(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 103250106 B
(45)授权公告日 2014.12.24

(21)申请号 201180056419.3
(22)申请日 2011.09.29
(30)优先权数据
12/893,670 2010.09.29 US
13/026,823 2011.02.14 US
(85)PCT国际申请进入国家阶段日
2013.05.23
(86)PCT国际申请的申请数据
PCT/US2011/054008 2011.09.29
(87)PCT国际申请的公布数据
WO2012/050977 EN 2012.04.19
(73)专利权人 数学工程公司
地址 美国马萨诸塞州
(72)发明人 P·加希奈特 P·阿帕卡瑞恩 D·诺尔
(74)专利代理机构 北京泛华伟业知识产权代理有限公司 11280
代理人 王勇
(51)Int.Cl.
G05B 13/04(2006.01)

(56)对比文件
CN 1327551 A,2001.12.19, 全文.
CN 101008840 A,2007.08.01, 全文.
Musa A. Mammadov, et al..A Nonsmooth Optimization Approach to H∞ Synthesis.《Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005》.2005,6893-6898页.
Pierre Apkarian, et al..Nonsmooth optimization algorithm for mixed H2/H∞ synthesis.《Proceedings of the 46th IEEE Conference on Decision and Control》.2007,4110-4115页.

审查员 金川

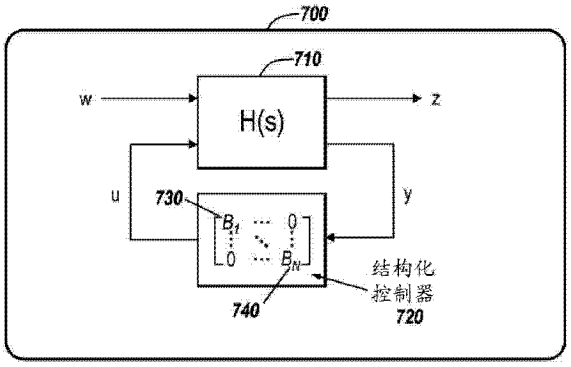
权利要求书3页 说明书29页 附图27页

(54)发明名称

多输入多输出控制结构的交互控制

(57)摘要

示例性实施例允许用户交互地使用公式表示并且求解多变量反馈控制问题。例如,用户可以解决问题,其中,多个控制元素分布在一个或多个反馈回路上,并且需要被联合调整以最优化控制系统的整体性能和健壮性。实施例允许用户以用户熟悉的格式来指定设计要求和目标。实施例可以对可调整参数进行操作,以便以满足由用户提供的设计要求和/或目标的方式来解决控制问题。



CN 103250106 B

1. 一种用于调整控制结构的模型的方法，包括：

获得所述模型中的一个或多个可调整组件，

所述获得通过计算装置来进行；

接收与所述模型相关联的设计要求，

所述接收通过计算装置来进行；

将所接收到的设计要求变换为用于调整所述模型中的一个或多个可调整组件的 H∞约束，

所述变换基于将与所述模型相关联的控制架构映射到特定结构，以及

所述变换通过计算装置来进行；

使用所述 H∞约束来提取控制器或函数中至少之一，

所述使用通过计算装置来进行，以及

所述 H∞约束表达以下中的至少之一：

与所述模型相关联的一个或多个设计目标，或者

与所述模型相关联的一个或多个设计要求；

使用优化器基于所提取的控制器或函数中至少之一并且基于所述模型中的所述一个或多个可调整组件来调整所述模型的一个或多个参数，

所述调整：

执行所述 H∞约束，并且

所述调整通过计算装置来进行；以及

基于使用所述优化器调整所述模型的所述一个或多个参数来生成结果，

所述结果包括一个或多个调整后的组件，

所述一个或多个调整后的组件满足与所述模型相关联的所述一个或多个设计目标或者与所述模型相关联的所述一个或多个设计要求中的至少之一，以及

所述生成通过计算装置来进行。

2. 根据权利要求 1 所述的方法，其中，当调整所述模型的一个或多个参数时，所述方法包括：

使用以下中的一种来调整所述模型的一个或多个参数：

MATLAB 兼容语言，

Simulink 兼容语言，或

LabVIEW 兼容语言。

3. 根据权利要求 1 所述的方法，其中，从模块集选择所述一个或多个可调整组件。

4. 根据权利要求 1 所述的方法，其中，所述调整支持所述控制结构的交互设计和调整。

5. 根据权利要求 1 所述的方法，其中，当使用所述 H∞约束来提取控制器或函数中至少之一时，所述方法包括：

将所述设计要求应用至所述函数以生成一个或多个设计任务。

6. 根据权利要求 1 所述的方法，其中，当获得所述一个或多个可调整组件时，所述方法包括：

基于用户输入来获得所述一个或多个可调整组件。

7. 根据权利要求 1 所述的方法，其中，所述特定结构为标准形式。

8. 根据权利要求 7 所述的方法，其中，所述标准形式包括：

线性模型，以及

所述一个或多个可调整组件。

9. 根据权利要求 8 所述的方法，其中：

所述线性模型包括一个或多个不可调整组件，以及

以块对角方式来布置所述一个或多个可调整组件。

10. 根据权利要求 8 所述的方法，其中，以块对角方式来布置所述一个或多个可调整组件的一部分。

11. 根据权利要求 1 所述的方法，进一步包括：

参数化所述一个或多个可调整组件，

其中，当调整所述模型的所述一个或多个参数时，所述方法包括：

基于参数化所述一个或多个可调整组件来与所述一个或多个可调整组件交互以调整所述模型的一个或多个参数。

12. 根据权利要求 1 所述的方法，其中：

所述一个或多个可调整组件包括一个或多个自由参数，以及其中，

当调整所述模型的一个或多个参数时，所述方法包括：

与所述一个或多个自由参数交互以调整所述模型的一个或多个参数。

13. 一种用于调整控制结构的模型的设备，包括：

用于获得所述模型中的一个或多个可调整组件的装置；

用于接收与所述模型相关联的设计要求的装置；

用于将所接收到的设计要求变换为用于调整所述模型中的一个或多个可调整组件的 H ∞约束的装置，

所述变换基于将与所述模型相关联的控制架构映射到特定结构；用于使用所述 H ∞约束来提取控制器或函数中至少之一的装置，

所述 H ∞约束表达以下中的至少之一：

与所述模型相关联的一个或多个设计目标，或者

与所述模型相关联的一个或多个设计要求；

用于使用优化器基于所提取的控制器或函数中至少之一并且基于所述模型中的一个或多个可调整组件来调整所述模型的一个或多个参数的装置，

所述调整：

执行所述 H ∞约束；以及

用于基于使用所述优化器调整所述模型的一个或多个参数来生成结果的装置，

所述结果包括一个或多个调整后的组件，以及

所述一个或多个调整后的组件满足与所述模型相关联的一个或多个设计目标或者与所述模型相关联的一个或多个设计要求中的至少之一。

14. 根据权利要求 13 所述的设备，还包括用于当调整所述模型的一个或多个参数时，使用以下中的一种来调整所述模型的一个或多个参数的装置：

MATLAB 兼容语言，

Simulink 兼容语言，或

LabVIEW 兼容语言。

15. 根据权利要求 13 所述的设备，其中，从模块集选择所述一个或多个可调整组件。

16. 根据权利要求 13 所述的设备，其中，所述调整支持所述控制结构的交互设计和调整。

17. 根据权利要求 13 所述的设备，还包括：用于当使用所述 H ∞约束来提取控制器或函数中至少之一时，将所述设计要求应用至所述函数以生成一个或多个设计任务的装置。

18. 根据权利要求 13 所述的设备，还包括：用于当获得所述一个或多个可调整组件时，基于用户输入来获得所述一个或多个可调整组件的装置。

19. 根据权利要求 13 所述的设备，其中，所述特定结构为标准形式。

20. 根据权利要求 19 所述的设备，其中，所述标准形式包括：

线性模型，以及

所述一个或多个可调整组件。

21. 根据权利要求 20 所述的设备，其中：

所述线性模型包括一个或多个不可调整组件，以及

以块对角方式来布置所述一个或多个可调整组件。

22. 根据权利要求 20 所述的设备，其中，以块对角方式来布置所述一个或多个可调整组件的一部分。

23. 根据权利要求 13 所述的设备，还包括：

用于参数化所述一个或多个可调整组件的装置，以及

用于当调整所述模型的一个或多个参数时，基于参数化所述一个或多个可调整组件来与所述一个或多个可调整组件交互以调整所述模型的一个或多个参数的装置。

24. 根据权利要求 13 所述的设备，其中，

所述一个或多个可调整组件包括一个或多个自由参数，以及其中，

所述设备还包括用于当调整所述模型的一个或多个参数时，与所述一个或多个自由参数交互以调整所述模型的一个或多个参数的装置。

多输入多输出控制结构的交互控制

[0001] 相关申请

[0002] 本申请要求在 2011 年 2 月 14 日提交的美国专利申请 No. 13/026,823 和在 2010 年 9 月 29 日提交的美国专利申请 No. 12/893,670 的优先权，其内容通过引用包含于此。

附图说明

[0003] 包含在本说明书中并且构成本说明书的一部分的附图示出了本发明的一个或多个实施例，并且与说明书一起说明本发明。在附图中，

[0004] 图 1 示出了用于使用 H ∞合成技术的传统配置；

[0005] 图 2 示出了用于实现本发明的实施例和技术的示例性系统；

[0006] 图 3 示出了可以用于实现本发明的方面的建模环境的示例性实现方式；

[0007] 图 4-6 示出了可以应用本发明的示例性实施例以调整控制器架构的组件的示例性控制器架构；

[0008] 图 7 示出了本发明的示例性实现方式，该实现方式使用结构化的控制器来与系统的 H ∞合成表示相结合地调整参数；

[0009] 图 8 示出了用于与本发明的实施例结合使用的反馈回路的示例性期望回路形状；

[0010] 图 9 示出了包括被控对象（plant）和控制器的示例性配置，该控制器具有可以应用示例性技术以调整组件的可调整元素；

[0011] 图 10 示出了用于可以使用本发明的实施例调整的多输入多输出控制问题的示例性期望回路形状；

[0012] 图 11-14 示出了用于显示可以用于本发明的实施例的命令、对象和输出的示例性界面；

[0013] 图 15 示出了具有可以使用本发明的实施例调整的可调整组件的示例性控制器；

[0014] 图 16A 和 16B 示出了用于显示可以用于本发明的实施例的命令和 / 或输出的示例性界面；

[0015] 图 17 和 18 示出了包括可以使用本发明的实施例调整的可调整组件的示例性自动驾驶仪系统；

[0016] 图 19 示出了示例性界面，该示例性界面可以用于接收与图 17 和 18 的自动驾驶仪系统相关的输入和 / 或显示与图 17 和 18 的自动驾驶仪系统相关的输出；

[0017] 图 20 示出了调整图 17 和 18 的自动驾驶仪系统的组件的、可以用于本发明的实施例的示例性目标回路形状；

[0018] 图 21 和 22 示出了用于接收与图 17 和 18 的自动驾驶仪系统相关的输入和 / 或显示与图 17 和 18 的自动驾驶仪系统相关的输出的示例性界面；

[0019] 图 23 示出了与图 17 和 18 的自动驾驶仪系统相关联的示例性阶跃响应；

[0020] 图 24 示出了可以用于评估与图 17 和 18 的自动驾驶仪系统相关联的灵敏度函数的增益的图；

[0021] 图 25 示出了可以用于实施本发明的实施例的示例性处理；

[0022] 图 26 示出了可以用于实现本发明的实施例的示例性架构；以及

[0023] 图 27 示出了用于实现本发明的分布式实施例的示例性系统。

具体实施方式

[0024] 用于设计多输入多输出(MIMO)控制器的传统手段可以包括被称为 H∞合成的技术。

[0025] 传统的 H∞合成技术

[0026] 图 1 示出了用于表示标准 H∞合成技术的应用的配置。图 1 可以包括被控对象 H(s)110 和控制器 C(s)120。在图 1 中，标准 H∞合成计算最小化从输入 w130 至输出 z140 的闭环峰值增益的控制器 C(s)120。控制器 C(s)120 可以表示集中 H∞控制器，该控制器可以作为黑盒工作(例如，用户可能未准备好访问在黑盒内的内部表示)。H(s)110 可以表示可以使用控制器 C(s)120 控制的被控对象。被控对象 H(s)110 可以是线性被控对象模型。

[0027] 传统的 H∞合成可以用于最小化用于系统 100 的闭环响应(所谓的 H∞规范)的峰值输入 / 输出增益。用户可以具有用于控制问题的设计规范，该控制问题指定设计的方面或目标，诸如用于被控对象 H(s) 的带宽、滚降(roll-off)、过冲和 / 或稳定裕度。基于 H∞合成的框架可以适用于帮助用户实现规范；然而，极少用户(诸如工程师)对于使用传统的 H∞技术舒适或熟练。例如，用户可能发现将用户熟悉的普通规范转换为传统的 H∞合成技术所需的标准化的闭环增益约束是乏味、不直观和 / 或耗时的。另外，传统的 H∞合成工具和技术的技术限制可能进一步妨碍和 / 或迷惑通常的用户。

[0028] 传统的 H∞合成技术还可能对用户来说是不期望的，因为它们不支持用户的通常的设计工作流程。另外，传统的 H∞合成技术将控制器 C(s)120 看作黑盒，该黑盒不允许用户容易地访问控制器 C(s)120 表示的系统结构的内部表示。而且，传统的 H∞合成技术当产生用于给定系统的 C(s)120 时耦合独立的输入连接，并且在那些连接之间插入控制块。该手段导致不表示从其得到 C(s)120 的系统结构的 C(s)120 的结构，这进一步加剧了通常的用户在使用传统的 H∞合成技术工作时面对的困难。

[0029] 例如，用户可能期望能够提供基本上实时的操作的设计工具(即，工具允许用户交互地设计、修改和运行设计的方面，而不遭遇过度地烦扰用户或不利地干扰用户与设计工具的交互的处理延迟)。传统的 H∞合成技术可能因为多个原因而不支持交互操作。例如，传统技术可能试图当在控制器 C(s)120 上未施加特定结构或顺序约束时通过重复地求解一对黎卡提(Riccati)方程来计算最佳控制器。这些用于求解一对方程的重复的尝试可能是耗时的并且在计算上昂贵的，这倾向于使得传统的 H∞合成技术是不期望的，特别是对于交互设计应用而言。

[0030] 示例性技术和实施例的概述

[0031] 示例性实施例提供了一种新颖技术，该技术允许用户使用直观的用户界面来用公式表示和求解多变量反馈控制问题。例如，示例性实施例允许用户处理具有在控制系统中分布的、需要联合调整的多个控制元素的控制问题。实施例可以例如包括在一个或多个反馈回路上分布的、需要联合调整以最优化系统的整体性能和健壮性的控制元素。示例性实施例是可扩展的，并且可以被应用到具有实质上任何数量的组件、回路和 / 或任何复杂度的控制问题。

[0032] 示例性实施例可以利用类似于图 1 但是允许用户将控制器 C(s) 当作白盒而不是黑盒的系统表示。例如，示例性实施例允许控制器 C(s) 包括块对角结构的排列，其中，在对角线上的块本身具有某种固定结构和复杂度。实施例可以包括下述配置：其中，块对角仅表示可调整的组件，或者其中，块对角表示可调整的组件和模型的一个或多个固定或已知动力学（例如，固定 / 已知组件）。C(s) 的另外的复杂度允许用户以与被分析的系统的控制架构一致的方式来表示控制器 C(s)。示例性实施例可以使用非平滑 H ∞优化器来自动地调整任意的 MIMO 控制结构。示例性技术将求解器专门化为其中 C(s) 包括块对角结构的 H ∞合成表示的典型结构。

[0033] 示例性实施例和技术移除了当试图执行 MIMO 调整任务时由传统技术带来的阻碍。例如，示例性实施例和技术自动化将控制架构和控制器结构转换为适合于优化器的成本函数和参数向量的过程。示例性实施例和技术还允许花费不多地计算梯度（每一个迭代 o(N)，其中，N 是参数的数量），并且还允许以面向对象的方式来计算梯度，其中，每一个可调整块类型（增益 PID、传递函数、状态空间、原参数等）向梯度提供其本身的贡献。示例性实施例利用梯度，而不要求用户如在传统手段中进行的那样计算逐块的梯度并且将它们组合在一起以获得成本函数的整体梯度。传统手段还可以是手动难处理的（除了简单的架构之外），并且与在本发明的方面中使用的技术相比在效率上相差数量级。

[0034] 本发明的示例性技术当以计算硬件实现时允许用户交互地执行 MIMO 调整任务。例如，示例性实施例允许用户使用标准的个人计算机（PC）和诸如 MATLAB 技术计算和编程环境的技术计算环境来体验范围从小于一秒至小于 30 秒的典型调整时间。可以在多核或其他类型的多处理装置或环境中部署实施例，以减少处理时间。

[0035] 为了容易呈现，将结合线性控制系统论述本文所述的实施例；然而，本发明的实施例可以用于解决非线性控制问题。例如，示例性技术可以支持对于非线性控制设计的手段，包括但是不限于增益调度。

[0036] 示例性系统

[0037] 图 2 示出了用于实施实施例的示例性系统 200。系统 200 可以用于构造包括一个或多个实体的模型、设计和实现用于该模型的 PID 控制器并且 / 或者从该模型生成代码，例如，生成用于控制器的代码。系统 200 可以包括计算机 205、获取逻辑 210、操作系统 215、建模环境 220、模型 230、输入装置 240、显示装置 250、模型表示 260 和被控对象 270。在图 2 中的系统是说明性的，并且系统 200 的其他实施例可以包括更少的装置、更多的装置和 / 或具有与图 2 的配置不同的配置的装置。

[0038] 计算机 205 可以包括执行处理操作、显示操作、通信操作等的装置。例如，计算机 205 可以包括诸如一个或多个处理或存储装置的逻辑，其可以用于为用户执行和 / 或支持处理行为。计算机 205 的实施例可以包括台式计算机、膝上型计算机、客户机、服务器、大型计算机、个人数字助理（PDA）、能够使用网络的蜂窝电话、智能电话、智能传感器 / 致动器或执行指令以执行一个或多个行为和 / 或产生一个或多个结果的其他计算或通信装置。

[0039] 计算机 205 可以通过向另一个装置（在图 2 中未示出）发送数据或从其接收数据来进一步执行通信操作。数据可以指任何类型的机器可读信息，该信息实质上具有可以被适配来用在一个或多个网络中和 / 或用于一个或多个装置的任何格式。数据可以包括数字信息或模拟信息。数据可以进一步被分组和 / 或不分组。

[0040] 获取逻辑 210 可以从计算机 205 外部的装置获取数据，并且可以使得该数据可用于计算机 205。例如，获取逻辑 210 可以包括模数转换器、数模转换器、滤波器、复用器等，它们用于使得数据可用于计算机 205。计算机 205 可以使用所获取的数据来执行建模操作、控制器设计行为等。

[0041] 操作系统 215 可以管理与计算机 205 相关联的硬件和 / 或软件资源。例如，操作系统 215 可以管理与接收用户输入、操作计算环境 205、分配存储器、对系统请求区分优先次序等相关联的任务。在一个实施例中，操作系统 215 可以是虚拟操作系统。操作系统 215 的实施例可以包括 Linux、Mac OS、Microsoft Windows、Solaris、UNIX 等。操作系统 215 可以进一步在可以由计算机 205 提供的虚拟机上运行。

[0042] 建模环境 220 可以提供计算环境，该计算环境允许用户执行与学科相关的模拟或建模任务，该学科例如但是不限于数学、科学、工程、医学、商务等。建模环境 220 可以支持一个或多个应用，该一个或多个应用执行指令来允许用户构造具有可执行语义的模型。例如，在一个实施例中，建模环境 220 可以允许用户创建具有可执行语义的自由形式的模型 (例如，第一、第二、第三、第四、第五等级模型等)。建模环境 220 还可以支持基于时间、基于事件等的建模行为。

[0043] 模型 230 可以包括用于可执行的文本或图形模型的信息。例如，模型 240 可以包括用于可以是基于时间的模型、基于事件的模型、状态转换模型、数据流模型、组件图、实体流图表、基于等式的语言图表等的文本模型或图形模型的信息。模型 230 的图形实施例可以包括表示用于执行操作的可执行代码的实体 (例如，块、图标等)。可以执行用于该实体的代码以使用模型来执行模拟。可以使用表示用于在模型中从一个实体向另一个传送数据的路径的线来将实体连接在一起。

[0044] 输入装置 240 可以接收用户输入。例如，输入装置 240 可以将用户动作或行为转换为可以被计算机 205 解释的信号或消息。输入装置 240 可以包括但是不限于键盘、指示装置、生物计量装置、加速计、麦克风、照相机、触觉装置等。

[0045] 显示装置 250 可以向用户显示信息。显示装置 250 可以包括阴极射线管 (CRT)、等离子体显示装置、发光二极管 (LED) 显示装置、液晶显示 (LCD) 装置等。如果需要，显示装置 250 的实施例可以被配置来 (例如，经由触敏屏幕) 接收用户输入。在一个实施例中，显示装置 250 可以向用户显示一个或多个图形用户界面 (GUI)。GUI 可以包括模型 240 和 / 或其他类型的信息。

[0046] 模型表示 260 可以包括模型 230 的视觉表示和 / 或由模型 230 提供的视觉表示，例如绘图窗口。例如，模型表示 260 可以被向用户显示，并且可以包括通过线连接的多个实体。当执行模型 230 时，模型表示 260 可以改变以通过模型示出例如数据流。

[0047] 被控对象 270 可以包括向计算机 205 提供数据的一个或多个装置。例如，被控对象 270 可以包括使用诸如加速计、热电偶、光电收发器、应变仪等的传感器监控的引擎系统。在一个实施例中，获取逻辑 210 可以以模拟或数字形式从被控对象 270 接收信号，并且可以将该信号转换成适合于在计算机 205 中使用的形式。

[0048] 示例性建模环境

[0049] 图 3 示出了建模环境 220 的示例性实施例。建模环境 220 可以包括模拟工具 310、实体库 320、接口逻辑 330、编译器 340、控制器逻辑 350、优化器 360、模拟引擎 370、报告引

擎 380 和代码生成器 390。在图 3 中所示的建模环境 220 的实施例是说明性的，并且在不偏离本发明的精神的情况下，建模环境 220 的其他实施例可以包括更多实体或更少实体。

[0050] 模拟工具 310 可以是用于建立模型的应用。模拟工具 310 可以用于建立具有可执行语义的文本模型或图形模型。在图形模型的情况下，模拟工具 310 可以允许用户对模型实体和 / 或连接进行创建、修改、诊断、删除等。模拟工具 310 可以与图 2 或 3 中所示的其他实体交互，以接收用户输入、执行模型、显示结果、生成代码等。模拟工具 310 可以向用户提供 :编辑窗口，用于构造文本模型或与其交互 ;和 / 或，GUI，用于创建图形模型或与其交互。

[0051] 实体库 320 可以包括用户可以向包括模型表示 360 的显示窗口内拖放的代码模块或实体(例如，块 / 图标)。在图形模型的情况下，用户可以使用连接来进一步耦合实体，以产生系统的图形模型，诸如被控对象 370。

[0052] 接口逻辑 330 可以允许建模环境 220 向 / 从装置(例如，被控对象 270、目标环境等)或软件模块(例如，函数、应用程序接口等)发送或接收数据和 / 或信息。在一个实施例中，接口逻辑 330 可以将获取逻辑 310 与建模环境 220 连接。

[0053] 编译器 340 可以将模型编译成可执行格式。可以在计算机 205 上执行由编译器 340 产生的编译代码以产生建模结果。在一个实施例中，编译器 340 也可以提供用于诊断与模型相关联的错误的调试能力。

[0054] 控制器逻辑 350 可以用于在模型 330 中创建和实现控制器。例如，控制器逻辑 350 可以提供用于表示在模型表示 260 中的控制器的类型的实体的功能。当模型执行时，控制器逻辑 350 可以通过与模型表示 260 中的实体交互来对模型执行控制操作。在一个实施例中，控制器逻辑 350 可以包括在模型表示 360 中实现控制器的控制算法。控制器逻辑 350 的实施例可以被配置来以单独或分布实现方式运行。

[0055] 优化器 360 可以优化模型的代码、参数、性能(例如，执行速度)等。例如，优化器 360 可以优化代码以与在未优化代码的情况下该代码执行的情况相比较，使得该代码占用更少的存储空间，使得该代码更有效地执行，使得该代码更快地执行等。优化器 360 也可以对控制器逻辑 350 执行优化，以例如优化用于控制器的参数。在一个实施例中，优化器 360 可以与编译器 340、控制器逻辑 350、代码生成器 390 等一起运行或可以被集成到编译器 340、控制器逻辑 350、代码生成器 390 等内。可以经由软件对象来实现优化器 360 的实施例，该软件对象与其他面向对象的软件交互，以例如接收优化器 360 所操作的数据。

[0056] 模拟引擎 370 可以执行用于执行模型以模拟系统的操作。模拟引擎 370 可以被配置来基于用户偏好或系统偏好来执行单独或远程模拟。

[0057] 报告引擎 380 可以基于建模环境 220 中的信息来产生报告。例如，报告引擎 380 可以产生用于指示控制器是否满足设计规范的报告、用于指示控制器是否以稳定方式运行的报告、用于指示模型是否正确地编译的报告等。报告引擎 380 的实施例可以以用于在显示装置 250 上显示的电子格式、以硬拷贝格式和 / 或以适用于在存储装置中存储的格式来产生报告。

