第5卷 第4期 自动化学报 Vol. 5, No. 4
1979年10月 ACTA AUTOMATICA SINICA Oct., 1979

关于递阶控制的几个问题

陈 珽
（华中工学院自控系）

摘 要

本文论述了近几年关于递阶控制的几个问题．全文分四节．第一节较详细地介绍了从无限维凸规划出发提出的辅助问题原则和松弛原则，根据这两个原则形成的统一方法可以推导出若干分解-协调算法． 第二节介绍了用递阶方法离线地计算线性和非线性系统的闭环控制律，并对稳态系统的在线控制问题作了较详细的介绍．第三节论述了 Stackelberg 策略在递阶控制中的应用．第四节介绍了两种递阶系统状态估计的算法．

本文论述了近几年关于递阶控制的几个理论问题，这就是统一的方法、在线及闭环控制、对策论的应用、状态估计及随机控制，现分述如下．

一、统一的方法

过去已有许多人提出了各种递阶最优化的分解-协调算法，如何把这些算法统一起来，形成更广泛的统一的方法是有意义的．下面将介绍这方面的工作，其中特别是 Cohen 的工作[1]是有成果的．

Cohen 的工作建立在无限维凸规划的基础上，他提出了两个基本原则，可由此推导若干迭代算法．

现有的数值计算方法已使用了“辅助问题原则”的概念．这个原则是如果要求得某个受约束的函数的极值（这个问题称为主问题），可以构成另一个函数去求它的极值（这个问题称为辅助问题），只要这两个函数在空间的同一点都达到它们的极值． 如果解答的存在性和唯一性对主问题和辅助问题都成立，则求得辅助问题的解也就求得了主问题的解．

在研究动态系统时，通常是求泛函的极值．设有一泛函 J（即系统的目标函数），在希尔伯特空间 U 中是凸的和可微的，而 U^f 为 U 的一闭凸子集，则主问题为

$$\min_{\boldsymbol{u}\in U^f} J(\boldsymbol{u}). \tag{1}$$

J 达到极小值的必要条件是：对于所有 $\boldsymbol{u}\in U^f$，存在一 $\boldsymbol{u}^*\in U^f$，使

$$\langle J'(\boldsymbol{u}^*),\quad \boldsymbol{u}-\boldsymbol{u}^*\rangle \geqslant 0. \tag{2}$$

在此 $\boldsymbol{u}^*$ 为 J 在子集 U^f 中的极小值点，J' 为 J 对 $\boldsymbol{u}$ 的导数，$\langle\cdot,\cdot\rangle$ 表示空间 U 和它的对偶空间 U^* 的对偶积，在希尔伯特空间中就是内积．

如果我们构成另一个泛函 $K(\boldsymbol{u})$，它有和 $J(\boldsymbol{u})$ 同样的假设，由此又构成泛函 $G^v(\boldsymbol{u})$，

$$G^v(\boldsymbol{u}) = K(\boldsymbol{u}) + \langle \epsilon J'(\boldsymbol{v}) - K'(\boldsymbol{v}), \boldsymbol{u} \rangle, \tag{3}$$

式中 ϵ 为一大于零的常数，$\boldsymbol{v} \in U^f$ 它如此选择使

$$G^v(\boldsymbol{v}) = \min_{\boldsymbol{u} \in U^f} G^v(\boldsymbol{u}). \tag{4}$$

按照 (2) 式的必要条件 $G^v(\boldsymbol{v})$ 应适合

$$\langle G'^v(\boldsymbol{v}), \quad \boldsymbol{u} - \boldsymbol{v} \rangle \geqslant 0, \tag{5}$$

对于所有 $\boldsymbol{u} \in U^f$. 不难证明，当 $\boldsymbol{v}$ 使 $G^v(\boldsymbol{u})$ 达到极小时，它也使 $J(\boldsymbol{u})$ 达到极小，即

$$J(\boldsymbol{v}) = \min_{u \in U^f} J(\boldsymbol{u}). \tag{6}$$

因而求 $J(\boldsymbol{u})$ 的极小等效于求 $G^v(\boldsymbol{u})$ 的极小.

上述方法称为辅助问题原则. 这个原则的主要优点是: 求解辅助问题依赖于泛函 $K(\boldsymbol{u})$ 的选择 (因而把 $K(\boldsymbol{u})$ 称为核)，而 $K(\boldsymbol{u})$ 的选择所受的约束是相当温和的. 而且在 $G^v(\boldsymbol{u})$ 中所包含的另一项是 $\boldsymbol{u}$ 的一线性修正项，比较便于计算. 当 $K(\boldsymbol{u})$ 选择为不同的泛函时，可以形成不同的算法.

对于求 (4) 式的解答，Cohen 采用迭代算法. 这个算法是: 首先选择 (3) 式中的 $\boldsymbol{v}$ 为 U^f 中的某一 $\boldsymbol{u}^0$，然后对 $\boldsymbol{u}$ 求 $G^v(\boldsymbol{u})$ 的极小，求得一 $\boldsymbol{u}^1$. 再把 (3) 式的 $\boldsymbol{v}$ 用 $\boldsymbol{u}^1$ 代入，又求 $G^v(\boldsymbol{u})$ 的极小，求得一 $\boldsymbol{u}^2$，如此等等. Cohen 证明了在一定条件下这个算法的收敛性.

由于把主问题的求解变换为辅助问题的求解给计算带来了很大的灵活性. 不论原来的主问题是否可以分解，原则上都可以把辅助问题分解为若干子问题，用分解-协调的方法去计算，这是以前的方法办不到的，以前的方法必须分解主问题.

进行分解可采用两种办法，一种是顺序分解. 把空间 U 分解为

$$U = U_1 \times \cdots \times U_N \tag{7}$$

$$U^f = U_1^f \times \cdots \times U_N^f \tag{8}$$

$$U_1^f \subset U_1, \cdots, U_N^f \subset U_N \tag{9}$$

得到"松弛算法". 所谓松弛算法是: 设系统的目标函数为

$$J(u_1, \cdots, u_N), \tag{10}$$

固定其中的 $N-1$ 个控制，松弛其中的一个，如 u_i，以求

$$\min_{u_i \in U_i^f} J(u_1^{k+1}, \cdots, u_{i-1}^{k+1}, u_i, u_{i+1}^k, \cdots, u_N^k), \tag{11}$$

在求得 u_i^{k+1} 以后再顺序松弛 u_{i+1}，如此等等.

