



(12) 发明专利

(10) 授权公告号 CN 103217909 B

(45) 授权公告日 2015.04.22

(21) 申请号 201310129989.1

CN 102156786 A, 2011.08.17,

(22) 申请日 2013.04.15

徐端等. “基于 OpenGL 的炼钢-连铸过程 3D 自适应仿真设备控件模型”. 《辽宁科技大学学报》. 2012, 第 35 卷 (第 6 期),

(73) 专利权人 东北大学

地址 110819 辽宁省沈阳市和平区文化路 3 号巷 11 号

俞胜平等. “基于虚拟现实的炼钢连铸调度仿真系统”. 《中南大学学报(自然科学版)》. 2009, 第 40 卷

(72) 发明人 朱坦 魏征 赫天章 刘士新

(74) 专利代理机构 沈阳东大知识产权代理有限公司 21109

审查员 盛琳

代理人 崔兰蔚

(51) Int. Cl.

G05B 17/02(2006.01)

(56) 对比文件

JP 特开 2001-290516 A, 2001.10.19,

US 7110835 B2, 2006.09.19,

CN 101604145 A, 2009.12.16,

CN 1556486 A, 2004.12.22,

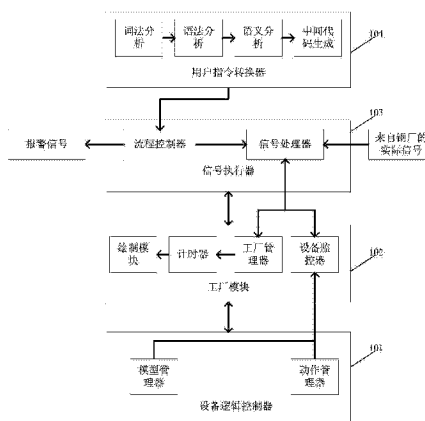
权利要求书2页 说明书20页 附图8页

(54) 发明名称

一种数字化钢厂设备控制仿真系统

(57) 摘要

一种数字化钢厂设备控制仿真系统,属于炼钢连铸技术领域。包括用于将用户提供的指令转换为可被信号执行器接受的信号格式的用户指令转换器、对双向精炼炉、精炼炉、转炉、行车、RH 精炼炉、回转台设备进行 3D 模拟,并对上述设备的外观进行拆分,赋予动作的设备逻辑控制器、用于接受用户指令或者钢厂内设备传递的实际信号,对数字钢厂中的设备信息进行修改并绘制数字钢厂的运行画面的工厂模块,用于接受用户指令转换器发出的用户指令,控制工厂管理器和设备逻辑控制器,实现实际工厂与虚拟工厂的交互及用户与虚拟工厂的交互的信号执行器。本发明采用数字化工厂方式与钢厂设备进行交互方式,通过可视化界面设计工厂布局并对其进行导入和导出,实现兼容外置设备。



1. 一种数字化钢厂设备控制仿真系统,包括:

用户指令转换器(104):用于将用户提供的指令转换为可被信号执行器接受的信号格式;

其特征在于:还包括:

设备逻辑控制器(101):对双向精炼炉、精炼炉、转炉、行车、RH精炼炉、回转台设备进行3D模拟,并对上述设备的外观进行拆分,赋予动作;

工厂模块(102):用于接受用户指令或者钢厂内设备传递的实际信号,对数字化钢厂中的设备信息进行修改并绘制数字化钢厂的运行画面;其中,工厂模块进一步包括:

设备监控器:用于根据接收到的用户指令或钢厂内设备传递的实际信号,调用设备逻辑控制器输出的经拆分后的设备模型及设备动作类型,使工厂模块内的设备增加一种动作、告知用户设备当前正在进行的动作、告知用户行车所处的位置、告知工厂模块行车的跨距、设置行车设备的跨距;

工厂管理器:(1)用于管理钢厂的设备位置信息、设备的旋转角度信息、设备的名称和行车设备的跨距信息;(2)用于根据接收到的用户指令或钢厂内设备传递的实际信号,控制计时器停止计时或者继续运行或者复位或者置数;(3)用于根据接收到的用户指令或钢厂内设备传递的实际信号,在工厂模块增加一种设备;(4)用于根据接收到的用户指令或钢厂内设备传递的实际信号,在工厂模块删除一种设备;(5)用于根据接收到的用户指令或钢厂内设备传递的实际信号,在工厂模块中修改设备的位置;(6)用于根据接收到的用户指令或钢厂内设备传递的实际信号,告知用户设备坐标和旋转角度;

计时器:用于根据工厂管理器的输出,向绘制模块提供时间参数;

绘制模块:用于根据计时器输出的时间参数,绘制钢厂在当前时间参数下的运行画面;对于任一设备的任一视角,根据设备的运动幅度,采用帧率优化算法来判断当前帧是否需要绘制;

若有必要则重绘,若没有必要,则跳过此次重绘;

信号执行器(103):用于接受用户指令转换器(104)发出的用户指令,控制工厂管理器和设备监控器,实现实际钢厂与数字化钢厂的交互及用户与数字化钢厂的交互。

2. 根据权利要求1所述的数字化钢厂设备控制仿真系统,其特征在于:所述的设备逻辑控制器还进一步包括:

模型管理器:用于建立实体设备,包括双向精炼炉、精炼炉、转炉、行车、RH精炼炉、回转台的3D模型,并对上述的3D模型进行拆解并保存;

动作管理器:用于使经模型管理器拆解后的3D模型进行动作,动作类型包括:转炉设备的动作包括空闲、加料、冶炼、溅渣和排渣操作;双向精炼炉、精炼炉和RH精炼炉的动作类型均包括:空闲、钢包进站、精炼炉处理、钢包出站、等待钢包被行车调走;回转台动作类型包括:空闲;在内侧无钢包的情况下,空回转台外侧等待接受首个钢包;在内侧无钢包的情况下,回转台将外侧满钢包转入内侧;在外侧无钢包的情况下,一个钢包在回转台内侧浇铸;在外侧无钢包的情况下,一个浇铸完毕的空钢包在回转台内侧等待;在外侧无钢包的情况下,一个浇铸完毕的空钢包在回转台由内侧转向外侧;回转台内侧钢包浇铸完毕,外侧有满钢包;回转台内侧钢包正在浇铸,外侧有满钢包;行车的动作类型包括:空闲、空载运行和吊钢包运行。

3. 根据权利要求 1 所述的数字化钢厂设备控制仿真系统,其特征在於:所述的信号执行器进一步包括:

流程控制器:用于接受用户指令转换器发来的用户指令,将指令按执行顺序发送给信号处理器;

信号处理器:(1) 用于接受流程控制器发来的用户指令,控制工厂管理器的设备信息和设备位置信息、控制计时器停止计时或者继续运行或者复位或者置数、控制设备监控器调用的设备信息及设备动作信息;(2) 用于接受钢厂的实际信号,按照存储在信号处理器内的、用户自定义的处理该信号的方法来控制工厂管理器的设备信息和设备位置信息、控制计时器停止计时或者继续运行或者复位或者置数、控制设备监控器调用的设备信息及设备动作信息;(3) 用于根据设备监控器、工厂管理器的输出,与设置在信号处理器内部的、由用户自定义的报警条件进行比较,若满足,则报警。

4. 根据权利要求 2 所述的数字化钢厂设备控制仿真系统,其特征在於:所述的对所述的 3D 模型进行拆解,具体拆解过程如下:

将双向精炼炉、精炼炉和 RH 精炼炉均做如下拆分:拆分为炉盖、精炼钢水罐车、钢包、设备支架、升降装置;

将转炉做如下拆分:拆分为钢包车、渣罐车、钢包、倾动装置及设备支架、转炉炉体及托圈、左炉门和右炉门;

将行车做如下拆分:拆分为行车左端梁、行车右端梁、行车主梁、钢丝绳、滑轮组、吊机、钢包;

将回转台做如下拆分:拆分为连铸台、钢包和设备底座。

5. 根据权利要求 1 所述的数字化钢厂设备控制仿真系统,其特征在於:所述的帧率优化算法过程如下:

使数字化钢厂界面满帧运行一段时间,在其间对钢厂中所有的设备的最大运动速度进行采样得到速度集合 S;取集合 S 中元素的中位数  $v\_middle$ ,计算帧率调整函数  $f(v)$ ,公式如下:

$$f(v) = \begin{cases} \frac{v}{v\_middle} & v < v\_middle \\ 1 & v \geq v\_middle \end{cases} \quad (1)$$

式中,  $v$  为当前重绘下设备的最大速度;

把一秒中的用户自定义的最大重绘次数等分成  $N$  组;

统计摄像机视角内的设备的各个组件的运动速度,得出当前重绘下设备的最大运动速度  $v$ ;

将得到的最大运动速度  $v$  带入公式 (1),来判断当前帧是否需要绘制,公式如下:若满足如下公式则重绘:

$$(T_{current} - T_{begin}) \bmod \left( \frac{T}{N} \right) < f(v) \times \left( \frac{T}{N} \right) \quad (2)$$

式中,  $T_{current}$  为当前时间,  $T_{begin}$  为标记时间段的开始时间,  $T$  为标记时间段的长度,  $N$  为时间段的分组数;

若不满足公式 (2) 则跳过本次重绘。

## 一种数字化钢厂设备控制仿真系统

### 技术领域

[0001] 本发明属于炼钢连铸技术领域,特别涉及一种数字化钢厂设备控制仿真系统。

### 背景技术

[0002] 对于生产调度计划较为复杂的工厂,对设备的控制、对钢厂生产计划的调度仍由人来参与制定,这种控制更多的取决于操作者的经验,一旦疏忽大意将造成不可逆转的损失。

[0003] 目前在生产过程中,有人利用数字化工厂产生动态仿真画面的特点来对实际生产过程进行模拟,但这种方法能事先发现生产计划的缺陷,但却往往只是在生产活动之前进行生产过程的仿真或是在生产活动后进行生产活动的重现,很难实现在生产过程中指导生产的目的。一方面是因为数字化工厂很难与灵活的外部监控信号很好的兼容,另一方面,即使可以很好的兼容外部信号,先有后的数字化工厂也很难实现自身对信号的处理加工,因为数字化工厂的功能在开发时就已经被确定,在使用时难以改变。

[0004] 传统的数字化工厂所展示的工厂布局往往难于修改,在实际生产过程中每一个工厂都要使用针对自己开发的适合自己工厂布局的一套数字工厂,这使得数字工厂的普及变得十分困难,对于工厂布局的修改往往停留在数字化工厂的开发阶段,用户难以按照自己的意愿方便的对工厂布局进行改变。

[0005] 除此之外现行的数字工厂模块间的接口对于用户是不可见的,这进一步的限制了数字工厂的通用性,用户只能依靠数字工厂的开发者所规定的方法来操作数字工厂,不能自由的增加外置设备,更不能在已经搭建完毕的工厂上按照自己的所愿意提供的格式提供指导数字工厂运转的信息,这种死板的接口模式也进一步局限了它的发展趋势。

