

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

G05B 19/42 (2006.01)

G06F 19/00 (2006.01)



[12] 发明专利说明书

专利号 ZL 01809674.3

[45] 授权公告日 2008 年 2 月 13 日

[11] 授权公告号 CN 100368950C

[22] 申请日 2001.5.10 [21] 申请号 01809674.3

[30] 优先权

[32] 2000.5.16 [33] US [31] 60/204,609

[86] 国际申请 PCT/US2001/015051 2001.5.10

[87] 国际公布 WO2001/088647 英 2001.11.22

[85] 进入国家阶段日期 2002.11.18

[73] 专利权人 杨伯翰大学

地址 美国犹他州

[72] 发明人 马克·S·埃文斯

沃尔特·爱德华·雷德

查尔斯·格雷戈里·詹森

波特·C·巴西特

吉里什·C·吉米雷

科里·林恩·麦克布赖德

乔纳森·爱德华·博斯利

[56] 参考文献

EP0706103A1 1996.4.10

CN1037997A 1989.11.29

US5703782A 1997.12.30

WO98/37467A1 1998.8.27

US5907494A 1999.5.25

US5506787A 1996.4.9

审查员 沈乐平

[74] 专利代理机构 中科专利商标代理有限责任公司

代理人 王 玮

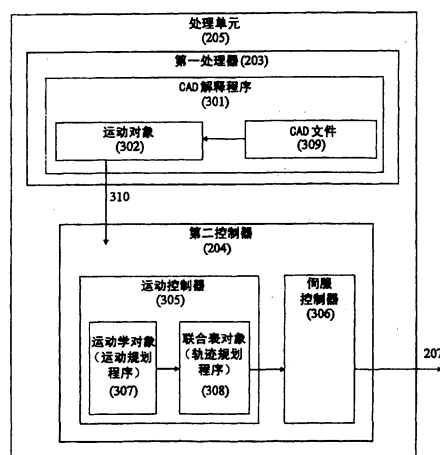
权利要求书 6 页 说明书 15 页 附图 6 页

[54] 发明名称

通过直接传送加工数据控制机床的方法和系统

[57] 摘要

本发明公开了一种方法和系统，用于从生产系统(204)向用于控制伺服驱动的机器的控制系统(205)直接传输运动路线数据(310)。例如，用于加工坯料材料的机床，所述数据生成系统包括一个计算机辅助设计软件包(301)。用于导引机床的工具构件的工具路线数据可以通过 CAD 软件(302)的操作接口从一个设计文件中提取，该设计文件由所述 CAD 软件打开。另外，所述设计文件可用于规定运动路线而不是最后的产品几何形状。所述 CAD 解释程序应用程序从所述设计文件中提取运动路线数据，消除了将所述数据翻译成用于控制所述机床的另外一种形式或者中间形式数据的需求。



1. 一种用于将控制系统与伺服驱动的机器连接的系统，包括：
第一处理器，执行产生运动路线数据而不使用M&G码的设计应用程序；
与所述第一处理器连接的第二处理器，所述第二处理器执行从所述第一处理器上的所述设计应用程序存取所述运动路线数据的运动控制软件；
和

其中所述运动控制软件根据所述运动路线数据控制伺服驱动的机器，并且所述运动控制软件是面向对象的，以致所述运动控制软件适用于允许使控制系统与不同的伺服驱动的机器连接。

2. 根据权利要求1所述的系统，其中伺服驱动的机器是根据设计规格加工坯料材料的机床。

3. 根据权利要求1所述的系统，其中所述设计应用程序是计算机辅助设计/计算机辅助制造软件包。

4. 根据权利要求1所述的系统，其中运动控制软件使用所述设计应用程序的应用程序接口从所述设计应用程序提取所述运动路线数据。

5. 根据权利要求1所述的系统，其中所述设计应用程序通过所述运动控制软件的应用程序接口向所述运动控制软件传送所述运动路线数据。

6. 根据权利要求1所述的系统，其中所述系统还包括一个显示装置和与所述第一处理器连接的用户输入装置，用于使用所述设计应用程序来产生、检查或编辑所述运动路线数据。

7. 根据权利要求1所述的系统，其中：

所述设计应用程序产生包含所述运动路线数据的运动对象，所述运动路线数据定义伺服驱动的机器的工具头要沿其移动的运动路线；和

所述设计应用程序将所述运动对象传递给所述第二处理器上的所述运动控制软件。

8. 根据权利要求7所述的系统，其中所述运动对象包括：

一个速率 对象，该速率对象确定进给速度和速率，用于控制伺服驱动的机器的构件；

一个速度分布产生器对象，用于控制所述运动路线中的速度跃变；
一个路线对象，包含定义每个运动路线的信息。

9. 根据权利要求1所述的系统，其中：

所述运动控制软件通过所述设计应用程序存取运动路线数据，所述运动路线数据定义伺服驱动的机器的工具头要沿其移动的运动路线；和
所述运动控制软件产生包含所述运动路线数据的运动对象。

10. 根据权利要求1所述的系统，其中由所述设计应用程序输出的所述运动路线数据以笛卡尔坐标系统规定工具路线，所述运动控制软件将工具路线翻译成用于驱动所述伺服驱动的机器的电机的伺服控制信号。

11. 根据权利要求1所述的系统，其中：

所述运动控制软件包括运动控制器应用程序和伺服控制器应用程序；
所述运动控制器应用程序将所述运动路线数据映射到由所述伺服驱动的机器定义的坐标系统中；

所述伺服控制器应用程序根据来自所述运动控制器应用程序的所述映射的运动路线数据产生传输给所述伺服驱动的机器的伺服机构的控制信号。

12. 根据权利要求1所述的系统，其中所述第一处理器和所述第二处理器在多处理器工作站中连接。

13. 一种将控制系统与伺服驱动的机器连接的方法，包含：

执行第一处理器上的设计应用程序，所述设计应用程序产生用于所述伺服驱动的机器的运动路线数据，所述运动路线数据是未使用M&G码产生的；

执行第二处理器上的运动控制软件，所述运动控制软件以与所述设计应用程序输出的相同的形式接收所述运动路线数据；和

根据来自所述设计应用程序的所述运动路线数据，用所述运动控制软件控制所述伺服驱动的机器。

14. 根据权利要求13所述的方法，进一步包括在不同的处理器上执行所述设计应用程序和所述运动控制软件。

15. 根据权利要求13所述的方法，进一步包括用所述伺服驱动的机器加工坯料材料以制造由所述运动路线数据定义的产品，该伺服驱动的机器

为机床。

16. 根据权利要求13所述的方法，其中所述设计应用程序是计算机辅助设计/计算机辅助制造软件包。

17. 根据权利要求13所述的方法，进一步包括：

使用计算机辅助设计软件包产生其中规定了工具路线的文件而不是最后的产品几何形状；和

用所述设计应用程序从所述文件中产生所述运动路线数据。

18. 根据权利要求13所述的方法，进一步包括用所述设计应用程序编辑所述运动路线数据。

19. 根据权利要求13所述的方法，进一步包括用所述设计应用程序检查所述运动路线数据。

20. 根据权利要求13所述的方法，进一步包括：

用所述设计应用程序产生一个运动对象，该运动对象包含所述运动路线数据，所述运动路线数据定义运动路线，所述伺服驱动的机器的构件沿所述运动路线移动；和

将所述运动对象传递到所述运动控制软件。

21. 根据权利要求13所述的方法，其中由所述设计应用程序输出的所述运动路线数据以笛卡尔坐标系统规定工具路线，所述方法还包括将所述工具路线翻译成用于驱动所述伺服驱动的机器的电机的伺服控制信号。