辅助问题算法和松弛算法可以合并使用. 为简便，设有两个变量 $\boldsymbol{u}$ 及 $\boldsymbol{v}$，首先固定 $\boldsymbol{v}$ 松弛 $\boldsymbol{u}$ 求

$$\min_{u \in U^f} K(\boldsymbol{u}, \boldsymbol{v}^k) + \langle \epsilon_1 J'_u(\boldsymbol{u}^k, \boldsymbol{v}^k) - K'_u(\boldsymbol{u}^k, \boldsymbol{v}^k), \boldsymbol{u} \rangle, \tag{12}$$

令 $\boldsymbol{u}^{k+1}$ 为其解. 然后固定 $\boldsymbol{u}$ 松弛 $\boldsymbol{v}$ 求

$$\min_{v \in V^f} K(\boldsymbol{u}^{k+1}, \boldsymbol{v}) + \langle \epsilon_2 J'_v(\boldsymbol{u}^{k+1}, \boldsymbol{v}^k) - K'_v(\boldsymbol{u}^{k+1}, \boldsymbol{v}^k), \boldsymbol{v} \rangle, \tag{13}$$

求得 $\boldsymbol{v}^{k+1}$ 为其解，如此反复进行迭代. 这个方法可以推广到 N 个控制变量.

分解的另一种办法是并行分解．设

$$K(\boldsymbol{u}) = \sum_{i=1}^{N} K_i(u_i), \tag{14}$$

(14) 式中的 $K_i(u_i)$ 可以选择为

$$K_i(u_i) = K_i(\mathscr{R}_i\boldsymbol{u}), \quad i = 1, \cdots, N. \tag{15}$$

(15)式中的 $\mathscr{R}_i\boldsymbol{u}$ 定义为

$$\mathscr{R}_i\boldsymbol{u} \triangleq u_1^k, \cdots, u_{i-1}^k, u_i, u_{i+1}^k, \cdots, u_N^k, \tag{16}$$

即N个变量中仅有一个 u_i 是可变的，其余都是固定的．显然，采用这样一种分解可以利用N部小计算机对N个子问题同时进行计算，而前面的顺序算法则是用一部计算机对N个子问题顺序地进行计算．

由于篇幅关系，本文只对 Cohen 工作中基本的部分，即两个原则作了介绍．要运用这两个原则去求得实际应用的分解-协调算法一般还有许多工作要做．这些工作是：第一，实际控制问题常常有一些明显的约束条件要考虑，例如系统的状态方程，控制作用 $\boldsymbol{u}$ 的等或不等的约束等等．因此需要把上面的算法作某些推广以求解这一类的问题．这需要寻求拉格朗日泛函的鞍点，如果它存在的话．第二，迭代算法能否收敛到它的真实解是一个十分重要的问题．在这方面 Cohen 做了大量的工作但没有解决所有的问题，更没有形成统一的收敛性理论．第三，这个方法最主要的一点是关于核 $K(\boldsymbol{u})$ 的选择，Cohen 没有给出一个规则，还得靠经验，他称为“问题的处理”．

但是无论如何这个方法是有价值的．第一，他把原有的若干分解-协调方法联系起来，而且在理论上推进了一步，即采用辅助问题原则以后主问题能不能分解，原则上并无关系，分解在这里是作为一种计算手段，它是对核的特殊选择所产生的．因此如果核选择不同，一方面它可以推得某些经典的迭代算法，如梯度算法、牛顿-拉夫申算法等，而另一方面又可以得到分解-协调算法，这就把这两类算法勾通起来了．第二，把以前几乎所有的分解-协调算法建立在上述两个原则统一的基础上，Cohen 推导了关联预估法、关联平衡法、Takahara 算法、Findeisen 的直接法等．第三，有可能探索新的算法．

Grateloup 和 Titli[2] 是从另一个方面，即从最优控制理论的角度对现有的一些分解-协调算法找出它们之间的关系．

他们的基本想法是用变分法去求整个系统的最优解，得到了通常的一级算法．但如果把从变分法得到的方程中的某几个(和它相应的变量)放在第一级，即局部级去求解，而把另外的几个方程(及其相应的变量)放到第二级，即协调级去求解，然后在两级之间进行迭代，就形成了两级最优控制算法．如果在计算步骤中把第一级和第二级的方程安排得不同，就得到不同的算法．Grateloup 和 Titli 研究了连续系统，他们推导出可行法 (feasible method) 和不可行法 (unfeasible method)．

Mahmoud 把这种想法用到离散系统[3]，并作了一些推广，但他的推导有错误，见 Cohen 的评论[4]．

Hassan 根据这种想法形成三级算法，避免了求解黎卡提方程，给计算上带来了很大的方便[5]．

二、在线及闭环控制

目前流行的离线控制算法在实际应用中存在不少缺点，因而发展在线和闭环控制算法是有实际意义的．本节就这方面所进行的工作作一介绍．

1. 闭环控制

目前的工作主要企图对子系统进行闭环控制． 这些方法都是离线计算最优控制律，然后运用得到的控制律在子系统实现闭环控制． 采用这种方法，系统的控制质量在很大程度上取决于模型的准确度．在文献 [6] 中研究了线性二次型系统．

一般，线性二次型的子系统如实现闭环控制，其最优控制律的形式为

$$\boldsymbol{u}_i(t) = K_{ii}(t)\boldsymbol{x}_i(t) + \boldsymbol{v}_i(t), \tag{17}$$

式中 $\boldsymbol{u}_i(t)$ 是第 i 个子系统的控制向量，$\boldsymbol{x}_i(t)$ 是它的状态向量，$K_{ii}(t)$ 是反馈增益，$\boldsymbol{v}_i(t)$ 为其它子系统对第 i 个子系统的影响．