[0006] 因此,目前尚无法使用数字化工厂与实际钢厂内的设备进行交互,对实际钢产设备进行控制。

### 发明内容

[0007] 针对现有技术存在的不足,本发明的目的是提供一种可与实际钢厂内设备进行交互、通过调整生产计划或钢厂内设备的布局已达到对实际工厂设备进行控制的目的。

[0008] 本发明的技术方案是这样实现的:一种数字化钢厂设备控制仿真系统,包括:

[0009] 用户指令转换器:用于将用户提供的指令转换为可被信号执行器接受的信号格式;

[0010] 设备逻辑控制器:对双向精炼炉、精炼炉、转炉、行车、RH精炼炉、回转台设备进行3D模拟,并对上述设备的外观进行拆分,赋予动作;

[0011] 工厂模块:用于接受用户指令或者钢厂内设备传递的实际信号,对数字钢厂中的设备信息进行修改并绘制数字钢厂的运行画面:其中,工厂模块进一步包括:

[0012] 设备监控器:用于根据接收到的用户指令或工厂内设备传递的实际信号,调用设备逻辑控制器输出的经拆分后的设备模型及设备动作类型,使工厂模块内的设备增加一种

动作、告知用户设备当前正在进行的动作、告知用户行车所处的位置、告知工厂模块行车的跨距、设置行车设备的跨距；

[0013] 工厂管理器：(1) 用于管理工厂的设备位置信息、设备的旋转角度信息、设备的名称和行车设备的跨距信息；(2) 用于根据接收到的用户指令或工厂内设备传递的实际信号，控制计时器停止计时或者继续运行或者复位或者置数；(3) 用于根据接收到的用户指令或工厂内设备传递的实际信号，在工厂模块增加一种设备；(4) 用于根据接收到的用户指令或工厂内设备传递的实际信号，在工厂模块删除一种设备；(5) 用于根据接收到的用户指令或工厂内设备传递的实际信号，在工厂模块中修改设备的位置；(6) 用于根据接收到的用户指令或工厂内设备传递的实际信号，告知用户设备坐标和旋转角度；

[0014] 计时器：用于根据工厂管理器的输出，向绘制模块提供时间参数；

[0015] 绘制模块：用于根据计时器输出的时间参数，绘制钢厂在当前时间参数下的运行画面；对于任一设备的任一视角，根据设备的运动幅度，采用帧率优化算法来判断当前帧是否需要绘制；

[0016] 若有必要则重绘，若没有必要，则跳过此次重绘；

[0017] 信号执行器：用于接受用户指令转换器发出的用户指令，控制工厂管理器和设备监控器，实现实际工厂与虚拟工厂的交互及用户与虚拟工厂的交互。

[0018] 所述的设备逻辑控制器还进一步包括：

[0019] 模型管理器：用于建立实体设备，包括双向精炼炉、精炼炉、转炉、行车、RH 精炼炉、回转台的 3D 模型，并对上述的 3D 模型进行拆解并保存；

[0020] 动作管理器：用于使经模型管理器拆解后的 3D 模型进行动作，动作类型包括：转炉设备的动作包括空闲、加料、冶炼、溅渣和排渣操作；双向精炼炉、精炼炉和 RH 精炼炉的动作类型均包括：空闲、钢包进站、精炼炉处理、钢包出站、等待钢包被行车调走；回转台动作类型包括：空闲；在内侧无钢包的情况下，空回转台外侧等待接受首个钢包；在内侧无钢包的情况下，回转台将外侧满钢包转入内侧；在外侧无钢包的情况下，一个钢包在回转台内侧浇铸；在外侧无钢包的情况下，一个浇铸完毕的空钢包在回转台内侧等待；在外侧无钢包的情况下，一个浇铸完毕的空钢包在回转台由内侧转向外侧；回转台内侧钢包浇铸完毕，外侧有满钢包；回转台内侧钢包正在浇铸，外侧有满钢包；行车的动作类型包括：空闲、空载运行和吊钢包运行。

[0021] 所述的信号执行器进一步包括：

[0022] 流程控制器：用于接受用户指令转换器发来的用户指令，将指令按执行顺序发送给信号处理器；

[0023] 信号处理器：(1) 用于接受流程控制器发来的用户指令，控制工厂管理器的设备信息和设备位置信息、控制计时器停止计时或者继续运行或者复位或者置数、控制设备监控器调用的设备信息及设备动作信息；(2) 用于接受钢厂的的实际信号，按照存储在信号处理器内的、用户自定义的处理该信号的方法来控制工厂管理器的设备信息和设备位置信息、控制计时器停止计时或者继续运行或者复位或者置数、控制设备监控器调用的设备信息及设备动作信息；(3) 用于根据设备监控器、工厂管理器的输出，与设置在信号处理器内部的、由用户自定义的报警条件进行比较，若满足，则报警。

[0024] 所述的对 3D 模型进行拆解，具体拆解过程如下：

[0025] 将双向精炼炉、精炼炉和 RH 精炼炉均做如下拆分：拆分为炉盖、精炼钢水罐车、钢包、设备支架、升降装置；

[0026] 将转炉做如下拆分：拆分为钢包车、渣罐车、钢包、倾动装置及设备支架、转炉炉体及托圈、左炉门和右炉门；

[0027] 将行车做如下拆分：拆分为行车端梁、行车右端梁、行车主梁、钢丝绳、滑轮组、吊机、钢包；

[0028] 将回转台做如下拆分：拆分为连铸台、钢包和设备底座。

[0029] 所述的帧率优化算法过程如下：

[0030] 使数字工厂界面满帧运行一段时间，在其间对工厂中所有的设备的最大运动速度进行采样得到速度集合 S；取集合 S 中元素的中位数  $v_{middle}$ ，计算帧率调整函数  $f(v)$ ，公式如下：

$$[0031] \quad f(v) = \begin{cases} \frac{v}{v_{middle}} & v < v_{middle} \\ 1 & v \geq v_{middle} \end{cases} \quad (1)$$

[0032] 式中， $v$  为当前重绘下设备的最大速度；

[0033] 把一秒中的用户自定义的最大重绘次数等分成 N 组；

[0034] 统计摄像机视角内的设备的各个组件的运动速度，得出当前重绘下设备的最大运动速度  $v$ ；

[0035] 将得到的最大运动速度  $v$  带入公式 (1)，来判断当前帧是否需要绘制，公式如下：

[0036] 若满足如下公式则重绘：

$$[0037] \quad (T_{current} - T_{begin}) \bmod \left(\frac{T}{N}\right) < f(v) \times \left(\frac{T}{N}\right) \quad (2)$$

[0038] 式中， $T_{current}$  为当前时间， $T_{begin}$  为标记时间段的开始时间， $T$  为标记时间段的长度， $N$  为时间段的分组数；

[0039] 若不满足公式 (2) 则跳过本次重绘。

[0040] 本发明的优点：本发明提供一种采用数字化工厂方式与钢厂设备进行交互的系统。数字工厂可以通过可视化界面设计工厂布局并对其进行导入和导出。本发明的系统可以方便的兼容外置设备，用户只需要按照自己的意愿编写相应功能控制指令，就可以实现拓展数字化工厂功能的目的，大大增加了数字化工厂的可拓展性。使数字化工厂可以在实时监控并重现工厂生产活动的过程中对其工厂中的一些调度问题进行判断，从而达到纠正优化不适当的生产过程，促进生产的目的。

[0041] 数字化工厂提供一种语法相对简单的语言，使用者可以通过本语言结合响应的设备所提供的操作与监听方法来方便的完成外部信号的识别，通过编写一些监听脚本，数字化工厂可以及时的检测工厂内部的问题并向外发送信息。使本数字化工厂不但作为一种仿真系统，更可以作为一种信息处理系统。

[0042] 应用重绘方法后，重绘次数明显减少，为其他的并行任务节省资源，有很大的实际意义。

[0043] 数字化工厂可以方便的通过外部信号监听脚本来识别外部信号，实时显示工厂的生产情况，并根据相应的监控发现工厂生产过程中出现或将要出现的问题，并及时的将相

关问题反馈给需要处理问题的相关人员。

[0044] 在脚本系统的支持下,本数字化工厂还可以作为一个接口自由可拓展的控件出现,方便的接受其他信号的控制,在检验与生产调度有关的优化算法时,可以通过内建的监控脚本发现调度过程中出现的关键错误,由于本数字化工厂所实现的是一种 3D 全景仿真,与 2D 的演示系统相比从更加全面的信息的基础上发现问题。

#### 附图说明

- [0045] 图 1 为本发明一种实施方式的钢厂设备控制系统结构框图；  
[0046] 图 2 为本发明一种实施方式的确定有限状态自动机的构造图；  
[0047] 图 3 为本发明一种实施方式的双向精炼炉的效果图；  
[0048] 图 4 为本发明一种实施方式的双向精炼炉的 3D 模型拆解图；  
[0049] 图 5 为本发明一种实施方式的精炼炉的效果图；  
[0050] 图 6 为本发明一种实施方式的转炉的效果图；  
[0051] 图 7 为本发明一种实施方式的转炉的 3D 模型拆解图；  
[0052] 图 8 为本发明一种实施方式的行车的效果图；  
[0053] 图 9 为本发明一种实施方式的行车的 3D 模型拆解图；  
[0054] 图 10 为本发明一种实施方式的 RH 精炼炉的效果图；  
[0055] 图 11 为本发明一种实施方式的 RH 精炼炉的 3D 模型拆解图；  
[0056] 图 12 为本发明一种实施方式的回转台的效果图；  
[0057] 图 13 为本发明一种实施方式的回转台的 3D 模型拆解图。

#### 具体实施方式

[0058] 下面结合附图对本发明的实施方式作进一步详细的说明。

[0059] 本实施方式给出一种数字化钢厂设备控制仿真系统,如图 1 所示。包括:用户指令转换器 104、设备逻辑控制器 101、工厂模块 102 和信号执行器 103,其中,设备逻辑控制器其进一步包括模型管理器和动作管理器,工厂模块进一步包括设备监控器、工厂管理器、计时器和绘制模块,下面对这些模块作详细的说明:

[0060] (1) 用户指令转换器:使用者可以通过操作平台向钢厂模块发送驱动钢厂运动的命令,此命令经过脚本编译器的解释,产生执行器所能识别的命令,完成用户与数字钢厂之间的交互。该转换器包括语法分析、语义分析和中间代码生成。

[0061] 首先钢厂设备控制系统为用户提供的输入接口,用户在该装置内以指令的形式输入实际钢厂的设备信息及设备的动作信息,其次,该转换器采用构造确定有限状态自动机(DFA, deterministic finite automaton)的方法(如图 2 所示)进行词法分析、使用了 LL(1) 分析法进行语法分析。

[0062] 由于现在的编译技术已经非常完善,故不给出详细的实现过程,只给出用户指令转换器为用户提供的输入指令用的脚本语言语法列表。

[0063] 下面给出脚本语言的文法定义,法规则中的终结符非终结符如表 1 和表 2 所示:

[0064] 表 1 位非终结符列表

[0065]

符号名称	符号意义
Z	命令集合，推导从本符号开始
type	所有变量的类型
declarelist	函数的参数列表
E	函数体所表示的命令集合
numberlist	结构体定义时的变量集合
sen	函数体内部的一条命令
ELSE	分支判断的假分支
E2	算数表达式
E3	对符号左边的数据进行赋值操作
E4	除赋值操作的算数表达式
E6	对符号左边的数据进行是否大于、大于等于、小于、小于等于、等于的逻辑运算
E5	加减乘除取余的算数表达式
E1	对符号左边的数据进行加减运算操作
T	包含乘除取余三种运算的算数表达式

[0066]

T1	对符号左边的数据进行乘、除、取余运算操作
F	变量、函数、或由括号提升优先级的算数表达式
A	对用户自定义符号的相关操作，在推导过程中自定义符号出现在 A 的左边，包括对用户自定义符号以 valuelist 为参数列表调用函数，以 E2 的值为索引取得数组型变量的一个成员，以另一个用户自定义符号为索引获取结构体的一个成员变量和指定用户自定义符号为一个变量名（即 A 不代表任何动作）

[0067] 表 2 为终结符列表



[0068]

符号名称	符号意义
function	函数定义开始的标志
name	用户自定义的所有名称
~	空, 推导出此符号的符号消失
#	结束标志
struct	结构体定义和声明的标志
do	循环结构开始的标志
while	循环体结束的标志
if	分支判断的开始标志
else	分支判断第假分支开始的标志
flag	跳转标志声明命令的标志
jump	跳转命令的标志
typename	int, float 或 string
return	函数返回变量的标志
c	立即数
() {} [] . 7 种符号	在特定的规则中有特定的意义, 在规则集中详述

[0069] 再给出转换规如表 3 所示 :

[0070] 表 3 位规则集表

[0071]

编 码	类 型	语法规则	说明
1	变 量 控 制	$A \rightarrow ( \text{valuelist} )$	调用函数
2		$A \rightarrow [ E2 ] A$	读取数组中的成员
3		$A \rightarrow . \text{name } A$	读取结构体中的成员
4		$A \rightarrow \sim$	空动作
5		$E1 \rightarrow w1 \ T \ E1$	中等优先级运算
6		$E1 \rightarrow \sim$	空动作
7		$E2 \rightarrow E4 \ E3$	最低优先级运算
8		$E3 \rightarrow = \ E4 \ E3$	最低优先级运算
9		$E3 \rightarrow \sim$	空动作
10		$E4 \rightarrow E5 \ E6$	次低优先级运算
11		$E5 \rightarrow \& \ E1$	中等优先级运算
12		$E6 \rightarrow w3 \ E5 \ E6$	次低低优先级运算
13		$E6 \rightarrow \sim$	空动作
14		$F \rightarrow i$	取最小运算单元

[0072]	流 程 控 制	15	$F \rightarrow ( E2 )$	最优先运算 (括号)
		16	$T \rightarrow F T1$	最高优先级运算
		17	$T1 \rightarrow w2 F T1$	最高优先级运算
		18	$T1 \rightarrow \sim$	空动作
		19	$valuelist \rightarrow E2, valuelist$	参数表 (调用函数时使用)
		20	$valuelist \rightarrow \sim$	空动作
		21	$i \rightarrow c$	最小运算单元的常数推导
		22	$i \rightarrow name A$	最小运算单元的变量推导
		23	$Z \rightarrow function\ type\ name$ $(\ declarelist ) \{ E \} Z$	子程序定义
		24	$Z \rightarrow \#$	结束标志
		25	$Z \rightarrow type\ name ; Z$	全局变量声明
		26	$Z \rightarrow struct\ name$ $\{ numberlist \} ; Z$	结构体变量声明
		27	$Z \rightarrow \sim$	空动作
		28	$E \rightarrow \{ E \} E$	拓展变量空间
		29	$E \rightarrow sen\ E$	增加顺序语句
		30	$E \rightarrow \sim$	空动作
		31	$declarelist \rightarrow type\ name ;$ $declarelist$	参数列表 (声明过程用)
		32	$declarelist \rightarrow \sim$	空动作
		33	$numberlist \rightarrow type\ name ;$ $numberlist$	结构体变量的变量集合
		34	$numberlist \rightarrow \sim$	空动作
		35	$sen \rightarrow do \{ E \} while ( E2 ) ;$	带判断的循环结构
		36	$sen \rightarrow type\ name ;$	局部变量声明
37	$sen \rightarrow E2 ;$	运算及功能调用		
38	$sen \rightarrow flag\ name ;$	跳转标志		
39	$sen \rightarrow jump\ name ( E2 ) ;$	跳转指令		
40	$sen \rightarrow return\ E2 ;$	函数返回值		
41	$sen \rightarrow if ( E2 ) \{ E \} ELSE$	判断结构		
42	$ELSE \rightarrow else \{ E \}$	判断结构的后缀		
43	$ELSE \rightarrow \sim$	空动作		
44	$type \rightarrow [ E2 ] type$	数组类型		
45	$type \rightarrow typename$	基本类型		
46	$type \rightarrow struct\ name$	结构体类型		

[0073] 注：

[0074] 1. 上述语法符合 LL(1) 分析法的基本要求, 可以采用此法进行语法分析, 推导从 Z 开始。

[0075] 2. 变量控制模块中, w1, w2, w3 分别表示运算符的集合, 在本钢厂仿真系统中, 采用  $w1 \{ +, - \} > w2 \{ *, /, \% \} > w3 \{ ==, >, <, >=, <= \} > \{ = \}$  的规则构造运算体系, 若有额外需求, 可以通过以下规则方便的扩展：

[0076] 若存在  $A \rightarrow BC, C \rightarrow w_xBC, C \rightarrow \sim, B \rightarrow DE, E \rightarrow w_zDE, E \rightarrow \sim$ , 此时满足  $w_z > w_x$

[0077] 若想增加一种运算, 其优先级为  $w_y < w_z < w_x$ , 则需要执行以下两步：

[0078] a) 将  $A \rightarrow BC, C \rightarrow w_xBC$  中的 B 改为 F。

[0079] b) 增加  $F \rightarrow BH, H \rightarrow w_yGH, H \rightarrow \sim$ 。

[0080] 3. 本实施方式中的用户指令转换器并不在语法分析阶段进行类型检查,若进行函数调用时(包括对最小动作集合与自定义函数的调用)出现参数类型不匹配的情况,将在运行时提示。

[0081] 在本数字化钢厂中,用户指令转换器按照上述规则。以 LL(1) 分析法和有限自动机解释上述命令,在语法分析的同时进行语义分析,产生流程控制器所能识别的信号,这里给出语义分析规则如下表,其中由方括号标注的语句为流程控制器所能识别的操作(流程控制器所能识别的操作将在对信号执行器的介绍中给出)有的操作可能有多个参数,以空格标明,临时变量是在解释某一种推导的情况下临时生成的变量,这些变量的变量名互不冲突,而且可以转换为任意一种变量类型。上表中的“推导非终结符 XX”是指按照 LL(1) 分析法的分析,命令可以被确定为由若干语法规则推演的结果,由于每个非终结符的推导式已经被确定,故可以由其已经确定的推导式继续进行推导,从而生成完整的命令列表,如表 4 所示:

[0082] 表 4 命令列表

[0083]

编码	类型	语法规则	说明
1	变量控制	$A \rightarrow ( \text{valuelist} )$	[POP 临时变量 1] 推导非终结符 valuelist [PUSHIP] [EXEF 临时变量 1]
2		$A \rightarrow [ E2 ] A$	推导非终结符 E2 [POP 临时变量 1] [ARRAY 临时变量 1] 推导非终结符 A
3		$A \rightarrow . \text{ name } A$	[NUMBER name] 推导非终结符 A
4		$A \rightarrow \sim$	
5		$E1 \rightarrow w1 T E1$	推导非终结符 T 推导非终结符 E1 [POP 临时变量 1] [POP 临时变量 2] [FINGER_w1 w1 临时变量 2 临时变量 1 临时变量 3] [PUSH 临时变量 3]
6		$E1 \rightarrow \sim$	
7		$E2 \rightarrow E4 E3$	推导非终结符 E4 推导非终结符 E3
8		$E3 \rightarrow = E4 E3$	推导非终结符 E4 推导非终结符 E3 [POP 临时变量 1] [POP 临时变量 2] [MOVE 临时变量 2 临时变量 1] [PUSH 临时变量 2]
9		$E3 \rightarrow \sim$	
10		$E4 \rightarrow E5 E6$	推导非终结符 E5 推导非终结符 E6

[0084]

11		E5->T E1	推导非终结符 T 推导非终结符 E1
12		E6->w3 E5 E6	推导非终结符 T 推导非终结符 E1 [POP 临时变量 1] [POP 临时变量 2] [FINGER W3 w3 临时变量 2 临时变量 1 临时变量 3] [PUSH 临时变量 3]
13		E6->~	空动作
14		F->i	推导非终结符 i
15		F->( E2 )	推导非终结符 E2
16		T->F T1	推导非终结符 F 推导非终结符 T1
17		T1->w2 F T1	推导非终结符 T 推导非终结符 E1 [POP 临时变量 1] [POP 临时变量 2] [FINGER W1 w1 临时变量 2 临时变量 1 临时变量 3] [PUSH 临时变量 3]
18		T1->~	
19		valuelist->E2 , valuelist	推导非终结符 E2 推导非终结符 valuelist
20		valuelist->~	
21		i->c	获得立即数 c 的类型 type_c [CMOVET 临时变量 1 c type_c] [PUSH 临时变量 1]
22		i->name A	[VMOVETBEGIN 临时变量 1 name] [PUSH 临时变量 1] [VMOVETEND] 推导非终结符 A
23	流程控制	Z->function type name ( declarelist ) { E } Z	[ADDFUNCTION name] [TYPELISTBEGIN] 推导非终结符 type [TYPELISTEND] [IN] [DECLEARLISTBEGIN] 推导非终结符 declarelist [DECLEARLISTEND] 推导非终结符 E [OUT] [ADDFUNCTION_OVER] 推导非终结符 Z
24		Z->#	
25		Z->type name ; Z	[ADDNAME name] [TYPELISTBEGIN] 推导非终结符 name [TYPELISTEND] 推导非终结符 Z
26		Z->struct name { numberlist } ; Z	[ADDSTRUCT name] [IN]