22. 一种将控制系统与伺服驱动的机器连接的系统，包括：

产生用于伺服驱动的机器的运动路线数据的装置，所述运动路线数据是未使用M&G码产生的，所述装置用于产生包括第一处理器的运动路线数据；

用于执行运动控制软件的装置，该运动控制软件以与由产生所述运动路线数据的所述装置输出的相同的形式接收所述运动路线数据，所述装置用于执行包括第二处理器的运动控制软件；和

根据所述运动路线数据控制所述伺服驱动的机器的装置。

23. 根据权利要求22所述的系统，其中所述伺服驱动的机器包括用于加工坯料材料以制造由所述运动路线数据定义的产品装置。

24. 根据权利要求22所述的系统，其中产生运动路线数据的所述装置

包括计算机辅助设计 / 计算机辅助制造软件包。

25. 根据权利要求22所述的系统，还包括检查和编辑所述运动路线数据的装置。

26. 一种将控制系统与伺服驱动的机器连接的系统，包括：

执行设计应用程序的第一处理器，所述设计应用程序产生运动路线数据而不使用M&G码；

与所述第一处理器连接的第二处理器，所述第二处理器执行运动控制软件，所述运动控制软件通过所述运动控制软件的应用程序接口从所述第一处理器上的所述设计应用程序接收所述运动路线数据；和

其中所述运动控制软件根据所述运动路线数据控制伺服驱动的机器。

27. 根据权利要求26所述的系统，其中伺服驱动的机器是根据设计规格加工坯料材料的机床。

28. 根据权利要求26所述的系统，其中所述设计应用程序是计算机辅助设计/计算机辅助制造软件包。

29. 根据权利要求26所述的系统，其中所述系统还包括一个显示装置和与所述第一处理器连接的用户输入装置，用于使用所述设计应用程序来产生、检查或编辑所述运动路线数据。

30. 根据权利要求26所述的系统，其中：

所述设计应用程序产生包含所述运动路线数据的运动对象，所述运动路线数据定义伺服驱动的机器的工具头要沿其移动的运动路线；和

所述设计应用程序将所述运动对象传递给所述第二处理器上的所述运动控制软件。

31. 根据权利要求30所述的系统，其中所述运动对象包括：

一个速率对象，该速率对象确定进给速度和速率，用于控制伺服驱动的机器的构件；

一个速度分布产生器对象，用于控制所述运动路线中的速度跃变；

一个路线对象，包含定义每个运动路线的信息。

32. 根据权利要求26所述的系统，其中由所述设计应用程序输出的所述运动路线数据以笛卡尔坐标系统规定工具路线，所述运动控制软件将工具路线翻译成用于驱动所述伺服驱动的机器的电机的伺服控制信号。

33. 根据权利要求26所述的系统，其中：

所述运动控制软件包括运动控制器应用程序和伺服控制器应用程序；

所述运动控制器应用程序将所述运动路线数据映射到由所述伺服驱动的机器定义的坐标系统中；

所述伺服控制器应用程序根据来自所述运动控制器应用程序的所述运动路线数据产生传输给所述伺服驱动的机器的伺服机构的控制信号。

34. 根据权利要求 26 所述的系统，其中所述第一处理器和所述第二处理器在多处理器工作站中连接。

35. 一种用于将控制系统与伺服驱动的机器连接的系统，包括：

执行设计应用程序的第一处理器，所述设计应用程序产生包含运动路线数据的运动对象，所述运动路线数据是未使用M&G码产生的，所述运动路线数据定义伺服驱动的机器的工具头要沿其移动的运动路线；

与所述第一处理器连接的第二处理器，所述第二处理器执行从所述第一处理器上的所述设计应用程序接收所述运动对象的运动控制软件；和

其中所述运动控制软件根据所述运动路线数据控制伺服驱动的机器。

36. 根据权利要求35所述的系统，其中伺服驱动的机器是根据设计规格加工坯料材料的机床。

37. 根据权利要求 35 所述的系统，其中所述设计应用程序是计算机辅助设计/计算机辅助制造软件包。

38. 根据权利要求 35 所述的系统，其中所述系统还包括一个显示装置和与所述第一处理器连接的用户输入装置，用于使用所述设计应用程序来产生、检查或编辑所述运动路线数据。

39. 根据权利要求35所述的系统，其中所述运动对象包括：

一个速率对象，该速率对象确定进给速度和速率，用于控制伺服驱动的机器的构件；

一个速度分布产生器对象，用于控制所述运动路线中的速度跃变；

一个路线对象，包含定义每个运动路线的信息。

40. 根据权利要求 35 所述的系统，其中由所述设计应用程序输出的所述运动路线数据以笛卡尔坐标系统规定工具路线，所述运动控制软件将工具路线翻译成用于驱动所述伺服驱动的机器的电机的伺服控制信号。

41. 根据权利要求35所述的系统，其中：

所述运动控制软件包括运动控制器应用程序和伺服控制器应用程序；

所述运动控制器应用程序将所述运动路线数据映射到由所述伺服驱动的机器定义的坐标系中；

所述伺服控制器应用程序根据来自所述运动控制器应用程序的所述运动路线数据产生传输给所述伺服驱动的机器的伺服机构的控制信号。

42. 根据权利要求 35 所述的系统，其中所述第一处理器和所述第二处理器在多处理器工作站中连接。

通过直接传送加工数据控制机床的方法和系统

技术领域

本发明涉及数字地连接机器人、机床或其它伺服驱动的机器的领域，这些设备具有提供用于控制伺服驱动的机器的数据的系统。本发明还涉及用数字控制的或计算机控制的机床由适当材料制造设备零件的领域，这种制造是根据要被制造的零件的技术要求粗加工和/或精加工初始材料片。更具体地说，本发明涉及一种通过把定义要制造的工件的加工数据从CAD/CAM应用程序直接传递给机器控制器来控制机床的操作的新范例。

背景技术

在现代工业中有许多机械具有活动部件，这些活动部件由伺服电动机驱动。这些机械包括工业机器人、坐标测量机、机床等等。这些类别的机械在这里总称为“伺服驱动的机械。”

通常，主系统向伺服控制器提供数据，伺服控制器控制伺服电动机适当地运转和移动伺服驱动的机器的部件。机床是这种伺服操作的机械的突出例子，本发明特别涉及这种机床。

当这些机械最初被开发时或者在有限的基础上被制造时，专用于这些机械的部件必须由适当的材料定制。这通常通过“加工”过程来完成，这是一个根据需要将一件或一块材料机械地切割、磨削、钻孔和精加工以生产所需要的零件的过程。用于将原材料转变成规定零件的机器在此之后被称为“机床”。现代的机床高度复杂化并被精确地控制，以便根据精密的技术要求生产零件。大多数这种现代机床由处理定义要制造的零件的精确数字指令的计算机控制。

第一台数控机床为1952年建造的3轴线铣床。使用了数字在一个三轴线系统中规定铣刀头的移动以便加工需要的零件。然而，随着加工的发展，根据规范数字用手机械地设定所述机床。随着后来电子技术的发展，“数

控”机床被越来越多地配备精密电子设备以自动进行加工处理。如今，数控机床包括计算机处理器和电子存储单元（RAM和ROM）。因而，如今的机床是电脑控制的机床，称为“计算机数控机床”。

早期的计算机控制机床从穿孔纸带接受输入。记录在这些纸带上的指令是输入/输出（“I/O”）指令（例如打开/关闭纺锤或冷却剂，设置缺省进料量和速度，等等。）和运动指令（例如，以直线或圆形的方式移到一些x y z位置）。写这些指令的语言被称为M&G码。可以认为M&G码类似于用于个人电脑的汇编语言；它是用于机床的最低水平的编程，并被所有机床普遍接受，而与构造或型号无关。即使在大约五十多年前制造出了第一台数控机器，但是我们仍然使用这种相同的仅有较小修的M&G码语言来对机床编程。

