



(12) 发明专利

(10) 授权公告号 CN 111030552 B

(45) 授权公告日 2021.10.15

(21) 申请号 201911256485.X

H04L 12/40 (2006.01)

(22) 申请日 2019.12.09

H04L 29/08 (2006.01)

(65) 同一申请的已公布的文献号
申请公布号 CN 111030552 A

(56) 对比文件

CN 1157515 A, 1997.08.20

CN 108983672 A, 2018.12.11

(43) 申请公布日 2020.04.17

CN 105119702 A, 2015.12.02

(73) 专利权人 常州节卡智能装备有限公司
地址 213100 江苏省常州市武进区武进国
家高新技术产业开发区武宜南路377
号10号楼东幢一层

CN 1702970 A, 2005.11.30

CN 108494313 A, 2018.09.04

CN 1828464 A, 2006.09.06

JP 2015002596 A, 2015.01.05

JP 2010283901 A, 2010.12.16

(72) 发明人 李明洋

CN 108791473 A, 2018.11.13

(74) 专利代理机构 北京超凡宏宇专利代理事务
所(特殊普通合伙) 11463

CN 206441006 U, 2017.08.25

CN 102130640 A, 2011.07.20

CN 104850065 A, 2015.08.19

代理人 张磊

审查员 舒红芳

(51) Int. Cl.

H02P 29/00 (2016.01)

H04L 7/00 (2006.01)

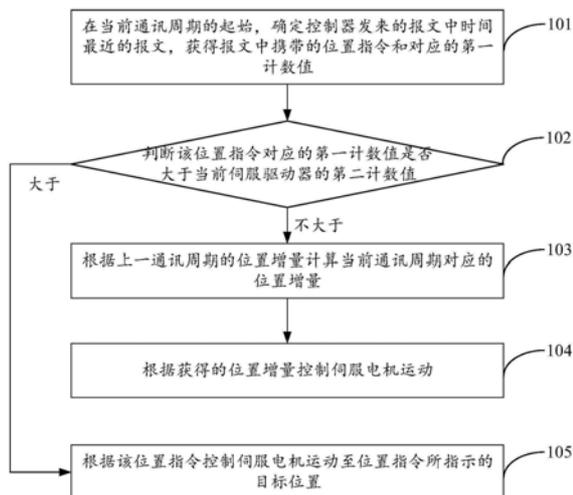
权利要求书2页 说明书8页 附图3页

(54) 发明名称

一种伺服驱动器的同步控制方法及伺服驱动器

(57) 摘要

本申请提供一种伺服驱动器的同步控制方法及伺服驱动器,该方法包括:在当前通讯周期的起始,获取控制器发来的位置指令中时间最近的位置指令,以及获取控制器对位置指令的第一计数值;判断所述位置指令所对应的第一计数值是否大于当前所述伺服驱动器对通讯周期的第二计数值;在所述第一计数值不大于所述第二计数值时,根据上一通讯周期的位置增量计算当前通讯周期对应的位置增量;根据所述位置增量控制伺服电机运动。本申请能够使不具有同步时钟的通讯方案也能够用于同步位置模式,有效解决在没有同步时钟的情况下,由于控制器的位置指令错位带来的抖动问题,使电机的运行更加平稳。



1. 一种伺服驱动器的同步控制方法,其特征在于,所述方法包括:

在当前通讯周期的起始,获取控制器发来的位置指令中时间最近的位置指令,以及获取控制器对位置指令的第一计数值;

判断所述位置指令所对应的第一计数值是否大于当前所述伺服驱动器对通讯周期的第二计数值;在所述第一计数值不大于所述第二计数值时,根据上一通讯周期的位置增量计算当前通讯周期对应的位置增量;

根据所述位置增量控制伺服电机运动;

所述第一计数值和所述第二计数值均由n个比特的二进制数字组成,所述第一计数值和第二计数值在取值范围 $[0, 2^n - 1]$ 内循环取值,n为正整数;

所述根据上一通讯周期的位置增量计算当前通讯周期对应的位置增量,包括:

获取对当前通讯周期之前的每个通讯周期的位置增量进行累加后得到的累加值Pos_Bk_m;

在 $K_m < K_{(m+1)} \leq 2^n - 1$ 时,通过如下公式确定当前通讯周期对应的位置增量Delt_Pos_(m+1):

$$\text{Delt_Pos_}(m+1) = \frac{\text{Pos_}(m+1) - \text{Pos_Bk_}m}{K_{(m+1)} - K_m};$$

其中,Pos_(m+1)为当前通讯周期起始时获取的时间最近的位置指令,K_(m+1)是与位置指令Pos_(m+1)对应的第一计数值,K_m是与位置指令Pos_m对应的第一计数值,Pos_m为上一通讯周期起始时获取的时间最近的位置指令。

2. 根据权利要求1所述的方法,其特征在于,在获取对当前通讯周期之前的每个通讯周期的位置增量进行累加后得到的累加值Pos_Bk_m之后,所述方法还包括:

在 $K_{(m+1)} < K_m < K_{(m+1)} + 2^n$ 时,通过如下公式确定当前通讯周期对应的位置增量Delt_Pos_(m+1):

$$\text{Delt_Pos_}(m+1) = \frac{\text{Pos_}(m+1) - \text{Pos_Bk_}m}{K_{(m+1)} + 2^n - K_m}。$$

3. 根据权利要求1所述的方法,其特征在于,所述根据上一通讯周期的位置增量计算当前通讯周期对应的位置增量,包括:

将上一通讯周期的位置增量确定为当前通讯周期对应的位置增量。

4. 根据权利要求1所述的方法,其特征在于,在判断所述位置指令所对应的第一计数值是否大于当前所述伺服驱动器对通讯周期的第二计数值之后,所述方法还包括:

