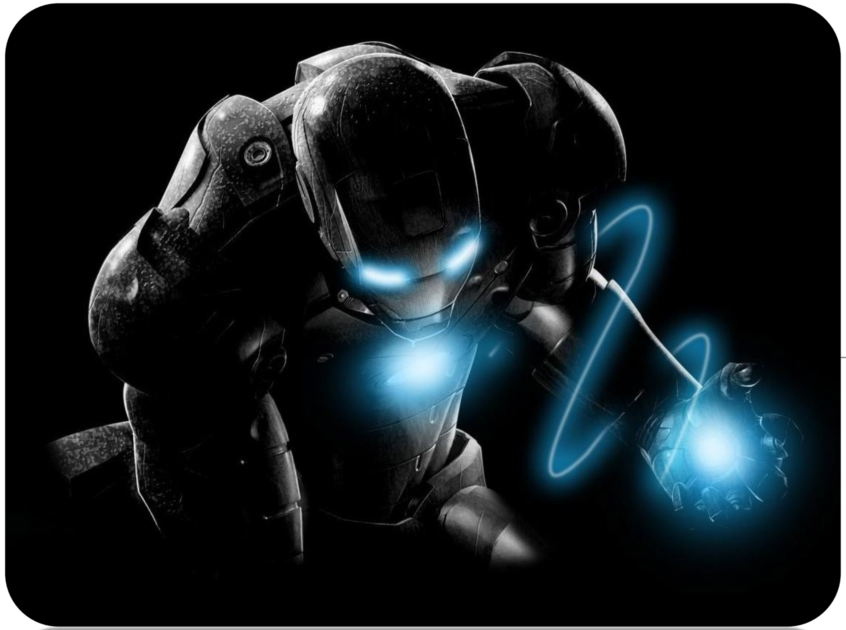
Automatic Control of Mobile Robot
MECT 611
Practical Season



