Mobile Robot Navigation – some issues in controller design and implementation

L. Huang

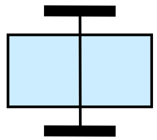
School of Engineering and Advanced Technology
Massey University

Outlines

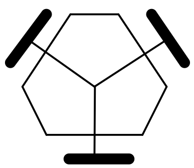
1. Introduction
2. Target tracking control schemes based on
   - Lyapunov method
   - Potential field method
3. Speed control
4. Conclusion

Introduction

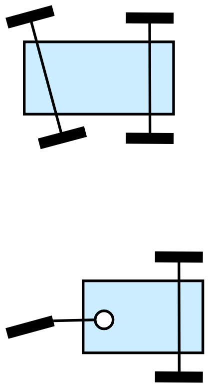
A wheeled mobile robot (WMR) can be driven by wheels in various formations:



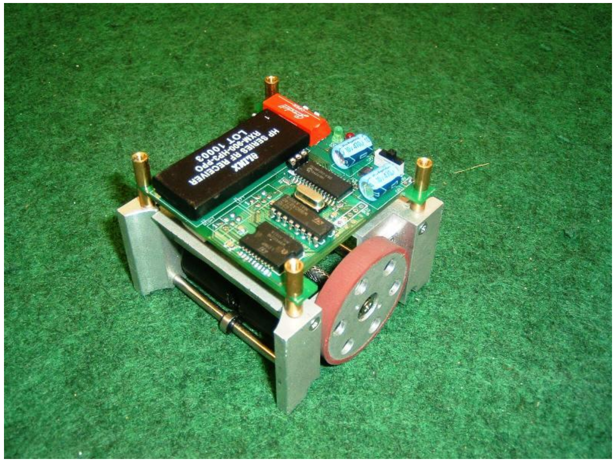
Differential

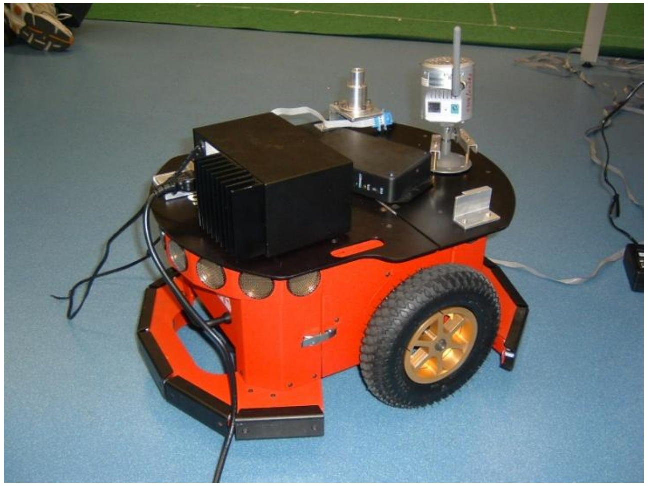


Omni Directional

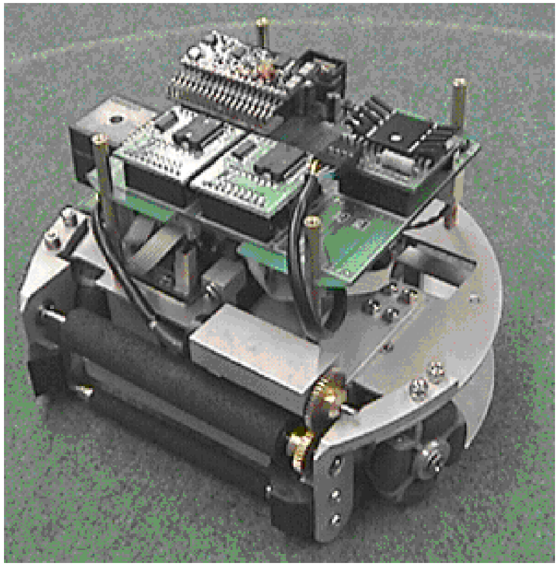


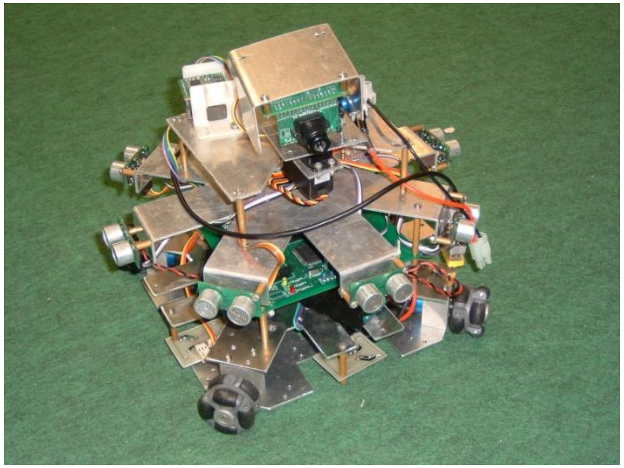
Steering





Differential Wheel Robot

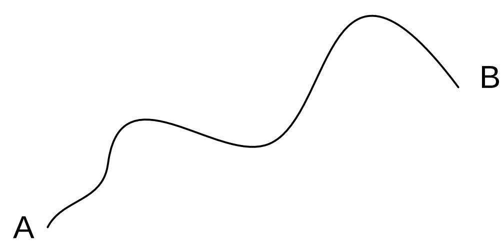




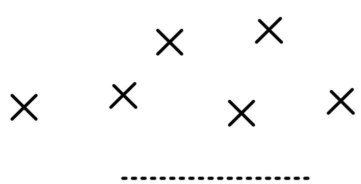
Omni Wheel Robot

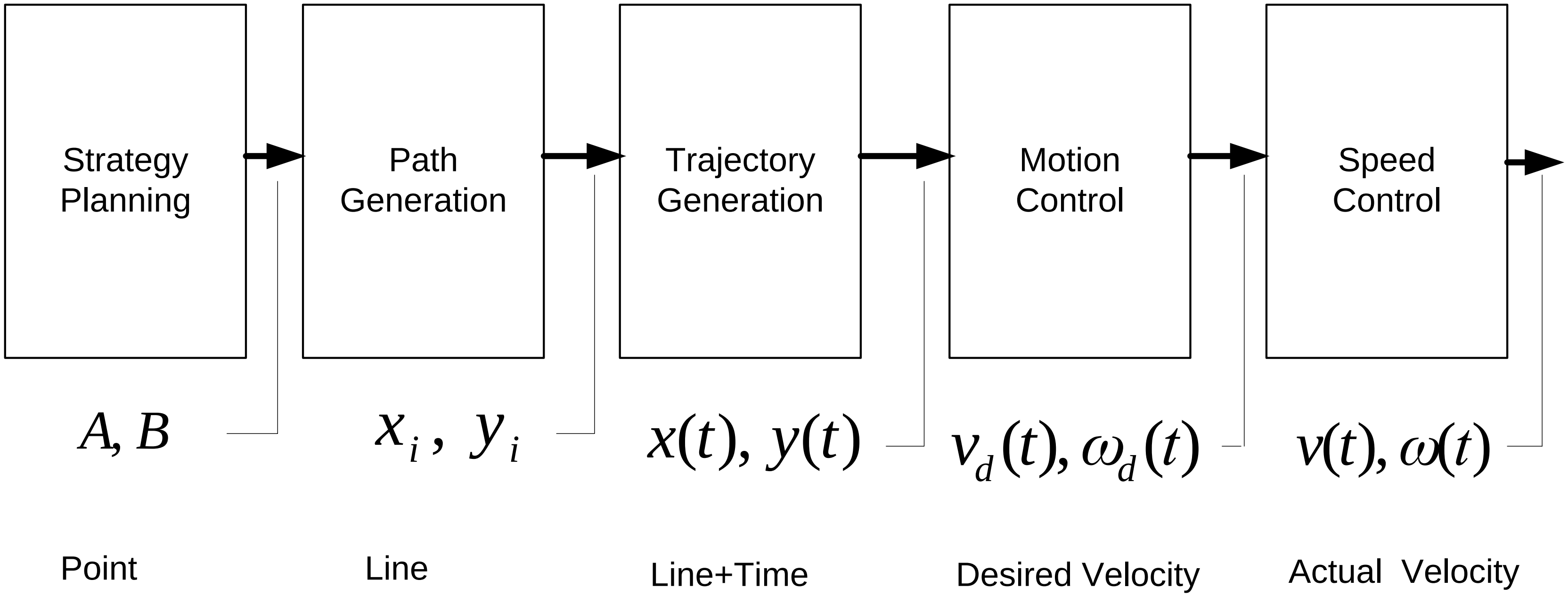
Two basic issues:

1. How to move a robot from posture A to posture B stand alone ?



2. How to determine postures A and B for a robot when a group of robots performing a task (such as soccer playing) ?




Strategy Planning
Path Generation
Trajectory Generation
Motion Control
Speed Control
A, B
$x_i\ ,\ y_i$
$x(t),\ y(t)$
$v_d(t), \omega_d(t)$
$v(t), \omega(t)$
Point
Line
Line+Time
Desired Velocity
Actual Velocity

Differential wheel driven robot (no-holonomic):

- Robot's posture (Cartesian coordinates) cannot be stablized by time-invariant feedback control or smooth state feedback control (*Brockett R. W.* etc.).

- Stabilization problem was solved by discontinuous or time varying control in Cartesian space (*Campion G. B., Samson C.* etc.)

- Asymptotic stabilization through smooth state feedback was achieved by Lyapunov design in Polar coordinates – the system is singular in origin, thus avoids the Brockett's condition (*Aicardi M.* etc).

- Trajectory tracking control is easier to achieve and is more significant in practice (desired velocity nonzero) (*Caudaus De Wit*, *De Luca A* etc.).

Omni-wheel driven robot

- It is fully linearisable for the controller design (*D'Andrea-Novel* etc.)
- Dynamic optimal control was implemented (*Kalmar-Nagy etc*.)

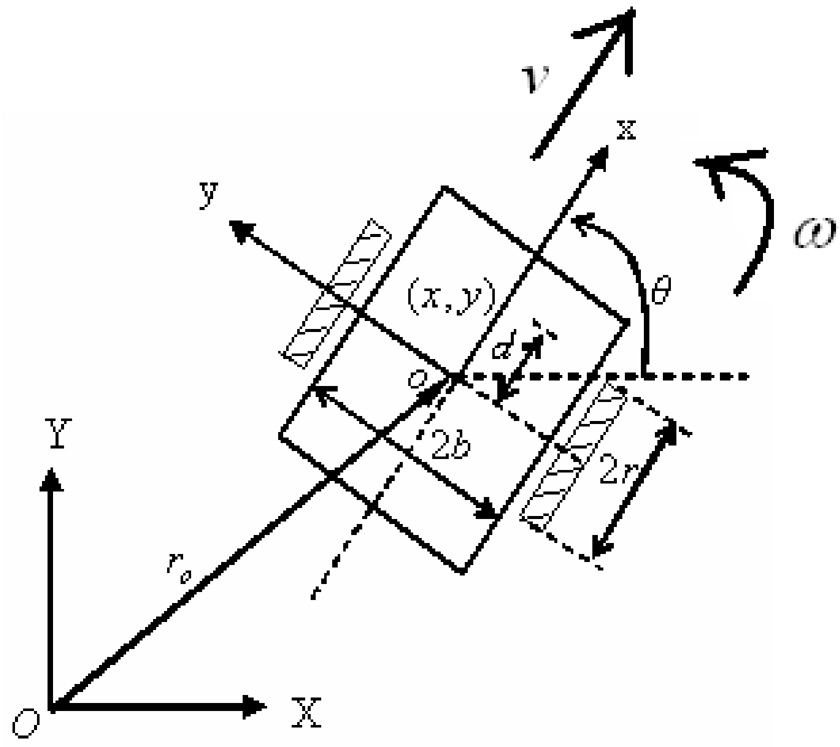
Robot modeled as a point-mass

- Potential field method was used for robot path planning (*Y.Koren and J. Borenstein*)

Issues to be addressed

- Application of Lyapunov-based and potential field based methods in the development of target tracking control scheme