在文献 [6] 中采用关联预估法对一线性定常系统具有二次型目标函数的最优控制律进行离线计算，得到

$$\boldsymbol{u}_i(t) = -R_i^{-1}B_i^T[K_i\boldsymbol{x}_i(t) + \boldsymbol{s}_i(t)], \tag{18}$$

式中 R_i 为二次型目标函数中 $\boldsymbol{u}_i$ 的加权矩阵，B_i 为第 i 个子系统状态方程中 $\boldsymbol{u}_i$ 的系数矩阵，K_i 为矩阵黎卡提方程的解，它与 $\boldsymbol{x}(t_0)$ 无关．因而由此实现的部分闭环控制，其控制律不受子系统初始条件变化的影响．文献 [6] 进一步寻求整个系统 $\boldsymbol{s}(t)$ 与 $\boldsymbol{x}(t)$ 的关系．在此 $\boldsymbol{s}(t) = [\boldsymbol{s}_1^T(t), \cdots, \boldsymbol{s}_N^T(t)]^T$，$\boldsymbol{s}_i(t)$ 即(18)式中的关联项，而

$$\boldsymbol{x}(t) = [\boldsymbol{x}_1^T(t), \cdots, \boldsymbol{x}_N^T(t)]^T.$$

不难推得

$$\boldsymbol{s}(t) = Y(t_f, t)\boldsymbol{x}(t), \tag{19}$$

式中 $Y(t_f, t)$ 为一矩阵方程的解，t_f 为过程的终点时刻． 只有当 $t_f \to \infty$，$Y(t_f, t)$ 成为常系数矩阵才便于求解，但仍需要对一很大的矩阵 X 求逆．

在此，分解-协调仅作为推导控制律时的一种离线算法，而实施时并不是两级控制．由于在一般情况 $Y(t_f, t)$ 并不是以对角块矩阵的形式出现，即不能去偶，因而 $\boldsymbol{s}_i(t)$ 将与整个状态 $\boldsymbol{x}(t)$ 有关，从而 $\boldsymbol{u}_i(t)$ 依赖于整个系统的 $\boldsymbol{x}(t)$． 这种形式的反馈控制并不是分散的而是集中的．

非线性系统的问题当然要比线性系统的问题复杂得多．首先需要采用一种方法去求解非线性系统的最优控制问题，例如可以采用拟线性化方法[7]，或用 Hassan 的预估方法[8,9]．但是无论采用什么方法，得到的非线性系统的闭环控制律总与系统的初始状态有关，这是由非线性系统的本质所决定的．为了解决这个困难，人们试图从两个方面采取一些办法[10]．一是定义一灵敏度去判断系统的目标函数受初始条件影响的程度．如果在某些特殊情况下这个灵敏度的值很小，就可以认为初始条件的变化（在一定范围内）可以忽略．二是采用一种方法在线补偿初始条件变化的影响． 例如，设闭环最优控制为

$$\boldsymbol{u} = H\boldsymbol{x} + \boldsymbol{q}, \tag{20}$$

式中 H 不受而 $\boldsymbol{q}$ 则受初始条件变化的影响．如在某时刻 τ，由于未知的扰动，系统的状态

改变到 $\boldsymbol{x}(\tau)$，就可以把它作为初始状态，采用一种迭代方法去计算 $\boldsymbol{q}$ 的新值，从而对 $\boldsymbol{q}$ 作修正．显然，上述两种方法都不能从根本上改善非线性闭环控制．

2. 在线算法

首先讨论稳态系统，这是目前研究得比较仔细的一种系统[11]．这类系统由于扰动和系统参数的改变要求系统中控制器的设定值作相应的改变．设定值的改变要求实现在线最优化．如果设定值的改变次数不多，一次改变和下次改变的时间间距相对于系统的时间常数很大，就可以忽略它的过渡过程，把系统看成是在一系列稳态下运行．

线性或非线性稳态系统的状态方程、约束条件和目标函数等是一些等或不等的代数方程．这个问题当然也可以用分解-协调方法求解．这里遇到的问题和数学规划问题不同之处在于数学规划问题的求解完全依赖于数学模型，而现在作为在线控制就要充分利用由测量得到的真实系统的信息去改善控制．

设提出的问题为第 i 个子系统有

$$\boldsymbol{y}_i = \boldsymbol{f}_i(\boldsymbol{u}_i, \boldsymbol{z}_i), \quad i = 1, \cdots, N, \tag{21}$$

式中 $\boldsymbol{y}_i$、$\boldsymbol{u}_i$、$\boldsymbol{z}_i$ 各为子系统 i 的输出、控制和关联向量，而关联向量 $\boldsymbol{z}_i$ 又决定于系统的输出 $\boldsymbol{y}$：

$$\boldsymbol{z}_i = H_i \boldsymbol{y}, \tag{22}$$

式中 H_i 为给定的关联矩阵．设在约束条件

$$\sum_{i=1}^{N} r_i(\boldsymbol{u}_i, \boldsymbol{z}_i) \leqslant r \tag{23}$$

以及

$$G_i(\boldsymbol{u}_i, \boldsymbol{z}_i) \leqslant 0, \quad i = 1, \cdots, N \tag{24}$$

之下使目标函数

$$J = \Psi(J_1, \cdots, J_N) \tag{25}$$

为极小，其中 $J_i(\boldsymbol{u}_i, \boldsymbol{z}_i)$ 为第 i 个子系统的目标函数，而 Ψ 为一严格的次序保持函数．

从协调量的选择 Findeisen 把协调方法区分为两类，一类称为直接法，另一类称为价格法．直接法所选择的协调量为子系统的输出量 $\boldsymbol{y}_i$．如果子系统还有不等的约束 (23) (例如由资源分配所产生)，则 r_i 也可以当作协调量．采用这个方法实际遇到的困难是协调器规定 $(\boldsymbol{y}_{d1}, r_{d1})$ 的值以后，子系统在约束条件 (21)、(22)、(24) 之下可能无最优解．因此 $(\boldsymbol{y}_d, r_d)$ 的集必须如此选择以保证系统的目标函数能达到其极小值，即规定

$$(\boldsymbol{y}_d, \boldsymbol{r}_d) \in Y_R. \tag{26}$$

但这个集 Y_R 不易事先确定．

为了绕过这个困难，在直接协调中运用罚函数，即把目标函数修改为

$$J'_i = J_i(\boldsymbol{u}_i, \boldsymbol{z}_i) + K_i(\boldsymbol{y}_i - \boldsymbol{y}_{di}), \tag{27}$$