[0058] 代码生成器 390 可以从模型生成代码。在一个实施例中，代码生成器 390 可以接收第一格式的代码，并且可以将该代码从第一格式转换成第二格式。在一个实施例中，代码生成器 390 可以从模型的至少一部分生成源代码、汇编语言代码、二进制代码、接口信息、

配置信息、性能信息、任务信息等。例如，代码生成器 390 可以从模型生成 C、C++、SystemC、Java、结构化文本等代码。

[0059] 代码生成器 390 的实施例可以进一步从图形模型(例如，系统建模语言(SysML)、可扩展标记语言(XML)、实时和嵌入系统的建模和分析(MARTE)、硬件描述语言(HDL)、汽车开放系统架构(AUTOSAR)等)的一些或全部生成基于统一建模语言(UML)的表示和 / 或扩展。在一个实施例中，优化器 370 可以与代码生成器 390 交互，以生成根据参数(例如，存储器使用、执行速度、多处理等)优化的代码。与本发明的原理一致的建模环境的实施例可以进一步包括诸如验证组件、校验组件等的组件。

[0060] 本发明的实施例可以用于交互地以公式表示和求解多变量反馈控制问题，并且设计控制器以用在实质上任何顺序和 / 或延迟的非线性模型中。实施例可以被配置来使用精确的线性化技术以产生可以表示非线性模型的至少一部分的线性时间不变模型。

[0061] 示例性控制架构

[0062] 例如，本发明的实施例可以被应用到具有多个组件并且具有一个或多个反馈回路的控制架构，该多个组件除了别的之外可以包括以实质上任何顺序排列的控制器块。

[0063] 图 4 示出了可以应用本发明的实施例的示例性控制架构，即，在高攻击角度模式中的 F-14 中使用的自动驾驶仪。可以在 GUI400 中显示该自动驾驶仪，并且该自动驾驶仪可以包含包括增益和时间常数的 8 个可调整参数。

[0064] 图 5 示出了用于在蒸馏塔中使用的控制器的示例性架构。图 5 的控制器可以经由 GUI500 向用户显示，并且可以包括需要调整的四个比例积分(PI)增益和 2×2 增益矩阵。

[0065] 图 6 示出了用于对于风力涡轮机的俯仰与偏航控制的示例性架构。图 6 的架构可以经由 GUI600 向用户显示，并且可以包括需要调整的三个 PI 控制器和两个增益。如图 6 所示，用于本发明的实施例的 GUI 可以包括多个界面，诸如窗口、窗格等。

[0066] 本发明的示例性实施例可以用于其他类型的架构和与在图 4-6 中和在本申请的其他位置所示的架构相比更复杂或更不复杂的架构。例如，实施例可以用于包括具有实质上任何数量的固定和 / 或可调整组件的反馈布置或前馈布置的架构。

[0067] 在此公开的示例性实施例和 / 或技术通过允许将结构简化为单个一般表示来允许任意控制结构的有效调整。

[0068] 示例性典型结构

[0069] 图 7 示出了可以用于表示变换的任意控制结构的示例性典型结构。

[0070] 参见图 7，系统 700 可以经由界面向用户显示，并且可以包括 H(s)710，H(s)710 可以是线性模型，该线性模型将控制系统的非调整(例如，固定的、不变的、没有要调整的自由参数等)组件组合为单个集中模型。实施例可以适用于可以被简化为线性模型(例如，分布式模型)的任何系统表示。例如，在本发明的一个实施例中，H(s)710 可以表示系统的固定控制动力学(例如，组件)。在一个实施例中，固定组件可以是不被调整的、不包括可以变化的自由参数等的组件。

[0071] 系统 700 可以进一步包括控制器 720，控制器 720 可以包括要调整的元素。在一个实施例中，可调整元素可以是具有变化的参数的组件，该参数例如是将被调整的自由参数。例如，在一个示例性实施例中，控制器 720 可以是包括一个或多个块 $B_1$730 至 B_N740 的结构化控制器，该一个或多个块表示要调整的控制元素。例如，当控制器 720 包括单个块时，该

块被称为 B_1，并且，具有三个块的控制器 720 包括块 B_1、B_2 和 B_3。要调整的控制元素可以在设计上变化。例如，要调整的元素可以包括在被控对象或控制器中的增益、动态元素（例如，传递函数、状态空间模型等）和 / 或设计参数。

[0072] 控制器 720 的示例性实施例可以包括能够被简化为块 B_1 至 B_N 的块对角集合的项，并且可以被称为结构化控制器 720。结构化控制器 720 可以包括重复的块（即，诸如 B_2 的特定块可以沿着对角线出现多次）。结构化控制器 720 可以被配置为仅包括沿着对角线的可调整组件或者包括沿着对角线的可调整和固定组件。结构化控制器 720 还可以包括在除了在块对角线上的位置之外的位置处的至少一个非零项。实施例可以进一步包括在块对角线上的为零的元素。

[0073] 可以使用在频域中的对应的等式来表示系统 700，该等式例如是：

[0074]
$$\begin{cases} \begin{bmatrix} z \\ y \end{bmatrix} = H(s) \begin{bmatrix} w \\ u \end{bmatrix} \\ u = \begin{bmatrix} B_1(s) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & B_N(s) \end{bmatrix} y \end{cases} \quad \text{（等式 1）}$$

[0075] 在等式 1 中，H(s) 和被调整的块（B_1 至 B_N）通过信号 u 和 y 以反馈方式来交互。等式 1 的典型结构具有三个属性：

[0076] （1）其中 B_1、…、B_N 显现为块的信号流程图可以通过下述方式被变换为这个典型结构：将块 B_1、…、B_N 与图的剩余部分分离，并且附加它们的输入和输出以创建块对角结构。

[0077]
$$C(s) = \begin{bmatrix} B_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & B_N \end{bmatrix} \quad \text{（等式 2）}$$

[0078] （2）其系数是一些参数 p_1、…、p_N 的有理函数的传递函数或状态空间模型可以被变换为这个典型形式，其中，块 B_1、…、B_N 具有下述形式

[0079]
$$B_j = \begin{bmatrix} p_j & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & p_j \end{bmatrix} \quad \text{（等式 3）}$$

[0080] （3）该典型表示在串连、并联和反馈互连下是闭合的，用于表示这样的典型结构的任何互连得到具有较大的块列表的另一种典型结构。

[0081] 上面的属性（1）-（3）确保用于调整与本发明的原理一致的图 7 的典型结构的任何方法适用于实际上任何控制结构，该控制结构具有任何顺序和结构的线性控制元素。

[0082] 用于参数化可调整块的示例性技术

[0083] 示例性实施例可以被配置来参数化在系统中的可调整元素，以使用公式表示作为优化问题的调整任务。例如，在一个实施例中，可以参数化要调整的系统元素。表 1 汇总了诸如比例积分微分（PID）块、固定阶传递函数块和固定阶状态空间块的一些公共控制元素的参数化。

[0084]

可调整控制元素	参数化

[0085]

增益块	增益矩阵的每一个条目是参数
PID 块 $B(s) = K_p + \frac{K_i}{s} + \frac{K_d s}{1 + T_f s}$	Kp、Ki、Kd 和 Tf 是参数
固定阶传递函数块 $B(s) = \frac{b_m s^m + \cdots + b_0}{s^n + a_{n-1} s^{n-1} + \cdots + a_0}$	分子和分母系数是参数
固定阶状态空间块 $B(s)\begin{cases}\dot{x} = Ax + Bu \\ y = Cx + Du\end{cases}$	A、B、C 和 D 矩阵的条目是参数。默认地，我们将 A 限制为三对角线的，这是完全通用的，并且减少了所需的参数的数量。

[0086] 表 1. 所选择的控制元素的参数化

[0087] 示例性实施例可以进一步被配置来提供和 / 或使用基于软件的工具，以允许用户通过写入涉及基本参数的表达式来创建控制元素的定制参数化。例如，在本发明的实施例中使用的示例性软件工具可以允许用户创建具有另外的结构的可调整块，诸如但是不限于：

[0088] 通过实数值 a 参数化的低通滤波器 a/(s+a)

[0089] 具有观测器形式的状态空间控制器：

[0090] $$\begin{cases}\dot{x} = Ax + Bu + L(y - Cx - Dy) \\ u = -Kx\end{cases}$$ （等式 4）

[0091] 在等式 4 中，参数是增益矩阵 K 和 L。

[0092] 用于多目标支持的示例性技术

[0093] 例如，可以提供两个 H ∞约束，即

[0094] $||T_1||_\infty < 1, ||T_2||_\infty < 1$

[0095] 涉及两个单独的闭环传递函数

[0096] $$\begin{cases}T_1 = F(H_1, C) \\ T_2 = F(H_2, C)\end{cases}$$ （等式 5）

[0097] 其中，F(.,.) 表示在图 7 中描述的线性分式变换（LFT）互连，并且

[0098] $$C(s) = \begin{bmatrix} B_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & B_N \end{bmatrix}$$ （等式 2）

[0099] 表示要调整的结构化补偿器。

[0100] 可以将两个 H ∞约束（$||T_1||_\infty < 1$，$||T_2||_\infty < 1$）组合为对于集合传递函数

[0101] $$T(s) = F\left(\begin{bmatrix} T_1 & 0 \\ 0 & T_2 \end{bmatrix}, \begin{bmatrix} C & 0 \\ 0 & C \end{bmatrix}\right)$$ （等式 6）

[0102] 的单个 H ∞约束

[0103] $||T||_\infty < 1$

[0104] 即使当 C(s) 是全阶多输入多输出（MIMO）控制器时，因为结果产生的控制器的 $\begin{bmatrix} C & 0 \\ 0 & C \end{bmatrix}$ 块对角结构，传统的 H ∞合成技术不能解决结果产生的问题。

[0105] 与传统的 H ∞合成技术相反，本发明的示例性实施例支持 $\begin{bmatrix} C & 0 \\ 0 & C \end{bmatrix}$ 块对角结构以及块重复。因此，本发明的示例性实施例可以用于解决传统的 H ∞合成技术失败的问题，诸如多目标问题：

[0106] 调整 C(s) 以使得 $||T_1||_\infty < 1$ 和 $||T_2||_\infty < 1$

[0107] 或者等同地

[0108] 调整 C(s) 以使得 $||T||_\infty < 1$

[0109] 与求解单个目标问题没有不同。从实践的观点看，示例性实施例允许用户独立地约束每一个点对点传递函数（例如，环路传递函数），以使得用户不必提出捕获所有要求的单个 MIMO 约束。这是许多用户期望的特征，因为由于干扰用户建立问题的方式的交叉项，提出捕获所有要求的单个 MIMO 约束通常对于用户是挑战性的任务。

[0110] 用于自动地使用公式表示回路成形要求的示例性技术

[0111] 回路成形是用于调整反馈控制系统的常用频域技术。例如，对于单输入单输出（SISO）反馈回路，回路成形过程可以由下述部分构成：

[0112] 1. 在用于开环响应 L(s) 的期望增益配置（作为频率的函数的增益）上表达设计要求。例如，增益应当在低频处高（>1）以用于良好的跟踪和干扰抑制，并且，增益应当在高频处低（<1）以用于对于噪声和建模误差的不灵敏。转换频率（增益 = 1）被称为交叉频率 ω_c，并且直接地影响控制系统的响应速度。

[0113] 2. 在保持闭环稳定性和足够的稳定裕度的同时，调整补偿器的可调整参数以接近期望的回路形状。

[0114] 对于多回路控制系统，该过程可以以如下所述的一些调整被应用到 MIMO 开环响应。

[0115] 上文中紧邻的步骤（1）通常被熟悉频域技术的控制工程师良好地理解。例如，可以通过在图 8 中所示的绘图 800 来表示用于具有积分作用的反馈回路的期望的回路形状。在一个实施例中，绘图 800 可以使用显示装置 250 经由 GUI 向用户显示。

[0116] 参见图 8，0dB 交叉频率是 ω_c=10（ω_c 在图 8 中的插图编号 805 处），其对应于大约 1/10 = 0.1 秒的响应时间。在图 8 中，响应时间可以指示反馈回路多快地响应于改变。增大或减小 ω_c 可以分别加速或减缓响应。在图 8 中的回路形状对于 w<10（区域 810）具有高增益，并且对于 w>10（区域 820）具有低增益。

[0117] 示例性实施例和技术适合于调整在系统中的补偿器的可调整参数，以达到或接近期望的回路形状。用于本发明的实施例的技术可以受益于将该回路成形目标（即，将开环增

益成形以匹配指定的配置）使用公式表示为一些 H ∞规范约束。存在用于在 H ∞规范约束中将回路成形目标使用公式表示的几种已知技术；然而，这样的转换经常妨碍在健壮控制理论和 H ∞合成方面未培训过的工程师。示例性实施例使用简化重新形成步骤的技术以便消除工程师（例如，用户）在使用传统手段时所面临的困难。例如，实施例可以自动化重新形成步骤以从用户去除配置和 / 或计算负担。

[0118] 图 9 示出了可以用于实现本发明的方面的系统 900。系统 900 可以包括与输入信号 r、输出信号 y、控制器 C(s)910 和被控对象 G(s)920 一起的 SISO 和 MIMO 反馈回路。系统 900 还包括信号 u 和 n。在图 9 中，C(s)910 可以包括一个或多个可调整块，并且可以假定信号 u 和 y 包括相同数量的条目（即，相同数量的控制和测量）。

[0119] 在图 9 中，可以将开环传递函数表示为：

[0120] L=GC （等式 7）

[0121] 并且，闭环传递函数是具有注释：

[0122] $S=(I+L)^{-1}$，T=LS （等式 9）

[0123] 的

[0124] $$\begin{bmatrix} y \\ e \end{bmatrix} = \begin{bmatrix} T & -T & GS \\ S & -S & GS \end{bmatrix} \begin{bmatrix} r \\ n \\ d \end{bmatrix} \quad \text{（等式 8）}$$

[0125] 在给出目标回路形状 ρ（s）和干扰抑制因子 β 的情况下，考虑

[0126] $$X(s) = \begin{bmatrix} 1 & 0 \\ 0 & \rho \end{bmatrix} \begin{bmatrix} T & -T & GS \\ S & -S & GS \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1/\rho & 0 \\ 0 & 0 & 1/\beta \end{bmatrix} \quad \text{（等式 10）}$$

[0127] 可以验证，H ∞约束

[0128] $||X||_{\infty} < 1$

[0129] 确保：

[0130] $$\begin{cases} \|S(j\omega)\| < \min\left(1, \frac{1}{\rho(\omega)}\right) \\ \|T(j\omega)\| < \min\left(1, \rho(\omega)\right) \\ \|GS(j\omega)\| < \beta \min\left(1, \frac{1}{\rho(\omega)}\right) \end{cases} \quad \text{（等式 11）}$$

[0131] 因为

[0132] ● S ≈ 1/L 和 T ≈ 1，其中，回路增益 ||L（jω）|| 大于 1

[0133] ● T ≈ L 和 S ≈ 1，其中，回路增益小于 1

[0134] 所以，这等同于：

[0135]
$$\begin{cases} \|L(j\omega)\| > \rho(\omega) \quad \text{其中} \ \|L(j\omega)\| > 1 \\ \|L(j\omega)\| < \rho(\omega) \quad \text{其中} \ \|L(j\omega)\| < 1 \\ \|S(j\omega)\| < 1 \\ \|T(j\omega)\| < 1 \\ \|GS(j\omega)\| < \beta \min\left(1, \frac{1}{\rho(\omega)}\right) \end{cases} \qquad \text{（等式 12）}$$

[0136] 等式 12 的前两个约束通过使得回路增益足够大或足够小以远离交叉频率来实现期望的回路形状。例如，实现方式可以包括用于回路增益的期望值，诸如远离交叉频率的上值和下值。等式 12 的第三和第四约束可以用于将 S 和 T 的增益保持为在交叉附近接近 1，这对于确保足够的过冲和稳定裕度是重要的。最后，图 12 的最后约束使得用户通过减小 β 来改善干扰抑制属性。

[0137] 通过总结上面关于图 9 的论述，已经说明了 H ∞约束

[0138] $||X||_{\infty} < 1$

[0139] 传递了期望的回路形状，提供了足够的裕度，并且通过 β 来优化干扰抑制。这提供了用于将回路成形目标转换为适合用于本发明的示例性技术的 H ∞约束的系统方式。例如，示例性实施例和技术允许创建可以仅使用来自用户的下面的基本信息来工作的软件工具，该基本信息例如是：

[0140] ●目标回路形状 ρ (s)（在一个实施例中，仅增益可能有影响）

[0141] ●可调整块

[0142] ●在闭环系统的框图中的信号 r、d、y 的位置

[0143] 从该信息，用于实现本发明的方面的软件工具可以使用与本发明的方面一致的结构化 H ∞合成技术来自动地提取 G 和 C，构造 X(s) 并且调整块参数。这些软件工具和技术可以通过下述方式进一步消除对于目标回路形状 ρ (s) 的需要：从期望的交叉频率 ω_c 自动地生成适当的回路形状，并且判断是否需要积分作用。

[0144] 本发明的软件实现的方面也可以适用于 MIMO 问题。例如，示例性技术可以被应用到具有四个单独的反馈回路的机器人臂的控制，每一个反馈回路控制特定的关节。在 MIMO 情况下，细微的改良可以包括使用可以与用于 SISO 问题的回路形状不同的回路形状。

[0145] 图 10 示出了根据本发明的方面的可以用于处理 MIMO 问题的回路形状。在图 10 中，轨迹 1010 包括具有 0dB 增益的数量级的整平部分 1015。整平部分 1015 允许一定范围的交叉频率。允许一定范围的交叉频率可以帮助解释下述事实：在耦合的多回路控制系统中，所有的回路不可能在同一频率下跨越 0dB。

[0146] 用于实现本发明的方面的示例性的基于软件的工具

[0147] 本发明的软件实现的实施例向用户提供了界面，诸如允许用户以通常的用户容易明白的格式和 / 或使用通常的用户容易明白的术语来输入信息的图形用户界面（GUI）。软件实现的实施例进一步使用支持控制系统的快速设计和 / 或分析的算法和处理技术。事实上，本发明的实施例可以被配置来支持允许用户利用工作流的交互控制设计和分析应用，该工作流促进控制系统的有效设计和分析。

[0148] 例如，当示例性实施例使用用于实现本文所述的技术的一个或多个软件工具时，

软件工具可以向用户提供服务，诸如但是不限于：

[0149] 1．指定控制元素的结构，并且自动地生成该元素的参数化；

[0150] 2．创建定制元素和参数化；

[0151] 3．将给定的控制架构映射到图 7 的对应的典型结构；

[0152] 4．将简单设计要求转换为 H ∞约束；

[0153] 5．通过求解基础结构化的 H ∞合成问题来自动地调整控制元素；

[0154] 6．在原始控制元素方面呈现结果；以及

[0155] 7．通过与普通线性分析工具的紧密集成来促成设计的验证。

[0156] 为了呈现的清楚，结合 MATLAB 编程语言和 / 或诸如 MathWorks 的控制系统工具箱的 MATLAB 语言兼容工具箱来描述示例性实施例和技术。在不偏离本发明的精神的情况下，可以在其他编程环境中、使用其他软件包和 / 或应用等来实现与本发明的实施例和 / 或与实施例相关联的技术。例如，可以使用应用来实现实施例和 / 或技术，该应用包括但是不限于 MATLAB 兼容产品 / 语言 Octave、Python、Comsol Script、来自 National Instruments 的 MATRIXx、来自 Wolfram Research 公司的 Mathematica、来自 Mathsoft Engineering&Education 公司的 Mathcad、来自 Maplesoft 的 Maple、来自 Imagine That 公司的 Extend、来自法国计算机科学和控制研究所（INRIA）的 Scilab、来自 Cadence 的 Virtuoso、来自 Dynasim 的 Modelica 或 Dymola、C/C++ 库等等。

[0157] 在一些实施例中，用于实现本发明的技术的环境可以包括提供计算环境的硬件或基于硬件 / 软件的逻辑，该计算环境允许用户比在另一种类型的计算环境中执行任务更有效地执行与学科相关的任务，该学科例如但是不限于数学、科学、工程、医学、商务等，该另一种类型的计算环境例如是要求用户以诸如 C++、C、Fortran、Pascal 等的传统编程语言来开发代码的环境。

[0158] 在一种实现方式中，该环境可以包括动态键入的语言，该语言可以用于以相关领域内的技术人员熟悉的数学表示来表达问题和 / 或解。例如，该环境可以使用阵列来作为基本元素，其中，该阵列可能不要求确定尺寸。这些阵列可以用于在下述方面支持阵列编程：操作可以应用到整个一组值，诸如在阵列中的值。阵列编程可以允许基于阵列的操作被看作高级编程技术或模型，这使得编程者考虑和操作于整个数据集合，而不必求助于单独的非阵列的显示循环，即，标量操作。

[0159] 该环境可以进一步被适用于执行矩阵和 / 或向量表述，该表述可以用于数据分析、数据可视化、应用开发、模拟、建模、算法开发等。可以在许多领域中使用这些矩阵和 / 或向量表述，该许多领域例如是统计、金融、图像处理、信号处理、控制设计、生命科学、教育、离散事件分析和 / 或设计、基于状态的分析和 / 或设计等。

[0160] 在必要时，可以在基于图形的环境中使用以下产品来实现与本发明的方面一致的实施例和技术，该产品例如但是不限于：Math Works 公司的 Simulink® 建模环境、Stateflow® 环境、SimEvents™ 环境等；Simulink/Stateflow/SimEvents 兼容的产品；Visual Solutions 的 VisSim；National Instruments 的 Lab VIEW®; Dynasim 的 Dymola；Measurement Computing 的 SoftWIRE；DALSA Coreco 的 WiT；Agilent 的 VEE Pro 或 System Vue；来自 PPT Vision 的 Vision Program Manager；来自 Khoral Research 的 Khoros；Gedae

公司的 Gedae ;来自(INRIA)的 Scicos ;来自 Cadence 的 Virtuoso ;来自 IBM 的 Rational Rose ;来自 Telelogic 的 Rhopsody 或 Tau ;来自加州大学伯克利分校的 Ptolemy ;或者,统一建模语言(UML)或 SysML 环境的方面。

[0161] 用于指定可调整块的示例性技术

[0162] 本发明的实施例可以允许用户指定在系统中的可调整块,其中,可调整块是被调整以获得最佳性能和健壮度的控制系统的组件。例如,在本发明的一个实施例中,控制系统的可调整组件可以对应于在图 7 的典型结构中的块 B_1、…、B_N。

[0163] 示例性实施例可以使用用于控制诸如增益、PID、传递函数和状态空间模型的元素的对象。该对象可以进一步用于解释可调整参数,并且允许用户初始化、设置(例如,固定)或释放参数。

[0164] 例如,用户可以与诸如文本用户界面的界面交互,以创建具有两个输入和一个输出的可调整增益。建模环境 220 可以生成界面 1100。例如,界面 1100 可以是经由显示装置 250 向用户显示的文本编辑窗口。参见图 11,在一个示例性实施例中,用户可以在提示(例如,“>>”)下输入命令 1105。

[0165] 响应于命令 1105,建模环境 220 可以返回可以经由界面 1100 向用户显示的对象 G1110。在一个实施例中,建模环境 220 可以被配置来以与用于显示与建模环境 220 兼容的其他类型的对象的格式类似的格式来显示对象 G1110。例如,对象 G1110 可以类似于在控制系统工具箱中的普通线性时间不变(LTI)对象,除了可以包括可调整参数的说明而不是数值的“增益”属性之外。例如,并且参见图 12,用户可以输入命令 1205,并且建模环境 220 可以返回可以包括用于 G.Gain 的值的回答 1210。

[0166] 仍然参见图 12,回答 1210 可以包括用于 G.Gain 的默认值。在一个示例性实施例中,默认值可以是用于建模环境 220 的当前值,直到用户将一个或多个值从默认值改变为另一个值。例如,增益的默认 / 当前“值”在回答 1210 中是 [00]。用户能够通过输入命令 1215 和 1220 来将增益矩阵的第一条目设置为 1。例如,用户可以输入 :

[0167] >>G.Gain.Value(1)=1

[0168] >>G.Gain.Free(1)=false

[0169] 然后用户可以输入 G.Gain,并且建模环境 220 可以返回输出 1225。在图 12 中,输出 1225 可以显示反映这个新的约束的增益矩阵的参数化(即,将“值”从默认设置改变为用户指定的设置)。

[0170] 示例性实施例可以允许用户在界面 1300 内分别使用在图 13 中所示的命令 1305、1310 和 1315 来创建其他类型的块,诸如可调整 PID 块、传递函数块或状态空间块。

[0171] 示例性实施例可以向用户提供可以处理典型的或主流的需要的预定义块。对于一些用户,预定义块可能足以满足它们的需要。然而,其他用户可能具有不同的需要,诸如对于诸如具有与在预定义块中的结构相比较更多结构的块的、更复杂和 / 或健壮的控制元素的需要。

[0172] 示例性实施例可以适用于支持用户要求与预定义块不同的块的需要,与预定义块不同的块例如是低通滤波器块、陷波滤波器块、观测器形式的状态空间模型等。例如,实施例可以被配置来提供与预定义块不同的一套块 ;然而,该套可能需要包括大量的块来满足特定用户的需要。在本发明的实施例中,提供了框架,其允许用户使用简单的代数来构造它