General control approaches



Differential Wheel Robot

- *Kinematic Model*

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix}$$

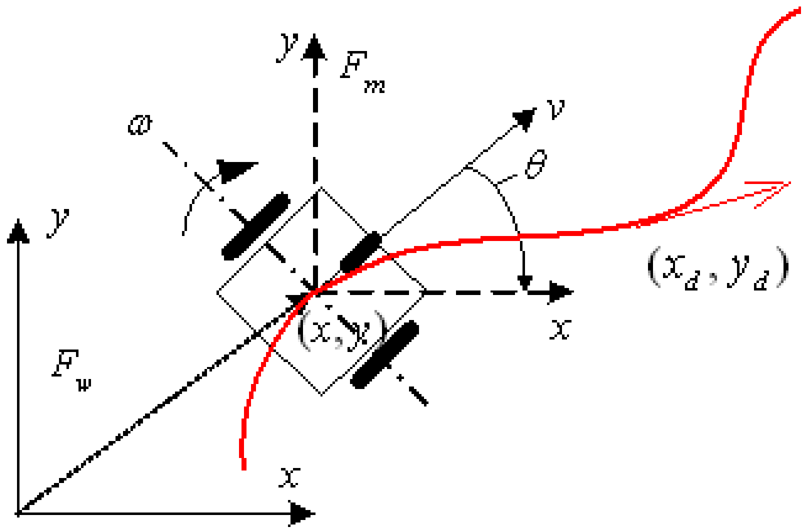
- *Nonholonomic Constraint* (rolling contact without slipping)

$$\dot{x}\sin\theta - \dot{y}\cos\theta = 0$$

❑ *Nonhonolonic (No-integrable) and under actuated (2-input~3-output)*

❑ *cannot be stabilized by time-invariant or smooth feedback control*

Trajectory tracking (Cartesian coordinates based)



Given $x_d, y_d, \dot{x}_d$ and $\dot{y}_d$

find v and ω

to make $x \to x_d, y \to y_d$

It can be proved (due to Lyapunov and Barbalat), the following control can meet the objective：

$$v = v_d \cos(\theta_d - \theta) + k_1[\cos\theta(x_d - x) + \sin\theta(y_d - y)]$$

$$\omega = \omega_d + k_2 \operatorname{sgn}(v_d)[\sin\theta(x_d - x) - \cos\theta(y_d - y)] + k_3(\theta_d - \theta)$$

$$v_d = \pm\sqrt{\dot{x}_d^2 + \dot{y}_d^2}$$

Desired linear velocity (along the trajectory)

$$\omega_d = \frac{\ddot{y}_d \dot{x}_d - \ddot{x}_d \dot{y}_d}{\dot{x}_d^2 + \dot{y}_d^2}$$

Desired angular velocity

$$\theta_d = ATAN2(\dot{y}_d, \dot{x}_d) + k\pi$$

Desired direction

$$k_1 = k_3 = 2\xi\sqrt{\omega_d^2 + b v_d^2}, \quad k_2 = b|v_d|$$

From the planned trajectory

Note： The trajectory needed to be specified in prior； the controller fails when $v_d = 0$

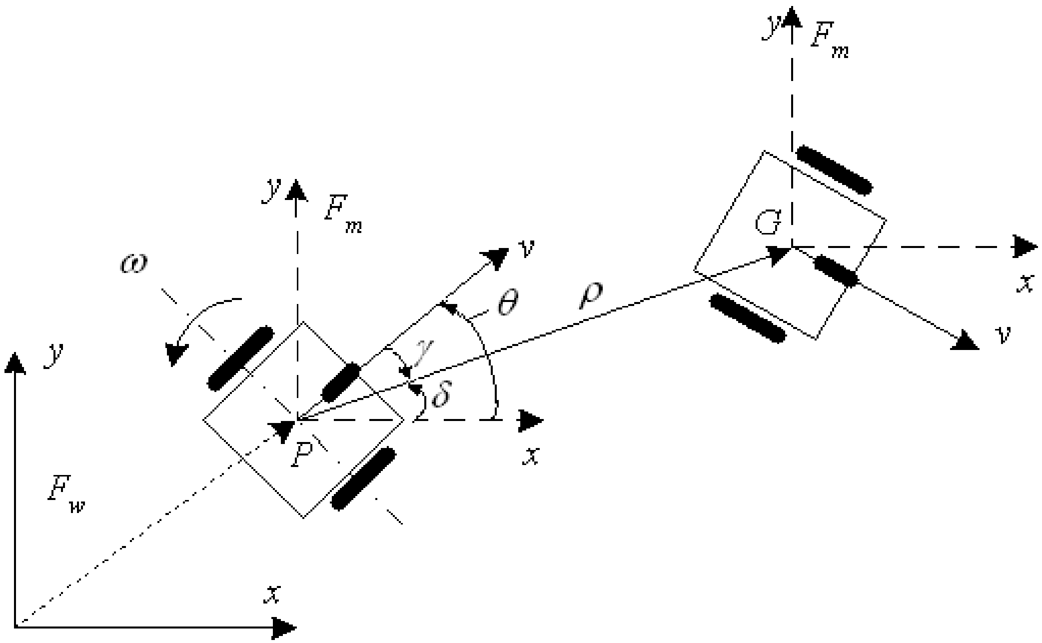
with nonlinear modifications to adjust angular motion:

$$v = v_d \cos(\theta_d - \theta) + k_1[\cos\theta(x_d - x) + \sin\theta(y_d - y)]$$

$$\omega = \omega_d + \bar{k}_2 v_d \frac{\sin(\theta_d - \theta)}{\theta_d - \theta}[\sin\theta(x_d - x) - \cos\theta(y_d - y)] + k_3(v_d, \omega_d)(\theta_d - \theta)$$

, where $\bar{k}_2 = b$

Goal / target tracking (Polar coordinates based)



Control task: move the robot from its original posture: (x_p, y_p, θ_p) to the target posture (x_g, y_g, θ_g) $(\theta_g = 0:$ *parallel parking*).

The system model described in polar coordinates:

$$\dot{\rho} = -v\cos\gamma, \ \dot{\gamma} = -\omega + v\frac{\sin\gamma}{\rho}, \ \dot{\delta} = v\frac{\sin\gamma}{\rho}$$

$$\rho = \sqrt{(x_g - x)^2 + (y_g - y)^2}$$

$$\delta = \tan^{-1}(\frac{y_g - y}{x_g - x})$$

$$\gamma = \delta - \theta$$

The model is singular at $\rho = 0$

Let $v = k_1 \rho \cos\gamma$ $\omega = k_2 a_2 \gamma + \frac{k_1 \sin 2\gamma}{2 a_2 \gamma}(a_2\gamma + a_3\delta)$

It can be proved that (*due to Lyapunov and Barbalat*)

$$\rho \to 0, \gamma \to 0, \delta \to 0$$

with the Lyapunov function candidate

$$V = \frac{1}{2} a_1 \rho^2 + \frac{1}{2} \mathrm{a}_2 \gamma^2 + \frac{1}{2} \mathrm{a}_3 \delta^2, \mathrm{a}_1, \mathrm{a}_2, \mathrm{a}_3 > 0$$

$$\dot{V} = -a_1 k_1 \cos^2 \gamma \, \rho^2 - k_2 {a_2}^2 \gamma^2 \le 0$$

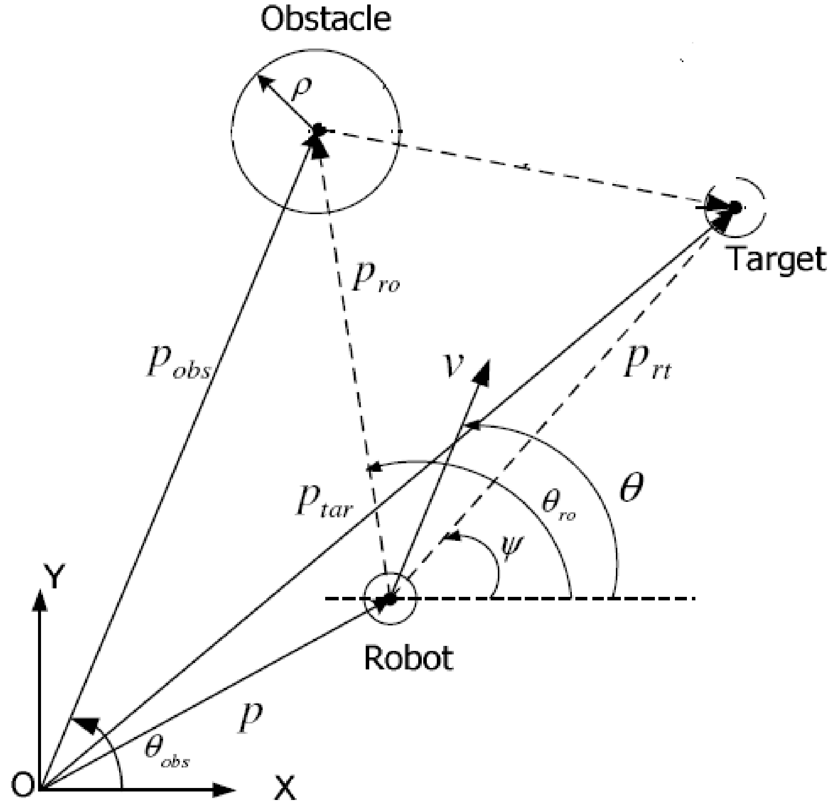
- large control effort or fluctuation when the angle tracking error is near zero or the linear tracking error is big
- the target is assumed to be stationary

Potential field approach (point mass model)

Attractive and repulsive fields:

$$U_{att} = \frac{1}{2}\xi_1 p_{rt}^T p_{rt}$$

$$U_{rep} = \begin{cases} \frac{1}{2}\xi_2(\rho^{-1} - \rho_0^{-1})^2, \text{if } \rho \le \rho_0 \\ 0 \qquad \text{else} \end{cases}$$



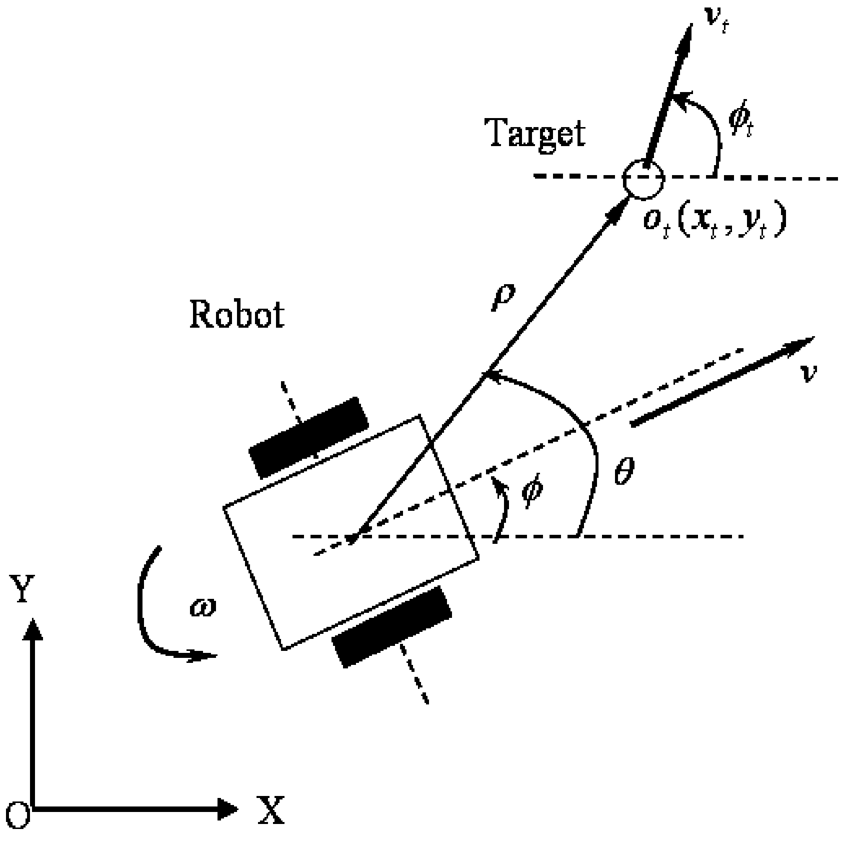
Robot move along the negative gradient of the combined field:

$$v = -\nabla_p U_{att}(p) - \nabla_p U_{rep}(p)$$

$$= \begin{cases} \xi_1 p_{rt} + \xi_2(\rho^{-1} - \rho_0^{-1})\rho^{-2}\nabla_p \rho, \text{ if } \rho \le \rho_0 \\ \xi_1 p_{rt}, \qquad \text{else} \end{cases}$$

- The law only specifies the direction of the robot velocity
- target is assumed to be stationary
- local minima

Lyapunov based target tracking controller with limited control efforts



System model (*extended from the conventional one by including the velocity of the target*):

$$
\begin{cases}
\dot{\rho} = v_t \cos\beta - v\cos\alpha \\
\dot{\alpha} = v\dfrac{\sin\alpha}{\rho} - v_t\dfrac{\sin\beta}{\rho} - \omega \\
\dot{\beta} = v\dfrac{\sin\alpha}{\rho} - v_t\dfrac{\sin\beta}{\rho} - \dot{\phi}_t, \quad \rho \neq 0
\end{cases}
$$

$$\alpha = \theta - \phi, \;\; \beta = \theta - \phi_t$$

Controller 1: Extension of the general control approach

$$v = (v_t \cos\beta + \lambda_v \rho)\cos\alpha$$

$$\omega = \lambda_\alpha \alpha + \frac{\alpha+\beta}{\rho}\left(\frac{\sin 2\alpha}{2\alpha}\cos\beta - \frac{\sin\beta}{\alpha}\right)v_t - \frac{\beta}{\alpha}\dot{\phi}_t + \frac{\sin 2\alpha}{2\alpha}\lambda_v(\alpha+\beta)$$

It can be proved with Lyapunov method, that under the controller,

$$\alpha \to 0, \rho \to 0 \text{ and } \beta \to 0$$

(Lyapunov function candidate: $V = \frac{1}{2}(\rho^2 + \alpha^2 + \beta^2)$)

Note:

- target motions directly affects the control efforts
- sinusoidal functions of the systems states attenuate the magnitude of control
- tracking errors appear as the denominators in the terms of the controller
- linear tracking and angular tracking errors are treated equally – too demanding ?

