control is the current mainstream control method for permanent magnet synchronous motors (PMSM). Aiming at its disadvantages such as complex calculations and dependence on motor parameters identification, a formal method of signal temporal logic (STL) is proposed to identify the running states of the motor, so that the maximum torque per ampere (MTPA) can be realized by controlling the pulse width modulation (PWM) of the motor drive circuit. The voltage data of the shunt resistance in series with the DC bus of the driving circuit under the same working condition and different running states of the motor were collected. The 5-fold cross validation is adopted, and the STL formula is learned based on the decision tree. Finally, the STL formula is used to determine whether the motor is running normal, under modulation or over modulation. The first-level and second-level primitives are defined as the nodes of the decision tree respectively. Particle swarm optimization (PSO) is used in the learning process, and different impurity measures are used as the loss functions. The experimental results show that the accuracy of motor states recognition by STL with first-level primitives can reach 98.78%, and the program takes 0.150 9 s. The recognition accuracy with second-level primitives can reach 95.06% and the program takes 2.397 9 s. It is of great significance to the implementation of motor control algorithms based on STL.

Keywords: signal temporal logic; decision tree; motor states recognition; impurity measure; anomaly detection

0 引 言

永磁同步电机(permanent magnetic synchronous machine, PMSM)的结构与直流电机类似,具有力矩惯量大、结构简单、运行可靠、调速性能好等优点^[1-3]。日前,永磁同步电机的主要控制技术之一是矢量控制,也被称为磁场定向控制,包括转速闭环控制,电流闭环控制和脉宽调制

(pulse width modulation, PWM)控制算法 3 个主要部分,而其中 PWM 又占据着主导作用^[4]。所以,很多永磁同步电机控制最终是控制驱动电路的 PWM, PWM 占空比是否合适对于电机能否以最佳状态运行具有重要意义^[5]。

信号时序逻辑^[6](signal temporal logic, STL)是一类形式化语言,能够描述动态系统的时间约束与参数变化范围约束,从而表达系统的某些属性,例如“如果一艘船在 A 区

域,并保持不低于 10 km/h 的速度继续行驶 10 min,它将确保能在 15 min 内到达河岸”。STL 具有严格的数学定义,其句式生成的本质是混杂系统全局优化^[7],基于数据生成新的句式,表达数据中蕴含的逻辑规律,同时优化其时间参数与物理参数^[8],描述连续系统的某些属性或行为。目前在控制领域,Tiger 等^[9]利用 STL 对机器人、无人机的运动路径进行规划与控制,将无人机的高度建模为一个概率分布而不是固定的范围,增加了其安全性;Jones 等^[10]提出一种与支持向量机(support vector machine, SVM)类似的算法,该算法用 STL 描述系统的已知正常行为,从而对列车制动装置等信息-物理系统(cyber-physical systems, CPS)的异常行为进行识别;Lindemann 等^[11]采用 STL 进行模型预测控制的优化计算,通过定义离散平均空间鲁棒性(discrete average space robustness, DASR)来提高离散信号的鲁棒性。在国内,舒新峰等^[12]基于 STL 提出一种命题投影时序逻辑的分布式模型检验方法,以缓解模型检验中的状态空间爆炸等问题;谢健等^[13]利用状态事件线性时序逻辑对状态事件系统进行规约,能够被应用于状态事件系统,尤其是与安全有关的嵌入式系统的模型验证中;聂凯等^[14]基于时序逻辑对几种常见的网络攻击模式建立 STL 公式,从而能有效地检测许多复杂的网络攻击。

电机控制一般基于 32 位微控制器(MCU),传统的矢量控制不能保证电机在不同工况、突加负载等情况下依然保持最佳状态运行。根据 PWM 占空比过大、合适、过小,驱动电路电压波形有差异的现象,本文利用电机运行时实时采集到的电机驱动电路中与交流母线串联的 shunt 电压波形,采用不同类型的损失函数和 STL 基元(primitives),基于决策树方法学习 STL 表达式,从而利用 STL 公式度量这种差异,描述电机运行在正常状态、欠调制状态或过调制状态下的 shunt 电压特征,识别电机运行状态,并验证了不同损失函数和基元对于 STL 识别电机运行状态的影响。STL 通过 shunt 电压波形对电机运行状态的识别结果,可以作为反馈量通过 MCU 调节电机 PWM,从而使电机运行于最佳状态。

1 信号时序逻辑基本理论

信号时序逻辑(STL)作为一类形式化语言,可以通过描述系统时间和物理参数约束,从而表达系统某些属性和行为。由于 STL 在语法和语义上非常接近自然语言,因此具有很强的可读性与可解释性。STL 的具体语法定义如下:

$$\phi ::= g \mid \neg \phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid F_{[a,b)} \phi \mid G_{[a,b)} \phi \quad (1)$$

其中, g 是一个术语,描述系统的某种属性或行为,可映射为真(true)或假(false),若为真则系统符合该 STL 语句表达的含义,反之为假。 \neg, \wedge, \vee 分别是布尔运算中的非,与,或运算。 F 和 G 分别是时间运算符“最终”(finally)