在所述第一计数值大于所述第二计数值时,根据所述位置指令控制伺服电机运动至所述位置指令所指示的目标位置。

5. 根据权利要求1-4任一项所述的方法,其特征在于,所述伺服驱动器与所述控制器之间的通讯使用控制器局域网络CAN通信或者Modbus通信。

6. 一种伺服驱动器,其特征在于,包括:

获取模块,用于在当前通讯周期的起始,获取控制器发来的位置指令中时间最近的位置指令,以及获取控制器对位置指令的第一计数值;

拟合模块,用于判断所述位置指令所对应的第一计数值是否大于当前所述伺服驱动器对通讯周期的第二计数值;并在所述第一计数值不大于所述第二计数值时,根据上一通讯周期的位置增量计算当前通讯周期对应的位置增量;所述第一计数值和所述第二计数值均由n个比特的二进制数字组成,所述第一计数值和所述第二计数值在取值范围 $[0, 2^n - 1]$ 内循环取值,n为正整数;所述拟合模块具体用于:获取对当前通讯周期之前的每个通讯周期的位置增量进行累加后得到的累加值Pos_Bk_m;并在 $K_m < K_{(m+1)} \leq 2^n - 1$ 时,通过如下公式确定当前通讯周期对应的位置增量Delt_Pos_(m+1):

$$\text{Delt_Pos_}(m+1) = \frac{\text{Pos_}(m+1) - \text{Pos_Bk_}m}{K_{(m+1)} - K_m};$$

其中,Pos_(m+1)为当前通讯周期起始时获取的时间最近的位置指令,K_(m+1)是与位置指令Pos_(m+1)对应的第一计数值,K_m是与位置指令Pos_m对应的第一计数值,Pos_m为上一通讯周期起始时获取的时间最近的位置指令;

控制模块,用于根据所述位置增量控制伺服电机运动。

7.根据权利要求6所述的伺服驱动器,其特征在于,所述拟合模块具体还用于:在 $K_{(m+1)} < K_m < K_{(m+1)} + 2^n$ 时,通过如下公式确定当前通讯周期对应的位置增量Delt_Pos_(m+1):

$$\text{Delt_Pos_}(m+1) = \frac{\text{Pos_}(m+1) - \text{Pos_Bk_}m}{K_{(m+1)} + 2^n - K_m}。$$

一种伺服驱动器的同步控制方法及伺服驱动器

技术领域

[0001] 本申请涉及运动控制技术领域,具体而言,涉及一种伺服驱动器的同步控制方法及伺服驱动器。

背景技术

[0002] 伺服系统是目前定位控制中使用较为广泛的系统,和步进系统相比具有控制精度高、转速快、带负载能力强等特点。伺服系统在定位控制中包含三个方面的设备:伺服电机、伺服驱动器以及控制器,控制器向伺服驱动器发送位置指令,伺服驱动器根据位置指令驱动伺服电机到达指定位置。

[0003] 伺服系统的位置控制方式包括轮廓位置模式和同步位置模式。在轮廓位置模式下,控制器间隔较长时间(通讯周期达几百毫秒)向伺服驱动器发送位置指令,由于通讯周期长,从一个位置指令到另一个位置指令的跨度较大,指令的点位不够密集,导致伺服插值出来的位置,与控制器期望的位置轮廓有很大差别,控制精度不够高;在同步位置模式下,控制器发送位置指令的间隔时间很短(通讯周期仅几个毫秒),可以让伺服插值出的位置轮廓与控制器期望的轮廓基本一致,但这要求每次指令到达的时间必须很精准,一个通讯周期内,必须有且仅有一次位置指令。

[0004] 目前,同步位置模式通常使用的是以太网控制自动化技术(EtherCAT)来实现控制器与伺服驱动器的通讯,在EtherCAT的通讯机制里,硬件有精确的同步时钟,所以可以保证,一个通讯周期内有且仅有一次位置指令;而对于其他不具有同步时钟的通讯方案,比如控制器局域网络(Controller Area Network,CAN)通讯或者Modbus通讯等,由于无法保证一个通讯周期有且仅有一次位置指令,所以一般只能用于轮廓位置模式,而无法用于同步位置模式。

发明内容

[0005] 本申请实施例的目的在于提供一种伺服驱动器的同步控制方法及伺服驱动器,使得不具有同步时钟的通讯方案也能够用于同步位置模式,且使伺服电机运行更加平稳。

[0006] 第一方面,本申请实施例提供一种伺服驱动器的同步控制方法,所述方法包括:在当前通讯周期的起始,获取控制器发来的位置指令中时间最近的位置指令,以及获取控制器对位置指令的第一计数值;判断所述位置指令所对应的第一计数值是否大于当前所述伺服驱动器对通讯周期的第二计数值;在所述第一计数值不大于所述第二计数值时,根据上一通讯周期的位置增量计算当前通讯周期对应的位置增量;根据所述位置增量控制伺服电机运动。

[0007] 在上述方案中,通过对第一计数值和第二计数值进行比较,能够准确判别控制器发送的位置指令是否正常,并在位置指令出现错位或者丢帧时,通过使用上一通讯周期的位置增量进行拟合计算,能够使得伺服驱动器获得相对平滑、连续的位置增量,从而使伺服电机平稳运行,这样一来,即便不具有同步时钟的通讯方案也可以用于同步位置模式。

[0008] 可选的,所述第一计数值和所述第二计数值均由n个比特的二进制数字组成,所述第一计数值和第二计数值在取值范围 $[0, 2^n - 1]$ 内循环取值,n为正整数。

[0009] 可选的,所述根据上一通讯周期的位置增量计算当前通讯周期对应的位置增量,包括:获取对当前通讯周期之前的每个通讯周期的位置增量进行累加后得到的累加值Pos_{Bk_m};在 $K_m < K_{(m+1)} \leq 2^n - 1$ 时,通过如下公式确定当前通讯周期对应的位置增量Delt_{Pos_(m+1)}:

$$[0010] \quad \text{Delt_Pos}_{(m+1)} = \frac{\text{Pos}_{(m+1)} - \text{Pos_Bk}_m}{K_{(m+1)} - K_m};$$