Controller 2: Improvement from Controller 1

Prioritise and change the control objectives:

$$\rho \to 0,\ \alpha \to 0\,\text{(or bounded)},\ \alpha - \beta \to 0\ \text{(or bounded)}$$

and reflect them in the definition of the Lyapunov function:

$$V = \frac{1}{2}\rho^2 + \frac{1}{2}\alpha^2 + \frac{1}{2}(\alpha - \beta)^2$$

New controller:

$$v = (v_t \cos\beta + \lambda_v \rho)\cos\alpha$$

$$\omega = \frac{\alpha}{2\alpha - \beta}\left(\lambda_\alpha \alpha + \frac{v_t}{\rho}\left(\frac{\sin 2\alpha}{2}\cos\beta - \sin\beta\right) + \frac{\sin 2\alpha}{2}\lambda_v\right) + \frac{\alpha - \beta}{2\alpha - \beta}\dot{\phi}_t$$

which can also achieve the convergence of the tracking errors, but with less control efforts

Comparison: control efforts of Controllers 1 and Controller 2

$$\omega_1 = \lambda_\alpha \alpha + \frac{\alpha+\beta}{\rho}(\frac{\sin 2\alpha}{2\alpha}\cos\beta - \frac{\sin\beta}{\alpha})v_t - \frac{\beta}{\alpha}\dot{\phi}_t + \frac{\sin 2\alpha}{2\alpha}\lambda_v(\alpha+\beta)$$

$$\omega_2 = \frac{\alpha}{2\alpha-\beta}(\lambda_\alpha \alpha + \frac{v_t}{\rho}(\frac{\sin 2\alpha}{2}\cos\beta - \sin\beta) + \frac{\sin 2\alpha}{2}\lambda_v) + \frac{\alpha-\beta}{2\alpha-\beta}\dot{\phi}_t$$

or

$$\omega_1 = \xi - \gamma_1 \eta - \lambda_\alpha \beta$$
$$\omega_2 = \xi - \gamma_2 \eta$$

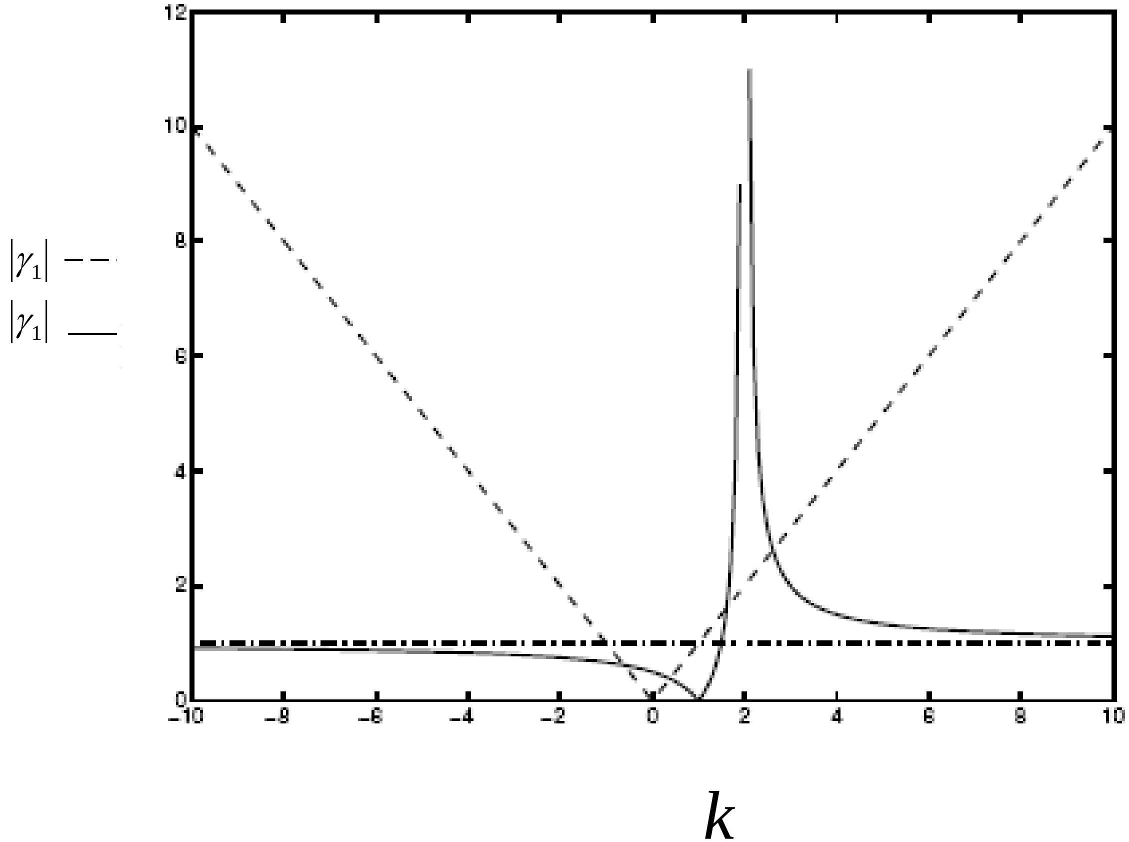
$$\xi = \lambda_\alpha \alpha + \frac{v_t}{\rho}(\frac{\sin 2\alpha}{2}\cos\beta - \sin\beta) + \frac{\sin 2\alpha}{2}\lambda_v$$

$$\eta = \xi - \dot{\phi}_t,\ \ \gamma_1 = k,\ \ \gamma_2 = \frac{1-k}{2-k}$$

$$k = \frac{\beta}{\alpha}$$

- By observation, the magnitude of controller 2 is less than that controller 1

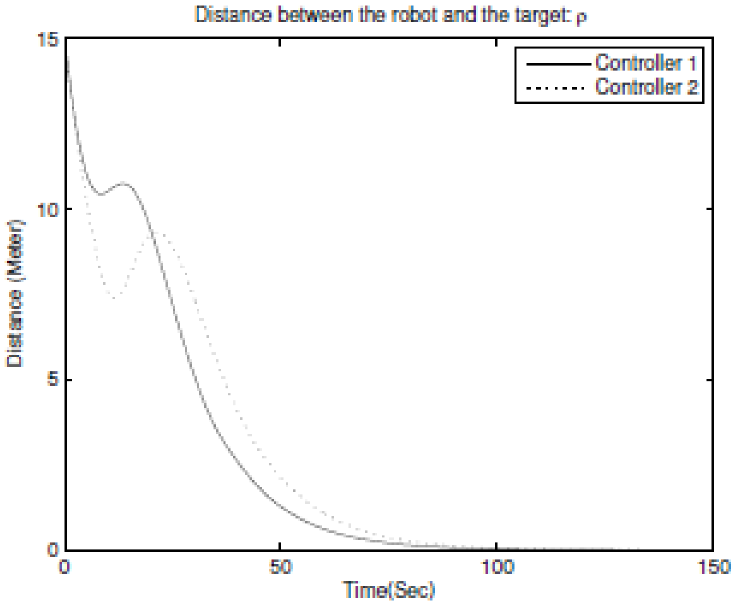
- Analysing the factors (γ) affecting the controller magnitude, it is obvious that, except for the region near $k = \frac{\beta}{\alpha} = 2$ that affecting Controller 1 is larger in magnitude than that affecting Controller 2.



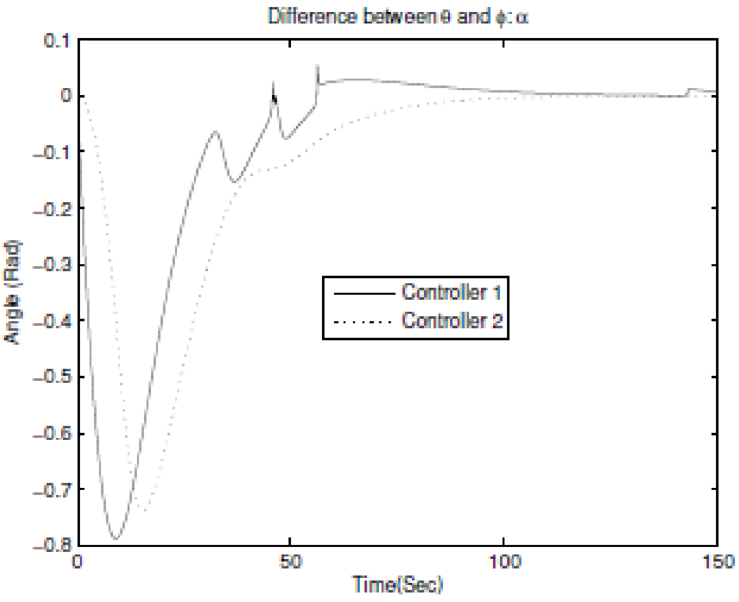
Simulation Results (*tracking a target Moving along a circle)*

$$x_t = 3 - 15\cos(0.08t),\ \ y_t = 47 + 15\sin(0.08t),\ \ v_t = 1.2$$

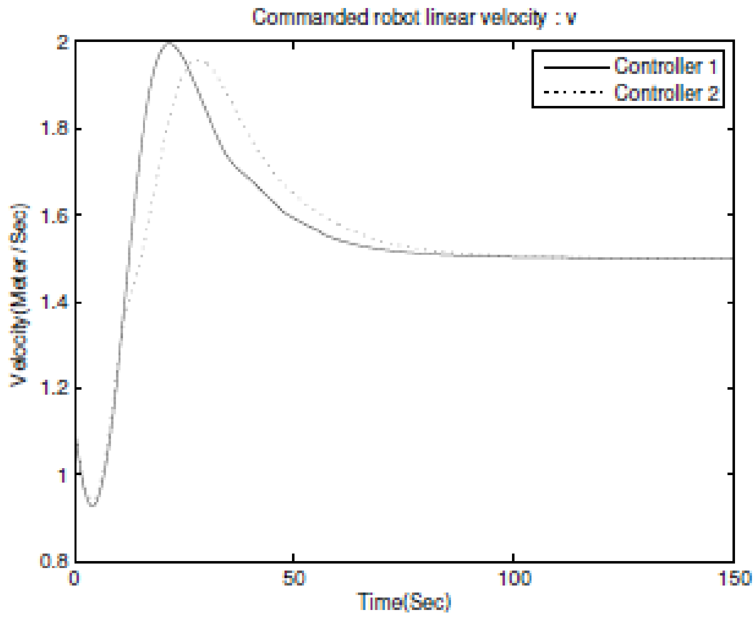
$$\lambda_v = 0.075,\ \lambda_\alpha = 0.15$$



