The Kitting Cell

Introduction

The kitting cell is a sophisticated real-time control application. The kitting cell assembles kits full of parts for use by other assembly cells. It has many devices (robots, cameras, etc.) that must be controlled to perform its kitting application. In the factory, each device has its own controller and sensors, which are native to it. A cell control program (a collection of agents) runs on its own computer. The agents “communicate” with the device controllers using wires and/or proprietary networks.

In this project all the devices are simulated in an animation. The agents will communicate with the animation using an asynchronous messaging interface, the so-called DoXXX( ) API between the csci201 teams and the csci200 teams.

Here is a link to the kitting cell schematic. Your animation will resemble the schematic.

Specification/Requirements

The normative scenario

- initialize the devices/animation;
- get the cell configuration information;
- get the kit configuration information; each kit holds anywhere from 4 to 8 various parts;
- make the gantry requests;
- load the feeders and lanes;
- move part robot to home;
- take pictures when parts are in nests;
- pickup kit from conveyor and put it on kitting stand;
- pick up parts for pass 1;
- put parts in kit;
- pickup parts for pass 2;
- put parts in kit;
- move kit to inspection stand;
- inspect kit put good kit back on conveyor.

Of course, many of the steps above can be done concurrently.

Basic operation

The cell has three logical divisions: kit management, feeder management, and lane/nest management

1. *Kit management:* The kitting stand has 3 kitting positions. Kits are brought to the kit delivery station on a pallet by a conveyor. The kit robot picks up the kit and puts it on one of the two furthest-right positions on the kit stand. When the kit is complete, the kit robot moves it to the furthest-left position for a camera inspection. If successful, the kit robot moves the kit back to the kit delivery station where it exits the cell.

2. Feeder management: Bins holding a single kind of part are dumped into the feeders by the gantry robot. The gantry puts the empty bin in the purge station behind the feeder. The feeders move the parts continuously one at a time onto a vibrating panel. A diverter orients the part toward one of two long lanes. From there the part vibrates down the lane. Parts queue up behind one another until they arrive at the nest at the lane's end. In a purge cycle the feeder drops its rear gate and empties itself into the purge bin. The gantry robot moves the purged bin to a temporary storage location and refills the bin with the required part. The lanes continue to vibrate parts to the nests during this purge cycle. Since one feeder services two lanes, purging and refilling is part of the normative operation.

3. Lane/nest management: Parts vibrate down the lanes into the nests. Parts lineup behind those already in the nest to create a stable state. Depending upon the part size, those nests may hold anywhere from one to ten parts. An overhead camera takes pictures of the nests in order to determine part quality and coordinate information. If satisfactory, the part robot, sliding along a rail over the lanes, uses one of its 4 grippers to pick up the part at the known coordinates. After picking up a load of parts, it puts them in the kit. Generally, two passes are required across the lanes to complete a kit. All eight nests can be individually dumped (parts just fall on the floor) in case no good parts can be located by the camera. The long lanes, which are generally always vibrating, must be stopped before a nest is dumped.

Device Capabilities

The hardware devices have the following capabilities:

l. *Feeder:* On/off switch, part low sensor, feed parts switch, part fed counter, lower/raise rear gate switch, purge bin switch, diverter left/right switch.

2. Long lane: On/off switch, vibration amplitude setting.

3. Nest: Up/down switch.

4. Cameras: Initialize; load configuration information; shoot camera x. (There is a camera over each pair of lanes and one to shoot a kit)

5. *The kit robot:* Has a normal set of robotic capabilities: move to position, pickup kit, put kit.

6. *The part robot:* Has a normal set of robotic capabilities: move to position, pickup part in gripper n, put part using gripper n. The robot hand has 4 grippers. Each gripper is designed to pick up a variety of parts. With its 4 grippers the part robot can hold 4 parts at a time.

The teams will design the DoXXX(…) methods that will engage these capabilities in the animation. The DoXXX(…) routines don't "do" anything. Like all messages, they just set some data in the address space of its device. The device/animation must be prepared to look at that data when appropriate.

Cell Controller

A cell controller full of agents will coordinate all these devices. The csci201 will design the agents. They will use the DoXXX(..) API in their actions.

The Environment

1. Factory Control System (FCS): The FCS sends configuration information to the cell. For the kitting cell, this includes kit configuration information (which parts are to be put into the kit during this run), bin configuration information (where bins are and what parts they contain). Generally, a cell will do its job until alerted by the FCS to do a new configuration (in which case the cell must purge irrelevant parts and load up for the new cycle). The FCS can also tell a cell to pause or purge--each cell must interpret how to do it. You must build a GUI for the FCS.

2. The gantry robot: This is shared among the various cells. The gantry robot moves bins around and can dump them into feeders. It has normal robotic capabilities: move to position, pickup bin, dump bin into feeder, place bin. There are many gantry robots. There is a gantry controller that controls them all. Cell agents make requests to this gantry controller.

3. The kit conveyor system: This is shared among the various cells. The kit conveyor system moves pallets around the cells in the assembly area. Its capabilities: move into cell, lock into position, leave cell. There are many pallet conveyors. There is a conveyor controller that controls them all. Cell agents make requests to this conveyor controller.

System analysis and design issues

The simplicity of the kitting cell's operation is striking. Each aspect is straightforward to describe and easy to understand. However, like most real world control problems, a closer inspection uncovers numerous constraints and boundary conditions. Some of the most serious are as follows.

1. Even though there are numerous cameras, only one can be fired at a time.

2. [v2] The lane cameras shoot from above and can be obstructed by the part robot as it moves from lane to lane. The robot blocks the lane it is on and the lane to its right while picking up a part. Picture-taking is an expensive operation; wasted pictures must be avoided.

3. [v2] The lane cameras shoot two nests with each exposure. Those cameras should wait to shoot until both nests are filled with unanalyzed parts.

4. After a part is picked up, the nest must be given time to stabilize before the next picture is taken. The amount of time depends on the part in the lane.

5. The gantry robot is a scarce resource shared across the entire assembly area. It is impractical (and selfish) to try to keep the lanes as full as possible. Instead, an algorithm must be devised

which asks for gantry service as late as possible, but before a lane runs out of parts. It would be nice if the algorithm were adaptive to the behavior of the gantry [v9].

6. In case of error, restarting the cell from an empty configuration is hopelessly expensive. The cell must be able to recover without needlessly purging the lanes and feeders.

7. Similarly, reconfiguring the cell for different kits could require some lanes and feeders to be purged.

8. The kit and part robots could collide at the two far right kitting stations.

9. The camera could report a part as good, bad, or missing. The part reported as missing could be indicative of a lane being jammed; you could increase the lane's amplitude to try to shake the parts free.

10. If a kit fails the final vision inspection, dump it somewhere.

11. [v3] Add to the cell a rack that holds kits. The idea is to decouple the kitting operation from the kit conveyor system. The rack could hold kits of various dispositions – empty, complete, partially completed (however they got that way), or bad kits (perhaps for later manual inspection). Presumably, the kit robot would know how to deal with them.