		[DECLEARLISTBEGIN] 推导非终结符 declarelist [DECLEARLISTEND] 推导非终结符 E [OUT] [ADDSTRUCT_OVER] 推导非终结符 Z
27	Z->~	
28	E->{ E } E	[IN] 推导第一个非终结符 E [OUT] 推导第二个非终结符 E
29	E->sen E	推导非终结符 sen 推导非终结符 E
30	E->~	
31	declarelist->type name ; ddeclarelist	推导非终结符 declarelist [TYPELISTBEGIN] [ARGNAME name] 推导非终结符 type [GETVALUE]
32	declarelist->~	
33	numberlist->type name ; numberlist	推导非终结符 numberlist [TYPELISTBEGIN] [NUMBER_NAME name] 推导非终结符 type [TYPELISTEND]
34	numberlist->~	
35	sen->do { E } while ( E2 ) ;	[SETFLAG 临时变量 1] [IN] 推导非终结符 E [OUT] 推导非终结符 E2 [JUMP]
36	sen->type name ;	[ADDNAME name] [TYPELISTBEGIN] 推导非终结符 type [TYPELISTEND]
37	sen->E2 ;	推导非终结符 E2 [POP]
38	sen->flag name ;	[SETFLAG name]
39	sen->jump name ( E2 ) ;	推导非终结符 E2 [JUMP name]
40	sen->return E2 ;	推导非终结符 E2 [OUT] [POPIP]
41	sen->if ( E2 ) { E } ELSE	推导第一个非终结符 E2 [JUMP1 临时变量 1] [IN] 推导第二个非终结符 E2 [OUT] [GOTO 临时变量 2] [SETFLAG 临时变量 1]

[0085]

[0086]			推导非终结符 ELSE [SETFLAG 临时变量 2]
	42	ELSE->else { E }	[IN] 推导非终结符 E2 [OUT]
	43	ELSE->~	
	44	type->[ E2 ] type	推导非终结符 E2 [TYPEARRAY] 推导非终结符 type
	45	type->typename	[TYPENAME typename]
	46	type->struct name	[SETSTRUCT name]

[0087] 通过语法、语义分析产生编译后的代码传递给信号执行器。

[0088] (2) 信号执行器进一步包括：

[0089] 流程控制器：用于接受用户指令转换器发来的用户指令，将指令按执行顺序发送给信号处理器。如流程控制器接受来自用户指令转换器的三个按顺序排列的指令片段为 PUSH1、EXEF2、POP，此时 2 中所保存的为操作钢厂的标志 &SETHCLENGTH&1 为需要设置的跨距长度，则流程控制器就会将 1 中的数据和执行操作 2 的命令发送给信号处理器，由信号处理器操作设备监控器修改行车的跨距，完成用户的一次操作。

[0090] 下面给出流程控制器所能识别的信号表，如表 4 所示：

[0091] 表 4 为流程控制器所能识别的信号表

识别代码	功能说明	参数说明	执行动作
ADDFUNCTION	函数开始的标志	参数 1：函数名	*根据参数 1 所指定的函数名记录下函数的出现位置，填写函数地址表供其它命令查阅。
ARGNAME	函数参数声明的标志	参数 1：变量名	1. 建立名为参数 1 的变量 2. 变量存入当前变量空间的 用户定义变量区。 3. 将此变量压入全局栈中。 4. 指令指针 IP 下移。
TYPEARRAY	拓展所声明的类型为数组类型		1. 全局栈弹出变量，将此变量的类型作为数组的长度。 2. 全局栈弹出变量，对此变量进行数组拓展。 3. 指令指针 IP 下移。
TYPELISTBEGIN	类型说明开始标志		1. 指令指针 IP 下移。
TYPELISTEND	类型说明结束标志		1. 指令指针 IP 下移。
TYPENAME	修改栈顶变量类型	参数 1：类型码	1. 将全局栈栈顶变量的类型修改为参数 1 所指定的类型。 2. 指令指针 IP 下移。
IN	在当前变量		1. 创建新的变量空间

[0093]

	空间的基础上开辟新的变量空间		<ol style="list-style-type: none"> <li>2. 变量空间作为当前空间的孩子。</li> <li>3. 进入所创建的变量空间。</li> <li>4. 指令指针 IP 下移。</li> </ol>
OUT	当前变量空间访问完毕, 返回上一级变量空间		<ol style="list-style-type: none"> <li>1. 销毁当前变量空间中的常数变量空间与临时变量空间。</li> <li>2. 返回父空间。</li> <li>3. 指令指针 IP 下移。</li> </ol>
CMOVET	对常量建立临时变量的索引	参数 1: 临时变量名称 参数 2: 常量的值 参数 3: 常量的类型	<ol style="list-style-type: none"> <li>1. 填写临时变量映射表, 建立映射关系。</li> <li>2. 对参数 2、3 确定的常量建立类型。</li> <li>3. 将常量压入当前的变量空间的常量空间中。</li> <li>4. 指令指针 IP 下移。</li> </ol>
PUSH	将临时变量所指定的变量压入全局栈	参数 1: 待压入临时变量的名称	<ol style="list-style-type: none"> <li>1. 根据参数 1 搜索变量空间, 定位变量。</li> <li>2. 变量地址压栈。</li> <li>3. 指令指针下移。</li> </ol>
POP	全局栈弹出栈顶元素, 并将此与参数所指定的临时变量建立映射关系	参数 1: 将要与弹出变量建立映射关系的临时变量名称	<ol style="list-style-type: none"> <li>1. 全局栈弹出元素。</li> <li>2. 填写临时变量映射表, 将弹出的元素与参数 1 确定的变量名建立映射关系。</li> <li>3. 指令指针 IP 下移。</li> </ol>
VMOVETBEGIN	对变量建立临时变量的索引	参数 1: 临时变量名 参数 2: 变量名	<ol style="list-style-type: none"> <li>1. 根据参数 1 与参数 2 填写临时变量映射表。</li> <li>2. 指令指针 IP 下移。</li> </ol>
NUMBER	将全局栈栈顶的变量变为指令所指定的名称	参数 1: 需从结合类变量中取得的成员名	<ol style="list-style-type: none"> <li>1. 将全局栈栈顶的变量变为参数 1 所指定的名称。</li> <li>2. 指令指针 IP 下移。</li> </ol>
VMOVETEND	变量索引建立完毕		<ol style="list-style-type: none"> <li>1. 指令指针 IP 下移。</li> </ol>
PUSHIP	保存当前的指令指针所在的位置		<ol style="list-style-type: none"> <li>1. Push(IP+2);</li> <li>2. 指令指针 IP 下移。</li> </ol>
EXEF	执行函数		<ol style="list-style-type: none"> <li>1. 通过 ADDFUNCTION 建立的函数地址表找到函数的入口位置, 并跳转。</li> </ol>
POPIP	恢复所保存的指令指针位置。		<ol style="list-style-type: none"> <li>1. IP=POP ()</li> </ol>
MOVE	赋值操作	参数 1: 被赋值变量。 参数 2: 赋值变量。	<ol style="list-style-type: none"> <li>1. 查找变量空间, 得到两个变量实例。</li> <li>2. 进行赋值操作。</li> <li>3. 指令指针 IP 下移。</li> </ol>

DECLLEARLISTBEGIN	声明开始		1. 指令指针 IP 下移。
DECLLEARLISTEND	声明结束		1. 指令指针 IP 下移。
ADDNAME	变量声明的标志	参数 1: 变量名	1. 建立名为参数 1 的变量 2. 变量存入当前变量空间的 用户定义变量区。 3. 将此变量压入全局栈中。 4. 指令指针 IP 下移。
FINGER_W2	乘、除、取 余运算标志	参数 1: 运 算类型 参数 2&参 数 3, 参与 运算的量。 参数 4: 结 果保存的变 量名称	1. 根据参数 2 与参数 3 指明 的临时变量, 查变量空间 找到临时变量所指的变 量。 2. 根据参数 1 所指的类型进 行运算, 并根据参数 4 所 指出的名字在变量空间 中的临时变量区建立变 量。 3. 指令指针 IP 下移
FINGER_W3	逻辑运算符 运算标志 (=、>、 <=、>、<)	参数 1: 运 算类型 参数 2&参 数 3, 参与 运算的量。 参数 4: 结 果保存的变 量名称	1. 根据参数 2 与参数 3 指明 的临时变量, 查变量空间 找到临时变量所指的变 量。 2. 根据参数 1 所指的类型进 行运算, 并根据参数 4 所 指出的名字在变量空间 中的临时变量区建立变 量。 3. 指令指针 IP 下移
FINGER_W1	加减运算符 运算标志 (+, -)	参数 1: 运 算类型 参数 2&参 数 3, 参与 运算的量。 参数 4: 结 果保存的变 量名称	1. 根据参数 2 与参数 3 指明 的临时变量, 查变量空间 找到临时变量所指的变 量。 2. 根据参数 1 所指的类型进 行运算, 并根据参数 4 所 指出的名字在变量空间 中的临时变量区建立变 量。 3. 指令指针 IP 下移
JUMP	真跳转指令	参数 1: 跳 转的标志名	1. 从全局栈中弹出栈顶元 素。 2. 若此元素大于 0, 则跳转 至参数 1 所代表的标志所 指出的中间代码, 否则指 令指针下移。
JUMPI	假跳转指令	参数 1: 跳 转的标志名	1. 从全局栈中弹出栈顶元 素。 2. 若此元素不大于 0, 则跳

[0094]



[0095]

			转至参数 1 所代表的标志所指出的中间代码，否则指令指针下移。
GOTO	无条件跳转指令	参数 1: 跳转的标志名	1. 跳转至参数 1 所代表的标志所指出的中间代码
GETVALUE	参数赋值指令		1. 从栈中弹出待赋值参数的地址。 2. 从栈中弹出传入的参数地址。 3. 将传入参数的值赋予待赋值参数。 4. 指令指针 IP 下移。
ARRAY	取数组成员指令	参数 1: 索引号	1. 将全局栈栈顶的变量地址更换为对其按照参数 1 进行索引所得到的成员地址。 2. 指令指针 IP 下移。
NUMBER_NAME	对一个变量建立变量成员，构建集合类变量	参数 1: 集合类变量成员的名称	1. 将全局栈栈顶的变量增加一个名为参数 1 的孩子变量。 2. 新建的孩子变量压入栈中。 3. 指令指针 IP 下移。
SET_STRUCT	将全局栈栈顶的元素拓展为集合类类型	参数 1: 将要拓展的集合类变量的名称。	1. 按照参数 1 将栈顶元素的所有叶子拓展成参数 1 指定的类型。 2. 指令指针 IP 下移。

[0096] 信号处理器：(1) 用于接受流程控制器发来的用户指令，如根据 SETHCPOS 和 GETHCPOS 指在设备监控器中设置或取得行车的位置。(2) 用于接受钢厂的实际信号（如暂停生产信号），按照存储在信号处理器内的、用户自定义的处理该信号的方法（如本实施方式中定义的若实际工厂停止生产，则数字钢厂的仿真生产过程也停止这一方法）来取消计时器对重绘模块所提供的时间参数的变化 (3) 用于根据设备监控器、工厂管理器的输出（如数字钢厂中多行车的位置信息），与设置在信号处理器内部的、由用户自定义的报警条件（如本实施方式中设置的两行车之间的距离不能太近的这一约束）进行比较，若满足，则报警。