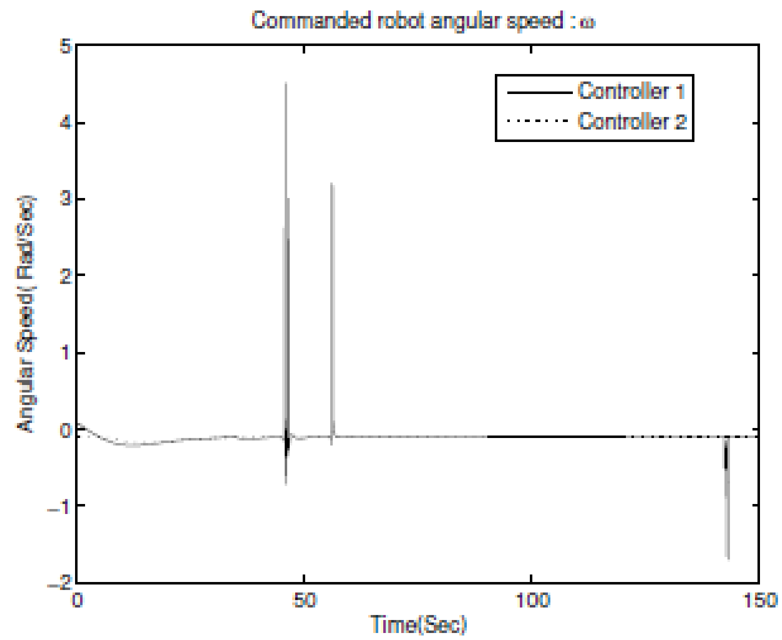
Linear tracking



Angular tracking

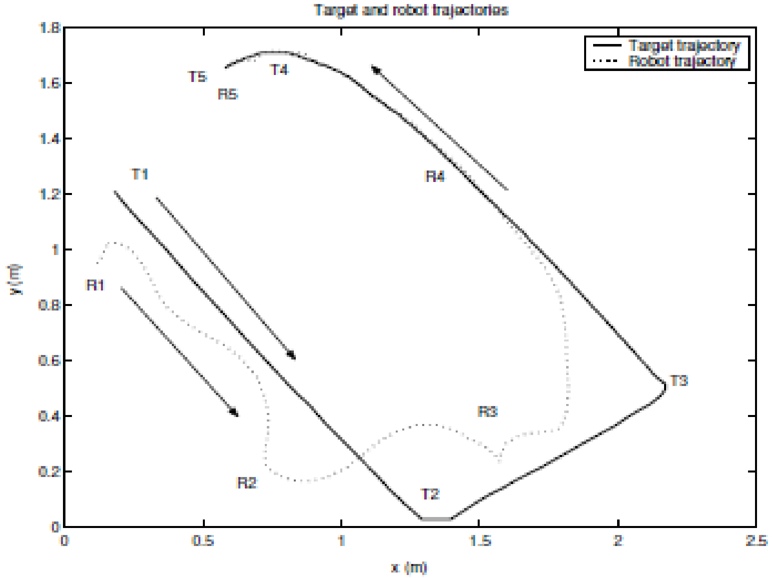


Linear velocity

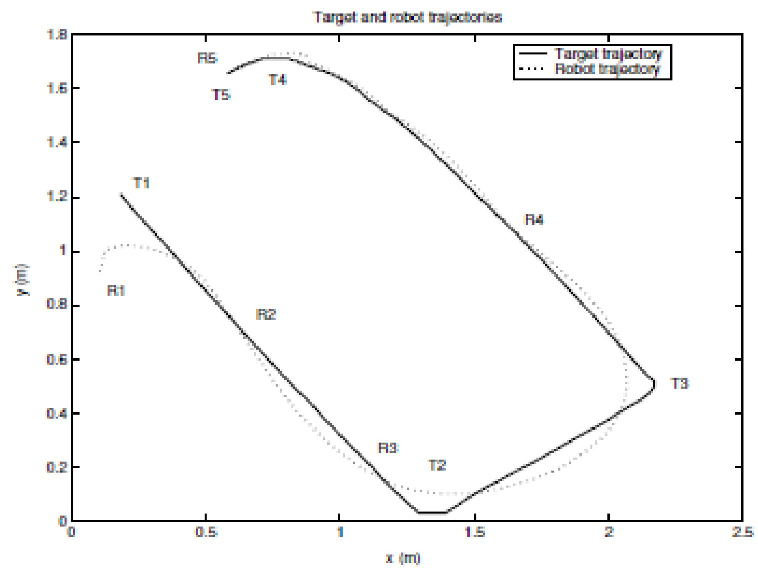


Angular velocity

Experiments

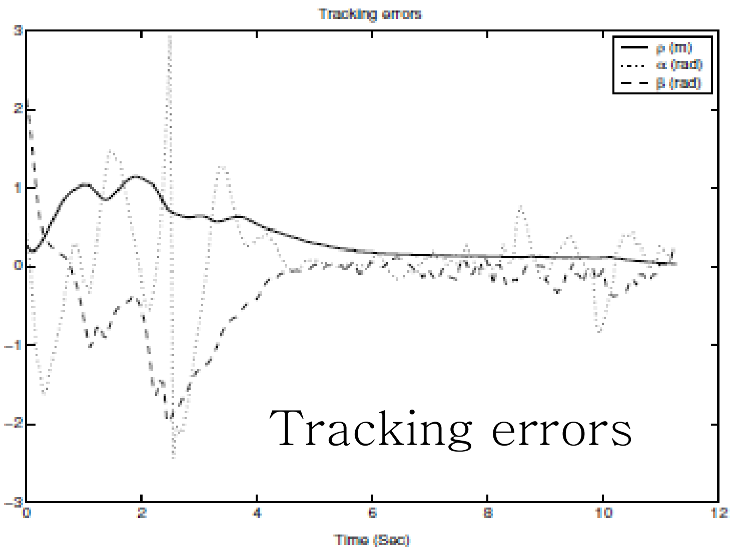


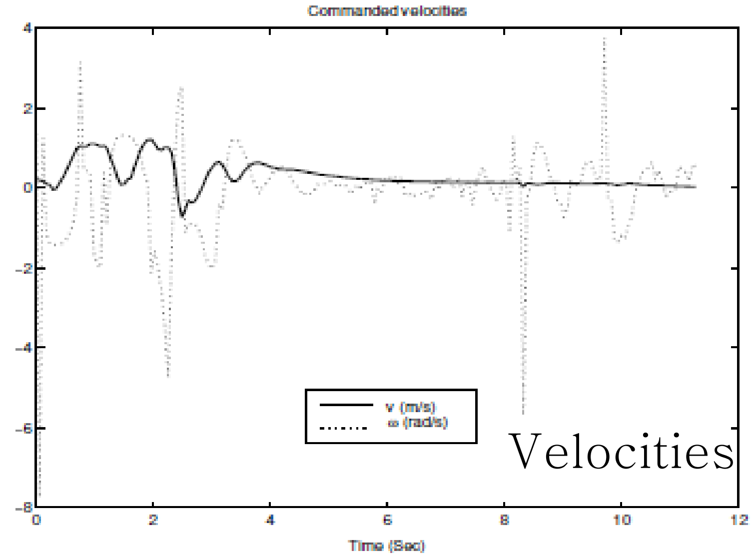
Robot trajectory under Controller 1



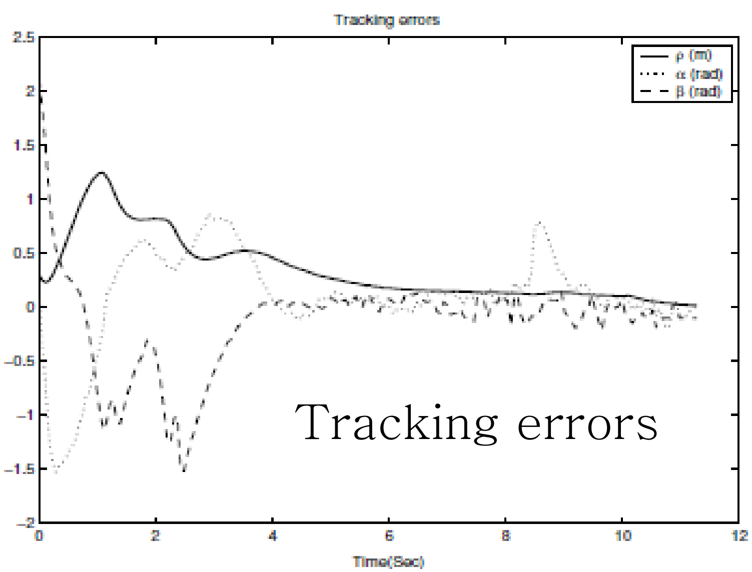
Robot trajectory under Controller 2

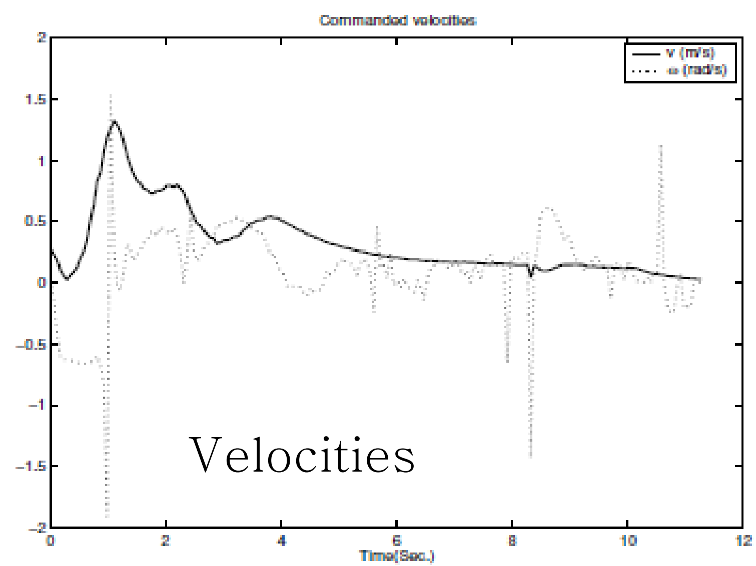
Under Controller 1:





Under Controller 2:





Demonstrations

Controller 1

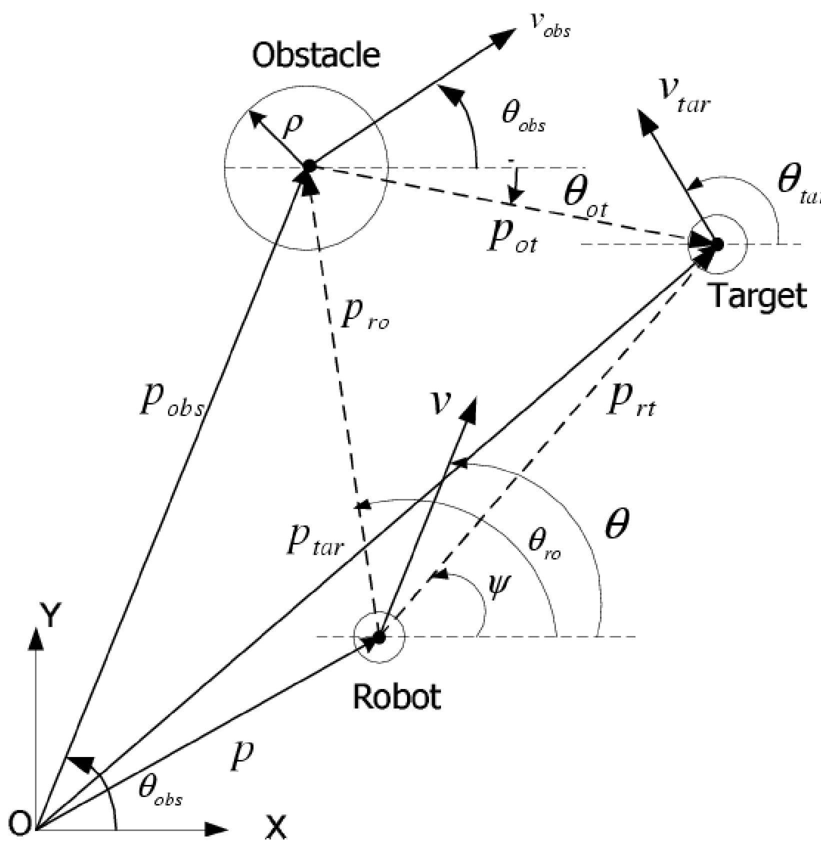
Controller 2

Conclusions:

- It is feasible to reduce the control efforts through
  - prioritization of control objectives
  - defining of Lyapunov function to reflect that priority
  - attenuation of controller outputs with some special functions of the system states (like sinusoidal functions etc.)

  while achieving the same or better control results in comparison with the conventional controllers
- The performance of the controller is affected by the noises of the sensors for state feedback (esp. velocity).

