

文章编号: 1001-0920(2009)02-0191-05

# 传感器网络中自顶向下的轮廓监控算法

信俊昌, 王国仁

(东北大学 信息科学与工程学院, 沈阳 110004)

**摘要:** 根据无线传感器网络的特殊性质, 提出了自顶向下的轮廓监控算法. 利用为节点分配的超立方, 可以有效地对轮廓监控过程中的数据进行过滤, 从而高效地完成轮廓的维护工作. 仿真实验结果表明, 该算法能有效地避免无用的数据传输, 减少节点的能量消耗, 从而延长传感器网络的使用寿命.

**关键词:** 无线传感器网络; 轮廓监控; 过滤; 能量有效性

**中图分类号:** TP393

**文献标识码:** A

## Top-down skyline monitor algorithm for wireless sensor networks

XIN Jurchang, WANG Guoren

(College of Information Science and Engineering, Northeastern University, Shenyang 110004, China. Correspondent: XIN Jurchang, E-mail: xinjunchang@ise.neu.edu.cn)

**Abstract:** According to the special properties of wireless sensor network, a top-down skyline monitor algorithm is proposed. Using the hyper-cube assigned for each node, the proposed algorithm efficiently processes data filtering and maintenance in the skyline monitoring. Experimental results show that the algorithm can efficiently suppress unnecessary data transmission and reduce the energy cost, so that the life-span of the network is prolonged.

**Key words:** Wireless sensor network; Skyline monitor; Filter; Energy-efficiency

### 1 引言

无线传感器网络由基站和大量具有感知、计算和无线通信能力的微型传感器节点组成, 传感器节点以自组织的方式构成网络并协作完成数据的采集、处理和传输等工作. 无线传感器网络逐渐得到学术界的重视, 各种研究成果不断涌现<sup>[1,2]</sup>.

传感器网络中的轮廓监控查询始终返回网络中属于轮廓<sup>[3]</sup>的节点, 只注重节点的编号而不注重节点的数值. 设传感器节点集合为  $N$ , 轮廓监控返回所有不被其他节点所支配的节点集合  $S$ . 其中, 节点  $n_i$  支配节点  $n_j$  当且仅当  $n_i$  采集的数据  $r_i$  支配  $n_j$  采集的数据  $r_j$ . 而  $r_i$  支配  $r_j$  当且仅当  $r_i$  在所有的维都不比  $r_j$  差, 并且在至少一维比  $r_j$  好. 这里, 好与差可以是用户定义的任何标准.

轮廓监控在无线传感器网络应用中起着非常重要的作用, 例如, 可以时刻监控网络中低能量、高负载的节点, 即能量和负载二维空间上的轮廓节点, 以便及时地给出应对措施来延长传感器网络的使用寿命.

轮廓往往只占整个数据集的一小部分, 因此采用集中式算法将导致大量无用数据的传输. 利用 TAG<sup>[4]</sup> 中提出的网内计算的解决办法, 可在一定程度上避免这种情况的发生, 但是还有许多无用的数据依然在网络中传输.

本文提出了自顶向下的轮廓监控算法 (TDSM), 可有效地节约监控轮廓时传感器网络中的能量消耗. 实验表明, TDSM 可以有效地减少轮廓监控过程中传感器网络内的数据传输量, 进而延长传感器网络的使用寿命.

### 2 相关工作

TAG<sup>[4]</sup> 和 COUGAR<sup>[5]</sup> 是传感器网络查询处理的两个典型系统, 都利用网内聚合的方式来减少计算 Min, Max 等聚合查询时的数据传输量, 并且提供一个类 SQL 的接口来方便用户的操作.

近年来, 无线传感器网络中轮廓查询的研究逐渐得到了重视. Chen 等<sup>[6]</sup> 提出了基于层次结构路由的过滤算法 MINMAX 来计算网络中的轮廓. Xin

收稿日期: 2008-03-15; 修回日期: 2008-06-02.