和“总是”(globally), a 和 b 都是时间参数。 ϕ 表示一个线性谓语,其形式为 $(y_s \sim \pi)$, 其中 y_s 是系统的某个物理参数, π 为一个标量,式中的 $\sim \in \{\leq, >\}$ 。用自然语言描述, $F[a,b)\phi$ 表示“在时间区间 $[a,b)$ 中, ϕ 描述的系统行为至少会发生一次”;而 $G[a,b)\phi$ 表示“ ϕ 描述的系统属性在时间区间 $[a,b)$ 内总是为真”。用 $s[t] \mid = \phi$ 表示信号 s 从 t 时刻满足 STL 表达式 ϕ , 则式(1)中各子式的形式化定义可递归定义如下:

$$s[t] \mid = (y_s \sim \pi) \text{ iff } y_s(t) \sim \pi \quad (2)$$

$$s[t] \mid = \phi_1 \wedge \phi_2 \text{ iff } s[t] \mid = \phi_1 \text{ and } s[t] \mid = \phi_2 \quad (3)$$

$$s[t] \mid = \phi_1 \vee \phi_2 \text{ iff } s[t] \mid = \phi_1 \text{ or } s[t] \mid = \phi_2 \quad (4)$$

$$s[t] \mid = G_{[\tau_1, \tau_2)}(y_s \sim \pi) \text{ iff } y_s(t') \sim \pi, \forall t' \in [t + \tau_1, t + \tau_2) \quad (5)$$

$$s[t] \mid = F_{[\tau_1, \tau_2)}(y_s \sim \pi) \text{ iff } \exists t' \in [t + \tau_1, t + \tau_2), \text{ s. t. } y_s(t') \sim \pi \quad (6)$$

信号 s 对于 STL 公式 ϕ 的满足程度可用 s 关于 ϕ 的鲁棒程度 $r(s, \phi, t)$ 衡量^[15]。鲁棒程度按照以下递归的定量语义进行计算:

$$r(s, (y_s \geq c_1), t) = y_s(t) - c_1 \quad (7)$$

$$r(s, (y_s < c_1), t) = c_1 - y_s(t) \quad (8)$$

$$r(s, \phi_1 \wedge \phi_2, t) = \min(r(s, \phi_1, t), r(s, \phi_2, t)) \quad (9)$$

$$r(s, \phi_1 \vee \phi_2, t) = \max(r(s, \phi_1, t), r(s, \phi_2, t)) \quad (10)$$

$$r(s, F_{[\tau_1, \tau_2)} \phi, t) = \min_{t' \in [t + \tau_1, t + \tau_2)} r(s, \phi, t') \quad (11)$$

$$r(s, G_{[\tau_1, \tau_2)} \phi, t) = \max_{t' \in [t + \tau_1, t + \tau_2)} r(s, \phi, t') \quad (12)$$

其中, c_i 为实数。 $r(s, \phi, t)$ 表示从 t 时刻开始信号 s 对于 STL 公式 ϕ 的鲁棒程度。本文直接使用 $r(s, \phi)$ 来表示 $r(s, \phi, 0)$ 。当 $r(s, \phi)$ 为正时,表示信号 s 满足 STL 公式 ϕ 的描述;当 $r(s, \phi)$ 为负时则表示 s 不满足 ϕ 。 $r(s, \phi)$ 的幅值代表了信号 s 对于 ϕ 的满足程度或背离程度。

STL 可以描述许多重要的系统属性,例如:

1) 系统物理参数有限时间范围内的不变性。STL 公式 $F[0, \tau_1)(G[\tau_2, \tau_3)(y_s < \pi))$ 表示存在时间 $t \in [0, \tau_1)$, 使得在 $[t + \tau_2, t + \tau_3)$ 内 y_s 总是小于 π 。

(2) 系统状态空间中多个区域的可达性。STL 公式 $F[0, \tau_1)(F[\tau_2, \tau_3)(y_s \geq \pi_1) \vee F[\tau_2, \tau_3)(y_s < \pi_2))$ 表示存在时间 $t \in [0, \tau_1)$, 使得在 $[t + \tau_2, t + \tau_3)$ 内 y_s 最终会 $\geq \pi_1$ 或 $< \pi_2$ 。

文献[16]采用 STL 监测港口进出船舶的航行轨迹是否正常,如图 1 所示,正常的航行轨迹用实线标识,虚线为异常轨迹。异常轨迹包含两种,一种是船只航行偏离到了岛屿上,意味着可能存在非法交易的情况;另一种船先靠近了港口或其他船只,游荡一阵后迅速返回了公海区域,预示可能存在恐怖活动或者间谍活动的风险。式(13)所示的 STL 公式将正常航迹与异常航迹进行了有效区分:

$$\phi_N = F_{[0, 320)}(G_{[28, 227)}(y > 3.5) \wedge (G_{[308, 313)}(x < 3.4) \quad (13)$$

式(13)的 STL 公式可用自然语言解释为“若存在 $t \in$

$[0, 320)$, 使得在 $[t + 28, t + 227)$ 时间内 y 总是大于 3.5 km , 并且在 $[t + 308, t + 313)$ 时间内 x 总是小于 3.4 km 则该船只航行轨迹正常, 否则为异常船只”。