Potential field based control approach for robot's target tracking



System model:

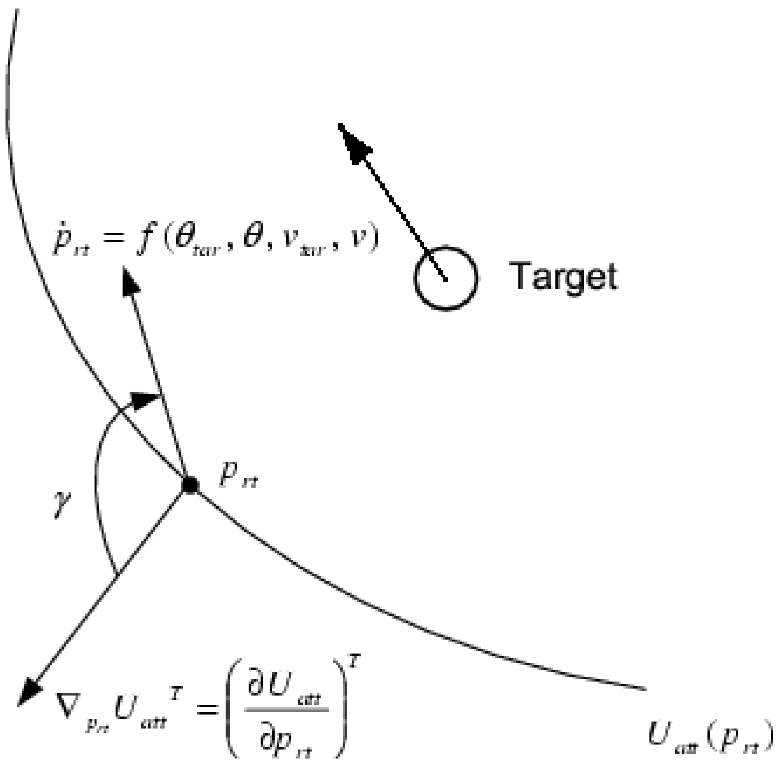
$$p_{rt} = [x_{rt} \quad y_{rt}]^T$$
$$\dot{x}_{rt} = v_{tar} \cos\theta_{tar} - v\cos\theta$$
$$\dot{y}_{tar} = v_{tar} \sin\theta_{tar} - v\sin\theta$$

Potential fields:

$$U = U_{att} + U_{rep}$$
$$U_{att} = \frac{1}{2}\xi_1 p_{rt}^T p_{rt}$$
$$U_{rep} = \begin{cases} \frac{1}{2}\xi_2(\rho^{-1} - \rho_0^{-1})^2, \text{if } \rho \le \rho_0 \\ 0 \qquad \text{else} \end{cases}$$

Case 1: Moving target free of obstacles

Minimization of the angle between the gradient of the field and the direction of robot motion relative to the target.



- Direction

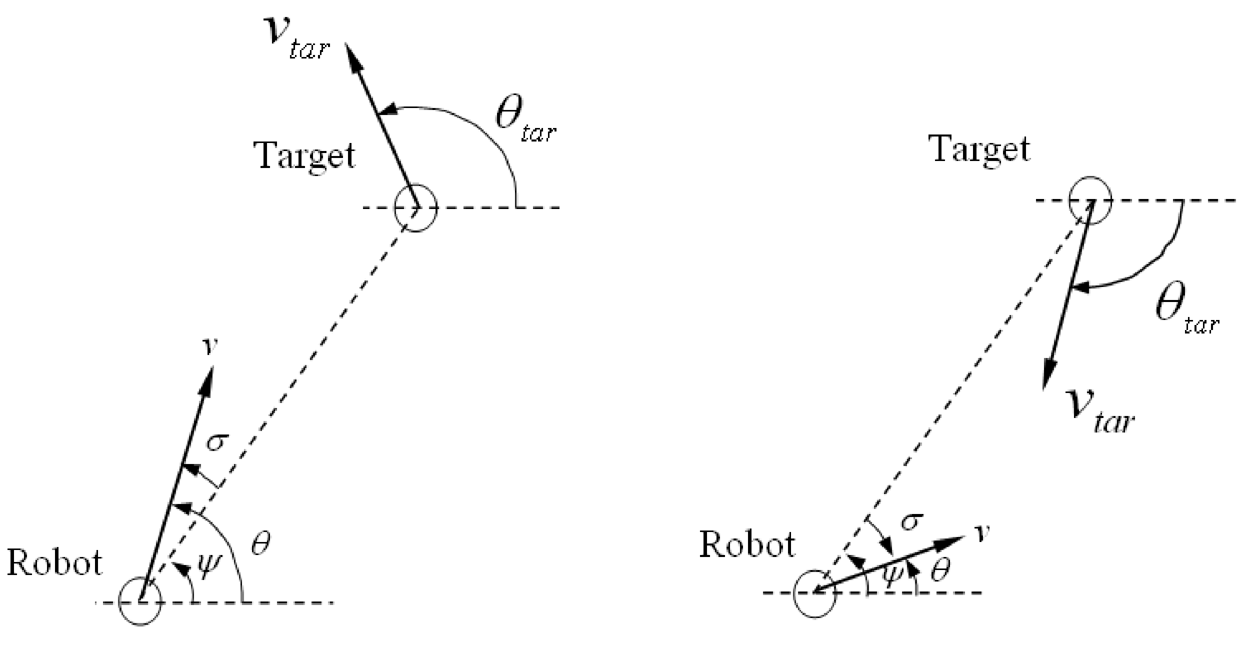
Minimisati on of $\overline{U}_{att} = \nabla_{p_{rt}} U_{att} \frac{\dot{p}_{rt}}{\|\dot{p}_{rt}\|}$

$$\frac{\partial \overline{U}_{att}}{\partial \theta} = \|\dot{p}_{rt}\|^{-3} \left(\frac{\partial U_{att}}{\partial x_{rt}} \dot{y}_{rt} - \frac{\partial U_{att}}{\partial y_{rt}} \dot{x}_{rt}\right)\left(\frac{\partial \dot{x}_{rt}}{\partial \theta} \dot{y}_{rt} - \frac{\partial \dot{y}_{rt}}{\partial \theta} \dot{x}_{rt}\right) = 0$$

$$\theta = \psi + \sigma$$

$$\sigma = \arcsin\left(\frac{v_{tar} \sin(\theta_{tar} - \psi)}{v}\right), \quad |\sigma| \le \frac{\pi}{2}$$

Robot direction is adjusted around the directional line pointing to the target



The target moves away from the robot *The target moves to the robot*

- **Speed** Intuitively $v \geq \left\| v_{tar} \sin(\theta_{tar} - \psi) \right\|$

It is chosen to decrease U_{att} , or

$$\dot{U}_{att} = \xi_1 p_{rt}^T \dot{p}_{rt} = \xi_1 \left\| p_{rt} \right\| (v_{tar} \cos(\theta_{tar} - \psi) - v \cos \sigma)$$

$$= \xi_1 \left\| p_{rt} \right\| (v_{tar} \cos(\theta_{tar} - \psi) - (v^2 - v_{tar}^2 \sin^2(\theta_{tar} - \psi))^{\frac{1}{2}}) < 0$$

One of the choices is:

$$v = (v_{tar}^2 + 2\lambda_1 v_{tar} \left\| p_{rt} \right\| \cos(\theta_{tar} - \psi) + \lambda_1^2 \left\| p_{rt} \right\|^2)^{\frac{1}{2}}$$

It leads to:

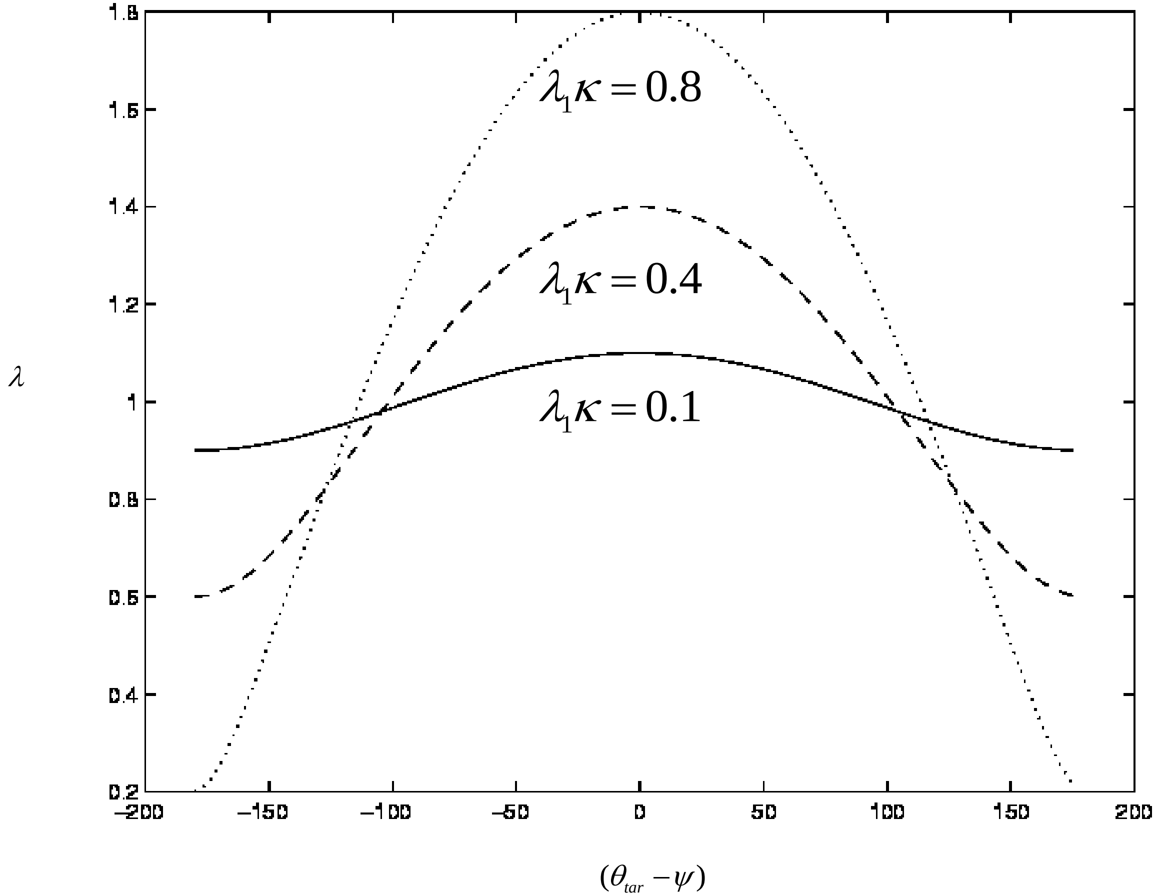
$$U_{att} = U_{att}(0) e^{-2\lambda_1 t} \rightarrow 0$$

$$\left\| p_{rt} \right\| \rightarrow 0$$

The speed determined by the relative linear distance, the target velocity and there directional relationship.

Comparison of the robot and target speeds:

$$\lambda = \frac{v}{v_{tar}} = (1 + 2\lambda_1 \kappa \cos(\theta_{tar} - \psi) + \lambda_1^2 \kappa^2)^{\frac{1}{2}}, \quad \kappa = \frac{\|p_{rt}\|}{v_{tar}}$$



The robot does not need to be always faster than the target (e.g. . when $\|(\theta_{tar} - \psi)\| > \frac{\pi}{2}$)

Case 2: Moving target with moving obstacles

The approach can be extended to solve the path/speed planning of the robot surrounded by multiple obstacles.

$$\theta = \overline{\psi} + \arcsin \frac{v_{tar} \sin(\theta_{tar} - \overline{\psi})}{v}$$

$$v = \sqrt{v_{tar} \cos(\theta_{tar} - \psi) - \sum_{i=1}^{n} \beta_i v_{obsi} \cos(\theta_{obsi} - \theta_{roi}) + \lambda_1 \left\| p_{rt} \right\|)^2 + v_{tar}^2 \sin^2(\theta_{tar} - \overline{\psi})}$$