基金项目: 国家自然科学基金项目 (60773221, 60773219); 国家高技术研究发展计划项目 (2006AA09Z139); 高等学校科技创新工程重大项目培育资金项目 (706016).

作者简介: 信俊昌 (1977—), 男, 辽宁辽阳人, 助教, 博士生, 从事传感器网络中数据管理技术的研究; 王国仁 (1966—), 男, 湖北崇阳人, 教授, 博士生导师, 从事数据库理论和技术等研究.

等<sup>[7,8]</sup> 提出并研究了传感器网络中的滑动窗口轮廓查询. 其中, 文献[7] 利用元组和格两种过滤器来减少数据传输量; 而[8] 则利用映射的方式将数据映射到整数空间, 以整数空间的轮廓作为过滤器来减少数据的传输量. 与[7] 相比, [8] 中使用的是整数而不是实数来表示过滤器, 因而在保证相同过滤效果的情况下, 过滤器维护代价更低.

轮廓监控只考虑哪些节点属于轮廓, 而现有的研究成果都针对感知数据集的轮廓计算, 未考虑轮廓监控的特点, 不适合轮廓监控查询的高效处理.

### 3 自顶向下的轮廓监控算法

#### 3.1 预备知识

不失一般性, 假设本文后面提到的“好”均指“小于”. 首先分析区间以及超立方之间的关系, 它们是过滤原理的基础.

**定义 1** 设两个区间  $a([a.l, a.u])$  和  $b([b.l, b.u])$ . 如果  $a.u < b.l$ , 则  $a$  支配  $b$ , 记为  $a > b$ ; 如果  $a.l < b.l$  且  $a.u > b.l$ , 则  $a$  和  $b$  支配关系不确定, 记为  $a ? b$ .

**定义 2** 设两个超立方  $c$  和  $d$ . 如果  $c$  所有维的区间均支配  $d$  的对应区间, 那么  $c$  支配  $d$ , 记为  $c > d$ ; 如果在某一维  $c$  的区间支配  $d$  的区间, 而在另外一维  $d$  的区间支配  $c$  的区间, 那么两者互相不支配, 记为  $c < > d$ ; 否则两者之间支配关系不确定, 记为  $c ? d$ .

在定义 2 的基础上, 进一步分析超立方关系对数据之间关系的影响.

**引理 1** 设两个超立方  $h_i$  和  $h_j$ , 两个点  $p_i \in h_i$  和  $p_j \in h_j$ . 如果  $h_i > h_j$ , 则  $p_i > p_j$ ; 如果  $h_i < > h_j$ , 那么  $p_i < > p_j$ .

证明略.

**引理 2** 设 3 个超立方  $h_i, h_j$  和  $h_k$ , 相应的点  $p_i \in h_i, p_j \in h_j$  和  $p_k \in h_k$ . 如果  $h_i < > h_j$  并且  $h_j > h_k$ , 那么  $p_k$  肯定不支配  $p_i$ .

**证明** 用反证法. 假设  $p_k > p_i$ . 根据引理 1, 因为  $h_j > h_k$ , 所以  $p_j > p_k$ , 可得  $p_j > p_i$ . 又因为  $h_i < > h_j$ , 所以  $p_i < > p_j$ , 与  $p_j > p_i$  相矛盾, 因此结论成立.

根据上述引理, 可以得到下面的定理. 它们是过滤器设置维护和轮廓监控的基础.

**定理 1** 设  $H = \{h_1, h_2, \dots, h_n\}$  是超立方的集合.  $H_s$  是  $H$  的子集, 如果  $H_s$  中的任意两个超立方  $h_i$  和  $h_j$  满足  $h_i < > h_j$ , 同时对  $H - H_s$  中任何超立方  $h_k, H_s$  中都存在一个  $h_l$  满足  $h_l > h_k$ , 那么  $H_s$  是  $H$  上的轮廓, 记为  $Skyline(H)$ .

**证明** 由已知条件可知,  $H - H_s$  中任何超立

方  $h_k$  都不属于轮廓, 因此  $Skyline(H) \subset H_s$ .

