

摘要

生产实践发展的需要催生了模型预测控制方法,模型预测控制理论发展的目的是更好地服务于生产实践。因此,如何把先进的模型预测控制算法应用于实际控制中是模型预测控制发展的根本性问题之一。论文正是从这一观念出发,在前人研究的基础上,对广义预测控制算法提出了一系列的简化实现方法。论文包括以下内容:

首先,介绍了模型预测控制方法的发展过程、现状、目前存在的局限性以及发展趋势。分析了广义预测控制算法的特点及应用时存在的困难。在总结模型预测控制方法常用简化实现方法和广义预测控制算法简化实现已取得的成果的基础上,提出论文所要进行的研究工作。

第二,对于物理可实现的多变量系统,其 CARIMA 模型的 $A(z^{-1})$ 与 $C(z^{-1})$ 多项式矩阵总可以构造成对角形式,从而可以给出这种模型的广义预测控制算法的完整求解过程。在算法的推导过程中,显式地考虑了过程纯滞后项,以提高计算效率。利用这种形式的模型结构,不但广义预测控制算法的求解过程可以得到很大程度的简化,而且相应的模型参数辨识问题也得到了简化,可以把一个多输入多输出模型的大型参数辨识问题分解成多个多输入单输出模型的小型参数辨识问题。

第三,通过对模型预测输出自由响应项的进一步分析,得到了状态反馈结构形式的广义预测控制器,控制增量等于控制器系数与设定值、过程输入输出历史数据的乘积,控制器系数只与模型参数和设计参数有关,控制器系数维数由预测时域与模型结构参数决定。消除了在非自适应模式下在线求解模型输出自由响应的必要,可以像 PID 控制器一样实现广义预测控制器。

第四,利用 CARIMA 模型直接递推得到了更加简洁的广义预测控制器,控制增量等于控制器系数与设定值、过程输入输出历史数据、模型预测误差历史数据的乘积,控制器系数只与模型参数和设计参数有关,控制器系数维数只由模型结构参数决定。在自适应模式下无需进行 Diophantine 方程的求解,在非自适应模式下将实现难度与计算量降到了最低。

第五,通过对多变量广义预测控制算法内在机理的分析,指出多变量广义预测控制算法实际上是一种函数映射,是从模型参数空间到控制器系数空间的映射,并利用神经网络的映射能力来实现多变量广义预测控制器系数的快速计算,

极大程度地降低了在线计算量，简化了算法的实现形式。

第六，在非线性液位装置上对以上简化实现的多变量广义预测控制算法进行了对比实验研究。实验结果证明了各个方法的可行性、有效性及等效性。

最后，对论文的内容进行了总结，讨论了各个简化方法的适应范围及在线实现时需要注意的问题，提出了有待进一步研究的问题。

关键词：广义预测控制，自适应控制，多变量控制，神经网络，快速算法

ABSTRACT

It is the development of industrial process control that has speeded up the emerging of model predictive control (MPC) methodology and the basic motivation of using MPC technology is to ensure significant economic benefits. Thus, one of the most essential problems of MPC technology development is how to implement these advanced algorithms in real world effectively. Just from this view, a set of simplified implementation for generalized predictive control (GPC) are presented based on the current framework of MPC. The dissertation is organized as follows:

First, a brief history of MPC technology development is presented, followed by its current features and limitations of existing technology and its future trends. The distinguishing features of GPC approach and its application difficulty are discussed. After summarizing the general ways to simplify the implementation of MPC and the achievements in simplifying implementation for GPC, the main research works are given.

Second, for most of the physical realizable processes, the matrices $C(z^{-1})$ and $A(z^{-1})$ of their Controlled Autoregressive Integrated Moving Average (CARIMA) model can always be diagonally constructed, so that the formulation of GPC can be developed in more detail while explicitly considering the dead time in order to improve the computational efficiency. This model structure greatly simplifies not only the development of the GPC but also its parameter identification which can be transformed into a set of multiple input single output model parameter identification problems.

Third, a state-feedback like controller of GPC is obtained by further manipulating the free response of the output predictor, whose control increment equals to the controller's coefficients multiplied by set-points and historical plant input and output data. The controller's coefficients are only determined by the model parameters and design parameters and its dimension is determined by the model structure parameters and predictive horizon, which eliminates the need to compute the free response on-line and makes the implementation of GPC controller as easy as that

of PID under the non-adaptive mode.

Fourth, a more concise GPC controller is obtained by directly manipulating the output predictor using the multivariable CARIMA model recursively, whose control moves are the product of the controller's coefficients and set-points, historical input/output data of the plant and predictive errors of the predictor. The controller's coefficients are determined only by the model parameters and design parameters and its dimension only depends on the orders of the model, which avoids solving Diophantine equations on-line under adaptive mode and reduces difficulties of implementing the GPC controller and the computational overhead to the lowest limit under the non-adaptive mode.

Fifth, it is pointed out that the multivariable GPC algorithm is essentially a kind of functional mapping from the multivariable process model parameters' space to the multivariable GPC controller's coefficients' space by analyzing its intrinsic mechanism. This mapping can be realized by BP neural network to obtain the GPC controller's coefficients from the model parameters directly, which can extremely reduce the computational overhead on-line and simplify the implementation of the GPC controller.

Sixth, the above schemes developed in this dissertation are compared by a set of contrast experiments on a nonlinear liquid level equipment. Their feasibility, validity and equivalency are demonstrated by experiment results.

Last, a summary is given to show what has been done in this paper. The applicable scope of these methods developed in this dissertation is discussed, followed by some items that must pay attention to when implement these methods and future potential research opportunities.

Keywords: generalized predictive control, adaptive control, multivariable control, neural networks, fast algorithm

致谢

攻读博士学位期间,王树青教授给予我的不仅仅是学术上的指导、生活上的关心,更多的是人生哲理上的点拨。很多的时候,人生的成功不只是依赖于我们的学识,更多地是成就于我们的人生观与世界观。求真、务实、乐观、宽容、诚挚、上进,是王树青教授传授于我的人生之道和求学之本。我不但要衷心感谢王树青教授对我的谆谆教诲,还要感谢王树青教授给我创造的各种机会,使我在多方面得到锻炼和发展。

同时,我也得到感谢王宁教授、张建明老师在学业与生活方面的长期指点与帮助。

实验过程中,得到苏宏业教授、徐巍华老师、冯毅萍老师及张克勤、张晓宇、顾海杰、章建栋、王旭、靳碧、刘进峰、张艳辉等师兄弟的热情帮助与支持,在此一并表示感谢。

在论文的撰写过程中得到许多师弟师妹的大力帮助,特别是智力师妹校对了大量的图片与文字,张柯师弟统计了矩阵求逆和正切函数计算的运算量,潘彩霞师妹阅读了部分文稿。感谢赵小强、李荣雨、朱炜等等师弟帮我打印了大量的论文。

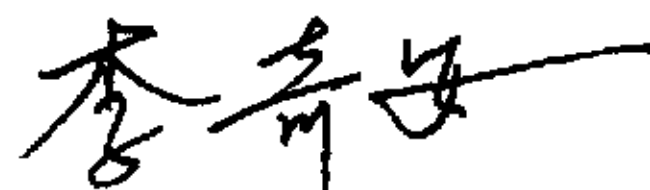
另外,感谢陈良师弟协助我完成许多有意义的工作,使我节省了大量的时间与精力。

感谢张智焕、王林、常宾宝、郭明、郭敏强、何宁、谢磊、蒋丽英、周韵园、祝雪妹、洪伟荣、张惠良、黄海、刘峙飞、韩安太、刘益剑、陈亚华、邓世普、王鹤曾、郑重、张丽、陈庆更、卢帅、杨项方、徐伟民、苏成利、张日东、董胜利、陈刚、杜鹃、陈雪丽、黄亮、陈铭、黄少锋、陶吉利、张奕、徐业健、阚晓旭、杜亚萍、徐志诚、孙世国、徐丰、曾建武、谢懿等等师兄姐妹们及来国妹女士在共同相处的日子里带来的欢乐与开心。

感谢徐国良老师、梁军老师及其他老师的指点与帮助

感谢 2001 秋博、2001 支部的同窗及其他朋友的帮助与支持。

当然,我也要感谢家人一直以来对我学业的支持与理解。



2005 年 5 月于浙大求是园

第一章 绪论

摘要：回顾了模型预测控制方法的发展历程，介绍了模型预测控制方法的发展现状、发展趋势和模型预测控制常用的简化实现方法。综述了广义预测控制算法及简化实现方法的研究现状。指出了论文对广义预测控制算法所要进行的改进工作。

1.1 引言

模型预测控制，Model Predictive Control (MPC)，是一类计算机控制算法，其核心特征是利用过程模型预测对象的未来行为。在每个控制周期，通过计算一系列控制作用来优化对象的未来行为，并把优化得到的控制序列的第一个输出信号作用于过程，在下一个控制周期重复所有的计算过程。起初，模型预测控制主要是为了满足炼油、电力等部门控制的需要，现在已广泛应用于化工、冶金、轻工等领域。

三十多年的研究与发展，模型预测控制已取得了丰富的理论成果，并在工程领域占有了一席之地，文献[1]-[25]对模型预测控制的理论发展与实际应用进行了全面的介绍与综述，另外许多模型预测控制方面的专著^{[26]-[31]}对模型预测控制的机理、方法及应用实例进行了详尽的阐述。

大量的文献显示越来越多的研究者与工程人员对模型预测控制的兴趣正以惊人的速度增长，面对如此迅猛的发展态势，人们不禁要问：模型预测控制到底解决什么问题？是否还存在不足？将来的发展方向在哪里？

要回答这个问题，就必须对模型预测控制的起源与发展过程有一个清晰的了解。模型预测控制起源于工程应用的需要，其归宿也在于更好地服务于生产实践，因此从应用的角度来分析模型预测控制更有现实意义。工业化模型预测控制技术几乎囊括了模型预测控制方法的所有特点，并一直走在模型预测控制发展的最前列，它的发展历程就是模型预测控制发展的真实写照。因此，下面就从工业化模型预测控制技术发展的角度来分析模型预测控制的发展。

1.2 模型预测控制方法的发展历程

在 2003 年发表的一篇关于工业化模型预测控制技术综述中, Qin 和 Badgwell 全面地总结了工业化模型预测控制技术发展的历史及趋势^[25], 其分析基于对现有知名模型预测控制技术厂商所作的调查, 很有借鉴意义。在这篇文献中, 作者把线性模型预测控制技术划分成图 1.1 所示的几个发展阶段, 并对每一代的特征进行了说明。

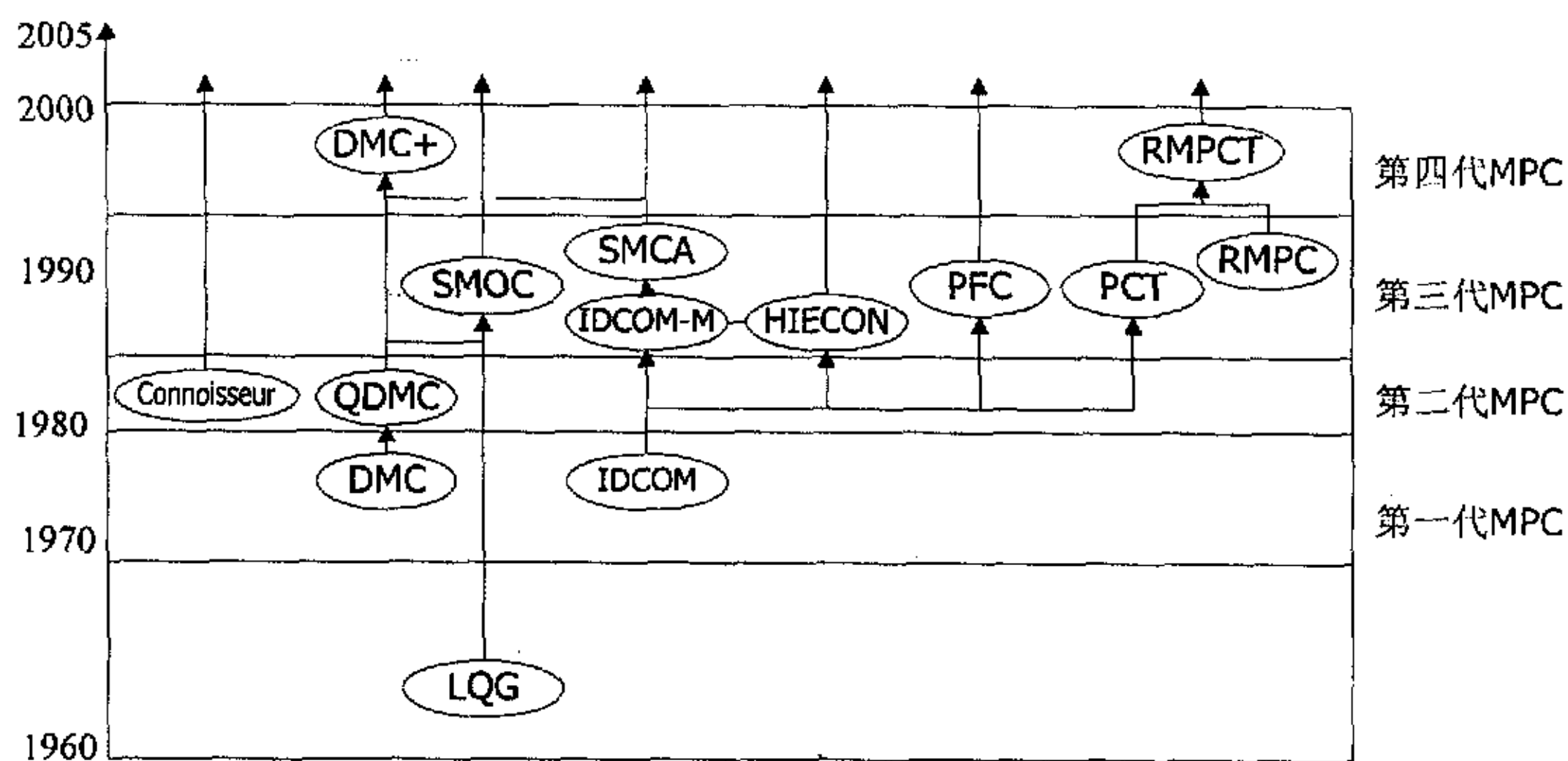


图 1.1 线性模型预测控制技术发展阶段

1.2.1 模型预测控制方法发展参照点

1960 年由 Kalman 等人提出的线性二次高斯控制(linear quadratic Gaussian (LQG))方法^{[32]-[33]}标示着现代控制理论的到来。LQG 采用无限预测时域, 具有优异的稳定性能, 对任意有理的线性对象, 在模型准确的情况下, 只要选择合适的加权矩阵, 就可以获得稳定的控制。在许多应用领域, LQG 已成为解决控制问题的标准方法^[34]。但是, LQG 对过程工业控制技术的发展影响很小, 主要原因是^{[2][35]}:

- 不能处理约束: 众所周知, 过程单元的经济工作点一般落在约束条件交叉区域内^[36], 因此一个成功的工业过程控制器必须使过程尽可能地工作在约束边界值附近, 但不要违背约束条件。而 LQG 没有考虑过程输入、状态及输出的约束。
- 不适合非线性过程: 实际的工业过程一般是复杂、非线性、受约束的多

变量系统，工作点变化及催化剂老化等类似原因会造成其动态特性随时间而变化。

- 鲁棒性差：每个过程单元的特性各异，要想从物理化学的角度建立统一的机理模型是不太合理的。当然有的领域，如航空工业，对象的特征可由物理系统确定，建立准确的机理模型是经济可行的。正是在这样的领域内，LQG 才得到了充分的发展。
- 性能指标单一：有些过程单元的性能指标很独特，其输出权系数是时变的，或者需要加入逻辑判断以描述不同的工作模式，很难表达成 LQG 单一的目标函数形式。
- 不符合当时的文化背景：影响 LQG 在过程工业成功运用的最重要的原因可能是当时的仪表技术员和控制工程师要么不了解 LQG 的概念，要么认为其不实用。

在这种情形下，为了满足复杂工业过程控制的需要，产生了一种建模与控制的新方法，即模型预测控制技术。这也是为什么把 LQG 作为模型预测控制方法发展参照点的原因。

1.2.2 第一代模型预测控制方法

尽管 1963 年 Propoi 就提出了滚动时域控制的方法^[37]，1967 年 Lee 和 Markus 在其最优控制教科书中也预言了如今 MPC 的实际形式^[38]，但直到 1976 年在第四届 IFAC 辨识与系统参数估计讨论会上 Richalet 等人才首次描述了 MPC 的应用^[35]。在 1978 年发表的论文中，Richalet 等人对这种 MPC 方法进行了总结，称之为模型预测启发式控制(Model Predictive Heuristic Control (MPHC))，其实现软件称为：IDCOM (Identification and Command)。IDCOM 的显著特征为：

- 使用有限脉冲响应(Finite Impulse Response (FIR))模型，输入变量与中间变量为线性
- 使用有限预测时域二次性能目标函数
- 使用参考轨迹来指定输出的未来行为
- 算法中包含了对输入输出约束的描述
- 使用启发式递推算法求解最优控制量，它是辨识问题的对偶解

Richalet 等人认为要想进行有效的控制，就必须采用功能分层的控制结构，并提出了一种四层结构控制模式：

- 第三层：生产的时间与空间调度层
- 第二层：设定值优化层，其目标是减少成本，确保产品的数量与质量

- 第一层：多变量动态控制层

- 第零层：常规控制层

他们指出：仅仅在第一层使用良好的动态控制减少被控变量的方差，并不会带来巨大的利益，真正的经济效益来自第二层对设计值的优化。这种分层控制的概念是应用先进控制的一个重要组成部分，并被许多实践工作者所沿用。

在二十世纪七十年代早期，Shell Oil 的工程师独立开发了动态矩阵控制 (Dynamic Matrix Control (DMC)) 技术，并在 1973 初次投入了使用。在 1979 年 National AIChE 会议与 1980 年 Joint Automatic Control 会议上，Culter 与 Ramaker 对 DMC 进行了详细说明^{[39][40]}。1980 年，Prett 和 Gillette 合作发表的论文中，描述了改进的 DMC 算法在 FCCU (fluid catalytic cracking unit) 反应/再生中的应用^[36]。DMC 控制算法的主要特征包括：

- 使用线性阶跃响应模型
- 使用有限预测时域二次目标函数
- 通过参考轨迹指定对象的未来输出尽可能地靠近设定值
- 优化方法使用最小方差求解方法

最初的 IDCOM 和 DMC 算法代表了 MPC 的第一代，他们定义了工业 MPC 方法的理论框架，对过程工业的控制产生了巨大影响。

1.2.3 第二代模型预测控制方法

起初的 IDCOM 和 DMC 对没有约束的多变量过程能获得非常优异的控制效果，但却需要采取一些特殊的方式来处理约束。Shell Oil 公司的工程人员把 DMC 算法构造成一个二次规划问题 (quadratic program (QP))，即 QDMC 算法，克服了 DMC 不能显式地处理输入输出约束的弱点。Culter 等人在 1983 年的一篇会议论文中首次对 QDMC 算法进行了描述^[41]。几年后，Garcia 和 Morshedi 在其发表的论文中进行了更详尽的表述^[42]。

QDMC 的关键特征包括：

- 使用线性阶跃响应模型
- 使用有限预测时域二次目标函数
- 引入输入增量抑制
- 采用二次规划

QDMC 算法代表了 MPC 技术的第二个发展阶段，能够系统地处理输入输出约束。

1.2.4 第三代模型预测控制方法

随着 MPC 技术日益被人们所接受, MPC 解决的问题变得越来越大、越来越复杂, 第二代 MPC 方法又遇到了许多实际问题。尽管 QDMC 算法能够系统地处理输入输出约束, 但对于不可行解的问题并没有明确的解决方案。实际过程中, 由于硬件故障、阀位饱和、或者操作人员的直接介入操作都可能会造成输入输出变量丢失, 致使被控制对象的结构发生变化, 那么怎样处理由此引发的容错控制问题呢? 另外, 对于大型复杂的控制问题, 要把众多的控制要求在一个单一的目标函数中表达成相对的权重并不是一件容易的事。这些问题激励着工程师们不断开发新的 MPC 算法。Setpoint 公司推出了 IDCOM-M(Multivariable), 而 Adersa 公司推出的 HIECON(hierarchical constraint control)与其具有相似的功能。IDCOM-M 的特征为^{[43][44]}:

- 使用线性脉冲响应模型
- 具有可控制性监控程序, 可以实时监视子系统的恶化情况
- 使用多个目标函数, 包括二次输出目标函数与二次输入目标函数
- 引入拟合点的概念, 只要求输出变量在预测控制时域的一些点上与参考轨迹一致
- 在每个控制周期只计算输入变量的一个增量信号
- 可以考虑硬约束与软约束, 硬约束赋予优先级

在此基础上, Setpoint 的工程师不断改进 IDCOM-M 技术, 最终把辨识、仿真、整定和控制产品融合为一个整体—SMCA(Setpoint Multivariable Control Architecture)。

二十世纪八十年代末期, Shell Research in France 的工程师们结合 LQG 与 DMC 的思想开发出了 SMOC(Shell Multivariable Optimizing Controller)^{[45][46]}, 它既可以利用 MPC 处理约束的能力, 又可以利用状态反馈控制丰富的理论成果。SMOC 的许多特征被认为是现代 MPC 算法不可缺少的组成部分:

- 使用状态空间模型, 可以描述稳定、不稳定和积分对象
- 使用扰动模型来描述不可测量的干扰, 常值扰动只是其特例
- 采用 Kalman 滤波器估计状态与不可测量扰动
- 区分被控变量与反馈变量: 被控制变量出现在目标函数中; 反馈变量用于状态估计
- 采用二次规划优化方法

IDCOM-M、HIECON、SMCA、SMOC 和 Profimatics 公司的 PCT 以及 Honeywell 公司的 RMPC(Robust MPC)等产品一起构成了 MPC 技术的第三代。

这一代 MPC 技术对约束条件进行了区分(可考虑硬约束、软约束、等级约束),提出了一些处理不可行解的方案,考虑到了控制结构实时变更的问题,可以选择不同形式的反馈方式,适应稳定、积分和不稳定的过程,并可采用多种形式的目标函数。

1.2.5 第四代模型预测控制方法

1995 年末 Honeywell 公司收购了 Profimatics 公司,组成了 Honeywell Hi-Spec Solutions,原来 Honeywell 的 RMPCT 和 Profimatics 的 PCT 融合成为 RMPCT。1996 年初 Aspen Technology 公司收购了 Setpoint 公司与 DMC 公司,1998 年又收购了 Treiber Controls 公司,把 SMCA 与 DMC 技术合为一体,构成了现在的 DMC-plus。RMPCT 与 DMC-plus 代表了 MPC 技术的第四代,它们的特征是:

- 基于 Windows 的图形用户界面
- 采用多层优化,以实现不同等级的控制目标
- 采用灵活的优化方法
- 采用鲁棒设计方法,直接考虑模型的不确定性
- 采用改进的辨识技术

以上发展阶段的划分只是相对的,主要参照文献[25]的调查结果,尽管还有许多的预测控制算法及软件没有包括在讨论中,但这并不影响对模型预测控制方法发展阶段的认识。当然各代之间并不是绝对分开的,在同一时期往往是各种方法交错并存。

1.3 模型预测控制方法的现状

RMPCT、DMC-plus 与其它现有的 MPC 产品一起代表着当前模型预测控制方法的研究与发展现状。

1.3.1 模型预测控制在全厂控制中的位置

在现代过程工业控制中, MPC 是多层控制结构中的一部分,其典型位置如图 1.2 所示^{[47][48][25]}。图 1.2 左边单元 1 是常规控制结构,右边单元 2 是 MPC 控制结构。最顶层是工厂级全局优化器,决定厂内各个单元的最优稳态设置,这些设置送入下一级单元局部优化器。单元级优化器计算出来的最优经济稳态值传递

到动态约束控制系统去实现。动态约束控制系统把装置从一个约束稳态控制到另一个约束稳态。在常规控制结构中这些功能是通过 PID 算法模块、超前滞后模块及逻辑判断模块的组合来实现的，而在 MPC 结构中这些组合模块由 MPC 控制器所代替。因为 MPC 能整体地考虑各种因素，所以能够真正实现常规组合模块所不能实现的控制要求。

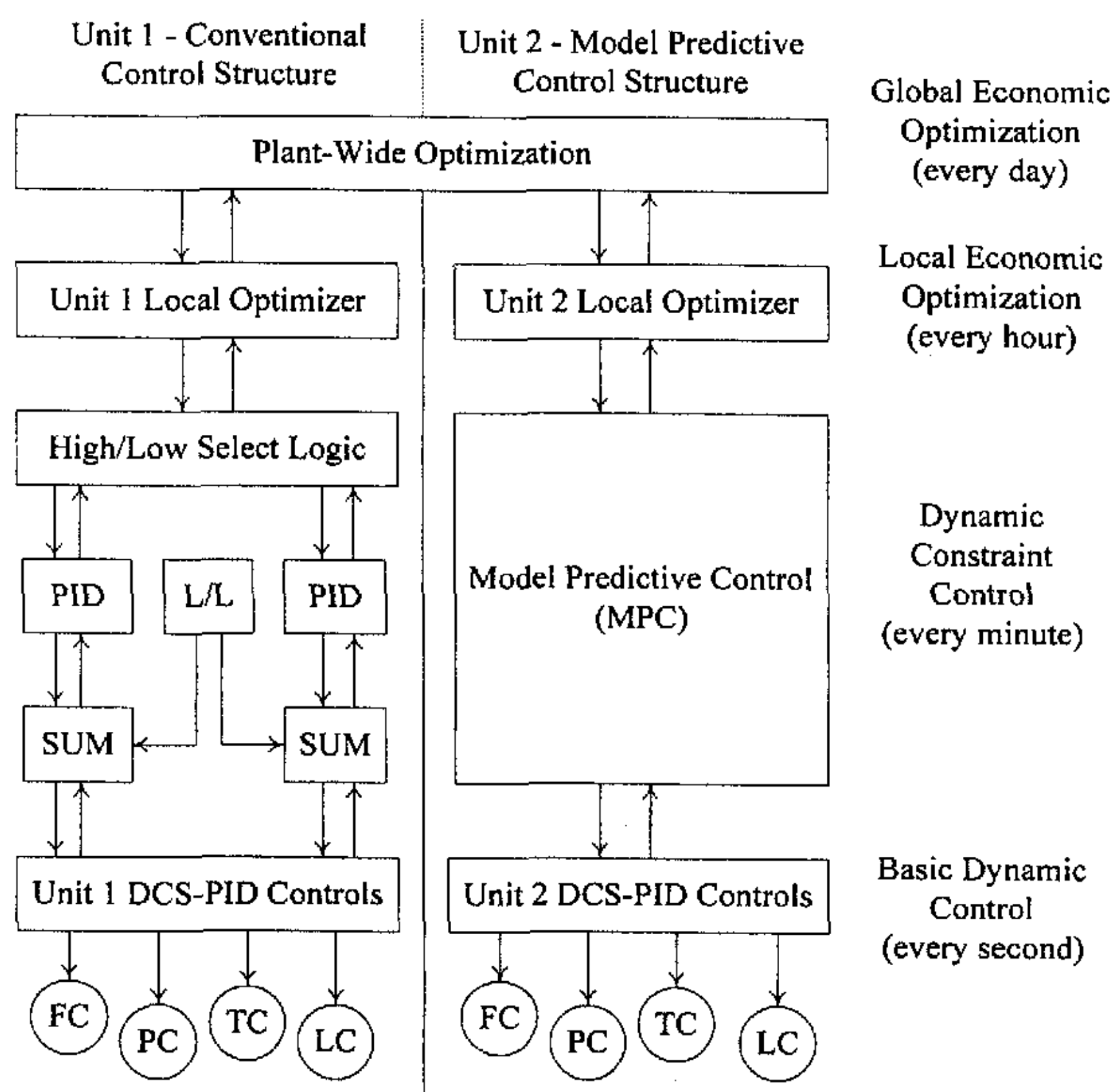


图 1.2 典型过程工业控制系统功能结构

1.3.2 模型预测控制器所要实现的目标

设计 MPC 控制器的目的是把过程从一个受约束优化稳态转移到另一个受约束优化稳态，因此 MPC 控制器所要达到的目标按重要性排列为：

1. 阻止输入输出变量违背约束条件
2. 把输出变量控制在稳态最优值(动态输出优化)
3. 在允许的自由度下把输入变量控制在最优稳态值(动态输入优化)
4. 防止控制变量增速过快
5. 当信号丢失或者控制器失灵时，尽可能控制好剩下的子系统

为实现这些目标，在每一个控制周期内，MPC 要完成如图 1.3 所示的计算步

骤。首先是测量过程输入变量与输出变量的值，利用测量值附带的信息分析传感器的状态和下层控制器与执行器的运行状态。然后进行反馈校正、稳态值计算和动态优化。

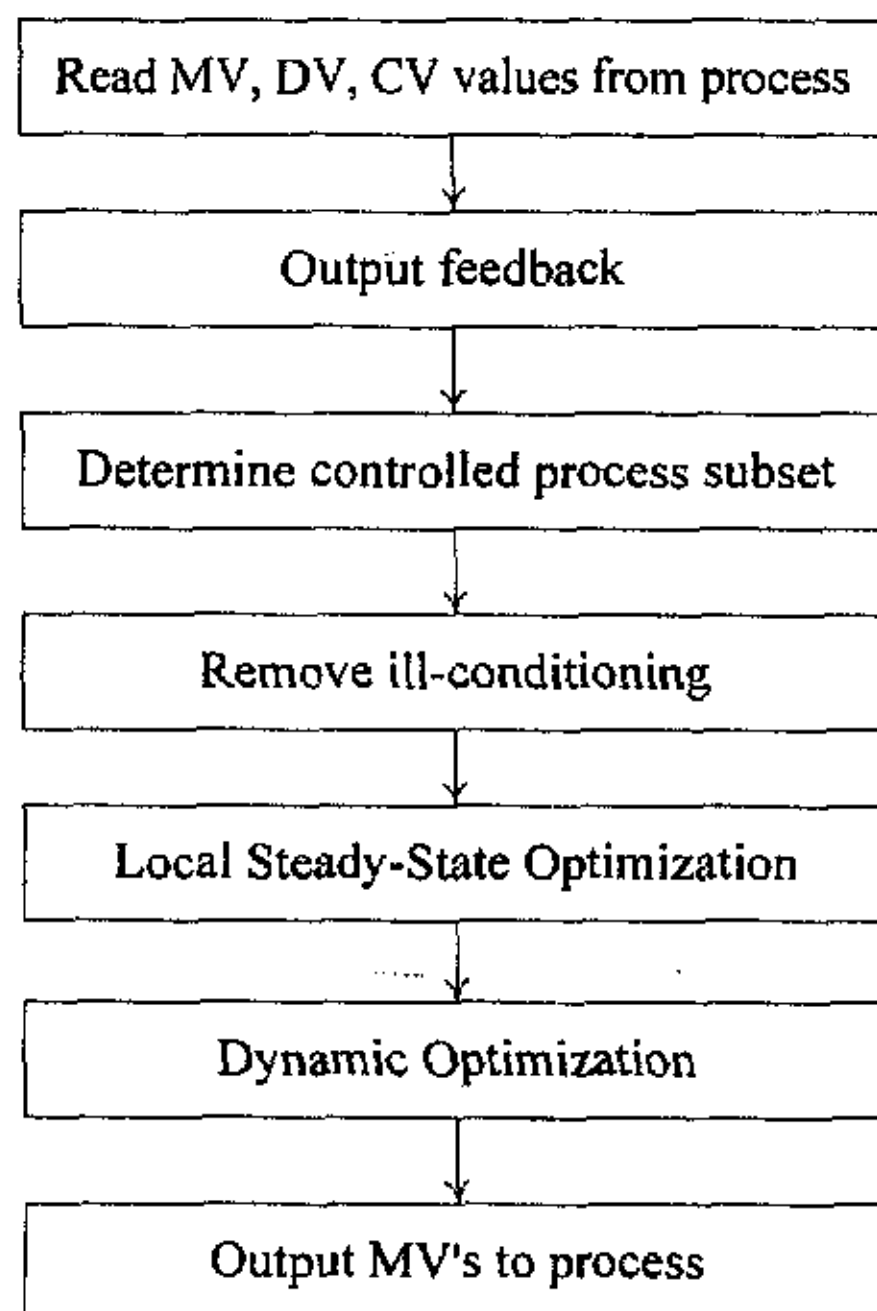


图 1.3 每个控制周期 MPC 的计算流程

在每个计算步上，各 MPC 产品都有其自身的实现方式。下面的部分对当前主流的 MPC 产品的相应实现方法进行简要的介绍。

1.3.3 当前 MPC 产品的主要特征

随着 MPC 技术的日益成熟，市场上涌现出了各种各样的线性 MPC 产品和非线性 MPC 产品。表 1.1 和表 1.2 分别列出了当前有代表性的 MPC 产品。

表 1.1 线性 MPC 产品

公司	产品名称	描述
Adersa	HIECON	Hierarchical constraint control
	PFC	Predictive functional control
Aspen Tech	DMC-plus	Dynamic matrix control package
Honeywell Hi-Spec	RMPCT	Robust model predictive control technology
Shell Global Solutions	SMOC-II	Shell multivariable optimizing control
Invernsys	Connoisseur	Control and identification package

表 1.2 非线性 MPC 产品

公司	产品名称	描述
Adersa	PFC	Predictive functional control
Aspen Tech	Aspen Target	Nonlinear MPC package
Continental Controls, Inc.	MVC	Multivariable control
DOT Products	NOVA-NLC	NOVA nonlinear controller
Pavilion Technologies	Process Perfecter	Nonlinear control

表 1.3 和表 1.4 分别列出了线性 MPC 与非线性 MPC 产品各自的实现形式及相关特征。

表 1.3 线性 MPC 对比表

公司	Aspen Tech	Honeywell Hi-Spec	Adersa	Adersa	Invensys	SGS
产品	DMC-plus	RMPCT	HIECON	PFC	Connoisseur	SMOC
模型形式 ^[a]	FSR	ARX,TF	FIR	LSS,TF,ARX	ARX,FIR	LSS
	L,S,I,U	L,S,I,U	L,S,I	L,N,S,I,U	L,S,I,U	L,S,I,U
反馈校正 ^[b]	CD,ID	CD,ID	CD,ID	CD,ID	CD,ID	KF
可控性监控 ^[c]	IMS	SVT	—	—	IMS	IMS
稳态优化目标 ^[d]	L/Q[I,O],...,R	Q[I,O]	—	Q[I,O]	L[I,O]	Q[I,O],R
稳态优化约束 ^[e]	IH,OS,R	IH,OH	—	IH,OH	IH,OH	IH,OS
动态优化目标 ^[f]	Q[I,O,M],S	Q[I,O]	Q[O],Q[I]	Q[I,O],S	Q[I,O,M]	Q[I,O]
动态优化约束 ^[g]	IH	IH,OS	IH,OH,OS,R	IA,OH,OS,R	IH,OS,R	IH,OS
输出轨迹 ^[h]	S,Z	S,Z,F	S,Z,RT	S,Z,RT	S,Z	S,Z,RTB,F
输出时域 ^[i]	FH	FH	FH	CP	FH	FH
输入参数化 ^[j]	MMB	MM	SM	BF	MMB	MMB
求解方法 ^[k]	SLS	ASQP	ASQP	LS	ASQP	ASQP
参考文献	[39][49]	[50]	[51]	[51]		[45]

表 1.4 非线性 MPC 对比表

公司	Adersa	Aspen Technology	Continental Controls	DOT Products	Pavilion Technologies
产品	PFC	Aspen Target	MVC	NOVA NLC	Process Perfecter
模型形式 ^[a]	NSS	NSS-NNN	SNP-ARX	NSS-FP	NNN-ARX
	S,I,U	S,I,U	S	S,I	S,I,U
反馈校正 ^[b]	CD,ID	CD,ID,EKF	CD	CD	CD,ID
可控性监控 ^[c]	—	IMS	IMS	IMS	—
稳态优化目标 ^[d]	Q[I,O]	Q[I,O]	Q[I,O]	—	Q[I,O]
稳态优化约束 ^[e]	IH,OH	IH,OH	IH,OS	—	IH,OH,OS
动态优化目标 ^[f]	Q[I,O],S	Q[I,O,M]	Q[I,O,M]	(Q,A)[I,O,M]	Q[I,O]
动态优化约束 ^[g]	IA,OH,OS,R	IH,OS-11	IH,OS	IH,OH,OS	IH,OS
输出轨迹 ^[h]	S,Z,RT	S,Z,RT	S,Z,RT	S,Z,RTUL	S,Z,TW
输出时域 ^[i]	CP	CP	FH	FH	FH
输入参数化 ^[j]	BF	MM	SM	MM	MM
求解方法 ^[k]	NLS	QPKWIK	GRG2	NOVA	GRG2
参考文献	[51]	[52]-[56]	[57]-[60]	[61][23]	[62]-[66]

表 1.3 与表 1.4 的内容引自文献[25]，现对其进行简单的注解：

- [a] 模型形式: Finite impulse response (FIR), finite step response (FSR), Laplace transfer function (TF), linear state-space (LSS), auto-regressive with exogenous input (ARX), linear (L), nonlinear (N), stable (S), integrating (I), unstable (U), Input-output (IO), first-principles (FP), nonlinear state-space (NSS), nonlinear neural net (NNN), static nonlinear polynomial(SNP).
- [b] 反馈校正: Constant output disturbance (CD), integrating output disturbance (ID), Kalman filter (KF), extended Kalman filter (EKF).
- [c] 可控性监控: Singular value thresholding (SVT), input move suppression (IMS).
- [d] 稳态优化目标: linear (L), quadratic (Q), inputs (I), outputs (O), multiple sequential objectives (...), outputs ranked in order of priority (R).
- [e] 稳态优化约束: Input hard maximum, minimum, and rate of change constraints (IH), output hard maximum and minimum constraints (OH), constraints ranked in order of priority (R).
- [f] 动态优化目标: Quadratic (Q), inputs (I), Outputs (O), input moves (M), sub-optimal solution (S), one norm (A).
- [g] 动态优化约束: Input hard maximum, minimum and rate of change constraints (IH), IH with input acceleration constraints (IA), output hard maximum and minimum constraints (OH), output soft maximum and minimum constraints (OS), constraints ranked in order of priority (R), output soft constraints with l_1 exact penalty treatment (OS-11).
- [h] 输出轨迹: Setpoint (S), zone (Z), reference trajectory (RT), RT bounds (RTB), funnel (F), upper and lower reference trajectories (RTUL), trajectory weighting (TW).
- [i] 输出时域: Finite horizon (FH), coincidence points (CP).
- [j] 输入参数化: Single move (SM), multiple move (MM), MM with blocking (MMB), basis

functions (BF).

- [k] 求解方法: Least squares (LS), sequential LS (SLS), active set quadratic program (ASQP). Nonlinear least squares (NLS), multi-step Newton method (QPKWIK) generalized reduced gradient (GRG), mixed complementarity nonlinear program (NOVA).

1.3.4 当前 MPC 方法的局限性

从图 1.1 看, 大部分现存的 MPC 产品都是在第一代 MPC 方法 IDCOM 和 DMC 的基础上发展而来, 它们的许多局限性也就沿袭至今^{[5][6][25]}.

- 模型形式简单: DMP-plus 和 HIECON 使用卷积模型, 当被控制对象的时间常数变化很大时, 使用这些模型会带来许多麻烦。有时为了兼顾模型的长度, 只得牺牲控制的快速性。脉冲响应模型与阶跃响应模型一个致命的弱点是只适合于严格稳定的过程, 尽管可以通过修改算法来适应纯积分过程, 但这样做会引发别的问题, 例如, 可能会在反馈通道引入噪声信号的导数。另外, 脉冲响应模型无法描述不稳定的过程。如果使用自回归参数模型, 这些问题便会迎刃而解。
- 反馈方式欠佳: 除 Aspen Target 和 SMOC 使用卡尔曼滤波器作为反馈校正外, 其它的 MPC 产品都把模型预测偏差值作为校正量。使用偏差校正方法是假定系统没有随机误差扰动, 也没有测量噪声。对于稳定对象这是可行的, 但对于纯积分过程和有测量噪声的系统, 控制效果就不容乐观。如果能对噪声和扰动进行仔细的分析, 建立合理的模型, 就会得到更加满意的反馈校正方法。
- 缺乏标称稳定性: 当有约束存在时, 即使模型准确, 要想得到稳定的控制, 正确整定控制器参数也不容易的。一般是在控制器投运之前, 采用仿真的办法来进行参数调整。但这种方法的缺点是不能全面地考虑所有可能出现的情形。最好的方法是发展能确保标称稳定性的算法, 以保证在所有可能的状态下系统都是稳定的。已有一些理论上的探讨^{[20][68][69]}, 但在工业 MPC 产品中却很少包含这些思想。
- 动态优化效率不高或者采用次优化方法: 为了加快求解速度, 一般会采用次优化方法。对于快速系统的控制, 这可能是一种无奈的选择, 因为要在很短的时间内完成二次规划不太可行。如果采样周期很长, 计算资源不受限, 采用次优化方法就不是很合理了。文献[70]指出, 如果能合理利用优化问题的结构, 就可以实现更快的控制, 解决更大维数的控制问题。

1.4 模型预测控制方法的发展趋势

1.4.1 模型预测控制的发展方向

Qin 和 Badgwell 在咨询 MPC 技术供应商的基础之上^[25], 结合 Froisy 的观点^[7], 指出了未来 MPC 发展方向:

1. 未来 MPC 控制器的基本架构: 因为要把所有控制目标表达成一个单一的目标函数很困难, 未来的 MPC 算法将使用多目标函数。无限预测时域具有很好的理论特征, 可能会成为 MPC 的一个标准要求。输入与输出轨迹的形式将包括设定值、区域、参考轨迹及“漏斗型”区域等多种选择。输入参数化将会更多地采用基函数的形式, 无限控制时域亦将成为可能。
2. 自适应 MPC: 尽管实际应用中很欢迎自校正 MPC 控制器, 但只有 Invensys 的 Connoisseur 和 Dot Products 的 STAR 是采用自适应方案的商业化 MPC 产品^[71]。实时实现自适应控制存在一定的困难, 如果没有理论突破, 近期这方面可能不会有很大的进展。
3. 鲁棒 MPC: 除了 Honeywell 公司的 RMPCT 在控制器设计时考虑到模型的不确定性外, 其他产品是通过大量的仿真实验来评估模型失配的影响。设计具有鲁棒稳定性保障的 MPC 控制器将减少整定与测试时间。
4. 非线性 MPC: 未来的 MPC 技术应该能将过程知识与测试数据有机地接合起来发展非线性模型。由于建立非线性过程数据模型比较困难, 使用机理模型或其它替代模型就尤为重要^[72]。

1.4.2 模型预测控制的应用趋势

不是所有的预测控制器都处于图 1.2 中所示的位置, 在某些机械伺服控制系统应用中 MPC 控制器已包含了下层的常规控制回路。另外, 当采用自适应控制时, 也需要预测控制直接覆盖下层的控制回路^{[73]-[75]}。随着预测控制不断向新的领域渗透, 将来预测控制直接涵盖常规控制回路的现象会越来越普遍。这样, 最下层就只留下了执行机构伺服层, 如图 1.4 所示。在飞机与航天器的控制中, 执行机构层不存在单回路控制, 因为每个设定值的实现都需要多个执行机构协同作用, 例如爬升速度控制、三维姿态控制等等。当包含底层控制功能时, 可能要求

更快的控制周期。图 1.4 所示控制结构的另一个特点是可以把经济效益优化与动态性能优化整合在一起，全部由预测控制来实现。

从理论服务于实践这个辩证关系的角度出发，尽管有必要进一步完善或发展新的 MPC 方法，但大力推广 MPC 的应用更加重要，更富有挑战性。

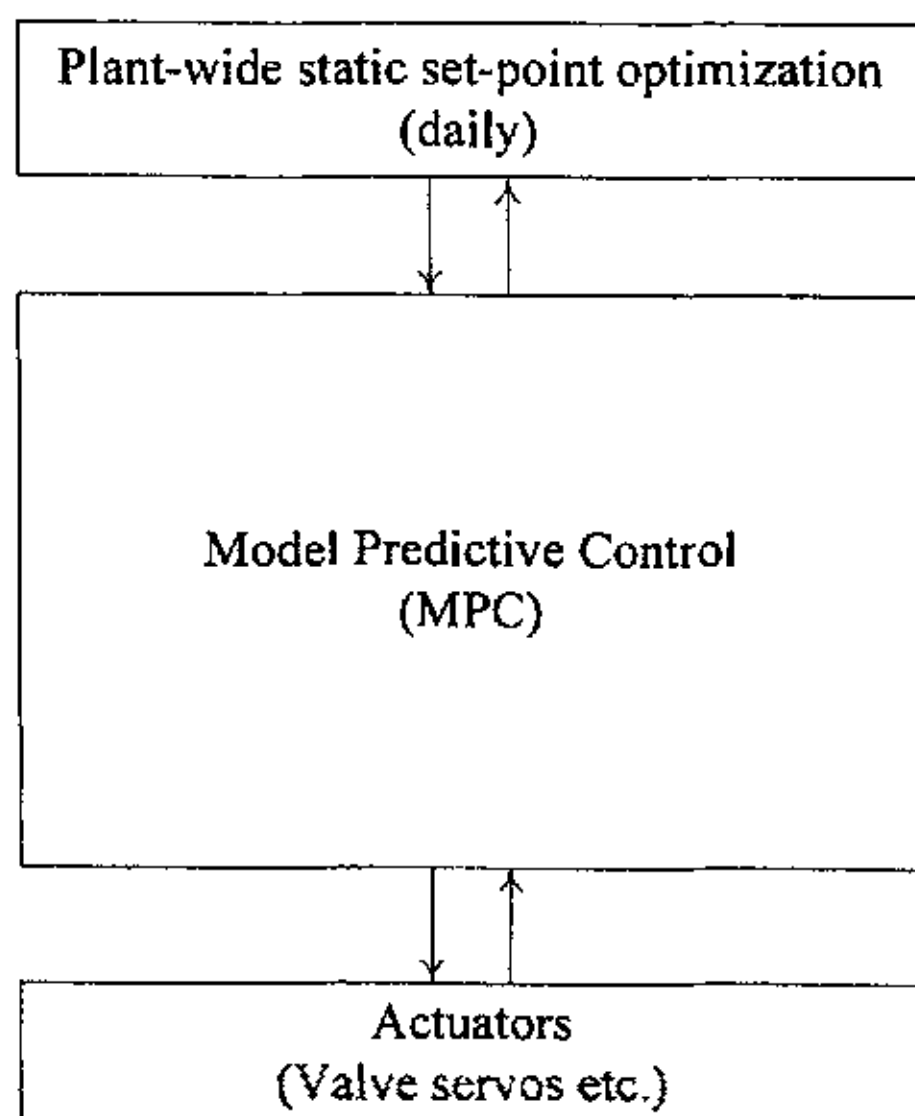


图 1.4 未来预测控制的应用趋势

1.5 广义预测控制

自从 Clarke 等人提出广义预测控制(Generalized Predictive Control (GPC))算法以来^{[76][77]}，GPC 算法已成为最受欢迎的模型预测控制算法之一。文献[28][78]-[81]对广义预测控制的发展概况、理论基础及应用细节做了深入的探讨。文献[28]对 GPC 算法论述较全面，介绍了 GPC 单变量、多变量形式的推导方法、工业应用简化办法、有约束 GPC 的形式和鲁棒 GPC 等内容。在进一步讨论 GPC 算法之前，先简要介绍一下 GPC 算法。由于一般的被控制对象是由多个输入变量多个输出变量(Multi-Input Multi-Output (MIMO))构成，纯粹由单个输入变量单个输出变量(Single-Input Single-Output (SISO))组成的系统只是 MIMO 的特例，因此以下只介绍多变量 GPC(Multivariable GPC (MGPC))的算法。

1.5.1 多变量广义预测控制算法

1.5.1.1 过程模型

m 个输入 n 个输出的多变量过程的 CARIMA 模型可表达如下:

$$\mathbf{A}(z^{-1})\mathbf{y}(t) = \mathbf{B}(z^{-1})\mathbf{u}(t-1) + \frac{1}{\Delta}\mathbf{C}(z^{-1})\mathbf{e}(t) \quad (1.1)$$

其中 $\mathbf{A}(z^{-1})$ 和 $\mathbf{C}(z^{-1})$ 为 $n \times n$ 首一多项式矩阵, $\mathbf{B}(z^{-1})$ 为 $n \times m$ 维多项式矩阵, 即:

$$\begin{aligned} \mathbf{A}(z^{-1}) &= \mathbf{I}_{n \times n} + \mathbf{A}_1 z^{-1} + \mathbf{A}_2 z^{-2} + \cdots + \mathbf{A}_{na} z^{-na} \\ \mathbf{B}(z^{-1}) &= \mathbf{B}_0 + \mathbf{B}_1 z^{-1} + \mathbf{B}_2 z^{-2} + \cdots + \mathbf{B}_{nb} z^{-nb} \\ \mathbf{C}(z^{-1}) &= \mathbf{I}_{n \times n} + \mathbf{C}_1 z^{-1} + \mathbf{C}_2 z^{-2} + \cdots + \mathbf{C}_{nc} z^{-nc} \end{aligned}$$

算子 $\Delta = 1 - z^{-1}$, $\mathbf{y}(t) = [y_1(t), y_2(t), \dots, y_n(t)]^T$, $\mathbf{u}(t) = [u_1(t), u_2(t), \dots, u_m(t)]^T$ 与 $\mathbf{e}(t) = [e_1(t), e_2(t), \dots, e_n(t)]^T$ 分别为 t 时刻的 $n \times 1$ 输出向量, $m \times 1$ 输入向量和 $n \times 1$ 噪声向量。噪声向量假设为零均值白噪声。 na 、 nb 、 nc 分别是矩阵多项式 $\mathbf{A}(z^{-1})$ 、 $\mathbf{B}(z^{-1})$ 、 $\mathbf{C}(z^{-1})$ 的阶次。 $\mathbf{A}_i (i=1, \dots, na)$ 、 $\mathbf{B}_j (j=0, \dots, nb)$ 、 $\mathbf{C}_k (k=1, \dots, nc)$ 分别是 $\mathbf{A}(z^{-1})$ 、 $\mathbf{B}(z^{-1})$ 、 $\mathbf{C}(z^{-1})$ 的系数列向量。

1.5.1.2 目标函数

考虑如下的有限时域二次目标函数:

$$J(N_1, N_2, N_3) = \sum_{j=N_1}^{N_2} \|\hat{\mathbf{y}}(t+j|t) - \mathbf{w}(t+j)\|_{\mathbf{R}}^2 + \sum_{j=1}^{N_3} \|\Delta \mathbf{u}(t+j-1)\|_{\mathbf{Q}}^2 \quad (1.2)$$

其中 $\hat{\mathbf{y}}(t+j|t)$ 为系统输出的最优 j 步向前预报, 它基于 t 时刻已知的输入输出数据, 即如果已知了过去的输入输出值和未来时刻的控制序列, 那它即为 t 时刻输出矢量的期望值, $\mathbf{w}(t+j)$ 为未来输出矢量的设定值或参考序列。 N_1 , N_2 为最小和最大预测时域, N_3 为控制时域。 \mathbf{R} 和 \mathbf{Q} 是正定的加权矩阵。

1.5.1.3 输出预测表达式

首先考虑 Diophantine 方程:

$$\mathbf{C}(z^{-1}) = \mathbf{E}_j(z^{-1})\tilde{\mathbf{A}}(z^{-1}) + z^{-j}\mathbf{F}_j(z^{-1}) \quad (1.3)$$

其中, $\tilde{\mathbf{A}}(z^{-1}) = \mathbf{A}(z^{-1})\Delta$, $\mathbf{E}_j(z^{-1})$ 、 $\mathbf{F}_j(z^{-1})$ 分别为 $j-1$ 和 n_a 阶的唯一的多项式矩阵。

定义多项式矩阵 $\bar{\mathbf{E}}_j(z^{-1})$ 和 $\bar{\mathbf{C}}_j(z^{-1})$, 满足

$$\bar{\mathbf{E}}_j(z^{-1})\mathbf{C}(z^{-1}) = \bar{\mathbf{C}}_j(z^{-1})\mathbf{E}_j(z^{-1}) \quad (1.4)$$

其中 $\bar{\mathbf{E}}_0 = \mathbf{I}$, 且 $\det(\bar{\mathbf{C}}_j(z^{-1})) = \det(\mathbf{C}_j(z^{-1}))$ 。

定义

$$\bar{\mathbf{F}}_j(z^{-1}) = z^j(\bar{\mathbf{C}}_j(z^{-1}) - \bar{\mathbf{E}}_j(z^{-1})\tilde{\mathbf{A}}(z^{-1})) \quad (1.5)$$

用 $\bar{\mathbf{E}}_j(z^{-1})\Delta$ 左乘(1.1)式:

$$\bar{\mathbf{E}}_j(z^{-1})\tilde{\mathbf{A}}(z^{-1})\mathbf{y}(t+j) = \bar{\mathbf{E}}_j(z^{-1})\mathbf{B}(z^{-1})\Delta\mathbf{u}(t+j-1) + \bar{\mathbf{E}}_j(z^{-1})\mathbf{C}(z^{-1})\mathbf{e}(t+j) \quad (1.6)$$

利用(1.4)和(1.5)式, 得到:

$$\bar{\mathbf{C}}_j(z^{-1})(\mathbf{y}(t+j) - \bar{\mathbf{E}}_j(z^{-1})\mathbf{e}(t+j)) = \bar{\mathbf{E}}_j(z^{-1})\mathbf{B}(z^{-1})\Delta\mathbf{u}(t+j-1) + \bar{\mathbf{F}}_j(z^{-1})\mathbf{y}(t) \quad (1.7)$$

根据噪声的分布特征, 上式左边取期望值, 可得

$$E[\bar{\mathbf{C}}_j(z^{-1})\mathbf{y}(t+j) - \bar{\mathbf{E}}_j(z^{-1})\mathbf{e}(t+j)] = \bar{\mathbf{C}}_j(z^{-1})\hat{\mathbf{y}}(t+j|t), \text{ 则最优输出预测}$$

$\hat{\mathbf{y}}(t+j|t)$ 由下式产生:

$$\bar{\mathbf{C}}_j(z^{-1})\hat{\mathbf{y}}(t+j|t) = \bar{\mathbf{E}}_j(z^{-1})\mathbf{B}(z^{-1})\Delta\mathbf{u}(t+j-1) + \bar{\mathbf{F}}_j(z^{-1})\mathbf{y}(t) \quad (1.8)$$

再引入如下 Diophantine 方程:

$$\mathbf{I} = \mathbf{J}_j(z^{-1})\bar{\mathbf{C}}_j(z^{-1}) + z^{-j}\mathbf{K}_j(z^{-1}) \quad (1.9)$$

其中 $\delta(\mathbf{J}_j(z^{-1})) < j$ 。(1.8)式乘以 $\mathbf{J}_j(z^{-1})$ 并利用(1.9)得:

$$(\mathbf{I} - z^j\mathbf{K}_j(z^{-1}))\hat{\mathbf{y}}(t+j|t) = \mathbf{J}_j(z^{-1})\bar{\mathbf{E}}_j(z^{-1})\mathbf{B}(z^{-1})\Delta\mathbf{u}(t+j-1) + \mathbf{J}_j(z^{-1})\bar{\mathbf{F}}_j(z^{-1})\mathbf{y}(t) \quad (1.10)$$

即:

$$\hat{\mathbf{y}}(t+j|t) = \mathbf{J}_j(z^{-1})\bar{\mathbf{E}}_j(z^{-1})\mathbf{B}(z^{-1})\Delta\mathbf{u}(t+j-1) + (\mathbf{K}_j(z^{-1}) + \mathbf{J}_j(z^{-1})\bar{\mathbf{F}}_j(z^{-1}))\mathbf{y}(t) \quad (1.11)$$

令: $\mathbf{J}_j(z^{-1})\bar{\mathbf{E}}_j(z^{-1})\mathbf{B}(z^{-1}) = \mathbf{G}_j(z^{-1}) + z^{-j}\mathbf{G}_{pj}(z^{-1})$, 其中 $\delta(\mathbf{G}_j(z^{-1})) < j$, 则

j 步最优预测输出可表示为:

$$\hat{\mathbf{y}}(t+j|t) = \mathbf{G}_j(z^{-1})\Delta\mathbf{u}(t+j-1) + \mathbf{G}_{pj}(z^{-1})\Delta\mathbf{u}(t-1) + (\mathbf{K}_j(z^{-1}) + \mathbf{J}_j(z^{-1})\bar{\mathbf{F}}_j(z^{-1}))\mathbf{y}(t) \quad (1.12)$$

令:

$$\mathbf{f}_{Cj} = \mathbf{G}_{pj}(z^{-1})\Delta\mathbf{u}(t-1) + (\mathbf{K}_j(z^{-1}) + \mathbf{J}_j(z^{-1})\bar{\mathbf{F}}_j(z^{-1}))\mathbf{y}(t) \quad (1.13)$$

则(1.12)式变为:

$$\hat{\mathbf{y}}(t+j|t) = \mathbf{G}_j(z^{-1})\Delta\mathbf{u}(t+j-1) + \mathbf{f}_{Cj} \quad (1.14)$$

输出预测式的右边第一项是未来控制增量的强迫响应, 后一项是由过去的输入量和输出量决定的自由响应。

j 取 1 到 N_2 , 则从(1.14)得到:

$$\begin{aligned} \hat{\mathbf{y}}(t+1|t) &= \mathbf{G}_1(z^{-1})\Delta\mathbf{u}(t) + \mathbf{f}_{C1} \\ \hat{\mathbf{y}}(t+2|t) &= \mathbf{G}_2(z^{-1})\Delta\mathbf{u}(t+1) + \mathbf{f}_{C2} \\ &\vdots \\ \hat{\mathbf{y}}(t+N_2|t) &= \mathbf{G}_{N_2}(z^{-1})\Delta\mathbf{u}(t+N_2-1) + \mathbf{f}_{CN_2} \end{aligned} \quad (1.15)$$

定义:

$$\mathbf{y}_{N_{12}} = \begin{bmatrix} \hat{\mathbf{y}}(t+N_1|t) \\ \hat{\mathbf{y}}(t+N_1+1|t) \\ \vdots \\ \hat{\mathbf{y}}(t+N_2|t) \end{bmatrix} \quad (1.16)$$

$$\mathbf{G}_{N_{12}} = \begin{bmatrix} \mathbf{G}_{N_1}(z^{-1}) \\ \mathbf{G}_{N_1+1}(z^{-1}) \\ \vdots \\ \mathbf{G}_{N_2}(z^{-1}) \end{bmatrix} \quad (1.17)$$

$$\mathbf{f}_{CN_{12}} = \begin{bmatrix} \mathbf{f}_{CN_1} \\ \mathbf{f}_{C(N_1+1)} \\ \vdots \\ \mathbf{f}_{CN_2} \end{bmatrix} \quad (1.18)$$

$$\mathbf{u} = \begin{bmatrix} \Delta\mathbf{u}(t) \\ \Delta\mathbf{u}(t+1) \\ \vdots \\ \Delta\mathbf{u}(t+N_2-1) \end{bmatrix} \quad (1.19)$$

则从最小预测时域到最大预测时域的输出预测式为:

$$\mathbf{y}_{N_{12}} = \mathbf{G}_{N_{12}}\mathbf{u} + \mathbf{f}_{CN_{12}} \quad (1.20)$$