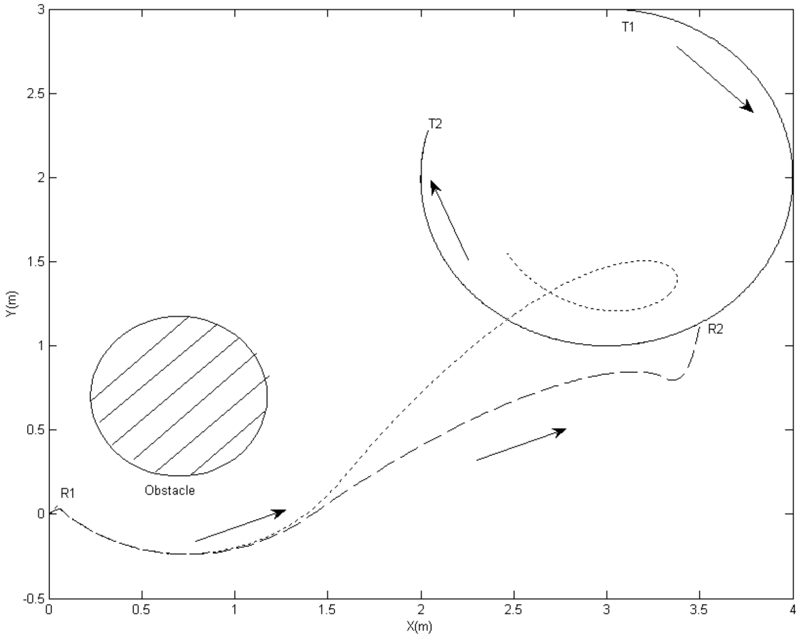
$$\overline{\psi} = \arctan \frac{\sin\psi - \sum_{i=1}^{n} \beta_i \sin\theta_{roi}}{\cos\psi - \sum_{i=1}^{n} \beta_i \cos\theta_{roi}}$$

$$\beta_i = \frac{\eta_i \left\| p_{roi} \right\|}{\xi_1 \left\| p_{rt} \right\|}, \quad \eta_i = \xi_2 \rho_i^{-2} \left\| p_{roi} \right\|^{-1} (\rho_i^{-1} - \rho_0^{-1})$$

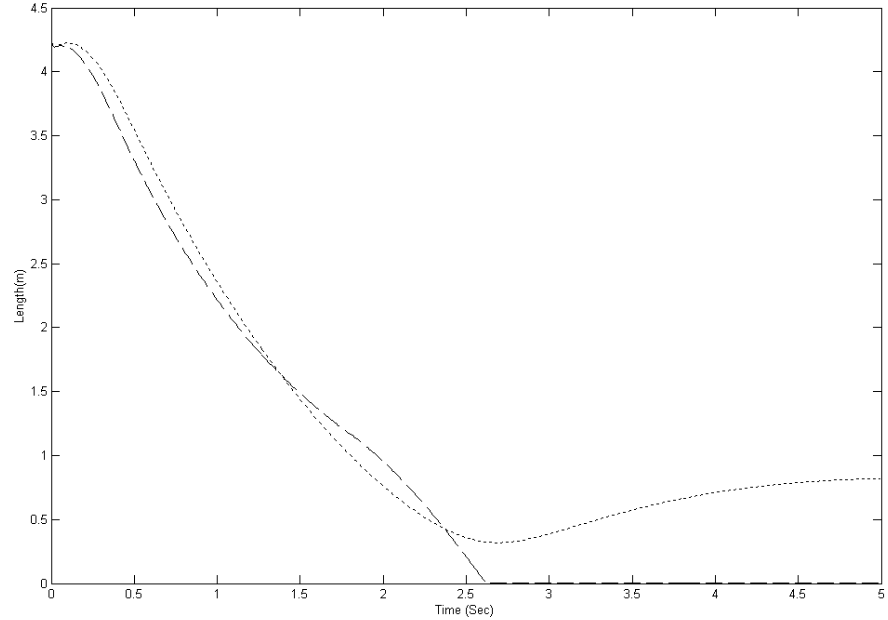
Simulation Results:

$$x_t = 3.0 + \sin t,\, y_t = 2.0 + \cos t$$

$$\theta_{tar} = -t, \quad v_{tar} = 1.0, \quad \lambda_1 = 1$$

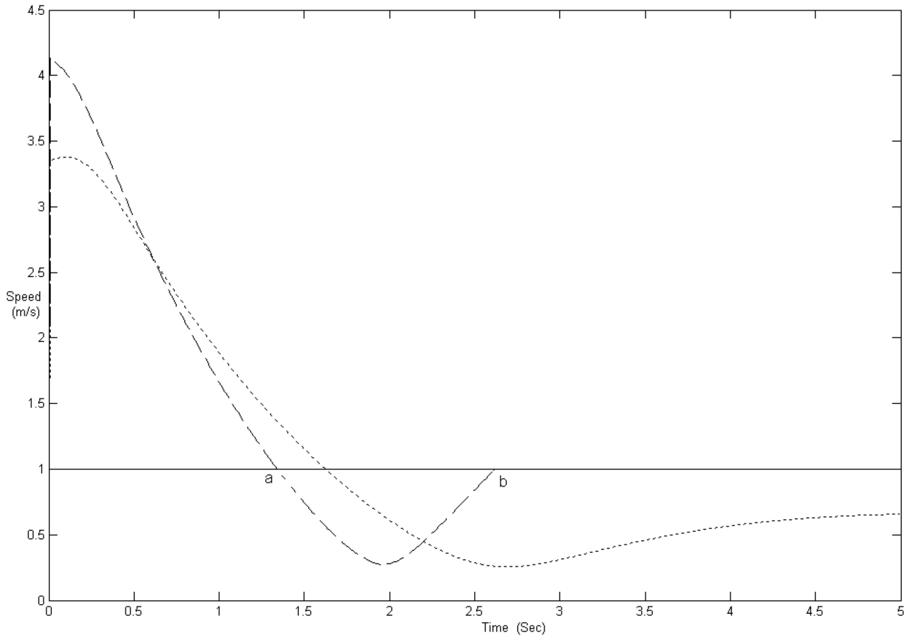


Trajectories

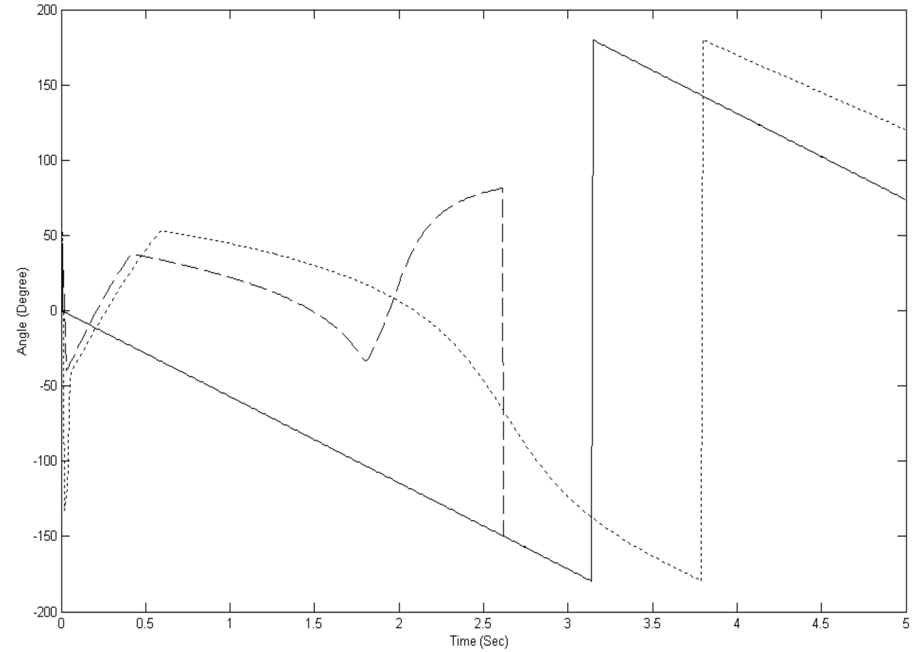


Relative Distance

Solid line: target
Dashed line : robot under the proposed controller
Dotted line :robot under the conventional potential field controller



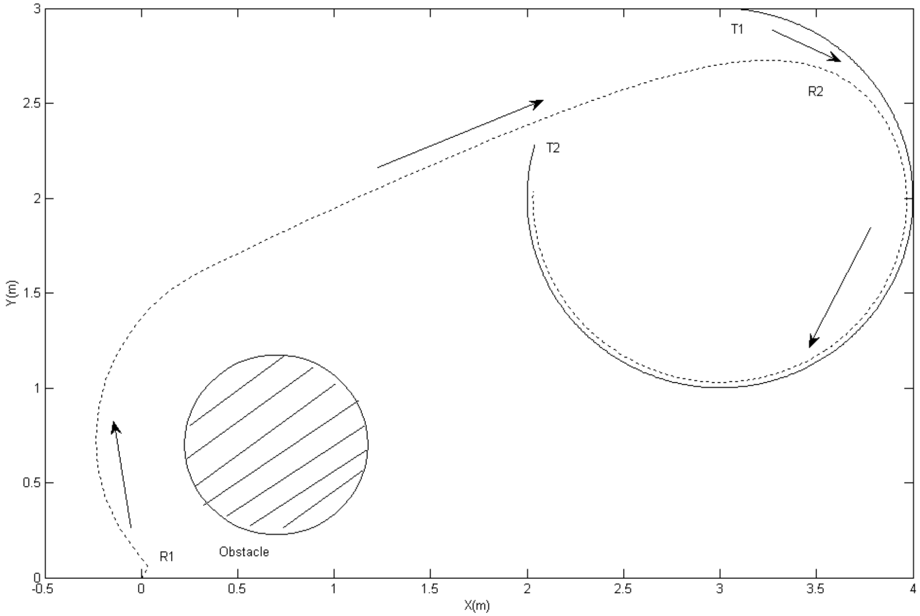
Speed



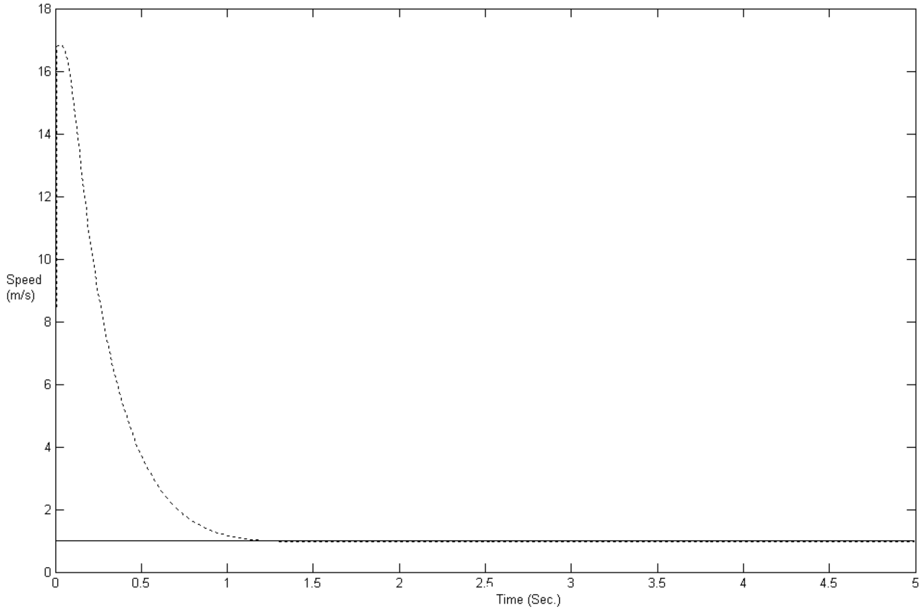
Angle

Solid line: target
Dashed line : robot under the proposed controller
Dotted line :robot under the conventional potential field controller

Performance of the conventional field method with a high gain



Trajectories



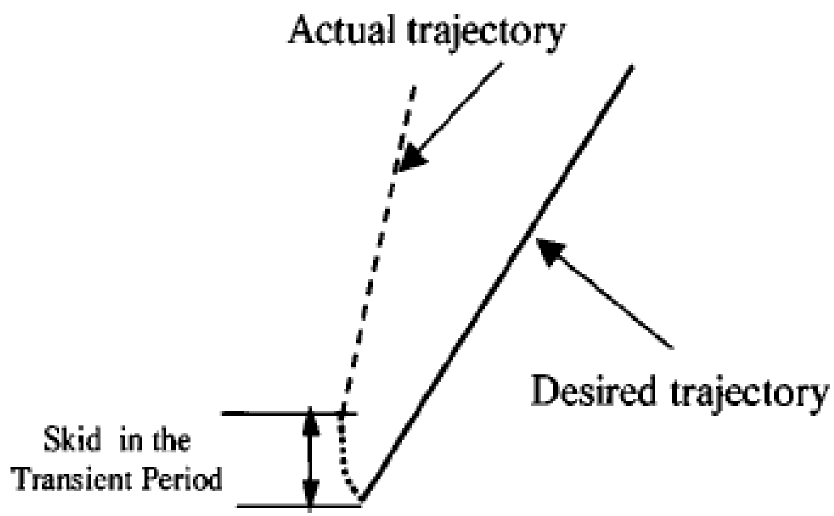
Speed

Conclusion:

- the speed as well as the direction of the robot motion are determined with potential field method
- the velocity of the moving target is taken into consideration
- the proposed approach maintains or improves tracking accuracy and reduce control efforts, in comparison to the traditional approaches
- further study on the determination of the optimum speed of the robot can be done by specifying additional performance requirements.