们本身的参数化。该框架消除了保持试图覆盖用户的所有需要的一大套块的需要。

[0173] 在一个实施例中，该框架可以包括：

[0174] 1. 包含参数的基本概念的“参数”对象；以及

[0175] 2. 将对于参数的普通操作自动地变换为图 7 的典型结构的计算机代数。

[0176] 例如，考虑低通滤波器：

[0177] $$F(s) = \frac{a}{s+a} = \frac{a/s}{1+a/s}$$ （等式 13）

[0178] 其中，a 是实数参数。等式 13 是一阶传递函数；然而，上面引入的传递函数块（图 13，命令 1310）不能表达下述约束：同一系数 a 在分子和分母两者中出现（换句话说，DC 增益被约束为 1）。在一个示例性实施例中，用户可以通过输入命令 1320（图 13）来指定这个控制元素，该命令例如是：

[0179] >> s=tf('s');

[0180] >> a=realp('a',2);

[0181] >> F=feedback(a/s,1)

[0182] 建模环境 220 处理命令 1320，并且，返回 @genss 对象“F”，其中，返回的对象包含以“a”为唯一的块的图 7 的典型结构。用户可以向界面 1300 内键入命令 1325，并且建模环境 220 可以返回回答 1330，诸如：

[0183] ans=

[0184] a:[1x1 realp]

[0185] 本发明的实施例可以使用用于创建 F 的句法，该句法与“a”仅是普通的双值的情况相同。该特征消除了用户必须学习用于与参数交互的新的 API 的需要。例如，用于在控制系统工具箱中操纵普通 LTI 对象的句法被无缝地延伸以处理参数和参数模型。在必要时，本发明的实施例也可以支持用于与参数交互的替换句法。例如，可以经由命令 1335（图 13）来表达用于指定参数低通滤波器 F(s) 的替换句法，该命令例如是：

[0186] >> a=realp('a',2);

[0187] >> F=tf(a,[1a])

[0188] 命令 1335 的替换句法也类似于（即，并行于）经由命令 1340 表达的用于在控制系统工具箱中创建普通传递函数的句法，该命令 1340 例如是：

[0189] >> a=2;

[0190] >> F=tf(a,[1a])

[0191] 输入命令 1335 或 1340 可以返回回答 1345，该回答 1345 经由界面 1300 来显示结果产生的传递函数。

[0192] 本发明的实施例可以用于解决比上面的示例更复杂的特性的控制问题。例如，可以将观测器形式的状态空间控制器提供为：

[0193] $$\begin{cases} \dot{x} = Ax + Bu + L(y - Cx - Dy) \\ u = -Kx \end{cases}$$ （等式 14）

[0194] 在等式 14 中，A、B、C、D 是已知矩阵，并且可调整变量是增益矩阵 K（状态反馈增益）和 L（观测器增益）。该控制器的结构化的状态空间参数化例如有益于增益调度。为了

说明性目的，当使用本发明的实施例时，可以假定存在两个测量 y、一个控制信号 u 和三个状态 x。用户可以通过键入命令 1410（图 14）来创建该可调整的块，该命令 1410 例如是：

[0195] >> K=realp('K',zeros(1,3));

[0196] >> L=realp('L',zeros(3,2));

[0197] >> OBC=ss(A-B*K-L*C+L*D*K,L,-K,0)

[0198] 用户可以然后键入命令 1420，诸如：

[0199] >> OBC.Blocks

[0200] 建模环境 220 可以处理该命令，并且可以经由界面 1400 来显示回答 1430，诸如：

[0201] ans=

[0202] K:[1x3realp]

[0203] L:[3x2realp]

[0204] 与结合图 13 所述的示例一样，结合图 14 描述的示例使用与用于在建模环境 220 中创建普通状态空间模型的句法类似（即，大体成镜像）的句法。在一个实施例中，输入图 14 的命令的结果是封装图 7 的典型结构的 @genss 模型，其具有结构补偿器：

[0205] $$C(s)=\begin{bmatrix} K & 0 & 0 & 0 & 0 & 0 \\ 0 & K & 0 & 0 & 0 & 0 \\ 0 & 0 & K & 0 & 0 & 0 \\ 0 & 0 & 0 & L & 0 & 0 \\ 0 & 0 & 0 & 0 & L & 0 \\ 0 & 0 & 0 & 0 & 0 & L \end{bmatrix}$$ （等式 15）

[0206] 往回参见图 7，对于图 7 的典型结构存在两段，即，（1）互连模型 H(s) 和（2）可调整块 B_1、…、B_N。图 8-14 和相关文本已经论述了用于指定可调整块（在前一个句子中的项（2））的工具和技术。下面讨论用于建立互连模型 H(s) 的工具和技术。

[0207] 用于得出典型结构的示例性技术

[0208] 示例性实施例和技术可以用于支持用于与任意控制结构交互和 / 或调整任意控制结构的两种或更多的手段。例如，实施例和技术可以支持（1）使用在建模环境（例如，图形建模环境，诸如 Simulink 建模环境、Simulink 兼容环境等）内存在的工具来自动地提取 H(s)。例如，可以使用在 Simulink 建模环境中的现有线性化工具来自动提取 H(s)。并且，（2）可以使用与文本计算和 / 或建模环境（例如，MATLAB 编程环境、MATLAB 兼容环境等）相关联的标准命令来自动地建立 H(s)。例如，与 MATLAB 编程环境交互的用户可以使用用于建立框图的命令来建立 H(s)。

[0209] 示例性实施例可以用于在诸如图形建模环境的基于图形的环境中指定可调整块。例如，用户可以与 Simulink 建模环境交互以指定用在控制系统中的可调整块。

[0210] 在 Simulink 环境中，可以通过了解下述部分来完全确定 H(s)：（1）要调整的块 B_1、…、B_N（图 7）；以及，（2）外部 I/O 信号 w 和 z（图 7）。当可调整块在框图中作为 Simulink 块出现时，该信息容易被提供为：

[0211] 1. 到块 B_1、…、B_N 的每一个的块路径的列表；以及

[0212] 2. 线性化输入和输出点，用于指定构成 w 和 z 的信号在何处进入和退出框图。

[0213] 一旦指定这个信息，诸如 LINLFT 的命令可以计算和返回作为 LTI 模型的 H(s)。上

面的过程可以被应用到线性和非线性模型。当向非线性模型应用该过程时，产生控制架构的线性化模型。

[0214] 在诸如MATLAB环境的文本编程环境中的H(s)的得出可以依赖于上面对于可调整块的指定所讨论的计算机代数(例如，图11-14和相关文本)。示例性实施例使得用户有可能扩展用于组合普通LTI模型的典型操作(例如，如可以使用MathWorks控制系统工具箱进行那样)，以也处理在图7中所示的参数对象、参数块和参数LFT模型的类型。一些典型操作的示例可以包括但是不限于：

[0215] ●普通代数：加、减、乘、除、求逆等；以及

[0216] ●互连操作：输入或输出级联、附加、并行、串行、反馈、LFT、连接等。

[0217] 示例性实施例利用静态和动态线性分式变换(LFT)模型(被定义为具有可调整块的块对角布置的矩阵或LTI模型的LFT互连，如在图7中所示)的面向对象的框架与概念。示例性实施例可以创建对象系统，该对象系统允许用户组合双阵列、LTI模型、参数、可调整块和LFT模型。例如，LFT模型是最一般的模型，并且可以是三种类型，诸如与H的三种不同的数值表示对应的@genmat、@genss和@genfrd，该三种不同的数值表示分别例如是双阵列、状态空间模型、频率响应数据模型。结果产生的对象系统是闭合的，并且实质上，LTI模型与可调整(参数)元素的任何组合将产生具有图7的典型结构的@genss或@genfrd模型。

[0218] 图15示出了可以使用由本发明的示例性实施例提供的基于对象的框架来交互的可执行图形模型1500。图15可以包括模型1500，模型1500可以包括多个可执行块。在模型1500中的一些块可以不是可调整的，诸如范围块1505和蒸馏塔(被控对象)块1510(以下称为被控对象块1510)。相反，在模型1500中的其他块可以是可调整的，诸如增益块1515与PI控制器块1520和1525。建模环境220的实施例可以被配置来使用第一类型的视觉标识符(例如，线宽、阴影、颜色、填充图案、文本图案等)来指示可调整块，并且使用第二类型的视觉标识符来指示不可调整的块。

[0219] 仍然参见图15，可以在模型1500中提供被控对象块1510的LTI模型G，并且用户能够与模型1500交互，以构造从r向y的闭环传递的LFT模型。例如，用户可以输入命令1605和1610(图16A)，诸如：

[0220] %指定可调整块

[0221] >> DM=ltiblock.gain('Decoupler',eye(2));

[0222] >> PI_L=ltiblock.pid('PI_L','pi');

[0223] >> PI_V=ltiblock.pid('PI_V','pi');

[0224] 以及

[0225] %得出闭环传递T

[0226] >> C=append(PI_L,PI_V)*DM;

[0227] >> T=feedback(G*C,eye(2));

[0228] 建模环境220可以处理命令1605和1610，并且可以产生@genss模型T。用户可以与建模环境220交互，并且可以通过输入命令1615将模型T分解为其部分H和B(如在图7中那样)，命令1615例如是：

[0229] >> [H,B]=getLFTModel(T)

[0230] 建模环境220可以处理命令1615，并且可以返回回答1620，诸如：

[0231]

a =

	x1	x2
x1	-0.01333	0
x2	0	-0.01333

[0232]

b =

	u1	u2	u3	u4	u5	u6
x1	0	0	2	0	0	0
x2	0	0	0	2	0	0

[0233]

c =

	x1	x2
yD	0.5853	-0.576
yB	0.7213	-0.7307
?	-0.5853	0.576
?	-0.7213	0.7307
?	-0.5853	0.576
?	-0.7213	0.7307

[0234]

d =

	u1	u2	u3	u4	u5	u6
yD	0	0	0	0	0	0
yB	0	0	0	0	0	0
?	1	0	0	0	1	0
?	0	1	0	0	0	1
?	1	0	0	0	0	0
?	0	1	0	0	0	0

[0235] 连续时间模型。

[0236] B=

[0237] [1x1 ltiblock.pid]

[0238] [1x1 ltiblock.pid]

[0239] [2x2 ltiblock.gain]

[0240] 用于优化可调整参数的示例性技术

[0241] 一旦用户已经与模型 1500 和建模环境 220 交互以使用公式将设计要求表示为在某个加权闭环模型 T(s) 上的 H ∞约束

[0242] $||T(s)||_\infty < 1$

[0243] 则用户可以通过输入命令来调用优化器，该命令例如是：

[0244] >> T=hinfstruct(T0,options)

[0245] 其中，T0 是未调整的闭环模型，并且 options 是用于优化器的某个选项集。建模

环境 220 可以处理命令，并且可以返回调整后的闭环模型 T，其中，可调整参数被设置为由优化器发现的某些值。例如，在一个实施例中，所有可调整参数可以被设置为由优化器 360 确定的最佳值。

[0246] 用户可能希望访问用于一个或多个可调整块的调整值。例如，在一个实施例中，用户可以通过输入命令来访问每一个可调整块的调整值，该命令例如是：

[0247] >> T.Blocks

[0248] 在特定情况下，C0 可以由包括一个或多个可调整块的补偿器构成。在这些情况下，用户可以使用本发明的示例性实施例向用于 C0 的 LFT 模型“推送”调整后的参数值。例如，用户可以键入命令，诸如：

[0249] >> C=replaceBlock(C0,T.Blocks)

[0250] 建模环境 220 可以处理该命令，并且可以将在 C0 中的相关参数值自动地替换为它们在 T 中的调整值。示例性实施例可以包括诸如帮助函数的函数，以用于促进合理化在诸如 Simulink 或 Simulink 兼容模型的模型中的块参数的调整。例如并且参考 Simulink 模型，

[0251] ● LINLFT 可以被扩展来基于被选择来用于调整的 Simulink 块而自动地拾取和配置正确的可调整块对象（ltiblock.gain、ltiblock.tf、...）

[0252] ●帮助函数可以被提供来向 Simulink 块推回由 hinfstruct 计算的调整后的参数值，

[0253] 帮助函数可以包括诸如未文档化函数（undocumented function）的函数，未文档化函数当文档化函数运行时由文档化函数使用。

[0254] 用于验证结果的示例性技术

[0255] 示例性实施例可以将可调整块和 LFT 模型当作普通 LTI 模型的对等者。这样做可以允许示例性实施例使用诸如 STEP 或 BODE 的标准命令来直接地分析调整后的控制系统。例如，用户可以键入命令，诸如：

[0256] >> bode(T)% 绘制 T 的波德 (Bode) 响应

[0257] 建模环境 220 可以处理该命令，并且可以经由显示装置向用户显示一个或多个波德图。

[0258] 示例性实施例可以进一步允许用户使用标准 tf()、ss()、zpk()、frd()、pid() 命令将可调整元素转换为普通 LTI 模型。例如，如果 F 是构成 T 的可调整块之一，则用户可以使用命令将 F 转换为传递函数，该命令例如是：

[0259] >> Ftuned=tf(T.Blocks.F)% 获得 F 的调整值作为传递函数

[0260] 如上所述的命令和技术向用户提供当用户与传统工具交互时不可获得的健壮调整能力。这些技术 / 命令进一步提供健壮能力，而不使得文本或图形建模环境对于用户来说过度复杂或困惑。

[0261] 用于实施本发明的实施例的示例性工作流

[0262] 图 17 示出了用于 F14 喷气战斗机的模拟自动驾驶仪的示例性模型 1700。本发明的示例性实施例和技术可以被应用到模型 1700 以调整用于 F14 的模拟自动驾驶仪系统。例如，实施例和技术可以使用 Simulink 建模环境来调整用于 F14 的纵轴的自动驾驶仪。模型 1700 可以包括可以实现标准级联回路自动驾驶仪的控制器块 1705。

[0263] 参见图 17，模型 1700 可以包括：内部回路 1710，其命令俯仰率 q；以及，外部回路 1715，其命令攻击角 alpha。

[0264] 图 18 更详细地示出了控制器块 1705。如图 18 中所示，窗口 1705-A 包括在图 17 中的控制器块 1705 中包括的组件和连接。在图 18 中，自动驾驶仪具有固定结构，并且由可能需要调整的增益和预滤波器块构成。例如，可调整增益块 1820、1825 和 1830 被示出具有三角形的形状和阴影，其中，阴影可以向用户指示增益块是可调整的。预滤波器块 1805、1810 和 1815 可以在形状上是矩形的，并且可以被加阴影，其中，该阴影可以指示预滤波器块是可调整的。图 18 可以进一步包括可以调整的加阴影的 PI 补偿器 1840。本发明的实施例可以使用第一颜色和 / 或阴影密度来指示可调整的增益块，并且使用第二颜色和 / 或阴影密度来指示可调整的预滤波器块和 / 或 PI 补偿器块。

[0265] 示例性实施例和技术允许用户自动地和联合地调整所选择的控制元素，以获得期望水平的性能和健壮度。

[0266] 参见图 19，用户可以输入命令 1905，诸如：

[0267] s = tf(' s ');

[0268] pKa = realp(' ka',0);

[0269] pKq = realp(' Kq',0);

[0270] pKf = realp(' Kf',0);

[0271] pKi = reale(' Ki',1);

[0272] pTal = realp(' Tal',1);

[0273] pW1 = realp(' W 1',0);

[0274] pW2 = realp(' W2',0);

[0275] 以定义图 17 和 18 的自动驾驶仪的参数。在图 19 中，用户可以选择将所有的参数定义为实数标量参数。在命令 1905 中，用户可以决定将大多数参数初始化为 0。

[0276] 用户可以与建模环境 220 交互以将积分器、alpha 和俯仰预滤波器创建为涉及命令 1905 的参数的表达式。例如，用户可以输入命令 1910 以创建积分器、alpha 和俯仰预滤波器，诸如：

[0277] Integrator=pKi/S; % Ki/s

[0278] AlphaSensor=tf(1,[pTal1]); %1/(Tal*s+1)

[0279] PitchFilter=tf([1pW1],[1pW2]);%(s+W1)/(s+W2)

[0280] 用于跟踪目标的示例性技术

[0281] 用户在与图 17 和 18 的自动驾驶仪模型交互时可能希望将跟踪目标使用公式表示为 H ∞约束。期望的带宽可以是 1rad/s（弧度 / 秒），并且用户可能期望使得攻击角 alpha 以较小的过冲追踪操纵杆输入（飞行员要求）。示例性实施例和技术可以用于使用公式表示在用于外部回路（alpha）1715（图 17）的开环形状上的用户要求。

[0282] 用户可以通过输入命令 1915 来定义他相信什么是真实的目标回路形状，该命令例如是：

[0283] wc=1;% 目标交叉

[0284] LS=(1+0.01*s/wc)/(0.01+s/wc);

[0285] bodemag(LS,{1e-3,1e3}),grid,title('Target loop sgape')

[0286] 建模环境 220 可以处理命令 1915，并且可以经由 GUI2000 来显示如图 20 中所示的波德图。GUI2000 显示回路形状，该回路形状确保在 1rad/s（0dB 交叉）的数量级上的带宽和可能不超过 1% 的稳态误差（即，在 w=0 处的 40dB 增益）。用户可以认识到他不能实际地在 w ＝ 0 处预期更大的增益，因为外部回路不包括积分器（即，积分器在图 17 的俯仰率回路中）。

[0287] 可能需要得出向 alpha 的闭环模型映射（r,n,d）以使得可以使用以上描述的示例性回路成形技术，其中，r 是 alpha 命令，n 是在 alpha 上的测量噪声，并且 d 是在致动器处进入的干扰。在一个示例性实施例中，用户可以通过下述方式来得出闭环模型：（1）列出在模型中的调整块，（2）将信号 r、n、d、alpha 标记为线性化输入 / 输出（I/O），并且（3）使用 LINLFT 命令来提取从（r,n,d）向 y ＝ alpha 的闭环传送的 LFT 模型。例如，用户可以创建图 21 中所示的代码。例如，用户可以通过向界面 2100 内输入文本来创建代码部分 2105 和 2110，该文本例如是：

[0288] % 调整的块

[0289] TunedBlocks={…

[0290] 'f14/Controller/Alpha-sensor Low-pass Filter', …

[0291] 'f14/Controller/Pitch Rate Lead Filter',…

[0292] 'f14/Controller/Ka',…

[0293] 'f14/Controller/Kq',…

[0294] 'f14/Controller/Kf',…

[0295] 'f14/Controller/Integrator'};

[0296] %图 4 的结构化补偿器

[0297] C=blkdiag(AlphaSensor,PitchFilter,pKa,pKq,pKf,Integrator);

[0298] linios=[…

[0299] linio('f14/Controller / Stick Prefilter',1,'in');…% r

[0300] linio('f14/Aircraft Dynamics Model',4,'outin');…% alpha,n

[0301] linio('f14/Controller',1,'in')]; % d

[0302] H=linlft('f14'' linios,TunedBlocks);

[0303] %闭环传递 [r ;n ;d]- ＞ y=alpha

[0304] Ta0=lft(H,C);

[0305] 在此，用户可能通过输入下述内容来要求可以添加作为 T10 的输出的误差信号 e=r-y：

[0306] T10=[0 0 0;1 0 0]+[1;-1]*Ta0;

[0307] 用户可以通过添加如上所述的回路成形权重来完成作为 H ∞约束的公式化。例如，用户可以向界面 220（图 22）内输入命令 2205，诸如：

[0308] beta=3：

[0309] Win=blkdidg(1,1 / LS,beta);

[0310] Wout=blkdiag(1,LS)；

[0311] T10=Wout*T10*Win;

[0312] T10.InputName={'r','n','d'};

[0313] T10.OutputName={'y','e'};

[0314] 在一个替换实施例中，可以通过建模环境 2200 来以编程方式添加回路成形权重，而不要求用户输入。

[0315] 当已经向建模环境 220 内输入与图 21 和图 22 相关联的输入时，可以通过下面的 H ∞约束来捕获图 20 的回路成形目标：

[0316] $||T_1(s)||_\infty < 1$。

[0317] 用于识别健壮性目标的示例性技术

[0318] 图 17 的级联回路结构是具有两个测量 (alpha 和 q) 和一个控制 (升降舵偏转 delta(德尔塔)) 的 MIMO 控制结构。在一些情况下，用户可能尝试估计每一个 SISO 回路的稳定裕度；然而，这样的手段不总是足以保证足够的 MIMO 裕度。由本发明的实施例支持的更可靠和精确的手段确保灵敏度函数

[0319] $S=(I+GK)^{-1}$ (等式 16)

[0320] 的峰值增益保持接近 1。在等式 16 中，G 将 delta 映射到 (alpha，q)，并且 K 是用于从 (alpha，q) 生成 delta 的集合控制器。换句话说，第二要求是

[0321] $||T_2(s)||_\infty < 1$，其中，$T_2=S$ (等式 18)

[0322] 可以通过观察在被控对象的输出 (alpha，q) 处测量到灵敏度函数 S 来得出 T_2 的闭环模型，并且 T_2 的闭环模型使用如上所述的 LINLFT。例如，用户可以输入命令 2210(图 22)，诸如：

[0323] linios=[…

[0324] linio('f14 / Aircraft Dynamics Model',4,'inout');…% alpha

[0325] linio('f14 / Aircraft Dynamics Model',3,'inout')]; % q

[0326] H=linlft('f14',linios,TunedBlocks);

[0327] S0=lft(H,C);

[0328] T20=S0;

[0329] 示例性调整技术

[0330] 一旦得出 T_2 的闭环模型，则用户可能希望调整控制器参数。示例性实施例促进将用于跟踪和健壮性的 H ∞约束组合为单个约束。例如，当用户限定

[0331] T0=blkdiag(T10,T20)

[0332] 时，可以表示单个约束。

[0333] 建模环境 220 可以指示 HINFSTRUCT 求解器使用优化器 360 运行 5 个优化。例如，优化器 360 可以从五个不同的起点运行该 5 个优化，以减轻在局部最小值处的提前终止的风险。优化器 360 可以进一步将闭环极点的量级限制为 50，以防止高增益设计。用户可以配置优化器 360 来通过输入诸如下述的命令来执行上面的优化：

[0334] Options=hinfstructOptions('Display','final','RandomStart',4,'SpecRadius',50);

[0335] 用户可以通过输入诸如下述的命令来运行指定的优化：

[0336] T=hinfstruct(T0,Options);

[0337] 建模环境 220 可以当命令被执行时返回调整后的目标 T。

[0338] 用于验证解的示例性技术

[0339] 用户可能希望验证控制器设计，并且示例性实施例允许用户交互地执行验证。在一个实施例中，可以通过向闭环传递 Ta0（用于 alpha 回路）和 S（灵敏度函数）推入调整后的参数值来验证该设计。用户可以输入两个命令，诸如：

[0340] Ta=replaceBlock(Ta0,T.Blocks);

[0341] S=replaceBlock(S0,T. Blocks);

[0342] 建模环境 220 可以处理命令，并且可以模拟 alpha 回路的闭环响应。

[0343] 图 23 示出了示例绘图 2300，其示出了用于模型 1700 的 alpha 回路的闭环响应。例如，用户可以输入命令，诸如：

[0344] step(Ta(1,1))%对于 alpha 的 alpha 命令

[0345] 以使得经由显示装置 250 来显示绘图 2300。

[0346] 示例性实施例可以进一步允许用户验证 S 的峰值增益接近 0dB。例如，用户可以键入：

[0347] sigma(S)%灵敏度函数的峰值增益

[0348] 建模环境 220 可以处理该命令，并且可以经由显示装置 250 显示图 24 的绘图 2400。

[0349] 示例性实施例通过与建模环境 220 交互来进一步允许用户访问调整后的参数（例如，Kq）的值。例如，用户可以键入：

[0350] T.Blocks.Kq.Value% Kq 增益的调整后值

[0351] 并且，建模环境 220 可以当命令被处理时产生回答。例如，可以向用户显示下述部分：

[0352] ans=

[0353] 4.3486e-001

[0354] 示例性处理

[0355] 图 25 示出了用于实施本发明的一个或多个实施例的示例性处理。用户可以创建包括一个或多个可调整组件的模型（行为 2505）。例如，用户可以与图形模型交互，并且可以建立自由图形模型，该自由图形模型包括通过表示信号的线连接的多个组件。该模型可以包括一个或多个反馈回路，并且可以包括可以被表示为被控对象的固件组件和可以经由控制器表示的一个或多个可调整组件。用户可以进一步使用其中用户输入命令以创建固定和可调整模型组件的文本环境来创建模型。在一个实施例中，行为 2505 可以是可选的，诸如当本发明的实施例接收到现有模型（例如，从永久存储介质检索到的模型）时。

[0356] 当识别了模型时，可以接收与模型相关联的设计要求（行为 2510）。例如，用户可以指定用于模型的开环响应的期望的增益配置，可以指定模型中的可以被组合为结构化控制器的可调整块，并且 / 或者可以识别模型中的用于对模型的闭环表示的输入、输出和干扰的位置。用户可以替换地与可以用于指定对于模型的设计要求的在文本环境中的命令交互。在其他实施例中，设计要求可以从数据存储器被读取，经由 API 以编程方式被接收等。

[0357] 可以利用本发明的示例性实施例将设计要求使用公式表示为 H∞约束（行为 2515）。例如，本发明的一个实施例可以将经由用户输入装置接收到的设计要求（例如，回路形状目标）自动地变换为在调整模型的组件中使用的 H∞约束。计算机 205 可以使用 H∞合成技术来执行处理操作，并且可以提取被控对象、控制器和传递函数（行为 2520）。在一个