[0097] (3) 工厂模块进一步包括：

[0098] 设备监控器：用于根据接收到的用户指令或工厂内设备传递的实际信号，调用设备逻辑控制器输出的经拆分后的设备模型及设备动作类型，使工厂模块内的设备增加一种动作、告知用户设备当前正在进行的动作、告知用户行车所处的位置、告知工厂模块行车的跨距、设置行车设备的跨距；

[0099] 工厂管理器：(1) 用于管理工厂的设备位置信息（如 20、25、0 表示设备在数字钢厂中的坐标为 20、25、0）、设备的旋转角度信息（如 90 表示设备在 XY 平面旋转 90 度）、设备

的名称 (BOF1, BOF2 表示两个不同名称的转炉设备) 和行车设备的跨距信息 (如 30 表示行车在三维空间中的跨距是 30 个单位长度) 这些数据可以以设备名称作为主键以一个表的形式保存在工厂管理器之中, 对相关数据的查询和修改都从这个表入手; (2) 用于根据接收到的用户指令或工厂内设备传递的实际信号 (如某设备开始工作、某设备停止工作、某设备发生故障等), 控制计时器停止计时 (如遇到整个工厂停止工作的信号) 或者继续运行 (如遇到整个工厂开始工作的信号) 或者复位或者置数; (3) 用于根据接收到的用户指令或工厂内设备传递的实际信号 (如在新的一轮生产计划中有新设备加入), 在工厂模块增加一种设备 (如 LF 精炼炉, RH 精炼炉, 转炉, 行车, 回转台中的一种); (4) 用于根据接收到的用户指令或钢厂内设备传递的实际信号 (如钢厂中有一台设备不参与此轮生产), 在工厂模块删除一个设备 (如为 LF 精炼炉, RH 精炼炉, 转炉, 行车, 回转台中的一种); (5) 用于根据接收到的用户指令或工厂内设备传递的实际信号, 在工厂模块中修改设备的位置; (6) 用于根据接收到的用户指令或工厂内设备传递的实际信号, 告知用户设备坐标和旋转角度 (如行车设备, 获得的坐标为当前行车跨所处的位置); 计时器: 用于根据工厂管理器的输出 (停止计时、继续开始计时、复位、置位), 向绘制模块提供时间参数 (1000 表示从数字钢厂开始运行开始计算已经过去 1000 秒)。与计时器有关的操作为: 计时器停止计时, 计时器继续运行, 计时器复位, 计时器置数, 计时器停止计时即暂停数字工厂中的时间流逝, 使数字工厂停止运作, 计时器继续运行即撤销计时器停止计时的操作, 计时器复位即计时器所提供的时间从零开始, 计时器置数即用户提供一个具体的时间点, 将计时器提供的时间与用户提供的时间同步。

[0100] 对设备监控器、工厂管理器和计时器三者所提供的操作总结如下:

[0101] 表 5 设备监控器和工厂管理器所提供的操作集合

[0102]

操作名称	含义	参数
SETHCLENGTH	设置行车的跨距	2 个, 字符型表示设备名称, 浮点型表示长度
SETHCPOS	设置行车当前位置	两个个, 字符型表示设备名称, 二维坐标类型表示位置
GETHCLENGTH	得到行车的跨距	
GETHCPOS	得到行车当前位置	
ADDACTION	增加动作节点	开始时间, 结束时间, 动作编码, 额外参数
PAUSE	计时器停止计时	
CONTINUE	计时器继续运行	
RESET	计时器复位	1 个整形, 表秒数
SETTIME	计时器置数	
ADDEQUIPMENT	增加设备	1 个, 设备名称
DELETEEQUIPMENT	删除设备	1 个, 设备名称

[0103]

GETEQUIPMENTPOS	获得设备坐标	1 个, 设备名称
SETEQUIPMENTPOS	设置设备坐标	2 个, 设备名称, 坐标
SENDMESSAGE	向外发送信息	2 个, 信道名称, 信息内容
GETEQUIPMENTR	获得设备旋转角度	2 个, 设备名称, 角度
SETEQUIPMENTR	设置设备旋转角度	2 个, 设备名称, 坐标

[0104] 绘制模块：用于根据计时器输出的时间参数，绘制钢厂在当前时间参数下的运行画面；对于任一设备的任一视角，根据设备的运动幅度，采用帧率优化算法来判断当前帧是否需要绘制；

[0105] 若有必要则重绘，若没有必要，则跳过此次重绘；

[0106] 所述的帧率优化算法过程如下：

[0107] 使数字工厂界面满帧运行一段时间，在其间对工厂中所有的设备的最大运动速度进行采样得到速度集合  $s$ ；取集合  $s$  中元素的中位数  $v\_middle$ ，计算帧率调整函数  $f(v)$ ，公式如下：

$$[0108] \quad f(v) = \begin{cases} \frac{v}{v\_middle} & v < v\_middle \\ 1 & v \geq v\_middle \end{cases} \quad (1)$$

[0109] 式中， $v$  为当前重绘下设备的最大速度；

[0110] 把一秒中的用户自定义的最大重绘次数等分成  $N$  组；

[0111] 统计摄像机视角内的设备的各个组件的运动速度，得出当前重绘下设备的最大运动速度  $v$ ；

[0112] 将得到的最大运动速度  $v$  带入公式 (1)，来判断当前帧是否需要绘制，公式如下：

[0113] 若满足如下公式则重绘：

$$[0114] \quad (T_{current} - T_{begin}) \bmod \left(\frac{T}{N}\right) < f(v) \times \left(\frac{T}{N}\right) \quad (2)$$

[0115] 式 +， $T_{current}$  为当前时间， $T_{begin}$  为标记时间段的开始时间， $T$  为标记时间段的长度， $N$  为时间段的分组数；

[0116] 若不满足公式 (2) 则跳过本次重绘。

[0117]  $T$  与  $N$  是人为限定的，如在最大帧率为 15 的情况下，可以设  $N=3$ ，则  $T = 0.3333s$ ，在一个单位的  $T$  内最多可以出现 5 次重绘，经过命中检测，每组的重绘次数可能会小于 5 次，达到了减小重绘次数的目的。本方法在存在固定相机用以监视单个设备运作的情况下能起到很好的降低帧数的作用。

[0118] (4) 设备逻辑控制器还进一步包括：

[0119] 模型管理器：作用是管理数字化工厂设备的模型信息。它有两方面的功能：用于建立实体设备，包括双向精炼炉、精炼炉、转炉、行车、RH 精炼炉、回转台的 3D 模型；对上述的 3D 模型进行拆解并保存。

[0120] 本实施方式采用 3DMAX 软件建立双向精炼炉 3D 模型、精炼炉 3D 模型、转炉 3D 模型、行车 3D 模型、RH 精炼炉 3D 模型和回转台 3D 模型，6 种设备的 3D 模型建立完毕后，需要

对模型进行拆解导出,方便利用分离的模型组织动画,进行拆解时,应当遵循以下规则:

[0121] (1) 固定不动的部分(如精炼炉的设备支架、转炉的倾动装置及设备支架)需要拆解成为独立一部分;

[0122] (2) 有不同运动方式的组件拆解成不同的模型,如行车的端梁和吊机。

[0123] 将双向精炼炉(如图3所示)和精炼炉(如图5所示)拆分,如图4所示:拆分为炉盖401、精炼钢水罐车402、钢包404、设备支架405、升降装置403;

[0124] 将RH精炼炉(如图10所示)均做如下拆分,如图11所示:炉盖1102、精炼钢水罐车1104、钢包1105、设备支架1103、升降装置1101。

[0125] 将转炉(如图6所示)做如下拆分,如图7所示:拆分为

[0126] 钢包车704、渣罐车703、钢包706、倾动装置及设备支架707、转炉炉体及托圈702、左炉门701和右炉门705。

[0127] 将行车(如图8所示)做如下拆分,如图9所示:拆分为:

[0128] 拆分为行车左端梁901、行车右端梁902、行车主梁903、钢丝绳905、滑轮组907、吊机908、钢包904。

[0129] 将回转台(如图12所示)做如下拆分:拆分为连铸台1301、钢包1302、钢包1303和设备底座1304,如图13所示。

[0130] 动作管理器:用于管理设备的动作信息,通过实现每种设备具体的动作,来使模型管理器中的模型具有意义,在上述模型分割原则的前提下,使用如下的动作分割方法:

[0131] 对于工厂的各个工位均有操作流程并有对应的编码,但该事件编码通常是不易于进行动画演示的,故应进行重新分割并进行细化,以便动画模型的控制和制作。

[0132] 每个工位首先必须分成两个部分:静态部分和动态部分。

[0133] 其中动态部分是指刚体部分。

[0134] 在刚体部分中,对会发生相对位移的不同部分独立分割成一个部件。

[0135] 需要注意的是,在工件运行全过程中出现在该工位的所有形体(有些会中间出现一段时间然后消失)都必须考虑为工件的一个部分。

[0136] 在工位确定被分割成对应部件以后,可以开始进行事件分割。事件分割在系统的方案设计过程中起着非常关键的作用,合理且符合实际的分割会给3D部件控制和仿真计划数据的分析带来很大的便利,故需要按照一定原则进行分析和分割。原则如下:

[0137] (1) 以做刚体运动的部件开始时间和结束时间作为事件的开端和结束;

[0138] (2) 一个事件的时间开区间内不能有任何的实体消失(撤离)或加入;

[0139] (3) 两个或两个以上刚体运动部件区间若重叠,以具有更显著差异的刚体运动为准,将其他刚体运动区间分解在两个事件中;

[0140] (4) 所有以初始静态动画呈现的外部事件一律不额外增加事件,均以统一的空闲态表示;

[0141] (5) 两个工位之间若有工件衔接和联系的,两个工位的对应事件必须在时间上无缝对接。

[0142] (6) 若多个事件是连续的,而且他们运行时长的比例是确定的,即可将他们合并成一个事件。

[0143] 如上所述,对于数字化钢厂,设备所要实现的动作为如表6所示:

[0144] 表 6 为数字化钢厂所要实现的动作表

[0145]

<p>转炉设备加料          开始：静止状态          过程：（1）转炉转动一定角度，转炉门打开；（2）转炉恢复垂直，转炉门关闭。          结束：静止状态</p>
<p>转炉设备冶炼          开始：静止状态          结束：静止状态</p>
<p>转炉设备冶炼          开始：续上          过程：（1）转炉转动，转炉门打开，钢包小车进入；（2）倾倒入钢水，钢包液面抬升；          （3）转炉、转炉门、钢包小车均复位          结束：复位完毕</p>
<p>转炉设备溅渣          开始：续上          过程：火焰效果，期间钢包可能消失（被调走转移到吊机上）          结束：火焰停止</p>
<p>转炉设备排渣          开始：续上          过程：（1）转炉转动，受渣小车进入；（2）倾倒入渣，液面抬升；（3）复位。期间钢包可能消失（被调走转移到吊机上）          结束：复位完毕</p>
<p>LF 或 RH 精炼炉钢包进站          开始：钢包由吊机装入小车          过程：小车驶入精炼炉          结束：小车停止</p>
<p>LF 精炼炉处理          开始：续上          过程：（1）旋转炉盖至钢包上方时（2）炉盖下压，开始处理（3）处理过程无动作（4）处理后按照（1），（2）步骤的相反动作复位。          结束：复位完毕</p>