[0011] 其中,Pos_(m+1)为当前通讯周期起始时获取的时间最近的位置指令,K_(m+1)是与位置指令Pos_(m+1)对应的第一计数值,K_m是与位置指令Pos_m对应的第一计数值,Pos_m为上一通讯周期起始时获取的时间最近的位置指令。

[0012] 可选的,在获取对当前通讯周期之前的每个通讯周期的位置增量进行累加后得到的累加值Pos_{Bk_m}之后,所述方法还包括:

[0013] 在 $K_{(m+1)} < K_m < K_{(m+1)} + 2^n$ 时,通过如下公式确定当前通讯周期对应的位置增量Delt_{Pos_(m+1)}:

$$[0014] \quad \text{Delt_Pos}_{(m+1)} = \frac{\text{Pos}_{(m+1)} - \text{Pos_Bk}_m}{K_{(m+1)} + 2^n - K_m}。$$

[0015] 在上述两种实施方式中,在每个通讯周期备份保存位置增量的累加值,可以通过位置指令与备份的累加值对当前通讯周期的位置增量进行拟合。

[0016] 可选的,所述根据上一通讯周期的位置增量计算当前通讯周期对应的位置增量,包括:将上一通讯周期的位置增量确定为当前通讯周期对应的位置增量。

[0017] 在位置指令出现错位或丢帧时,可以直接使用上一通讯周期的位置增量作为当前通讯周期的位置增量,这样一来,连续的两个通讯周期的位置增量相同,因此伺服电机可以保持匀速运动,运行平稳、流畅,且计算量小。

[0018] 可选的,在判断所述位置指令所对应的第一计数值是否大于当前所述伺服驱动器对通讯周期的第二计数值之后,所述方法还包括:在所述第一计数值大于所述第二计数值时,根据所述位置指令控制伺服电机运动至所述位置指令所指示的目标位置。

[0019] 可选的,所述驱动伺服器与所述控制器之间的通讯使用CAN通信或者Modbus通信。

[0020] 第二方面,本申请实施例提供一种伺服驱动器,包括:获取模块,用于在当前通讯周期的起始,获取控制器发来的位置指令中时间最近的位置指令,以及获取控制器对位置指令的第一计数值;拟合模块,用于判断所述位置指令所对应的第一计数值是否大于当前所述伺服驱动器对通讯周期的第二计数值;并在所述第一计数值不大于所述第二计数值时,根据上一通讯周期的位置增量计算当前通讯周期对应的位置增量;控制模块,用于根据所述位置增量控制伺服电机运动。

[0021] 可选的,所述第一计数值和所述第二计数值均由n个比特的二进制数字组成,所述第一计数值和第二计数值在取值范围 $[0, 2^n - 1]$ 内循环取值,n为正整数;所述拟合模块具体用于:获取对当前通讯周期之前的每个通讯周期的位置增量进行累加后得到的累加值Pos_{Bk_m}

Bk_m;并在 $K_m < K_{(m+1)} \leq 2^n - 1$ 时,通过如下公式确定当前通讯周期对应的位置增量 $\text{Delt_Pos}_{(m+1)}$:

$$[0022] \quad \text{Delt_Pos}_{(m+1)} = \frac{\text{Pos}_{(m+1)} - \text{Pos_Bk}_m}{K_{(m+1)} - K_m};$$

[0023] 其中, $\text{Pos}_{(m+1)}$ 为当前通讯周期起始时获取的时间最近的位置指令, $K_{(m+1)}$ 是与位置指令 $\text{Pos}_{(m+1)}$ 对应的第一计数值, K_m 是与位置指令 Pos_m 对应的第一计数值, Pos_m 为上一通讯周期起始时获取的时间最近的位置指令。

[0024] 可选的,所述拟合模块具体还用于:在 $K_{(m+1)} < K_m < K_{(m+1)} + 2^n$ 时,通过如下公式确定当前通讯周期对应的位置增量 $\text{Delt_Pos}_{(m+1)}$:

$$[0025] \quad \text{Delt_Pos}_{(m+1)} = \frac{\text{Pos}_{(m+1)} - \text{Pos_Bk}_m}{K_{(m+1)} + 2^n - K_m}。$$

附图说明

[0026] 为了更清楚地说明本申请实施例的技术方案,下面将对本申请实施例中所需要使用的附图作简单地介绍,应当理解,以下附图仅示出了本申请的某些实施例,因此不应被看作是对范围的限定,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他相关的附图。

[0027] 图1为本申请实施例提供的伺服驱动器的同步控制方法的流程图;

[0028] 图2为按照本申请实施例提供的同步控制方法进行实施后各通讯周期获得的位置增量的变化示意图;

[0029] 图3为现有技术中各通讯周期的位置增量的变化示意图;

[0030] 图4为本申请实施例提供的伺服驱动器的示意图。

具体实施方式

[0031] 下面将结合本申请实施例中的附图,对本申请实施例中的技术方案进行描述。

[0032] 在同步位置模式下,控制器内部进行插值运算,并按照设定的通讯周期向伺服驱动器发送位置指令。这一模式要求伺服驱动器的时钟基准必须与控制器的时钟基准保持一致,并且通讯传输及位置信息采集周期保持稳定。但在实际的伺服系统中,控制器与伺服驱动器由于时钟偏移、通讯波动等因素,很难达到保持同步。

[0033] 如果要使用CAN通讯或者Modbus通讯实现同步位置模式,由于控制器和伺服驱动器的时钟不同步,连续长期运行,会导致位置指令错位,即出现:1、一个通讯周期内,控制器的位置指令应该只跳变一次,但实际上控制器的位置指令跳变两次甚至多次;2、两个连续的通讯周期内,控制器的位置指令应该是连续变化,但实际的位置指令与上一周期完全一致,甚至连续多个周期都保持完全一致;3、由于CAN总线物理层的问题,比如线接触不良,导致某一帧数据接收失败,出现漏帧。