式中 $K_i(\boldsymbol{y}_i - \boldsymbol{y}_{di})$ 称为罚函数．(27) 式仍受约束于 (21)、(23)、(24) 式．在引入罚函数以后就不必硬性要求 $\boldsymbol{y}_i$ 一定等于 $\boldsymbol{y}_{di}$，从而使问题有了伸缩性．

采用价格法所选择的协调量为拉格朗日乘子或关联向量，因此通常的关联平衡法、关联预估法都属于这一类．

把价格法用到在线，是把按模型计算的子系统的控制施加到真实系统，由此得到各子

系统的输出 $\boldsymbol{y}_1^*, \cdots, \boldsymbol{y}_N^*$,（不加星号的量表示模型输出，加星号的量表示真实输出）．将这些真实输出再反馈到协调级，在协调级计算最优的拉格朗日乘子 $\boldsymbol{\beta}_i$． Findeisen 所采用的一种协调算法是寻求 $\boldsymbol{\beta}=\tilde{\boldsymbol{\beta}}$ 使

$$\hat{\boldsymbol{z}}_i(\tilde{\boldsymbol{\beta}})=\boldsymbol{z}_i^*(\hat{\boldsymbol{u}}(\tilde{\boldsymbol{\beta}})),\quad i=1,\cdots,N, \tag{28}$$

式中 $\hat{\boldsymbol{z}}_i(\tilde{\boldsymbol{\beta}})$ 是令 $\boldsymbol{\beta}=\tilde{\boldsymbol{\beta}}$ 按模型计算得到的关联向量，而

$$\boldsymbol{z}_i^*=H_i^*\boldsymbol{y}^* \tag{29}$$

为从真实系统测量的关联向量．这就是说，协调级应选择一 $\tilde{\boldsymbol{\beta}}$ 使计算得到的关联向量 $\hat{\boldsymbol{z}}$ 和测量得到的关联向量 $\boldsymbol{z}^*$ 相等，从而使计算结果尽量接近真实结果．

从直观上看，由于利用真实系统的信息，这个算法所得到的结果至少不会比纯粹按模型计算的结果要差．

直接求解 (28) 式是有困难的，因为式中的 $\boldsymbol{z}_i^*$ 是由实测得到的并不知道它是 $\boldsymbol{\beta}$ 的什么函数，而且在不等约束条件下函数

$$\boldsymbol{F}^*(\boldsymbol{\beta})=\hat{\boldsymbol{z}}(\boldsymbol{\beta})-\boldsymbol{z}^*(\boldsymbol{\beta}) \tag{30}$$

可能不可微．解决这个问题的一种方法是用一已知的可微的函数 $\boldsymbol{F}(\cdot)$ 去逼近 $\boldsymbol{F}^*(\cdot)$，然后用牛顿算法去迭代

$$\boldsymbol{\beta}^{k+1}=\boldsymbol{\beta}^k-[\boldsymbol{F}'(\boldsymbol{\beta}^k)]^{-1}\boldsymbol{F}^*(\boldsymbol{\beta}^k), \tag{31}$$

式中 $\boldsymbol{F}'(\cdot)$ 为 $\boldsymbol{F}(\cdot)$ 的导数，这个问题的详细讨论见文献[11]．

关于这种在线算法还需要讨论的一个问题是可行性的问题．一种迭代算法即使其最终结果能符合所有约束条件，但在迭代过程中某些约束条件被突破则这种算法称为不可行的．显然不可行的算法不能在线使用，如何产生可行算法在文献[11]中有讨论．

动态系统的在线控制要比稳态系统难得多，这是因为在这种情况采用迭代算法有困难．迭代算法是对一个动态系统的全过程进行迭代，通过多次迭代逐步接近最优．因此严格地说，除非是多次重复的批量生产过程，否则在迭代过程中无法利用从真实系统反馈回来的信息．

解决这类问题的一个途径是定期地（例如在时刻 0, T_1, $2T_1$, $\cdots$）把从真实过程测得的状态送到各子系统，各子系统把这些实测的状态作为它在该时刻的初始状态重复地计算它的最优控制[12,13]．由于各子系统利用了从过程获得的真实信息，由此产生的控制应当要比纯粹由模型计算的控制要好．如果不把实测的状态送到协调级，则由协调级计算的拉格朗日乘子或关联向量仍可能有较大的误差．为此，还应当把实测的状态周期地（一般比子系统的周期要长）送到协调级，并重复地计算其相应的轨线．为了简化协调级的计算可以在这级采用降阶的模型．

显然，这种方法要能实际应用，除非动态过程的持续时间很长，而利用实测信息计算最优轨线足够快．否则计算速度跟不上过程变化，这个算法没有实用价值．

三、对策论的应用

目前研究的一种方法是把 Stackelberg 策略用到多级控制[14,15]． Stackelberg 策略是一种求解二人非零和对策的方法，现在先用静态问题说明．

设有两个局中人，局中人 1 选择策略（对于系统也可以称为控制）$u_1\in R$，局中人 2 选择策略 $u_2\in R$。联系到局中人 1 的纯量目标函数为 $J_1(u_1, u_2)$，联系到局中人 2 的纯量目标函数为 $J_2(u_1,u_2)$。局中人 1 为领导，局中人 2 为随从．局中人 1 知道 J_1 及 J_2，还知道他选择 u_1 以后局中人 2 的反应，即知道局中人 2 一定会选择 $u_2=T_2(u_1)$，T_2 为从 u_1 到 u_2 的 1 对 1 的映射．局中人 2 仅知道本身的 J_2 而不知道 J_1，但知道局中人 1 的策略 u_1．各人都企图使自己的目标函数为最小而不考虑别人的．由此可知，当局中人 1 选择 u_1 以后局中人 2 选择的 u_2 将使

$$J_2(u_1, T_2(u_1))\leqslant J_2(u_1, u_2), \tag{32}$$

对于所有 $u_2\in R$．局中人 1 选择 u_1^* 使

$$J_1(u_1^*, T_2(u_1^*))\leqslant J_1(u_1, T_2(u_T)), \tag{33}$$

对于所有 $u_1\in R$．策略 u_1^* 称为局中人 1 的 Stackelberg 策略，而策略 $u_2=T_2(u_1)$ 称为局中人 2 的 Stackelberg 策略．

