

多路分时控制系统中的 模拟保持技术

吕 东 庚

〔计算机科学(电脑)系〕

摘 要

本文论述在微机多路分时控制系统中,如何用单个 D/A 变换器,对多路的执行机构实现有效地驱动。讨论采用这一技术时,电路系统的响应时间,所需占用的 CPU 工作时间与控制路数及电路参数的关系,最后给出电路的设计步骤及程序流程。

1. 引言

一

在多路工业控制系统中,常采用分时处理方法把计算机中央处理器的有效工作时间,按一定的原则分配给各路使用。因此被控制的每一路所能占用的 CPU 和输出端口的有效时间都只能是短暂的,而且被控制的路数越多,这一时间就越短。对此,如何把不同时刻出现在计算机输出端口上的数字控制量,转换为相应路的模拟输出量,并使它有足够的保持时间,是多路控制系统中的一个重要问题。这一问题实质上是短脉冲的展宽问题。利用电容器的贮电特性与场效应晶体管或具有深度负反馈的宽带运算放大器,把微秒甚至毫微秒级的短脉冲展宽到毫秒级,近十多年来已广泛用于高速脉冲的测量中。然而采用这一技术由于大容量电容器的漏电,因而即使采用多级展宽技术也难于把脉冲展宽到秒级以上。七十年代中后期,提出了采用存贮器和 A/D, D/A 变换器相结合的方法,它为短脉冲的展宽和长期保持提供了良好的解决办法。但是这种办法由于每一路需有一套寄存器和 D/A 变换器,因此在控制路数很多时(例如几十路甚至近百路),则电路就过于庞大,整机功率明显上升,可靠性下降,并使造价显著提高。为此我们探讨了一种综合性的解决方案。这一方案每一路仅用一个电容保持电路,利用它获得各路输出信号的短时间保持,并利用计算机中少量的内存单元做为计算机输出控制量的长期存贮。然后利用分时控制系统中 CPU 的空间时间,定期把内存中的输出控制量通过一个公用的 D/A 变换器,转换为模拟量,而后分别送到相应路的保持电容上,利用它来补充电容本身及外界的漏电。从而使各路的输出既能根据计算机的运算结果做出快速反应,又能在要求输出恒定时保持长期不变的跟踪目的。

本文 1985 年 10 月 5 日收到。

二

图 1 给出了采用上述方案的 M 路控制系统的输出原理框图。图中的公用 D/A 变换器把各个时刻从计算机送出的数字量转换为相应的模拟量。多路模拟开关则在计算机的控制下把相应时刻中经 D/A 变换及缓冲放大后的模拟信号送入相应输出通路的电容保持电路，而后送往相应路的控制执行机构。由于系统设计时采用了窃取 CPU 空闲时间对各电容电路的电荷泄漏实行定时修正，因而保证了各路模拟输出 U_{Ao} 既能根据计算机的运算结果做出快速反应，而在要求输出恒定时又能保持长期不变的跟踪目的。

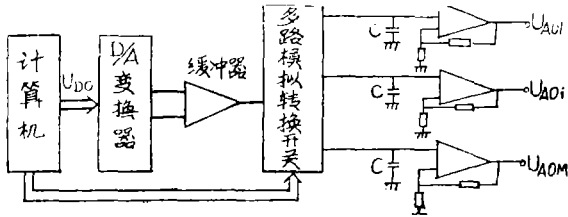


图 1

图 2 为系统输出电路的波形示意图，图中实线为未引入 CPU 修正时的波形图，虚线为引入修正后的波形图。图 2 (a) 中各输出量的数字标示 1, 2, ..., M ，分别表示该输出量为 1, 2, ..., M 路的输出控制量。