[0034] 在伺服电机正常匀速运行过程中,位置增量应该保持一个固定值,但是由于上述指令错位问题,会导致某一通讯周期内的位置增量突变成0(甚至多个周期的位置增量均为

0),再下一个通讯周期内的位置增量突变成正常值的2倍(甚至多倍);伺服驱动器基于这样突变的位置增量,插值算出的每一个细分周期的小增量和速度前馈值也有突变,那么速度、电流也同样会出现突变,就会造成伺服电机抖动,具体表现为顿挫感、不流畅,严重影响控制性能。

[0035] 由此,本申请实施例提供一种伺服驱动器的同步控制方法,能够有效解决上述由于位置指令的错位带来的抖动问题,使得不具有同步时钟的通讯方案也能够用于同步位置模式,本实施例以CAN通讯为例对该方法进行具体介绍。

[0036] 首先,在通讯之前,在控制器端定义一个n个比特的指令同步计数器,该指令同步计数器用于对控制器生成的位置指令进行计数,获得第一计数值。其中,第一计数值由n位的二进制数字组成,并在取值范围 $[0, 2^n - 1]$ 内循环取值,n为正整数。每一个通讯周期内,控制器将会产生一个位置指令并向伺服驱动器发出,因此,控制器每产生一个位置指令,则指令同步计数器的值就加一,当计数值到达 $2^n - 1$ 后,指令同步计数器的值再变为0,以此循环。

[0037] 控制器与伺服驱动器连接在同一CAN总线上,控制器将待发送的位置指令和第一计数值添加到CAN报文中,并向CAN总线发送该CAN报文。伺服驱动器从CAN总线上读取到该CAN报文,同时获得控制器发出的位置指令以及控制器对位置指令的第一计数值。

[0038] 同时,在伺服驱动器端也定义一个n个比特的拟合计数器,该拟合计数器对通讯周期进行计数,获得第二计数值。伺服驱动器的内部定时器会统计本地时间,每10毫秒(ms)记作一个通讯周期(当然,通讯周期可以根据需求进行设置),这个时间是十分准确的,每过一个通讯周期,则拟合计数器计数一次,即拟合计数器的值就加一。第二计数值由n位的二进制数字组成,并在取值范围 $[0, 2^n - 1]$ 内循环取值,当计数值到达 $2^n - 1$ 后,拟合计数器的值再变为0,以此循环。

[0039] 本实施例中的控制器可以是可编程逻辑控制器(Programmable Logic Controller,PLC)或者单片机,也可以是专用的定位控制单元或模块,如三菱公司生产的型号为FX2N-1PG、FX2N-10GM、FX2N-20GM等的定位模块。

[0040] 然后,在通讯过程中,伺服驱动器根据第一计数值和第二计数值执行同步控制方法,请参阅图1,该方法包括如下步骤:

[0041] 步骤101:在当前通讯周期的起始,确定控制器发来的报文中时间最近的报文,获得报文中携带的位置指令和对应的第一计数值。

[0042] 控制器按照设定的通讯周期向CAN总线发送报文,伺服驱动器可从CAN总线上读取到该报文,同时获得报文中携带的位置指令和第一计数值。伺服驱动器在每个通讯周期的起始,首先确定从CAN总线上收到的报文中时间最近的报文,然后根据该报文中的位置指令确定当前通讯周期结束时伺服电机应当运动至的目标位置,在该位置指令未出现错位时,该位置指令所指示的绝对位置为当前通讯周期结束时的目标位置,在位置指令出现错位或者丢帧等情况时,则需要先拟合获得当前通讯周期内的位置增量,再确定其目标位置。

[0043] 步骤102:判断该位置指令对应的第一计数值是否大于当前伺服驱动器的第二计数值,若不大于,转至步骤103,若大于,转至步骤105。

[0044] 在每个通讯周期的起始,伺服驱动器读取控制器的指令同步计数器的值,并和伺服驱动器在当前通讯周期的拟合计数器的值进行对比:

[0045] (1)如果指令同步计数器的值大于拟合计数器的值,说明位置指令正常。在正常情

况下,需要先由控制器产生位置指令,之后才能够控制伺服电机运动,所以指令同步计数器的值应当超前于拟合计数器的值,即正常情况下,指令先到,所以应当具有这样的关系: $N=M+1$,其中, N 为第一计数值, M 为第二计数值。

[0046] (2) 如果指令同步计数器的值小于或者等于拟合计数器的值,说明位置指令存在位置错位或者丢帧。如果一段时间内,伺服驱动器没有收到位置指令,那么得到的 N 的值不变,但每隔10ms, M 的值就加一,所以在 $N \leq M$ 时,表明出现了指令错位或者丢帧,此时,需要使用上一通讯周期获得的位置增量来拟合当前通讯周期的位置增量,即执行步骤103。

[0047] 步骤103:根据上一通讯周期的位置增量计算当前通讯周期对应的位置增量。

[0048] 在一种实施方式中,伺服驱动器根据该位置指令以及伺服驱动器对之前每一通讯周期的位置增量累加后获得的累加值,确定当前通讯周期对应的位置增量。

[0049] 位置指令变化后,伺服驱动器在每个通讯周期的起始时计算当前通讯周期对应的位置指令的位置增量,并将获得的位置增量与之前每个通讯周期的位置增量进行累加,获得累加值,并将该累加值作为备份。在下一通讯周期的起始时,将下一通讯周期计算出的位置增量与该累加值进行累加,然后获得新的累加值,累加值随通讯周期的变化不断更新。第 m 个通讯周期的位置增量的累加值记作 Pos_Bk_m ,例如,在第1个通讯周期的起始,将计算出的第1个通讯周期的位置增量与第0个通讯周期的位置增量累加,获得累加值 Pos_Bk_1 ,在第2个通讯周期的起始,将计算出的第2个通讯周期的位置增量与累加值 Pos_Bk_1 累加,获得累加值 Pos_Bk_2 。