同时, 因为  $H_s$  中的任意两个超立方  $h_i$  和  $h_j$  满足  $h_i < > h_j$ , 所以  $H_s$  中的超立方  $h_i$  都属于轮廓, 因此  $H_s \subset Skyline(H)$ . 故有  $H_s = Skyline(H)$ .

**定理 2** 设  $H = \{h_1, h_2, \dots, h_n\}$  是超立方的集合, 数据集  $P = \{p_1, p_2, \dots, p_n\}$  满足任意  $i, p_i \in h_i$ . 另  $H_s = Skyline(H)$ , 如果超立方  $h_i$  与  $H_s$  中的任何超立方  $h_j$  都满足  $h_i < > h_j$ , 那么  $p_i \in Skyline(P)$ .

**证明** 根据轮廓的定义,  $\forall h_k \in H - H_s$ , 存在  $h_l \in H_s$ , 满足  $h_l > h_k$ . 根据已知,  $h_l$  与  $H_s$  中的任何超立方  $h_j$  都满足  $h_l < > h_j$ . 那么根据引理 2,  $p_k$  肯定不支配  $p_i$ ; 根据引理 1,  $p_j$  肯定不支配  $p_i$ . 由于不存在可以支配  $p_i$  的点, 因此  $p_i \in Skyline(P)$ .

#### 3.2 算法综述

自顶向下的轮廓监控算法 (TDSM) 通过为每个节点分配一个过滤器来阻止无用的数据传输, 同时基站保存这些过滤器的副本以便可以了解网络中数据分布的大致情况. 其中, 节点只向基站传输已通过过滤的数据, 而基站则根据过滤器和收到的数据来求解网络的轮廓. 显然, 当所有节点的数据都在各自的过滤器内变化时, 基站必须要在不从网络中获取任何信息的情况下正确求解轮廓.

如图 1 所示, 传感器节点 1 ~ 5 构成了一个路由树, 如果利用  $TAG^{(4)}$  来计算网络中的轮廓, 那么 3 个时刻共需要传输 21 个数据包. 而这些时刻的轮廓始终为 {节点 3, 节点 4}, 这说明还有许多无用的数据依然在网络中传输. 如果采用 TDSM 算法, 每个节点被分配了右侧的超立方 (矩形) 作为过滤器. 在时刻 1 和 2, 没有数据通过过滤, 因此没有数据传输. 节点 4 的过滤器支配节点 5 的过滤器, 因此节点 4 的数据也一定支配节点 5 的数据. 节点 3 的过滤器支

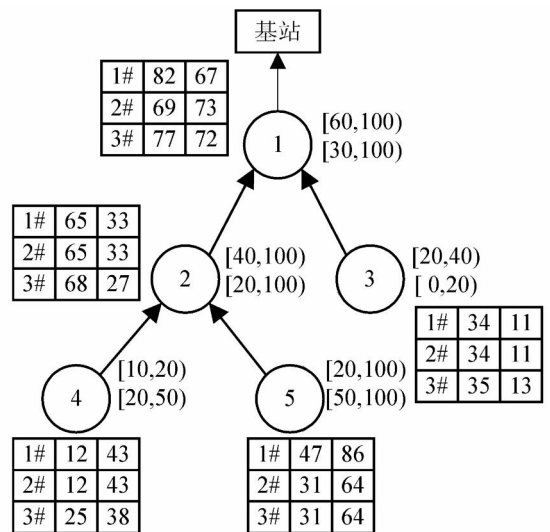


图 1 自顶向下轮廓监控举例

配节点 1, 2 和 5 的过滤器, 同样地, 节点 3 的数据也支配节点 1, 2 和 5 的数据. 节点 4 的过滤器在第 1 维支配节点 3, 而节点 3 在第 2 维支配节点 4, 因此它们的数据之间互不支配. 基站可以断定此时的轮廓一定是 {节点 3, 节点 4}. 在时刻 3, 节点 4 的数据通过了过滤器, 将它传输给基站, 基站不能准确判定它与节点 3 和 5 的数据之间的关系, 因而也就不能确定最终的轮廓结果. 这时, 需要基站从节点 3, 5 取回数据与节点 4 的数据比较来得到最终的轮廓. 此时共传输 8 个数据包, 与 TAG 相比, TDSM 只花费了一个广播包而节约了 13 个数据包的传输.