	<p>RH 精炼炉处理                  开始：续上                  过程：（1）真空槽平移至钢包上方时（2）真空槽下压，开始处理（3）处理过程无动作（4）处理后按照（1），（2）步骤的相反动作复位。                  结束：复位完毕</p>
	<p>LF 或 RH 精炼炉出站                  开始：精炼炉处理结束                  过程：小车驶出                  结束：小车停止</p>
	<p>LF 或 RH 精炼炉等待钢包被调走                  开始：续上                  过程：等待，钢包被调走                  结束：钢包被调走</p>
	<p>空回转台外侧接受首个钢包等待（内侧无钢包）                  开始：钢包吊入回转台外侧                  过程：静止                  结束：回转台开始旋转</p>
	<p>回转台将外侧满钢包转入内侧（内侧无钢包）                  开始：回转台开始旋转（续上）                  过程：回转台旋转                  结束：回转台旋转到位</p>
	<p>一个钢包在回转台内侧浇铸（外侧无钢包）                  开始：回转台旋转到位（续上）                  过程：钢包开始浇铸，液面下降，触发出钢                  结束：浇铸完毕或下一个钢包吊入回转台外侧</p>
[0146]	<p>一个浇铸完毕的空钢包在回转台内侧等待（外侧无钢包）                  开始：一个钢包在内侧浇铸完毕                  过程：等待                  结束：外侧钢包吊入</p>
	<p>一个浇铸完毕的空钢包在回转台由内侧转向外侧（外侧无钢包）                  开始：外侧钢包吊入（续上）                  过程：回转台旋转                  结束：钢包消失，回转台到位</p>
	<p>回转台内侧钢包浇铸完毕，外侧有满钢包                  开始：外侧有钢包吊入                  过程：回转台旋转                  结束：回转台到位，空钢包消失</p>
	<p>回转台内侧钢包正在浇铸，外侧有满钢包                  开始：外侧有钢包吊入                  过程：内侧钢包液面下降                  结束：内侧钢包浇铸完毕</p>
	<p>行车空载运行                  开始：计划规定移动开始                  过程：吊机位置作平移                  结束：计划规定移动结束</p>
	<p>行车吊钢包运行                  开始：计划规定移动开始                  过程：吊机位置作平移，吊有钢包                  结束：计划规定移动结束</p>
	<p>空闲                  开始：静止不动（初始状态）                  过程：静止不动（初始状态）</p>

[0147]

<p>结束：静止不动（初始状态）</p>
----------------------

[0148] 虽然以上描述了本发明的具体实施方式,但是本领域内的熟练的技术人员应当理解,这些仅是举例说明,可以对这些实施方式做出多种变更或修改,而不背离本发明的原理和实质。本发明的范围仅由所附权利要求书限定。