Speed control considering dynamic coupling between the actuators

- Synchronisation of the wheels' motion affects the robot's trajectory
- Coupling between the actuators needs to be considered



Model based adaptive control

Dynamic model: $M\dot{\omega} + \beta C(\omega) = \tau$

$$\omega = [\omega_r \quad \omega_l]^T$$

$$M = \begin{bmatrix} \frac{r^2}{4b^2}(mb^2 + I) + I_w & \frac{r^2}{4b^2}(mb^2 - I) \\ \frac{r^2}{4b^2}(mb^2 - I) & \frac{r^2}{4b^2}(mb^2 + I) + I_w \end{bmatrix}$$

$$C(\omega) = \begin{bmatrix} 0 & \omega_r - \omega_l \\ \omega_l - \omega_r & 0 \end{bmatrix} \qquad \beta = \frac{m_c d r^3}{4b^2}, \quad m = m_c + 2m_w$$

$$I = m_c d^2 + 2m_w b^2 + I_c + 2I_m$$

m_c, m_w, I_m, I_c are the inertia parameters of the robot and the wheels

b, d, r are the geometric parameters

Introducing new variables

$$\omega' = [\omega_1' \quad \omega_2']^T, \; \tau' = [\tau_1' \quad \tau_2']^T$$
$$\omega_1' = \omega_r + \omega_l, \; \omega_2' = \omega_r - \omega_l$$
$$\tau_1' = \tau_r' + \tau_l, \tau_2' = \tau_r' - \tau_l$$

then

$$\omega = T\omega', \quad \tau = T\tau', \; T = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Dynamic model is transformed to a more compact form :

$$M'\dot{\omega}' + \beta\omega_2' C'\omega' = \tau'$$

$$M' = T^{-1}MT = \begin{bmatrix} \alpha_1 & 0 \\ 0 & \alpha_2 \end{bmatrix}, \quad C' = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

$$\alpha_1 = \frac{mr^2}{2} + I_w, \; \alpha_2 = \frac{Ir^2}{2b^2} + I_w$$

Based on the transformed dynamic model, the adaptive speed controllers are derived:

$$\tau_r = \frac{k_1 + k_2}{2}(\omega_{rd} - \omega_r) + \frac{k_1 - k_2}{2}(\omega_{ld} - \omega_l) + \frac{1}{2}(\hat{\alpha}_1 \dot{\omega}'_{ld} + \hat{\alpha}_2 \dot{\omega}'_{2d}) + \hat{\beta}\omega'_2 \omega_{ld}$$

$$\tau_l = \frac{k_1 + k_2}{2}(\omega_{ld} - \omega_l) + \frac{k_1 - k_2}{2}(\omega_{rd} - \omega_r) + \frac{1}{2}(\hat{\alpha}_1 \dot{\omega}'_{ld} - \hat{\alpha}_2 \dot{\omega}'_{2d}) - \hat{\beta}\omega'_2 \omega_{rd}$$

$$\dot{\hat{\alpha}}_1 = -\gamma \dot{\omega}_{1d} e_1, \quad \dot{\hat{\alpha}}_2 = -\gamma \dot{\omega}_{2d} e_2, \quad \dot{\hat{\beta}} = -\gamma \omega'_2 (\omega'_2 \omega_{1d} - \omega'_1 \omega_{2d})$$

$$e_1 = \omega'_{1d} - \omega_1, \quad e_2 = \omega'_{2d} - \omega_2$$

Modified to reduce the amplitudes of the control outputs:

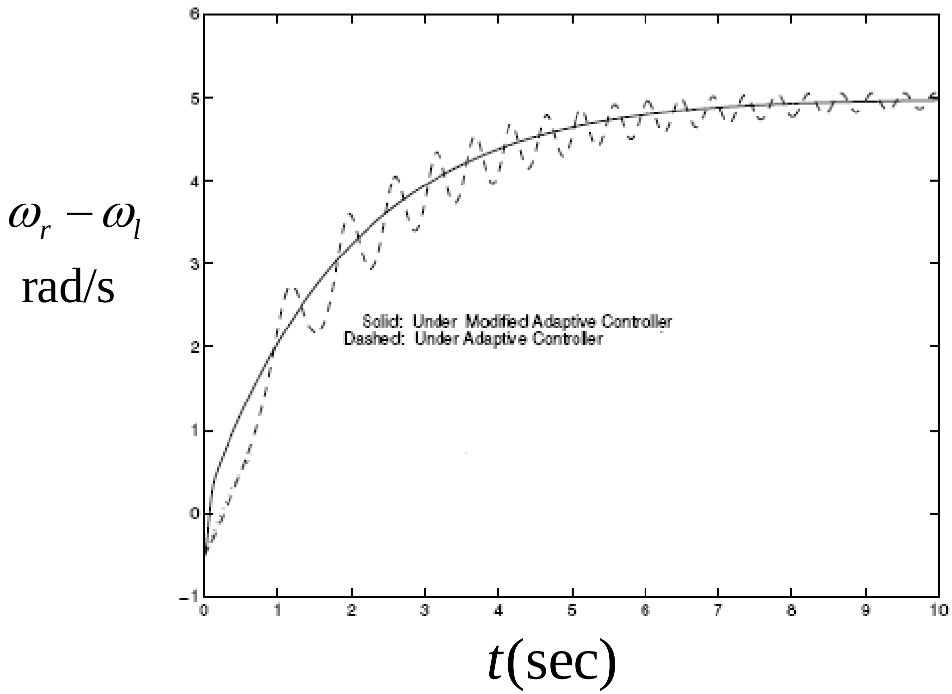
$$\tau_r = \frac{k_1 + k_2}{2}(\omega_{rd} - \omega_r) + \frac{k_1 - k_2}{2}(\omega_{ld} - \omega_l) + \frac{1}{2}(\hat{\alpha}_1 \dot{\omega}'_{ld} + \hat{\alpha}_2 \dot{\omega}'_{2d}) + k\hat{\beta}\omega_l$$

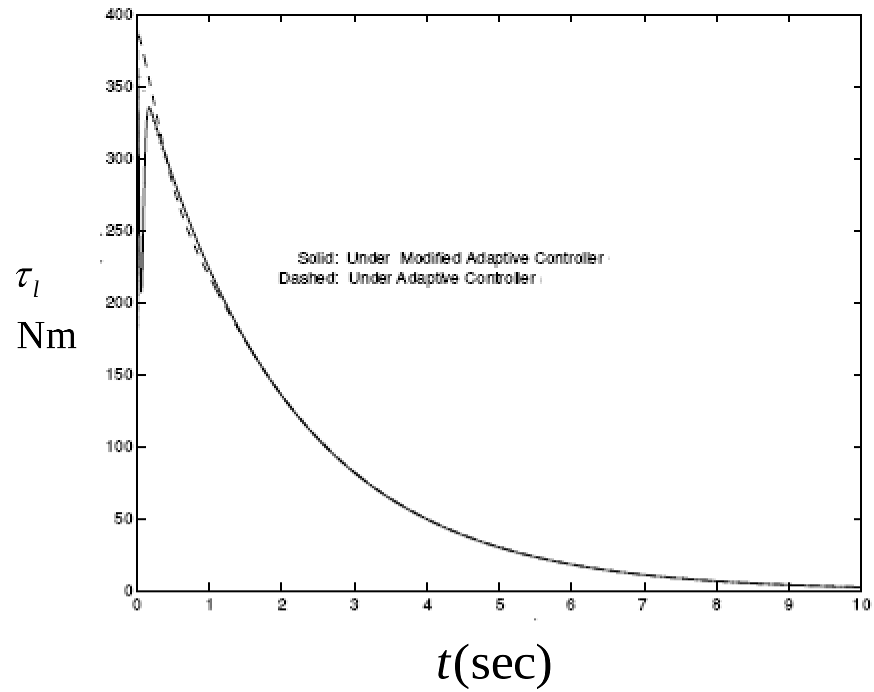
$$\tau_l = \frac{k_1 + k_2}{2}(\omega_{ld} - \omega_l) + \frac{k_1 - k_2}{2}(\omega_{rd} - \omega_r) + \frac{1}{2}(\hat{\alpha}_1 \dot{\omega}'_{ld} - \hat{\alpha}_2 \dot{\omega}'_{2d}) - k\hat{\beta}\omega_r$$

$$\dot{\hat{\beta}} = -k\gamma(\omega'_2 e_1 - \omega'_1 e_2)$$

$$k = \omega'_2 - \gamma_k(\omega'_2 e_1 - \omega'_1 e_2)$$

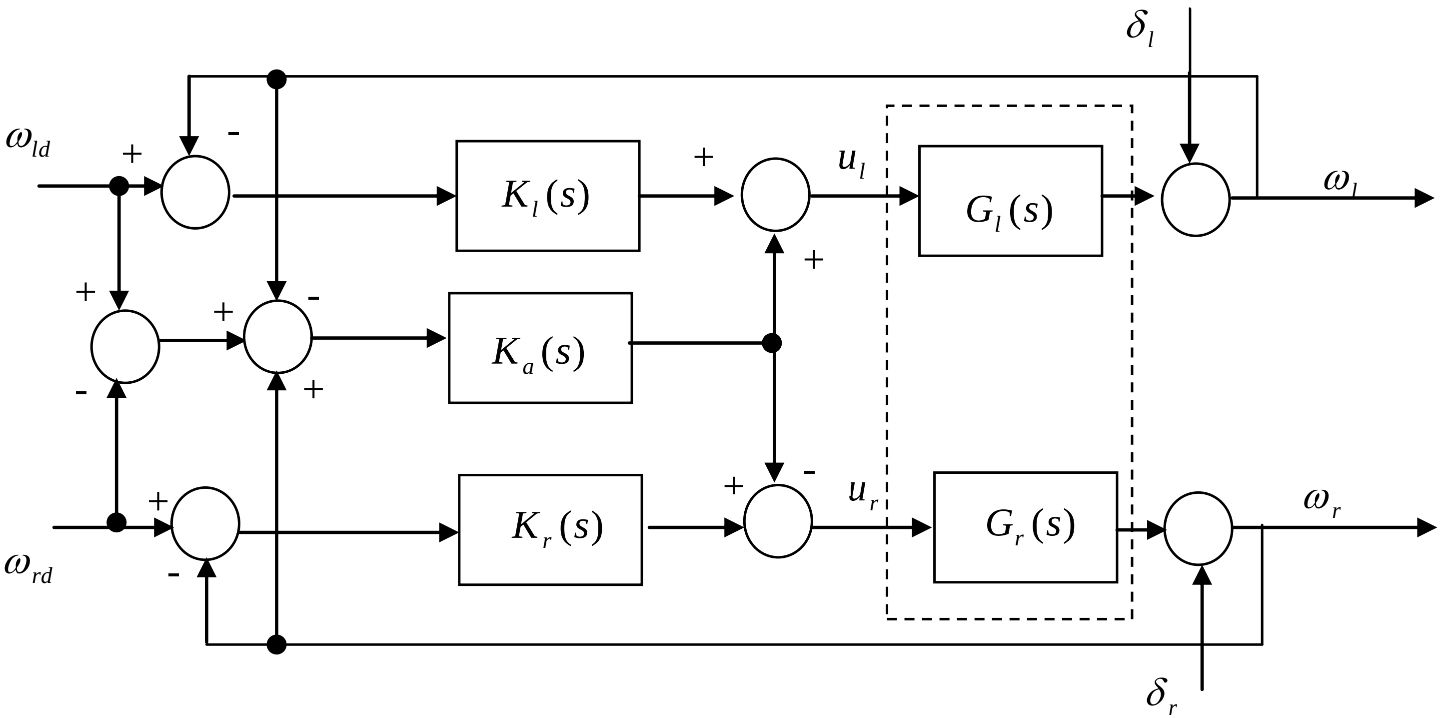
Simulation results





Model free PID control

A loop for the coupling of the wheels' speeds is added.