[0050] 在第 m 个通讯周期的起始,伺服驱动器获得控制器的位置指令 Pos_m 和指令同步计数器的值 K_m ;紧接着,在第 $m+1$ 个通讯周期的起始,伺服驱动器获得控制器的位置指令 $Pos_{(m+1)}$ 和指令同步计数器的值 $K_{(m+1)}$ 。需要注意的是, $Pos_{(m+1)}$ 看起来是 Pos_m 的下一帧,但因为可能发生了指令错位、漏帧,所以 $Pos_{(m+1)}$ 和 Pos_m 之间实际上可能间隔了 x 帧,而这个 x 会体现在指令同步计数器的第一计数值里,也就是说: $x=K_{(m+1)}-K_m$,这样通过指令同步计数器的差值,可以间接知道位置指令实际跳变了多少帧。

[0051] 可以确定的是,在第 $m+1$ 个通讯周期的起始,伺服驱动器可以获得这些值:第 m 个通讯周期获得的位置增量的累加值 Pos_Bk_m ,位置指令 Pos_m ,第一计数 K_m ,以及第 $m+1$ 个通讯周期获得的位置指令 $Pos_{(m+1)}$,第一计数值 $K_{(m+1)}$ 。然后,伺服驱动器根据如下方式拟合第 $m+1$ 个通讯周期的位置增量:

[0052] (1) 若指令同步计数器的值未过零点,即 $K_m < K_{(m+1)} \leq 2^n - 1$ 时,本次通讯周期的位置增量 $Delt_Pos_{(m+1)}$ 为:

$$[0053] \quad Delt_Pos_{(m+1)} = \frac{Pos_{(m+1)} - Pos_Bk_m}{K_{(m+1)} - K_m}$$

[0054] (2) 若指令同步计数器的值刚好过零点,即 $K_{(m+1)} < K_m < K_{(m+1)} + 2^n$ 时,本次通讯周期的位置增量 $Delt_Pos_{(m+1)}$ 为:

$$[0055] \quad Delt_Pos_{(m+1)} = \frac{Pos_{(m+1)} - Pos_Bk_m}{K_{(m+1)} + 2^n - K_m}$$

[0056] 在计算出本次通讯周期的位置增量 $Delt_Pos_{(m+1)}$ 之后,将位置增量 $Delt_Pos_{(m+1)}$ 与上一通讯周期的位置增量累加值 Pos_Bk_m 相加,获得第 $m+1$ 个通讯周期的累加值

Pos_Bk_(m+1), Pos_Bk_(m+1)可以用于第m+2个通讯周期的位置增量的计算。

[0057] 在另一种实施方式中,位置指令变化后,伺服驱动器在每个通讯周期的起始时计算当前通讯周期对应的位置指令的位置增量,并将获得的位置增量作为备份,第m个通讯周期的位置增量记作Delt_Pos_m。在下一通讯周期的起始时,计算出下一通讯周期对应的位置增量,并对备份的位置增量进行更新。

[0058] 伺服驱动器在判断位置指令出现错位或者丢帧后,直接使用上一通讯周期的位置增量来拟合当前通讯周期的位置增量,即:

[0059] $Delt_Pos_(m+1) = Delt_Pos_m$

[0060] 在步骤103之后,执行步骤104:根据获得的位置增量控制伺服电机运动。

[0061] 伺服驱动器对通讯周期内的位置增量进行插值计算,计算出每一个细分周期的小增量,并根据每个细分周期的小增量控制伺服电机运动。例如,控制器每10个ms发送一次位置指令,而由于伺服驱动器内部运算的速度远高于10个ms,可能每100个微秒(μs)计算一次,所以在一个通讯周期里面可以插值计算一百步,从而对位置增量再次细分,然后根据每一步的小增量控制伺服电机运动。

[0062] 步骤105:根据该位置指令控制伺服电机运动至位置指令所指示的目标位置。

[0063] 在第一计数值大于第二计数值时,表明位置指令未出现错位,此时,直接根据控制器的位置指令进行插值计算即可,然后根据插值出的每个细分周期的小增量控制伺服电机运动。

[0064] 在具体的实施过程中,请参阅图2,图2示出了前7个通讯周期内指令同步计数器和拟合计数器的计数值的变化以及每一通讯周期内位置增量的变化。可以看出,各通讯周期的同步控制过程包括:

[0065] A、在拟合计数器的值从0到1对应的通讯周期内,伺服驱动器收到控制器发来的位置指令,第一计数值为1,在下一个通讯周期(从1到2对应的通讯周期)的起始,伺服驱动器将指令同步计数器的计数值1与拟合计数器的计数值0进行比较,由于 $1 > 0$,因此按照位置指令控制伺服电机移动;

[0066] B、在拟合计数器的值从1到2对应的通讯周期内,伺服驱动器未收到控制器发来的位置指令,第一计数值仍为1,在下一个通讯周期(从2到3对应的通讯周期)的起始,伺服驱动器将指令同步计数器的计数值1与拟合计数器的计数值1进行比较,由于 $1 = 1$,因此根据上一通讯周期(从1到2对应的通讯周期)的位置增量拟合本次通讯周期(从2到3对应的通讯周期)的位置增量,然后按照获得的位置增量控制伺服电机移动;

[0067] C、在拟合计数器的值从3到4对应的通讯周期内,伺服驱动器收到控制器发来的两个位置指令,第一计数值分别为3和4,在下一个通讯周期(从4到5对应的通讯周期)的起始,伺服驱动器将指令同步计数器的计数值4与拟合计数器的计数值3进行比较,由于 $4 > 3$,因此按照位置指令控制伺服电机移动。