TDSM 涉及的问题主要包括感知数据建模、过滤器计算及维护和轮廓监控, 下面将分别加以介绍.

### 3.3 感知数据建模

传感器节点采集的是真实环境中的数据, 因此, 数据表现出非常强烈的时间相关性. 感知数据建模过程如下: 1) 用户给出概率密度函数的具体形式; 2) 节点利用已有的机器学习的算法求解其中的参数并将其发送给基站; 3) 基站根据收集到的参数确定各个节点的方程并根据方程计算出网络中的过滤器设置方案.

感知数据动态变化, 因此模型参数也动态地变化. 通常, 模型的更新需要及时地通知基站, 以便快速作出调整提高效率. 如果更新的频率太高, 模型的维护代价将很大. 解决该问题的方案如下: 如果新采集的数据没有通过过滤器, 说明过滤器有效, 则认为模型有效; 如果新采集的数据通过了过滤器, 则利用假设-检验的方法来对模型中的参数逐一进行检验, 一旦某个参数未通过检验, 说明模型失效, 将新的参数传输给基站.

在此, 以多元高斯分布为例说明具体的计算及维护过程. 多元高斯分布是一元高斯分布的扩展, 函数表示如下:

$$p(\vec{x}) = e^{-1/2(\vec{x}-\vec{\mu})^T \Sigma^{-1}(\vec{x}-\vec{\mu})} / (2\pi)^{n/2} |\Sigma|^{1/2}. \quad (1)$$

其中:  $\vec{\mu}$  为数据的均值,  $\Sigma$  为数据的协方差矩阵. 节点在保存  $p(\vec{x})$  的同时, 记录一些必要的统计值, 如均值  $\bar{X}$  和协方差矩阵  $S$ . 一旦新采集的数据通过了过滤就开始进行假设-检验. 首先, 进行均值检验, 检验方程如下:

$$T^2 = n(\bar{X} - \mu)^T \Sigma^{-1}(\bar{X} - \mu), \quad (2)$$

其中  $\Sigma^{-1} = S / (n - 1)$ . 一旦  $T^2$  大于给定的阈值, 则  $\mu$  检验失败, 需要重新计算模型并将新模型传输给基站. 如果通过了  $\mu$  检验, 则继续进行方差检验, 方程如下:

$$L = (n - 1) [\ln |\Sigma| - p - \ln |\hat{\Sigma}| + \text{tr}(\hat{\Sigma}^{-1})]. \quad (3)$$

同样地, 当检验失败时需要重新计算模型并通知基站.

假设-检验的时间复杂度是  $O(d^2)$ , 与巨大的通信代价相比, 只消耗较少的能量, 因此十分有效.

### 3.4 过滤器计算及维护

如果过滤器的设置可以保证所有数据均未通过过滤时, 基站可以不从网络中获取任何信息便可准确地计算出网络的轮廓, 那么它就是一个合理的过滤器设置方案. 此时要求过滤器的轮廓等于节点的轮廓. 根据定理 1, 需要过滤器轮廓内的超立方之间互不支配, 同时非轮廓内的超立方必须被至少一个轮廓内的超立方所支配. 并且直观地可以认为这种情况出现的越多, 过滤器的设置方案越合理.

$$\max_{i=1}^n P(r_i, h_i), \quad (4)$$

$$\text{s. t. } 1) \forall h_i, h_j \in H_s, h_i \not\supset h_j,$$

$$2) \forall h_k \in H - H_s, \exists h_i \in H_s, h_i \supset h_k.$$

这是一个非线性约束的非线性优化问题, 计算代价很高. 为了适应传感器网络的需要, 将两个条件进一步放宽: 对于  $H_s$  中的任何两个超立方, 都选定两维, 使其保证互不支配关系, 这样就变成了线性约束; 为  $H - H_s$  中的每个超立方都选定一个  $H_s$  的超立方来保证其支配关系, 这样也变成了线性约束. 该问题可以利用粒子群优化算法<sup>[9, 10]</sup>来解决.