1.5.1.4 控制律求解

引入控制时 N_3 ，假定从 $t + N_3$ 个采样时刻起，控制信号保持不变。定义：

$$\mathbf{u}_{N_3} = \begin{bmatrix} \Delta \mathbf{u}(t) \\ \Delta \mathbf{u}(t+1) \\ \vdots \\ \Delta \mathbf{u}(t+N_3-1) \end{bmatrix} \quad (1.21)$$

则输出预测式(1.20)变为：

$$\mathbf{y}_{N_{12}} = \mathbf{G}_{N_{123}} \mathbf{u}_{N_3} + \mathbf{f}_{CN_{12}} \quad (1.22)$$

其中， $\mathbf{G}_{N_{123}}$ 为 $\mathbf{G}_{N_{12}}$ 的子矩阵。

将输出预测式(1.22)代入目标函数(1.2)式，得：

$$J = (\mathbf{G}_{N_{123}} \mathbf{u}_{N_3} + \mathbf{f}_{CN_{12}} - \mathbf{w})^T \bar{\mathbf{R}} (\mathbf{G}_{N_{123}} \mathbf{u}_{N_3} + \mathbf{f}_{CN_{12}} - \mathbf{w}) + \mathbf{u}_{N_3}^T \bar{\mathbf{Q}} \mathbf{u}_{N_3} \quad (1.23)$$

其中 $\bar{\mathbf{R}} = \text{diag}(\mathbf{R}, \dots, \mathbf{R})$ ， $\bar{\mathbf{Q}} = \text{diag}(\mathbf{Q}, \dots, \mathbf{Q})$ 。

如果没有限制条件，最优值可表示为：

$$\mathbf{u}_{N_3} = (\mathbf{G}_{N_{123}}^T \bar{\mathbf{R}} \mathbf{G}_{N_{123}} + \bar{\mathbf{Q}})^{-1} \mathbf{G}_{N_{123}}^T \bar{\mathbf{R}} (\mathbf{w} - \mathbf{f}_{CN_{12}}) \quad (1.24)$$

使用滚动控制策略， t 时刻只有 $\Delta u_j(t)$, $j = 1, \dots, m$ 是需要的，定义矩阵：

$$\mathbf{M} = \begin{bmatrix} \underbrace{1 \ 0 \ \dots \ 0}_{N_3+1} & \underbrace{0 \ 0 \ \dots \ 0}_{N_3+1} & \dots & \underbrace{0 \ 0 \ \dots \ 0}_{N_3+1} \\ \underbrace{0 \ 0 \ \dots \ 0}_{N_3+1} & \underbrace{1 \ 0 \ \dots \ 0}_{N_3+1} & \dots & \underbrace{0 \ 0 \ \dots \ 0}_{N_3+1} \\ \vdots & \vdots & \ddots & \vdots \\ \underbrace{0 \ 0 \ \dots \ 0}_{N_3+1} & \underbrace{0 \ 0 \ \dots \ 0}_{N_3+1} & \dots & \underbrace{1 \ 0 \ \dots \ 0}_{N_3+1} \end{bmatrix} \quad (1.25)$$

令： $\mathbf{K} = \mathbf{M}(\mathbf{G}_{N_{123}}^T \bar{\mathbf{R}} \mathbf{G}_{N_{123}} + \bar{\mathbf{Q}})^{-1} \mathbf{G}_{N_{123}}^T \bar{\mathbf{R}}$ ，则控制律可表示为：

$$\Delta \mathbf{u}(t) = \mathbf{K}(\mathbf{w} - \mathbf{f}_{CN_{12}}) \quad (1.26)$$

其中： $\Delta \mathbf{u}(t) = [\Delta u_1(t), \Delta u_2(t), \dots, \Delta u_m(t)]^T$ 。即一个线性增益矩阵乘以参考轨迹与输出自由响应之差。对于非自适应情况， \mathbf{K} 可以提前算出，而对于自适应情况，则在模型更新的每一时刻需要重新计算 \mathbf{K} 。

1.5.2 广义预测控制算法的优点

广义预测控制算法使用 CARIMA 模型,不但可以描述各种各样的过程,例如稳定过程、积分过程或者不稳定过程,而且引入了扰动与噪声模型,可以对可测量扰动、不可测量扰动及测量噪声进行显式地考虑,优化了反馈校正方式。GPC 已成功地在许多工业应用中得以实现,在参数和时间滞后不确定的情况下显示了良好的控制性能和鲁棒性。采用二次性能目标函数可以很自然地引入输入输出约束,在不考虑约束条件时可以得到解析解,加快了求解速度。

另外,广义预测控制是从广义最小方差控制器的基础上发展而来,其研究工作是围绕自适应控制思想展开的。由于其模型参数少,比其它算法更容易实现自适应控制。

广义预测控制算法不但可以克服现行 MPC 方法中的许多不足,而且与 MPC 的发展趋势一致,加强其理论与应用研究,必将对 MPC 的整体发展产生深远影响。

1.6 广义预测控制算法简化实现方法

1.6.1 广义预测控制算法推导及实现的复杂性

从前面的 GPC 算法的推导过程可知, GPC 算法的推导和求解有一定的数学难度,例如需要求解大量的 Diophantine 方程、得到矩阵 $\mathbf{G}_{N_{12}}$ 和 $\mathbf{f}_{CN_{12}}$ 及高维矩阵求逆 $(\mathbf{G}_{N_{12}}^T \bar{\mathbf{R}} \mathbf{G}_{N_{12}} + \bar{\mathbf{Q}})^{-1}$ 等等。特别是对自适应过程,当参数改变时,以上运算都要重新进行,需要占用大量的计算资源。这对于计算能力有限的控制系统(例如 DCS 系统)、控制周期短的快速控制系统(例如电力传动控制系统)、及控制回路多的大型控制系统来说,有时是不可实现的。这些困难为 GPC 的工程应用增加了难度,也阻碍了其在应用领域的发展。

如何简化 GPC 算法的在线实现复杂性及减少其在线计算量,一直以来是大量研究人员关注的热点问题,也是拓展广义预测控制应用范围必须解决的关键问题之一。因此,本论文将着力于这方面的研究,通过探询 GPC 的内在机理尽可能地降低 GPC 算法的实现难度。

GPC 算法是 MPC 家族中的一员,有着相同的理论框架,可以借鉴其它方法

的简化实现思想。因此,在进一步研究 GPC 算法简化实现方法之前,首先回顾一下 MPC 算法常用的一些简化实现方法,及 GPC 算法在简化实现方面已取得的一些成果。

1.6.2 常用的几种简化实现方法

预测控制方法具有优秀的控制性能,但却需要更多的计算。上面介绍的工业化模型预测控制方法都考虑到了预测控制方法的简化实现问题。为了减少计算量,简化在线实现,通常采取以下手段:

1. 定义有限预测时域

最小预测时域的物理意义是避开无意义的计算。当对象存在纯滞后 d 时,输入变量的作用要等到 $t+d$ 时刻才能在被控制变量中反应出来,如果在目标函数中考虑 $t+d$ 时刻前的输出跟踪误差显然是没有意义的。另外,为得到平滑控制,也要求对输出预测的前几步不作要求,例如在非最小相位系统控制中,反冲阶段的输出预测值可以不作计算。对于最大预测时域的选择一般以覆盖对象的瞬态特性为宜,太大的预测时域会增加在线计算量。表 1.3 和表 1.4 中所示的 MPC 方法,除 Adersa 的 PFC 和 Aspen Technology 的 Aspen Target 外,其它 MPC 产品都用到有限预测时域的概念。

“拟合点”的方法也是用来减少运算量,它只对预测时域上一些点的拟合精度有要求,其它点不作计算。采用这种方法的产品有 Adersa 的 PFC 和 Aspen Technology 的 Aspen Target。

极限情况下,只要求计算一个时间点的预测值,如扩展时域自适应控制中只要求 $t+N$ 时间点上的跟踪误差最小^[82]。

2. 构造特定形式的控制律

输入参数化是 MPC 的主要特征之一,几乎所有的预测控制算法都引入控制时域 N_u 的概念,即假定 $\Delta u(t+k)=0, k \geq N_u$ 。一般情况下是控制时域越大,则控制性能会越好,但随之引入的计算量会大幅度增加。适当选择控制时域,以便在控制性能与在线计算量之间进行折中。

Aspen Technology 的 DMC-plus、Invensys 的 Connoisseur、和 SGS 的 SMOC 采用所谓的“blocking”技术,亦是待计算的控制信号参数化成多个值,却允许一些点上的值不进行计算,相当于减小了控制时域。

最简单的方法是令控制时域 $N_u=1$,则控制量的求解已得到完全的简化,例如 Adersa 的 HIECON、Continental 的 MVC、及扩展预测自适应控制等,但此时

控制性能的损失却是无法估量的。

Adersa 的 PFC 使用“基函数”方法,把控制变量表达成基函数的线性组合,只需要使用较少的参数便能得到满意的输入曲线,同样达到减少计算量的目的。

3. 采用分层优化、次优化或“cut-off”策略

表 1.3 和表 1.4 中所列产品,除 Aderda 的 HIECON 和 Dot Products 的 NOVA-NLC 外,其它产品中都采用稳态优化和动态优化相结合的方法,目的就是简化在线实现。在不同的优化层中,考虑不同目标函数和约束条件,尽可能地减少计算量。一般上层是静态优化,考虑动态过程的信息相对较少,主要是为求出最优稳态工作点。下层是动态优化,一般要考虑输入输出约束,但为加快运算和保证有可行解,有时采用罚函数方法来处理输入输出约束,如 Aspentech 的 DMCPlus、Honeywell 的 RMPCT 等。还有的商品化预测控制软件在计算控制量时,按控制目标的需要分成几种控制模式,不同的模式中考虑的约束不同,优化方法也不一样,例如 Connoisseur 就有三种控制模式^[31]: LR(Long-Range)、QP-LR、和 QP 模式。在 LR 模式中不考虑约束, QP-LR 模式中考虑输入约束,而 QP 模式中要考虑输入与输出约束,所有约束都用罚函数定义成的软约束形式。

在变量非常多或者速度非常快的控制问题中,没有足够的时间来求解 QP 问题,只好采取快速次优化算法求得动态优化的近似解。DMC-plus 算法中,当预测到输入信号超出其上、下幅值约束时,直接取其最近的约束边界值,并从目标函数中移除相应的控制变量,重新进行优化计算。PFC 算法却直接采用“cut-off”方法,优化计算时不考虑约束条件,如果计算得到的输入信号值超出其边界,则“剪掉”多余的部分,取其最近的边界值。这些方法一般可以避免输入违背约束条件,但有时会导致控制性能的下降。

在预测控制方案中,采用各种办法来处理约束及优化问题,其目的就是要在保证优化可行的条件下,尽可能简化在线计算。

4. 使用“区域”或“管道形区域”轨迹

“区域”或者“管道形区域”两种形式的轨迹不同于常规的设定值或参考轨迹形式,只要求变量的值落在某个范围内。一方面,可以减少执行机构的来回振荡,保持系统更加平稳;另一方面,就是减少了在线计算量,因为当变量落在预测区域内时,目标函数就可以不对其进行考虑。

1.6.3 广义预测控制简化实现方法

GPC 使用 CARIMA 模型,参数少,可以消除负载扰动。对于可线性化的系统,很容易地实现 GPC 的自适应控制。但是 GPC 算法也存在一些缺点,最主要

就是在线计算量非常大,特别是当采用自适应控制模式时,需要占用大量的计算资源。尽管某些情况下,非自适应模式的广义预测控制方法便能满足控制需要,但广义预测控制本身是在自适应理论框架下发展起来的,自适应是其重要的特征。因此,为了推广 GPC 的应用范围,就必须减少计算量,简化实现难度。除采用以上提到的 MPC 方法常用的几种简化实现方法外,研究人员已从以下几个方面对 GPC 进行了改进:

1. 递推求解 Diophantine 方程

早在 1987 年 Clarke 等人提出广义预测控制算法时^[76],便已考虑到了计算简化问题。白噪声情形时,即 CARIMA 模型中 $C(z^{-1})=1$,对如下的 Diophantine 方程可以采用递推的方法进行求解:

$$1 = E_j(z^{-1})A\Delta + z^{-j}F_j(z^{-1}) \quad (1.27)$$

其中 E_j 和 F_j 是由 $A(z^{-1})$ 和预测步 j 唯一确定的多项式。在之后的许多文献中都引用了类似的做法,如文献[26]-[28][81]。

有色噪声情形,即 $C(z^{-1}) \neq 1$,为了获得类似的递推解法,文献[28]中对输入输出变量进行了滤波:

$$y^f(t) = \frac{1}{C(z_{-1})} y(t) \quad u^f(t) = \frac{1}{C(z_{-1})} u(t) \quad (1.28)$$

原过程模型变为如下形式:

$$A(z^{-1})y^f(t) = B(z^{-1})u^f(t) + \frac{e(t)}{\Delta} \quad (1.29)$$

此时,可以采用与白噪声时一样的方法求解 Diophantine 方程。不过要得到预测信号 $\hat{y}(t+j|t)$ 则必需对 $\hat{y}^f(t+j|t)$ 进行 $C(z^{-1})$ 滤波。实际上,有色噪声时,Diophantine 方程也可以递推求解^[28]。文献[28]还将 Diophantine 方程递推求解的思想推广到了多变量 GPC 中。

2. 矩阵求逆采用特殊算法

文献[83][84]对 $(G^T G + \lambda I)^{-1}$ 分别采用了两种不同的递推算法。文献[85]利用下三角矩阵元素排列的特殊性,对 $(G^T G + \beta^2 I)$ 进行分解,实现了矩阵的递推求逆。文献[86]通过改变目标函数的形式,使得求解控制律时的求逆矩阵变为: $(G + \lambda I)$,而这个矩阵是一个下 Teoplitz 矩阵,存在相应的快速求解公式^[87]。

3. 直接利用模型参数辨识值计算输出预测值

在文献[88][89]中, 分别对单变量对象与多变量对象的广义预测控制方法作了改进, 在计算输出预测值时避开 Diophantine 方程求解。其思路如下: 利用模型表达式, 先得到 $\hat{y}(t+1|t)$ 的表达式, 再到得到 $\hat{y}(t+2|t)$ 表达式, 然后把 $\hat{y}(t+2|t)$ 表达式中出现的 $\hat{y}(t+1|t)$ 项用 $\hat{y}(t+1|t)$ 的表达式替换, 并以此类推。得到的输出预测表达式中的系数可以由过程模型参数计算出来, 从而避开了 Diophantine 方程的求解, 在一定程度上减少了计算量。

4. 采用隐式自适应方法来实现广义预测控制

文献[90]中对 CARIMA 模型的对象, 通过构造特殊形式的目标函数, 得到如下形式的控制律:

$$\Delta u(t) = P(z^{-1})y_r(t+N_1) - X(t)^T \hat{\theta}(t) \quad (1.30)$$

上式中的 $\theta(t)$ 可以通过辨识方法得出, $X(t)$ 由过程输入输出信息得出; 第一项中的 $P(z^{-1})$ 在计算过程中是不变的, $y_r(t+N_1)$ 为设定值。因此可以通过辨识方法得到直接算法。文献[91]对于以下被控对象:

$$A(z^{-1})\Delta y(t) = B(z^{-1})\Delta u(t-1) \quad (1.31)$$

得到了类似的算法。

在文献[92]中, 对一类可线性化系统, 利用文献[81]中得到的广义预测控制器:

$$u(k) = l_0(k)y(k) + \dots + l_n(k)y(k-n) + p_1(k)y_r(k+1) + \dots + p_N(k)y_r(k+N) + \lambda_1(k)u(k-1) + \dots + \lambda_m(k)u(k-m) \quad (1.32)$$

采用小波网络直接辨识控制器的参数。

文献[93]-[96]分别采用了一个或多个辨识器来实现类似的隐式算法。

5. 采用并行计算技术

文献[97][98]中对自适应广义预测算法的参数自校正、柔化、调节和预测四部分的结构作了分析, 给出了能实现并行运算的结构, 在并行计算机存在的条件下, 可以使用并行算法来加快广义预测控制的实现。在文献[99][100]中, 分析了广义预测的阵列结构, 提出以 systolic 算法来实现自适应广义预测控制的并行算法。但这些设想的实现依赖于并行机或并行运算芯片的发展。

6. 采用经验公式

Camacho 等人指出^{[28][101]}, 对于如下形式的典型工业过程:

$$G(s) = \frac{K}{1+Ts} e^{-\tau s}, \quad G(s) = \frac{K}{s(1+Ts)} e^{-\tau s} \quad (1.33)$$

其广义预测控制器系数与过程离散参数之间可以用一套简单的 Ziegler-Nichols 函数联系起来,通过这些关系式可以直接计算控制器系数,从而简化了广义预测控制在工业过程中的应用。

7. 采用网络映射技术

这种方法利用控制器系数与对象模型参数之间的对应关系,根据对象模型参数变化的范围,离线训练好映射网络,在线运算时只需给出模型参数值,便可以从网络直接得到控制器的系数。这种映射可以采用小脑模型来实现^{[102][103]},也可采用 BP 网络来实现^{[104][105]}。

以上介绍的一些方法主要是针对单变量系统,对于多变量系统而言,由于其复杂性,只有较少的文献提到了对多变量广义预测控制方法进行改进^{[89][94][106]}。

1.7 本文的主要内容

不论多么完美的理论如果不能很好地服务于实践,其对于人类社会发展的贡献便很难充分展示出来。同时,如果人们所拥有的资源是无限的,我们甚至不需要殚思极虑地进行优化生产、发展优化算法。广义预测控制吸收了自适应、优化、预测等多种先进思想,拥有一个完美的理论框架,是一种很有前途的 MPC 方法。但受到计算资源的限制,阻碍了其向应用领域的大力发展,急需寻找各种简单易行的实现方法。另外,从提高效率的角度考虑,也需要我们简化实现方法。因为,效率是投入与产出之比,它要求我们用比较经济的方法来达到预定的目的。如果对于一定的投入,取得了更多的产出,即为提高了效率;同样,若对于一定的产出,我们减少了投入,那么也是提高了效率^[107]。使用 GPC 算法可以提高控制性能,实现优化生产,获得更大的经济效益;同样,降低 GPC 的实现成本,是减少了投入,扩大了使用范围,同样可以带来可观的经济效益与社会效益。当然简化实现,不应当以牺牲控制性能为代价,而应当是通过对 GPC 算法本身的深入分析,寻找更加经济的途径来实现优化控制。

基于此,本文在前人已作改进的基础之上,进一步对 GPC 算法进行研究,寻找更加经济、更加有效的实现手段。

首先,在 $A(z^{-1})$ 和 $C(z^{-1})$ 多项式矩阵为对角形式的多变量 CARIMA 模型 GPC 方法求解过程中,显式地考虑了过程可能存在的纯滞后,以提高计算效率。当被控制对象输入输出维数不是很大、或者过程不存在明显纯滞后时,这种考虑的意义也许不是很明显。但当过程纯滞后很大,或者被控对象维数很高时,这种细致的分析带来的效益便不可忽视。例如在 1999 年 Qin 和 Badgwell 两人对工业

MPC 供应商进行的一次调查中显示^[25], Aspen Technology 实现的 MPC 控制中被控制对象输入输出变量的维数高达 283×603 (输入 \times 输出), 对于这样的过程, 任何计算效率的提高都是有益的。

其次, 在非自适应模式下, 如果不考虑约束, 尽管可以提前计算出广义预测控制器的系数矩阵, 但在每一个控制周期还是需要计算自由响应项, 不利于 GPC 算法的实现。通过对自由响应项的进一步分析发现, 控制律完全可以表达成控制器系数与设定值、输入历史数据和输出历史数据乘积的形式。控制器系数完全由对象模型参数及设计参数决定, 在非自适应模式下可以提前算出, 则在线实现时 GPC 控制器就如 PID 一样简单。

再次, 利用对象的 CARIMA 模型直接进行递推, 不但可以避开 Diophantine 方程的求解, 而且控制器可表达成控制器系数与设定值、输入历史数据、输出历史数据及模型预测误差历史数据乘积的形式。控制器系数完全由模型参数及设计参数确定, 系数个数只与模型的阶次相关。不论是自适应模式, 还是非自适应模式, 都得到了极大的简化。

最后, 指明广义预测控制算法本质上是一种函数映射关系: 按照广义预测控制算法所确定的对应关系, 将对象模型参数所确定的空间映射到控制器系数所确定的空间。并利用神经网络的空间映射能力实现了 CARIMA 模型参数到广义预测控制器系数的直接映射, 彻底地简化了多变量自适应广义预测控制算法的在线实现。

论文的第二章到第五章是简化实现方法的理论推导, 第六章是广义预测控制简化实现方法在实验装置上的对比实验研究, 最后是论文的总结和展望。

第二章 多变量广义预测控制算法

摘要: 指出大多数物理可实现多变量过程的 CARIMA 模型中的 $A(z^{-1})$ 与 $C(z^{-1})$ 项可以构造成对角多项式矩阵形式。对这种特殊结构形式的模型, 先求出每个输出的预测表达式, 再组合在一起, 就得到了整个多变量过程的输出预测表达式, 最后得到多变量广义预测控制算法的控制律。算法推导过程中显式地考虑了纯滞后项, 以提高计算效率。介绍了多变量过程模型参数辨识方法。

2.1 引言

现实世界大部分的过程是多变量过程, 由多个输入变量和多个输出变量组成, 有时还可能存在可测量或不可测量的扰动。由于其表述与计算的复杂性, 常规的方法是把多变量过程进行解耦, 变成多个单变量过程, 再用单变量控制方法进行控制。但解耦方法需要输入变量与输出变量的个数相等, 而且输入与输出之间要能够进行配对。对于复杂的动态过程达到完全解耦是不可能的。

多变量广义预测控制是控制多变量过程的一种有效方法。但其复杂的数学推导及繁重的在线计算增加了其应用难度。文献[28]指出, 如果多变量 CARIMA 模型中的 $A(z^{-1})$ 与 $C(z^{-1})$ 取对角多项式矩阵的形式, 则其模型输出预测可以转变成多个多输入单输出子模型的输出预测, 其求解过程会大大简化, 而且计算量也会显著减少。并讨论了白噪声 CARIMA 模型滞后多项式已知、有色噪声 CARIMA 模型滞后多项式未知两种情况下多变量广义预测控制算法的求解问题。在多数情况下, 纯滞后可以从过程输入输出数据中分析得到, 有色噪声多项式也可以作为设计参数而存在。为了多变量广义预测控制算法的完整性, 本章对有色噪声 CARIMA 模型纯滞后情况已知、 $A(z^{-1})$ 与 $C(z^{-1})$ 取对角多项式矩阵形式的多变量广义预测控制算法的求解过程作了详细的推导。推导中显式地考虑了纯滞后, 以提高计算效率。

2.2 多变量过程的 CARIMA 模型

左 MFD: 给定 $q \times p$ 的具有有理分式矩阵形式的传递函数 $\mathbf{G}(s)$, 则一定存在 $q \times q$ 和 $q \times p$ 的多项式矩阵 $\mathbf{A}(s)$ 和 $\mathbf{B}(s)$ ^[108], 使下式成立:

$$\mathbf{G}(s) = \mathbf{A}^{-1}(s)\mathbf{B}(s) \quad (2.1)$$

称 $\mathbf{A}^{-1}(s)\mathbf{B}(s)$ 为 $\mathbf{G}(s)$ 的一个左矩阵分式描述, 简称为左 MFD (Matrix Fraction Description)。其中 $\mathbf{A}^{-1}(s)$ 为对角多项式矩阵。

(2.1)式的离散形式为:

$$\mathbf{T}(z^{-1}) = \mathbf{A}(z^{-1})^{-1}\mathbf{B}(z^{-1})z^{-1} \quad (2.2)$$

实际上, 要得到有色噪声多项式是一件很困难的工作, 一般可作为设计多项式, 因此可以把其设计成对角多项式矩阵的形式。

所以, 对于物理可实现的 m 个输入 n 个输出的多变量对象, 总可建立如下的 CARIMA 模型:

$$\mathbf{A}(z^{-1})\mathbf{y}(t) = \mathbf{B}(z^{-1})\mathbf{u}(t-1) + \frac{1}{\Delta}\mathbf{C}(z^{-1})\mathbf{e}(t) \quad (2.3)$$

其中: $\mathbf{A}(z^{-1}) = \text{diag}(A_{ii}(z^{-1}))$ 和 $\mathbf{C}(z^{-1}) = \text{diag}(C_{ii}(z^{-1}))$ 为 $n \times n$ 首一多项式矩阵,

$\mathbf{B}(z^{-1}) = [z^{-d_{ij}}B_{ij}(z^{-1})]$ 为 $n \times m$ 维多项式矩阵, d_{ij} 是第 i 个输出对于第 j 个输入的纯滞后。即:

$$\mathbf{A}(z^{-1}) = \begin{bmatrix} A_{11}(z^{-1}) & 0 & \cdots & 0 \\ 0 & A_{22}(z^{-1}) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A_{nn}(z^{-1}) \end{bmatrix}$$

$$\mathbf{B}(z^{-1}) = \begin{bmatrix} B_{11}(z^{-1})z^{-d_{11}} & B_{12}(z^{-1})z^{-d_{12}} & \cdots & B_{1m}(z^{-1})z^{-d_{1m}} \\ B_{21}(z^{-1})z^{-d_{21}} & B_{22}(z^{-1})z^{-d_{22}} & \cdots & B_{2m}(z^{-1})z^{-d_{2m}} \\ \vdots & \vdots & \ddots & \vdots \\ B_{n1}(z^{-1})z^{-d_{n1}} & B_{n2}(z^{-1})z^{-d_{n2}} & \cdots & B_{nm}(z^{-1})z^{-d_{nm}} \end{bmatrix}$$

$$\mathbf{C}(z^{-1}) = \begin{bmatrix} C_{11}(z^{-1}) & 0 & \cdots & 0 \\ 0 & C_{22}(z^{-1}) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & C_{nn}(z^{-1}) \end{bmatrix}$$

$$A_{ii}(z^{-1}) = 1 + a_{ii} z^{-1} + \cdots + a_{ii_{na_{ii}}} z^{-na_{ii}} \quad (i = 1, \cdots, n)$$

$$B_{ij}(z^{-1}) = b_{ij_0} + b_{ij_1} z^{-1} + \cdots + b_{ij_{nb_{ij}}} z^{-nb_{ij}} \quad (i = 1, \cdots, n, \quad j = 1, \cdots, m)$$

$$C_{ii}(z^{-1}) = 1 + c_{ii} z^{-1} + \cdots + c_{ii_{nc_{ii}}} z^{-nc_{ii}} \quad (i = 1, \cdots, n)$$

算子 $\Delta = 1 - z^{-1}$, $\mathbf{y}(t) = [y_1(t), y_2(t), \cdots, y_n(t)]^T$, $\mathbf{u}(t) = [u_1(t), u_2(t), \cdots, u_m(t)]^T$ 与 $\mathbf{e}(t) = [e_1(t), e_2(t), \cdots, e_n(t)]^T$ 为 t 时刻的 $n \times 1$ 维输出向量, $m \times 1$ 维输入向量和 $n \times 1$ 维噪声向量。假设噪声向量为零均值白噪声。

2.3 目标函数

考虑如下的有限时域二次目标函数:

$$J(\mathbf{N}_1, \mathbf{N}_2, \mathbf{N}_3) = \sum_{i=1}^n \sum_{k=N_{1i}}^{N_{2i}} \{ \delta_i(k) [\hat{y}_i(t+k|t) - w_i(t+k)]^2 \} + \sum_{j=1}^m \sum_{k=1}^{N_{3j}+1} \{ \lambda_j(k) [\Delta u_j(t+k-1)]^2 \} \quad (2.4)$$

其中: $\hat{y}_i(t+k|t)$ 为系统第 i 个输出的最优 k 步向前预报, $w_i(t+k)$ 为第 i 个输出 $t+k$ 时刻的设定值或参考轨迹, $\Delta u_j(t+k-1)$ 为第 j 个输入 $t+k-1$ 时刻的增量。
 $\mathbf{N}_1 = [N_{1_1}, N_{1_2}, \cdots, N_{1_n}]^T$ 为最小预测时域向量, $\mathbf{N}_2 = [N_{2_1}, N_{2_2}, \cdots, N_{2_n}]^T$ 为最大预测时域向量, $\mathbf{N}_3 = [N_{3_1}, N_{3_2}, \cdots, N_{3_m}]^T$ 为控制时域向量。 $\delta_i(k)$ 和 $\lambda_j(k)$ 是加权系数, 一般取常值。

2.4 输出预测表达式

对于第 i 个输出 y_i , 有:

$$A_{ii}(z^{-1})y_i(t) = \sum_{j=1}^m \{ z^{-d_{ij}} B_{ij}(z^{-1})u_j(t-1) \} + \frac{1}{\Delta} C_{ii}(z^{-1})e_i(t) \quad (2.5)$$

定义: $z^{-d_{ij}} B_{ij}(z^{-1}) = z^{-d_i} B_{ij}^*(z^{-1})$, 其中 $d_i = \min_j d_{ij}$ 是第 i 个输出对所有输入的最小时滞。

则(2.5)式变为:

$$A_{ii}(z^{-1})y_i(t) = \sum_{j=1}^m \{B'_{ij}(z^{-1})u_j(t-d_i-1)\} + \frac{1}{\Delta} C_{ii}(z^{-1})e_i(t) \quad (2.6)$$

即:

$$y_i(t) = - \sum_{k=1}^{na_{ii}+1} \tilde{a}_{ii_k} y_i(t-k) + \sum_{j=1}^m \left\{ \sum_{k=0}^{nb_{ij}} b_{ij_k} \Delta u_j(t-d_{ij}-k-1) \right\} + e_i(t) + \sum_{k=1}^{nc_{ii}} c_{ii_k} e_i(t-k) \quad (2.7)$$

考虑如下的 Diophantine 方程:

$$C_{ii}(z^{-1}) = E_{i_{k+d_i}}(z^{-1}) \tilde{A}_{ii}(z^{-1}) + z^{-(k+d_i)} F_{i_{k+d_i}}(z^{-1}) \quad (2.8)$$

其中, $\tilde{A}_{ii}(z^{-1}) = \Delta A_{ii}(z^{-1}) = 1 + \tilde{a}_{ii_1} z^{-1} + \tilde{a}_{ii_2} z^{-2} + \cdots + \tilde{a}_{ii_{na_{ii}}} z^{-na_{ii}} + \tilde{a}_{ii_{na_{ii}+1}} z^{-na_{ii}-1}$,

$E_{i_{k+d_i}}(z^{-1})$ 和 $F_{i_{k+d_i}}(z^{-1})$ 分别为 $k+d_i-1$ 阶和 na_{ii} 阶的多项式。此处约定

$\delta(C(z^{-1})) \leq d_i + na_{ii} + 1$, 以保证 Diophantine 方程 (2.8) 式解唯一。

如果(2.6)式两边同时乘以 $\Delta E_{i_{k+d_i}}(z^{-1})z^{k+d_i}$, 并利用公式(2.8)则有:

$$\begin{aligned} C_{ii}(z^{-1})y_i(t+k+d_i) &= F_{i_{k+d_i}}(z^{-1})y_i(t) + \\ &\sum_{j=1}^m \{E_{i_{k+d_i}}(z^{-1})B'_{ij}(z^{-1})\Delta u_j(t+k-1)\} + E_{i_{k+d_i}}(z^{-1})C_{ii}(z^{-1})e_i(t+k+d_i) \end{aligned} \quad (2.9)$$

注意到 $E_{i_{k+d_i}}(z^{-1})$ 为 $k+d_i-1$ 阶的, 故公式(2.9)中的噪声项都在将来时刻。

采用期望算子, 则 $y_i(t+k+d_i)$ 的期望值为:

$$C_{ii}(z^{-1})\hat{y}_i(t+k+d_i|t) = F_{i_{k+d_i}}(z^{-1})y_i(t) + \sum_{j=1}^m \{E_{i_{k+d_i}}(z^{-1})B'_{ij}(z^{-1})\Delta u_j(t+k-1)\} \quad (2.10)$$

引入如下的 Diophantine 方程:

$$1 = C_{ii}(z^{-1})M_{i_{k+d_i}}(z^{-1}) + z^{-(k+d_i)}N_{i_{k+d_i}}(z^{-1}) \quad (2.11)$$

其中: $\delta(M_{i_{k+d_i}}(z^{-1})) = k+d_i-1$, $\delta(N_{i_{k+d_i}}(z^{-1})) = nc_{ii}-1$ 。

用 $M_{i_{k+d_i}}(z^{-1})$ 乘以公式(2.10), 并利用公式(2.11), 则得:

$$\begin{aligned} \hat{y}_i(t+k+d_i|t) &= [N_{i_{k+d_i}}(z^{-1}) + M_{i_{k+d_i}}(z^{-1})F_{i_{k+d_i}}(z^{-1})]y_i(t) \\ &+ \sum_{j=1}^m \{M_{i_{k+d_i}}(z^{-1})E_{i_{k+d_i}}(z^{-1})B'_{ij}(z^{-1})\Delta u_j(t+k-1)\} \end{aligned} \quad (2.12)$$

令:

$$\bar{G}_{ij_{k+d_i}}(z^{-1}) = M_{i_{k+d_i}}(z^{-1})E_{i_{k+d_i}}(z^{-1})B_{ij}(z^{-1}) = G_{ij_k}(z^{-1}) + z^{-k}G_{p_{ij_k}}(z^{-1})$$

其中: $\delta(G_{ij_k}(z^{-1})) < k$, $G_{ij_k}(z^{-1}) = g_{ij_{k,0}} + g_{ij_{k,1}}z^{-1} + \cdots + g_{ij_{k,k-1}}z^{-k+1}$ 。则公式(2.12)可表示为:

$$\begin{aligned} \hat{y}_i(t+k+d_i|t) = & \sum_{j=1}^m \{G_{ij_k}(z^{-1})\Delta u_j(t+k-1)\} + \sum_{j=1}^m \{G_{p_{ij_k}}(z^{-1})\Delta u_j(t-1)\} \\ & + [N_{i_{k+d_i}}(z^{-1}) + M_{i_{k+d_i}}(z^{-1})F_{i_{k+d_i}}(z^{-1})]y_i(t) \end{aligned} \quad (2.13)$$

令:

$$f_{i_k} = \sum_{j=1}^m \{G_{p_{ij_k}}(z^{-1})\Delta u_j(t-1)\} + [N_{i_{k+d_i}}(z^{-1}) + M_{i_{k+d_i}}(z^{-1})F_{i_{k+d_i}}(z^{-1})]y_i(t) \quad (2.14)$$

则公式(2.13)为:

$$\hat{y}_i(t+k+d_i|t) = \sum_{j=1}^m \{G_{ij_k}(z^{-1})\Delta u_j(t+k-1)\} + f_{i_k} \quad (2.15)$$

2.5 控制律的求解

考虑 k 取 1 到 N_i , ($N_i = N_{2_i} - N_{1_i} + 1$, N_{1_i} 、 N_{2_i} 和 N_i 分别是第 i 个输出变量的最小预测时域、最大预测时域和预测时域), 则有:

$$\begin{aligned} \hat{y}_i(t+d_i+1|t) &= \sum_{j=1}^m \{G_{ij_1}(z^{-1})\Delta u_j(t)\} + f_{i_1} \\ \hat{y}_i(t+d_i+2|t) &= \sum_{j=1}^m \{G_{ij_2}(z^{-1})\Delta u_j(t)\} + f_{i_2} \\ &\vdots \\ \hat{y}_i(t+d_i+N_i|t) &= \sum_{j=1}^m \{G_{ij_{N_i}}(z^{-1})\Delta u_j(t+N_i-1)\} + f_{i_{N_i}} \end{aligned} \quad (2.16)$$

定义: $N_{1_i} = d_i + 1$, 第 i 个输出的预测式可表示为简洁的形式:

$$\hat{\mathbf{Y}}_{i_{N_i}} = \sum_{j=1}^m \{\mathbf{G}_{ij} \mathbf{U}_{ij}\} + \mathbf{f}_i \quad (2.17)$$

其中:

$$\hat{\mathbf{Y}}_{i_{N_i}} = [\hat{y}_i(t+N_{1_i}|t), \hat{y}_i(t+N_{1_i}+1|t), \dots, \hat{y}_i(t+N_{2_i}|t)]^T$$

$$\mathbf{U}_{ij} = [\Delta u_j(t), \Delta u_j(t+1), \dots, \Delta u_j(t+N_i-1)]^T$$

$$\mathbf{f}_i = [f_{i_1}, f_{i_2}, \dots, f_{i_{N_i}}]^T$$

$$\mathbf{G}_{ij} = \begin{bmatrix} g_{ij,1,0} & 0 & \dots & 0 \\ g_{ij,2,1} & g_{ij,2,0} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{ij,N_i,N_i-1} & g_{ij,N_i,N_i-2} & \dots & g_{ij,N_i,0} \end{bmatrix}$$

引入控制时域, 如果从第 $N_{3_j} + 1$ 个采样时刻后, 控制信号保持不变, (其中 N_{3_j} 是第 j 个输入的控制时域, $j=1, \dots, m$), 则(2.17)式可表示为:

$$\hat{\mathbf{Y}}_{i_{N_i}} = \sum_{j=1}^m \{ \mathbf{G}_{ij, N_{3_j}} \mathbf{U}_{j, N_{3_j}} \} + \mathbf{f}_i \quad (2.18)$$

其中: $\mathbf{U}_{j, N_{3_j}} = [\Delta u_j(t), \dots, \Delta u_j(t + N_{3_j})]^T_{(N_{3_j}+1) \times 1}$, $\mathbf{G}_{ij, N_{3_j}}$ 为 \mathbf{G}_{ij} 的 $N_{3_j} + 1$ 列元素构成的子矩阵。

考虑所有 n 个输出的预测式, 由(2.18)式可得:

$$\begin{bmatrix} \hat{\mathbf{Y}}_{1, N_1} \\ \hat{\mathbf{Y}}_{2, N_2} \\ \vdots \\ \hat{\mathbf{Y}}_{n, N_n} \end{bmatrix} = \begin{bmatrix} \mathbf{G}_{11, N_{3_1}} & \mathbf{G}_{12, N_{3_2}} & \dots & \mathbf{G}_{1m, N_{3_m}} \\ \mathbf{G}_{21, N_{3_1}} & \mathbf{G}_{22, N_{3_2}} & \dots & \mathbf{G}_{2m, N_{3_m}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{G}_{n1, N_{3_1}} & \mathbf{G}_{n2, N_{3_2}} & \dots & \mathbf{G}_{nm, N_{3_m}} \end{bmatrix} \begin{bmatrix} \mathbf{U}_{1, N_{3_1}} \\ \mathbf{U}_{2, N_{3_2}} \\ \vdots \\ \mathbf{U}_{m, N_{3_m}} \end{bmatrix} + \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \vdots \\ \mathbf{f}_n \end{bmatrix} \quad (2.19)$$

定义:

$$\hat{\mathbf{Y}} = \begin{bmatrix} \hat{\mathbf{Y}}_{1, N_1} \\ \hat{\mathbf{Y}}_{2, N_2} \\ \vdots \\ \hat{\mathbf{Y}}_{n, N_n} \end{bmatrix}, \quad \mathbf{U}_{N_3} = \begin{bmatrix} \mathbf{U}_{1, N_{3_1}} \\ \mathbf{U}_{2, N_{3_2}} \\ \vdots \\ \mathbf{U}_{m, N_{3_m}} \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \vdots \\ \mathbf{f}_n \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} \mathbf{G}_{11, N_{3_1}} & \mathbf{G}_{12, N_{3_2}} & \dots & \mathbf{G}_{1m, N_{3_m}} \\ \mathbf{G}_{21, N_{3_1}} & \mathbf{G}_{22, N_{3_2}} & \dots & \mathbf{G}_{2m, N_{3_m}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{G}_{n1, N_{3_1}} & \mathbf{G}_{n2, N_{3_2}} & \dots & \mathbf{G}_{nm, N_{3_m}} \end{bmatrix}$$

则(2.19)式可以写成:

$$\hat{\mathbf{Y}} = \mathbf{G} \mathbf{U}_{N_3} + \mathbf{f} \quad (2.20)$$

将目标函数(2.4)写成矩阵形式,

$$J = (\hat{\mathbf{Y}} - \mathbf{W})^T \mathbf{R} (\hat{\mathbf{Y}} - \mathbf{W}) + \mathbf{U}_{N_3}^T \mathbf{Q} \mathbf{U}_{N_3} \quad (2.21)$$

其中:

$$\mathbf{W} = [\mathbf{w}_1^T, \mathbf{w}_2^T, \dots, \mathbf{w}_n^T]^T, \quad \mathbf{w}_i = [w_i(t + N_{1_i}), w_i(t + N_{1_i} + 1), \dots, w_i(t + N_{2_i})]^T \quad (i=1, \dots, n)$$

$$, \mathbf{R} = \text{diag}\{\delta_1, \delta_2, \dots, \delta_n\}, \delta_i = \text{diag}\{\delta_i(N_{1_i}), \delta_i(N_{1_i}+1), \dots, \delta_i(N_{2_i})\} (i=1, \dots, n),$$

$$\mathbf{Q} = \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_m\}, \lambda_j = \text{diag}\{\lambda_j(1), \lambda_j(2), \dots, \lambda_j(N_{3_j})\} (j=1, \dots, m)。$$

将来输出参考值未知时, 参考轨迹可设为 $y_i(t)$ 向着现时刻设定值 $r_i(t)$ 的一阶光滑近似:

$$\mathbf{w}_i = \begin{bmatrix} \alpha_i^{N_{1_i}} & 1 - \alpha_i^{N_{1_i}} \\ \vdots & \vdots \\ \alpha_i^{N_{2_i}} & 1 - \alpha_i^{N_{2_i}} \end{bmatrix} \begin{bmatrix} y_i(t) \\ r_i(t) \end{bmatrix} = \begin{bmatrix} \mathbf{T}_{y_i} & \mathbf{T}_r \end{bmatrix} \begin{bmatrix} y_i(t) \\ r_i(t) \end{bmatrix} = \mathbf{T}_{y_i} y_i(t) + \mathbf{T}_r r_i(t) \quad (2.22)$$

其中 $\alpha_i \in [0, 1]$ 为第 i 个输出的柔化因子。则:

$$\mathbf{W} = \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \vdots \\ \mathbf{w}_n \end{bmatrix} = \begin{bmatrix} \mathbf{T}_{y_1} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_{y_2} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{T}_{y_n} \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_n(t) \end{bmatrix} + \begin{bmatrix} \mathbf{T}_r & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_r & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{T}_r \end{bmatrix} \begin{bmatrix} r_1(t) \\ r_2(t) \\ \vdots \\ r_n(t) \end{bmatrix} \quad (2.23)$$

如果令:

$$\mathbf{r}(t) = \begin{bmatrix} r_1(t) \\ r_2(t) \\ \vdots \\ r_n(t) \end{bmatrix}, \mathbf{T}_y = \begin{bmatrix} \mathbf{T}_{y_1} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_{y_2} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{T}_{y_n} \end{bmatrix}, \mathbf{T}_r = \begin{bmatrix} \mathbf{T}_r & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_r & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{T}_r \end{bmatrix}$$

则有:

$$\mathbf{W} = \mathbf{T}_y \mathbf{y}(t) + \mathbf{T}_r \mathbf{r}(t) = \begin{bmatrix} \mathbf{T}_y & \mathbf{T}_r \end{bmatrix} \begin{bmatrix} \mathbf{y}(t) \\ \mathbf{r}(t) \end{bmatrix} = \mathbf{T} \begin{bmatrix} \mathbf{y}(t) \\ \mathbf{r}(t) \end{bmatrix} \quad (2.24)$$

将(2.20)式代入(2.21)式, 如果没有限制条件, 最优控制增量可表示为:

$$\mathbf{U}_{N_1} = (\mathbf{G}^T \mathbf{R} \mathbf{G} + \mathbf{Q})^{-1} \mathbf{G}^T \mathbf{R} (\mathbf{W} - \mathbf{f}) \quad (2.25)$$

使用滚动控制策略, t 时刻只有 $\Delta u_j(t)$ ($j=1, \dots, m$) 是需要的, 定义矩阵:

$$\mathbf{M} = \begin{bmatrix} \underbrace{1 \ 0 \ \dots \ 0}_{N_{3_1}+1} & \underbrace{0 \ 0 \ \dots \ 0}_{N_{3_2}+1} & \dots & \underbrace{0 \ 0 \ \dots \ 0}_{N_{3_m}+1} \\ \underbrace{0 \ 0 \ \dots \ 0}_{N_{3_1}+1} & \underbrace{1 \ 0 \ \dots \ 0}_{N_{3_2}+1} & \dots & \underbrace{0 \ 0 \ \dots \ 0}_{N_{3_m}+1} \\ \vdots & \vdots & \ddots & \vdots \\ \underbrace{0 \ 0 \ \dots \ 0}_{N_{3_1}+1} & \underbrace{0 \ 0 \ \dots \ 0}_{N_{3_2}+1} & \dots & \underbrace{1 \ 0 \ \dots \ 0}_{N_{3_m}+1} \end{bmatrix}_{m \times \sum_{j=1}^m (N_{3_j}+1)}$$

令: $\mathbf{K} = \mathbf{M}(\mathbf{G}^T \mathbf{R} \mathbf{G} + \mathbf{Q})^{-1} \mathbf{G}^T \mathbf{R}$, 则控制律可表示为:

$$\Delta \mathbf{u}(t) = \mathbf{K}(\mathbf{W} - \mathbf{f}) \quad (2.26)$$

其中: $\Delta \mathbf{u}(t) = [\Delta u_1(t), \Delta u_2(t), \dots, \Delta u_m(t)]^T$ 。即一个线性增益矩阵乘以参考轨迹与输出自由响应之间的预测误差。对于非自适应情况, \mathbf{K} 可以提前算出, 而对于自适应情况, 则在模型更新的每一时刻需要重新计算 \mathbf{K} 。

得到的控制系统框图如图 2.1 所示。

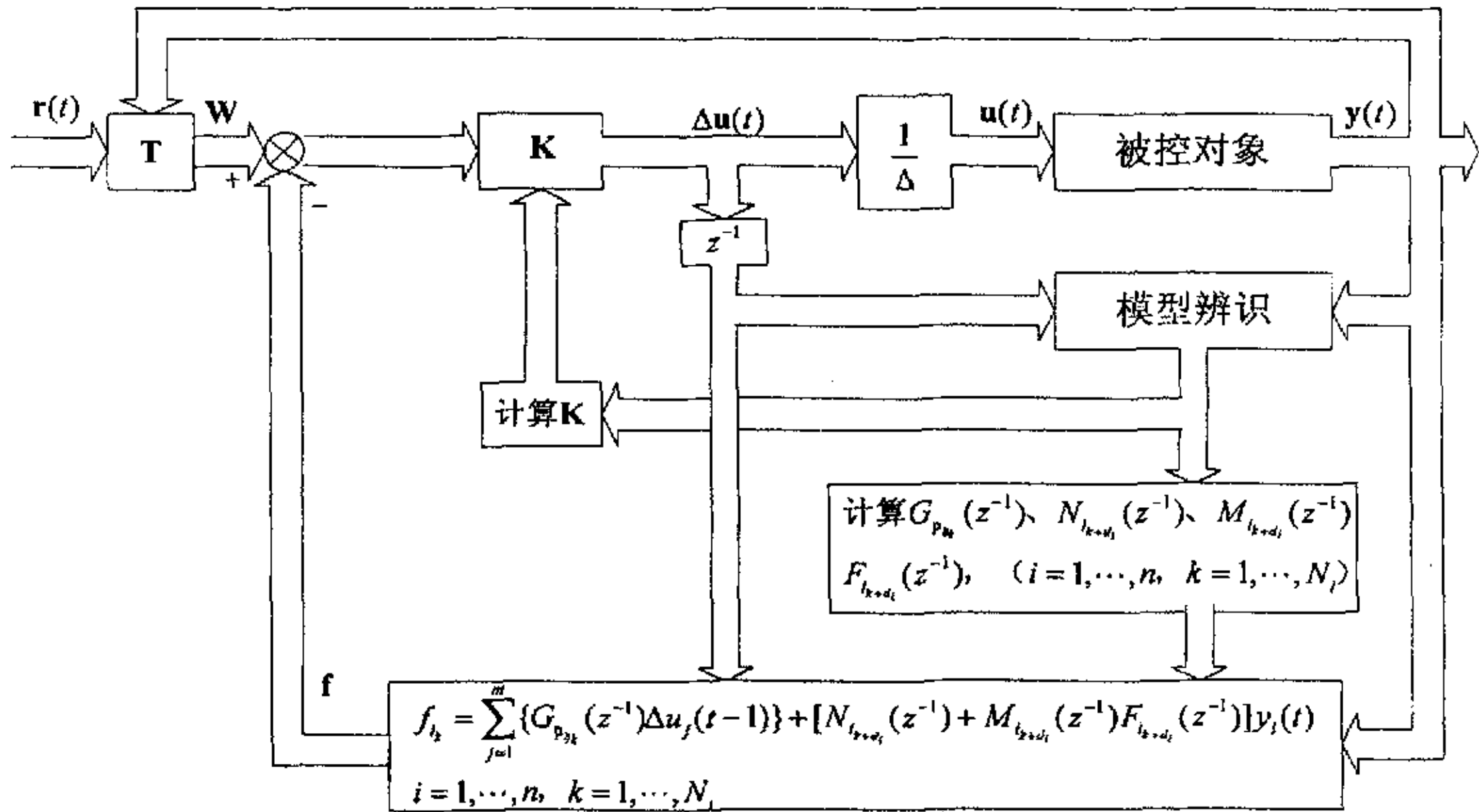


图 2.1 多变量自适应广义预测控制系统框图

2.6 参数辨识

由(2.7)式, 可以把第 i 个输出写出成如下的形式:

$$y_i(t) = \mathbf{H}_i^T(t) \boldsymbol{\theta}_i + e_i(t) \quad (2.27)$$

其中:

$$\mathbf{H}_i(t) = \begin{bmatrix} \left. \begin{array}{c} -y_i(t-1) \\ -y_i(t-2) \\ \vdots \\ -y_i(t-na_i-1) \end{array} \right\} \\ \left. \begin{array}{c} \Delta u_1(t-d_{i1}-1) \\ \Delta u_1(t-d_{i1}-2) \\ \vdots \\ \Delta u_1(t-d_{i1}-1-nb_{i1}) \end{array} \right\} \\ \left. \begin{array}{c} \Delta u_2(t-d_{i2}-1) \\ \Delta u_2(t-d_{i2}-2) \\ \vdots \\ \Delta u_2(t-d_{i2}-1-nb_{i2}) \end{array} \right\} \\ \vdots \\ \left. \begin{array}{c} \Delta u_m(t-d_{im}-1) \\ \Delta u_m(t-d_{im}-2) \\ \vdots \\ \Delta u_m(t-d_{im}-1-nb_{im}) \end{array} \right\} \\ \left. \begin{array}{c} -e_i(t-1) \\ -e_i(t-2) \\ \vdots \\ -e_i(t-nc_{ii}-1) \end{array} \right\} \end{bmatrix}$$

$$\boldsymbol{\theta}_i = \begin{bmatrix} \left. \begin{array}{c} \tilde{a}_{ii_1} \\ \tilde{a}_{ii_2} \\ \vdots \\ \tilde{a}_{ii_{na_i+1}} \end{array} \right\} \\ \left. \begin{array}{c} b_{i1_0} \\ b_{i1_1} \\ \vdots \\ b_{i1_{nb_{i1}}} \end{array} \right\} \\ \left. \begin{array}{c} b_{i2_0} \\ b_{i2_1} \\ \vdots \\ b_{i2_{nb_{i2}}} \end{array} \right\} \\ \vdots \\ \left. \begin{array}{c} b_{im_0} \\ b_{im_1} \\ \vdots \\ b_{im_{nb_{im}}} \end{array} \right\} \\ \left. \begin{array}{c} c_{ii_1} \\ c_{ii_2} \\ \vdots \\ c_{ii_{nc_{ii}}} \end{array} \right\} \end{bmatrix}$$

则所有输出表达式可以表示成如下的矩阵形式:

$$\mathbf{y}(t) = \mathbf{H}(t)\boldsymbol{\theta} + \mathbf{e}(t) \quad (2.28)$$

其中:

$$\mathbf{y}(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_n(t) \end{bmatrix}, \quad \mathbf{e}(t) = \begin{bmatrix} e_1(t) \\ e_2(t) \\ \vdots \\ e_n(t) \end{bmatrix}, \quad \mathbf{H}(t) = \begin{bmatrix} \mathbf{H}_1^T(t) & 0 & \dots & 0 \\ 0 & \mathbf{H}_2^T(t) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{H}_n^T(t) \end{bmatrix}, \quad \boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{\theta}_1 \\ \boldsymbol{\theta}_2 \\ \vdots \\ \boldsymbol{\theta}_n \end{bmatrix}$$

参数辨识采用渐消记忆的递推最小二乘算法^{[109]~[111]},

$$\begin{aligned} \hat{\boldsymbol{\theta}}(t) &= \hat{\boldsymbol{\theta}}(t-1) + \mathbf{K}(t)[\mathbf{y}(t) - \mathbf{H}(t)\hat{\boldsymbol{\theta}}(t-1)] \\ \mathbf{K}(t) &= \mathbf{P}(t-1)\mathbf{H}^T(t)[\mathbf{H}(t)\mathbf{P}(t-1)\mathbf{H}^T(t) + \mu\mathbf{I}_n]^{-1} \\ \mathbf{P}(t) &= [\mathbf{I} - \mathbf{K}(t)\mathbf{H}^T(t)]\mathbf{P}(t-1)/\mu \end{aligned} \quad (2.29)$$

其中: $0 < \mu \leq 1$ 为遗忘因子, 常可选 $0.95 < \mu \leq 1$ 。 $\mathbf{K}(t)$ 为权因子, $\mathbf{P}(t)$ 为正定的协方差阵。初值 $\hat{\boldsymbol{\theta}}(0)$ 可取标称值或零值, $\mathbf{P}(0) = \alpha^2 \mathbf{I}$, α 是一个足够大的正数。

在实际过程中, 为了避免(2.29)引入的矩阵求逆运算, 可以把 m 个输入 n 个输出多变量对象的参数辨识问题分成 n 个 m 个输入 1 个输出的多输入单输出对象的参数辨识问题。第 i 个子辨识问题对应的对象如(2.27)式所示。

2.7 算法实现

根据被控制对象特征, 确定合适的模型结构, 选择合适的预测时域、控制时域、加权因子、柔化因子、模型参数辨识初值、遗忘因子。每个控制周期执行以下计算:

1. 得到输入输出变量新的测量值, 进行模型参数辨识, 得到 $\boldsymbol{\theta}$;
2. 计算 Diophantine 方程得到矩阵 \mathbf{G} 和自由向量 \mathbf{f} ;
3. 计算参考轨迹, 得到矩阵 \mathbf{W} ;
4. 计算控制器系数矩阵 \mathbf{K} ;
5. 计算控制增量 $\Delta \mathbf{u}(t) = \mathbf{K}(\mathbf{W} - \mathbf{f})$;
6. 计算 $\mathbf{u}(t) = \mathbf{u}(t-1) + \Delta \mathbf{u}(t)$ 。

非自适应模式下, 不执行第 1 步。

2.8 小结

如果多变量对象 CARIMA 模型中的多项式矩阵 $\mathbf{A}(z^{-1})$ 与 $\mathbf{C}(z^{-1})$ 能构造成对角形式, 则按本章提出的算法, 多变量广义预测控制的推导与求解过程就能得到简化, 在线计算量也会大幅度下降。算法中显式地考虑了纯滞后项, 可以避开无益的计算, 提高计算效率。这种模型结构下, 参数辨识问题也可以得到简化, 把一个大型参数辨识问题化为 n 个相对较小的参数辨识问题, 避免辨识过程中引入的矩阵求逆。

第三章 多变量广义预测控制器系数求解方法

摘要：在经典控制理论与现代控制理论中，人们习惯于把控制器设计成设计变量与过程已知数据乘积的形式，事实上广义预测控制算法的控制器也可以写成类似形式。通过对广义预测控制算法自由响应项的进一步推导，得到了状态反馈形式的广义预测控制器，控制增量等于控制器系数与参考轨迹及过程历史输入输出数据的乘积，并给出了控制器系数的计算公式。

3.1 引言

在传统的 PID 设计与状态反馈控制设计中，控制器表达成设计参数与过程信号的乘积形式。例如，PID 控制器，其离散形式为：

$$\Delta u(t) = K_p[e(t) - e(t-1)] + \frac{K_p T}{T_i} e(t) + \frac{K_p T_d}{T} [e(t) - 2e(t-1) + e(t-2)] \quad (3.1)$$

其中， $e(t) = r(t) - y(t)$ 为 t 时刻设定值与对象输出的偏差， $r(t)$ 为设定值， $y(t)$ 为对象输出， $\Delta u(t)$ 为控制增量， T 为采样周期， K_p 为比例系数， T_i 为积分时间， T_d 为微分时间。

在控制系统的状态空间设计中^[112]，考虑系统：

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}u \\ y &= \mathbf{C}\mathbf{x} \end{aligned} \quad (3.2)$$

式中： \mathbf{x} 为 n 维向量状态向量， u 为控制信号， \mathbf{A} 、 \mathbf{B} 、 \mathbf{C} 分别为 $n \times n$ 、 $n \times 1$ 、 $1 \times n$ 维矩阵。则状态反馈控制为：

$$u(t) = -\mathbf{K}_s \mathbf{x} \quad (3.3)$$

其中： \mathbf{K}_s 为状态反馈增益矩阵。

能否将广义预测控制器写成类似的形式呢？

不考虑硬约束条件, 第二章推导的广义预测控制算法控制增量可以表示为线性的形式, 等于增益系数矩阵乘以参考轨迹与自由响应之差, 如(2.26)所示:

$$\Delta u(t) = K(W - f) \quad (3.4)$$

其中: 增益系数矩阵 K 只与对象的模型参数及设计参数有关, 而自由响应 f 却涉及到过程的输入输出数据。进一步分析, 发现控制增量总是由模型参数、设计参数、参考轨迹、过程输入输出历史信息组成。如果能把控制量写成线性增益系数与参考轨迹、过程历史信息向量乘积的形式, 不但可以省掉输出预测器、简化设计思路, 而且更符合设计习惯、便于工程人员更好的实现控制算法。特别是在非自适应情况下, 只需离线计算控制器系数, 在线实现就如常规 PID 控制器一样简单。

3.2 控制器系数的求解方法

对于(2.3)式的过程模型, 引入(2.4)式的目标函数, 根据第二章的推导过程, 可以得到(2.26)式所示的控制律, 其中控制器系数矩阵 K 只与模型参数及设计参数相关。

考虑(2.14)式:

$$f_{i_k} = \sum_{j=1}^m \{G_{p_{jk}}(z^{-1})\Delta u_j(t-1)\} + [N_{i_{k+d_i}}(z^{-1}) + M_{i_{k+d_i}}(z^{-1})F_{i_{k+d_i}}(z^{-1})]y_i(t) \quad (3.5)$$

令:

$$G_{p_{jk}}(z^{-1}) = f_{u_{jk0}} + f_{u_{jk1}}z^{-1} + \cdots + f_{u_{jkk+d_i+d_{ij}+nb_{ij}-1}}z^{-(k+d_i+d_{ij}+nb_{ij}-2)} \quad (3.6)$$

$$[N_{i_{k+d_i}}(z^{-1}) + M_{i_{k+d_i}}(z^{-1})F_{i_{k+d_i}}(z^{-1})] = f_{y_{ik0}} + f_{y_{ik1}}z^{-1} + \cdots + f_{y_{ikk+d_i+na_{ii}-1}}z^{-(k+d_i+na_{ii}-1)} \quad (3.7)$$