[0068] 需要说明的是,以上述过程A为例,在第一次位置指令到来之前,指令同步计数器的值为0,且拟合计数器的值也为0(因为通讯还未开始),在收到位置指令后的下一次通讯周期的起始,指令同步计数器的值为1,拟合计数器的值仍保持为0,在完成计数值的比较和位置增量的插值计算后,伺服驱动器再将拟合计数器的值加一,也就是说,在将两个计数值进行比较时,拟合计数器的值实际上应当是上一周期的计数值。

[0069] 从图2所示的位置增量的曲线可以看出,按照本实施例提供的同步控制方法进行实施,多个通讯周期内的位置增量没有出现突变,始终保持平滑,因此并不会导致伺服电机的抖动。而在现有技术中,如图3所示,同样的位置指令,由于伺服驱动器没有对每个通讯周期的位置增量进行拟合,而是直接使用位置指令所指示的目标位置控制伺服电机运动,可以看出,位置增量的突变异常明显。本实施例的同步控制方法相比现有技术,具有十分显著的效果。

[0070] 应当理解,伺服电机在一个通讯周期内从起始到结束时位置的增量具体表现为运动的速度,在伺服电机正常匀速运行过程中,经过拟合计算后,连续的两个通讯周期的位置增量相同,因此伺服电机的运动速度仍然保持匀速运动,运动流畅,不存在抖动。当然,本实施例提供的同步控制方案也可以不限于匀速的拟合,技术人员可以根据上述匀速的拟合方案,在匀加速运动或者变加速运动中,按照固定的加速度或者变化的加速度对每个存在指令错位的通讯周期的位置增量进行拟合。例如,在通讯周期的起始,伺服驱动器可以计算之前的通讯周期的位置增量的增量,即计算伺服电机运动的加速度,在加速度恒定时,按照上一个通讯周期的加速度进行拟合;而在加速度变化时,这属于一种更加复杂的轨迹规划,本实施例在此不做详述。

[0071] 综上所述,本申请实施例提供的同步控制方法,通过设置指令同步计数器和拟合计数器,并备份保存通讯周期的位置增量或位置增量累加值,可以通过控制算法来模拟同步过程,使得伺服驱动器获得了相对平滑、连续的位置增量,且使伺服电机的运行更加平稳,极大改善了现有技术中存在的问题。

[0072] 基于同一发明构思,本申请实施例还提供一种伺服驱动器,请参阅图4,该伺服驱动器包括:

[0073] 获取模块201,用于在当前通讯周期的起始,获取控制器发来的位置指令中时间最近的位置指令,以及获取控制器对位置指令的第一计数值;

[0074] 拟合模块202,用于判断所述位置指令所对应的第一计数值是否大于当前所述伺服驱动器对通讯周期的第二计数值;并在所述第一计数值不大于所述第二计数值时,根据上一通讯周期的位置增量计算当前通讯周期对应的位置增量;

[0075] 控制模块203,用于根据所述位置增量控制伺服电机运动。

[0076] 可选的,第一计数值和第二计数值均由n个比特的二进制数字组成,第一计数值和第二计数值在取值范围 $[0, 2^n - 1]$ 内循环取值,n为正整数。

[0077] 可选的,拟合模块202具体用于:获取对当前通讯周期之前的每个通讯周期的位置增量进行累加后得到的累加值 Pos_Bk_m ;并在 $K_m < K_(m+1) \leq 2^n - 1$ 时,通过如下公式确定当前通讯周期对应的位置增量 $Delt_Pos_(m+1)$:

$$[0078] \quad Delt_Pos_(m+1) = \frac{Pos_(m+1) - Pos_Bk_m}{K_(m+1) - K_m};$$

[0079] 其中, $Pos_(m+1)$ 为当前通讯周期起始时获取的时间最近的位置指令, $K_(m+1)$ 是与位置指令 $Pos_(m+1)$ 对应的第一计数值, K_m 是与位置指令 Pos_m 对应的第一计数值, Pos_m 为上一通讯周期起始时获取的时间最近的位置指令。

[0080] 可选的,拟合模块202具体还用于:在 $K_(m+1) < K_m < K_(m+1) + 2^n$ 时,通过如下公式

确定当前通讯周期对应的位置增量 $\text{Delt_Pos}_{(m+1)}$ ：

$$[0081] \quad \text{Delt_Pos}_{(m+1)} = \frac{\text{Pos}_{(m+1)} - \text{Pos_Bk}_m}{K_{(m+1)} + 2^n - K_m}。$$

[0082] 可选的,拟合模块202具体还用于:将上一通讯周期的位置增量确定为当前通讯周期对应的位置增量。

[0083] 可选的,控制模块203还用于:在所述第一计数值大于所述第二计数值时,根据所述位置指令控制伺服电机运动至所述位置指令所指示的目标位置。

[0084] 上述提供的伺服驱动器与前一方法实施例的基本原理及产生的技术效果相同,为简要描述,本实施例部分未提及之处,可参考上述的方法实施例中的相应内容,在此不做赘述。

[0085] 在本申请所提供的实施例中,应该理解到,所揭露装置和方法,可以通过其它的方式实现。以上所描述的装置实施例仅仅是示意性的,例如,所述单元的划分,仅仅为一种逻辑功能划分,实际实现时可以有另外的划分方式,又例如,多个单元或组件可以结合或者可以集成到另一个系统,或一些特征可以忽略,或不执行。另一点,所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些通信接口,装置或单元的间接耦合或通信连接,可以是电性,机械或其它的形式。

[0086] 另外,作为分离部件说明的单元可以是或者也可以不是物理上分开的,作为单元显示的部件可以是或者也可以不是物理单元,即可以位于一个地方,或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