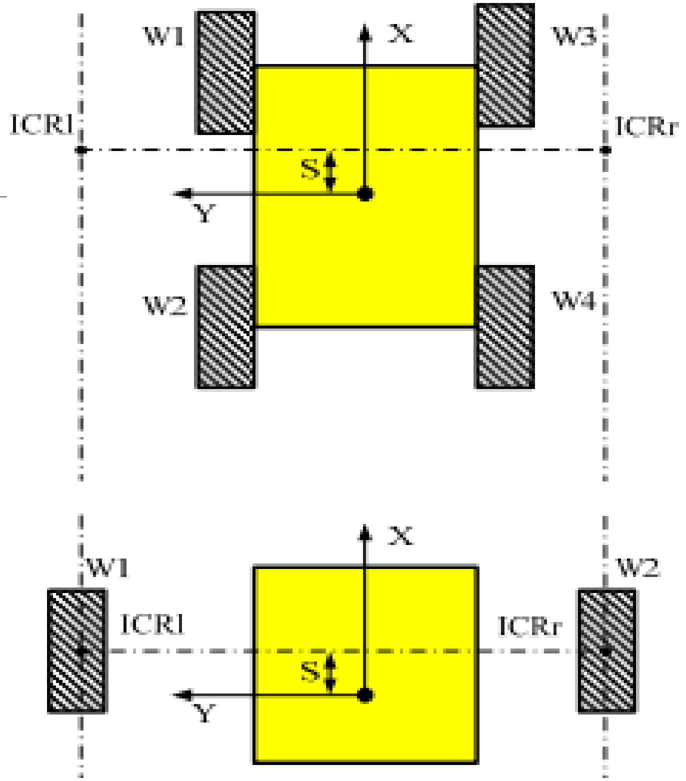
Hossam Hassan Ali

B.SC and M.SC in Industrial Electronics and Automatic Control

Hhassan@nu.edu.eg

Wheeled Mobile Robot Control

- Relation between the control input and speed of wheels



$$V_L = r\ \omega_L \qquad V_R = r\ \omega_R$$

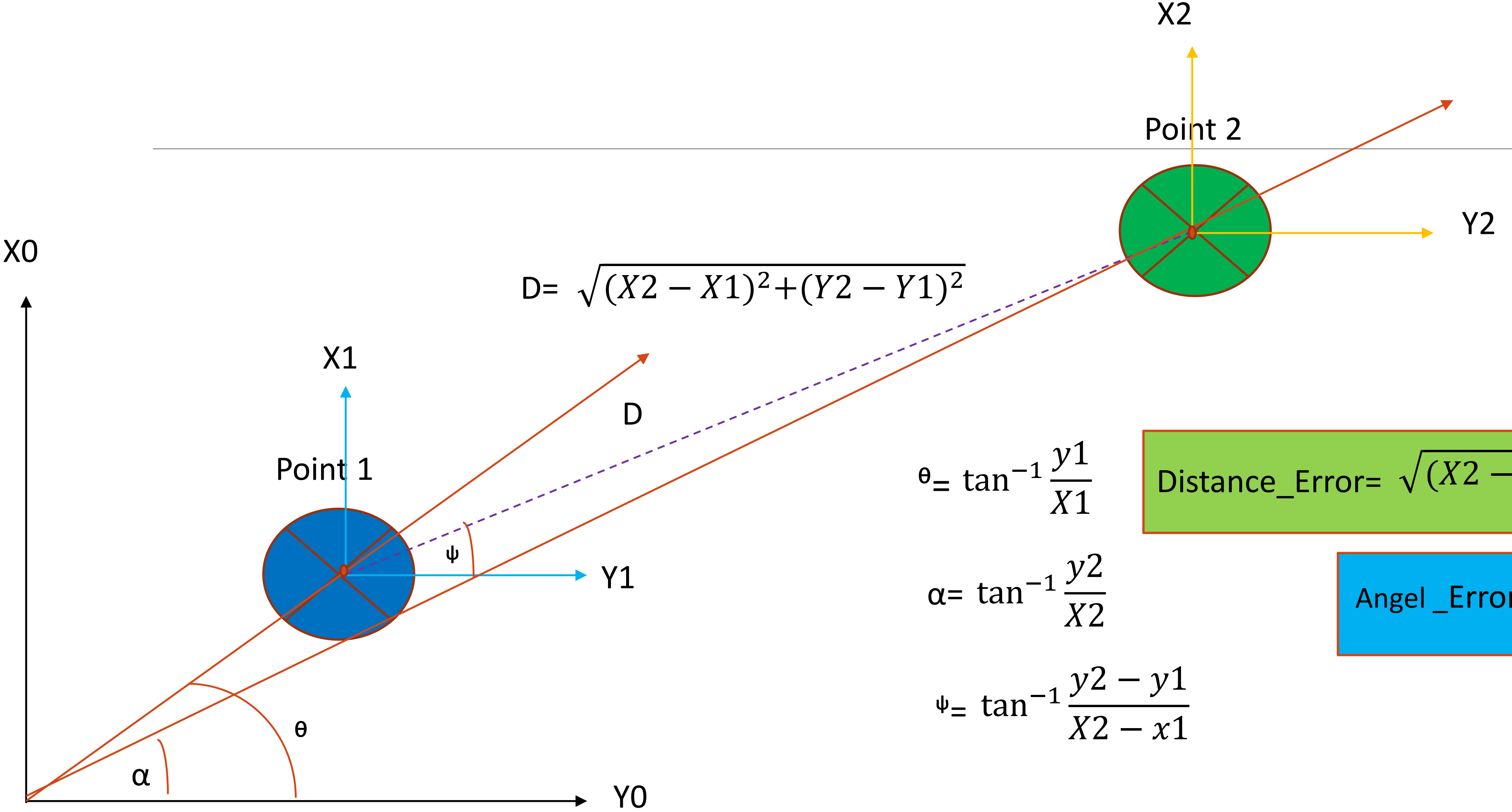
V-linear velocity of the robot
w – angular velocity of the robot

$$\omega = \frac{V_R - V_L}{L} \qquad v = \frac{V_R + V_L}{2}$$

$V_R(t)$ – linear velocity of right wheel
$V_L(t)$ – linear velocity of left wheel
r – nominal radius of each wheel
L–The distance between the two wheels).