经过了50年，许多公司一直在努力地建造理想的机床控制器。这些控制器包含常规的硬件元件和高度专门化各不相同的操作系统。这些控制器中有许多已经是专有的封闭系统。虽然用于这些控制系统的硬件的开放式体系结构可能刚刚出现，但是如今的机床在很大程度上使用已经有几十年历史的技术。专有控制继续处在机床硬件控制器技术，而对机器的编程只能通过M&G码环境来执行。

M&G代码在一个单个的文件或程序中使用文字，或单独的指令，和模块，或者行。读出一行中的所有指令，并且同时执行所有指令。为了生成M&G程序，必须为所有的刀具路径计算所有的几何形状，例如，在加工所需零件时机床的刀尖通过的线段和弧线。从这种几何形状计算出运动和I/O指令。然后这些指令必须被写成M&G码，所述代码必须输入到控制器中。最初，必需用手将M&G码输入到所述控制器，或者打孔在聚酯薄膜或者纸带上，随后将聚酯薄膜或者纸带送到控制器上的一个阅读器中。即使现在，为了编写一个加工复杂零件的M&G程序也是一项漫长和乏味的任务。

在50年代末和60年代初，出现了许多新的更灵活的编程语言。自动编程工具（“APT”）就是这些新语言中最先出现的。使用APT，工程师可以规定机床的几何形状，刀尖的几何形状，和控制机床的I/O的加工参数。然而，由于现有机器系统的专用性和不一致性，必须把用APT生成的指令最终翻译成更基本的M&G码。APT应用程序根据APT程序中的指令执行计算以

生成一个刀具位置 (“CL”) 文件。CL文件是ASCII码文件，该文件包含机床的刀尖在加工过程中必须通过的路径的几何形状。然后使用一种被称为后处理程序的程序将该CL文件翻译成M&G码。

在60年代和70年代，随着计算机变得更低廉和更强大，更多机械设计采用计算机辅助设计 (“CAD”) 程序进行。计算机辅助制造 (“CAM”) 程序允许用户使用CAD程序中规定的几何形状来筹划出用于对机床进行编程的刀具路径。一般认为CAD，CAM和CAD/CAM程序在此是属于一般种类的软件包，通常称为“设计应用程序”。以CL或APT文件的形式输出由CAM应用程序定义的刀具路径，在将该文件传送给机器控制器前必须将其翻译成M&G代码。这些作法仍然是如今使用的标准。

图1中示出使用CAD/CAM应用程序对机床进行编程的典型过程。如图1中所示，CAD/CAM应用程序在工作站 (100) 上运行。使用CAD/CAM应用程序，生成一个CAD/CAM文件 (101)，这个文件 (101) 规定了机床和要被制造的零件的几何形状。包含由CAD/CAM或者类似软件包生成的设计资料的文件，诸如文件 (101) 在这里统称为“设计文件”。

然后CAD/CAM应用程序或者也许是一种独立的应用程序使用CAD/CAM文件计算用于机床端部在整个加工过程中的刀具定位数据并生成CL文件 (102)。CL文件 (102) 是一个ASCII码文件。然后CL文件 (102) 可以被译成M&G码 (103)，M&G码也是一个ASCII码文件。从CL文件 (102) 翻译成M&G码 (103) 通常由一个后处理程序执行。然后将M&G码 (103) 传输到所述机床控制器 (104)，该机床控制器在加工规定零件的过程中使用该码来控制所述机床 (未示出)。

由CL文件 (102) 从CAD/CAM软件的输出开始，每个CL文件都与源CAD/CAM文件 (101) 单方向相关联。因而，任何对源CAD/CAM文件 (101) 的修改不会反映在一个输出的CL文件 (102) 中，需要产生一个新的“子”ASCII CL文件 (102)，修改总是从源文件到子文件。同样，可能由机械师制备M&G码文件而对子文件 (102) 进行的修改也不会对源设计文件 (101) 的修改或者改正产生作用。

作为设计的进化，这种单方向的数据流动要求必须保存所有相关的现有文件和删除陈旧的文件，这会带来很多问题。因此，在本领域中需要一

种能够更好地控制机床的方法和系统，有利于修改、更新和管理提供给机床控制器的控制文件。

另外的问题就是在机床控制器之间即使是“标准”M&G码也存在不同。实际上，M&G码在一种机床与另外一种机床之间是不一致的。一直都在进行着M&G码的标准化努力，但是一直没有成功。这些问题主要由下述事实带来，即，M&G标准不具备实施许多功能的灵活性，这些功能是高级加工机床可以实现的。这些高级机床的制造商不得不使用特有的新指令延续现有的标准，以便将新的加工技术加到代码架构上，如果不这么作，该代码架构就不支持这些技术。

例如，许多现代的机床控制器能够使机床加工真实的非均匀有理B样条曲线（“NURBS”）。即使它不是M&G代码标准的一部分，这些控制器将非标准M&G代码识别为用于NURBS的代码。

即使声称基于标准M&G码，其效果是编程接口和语言随着机器制造商和型号而改变。因此，本领域需要一种避免不同制造和型号的机床之间不一致的M&G码版本的问题的方法和系统。

在CL文件和M&G码中，几何形状表示为刀具顶端通过以便制造该几何形状的一系列点。这些点限定了刀具的路线。这条路线由一些直线段、圆形或者螺旋形的弧线组成。即不是直线又不是圆形的路线近似为离散点或者小线段。造型的和自由形式的表面被越来越多地使用在消费产品中。CAD销售商在它们的软件塑造复合曲线、表面和实体方面已经取得了长足的进步，但是用于制造它们的工具和技术还和几十年前使用的一样。

一个复合表面在CAD程序中可以由单个方程式表示，但是在提供给机床控制器的指令中可能需要表示为数千条线段，以便获得所需的精确度。在使用图1示出的传统数据流时，没有办法避免这种精确度上的损失。

如上所述，有一些非常昂贵的机床控制器具有处理象NURBS这样复杂的路线的能力。这使得这种机床能够比使用线性内插更快和更平滑地制造复合表面。问题在于将用于所需要NURBS的技术要求传递给机床控制器。为了与机床控制器通信，在CAD/CAM设计文件中规定的NURBS（也许为每个曲线使用一个单独的方程式）必须被译成M&G码使用的线段几何形状。然后机床控制器可以将M&G码的线段指令翻译成机床使用的（也可能是通过

一系列点内插一条NURBS曲线规定的) NURBS指令。这种NURBS转换成线段, 然后返回到真正的NURBS表达式导致在定义所述NURBS时毫无意义的精度降低。

因此, 本领域也需要通过基于线段的M&G码的中间状态来避免不必要的精度损失的系统和方法, 这种精度损失与将由CAD/CAM应用程序生成的一个NURBS表达式发送到由高级机床控制器使用的可能不同的NURBS表达式有关。

发明内容

本发明的一个目的是满足上述和其它需要。具体的说, 本发明的一个目的就是提供一种改进的系统和方法, 用于对机床控制器编程, 以便根据由CAD/CAM或其它设计应用程序产生的数据加工规定的零件。本发明的另外一个目的是提供一种系统和方法, 用于对机床控制器编程, 以避免与管理反映源设计文件中零件设计的进化的一系列子文件有关的问题。

本发明的另外一个目的是提供一种系统和方法, 用于对机床控制器编程, 以避免由于在不同构造和型号的机床中使用的各种版本的M&G码间高级指令的变化引起的问题。本发明的再一个目的是提供一种系统和方法, 用于对机床控制器编程, 以避免在规定诸如NURBS之类的曲线中的不必要的精度损失, 该精度损失与通过例如M&G码中使用的线段近似值翻译这些曲线的表达式有关。

本发明的其它目的、优点和新颖的特性将在下文的说明书中描述, 本领域技术人员通过阅读这些材料或实施本发明可理解这些目的、优点及特性。本发明的目的和优点可以通过在所附的权利要求书中叙述的装置来实现。