[0087] 再者,在本申请各个实施例中的各功能模块可以集成在一起形成一个独立的部分,也可以是各个模块单独存在,也可以两个或两个以上模块集成形成一个独立的部分。

[0088] 需要说明的是,功能如果以软件功能模块的形式实现并作为独立的产品销售或使用,可以存储在一个计算机可读取存储介质中。基于这样的理解,本申请的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的部分可以以软件产品的形式体现出来,该计算机软件产品存储在一个存储介质中,包括若干指令用以使得一台计算机设备(可以是个人计算机,服务器,或者网络设备等)执行本申请各个实施例所述方法的全部或部分步骤。而前述的存储介质包括:U盘、移动硬盘、只读存储器(Read-Only Memory,ROM)随机存取存储器(Random Access Memory,RAM)、磁碟或者光盘等各种可以存储程序代码的介质。

[0089] 在本文中,诸如第一和第二等之类的关系术语仅仅用来将一个实体或者操作与另一个实体或操作区分开来,而不一定要求或者暗示这些实体或操作之间存在任何这种实际的关系或者顺序。

[0090] 以上所述仅为本申请的实施例而已,并不用于限制本申请的保护范围,对于本领域的技术人员来说,本申请可以有各种更改和变化。凡在本申请的精神和原则之内,所作的任何修改、等同替换、改进等,均应包含在本申请的保护范围之内。