$$P = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix}$$

Point-to –Point Movement



θ= $\tan^{-1}\dfrac{y1}{X1}$

α= $\tan^{-1}\dfrac{y2}{X2}$

ψ= $\tan^{-1}\dfrac{y2-y1}{X2-x1}$

Distance_Error= $\sqrt{(X2-X1)^2+(Y2-Y1)^2}$

Angel _Error= θ _ ψ

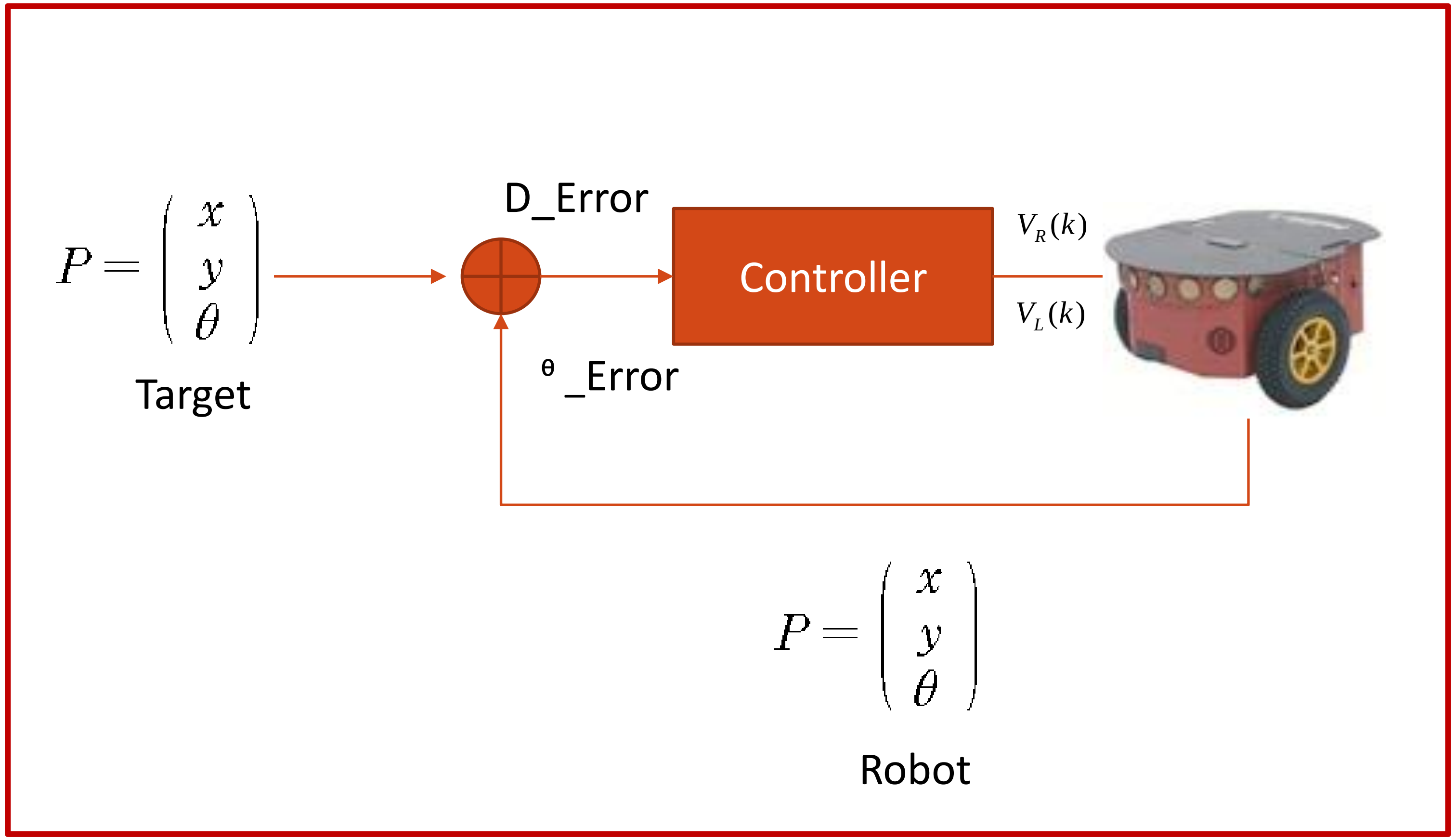