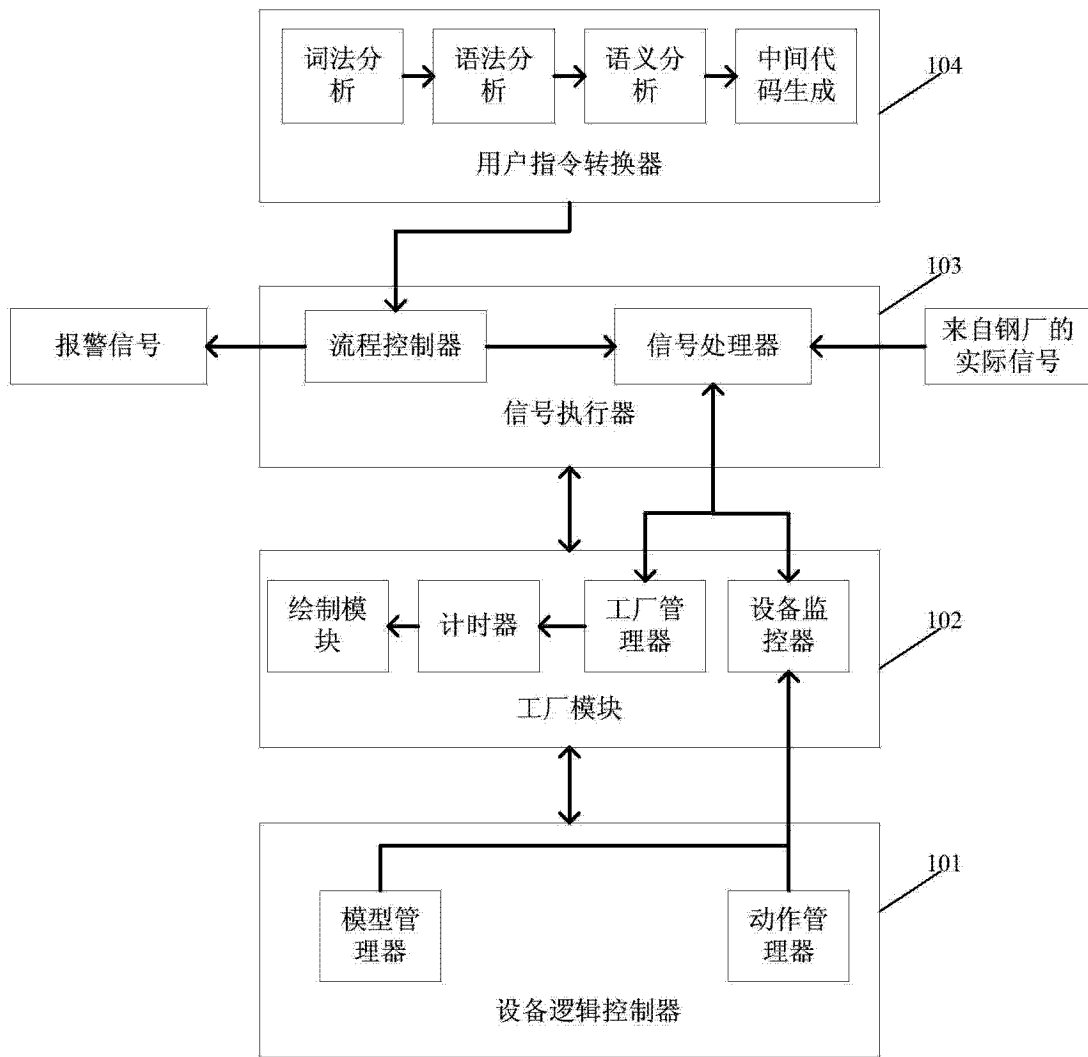


图 1



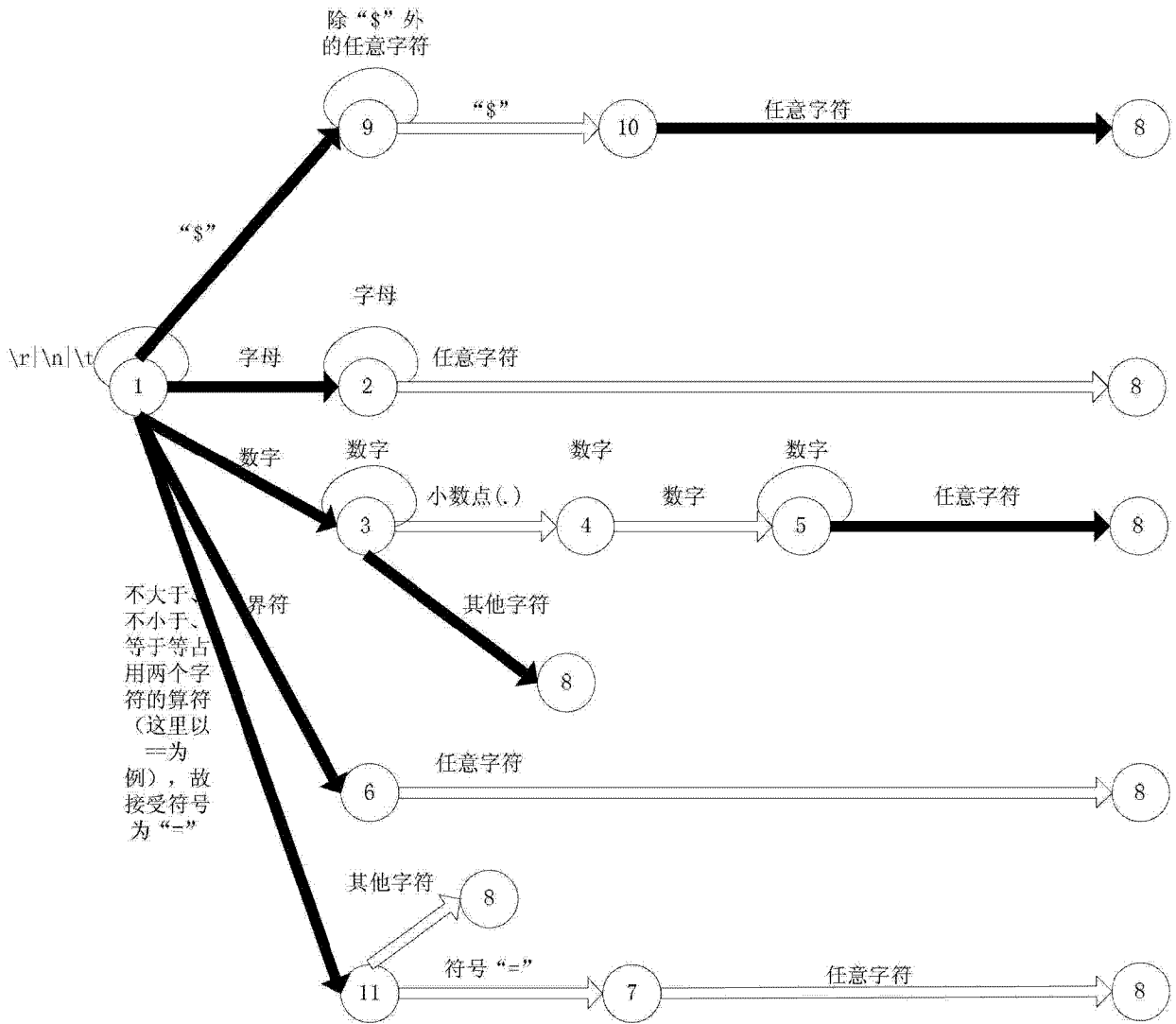


图 2

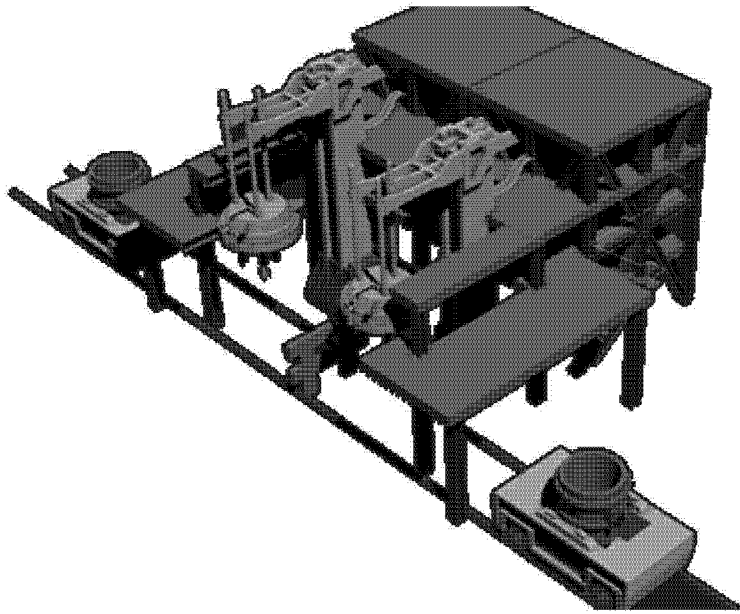


图 3

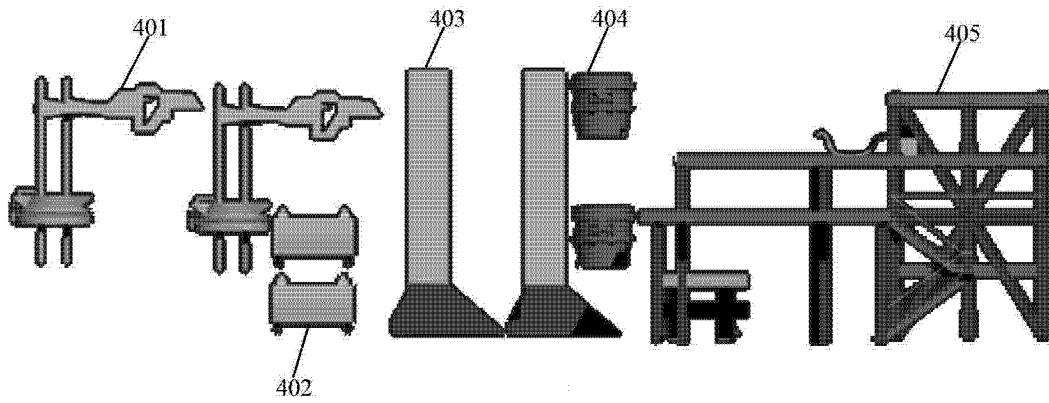


图 4

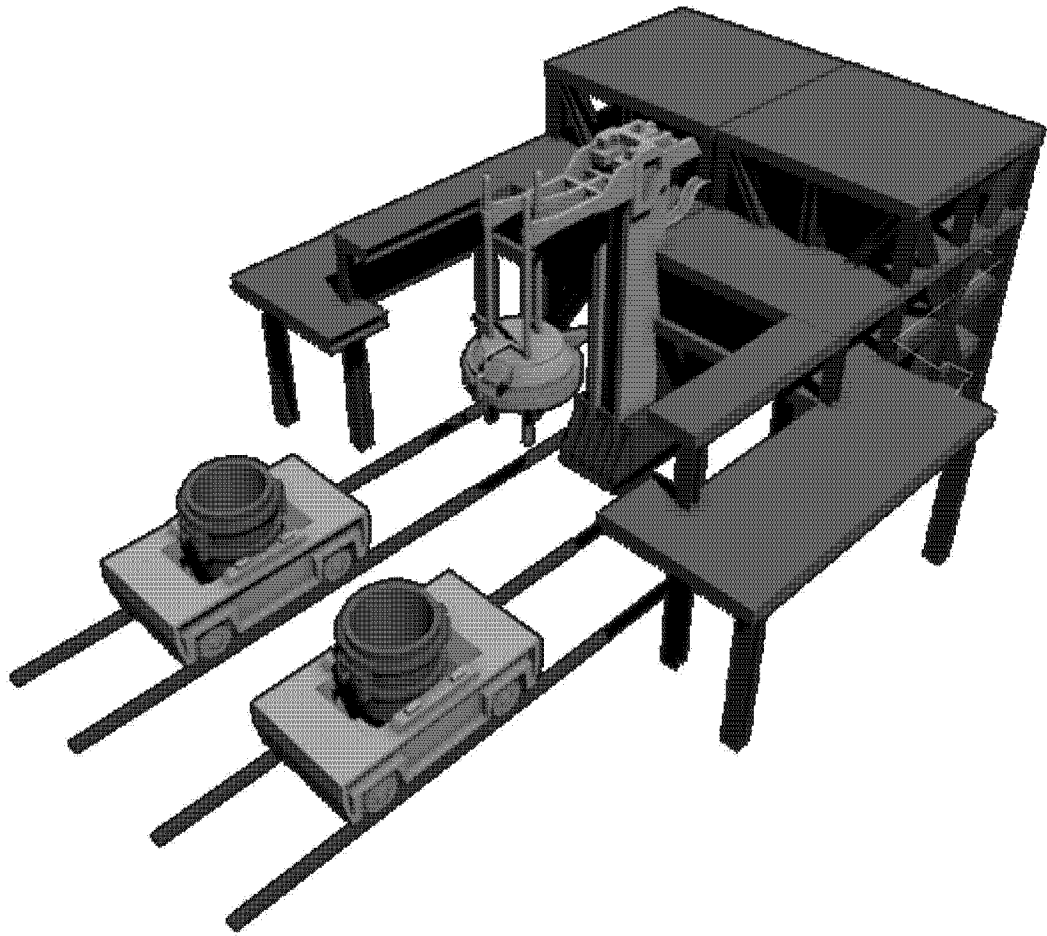


图 5

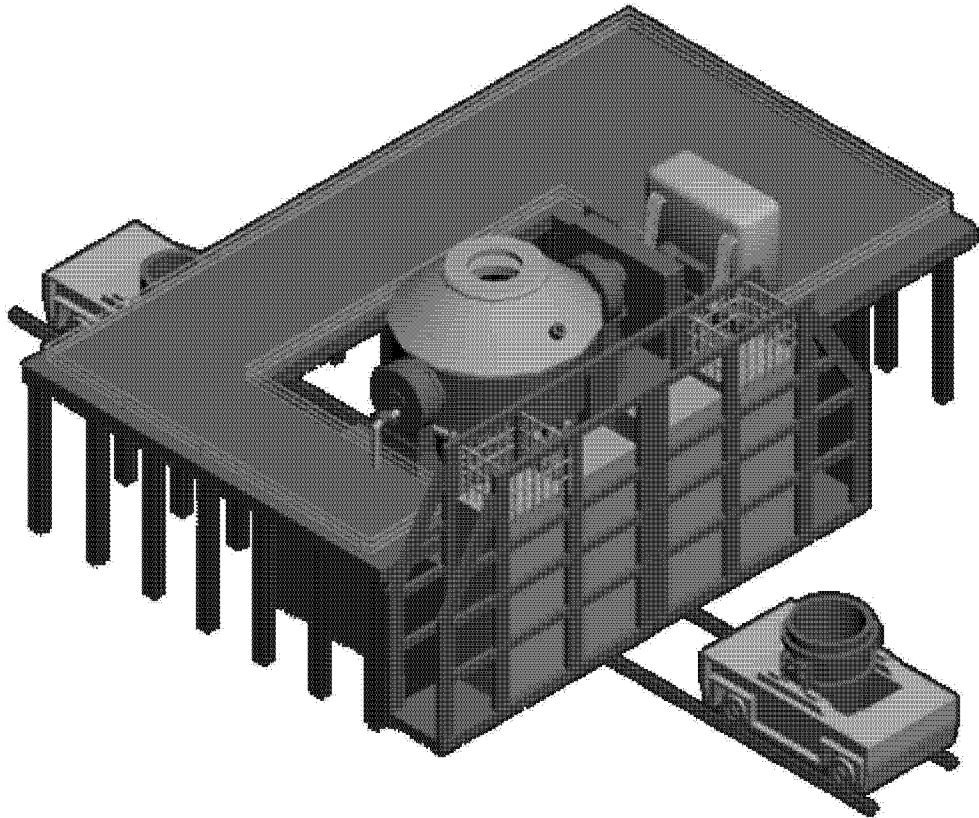


图 6

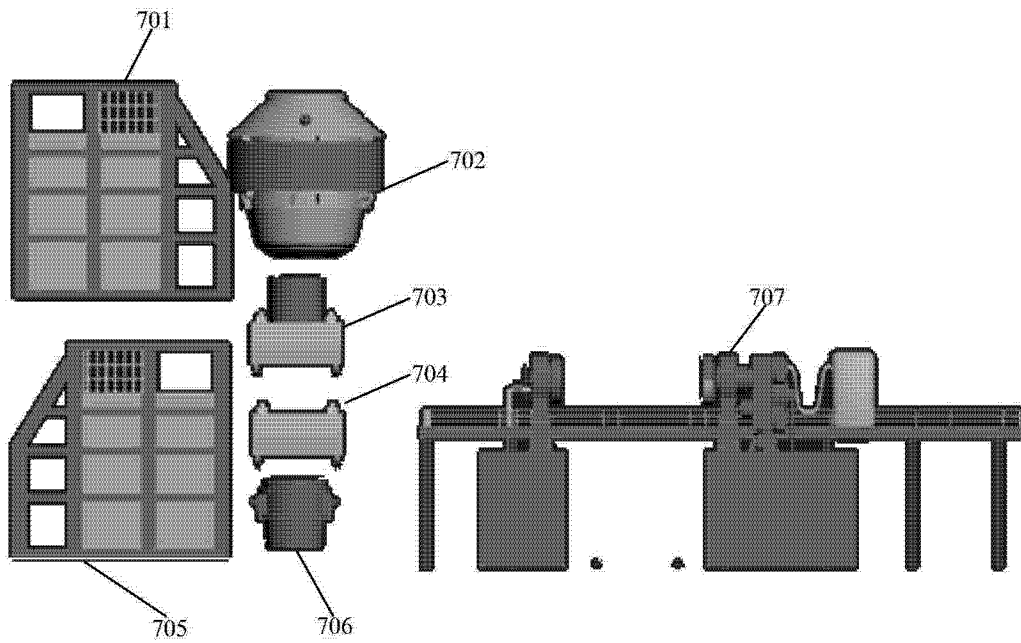