当节点感知数据模型的参数发生变化时, 需要基站对过滤器的设置方案进行调整. 如果每次都重复上面的计算过程显然不合算, 这时可以只改变某些相关节点的过滤器, 其他的保持不变. 主要分为 4 种情况: 1) 轮廓节点, 变化后还是轮廓节点. 此时, 除了它本身的过滤器, 它原来的被支配节点和新的被支配节点的过滤器也需要调整. 2) 轮廓节点, 变化后为非轮廓节点. 除了它本身的过滤器, 它原来的被支配节点和新的支配节点以及新支配节点的被支配节点的过滤器都需要调整. 3) 非轮廓节点, 变化后还是非轮廓节点. 除了它本身的过滤器, 它的新支配节点和原来的支配节点的过滤器也需要调整. 4) 非轮廓节点, 变化后为轮廓节点. 除了它本身的过滤器, 所有的轮廓节点的过滤器都需要调整.

### 3.4 轮廓监控

节点时刻采集周围环境中的数据, 一旦数据通过了过滤就将它发送到基站. 由于采用的是树形的路由结构, 因此中间节点可通过网内计算来滤掉一些不必要的数据. 数据的变化需要通知基站, 而数据的内容只有在它影响了轮廓结果时才需要通知基站. 因此, 在中间节点可进行计算, 那些通过了过滤却被其他数据支配的节点, 只传输 id 即可通知基站

它发生了变化却未影响结果,仍是非轮廓节点.这样既能保证结果的正确性又节约了能量.

基站接收到的信息可以分为两个部分:节点 id 信息和节点的完整数据.首先从过滤器集合中删除只有 id 信息的节点,然后用完整的数据替换节点对应的过滤器.这样就得到一个数据点和超立方的混合集,然后求出这个集合的轮廓.根据定理 2 可以判断其中的哪些节点确定属于轮廓,而剩余的不确定是否属于轮廓的节点则需要基站取回相应信息进一步判断,同时相应地调整过滤器.这样,基站便可以时刻监控传感器网络中的轮廓,及时地对监控环境的情况的变化作出反应.

#### 4 仿真实验

本节给出了 TDSM 与 TAG 在各种参数变化时的性能验证结果.所有实验在主频 2.8 G,内存 512 M,操作系统为 Windows XP 的 PC 机上运行.首先,利用文献 [3] 中的仿真数据生成器生成独立分布和反相关分布的数据,以此作为各节点的均值,利用多元高斯分布生成器生成所需要的仿真数据.

图 2 表明各算法的通信代价都随着节点数量的增加而增加.这是因为节点数量的增加意味着每个时刻参与计算的数据点个数的增加,也就导致了轮廓结果数的增加,即通信代价增加.同时, TDSM 的通信代价始终小于 TAG 的通信代价,且增加速度比 TAG 缓慢,这说明了 TDSM 算法的有效性.

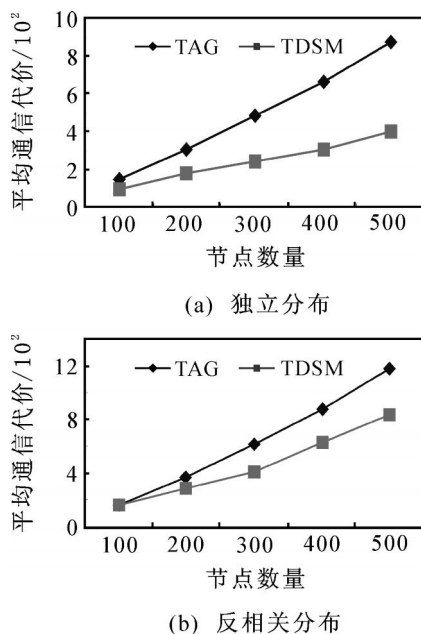


图 2 节点数量对性能的影响

图 3 表明,随着维度的增加,通信代价也相应增加, TDSM 的通信代价始终小于 TAG 的通信代价,而增长速度较快.因为维度的增加使得数据成为轮廓的概率增加,所以轮廓结果增加,也就导致了通信