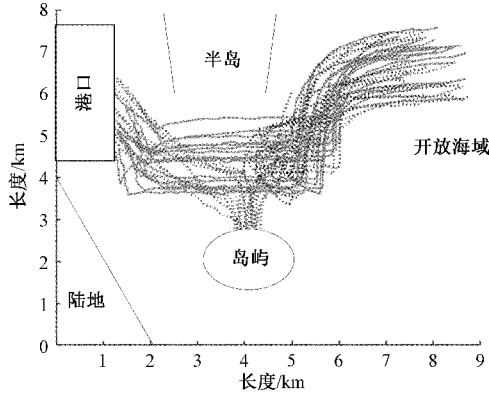


图 1 STL 监测港口船舶航行轨迹

2 基于决策树构建信号时序逻辑公式

2.1 决策树学习算法

一般的 STL 公式生成与参数优化往往伴随大规模的全局非凸寻优过程, 本文通过二元决策树构造 STL 公式, 在决策树的每个节点需要优化的参数数量很少且是固定的, 从而能够极大提高算法执行速度。

决策树可以看作一个关于数据的树状问题序列, 当信号数据流经决策树之后就能对该数据类型做出预测^[17]。在决策树中我们定义: 起始节点为根节点; 一个节点的深度为从根节点到该节点的路径长度; 节点的父节点为比其深度小 1 的邻节点; 节点的子节点为比其深度大 1 的邻节点。没有子节点的节点称为叶节点, 其他节点称为非终端(non-terminal)节点。

构造决策树的伪代码如算法 1 所示, 该算法包含 3 个可以调整的超参数, 即 STL 基元集合 \mathbf{P} , 损失函数类型 J 和停止条件 $stop$ 。算法 1 采用递归形式, 将当前路径节点的公式 ϕ^{path} , 到达当前节点的数据 \mathbf{S} 和当前节点深度 h 作为输入。

在算法 1 中, 首先检查是否满足算法停止的条件, 包括树的深度、当前节点的信号数和当前节点的不纯度度量 (impurity measure)。如果条件满足, 则返回叶节点, 其标签 c 通过 2.4 节定义的 $p(\mathbf{S}, c; \phi^{path})$ 取得的最好分类效果来确定。若条件不满足, 算法继续从 2.3 节定义的 STL 基元中找出最优的 STL 公式及其参数, 优化中使用的损失函数是不纯度度量 J , 它用于度量学习的 STL 公式对于数据的划分效果。 $partition(\mathbf{S}, \phi)$ 将信号 \mathbf{S} 分为满足 STL 公式 ϕ 的信号 \mathbf{S}_+ 和不满足公式 ϕ 的信号 \mathbf{S}_- , 即 $\mathbf{S}_+, \mathbf{S}_- = partition(\mathbf{S}, \phi)$:

$$\mathbf{S}_+ = \{(s^i, l^i) \in \mathbf{S} \mid s^i = \phi\} \quad (14)$$

$$\mathbf{S}_- = \{(s^i, l^i) \in \mathbf{S} \mid s^i \neq \phi\} \quad (15)$$

算法 1: 决策树构造 $BuildTree(\cdot)$

参数: \mathbf{P} - STL 基元集合

参数: J - 作为损失函数类型的不纯度度量

参数: $stop$ - 算法停止准则

输入: ϕ^{path} - 当前决策树路径表示的 STL 公式

输入: $\mathbf{S} = \{(s^i, l^i)\}_{i=1}^N$ - 带类型标签的信号数据集

输入: h - 当前决策树深度

输出: t - 等价于一个 STL 公式的决策树

1: if $stop(\phi^{path}, h, \mathbf{S})$ then

2: $t \leftarrow leaf(\arg \max_{c \in \mathcal{C}} \{p(\mathbf{S}, c; \phi^{path})\})$

3: return t

4: $\phi^* = \arg \max_{\phi \in \mathbf{P}, \theta \in \Theta} J(\mathbf{S}, partition(\mathbf{S}, \phi \wedge \phi^{path}))$

5: $t \leftarrow non_terminal(\phi^*)$

6: $\mathbf{S}_+, \mathbf{S}_- \leftarrow partition(\mathbf{S}, \phi^{path} \wedge \phi^*)$

7: $t.left \leftarrow buildTree(\phi^{path} \wedge \phi^*, \mathbf{S}_+, h + 1)$

8: $t.right \leftarrow buildTree(\phi^{path} \wedge \neg \phi^*, \mathbf{S}_-, h + 1)$

9: return t

2.2 决策树与 STL 公式的转换

通过算法 1 得到的决策树可直接用于对信号的分类, 该决策树等价于一个 STL 公式。图 2 是决策树的一个示例, 其中 ϕ_i 表示一个 STL 基元, \mathbf{S}_p 为满足该决策树表示的 STL 公式的信号, \mathbf{S}_n 为不满足的信号。在树的每一个节点, 该节点及其所有子节点所等价的 STL 公式通过以下 3 个步骤计算: 1) 将该节点公式与其左子树所表示的公式逻辑与; 2) 将该节点逻辑非, 然后跟其右子树表示的公式逻辑与; 3) 将步骤 1) 和 2) 进行逻辑或运算。要得到最终的 STL 公式, 需要从根节点开始递归执行上面的过程。图 2 的决策树等价的 STL 公式如式 (16) 所示。

$$\phi = (\phi_1 \wedge \phi_2 \wedge \phi_3) \vee (\neg \phi_1 \wedge ((\phi_4 \wedge \phi_5) \vee (\neg \phi_4 \wedge \phi_6))) \quad (16)$$