容易看出，如果把 Stackelberg 策略中的领导（局中人 1）看成递阶控制中的协调器，而把随从（局中人 2）看成局部控制器，就可以用来求解某类递阶控制问题，但是需要把 Stackelberg 策略作某些推广，一是把它推广到动态对策，二是把局中人 2（随从）推广到 N 个．

首先研究一动态系统，设其状态方程为

$$\dot{\boldsymbol{x}}=\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}_1, \boldsymbol{u}_2), \tag{34}$$

式中 $\boldsymbol{x}$ 为系统的状态．在此我们把系统的控制 $\boldsymbol{u}_1$ 和 $\boldsymbol{u}_2$ 作为局中人 1 和 2 的策略，并设局中人 1 为领导，局中人 2 为随从．又设他们的纯量目标函数各为

$$J_i=\Phi_i[\boldsymbol{x}(T)]+\int_0^T \Psi_i(\boldsymbol{x}, \boldsymbol{u}_1, \boldsymbol{u}_2)dt,\quad i=1,2. \tag{35}$$

运用变分法不难求得局中人 2 的开环 Stackelberg 策略为

$$\frac{\partial H_2}{\partial \boldsymbol{u}_2}=0, \tag{36}$$

$$\dot{\boldsymbol{p}}^T=-\frac{\partial H_2}{\partial \boldsymbol{x}}, \tag{37}$$

$$\boldsymbol{p}^T(T)=\frac{\partial \Phi_2}{\partial \boldsymbol{x}(T)}[\boldsymbol{x}(T)], \tag{38}$$

其中哈密顿量 H_2 为

$$H_2(\boldsymbol{x}, \boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{p})=\Psi_2(\boldsymbol{x}, \boldsymbol{u}_1, \boldsymbol{u}_2)+\boldsymbol{p}^T\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}_1, \boldsymbol{u}_2). \tag{39}$$

局中人 1 的 Stackelberg 策略为

$$\frac{\partial H_1}{\partial \boldsymbol{u}_i}=0,\quad i=1,2, \tag{40}$$

$$\dot{\boldsymbol{\lambda}}_1^T=-\frac{\partial H_1}{\partial \boldsymbol{x}}, \tag{41}$$

$$\dot{\boldsymbol{\lambda}}_2^T=-\frac{\partial H_1}{\partial \boldsymbol{p}}, \tag{42}$$

$$\boldsymbol{\lambda}_1^T(T)=\frac{\partial \Phi_1}{\partial \boldsymbol{x}[T]}[\boldsymbol{x}(T)]-\boldsymbol{\lambda}_2^T(T)\left[\frac{\partial^2 \Phi_2}{\partial \boldsymbol{x}(T)^2}[\boldsymbol{x}(T)]\right], \tag{43}$$

$$\boldsymbol{\lambda}_2(0)=0, \tag{44}$$

哈密顿量 H_1 为

$$H_1(\boldsymbol{x}, \boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{p}, \boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \boldsymbol{\beta}_2) = \Psi_1(\boldsymbol{x}, \boldsymbol{u}_1, \boldsymbol{u}_2) + \boldsymbol{\lambda}_1^T \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}_1, \boldsymbol{u}_2) + \boldsymbol{\lambda}_2^T \left(-\frac{\partial H_2}{\partial \boldsymbol{x}}\right)^T + \boldsymbol{\beta}_2^T \left(\frac{\partial H_2}{\partial \boldsymbol{u}_2}\right)^T. \tag{45}$$

在 H_1 中把 (36)、(37) 式作为约束条件考虑进去，因为按照 Stackelberg 策略 $\boldsymbol{u}_2 = T_2(\boldsymbol{u}_1)$，这个关系包含在 (36) 及 (37) 式中. 这组方程的推导及用于求解线性二次型问题见文献[16].

使用开环 Stackelberg 策略的缺点是动态规划不能应用. 如果要用动态规划就需要对开环 Stackelberg 策略作修改，这就形成了所谓反馈的 Stackelberg 策略.

当把 Stackelberg 策略用到递阶控制时还需要把局中人 2 推广到 N 个. 为简便，现设局中人 2 有两个，其目标函数各为 J_1 和 J_2，而策略各为 u_1 和 u_2. 当局中人 1 的策略 u_0 给定时，u_1 及 u_2 根据 Nash 平衡条件去选择，即

$$J_1(u_0, T_1(u_0), T_2(u_0)) \leqslant J_1(u_0, u_1, T_2(u_0)), \tag{46}$$

$$J_2(u_0, T_1(u_0), T_2(u_0)) \leqslant J_2(u_0, u_2, T_1(u_0)). \tag{47}$$

局中人 1 选择一 u_0 使其目标函数 $J_0(u_0, u_1, u_2)$ 在条件 $u_1 = T_1(u_0)$ 及 $u_2 = T_2(u_0)$ 下为极小，即选择 u_0 为 u_0^s 使

$$J_0(u_0^s, T_1(u_0^s), T_2(u_0^s)) \leqslant J_0(u_0, T_1(u_0), T_2(u_0)), \tag{48}$$

对于所有 u_0.