为了实现本发明的目的, 根据本发明的一个方面, 提供一种用于将控制系统与伺服驱动的机器连接的系统, 包括: 第一处理器, 执行产生运动路线数据而不使用M&G码的设计应用程序; 与所述第一处理器连接的第二处理器, 所述第二处理器执行从所述第一处理器上的所述设计应用程序存取所述运动路线数据的运动控制软件; 和其中所述运动控制软件根据所述运动路线数据控制伺服驱动的机器, 并且所述运动控制软件是面向对象

的，以致所述运动控制软件适用于允许使控制系统与不同的伺服驱动的机器连接。

根据本发明的另一个方面，提供一种将控制系统与伺服驱动的机器连接的方法，包含：执行第一处理器上的设计应用程序，所述设计应用程序产生用于所述伺服驱动的机器的运动路线数据，所述运动路线数据是未使用M&G码产生的；执行第二处理器上的运动控制软件，所述运动控制软件以与所述设计应用程序输出的相同的形式接收所述运动路线数据；和根据来自所述设计应用程序的所述运动路线数据，用所述运动控制软件控制所述伺服驱动的机器。

根据本发明的再一个方面，提供一种将控制系统与伺服驱动的机器连接的系统，包括：产生用于伺服驱动的机器的运动路线数据的装置，所述运动路线数据是未使用M&G码产生的，所述装置用于产生包括第一处理器的运动路线数据；用于执行运动控制软件的装置，该运动控制软件以与由产生所述运动路线数据的所述装置输出的相同的形式接收所述运动路线数据，所述装置用于执行包括第二处理器的运动控制软件；和根据所述运动路线数据控制所述伺服驱动的机器的装置。

根据本发明的再一个方面，提供一种将控制系统与伺服驱动的机器连接的系统，包括：执行设计应用程序的第一处理器，所述设计应用程序产生运动路线数据而不使用M&G码；与所述第一处理器连接的第二处理器，所述第二处理器执行运动控制软件，所述运动控制软件通过所述运动控制软件的应用程序接口从所述第一处理器上的所述设计应用程序接收所述运动路线数据；和其中所述运动控制软件根据所述运动路线数据控制伺服驱动的机器。

根据本发明的再一个方面，提供一种用于将控制系统与伺服驱动的机器连接的系统，包括：执行设计应用程序的第一处理器，所述设计应用程序产生包含运动路线数据的运动对象，所述运动路线数据是未使用M&G码产生的，所述运动路线数据定义伺服驱动的机器的工具头要沿其移动的运动路线；与所述第一处理器连接的第二处理器，所述第二处理器执行从所述第一处理器上的所述设计应用程序接收所述运动对象的运动控制软件；和其中所述运动控制软件根据所述运动路线数据控制伺服驱动的机器。

总之，本发明提供了传统CAD工作站和机床控制器的概念集成。机床控制器最好包括两个处理器，或各提供一个处理器的两台计算机。第一处理器执行一种设计应用程序，即，CAD、CAD/CAM或类似软件包，该设计应用程序可用于产生、检查或修改设计文件。第二处理器执行控制所述机床所需的软件。

可以完全运行在第二处理器上的软件实现整个机床的控制。这种可以包括几个应用程序或对象的软件通常被称为“运动控制软件”。不再需要其它硬件架构作为机床控制器的一部分。

运动路线数据规定机床的一个或多个刀头必须沿其移动以制造所需产品的一条路线或多条路线。如果设计应用程序的应用程序接口（API），例如，CAD/CAM软件包，开放并可供使用，可以直接从设计文件中提取运动路线数据。然后，运动控制软件可以使用该刀具路线数据驱动机床并制造规定的产品。

在不能获得设计应用程序的API的情况，设计应用程序，例如，CAD/CAM软件包，可以将运动路线数据传递给运动控制软件，这是一种直接加工运动控制软件的API组。然后，运动控制软件可以使用刀具路线数据驱动机床并制造出规定的产品。

最好是，以C++运动对象的形式传递运动路线数据，C++运动对象规定机床的一个或多个刀具头的位置和方向以及任何刀具路线中的任何点。最好是，运动对象包括三个其它对象：一个确定用于伺服驱动的机器元件的控制的进给速度和速率的速度对象；一个用于在那些运动路线中控制速度转换的速度分布产生器对象；和一个包含定义每个运动路线的信息的路线对象。

从CAD系统中提取的运动路线数据一般在笛卡尔座标系统中。因此，运动控制软件将笛卡儿坐标刀具路线数据翻译成适当的参考坐标系，伺服控制器应用程序从这个参考坐标系产生用于驱动机床的伺服电机的伺服控制信号。

附图说明

附图示出了本发明，并且是说明书的一部分。这些附图将与下文的说

说明书一起说明和解释本发明的原理。

图1是说明在对机床控制器编程中使用的传统系统和数据结构的方框图。

图2是说明根据本发明的机床控制器系统的方框图。

图3是图2所示的处理单元的更详细的方框图。

图4是根据第一实施例，图2所示的伺服控制器目标和系统接口的更详细的方框图。

图5是根据第二实施例，图2所示的伺服控制器目标和系统接口的更详细的方框图。

图6是由图3的CAD解释程序产生的运动对象的方框图。

具体实施方式

本发明提供一种用于数字控制接口的新的、开放的、软件架构。本领域技术人员能够意识到，这种新接口架构可以应用在将控制数据发送给与控制系统连接的伺服控制机器的任何系统。

本发明的数字控制接口和伴随原理的优选应用于根据预定技术要求制造机器零件的机床的机床控制器。例如，本发明提供了一种新的范例，该范例能够利用诸如CAD/CAM软件包之类的现有设计应用程序和机床控制器的能力，更直接和有效地将CAD产生的零件设计转化成正确控制机床所需的指令。本发明的过程被称为“直接加工”。

直接加工消除了对如CL文件和M&G码文件之类存在于CAD设计文件之外的ASCII文件的需求。而是，能够使机床控制器读取原始的CAD设计文件并从中提取控制机床制造规定产品所需的所有相关加工数据。

参照附图描述本发明的这种具体应用的优选实施例。如图2所示，根据本发明的机床控制器(200)可以包括传统CAD工作站(100)的一些主要特征。具体的说，所述控制器(200)可以包括执行CAD软件包的第一处理器(203)。该处理器(205)连接到显示装置(201)和用户输入装置(202)，所述显示装置例如是阴极射线管显示器，液晶显示器等等，所述用户输入装置例如是键盘和鼠标、轨迹球、操纵杆等等。因此，可以在机床控制器(200)的硬件产生或改变计算机辅助设计，而不是在分开的工作站上进

行。

当然，也可以在分开的CAD工作站产生的设计文件传送到处理器（203），以便在机床控制器（200）的硬件（201，202）上显示、检查和/或修改。可以将CAD文件传送到机床控制器（200），例如，通过软盘或者任何其它文件传输手段，包括工作站和控制器（200）之间的有线或者无线网络连接。

在图2的例子中，第一处理器（203）是同样包含第二处理器（204）的处理单元（205）的一部分。一般而言，第一处理器（203）运行CAD或者CAD/CAM软件包，或其它设计应用程序，而第二处理器（204）运行使用CAD软件产生的数据所需的软件，以便由此控制机床。

第二处理器（204）上执行的软件被统称为“运动控制软件”，并且是用于控制机床的完整系统。不再需要另外的控制器硬件。

系统接口（206）设置在机床控制器（200）和机床（在图2中未示出）之间，以使第二处理器（204）的软件可以与所述机床通信并对该机床进行控制。连接（207）设置在第二处理器（204）和系统接口（206）之间。系统接口（206）将在下文中更详细地描述。

虽然优选图2中示出的双处理器单元（200），但是本发明也可以用两个单独的计算机单元，例如，两个PC实现，每个单元提供在此所述的相应的第一或第二处理器。可用于Windows®NT型处理器实施本发明。