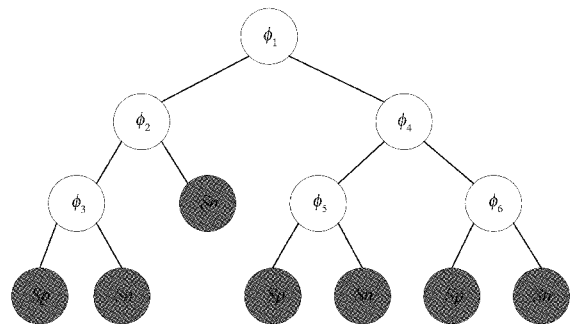


图 2 决策树等价于一个 STL 公式

2.3 STL 公式基元

为了在决策树每个节点进行数据分类, 通常会考虑一个有限的可能分割规则表, 本文将 STL 基元作为在每个节点对于数据的分类规则。STL 基元也叫原子命题, 即组成 STL 公式的最小单元, 这些单元的逻辑组合构成最终的

STL 公式。在本文中定义两类 STL 基元：

1) 一阶基元。令 \mathbf{S} 表示值为 n 维实数 \mathbf{R}^n 的信号数据集，其中 $n \geq 1$ ，定义以下一阶基元集合：

$$\mathbf{P}_1 = \{F_{[\tau_1, \tau_2]}(x_i \sim \mu) \text{ or } G_{[\tau_1, \tau_2]}(x_i \sim \mu) \mid i \in \{1, \dots, n\}, \sim \in \{\leq, >\}\} \quad (17)$$

\mathbf{P}_1 的参数包括 (μ, τ_1, τ_2) ，其参数空间为 $\Theta_1 = \mathbf{R} \times \{(a, b) \mid a < b, a, b \in \mathbf{R}_{\geq 0}\}$ ，其中 \mathbf{R} 为一维实数空间。

2) 二阶基元。令 \mathbf{S} 表示值为 n 维实数 \mathbf{R}^n 的信号数据集，其中 $n \geq 1$ ，定义以下二阶基元集合：

$$\mathbf{P}_2 = \{G_{[\tau_1, \tau_2]}F_{[0, \tau_3]}(x_i \sim \mu) \text{ or } F_{[\tau_1, \tau_2]}G_{[0, \tau_3]}(x_i \sim \mu) \mid i \in \{1, \dots, n\}, \sim \in \{\leq, >\}\} \quad (18)$$

\mathbf{P}_2 的参数包括 $(\mu, \tau_1, \tau_2, \tau_3)$ ，其参数空间为 $\Theta_2 = \mathbf{R} \times \{(a, b) \mid a < b, a, b \in \mathbf{R}_{\geq 0}\} \times \mathbf{R}_{\geq 0}$ ，其中 \mathbf{R} 为一维实数空间。

一阶基元的含义较为直观， $F_{[\tau_1, \tau_2]}(x_i \sim \mu)$ 表示在时间区间 $[\tau_1, \tau_2]$ 内述语 $(x_i \sim \mu)$ 必须在至少一个时间点为真，即 $(x_i \sim \mu)$ 所描述的系统属性或行为至少会发生一次； $G_{[\tau_1, \tau_2]}(x_i \sim \mu)$ 表示在其中所有时间点都为真。类似地，二阶基元可用以下自然语言解释：1) $F_{[\tau_1, \tau_2]}G_{[0, \tau_3]}(x_i \sim \mu)$ 意为“述语 $(x_i \sim \mu)$ 必须在起始时间位于 $[\tau_1, \tau_2]$ ，持续时间为 τ_3 中总是成立”；2) $G_{[\tau_1, \tau_2]}F_{[0, \tau_3]}(x_i \sim \mu)$ 表示“在区间 $[\tau_1, \tau_2]$ 的所有时间点中，述语 $(x_i \sim \mu)$ 必须在 τ_3 时间内为真”。一阶基元和二阶基元可以被认为具有不同程度灵活性的有界可达性和安全性的规范表达式。值得注意的是，STL 公式的基元可以根据不同研究对象、不同的数据进行自定义，使其更适用于所研究的具体问题。

2.4 不纯度度量

在定义了一系列基元集 \mathbf{P} 对数据进行分类之后，需要一个标准来选择哪个基元及其参数对于数据的分类效果最好。我们希望在决策树的每个节点通过 STL 基元分类之后的数据能够尽可能地属于同一种类型，即其中的“杂质”最少。不纯度度量 (impurity measure) 便是衡量数据同质性的常用标准。本节首先定义经典的不纯度度量，再定义加入鲁棒程度作为计算因子之后的扩展不纯度度量。

