```
function [pwm] = ctrlAlloc(y)
%ctrlAlloc Takes measurements from the robot's sensors and determines what
%control action to make.
yLeft = y(1);
yRight = y(2);

%The following is an example of how you might want to set up a state
%machine in matlab using persistent variables for the state, this value
%remains the same between calls to this function but is reset by the clear
%all at the beginning of the simulation.

%define states
STATE1 = 0;    % driving with no measurement
STATE2 = 1;     % right wall
STATE3 = 2;     % left wall
STATE4 = 3;     % finish state


reference = 520;
Kp = 0.4;
Kd = 0.005;
% Kp = 5;

persistent state rightwallcount counter last_error
if isempty(state)
    state = STATE1;
    rightwallcount =0;
    counter = 50;
    last_error = 0;
end

switch state
    case STATE1
        pwmL = 60;
        pwmR = 60;
        %Transition condition
        if(yRight ~= 1000)
            state = STATE2;
            rightwallcount = rightwallcount + 1;
        elseif (yLeft ~=1000)
            state = STATE3;
        end
    %break; would go here in C
    case STATE2 % measuring right wall
        error = reference - yRight;
        error_difference = (last_error - error)/0.05;

        pwmL = 60 - Kp*error - Kd*error_difference;
        pwmR = 60 + Kp*error + Kd*error_difference;
        %Transition condition
        if ((yRight == 1000) && (rightwallcount == 2))
            state = STATE4;
        elseif(yRight == 1000)
            state = STATE1;
        end
        last_error = error;
    case STATE3 % measuring left wall
        error = -(reference - yLeft);

        pwmL = 60 - Kp*error;
        pwmR = 60 + Kp*error;
        %Transition condition
        if(yLeft == 1000)
            state = STATE1;
        end
    case STATE4
        if (counter >0)
            pwmL = 60;
```

```
            pwmR = 60;
        else
            pwmL = 0;
            pwmR = 0;
        end
        counter = counter - 1;
    %break; would go here in C
    otherwise%this becomes default in C instead of otherwise
end

pwmL = max(0,pwmL);
pwmR = max(0,pwmR);

pwm = [pwmL;pwmR];
end
```