Autonomous Robot Systems

Programming

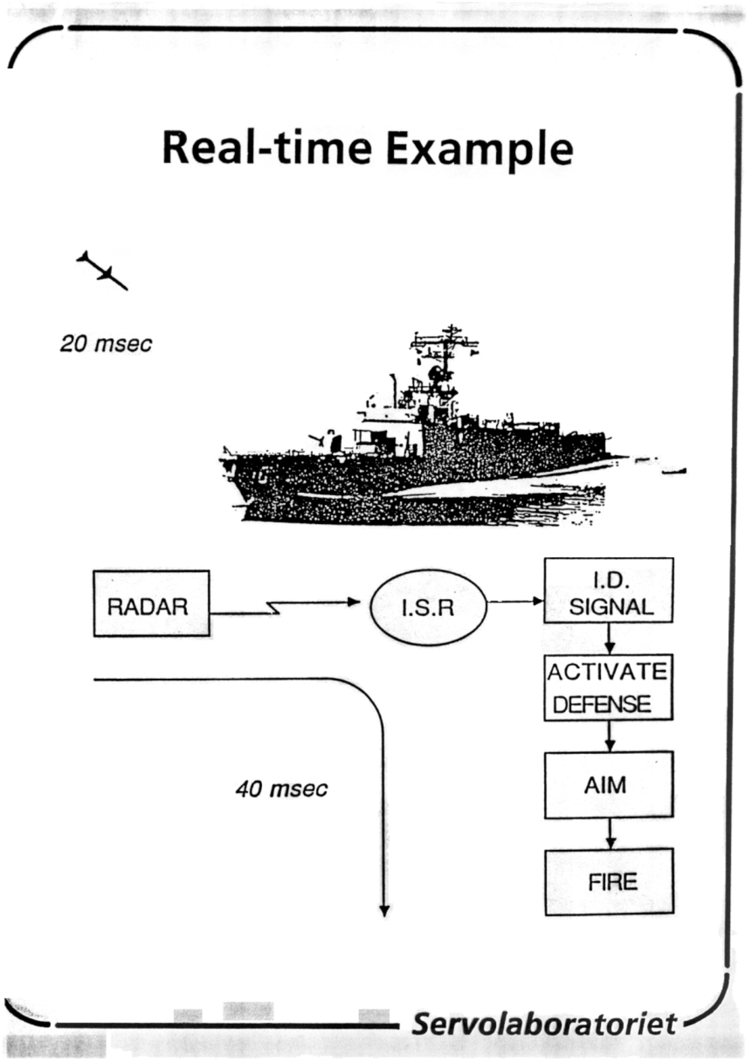
Nils Andersen and

Ole Ravn

Automation and Control

DTU Elektro

Real-time programming



Real-time Example

20 msec

RADAR

I.S.R

I.D. SIGNAL

ACTIVATE DEFENSE

AIM

FIRE

40 msec

Servolaboratoriet

Real-Time programming

- Even the right answer is wrong if it is late.

- Hard realtime is ability to garantee maximum reactiontimes that are sufficient for the controlled system.