令 \mathbf{S} 为一有限信号数据集， ϕ 为一个 STL 公式，且 $\mathbf{S}_\top, \mathbf{S}_\perp = \text{partition}(\mathbf{S}, \phi)$ 。下面的区分权重 (partition weights) 用于描述信号 s^i 基于其标签 l^i 和 STL 公式 ϕ 的分布情况：

$$p_- = \frac{|\mathbf{S}_\perp|}{|\mathbf{S}|} \quad (19)$$

$$p_+ = \frac{|\mathbf{S}_\top|}{|\mathbf{S}|} \quad (20)$$

$$p(\mathbf{S}, c; \phi) = \frac{|\{(s^i, l^i) \mid l^i = c\}|}{|\mathbf{S}|} \quad (21)$$

其中， \top 与 \perp 分别是 true 与 false 的简写， p_+ 和 p_- 分别代表信号 \mathbf{S} 中表示为 \mathbf{S}_\top 和 \mathbf{S}_\perp 的所占的比例， $p(\mathbf{S}, c; \phi)$ 代表信号 \mathbf{S} 中类型为 c 的信号所占的比例。不纯度度量

的经典定义^[17]如下：

1) 信息增益 (information gain, IG)

$$IG(\mathbf{S}, \{\mathbf{S}_\top, \mathbf{S}_\perp\}) = H(\mathbf{S}) - \sum_{\otimes \in \{\top, \perp\}} p_\otimes \cdot H(\mathbf{S}_\otimes) \quad (22)$$

$$H(\mathbf{S}) = - \sum_{c \in \mathbf{C}} p(\mathbf{S}, c; \phi) \log p(\mathbf{S}, c; \phi) \quad (23)$$

2) 基尼增益 (gini gain, GG)

$$GG(\mathbf{S}, \{\mathbf{S}_\top, \mathbf{S}_\perp\}) = Gini(\mathbf{S}) - \sum_{\otimes \in \{\top, \perp\}} p_\otimes \cdot Gini(\mathbf{S}_\otimes) \quad (24)$$

$$Gini(\mathbf{S}) = \sum_{c \in \mathbf{C}} p(\mathbf{S}, c; \phi) (1 - p(\mathbf{S}, c; \phi)) \quad (25)$$

3) 误分类增益 (misclassification gain, MG)

$$MG(\mathbf{S}, \{\mathbf{S}_\top, \mathbf{S}_\perp\}) = MR(\mathbf{S}) - \sum_{\otimes \in \{\top, \perp\}} p_\otimes \cdot MR(\mathbf{S}_\otimes) \quad (26)$$

$$MR(\mathbf{S}) = \min(p(\mathbf{S}, C_p; \phi), p(\mathbf{S}, C_n; \phi)) \quad (27)$$

如第 1 节所述，鲁棒程度可用于在对 STL 公式进行参数优化时评价一个 STL 公式对于信号的分类结果，我们考虑待分类信号鲁棒程度，将区分权重用以下式子定义，从而得到将鲁棒程度作为计算因子的扩展不纯度度量。

$$p_\top = \frac{\sum_{s^i \in \mathbf{S}_\top} r(s^i, \phi)}{\sum_{s^i \in \mathbf{S}} |r(s^i, \phi)|} \quad (28)$$

$$p_\perp = - \frac{\sum_{s^i \in \mathbf{S}_\perp} r(s^i, \phi)}{\sum_{s^i \in \mathbf{S}} |r(s^i, \phi)|} \quad (29)$$

$$p(\mathbf{S}, c; \phi) = \frac{\sum_{s^i \in \mathbf{S}_c} |r(s^i, \phi)|}{\sum_{s^i \in \mathbf{S}} |r(s^i, \phi)|} \quad (30)$$

其中， $\mathbf{S}_c = \{s^i \in \mathbf{S} \mid l^i = c\}$ 。将式(28)~(30)代入不纯度度量的定义中，即可得到扩展后的不纯度度量定义。为了与经典定义相区分，用脚标 r 表示加入鲁棒程度的扩展不纯度度量，例如 IGr 。

3 STL 识别电机运行状态实验方法

在工程实践中，最为简便的电机驱动电路测量方案是仅用一个采样电阻 (shunt) 串联在直流母线负端，如图 3 中的电阻 R_s 。以文献[18]为代表的许多论文探讨如何由 shunt 信号波形 V_s 重构三相电流波形，从而实现矢量控制。将 V_s 看作一个时序信号，将电机控制问题转化为时序信号模式识别与随机决策问题。本文基于图 3 的电机控制框图，验证基于信号时序逻辑方法的电机运行状态识别的有效性。

本文使用的实验测试平台如图 4 所示，该平台为车载变频空调涡旋压缩机综合测试台，由电源给三相桥式全控整流电路和控制电路供电，同时控制电路负责三相全桥的

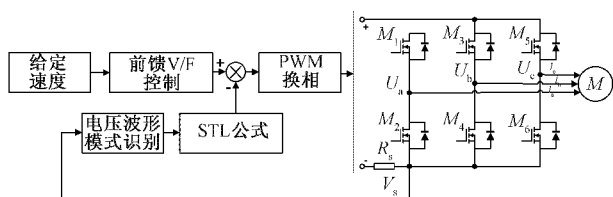


图 3 基于 STL 控制电机驱动电路

PWM 换相。给定不同作用量,使电机分别工作在正常状态、欠调制状态和过调制状态下,通过与直流母线串联的 shunt 采样电阻的电压波形和电机相电压波形,传递到 PC 机上,再在 PC 机上采用 STL 方法验证对 shunt 电压波形模式识别的效果,判断电机的工作状态,并与其真实值作比较。