Point –to –point mobile robot control

By solving the following equations we get

$$\omega = \frac{V_R - V_L}{L} \qquad v = \frac{V_R + V_L}{2}$$

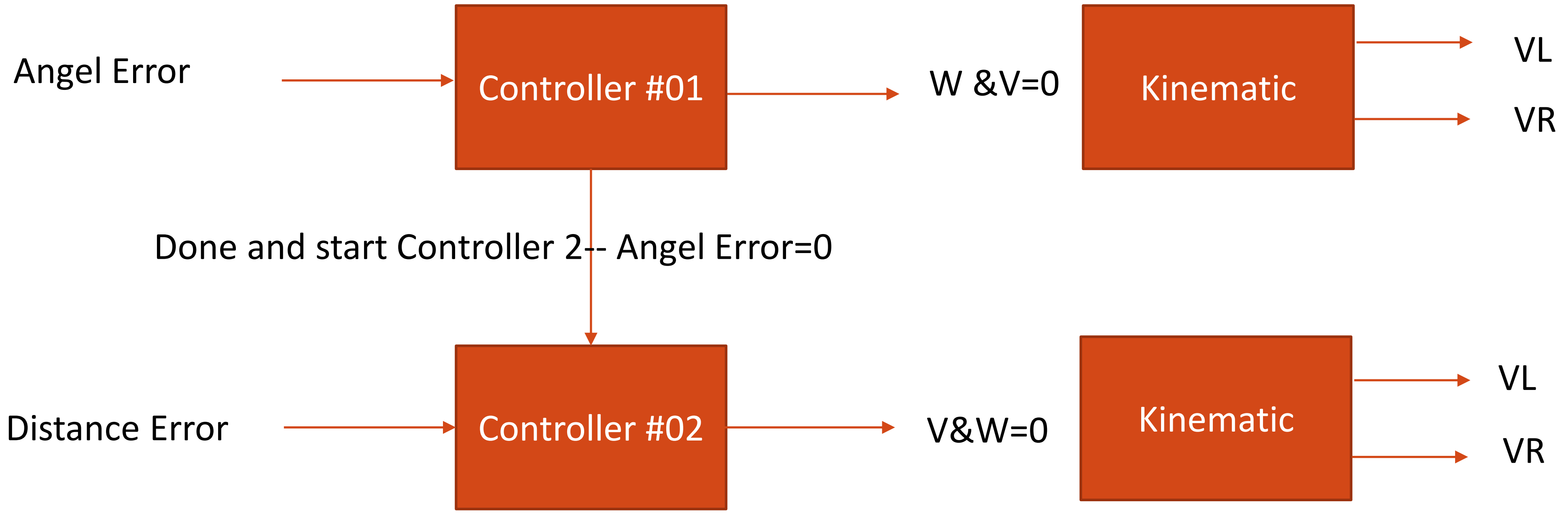
$$V_R(k) = (2*V + W*L)/2$$

$$V_L(k) = (2*V - W*L)/2$$



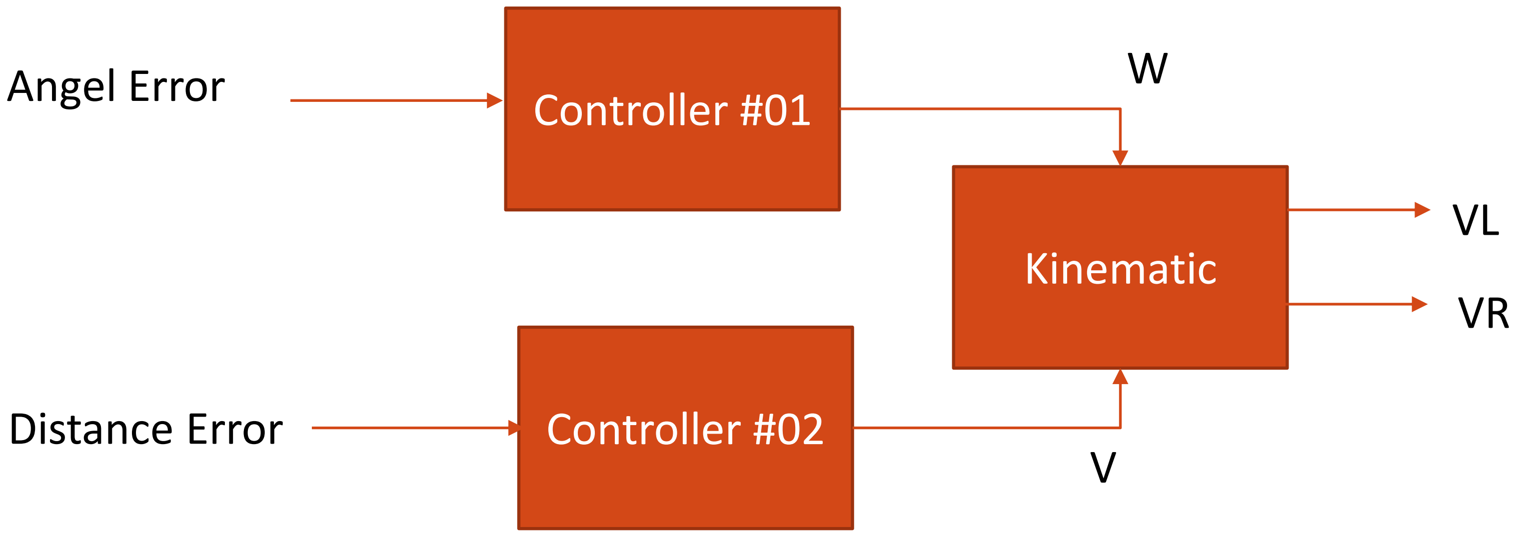
