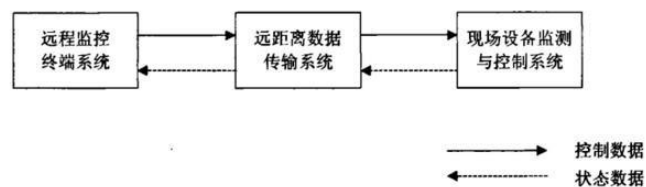


## 《Telerobot control systems》

### — modeling and analysis

We design a simple version of a Telerobot Control (TC) system. TC system is a remotely controlled robot. It consists of two main parts: a sensor glove and the remote RobotArm. Meanwhile, it also include the network connect between those two pars. Sensor glove will make a accurate measurement, then send the sensor data to robot-arm and execute the same action with the sensor glove. TC system is useful in hazardous environments because the operator can be yard away defusing a bomb or handling dangerous chemicals. To be useful, the robot must response to the hand movements of a sensor glove quickly enough to take advantage of the user's inherent motor-control system. If the robot responds too slowly, the user will overreact, creating an unstable control loop. Likewise, the control loop can became unstable if robot is slow to provide tactile or visual feedback.

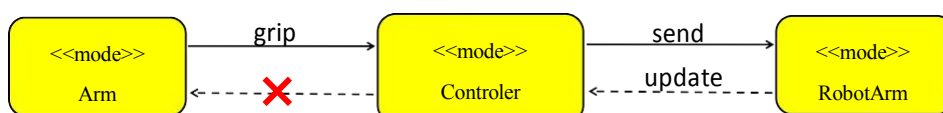


TC system have three parts, which it can be seen as three components.

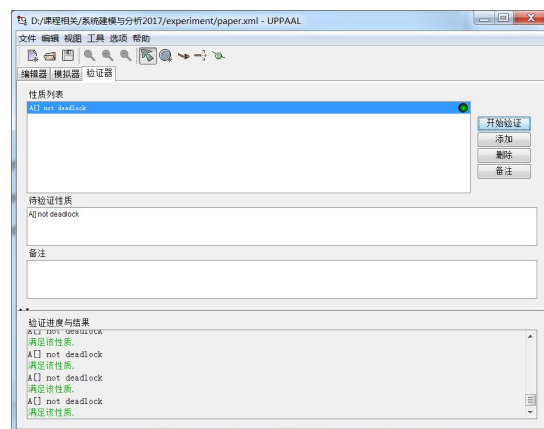
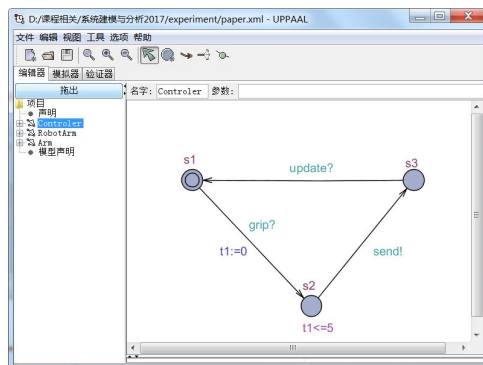
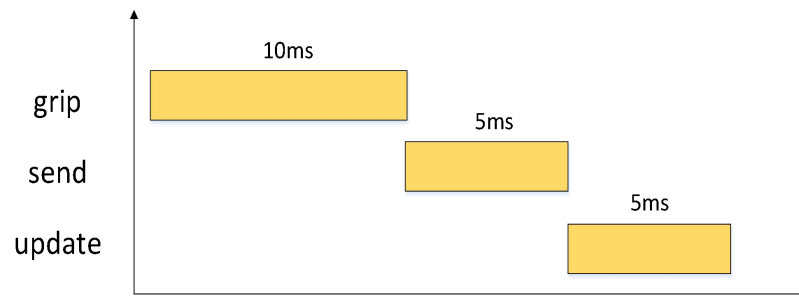
- The *Arm* components have **Grip** action, which get the glove power from the sensor, and send the grip command to the Controller.
- Another component is an station which is named *Controller*, containing the following actions that is triggered for executing the corresponding behaviors: Controller components will accept the grip command from the Arm, then **Send** action will send the control instruction to the remote Robot; and another action is **Update**, which will refresh the display after receive the update command.
- Another component is the remote *Robot*: the component will receive the send command using the network protocol stack; then **Update** action will send the update command back to the Controller.

The timing and functional properties of TeleRobot control system is specified following :

- R1: The **grip** will execute with periodic cycle 10ms (**Arm**);
- R2: **Send** is always followed by **grip** (**Controller**);
- R3: The behavior of **update** is always followed by **send** (**Controller**);
- R4: RobotArm will receive the **send** commands (**RobotArm**);
- R6:**update** is always followed by **send** within 5ms, when RobotArm have received the send command (**RobotArm**);



Time Budgets of TC system :



Config:

```
clock t1,t2,t3;
chan grip, send, update;
```

```
system Controler, RobotArm, Arm;
```

Some verification:

A [ ] not deadlock

Design the component of **RobotArm** and **Arm** ?