实施例中，控制器可以包括以块对角结构排列的可调整组件的可调整参数。例如，将设计要求变换为 H ∞约束可以包括将控制架构映射为如图 7 中所示的典型结构。

[0358] 在变换后，被控对象、控制器和传递函数可以具有与可以用于调整模型的参数的优化器兼容的格式。该兼容格式可以被输入到优化器，并且优化器可以调整与结构化的控制器相关联的组件(行为 2525)。在一个实施例中，优化器可以接收未调整的组件。

[0359] 优化器可以产生包括调整后的组件的结果(行为 2530)。例如，调整后的组件可以满足在行为 2510 中接收到的设计要求，并且同时满足用于模型的稳定性目标。

[0360] 可以验证由优化器产生的结果以确保该结果允许模型以期望的方式运行(行为 2535)。例如，验证结果可以确保不超过过冲裕度，实现期望的响应速率，保持稳定裕度，实现期望的回路增益，实现期望的干扰抑制等。

[0361] 示例性架构

[0362] 图 26 示出了可以用于实现图 2 的计算机 205 的示例性计算机架构。图 26 是与计算机 205 相对应的实体的示例性图。如所示，实体可以包括总线 2610、处理逻辑 2620、主存储器 2630、只读存储器(ROM)2640、存储装置 2650、输入装置 2660、输出装置 2670 和 / 或通信接口 2680。总线 2610 可以包括允许在实体的组件之间的通信的路径。

[0363] 处理逻辑 2620 可以包括可以解释和执行指令的处理器、微处理器或其他类型的处理逻辑(例如，现场可编程门阵列(FPGA)、图形处理单元(GPU)、数字信号处理器(DSP)、专用集成电路(ASIC)、专用集成处理器(ASIP)、可编程逻辑器件(PLD)等)。对于一种实现方式，处理逻辑 2620 可以包括单核处理器或多核处理器。在另一种实现方式中，处理逻辑 2620 可以包括单个处理装置或一组处理装置，诸如处理集群或计算网格。在另一种实现方式中，处理逻辑 2620 可以包括多个处理器，该多个处理器可以相对于彼此是本地或远程的，并且可以在处理时使用一个或多个线程。

[0364] 主存储器 2630 可以包括可以存储用于由处理逻辑 2620 执行的信息和指令的随机存取存储器(RAM)或另一种类型的动态存储装置。ROM2640 可以包括可以存储由处理逻辑 2620 使用的静态信息和 / 或指令的 ROM 装置或另一种类型的静态存储装置。存储装置 2650 可以包括磁性、固态和 / 或光学记录介质及其对应的驱动器或者可以存储由处理逻辑 2620 使用的静态信息和 / 或指令的另一种类型的静态存储装置。

[0365] 输入装置 2660 可以包括逻辑，该逻辑允许操作员向诸如键盘、鼠标、笔、触摸板、加速计、麦克风、语音识别、照相机、生物计量机构等的实体输入信息。在一个实施例中，输入装置 2660 可以对应于输入装置 240。

[0366] 输出装置 2670 可以包括向操作员输出信息的机构，包括显示器、打印机、扬声器、触觉接口等。通信接口 2680 可以包括任何收发器类的逻辑，该逻辑使得实体能够与其他装置和 / 或系统进行通信。例如，通信接口 2680 可以包括用于经由网络与另一个装置或系统进行通信的机构。

[0367] 在图 26 中描述的实体可以响应于处理逻辑 2620 执行在诸如主存储器 2630 的计算机可读存储介质中存储的软件指令来执行特定操作。计算机可读存储介质可以被定义为物理(例如，有形)或逻辑存储装置。可以从诸如存储装置 2650 的另一计算机可读存储介质或经由通信接口 2680 从另一装置向主存储器 2630 内读取软件指令。当在处理逻辑上执行软件指令时，在主存储器 2630 中包含的软件指令可以使得处理逻辑 2620 执行本文所述的

技术。可替换地，硬连线的电路可以取代软件指令或与软件指令组合使用以实现本文所述的技术。因此，本文所述的实现方式不限于硬件电路和软件的任何特定组合。

[0368] 虽然图 26 示出实体的示例性组件，但是在其他实现方式中，实体可以包含比在图 26 中所述更少、不同或增加的组件。在其他实现方式中，实体的一个或多个组件可以执行被描述为由实体的一个或多个其他组件执行的一个或多个任务。

[0369] 示例性分布式实施例

[0370] 分布式实施例可以使用两个或更多的处理资源来执行处理。例如，实施例可以使用在单个处理装置中的两个或更多的核来执行处理，在单个外壳内安装的多个处理装置上分布处理，并且 / 或者在通过网络连接的多种类型的处理逻辑上分布处理。

[0371] 图 27 示出示例性系统，该示例性系统可以支持使用分布式计算环境为客户机装置（例如，计算机 205）交互地设计用于非线性模型的控制器。系统 2700 可以包括计算机 205、网络 2730、服务提供商 2740、远程数据库 2750 和集群 2760。图 27 的实现方式是示例性的，并且本发明的其他分布式实现方式可以包括更多装置和 / 或实体、更少装置和 / 或实体和 / 或在配置上与图 27 的示例性配置不同的装置 / 实体。

[0372] 计算机 205 可以包括图形用户界面（GUI）2710 和建模环境 220。GUI2710 可以包括允许用户与计算机 205 和 / 或远程装置（例如，服务提供商 2740）交互的界面。在一个示例性实施例中，GUI2710 可以类似于图 4-6 和 15-18 的界面。

[0373] 网络 2730 可以包括能够传送数据（例如，分组数据或非分组数据）的任何网络。网络 2730 的实现方式可以包括局域网（LAN）、城域网（MAN）和 / 或诸如因特网的广域网（WAN），它们可以使用实质上任何网络协议来运行，网络协议诸如因特网协议（IP）、异步传输模式（ATM）、同步光学网络（SONET）、用户数据报协议（UDP）、IEEE802.10 等。

[0374] 网络 2730 可以包括网络装置，诸如路由器、交换机、防火墙和 / 或服务器（未示出）。网络 2730 可以是使用导线和 / 或光纤的硬连线网络，并且 / 或者可以是使用自由空间光学、射频（RF）和 / 或声音传输路径的无线网络。在一种实现方式中，网络 2730 可以是实质上开放的公共网络，诸如因特网。在另一种实现方式中，网络 2730 可以是更受限的网络，诸如公司虚拟网。在本文所述的网络上运行的网络和 / 或装置的实现方式不限于任何特定的数据类型、协议、架构 / 配置等。例如，在一个实施例中，网络 2730 可以是使用量子兼容的联网协议的量子网络。

[0375] 服务提供商 2740 可以包括使得另一个装置可获得服务的装置。例如，服务提供商 2740 可以包括实体，该实体使用服务器和 / 或其他装置向目的地提供一个或多个服务。服务可以包括由目的地执行的指令来执行操作。可替换地，服务可以包括代表目的地执行的指令以代表目的地执行操作。

[0376] 为了举例，假定服务提供商运行 web （网络）服务器，该 web 服务器向诸如计算机 205 的目的地提供一个或多个基于 web 的服务。基于 web 的服务可以允许计算机 205 使用由服务提供商运行的硬件来执行电子和 / 或机械系统的分布式模拟。例如，计算机 205 的用户可以被允许来使用服务提供商的硬件来交互地设计用于系统模型的 PID 控制器。在一种实现方式中，客户（用户）可以在预订的基础上接收服务。预订可以包括以下配置，诸如按月预订、每次使用收费、基于在服务提供商 2740 和客户之间交换的信息量的收费、基于由客户使用的处理器周期的数量的收费、基于由客户使用的处理器的数量的收费等。

[0377] 远程数据库2750可以包括装置，该装置存储由诸如计算机205的其他装置使用的机器可读信息。在一个实施例中，远程数据库2750可以包括存储包含关于系统模型、控制器等的信息的数据结构的存储装置(例如，硬盘、光盘、固态存储装置等)的阵列或网格。

[0378] 集群2760可以包括一组处理装置，诸如可以用于执行远程处理(例如，分布式处理、并行处理等)的执行单元2770。执行单元2770可以包括硬件和/或基于硬件/软件的装置，其为诸如计算机205的请求装置执行处理操作。在一个实施例中，执行单元2770可以各自计算部分结果，并且该部分结果可以被组合为用于模型的整体结果。

[0379] 示例性实现方式

[0380] 一个实施例可以包括计算机实现的方法或在永久计算机可读介质上驻留的机器可执行指令。该方法和/或指令可以被实现以调整在任意控制结构中的设计参数，其中，可以使用例如本地机器的单机或在分布式环境中(例如，使用计算装置的集群或网格)来实施计算机实现的方法。

[0381] 例如，当被实施为计算机实现的方法时，该方法可以包括：识别一个或多个可调整组件，该一个或多个可调整组件具有被调整的一个或多个自由参数。可调整组件可以是任意控制结构的文本或图形模型的一部分。该方法可以进一步包括：识别包括一个或多个可调整组件的模型的部分。

[0382] 该方法可以进一步包括：将该任意控制结构变换为标准形式，该标准形式包括集中线性模型，该集中线性模型包括在任意控制结构中不调整的组件和使用该可调整组件形成的集合。从可调整组件形成的集合可以被分组或布置为块对角方式。在该方法中，由于下述原因可以访问在块对角布置中的组件：在块对角布置中的单独组件或参数能够被用户看到、可用于算法、能够被读取或写入等。当必要时，该方法可以支持使得集合的一个或多个构件位于块对角布置之外。

[0383] 该方法可以使用H∞或H2目标或约束来表达设计目标和/或设计要求。在该方法的一种实现方式中，H∞/H2目标或约束与一个或多个点对点传递函数相关。例如，目标和/或约束可以与在开环或闭环系统中的点对点传递函数相关。该方法可以进一步参数化可调整组件。例如，该方法可以静态地参数化可调整组件或动态地参数化可调整组件。一旦参数化了可调整组件，该方法可以基于该参数化来与可调整组件的自由参数交互。

[0384] 该方法可以使用调整器来调整反馈控制结构。例如，在一种实现方式中，该方法可以使用在非平滑H∞/H2优化算法的类中的调整器来调整反馈控制结构。该调整器可以作用于标准形式，并且当可调整组件具有块对角形式时可以作用于可调整参数。该调整器可以进一步调整参数以最小化或最大化H∞/H2目标和/或执行H∞/H2约束。

[0385] 可以在诸如MATLAB兼容环境的技术计算环境中实现计算机实现的方法。还可以在诸如Simulink兼容的环境或LabView兼容的环境的文本或图形环境中实现该方法。实现方式可以进一步与模块集交互，该模块集可以例如包括可调整组件、固定组件、连接、算法等。

[0386] 还可以以支持任意反馈控制结构的交互设计和调整的方式来实现计算机实现的方法。例如，可以实现支持用户的实时交互设计和调整行为的方法。可以使用FPGA、ASIC、ASIP、PLD、GPU、DSP、多核装置、分布式计算资源等来实现该方法的实现方式。

[0387] 该方法的实现方式可以进一步允许用户或装置指定在多个输入/输出点(例如，

点对点传递函数)上的独立目标或约束,以简化多目标或多要求设计任务。该方法的实现方式可以通过支持由用户或用户组使用的传统工作流来如此进行。该方法可以被扩大以包括将设计要求应用至点对点传递函数,其中,该应用促进多目标和多要求设计任务的公式化。可以进一步在面向对象的框架中部署该方法。

[0388] 本发明的实施例可以进一步包括在永久计算机可读存储介质上驻留的计算机实现的方法行为和/或可执行指令。例如,当被实现为介质/可执行指令时,该介质可以存储指令,该指令当被执行时实施本发明的方面。例如,在一种实现方式中,执行的指令可以为用户执行方法或技术。在一种实现方式中,该介质可以存储可执行指令,该可执行指令当在处理器上被执行时实现 API,该 API 用于静态地指定可调整组件、动态地指定可调整组件和/或与可调整组件的参数交互。在一种实现方式中,API 可以是基于对象的。

[0389] 可执行指令当被执行时可以识别预定义界面,该预定义界面包含用于预定义的一组组件的参数化。该指令可以进一步实现一组算术运算和/或实现一组帮助函数。所执行的指令可以使用算术运算和帮助函数来进一步动态地创建可调整组件。在一种实现方式中,动态地创建可以包括组合基本参数组件和固定系数或固定组件。在必要时实施例可以进一步支持静态配置组件。

[0390] 所执行的指令可以进一步产生可调整组件的参数模型。在一种实现方式中,该参数模型可以解释可调整组件的可调整参数,并且可以允许代表用户接收到的输入与可调整参数交互。例如,用户输入可以初始化可调整参数,固定可调整参数,或释放可调整参数的所选择的一些。

[0391] 本发明的另一个实施例可以被实现为计算机实现的方法或经由驻留在一个或多个永久计算机可读存储介质上的可执行指令被实现。该实施例可以对用于建立任意反馈控制结构的标准形式或用于指定在设计要求的 H∞或 H2 公式化中使用的点对点传递函数的应用程序接口(API)进行实现、部署、运行等。当必要时,API 的实施例可以是基于对象的。

[0392] 实施例可以与用户输入机构交互,该用户输入机构使用输入句法来指定算术运算符和/或框图操作。例如,由于如下原因,输入句法可以用户熟悉的句法:用户已经使用该句法来与诸如 PID 控制器的其他类型的控制结构交互。输入句法可以允许用户输入线性时间不变模型组件,其中,该线性时间不变模型组件在线性时间不变模型中被使用并且被用作基于软件的输入界面,其中,基于软件的界面描述了具有可调整参数的可调整组件。

[0393] 实施例可以使用算术运算符的一个或多个和框图操作将线性时间不变模型组件与基于软件的界面进行组合。框图操作的示例可以包括但是不限于串联连接、并联连接或反馈连接。实施例可以与用户输入机构交互,并且该交互可以允许输入句法递增地构造整体控制系统的标准形式。该标准形式可以包括线性时间不变模型组件和可调整组件。输入句法可以进一步用于促进基于标准形式来生成参数模型。该参数模型可以具有与优化器兼容的形式,其中,该兼容形式允许向优化器输入参数模型。可以以下述方式来建立参数模型:以允许满足设计要求的方式来允许调整参数模型。

[0394] 该实施例可以进一步被配置来支持:使用优化器来优化参数模型,该优化器与标准形式和可调整参数交互以最小化或最大化 H∞/H2 目标,并且/或者执行 H∞/H2 约束。该实施例可以进一步被配置来向用户输入机构提供帮助函数。该实施例可以当用户使用输入句法与 API 交互时允许用户经由输入机构访问帮助函数。而且,实施例可以被配置来提

供与输入机构兼容并且可以与输入机构一起运行的函数。该函数可以用于查询控制系统，并且 / 或者分析控制系统。

[0395] 本发明的实施例可以包括计算机实现的方法和 / 或计算机可执行指令，该计算机可执行指令可以驻留在永久存储介质上，以便以编程方式以公式将回路成形要求表示为 H ∞或 H2 公式。该实施例当被实施为方法时可以使用单独或分布式装置被实现，并且可以被配置来代表一个或多个用户来支持交互（例如，实时）操作。该方法可以包括 ：接收用于开环增益的目标形状或用于目标形状的代理。例如，可以代表用户从输入机构（例如，键盘、GUI 等）接收目标回路形状或代理，或者可以例如从计算机可读存储介质以编程方式检索目标回路形状或代理。

[0396] 该方法可以包括 ：与软件工具交互，该软件工具从目标回路形状或代理得出控制结构和滤波器。软件工具可以从得出的控制结构和滤波器进一步构造标准形式，并且从得出的控制结构和滤波器构造 H ∞ /H2 约束。在该方法中，标准形式和 H ∞ /H2 约束可以捕获用于控制系统的设计要求。

[0397] 另一个实施例可以包括计算机实现的方法和 / 或在计算机可读介质上驻留的用于利用标准形式的结构的可执行指令。利用用于标准形式的结构可以增强可以代表用户支持交互工作流的调整器算法的性能。该实施例可以接收由优化器在优化过程期间供应的可调整参数的值。优化过程可以包括用于执行优化行为的机器实现的技术。

[0398] 该实施例可以进一步使用构造过程 / 技术来通过下述方式构造用于可调整参数值的标准形式的状态空间模型 ：使用软件对象来实现与可调整参数相关联的可调整组件，使得每一个软件对象负责针对所接收的参数值提供其状态空间表示，聚合状态空间矩阵以产生用于可调整组件的聚合的状态空间矩阵，并且 / 或者将可调整组件的聚合的状态空间矩阵与集中被控对象模型的状态空间表示进行组合以获得标准形式的期望的闭环状态空间模型。

[0399] 该实施例可以高速缓存构造过程的中间结果以加速随后的梯度计算。可以通过下述方式来计算梯度信息 ：应用诸如链规则的规则以区分目标和约束 ；使用用于构造标准形式的状态空间模型的相同软件对象 ；使得每一个软件对象负责针对所接收到的参数提供标量值函数的梯度，其中，该标量值函数是应用规则的副产品 ；并且 / 或者，使用高速缓存的中间结果来将由每一个可调整组件供应的梯度数据高效地聚合成目标和约束的整体梯度。

[0400] 该实施例可以使得梯度信息可获得，而不要求代表处理器的另外的计算。例如，与在未使得梯度信息可获得的情况下引发的计算成本相比较，可以在没有另外的计算成本的情况下使得梯度信息可获得。该实施例可以在几乎没有计算成本的情况下进一步得出用于可调整参数的给定值的闭环系统，例如，由于如下原因该成本可忽略 ：在成本上的分解没有不利地影响与系统设计、建模时间等相关联的计算预算。

[0401] 被实现为计算机实现的方法的实施例可以调整在任意控制结构中的设计参数。在计算机中运行的方法可以获得一个或多个可调整组件。例如，该方法可以从相对于实现该方法的计算机本地或远程执行的模型获得要调整的组件。该方法可以通过将要调整的组件与控制结构的固定动力学分离来变换控制结构。在一个实施例中，固定动力学可以包括固定和 / 或已知组件。该方法可以使用与在系统中的一个或多个点对点传递函数相关的 H ∞或 H2 目标或约束。在一个实施例中，H ∞或 H2 目标或约束可以用于表达设计目标或设计

要求。该方法可以基于非平滑 H ∞或 H2 优化算法使用调整器进一步调整控制结构。例如，可以从一组非平滑 H ∞或 H2 优化算法选择调整器，其中，调整器调整参数以优化 H ∞或 H2 目标，或执行 H ∞或 H2 约束。

[0402] 该计算机实现的方法可以从诸如键盘、麦克风、触敏显示器等的输入机构接收指令。该指令可以指定一个或多个可调整组件和 / 或固定组件。该方法可以进一步识别一个或多个点对点传递函数，并且可以使用点对点传递函数来用于变换控制结构。在一种实现方式中，所述方法可以将控制结构变换为可以包括线性模型和可调整组件的标准形式。例如，线性模型可以包括不可调整组件。该标准形式可以进一步包括以块对角方式布置的可调整组件或参数的一些或全部。

[0403] 计算机实现的方法可以参数化可调整组件，并且基于该参数化来与可调整组件交互。用于该方法的可调整组件可以包括一个或多个自由参数，并且该方法可以与自由参数交互。该方法可以使用诸如反馈控制结构、前馈控制结构等的控制结构的各种配置来运行。该方法还可以使用可以被简化为诸如集中线性模型的一种线性模型的模型来运行。

[0404] 一个实施例可以包括用于存储可执行指令的一个或多个永久计算机可读介质，该可执行指令当在处理器上被执行时实现应用程序接口（API）。该 API 可以用于静态地指定可调整组件，动态地指定可调整组件和 / 或与可调整组件的参数交互。该介质可以存储一个或多个可执行指令以例如当在计算装置上执行指令时实施方法。该方法可以包括 ：识别预定义界面，其中，该预定义界面包含用于预定义的一组组件的参数化，该预定义的一组组件例如是经由模块集、库等提供的组件。该方法可以使用算术运算或帮助函数来动态地创建可调整组件。该方法可以进一步产生与可调整组件的可调整参数交互的可调整组件的参数模型。

[0405] 该方法可以进一步包括 ：实现一组算术运算或帮助函数。该算术运算或帮助函数可以用于在必要时动态地创建可调整组件。利用该方法，参数模型可以响应于与诸如代表用户运行的输入机构之类的输入机构相关联的指令来与可调整参数交互。在一个实施例中，经由输入机构接收到的指令可以与下述的一个或多个相关联 ：初始化可调整参数，固定可调整参数，并且释放可调整参数的所选择的一些。

[0406] 一个实施例可以包括用于存储可执行指令的一个或多个永久计算机可读存储介质，该可执行指令当在处理器上被执行时实现应用程序接口（API）。例如，当执行指令时，计算机可以执行用于实现 API 的方法。在一个实施例中，该方法可以包括 ：将线性时间不变模型组件与一个或多个基于软件的界面组合，该一个或多个基于软件的界面用于描述可以调整的组件，诸如在模型中的可调整组件。该方法可以构造包括线性时间不变模型组件和可调整组件的控制系统的标准形式，并且可以生成用于支持调整的参数模型。例如，该方法可以基于标准形式来生成参数模型，并且可以基于调整行为使用该参数模型来满足设计要求。

[0407] 该方法可以使用优化器来进一步优化参数模型。在一个实施例中，该优化器可以与标准形式和可调整参数交互，以最小化 / 最大化 H ∞或 H2 目标和 / 或执行 H ∞或 H2 约束。该方法可以进一步将线性时间不变模型组件与用于描述可调整组件的一个或多个基于软件的界面组合。在这个实施例中，组合行为可以进一步包括一个或多个算术运算符或框图操作的使用。在该实施例中，框图操作可以包括串联连接、并联连接或反馈连接。

[0408] 一个实施例可以包括一个或多个永久计算机可读介质，用于存储用于以编程方式来使用公式表示回路成形要求的可执行指令。该实施例可以当在计算装置上执行指令时实现方法。例如，该方法可以包括与软件工具交互，该软件工具得出控制结构，得出滤波器、从所得出的控制结构和滤波器构造标准形式，并且从所得出的控制结构和滤波器构造 H ∞或 H2 约束。在该方法中，标准形式和 H ∞约束或 H2 约束可以捕获控制系统的设计要求。

[0409] 一个实施例可以包括用于存储可执行指令的一个或多个永久计算机可读介质。当在计算装置上执行指令时，可以通过该装置来执行一种方法。该方法可以包括 ：使用构造过程来构造用于可调整参数值的标准形式的状态空间模型。该方法可以通过下述方式来构造状态空间模型 ：将具有可调整参数的可调整组件的聚合状态空间矩阵与被控对象模型的状态空间表示组合，以获得具有标准形式的期望状态空间模型。例如，该方法可以使用集中被控对象模型来促进获得具有标准形式的状态空间模型。该方法可以通过下述方式来计算梯度信息 ：区分目标或约束 ；通过软件对象来提供标量值函数的梯度 ；并且 / 或者，将由可调整组件供应的梯度聚合成目标和约束的整体梯度。

[0410] 结论

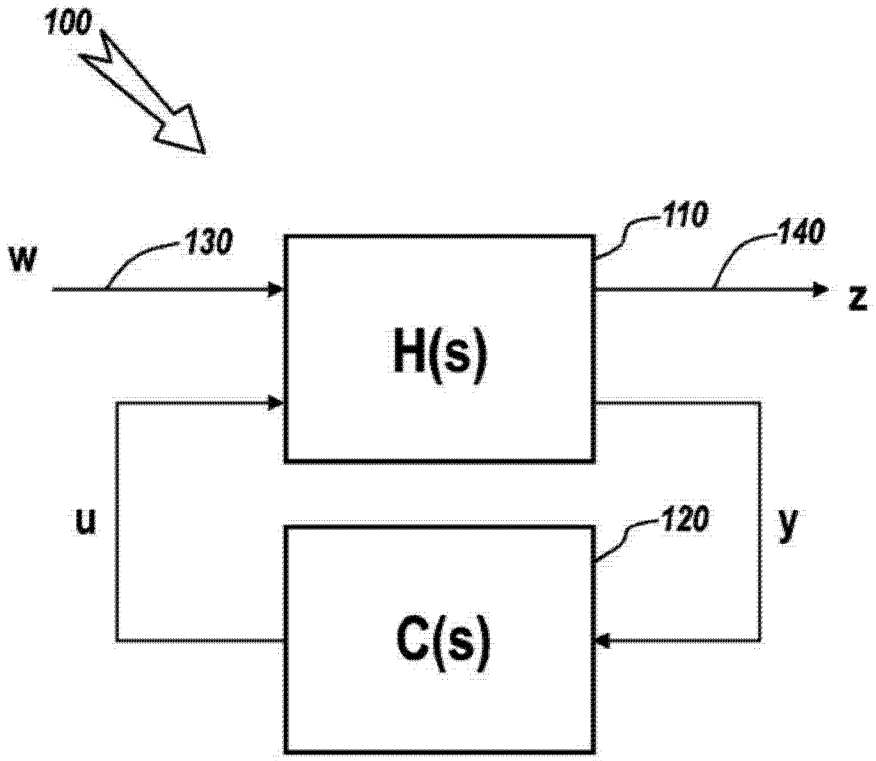
[0411] 实现方式可以允许用户使用用户熟悉的特性来交互地设计用于系统模型的控制器。

[0412] 本发明的示例性实施例的上述说明提供了例示和描述，但是不意欲是穷举的或将本发明限于所公开的精确形式。根据上面的教导可以进行修改和变化，或者，可以从本发明的实践获取修改和变化。例如，虽然已经参考图 25 描述了一系列行为，但是可以在与本发明的原理一致的其他实现方式中修改行为的顺序。而且，可以并行地执行不相互依赖的行为。