Cruz 利用以上结果研究一线性离散系统具有二次型目标函数的最优解. 对此，他把局中人 1 看成协调器，而把 N 个局中人 2 看成 N 个子系统的控制器（下面暂设两个）. 他提出的问题是：设系统表示为

$$\boldsymbol{x}(k+1) = A(k)\boldsymbol{x}(k) + B^0(k)\boldsymbol{u}^0(k) + B^1(k)\boldsymbol{u}^1(k) + B^2(k)\boldsymbol{u}^2(k) + \boldsymbol{v}(k), \tag{49}$$

式中 $\boldsymbol{u}^0(k)$ 为协调器的控制，而 $\boldsymbol{u}^1(k)$ 及 $\boldsymbol{u}^2(k)$ 为两个子系统的控制，$\boldsymbol{v}(k)$ 为高斯随机向量，它表示系统受到的干扰. 设协调器及控制器都能测量它们的输出

$$\boldsymbol{y}^i(k) = H^i(k)\boldsymbol{x}(k) + \boldsymbol{\xi}^i(k), \quad i = 0, 1, 2, \tag{50}$$

式中 $\boldsymbol{\xi}^i(k)$ 为高斯随机向量，它表示第 i 个输出中的噪声. 局中人 i 的目标函数为

$$J^i(\boldsymbol{u}^i) = \frac{1}{2}\boldsymbol{x}^T(N)K^i(N)\boldsymbol{x}(N) + \frac{1}{2}\sum_{k=0}^{N-1}[\boldsymbol{x}^T(k)Q^i(k)\boldsymbol{x}(k) + (\boldsymbol{u}^i)^T(k)R^i(k)\boldsymbol{u}^i(k)], \quad i = 0, 1, 2. \tag{51}$$

现在采用 Stackelberg 策略求系统的最优解 $\boldsymbol{u}^0, \boldsymbol{u}^1, \boldsymbol{u}^2$. Cruz 假设协调器和所有子系统控制器通过它们的测量都能得到整个系统的状态的完全的信息，即假设

$$\boldsymbol{y}^0(k) = \boldsymbol{y}^1(k) = \boldsymbol{y}^2(k) = \boldsymbol{x}(k), \tag{52}$$

式中 $\boldsymbol{y}^0$ 为协调器的输出而 $\boldsymbol{y}^1(k)$ 及 $\boldsymbol{y}^2(k)$ 为子系统 1 及 2 的输出. 他推导的结果是：对于子系统控制器

$$\boldsymbol{u}^i(k) = -\Delta^i(k)[A(k)\boldsymbol{x}(k) + B^0(k)\boldsymbol{u}^0(k)], \quad i = 1, 2. \tag{53}$$

对于协调器

$$\boldsymbol{u}^0(k) = -L^0(k)\boldsymbol{x}(k). \tag{54}$$

(53) 及 (54) 式中的增益矩阵 $L^0(k)$ 和 $\Delta^i(k)$ 都是利用递推公式从 $k = N-1$ 开始在时

间上逆向计算得到，即采用了动态规划的方法．

Cruz 还讨论了以下情况：$\boldsymbol{y}^1(k)=\boldsymbol{y}^2(k)$，协调器的测量至少包含 $\boldsymbol{y}^1(k)$，即它可能有比 $\boldsymbol{y}^1(k)$ 更多的信息．在这种情况所得的结果比上面要复杂，见文献[14]．

当协调器不知道第二级控制器所有的测量以及第二级控制器没有相同的测量时，就非常难于构成一最优反馈 Stackelberg 问题．只有在每个控制律的结构被标定，例如它们局限于线性时能推导其必要条件，见文献[14]．

Madanic 等人还研究了一类特殊的递阶控制，所谓M级递阶对策[17]．在此，最高一级的决策者为以下各级的领导，最下一级的决策者则为以上各级的随从，而中间各级决策者对上级为随从对下级为领导．显然，这种对策有它的现实意义．

请注意，现在在对策论中提出的问题和前两节中提出的问题有些不同．前面提出的问题整个系统只有一个目标函数，虽然这个目标函数可按子系统进行分解．现在研究的问题本质上是多目标决策问题，因为我们假设协调级及下级的各子系统各有其本身的目标函数．此外，这里还假设协调级有一控制直接作用到各子系统，这在前面的问题中也是没有的．因此虽然这些方法都用于求解递阶最优控制，但解决的问题还有所不同．

上面的讨论并没有对领导的目标函数 J_0 和下级决策者的目标函数 $J_i(i=1,\cdots,N)$ 标定任何关系．现在假设 J_0 为 J_1 及 J_2 的一凸线性组合，即设

$$J_0(u_0,u_1,u_2)=\alpha_1J_1(u_0,u_1,u_2)+\alpha_2J_2(u_0,u_1,u_2), \tag{55}$$

而

$$\alpha_1\geqslant 0,\quad \alpha_2\geqslant 0,\quad \alpha_1+\alpha_2=1. \tag{56}$$

如果选择 u_0 为 u_0^s 使

$$\begin{aligned}&\alpha_1J_1(u_0^s,T_1(u_0^s),T_2(u_0^s))+\alpha_2J_2(u_0^s,T_1(u_0^s),T_2(u_0^s))\\&\leqslant\alpha_1J_1(u_0,T_1(u_0),T_2(u_0))+\alpha_2J_2(u_0,T_1(u_0),T_2(u_0)),\end{aligned} \tag{57}$$

则得到的结果为约束的 Pareto 最优解 $(u_0^s,T_1(u_0^s),T_2(u_0^s))$．

四、状态估计及随机控制

上面各节除了在个别情况外，基本上没有考虑系统中的随机扰动和测量噪声．如果把这些因素考虑进去就涉及递阶结构的状态估计和随机控制问题．目前在这方面的研究还仅限于关联系统的 LQG 问题，现作简单的介绍．

如果分离定理成立则最优随机控制问题可分为两步求解，首先对系统的状态进行最优估计，然后根据已知的系统的状态去求解一确定的最优控制问题．因此必须研究在分散控制的情况下分离定理是否成立．据称 Hassan 等人已经对一类分散的线性离散系统建立了分离定理[10]，但作者还未见到他们的论文．最近 Yoshikawa 论述了分散随机控制分离定理存在的必要和充分条件[18]．

由于确定的递阶最优控制已有很多种算法，因此如分离定理成立则求解最优随机控制问题的重点必然是分散控制系统的最优状态估计．在本节我们将介绍求解状态估计问题的两种方法．

一种对关联系统进行状态估计的方法是采用递阶结构对各子系统的测量子空间逐步

进行正交化以获得最优估计[19]. 这种方法的理论基础是随机变量空间的正交射影定理.