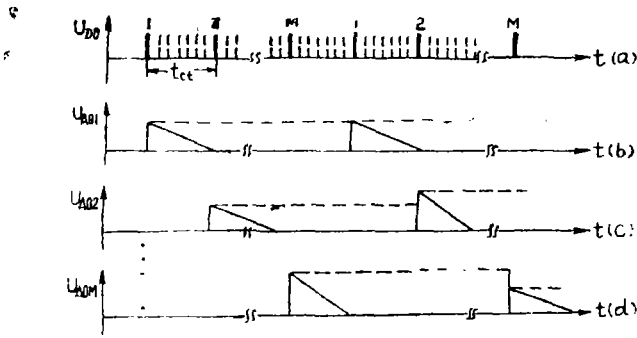


图 2

从图 2 (a) 可以看出，在分时控制系统中，每一路的输出控制量在计算机输出端口的持续时间是短暂的（其最长时间可达到 t_{ci} ）。同时，从图 2 可以看到在不引入修正时，它们在各路的最终输出点 U_{Ao} 上的模拟信号是一个指数衰减信号。CPU 修正的引入在图 2 (a) 中用虚线方框表示。每一虚线方框表示计算机在该时间内依序输出了 M 路的输出控制量。由于这一措施的引入使各路的输出保持跟踪的波形则分别用虚线表示在图 2 (b) 至 (d) 中，

为进一步分析系统的特性,图3给出了在计算机端口上对某一个特定输出路的输出时间波形图。图中每一矩形虚线方框表示CPU对各路进行一次修正时,在计算机端口上输出的各路输出控制数字量。它们在时间上是依照输出路的路号顺序被送出的。

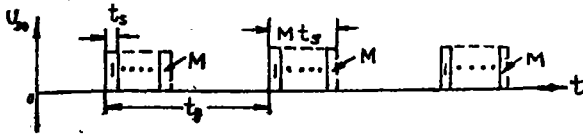


图 3

方框中的实线是与我们所指定的输出路(第一路)相对应的计算机输出数字量在计算机端口上的波形。它在端口寄存器中的持续时间用 t_s 表示,而该路相邻两次输出量出现的时间间隔用 t_g 表示。 t_g 恰为CPU对输出进行修正的时间间隔。根据图1、3可得出对某一路输出的等效电路如图4。

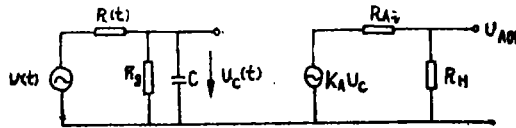


图 4

图中 R_g 为保持电容的漏电阻,电路布局和开关的漏电阻,以及保持放大器输入电阻等因素的并联等效电阻。 K_A 和 R_{A1} 分别为保持放大器的电压增益与输出阻抗, R_H 为它的负载电阻。由于上述工作过程可视为周期为 t_g 的非正弦准周期过程,因此 $U(t)$ 、 $R(t)$ 可分别表示成,

$$u(t) = \begin{cases} u_i & 0 < t \leq t_s \\ 0 & t_s < t \leq t_g \end{cases}, \quad R(t) = \begin{cases} R_g & 0 < t \leq t_s \\ \infty & t_s < t \leq t_g \end{cases}$$

对上述电路进行分析可求出,在计算机送出输出控制量之后的第 N 个到 $N+1$ 个周期时间内,保持电容器上的电压 $U_o(t)$ 与理想输出电压 U_i 之比值 K_{AH0} (理想输出电压 U_i 就是与计算机输出控制数字量精确对应的模拟电压值)。

$$K_{AH0} = \frac{u_o(t)}{u_i}$$

$$= \left[1 - \exp\left(-\frac{t_s}{\tau_s}\right) \right] \cdot \exp\left(-\frac{t-t_s}{\tau_g}\right) \frac{1 - \left[\exp\left(-\frac{t_s}{\tau_s}\right) \cdot \exp\left(-\frac{t_s-t_s}{\tau_g}\right) \right]^N}{1 - \left[\exp\left(-\frac{t_s}{\tau_s}\right) \cdot \exp\left(-\frac{t_s-t_s}{\tau_g}\right) \right]} \quad (1)$$

式中,

$$\tau_s = C \cdot (R_g // R_H) \approx C \cdot R_g, \quad \tau_g = C \cdot R_H$$