Real-Time programming

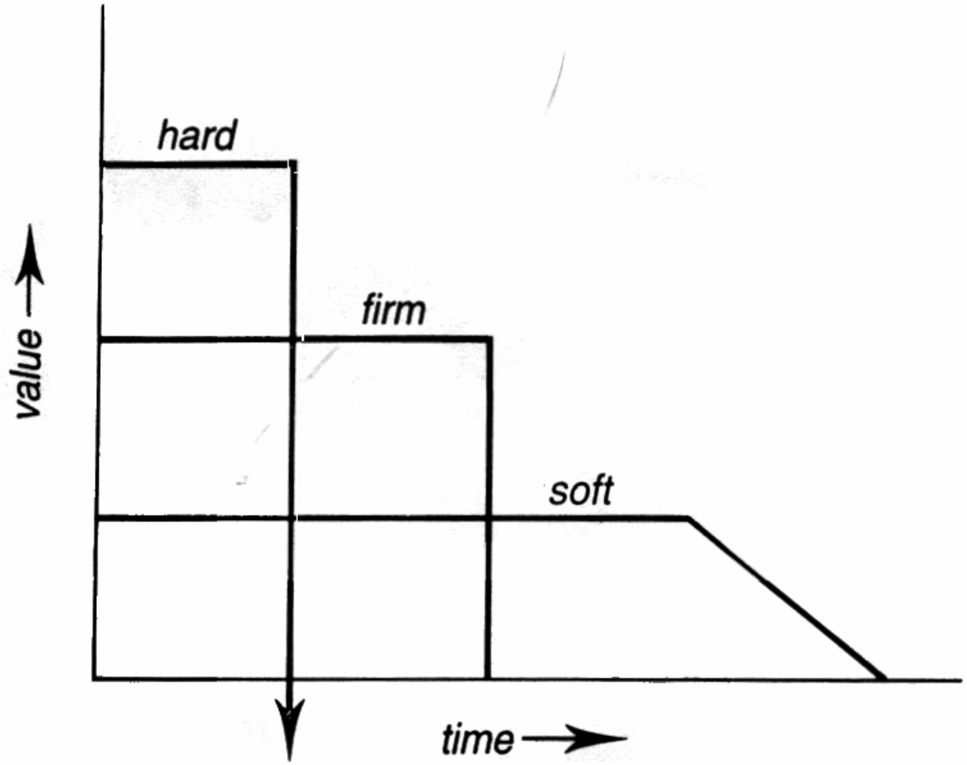