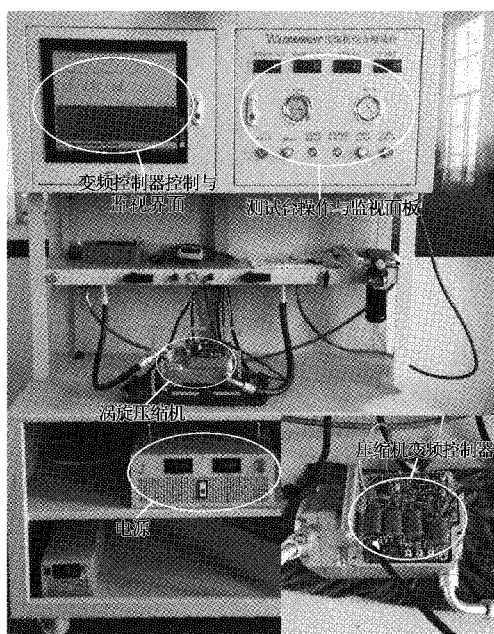


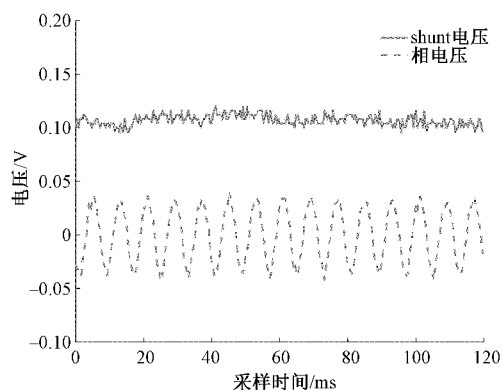
图 4 实验测试平台

4 实验结果与分析

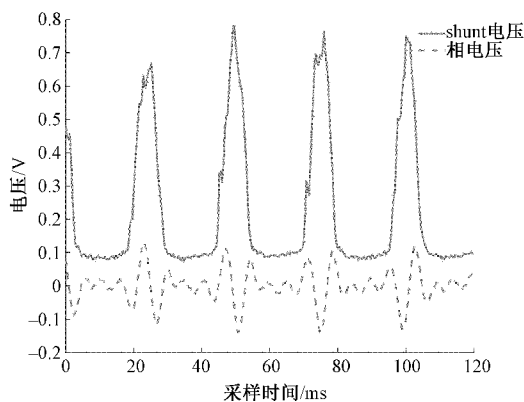
4.1 电机不同运行状态下电压波形

在同一工况下,通过控制器给定电机不同作用量,分别使电机工作在正常状态、欠调制状态和过调制状态,使用 Fluke 示波器采集到的典型 shunt 电压和电机 B 相电压波形如图 5 所示,采样频率为 2.5 kHz。

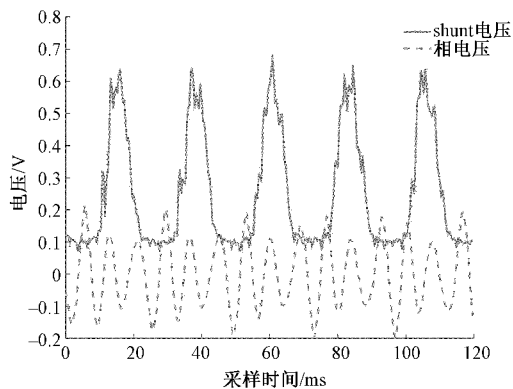
根据图 5,当电机工作在正常状态时,shunt 电压在小范围内震荡,比较符合理想的直流电压,电机相电压也接近正弦波形;而当电机欠调制或过调制时,shunt 电压波动幅值较大,表现出一定的周期性,电机相电压也不再是标准的正弦波形,发生了一定程度的畸变。在实际控制过程中,电机相电压对于控制电路是无法测量的,所以我们通过 shunt 电压来对电机运行状态进行识别。由于电机工作正常状态与异常状态可以通过一定时间区间内 shunt 电压的平均幅



(a) 正常状态



(b) 欠调制状态



(c) 过调制状态

图 5 电机不同运行状态下 shunt 电压与相电压波形

值、变化范围进行确定,所以本文主要通过 STL 判断异常状态中电机是欠调制运行还是过调制运行。应当指出,在一般工作情况下,电机不同程度的欠调制或过调制状态对于 shunt 电压波形并无明显影响,STL 公式仍能对其进行正确区分。

4.2 电压波形分割

由图 5 可知,电机欠调制运行与过调制运行时的 shunt 电压波形呈现出周期性,要从波形数据分辨电机运行状态,首先需要将单个周期波形分割出来。

滑动窗口十分适用于只要求对当前某时间段内的数据

进行处理的应用^[19],采用滑动窗口方法分割电压波形。首先将 shunt 电压信号进行离散傅里叶变换,对于长度为 n 的波形数据 \mathbf{X} ,其离散傅里叶变换

$$Y(k) = \sum_{j=1}^n X(j)W_n^{(j-1)(k-1)} \quad (31)$$

其中, $W_n = e^{(-2\pi i)/n}$ 为 n 次单位根之一。在频谱中找到幅值最大点对应的频率,其倒数即为 shunt 电压周期,结果约为 25 ms。使用对应大小的窗口在 shunt 电压波形中滑动,分割出单个周期波形如图 6 所示。