Driving Algorithms

Algorithm#01. sequential Drive

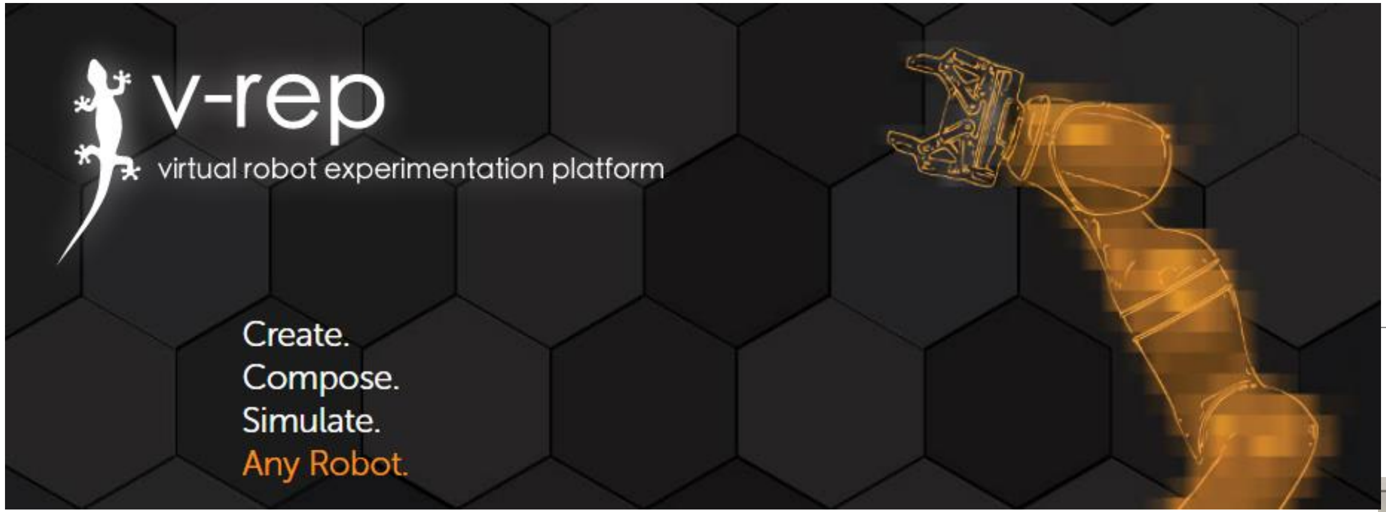


Driving Algorithms

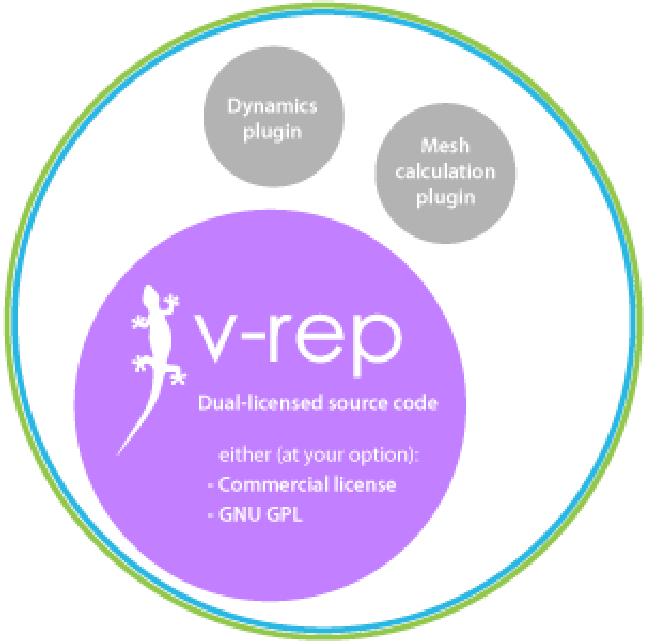
Algorithm#02 Hybrid Drive

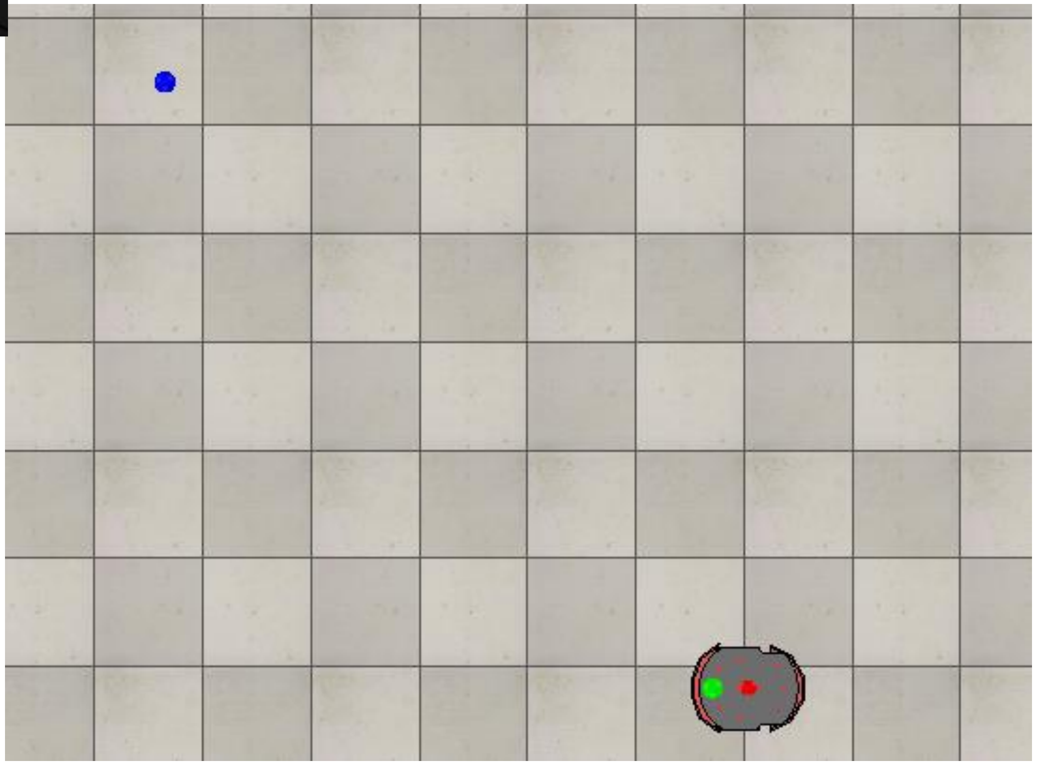


Mobile Robot Platform



simExtRemoteApiStart(19999)