为了最好地完成直接加工，必须兼备具有公知架构的CAD/CAM系统和机器控制器，以便能够访问它们的内部函数。例如，为了加工规定的零件，每个机床路线的数学表达式，即，运动路线数据，必须从要被制造的零件的设计产生。

CAD或CAD/CAM软件包具有一个应用程序接口（API）。API是操作系统和设计应用程序之间的接口。API规定了设计应用程序与操作系统通信的方式，以及操作系统可为设计应用程序提供的服务。

如果控制机床的运动控制软件可以通过它的API访问设计应用程序并直接从由设计应用程序打开的设计文件中提取机床的刀具路径，那么直接加工将很好地工作。不幸的是，传统CAD/CAM软件包并没有公开实现该提取工作所需的功能的可用API。然而，如果设计应用程序的API可供使用，

那么本发明的最佳实施可以通过设计应用程序的API简单地从设计文件数据中直接提取机床的路线。这种提取工作优选由运行在第二处理器（204）上的软件来执行。这些机床路线能因此被第二处理器（204）的机器控制器软件使用，以控制机床制造规定的零件。

如通常的情形一样，在不能使用CAD/CAM软件包的API的情况下，本发明的原理仍然可以提供直接加工。在这种情况下，DMAC运动控制软件将提供一个API组，CAD/CAM软件包可以调用API组将刀具路线信息传送给直接加工软件。

为了理解本发明中如何将刀具路线传送给控制器，需要理解一些有关在第一（203）和第二（204）处理器上执行的软件的架构的情况。如图3中所示，有两个在本发明的双处理器系统上运行的主要应用程序：CAD解释程序（301）和运动控制器（305）。与上面的解释一致，如图3所示，CAD解释程序（301）最好在第一处理器（203）上执行。而运动控制器（305）最好在第二处理器（204）上执行。

所述CAD解释程序（301）被认为是设计应用程序，并能够理解利用CAD软件包产生的设计文件（309）并从中提取相关的加工信息，即由设计文件（309）中的数据表示的机床路线。CAD解释程序（301）几乎不需要关于控制器或要被控制来制造规定的零件的机床的信息。然而，正是CAD解释程序（301）提供了运动路线数据，运动控制器（305）将使用该运动路线数据引导所述机床的刀具头。

如下文中更详细描述，运动控制器（305）是运行在第二处理器（204）上的应用程序，是本发明的运动控制软件的一部分，并从第一处理器（203）上的CAD解释程序接受刀具路线数据（310）。运动控制器（305）负责把来自CAD解释程序（301）的刀具路线数据（310）翻译成可用于正确控制机床制造规定产品的数据。运动控制器（305）还确定移动接触（**move tangency**）。运动控制器（305）包含两个完成这些任务的子系统。

由CAD解释程序（301）提供的运动路线数据（310）通常在笛卡尔座标系统中规定刀具路线。运动控制器（305）首先执行运动计划程序（307）（**Motion Planner**），以便将来自笛卡儿空间的刀具路线数据映射到联合空间（**joint-space**）。联合空间是定义机床的一个或多个刀具头的可能路

线的坐标系统。轨迹计划程序(308)从联合空间数据计算出正确移动机床的刀具头通过规定路线所需的执行器值。

如图3所示,执行器值随后被传送到伺服控制器应用程序(306),该应用程序也是本发明的运动控制软件的一部分。伺服控制器(306)负责执行伺服控制环路,以使机床的刀具头移动通过规定的刀具路线,该规定的刀具路线将导致在所述机床上由坯料生产所需的零件。

最好,本发明的系统采用由C++计算机语言编写的对象。C++对象为这样的数据结构,它们可以同时具有成员(member)(子数据结构或变量)和与所述成员相关的方法(在对象上可以调用的函数)。函数通过在对象上调用一个适当的函数还可以“问”对象一个问题。通过该函数的返回值接收“答案”。

C++对象还具有继承和多态性质。继承性描述的是一个对象类别从其父类继承性质的能力。同样地,任何为子(或派生)类别定义的特定成员或方法还将具有其父类(parent class)的所有成员、方法和性质。因此,派生类的对象可以被当作子类的成员或者当作父类的成员对待。多态性允许所有从相同父类派生的类可以具有相同的接口,使得对象可以从外部将它们都当作父类来对待。然而,当调用在该接口中的函数时,对象的行为可以根据对象的派生类而非常不同。

运动控制器(305)包含成员对象(307和308),该成员对象(307和308)执行上述的特殊函数。运动学对象(307)将笛卡儿坐标值转变成联合值(joint values),然后联合表对象(308)将联合值转变成执行器值。联合表对象(308)最好包含一个特定的联合执行器对象,用于被控制机床的每个联合。因而,由一个分开的联合执行器对象为每个联合执行联合向执行的变换。然后,将执行器位置和速度传送给软件伺服控制器(306)。运动控制器(305)使用的运动学对象和联合执行器对象实际上是从基本类派生的,并根据它们代表的变换种类(例如,用于运动学上的3或5轴线机床和用于联合执行器的齿轮或滚珠丝杠(ball screw))而不同。

伺服控制器应用程序(306)包含指向执行象靠近伺服控制环路和与电机通信这样的任务的对象的指针。(参见图4&5)。如同运动控制器(305),伺服控制器应用程序(306)包含同样从基本类派生的对象。例

如，伺服控制器包含执行伺服控制环路的对象，但是该对象是各种基于其实施的控制规律（例如PID等）的派生类之一。

最好是通过调用运动对象（302）的C++对象将加工信息从CAD解释程序（301）发送到运动控制器（305）。在运动控制器305）从CAD解释程序（301）接收到运动对象（302）之后，它通过查询运动对象（302）而获得所有必需的刀具路线信息。运动对象（302）由CAD解释程序（301）生成并将刀具头沿着刀具路线运动过程中的任何时候需要的刀具位置和方向通知运动控制器（305）。然后，如上所述，运动处理器（305）利用其所包含的C++对象将刀具位置，方向等变换成伺服控制器（306）使用的执行器值。然后，伺服控制器（306）使用一些已存储在系统中并专用于被控制的机床的控制规律计算施加到机床的电机的转矩，以实现那些数值。

图4示出本发明的实施例，其中使用模拟电机驱动机床的部件，以使机床刀头部移动通过所希望的刀具路线。在该实施例中，伺服控制器（306）包括把电机（406）的数字控制信号传递给系统接口（206）中的数模转换器板（404）的数字模拟驱动器（402）。然后，发来自转换器板（404）的模拟控制信号（407）并提供给电机（406）。与每个电机（406）相关的传感器（405）检测，例如电机（405）的实际转矩、位置、速度和加速度并向系统接口（206）中的模拟数字转换器板（403）提供模拟信号。然后，把数字化的传感器信号提供给伺服控制器（306）的模/数转换驱动器（401）以完成反馈环路，该反馈环路使伺服控制器（306）根据需要按照规定的刀具路线驱动电机（406）。

这个过程可以并且最好完全由软件执行，不需要任何其它硬件。因而可以消除封闭的硬件架构。

图5示出本发明的第二实施例，其中使用数字电机驱动机床的部件以移动机床刀头通过所希望的刀具路线。在该实施例中，系统接口（206）包括一个串行通信装置（502），该串行通信装置通过高速数据网络（506）与每个电机单元的数字电机接口（503）通信。数字电机接口（503）接收规定转矩设置点的控制信号并将其转换成模拟信号，然后，电机放大器（504）放大该模拟信号，并根据需要使用其驱动相应的电机（505），以便移动机床通过规定的刀具路线。同上，传感器完成可完全由软件驱动的

伺服环路。

如这里描述的，本发明的直接加工方法和系统提供了优于传统加工技术的优点。直接加工消除了与目前设计到制造过程相关的许多问题，因为使用的是真实的设计几何形状而不是仅在CL文件和M&G码中使用的离散点的数据。因为加工是基于原始的设计几何形状，所以没有几何信息丢失或者近似。