则(3.5)式变成:

$$\begin{aligned} f_{i_k} = & \sum_{j=1}^m \{f_{u_{jk0}}\Delta u_j(t-1) + f_{u_{jk1}}\Delta u_j(t-2) + \cdots + f_{u_{jkk+d_i+d_{ij}+nb_{ij}-1}}\Delta u_j(t-k-d_i-d_{ij}-nb_{ij}+1)\} \\ & + f_{y_{ik0}}y_i(t) + f_{y_{ik1}}y_i(t-1) + \cdots + f_{y_{ikk+d_i+na_{ii}-1}}y_i(t-k-d_i-na_{ii}+1) \end{aligned} \quad (3.8)$$

根据 f_i 的定义, 有:

$$\begin{aligned}
\mathbf{f}_i &= \begin{bmatrix} f_{i_1} \\ f_{i_2} \\ \vdots \\ f_{i_{N_i}} \end{bmatrix} \\
&= \sum_{j=1}^m \left\{ \begin{bmatrix} f_{u_{i10}} & f_{u_{i11}} & \cdots & 0 \\ f_{u_{i20}} & f_{u_{i21}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ f_{u_{iN_i0}} & f_{u_{iN_i1}} & \cdots & f_{u_{iN_iN_i+d_i+d_{ij}+nb_{ij}-2}} \end{bmatrix} \begin{bmatrix} \Delta u_j(t-1) \\ \Delta u_j(t-2) \\ \vdots \\ \Delta u_j(t-N_i-d_i-d_{ij}-nb_{ij}+1) \end{bmatrix} \right. \\
&\quad \left. + \begin{bmatrix} f_{y_{i10}} & f_{y_{i11}} & \cdots & 0 \\ f_{y_{i20}} & f_{y_{i21}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ f_{y_{iN_i0}} & f_{y_{iN_i1}} & \cdots & f_{y_{iN_iN_i+d_i+na_{ii}-1}} \end{bmatrix} \begin{bmatrix} y_i(t) \\ y_i(t-1) \\ \vdots \\ y_i(t-N_i-d_i-na_{ii}+1) \end{bmatrix} \right\} \quad (3.9)
\end{aligned}$$

令:

$$\mathbf{f}_{i_{uj}} = \begin{bmatrix} f_{u_{i10}} & f_{u_{i11}} & \cdots & 0 \\ f_{u_{i20}} & f_{u_{i21}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ f_{u_{iN_i0}} & f_{u_{iN_i1}} & \cdots & f_{u_{iN_iN_i+d_i+d_{ij}+nb_{ij}-2}} \end{bmatrix}$$

$$\mathbf{f}_{i_u} = [\mathbf{f}_{i_{u1}} \quad \mathbf{f}_{i_{u2}} \quad \cdots \quad \mathbf{f}_{i_{um}}]$$

$$\mathbf{f}_{i_y} = \begin{bmatrix} f_{y_{i10}} & f_{y_{i11}} & \cdots & 0 \\ f_{y_{i20}} & f_{y_{i21}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ f_{y_{iN_i0}} & f_{y_{iN_i1}} & \cdots & f_{y_{iN_iN_i+d_i+na_{ii}-1}} \end{bmatrix}$$

$$\mathbf{y}_{i_p} = [y_i(t), y_i(t-1), \cdots, y_i(t-N_i-na_{ii}-d_i+1)]^T$$

$$\mathbf{u}_{p_{ij}} = [\Delta u_j(t-1), \cdots, \Delta u_j(t-N_i-d_i-d_{ij}-nb_{ij}+1)]^T$$

$$\mathbf{u}_{p_i} = [\mathbf{u}_{p_{i1}}^T, \mathbf{u}_{p_{i2}}^T, \cdots, \mathbf{u}_{p_{im}}^T]^T$$

则有:

$$\mathbf{f}_i = \mathbf{f}_{i_u} \mathbf{u}_{p_i} + \mathbf{f}_{i_y} \mathbf{y}_{i_p} \quad (3.10)$$

其中: \mathbf{f}_{i_u} , \mathbf{f}_{i_y} 的元素只与过程模型参数有关。

(3.10)式中, i 从 1 到 n , 有:

$$\begin{aligned}
\mathbf{f}_1 &= \mathbf{f}_{1_u} \mathbf{u}_{p_1} + \mathbf{f}_{1_y} \mathbf{y}_{1_p} \\
\mathbf{f}_2 &= \mathbf{f}_{2_u} \mathbf{u}_{p_2} + \mathbf{f}_{2_y} \mathbf{y}_{2_p} \\
&\vdots \\
\mathbf{f}_n &= \mathbf{f}_{n_u} \mathbf{u}_{p_n} + \mathbf{f}_{n_y} \mathbf{y}_{n_p}
\end{aligned} \quad (3.11)$$

根据 \mathbf{f} 的定义, 有:

$$\mathbf{f} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \vdots \\ \mathbf{f}_n \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{1_u} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{f}_{2_u} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{f}_{n_u} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{p_1} \\ \mathbf{u}_{p_2} \\ \vdots \\ \mathbf{u}_{p_n} \end{bmatrix} + \begin{bmatrix} \mathbf{f}_{1_y} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{f}_{2_y} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{f}_{n_y} \end{bmatrix} \begin{bmatrix} \mathbf{y}_{1_p} \\ \mathbf{y}_{2_p} \\ \vdots \\ \mathbf{y}_{n_p} \end{bmatrix} \quad (3.12)$$

令:

$$\mathbf{u}_p = [\mathbf{u}_{p_1}^T, \mathbf{u}_{p_2}^T, \cdots, \mathbf{u}_{p_n}^T]^T$$

$$\mathbf{y}_p = [\mathbf{y}_{1_p}^T, \mathbf{y}_{2_p}^T, \cdots, \mathbf{y}_{n_p}^T]^T$$

$$\mathbf{f}_u = \text{diag}\{\mathbf{f}_{1_u}, \mathbf{f}_{2_u}, \cdots, \mathbf{f}_{n_u}\}$$

$$\mathbf{f}_y = \text{diag}\{\mathbf{f}_{1_y}, \mathbf{f}_{2_y}, \cdots, \mathbf{f}_{n_y}\}$$

则

$$\mathbf{f} = \mathbf{f}_u \mathbf{u}_p + \mathbf{f}_y \mathbf{y}_p \quad (3.13)$$

将公式(3.13)代入公式(3.4), 则控制律变为:

$$\Delta \mathbf{u}(t) = \mathbf{K} \mathbf{W} - \mathbf{K} \mathbf{f}_u \mathbf{u}_p - \mathbf{K} \mathbf{f}_y \mathbf{y}_p = [\mathbf{K}, -\mathbf{K} \mathbf{f}_u, -\mathbf{K} \mathbf{f}_y] [\mathbf{W}^T, \mathbf{u}_p^T, \mathbf{y}_p^T]^T \quad (3.14)$$

将来输出参考值未知时, 参考轨迹可设为 $y_i(t)$ 向着现时刻设定值 $r_i(t)$ 的一阶光滑近似, 根据(2.24)式, 则有:

$$\mathbf{W} = \mathbf{T}_y \mathbf{y}(t) + \mathbf{T}_r \mathbf{r}(t) = \begin{bmatrix} \mathbf{T}_y & \mathbf{T}_r \end{bmatrix} \begin{bmatrix} \mathbf{y}(t) \\ \mathbf{r}(t) \end{bmatrix} = \mathbf{T} \begin{bmatrix} \mathbf{y}(t) \\ \mathbf{r}(t) \end{bmatrix} \quad (3.15)$$

将(3.15)代入(3.14)则:

$$\begin{aligned} \Delta \mathbf{u}(t) &= \mathbf{K} \mathbf{T}_y \mathbf{y}(t) + \mathbf{K} \mathbf{T}_r \mathbf{r}(t) - \mathbf{K} \mathbf{f}_u \mathbf{u}_p - \mathbf{K} \mathbf{f}_y \mathbf{y}_p \\ &= [\mathbf{K} \mathbf{T}_y, \mathbf{K} \mathbf{T}_r, -\mathbf{K} \mathbf{f}_u, -\mathbf{K} \mathbf{f}_y] [\mathbf{y}^T(t), \mathbf{r}^T(t), \mathbf{u}_p^T, \mathbf{y}_p^T]^T \end{aligned} \quad (3.16)$$

如果令:

$$\mathbf{S} = \begin{bmatrix} \underbrace{1 \ 0 \ \cdots \ 0}_{N_1 + na_{11} + d_1} & \underbrace{0 \ 0 \ \cdots \ 0}_{N_2 + na_{22} + d_2} & \cdots & \underbrace{0 \ 0 \ \cdots \ 0}_{N_n + na_{nn} + d_n} \\ \underbrace{0 \ 0 \ \cdots \ 0}_{N_1 + na_{11} + d_1} & \underbrace{1 \ 0 \ \cdots \ 0}_{N_2 + na_{22} + d_2} & \cdots & \underbrace{0 \ 0 \ \cdots \ 0}_{N_n + na_{nn} + d_n} \\ \vdots & \vdots & \ddots & \vdots \\ \underbrace{0 \ 0 \ \cdots \ 0}_{N_1 + na_{11} + d_1} & \underbrace{0 \ 0 \ \cdots \ 0}_{N_2 + na_{22} + d_2} & \cdots & \underbrace{1 \ 0 \ \cdots \ 0}_{N_n + na_{nn} + d_n} \end{bmatrix}$$

则公式(3.16)可变为:

$$\Delta \mathbf{u}(t) = \mathbf{K} \mathbf{T}_r \mathbf{r}(t) + [\mathbf{K} \mathbf{T}_y \mathbf{S} - \mathbf{K} \mathbf{f}_y] \mathbf{y}_p - \mathbf{K} \mathbf{f}_u \mathbf{u}_p \quad (3.17)$$

令：

$$\begin{aligned} l_r &= KT_r \\ l_y &= KT_y S - Kf_y \\ l_u &= -K_u f_u \end{aligned} \quad (3.18)$$

则(3.17)变为：

$$\Delta u(t) = l_r r(t) + l_y y_p + l_u u_p \quad (3.19)$$

其中控制器系数 l_r 、 l_u 和 l_y 只与过程模型参数及设计参数有关，对于非自适应控制模式，可以离线计算得到，对于自适应情况，则在模型参数更新时需要重新计算。

控制系统结构如图 3.1 所示。

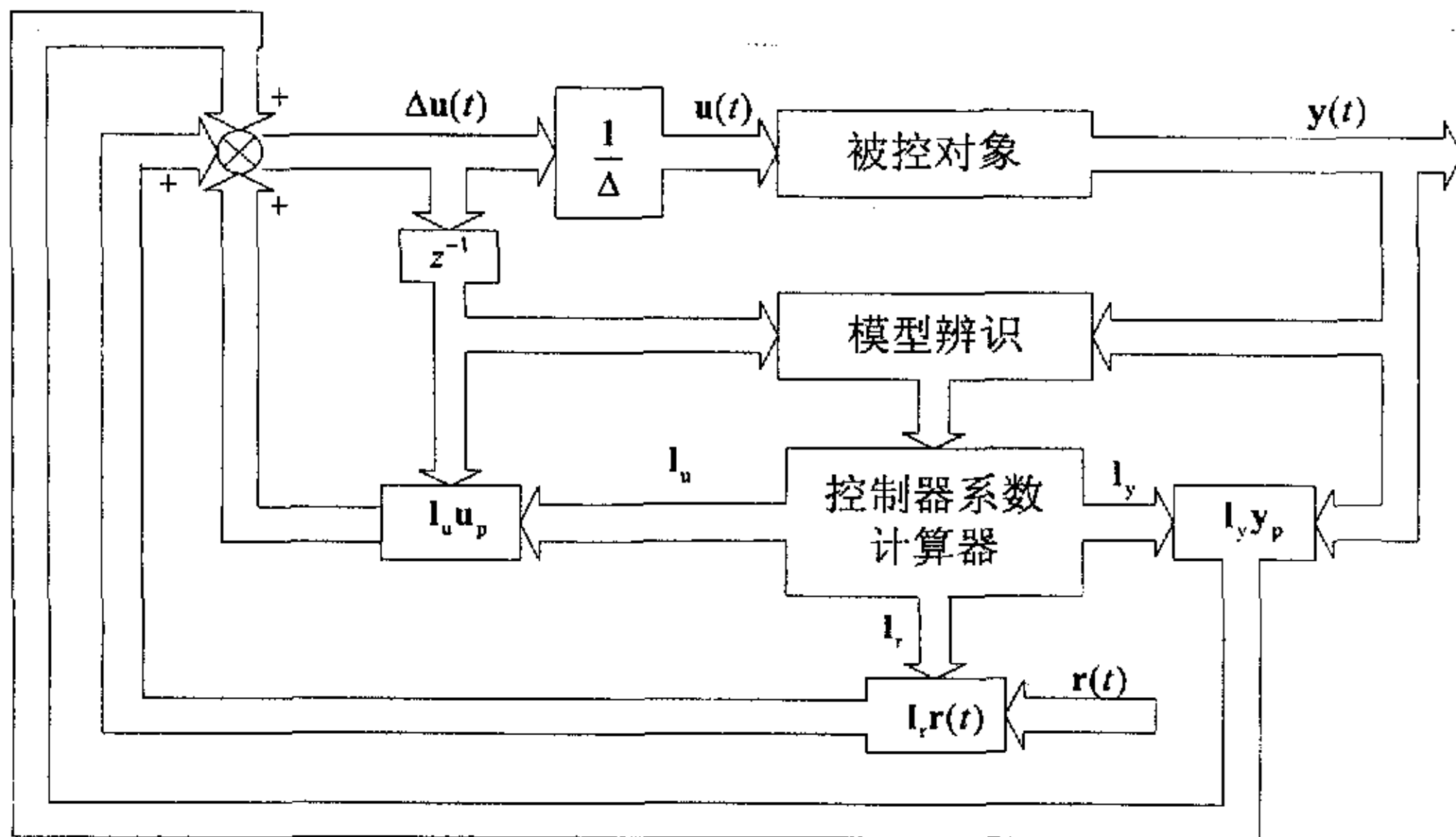


图 3.1 多变量自适应广义预测控制系统方框图

从图 3.1 可以看出，按本章所示方法整理后，所得到的控制系统与常规状态反馈控制系统极为相似。与图 2.1 相比，控制系统结构得到了简化。

3.3 算法实现

根据被控制对象特征，确定合适的模型结构，选择合适的预测时域、控制时域、加权因子、柔化因子、模型参数辨识初值、遗忘因子。每个控制周期执行以下计算：

1. 得到输入输出变量新的测量值，进行模型参数辨识，得到 θ ；
2. 计算 Diophantine 方程得到矩阵 G 、 f_u 、和 f_y ；
3. 计算矩阵 T_y 和 T_r ；
4. 计算控制器系数矩阵 l_r 、 l_u 和 l_y ；
5. 计算控制增量 $\Delta u(t) = l_r r(t) + l_y y_p + l_u u_p$ ；
6. 计算 $u(t) = u(t-1) + \Delta u(t)$ 。

非自适应模型下，由于控制器系数可以提前计算出，在线时只要执行第 5 和第 6 步，因此计算与实现都比第二章中的方法容易。

3.4 小结

本章通过对预测控制算法中自由响应项的处理，在第二章所示方法的基础上，把多变量广义预测控制方法表达成常规控制器的形式，简化了控制系统的结构，便于理解与应用。在自适应控制模式下，它与第二章中的方法具有相同的计算量；在非自适应模式下，相对于第二章的方法，本章方法可省去对自由响应项的求解，在线计算减少，实现方式更加简单。

第四章 多变量广义预测控制器系数 直接求解方法

摘要：利用多变量 CARIMA 模型直接进行递推，得到了每个输出的预测表达式，将其组合在一起便得到了整个输出预测表达式，避开了 Diophantine 方程的求解。得到了状态反馈形式的控制器，给出了控制器系数的求解公式，控制器系数维数只与过程模型的阶次相关，结构简单，计算量少。通过比较运算量，对算法的快速性进行了说明。

4.1 引言

第三章的推导过程说明，多变量广义预测控制器可以写成常规控制器的形式：

$$\Delta \mathbf{u}(t) = \mathbf{l}_r \mathbf{r}(t) + \mathbf{l}_y \mathbf{y}_p + \mathbf{l}_u \mathbf{u}_p \quad (4.1)$$

上式中控制器系数 \mathbf{l}_y 和 \mathbf{l}_u 的个数与控制时域 N 大小有关，如果 N 很大，则控制器的系数就会很多。另外，第三章算法中控制器系数的求解都得需要求解 Diophantine 方程，自适应情况下计算量依然繁重。

文献[88][89]提出通过递推的形式求得输出的预测值，不需要求解 Diophantine 方程，但其求解过程依然按照常规步骤，分为自由响应项求解与控制器增益系数矩阵的求解，设计与实现不够直观。本章直接利用过程模型(2.3)式进行递推，先得到单个输出的预测表达式，然后将它们组合在一起，便构成了整个输出表达式，将其代入目标函数(2.4)式便得到了控制器系数与模型参数之间的计算公式。

4.2 第 i 个输出预测表达式

由(2.5)式得：

$$\tilde{A}_{ii}(z^{-1})y_i(t) = \sum_{j=1}^m \{z^{-d_{ij}} B_{ij}(z^{-1})\Delta u_j(t-1)\} + C_{ii}(z^{-1})e_i(t) \quad (4.2)$$

其中 $\tilde{A}_{ii}(z^{-1}) = \Delta A_{ii}(z^{-1}) = 1 + \tilde{a}_{ii_1} z^{-1} + \tilde{a}_{ii_2} z^{-2} + \cdots + \tilde{a}_{ii_{na_{ii}}} z^{-na_{ii}} + \tilde{a}_{ii_{na_{ii}+1}} z^{-na_{ii}-1}$ 。

由(4.2)式得:

$$y_i(t) = - \sum_{k=1}^{na_{ii}+1} \tilde{a}_{ii_k} y_i(t-k) + \sum_{j=1}^m \left\{ \sum_{k=0}^{nb_{ij}} b_{ij_k} \Delta u_j(t-d_{ij}-k-1) \right\} + e_i(t) + \sum_{k=1}^{nc_{ii}} c_{ii_k} e_i(t-k) \quad (4.3)$$

递推 $d_i + N_i$ 步 (其中 $d_i = \min_j d_{ij}$ 是第 i 个输出对所有输入的最小时滞,

$N_i = N_{2i} - N_{1i} + 1$, N_{1i} 、 N_{2i} 和 N_i 分别是第 i 个输出变量的最小预测时域、最大预测时域和预测时域), 得到:

$$\begin{bmatrix} \mathbf{Y}_{i_{d_i}} \\ \mathbf{Y}_{i_{N_i}} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{ii_{pd_i}} \\ \mathbf{A}_{ii_{pN_i}} \end{bmatrix} \mathbf{Y}_{i_p} + \sum_{j=1}^m \left\{ \begin{bmatrix} \mathbf{B}_{ij_{pd_i}} \\ \mathbf{B}_{ij_{pN_i}} \end{bmatrix} \mathbf{U}_{ij_p} + \begin{bmatrix} \mathbf{0} \\ \mathbf{B}_{ij_{N_i}} \end{bmatrix} \mathbf{U}_{ij} \right\} + \begin{bmatrix} \mathbf{C}_{ii_{pd_i}} \\ \mathbf{C}_{ii_{pN_i}} \end{bmatrix} \mathbf{E}_{ii_p} + \begin{bmatrix} \mathbf{C}_{ii_{d_i}} \\ \mathbf{C}_{ii_{N_i}} \end{bmatrix} \mathbf{E}_{ii} \quad (4.4)$$

其中:

$$\mathbf{Y}_{i_{d_i}} = [y_i(t+1), \dots, y_i(t+d_i)]^T$$

$$\mathbf{Y}_{i_{N_i}} = [y_i(t+d_i+1), \dots, y_i(t+d_i+N_i)]^T$$

$$\mathbf{Y}_{i_p} = [y_i(t), \dots, y_i(t-na_{ii})]^T$$

$$\mathbf{U}_{ij_p} = [\Delta u_j(t-1), \dots, \Delta u_j(t-d_{ij}-nb_{ij})]^T,$$

$$\mathbf{U}_{ij} = [\Delta u_j(t), \dots, \Delta u_j(t+d_i+N_i-d_{ij}-1)]^T$$

$$\mathbf{E}_{ii_p} = [e_i(t), \dots, e_i(t-nc_{ii}+1)]^T$$

$$\mathbf{E}_{ii} = [e_i(t+1), \dots, e_i(t+d_i+N_i)]^T$$

如果用 \mathbf{X} 分别代表 $[\mathbf{A}_{ii_{pd_i}}^T \quad \mathbf{A}_{ii_{pN_i}}^T]^T$ 、 $[\mathbf{B}_{ij_{pd_i}}^T \quad \mathbf{B}_{ij_{pN_i}}^T]^T$ 和 $[\mathbf{C}_{ii_{pd_i}}^T \quad \mathbf{C}_{ii_{pN_i}}^T]^T$, X_c 代表矩阵 \mathbf{X} 的列数, 则矩阵 \mathbf{X} 中的元素可按如下方式产生:

$$\mathbf{X}(i, j) = \mathbf{T}(i+j-1) + \sum_{k=1}^{i-1} \alpha_k \mathbf{X}(i-k, j), (1 \leq i \leq N_i + d_i, 1 \leq j \leq X_c) \quad (4.5)$$

其中:

$$\mathbf{T} = \begin{cases} [-\tilde{a}_{i_1}, -\tilde{a}_{i_2}, \dots, -\tilde{a}_{i_{na_i+1}}, \underbrace{0, 0, \dots, 0}_{N_i+d_i-1}], \mathbf{X} = [\mathbf{A}_{i_{pd_i}}^T & \mathbf{A}_{i_{pN_i}}^T]^T \\ [b_{ij_1}, \dots, b_{ij_{nb_j}}, \underbrace{0, 0, \dots, 0}_{N_i+d_i-1}], d_{ij} = 0 \\ [0, 0, \dots, 0, b_{ij_0}, b_{ij_1}, \dots, b_{ij_{nb_j}}, \underbrace{0, 0, \dots, 0}_{N_i+d_i-1}], d_{ij} > 0 \end{cases}, \mathbf{X} = [\mathbf{B}_{i_{pd_i}}^T & \mathbf{B}_{i_{pN_i}}^T]^T$$

$$[c_{i_1}, c_{i_2}, \dots, c_{i_{nc_i}}, \underbrace{0, 0, \dots, 0}_{N_i+d_i-1}], \mathbf{X} = [\mathbf{C}_{i_{pd_i}}^T & \mathbf{C}_{i_{pN_i}}^T]^T$$

$$\alpha_k = \begin{cases} -\tilde{a}_{i_k}, 1 \leq k \leq na_i + 1 \\ 0, k > na_i + 1 \end{cases}$$

$\mathbf{B}_{i_{pN_i}} = [\mathbf{0}^T, \bar{\mathbf{B}}_{i_{pN_i}}^T]^T$, $\bar{\mathbf{B}}_{i_{pN_i}}$ 和 $[\mathbf{C}_{i_{pd_i}}^T \quad \mathbf{C}_{i_{pN_i}}^T]^T$ 是下三角矩阵, 且其第 $i+1$ 行的第 2 个至第 $i+1$ 个元素分别是第 i 行的第 1 个至第 i 个元素, 因此只有第一列的元素需要求解。如果用 \mathbf{X} 分别代表 $\bar{\mathbf{B}}_{i_{pN_i}}$ 和 $[\mathbf{C}_{i_{pd_i}}^T \quad \mathbf{C}_{i_{pN_i}}^T]^T$, X_r 代表矩阵 \mathbf{X} 的行数, 则:

$$\mathbf{X}(i, 1) = \mathbf{T}(i) + \sum_{k=1}^{i-1} \alpha_k \mathbf{X}(i-k, 1), (1 \leq i \leq X_r) \quad (4.6)$$

$$\text{其中: } \mathbf{T} = \begin{cases} [b_{ij_0}, b_{ij_1}, \dots, b_{ij_{nb_j}}, \underbrace{0, 0, \dots, 0}_{N_i+d_i-d_{ij}-nb_j-1}], \mathbf{X} = \bar{\mathbf{B}}_{i_{pN_i}} \\ [1, c_{i_1}, \dots, c_{i_{nc_i}}, \underbrace{0, 0, \dots, 0}_{N_i+d_i-nc_i-1}], \mathbf{X} = [\mathbf{C}_{i_{pd_i}}^T \quad \mathbf{C}_{i_{pN_i}}^T]^T \end{cases}$$

以上计算公式由公式(4.3)直接递推而来, 如果需要证明, 可用归纳法。

采用期望算子, 并考虑到公式(4.4) \mathbf{E}_{ii} 中的元素都在将来时刻, 则输出的期望值为:

$$\begin{bmatrix} \hat{\mathbf{Y}}_{i_{d_i}} \\ \hat{\mathbf{Y}}_{i_{N_i}} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{i_{pd_i}} \\ \mathbf{A}_{i_{pN_i}} \end{bmatrix} \mathbf{Y}_{i_p} + \sum_{j=1}^m \left\{ \begin{bmatrix} \mathbf{B}_{i_{pd_i}} \\ \mathbf{B}_{i_{pN_i}} \end{bmatrix} \mathbf{U}_{ij_p} + \begin{bmatrix} \mathbf{0} \\ \mathbf{B}_{i_{pN_i}} \end{bmatrix} \mathbf{U}_{ij} \right\} + \begin{bmatrix} \mathbf{C}_{i_{pd_i}} \\ \mathbf{C}_{i_{pN_i}} \end{bmatrix} \mathbf{E}_{i_{ij_p}} \quad (4.7)$$

其中:

$$\hat{\mathbf{Y}}_{i_{d_i}} = [\hat{y}_i(t+1|t), \dots, \hat{y}_i(t+d_i|t)]^T,$$

$$\hat{\mathbf{Y}}_{i_{N_i}} = [\hat{y}_i(t+d_i+1|t), \dots, \hat{y}_i(t+d_i+N_i|t)]^T.$$

由于 t 时刻的输入至少要经过 d_i 个时刻后才能在第 i 个输出中反映出来, 取

$N_{i_1} = d_i + 1$, $N_{2_i} = d_i + N_i$, 从公式(4.7)可得:

$$\hat{\mathbf{Y}}_{i_{N_i}} = \mathbf{A}_{i_{pN_i}} \mathbf{Y}_{i_p} + \sum_{j=1}^m \left\{ \mathbf{B}_{ij_{pN_i}} \mathbf{U}_{ij_p} + \mathbf{B}_{ij_{N_i}} \mathbf{U}_{ij} \right\} + \mathbf{C}_{i_{pN_i}} \mathbf{E}_{i_p} \quad (4.8)$$

4.3 控制律推导

引入控制时域 N_{3_j} , 第 j 个输入从第 $t + N_{3_j} + 1$ (约定

$N_{3_j} + 1 \leq d_i + N_i - d_{ij} - 1$) 个采样时刻起, 控制信号保持不变, 即

$\Delta u_j(t+k) = 0, k > N_{3_j}$, 则公式(4.8)可表示为:

$$\hat{\mathbf{Y}}_{i_{N_i}} = \mathbf{A}_{i_{pN_i}} \mathbf{Y}_{i_p} + \sum_{j=1}^m \left\{ \mathbf{B}_{ij_{pN_i}} \mathbf{U}_{ij_p} + \mathbf{B}_{ij_{N_i N_{3_j}}} \mathbf{U}_{j_{N_{3_j}}} \right\} + \mathbf{C}_{i_{pN_i}} \mathbf{E}_{i_p} \quad (4.9)$$

其中: $\mathbf{U}_{j_{N_{3_j}}} = [\Delta u_j(t), \dots, \Delta u_j(t + N_{3_j})]^T_{(N_{3_j}+1) \times 1}$, $\mathbf{B}_{ij_{N_i N_{3_j}}}$ 为 $\mathbf{B}_{ij_{N_i}}$ 前 $N_{3_j} + 1$ 列元素构成

的子矩阵。

考虑所有 n 个输出的预测式, 由(4.9)式可得:

$$\hat{\mathbf{Y}} = \mathbf{A}_{pN} \mathbf{Y}_p + \mathbf{B}_{pN} \mathbf{U}_p + \mathbf{B}_{N_3} \mathbf{U}_{N_3} + \mathbf{C}_{pN} \mathbf{E}_p \quad (4.10)$$

其中:

$$\begin{aligned}
\hat{\mathbf{Y}} &= \begin{bmatrix} \hat{\mathbf{Y}}_{1N_1} \\ \hat{\mathbf{Y}}_{2N_2} \\ \vdots \\ \hat{\mathbf{Y}}_{nN_n} \end{bmatrix}, \quad \mathbf{Y}_p = \begin{bmatrix} \mathbf{Y}_{1p} \\ \mathbf{Y}_{2p} \\ \vdots \\ \mathbf{Y}_{np} \end{bmatrix}, \quad \mathbf{U}_p = \begin{bmatrix} \mathbf{U}_{11p} \\ \mathbf{U}_{12p} \\ \vdots \\ \mathbf{U}_{1mp} \\ \mathbf{U}_{21p} \\ \mathbf{U}_{22p} \\ \vdots \\ \mathbf{U}_{2mp} \\ \vdots \\ \mathbf{U}_{n1p} \\ \mathbf{U}_{n2p} \\ \vdots \\ \mathbf{U}_{nmp} \end{bmatrix}, \quad \mathbf{U}_{N_3} = \begin{bmatrix} \mathbf{U}_{1N_3} \\ \mathbf{U}_{2N_3} \\ \vdots \\ \mathbf{U}_{mN_3} \end{bmatrix}, \\
\mathbf{E}_p &= \begin{bmatrix} \mathbf{E}_{11p} \\ \mathbf{E}_{22p} \\ \vdots \\ \mathbf{E}_{nnp} \end{bmatrix}, \quad \mathbf{A}_{pN} = \begin{bmatrix} \mathbf{A}_{11pN_1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{22pN_2} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{A}_{nnpN_n} \end{bmatrix}, \quad \mathbf{B}_{pN} = [\mathbf{B}_{pN_1} \quad \mathbf{B}_{pN_2} \quad \cdots \quad \mathbf{B}_{pN_n}], \\
\mathbf{B}_{pN_i} &= \begin{bmatrix} \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{B}_{i1pN_i} & \mathbf{B}_{i2pN_i} & \cdots & \mathbf{B}_{im_pN_i} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}, \quad \mathbf{B}_{N_3} = \begin{bmatrix} \mathbf{B}_{11N_1N_3} & \mathbf{B}_{12N_1N_3} & \cdots & \mathbf{B}_{1mN_1N_3} \\ \mathbf{B}_{21N_2N_3} & \mathbf{B}_{22N_2N_3} & \cdots & \mathbf{B}_{2mN_2N_3} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{B}_{n1N_nN_3} & \mathbf{B}_{n2N_nN_3} & \cdots & \mathbf{B}_{nmN_nN_3} \end{bmatrix}, \\
\mathbf{C}_{pN} &= \begin{bmatrix} \mathbf{C}_{11pN_1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{22pN_2} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{C}_{nnpN_n} \end{bmatrix}.
\end{aligned}$$

将目标函数(2.4)式写成(2.21)式的矩阵形式:

$$J = (\hat{\mathbf{Y}} - \mathbf{W})^T \mathbf{R} (\hat{\mathbf{Y}} - \mathbf{W}) + \mathbf{U}_{N_3}^T \mathbf{Q} \mathbf{U}_{N_3} \quad (4.11)$$

将(4.10)式代入(4.11)式, 如果没有限制条件, 最优控制增量可表示为:

$$U_{N_3} = [B_{N_3}^T R B_{N_3} + Q]^{-1} B_{N_3}^T R (W - A_{pN} Y_p - B_{pN} U_p - C_{pN} E_p) \quad (4.12)$$

使用滚动控制策略, t 时刻只有 $\Delta u_j(t), j=1, \dots, m$ 是需要的, 定义矩阵:

$$M = \begin{bmatrix} \underbrace{1 \ 0 \ \dots \ 0}_{N_{3_1}+1} & \underbrace{0 \ 0 \ \dots \ 0}_{N_{3_2}+1} & \dots & \underbrace{0 \ 0 \ \dots \ 0}_{N_{3_m}+1} \\ \underbrace{0 \ 0 \ \dots \ 0}_{N_{3_1}+1} & \underbrace{1 \ 0 \ \dots \ 0}_{N_{3_2}+1} & \dots & \underbrace{0 \ 0 \ \dots \ 0}_{N_{3_m}+1} \\ \vdots & \vdots & \ddots & \vdots \\ \underbrace{0 \ 0 \ \dots \ 0}_{N_{3_1}+1} & \underbrace{0 \ 0 \ \dots \ 0}_{N_{3_2}+1} & \dots & \underbrace{1 \ 0 \ \dots \ 0}_{N_{3_m}+1} \end{bmatrix} \quad (4.13)$$

令: $\Delta u(t) = [\Delta u_1(t), \Delta u_2(t), \dots, \Delta u_m(t)]^T$, 则:

$$\Delta u(t) = M[B_{N_3}^T R B_{N_3} + Q]^{-1} B_{N_3}^T R (W - A_{pN} Y_p - B_{pN} U_p - C_{pN} E_p) \quad (4.14)$$

令:

$$\begin{aligned} l_w &= [l_{w_1}, \dots, l_{w_n}] = M[B_{N_3}^T R B_{N_3} + Q]^{-1} B_{N_3}^T R \\ l_y &= [l_{y_1}, \dots, l_{y_n}] = -l_w A_{pN} \\ l_u &= [l_{u_1}, \dots, l_{u_n}] = -l_w B_{pN} \\ l_e &= [l_{e_1}, \dots, l_{e_n}] = -l_w C_{pN} \end{aligned} \quad (4.15)$$

则:

$$\Delta u(t) = [l_w, l_y, l_u, l_e][W^T, Y_p^T, U_p^T, E_p^T]^T \quad (4.16)$$

将来输出参考值未知时, 参考轨迹可设为 $y_i(t)$ 向着现时刻设定值 $r_i(t)$ 的一阶光滑近似, 则根据(2.24)式有:

$$W = T_y y(t) + T_r r(t) = \begin{bmatrix} T_y & T_r \end{bmatrix} \begin{bmatrix} y(t) \\ r(t) \end{bmatrix} = T \begin{bmatrix} y(t) \\ r(t) \end{bmatrix} \quad (4.17)$$

令:

$$\begin{aligned} [l_{w_y}, l_{w_r}] &= l_w T = l_w \begin{bmatrix} T_y & T_r \end{bmatrix} = [l_{w_y} T_y \quad l_{w_r} T_r] \\ l_{w_y} &= l_w T_y \\ l_{w_r} &= l_w T_r \end{aligned} \quad (4.18)$$

则控制律公式(4.16)变为:

$$\begin{aligned} \Delta u(t) &= [l_{w_y}, l_{w_r}, l_y, l_u, l_e][y^T(t), r^T(t), Y_p^T, U_p^T, E_p^T]^T \\ &= l_{w_y} y(t) + l_{w_r} r(t) + l_y Y_p + l_u U_p + l_e E_p \end{aligned} \quad (4.19)$$

令:

4.4 t 时刻的预测误差

噪声及未建模型动态引起的 t 时刻的输出预报误差:

$$\mathbf{e}(t) = \mathbf{y}(t) - \hat{\mathbf{y}}(t|t-1) \quad (4.21)$$

$\mathbf{e}(t) = [e_1(t), e_2(t), \dots, e_n(t)]^T$, $\mathbf{y}(t) = [y_1(t), y_2(t), \dots, y_n(t)]^T$ 是系统实际输出, $\hat{\mathbf{y}}(t|t-1) = [\hat{y}_1(t|t-1), \hat{y}_2(t|t-1), \dots, \hat{y}_n(t|t-1)]^T$ 为 $t-1$ 时刻对 t 时刻的模型预测输出。

4.5 算法实现

根据被控制对象特征, 确定合适的模型结构, 选择合适的预测时域、控制时域、加权因子、柔化因子、模型参数辨识初值、遗忘因子。每个控制周期执行以下计算:

1. 得到输入输出变量新的测量值, 进行模型参数辨识, 得到 θ ;
2. 计算预测误差 $\mathbf{e}(t) = \mathbf{y}(t) - \hat{\mathbf{y}}(t|t-1)$;
3. 计算矩阵 \mathbf{T}_y 和 \mathbf{T}_r ;
4. 根据模型参数计算 \mathbf{A}_{pN} 、 \mathbf{B}_{pN} 、 \mathbf{C}_{pN} 和 \mathbf{B}_{N_1} ;
5. 计算控制器系数矩阵 \mathbf{l}_{w_r} 、 \mathbf{l}_{w_y} 、 \mathbf{l}_u 和 \mathbf{l}_e ;
6. 计算控制增量 $\Delta \mathbf{u}(t) = \mathbf{l}_{w_r} \mathbf{r}(t) + \mathbf{l}_{w_y} \mathbf{Y}_p + \mathbf{l}_u \mathbf{U}_p + \mathbf{l}_e \mathbf{E}_p$;
7. 计算 $\mathbf{u}(t) = \mathbf{u}(t-1) + \Delta \mathbf{u}(t)$ 。

非自适应模式下, 由于控制器系数都可以提前计算出, 在线时只用执行第 5-7 步。

4.6 算法简化程度分析

在自适应模式下, 由于本章方法不再需要求解 Diophantine 方程, 其实现与计算显然比第二章与第三章的方法容易。

在非自适应模式下,本章方法相对第三章的方法,实现难度进一步下降。以第六章在多容液位系统中的应用为例,设计广义预测控制器使用的对象模型结构为:

$$\begin{bmatrix} 1+\tilde{a}_{11_1}z^{-1}+\tilde{a}_{11_2}z^{-2} & 0 \\ 0 & 1+\tilde{a}_{22_1}z^{-1}+\tilde{a}_{22_2}z^{-2} \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} b_{11_0} & b_{12_0}z^{-1} \\ b_{21_0}z^{-1} & b_{22_0} \end{bmatrix} \begin{bmatrix} \Delta u_1(t-1) \\ \Delta u_2(t-1) \end{bmatrix} + \begin{bmatrix} 1+c_{11_1}z^{-1} & 0 \\ 0 & 1+c_{22_1}z^{-1} \end{bmatrix} \begin{bmatrix} e_1(t) \\ e_2(t) \end{bmatrix} \quad (4.22)$$

根据上式,模型结构参数为: $na = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ 、 $nb = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$ 、 $nc = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ 、 $d = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ 。

控制器设计参数为:预测时域 $N_{1_1} = N_{1_2} = 73$, 控制时域 $N_{3_1} = N_{3_2} = 18$ 。则按照第三章的算法,控制器(3.19)式的系数矩阵的维数为:

$$m \times \left(n + \sum_{i=1}^n (N_i + na_{ii} + d_i) + \sum_{i=1}^n \sum_{j=1}^m (N_i + d_i + d_{ij} + nb_{ij} - 1) \right) \quad (4.23)$$

而按本章的方法,控制器(4.20)式的系数矩阵的维数为:

$$m \times \left(n + \sum_{i=1}^n (na_{ii} + 1) + \sum_{i=1}^n \sum_{j=1}^m (d_{ij} + nb_{ij}) + \sum_{i=1}^n (nc_{ii}) \right) \quad (4.24)$$

把相关参数代入(4.23)式与(4.24)式中,得到第三章控制器系数矩阵的维数为: 2×440 ; 本章控制器系数矩阵的维数为: 2×10 , 本章方法控制器系数只有第三章方法的 $1/44$ 。这种实现方式与计算量的简化是显然的。尽管第四章方法还需要计算(4.21)式一次,但其引入的计算量非常小。在本例中只需要 12 次加法与 10 次乘法运算。

4.7 小结

第三章与本章的内容表明,多变量广义预测控制算法无论是引入 Diophantine 方程求解,还是直接递推求解,最终都可以表示成控制器系数与参考轨迹及过程历史信息乘积的形式。求解 Diophantine 方程的方法复杂一些,而本章直接递推求解的方法简单直观,计算量少。

第五章 基于 BP 网络的多变量广义预测控制器系数快速计算方法

摘要：指出了多变量广义预测控制算法实际上是一种函数映射关系，将多变量系统模型参数构成的空间映射到多变量广义预测控制器系数构成的空间，并利用 BP 网络的空间映射能力实现多变量广义预测控制器系数的快速计算。通过运算量比较，举例说明了算法的快速性。

5.1 引言

在第一章绪论“广义预测控制简化实现方法”一节中提到的各种简化实现方法中，有一类是利用空间映射的方法。由第三章和第四章的内容知，广义预测控制器系数与其模型参数之间存在着直接的计算公式，说明了广义预测控制算法实质上是在设计参数所确定的广义预测控制方法下模型参数到控制器系数之间的一种映射。BP 网络从理论上具有任意的映射能力，而且具有最小的网络规模^[113]。因此本章在阐明广义预测控制内在规律的基础上，利用 BP 网络的空间映射能力来实现模型参数到控制器系数的映射，简化控制器系数的求解运算。本章方法是对作者在文献[104]和[105]中所示方法的完善与扩展。

5.2 广义预测控制算法控制器系数直接求解公式

对于(2.3)式的过程模型，引入(2.4)式的目标函数，根据第四章的推导过程，可以得到(4.20)式所示的控制律：

$$\Delta u(t) = l_w r(t) + l_{wy} Y_p + l_u U_p + l_e E_p \quad (5.1)$$

即：

$$\Delta u(t) = \begin{bmatrix} l_w & l_{wy} & l_u & l_e \end{bmatrix} \begin{bmatrix} r(t) \\ Y_p \\ U_p \\ E_p \end{bmatrix} \quad (5.2)$$

其中控制器系数矩阵 $\begin{bmatrix} \mathbf{l}_w & \mathbf{l}_{wy} & \mathbf{l}_u & \mathbf{l}_e \end{bmatrix}$ 只与模型参数及设计参数相关。

5.3 模型参数和广义预测控制器系数之间的关系

定义:

$$\mathbf{l}_{\text{MIMO GPC}} = \begin{bmatrix} \mathbf{l}_w & \mathbf{l}_{wy} & \mathbf{l}_u & \mathbf{l}_e \end{bmatrix} = \{l_{ij}\}_{m \times [n + \sum_{i=1}^n (na_{ii} + 1) + \sum_{i=1}^n \sum_{j=1}^m (d_{ij} + nb_{ij}) + \sum_{i=1}^n nc_{ii}]} \quad (5.3)$$

假定通过选择合适的 \mathbf{R} 与 \mathbf{Q} 总能保证多变量广义预测控制算法有解, 那么对于任意给定的一组对象模型参数:

$$\begin{aligned} & (a_{ii_k}, b_{ij_h}, d_{ij}, c_{ii_x}) \\ & i = 1, \dots, n \\ & j = 1, \dots, m \\ & k = 1, \dots, na_{ii} \\ & h = 0, \dots, nb_{ij} \\ & x = 1, \dots, nc_{ii} \end{aligned}$$

及广义预测控制算法设计参数 $\mathbf{R}, \mathbf{Q}, \mathbf{N}, \mathbf{N}_3$, 在第四章所示方法下, 总可以确定一组控制器系数 $\mathbf{l}_{\text{MIMO GPC}}$, 即:

$$\begin{aligned} \mathbf{l}_{\text{MIMO GPC}} &= f_{\text{MIMO GPC}}(a_{ii_k}, b_{ij_h}, d_{ij}, c_{ii_x}, \mathbf{R}, \mathbf{Q}, \mathbf{W}, \mathbf{N}, \mathbf{N}_3) \\ & i = 1, \dots, n \\ & j = 1, \dots, m \\ & k = 1, \dots, na_{ii} \\ & h = 0, \dots, nb_{ij} \\ & x = 1, \dots, nc_{ii} \end{aligned} \quad (5.4)$$

其中, $f_{\text{MIMO GPC}}$ 代表多变量广义预测控制算法。

一般设计中, 参考轨迹取设定值的一阶光滑近似, 柔化因子为

$\alpha = [\alpha_1, \alpha_2, \dots, \alpha_n]$ 。 $\mathbf{R} = \text{diag}\{\delta_1, \delta_2, \dots, \delta_n\}$ 与 $\mathbf{Q} = \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_m\}$ 中的

$\delta_i (i = 1, \dots, n)$ 和 $\lambda_j (j = 1, \dots, m)$ 经常是常值对角阵。因此公式(5.4)可以简化为:

$$\begin{aligned}
\mathbf{l}_{\text{MIMO GPC}} &= \mathbf{f}_{\text{MIMO GPC}}(a_{i_k}, b_{ij_h}, d_{ij}, c_{i_x}, \delta_i, \lambda_j, \alpha_i, N_i, N_{3_j}) \\
i &= 1, \dots, n \\
j &= 1, \dots, m \\
k &= 1, \dots, na_{ii} \\
h &= 0, \dots, nb_{ij} \\
x &= 1, \dots, nc_{ii}
\end{aligned} \tag{5.5}$$

当滞后 d_{ij} ($i = 1, \dots, n, j = 1, \dots, m$) 确定后, 模型参数可以任意取值, 每组取值构成一个点对:

$$\begin{aligned}
&(a_{i_k}, b_{ij_h}, c_{i_x}) \\
&i = 1, \dots, n \\
&j = 1, \dots, m \\
&k = 1, \dots, na_{ii} \\
&h = 0, \dots, nb_{ij} \\
&x = 1, \dots, nc_{ii}
\end{aligned}$$

所有点对的集合便构成一个 $\sum_{i=1}^n \left[na_{ii} + \left(\sum_{j=1}^m (nb_{ij} + 1) \right) + nc_{ii} \right]$ 维空间, 并可定义此空间为模型参数空间。对于纯滞后估计不准的部分, 离散时会包含在相应的脉冲传递函数的分子系数中^[114]。

相应地, 如果定义控制器系数点对集 $\{l_{ij}\}_{m \times [n + \sum_{i=1}^n (na_{ii} + 1) + \sum_{i=1}^n \sum_{j=1}^m (d_{ij} + nb_{ij}) + \sum_{i=1}^n nc_{ii}]}$ 的全体所组成的空间为控制器系数空间, 则多变量广义预测控制算法实际上定义了一种映射, 它的原象空间为模型参数空间, 象空间为控制器系数空间, 映射关系由 $\mathbf{W}, \mathbf{R}, \mathbf{Q}, \mathbf{N}, \mathbf{N}_1, \mathbf{N}_2, \mathbf{N}_3$ 确定。即:

$$f_{\text{MIMO GPC}} : R^{\sum_{i=1}^n \left[na_{ii} + \left(\sum_{j=1}^m (nb_{ij} + 1) \right) + nc_{ii} \right]} \xrightarrow[\mathbf{W}, \mathbf{R}, \mathbf{Q}, \mathbf{N}, \mathbf{N}_1, \mathbf{N}_2, \mathbf{N}_3]{\text{MIMO GPC}} R^{m \times [n + \sum_{i=1}^n (na_{ii} + 1) + \sum_{i=1}^n \sum_{j=1}^m (d_{ij} + nb_{ij}) + \sum_{i=1}^n nc_{ii}]}$$

如图 5.1 所示。

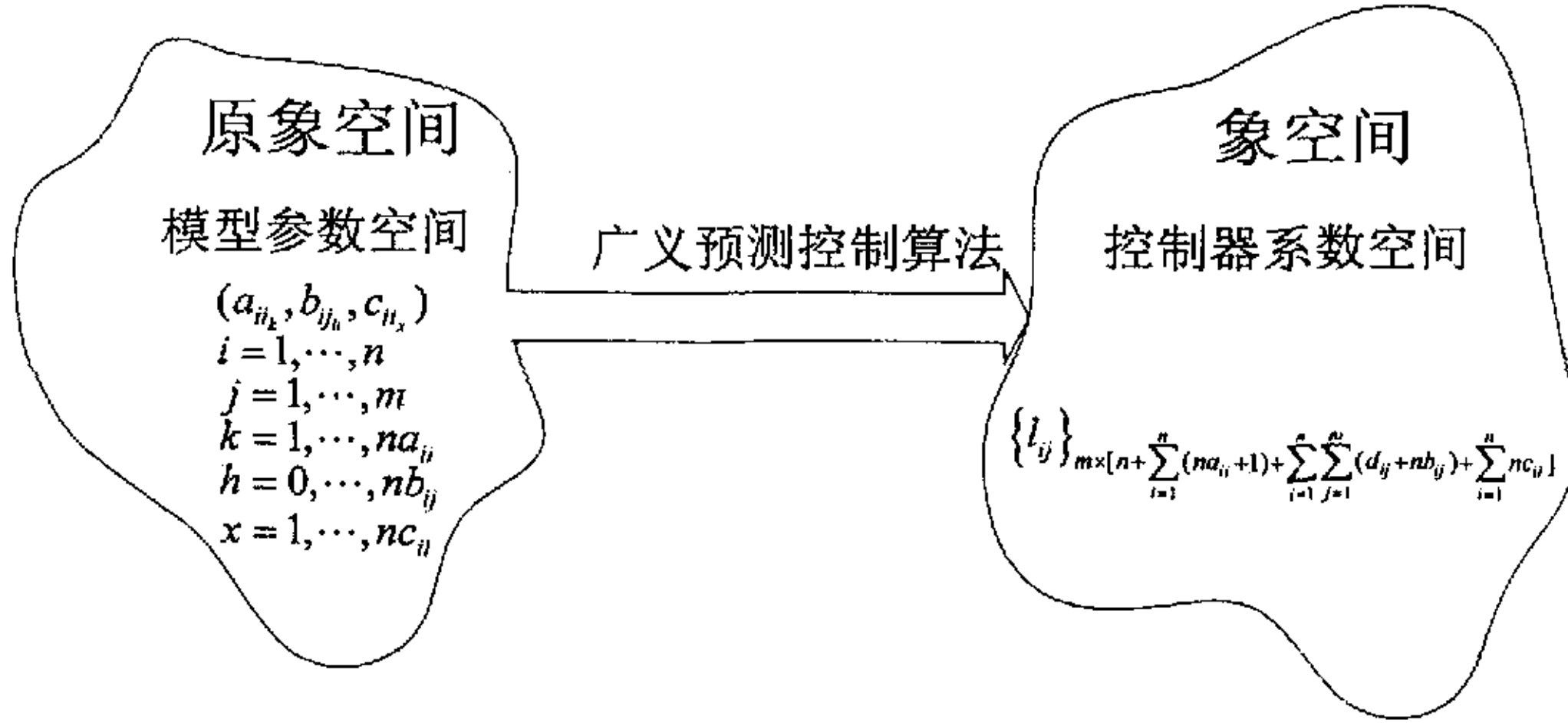


图 5.1 多变量广义预测控制算法映射示意图

对于很大一类线性或非线性、时变或非时变对象，只要选择合适的 $na_{ii}, nb_{ij}, nc_{ii}, d_{ij}$, ($i=1, \dots, n, j=1, \dots, m$)，在特定的时空坐标上总是可以近似地用 CARIMA 模型(2.3)式来描述。而且一个特定对象，在特定的时空中，当 d_{ij} ($i=1, \dots, n, j=1, \dots, m$) 确定后，这种近似的点对 (a_{ik}, b_{jh}, c_{ix}) 只会集中在这个

$\sum_{i=1}^n \left[na_{ii} + \left(\sum_{j=1}^m (nb_{ij} + 1) \right) + nc_{ii} \right]$ 维空间的某一个或者几个区域，即：

$$\begin{aligned} \underline{a}_{ik} \leq a_{ik} \leq \bar{a}_{ik} & \quad i=1, \dots, n \\ \underline{b}_{jh} \leq b_{jh} \leq \bar{b}_{jh} & \quad j=1, \dots, m \\ \underline{c}_{ix} \leq c_{ix} \leq \bar{c}_{ix} & \quad k=1, \dots, na_{ii} \\ & \quad h=0, \dots, nb_{ij} \\ & \quad x=1, \dots, nc_{ii} \end{aligned}$$

如图 5.2 所示。

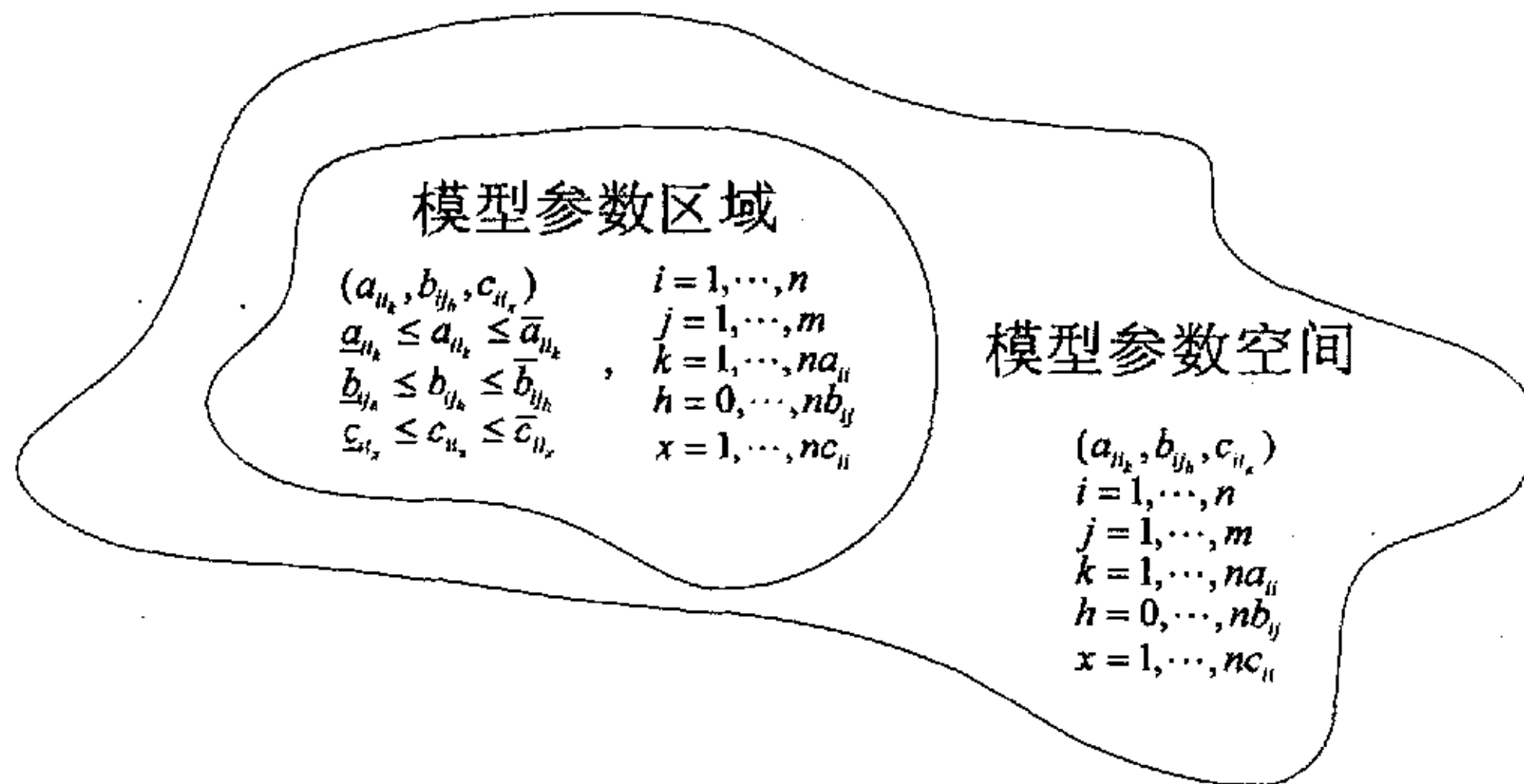


图 5.2 模型参数区域示意图

5.4 模型参数到广义预测控制器系数的映射

以上说明了多变量广义预测控制实际上是从模型参数空间到控制器系数空间的一种映射，可以利用神经网络来映射这种关系。神经网络的输入层节点为：

$$a_{i_k}, b_{ij_h}, d_{ij}, c_{i_x}, \delta_i, \lambda_j, \alpha_i, N_i, N_{3_j}$$

$$(i = 1, \dots, n, j = 1, \dots, m, k = 1, \dots, na_{ii}, h = 0, \dots, nb_{ij}, x = 1, \dots, nc_{ii})$$

输出层节点为：

$$l_{i,j}, (i = 1, \dots, m, j = 1, \dots, \left[n + \sum_{i=1}^n (na_{ii} + 1) + \sum_{i=1}^n \sum_{j=1}^m (d_{ij} + nb_{ij}) + \sum_{i=1}^n nc_{ii} \right])$$

在实际设计中，通常把 $\delta_i, \lambda_j, \alpha_i, N_i, N_{3_j}$ 作为设计参数。为减少神经网络的规模，只需把对象模型参数 $(a_{i_k}, b_{ij_h}, c_{i_x})$ 作为网络的输入层节点。此时神经网络代表的映射为：

$$\begin{aligned} l_{\text{MIMOGPC}} &= f_{\text{MIMOGPC}}(a_{i_k}, b_{ij_h}, c_{i_x}) \\ i &= 1, \dots, n \\ j &= 1, \dots, m \\ k &= 1, \dots, na_{ii} \\ h &= 0, \dots, nb_{ij} \\ x &= 1, \dots, nc_{ii} \end{aligned} \quad (5.6)$$