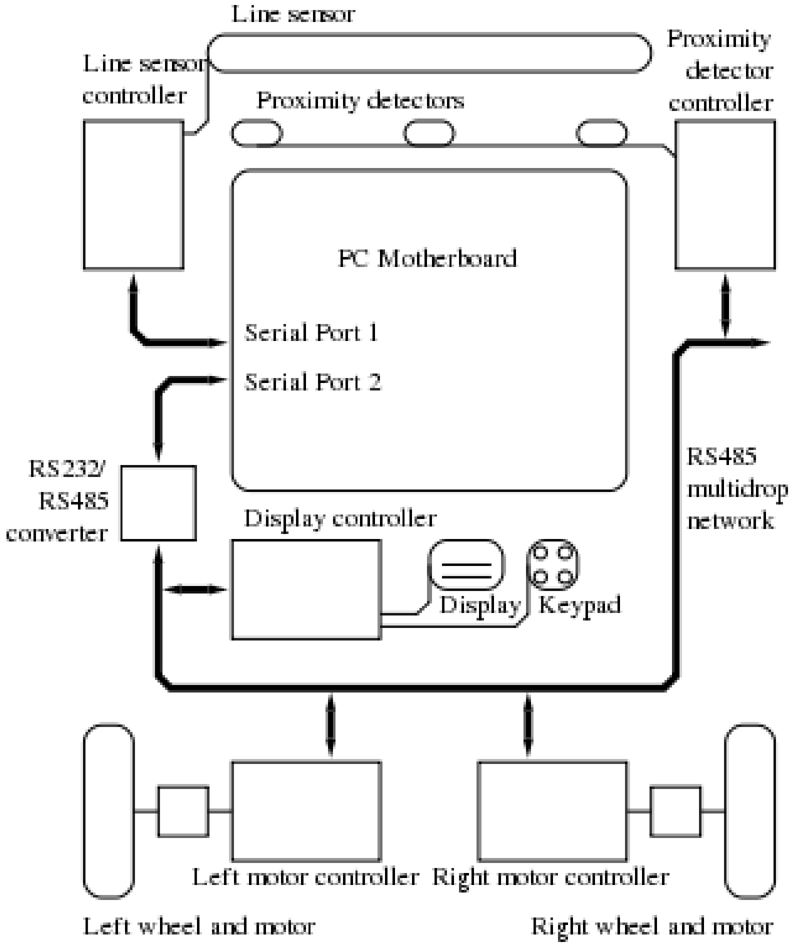
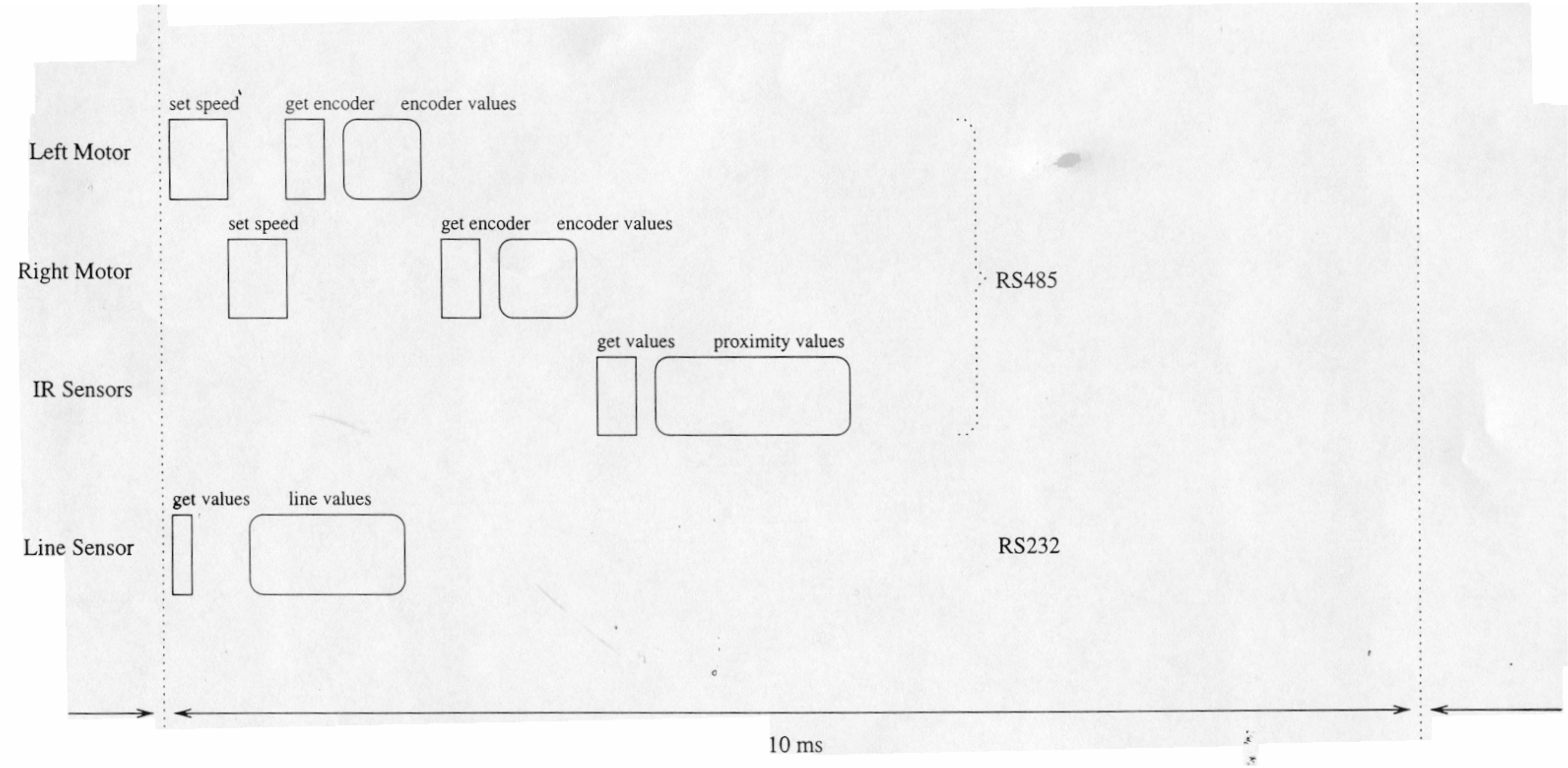