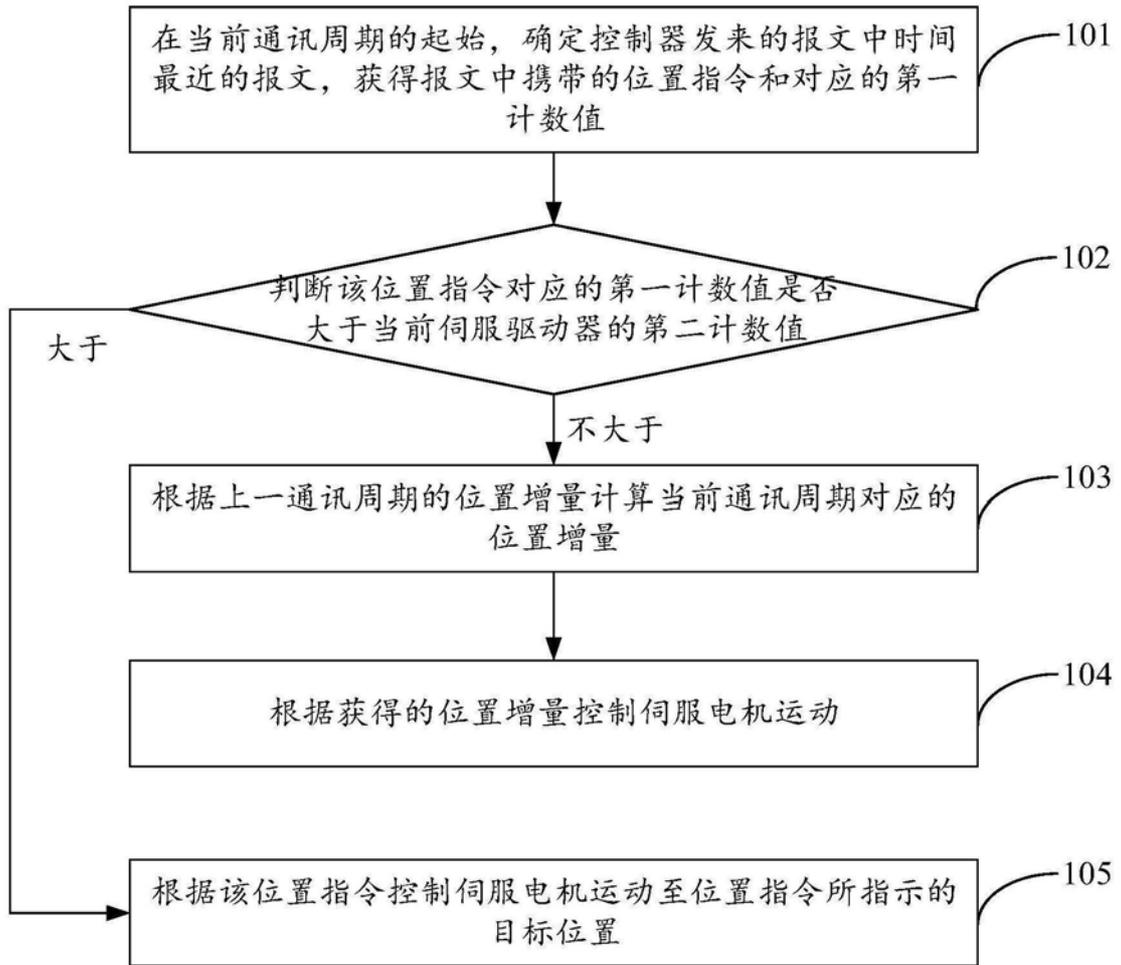


图1

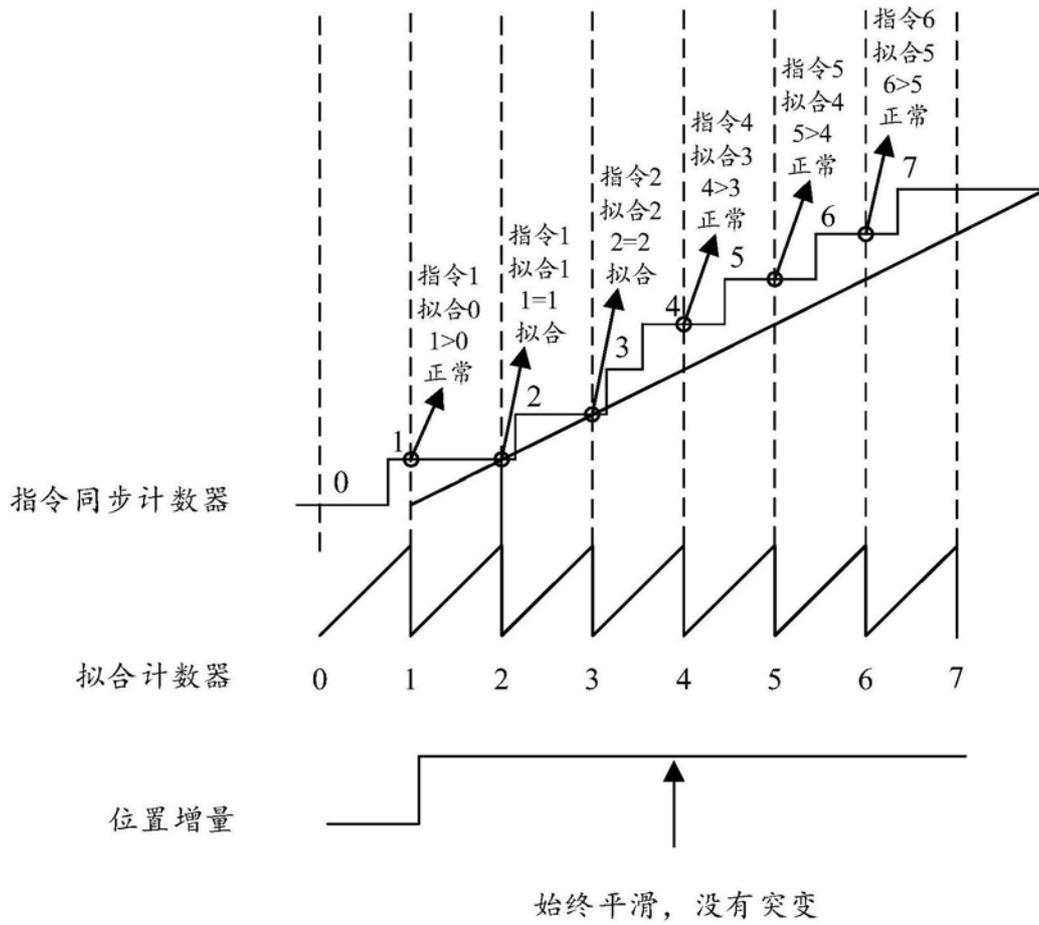


图2

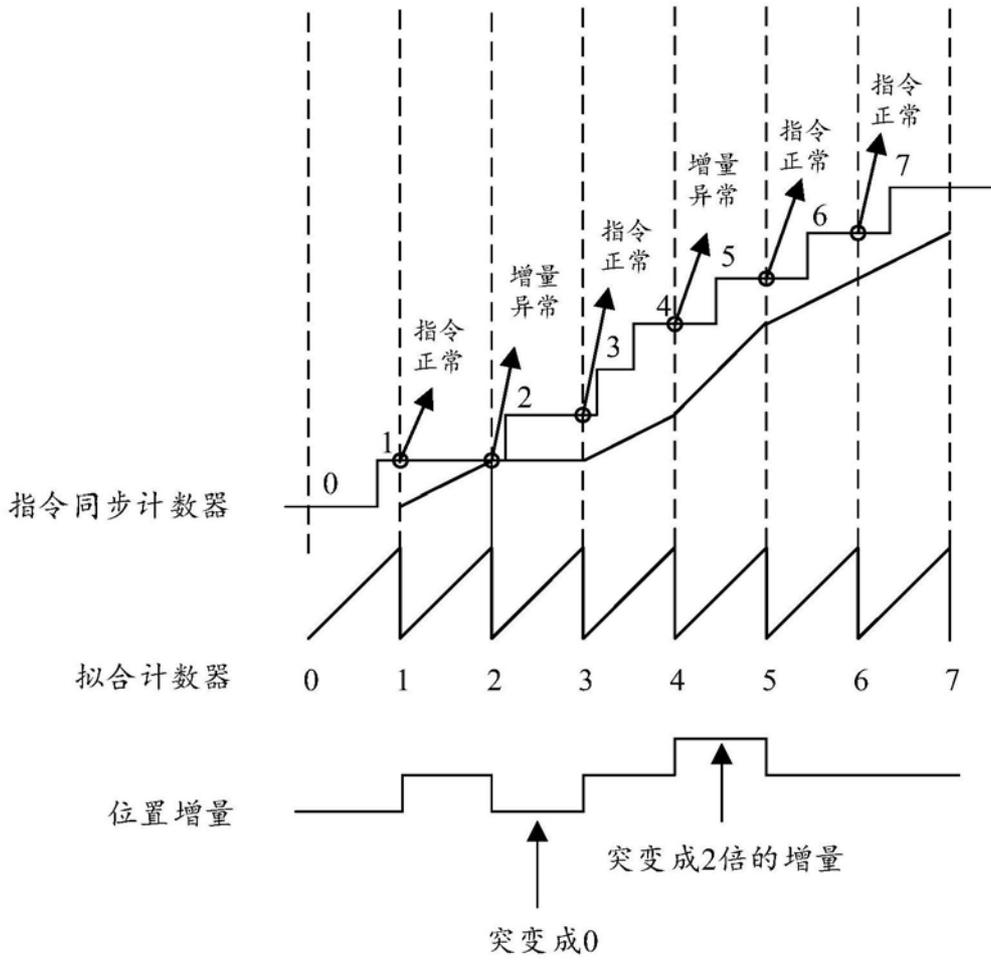


图3

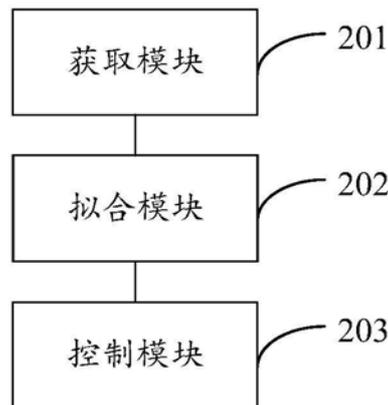


图4