设关联系统第 i 个子系统的状态方程为

$$x_i(k+1) = \Phi_{ii}x_i(k) + \sum_{i=1,\ i\neq j}^{N} \Phi_{ij}x_j(k) + w_i(k), \tag{58}$$

$$y_i(k+1) = H_i x_i(k+1) + v_i(k+1), \quad i = 1, \cdots, N, \tag{59}$$

式中 w_i 及 v_i 为不相关的零均值高斯白色噪声序列，y_i 为被测量的输出. 状态 x_i 在 $k+1$ 时刻的线性无偏最小方差估计为

$$\begin{aligned}\hat{x}_i(k+1|k+1) &= E\{x_i(k+1)|\mathscr{Y}(k+1)\} \\ &= E\{x_i(k+1)|\mathscr{Y}(k), y_1(k+1), \cdots, y_N(k+1)\},\end{aligned} \tag{60}$$

式中 $\mathscr{Y}(k)$ 及 $\mathscr{Y}(k+1)$分别为整个系统直到 k 及 $k+1$ 时刻的所有的测量，$y_i(k+1)$ $(i=1,\cdots,N)$ 为第 i 个子系统在 $k+1$ 时刻的测量. 在这里假设了每个子系统能得到整个系统过去所有时刻的全部输出信息 $\mathscr{Y}(k+1)$.

运用随机变量在希尔伯特空间中正交射影的概念可以推得

$$\begin{aligned}\hat{x}_i(k+1|k+1) = E\{x_i(k+1)|\mathscr{Y}(k), y_1(k+1), \cdots, y_{N-1}(k+1)\} \\ + E\{x_i(k+1)|\tilde{y}_N^{N-1}(k+1|k+1)\},\end{aligned} \tag{61}$$

式中 $\tilde{y}_N^{N-1}(k+1)$ 为

$$\begin{aligned}\tilde{y}_N^{N-1}(k+1|k+1) &= y_N(k+1) \\ &\quad - E\{y_N(k+1)|\mathscr{Y}(k), y_1(k+1), \cdots, y_{N-1}(k+1)\}.\end{aligned} \tag{62}$$

这就是说 x_i 的最优估计 $\hat{x}_i(k+1|k+1)$ 可以作为两个估计之和，其中第一个估计是 $x_i(k+1)$ 在由测量 $\mathscr{Y}(k)$、$y_1(k+1),\cdots,y_{N-1}(k+1)$ 所产生的子空间的正交射影，第二个估计是 $x_i(k+1)$ 在由 $\tilde{y}_N^{N-1}(k+1|k+1)$ 所产生的子空间的正交射影. 按照 (62) 式，后面这个子空间又是由 $y_N(k+1)$ 减去 $y_N(k+1)$ 在

$$\mathscr{Y}(k), y_1(k+1), \cdots, y_{N-1}(k+1)$$

所产生的子空间的正交射影所产生，因此 $\tilde{y}_N^{N-1}(k+1|k+1)$ 所产生的子空间和由

$$\mathscr{Y}(k), y_1(k+1), \cdots, y_{N-1}(k+1)$$

所产生的子空间彼此正交. 按照类似的推理可得

$$\begin{aligned}\hat{x}_i(k+1|k+1) = E\{x_i(k+1)|\mathscr{Y}(k)\} + E\{x_i(k+1)|\tilde{y}_1(k+1|k)\} \\ + \sum_{r=2}^{N} E\{x_i(k+1)|\tilde{y}_r^{r-1}(k+1|k+1)\}.\end{aligned} \tag{63}$$

由 (63) 式，$x_i(k+1)$ 的最优估计 $\hat{x}_i(k+1|k+1)$ 可以采用递推算法来计算. 这就是在计算 $E\{x_i(k+1)|\mathscr{Y}(k)\}$ 之后，顺次计算

$$E\{x_i(k+1)|\tilde{y}_1(k+1|k)\}, \quad E\{x_i(k+1)|\tilde{y}_2^1(k+1|k+1)\}$$

等等. 由此可推得第 i 个子系统 Kalman 滤波的递推表示式为

$$\hat{x}_i(k+1|k+1)_l = \hat{x}_i(k+1|k+1)_{l-1} + K_{il}^{l-1}(k+1)\tilde{y}_l^{l-1}(k+1|k+1), \tag{64}$$

式中 l 系从 1 开始到 N，校正项的系数由另一组递推表示式计算.

采用这种算法各个子系统的状态估计可以独立地进行，因而可采用多处理机的系统. 在任何时刻 k 整个系统的测量由协调计算机所收集并由它送给各子系统的计算机，各子

系统即独立地进行状态估计直到时刻 $k+1$ 得到新的测量.

据称采用这种算法较之全局的 Kalman 滤波算法在计算机的存储量及计算时间等两个方面都有很大的节省.

在估计了状态之后，根据分离定理去求得 LQG 问题的最优控制是一个很容易的问题,可以套用确定的最优控制的结果,见文献[10].

Hassan 等解决大系统的状态估计问题的另一种方法[20]是根据 Kalman 提出的对偶理论[21]把一最优估计问题化为一最优调节器问题求解. 设

$$\dot{\boldsymbol{x}} = F(t)\boldsymbol{x}(t) + G(t)\boldsymbol{w}(t) \tag{65}$$

$$\boldsymbol{y} = H(t)\boldsymbol{x}(t) + \boldsymbol{v}(t) \tag{66}$$

其中 $\boldsymbol{w}$ 及 $\boldsymbol{v}$ 假设为不相关的零均值高斯白色噪声，其协方差矩阵分别为 Q 及 R. 初始状态的协方差为 P.

根据对偶原理,上述状态估计问题可化为如下的对偶控制问题求解:

$$\dot{\boldsymbol{x}}^* = A(t^*)\boldsymbol{x}^*(t^*) + B(t^*)\boldsymbol{u}^*(t^*) \tag{67}$$

$$\boldsymbol{y}^*(t^*) = C(t^*)\boldsymbol{x}^*(t^*) \tag{68}$$

式中

$$\left.\begin{aligned} &A^T(t^*) = F(t), \quad C^T(t^*) = G(t), \\ &B(t^*) = H(t), \quad t^* = -t. \end{aligned}\right\} \tag{69}$$

其转移矩阵为

$$\Phi^*(t^*, t_0^*) = \Phi^T(t_0, t) \tag{70}$$

最优调节器问题为在上述约束条件下使目标函数

$$J = \frac{1}{2}\|\boldsymbol{x}(t_0^*)\|_P^2 + \frac{1}{2}\int_{t^*}^{t_0^*}\{\|\boldsymbol{x}^*(\tau^*)\|_{Q(\tau^*)}^2 + \|\boldsymbol{u}^*(\tau^*)\|_{R(\tau^*)}^2\}d\tau \tag{71}$$

为极小. 由此得到的解 $\boldsymbol{x}^*$ 即最优状态估计 $\hat{\boldsymbol{x}}$.