神经网络的结构如图 5.3 所示。

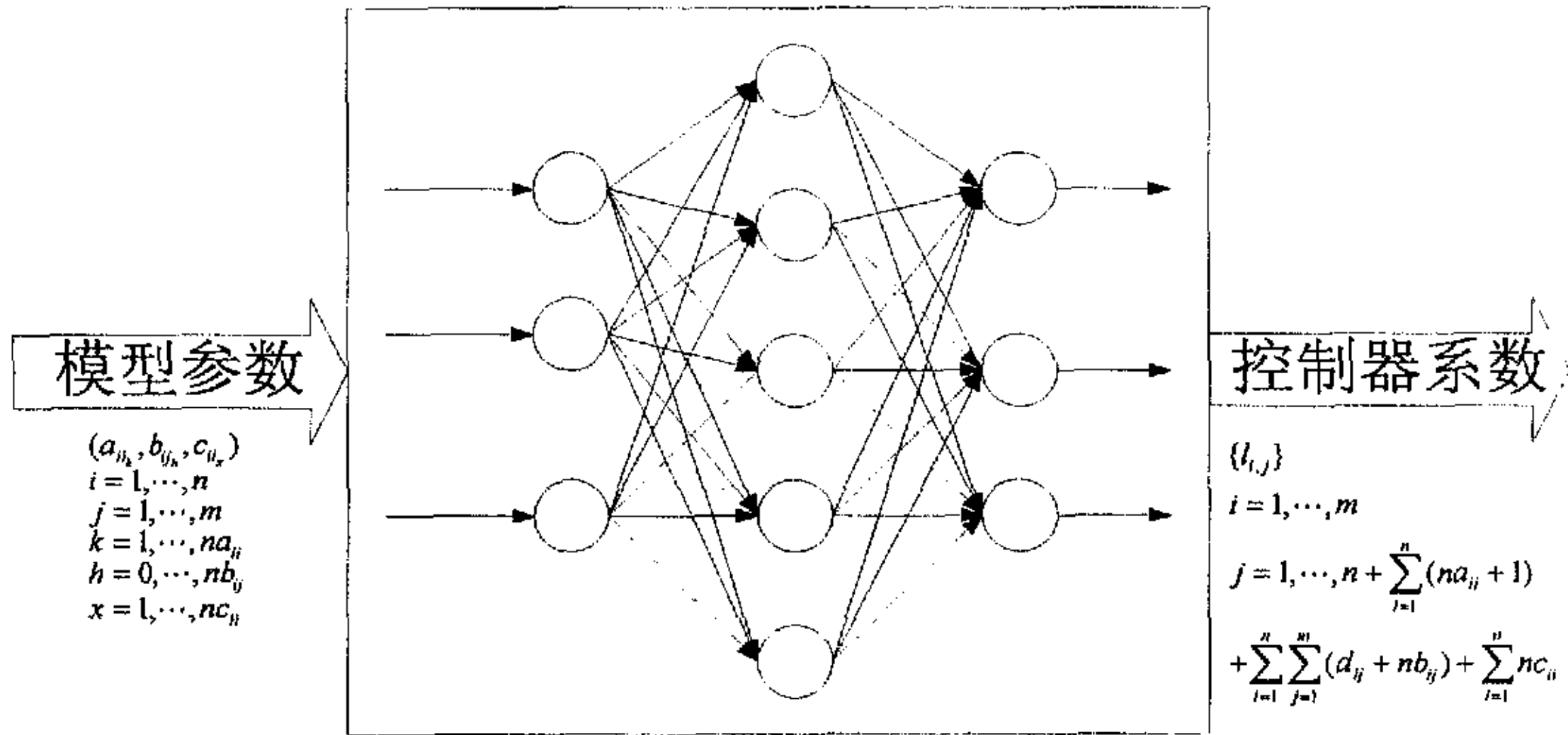


图 5.3 BP 网络结构示意图

5.5 控制系统结构

所得到的控制系统结构如图 5.4 所示。

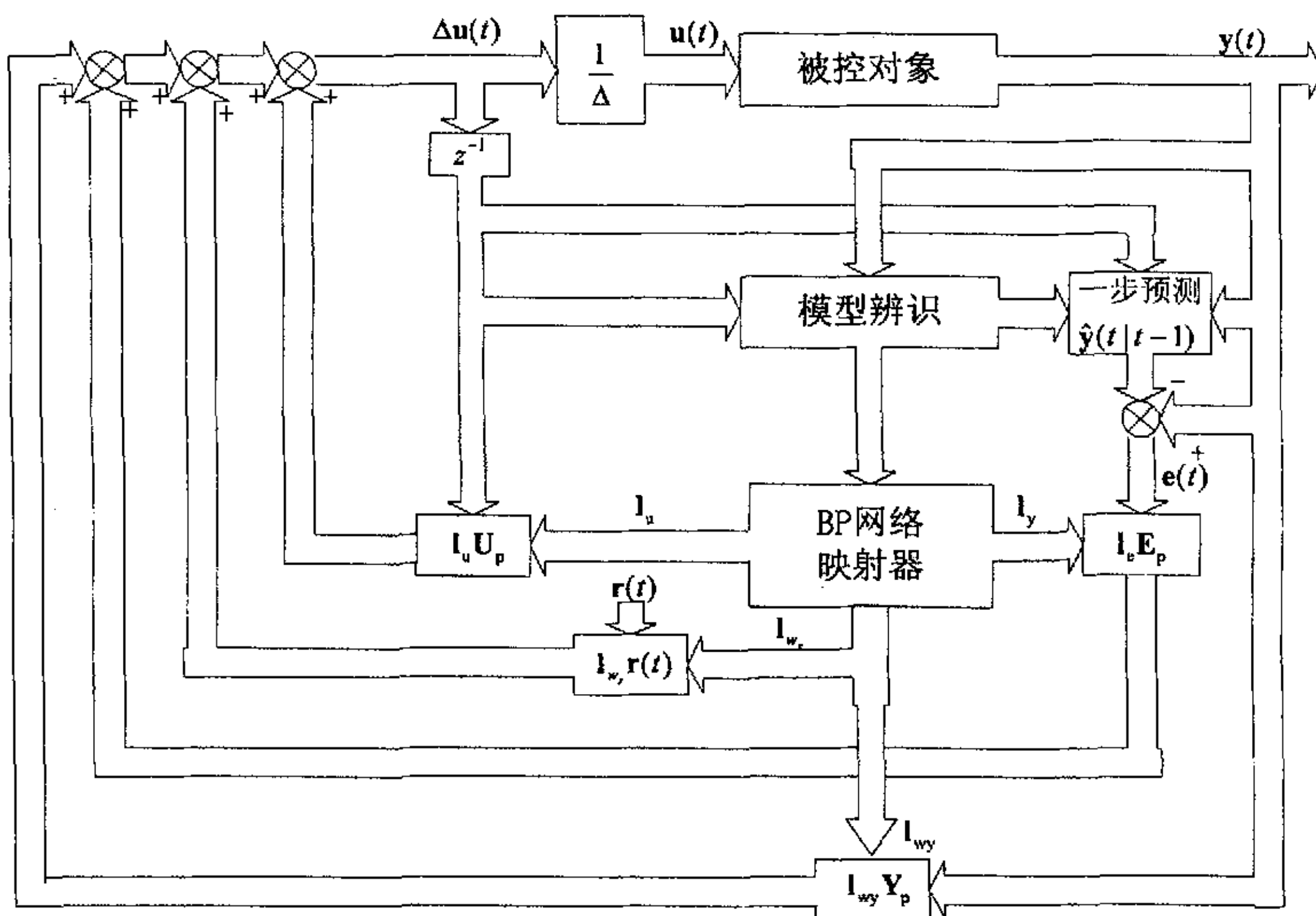


图 5.4 基于 BP 网络的多变量自适应广义预测控制快速算法系统框图

图 5.4 与图 4.1 的结构完全相同，只是图 4.1 中控制系数计算部分现在已由图 5.4 中的 BP 网络来完成。BP 网络是离线训练好的，投入控制时，直接从模型参数映射得到控制器系数，从而减少了在线计算量。

5.6 算法实现

算法实现分为离线与在线两部分。

离线部分为：在控制器投入在线运行前，选择可以满足需要的模型结构，并指定模型参数的变化区间，组合成一定数量的输入样本点对；根据控制目标需要确定多变量广义预测控制算法设计参数，按公式(4.15)到公式(4.20)求出相应的控制器系数，用来作为训练神经网络的样本。输入样本的选择，应以覆盖所有可能出现的模型参数为宜，以保证控制器的鲁棒性。然后选定神经网络的结构，训练

神经网络。可以把设计参数作为网络的输入层节点,也可以根据不同设计参数的组合训练不同的神经网络,以满足各种控制要求。

自适应模式下在线计算按以下步骤:

1. 采样当前时刻的输入输出,用合适的参数辨识方法得到当前时刻被控制对象的模型参数;

2. 计算预测误差 $e(t) = y(t) - \hat{y}(t|t-1)$;

3. 运算一次 BP 网络,获得控制器系数 l_{w_r} 、 l_{w_y} 、 l_u 、 l_e ;

4. 计算控制增量 $\Delta u(t) = l_{w_r} r(t) + l_{w_y} Y_p + l_u U_p + l_e E_p$;

5. 计算 $u(t) = u(t-1) + \Delta u(t)$ 。

非自适应模式下,不需要执行第 1 步。

5.7 算法快速性分析

为了对基于 BP 网络的多变量自适应广义预测控制快速算法(简记为 FMGPC)求解控制器系数的快速性有一个直观的了解,以第六章广义预测控制算法在多容液位系统中的应用为例,把本章提出的方法与第四章提出的多变量广义预测控制器系数直接求解方法(简记为 DMGPC)在计算量上作一比较。比较两种方法从模型参数计算控制系数所需要的加法与乘法的次数,其它的运算不作统计,统计计算量约定以下规则:

1. 矩阵加法, $q \times p$ 维的矩阵与 $q \times p$ 维的矩阵相加需要加法 $q \times p$ 次。
2. 矩阵乘法, $q \times p$ 维的矩阵与 $p \times h$ 维的矩阵相乘需要加法 $q \times h \times (p-1)$ 次,乘法 $q \times h \times p$ 次。
3. 矩阵求逆,根据全选主元 Gauss-Jordan 消去法^[115],一个 n 维矩阵不计赋值运算量,至少需要加法 $4n^3/3 - n^2/2 + n/6$,乘法 $n^3 - n$ 次。
4. 正切函数,按照 Sun Microsystem, Inc.(<http://www.netlib.org>)提供的算法,一个正切函数值计算最多由 31 次加法和 24 次乘法完成。

按以上规定,一个 $i-h-o$ 的 BP 网络,如果隐层为正切函数,则从网络输入直接计算一次网络输出的计算量为:加法 $h(i+31+o)$ 次,乘法 $h(i+24+o)$ 次。

按第四章的计算方法,计算控制器系数的过程中,需要先计算一些中间结果,

最后才能得到控制器系数。表 5.1 列出了各中间结果及相应控制器系数计算需要的加法运算与乘法运算的次数，每一项中不包含中间结果的计算量。

表 5.1 DMGPC 各项需要的加法与乘法次数表

项	加法运算次数	乘法运算次数
T_y	$\sum_{i=1}^n N_i$	$\sum_{i=1}^n \{N_i + d_i\}$
T_r	$\sum_{i=1}^n N_i$	$\sum_{i=1}^n \{N_i + d_i\}$
A_{pN}	$\sum_{i=1}^n \left\{ (1 + na_{ii}) \left[1 + (d_i + N_i - 1)na_{ii} - \frac{na_{ii}^2}{2} \right] \right\}$	$\sum_{i=1}^n \left\{ (na_{ii} + 1)^2 \left[-1 - \frac{na_{ii}}{2} + (d_i + N_i) \right] \right\}$
B_{pN}	$\sum_{i=1}^n \sum_{j=1}^m \left\{ (d_{ij} + nb_{ij}) \left[\frac{1 + nb_{ij} + 2d_{ij} - d_{ij}^2}{2} + (d_i + N_i - \frac{3}{2})na_{ii} - \frac{na_{ii}^2}{2} \right] \right\}$	$\sum_{i=1}^n \sum_{j=1}^m \left\{ [(d_{ij} + nb_{ij})(na_{ii} + 1)] \left[-1 - \frac{na_{ii}}{2} + (d_i + N_i) \right] \right\}$
C_{pN}	$\sum_{i=1}^n \left\{ \frac{nc_{ii}(nc_{ii} + 1)}{2} + \left[(d_i + N_i) - \frac{na_{ii}}{2} - \frac{3}{2} \right] nc_{ii} na_{ii} \right\}$	$\sum_{i=1}^n \left\{ [(na_{ii} + 1)nc_{ii}] \left[-1 - \frac{na_{ii}}{2} + (d_i + N_i) \right] \right\}$
B_{N_3}	$\sum_{i=1}^n \sum_{j=1}^m \left\{ nb_{ij} + 1 + (N_i + d_i - d_{ij} - \frac{3}{2})na_{ii} - \frac{na_{ii}^2}{2} \right\}$	$\sum_{i=1}^n \sum_{j=1}^m \left\{ \left[-1 - \frac{na_{ii}}{2} + (N_i + d_i - d_{ij}) \right] (na_{ii} + 1) \right\}$
I_w	$\frac{4}{3} \left[\sum_{j=1}^m \{N_{3j} + 1\} \right]^3 + \left[\sum_{j=1}^m \{N_{3j} + 1\} \right]^2 \left[2 \left[\sum_{i=1}^n \{N_i\} \right] - \frac{1}{2} \right] + \left\{ \frac{1}{6} - \left[\sum_{i=1}^n \{N_i\} \right] + m \left[\sum_{i=1}^n \{N_i\} \right] \right\} \sum_{j=1}^m \{N_{3j} + 1\} - m \left[\sum_{i=1}^n \{N_i\} \right]$	$\left[\sum_{j=1}^m \{N_{3j} + 1\} \right]^3 + 2 \left[\sum_{i=1}^n \{N_i\} \right] \left[\sum_{j=1}^m \{N_{3j} + 1\} \right]^2 + \left\{ 2 \left[\sum_{i=1}^n \{N_i\} \right] + m \left[\sum_{i=1}^n \{N_i\} \right] - 1 \right\} \left[\sum_{j=1}^m \{N_{3j} + 1\} \right]$

(此表续于下页)

(续前页表 5.1)

项	加法运算次数	乘法运算次数
I_y	$m \left(\sum_{i=1}^n \{na_{ii} + 1\} \right) \left(\sum_{i=1}^n \{N_i\} - 1 \right)$	$m \left(\sum_{i=1}^n \{na_{ii} + 1\} \right) \left(\sum_{i=1}^n \{N_i\} \right)$
I_u	$m \left(\sum_{i=1}^n \sum_{j=1}^m \{d_{ij} + nb_{ij}\} \right) \left(\sum_{i=1}^n \{N_i\} - 1 \right)$	$m \left(\sum_{i=1}^n \sum_{j=1}^m \{d_{ij} + nb_{ij}\} \right) \left(\sum_{i=1}^n \{N_i\} \right)$
I_e	$m \left(\sum_{i=1}^n \{nc_{ii}\} \right) \left(\sum_{i=1}^n \{N_i\} - 1 \right)$	$m \left(\sum_{i=1}^n \{nc_{ii}\} \right) \left(\sum_{i=1}^n \{N_i\} \right)$
I_{w_y}	$mn \left(\sum_{i=1}^n \{N_i\} - 1 \right)$	$mn \left(\sum_{i=1}^n \{N_i\} \right)$
I_{w_r}	$mn \left(\sum_{i=1}^n \{N_i\} - 1 \right)$	$mn \left(\sum_{i=1}^n \{N_i\} \right)$
I_{wy}	$m \left(\sum_{i=1}^n \{na_{ii} + 1\} \right) (n-1)$ $+ m \left(\sum_{i=1}^n \{na_{ii} + 1\} \right)$	$m \left(\sum_{i=1}^n \{na_{ii} + 1\} \right) n$

表 5.1 中 T_y 、 T_r 的计算量统计参照公式(2.22)和公式(2.24)， A_{pN} 、 B_{pN} 、 C_{pN} 参照公式(4.5)、公式(4.8)及公式(4.10)， B_{N_i} 参照公式(4.6)、公式(4.9)及公式(4.10)， I_w 、 I_y 、 I_u 、 I_e 参照公式(4.15)， I_{w_y} 、 I_{w_r} 参照公式(4.18)， I_{wy} 参照公式(4.20)。

在第六章多容液位对象控制中，设计广义预测控制器使用的对象模型结构为：

$$\begin{bmatrix} 1+\tilde{a}_{11}z^{-1}+\tilde{a}_{11_2}z^{-2} & 0 \\ 0 & 1+\tilde{a}_{22}z^{-1}+\tilde{a}_{22_2}z^{-2} \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} b_{11_0} & b_{12_0}z^{-1} \\ b_{21_0}z^{-1} & b_{22_0} \end{bmatrix} \begin{bmatrix} \Delta u_1(t-1) \\ \Delta u_2(t-1) \end{bmatrix} + \begin{bmatrix} 1+c_{11}z^{-1} & 0 \\ 0 & 1+c_{22}z^{-1} \end{bmatrix} \begin{bmatrix} e_1(t) \\ e_2(t) \end{bmatrix} \quad (5.7)$$

根据上式，模型结构参数为： $na = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ 、 $nb = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$ 、 $nc = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ 、 $d = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ 。

控制器设计参数为：预测时域 $N_{1_1} = N_{1_2} = 73$ ，控制时域 $N_{3_1} = N_{3_2} = 18$ 。选用 20 个结构为 8-10-1 的子网来实现控制器系统的快速计算，隐层节点使用正切函数。

把以上参数代入相应的公式，得到各个算法总的加法次数与乘法次数如表 5.2 所示。表中数据说明，用 BP 网络映射器求解控制器系数，其加法次数只有用公式计算的 1/63，而乘法次数只有原公式计算方法的 1/76。

以上事实说明，如果按照计算量来比较计算速度，则基于神经网络的方法具有明显的优势。

表 5.2 算法计算量比

算法	加法	乘法
DMGPC	504003	504206
FMGPC	8000	6600
DMGPC/FMGPC	63.004	76.3948

5.8 小结

从空间映射的角度解释了多变量广义预测控制算法的内在机理，为利用智能方法来实现广义预测控制算法找到了理论依据。BP 网络的引入，消除了多变量广义预测控制算法模型预测、反馈校正、滚动优化所需要的在线运算，降低了其设计与实现难度。这种快速算法，内存占用少，计算过程大大减化，对拓展自适应多变量广义预测控制的应用范围有一定的促进作用。

第六章 简化广义预测控制算法实验研究

摘要：将第二章至第五章所给出的简化算法应用于一个由 DCS 控制的非线性液位系统中，进行了实验与对比研究。对比了各个算法在自适应模式与非自适应模式下，跟踪设定值的能力，抗扰动的能力及鲁棒性能。实验结果表明，所有算法均是可行的，而且具有一致的控制品质。

6.1 引言

现实世界的被控制对象一般是非线性时变的，绝对线性的对象很少见。第二章到第五章的方法基于线性化思想，要求被控制对象在一定的时空范围内可线性化。为了检验第二章到第五章提出的方法的可行性，对比各章算法的控制效果，在一个 DCS 控制的非线性多容液位系统上对它们进行了对比实验研究。实验分为两大部分，第一部分是控制一个非线性单容液位系统，第二部分是控制一个多容液位系统。

为了在实验中区分第二章到第五章的算法，分别对它们进行了编号，如表 6.1 所示。

表 6.1 算法编号

序号	编号	算法对应章
1	MGPC-1	第二章
2	MGPC-2	第三章
3	MGPC-3	第四章
4	MGPC-4	第五章

6.2 多容液位装置简介

多容液位装置如图 6.1 所示，上半部分是六个水槽分两层排列，标号分别为：LT101、LT102、LT103、LT201、LT202、LT203；下半部分正中间是一个贮水箱，其左右两侧对称安装着变频泵 1 和泵 2。液位测量使用压力变送器，安装在各水槽的底部，如图 6.3 所示，压力测量信号为 4-20mA 的标准电流信号，经量纲变

换成 0-1 之间的无量纲数值，实际控制过程中被控变量的可变化范围在 0.16 到 0.36 之间。泵 1 和泵 2 各由一个变频器控制，变频器的可变频率范围是 0-50Hz，对应控制信号 0-100 之间的数值。泵 1 可以向 LT101、LT102、LT103 和 LT203 中供水；泵 2 可以向 LT201、LT202、LT203 和 LT103 中供水。

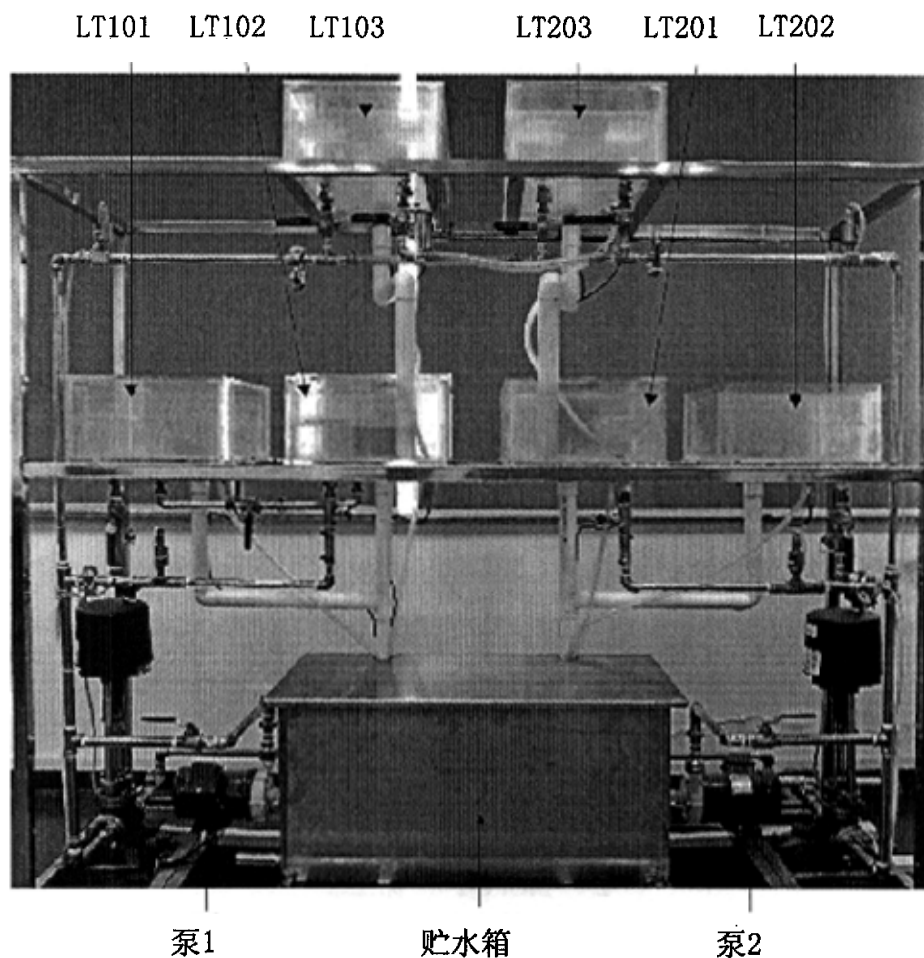


图 6.1 多容液位装置

控制器为浙大中控技术有限公司的 JX-300X 集散控制系统^{[116]-[120]}。

6.3 非线性单容液位控制系统实验研究

6.3.1 非线性单容液位系统简介

构造如图 6.2 所示的非线性单容液位系统，控制对象为 LT203 的液位，执行机构是泵 1。泵 2 为扰动泵，同时协助泵 1 改变工作点的位置。

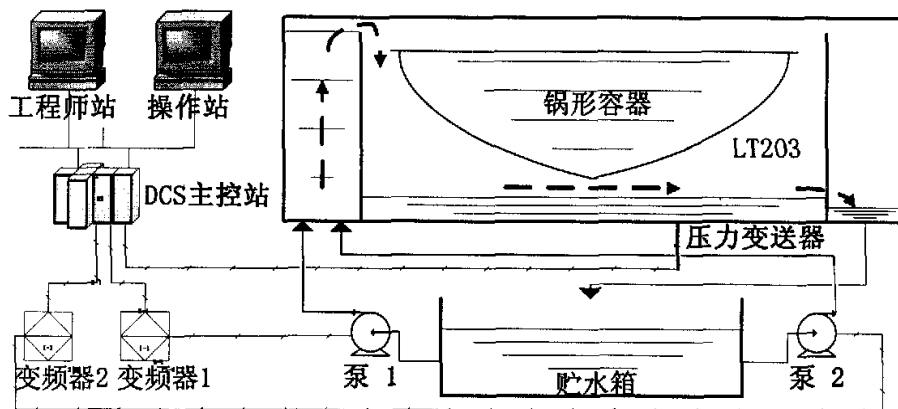


图 6.2 非线性单容液位系统工艺流程图

水槽分隔为三部分，如图 6.3 所示，泵 1 和泵 2 的水从左边水槽底部注入，溢流进入中间方形容器，然后向右从带纵向开口的隔板注入右边排水槽，再通过 LT201 流回到底部的贮水箱。为了强化液位变化的非线性，中间容器里面加入一个已盛满液体的有沿抛物面状锅形容器。

6.3.2 非线性单容液位系统的模型

为了得到实验对象的模型参数，作为整定控制器参数和选择模型辨识器参数初值的依据，对实验对象进行了一系列的测试，分别建立了传递函数模型与 CARIMA 模型。测试时，图 6.3 所示水槽中没有加锅形容器，泵 1 的平衡工作点选为 70，扰动泵 2 的控制信号保持 70 不变，数据采集周期为 1 秒。测试信号为伪随机二进制序列，共进行了十组测试。

采用一阶惯性加纯滞后的传递函数模型结构，从实验数据分析得到的平均传递函数模型为：

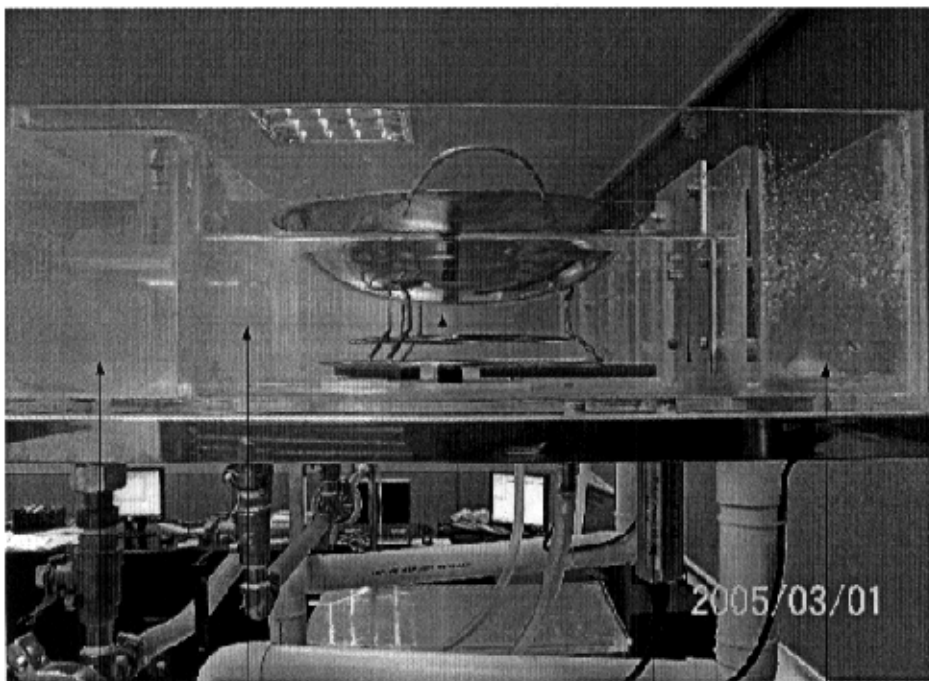
$$G(s) = \frac{0.0016}{1 + 29.1937s} e^{-1.9097s} \quad (6.1)$$

建立 CARIMA 离散模型时, 被控制对象的模型结构取为:

$$(1 + \tilde{a}_1 z^{-1} + \tilde{a}_2 z^{-2})y(t) = (b_0 + b_1 z^{-1})u(t) + (1 + c_1 z^{-1})e(t) \quad (6.2)$$

其中 $y(t)$ 是 LT203 的液位, $u(t)$ 是泵 1 的控制频率, $e(t)$ 是噪声。采样周期为 5 秒, 实验数据分析得到的平均 CARIMA 模型为:

$$(1 - 1.8458z^{-1} + 0.84581z^{-2})y(t) = (0.00020449 + 4.5644 * 10^{-5} z^{-1})u(t) + (1 - 1.01z^{-1})e(t) \quad (6.3)$$



入水槽 LT203 锅形容器 压力变送器 排水槽

图 6.3 非线性水槽

6.3.3 控制器参数选择

依据以上分析, 按照控制周期与预测控制算法参数选择的一般原则^{[11][12]}, 控制周期为 5 秒, 预测时域 $N = 30$, 控制时域 $N_3 = 5$ 。实验中参考轨迹取设定值的一阶光滑近似, 柔化因子 $\alpha = 0.5$ 。 \mathbf{R} 与 \mathbf{Q} 为对角阵, 对角元素分别为 $r = 1000$,

$q = 0.01$ 。

辨识方法采用带遗忘因子的最小二乘法, 遗忘因子为 0.98, 协方差矩阵初值 $P_0 = 1 \times 10^4 I$ (I 为单位矩阵)。辨识参数为 $\tilde{a}_1, \tilde{a}_2, b_0, b_1, c_1$ 。模型参数初值取一组近似值: $a_1 = -0.846$, $b_0 = 3 \times 10^{-4}$, $b_1 = 1.5 \times 10^{-6}$, $c_1 = 0$ 。

6.3.4 BP 网络结构、样本选择及训练

BP 网络用来实现基于 BP 网络的广义预测控制快速算法(MGPC-4)。

根据对象的模型结构(6.2)式及第四章的控制增量(4.19)式, 非线性单容液位系统广义预测控制器可以选择(6.4)式的结构:

$$\Delta u(t) = l_{w_{r1}} y(t) + l_{w_{r2}} r(t) + l_{y1} y(t) + l_{y2} y(t+1) + l_{u1} \Delta u(t) + l_{e1} e(t) \quad (6.4)$$

其中 $r(t)$ 为设定值, $l_{w_{r1}}, l_{w_{r2}}, l_{y1}, l_{y2}, l_{u1}, l_{e1}$ 为控制器系数。

由于 $\tilde{a}_1 = a_1 - 1$ 与 $\tilde{a}_2 = -a_1$, 都可由 a_1 决定, 因此对应第五章算法的 BP 网络的输入层节点取为 a_1 、 b_0 、 b_1 和 c_1 。输出层结点为: $l_{w_{r1}}, l_{w_{r2}}, l_{y1}, l_{y2}, l_{u1}, l_{e1}$ 。

根据实验结果分析, 选用表 6.2 所示的模型参数样本取值区间, 可以基本上覆盖控制过程中模型参数的变化范围。

表 6.2 模型参数样本取值区间

参数	最小值	最大值
a_1	-0.865	-0.825
b_0	3×10^{-4}	4.8×10^{-4}
b_1	-1.5×10^{-4}	0.8×10^{-4}
c_1	-0.5	0.5

神经网络的结构为: 4-9-6。

输入样本的获取方式为, 从表 6.2 中所示范围内将每一个输入参数等距分为五个点, 然后进行全组合, 共获得 625 组输入样本。输出样本是控制器系数, 按第四章(4.15)到(4.20)的公式计算。神经网络的训练目标为 0.00001, 训练 522 步达到目标。训练过程如图 6.4。

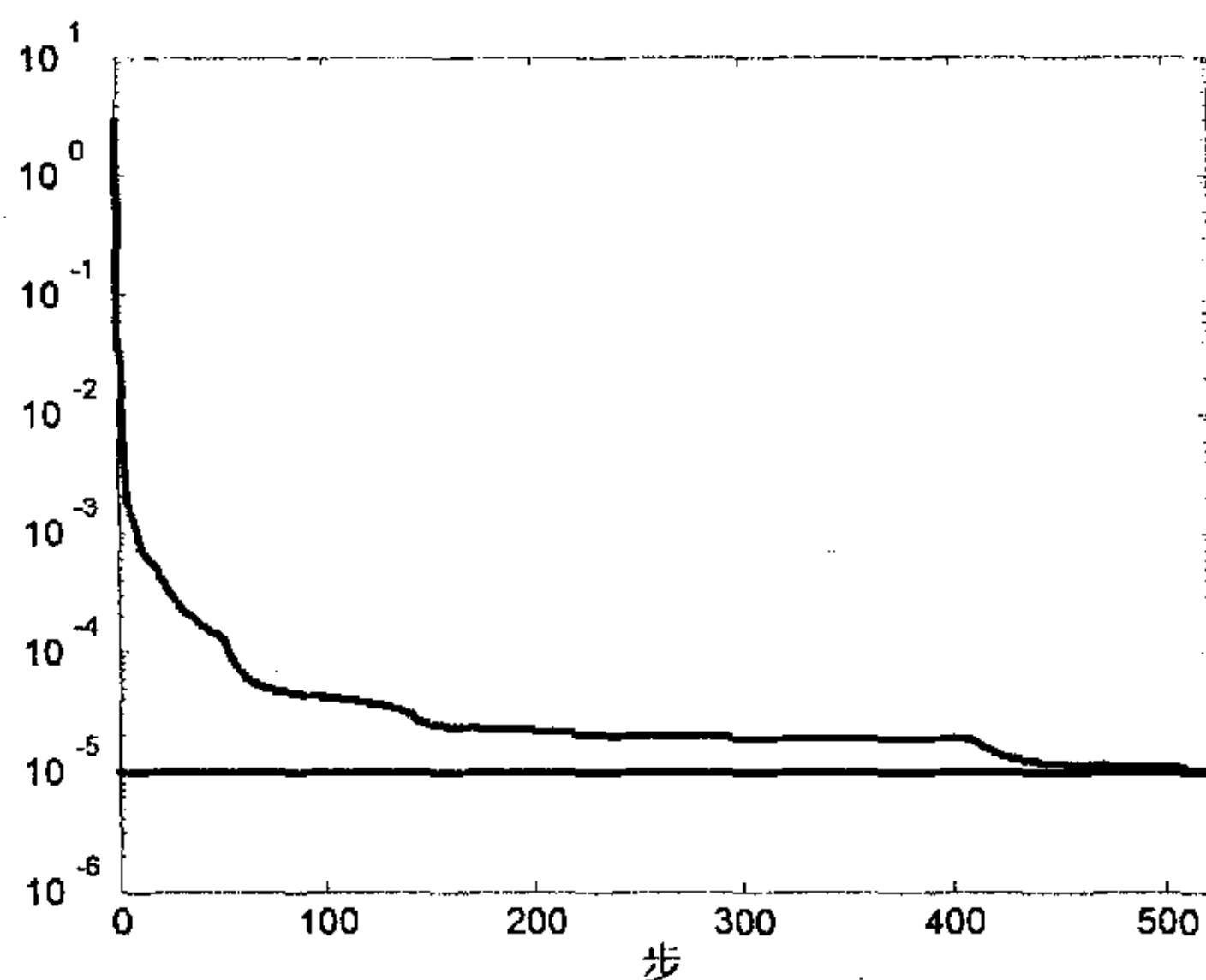


图 6.4 神经网络训练图

6.3.5 非线性单容液位系统控制实验

6.3.5.1 自适应模式

每一个算法进行控制时，时间分为 180 秒一段，1-360 秒时间段内泵 1 输入伪随机二进制控制信号进行参数辨识；361-540 秒时间段内让系统进入稳态；后面的 541-2880 秒时间段内系统投入自动控制。MGPC-1、MGPC-3 和 MGPC-4 设定值与扰动泵 2 在各个时间段内的变化情况如表 6.3，MGPC-2 设定值与扰动泵 2 在各个时间段内的变化情况如表 6.4。

扰动泵 2 的变化情况在四个控制算法实验中都是一样的。设定值的变化基本一样，只是表 6.3 中的 0.23 在表 6.4 中变成了 0.24，这是考虑到 MGPC-2 与 MGPC-1、MGPC-3、MGPC-4 完成的时间不同，测量装置对于同一高度测量结果出现稍微不同而进行调整的。

实验中，为了不使泵发生倒灌，泵的控制信号的最低值限制在 45。当计算得到的控制信号的幅值小于 45 或大于 100 时，采用“cut-off”方法，直接取其最近的边界值。

实验结果如图 6.5 到图 6.17 所示。图 6.5 到图 6.8 中实线为系统输出，点线为输出设定值，虚线、点划线分别为锅形容器的上下边界高度值。图 6.5 到图 6.8 中输出曲线上的尖脉冲是系统受到泵 2 扰动时的响应。图 6.9 到图 6.12 中的实线为泵 1 的控制信号，虚线为扰动泵 2 的控制信号。图 6.13 到图 6.17 中实线、虚线、点线和点划线分别为 MGPC-1、MGPC-2、MGPC-3 和 MGPC-4 的模型参数

在线辨识值，下三角形线和上三角形线分别为 MGPC-4 中相应模型参数在训练神经网络时输入样本点取值的上下界。

表 6.3 设定值与扰动泵 2 变化情况(一)

时间段(秒)	设定值	扰动泵 2
1-540	0	70
541-720	0.27	70
721-900	0.23	70
901-1080	0.23	45
1081-1260	0.19	45
1261-1440	0.23	45
1441-1620	0.23	70
1621-1800	0.27	70
1801-1980	0.31	70
1981-2160	0.31	95
2161-2340	0.35	95
2341-2520	0.31	95
2521-2700	0.31	70
2701-2880	0.27	70

表 6.4 设定值与扰动泵 2 变化情况(二)

时间段(秒)	设定值	扰动泵 2
1-540	0	70
541-720	0.27	70
721-900	0.24	70
901-1080	0.24	45
1081-1260	0.19	45
1261-1440	0.24	45
1441-1620	0.24	70
1621-1800	0.27	70
1801-1980	0.31	70
1981-2160	0.31	95
2161-2340	0.35	95
2341-2520	0.31	95
2521-2700	0.31	70
2701-2880	0.27	70