图 7

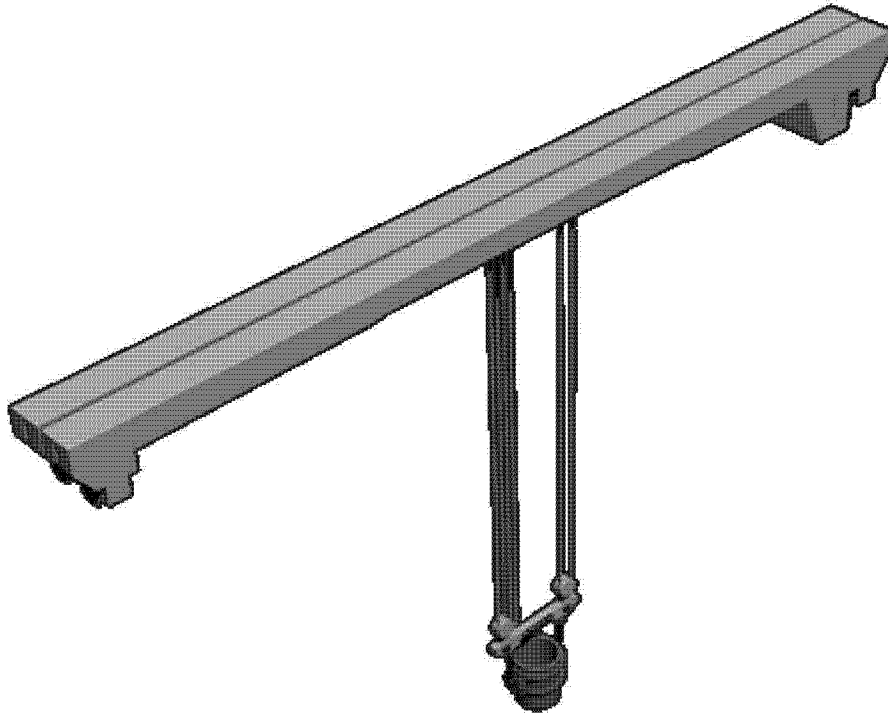


图 8

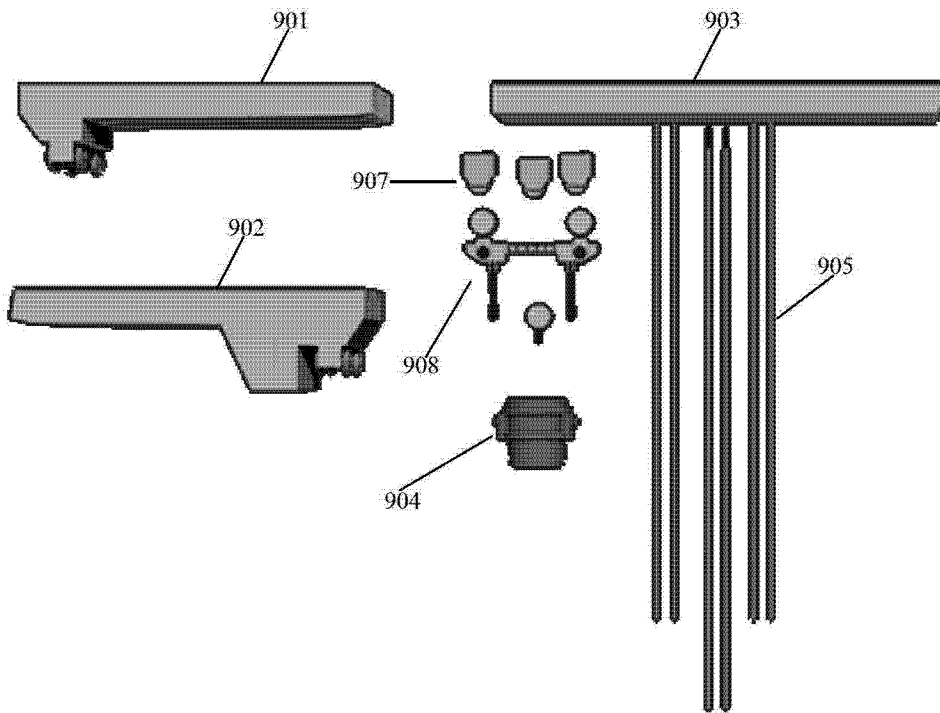


图 9

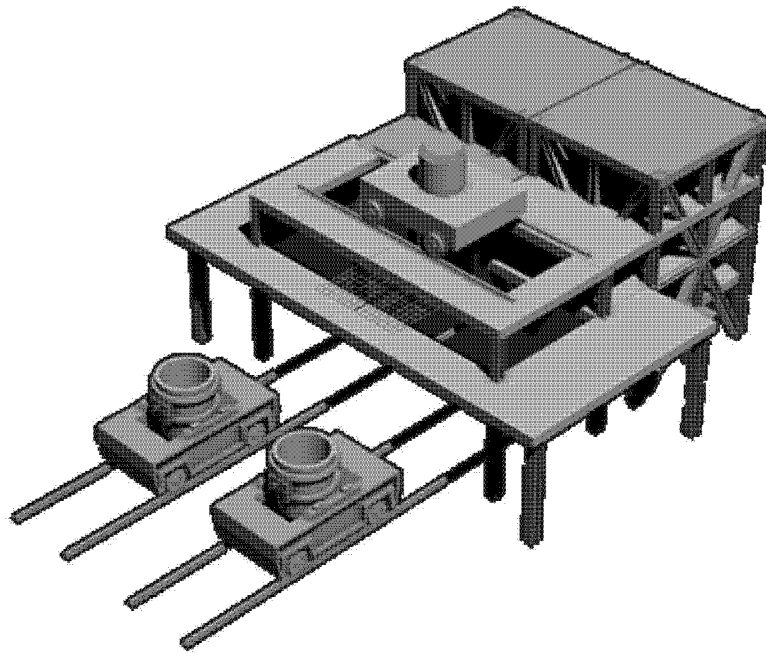


图 10

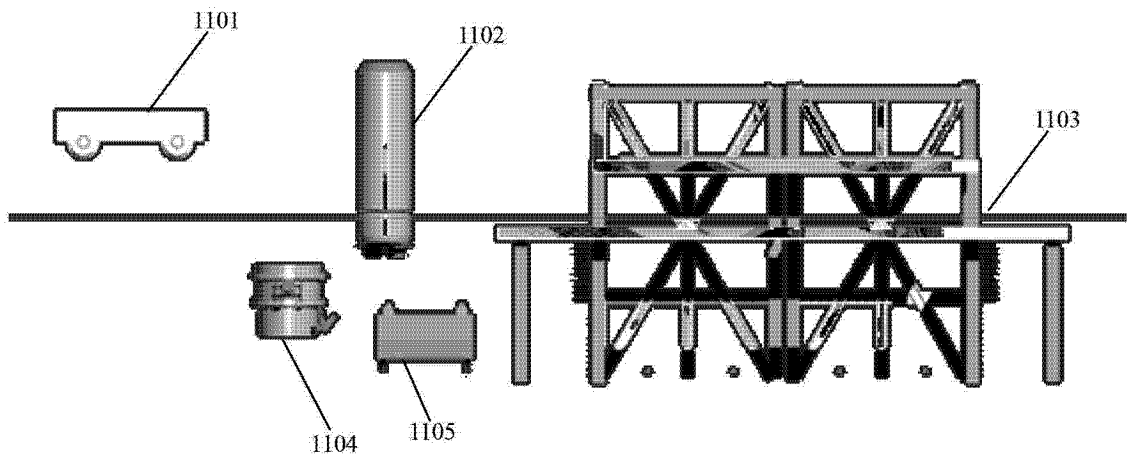


图 11

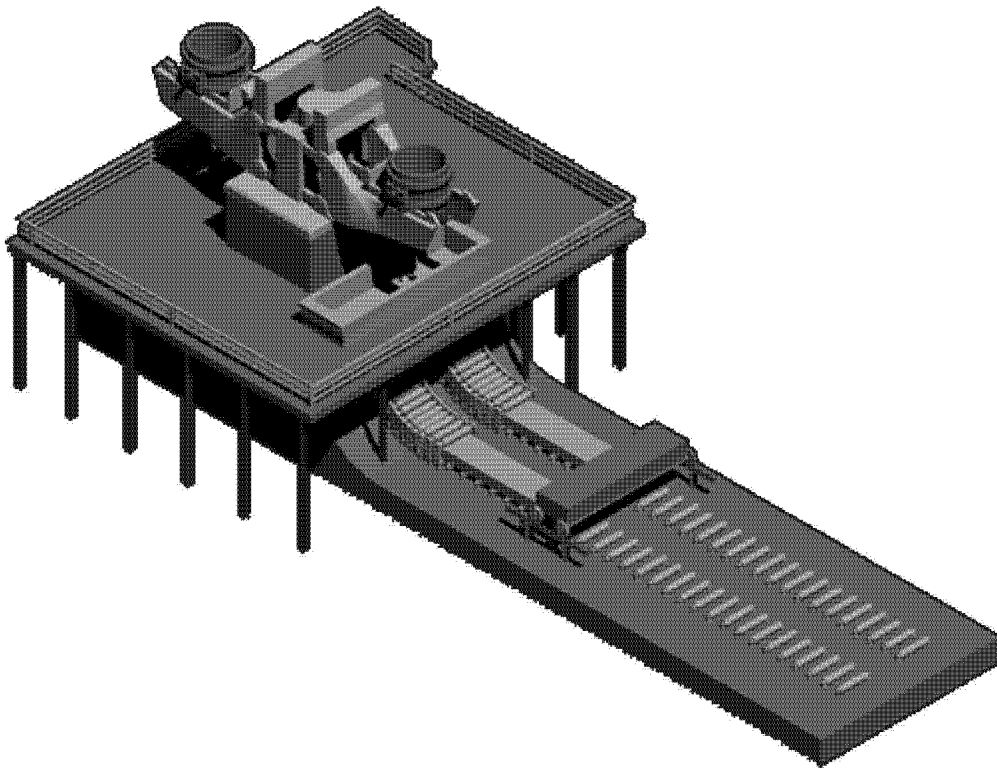


图 12

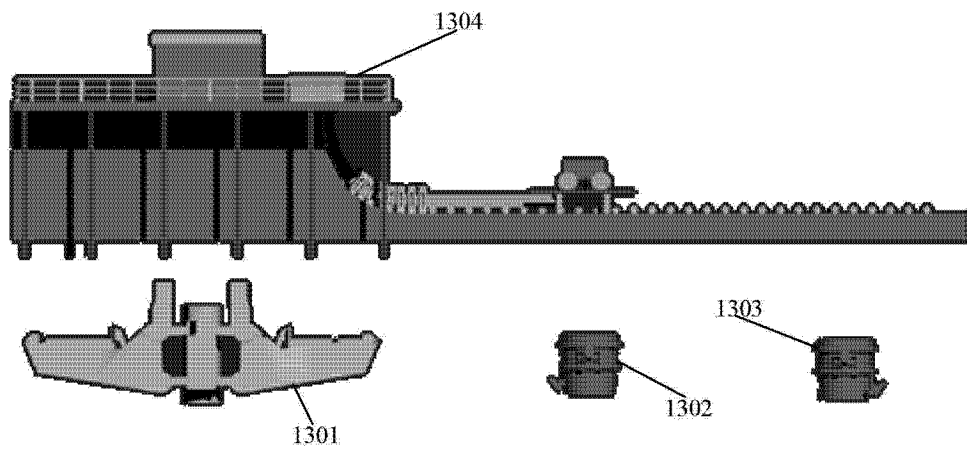


图 13