[0413] 另外，在不偏离本发明的精神的情况下，可以使用除了在附图中示出和在说明书中描述的装置和配置之外的装置和配置来实现与本发明的原理一致的实现方式。可以根据具体部署和 / 或应用来相对于图 2、3、26 或 27 的实现方式增加和 / 或去除装置和 / 或组件。而且，所公开的实现方式可以不限于硬件的任何特定组合。

[0414] 而且，本发明的特定部分可以被实现为执行一个或多个功能的“逻辑”。该逻辑可以包括诸如硬连线逻辑、专用集成电路、现场可编程门阵列、微处理器之类的硬件，或硬件和软件的组合。在本发明的说明书中使用的元素、行为或指令都不应当被解释为对于本发明的关键或必要的，除非这样明确地说明。而且，本文使用的冠词“一”意欲包括一个或多个项。当仅意欲表示一个项时，使用术语“一个”或类似语言。而且，本文使用的短语“基于”意欲表示“至少部分地基于”，除非另外明确地说明。

[0415] 本文使用的标题或小标题通过将说明书划分为子部分来帮助读者。这些标题和小标题不被解释为限制本发明的范围或限定本发明的特征。



（现有技术）

图 1

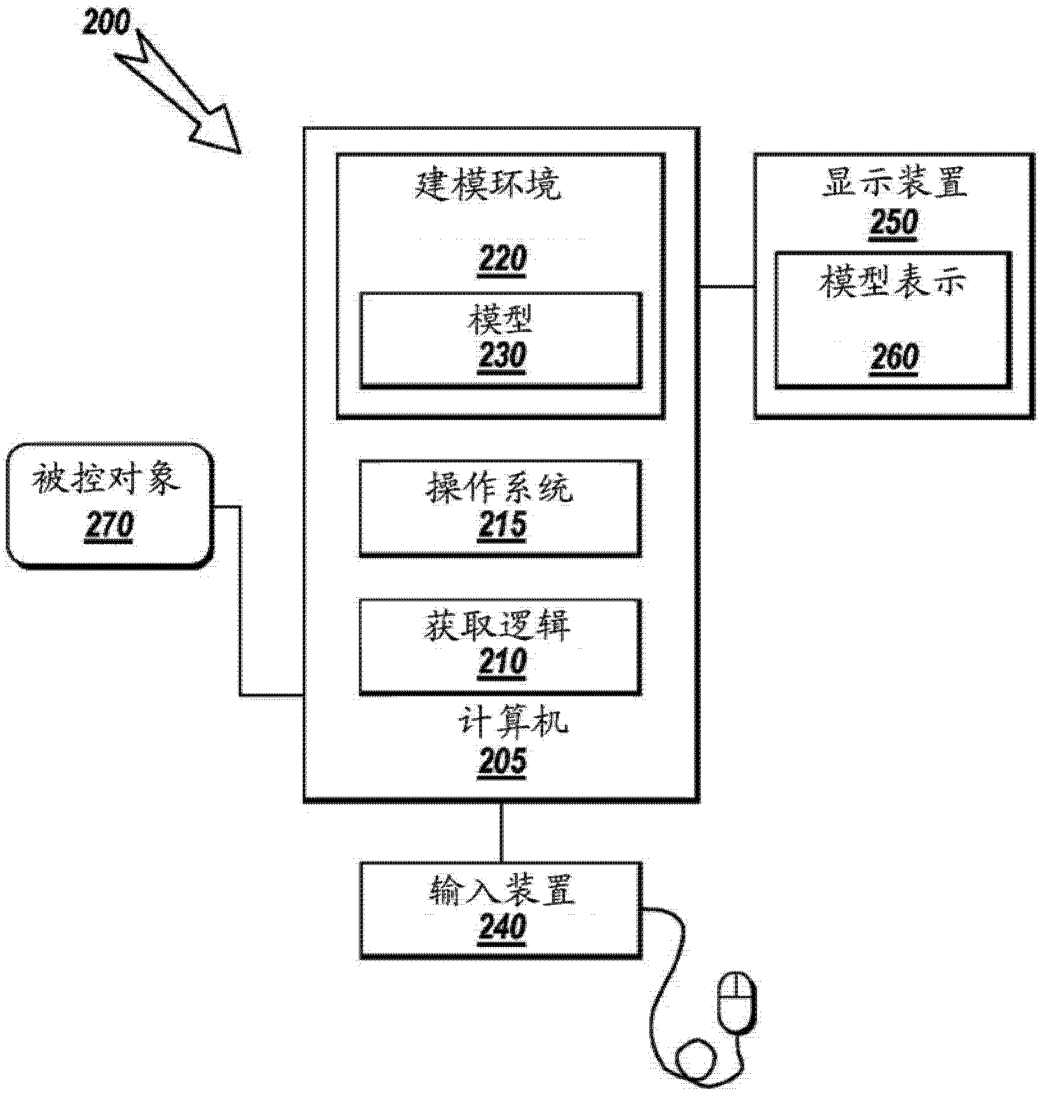

200
建模环境
220
模型
230
显示装置
250
模型表示
260
被控对象
270
操作系统
215
获取逻辑
210
计算机
205
输入装置
240

图 2

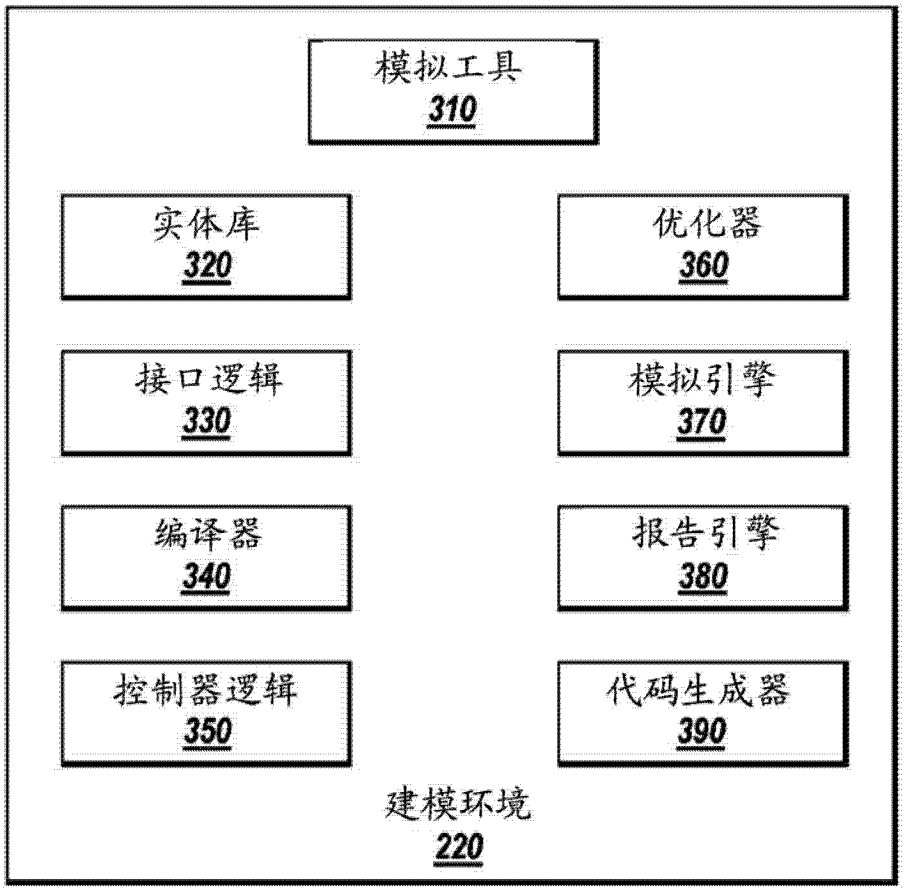

模拟工具
310
实体库
320
优化器
360
接口逻辑
330
模拟引擎
370
编译器
340
报告引擎
380
控制器逻辑
350
代码生成器
390
建模环境
220

图 3

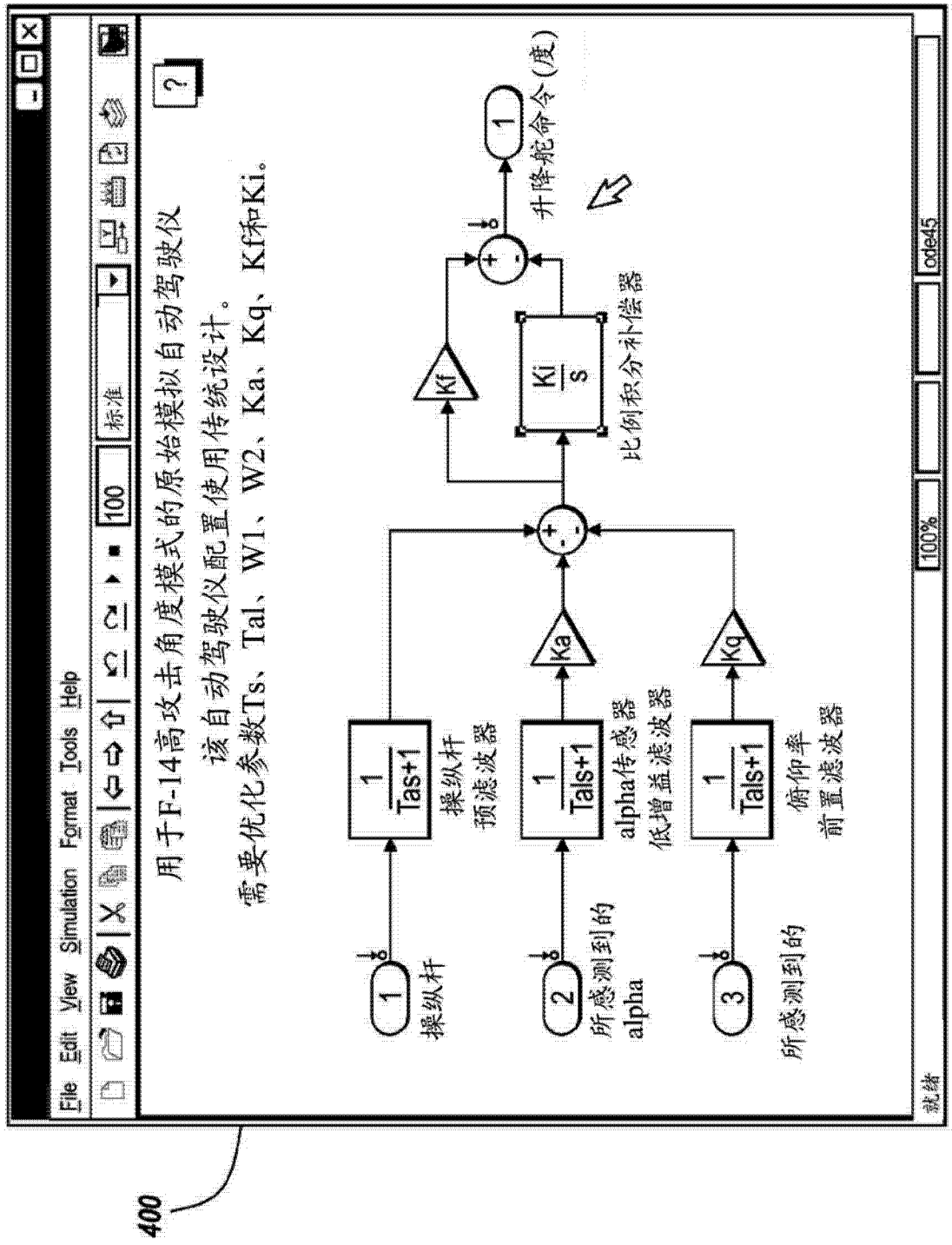

File Edit View Simulation Format Tools Help
100
标准
用于F-14高攻击角度模式的原始模拟自动驾驶仪
该自动驾驶仪配置使用传统设计。
需要优化参数Ts、Tal、W1、W2、Ka、Kq、Kf和Ki。
?
1
操纵杆
2
所感测到的
alpha
3
所感测到的
1/(Tas+1)
操纵杆
预滤波器
1/(Tals+1)
alpha传感器
低增益滤波器
1/(Tals+1)
俯仰率
前置滤波器
Ka
Kq
Kf
Ki/s
比例积分补偿器
1
升降舵命令(度)
400
就绪
100%
ode45

图 4

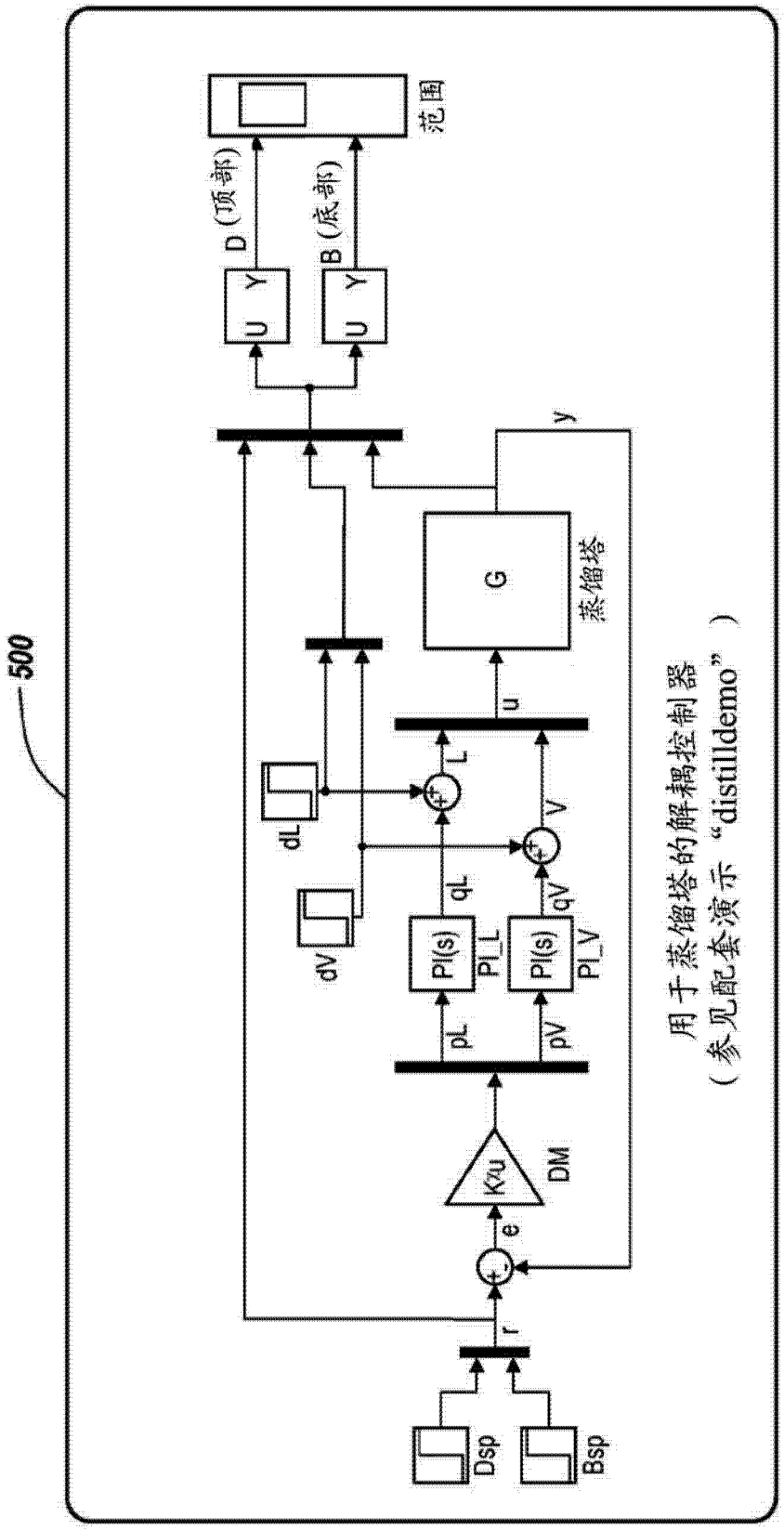

500
用于蒸馏塔的解耦控制器
（参见配套演示“distilldemo”）
范围
D (顶部)
B (底部)
U
Y
y
G
蒸馏塔
u
L
V
dL
dV
qL
qV
PI(s)
PI_L
PI_V
pL
pV
K*u
DM
e
r
Dsp
Bsp

图 5

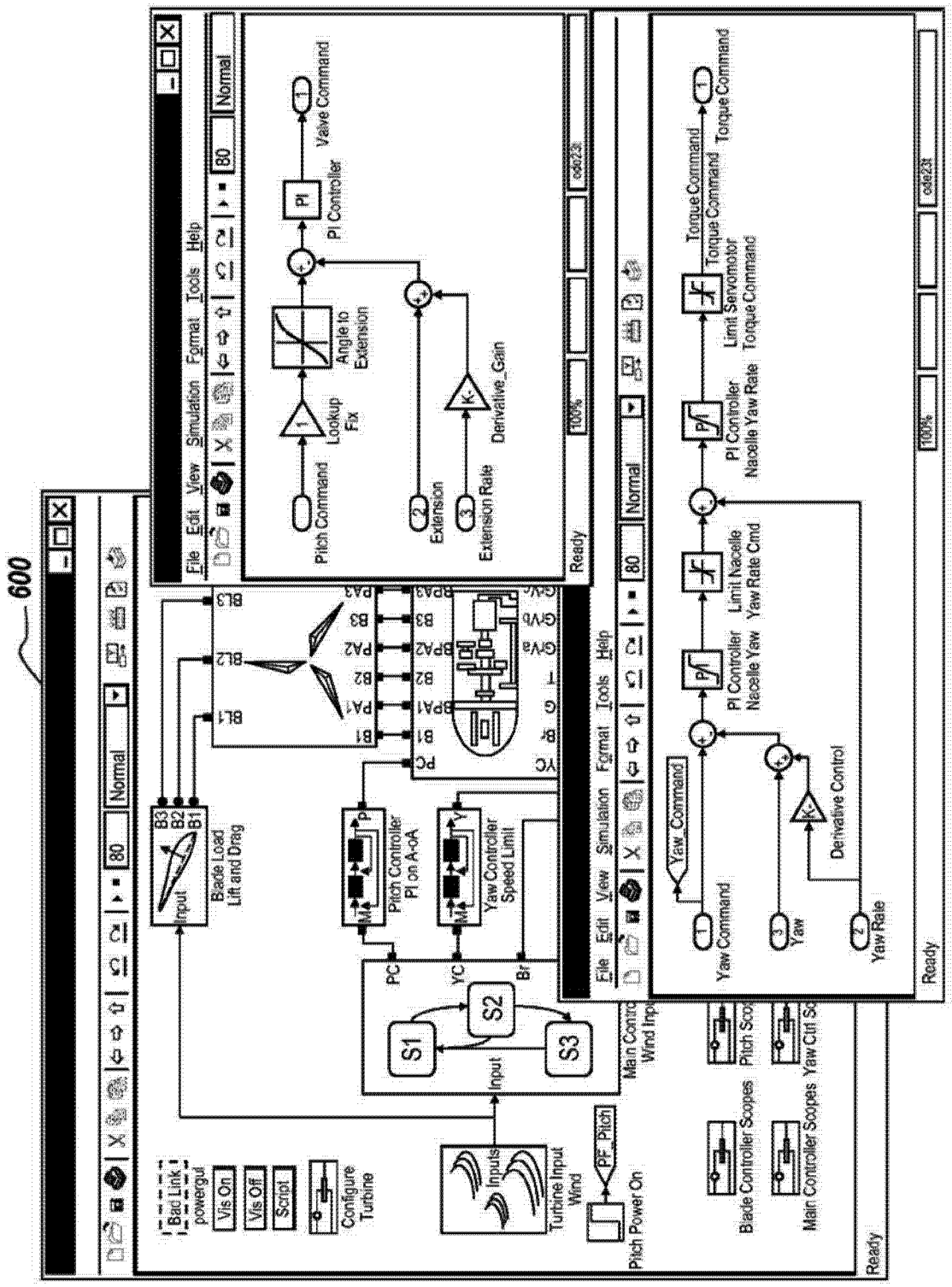

600
File Edit View Simulation Format Tools Help
Normal
Ready
Pitch Command
Lookup Fix
Angle to Extension
PI Controller
Valve Command
Extension
Extension Rate
Derivative_Gain
Blade Load Lift and Drag
Pitch Controller PI on A-oA
Yaw Controller Speed Limit
S1
S2
S3
Input
Inputs
Turbine Input Wind
Pitch Power On
Configure Turbine
Vis On
Vis Off
Script
Bad Link ! powergui
Blade Controller Scopes
Main Controller Scopes
Yaw_Command
Yaw Command
Yaw
Yaw Rate
Derivative Control
PI Controller Nacelle Yaw
Limit Nacelle Yaw Rate Cmd
PI Controller Nacelle Yaw Rate
Limit Servomotor Torque Command
Torque Command

图 6

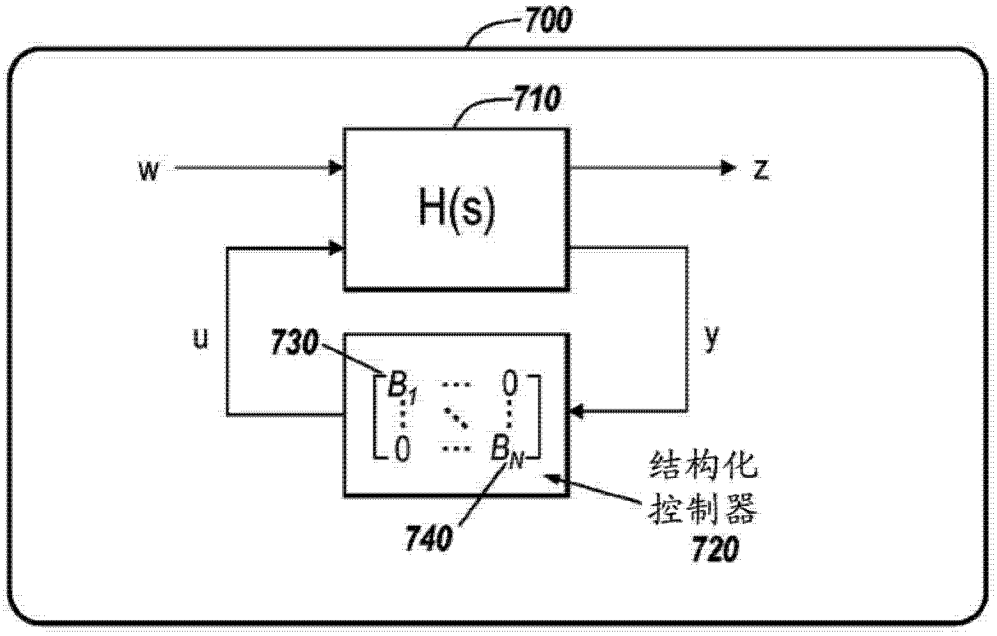


图 7

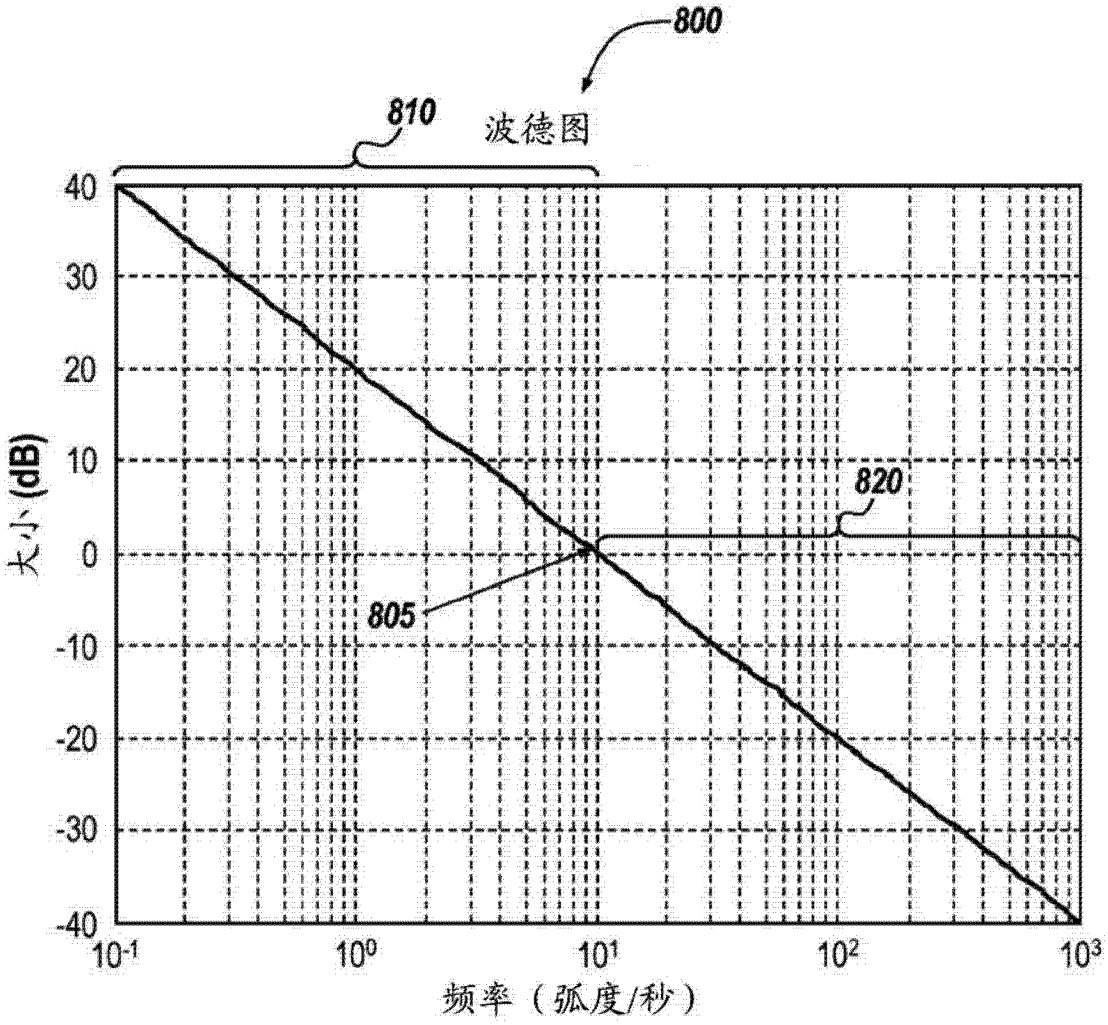

800
810
波德图
40
30
20
10
0
-10
-20
-30
-40
大小 (dB)
820
805
10-1
100
101
102
103
频率（弧度/秒）