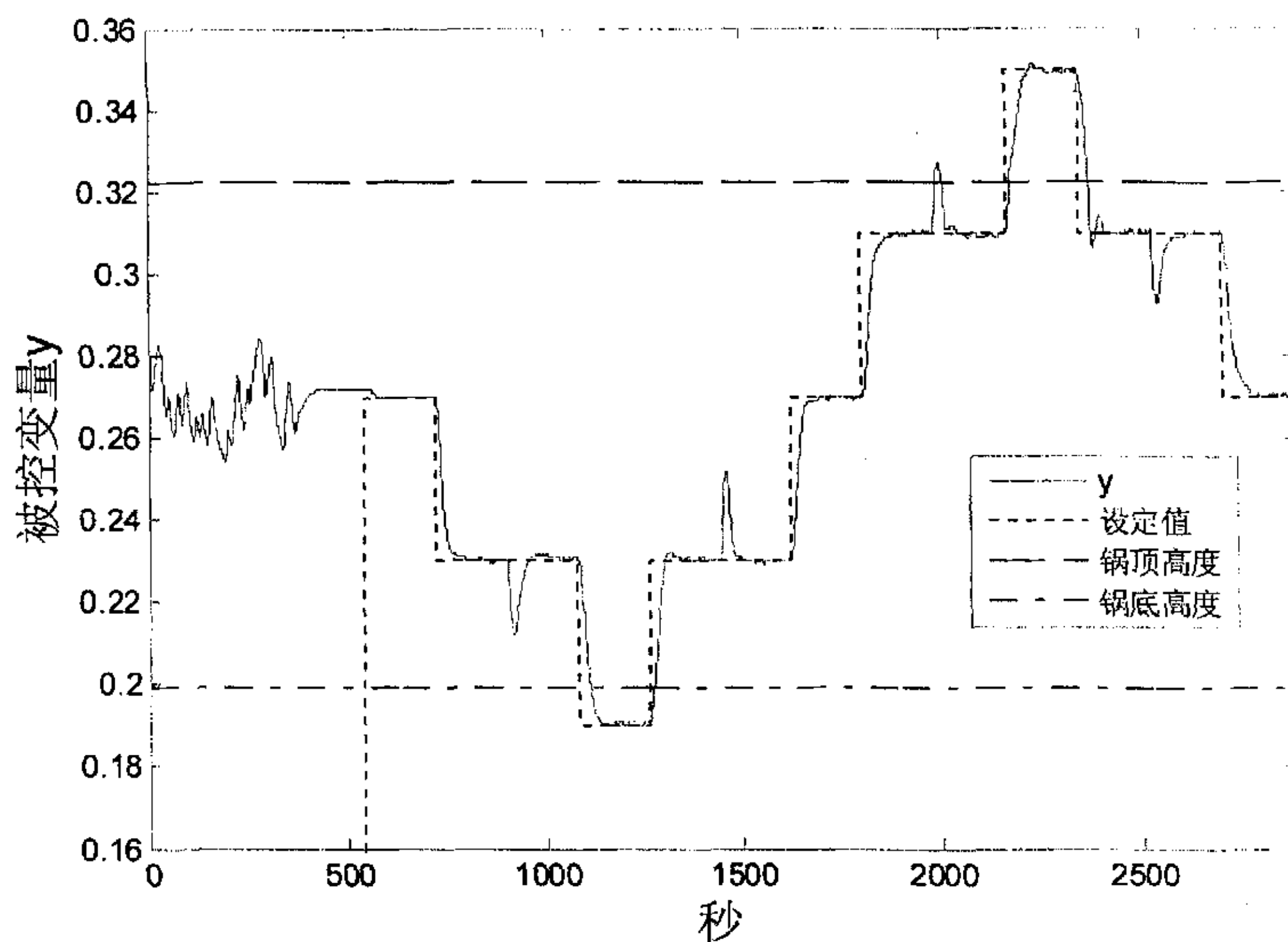


图 6.5 MGPC-1 被控变量 y

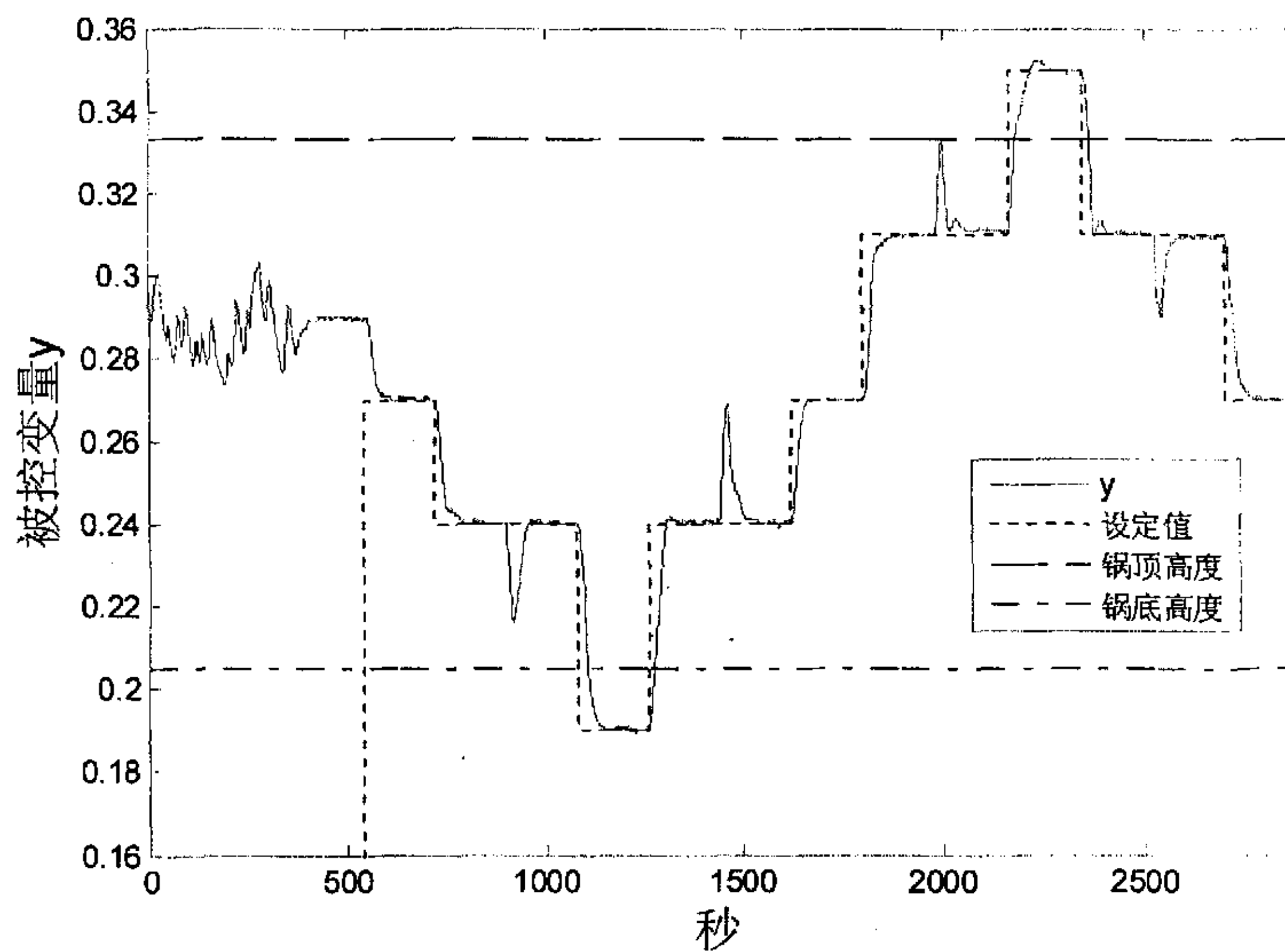


图 6.6 MGPC-2 被控变量 y

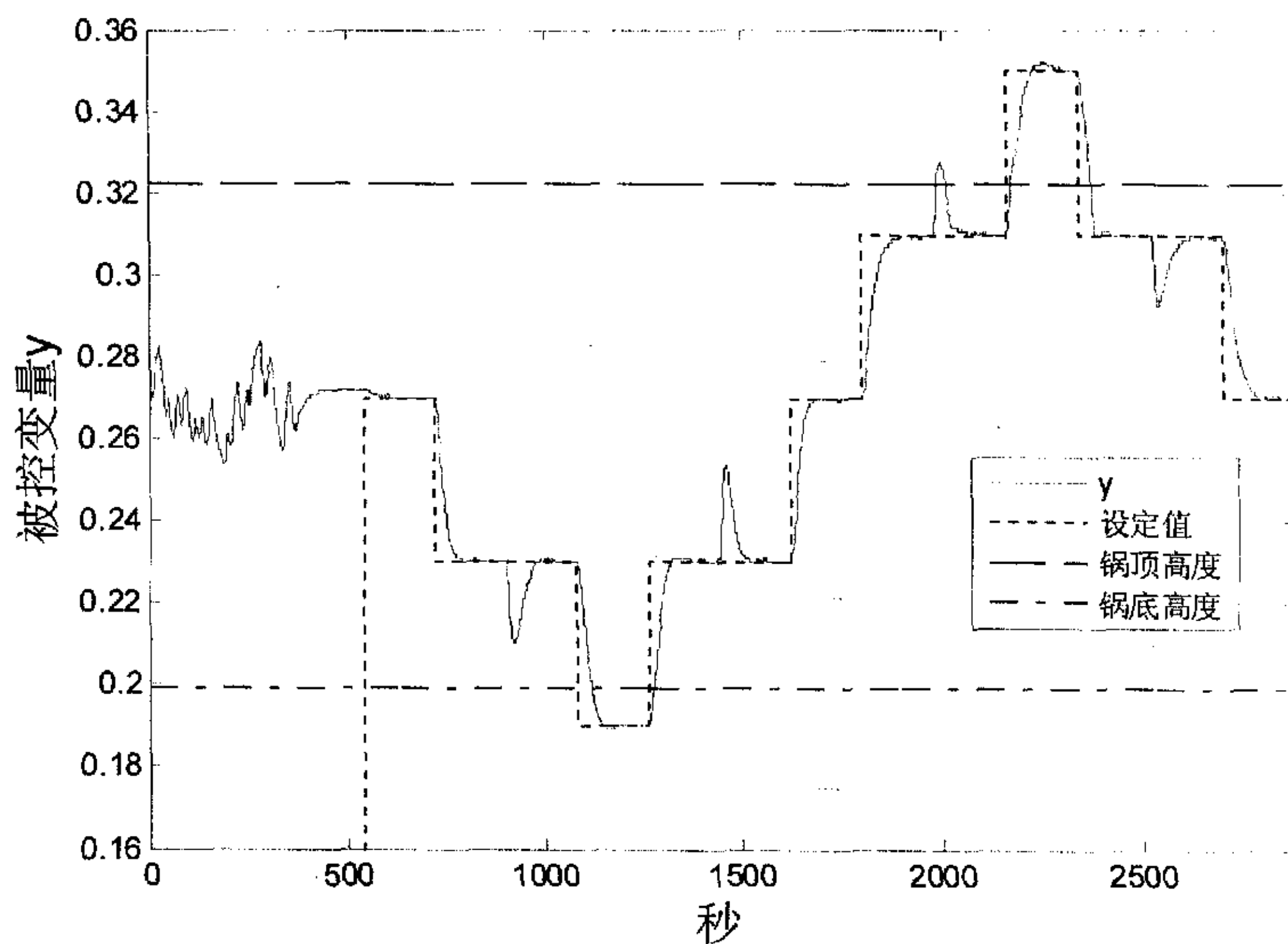


图 6.7 MGPC-3 被控变量 y

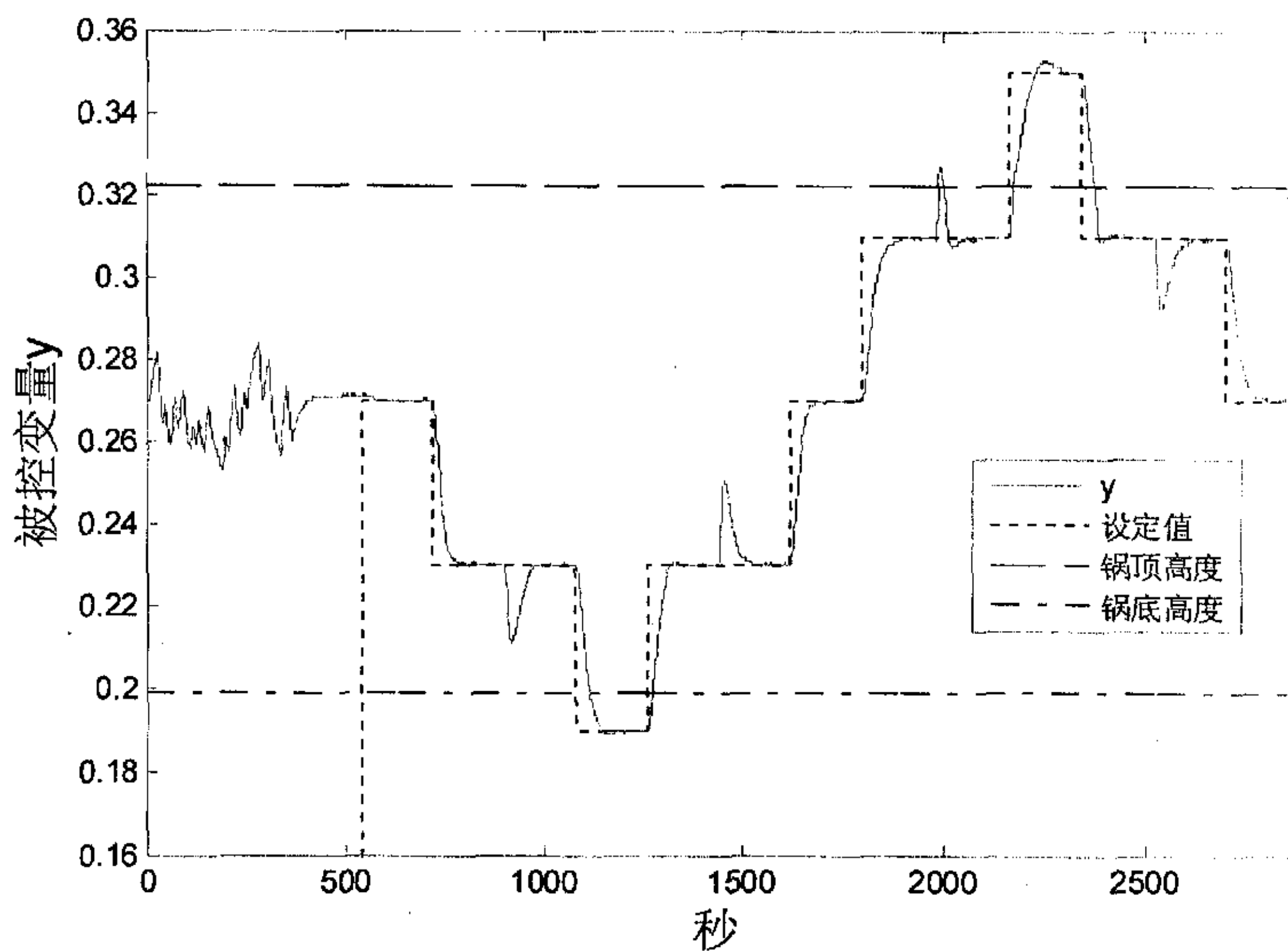


图 6.8 MGPC-4 被控变量 y

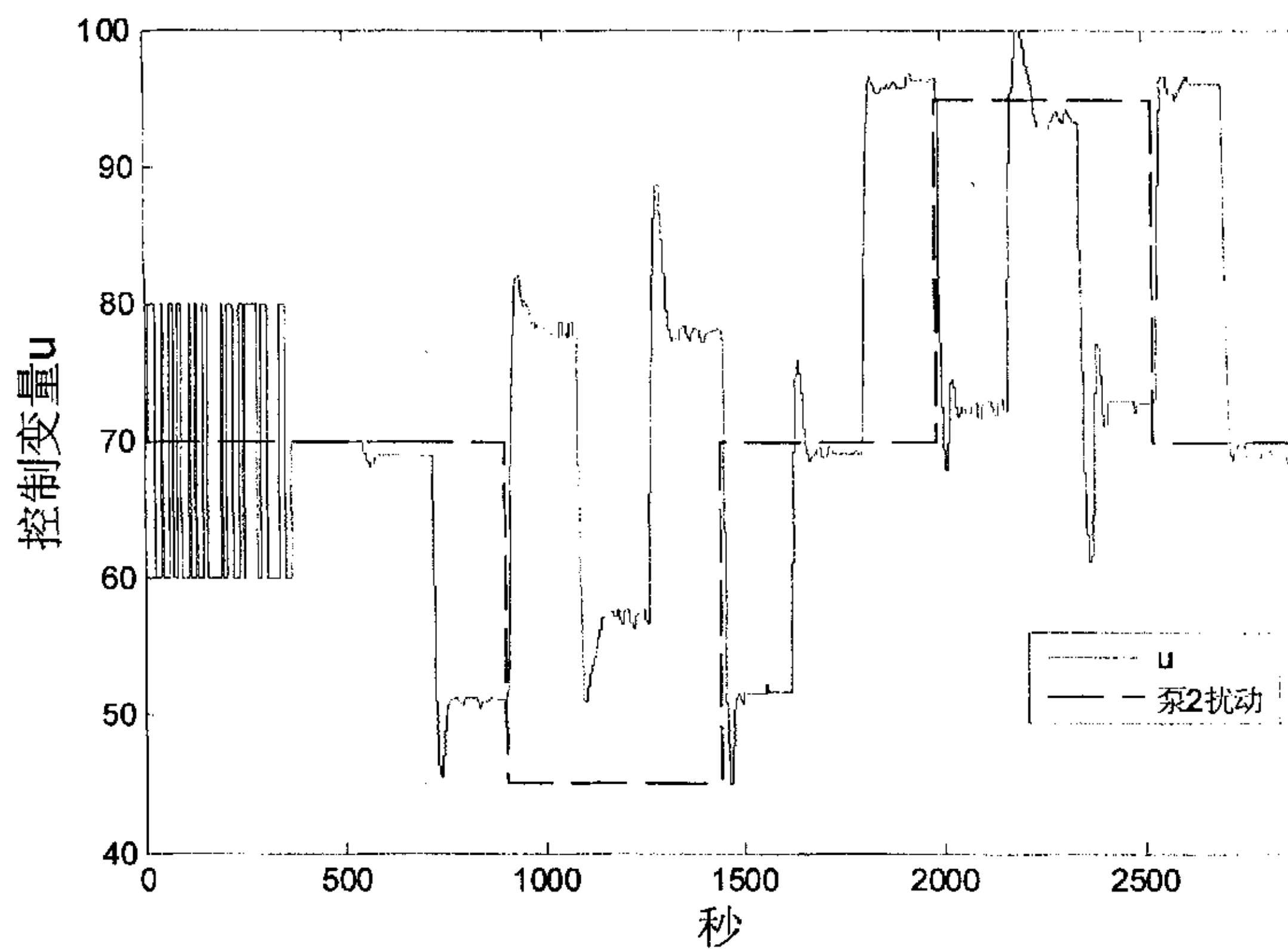


图 6.9 MGPC-1 控制变量 u

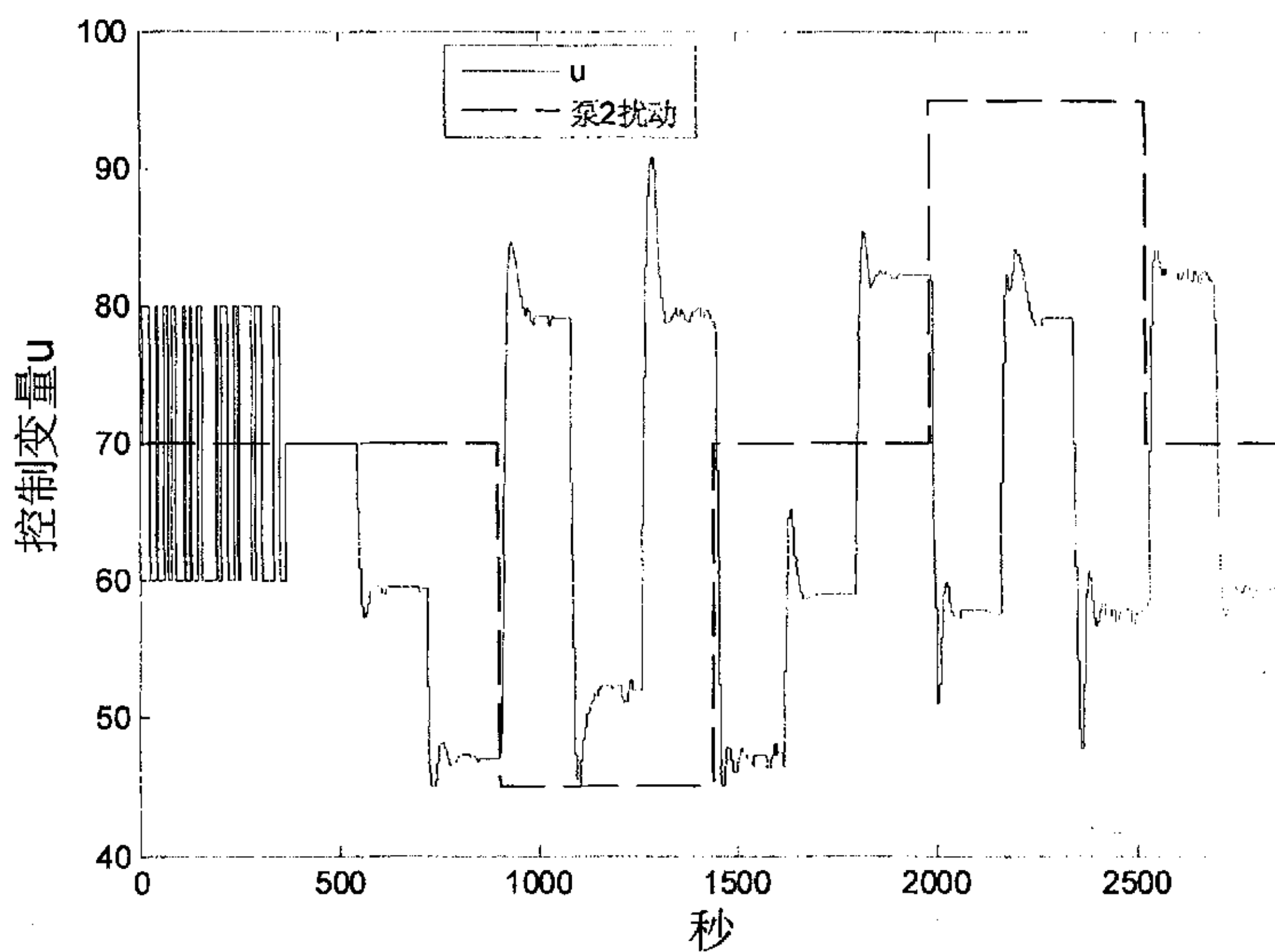


图 6.10 MGPC-2 控制变量 u

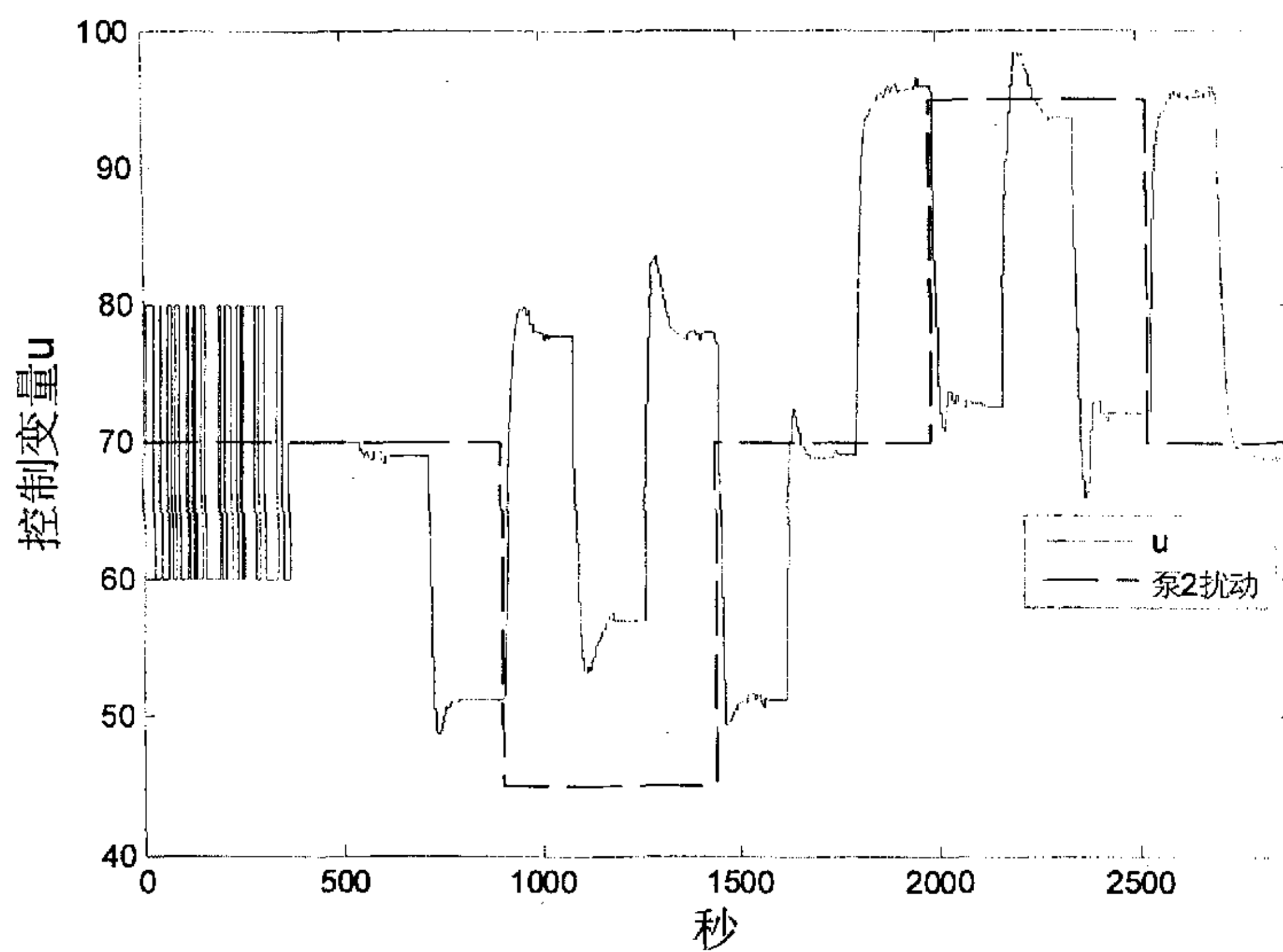


图 6.11 MGPC-3 控制变量 u

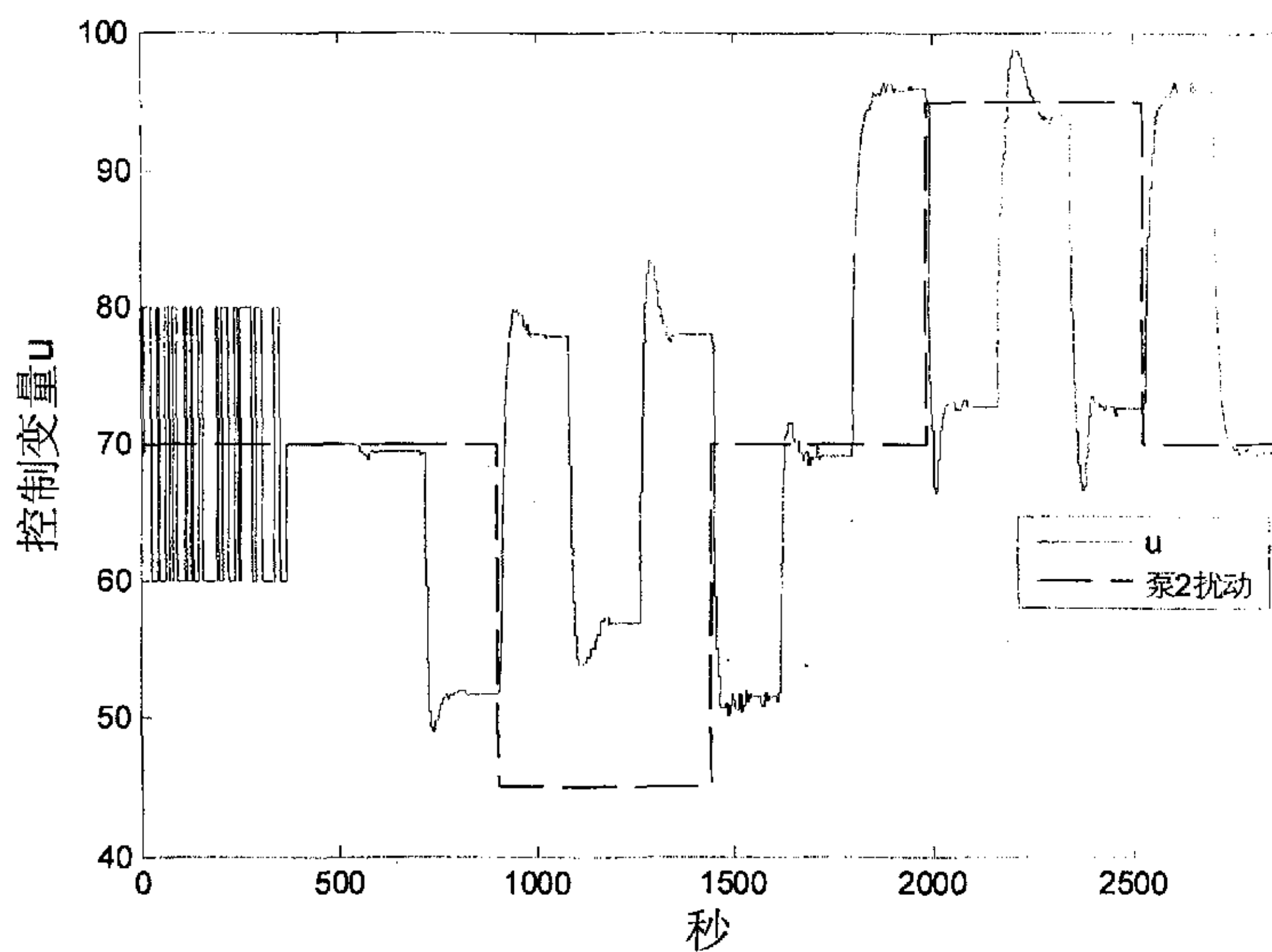


图 6.12 MGPC-4 控制变量 u

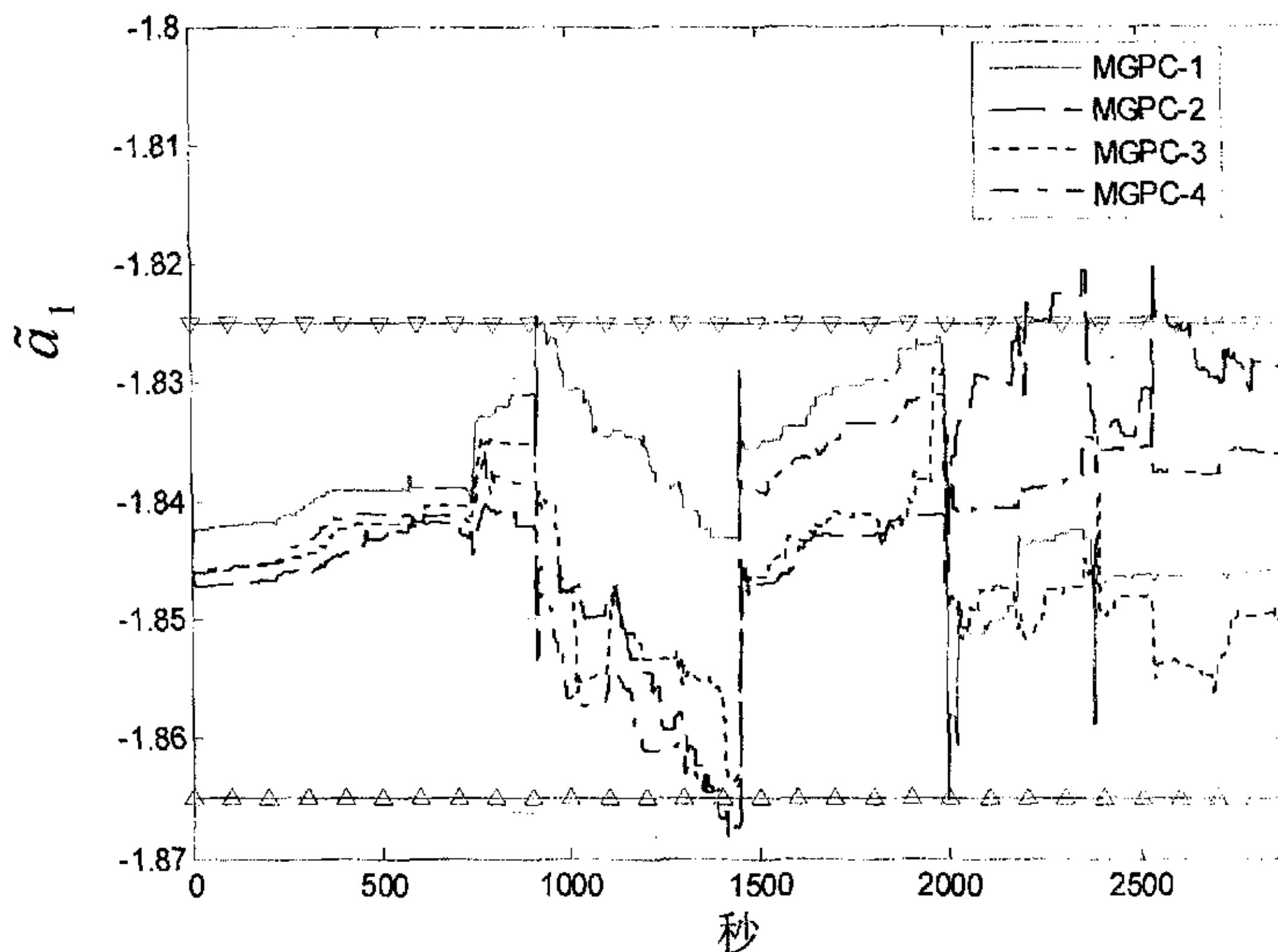


图 6.13 模型参数辨识值 \tilde{a}_1

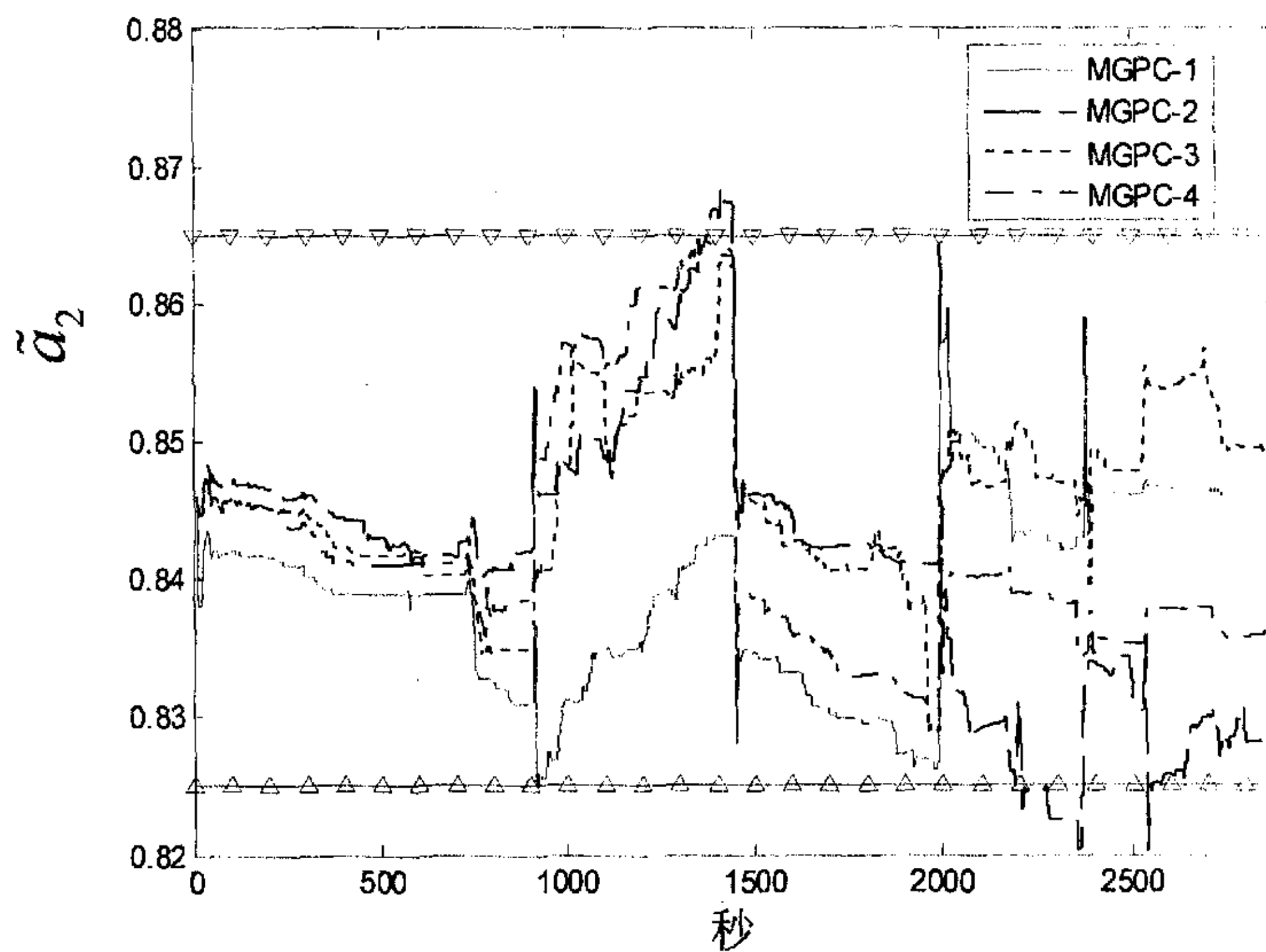


图 6.14 模型参数辨识值 \tilde{a}_2

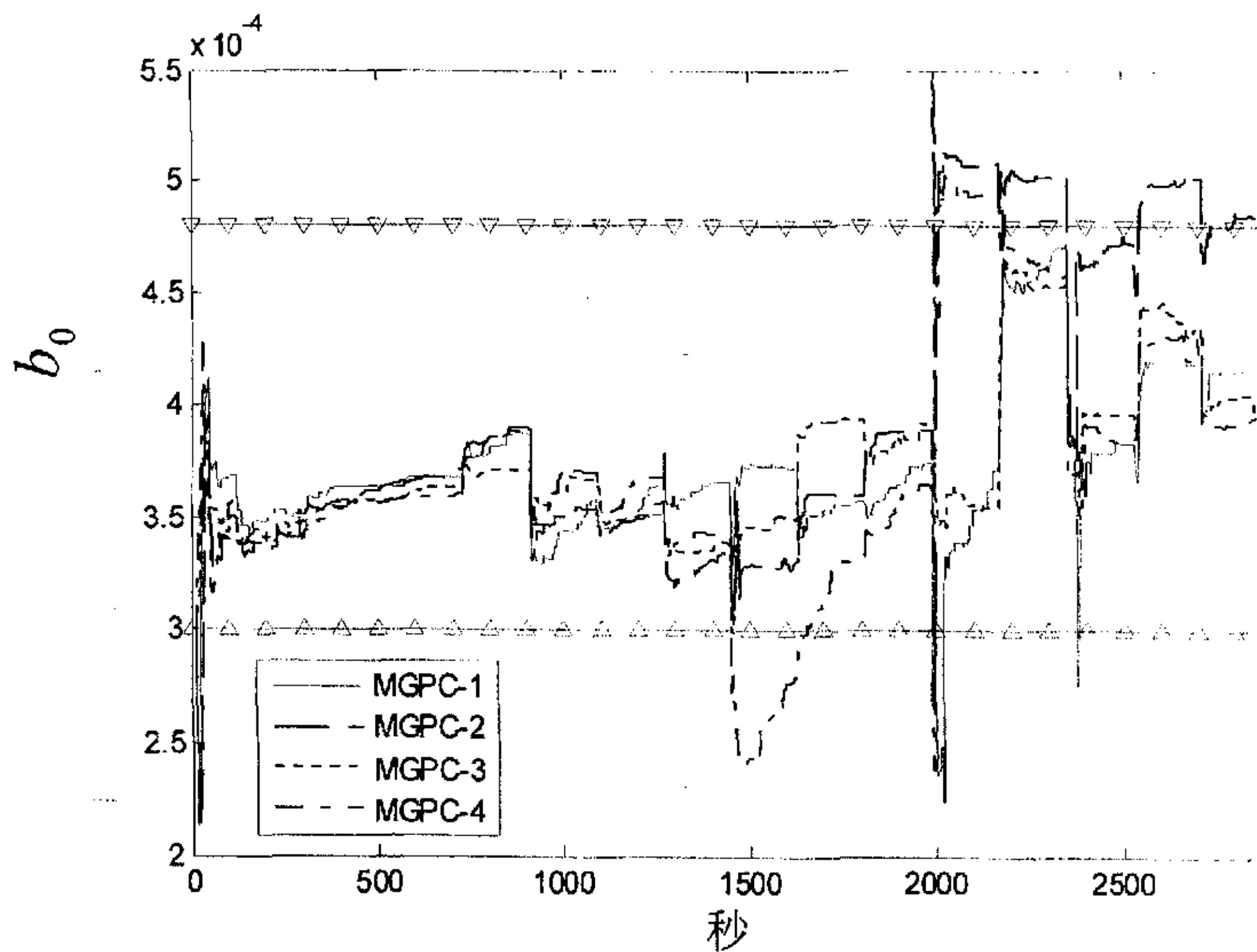


图 6.15 模型参数辨识值 b_0

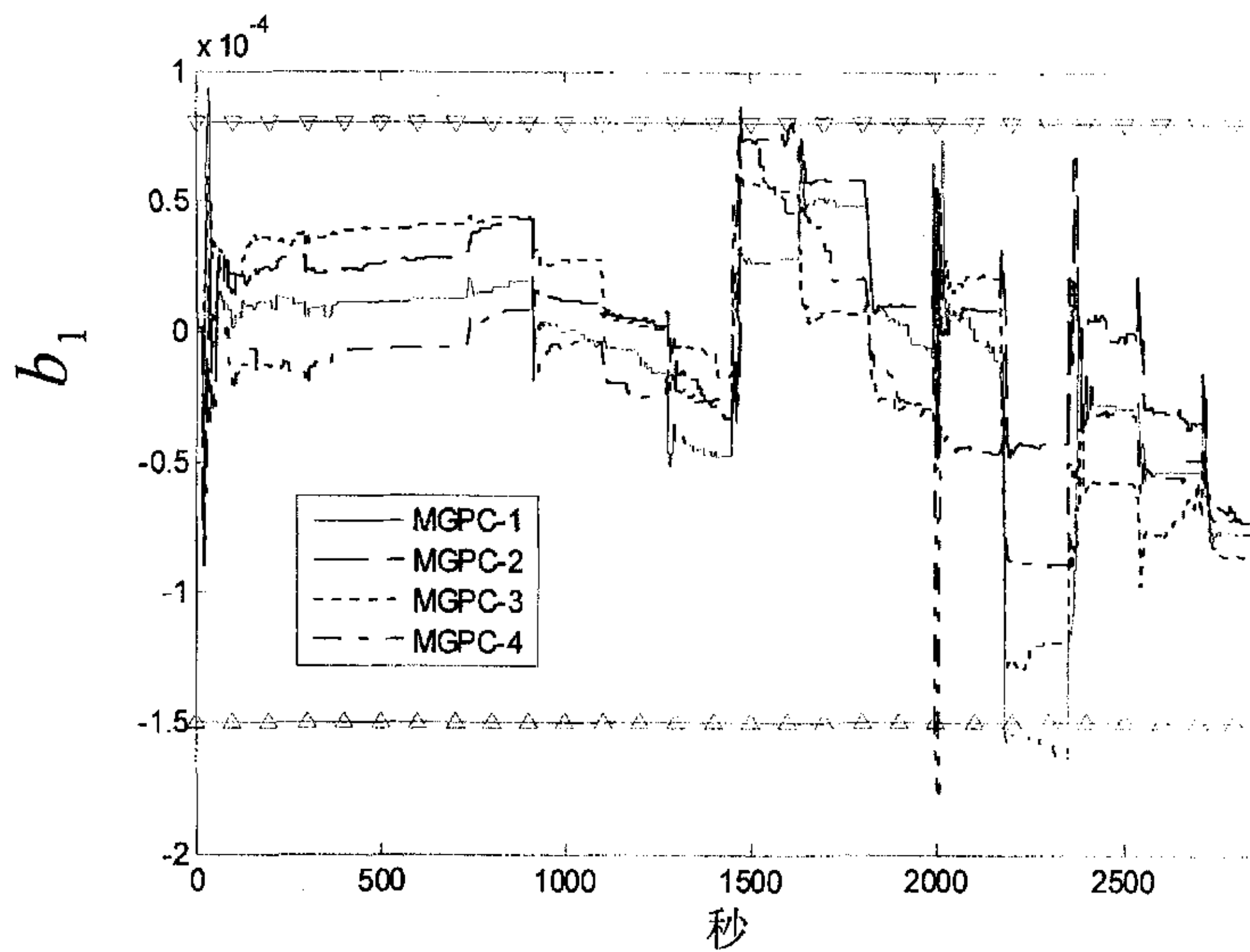
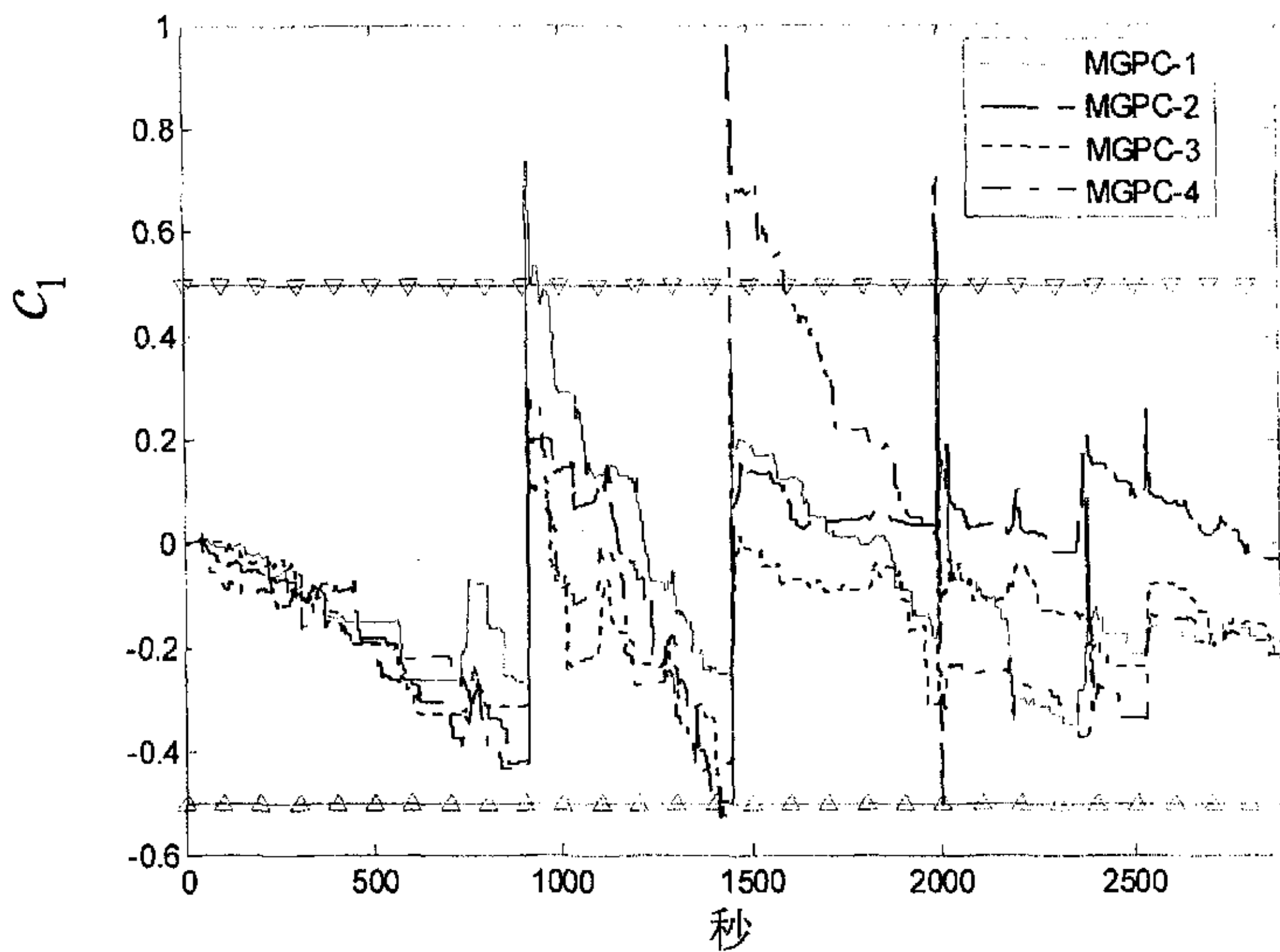


图 6.16 模型参数辨识值 b_1

图 6.17 模型参数辨识值 c_1

6.3.5.2 非自适应模式

在非自适应模式时, MGPC-1、MGPC-2 设定值及扰动的变化情况如表 6.4 所示, MGPC-3、MGPC-4 设定值及扰动的变化情况如表 6.3 所示。其它情况与自适应模式时一样。非自适应模式时, 设计广义预测控制器依据的模型为(6.3)式。控制结果如图 6.18 到图 6.25 所示。

为了与自适应模式进行对照, 整个物理过程力求与自适应模式时保持一致, 辨识器仍然在线运行, 只不过控制器不再按辨识结果在线进行校正了。模型参数的实时辨识结果不再列出。

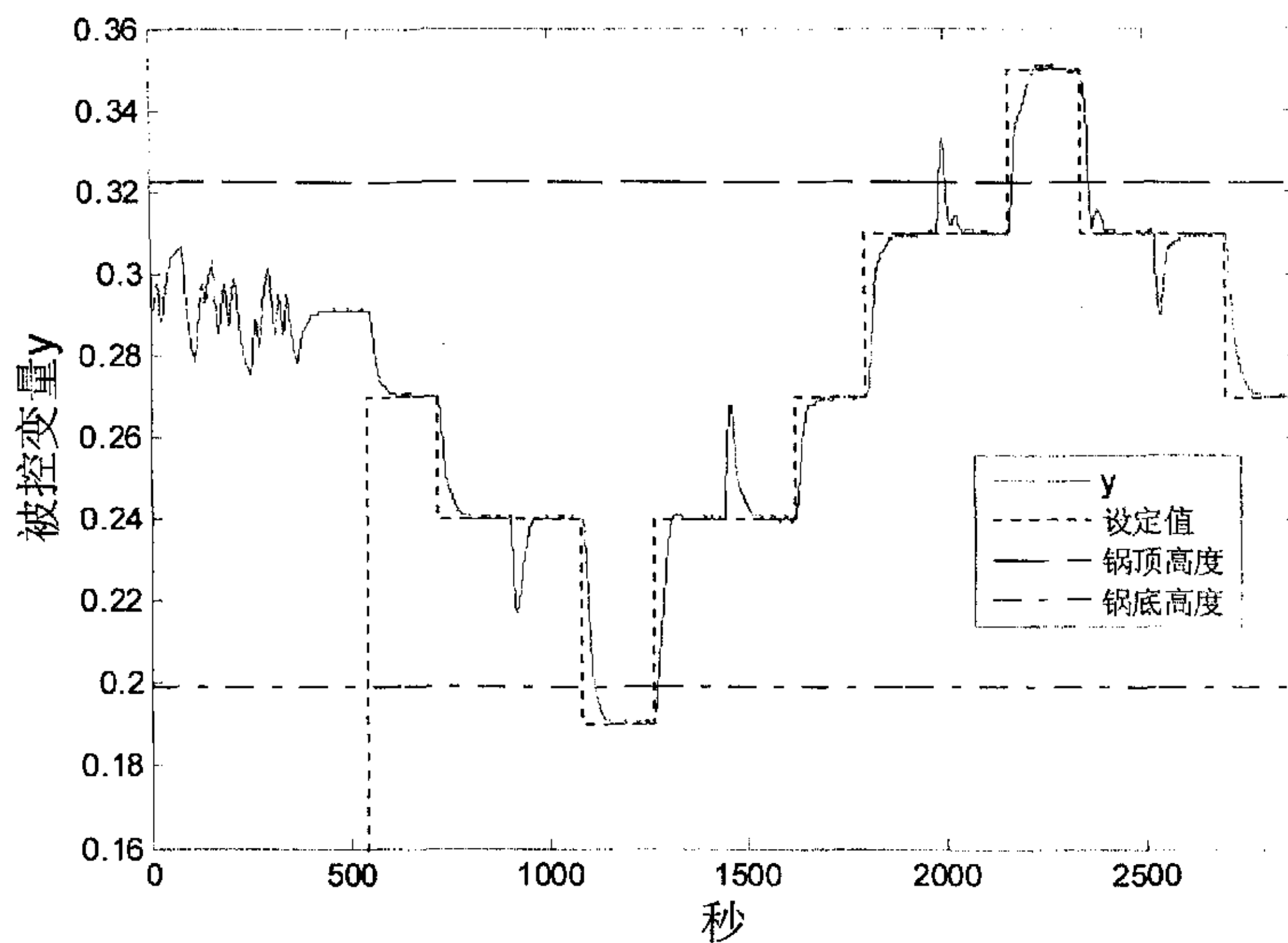


图 6.18 MGPC-1 被控变量 y

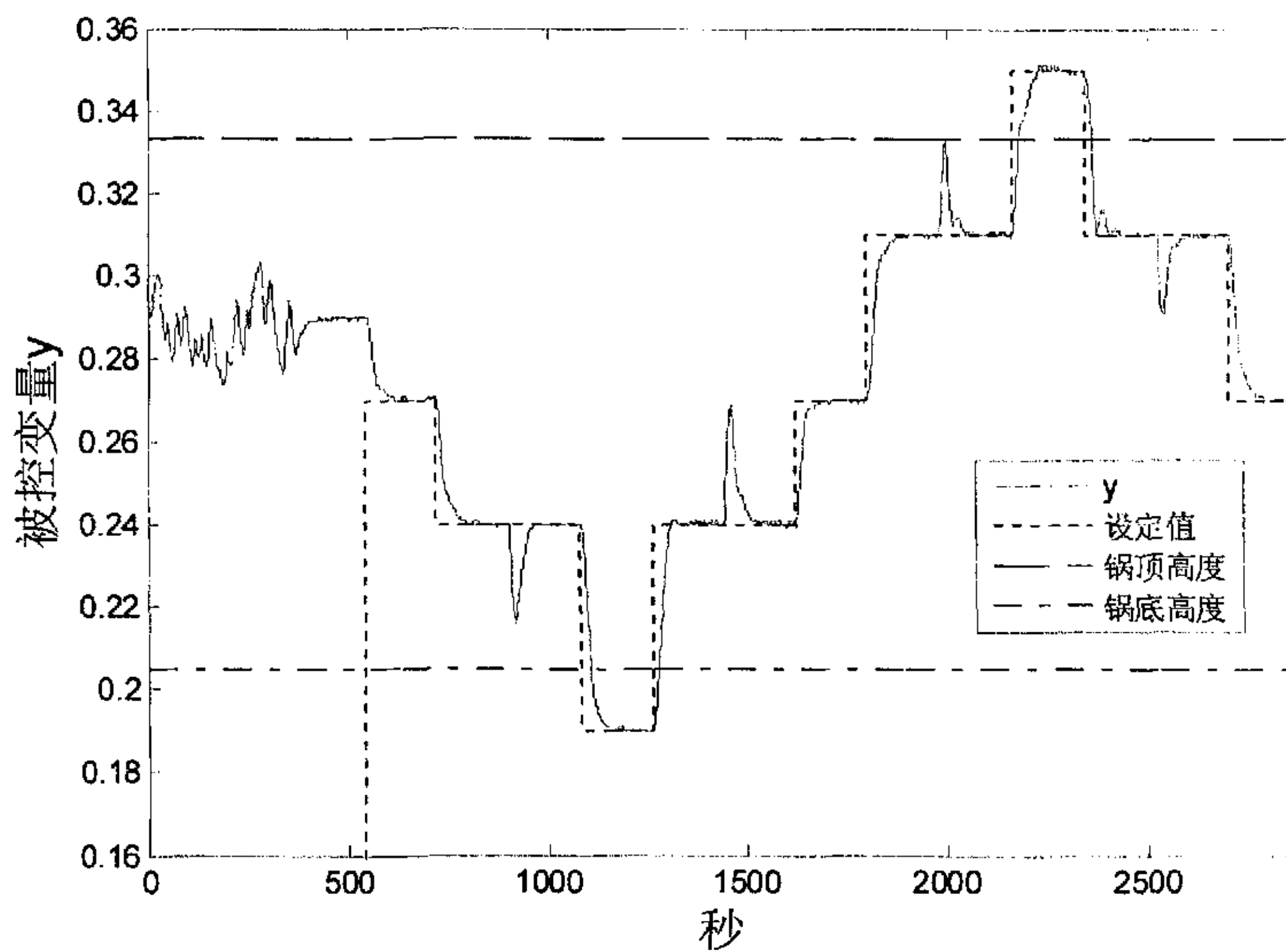


图 6.19 MGPC-2 被控变量 y

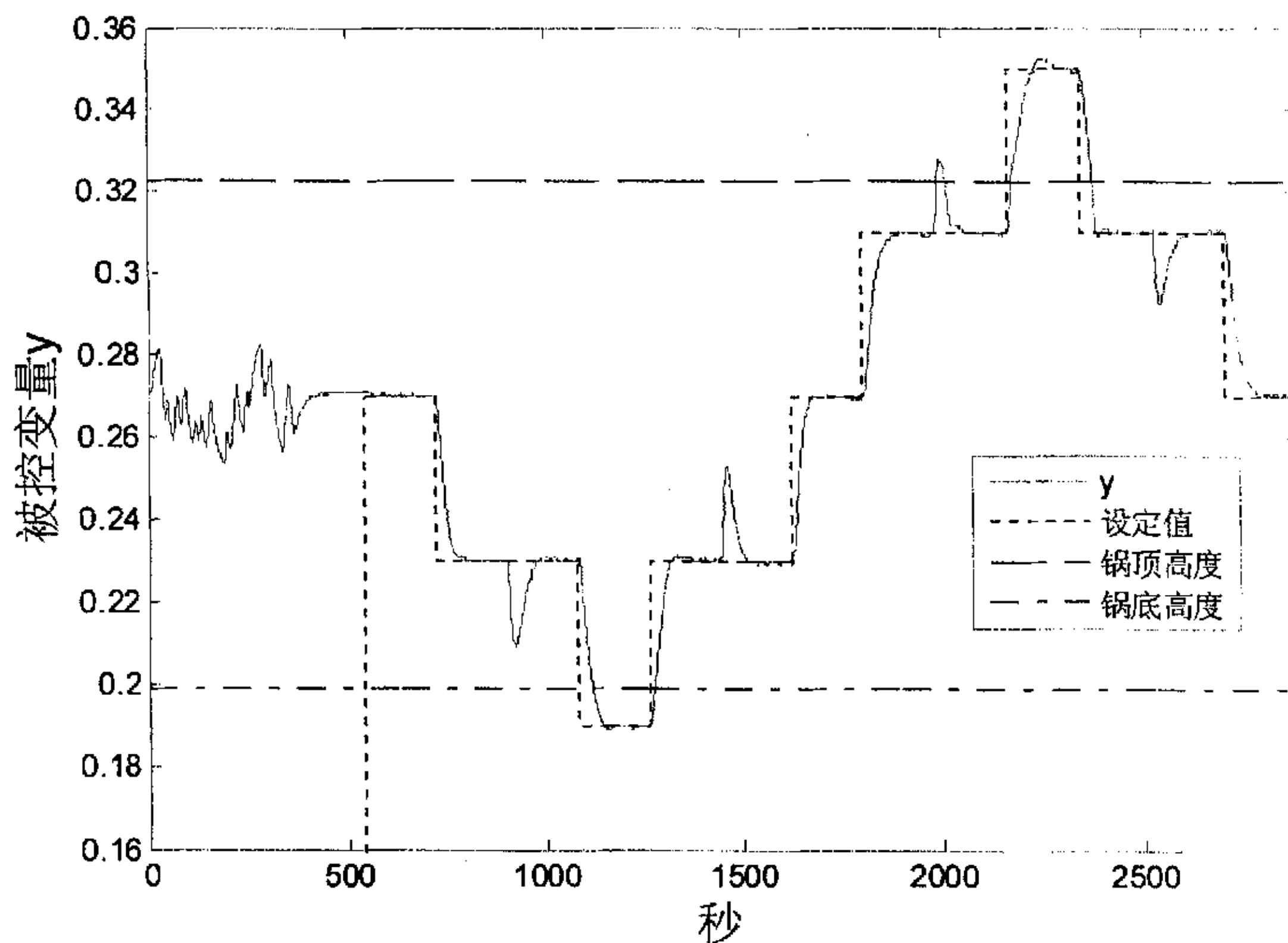


图 6.20 MGPC-3 被控变量 y

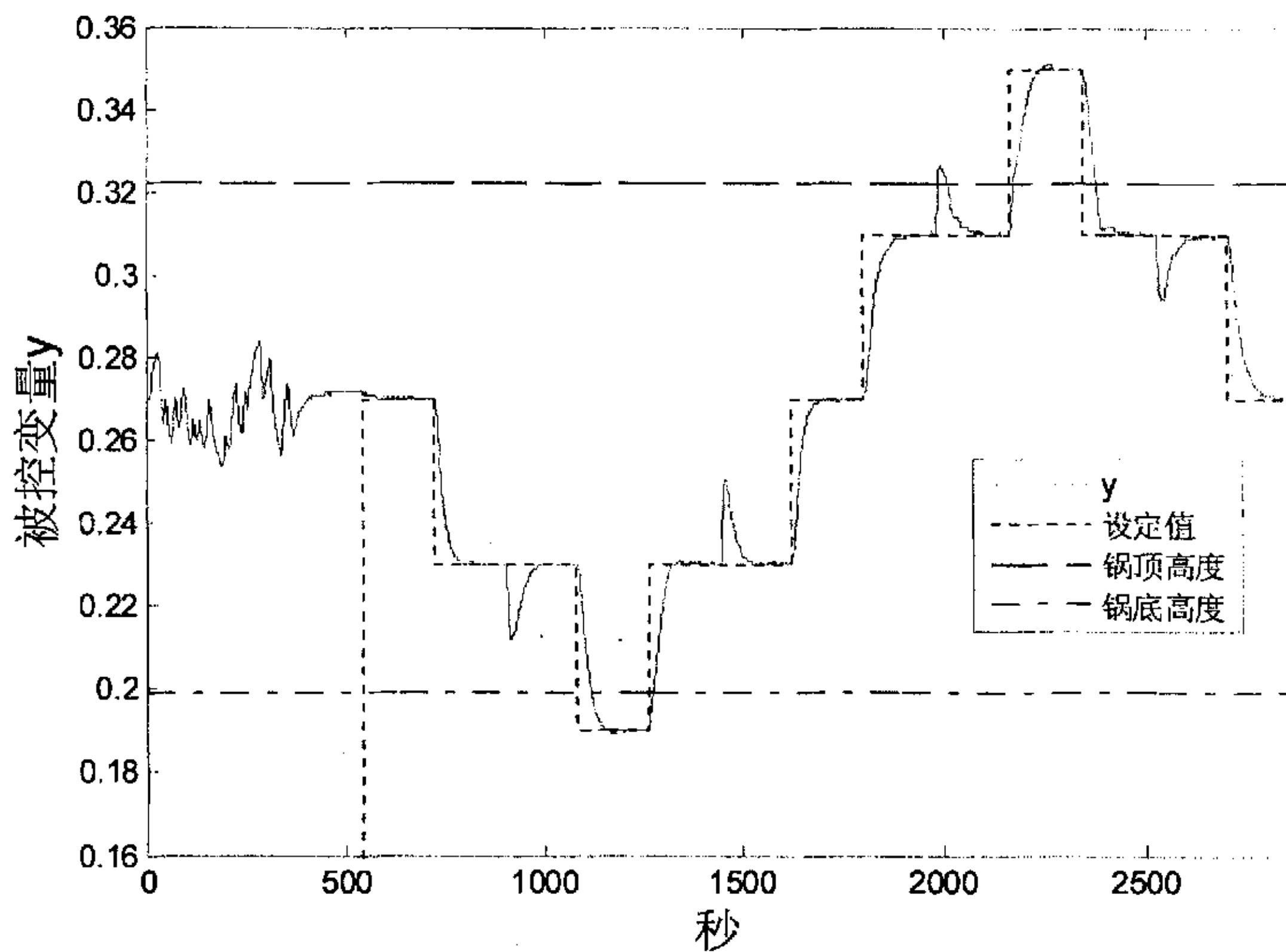


图 6.21 MGPC-4 被控变量 y

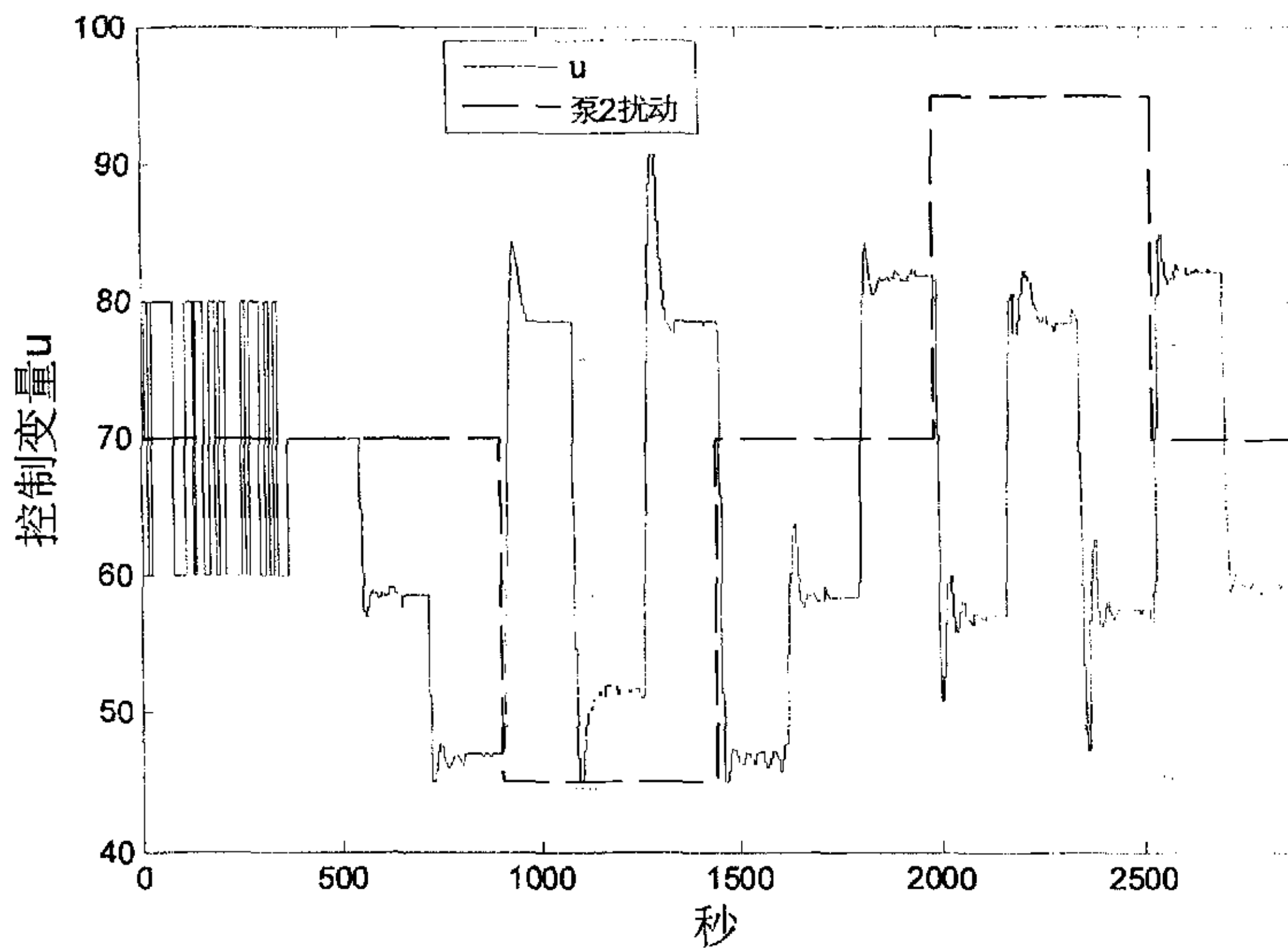


图 6.22 MGPC-1 控制变量 u

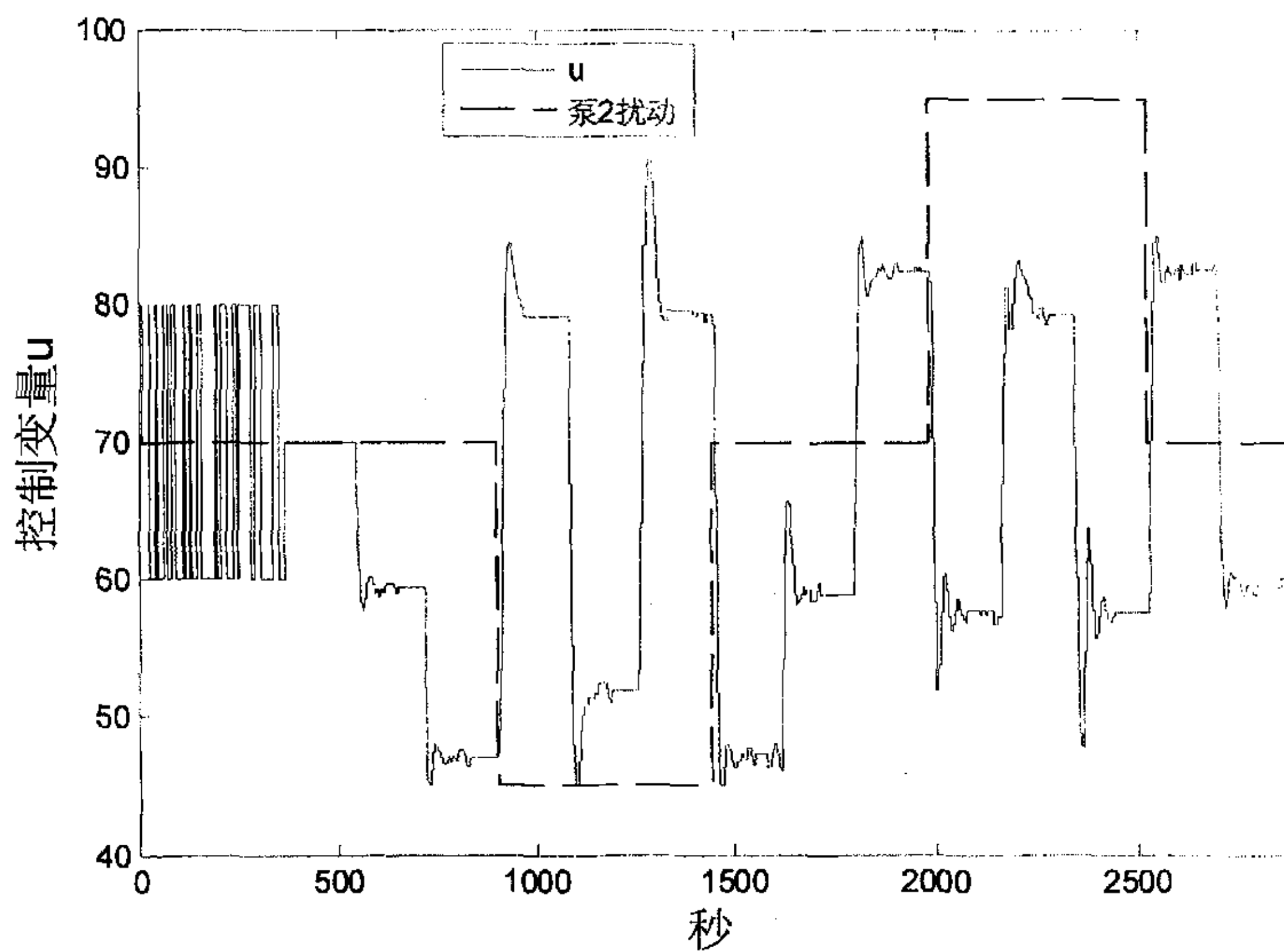


图 6.23 MGPC-2 控制变量 u

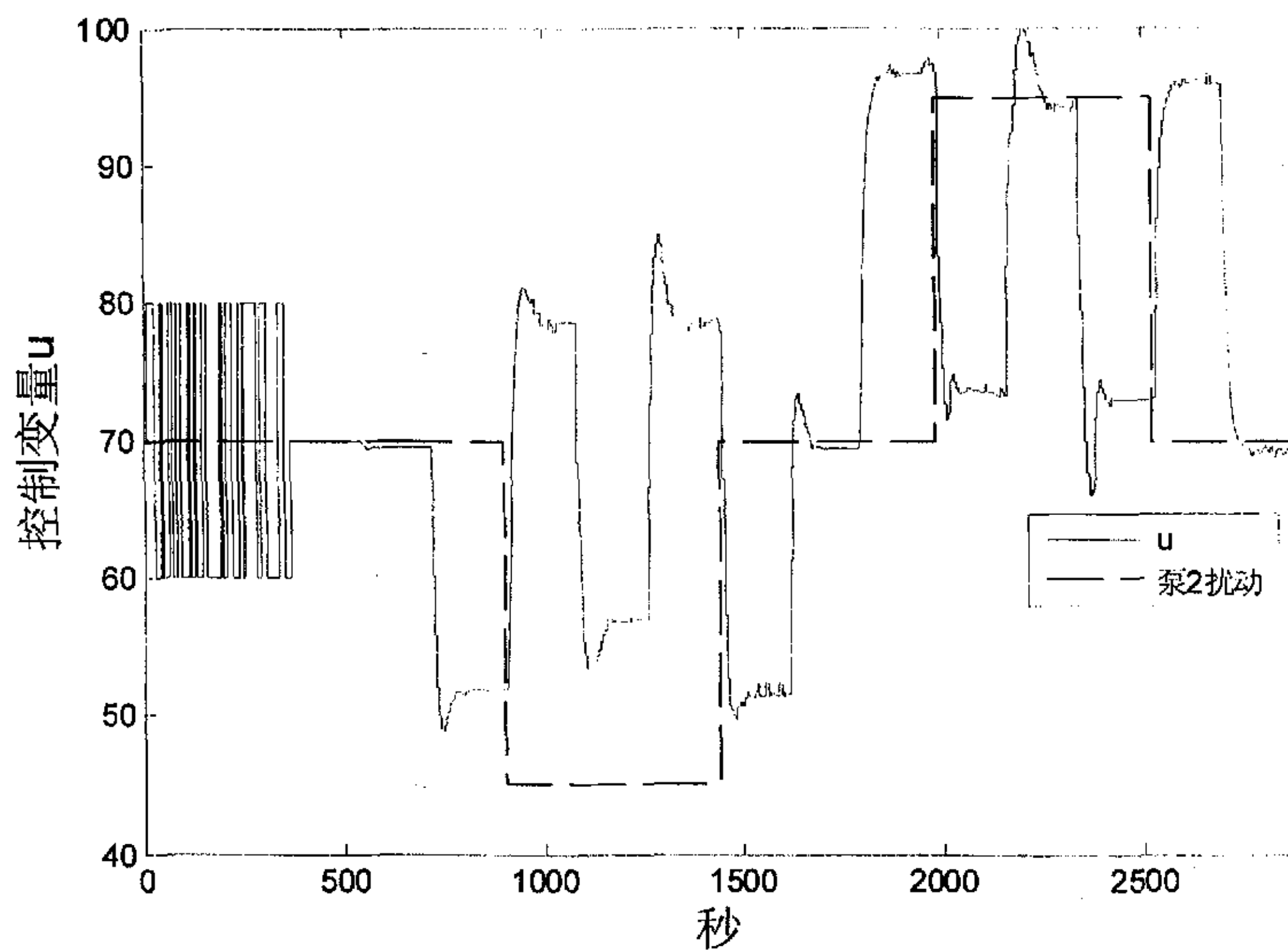


图 6.24 MGPC-3 控制变量 u

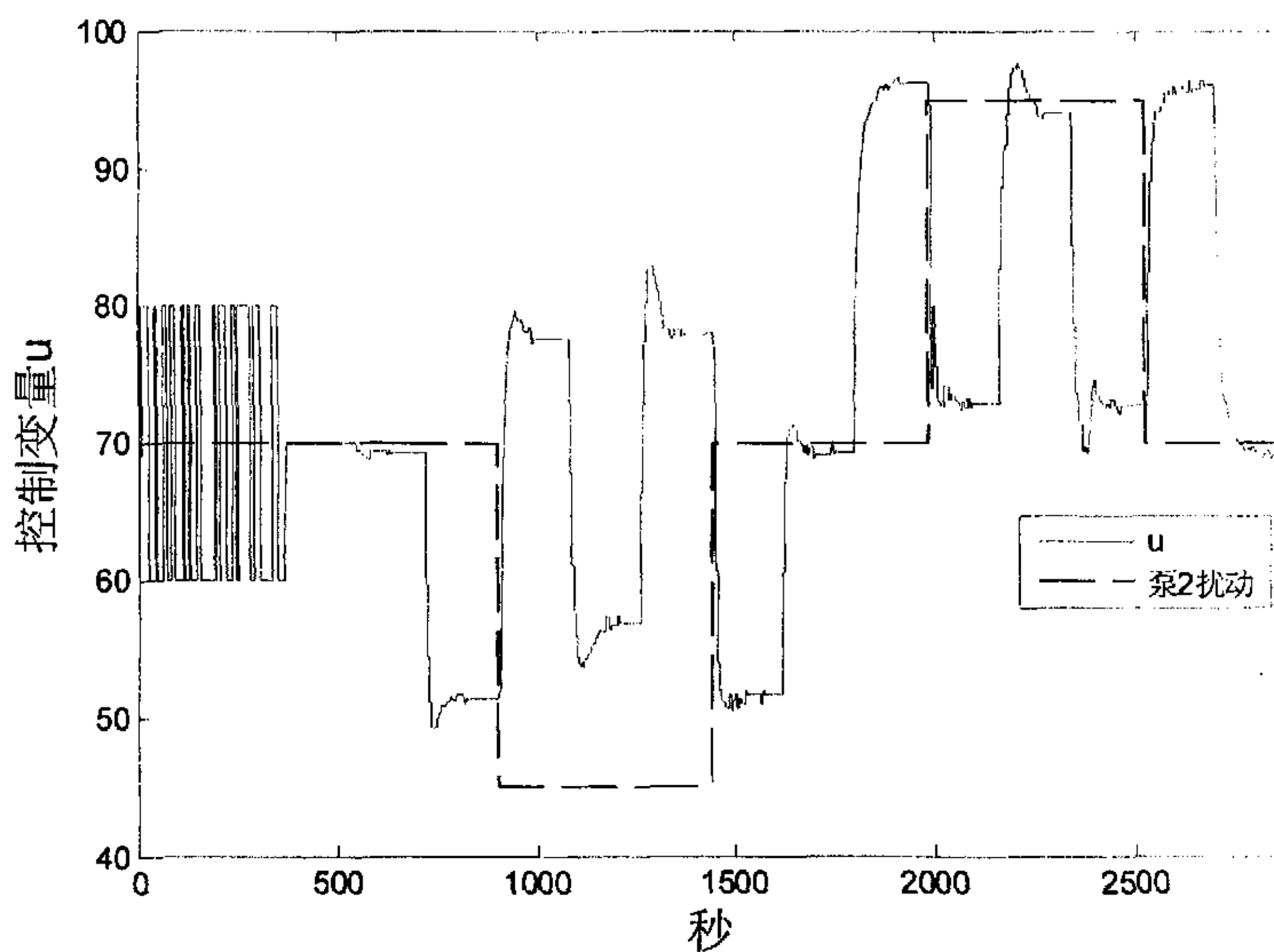


图 6.25 MGPC-4 控制变量 u

6.3.6 非线性单容液位系统实验结果分析

所有实验经历的物理过程基本相同，图 6.13 到图 6.17 的模型参数实时辨识曲线也具有相同的变化趋势，表明实验结果具有可比性。

在遇到强大扰动，或者液位正向或者反向穿越锅形容器顶部平面时，模型参数辨识值变化幅值很大。这是由系统的非线性引起的，当液位处于不同高度或扰动泵 2 处于不同工作频率时，对象的动态特性差异很大，如果用同样结构的线性化模型来描述系统，则其模型参数值就会变化很大。

表 6.5 和表 6.6 给出了各个实验在 721 秒到 2880 秒之间，系统输出跟踪设定值的误差绝对值和以及误差平方和。表中数据说明，各个算法不论是自适应模式，还是非自适应模式，控制效果几乎相同。在自适应模式下，跟踪速度与控制质量相对有一点提高，但不明显。

表 6.5 自适应模式下的误差绝对值和与误差平方和

算法	误差绝对值和	误差平方和
MGPC-1	9.4430	0.2202
MGPC-2	9.9260	0.2430
MGPC-3	11.2020	0.2586
MGPC-4	10.9490	0.2541

表 6.6 非自适应模式下的误差绝对值和与误差平方和

算法	误差绝对值和	误差平方和
MGPC-1	10.0500	0.2417
MGPC-2	10.0120	0.2413
MGPC-3	11.1990	0.2611
MGPC-4	11.1290	0.2497

实验中设定值的变化和泵 2 引入的扰动均非常大，设定值的变化范围几乎遍布了液位的可调区间，而泵 2 扰动的幅值每次都接近其可调频率范围的一半，但实验都取得了满意的控制效果，这表明所有算法具有良好的跟踪能力及抗扰动能力。在非自适应模式下，尽管实际对象的特性发生了较大变化，但控制器性能并没有明显的下降，亦说明各个算法具有很强的鲁棒性。

图 6.13 和图 6.17 实时辨识曲线表明，训练神经网络时所选的模型参数的变化区间基本上覆盖了实时控制时模型参数的变化范围，只有极少数的辨识值落在

已训练的区间之外。MGPC-4 的控制结果表明,在神经网络映射器已训练范围内, MGPC-4 完全能够实现 MGPC-3 的控制功能。

6.4 多容液位控制系统实验研究

6.4.1 多容液位系统简介

构造如图 6.26 所示的非线性多容液位系统。被控对象由图 6.1 多容液位装置的中间四个水槽组成,执行机构是泵 1 和泵 2。泵 1 的水分别注入 LT102、LT103、LT203; 泵 2 的水分别注入 LT201、LT203、LT103。LT103 的水流入 LT102, LT203 的水流入 LT201, LT102 和 LT201 的水流回底部贮水箱。每个水槽的结构均相同,分隔为三部分,泵 1 或泵 2 的水从左边水槽底部注入,溢流进入中间方形容器,然后向右从带纵向开口的隔板流入右边的排水槽。