为分析方便,令电路的额定传输系数 K 为,

$$K = \left[1 - \exp\left(-\frac{t_s}{\tau_s}\right) \right] \cdot \frac{1 - \left[\exp\left(-\frac{t_s}{\tau_s}\right) \cdot \exp\left(-\frac{t_s - t_s}{\tau_s}\right) \right]^N}{1 - \exp\left(-\frac{t_s}{\tau_s}\right) \cdot \exp\left(-\frac{t_s - t_s}{\tau_s}\right)} \quad (2)$$

则

$$K_{AH} = K \cdot \exp\left(-\frac{t - t_s}{\tau_s}\right) \quad (3)$$

並令:

$$I = (t_s - t_s) / \tau_s \quad (4)$$

$$J = t_s / \tau_s \cdot I \quad (5)$$

$$a = \tau_s / \tau_s \quad (6)$$

通过计算可求出在工程实用的 I, J 范围内的 $K-N$ 关系曲线如 5。该图表明当 $N \geq 4$ 时, 电路的额定传输系数 K , 在 N 值增加时並沒有明显的变化。这一事实说明, 在采用 CPU 进行定时修正的输出电路中, 电路对阶跃信号的响应时间 t_{ss} 不大于 $4t_s$ 。

当 N 取为 4 根据上述公式 (2) 可求出对于不同 K 值的 $J-I$ 关系曲线如图 6 所示。利用图 6 则可在给定的传输系数 K 及 I 值已知时, 求出相应的 J 值。並从系统规定的响应时间 t_{ss} 求出 t_s , 而后求出 t_s 。

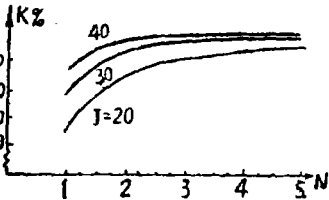


图 5 $K-N$ ($I=0.05, J=20-40$)

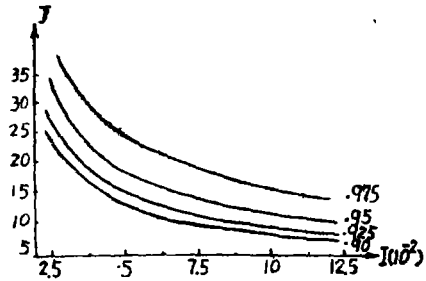


图 6 $J-I$ ($N=4, K=0.9, 0.925, 0.95, 0.975$)

对于这一电路方案的另一个重要问题是计算机对输出电路的定时修正究竟应占用 CPU 多少工作时间。如果这一占用的时间比率太大, 将使计算机对被控制对象工作状态变化的反映速度下降, 或使它所能控制的路数减少。若把 CPU 用于对输出进行修正的时间与 CPU 全部工作时间之比, 定义为输出修正对 CPU 的时间占用率, 並用 P 表示, 则从图 2 及式 (2) ~ (6) 可得:

$$P = \frac{Mt_s}{t_s} = \frac{J}{J+a} \cdot M \approx M \cdot \frac{J}{a} \quad (7)$$

式中 M 为系统的控制总路数, J, a 分别为式 (4), (5) 所定义的电路参数。当电路选用 CL11 涤纶电容器与运算放大器 (例如 CH14573) 组成保持电路, 並用 CH4051C 做为多路模拟选通开关时, 则上述 a 值为 $(5 \sim 10) \cdot 10^4$ 。如果要求电路的额定传输系数 $K=0.95$, 电路的响应时间为 $4 \cdot t_s$, 则从图 5 可查得 $J=20$ 。这时 $P = M \cdot (2 \sim 4) \times 10^{-4}$ 。对于 100 路的控制系统 CPU 用于对输出进行修正的时间不大于 CPU 全部工作的百分之四。

三

上述方案的电路设计可按下述步骤进行:

1. 根据执行机构的惰性, 决定电路输出信号在两次被修正的时间间隔内的最大容许波动量 ΔX . 它应大于或等于式(3) K_{AN} 中 $EXP\left(-\frac{t_0-t_s}{\tau_s}\right)$ 的最大波动值 $1-EXP\left(-\frac{t_0-t_s}{\tau_s}\right)$.