Figure 4.1 Different kinds of value function

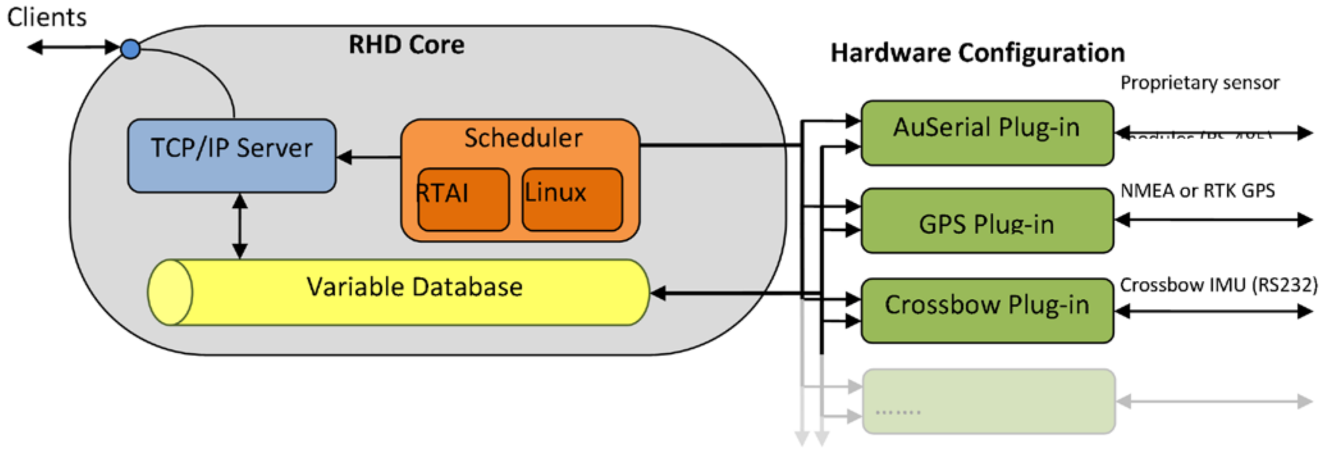
SMR Communication



SMR-RS485-timing



RHD-architechture



RHD functions

```
char    rhdConnect(char, char *, int);
char    rhdDisconnect(void);
char    rhdSync(void);
symTableElement*  getSymbolTable(char);
int     getSymbolTableSize(char);
symTableElement *    getinputref (const char
            *sym_name, symTableElement * tab)
```

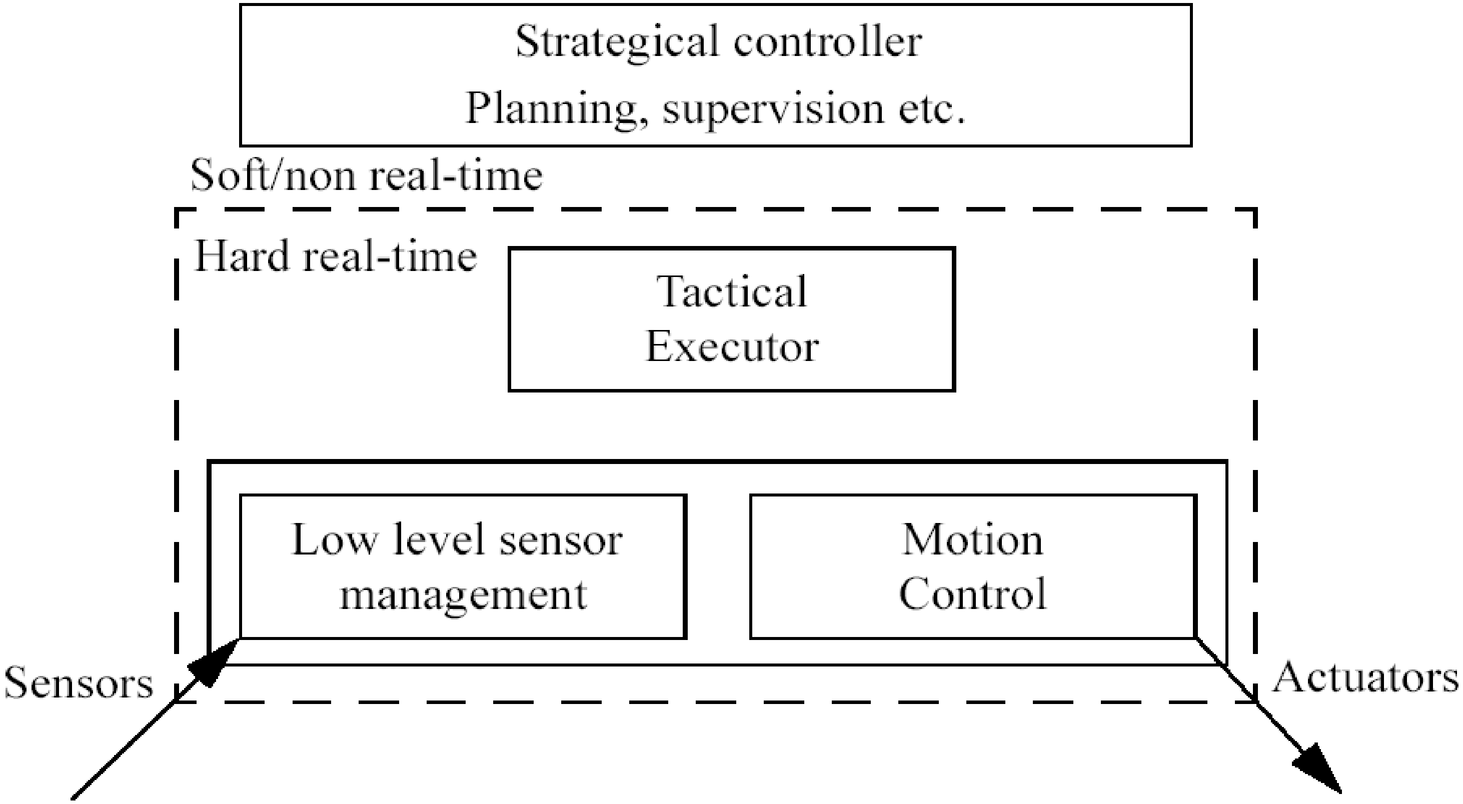
RHD symbol table element

```
 typedef struct  {
int32_t *data;                              //Pointer to the dataarea
int32_t length;      char    name[MAXNAMELEN+1];
uint8_t updated;
uint8_t changed;
struct timeval time;  //timestamp
struct timeval *timestamp;  //Pointer to the timestamp
double  *inputVar;          //Pointer to the MRC variable
 } symTableElement;
```