图 8

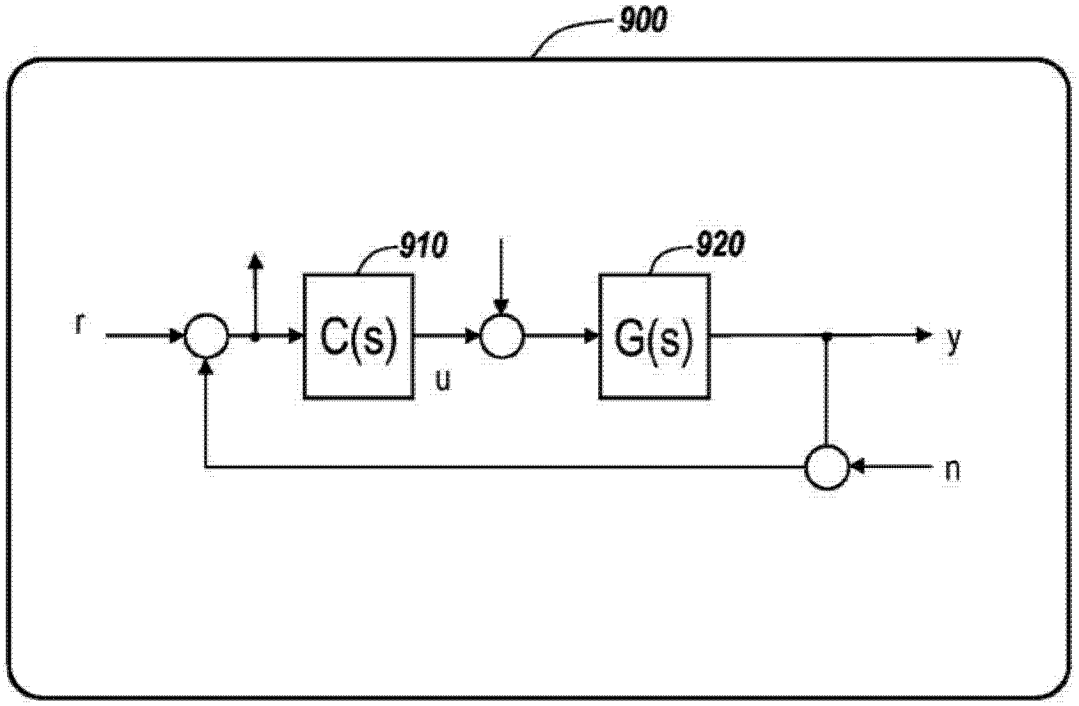


图 9

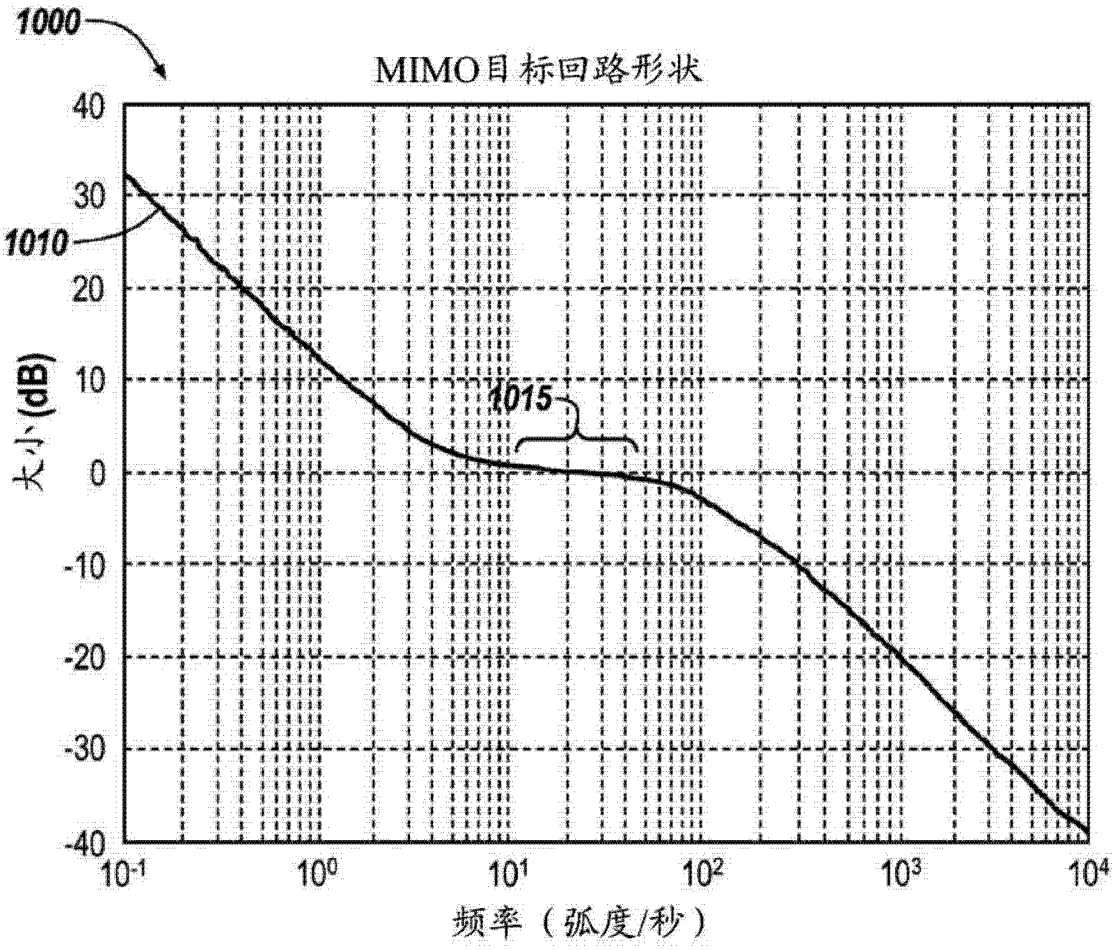

1000
MIMO目标回路形状
40
30
1010
20
10
0
-10
-20
-30
-40
大小(dB)
1015
10-1
100
101
102
103
104
频率（弧度/秒）

图 10

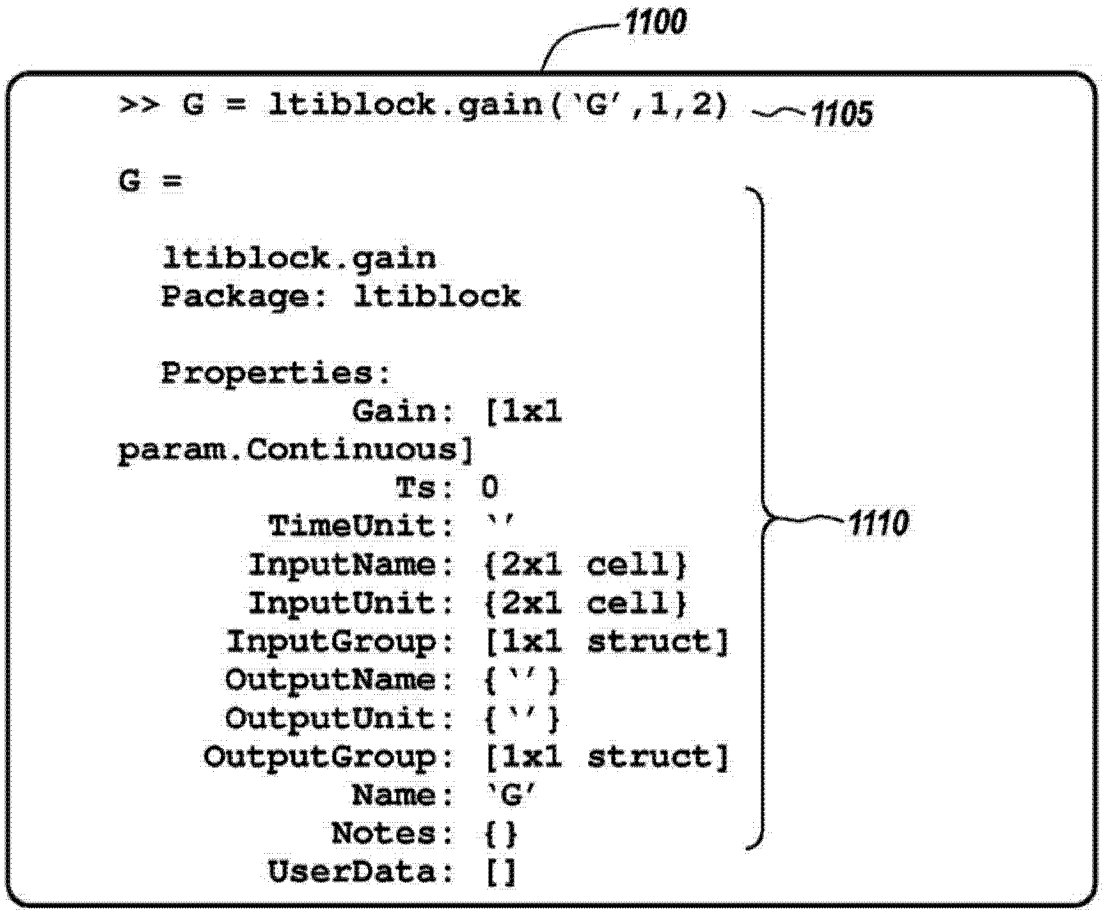

1100
>> G = ltiblock.gain('G',1,2)
1105
G =
ltiblock.gain
Package: ltiblock
Properties:
Gain: [1x1
param.Continuous]
Ts: 0
TimeUnit: ''
InputName: {2x1 cell}
InputUnit: {2x1 cell}
InputGroup: [1x1 struct]
OutputName: {''}
OutputUnit: {''}
OutputGroup: [1x1 struct]
Name: 'G'
Notes: {}
UserData: []
1110

图 11

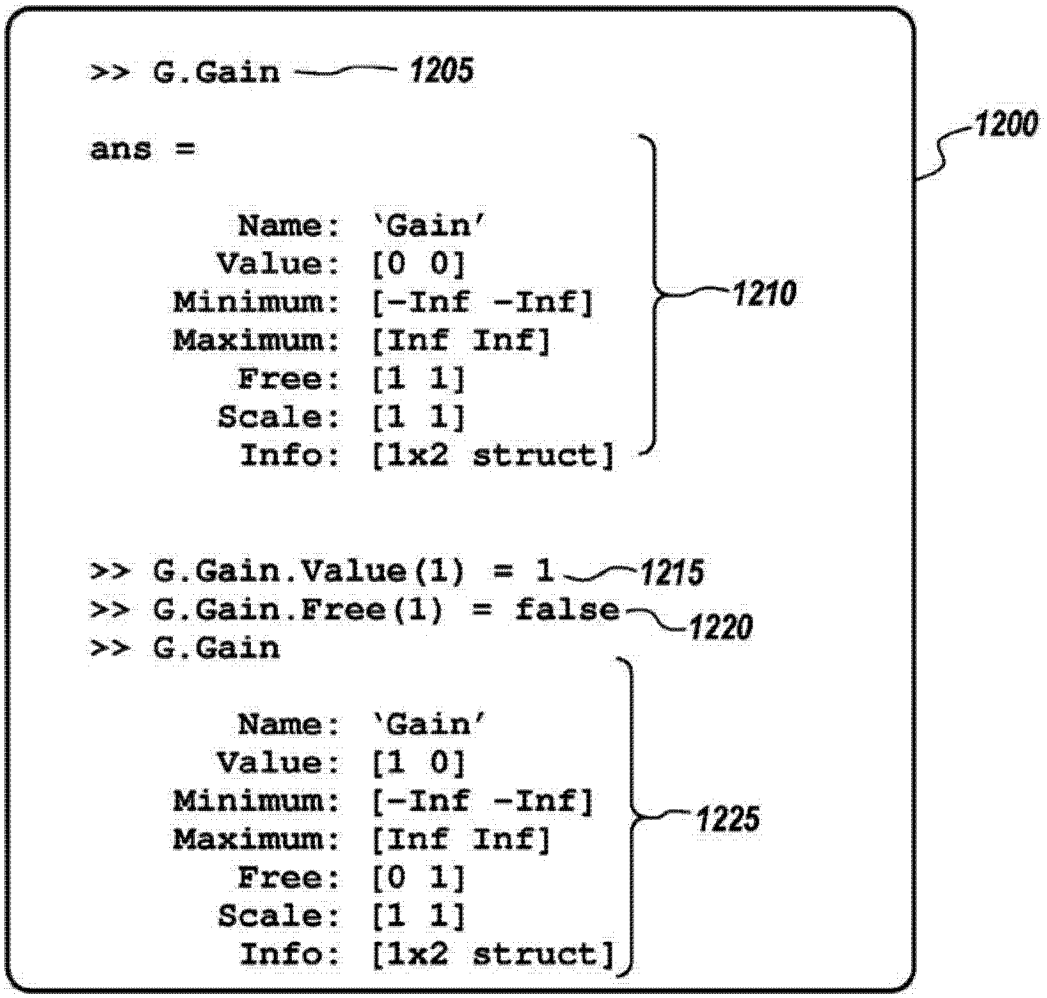

>> G.Gain
1205
1200
ans =
Name: 'Gain'
Value: [0 0]
Minimum: [-Inf -Inf]
Maximum: [Inf Inf]
Free: [1 1]
Scale: [1 1]
Info: [1x2 struct]
1210
>> G.Gain.Value(1) = 1
1215
>> G.Gain.Free(1) = false
1220
>> G.Gain
Name: 'Gain'
Value: [1 0]
Minimum: [-Inf -Inf]
Maximum: [Inf Inf]
Free: [0 1]
Scale: [1 1]
Info: [1x2 struct]
1225

图 12

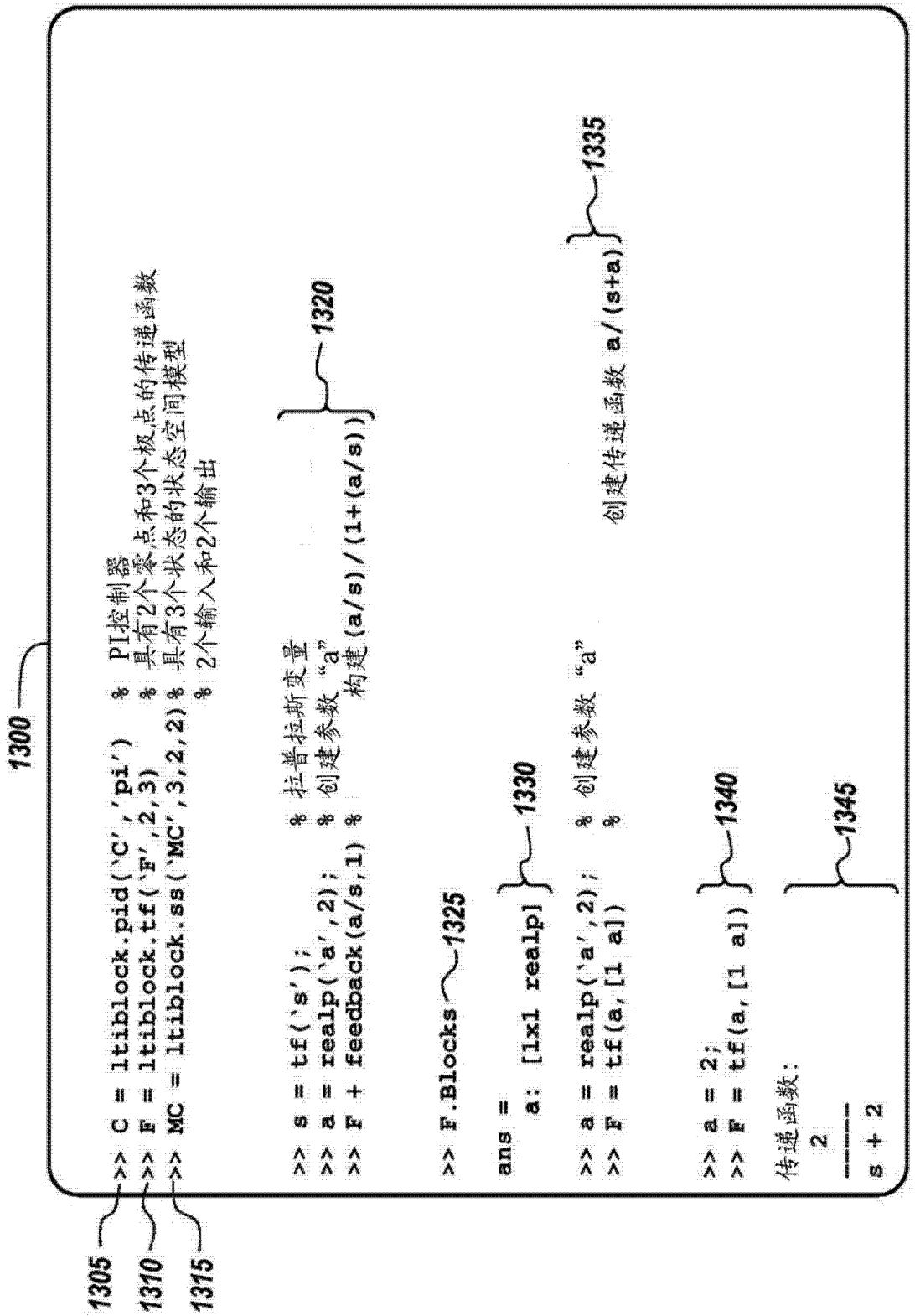

1300
>> C = ltiblock.pid('C','pi') % PI控制器
>> F = ltiblock.tf('F',2,3) % 具有2个零点和3个极点的传递函数
>> MC = ltiblock.ss('MC',3,2,2) % 具有3个状态的状态空间模型
% 2个输入和2个输出
1305
1310
1315
>> s = tf('s'); % 拉普拉斯变量
>> a = realp('a',2); % 创建参数 "a"
>> F + feedback(a/s,1) % 构建 (a/s)/(1+(a/s))
1320
>> F.Blocks
1325
ans =
a: [1x1 realp]
1330
>> a = realp('a',2); % 创建参数 "a"
>> F = tf(a,[1 a]) % 创建传递函数 a/(s+a)
1335
>> a = 2;
>> F = tf(a,[1 a])
1340
传递函数:
2
-----
s + 2
1345

图 13

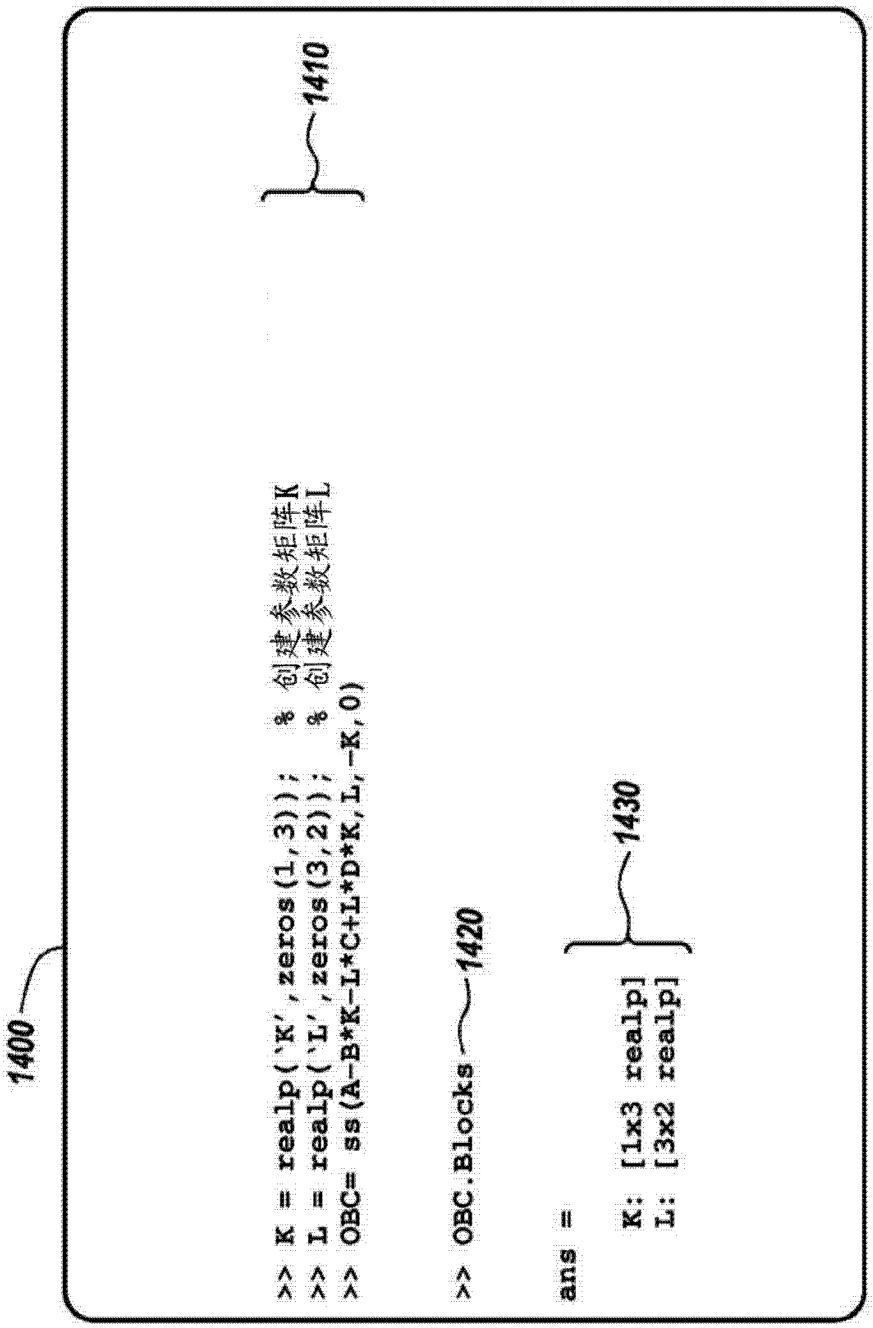

1400
1410
>> K = realp('K',zeros(1,3)); % 创建参数矩阵K
>> L = realp('L',zeros(3,2)); % 创建参数矩阵L
>> OBC= ss(A-B*K-L*C+L*D*K,L,-K,0)
>> OBC.Blocks
1420
ans =
K: [1x3 realp]
L: [3x2 realp]
1430

图 14

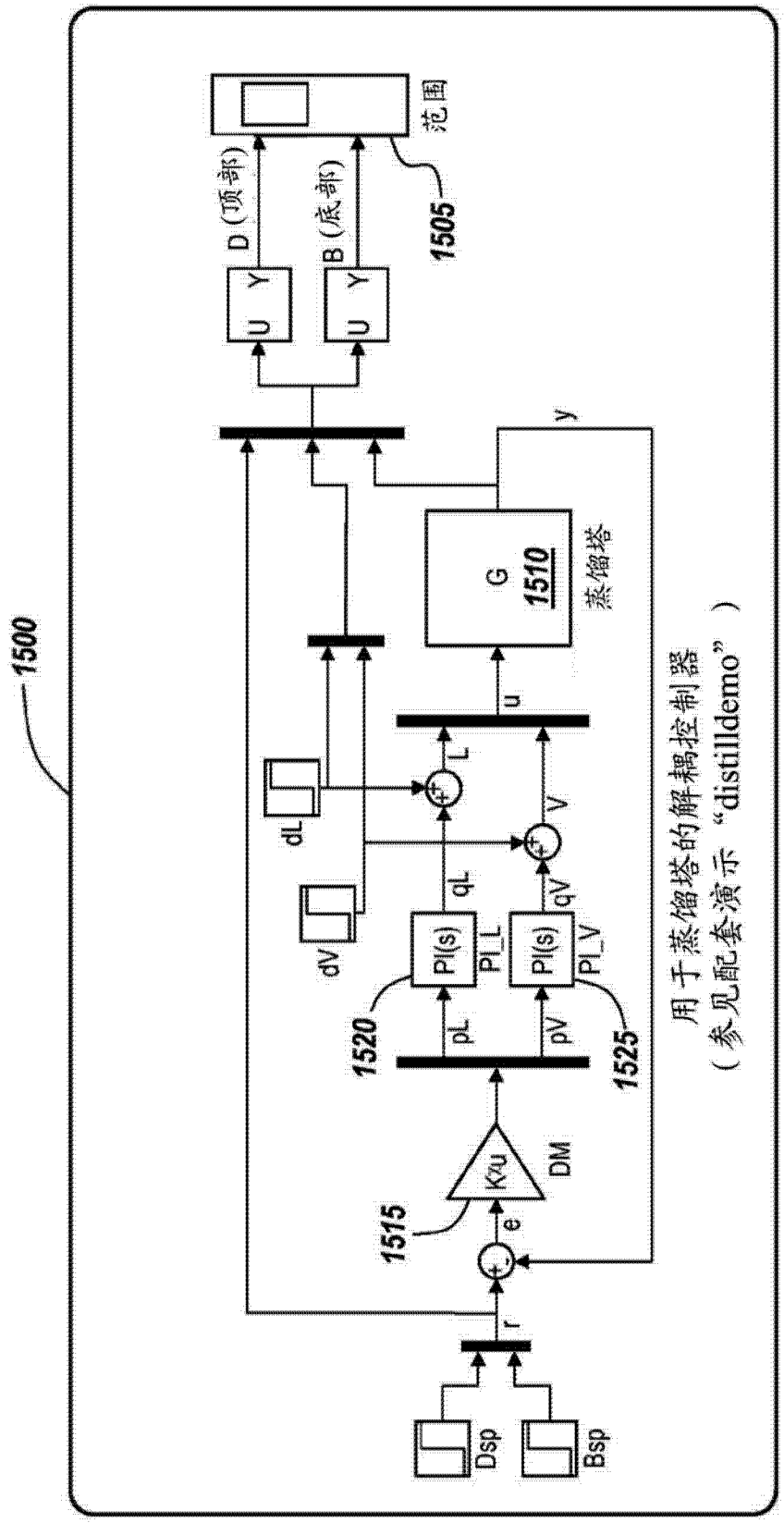

1500
范围
1505
D (顶部)
B (底部)
U Y
y
G
1510
蒸馏塔
u
L
V
dL
dV
qL
qV
PI(s)
PI_L
PI_V
pL
pV
1520
1525
K*u
DM
1515
e
r
Dsp
Bsp
用于蒸馏塔的解耦控制器
（参见配套演示“distilldemo”）