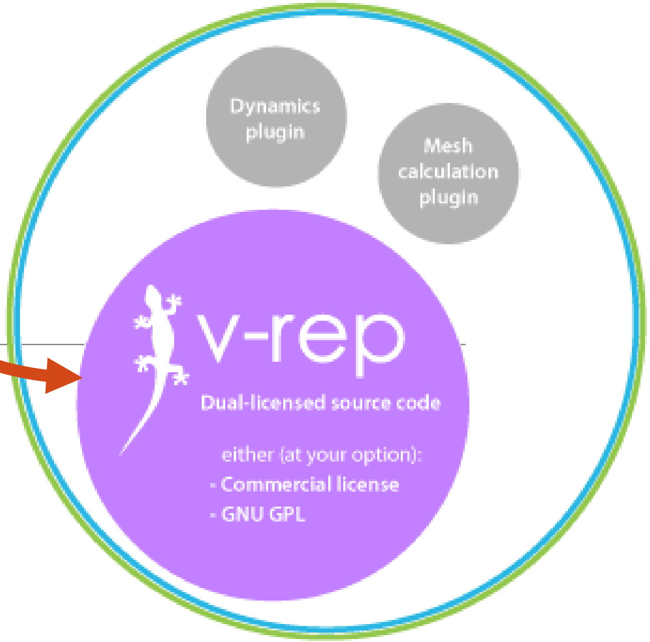




Mobile Robot Platform control via MATLAB

Import Library RemotApi to your MATLAB PATH

MATLAB




```
vrep=remApi('remoteApi');
vrep.simxFinish(-1);
clientID=vrep.simxStart('127.0.0.1',19999,true,true,5000,5);
```

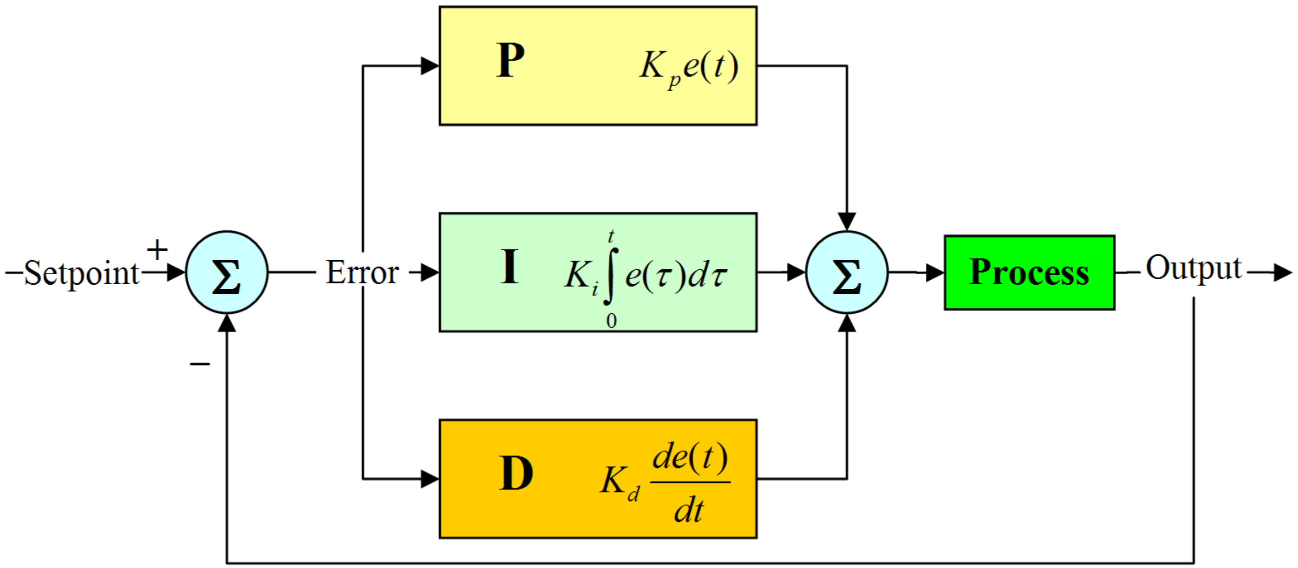
```
[Code, NAME] = vrep.simxGetObjectHandle(clientID, 'NAME', vrep.simx_opmode_oneshot_wait);
```

```
vrep.simxGetObjectPosition(clientID,NAME,-1,vrep.simx_opmode_streaming);
vrep.simxGetObjectOrientation(clientID,NAME,-1,vrep.simx_opmode_streaming);
```

```
[Code,data1]=vrep.simxGetObjectPosition(clientID,Base,-1,vrep.simx_opmode_buffer);
[Code,data2]=vrep.simxGetObjectOrientation(clientID,Base,-1,vrep.simx_opmode_buffer);
```

```
vrep.simxSetJointTargetVelocity(clientID,Name,(Value),vrep.simx_opmode_oneshot);
```

PID Controller




```
error_previous = 0
integral = 0

loop:
    Gosub get_set_point_value ' acquire set_point value
    Gosub acquire_sensor_ ' acquire sensor_value

     error = sensor - set_point
     integral = integral + error*DT
     derivative = (error - error_previous)/DT

     output = KP*error + KI*integral + KD*derivative
     Gosub send_output_to_system ' update command signal

     error_previous = error
Goto loop
```