另外，因为没有新文件产生，所以不存在由于陈旧的CL和M&G文件存留在所述系统中带来的有可能无意中被用于错误地加工一个零件。对设计文件的改变直接反映在所控制的机床的操作中。本发明还具有一些与直接加工有关的间接优点，以及具有接近原始模型的车间。同时许多加工中心能够用常驻在控制器（200）中的CAD模型在进行中做出检查，这种检查可以用一个接触探头进行，并可以与来自M&G文件的刀具位置点的外推法相反，相对于模型本身评估。另外，可要求控制器（200）从一个与控制器（200）联网的产品数据管理（PDM）系统中检验设计文件，进一步消除加工过时零件的可能性。

从CAD文件（309）中产生刀具路线和产生包含这个刀具路线数据的运动对象的方法的例子将在下文中详细地说明。一般来说，有两种类型的加工：粗加工和精加工。粗加工从坯料除去大部分材料，而精加工尝试使坯料的表面与为零件规定的精确表面相配。在这个例子中，我们仅仅考虑到与我们自己有关的切削操作中的精加工刀具路线。

刀具路线代表机床的刀具头的尖端中心所遵循的路线，而不是零件的表面和工具之间的切点的路线。在一些简单的操作，象铣面，刀具尖端就是所述的切点，所以刀具路线就直接位于加工面上。在稍微复杂的加工中，诸如加工凹处，端铣刀的侧面进行切割。在这种情况下，刀具路线落在与加工面偏离刀具半径的距离的表面上。

当加工形状不规则的表面时事情将更加复杂，因为这种表面经常用圆头槽铣刀来加工，这种圆头槽铣刀具有一个球形尖端，该球形尖端具有与工具相同的半径。使用下述等式为圆头槽铣刀计算刀具尖端位置：

$$P(s, t) = Q(s, t) + RN(s, t) - [0, 0, R]$$

其中 R 是圆头槽铣刀的半径, $Q(s, t)$ 是切点, $N(s, t)$ 是在每个切点 $Q(s, t)$ 处的表面法线。如果刀具的球形尖端与零件面相切, 那么刀具尖端的中心将位于与零件表面偏离刀具半径的距离的表面上。在3轴线加工中, 如果刀具垂直排列, 刀具尖端在 Z 方向上则整比刀具球形部份的中心低一个半径的距离。为了生成所述刀具尖端遵循的表面, 一个表面从加工面处偏离并沿 Z 轴降低圆头槽铣刀的半径的距离。然后将这个表面与垂直平面交叉定义实际的刀具路线。

在如上所述的C++环境中, 所有传送到运动控制器(305)的运动对象实际上是从运动对象(302)派生的类别成员(members of classes)。为了具体的控制目的, 可以有許多来自运动对象(302)不同派生类别。这种安排的好处在于运动控制器(305)不知道什么类型的运动对象传递到它。只要传送的对象具有运动对象(302)的接口, 运动控制器(305)不需要任何附加信息。这意味着可以开发新的运动对象并结合到CAD解释程序(301)中, 运动控制器(305)不用对其本身升级就可以理解它们。

如上所述, 传送到运动控制器(305)的运动对象通常以笛卡儿坐标规定刀具路线。这是一种方便的方式来确定遵循规定路线的运动, 就象在加工操作中一样。因此, 运动对象(302)的一个规定的子类可以被称为基本笛卡儿运动对象类别。为简单起见, 所有对运动对象的称呼在此后都理解但是指基本笛卡儿运动对象。

如图6中所示, 这些派生出来的运动对象(600)最好包含三个专门类别的成员: 一个速率对象(601)、一个速度分布产生器("VPG")对象(602)和一个路线对象(603)。速率对象(601)确定所使用的进给量和速度, VPG对象(602)控制速度跃变, 路线对象(603)包含关于路线的几何形状的全部信息。

正如运动对象(600)一样, 路线对象(603)也是一个基本类别, 从基本类别得出专用路线。路线对象(603)存储几何形状作为参数方程。用这些参数方程, 路线对象(603)可以告知运动对象(600)有关刀具在沿曲线的任何点的所需位置和方向。运动对象(600)将该几何形状与包含在速度对象(601)和VPG对象(602)中的信息结合起来, 并计算出在

任何时候的位置和方向。因为路线对象（603）总是得出的类别的成员，所以运动对象不需要对与之一起工作的路线对象进行规定。

这种将几何形状传送给控制器的模式与现在用M&G码的情形非常不同，并具有一些重要的含义。在图1的现有技术系统中，机床控制器（104）被给予了定义刀具路线所需的最低量的几何信息。然后控制器（104）使用这个几何形状内插一个连续运动路线。例如，两个端点定义一个线段，两个端点、一个中点和一个旋转方向定义一个螺旋线或弧。

使用本发明的多形态的C++对象来传递几何形状信息并不需要运动控制器（305）知道它已接收了什么类型的路线。另外，路线仅仅被称为一般的曲线，该一般曲线可以告知运动控制器（305）任何所需详细程度的几何形状。

使用这里描述的多形态的C++对象包括比如图1中所示的现有系统中所需的大得多的数据处理和存储量。然而，这不是一个重要的问题，因为现代控制器足够快并具有足够的存储器来处理这样的数据流增加。直接加工的一个重要优点就在于控制器准确地接收其所需的信息量。最终，当I/O指令发送到驱动电机时，必须将所有曲线分解为小的、不连续的段。按常规，在输出APT文件时，所述曲线被CAM程序离散化。在直接加工中，所述运动控制器将曲线离散化。这样使得控制器总是以其最好容限进行加工。

前述的说明仅仅是用来举例和描述本发明。并不意味着已经穷举了本发明或者将本发明限于在这里公开的任何形式。根据上文的教导下能够对本发明进行许多修改和变化。

选择和描述优选实施例是为了更好地解释本发明的原理和它的实际应用。上文的描述旨在使本领域技术人员能够最好地以各种实施例利用本发明，对于具体的应用可以进行各种改变。本发明的范围由所附的权利要求书限定。