图 15

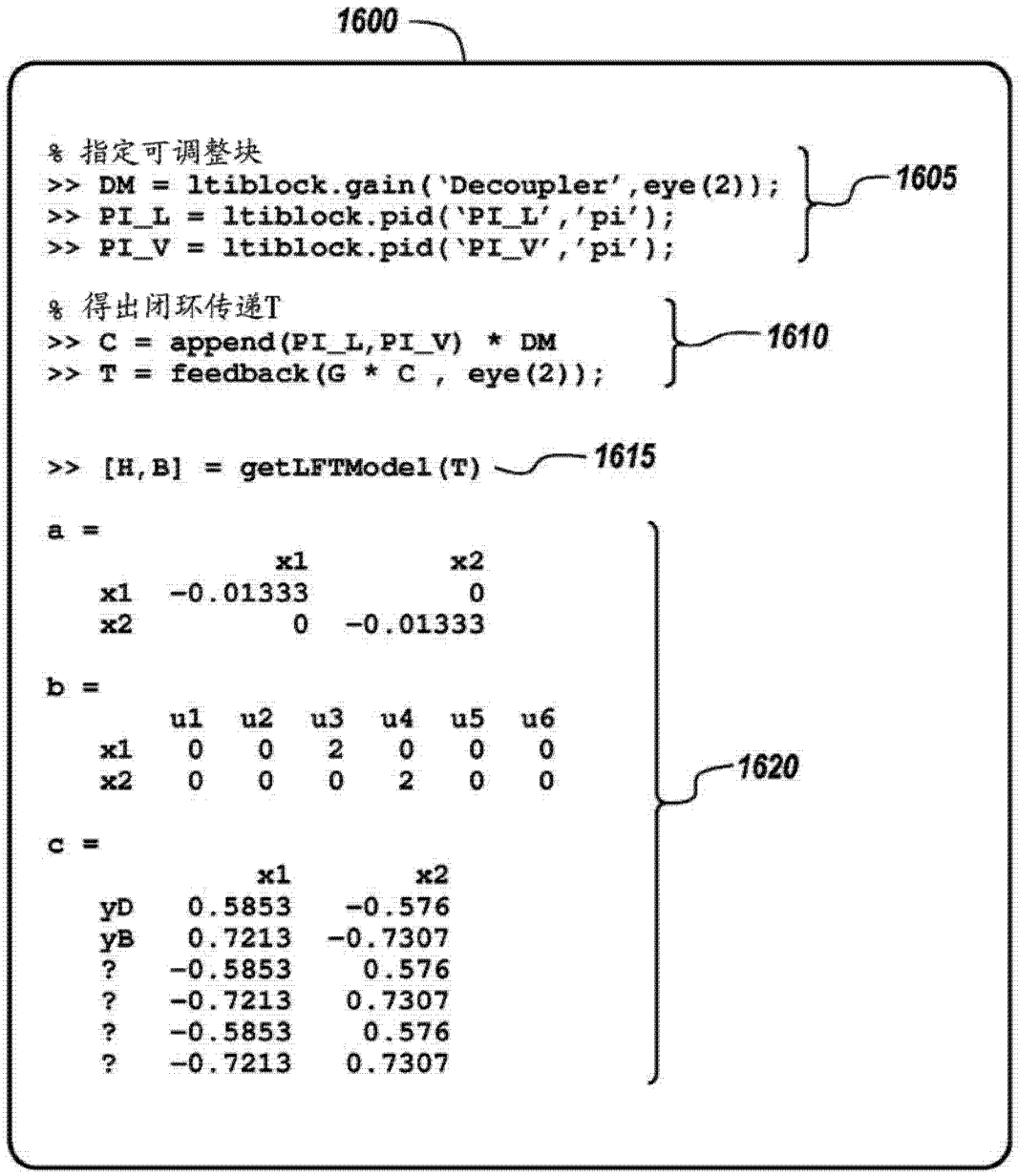

1600
% 指定可调整块
>> DM = ltiblock.gain('Decoupler',eye(2));
>> PI_L = ltiblock.pid('PI_L','pi');
>> PI_V = ltiblock.pid('PI_V','pi');
1605
% 得出闭环传递T
>> C = append(PI_L,PI_V) * DM
>> T = feedback(G * C , eye(2));
1610
>> [H,B] = getLFTModel(T)
1615
a =
x1 x2
x1 -0.01333 0
x2 0 -0.01333
b =
u1 u2 u3 u4 u5 u6
x1 0 0 2 0 0 0
x2 0 0 0 2 0 0
c =
x1 x2
yD 0.5853 -0.576
yB 0.7213 -0.7307
? -0.5853 0.576
? -0.7213 0.7307
? -0.5853 0.576
? -0.7213 0.7307
1620

图 16A

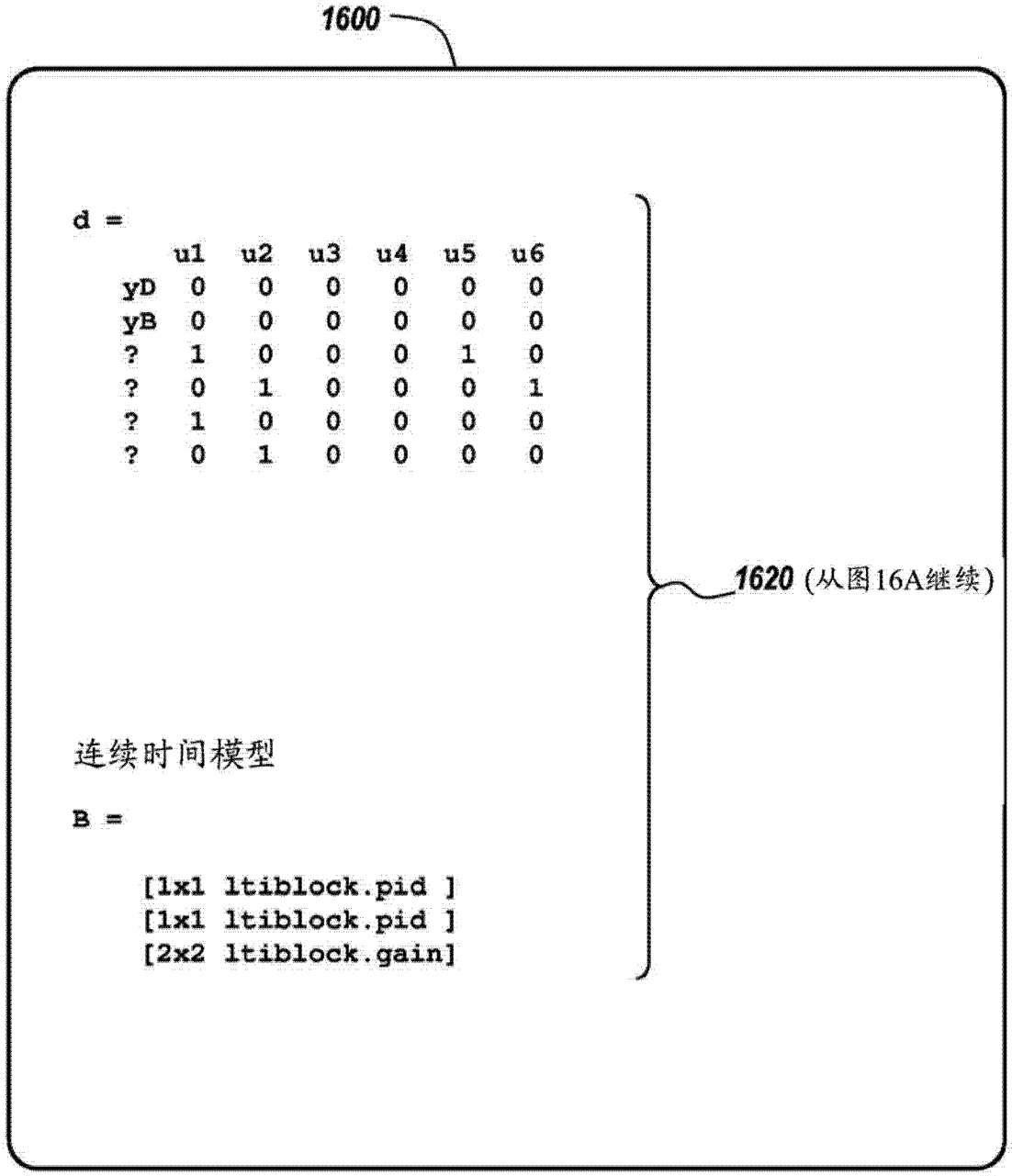

1600
d =
u1 u2 u3 u4 u5 u6
yD 0 0 0 0 0 0
yB 0 0 0 0 0 0
? 1 0 0 0 1 0
? 0 1 0 0 0 1
? 1 0 0 0 0 0
? 0 1 0 0 0 0
1620 (从图16A继续)
连续时间模型
B =
[1x1 ltiblock.pid]
[1x1 ltiblock.pid]
[2x2 ltiblock.gain]

图 16B

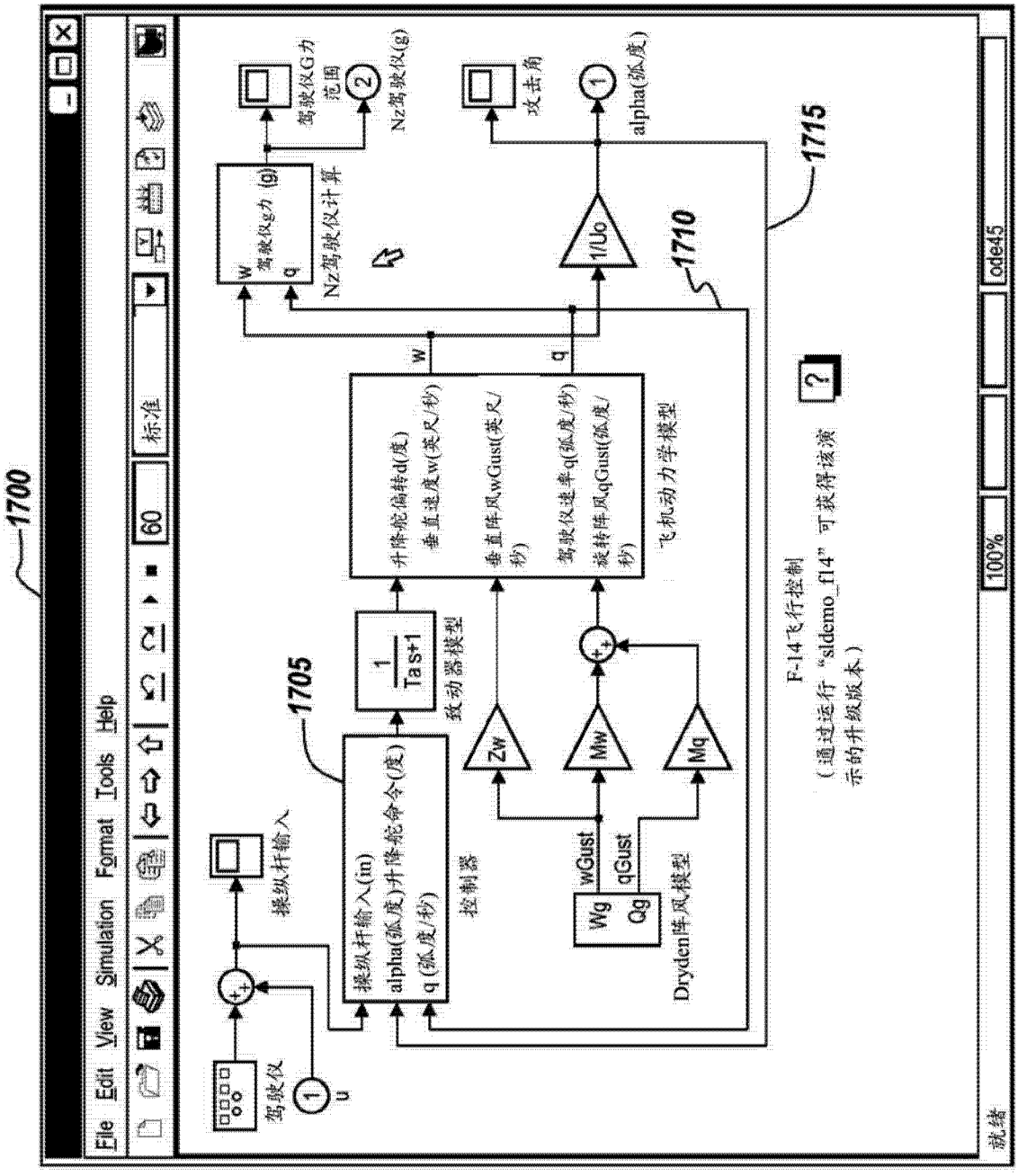

1700
File Edit View Simulation Format Tools Help
标准
60
驾驶仪
u
操纵杆输入
操纵杆输入(in)
alpha(弧度)升降舵命令(度)
q (弧度/秒)
控制器
1705
1
Tas+1
致动器模型
升降舵偏转d(度)
垂直速度w(英尺/秒)
驾驶仪速率q(弧度/秒)
飞机动力学模型
垂直阵风wGust(英尺/秒)
旋转阵风qGust(弧度/秒)
Zw
Mw
Mq
wGust
qGust
Wg
Qg
Dryden阵风模型
w
q
驾驶仪g力 (g)
Nz驾驶仪计算
驾驶仪G力范围
Nz驾驶仪(g)
攻击角
1/Uo
alpha(弧度)
1710
1715
F-14飞行控制
（通过运行“sldemo_f14”可获得该演示的升级版本）
100%
ode45
就绪

图 17

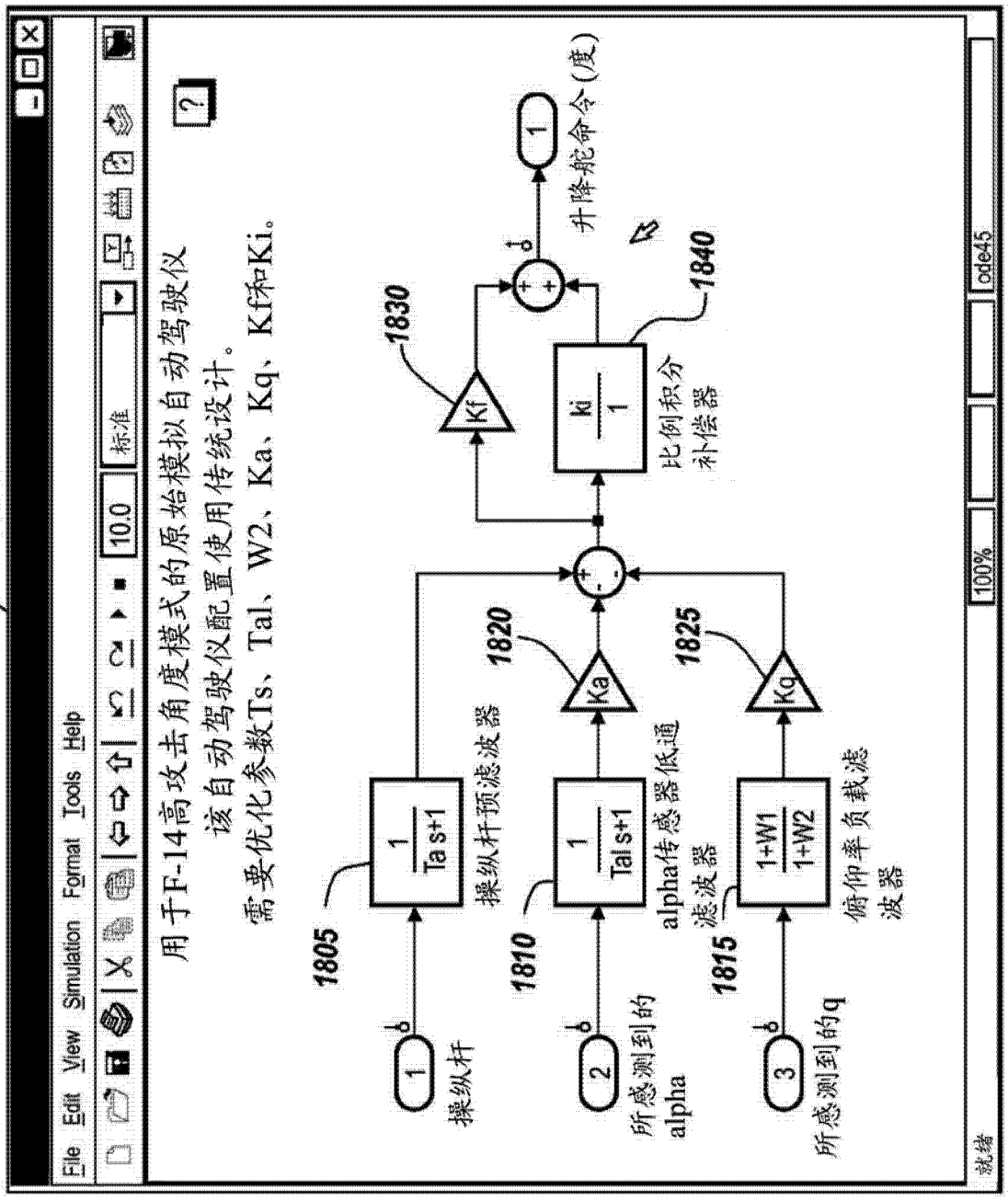

1705-A
File Edit View Simulation Format Tools Help
10.0
标准
用于F-14高攻击角度模式的原始模拟自动驾驶仪
该自动驾驶仪配置使用传统设计。
需要优化参数Ts、Tal、W2、Ka、Kq、Kf和Ki。
1805
1810
1815
1820
1825
1830
1840
操纵杆
所感测到的alpha
所感测到的q
操纵杆预滤波器
alpha传感器低通滤波器
俯仰率负载滤波器
比例积分补偿器
升降舵命令(度)
Kf
Ka
Kq
ki
就绪
100%
ode45

图 18

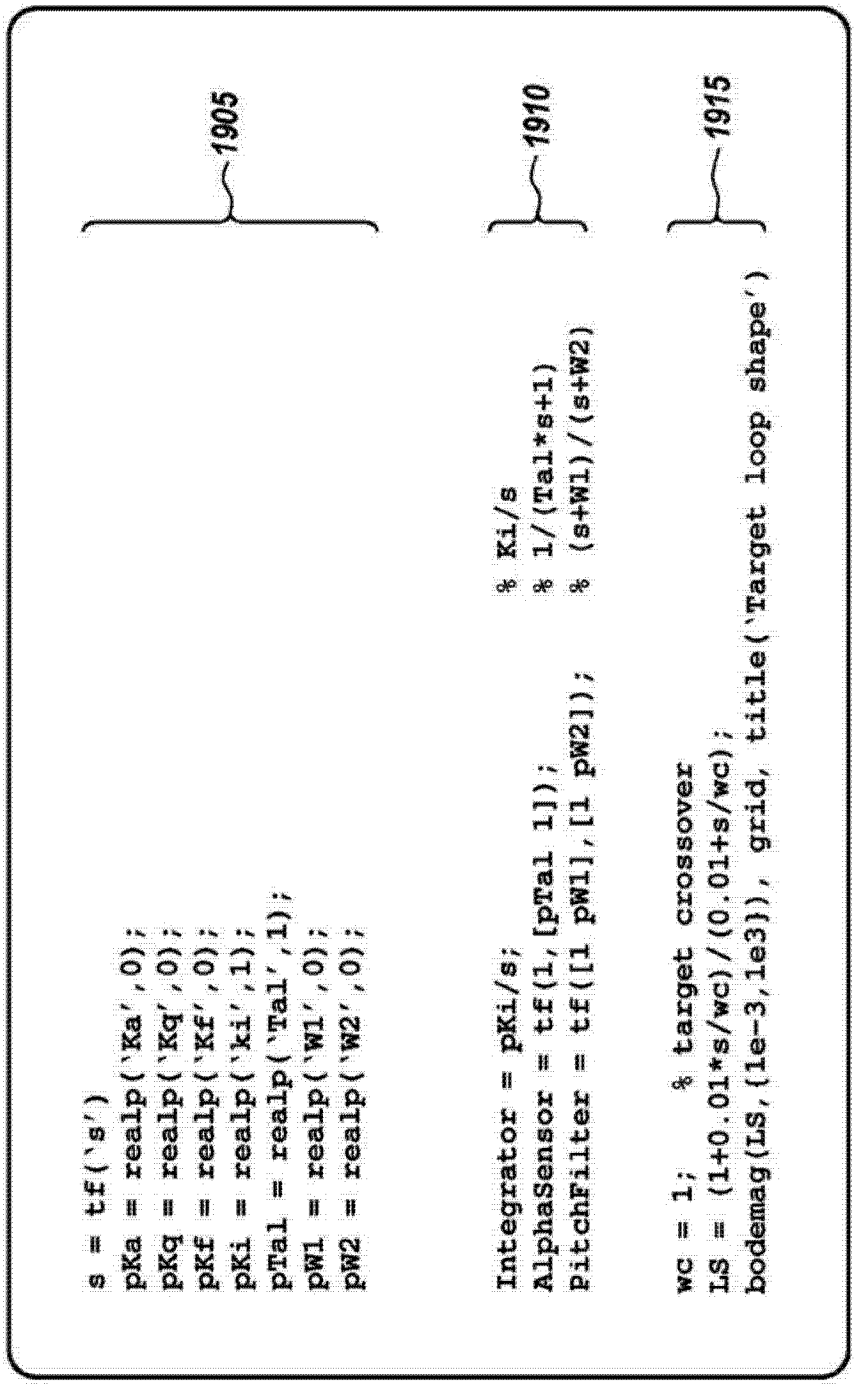

1905
s = tf('s')
pKa = realp('Ka',0);
pKq = realp('Kq',0);
pKf = realp('Kf',0);
pKi = realp('Ki',1);
pTal = realp('Tal',1);
pW1 = realp('W1',0);
pW2 = realp('W2',0);
1910
Integrator = pKi/s; % Ki/s
AlphaSensor = tf(1,[pTal 1]); % 1/(Tal*s+1)
PitchFilter = tf([1 pW1],[1 pW2]); % (s+W1)/(s+W2)
1915
wc = 1; % target crossover
LS = (1+0.01*s/wc)/(0.01+s/wc);
bodemag(LS,{1e-3,1e3}), grid, title('Target loop shape')