因此可从 $\Delta X = 1-EXP\left(-\frac{t_0-t_s}{\tau_s}\right)$ 求出 $\frac{t_0-t_s}{\tau_s}$ 之值, 即 I 值.

2. 根据控制系统的要求, 确定电路的最大响应时间 $t_{R.}$ 它应满足:

$$t_{R.} \geq 4t_s$$

从此式则可求出 t_s .

3. 从给定的电路额定传输系数 K , 以及已知的 I 值从图 6 求出, 当 $N=4$ 时的最小 I 值即为 I_0 . 并通过式(4)~(6)导出的公式:

$$t_s \geq \frac{I_0}{a} t_0$$

则可求出 t_s .

4. 从式(4)求出 τ_s , 并根据电路的实际条件决定电路的泄漏电阻 R_s , 则可求出保持电容 C 的相应数值.

5. 定时修正功能的实现:

为了对各路模拟输出信号进行定时修正, 必须在计算机内存中指定若干存贮单元. 它们分别存贮着计算机对每一路的当前控制数字量. 並利用计算机的定时系统, 产生一系列间隔为 t_s 的周期脉冲信号. 利用这一信号向 CPU 定时申请中断. 一旦 CPU 接受这一中断请求, 则执行一个输出修正子程序. 该子程序将把存贮在上述内存中的控制数字量依序送入一个公用的 D/A 变换器, 同时接通相应的多路模拟转换开关, 使计算机依序对各路的输出值进行修正. 图 7 为输出修正子程序的流程图. 从流程图中不难看出, 输出电路仅在子程序执行的一部分时间内才与计算机的输出端口接通. 因此 CPU 对每一路的服务时间将不仅包括实际的接通时间 t_s , 还包括了为实现这一接通所需的辅助服务时间. 即 CPU 对每一路进行输出修正的总服务时间为 $A \cdot t_s$. A 为大于 1 的数值, 它取决于 CPU 的型号, 计算机机的结构, 以及程序设计技术. 对于 Z80-CPU, A 可取为 2.5. 考虑到这一因素后, 引入输出修正对 CPU 的时间占用率应修改为:

$$P = A \cdot M \cdot \frac{I}{a}$$

当控制路数 $M=100$, 电路额定传输系数 $K=0.95$, $N=4$, $a=(5 \sim 10) \cdot 10^4$ 时, P 为 $(5 \sim 10)\%$.

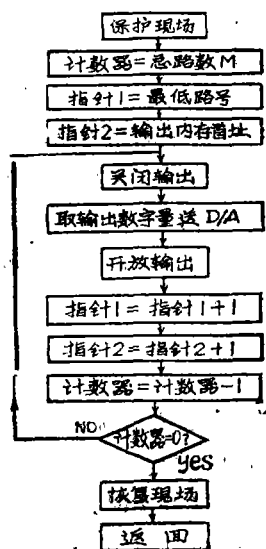


图 7 输出修正子程序流程图

本文的主要论点, 笔者曾在电脑系计算机控制课题组中做过介绍, 並得到组内同志们的热心支持。在此表示感谢。!

参 考 文 献

- [1] Impulse-peak Measument by Meansof pulse Lengthener circuit with Source Follower, IEEE ONI. M., Vol IM—30 August, (1971).
- [2] Г. В. 阿塔别柯夫著, 江泽佳等译, 线性电路理论(下册), 人民教育出版社, (1963).
- [3] 工业控制计算机使用技术编写组, 工业控制计算机使用技术, 化学工业出版社, (1982).

An Analog and Hold Technique for Multiplexing Time-sharing System

Lu Donggong

Abstract

This paper discusses how to drive the execution unit effectively with single D/A converter in a microcomputer controlled multiplexing time-sharing system. While this technique is introduced, the response time of the circuit, and the relationship of the CPU time and the paths number and the circuit parameters are discussed as well. And finally, the steps of circuit design and the program flowchart are given.