Driving Algorithms

Algorithm#02 Hybrid Drive

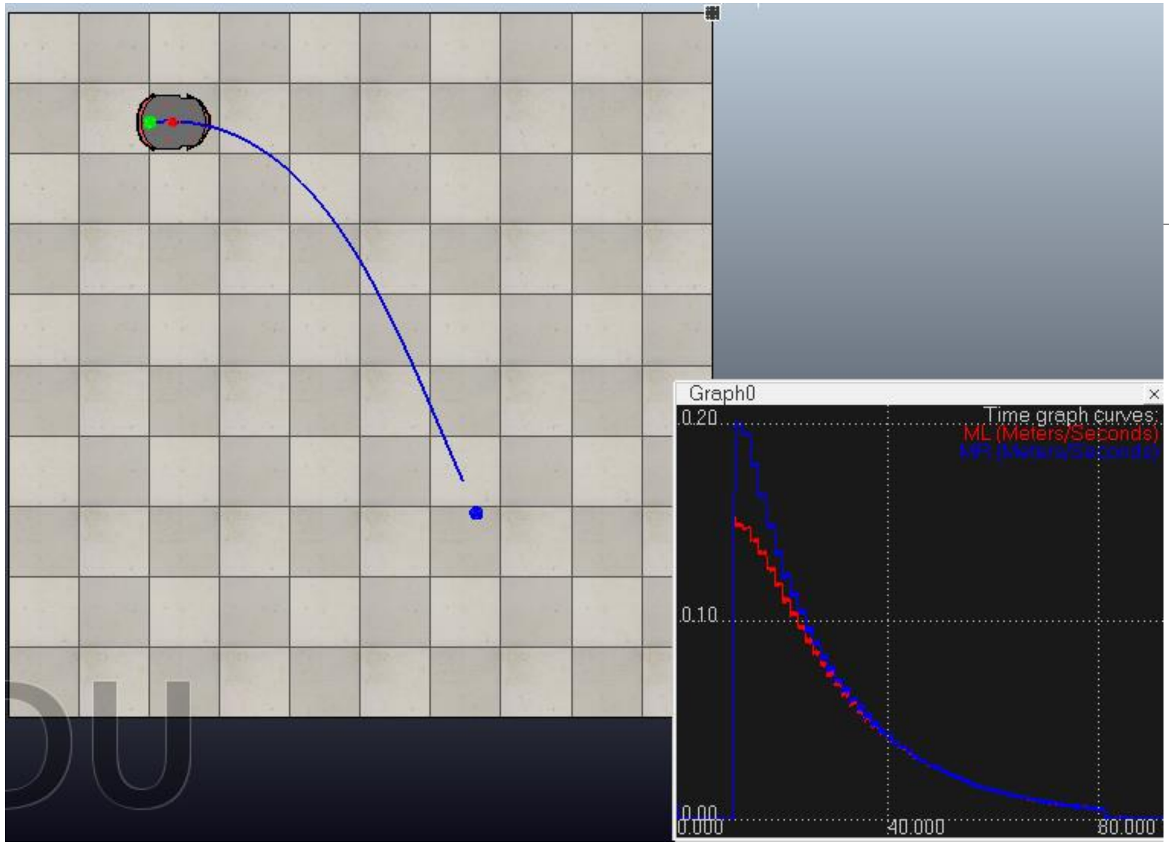
```
function D =D_Drive(KP1,KI1,KD1,KP2,KI2,KD2)
dT=0.1;
E_THO=0;
E_DO=0;
INT_TH=0;
DRif_TH=0;
INT_D=0;
DRif_D=0;
[XR,YR,THR,XT,YT,THT]=FRKB(0,0);
E_D=nearest(sqrt(((XT-XR)^2)+((YT-YR)^2)));
Err_TH=THT-THR;
[XR,YR,THR,XT,YT,THT]=FRKB(0,0);
E_D=sqrt(((XT-XR)^2)+((YT-YR)^2));
Err_TH=THT-THR;
while(abs(E_D)>10)
      Err_TH=THT-THR;
      E_D=nearest(sqrt(((XT-XR)^2)+((YT-YR)^2)))
     INT_TH=INT_TH+(Err_TH*dT);
    DRif_TH=(Err_TH-E_THO)/dT;
    W=KP1*Err_TH+KD1*DRif_TH+KI1*INT_TH;
     INT_D=INT_D+(E_D*dT);
    DRif_D=(E_D-E_DO)/dT;
    V=KP2*E_D+KD2*DRif_D+KI2*INT_D;
    VL=(-V+W)/2;
    VR=(-V-W)/2;
    [XR,YR,THR,XT,YT,THT]=FRKB(VL,VR,1);
    D='Go';
end
 [XR,YR,THR,XT,YT,THT]=FRKB(0,0);
 D='Done';
end
```