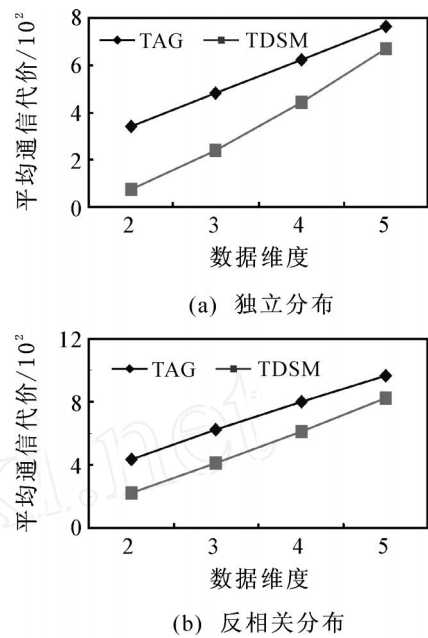


图 3 数据维度对性能的影响

代价的增加.而维度的增加也导致了过滤时需要判断的条件增加,因此过滤效果下降.

图 4 表明,随着协方差的增加, TAG 的通信代价比较稳定,而 TDSM 的通信代价则相应增加.这是由于协方差的增加并没有影响轮廓结果数量的显著变化,而 TAG 的通信代价依赖于结果数量,因此相对平稳.而协方差的增加意味着每个节点数据的变化范围的增加,同样的超立方的过滤效果也下降,因此通信代价增加.

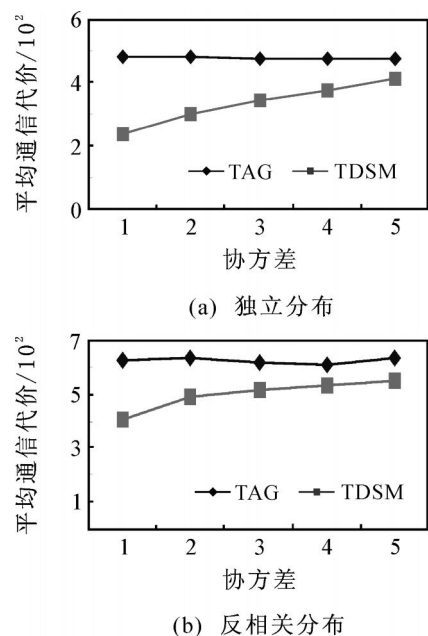


图 4 协方差对性能的影响

通过对上述实验结果的分析,可以得出如下结论:在传感器网络环境中, TDSM 算法与 TAG 算法相比可以显著地降低进行监控轮廓时的通信代价.

## 5 结 论

传感器网络应用中最重要问题就是如何节约节点的能量,本文提出了自顶向下的过滤算法来有效地监控网络中的轮廓节点.首先,提出了节点感知数据的建模和维护方案;其次,设计了详细的过滤器设置和维护过程;最后,介绍了如何在基站利用超立方之间的性质完成对网络内轮廓的有效监控.仿真结果表明:本文提出的 TDSM 算法能有效地避免不必要的数据传输,进而降低网络能耗,延长无线传感器网络的使用寿命.