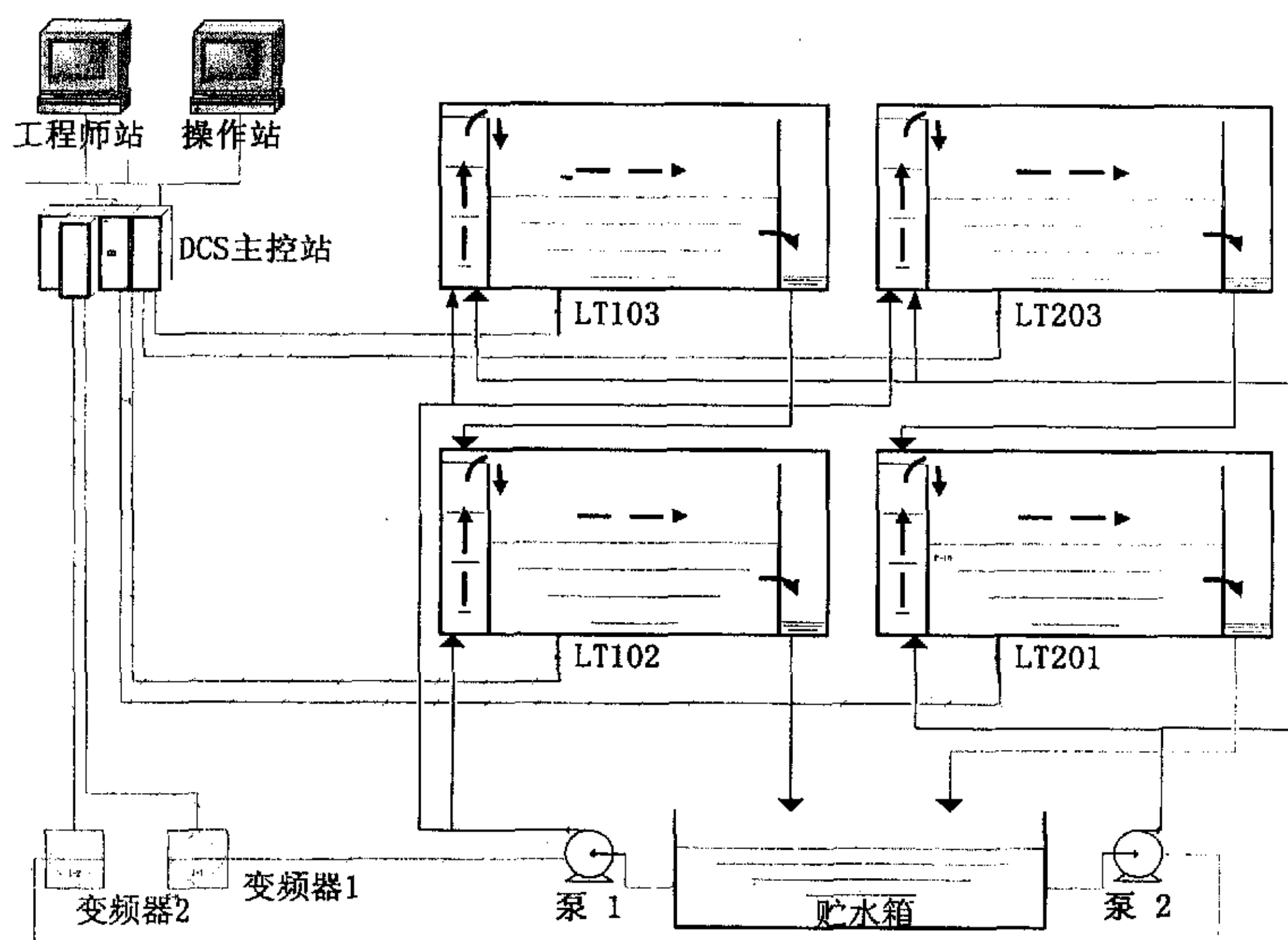


图 6.26 多容液位系统工艺流程示意图

6.4.2 多容液位系统的模型

为了解多容液位系统的动态特性分布情况,在表 6.7 所示测试点位置对系统进行了模型测试。测试信号为伪随机二进制信号,共进行了九组测试。

表 6.7 模型测试点

序号	泵 1	泵 2
1	60	60
2	60	75
3	60	90
4	75	60
5	75	75
6	75	90
7	90	60
8	90	75
9	90	90

数据采集周期为 1 秒。分别建立了过程的传递函数模型和 CARIMA 模型。建立模型时是分成两个子模型进行分析的,每个输出对应一个子模型。

传递函数模型选用(6.5)式的形式:

$$\begin{bmatrix} y_1(s) \\ y_2(s) \end{bmatrix} = \begin{bmatrix} \frac{K_{11}}{1+T_{11}s} e^{-\tau_{11}s} & \frac{K_{12}}{1+T_{12}s} e^{-\tau_{12}s} \\ \frac{K_{21}}{1+T_{21}s} e^{-\tau_{21}s} & \frac{K_{22}}{1+T_{22}s} e^{-\tau_{22}s} \end{bmatrix} \begin{bmatrix} u_1(s) \\ u_2(s) \end{bmatrix} \quad (6.5)$$

其中输入输出变量符号对应的物理量如表 6.8 所示:

表 6.8 符号对应表

符号	物理量
u_1	泵 1 的工作频率
u_2	泵 2 的工作频率
y_1	LT102 的液位
y_2	LT201 的液位

模型分析得到各个工作点的传递函数模型参数值如表 6.9 和 6.10 所示:

表 6.9 传递函数模型参数值表(一)

序号	K_{11}	T_{11}	τ_{11}	K_{12}	T_{12}	τ_{12}
1	0.0026	56.0986	1.8687	0.0013	78.7569	11.8563
2	0.0027	56.4672	1.8743	0.0010	81.8063	9.7990
3	0.0026	50.5239	2.3195	0.0007	72.8677	6.8193
4	0.0022	44.6791	2.3316	0.0012	82.2409	10.9572
5	0.0023	47.5274	2.2902	0.0011	93.7406	8.8159
6	0.0022	50.2323	2.5414	0.0008	72.0524	10.8511
7	0.0023	46.5133	2.2807	0.0015	94.3559	7.9381
8	0.0021	44.3918	2.2091	0.0012	119.1141	11.7024
9	0.0021	44.0035	2.1915	0.0007	69.5050	8.9750
均值	0.0023	48.9375	2.2119	0.0011	84.9378	9.7460

表 6.10 传递函数模型参数值表(二)

序号	K_{21}	T_{21}	τ_{21}	K_{22}	T_{22}	τ_{22}
1	0.0015	88.8676	10.8350	0.0028	50.6062	1.7217
2	0.0012	80.1735	10.8544	0.0023	45.6401	1.6543
3	0.0011	69.5888	12.0317	0.0021	39.7835	2.1446
4	0.0009	80.5170	14.2006	0.0028	51.7172	2.4466
5	0.0014	99.7560	9.6172	0.0023	44.8922	2.1160
6	0.0009	67.2114	8.8500	0.0022	46.1419	2.2914
7	0.0008	100.5422	13.3313	0.0025	45.9858	2.4688
8	0.0008	81.7948	13.0706	0.0022	44.7972	2.1093
9	0.0007	71.1300	10.1161	0.0019	38.1141	1.9545
均值	0.0011	82.1757	11.4341	0.0024	45.2976	2.1008

表 6.9 与表 6.10 所示各测试点的增益系数、时间常数和纯滞后分别绘于图 6.27、图 6.28、图 6.29 中。从图中曲线可知,在不同的测试点,对象的特性变化很大,说明过程具有较强的非线性。从图 6.27 可以看出,泵的工作频率越高,则对应的开环增益系数越小。这与系统的物理特征是一致的,因为系统中液体的总量不变,随着泵的工作频率增大,抽出的液体增多,贮水箱中的液位逐渐变低,

导致泵的供水能力相对下降。图 6.28 中时间常数 T_{11} 和 T_{22} 也存在类似的变化规律,这是因为当泵的工作频率较大时,管道上液体流量大,工作频率的较小变化能很快在水槽液位中反映出来,也就是对象的时间常数变小了;相反,当变频泵工作在较小频率时,管道上液体流量小,泵工作频率的变化要在较长的时间后才能从液位对象中反应出来,这时系统的时间常数较大。而图 6.29 显示纯滞后 τ_{11} 与 τ_{22} 在不同工作点变化不大,接近一个常数。

CARIMA 模型选择(6.6)式的形式:

$$\begin{bmatrix} 1+\tilde{a}_{11}z^{-1}+\tilde{a}_{12}z^{-2} & 0 \\ 0 & 1+\tilde{a}_{22}z^{-1}+\tilde{a}_{22}z^{-2} \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} b_{110} & b_{120}z^{-1} \\ b_{210}z^{-1} & b_{220} \end{bmatrix} \begin{bmatrix} \Delta u_1(t-1) \\ \Delta u_2(t-1) \end{bmatrix} + \begin{bmatrix} 1+c_{11}z^{-1} & 0 \\ 0 & 1+c_{22}z^{-1} \end{bmatrix} \begin{bmatrix} e_1(t) \\ e_2(t) \end{bmatrix} \quad (6.6)$$

对实验对象的结构进行分析知,这种模型结构显然不足以描述实际的物理对象。选择这样的模型结构是考虑到用神经网络实现多变量广义预测控制快速算法时,模型参数少可以减少样本的数量,同时也可以减少神经网络的规模,缩短离线训练时间。

采样周期为 5 秒,得到的 CARIMA 模型的参数值如表 6.11 和表 6.12。

得到 CARIMA 的平均模型为:

$$\begin{bmatrix} 1-1.9551z^{-1}+0.96159z^{-2} & 0 \\ 0 & 1-1.9424z^{-1}+0.94625z^{-2} \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} 0.15118e^{-3} & 1.8964e-5z^{-1} \\ 2.0352e-5z^{-1} & 0.17828e^{-3} \end{bmatrix} \begin{bmatrix} u_1(t-1) \\ u_2(t-1) \end{bmatrix} + \begin{bmatrix} 1-0.85306z^{-1} & 0 \\ 0 & 1-0.89747z^{-1} \end{bmatrix} \begin{bmatrix} e_1(t) \\ e_2(t) \end{bmatrix} \quad (6.7)$$

表 6.11 和表 6.12 中的模型参数分别绘制于图 6.30、图 6.31 和图 6.32 中。图中参数的分布情况同样表明过程具有较大的非线性。 $\tilde{\mathbf{A}}(z^{-1})$ 与 $\mathbf{B}(z^{-1})$ 多项矩阵的系数具有相似的变化规律,即,泵的工作频率越高,则对应系数值越小。这与前面传递函数中增益系数与时间常数的变化相一致,过程的时间常数直接决定着 $\tilde{\mathbf{A}}(z^{-1})$ 中系数的大小,而 $\mathbf{B}(z^{-1})$ 中系数的大小与开环增益系数的大小对应关系。

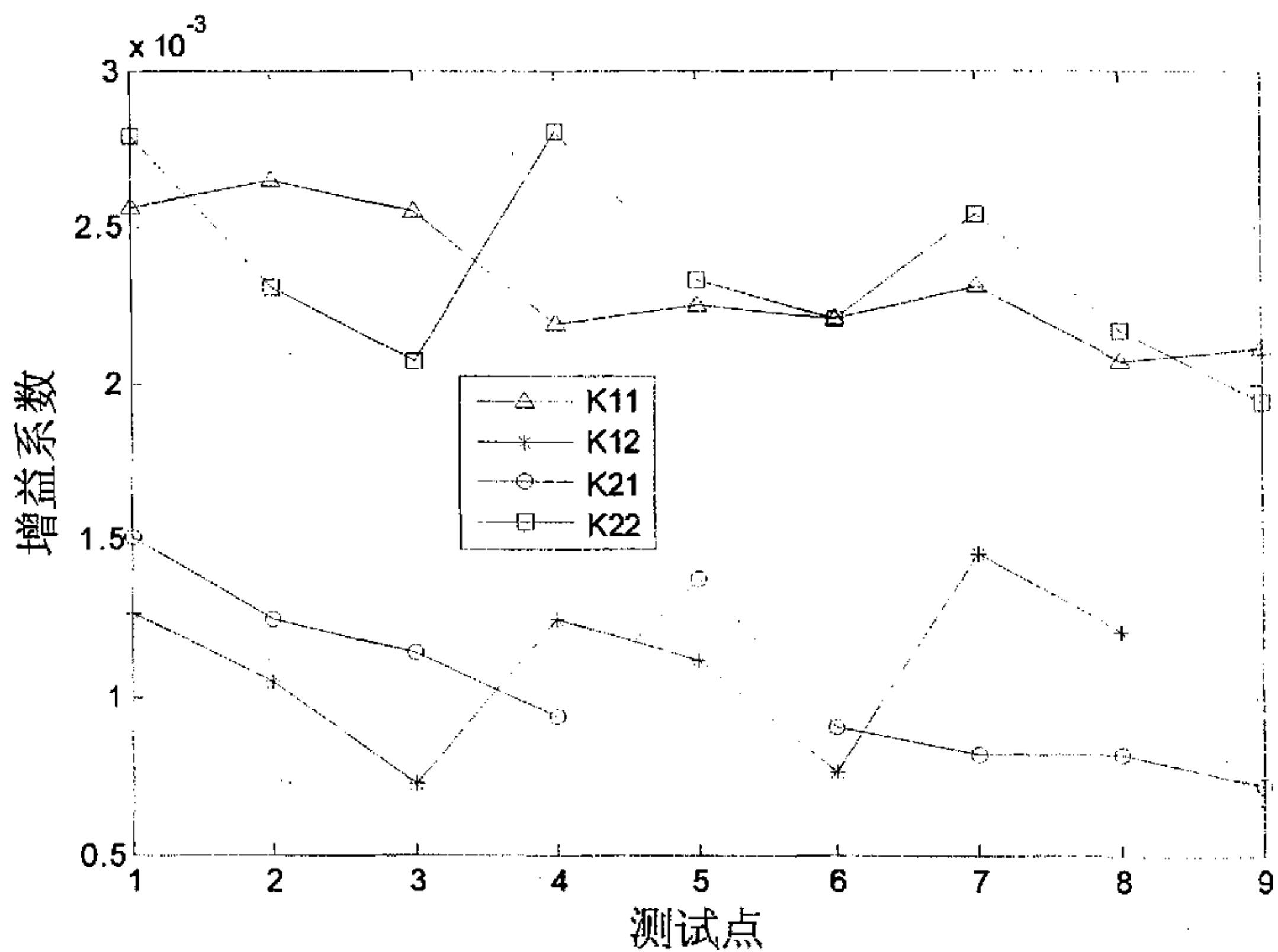


图 6.27 增益系数

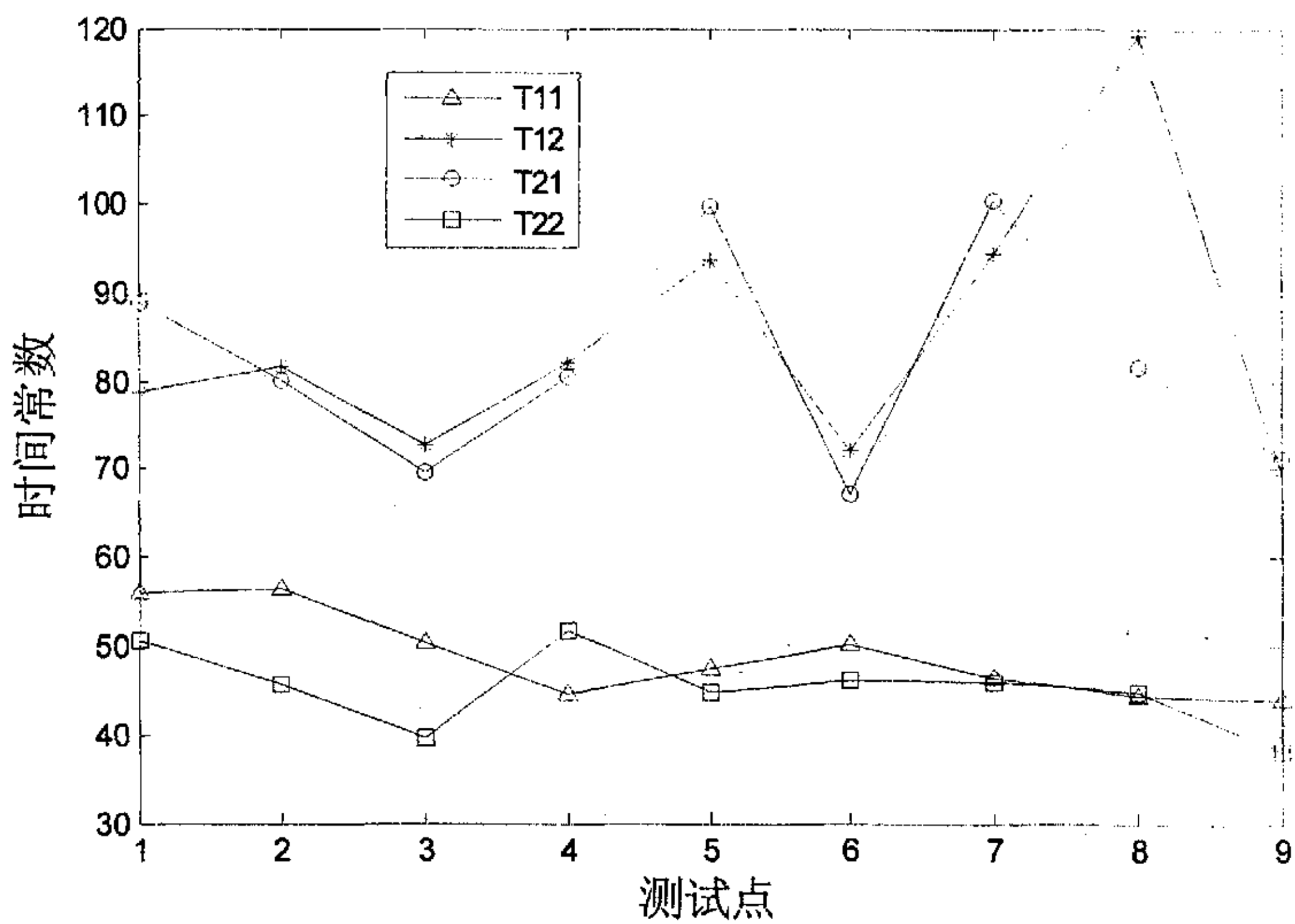


图 6.28 时间常数

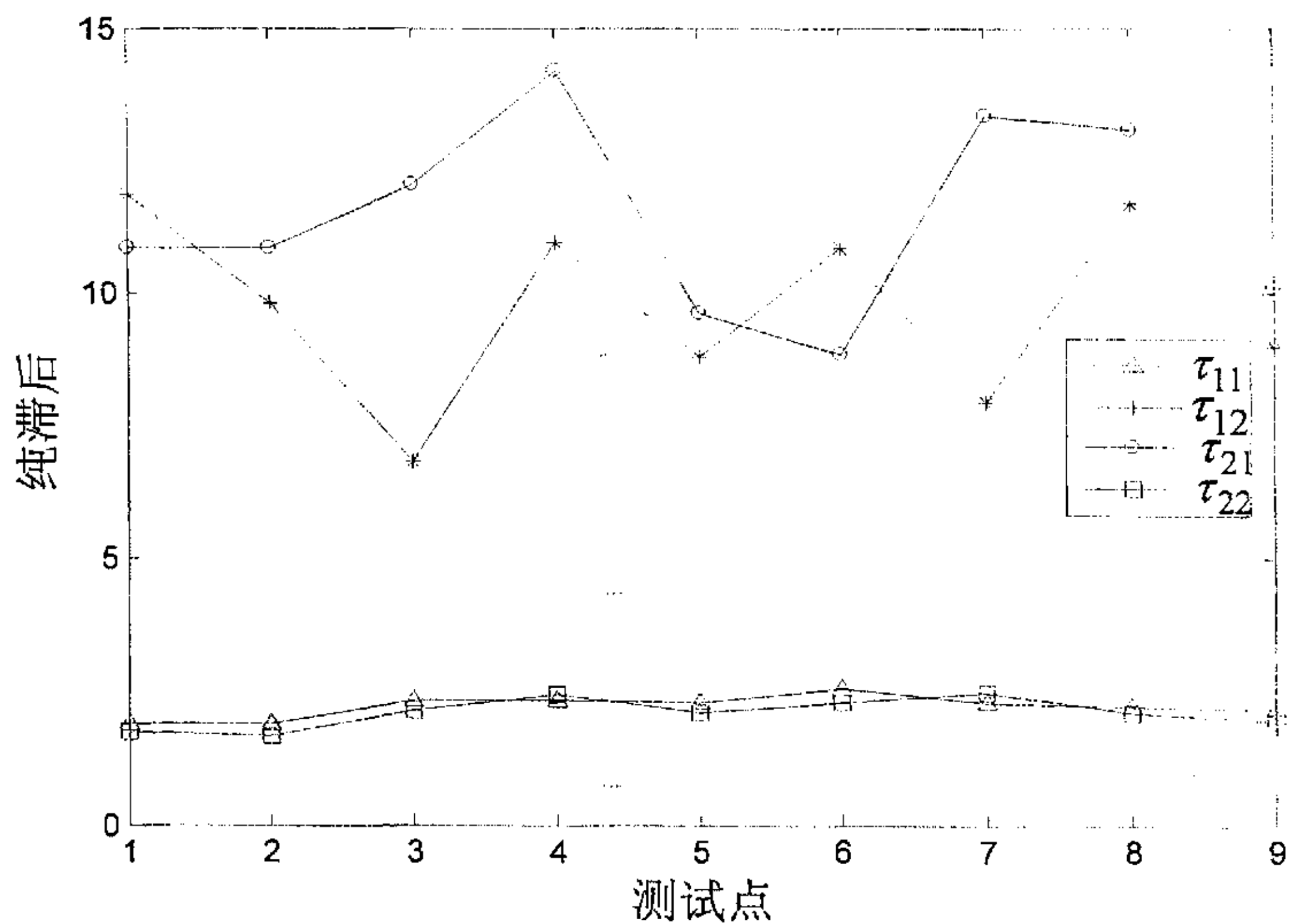


图 6.29 纯滞后

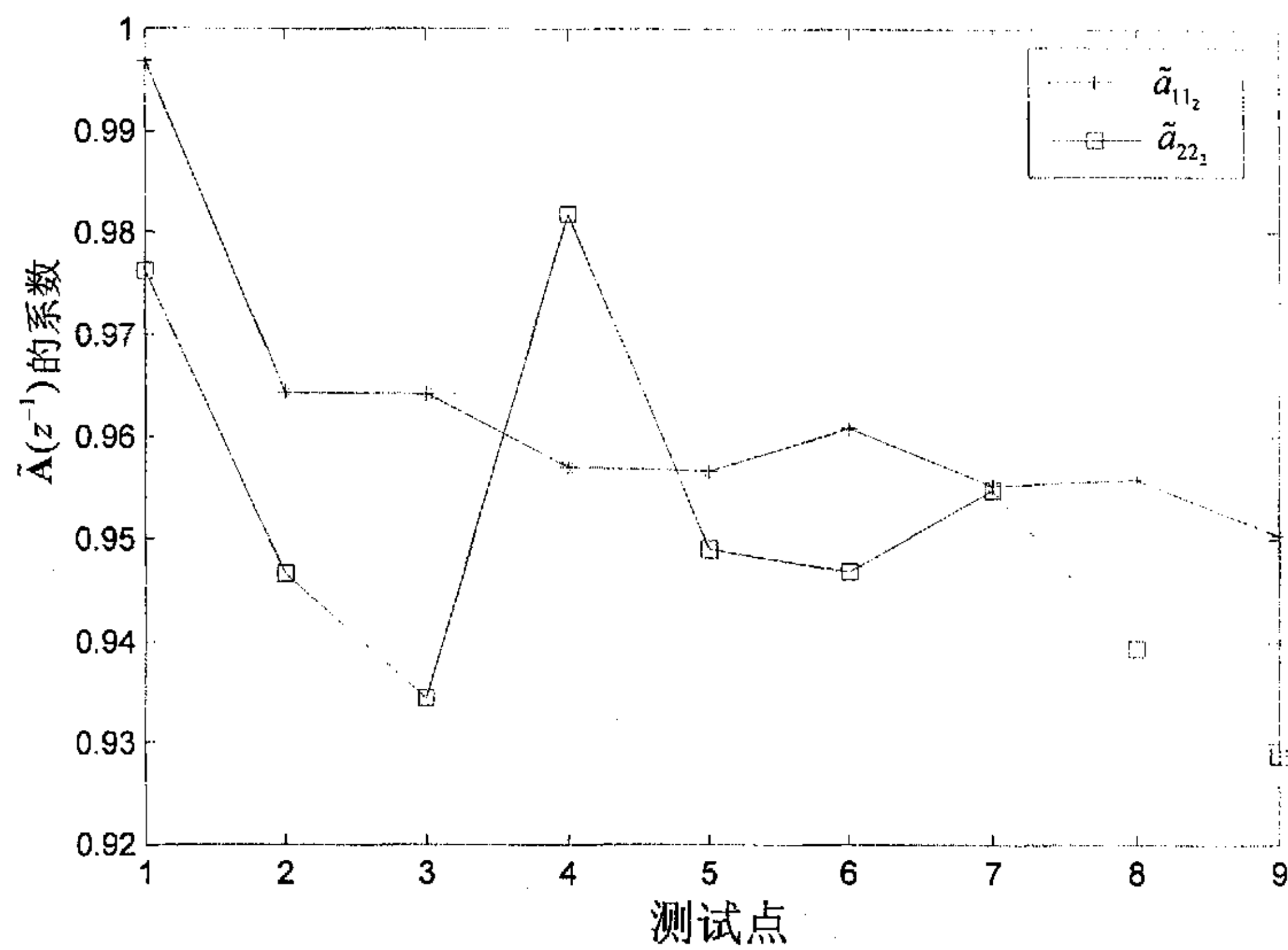


图 6.30 $\tilde{A}(z^{-1})$ 的系数 \tilde{a}_{11_2} 和 \tilde{a}_{22_2}

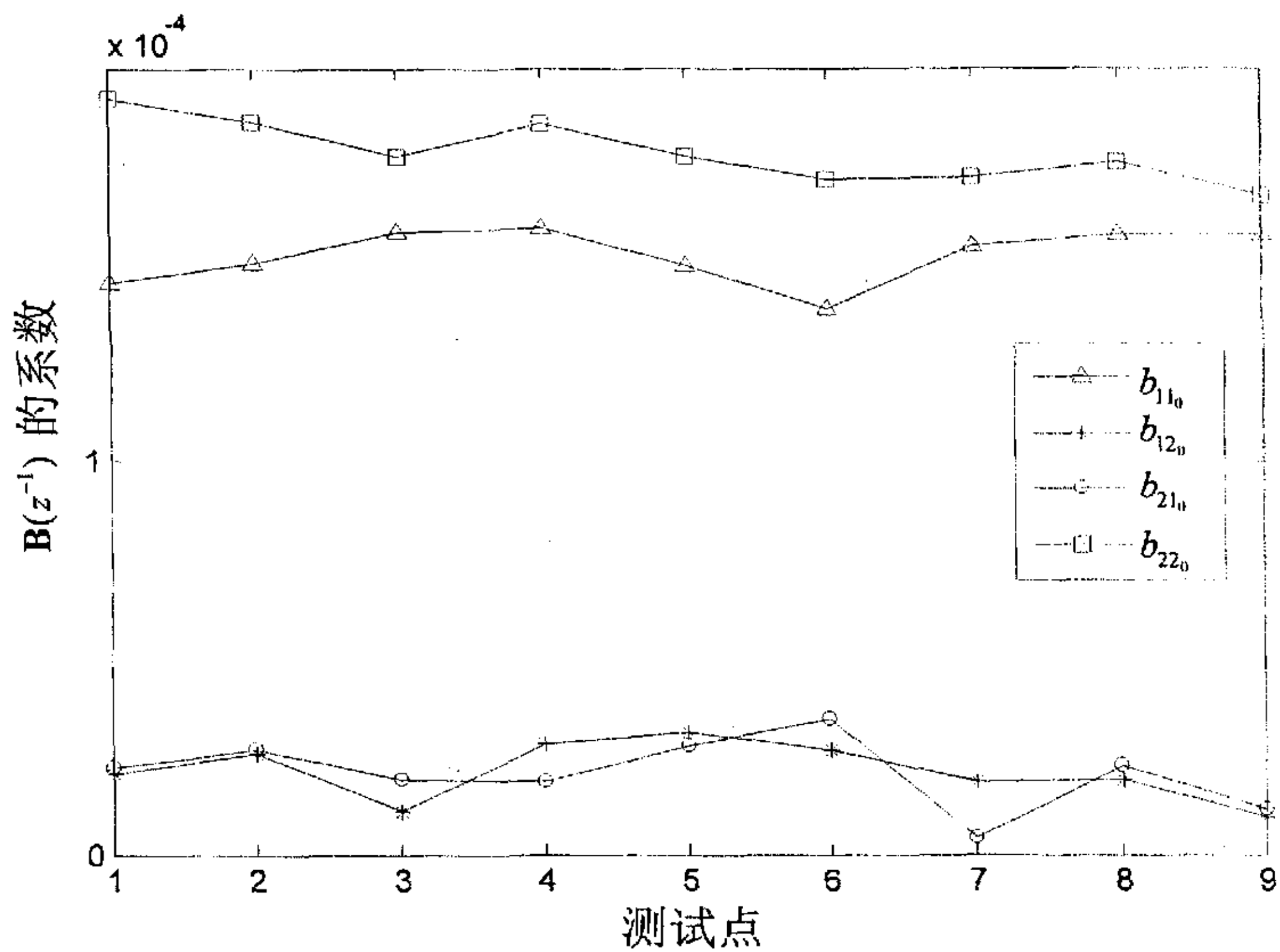


图 6.31 $B(z^{-1})$ 的系数 b_{11_0} , b_{12_0} , b_{21_0} 和 b_{22_0}

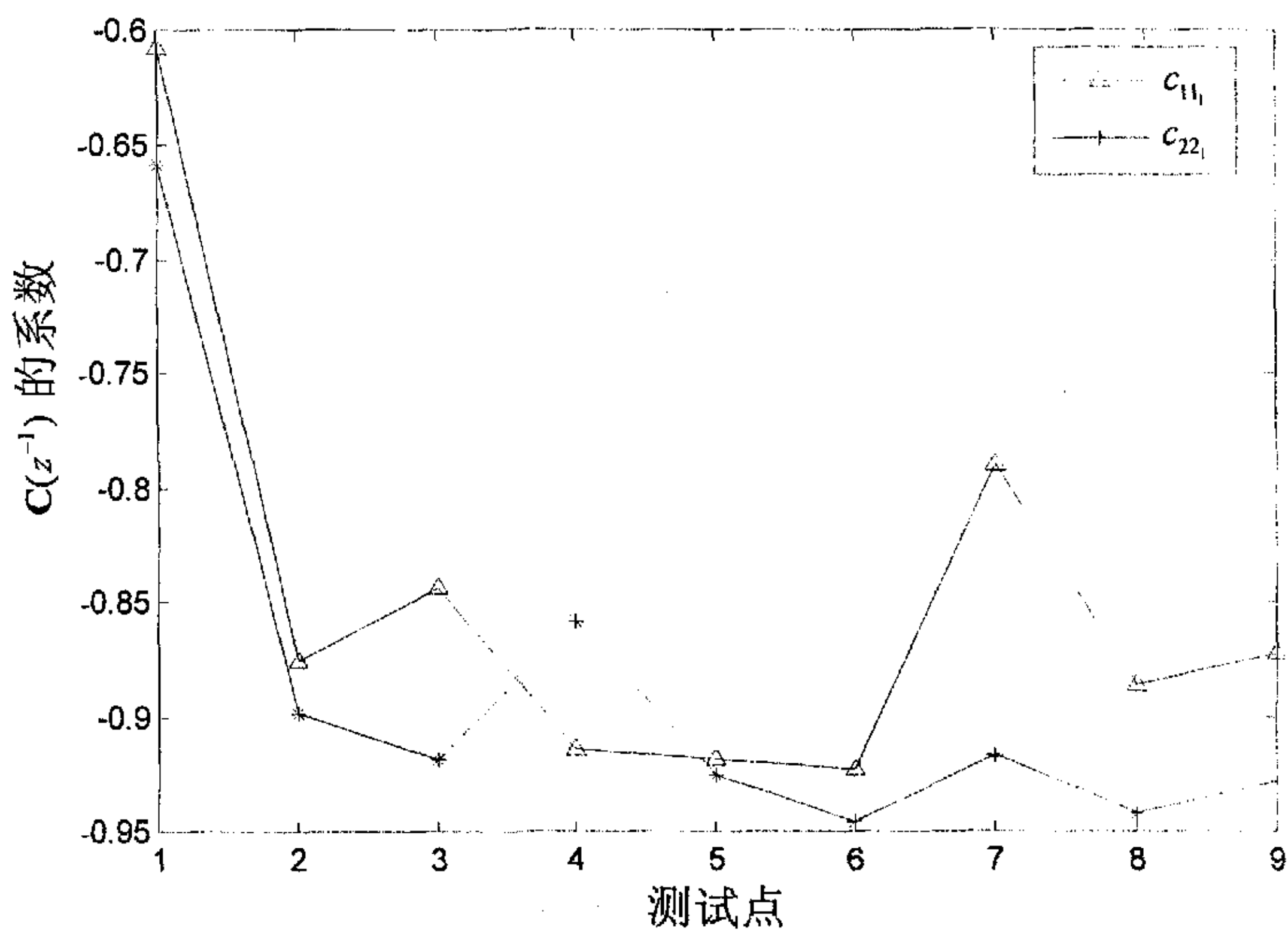


图 6.32 $C(z^{-1})$ 的系数 c_{11_1} 和 c_{22_1}

表 6.11 CARIMA 模型参数值表(一)

序号	\tilde{a}_{11_1}	\tilde{a}_{11_2}	b_{11_0}	b_{12_0}	c_{11_1}
1	-1.9849	0.99687	0.00014507	2.0717e-005	-0.60735
2	-1.959	0.96427	0.00015022	2.562e-005	-0.87561
3	-1.9566	0.96412	0.00015817	1.1337e-005	-0.84338
4	-1.9516	0.95702	0.00015892	2.7792e-005	-0.914
5	-1.952	0.95671	0.00014939	3.108e-005	-0.91862
6	-1.9564	0.96075	0.00013884	2.6636e-005	-0.92262
7	-1.9458	0.95529	0.00015461	1.8676e-005	-0.78979
8	-1.9509	0.9558	0.00015737	1.9312e-005	-0.88652
9	-1.944	0.95057	0.00015762	9.5024e-006	-0.87195

表 6.12 CARIMA 模型参数值表(二)

序号	\tilde{a}_{22_1}	\tilde{a}_{22_2}	b_{21_0}	b_{22_0}	c_{22_1}
1	-1.9665	0.97632	2.1904e-005	0.00019227	-0.65851
2	-1.9436	0.94672	2.6607e-005	0.00018659	-0.89854
3	-1.93	0.93443	1.9144e-005	0.00017753	-0.91846
4	-1.978	0.98187	1.8793e-005	0.0001861	-0.85874
5	-1.9462	0.94911	2.7386e-005	0.00017738	-0.92525
6	-1.9457	0.94699	3.4635e-005	0.0001713	-0.94519
7	-1.9498	0.95483	4.2761e-006	0.00017185	-0.91614
8	-1.9381	0.93939	2.2712e-005	0.00017585	-0.94211
9	-1.9247	0.92874	1.1627e-005	0.00016729	-0.92795

6.4.3 控制器参数选择

依据以上的分析, 按照控制周期与预测控制算法参数选择的一般原则^{[11][12]}, 取控制周期为 5 秒。预测时域 $N_{1_1} = N_{1_2} = 73$, 控制时域 $N_{3_1} = N_{3_2} = 18$ 。

实验中参考轨迹取设定值的一阶光滑近似, 柔化因子 $\alpha_1 = \alpha_2 = 0.5$ 。 \mathbf{R} 与 \mathbf{Q} 为常值对角阵, 对角元素分别为 5000 和 0.01。

辨识模型如(6.6)式所示, 初值取(6.7)式中相应的系数值。辨识方法采用带遗

忘因子的最小二乘法, 遗忘因子为 0.98, 协方差矩阵初值 $P_0 = 1 \times 10^4 I$ (I 为单位矩阵)。辨识系统模型参数时, 亦分成两个子模型, 每个输出对应一个双输入单输出辨识对象。

6.4.4 BP 网络结构、样本选择、及训练

神经网络用来实现基于 BP 网络的 GPC 快速算法 MGPC-4 对多容液位系统的控制。

按照第五章的说明, 控制器的结构为:

$$\Delta u(t) = \begin{bmatrix} l_{w_r} & l_{w_y} & l_u & l_e \end{bmatrix} \begin{bmatrix} r(t) \\ Y_p \\ U_p \\ E_p \end{bmatrix} \quad (6.8)$$

根据对象模型(6.6)式, 控制器(6.8)式的控制系数为 2×10 的矩阵, 即系数为 l_{ij} ($i=1,2, j=1,\dots,10$)。故 BP 网络的输出层节点应是 20 个。

由于 $A(z^{-1})$ 和 $\tilde{A}(z^{-1})$ 中的系数有对应关系, 故只选择 CARIMA 模型中 a_{11_1} 、 b_{11_0} 、 b_{12_0} 、 c_{11_1} 、 a_{22_1} 、 b_{21_0} 、 b_{22_0} 和 c_{22_1} 等 8 个模型参数作为 BP 网络的输入层节点。模型参数样本的取值区间如表 6.13 所示。

输入层样本获取方式为, 从表中所示范围内将每一个输入参数等距分为五个点, 然后进行全组合, 共得 390625 组输入样本。输出层样本按第四章公式(4.15)到公式(4.20)计算。由于样本很大, 加之需要映射的多变量广义预测算法关系极其复杂, 如果用一个网络来进行映射, 则需要较长的训练时间。为了加快实验进度, 采用了 20 个小网络来代替单个网络, 每一个小网络只映射一个控制器系数, 小网络编号与其所映射的控制器系数对应表见表 6.14。子网络的结构均为: 8—10—1, 训练目标为 0.00001, 每个子网络的训练步数如表 6.15 所示。BP 训练过程如图 6.33 所示。图 6.33 由 20 个子图构成, 每个子图对应一个小 BP 网络的训练过程, 中间的数字是相应小神经网络的编号, 横坐标是训练的步数, 纵坐标是训练的目标。

表 6.13 模型参数样本取值区间

参数	最小值	最大值
a_{11_1}	0.955	0.965
b_{11_0}	1.3e-4	2.3e-4
b_{12_0}	0.1e-4	0.9e-4
c_{11_1}	-0.9	-0.6
a_{22_1}	0.945	0.96
b_{21_0}	0.1e-4	0.9e-4
b_{22_0}	1.6e-4	2.5e-4
c_{22_1}	-0.9	-0.65

表 6.14 小神经网络输出层节点对应的控制器系数

网络编号	1	2	3	4	5	6	7	8	9	10
控制器系数	$l_{1,1}$	$l_{1,2}$	$l_{1,3}$	$l_{1,4}$	$l_{1,5}$	$l_{1,6}$	$l_{1,7}$	$l_{1,8}$	$l_{1,9}$	$l_{1,10}$
网络编号	11	12	13	14	15	16	17	18	19	20
控制器系数	$l_{2,1}$	$l_{2,2}$	$l_{2,3}$	$l_{2,4}$	$l_{2,5}$	$l_{2,6}$	$l_{2,7}$	$l_{2,8}$	$l_{2,9}$	$l_{2,10}$

表 6.15 小神经网络的训练步数

网络编号	1	2	3	4	5	6	7	8	9	10
训练步数	282	25	316	128	58	76	84	132	135	91
网络编号	11	12	13	14	15	16	17	18	19	20
训练步数	103	617	120	34	63	78	134	151	99	135

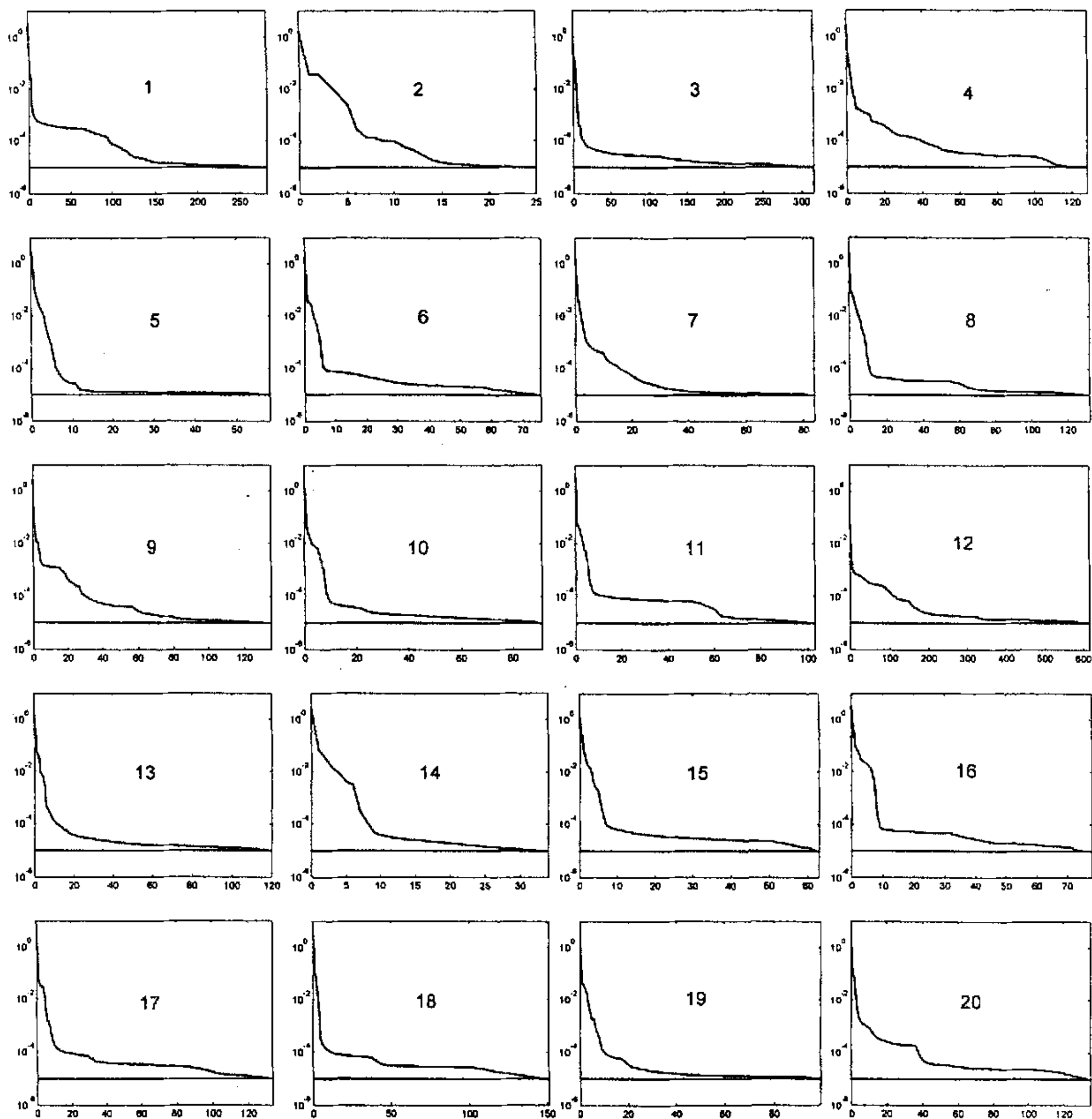


图 6.33 BP 网络训练图

6.4.5 多容液位系统控制实验

6.4.5.1 自适应模式

把实验时间分为每 300 秒钟一段, 每一组实验的前 600 秒时间段内泵 1 和泵 2 输入伪随机二进制序列控制信号进行模型参数辨识; 601-900 秒时间段内让系统进入稳态; 901-3000 秒时间段内系统投入自动控制。 y_1 与 y_2 设定值变化情况见表 6.16。

表 6.16 y_1 与 y_2 的设定值的变化情况表

时间段	y_1 设定值	y_2 设定值
1-900	0	0
901-1200	0.26	0.28
1201-1500	0.26	0.28
1501-1800	0.29	0.25
1801-2100	0.26	0.25
2101-2400	0.29	0.25
2401-2700	0.29	0.28
2701-3000	0.29	0.25

实验中，为了不使泵发生倒灌，控制信号的最低值限制在 50。当控制信号的幅值小于 50 或大于 100 时，采用“cut-off”方法，直接取其最近的边界值。

实验结果如图 6.34 到图 6.51 所示。图 6.34 到图 6.37 中实线、点线分别为系统输出 y_1 与 y_2 ，虚线、点划线分别为 y_1 与 y_2 的设定值。图 6.38 到图 6.41 中的实线和虚线分别系统的输入 u_1 和 u_2 。图 6.42 到图 6.51 中实线、虚线、点线和点划线分别为 MGPC-1、MGPC-2、MGPC-3 和 MGPC-4 的模型参数在线辨识值，下三角形线和上三角形线分别为 MGPC-4 中相应模型参数在训练神经网络时输入样本点取值的上下界。

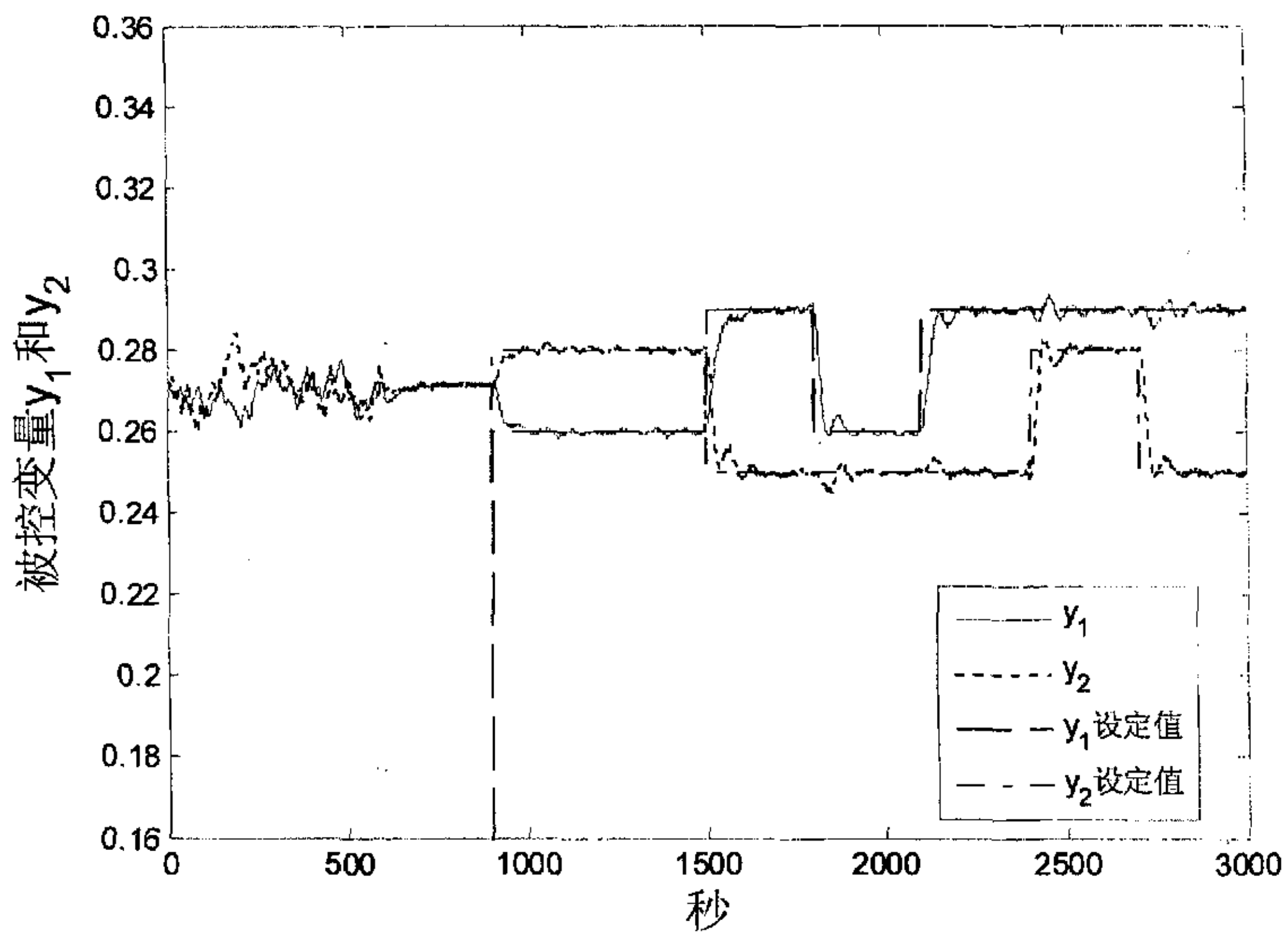


图 6.34 MGPC-1 被控变量 y_1 和 y_2

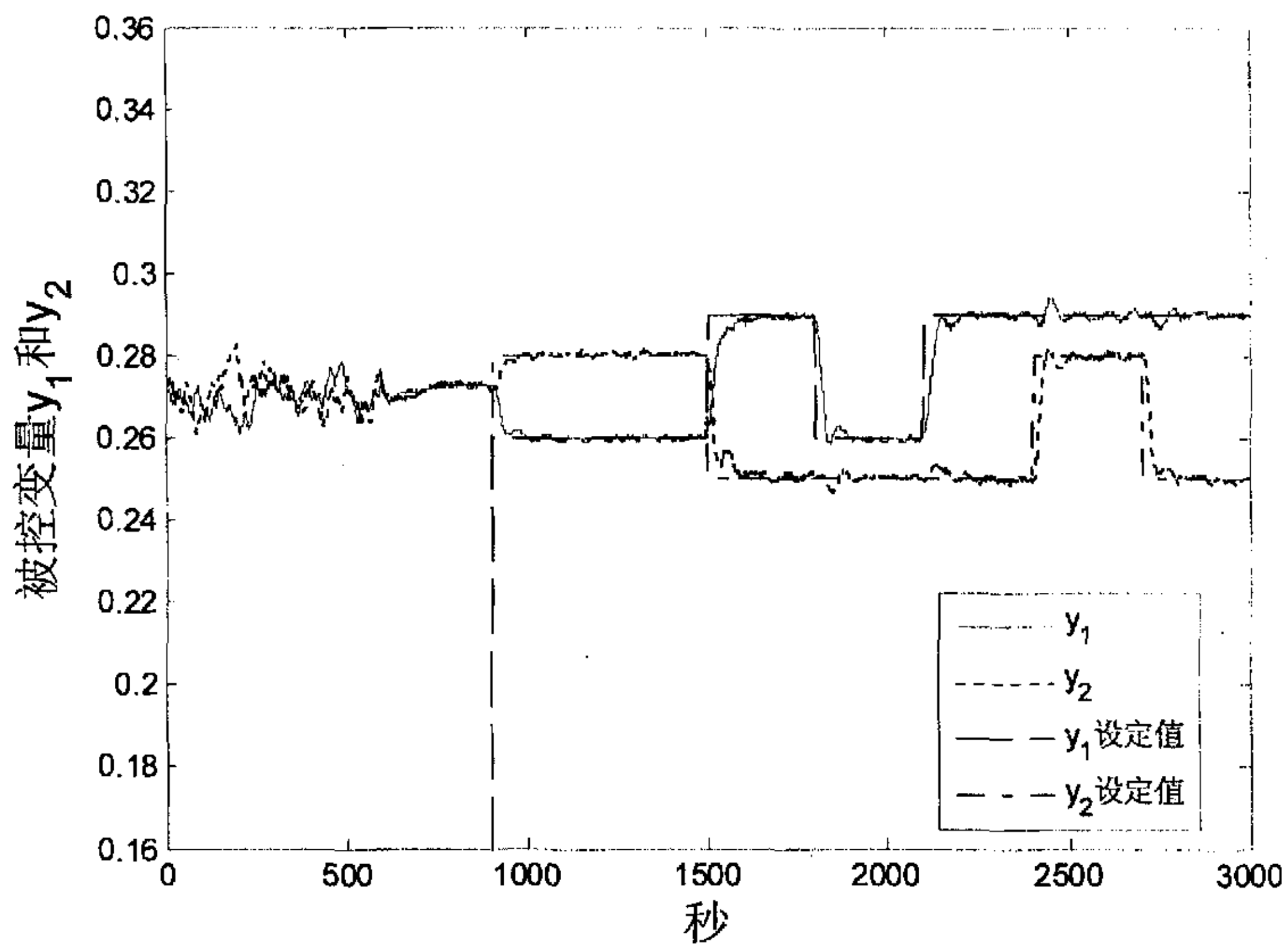
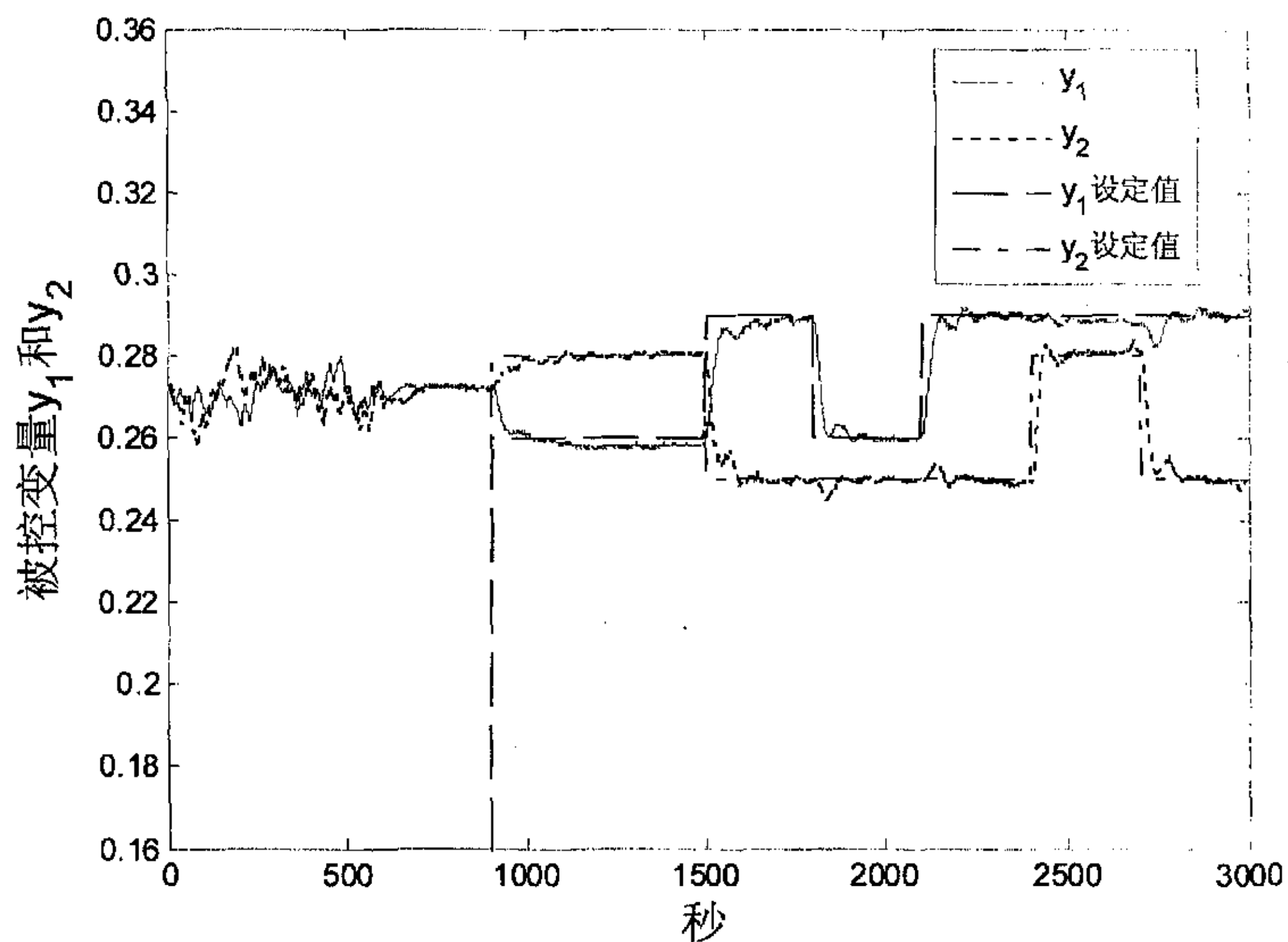
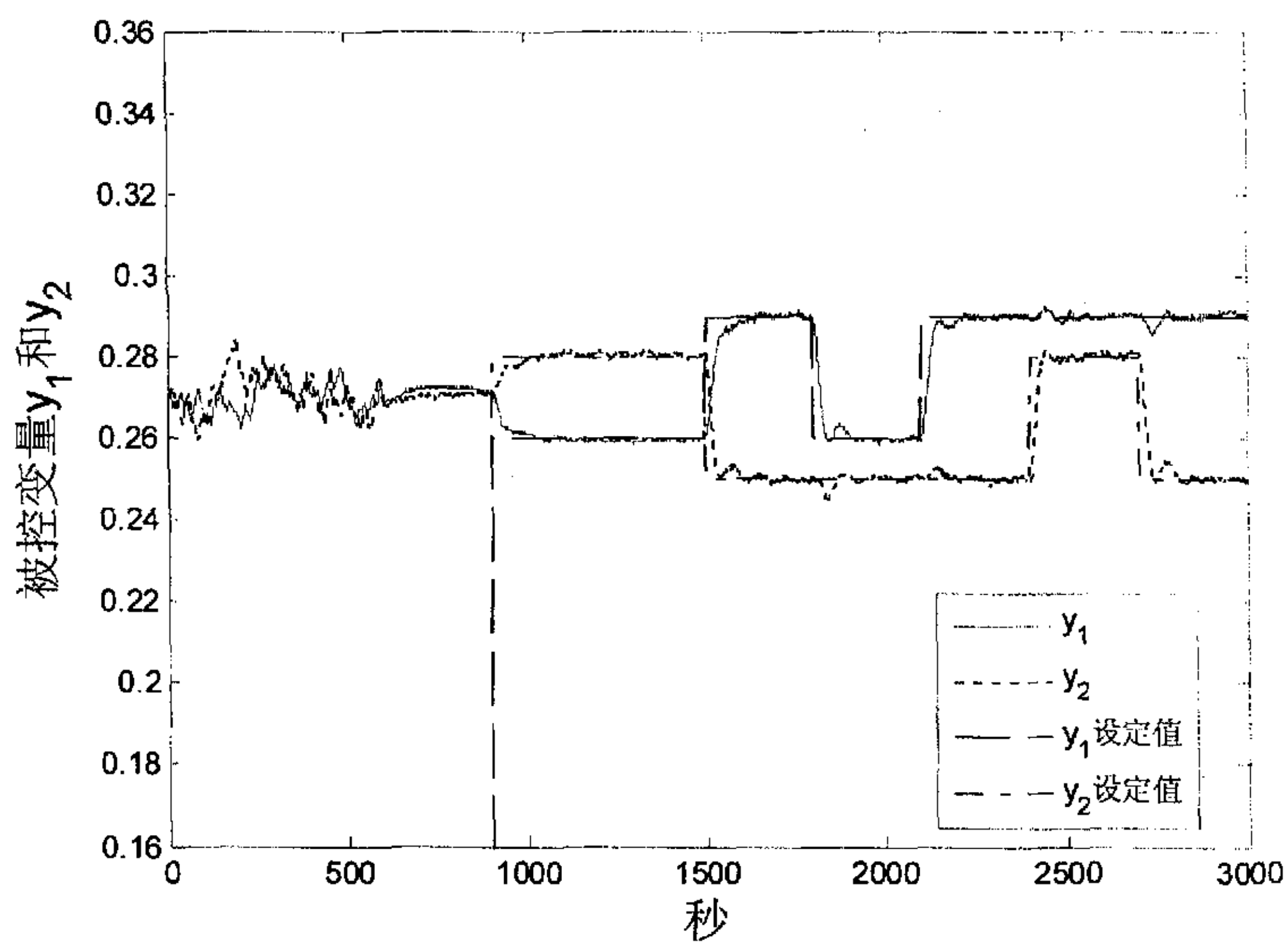