图 19

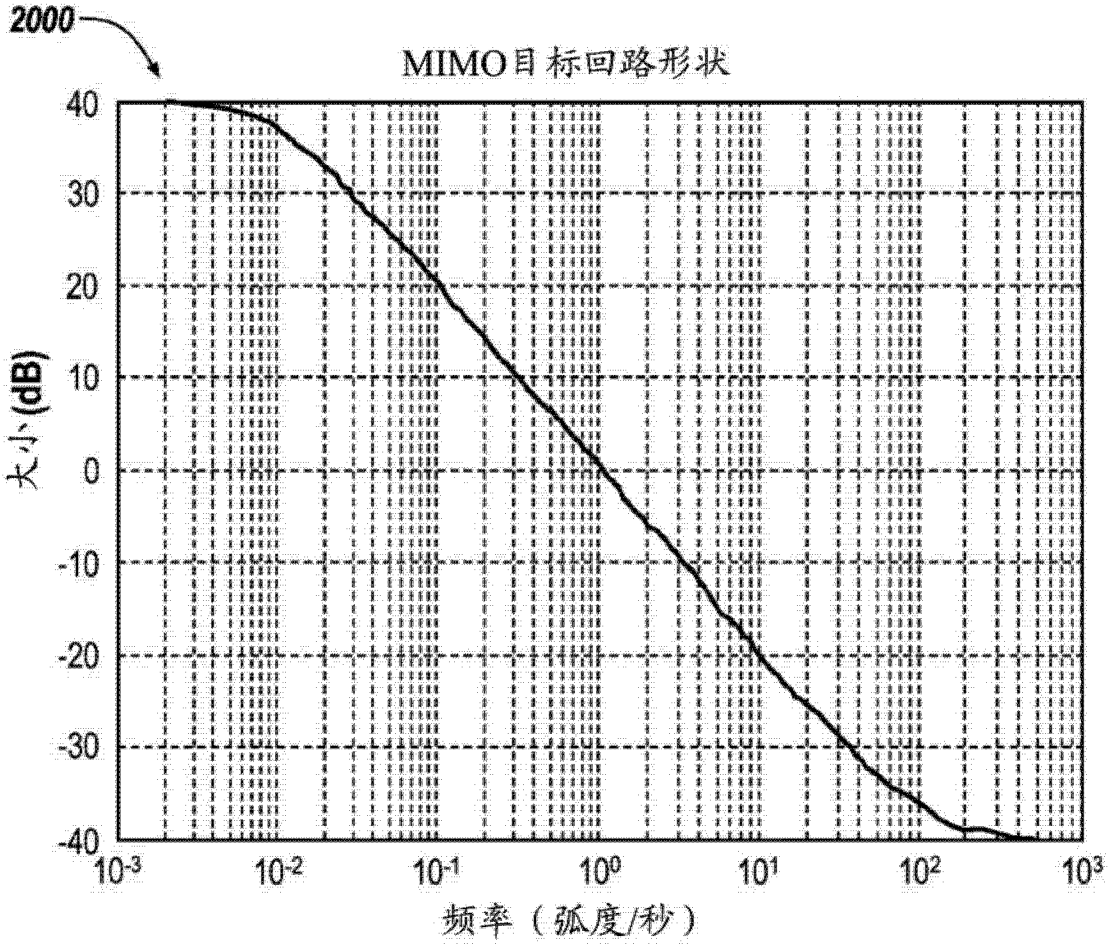

2000
MIMO目标回路形状
40
30
20
10
0
-10
-20
-30
-40
大小(dB)
10-3
10-2
10-1
100
101
102
103
频率（弧度/秒）

图 20

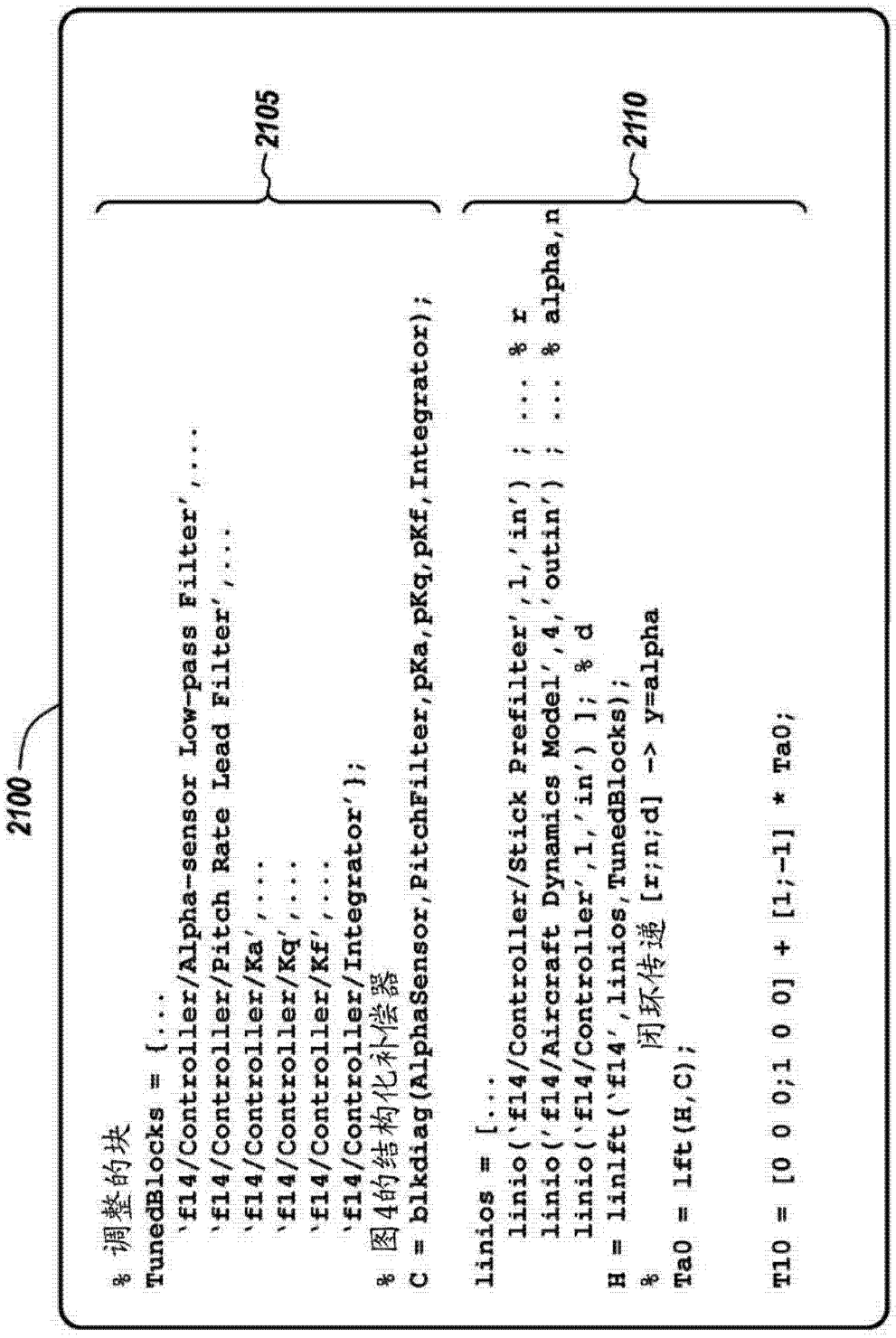

2100
2105
2110
% 调整的块
TunedBlocks = {...
'f14/Controller/Alpha-sensor Low-pass Filter',...
'f14/Controller/Pitch Rate Lead Filter',...
'f14/Controller/Ka',...
'f14/Controller/Kq',...
'f14/Controller/Kf',...
'f14/Controller/Integrator'};
% 图4的结构化补偿器
C = blkdiag(AlphaSensor,PitchFilter,pKa,pKq,pKf,Integrator);
linios = [...
linio('f14/Controller/Stick Prefilter',1,'in') ; ... % r
linio('f14/Aircraft Dynamics Model',4,'outin') ; ... % alpha,n
linio('f14/Controller',1,'in')]; % d
H = linlft('f14',linios,TunedBlocks);
% 闭环传递 [r;n;d] -> y=alpha
Ta0 = lft(H,C);
T10 = [0 0 0;1 0 0] + [1;-1] * Ta0;

图 21

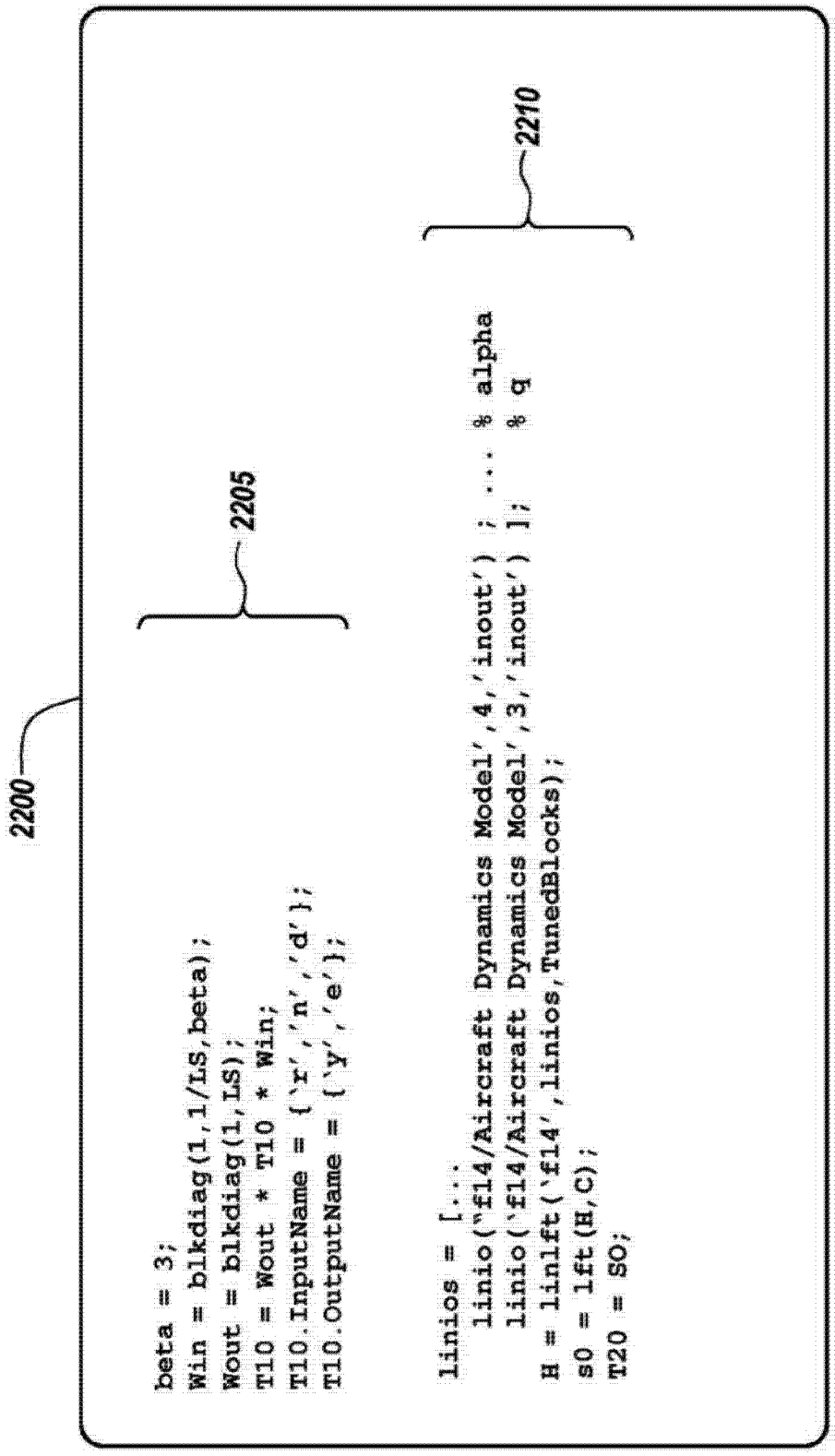

2200
2205
2210
beta = 3;
Win = blkdiag(1,1/LS,beta);
Wout = blkdiag(1,LS);
T10 = Wout * T10 * Win;
T10.InputName = {'r','n','d'};
T10.OutputName = {'y','e'};
linios = [...
linio("f14/Aircraft Dynamics Model',4,'inout') ; ... % alpha
linio('f14/Aircraft Dynamics Model',3,'inout')]; % q
H = linlft('f14',linios,TunedBlocks);
s0 = lft(H,C);
T20 = SO;

图 22

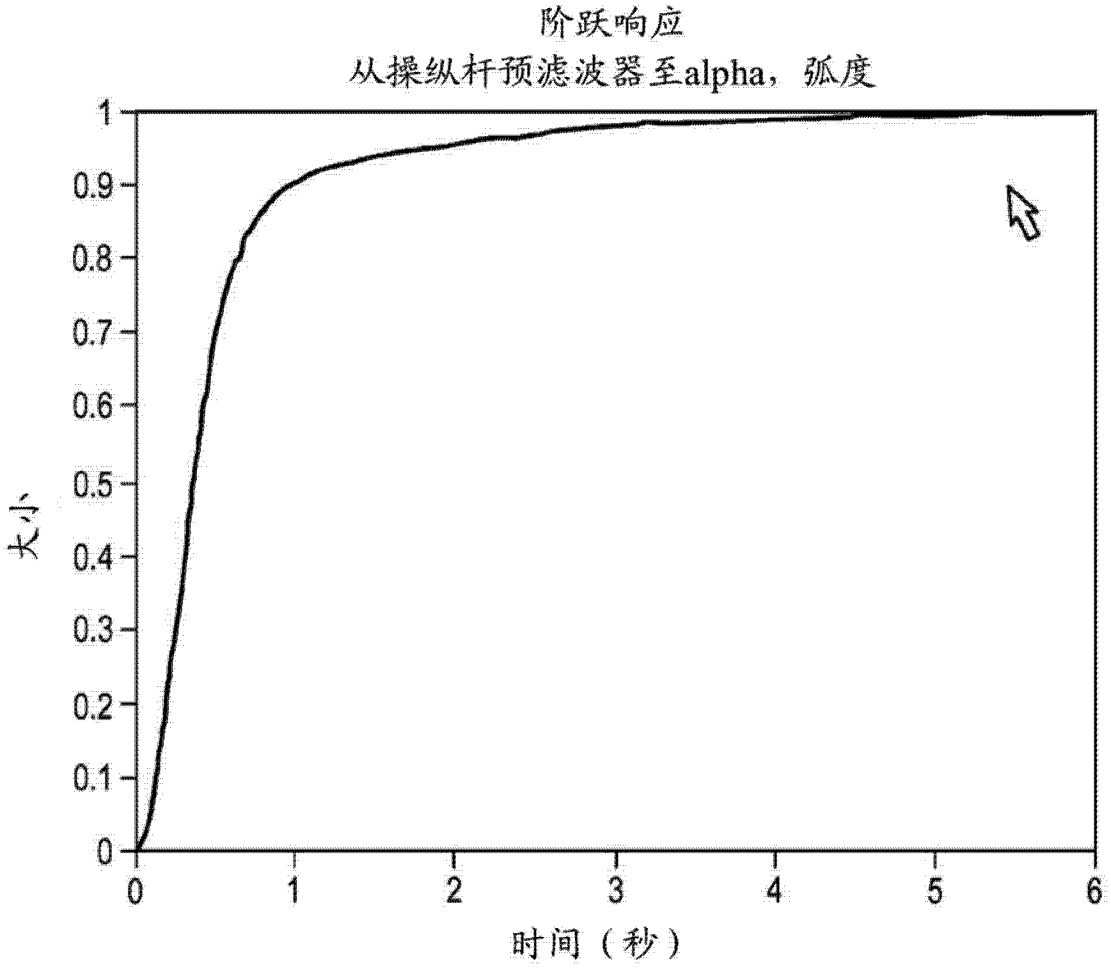

阶跃响应
从操纵杆预滤波器至alpha，弧度
大小
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
1
2
3
4
5
6
时间（秒）

图 23

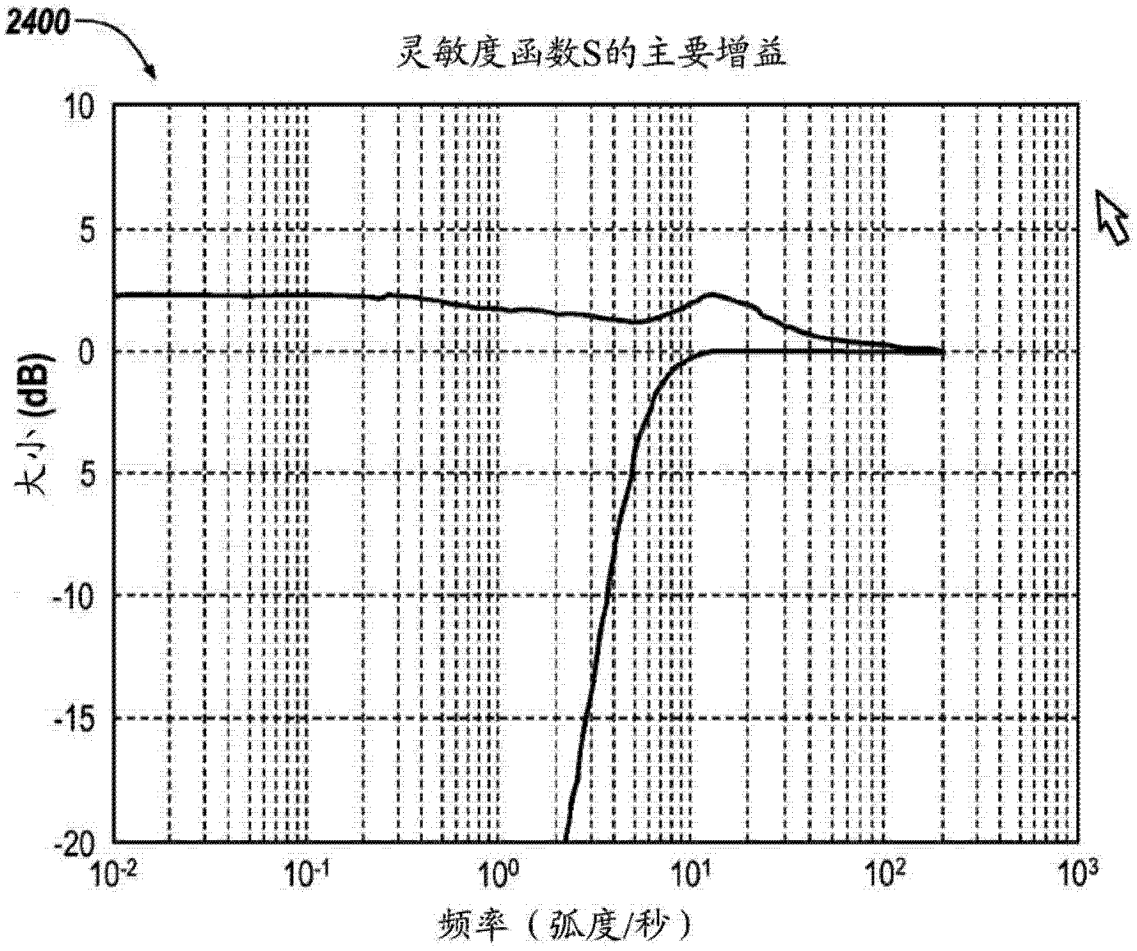

2400
灵敏度函数S的主要增益
10
5
0
5
-10
-15
-20
大小 (dB)
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
频率（弧度/秒）

图 24

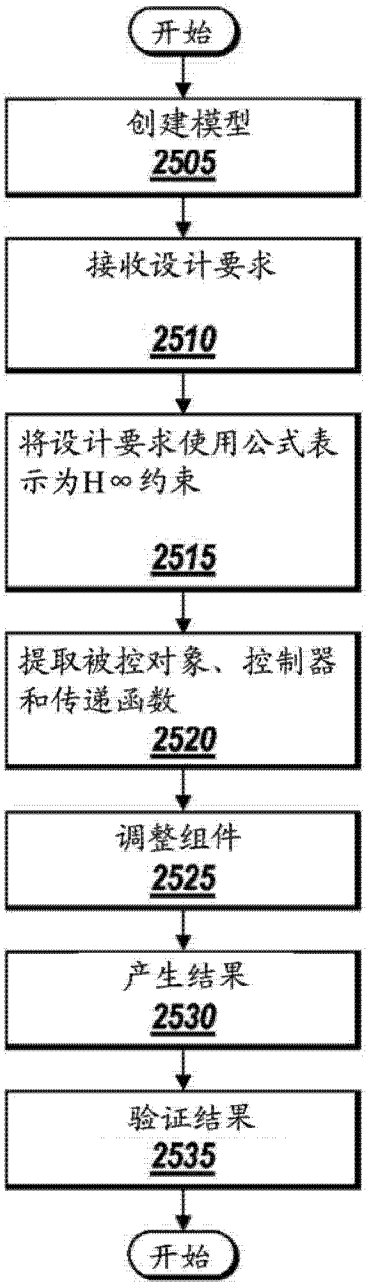

开始
创建模型
2505
接收设计要求
2510
将设计要求使用公式表示为H∞约束
2515
提取被控对象、控制器和传递函数
2520
调整组件
2525
产生结果
2530
验证结果
2535
开始

图 25

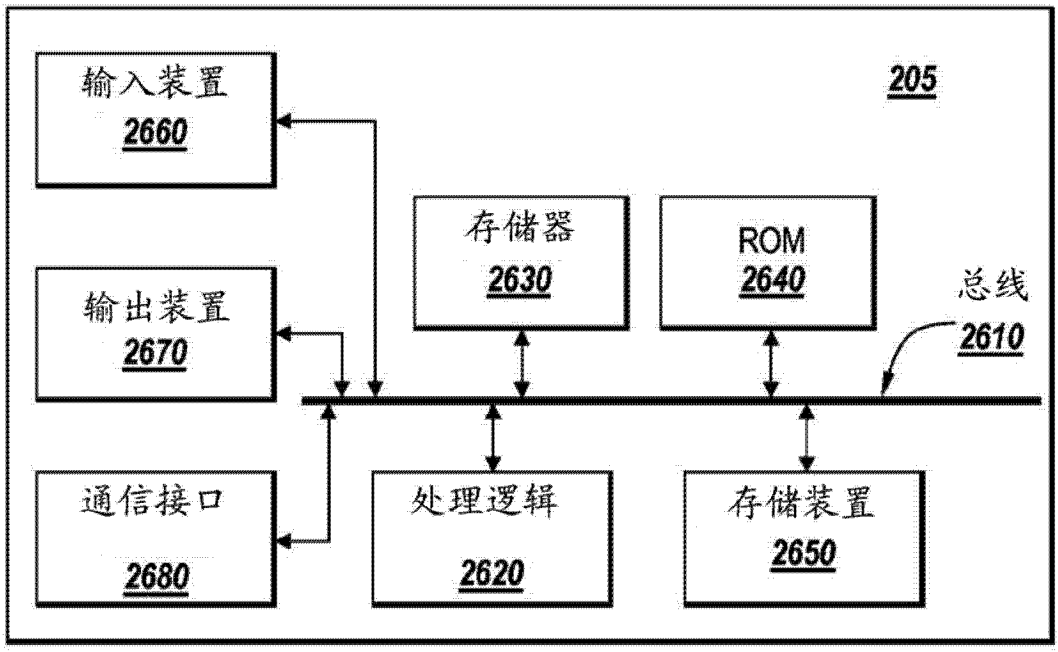

输入装置
2660
205
存储器
2630
ROM
2640
输出装置
2670
总线
2610
通信接口
2680
处理逻辑
2620
存储装置
2650

图 26

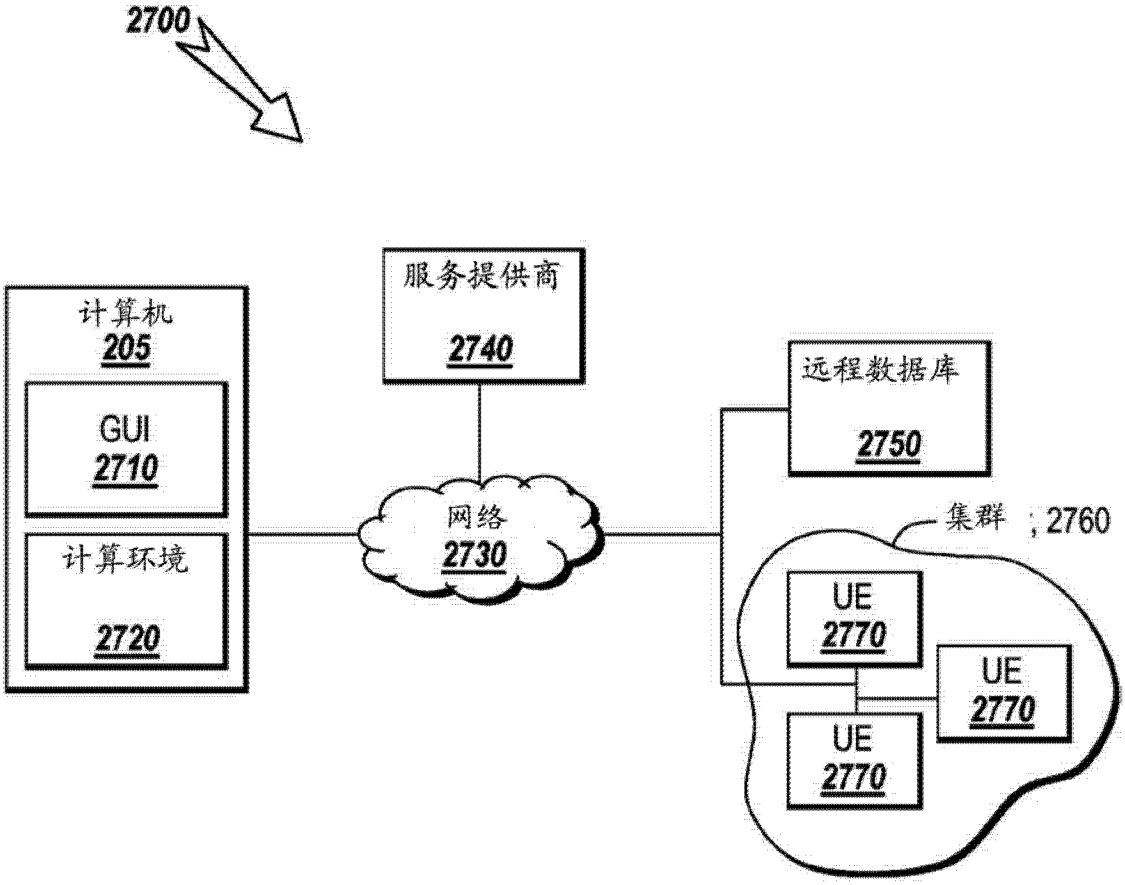

2700
计算机
205
GUI
2710
计算环境
2720
服务提供商
2740
网络
2730
远程数据库
2750
集群 ; 2760
UE
2770
UE
2770
UE
2770

图 27