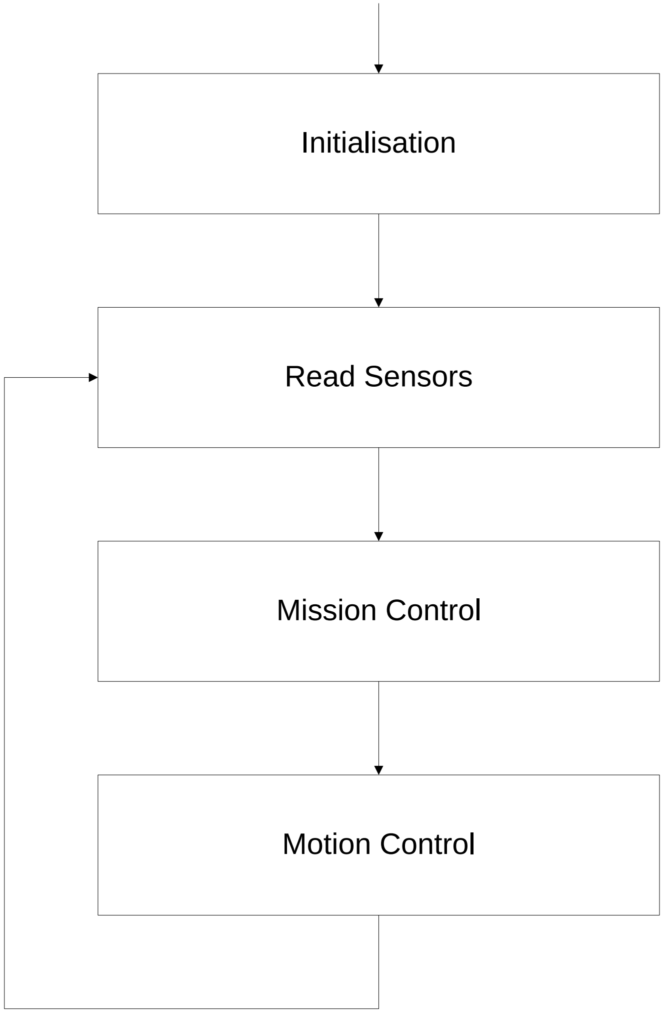
State Machine

```
enum (STATE1,STATE2,STATE3);


if (state != oldstate) {
  statetime=0;
  oldstate=state;
  updatestatelog();
}
else {
  statetime++
}


switch (state){

    case STATE1:
           if (shiftcondition) state=nextstate;
    break;
    case STATE2:

    break;
    case STATE3:

    break;
    default:

}
```