把最优估计问题化为最优调节器问题以后,就可以采用通常的两级算法去求解,根据得到的状态的最优估计再去计算最优控制. Hassan 等运用这个方法求解了一个 52 阶的河道污染问题.

五、结　论

本文对递阶控制的几个理论问题作了综述. 从本文的介绍可以看出递阶控制的研究目前仍然是一个相当活跃的领域. 虽然递阶控制在理论上和方法上已经取得了很多的研究成果,并且也有一些用于解决实际问题的报道,但仍然有许多问题值得进一步探讨.

这个方法到目前为止基本上是离线的计算方法，这就降低了它的实用价值. 对于动态系统，如果能找到较好的在线计算方法，即使是次最优的，将有助于扩大它的应用范围.

目前对于大系统的随机控制问题还研究得不够充分,局限于 LQG 问题. 如果能找到一种方法能解决更广泛的随机控制问题,将是很有帮助的.

由于大系统的结构很复杂，因此事先准确地知道它的结构和参数是很困难的. 现在

还不清楚在递阶控制中能否采用模糊集理论？目前还缺乏这方面的报道.

模型简化的工作已经作得不少，而且卓著成效. 如何把这方面的成果，例如集结法和奇异摄动法用到递阶控制中来，从而使计算简化，是值得进一步研究的问题.

参 考 文 献

[1] G. Cohen, Optimization by decomposition and coodination, A unified approach, *IEEE Trans. Automat. Contr*, **AC-23**, No. 2, Apr. (1978).

[2] G. Grateloup and A. Titli, Two level dynamic optimization methods, *J. Optimiz. Theory and Appl.*, **15**, No. 5, (1975).

[3] M. S. Mahmoud et al., Multilevel control and optimization using generalized gradients technique, *Int. J. Contr.*, **25**, No. 4, (1977).

[4] G. Cohen, Comments on paper ''Multilevel control and optimization using gradients technique'', *Int. J. Contr.*, **27**, No. 2, Feb. (1978).

[5] M. Hassan et al., A three level costate prediction method for continuous dynamical systems, *Automatica*, **14**, No. 2, March, (1978).

[6] M. G. Singh, Hierarchical feedback control for large dynamic systems, *Int. J. Syst. Sci.*, **8**, No. 1, (1977).

[7] M. G. Singh and A. Titli, Closed-loop hierarchical control for non-linear systems using quasilinearlization, *Automatica*, **11**, No. 5, Sept. (1975).

[8] M. F. Hassan and M. G. Singh, The optimization of non-linear systems using a new two level method, *Automatica*, **12**, July, (1976).

[9] M. G. Singh and M. F. Hassan, A two level prediction algorithm for non-linear systems, *Automatica*, **13**, Jan. (1977).

[10] M. G. Singh and A. Titli, Practical hierarchical optimization and control algorithms, 7th Triennial World Congress of IFAC, June, (1978), 32A. 2.

[11] W. Findeisen et al., On-line hierarchical control for steady-state systems, *IEEE Trans. Automt. Contr.*, **AC-23**, No. 2, Apr. (1978).

[12] W. Findeisen and K. Malinowski, A structure for on-line dynamic coodination, *Large-Scale Systems Theory and Applications*, Udine, Italy, **16—20**, June, (1976).

[13] C. Y. Chong and M. Athans, On the periodic coodination of linear stochastic systems, *IEEE Trans. Automat. Contr.*, **12**, (1976).

[14] J. B. Cruz, Leader-follower strategies for multilevel systems, *IEEE Trans. Automat. Contr.*, **AC-23**, No. 2, Apr. (1978).

[15] J. B. Cruz, Stackelberg strategies for multilevel systems. ''Directions in large-scale systems'', ed. by Y. C. Ho and S. K. Mitter., Plenum Press, New York, (1975).

[16] C. I. Chen and J. B. Cruz, Stackelberg solution for two person games with biased information pattern, *IEEE Trans. Automat. Contr.*, **AC-17**, No. 6, (1972).

[17] J. Medanic and D. Radojevic, On multilevel Stackelberg strategies in linear quadratic systems, *J. Opt. Theory Appl.* **24**, 1978.

[18] T. Yoshikawa and H. Kobayashi, Seperation of estimation and control for decentralized stochastic control, 7th Triennial world Congress of IFAC, June (1978), 44. 3.

[19] M. F. Hassan et al., A decentralized computational algorithm for the global kalman filter, *IEEE Trans. Automat. Contr.*, **AC-23**, No. 2, Apr. (1978).

[20] M. F. Hassan et al., Stochastic hierarchical control of a large scale river system, 7th Triennial World Congress of IFAC, June (1978), 32A. 3.

[21] R. E. Kalman, New results in linear filtering and prediction theory, *Trans. ASME Series D. J. Basic Eng.*, **82**, No. 1, March (1960).

SEVERAL PROBLEMS OF HIERARCHICAL CONTROL

CHEN TING
(*Huachung Institute of Technology*)

ABSTRACT

In this paper, the (status of) several problems of hierarchical control are surveyed. The contents of this paper are as follows: In section I, the auxiliary problem and relaxation principle which are based on infinite-dimensional programming are discussed in detail. By these two principles, a unified apprach is formed, which can be used to derive several basic decomposition-coodination algorithms. In section II, the hierarchical method are used to obtain the closed-loop optimal control laws of linear and nonlinear systems; the on-line hierarchical control of the steady-state systems is also discussed. In section III, the Stackelberg strategies applied to hierarchical control are surveyed. In section IV, two algorithms of state estimation for hierarchical systems are derived.