### 参考文献(References)

- [1] 任丰原, 黄海宁, 林闯. 无线传感器网络 [J]. 软件学报, 2003, 14(2): 1148-1157.  
(Ren F Y, Huang H N, Lin C. Wireless sensor networks[J]. J of Software, 2003, 14(2): 1148-1157.)
- [2] 李建中, 李金宝, 石胜飞. 传感器网络及其数据管理的概念、问题与进展[J]. 软件学报, 2003, 14(10): 1717-1727.  
(Li J Z, Li J B, Shi S F. Concepts, issues and advance of sensor networks and data management of sensor networks[J]. J of Software, 2003, 14(10): 1717-1727.)
- [3] Borzsonyi S, Stocker K, Kossmann D. The skyline operator[C]. Proc of ICDE. Bangalore, 2003: 717-719.
- [4] Madden S, Szewczyk R, Franklin M J, et al. TAG: A tiny aggregation service for Ad-Hoc sensor networks [C]. Proc of OSDI. Boston, 2002: 131-146.
- [5] Yao Y, Johannes G. The cougar approach to in network query processing in sensor networks [J]. SIGMOD Record, 2002, 31(3): 9-18.
- [6] Chen H, Zhou S, Guan J. Towards energy-efficient skyline monitoring in wireless sensor networks [C]. Proc of EWSN. Netherlands, 2007: 101-116.
- [7] Xin J, Wang G, Chen L, et al. Continuously maintaining sliding window skylines in a sensor network [C]. Proc of DASFAA. Bangkok, 2007: 509-521.
- [8] Xin J, Wang G, Zhang X. Energy efficient skyline queries over sensor network using mapped skyline filters [C]. Proc of APWeb/WAIM. Huangshan, 2007: 144-156.
- [9] Kennedy J, Eberhart R. Particle swarm optimization [C]. Proc of ICNN. Perth, 1995: 1942-1948.
- [10] Eberhart R, Kennedy J. A new optimizer using particle swarm theory [C]. Proc of MHS95. Nagoya, 1995: 39-43.

(上接第 190 页)

### 参考文献(References)

- [1] 席裕庚. 预测控制[M]. 北京: 国防工业出版社, 1993.  
(Xi Y G. Model predictive control [M]. Beijing: National Defense Publishing House, 1993.)
- [2] 李德伟, 席裕庚. 一种基于衰减集结的鲁棒预测控制器 [J]. 自动化学报, 2008, 34(1): 48-54.  
(Li D W, Xi Y G. An amplitude decaying aggregation based robust model predictive control [J]. Acta Automatica Sinica, 2008, 34(1): 48-54.)
- [3] Blanchini F. Set invariance in control[J]. Automatica, 1999, 35(11): 1747-1767.
- [4] Kothare M V, Balakrishnan V, Morari M. Robust constrained model predictive control using linear matrix inequalities[J]. Automatica, 1996, 32(11): 1361-1379.
- [5] Wan Z Y, Kothare M V. Robust output feedback model predictive control using off-line linear matrix inequalities [J]. J of Process Control, 2000, 12(5): 763-774.
- [6] Wan Z Y, Kothare M V. An efficient off-line formulation of robust model predictive control using linear matrix inequalities[J]. Automatica, 2003, 39(6): 837-846.
- [7] Pluymers B, Rossiter J A, Suykens J A K, et al. The efficient computation of polyhedral invariant sets for linear systems with polytopic uncertainty [C]. Proc of the American Control Conference. Portland, 2005: 804-809.
- [8] Lee Y I, Kouvaritakis B. Robust receding horizon predictive control for systems with uncertain dynamics and input saturation [J]. Automatica, 2000, 36(10): 1497-1504.
- [9] Lee Y I, Kouvaritakis B. A linear programming approach to constrained robust predictive control [J]. IEEE Trans on Automatic Control, 2000, 45(9): 1765-1770.
- [10] Lee Y I, Kouvaritakis B. Superposition in efficient robust constrained predictive control [J]. Automatica, 2002, 38(6): 875-878.
- [11] 丁宝苍, 杨鹏, 孙鹤旭, 等. 离线鲁棒预测控制器综合方法的改进方案 [J]. 控制与决策, 2005, 20(3): 312-315.  
(Ding B C, Yang P, Sun H X, et al. Improved off-line synthesis approach of robust model predictive control [J]. Control and Decision, 2005, 20(3): 312-315.)