Software architecture



PROGRAM STRUCTURE



Initialisation

Read Sensors

Mission Control

Motion Control

Square program, initialization

Connect to RHD-server

Get pointers to RHD-variables

Connect to camera server

Connect to laser server

Initialize some variables

RUN main-loop

Square Program, main loop

Get data from laser server

Get data from camera server

Syncronize rhd data

Run mission state machine

Run motion state machine

Set motor references

Check for keyboard stop

Update_odo

```
void update_odo(odotype *p){
  int delta;
  delta = p->right_enc - p->right_enc_old;
  if (delta > 0x8000) delta -= 0x10000;
  else if (delta < -0x8000) delta += 0x10000;
  p->right_enc_old = p->right_enc;
 p->right_pos += delta * p->cr;
  ****
}
```

Mobile Robot Controller

- Odometry
- Motion controller
- SMR-CL interpreter
- Socket interface to high level controllers
- XML-based socket interface to sensor servers
- Socket interface to RHD
- XML-based configuration file
- Calibration support for odometry, line-sensors and distance sensors.

SMR Software Modules

- Odometry
- Linesensor
- IR-sensor
- Motioncontrol
- SMR-CL interpreter

Motioncontroller

- Fwd
- Turn
- Drive
- Turnr
- Stop
- Followwall
- Followline

SMR-CL

command parameters [@v vel] [@a accel] [: stopconditions]

where [] means optional.

command   is the desired command

parameters are the parameters required by the command. The parameters can be numbers, variables or string literals (characters within “ “)

vel   desired vehicle velocity in m/s

accel   desired vehicle acceleration m/s^2

stopconditions (expression) [|(expression)]*

a list of ORed stopconditions, the interpreter will continue at the next line if one or more of the conditions are true. The system variable $condition will hold the number of the first true condition (from left)

Flow Control

label   *“labelname”*   defines a label

goto   *“labelname”*   continues execution at the line with label “labelname”

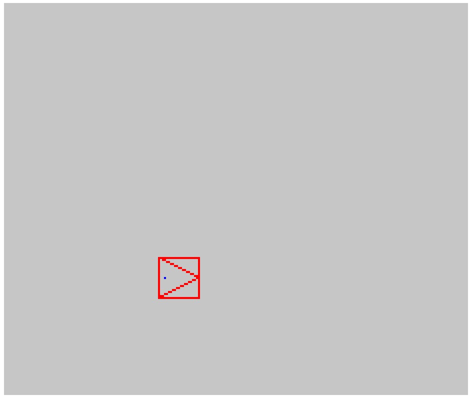
if (expression) “labelname”   jump to “labelname” if expression is not zero