When $G_l(s) = G_r(s) = G(s), \quad K_l(s) = K_r(s) = K(s)$

Transfer functions :

$$G_{ind}(S) = \frac{G(s)K(s)(K(s) + 2K_s(s))}{G(s)K(s)(K(s) + 2K_a(s)) + K(s) + K_a(s)},$$

$$G_{ind}(S) = \frac{K_s(s)}{G(s)K(s)(K(s) + 2K_a(s)) + K(s) + K_a(s)},$$

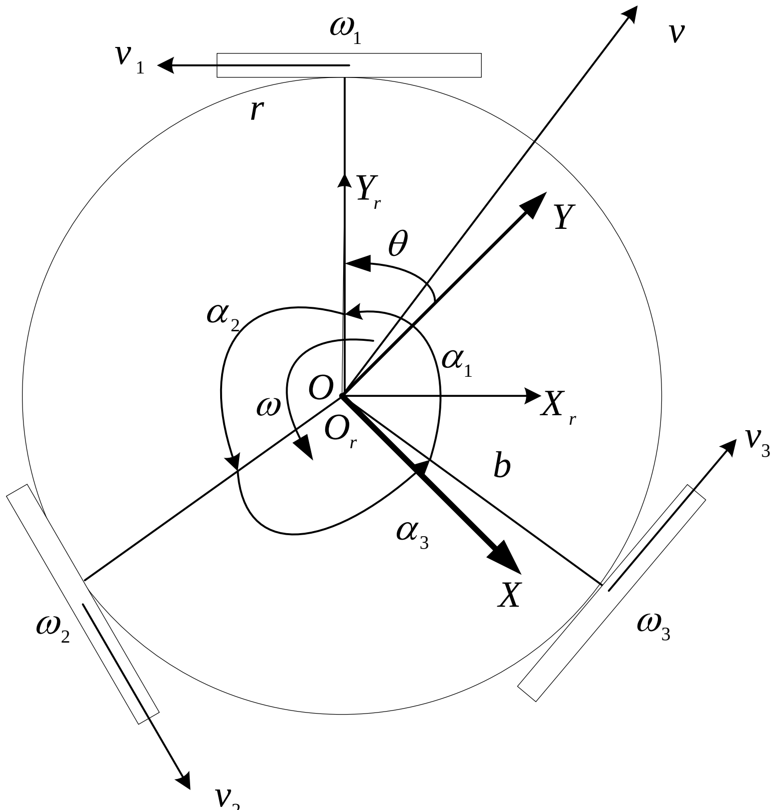
$$\omega_l(s) = G_{ind}(s)\omega_{ld}(s) - G_{syn}(s)\omega_r(s)$$

$$\omega_r(s) = G_{ind}(s)\omega_{rd}(s) - G_{syn}(s)\omega_l(s)$$

- First order motor model is adopted: $G(s) = \frac{k_m}{1 + \tau_m s}, \ \tau_m = JR_a / K_t^2$
- PID controller is used for the speed control
- Implemented with one PIC18F252 microcontroller

Speed Control of an Omni-wheel robots

Modeling (Kinematics)



Omni Wheel Robot

Inverse kinematic model:

$$r\omega_i = b\omega + v_r v_i \quad (i = 1, 2, 3)$$

$$v_1 = [-1 \;\; 0]^T$$

$$v_2 = [\cos\frac{\pi}{3} \quad -\sin\frac{\pi}{3}]^T$$

$$v_3 = [\cos\frac{\pi}{3} \quad \sin\frac{\pi}{3}]^T$$

$$\omega_1 = r^{-1}(b\omega - v_{rx})$$

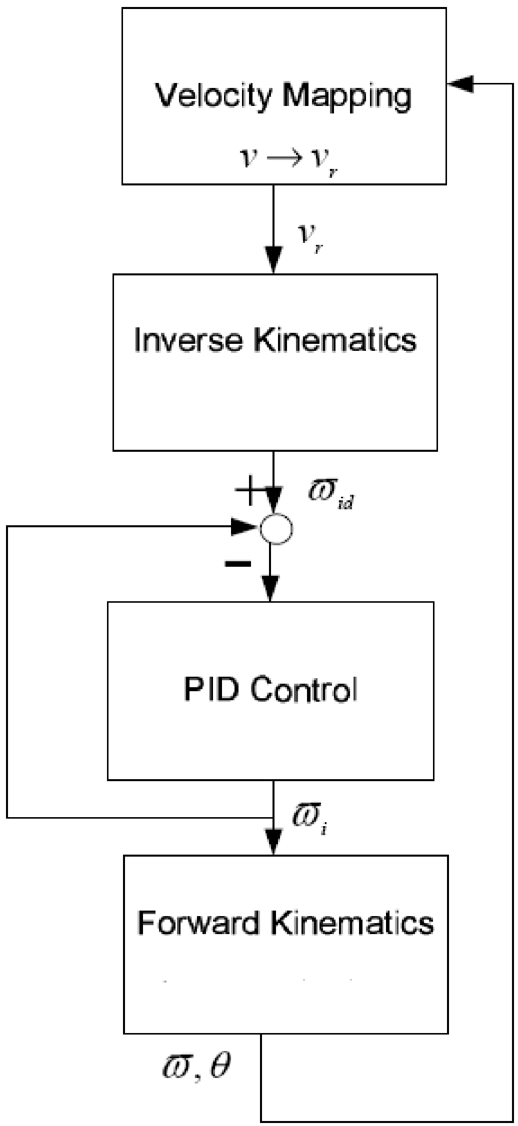
$$\omega_2 = r^{-1}(b\omega + v_{rx}\cos\frac{\pi}{3} - v_{ry}\sin\frac{\pi}{3})$$

$$\omega_3 = r^{-1}(b\omega + v_{rx}\cos\frac{\pi}{3} + v_{ry}\sin\frac{\pi}{3})$$

$$v_r = [v_{rx} \quad v_{ry}]^T$$

$$v_{rx} = v_x\cos\theta + v_y\sin\theta$$

$$v_{ry} = -v_x\sin\theta + v_y\cos\theta$$

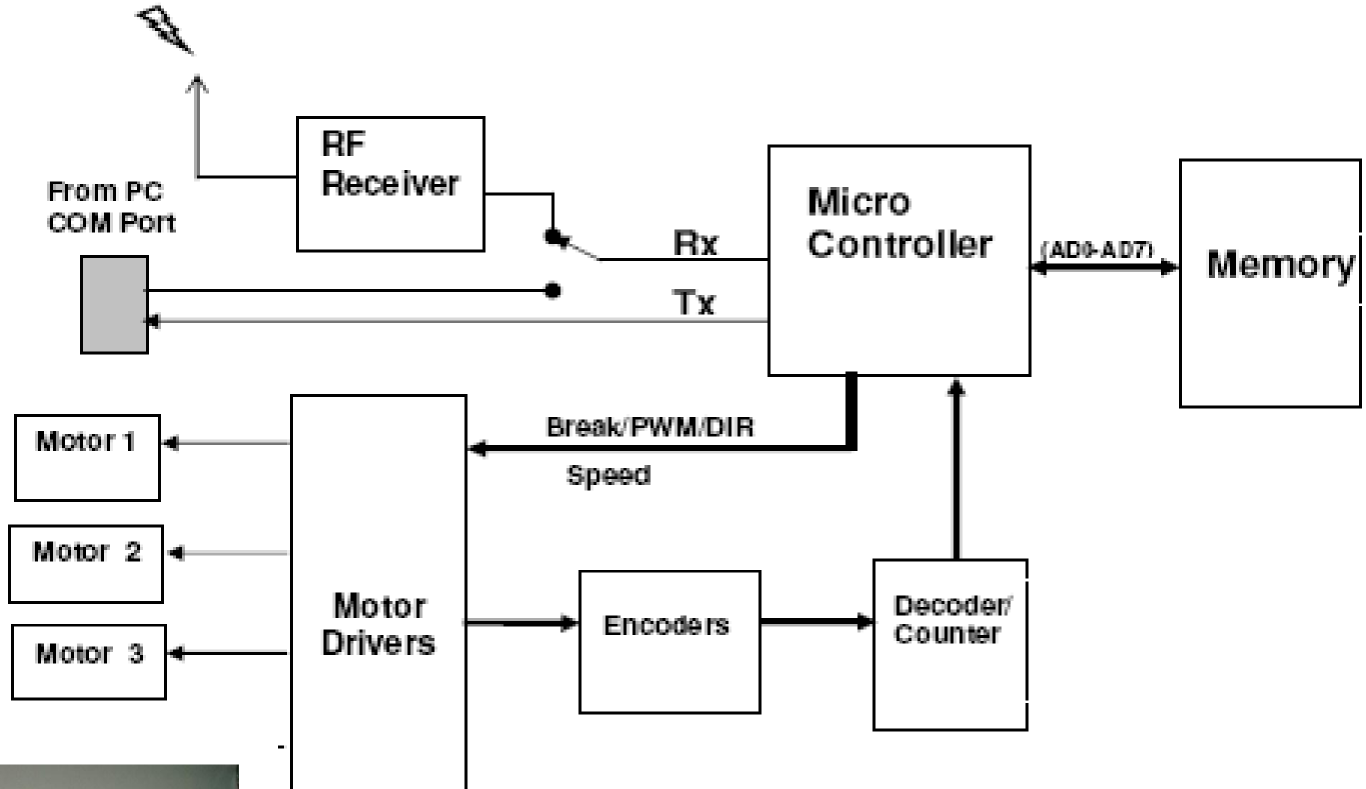


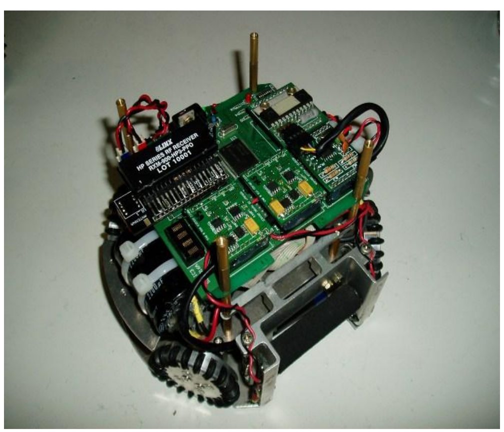
- Chooped fed motors with drivers to drive the wheels
- PID controller implemented with one one 80296 microcontrollers (three PWM outputs)
- Encoder resolution 512 ppr
- Sampling time 1 ms
- Control loop completed within 0.5ms

This is achieved through:

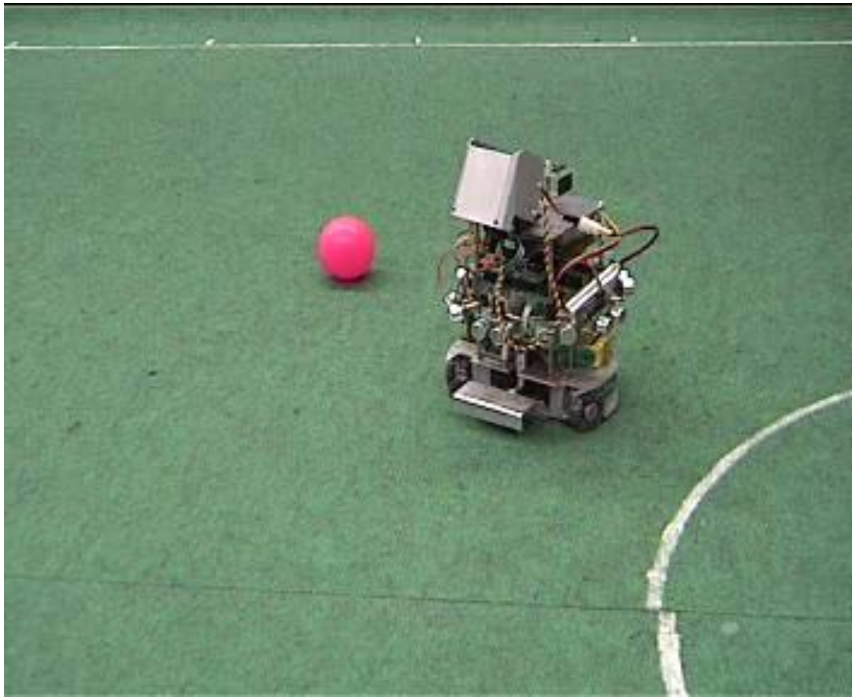
- codes written in an assembly language without using floating point libraries (too slow)
- fixed point notation and a look up table of whole numbers to represent a floating point number with reasonable accuracy
- only the simple operations like addition, substration, multiplication and bits-shifting are used.

Implementation





Demonstrations



Conclusion

- Lyapunov and potential field based target tracking controllers, and speed controller for dynamically coupled wheels for mobile robots were presented
- Both position and velocity of the target were considered in the target tracking controller design
- Functions of the system states, especially those of the target, are are designed to moderate the magnitude or fluctuation of the control effort
- The states of the system were assumed to be available; sensor noises affect the performance of the controller.
- To get a good system states estimation and prediction from the sensor data is another big issue to be addressed together with the controller design (*Kalman filtering, Bayesian method* etc.)
- Further study can be undertaken on integrating open-loop optimal control, closed-loop control and system states estimation and prediction