Algorithm#01. sequential Drive

```
function D=G_Drive(KP1,KI1,KD1,KP2,KI2,KD2)
dT=0.25;
E_THO=0;
E_DO=0;
INT_TH=0;
DRif_TH=0;
INT_D=0;
DRif_D=0;
[XR,YR,THR,XT,YT,THT]=FRKB(0,0);
E_D=nearest(sqrt(((XT-XR)^2)+((YT-YR)^2)));
Err_TH=THT-THR;
[XR,YR,THR,XT,YT,THT]=FRKB(0,0);
E_D=sqrt(((XT-XR)^2)+((YT-YR)^2));
Err_TH=THT-THR;
 while(abs(Err_TH)>0.1)
      Err_TH=THT-THR;
     INT_TH=INT_TH+(Err_TH*dT);
    DRif_TH=(Err_TH-E_THO)/dT;
    W=KP1*Err_TH+KD1*DRif_TH+KI1*INT_TH;
      VL=W;
      VR=-1*W;
       [XR,YR,THR,XT,YT,THT]=FRKB(VL,VR);
 end
 while(abs(E_D)>10)
     E_D=nearest(sqrt(((XT-XR)^2)+((YT-YR)^2)));
     INT_D=INT_D+(E_D*dT);
    DRif_D=(E_D-E_DO)/dT;
    V=KP2*E_D+KD2*DRif_D+KI2*INT_D;
    VL=-V;
    VR=-V;
    [XR,YR,THR,XT,YT,THT]=FRKB(VL,VR);
    D='Go';
 end
 [XR,YR,THR,XT,YT,THT]=FRKB(0,0);
D='Done';
end
```

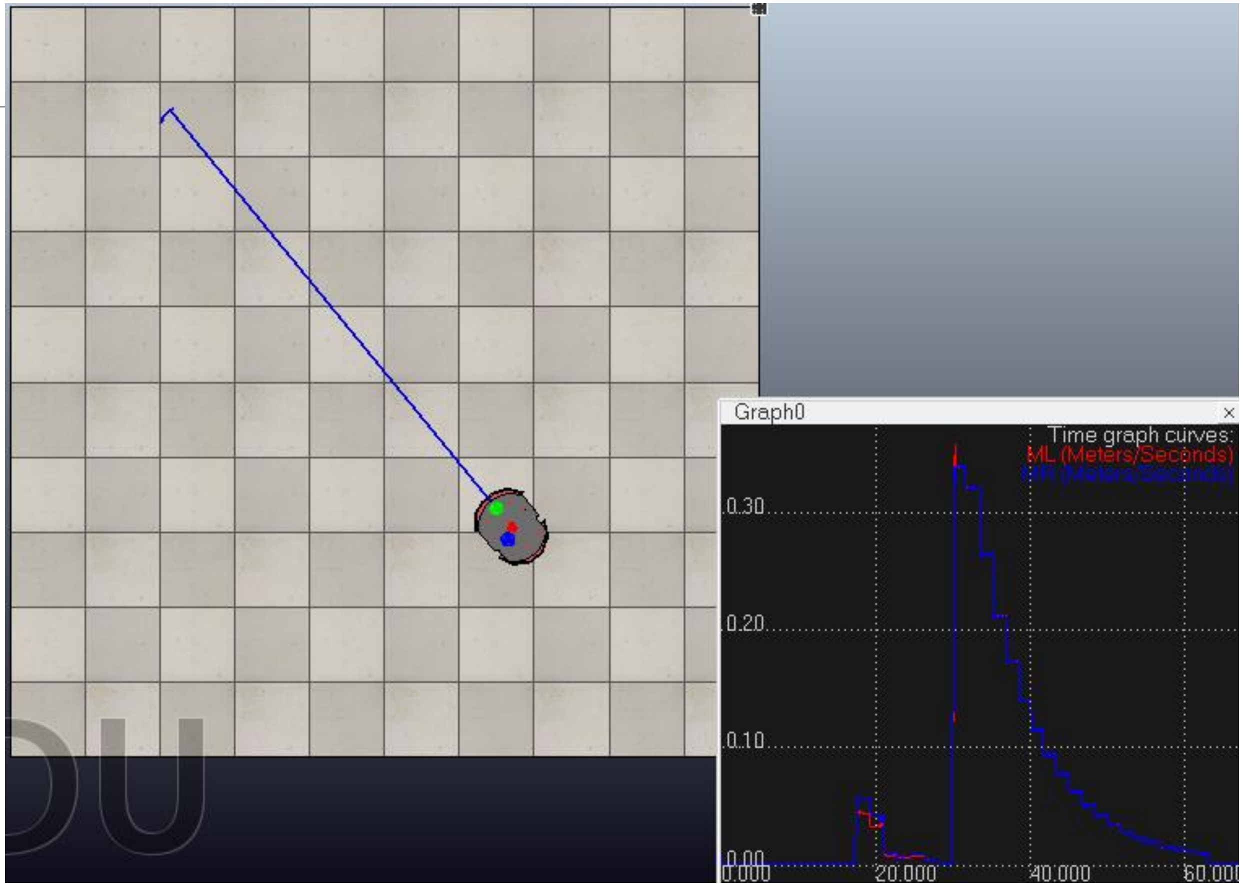
Results

Algorithm#02 Hybrid Drive



D_Drive(0.005,0,0,0.005,0,0)

G_Drive(0.01,0,0,0.005,0,0)



Algorithm#01. sequential Drive

Hossam Hassan

B.SC and M.SC in Industrial Electronics and Automatic Control

Hhassan@nu.edu.eg