%   The rest of the line after % is a comment

Program Examples

Square program

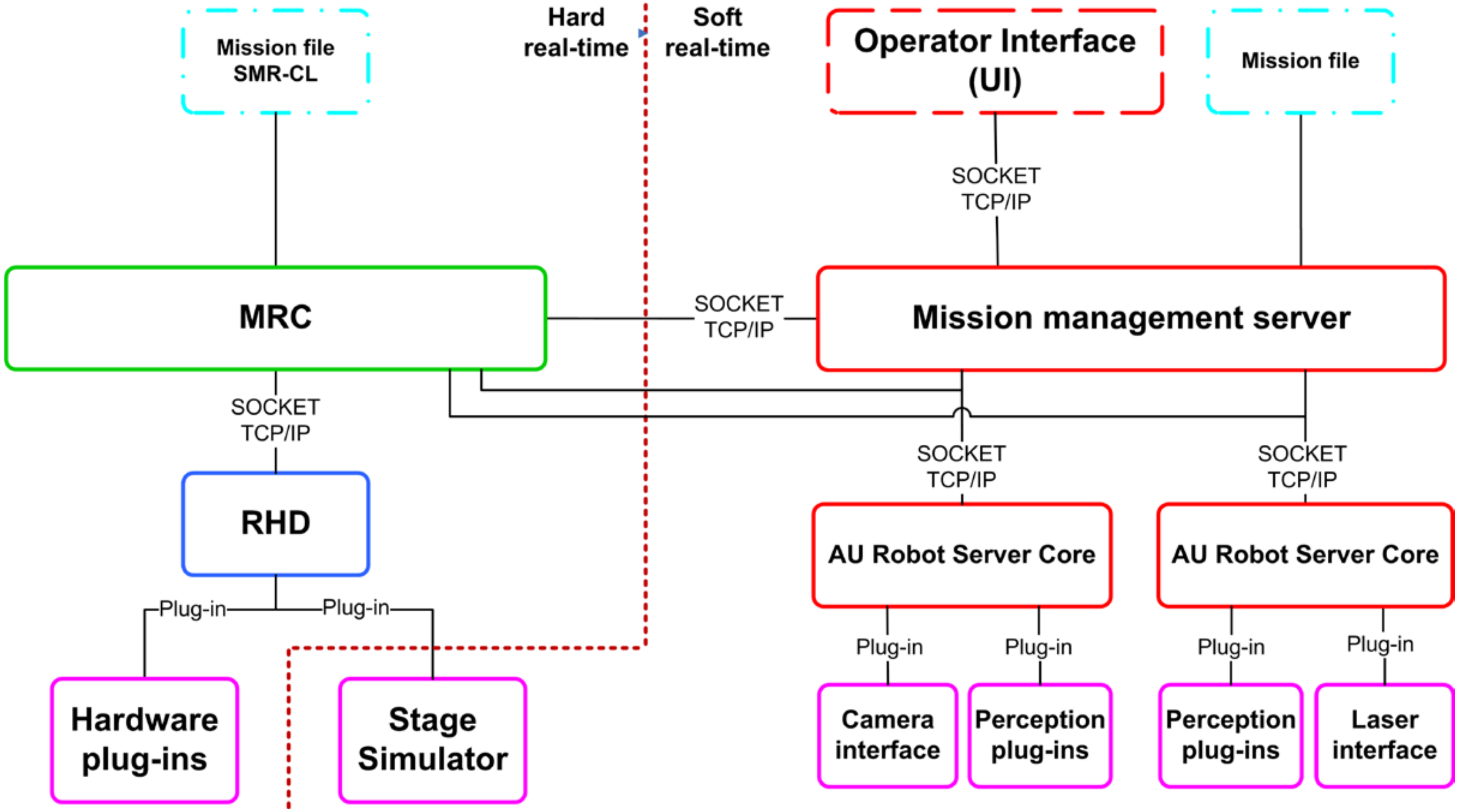
```
n=4
label “start”
drive  :($drivendist >1)
turnr 0.3 90
n=n-1
if (n > 0 ) “start”
Stop
```



Find Gate

```
% Go to a search line 0.4 m from the gate line
followline “wm” @v0.2 :($drivendist >0.4)

% Turn and back to the start of the search line
turn –90
fwd -0.9
% Save start of line dist
dist0= $ododist
% search until gate found or search line ended

drive @v0.05 :($irdistleft < 0.4)|($drivendist > 1.8)
if (condition == 1) “gatefound”
% failed
stop
label “gatefound”

% Calculate distance from middle of gate to track
gatedist=0.9-0.46-($ododist-dist0)

% Go to middle of gate and drive through it
fwd 0.46 @v0.3
turn 90
fwd 0.7
% Go back to track
turn –90
fwd gatedist
turn 90
```

MobotWare Overview


Mission file
SMR-CL
Hard
real-time
Soft
real-time
Operator Interface
(UI)
Mission file
SOCKET
TCP/IP
MRC
SOCKET
TCP/IP
Mission management server
SOCKET
TCP/IP
SOCKET
TCP/IP
SOCKET
TCP/IP
RHD
AU Robot Server Core
AU Robot Server Core
Plug-in
Plug-in
Plug-in
Plug-in
Plug-in
Plug-in
Hardware
plug-ins
Stage
Simulator
Camera
interface
Perception
plug-ins
Perception
plug-ins
Laser
interface