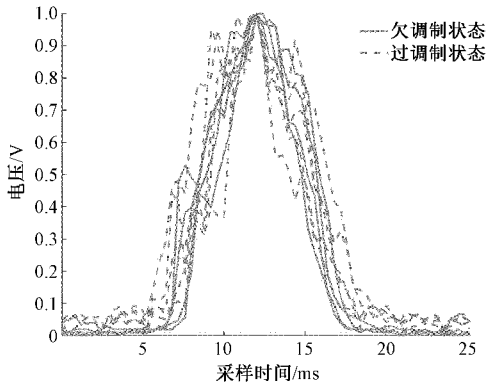


图 6 波形分割后电机不同状态 shunt 电压

4.3 STL 公式电机运行状态识别

本文采集了电机运行于同一工况下的欠调制与过调制状态的 shunt 电压波形,并将单个周期分割的共 142 组数据,首先对样本数据进行极差归一化

$$x_{out} = (x_{in} - x_{min}) / (x_{max} - x_{min}) \quad (32)$$

使数据范围归一化到 $[0, 1]$ 之间,再根据算法 1 构造用于判断电机运行状态的决策树。

在构造决策树时,采用 5 折交叉验证,设置训练样本分数为 0.8,将 shunt 电压波形数据分为训练样本和测试样本,并设置 2.3 节定义的一阶基元和二阶基元分别进行实验,式(29)与(30)分别为使用一阶、二阶基元学习到的 STL 公式的例子。

$$(G_{[13.2, 17.8]} x_1 > 0.0733 \wedge G_{[8.64, 9.8]} x_1 < 0.6 \wedge F_{[16.5, 25.2]} x_1 > 0.395) \vee (G_{[13.2, 17.8]} x_1 > 0.0733 \wedge F_{[8.64, 9.8]} x_1 > 0.6) \quad (33)$$

$$(G_{[0.925, 13.4]} F_{[0.8, 69]} x_1 < 0.315) \vee (F_{[0.925, 13.4]} G_{[0.8, 69]} x_1 > 0.315 \wedge G_{[2.61, 5.52]} F_{[0.6, 35]} x_1 < 0.0965) \quad (34)$$

在使用不同基元构造决策树的过程中,分别采用经典的信息增益、误分类增益和它们加入鲁棒程度后的扩展版本作为优化基元参数时的损失函数。对于基元参数的优化问题是一个非凸的全局优化,而粒子群优化 (particle swarm optimization, PSO) 具有良好的全局寻优能力^[20],所以本文优化算法选择粒子群算法。将算法 1 在拥有 2.6 GHz 处理器和 8 GB RAM 的计算机上通过 MATLAB 实现,分别测试了上述不同实验设置下的状态识别平均正确率、正确率标准差、平均用时和用时标准差,结果如表 1、2

所示,其中损失函数类型带有脚标 r 的为扩展的不纯度度量。

表 1 使用一阶基元识别电机状态结果

损失函数	平均正确率/%	标准差/%	平均用时/s	用时标准差/s
IG	85.06	6.16	0.344 0	0.059 6
MG	96.73	6.48	0.203 8	0.069 4
IGr	87.78	15.32	0.542 9	0.085 9
MGr	98.78	4.97	0.150 9	0.021 9

表 2 使用二阶基元识别电机状态结果

损失函数	平均正确率/%	标准差/%	平均用时/s	用时标准差/s
IG	87.56	9.85	4.739 8	0.931 9
MG	95.06	10.11	2.397 9	0.461 8
IGr	92.78	6.62	8.616 1	1.141 8
MGr	90.28	5.47	2.737 7	0.545 4

从表 1 可以看出,使用一阶基元作为节点构造决策树时,参数优化过程中损失函数为误分类增益的平均正确率明显高于信息增益,经典误分类增益与扩展误分类增益的 5 折交叉验证的平均正确率分别为 96.73% 与 98.78%,正确率标准差分别为 6.48% 与 4.97%,正确率波动不大;而使用经典信息增益和扩展信息增益的电机状态识别效果并不理想,正确率都在 90% 以下。在算法用时上,一阶基元学习训练与测试的平均用时极短,大多在 0.4 s 以下,用时标准差也小于 0.1 s。其中平均用时最短的是扩展误分类增益,仅为 0.150 9 s。以上结果表明该算法在计算资源宝贵的 MCU 上也可以取得良好的效果。

二阶基元由于需要优化的参数更多,计算也更复杂,所以算法运行时间有了相当的提升,如表 2 所示,经典误分类增益与扩展误分类增益的平均用时分别为 2.397 9 和 2.737 7 s,而信息增益分别达到了 4.739 8 和 8.616 1 s,用时标准差都较小。在正确率方面,对电机运行状态识别最好的是经典的误分类增益,5 折交叉验证的平均正确率为 95.06%,但其正确率标准差达到了 10.11%,在每一次的实验中正确率波动较为明显;其次是加入鲁棒程度的扩展信息增益,正确率为 92.78%。相比于二阶基元,一阶基元的 STL 公式在对电机运行状态识别中正确率更高,算法用时也更少。