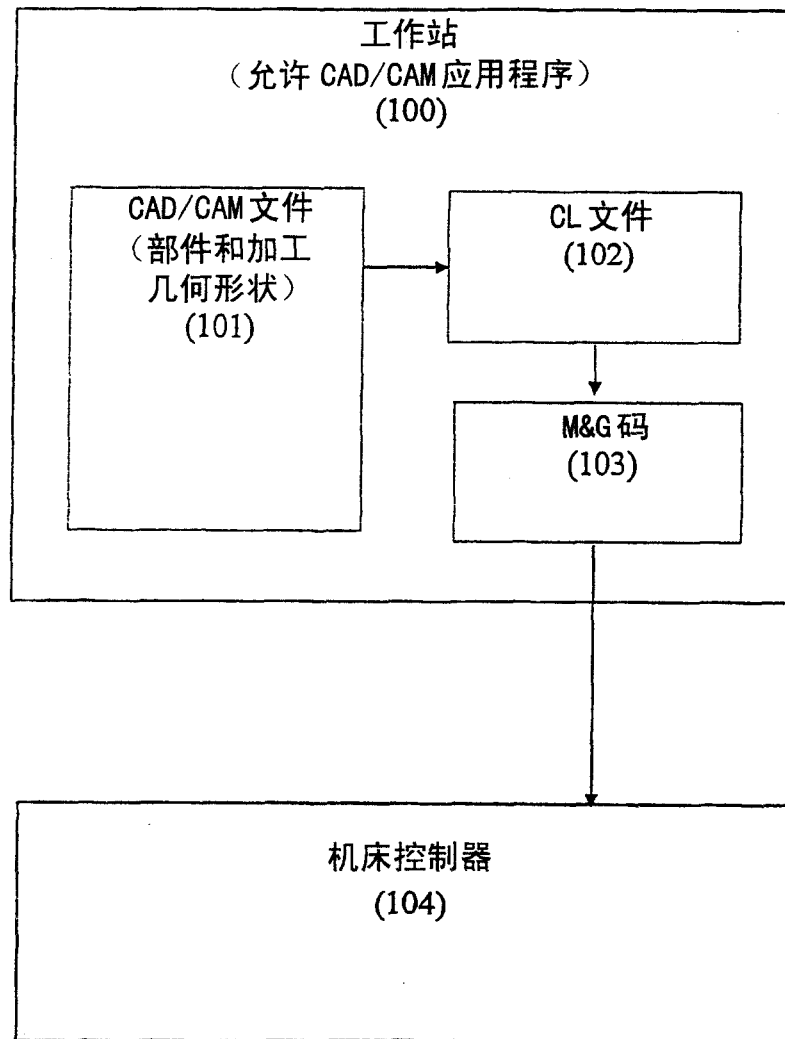


图 1

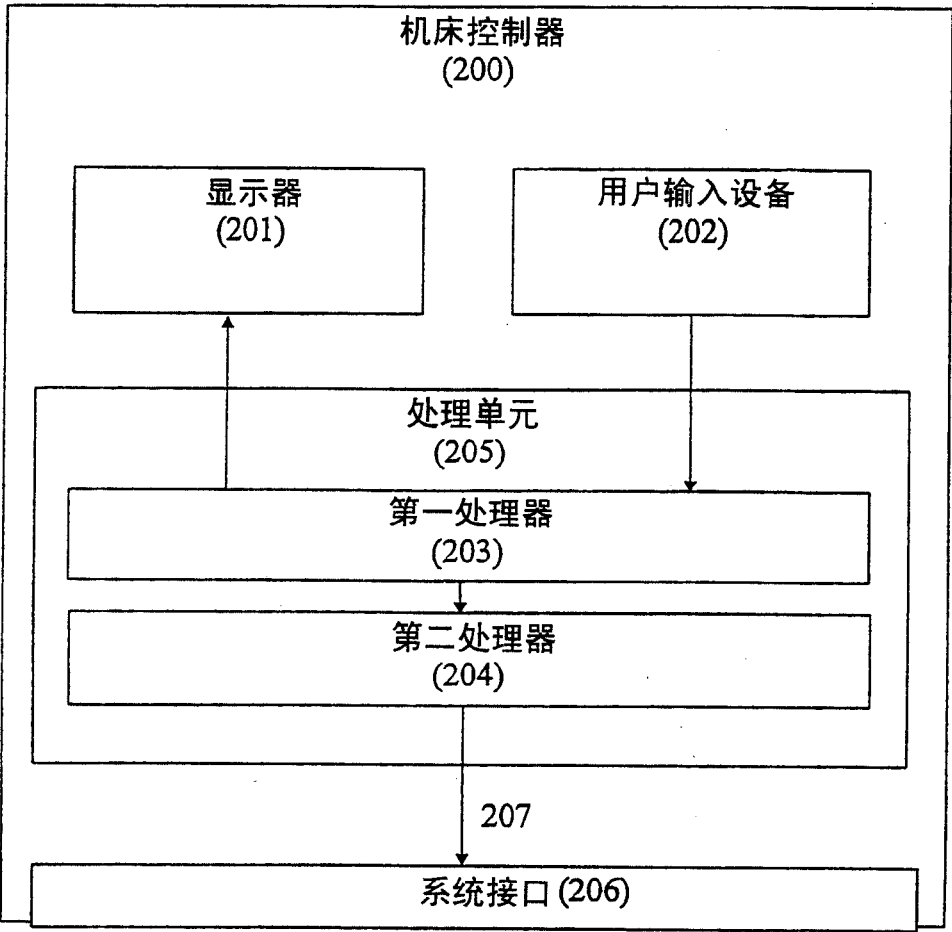


图 2

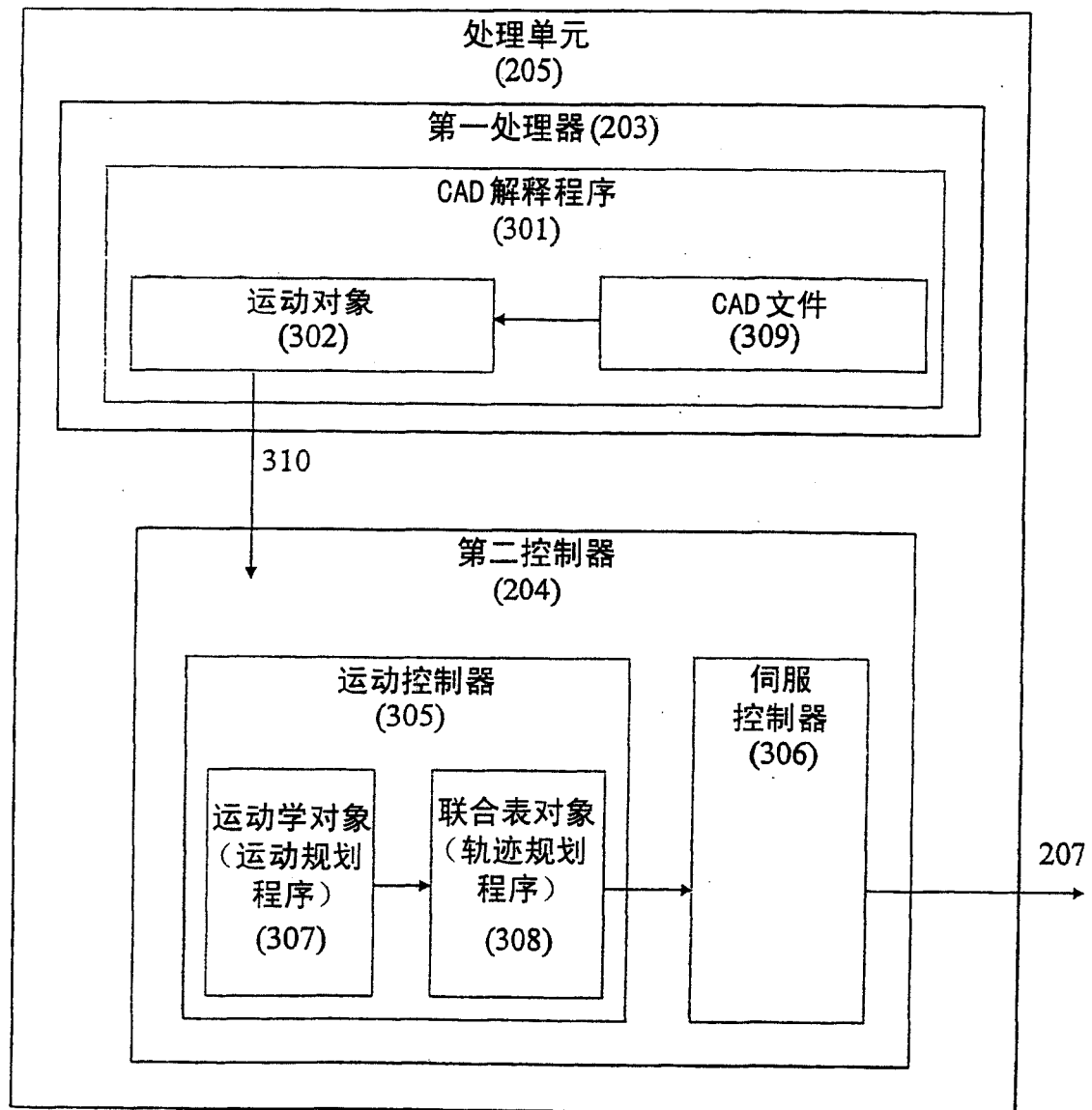


图 3