图 6.35 MGPC-2 被控变量 y_1 和 y_2



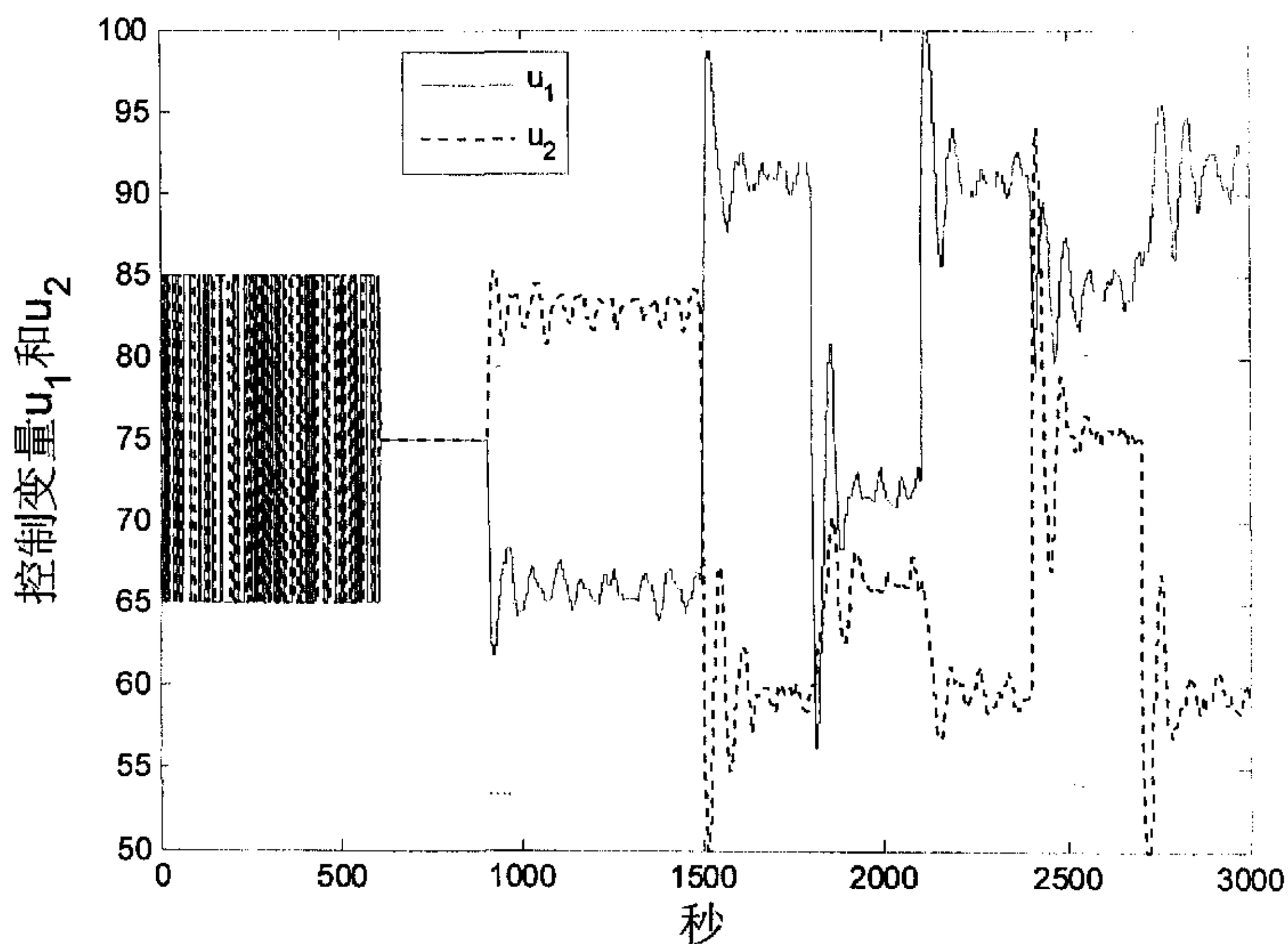


图 6.38 MGPC-1 控制变量 u_1 和 u_2

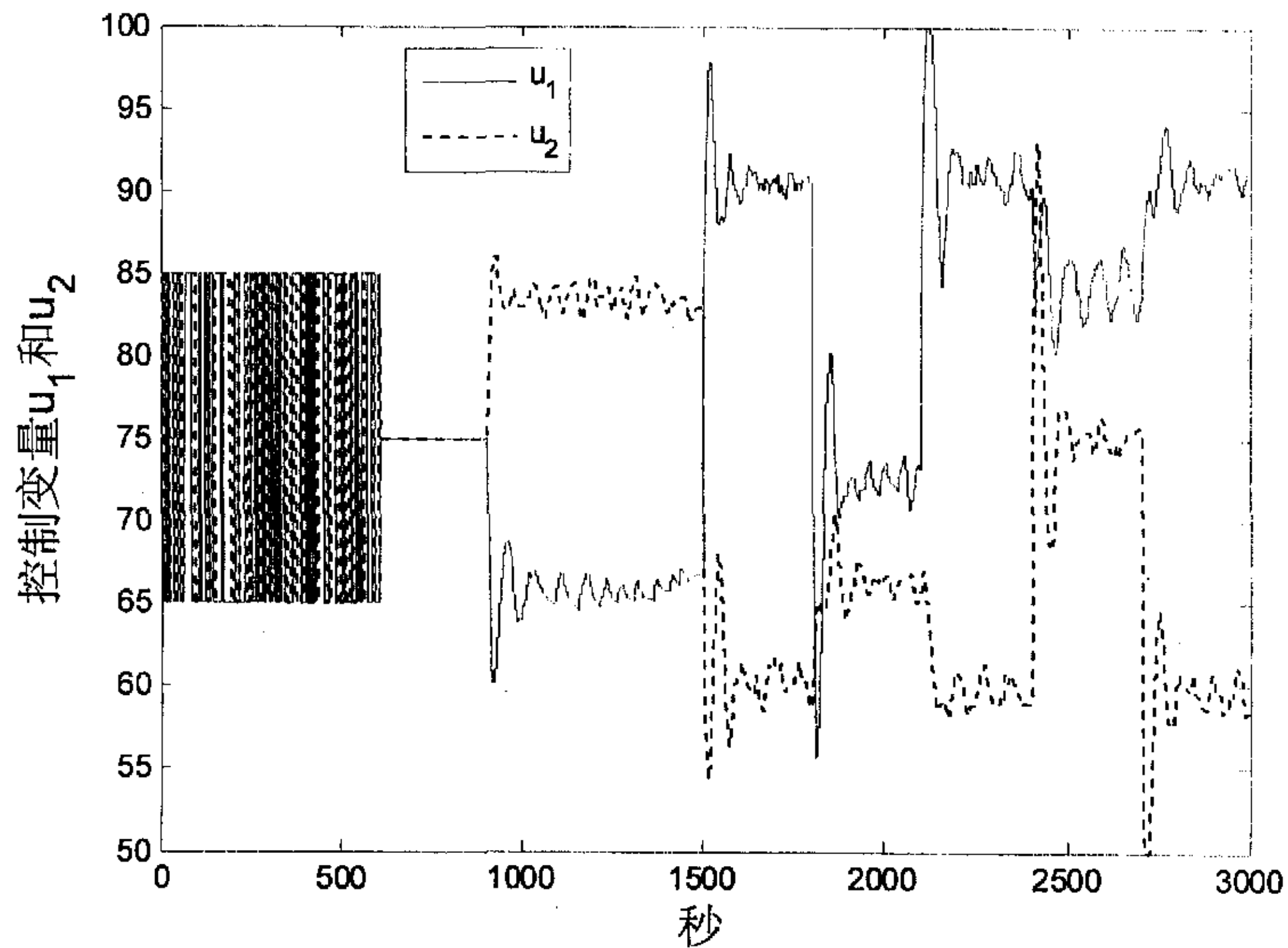


图 6.39 MGPC-2 控制变量 u_1 和 u_2

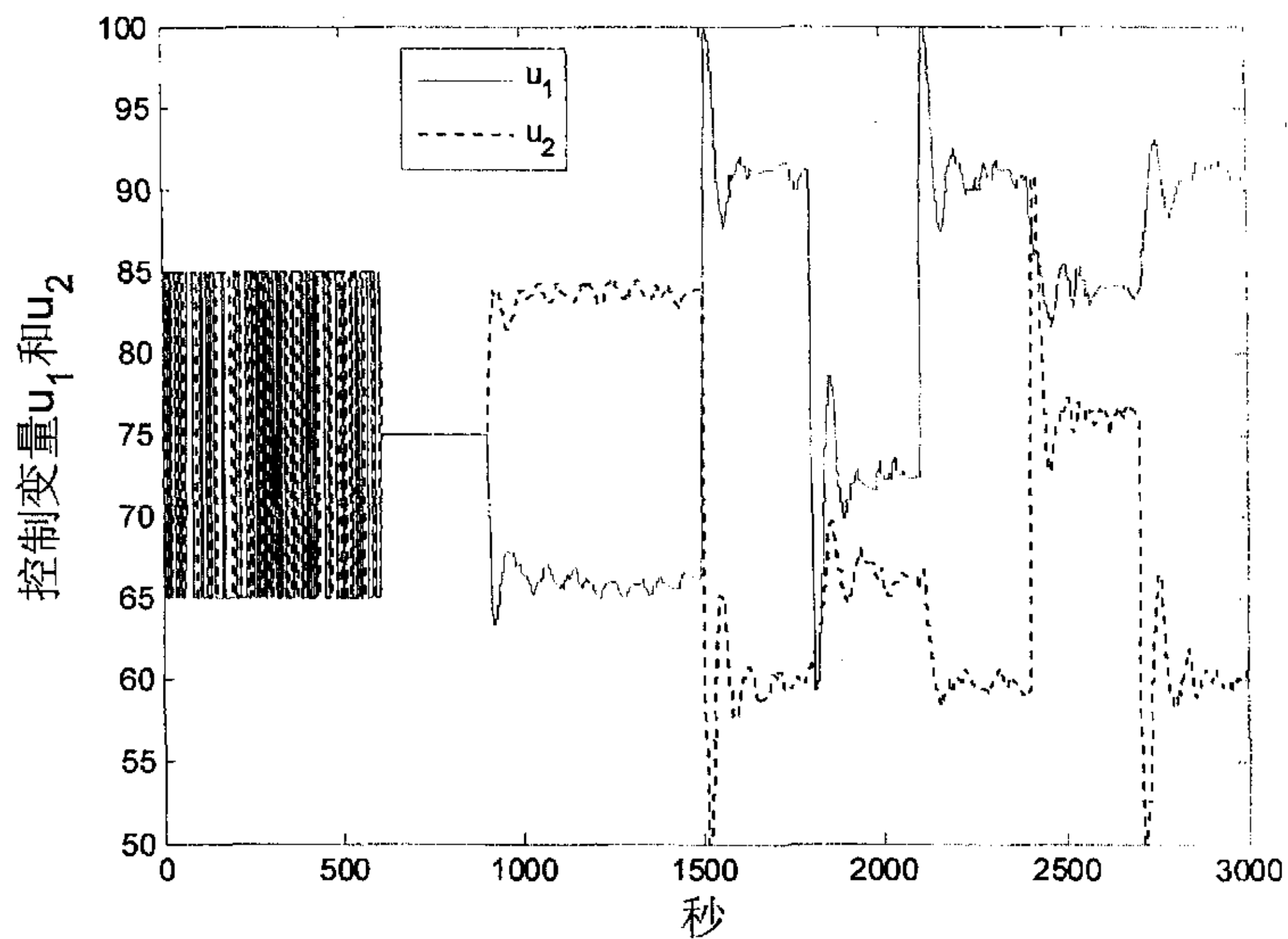


图 6.40 MGPC-3 控制变量 u_1 和 u_2

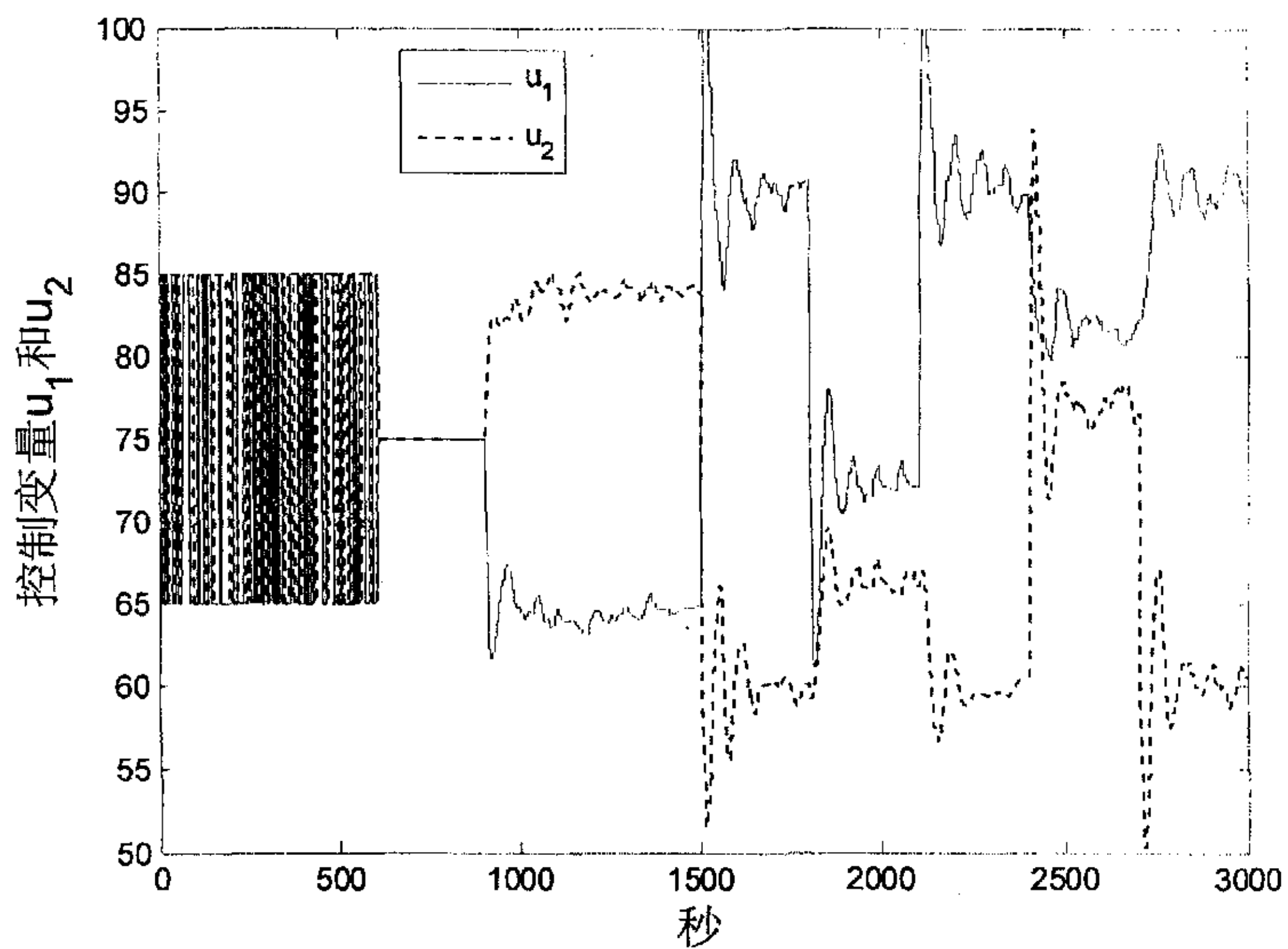


图 6.41 MGPC-4 控制变量 u_1 和 u_2

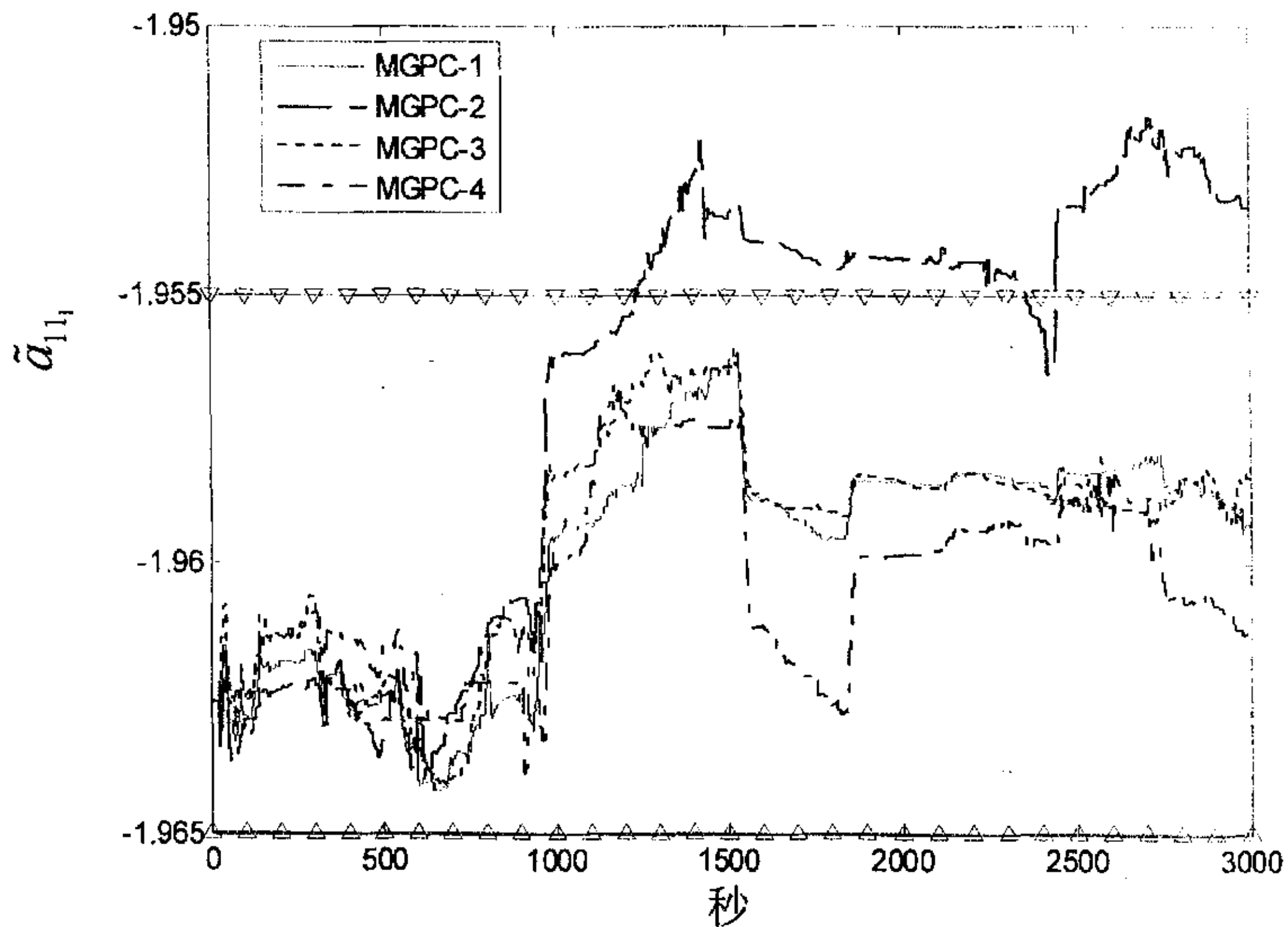


图 6.42 模型参数辨识值 \tilde{a}_{11_1}

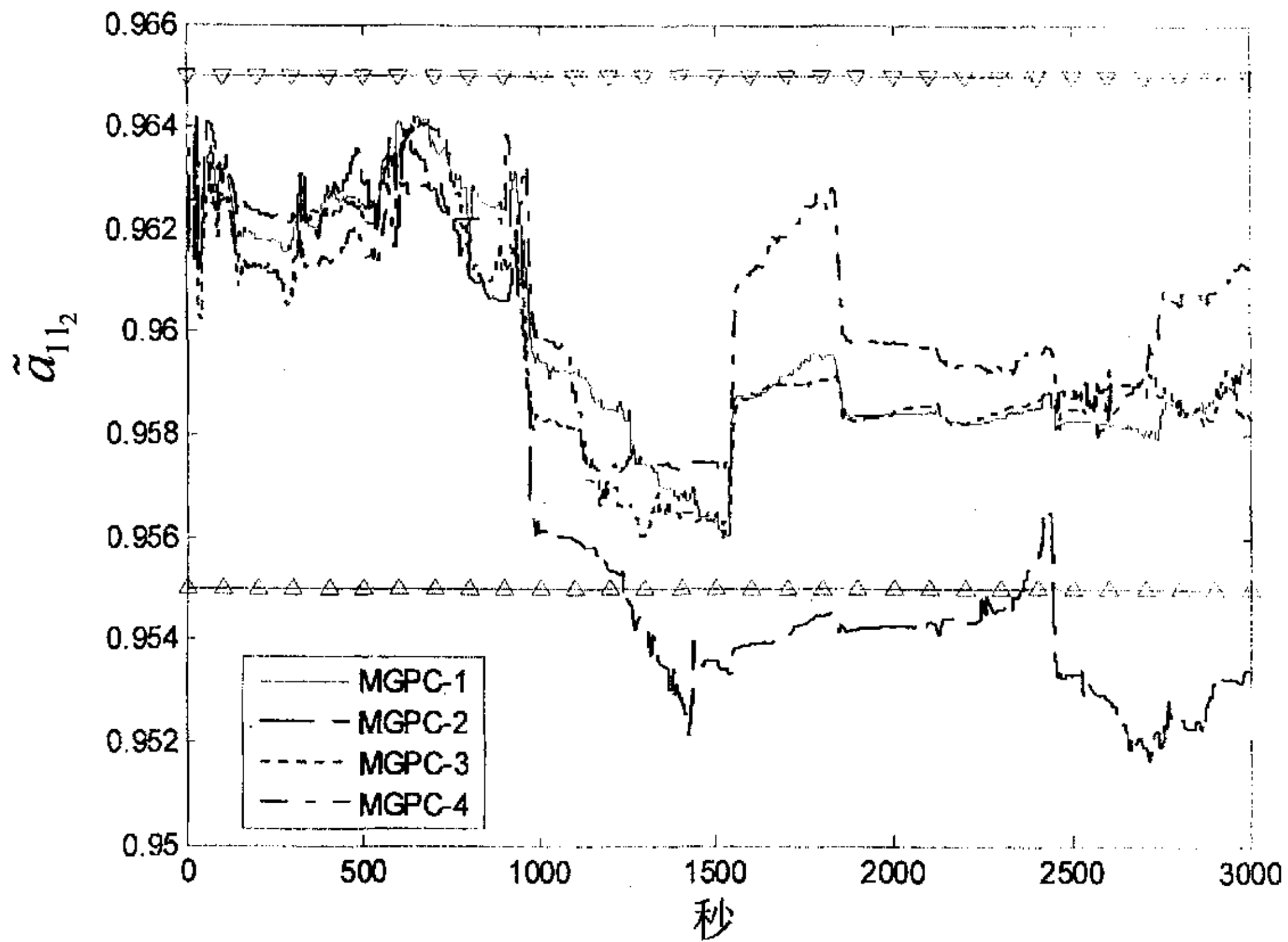


图 6.43 模型参数辨识值 \tilde{a}_{11_2}

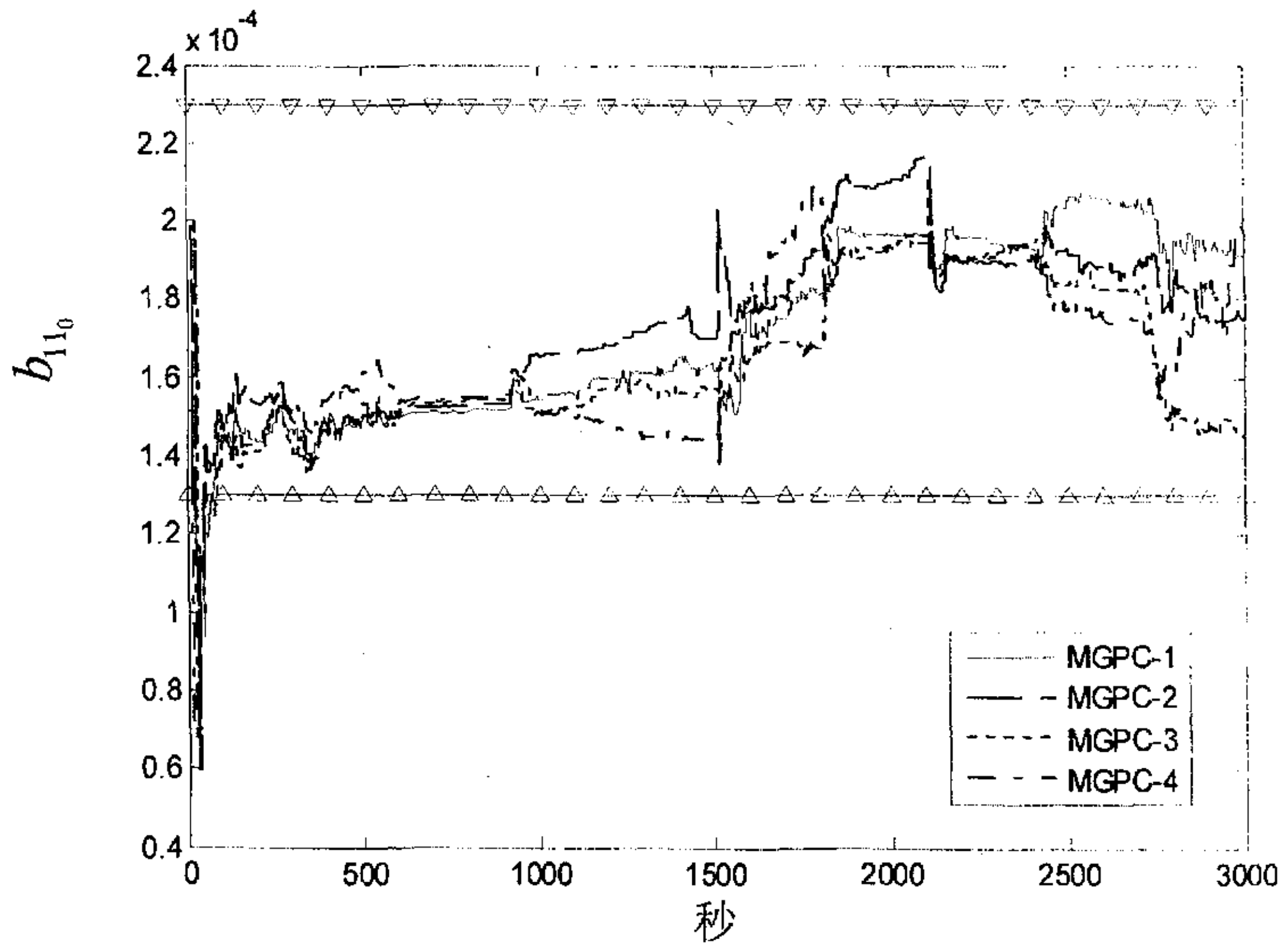


图 6.44 模型参数辨识值 b_{11_0}

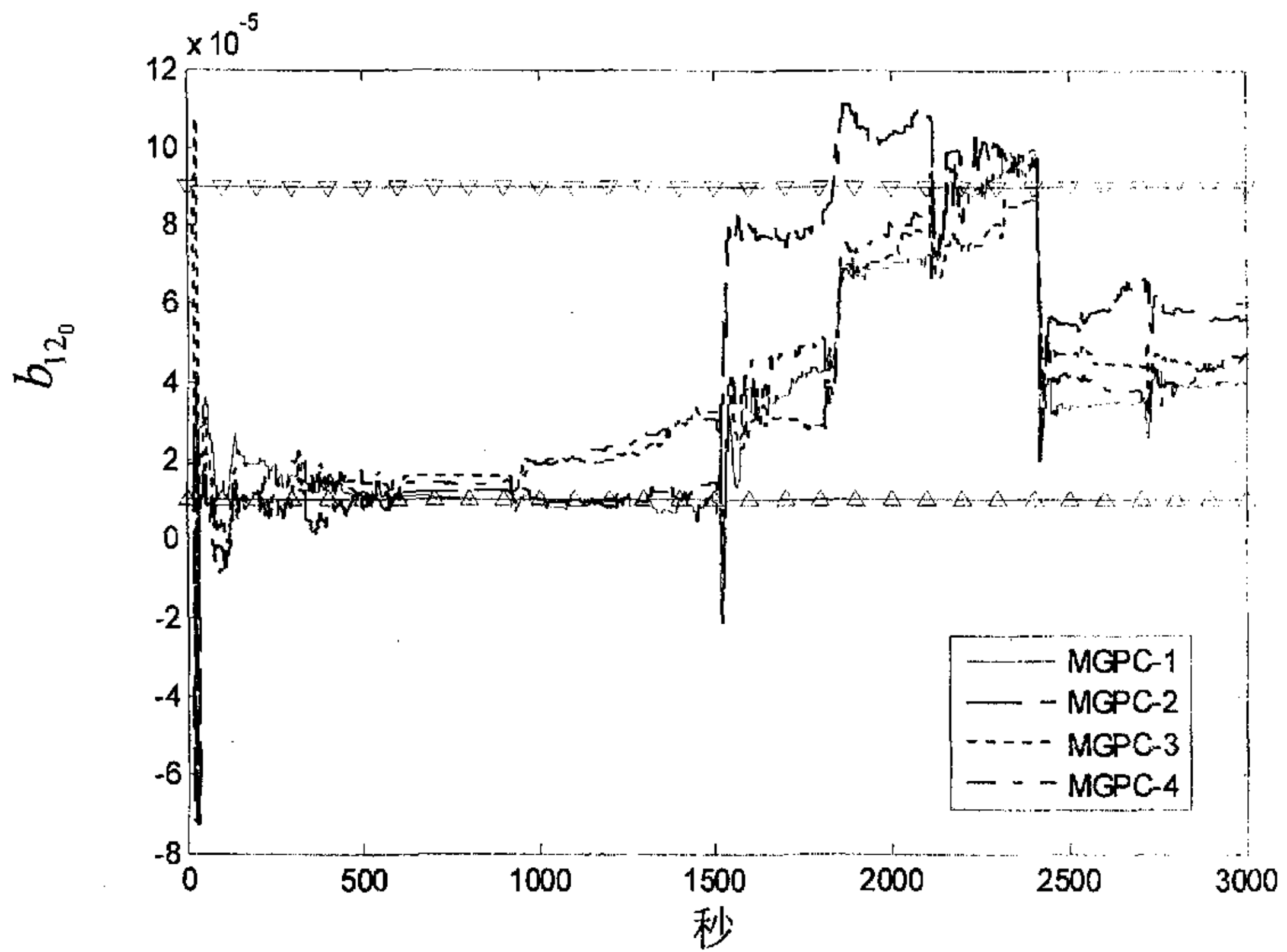
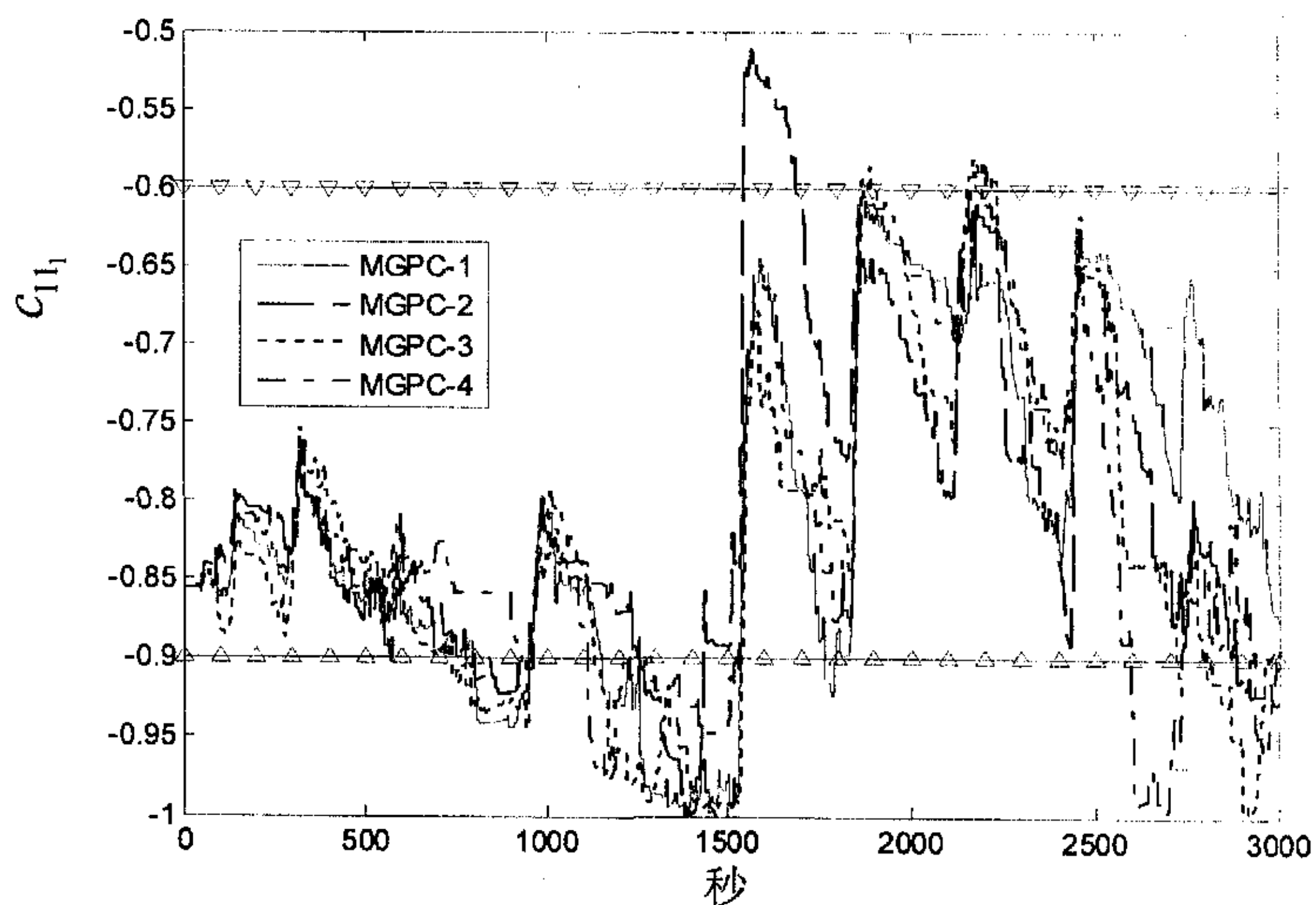
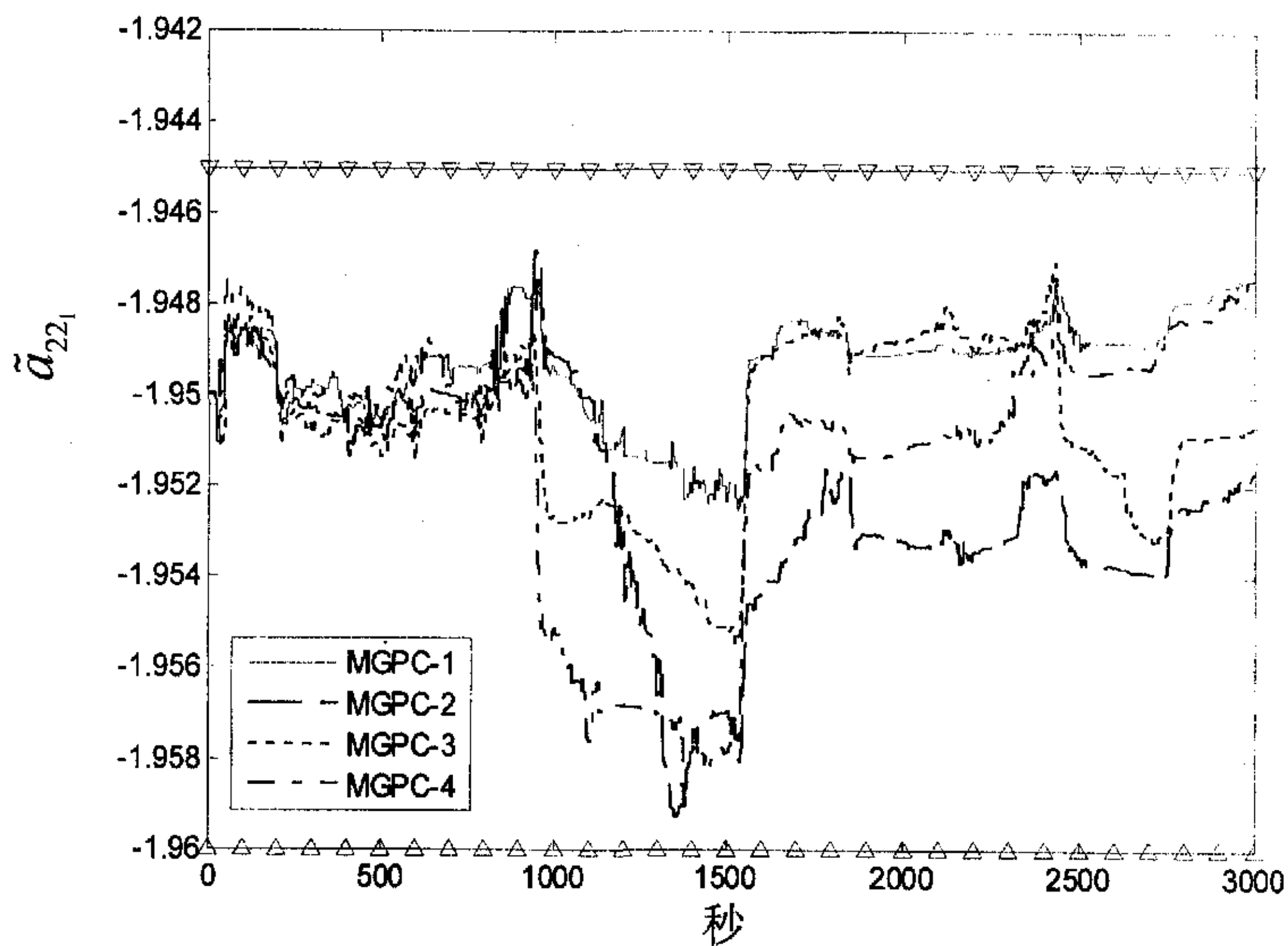


图 6.45 模型参数辨识值 b_{12_0}

图 6.46 模型参数辨识值 c_{11} 图 6.47 模型参数辨识值 a_{22}

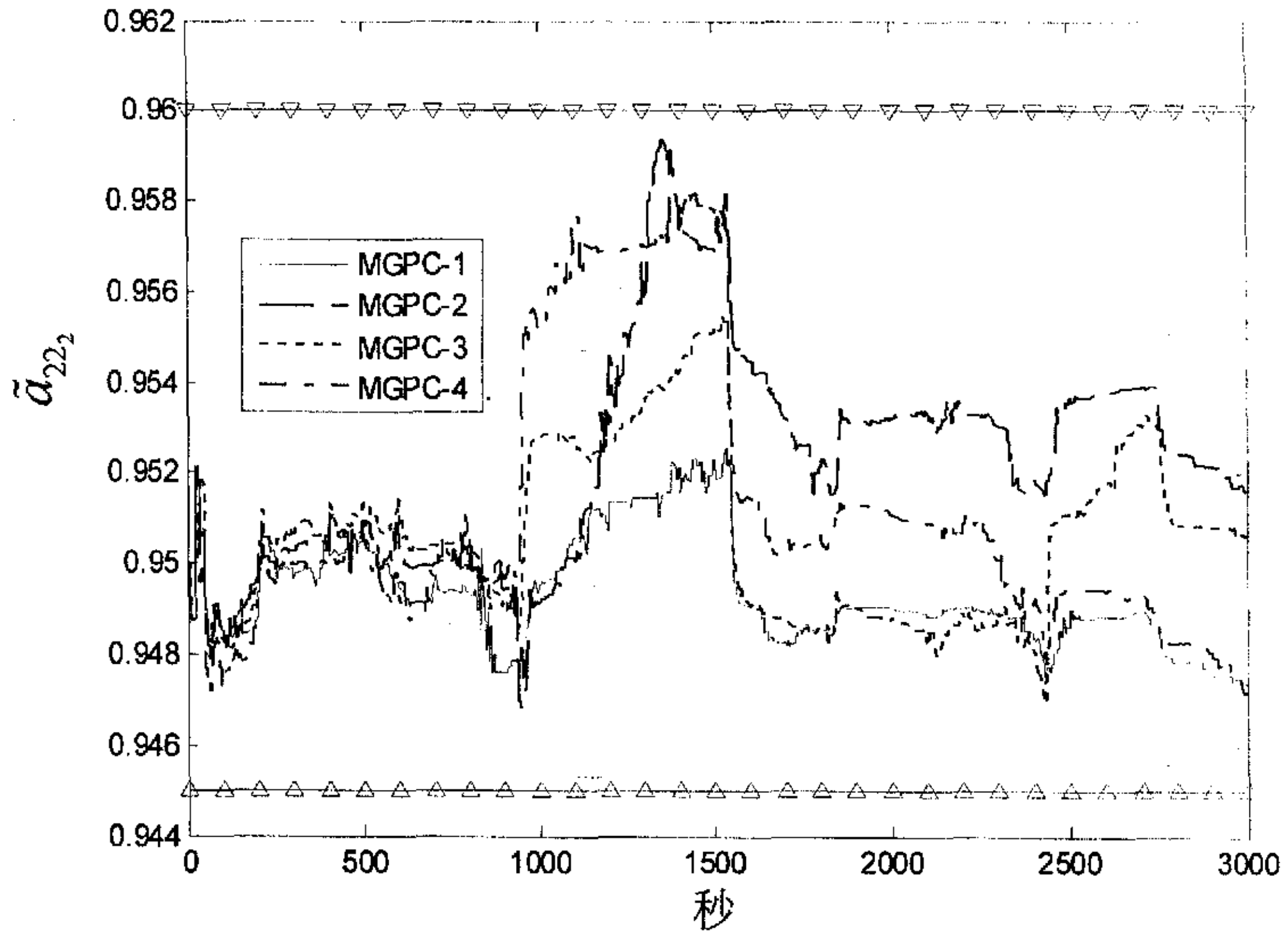


图 6.48 模型参数辨识值 \tilde{a}_{22_2}

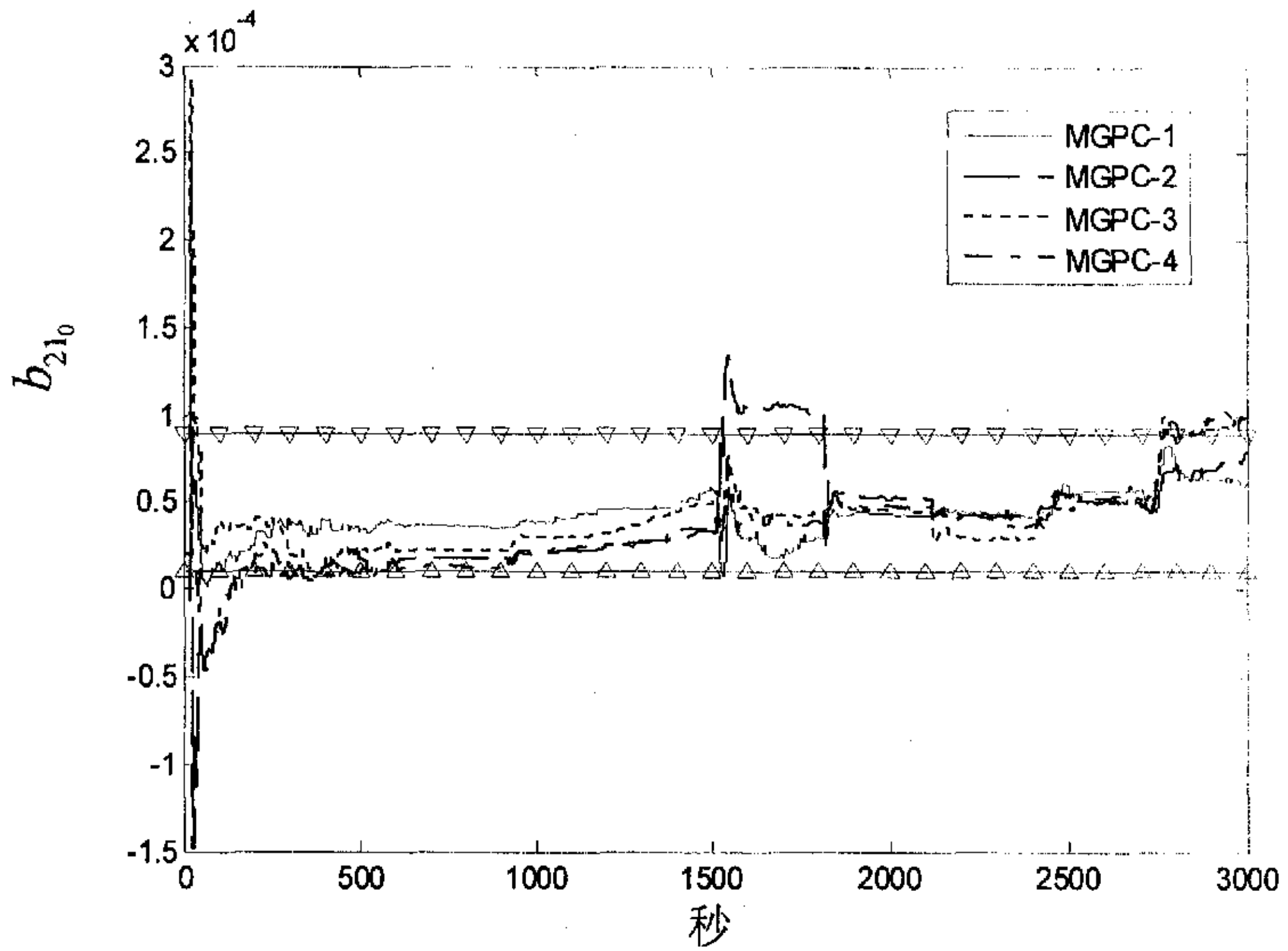


图 6.49 模型参数辨识值 b_{21_0}

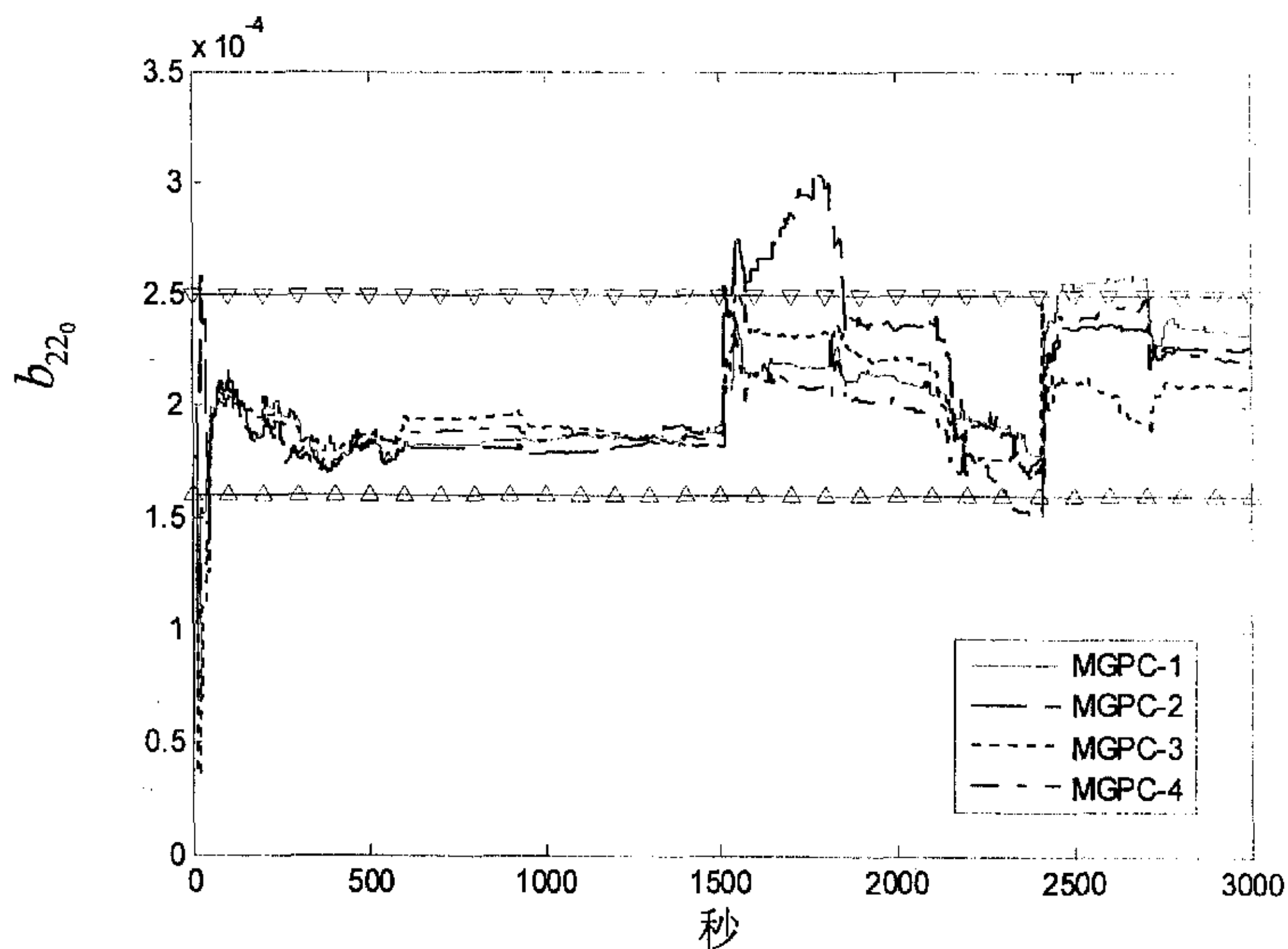


图 6.50 模型参数辨识值 b_{22_0}

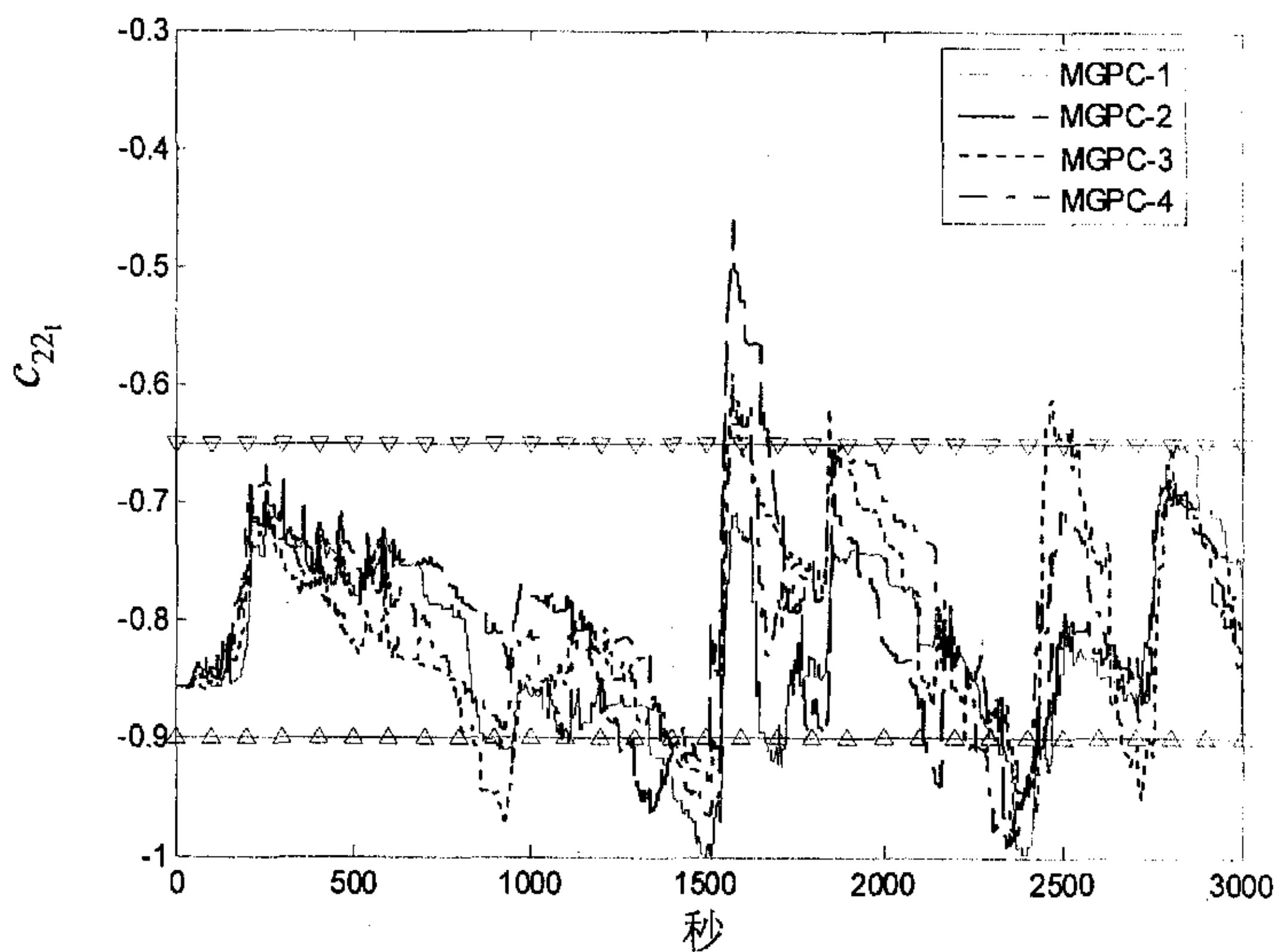


图 6.51 模型参数辨识值 c_{22_1}

6.4.5.2 非自适应模式

在非自适应模式时, 设定值变化情况与自适应模式一样, 见表 6.16。非自适应模式时, 广义预测控制器依据模型(6.7)设计。控制结果如图 6.52 到图 6.59 所示。

为了与自适应模式时的控制结果进行对照, 整个物理过程力求保持与自适应控制模式时的情形一致, 辨识器仍然在线运行, 只不过控制器不再按辨识结果在线进行校正了。模型参数的实时辨识结果不再列出。