5 结 论

本文对信号时序逻辑 (STL) 理论进行了分析,利用该理论实现了基于决策树的 STL 公式构造算法,并将该理论与算法应用于电机运行状态的识别中,用一阶基元、二阶基元作为决策树优化节点,并使用不同的不纯度度量作为参

数优化算法中的损失函数分别进行了实验,验证了方法的有效性。结果表明,一阶基元构造 STL 公式对于电机运行状态的识别正确率更高,算法运行用时也更短,其中使用扩展误分类增益作为损失函数时识别平均正确率最高,为 98.78%,算法平均用时为 0.1509 s,证实了该方法用于电机运行状态识别的有效性和高效性,对进一步将 STL 方法在 MCU 上实现,从而实现基于 STL 的电机控制算法的工程应用具有重要意义。

参考文献

- [1] HE C, WU T. Analysis and design of surface permanent magnet synchronous motor and generator [J]. *Ces Transactions on Electrical Machines and Systems*, 2019, 3(1): 94-100.
- [2] HU W, RUAN C, NIAN H, et al. Simplified modulation scheme for open-end winding PMSM system with common DC bus under open-phase fault based on circulating current suppression [J]. *IEEE Transactions on Power Electronics*, 2019, 35(1): 10-14.
- [3] 王帅, 郎宝华. 基于滑模观测器的 PMSM 转子位置估计[J]. *国外电子测量技术*, 2020, 39(11): 34-38.
- [4] 刘宇博. 基于滑模观测器的永磁同步电机控制系统关键技术研究[D]. 哈尔滨: 哈尔滨理工大学, 2020.
- [5] 巫春玲, 程琰清, 刘智轩, 等. 一种永磁同步电机系统效率优化控制策略研究[J]. *电子测量技术*, 2020, 43(10): 36-41.
- [6] ASARIN E, DONZÉ A, MALER O, et al. Parametric identification of temporal properties [C]. *International Conference on Runtime Verification*, 2011: 147-160.
- [7] LAKHDHAR W, MZID R, KHALGUI M, et al. Multiobjective optimization approach for a portable development of reconfigurable real-time systems: From specification to implementation [J]. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2018: 1-15.
- [8] BOMBARA G, BELTA C. Online learning of temporal logic formulae for signal classification [C]. *2018 European Control Conference (ECC)*, 2018: 2057-2062.
- [9] TIGER M, HEINTZ F. Incremental reasoning in probabilistic signal temporal logic [J]. *International Journal of Approximate Reasoning*, 2020, 119: 325-352.
- [10] JONES A, KONG Z, BELTA C. Anomaly detection in cyber-physical systems: A formal methods approach [C]. *53rd IEEE Conference on Decision and Control*, 2014: 848-853.
- [11] LINDEMANN L, DIMAROGONAS D V. Robust control for signal temporal logic specifications using discrete average space robustness [J]. *Automatica*, 2019, 101: 377-387.
- [12] 舒新峰, 王昌太, 王燕, 等. 一种命题投影时序逻辑的分布式模型检测方法[J]. *西安电子科技大学学报*, 2020, 47(4): 39-47.
- [13] 谢健, 阚双龙, 黄志球, 等. 嵌入偏序约简的状态事件线性时序逻辑验证[J]. *计算机学报*, 2019, 42(10): 2145-2159.
- [14] 聂凯, 周清雷, 朱维军, 等. 基于时序逻辑的 3 种网络攻击建模[J]. *计算机科学*, 2018, 45(2): 209-214.
- [15] DONZÉ A, MALER O. Robust satisfaction of temporal logic over real-valued signals [C]. *International Conference on Formal Modeling and Analysis of Timed Systems*, 2010: 92-106.
- [16] KONG Z, JONES A, BELTA C. Temporal logics for learning and detection of anomalous behavior [J]. *IEEE Transactions on Automatic Control*, 2016, 62(3): 1210-1222.
- [17] SONG Y Y, YING L. Decision tree methods: Applications for classification and prediction [J]. *Shanghai archives of psychiatry*, 2015, 27(2): 130.
- [18] LU J, HU Y, LIU J. Analysis and compensation of sampling errors in TPFS IPMSM drives with single current sensor [J]. *IEEE Transactions on Industrial Electronics*, 2018, 66(5): 3852-3855.
- [19] 付强, 景博, 何鹏举, 等. 动态信号源数目盲估计方法研究[J]. *仪器仪表学报*, 2020, 41(4): 119-128.
- [20] 尚雪梅, 徐远纲. PSO 优化的最大峭度熵反褶积齿轮箱故障诊断[J]. *电子测量与仪器学报*, 2020, 34(7): 64-72.

作者简介

覃姜, 硕士研究生, 主要研究方向为模式识别与智能系统。

E-mail: 505935310@qq.com

肖兵, 工学博士, 教授, 主要研究方向为新能源汽车电子控制系统、模式识别与智能控制、传感技术与智能检测技术等。

E-mail: aubxiao@scut.edu.cn