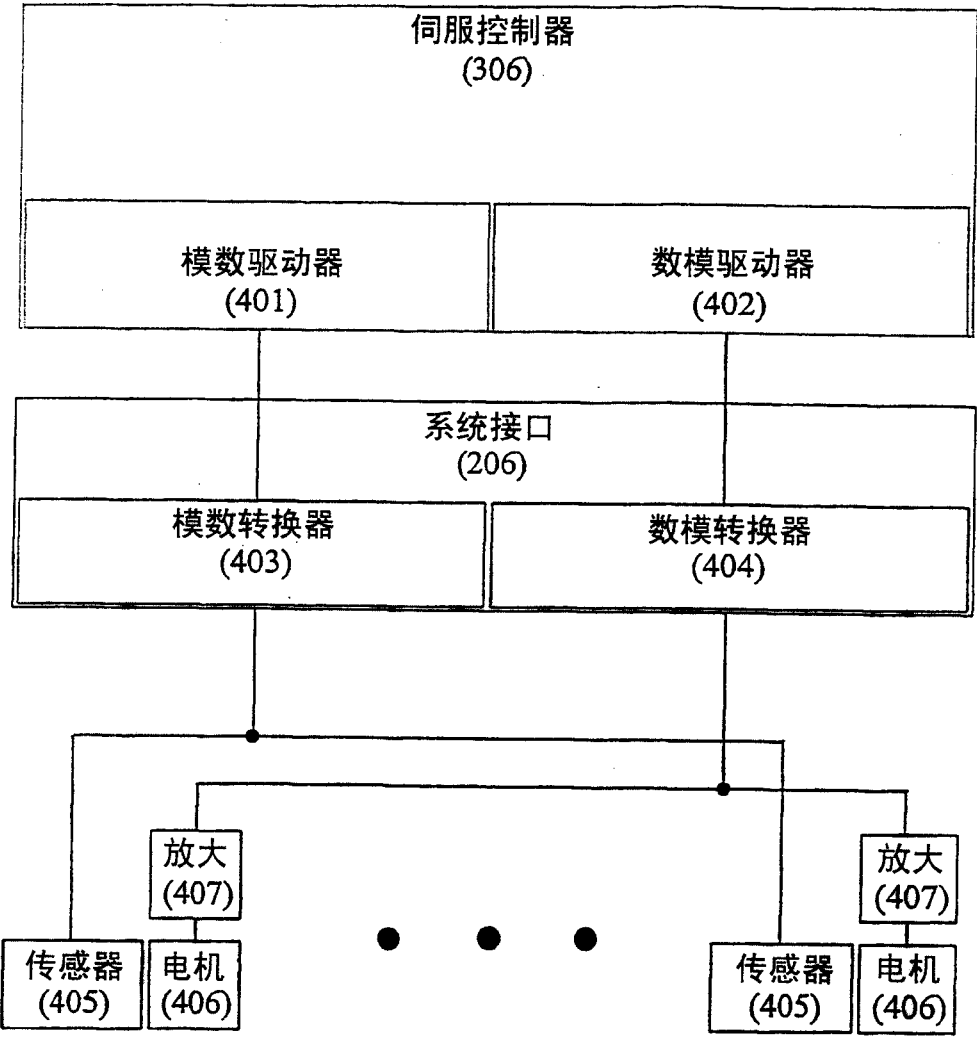


图 4

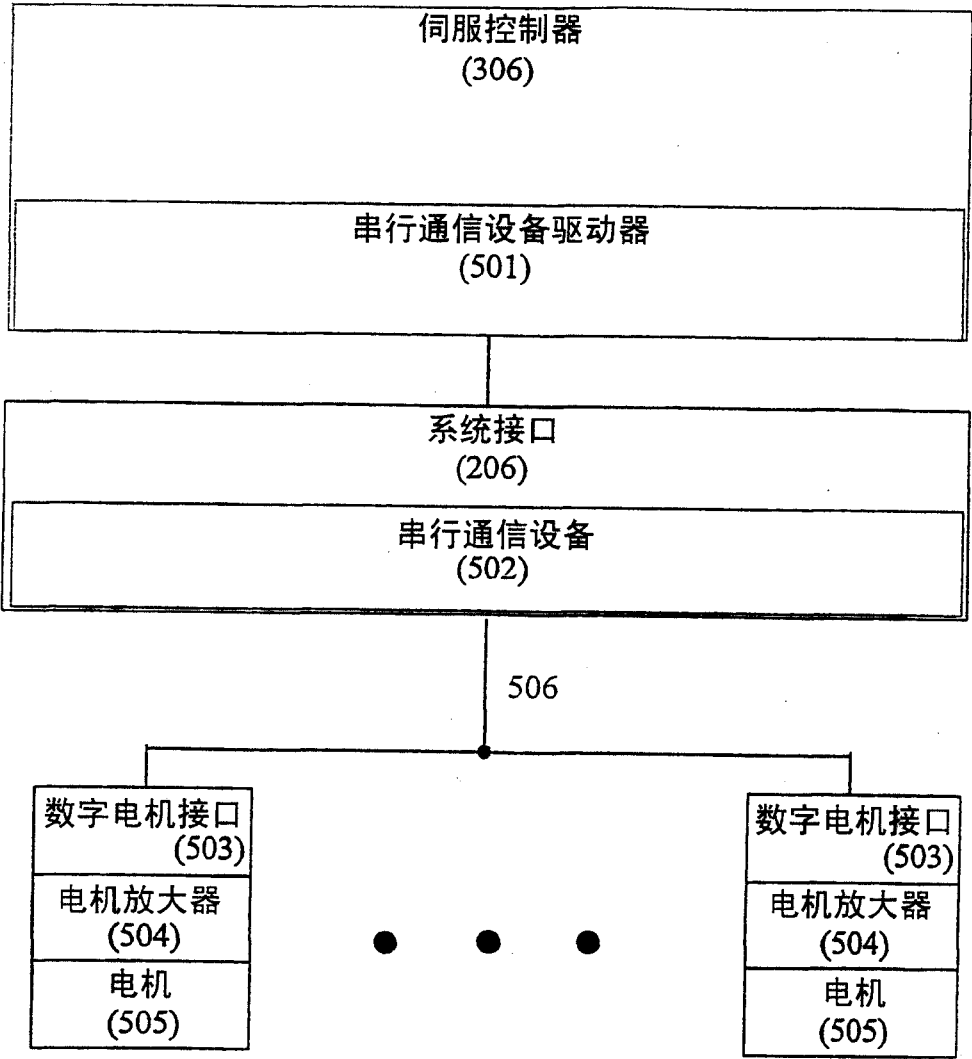


图 5

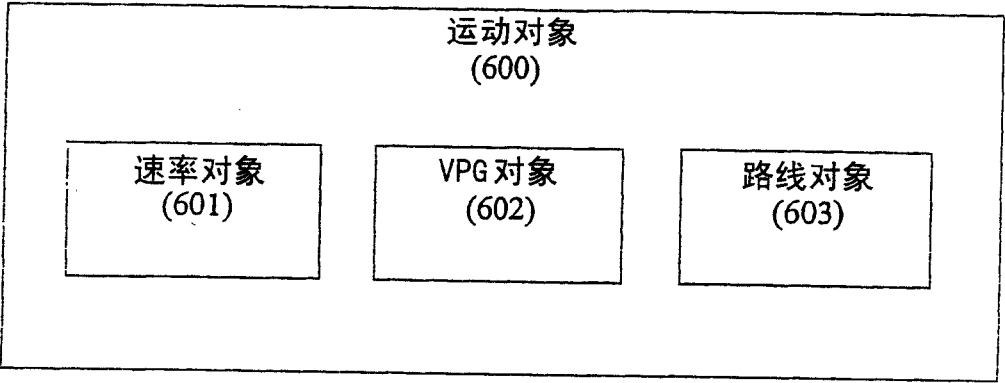


图 6