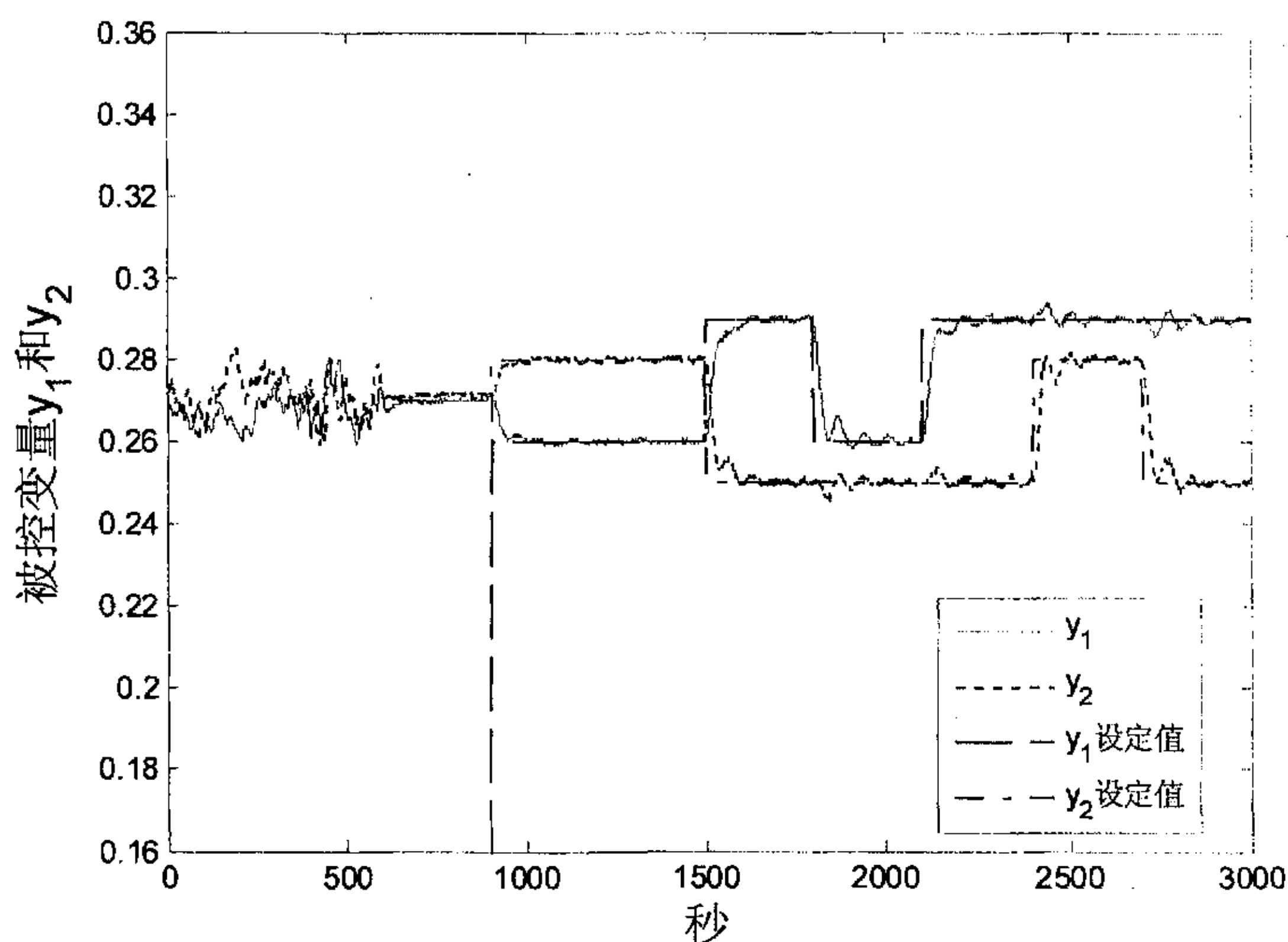
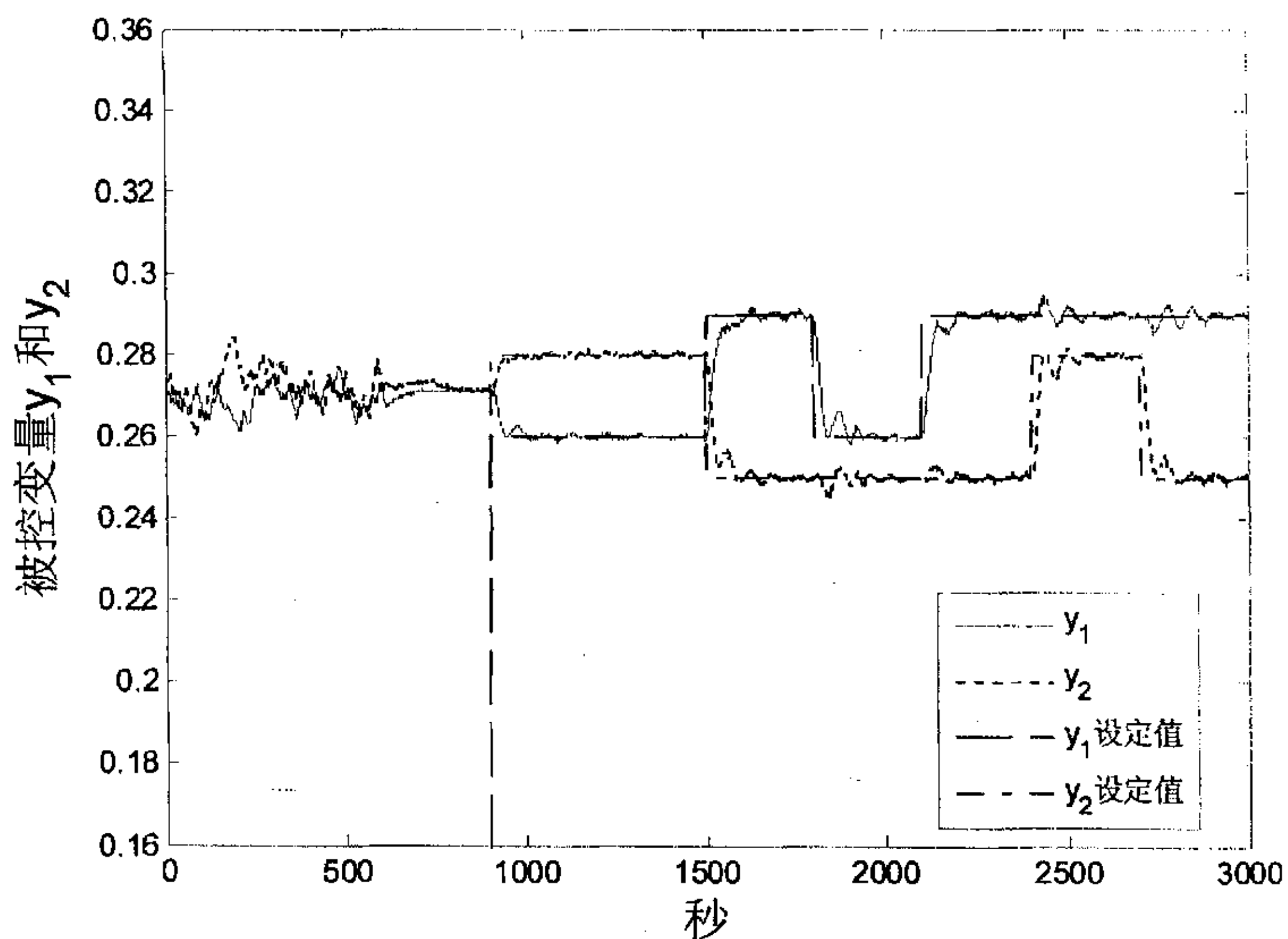
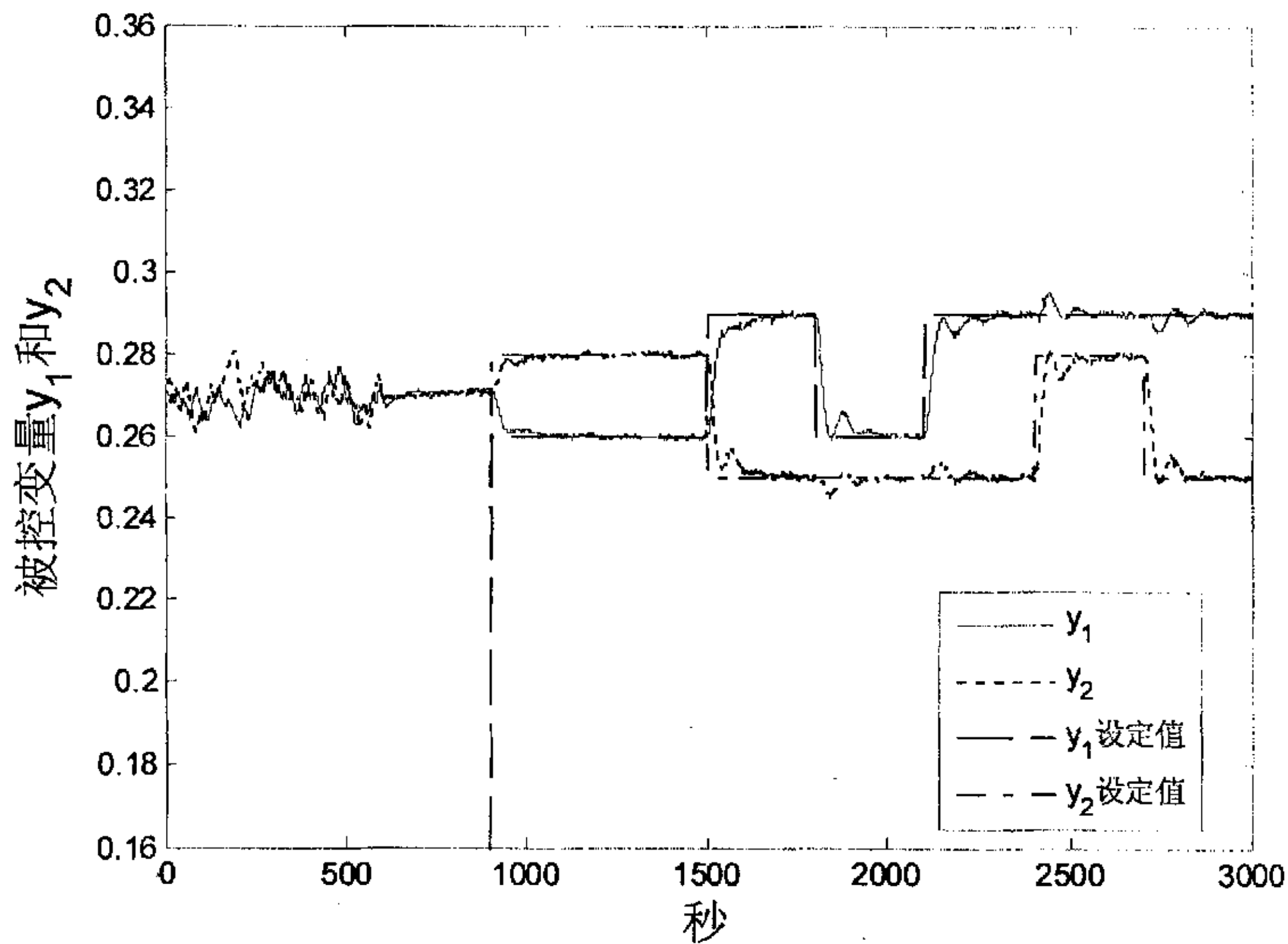


图 6.52 MGPC-1 被控变量 y_1 和 y_2

图 6.53 MGPC-2 被控变量 y_1 和 y_2 图 6.54 MGPC-3 被控变量 y_1 和 y_2

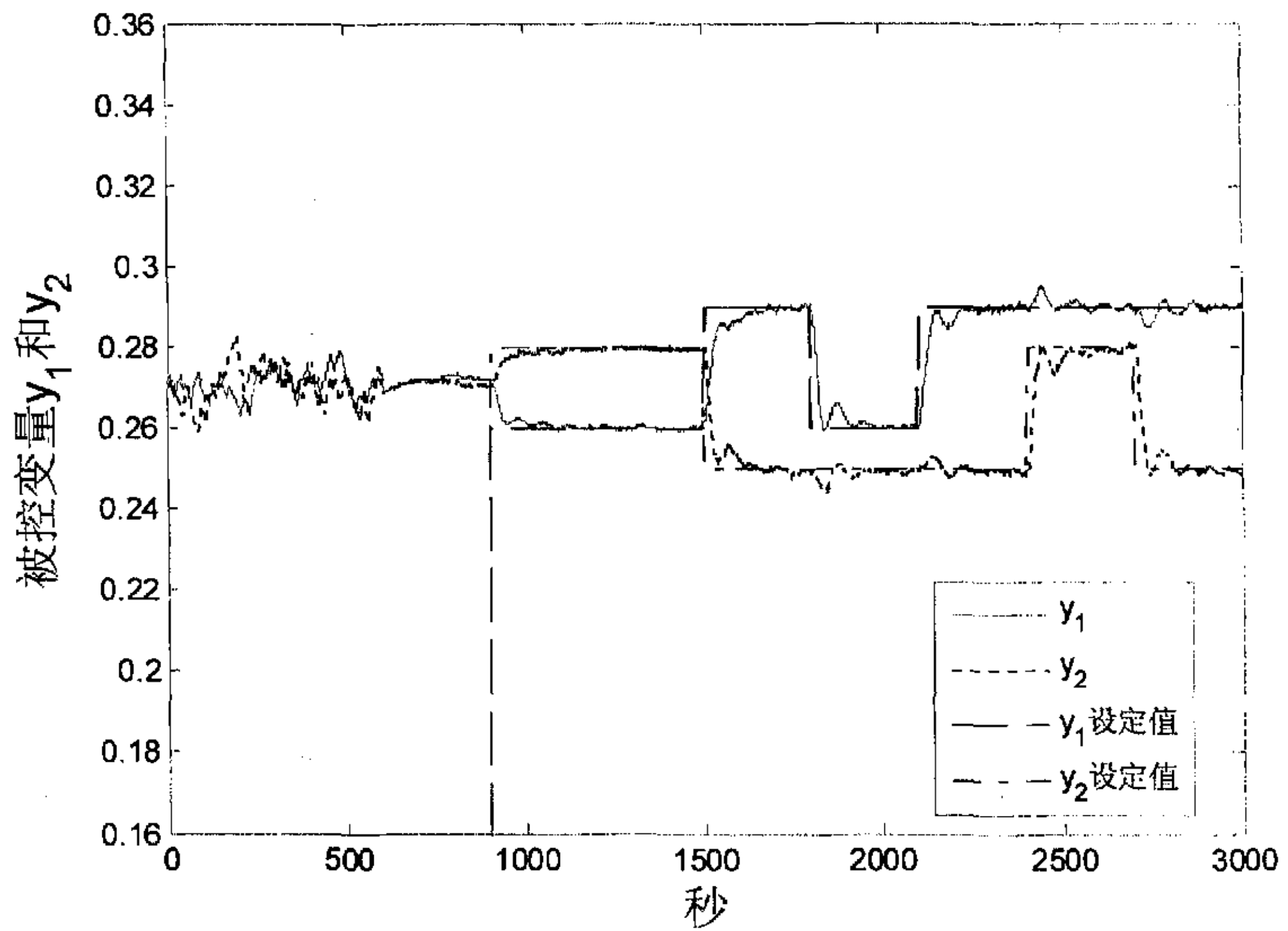


图 6.55 MGPC-4 被控变量 y_1 和 y_2

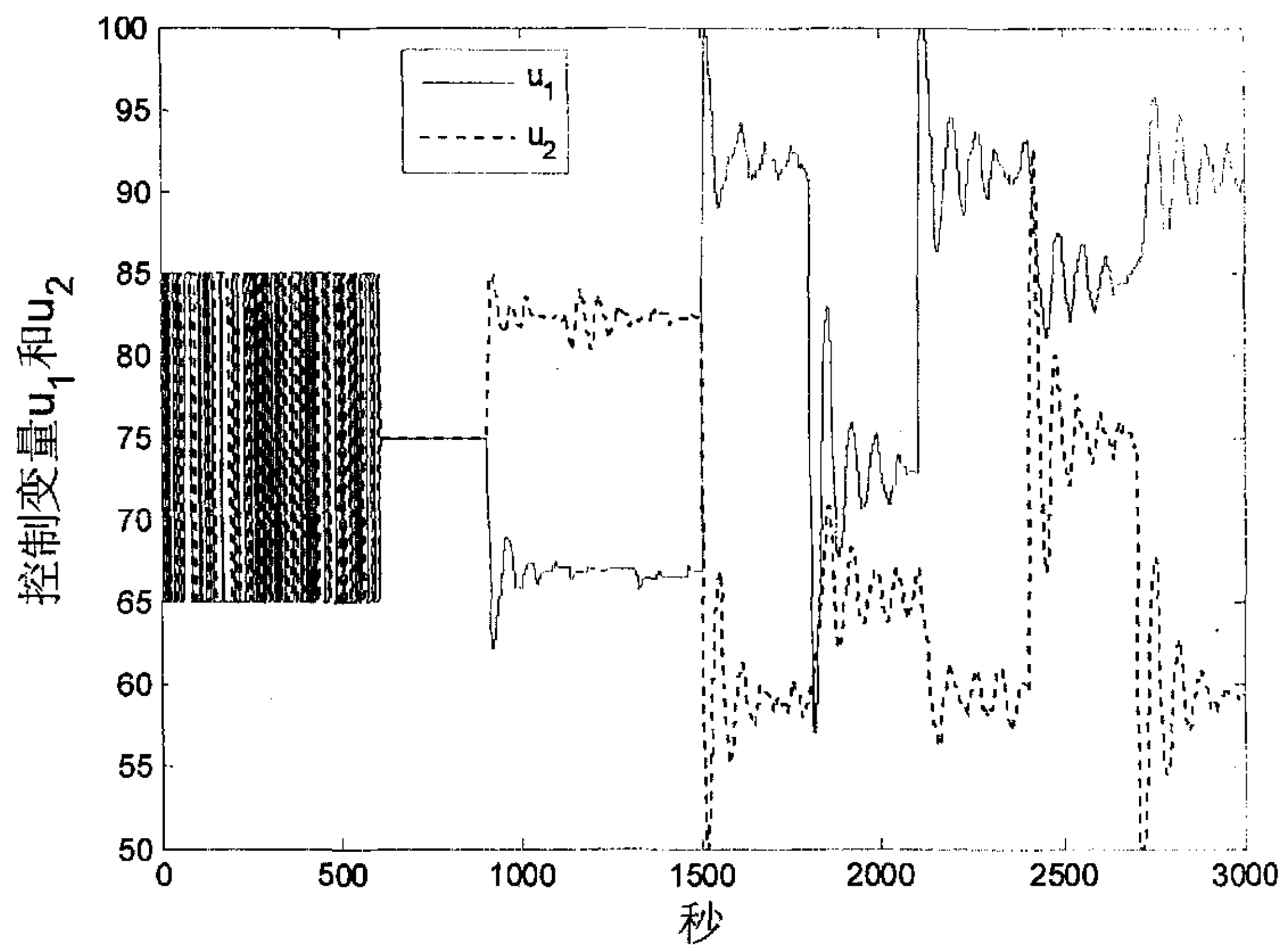


图 6.56 MGPC-1 控制变量 u_1 和 u_2

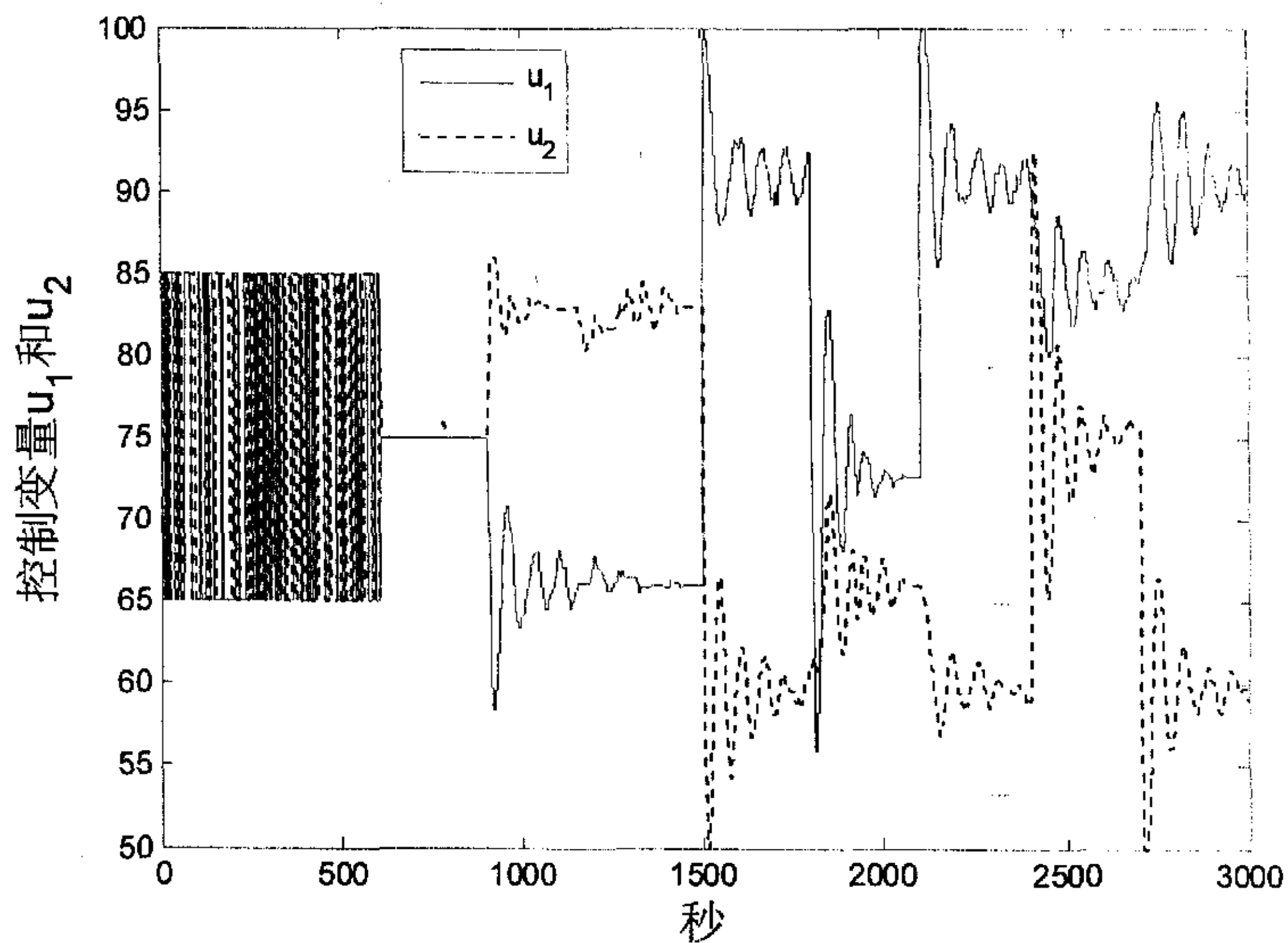


图 6.57 MGPC-2 控制变量 u_1 和 u_2

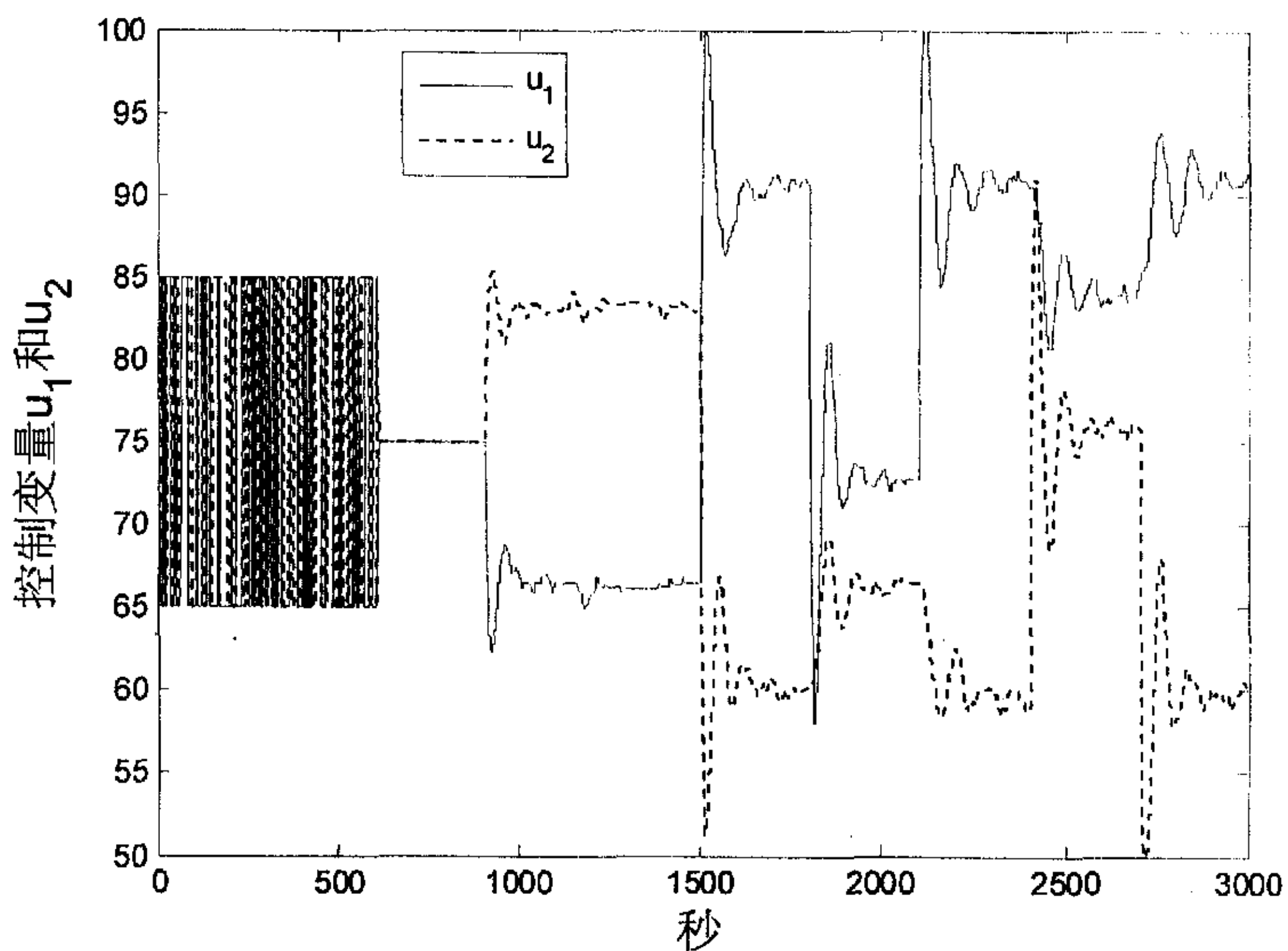
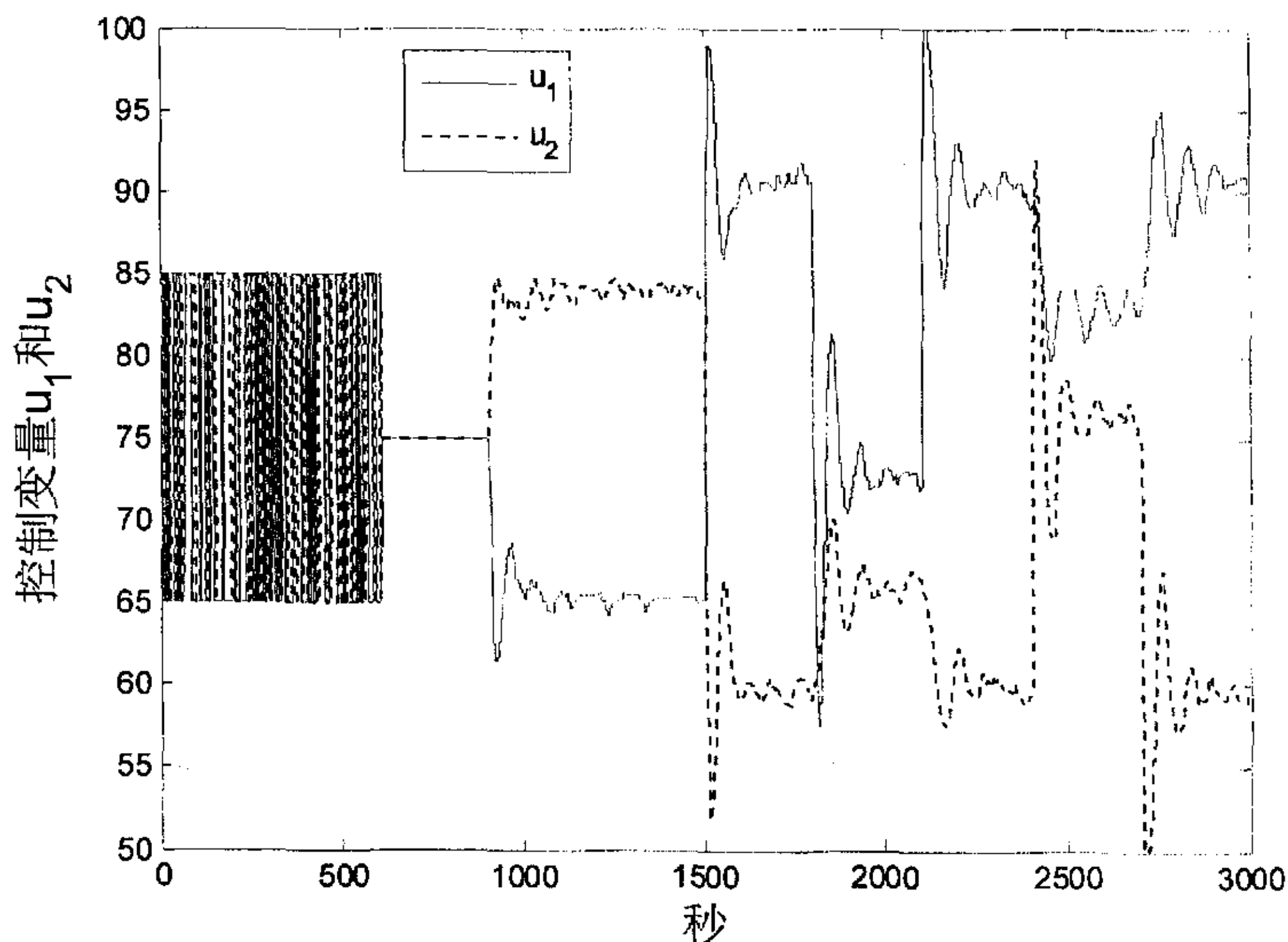


图 6.58 MGPC-3 控制变量 u_1 和 u_2

图 6.59 MGPC-4 控制变量 u_1 和 u_2

6.4.5 多容液位系统控制实验结果分析

多容液位控制实验中各算法所经历的物理过程基本相同, 图 6.42 到图 6.51 的模型参数实时辨识曲线也具有相同的变化趋势, 表明实验结果具有可比性。

图 6.42 到图 6.51 的实时辨识曲线表明, 当系统从一个平衡点变到另一个平衡点时, 模型参数也随之发生变化, 表明系统在不同工作点的特性不一样, 这是由于对象的非线性引起的。首先变频泵在不同的频率下扬程呈非线性变化; 其次, 整个系统的液体总量不变, 当抽出的液体多时, 贮水箱中的液位就会下降, 影响泵的供水能力。这些因素和装置中的其它非线性特性一起导致了整个系统在不同工作点时, 输入输出特征不一样。

表 6.17 到表 6.20 的数据为各个控制算法实验在 1200 秒到 3000 秒之间输出跟踪设定值的误差绝对值和以及误差平方和。表中数据说明各个算法不论是自适应模式, 还是非自适应模式, 控制效果几乎相同。只有 MGPC-4 在自适应模式下, 性能相对有所下降, 输出跟踪设定值的误差绝对值和明显比其它三种方法大, 图 6.37 所示的系统输出曲线也不如其它方法。而在非自适应模式下, MGPC-4 与其它三种算法的控制效果基本相同。这可能是在训练神经网络时, 所选模型参

数区间不能完全覆盖实时控制时对象模型参数变化的范围,导致了在没有训练的区域神经网络泛化能力变差的缘故。这一点可以从图 6.42 到图 6.51 实时参数辨识结果得到证实,在控制过程中 MGPC-4 的有些参数,特别是有色噪声多项式的系数 c_{11} 和 c_{22} , 实时辨识结果有相当一部分落在所训练的区间之外。而在非自适应模式时,标称模型的参数已包含在训练的区间之内,故 BP 网络能正确地实现映射。

表 6.17 自适应模式下的输出跟踪设定值的误差绝对值和

算法	y_1	y_2	和
MGPC-1	3.5010	3.1400	6.6410
MGPC-2	3.1940	3.1070	6.3010
MGPC-3	3.3010	3.0570	6.3580
MGPC-4	4.8720	3.5140	8.3860

表 6.18 非自适应模式下的输出跟踪设定值的误差绝对值和

算法	y_1	y_2	和
MGPC-1	3.5300	3.2480	6.7780
MGPC-2	3.5810	3.2590	6.8400
MGPC-3	3.6950	3.3610	7.0560
MGPC-4	3.8160	3.7850	7.7610

表 6.19 自适应模式下的输出跟踪设定值的误差平方和

算法	y_1	y_2	和
MGPC-1	0.0488	0.0435	0.0923
MGPC-2	0.0434	0.0409	0.0843
MGPC-3	0.0480	0.0435	0.0915
MGPC-4	0.0546	0.0465	0.1012

表 6.20 非自适应模式下的输出跟踪设定值的误差平方和

算法	y_1	y_2	和
MGPC-1	0.0462	0.0418	0.0879
MGPC-2	0.0468	0.0420	0.0889
MGPC-3	0.0483	0.0441	0.0923
MGPC-4	0.0496	0.0455	0.0950

实验结果还证明了广义预测控制算法具有极强的鲁棒性。对于双输入双输出的对象，用 8 个参数的多变量 CARIMA 模型来描述，显然是严重模型失配的，但所有算法都取得了满意的控制结果。说明 GPC 算法在模型失配的情形下仍然能够获得较好的控制品质，具有良好的鲁棒性能。

6.5 小结

在单容非线性液位系统控制和多容液位系统控制中的对比实验研究表明，第二章至第五章给出的简化算法不论是在自适应模式下，还是在非自适应模式下，都是可行的，并且具有一致的控制效果。同时，实验还证明了把 CARIMA 模型中的 $A(z^{-1})$ 和 $C(z^{-1})$ 构造成对角多项式矩阵形式的建模方法在现实应用中确实是可行的。另外，根据模型的这种特殊结构，采用辨识多个多输入单输出的子模型代替辨识单个多输入多输出模型的方法也是可行的。这样就能降低多变量系统在线辨识的实现难度。

第三章算法是对第二章算法在非自适应模式下的简化，第四章算法是对第二章算法在自适应模式与非自适应模式下的全面简化，而第五章算法是通过 BP 网络的映射能力对第四章算法的进一步简化。理论推导与实验研究表明，这种简化是在线计算与算法实现形式的简化，并没有以牺牲算法的控制性能为代价。当然，只有在离线训练时输入样本的取值区间能够基本上覆盖实时控制时对象模型参数辨识值的变化范围，才能保障第五章算法控制性能不下降。

第七章 结束语

摘要：对本文的工作进行了总结，讨论了本文给出的简化广义预测控制算法的适用范围、在线实现时需要注意的问题和需要进一步研究的工作。

7.1 研究工作总结

生产实践发展的需要催生了模型预测控制方法，预测控制理论发展的目的是更好地服务于生产实践。因此，如何把先进的预测控制算法更好在控制回路上实现是预测控制发展的根本性问题之一。本论文正是从这一观念出发，在前人研究的基础上，对广义预测控制算法提出了一系列的简化实现方法。

在绪论部分，介绍了模型预测控制方法的发展过程、现状、目前存在的局限性以及发展趋势。分析了广义预测控制算法的特点及应用时存在的困难。总结了模型预测控制方法中通常采用的简化实现方法，综述了广义预测控制算法简化实现方面已取得的成果，并指出了论文所要进行的研究工作。

第二章，首先说明对于物理可实现的多变量系统，其 CARIMA 模型总可以构造成 $A(z^{-1})$ 与 $C(z^{-1})$ 取对角多项式矩阵的形式。在这种模型结构的基础上，给出了多变量广义预测控制算法的完整求解过程。在算法的推导过程中，显式地考虑过程纯滞后项，以提高计算效率。这种形式的模型结构，不但广义预测控制算法的求解过程可以得到很大程度的简化，相应的参数辨识问题也得到简化，可以把一个多输入多输出模型的复杂辨识问题分解成多个多输入单输出的小型辨识问题。

第三章中，通过对广义预测控制算法中输出自由响应项的进一步推导，得到了状态反馈形式的广义预测控制器。消除了在非自适应模式下在线求解过程自由响应的必要，可以像 PID 控制器一样实现 GPC 控制器。

第四章利用 CARIMA 模型直接递推得到了更加简洁的广义预测控制器形式，不但在自适应模式下无需进行 Diophantine 方程的求解，即使在非自适应模式下，实现方式与计算量也得到极大的简化。

第五章通过对多变量广义预测控制算法内在机理的分析，指出 GPC 算法实际上是一种函数映射关系，是从对象的模型参数空间到控制器系数空间的映射，

并可用神经网络的映射能力来实现广义预测控制的快速算法。

第六章在非线性液位装置上对以上简化的广义预测控制算法进行了对比实验研究。实验结果证明了各个方法的可行性、有效性与等效性。

7.2 简化方法的适用范围

对于自适应情况下的在线计算来说,第二章、第三章的方法计算量最为繁重,都要求解 Diophantine 方程,并需要矩阵逆运算。第四章递推算法计算量相对减少,不需要求解 Diophantine 方程,但也需要矩阵求逆。第五章所示快速算法计算量大大降低,既不需要求解 Diophantine 方程,也不需要矩阵求逆。但基于 BP 网络的快速算法只能保证在已训练的模型参数范围内具有优良的控制品质。因此,在计算资源及控制周期允许的情况下,选用第二章、第三章、第四章的方法均可。而对于计算资源有限,或者要求快速控制的系统,在对象模型参数变化的范围大概知道的情形下,第五章所示快速算法是最合适的选择。

在非自适应模式下,第四章的方法计算量最小。第三章所示方法控制器系数维数与预测时域有关,计算量相对大一点。第二章的方法需要求解输出的自由响应项,不太适合在非自适应模式下选用。第五章提出的方法在非自适应模式下的计算量与自适应情况一样多,主要由网络规模决定。不过在非自适应模式下,基于神经网络的广义预测控制快速算法适应能力相对最强。因为如果控制器已经投运了,却需要调整模型参数,且变化的参数落在已训练的范围,则神经网络控制器不需要作任何变动,而其它方法都要重新计算相关系数。实验证明在非自适应情形下,这些方法都具有较好的鲁棒性,能在一定范围内保证控制效果。因此,可以根据控制的需要,选用合适的方法。

7.3 实现简化算法时需要注意的问题

采用第二章、第三章的方法时,涉及的 Diophantine 方程求解可以采用递推形式,以减小计算量。对于基于 BP 网络的广义预测控制快速算法以外的其它方法中要求的矩阵求逆运算,按矩阵的结构形式可以采用与文献[83][84][85]中类似的递推求解方法,以进一步减轻计算负担。

在线实现时,对输入输出数据采用适当的数值滤波方法^[122],有利于提高算法的控制效果。

另外, 论文中所提出来的各种方法都没有考虑控制变量的约束条件, 而是通过适当调整设计参数来满足控制性能要求。尽管在本文的实验中, 很少有违背约束的情形发生, 但执行机构一般会存在控制变量的幅值及速率约束。因此在运用这些算法前, 应通过闭环仿真实验适当调整设计参数以尽可能地满足约束条件, 在线实现时应结合“cut-off”方法来保证输入变量不要违背约束条件。

7.4 需要进一步研究的工作

在许多系统中, 除了不可测量扰动外, 有时会有可测量的扰动。当需要对这些可测量扰动进行前馈控制时, 是否也可以通过分析其内在规律, 找到一些简化的方法呢? 这种可能性是存在的。因为扰动本身是系统的一种输入, 对过程的影响满足一定的规律, 只要能获得其模型便可以按论文提供的思路找到相似的解决方案^[123]。

在第六章单容非线性液位系统的实验研究中, 由于训练的样本区间基本上覆盖了实时辨识的参数变化区间, 基于 BP 网络的快速算法完全能够实现第四章方法的控制功能。但在多容液位系统的对比实验中, 由于训练 BP 网络时所选的参数区间不能完全覆盖在线控制时辨识参数值的变化范围, 第五章的方法就没有完全达到第四章方法的控制效果。因此还需要进一步研究, 以寻找选择模型参数区间的规律。

实际的输入输出变量都存在着一定的约束, 如果不加以考虑, 则过程就不能运行在期望的状态下。如何更有效地引进约束条件一直是模型预测控制方法研究的难点与热点问题。引入硬约束, 优化时便需要数值解法, 就无法回避繁重的在线计算负担。许多商业化模型预测控制产品中使用的分层优化方法与“罚函数”方法, 可以引入到广义预测控制方法中。例如对第五章的快速算法, 可以按不同的设计参数训练多个实现 GPC 快速算法的神经网络, 在线实现时通过一定的“罚函数”来选择相应的神经网络, 则既可以避免违背约束条件, 又可以减轻计算负担。

文献[124]指出对于状态变量与输入变量有约束的离散线性时不变系统的有限时域或者无限时域线性二次最优控制问题, 可以找到显式状态反馈解。并指出这些闭环形式的解是分段线性和连续的, 在线运行时只需要计算这些预先确定的分段线性函数就可以得到控制量。这种思路很有借鉴意义, 说明对于有约束的预测控制问题, 确实存在通过离线计算来简化在线实现的途径。对于有约束的广义预测控制算法也应该存在类似的规律, 可以进一步进行探讨。例如可以通过不同

参数区域的划分训练不同的网络来保证广义预测控制方法的全局最优性。

另外，论文中提到的简化方法，可以推广到状态空间模型及其它模型形式的广义预测控制方法中，同时也可以把这种思路与其它的预测控制算法相结合，寻找更加可行的实现方法。

参考文献

- [1] 席裕庚, 许晓鸣. 预测控制的现状、机理及发展前景. 孙优贤, 钱积新编, *工业过程模型化及控制*. 杭州: 浙江大学出版社, 1988, 1-6.
- [2] García, C. E., Prett, D. M., Morari, M. Model predictive control: Theory and practice—a survey. *Automatica*, 1989, 25(3):335-348.
- [3] 席裕庚, 许晓鸣, 张钟俊. 预测控制的研究现状和多层智能预测控制. *控制理论与应用*, 1989, 6(2): 1-7.
- [4] Richer, N. L.. Model predictive control: State of the art. In Y. Arkun, W. H. Ray (Eds.), *Chemical process control—CPC IV, Fourth international conference on chemical process control* (pp. 271-296). Amsterdam:Elsevier, 1991.
- [5] Morari, M., Lee, J. H.. Model predictive control: The good, the bad, and the ugly. In Y. Arkun, W. H. Ray (Eds.), *Chemical process control—CPC IV, Fourth international conference on chemical process control* (pp. 419-444). Amsterdam:Elsevier, 1991.
- [6] Muske, K. R. Rawlings, J. B.. Model predictive control with linear models. *A.I.C.H.E Journal*, 1993, 39(2):262-287.
- [7] Froisy, J. B.. Model predictive control: Past, present and future. *ISA Transactions*, 1994, 33:235-243.
- [8] Hillestad, M., Andersen, K. S.. Model predictive control for grade transitions of a polypropylene reactor. In *Proceedings of the 4th European symposium on computer aided process engineering (ESCAPE 4)*, Dublin, March 1994.
- [9] Rawlings, J. B., Meadows, E. S., Muske, K.. Nonlinear model predictive control: a tutorial and survey. In *Proceedings of IFAC ADCHEM*, Japan, 1994.
- [10] 徐立鸿. 预测控制的研究现状及问题. *控制理论与应用*. 1994, 11(1): 121-125.
- [11] Ohshima, M. Ohno, H. Hashimoto, I.. Model predictive control: Experiences in the university-industry joint projects and statistics on MPC applications in Japan. *International workshop on predictive and receding horizon control*,

- Korea, October, 1995.
- [12] 李平, 王树青, 王骥程. 预测控制研究的概况. *化工自动化及仪表*. 1995, 22(6): 3-9.
- [13] Mayne, D. Q.. Nonlinear model predictive control: An assessment. In J. C. Kantor, C. E. Garcia, B. Carnahan (Eds.), *Fifth international conference on chemical process control* AICHE and CACHE, (pp. 217-231), 1997.
- [14] Lee, J. H. Cooley, B.. Recent advances in model predictive control and other related areas. In J. C. Kantor, C. E. Garcia, B. Carnahan (Eds.), *Fifth international conference on chemical process control* AICHE and CACHE, (pp. 201-216b), 1997.
- [15] Qin, S. J., Badgwell, T. A.. An overview of industrial model predictive control technology. In J. C. Kantor, C. E. Garcia, B. Carnahan (Eds.), *Fifth international conference on chemical process control* AICHE and CACHE, (pp. 232-256), 1997.
- [16] Allgower, F., Badgwell, T. A., Qin, S. J., Rawlings, J. B., Wright, S. J.. Nonlinear predictive control and moving horizon estimation—an introductory overview. In P. M. Frank (Eds.), *Advances in control: highlights of ECC'99*. Berlin: Springer, 1999.
- [17] 王献忠, 杜维. 模型预测控制发展概况. *自动化与仪器仪表*. 1999, 84(4): 4-9.
- [18] 梁春燕, 谢剑英. 预测控制中的若干问题研究. *自动化与仪器仪表*. 1999, 84(4): 10-13.
- [19] Rawlings, J. B.. Tutorial overview of model predictive control. *IEEE Control Systems Magazine*, 2000, 20:38-52.
- [20] Mayne, D. Q., Rawlings, J. B., Rao, C. V., Scokaert, P. O. M.. Constrained model predictive control: Stability and optimality. *Automatica*, 2000, 36:789-814.
- [21] Qin, S. J., Badgwell, T. A.. An overview of nonlinear model predictive control applications. In J. C. Kantor, C. E. Garcia, & B. Carnahan (Eds.), *Nonlinear model predictive control*. Basel: Birkhauser, 2000.
- [22] Kulhavy, R., Lu, J., Samad, T.. Emerging technologies for enterprise optimization in the process industries. In *chemical process control-6, assessment and new directions for research (CPC VI)*, Tuscon, Arizona, January 2001.

- [23] Young, R. E., Bartusiak, R. B., Fontaine, R. B.. Evolution of an industrial nonlinear model predictive controller. In *Chemical process control-CPC VI*, Tucson, Arizona, CACHE, 2001, pp. 399-410.
- [24] Downs, J. J.. Linking control strategy design and model predictive control. In *Chemical process control-6, assessment and new directions for research (CPC VI)*, Tuscon, Arizona, January 2001.
- [25] Qin, S. J., Badgwell, T. A.. A survey of industrial model predictive control technology. *Control Engineering Practice*, 2003, 733-764.
- [26] 席裕庚. 预测控制. 北京: 国防工业出版社, 1993.
- [27] 舒迪前. 预测控制系统及其应用. 北京: 机械工业出版社, 1996.
- [28] Camacho, E. F., Bordons C.. *Model Predictive Control*. Springer, 1999.
- [29] Allgower, F., Zheng, A., (Eds.). *Nonlinear model predictive control, progress in systems and control theory*. Vol. 26. Basel, Boston, Berlin: Birkhauser Verlag, 1999.
- [30] Kouvaritakis, B., Cannon, M. (Eds.). *Nonlinear predictive control, theory and practice*. London: The Institution of Electrical Engineers, 2001.
- [31] Maciejowski, J. M.. *Predictive control: with constraints*. Pearson Education Limited, 2002.
- [32] Kalman, R. E.. Contributions to the theory of optimal control. *Bulletin de la Societe Mathematique de Mexicana*, 1960, 5: 102-119.
- [33] Kalman, R. E.. A new approach to linear filtering and prediction problems. *Transactions of ASME, Journal of Basic Engineering*, 1960, 87: 35-45.
- [34] Goodwin, G. C., Graebe, S. F., Salgado, M. E.. *Control system design*. Englewood Cliffs, NJ:Prentice Hall, 2001.
- [35] Richalet, J., Rault, A., Testud, J. L., Papon, J.. Algorithmic control of industrial processed. In *Proceedings of the 4th IFAC symposium on identification and system parameter estimation*. (pp. 1119-1167), 1976.
- [36] Prett, D. M., Gillette, R. D.. Optimization and constrained multivariable control of a catalytic cracking unit. In *Proceedings of the joint automatic control conference*, 1980.
- [37] Propoi, A. I.. Use of linear programming methods for synthesizing sampled-data automatic systems. *Automatic Remote Control*, 1963, 24(7): 837-844.
- [38] Lee, e. B., Markus, L.. *Foundations of optimal control theory*. New York:

- Wiley, 1967.
- [39] Culter, C. R., Ramaker, B. L.. Dynamic matrix control—a computer control algorithm. *AIChE national meeting*, Houston, TX, April, 1979.
- [40] Culter, C. R., Ramaker, B. L.. Dynamic matrix control—a computer control algorithm. In *Proceeding of the joint automatic control conference*, 1980.
- [41] Culter, C. R., Morshedi, A., Haydel, J. An industrial perspective on advanced control. In *AIChE annual meeting*, Washington, DC, October, 1983.
- [42] Garcia, C. E., Morshedi, A. M.. Quadratic programming solution of dynamic matrix control (QDMC). *Chemical Engineering Communications*, 1986, 46: 73-87.
- [43] Grosdidier, P., Froisy, B., Hammann, M.. The IDCOM-M controller. In T. J. McAvoy, Y. Arkun, E. Zafiriou (Eds.), *Proceedings of the 1988 IFAC workshop on model based process control* (pp. 31-36). Oxford: Pergamon Press, 1988.
- [44] Froisy, J. B., Matsko, T.. IDCOM-M application to the shell fundamental control problem. *AIChE annual meeting*, November 1990.
- [45] Marquis, P., Broustail, J. P.. SMOC, a bridge between state space and model predictive controllers: Application to the automation of a hydrotreating unit. In T. J. McAvoy, Y. Arkun, E. Zafiriou (Eds.), *Proceedings of the 1988 IFAC workshop on model based process control* (pp. 27-43). Oxford: Pergamon Press, 1988.
- [46] Yousfi, C., Tournier, R.. Steady-state optimization inside model predictive control. In *Proceedings of ACC'91*, Boston, MA (pp. 1866-1870), 1991.
- [47] Richalet, J., Rault, A., Testud, J. L., Papon, J.. Model predictive heuristic control: Application to industrial processes. *Automatica*, 14: 413-428, 1978.
- [48] Prett, D. M., Garcia, C. E.. *Fundamental process control*. Boston: Butterworths, 1988.
- [49] DMC Corp. [DMC]TM. *Technology overview*. Product literature from DMC Corp., July 1994.
- [50] Honeywell Inc.. *RMPCT concepts reference*. Product literature from Honeywell, Inc., October 1995.
- [51] Richalet, J.. Industrial applications of model-based control. *Automatica*, 1993, 29: 1251-1274.
- [52] De Oliveira, N. M. C., Biegler, L. T.. Constraint handling and stability

- properties of model-predictive control. *A.I.C.H.E. Journal*, 1994, 40(7): 1138-1155.
- [53] De Oliveira, N. M. C., Biegler, L. T.. An extension of newton-type algorithms for nonlinear process control. *Automatica*, 1995, 31: 281-286.
- [54] Sentoni, G. B., Biegler, L. T., Guiver, J. B., Zhao, H.. State-space nonlinear process modeling. *A.I.C.H.E. Journal*, 1998, 44(10): 2229-2239.
- [55] Zhao, H., Guiver, J. P., Sentoni, G. B.. An identification approach to nonlinear state space model for industrial multivariable model predictive control. In *Proceedings of the 1998 American control conference*, Philadelphia, Pennsylvania (pp. 796-800), 1998.
- [56] Zhao, H., Guiver, J., Neelakantan, R., Biegler, L. T.. A nonlinear industrial model predictive controller using integrated PLS and neural state space model. In *IFAC 14th triennial world congress*, Beijing, Peoples Republic of China, 1999.
- [57] Berkowitz, P., Papadopoulos, M.. *Multivariable process control method and apparatus*. US Patent 5396416, 1995.
- [58] *MVC3.0 User Manual*. Continental Controls, Inc. Product Literature, 1995.
- [59] Berkowitz, P., Papadopoulos, M., Colwell, L., Moran, M.. *Multivariable process control method and apparatus*. US Patent 5488561, 1996.
- [60] Poe, W., Munsif, H.. Benefits of advanced process control and economic optimization to petrochemical processes. In *Hydrocarbon processing's process optimization conference*, Houston, TX, 1998.
- [61] Bartusiak, R. D., Fontaine, R. W.. *Feedback method ofr controlling non-linear processes*. US Patent 5682309, 1997.
- [62] Demoro, E., Axelrud, C., Johnston, D., Martin, G.. Neural network modeling and control of polypropylene process. In *Society of plastics engineers international conference*, Houston, TX, 1997.
- [63] Keeler, J., Martin, G., Boe, G., Piche, S., Mathur, U., Johnston, D.. *The process perfecter, the next step in multivariable control and optimization*. Technical report, Pavilion Technologies, Inc., Austin, TX, 1996.
- [64] Martin, G., Boe, G., Keeler, J., Timmer, D., Havener, J., *Method and apparatus for modeling dynamic and steady-state processes for prediction, control, and optimization*. US Patent, 1998.
- [65] Martin, G., Johnston, D.. Continuous model-based optimization. In

- Hydrocarbon processing's process optimization conference*, Houston, TX, 1998.
- [66] Piche, S., Sayyar-Rodsari, B., Johnson, D., Gerules, M.. Nonlinear model predictive control using neural networks. *IEEE Control System Magazine*, 2000, 20(3): 53-62.
- [67] Lasdon, L. S., Warren, A. D.. *GRG2 user's guide*. Cleveland, OH: Cleveland State University, 1986.
- [68] Scokaert, P. O. M., Rawlings, J. B.. Constrained linear quadratic regulation. *IEEE Transactions on Automatic Control*, 1998, 43(8): 1163-1169.
- [69] Chen, H., Allgower, F.. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 1995, 34(10): 1205-1218.
- [70] Rao, C. V., Wright, S. J., Rawling, J. B.. Application in interior-point methods to model predictive control. *Journal of Optimization Theory Application*, 1998, 99: 723-757.
- [71] Dollar, R., Melton, L. L., Morshedi, A. M., Glasgow, D. T., Repsher, K. W.. Consider adaptive multivariable predictive controllers. *Hydrocarbon Processing*, 1993, 10: 109-112.
- [72] Foss, B. A., Lohmann, B., Marquardt, W.. A field study of the industrial modeling process. *Journal of Process Control*, 1998, 8(5,6): 325-338.
- [73] Camacho, E. F., Berenguel, M., Bordons C.. Adaptive generalized predictive control of a distributed collector field. *IEEE Tran. Control Systems Technology*, 1994, 2: 462-468.
- [74] Clarke, D. W. Application of generalized predictive control to industrial processes. *IEEE Control Systems Magazine*, 1988, 122: 49-55.
- [75] Richalet, J., Abu el Ata-Doss, S., Arber, C., Kuntze, H. B.. Predictive functional control: application to fast and accurate robots. In *Proc. IFAC World Congress*, Munich, 1987.
- [76] Clarke, D. W., Mohtadi, C., Tuffs, P. S.. Generalized Predictive Control. Part I. The Basic Algorithm. *Automatica*, 1987, 23(2): 137-148.
- [77] Clarke, D. W., Mohtadi, C., Tuffs, P. S.. Generalized Predictive Control. Part II. Extensions and Interpretations. *Automatica*, 1987, 23(2): 149-160.
- [78] 徐立鸿, 冯纯伯. 论广义预测控制. *控制与决策*, 1992, 7(4): 241-246.
- [79] 谢克明, 李国勇. 广义预测控制技术的现状及展望. *电力学报*, 1994,

- 9(2): 1-5.
- [80] 王伟, 杨建军. 广义预测控制: 理论、算法及应用. *控制理论与应用*, 1997, 14(6): 777-786.
- [81] 王伟. *广义预测控制理论及其应用*. 北京: 科学出版社, 1998.
- [82] Ydstie, B. E.. Extended Horizon Adaptive Control. In Proc. 9th IFAC World Congress, Budapest, Hungary, 1984.
- [83] 袁著祉. 递推广义预测自校正控制器. *自动化学报*, 1989, 15(4):348-351.
- [84] 徐立鸿, 袁震东. ARMAX 模型的递推广义预测控制算法. *控制理论与应用*, 1990, 7(3):102-107.
- [85] 郭庆鼎, 金元郁, 胡耀华. 求解 GPC 中逆矩阵的递推算法. *控制与决策*, 1996, 11(4): 510-513.
- [86] 王一晶, 左志强. 一种新型广义预测控制快速算法. *模型识别与人工智能*, 2002, 15(3):295-298.
- [87] 徐仲, 张凯院, 陆全. TEOPLITZ 矩阵类的快速算法. 西安: 西北工业大学出版社, 1999.
- [88] 金元郁, 顾兴源. 改进的广义预测控制算法. *信息与控制*, 1990, 19(3):8-14.
- [89] 金元郁, 顾兴源. 改进的多变量广义预测控制算法. *信息与控制*, 1990, 19(6):20-22.
- [90] 王伟. 广义预测自适应控制的直接算法及全局收敛性分析. *自动化学报*, 1995, 21(1):57-62.
- [91] 王伟. 一种广义预测自适应控制的直接算法. *自动化学报*, 1996, 22(3):270-277.
- [92] 郭健, 陈庆伟, 吴晓蓓, 胡维礼, 吴宏鑫. 一类非线性系统的稳定自适应控制. *控制理论与应用*, 2003, 20(4):603-606.
- [93] 王倩, 荣德善. 单变量隐式广义预测自校正控制算法. *计算技术与自动化*, 1990, 9(3):1-5.
- [94] 王倩. 多变量隐式广义预测自校正控制算法. *计算技术与自动化*, 1992, 11(2):37-42.
- [95] 舒迪前, 石中锁. 隐式广义预测自校正控制器及其全局收敛性. *自动化学报*, 1995, 21(5):545-554.
- [96] 胡耀华, 贾欣乐. 广义预测控制的直接算法. *控制与决策*, 2000, 15(2):221-223.
- [97] 王秩, 席裕庚. 广义预测控制的并行算法. *第一届中国智能控制与智能*

- 自动化学术会议论文集(CICA94). 沈阳: 东北大学出版社, 1994, 463-468.
- [98] 王秩, 席裕庚. 广义预测控制的并行算法. *自动化学报*, 1996, 22(1):74-78.
- [99] 慕德俊, 戴冠中. 状态空间模型广义预测控制的并行算法. *控制理论与应用*, 1995, 12(5):646-652.
- [100] 慕德俊. 基于输入输出模型广义预测模型的并行算法. *控制理论与应用*, 1997, 14(1): 80-83.
- [101] Camacho, E. F., Bordons, C.. Implementation of self tuning generalized predictive controllers for the process industry. *International Journal of Adaptive Control and Signal Processing*, 1993, 7(1): 63-73.
- [102] 谢永斌, 李琳, 罗忠, 冯祖仁, 胡保生. 利用小脑模型提前计算的广义预测控制. *西安交通大学学报*, 1997, 31(10):30-34.
- [103] 谢永斌, 罗忠, 冯祖仁, 胡保生. 基于连续映射小脑模型的广义预测控制快速算法. *控制理论与应用*, 1997, 14(6): 842-846.
- [104] 李奇安, 李平, 李悦. 基于 BP 网络的典型工业过程自适应预测区域控制. *抚顺石油学院学报*, 2001, 21(2): 66-71.
- [105] QI-AN LI, SHU-QING, WANG. Fast Algorithm for Adaptive Generalized Predictive Control Based on BP Neural Networks. In *Proceedings of the 2004 IEEE International Conference on Machine Learning and Cybernetics*, Shanghai, pp.738-842, August 26-29, 2004.
- [106] 袁著祉, 崔保民. 新型多变量广义预测自校正控制器. *控制理论与应用*, 1992, 9(6): 672-677.
- [107] 邢以群. *管理学*. 杭州: 浙江大学出版社, 1997.
- [108] 郑大钟. *线性系统理论*. 北京: 清华大学出版社, 1990.
- [109] 方崇智, 萧德云. *过程辨识*. 北京: 清华大学出版社, 1988.
- [110] Lennart Ljung. *System identification - theory for the user*. Tsinghua University Press, Beijing, 2002
- [111] Lennart Ljung. *System identification Toolbox for Use with MATLAB - User's Guide*. The MathWorks, 2002.
- [112] Katsuhiko Ogata. *Modern Control Engineering, Fourth Edition*. Prentice Hall, 2002.
- [113] Howard Demuth, Mark Beale. *Neural Network Toolbox for Use with MATLAB - User's Guide*. The MathWorks, 2002.

- [114] 李清泉. *自适应控制系统理论、设计与应用*. 北京:科学出版社, 1990.
- [115] 徐士良, 孙甲松. *科学计算通用程序集*. 沈阳: 辽宁科学技术出版社, 1997.
- [116] *JX-300X 集散控制系统使用手册-系统硬件*. 浙江浙大中控技术有限公司, 2003.
- [117] *JX-300X 集散控制系统使用手册-系统组态*. 浙江浙大中控技术有限公司, 2003.
- [118] *JX-300X 集散控制系统使用手册-实时监控*. 浙江浙大中控技术有限公司, 2003.
- [119] *JX-300X 集散控制系统使用手册-SCX 语言*. 浙江浙大中控技术有限公司, 2003.
- [120] *JX-300X 集散控制系统使用手册-流程图绘制*. 浙江浙大中控技术有限公司, 2003.
- [121] Manfred Morari, N. Lawrence Richer. *Model Predictive Control Toolbox for Use with Matlab*. User's Guide Version 1. The MathWorks, 1998.
- [122] 王建华, 黄河清. *计算机控制技术*. 北京: 高等教育出版社, 2003.
- [123] 李奇安, 李平, 于海斌, 王树青. 串联系统的多前馈—反馈广义预测控制. *控制与决策*, 2002, 17(4): 402-406.
- [124] Alberto Bemporad, Manfred Morari, Vivek Dua, Efstratios N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 2002, 38:3-20.

作者在攻读博士期间发表与完成的论文

博士期间发表的论文:

1、李奇安、李平、于海斌、王树青, 串联系统的多前馈—反馈广义预测控制, 控制与决策, Vol.17, No.4, 402—406, 2002 年 4 月。(EI 收录)

2、李奇安、王树青、李平, 基于神经网络的自适应广义预测控制快速算法, 第二十一届中国控制会议论文集, 杭州, 646—650, 2002 年 8 月 12 日。

3、QI-AN LI, SHU-QING WANG. Fast Algorithm for Adaptive Generalized Predictive Control Based on BP Neural Networks. Proceedings of the 2004 IEEE International Conference on Machine Learning and Cybernetics, Shanghai, pp:738-842, August 26-29, 2004。(EI 收录)

已投稿的论文:

1、李奇安、王树青, 自适应广义预测控制器系数直接计算方法及实验研究, 信息与控制。

作者简介

李奇安，男，1971年8月8日出生于湖南华容。1991年考入辽宁石油化工大学自动化系学习，1995年获工学学士学位。1995年到1998年期间在湖南省岳阳化肥厂仪表车间任助理工程师。1998年考入辽宁石油化工大学控制理论与工程专业攻读硕士学位，师从李平教授，2001年获得工学硕士学位。2001年考入浙江大学工业控制技术国家重点实验室先进控制研究所攻读博士学位，师从王树青教授。研究兴趣为：预测控制，自适应控制，神经网络控制等。