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
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)?
Differential wheel driven robot (no-holonomic):

- Robot’s posture (Cartesian coordinates) cannot be stabilized 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)
  \[
  \dot{x} \sin \theta - \dot{y} \cos \theta = 0
  \]

- Nonholonomic (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 $\omega$

to make $x \rightarrow x_d$, $y \rightarrow y_d$
It can be proved (due to Lyapunov and Barbalat), the following control can meet the objective:

\[ v = v_d \cos(\theta_d - \theta) + k_1[\cos\theta(x_d - x) + \sin\theta(y_d - y)] \]
\[ \omega = \omega_d + k_2 \text{sgn}(v_d)[\sin\theta(x_d - x) - \cos\theta(y_d - y)] + k_3(\theta_d - \theta) \]

\[ v_d = \pm \sqrt{\dot{x}_d^2 + \dot{y}_d^2} \quad \text{Desired linear velocity (along the trajectory)} \]
\[ \omega_d = \frac{\ddot{y}_d \dot{x}_d - \dot{y}_d \dot{x}_d}{\dot{x}_d^2 + \dot{y}_d^2} \quad \text{Desired angular velocity} \]
\[ \theta_d = \text{ATAN2}(\dot{y}_d, \dot{x}_d) + k\pi \quad \text{Desired direction} \]

\[ k_1 = k_3 = 2\xi \sqrt{\omega_d^2 + b\dot{v}_d^2}, \quad k_2 = b|v_d| \]

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: \((x_p, y_p, \theta_p)\) to the target posture \((x_g, y_g, \theta_g)\) \((\theta_g = 0: \text{parallel parking})\).
The system model described in polar coordinates:

\[
\begin{align*}
\dot{\rho} &= -v \cos \gamma, \quad \dot{\gamma} = -\omega + v \frac{\sin \gamma}{\rho}, \quad \dot{\delta} = v \frac{\sin \gamma}{\rho} \\
\rho &= \sqrt{(x_g - x)^2 + (y_g - y)^2} \\
\delta &= \tan^{-1}\left(\frac{y_g - y}{x_g - x}\right) \\
\gamma &= \delta - \theta
\end{align*}
\]

The model is singular at \( \rho = 0 \)
Let \( v = k_1 \rho \cos \gamma \)  
\[ \omega = k_2 a_2 \gamma + \frac{k_1 \sin 2\gamma}{2a_2 \gamma} (a_2 \gamma + a_3 \delta) \]

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

\[ \rho \to 0, \gamma \to 0, \delta \to 0 \]

with the Lyapunov function candidate

\[ V = \frac{1}{2} a_1 \rho^2 + \frac{1}{2} a_2 \gamma^2 + \frac{1}{2} a_3 \delta^2, a_1, a_2, a_3 > 0 \]

\[ \dot{V} = -a_1 k_1 \cos^2 \gamma \rho^2 - k_2 a_2 \gamma^2 \leq 0 \]

- 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:

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

\[ U_{rep} = \begin{cases} \frac{1}{2} \xi_2 (\rho^{-1} - \rho_0^{-1})^2, & \text{if } \rho \leq \rho_0 \\ 0, & \text{else} \end{cases} \]

Robot move along the negative gradient of the combined field:

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

\[ = \begin{cases} \xi_1 p_{rt} + \xi_2 (\rho^{-1} - \rho_0^{-1}) \rho^{-2} \nabla_p \rho, & \text{if } \rho \leq \rho_0 \\ \xi_1 p_{rt}, & \text{else} \end{cases} \]

- 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):

\[
\begin{align*}
\dot{\rho} &= v_t \cos \beta - v \cos \alpha \\
\dot{\alpha} &= v \frac{\sin \alpha}{\rho} - v_t \frac{\sin \beta}{\rho} - \omega \\
\dot{\beta} &= v \frac{\sin \alpha}{\rho} - v_t \frac{\sin \beta}{\rho} - \dot{\phi}_t, \quad \rho \neq 0
\end{align*}
\]

\[\alpha = \theta - \phi, \quad \beta = \theta - \dot{\phi}_t\]
Controller 1: Extension of the general control approach

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

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

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

\[ \alpha \rightarrow 0, \rho \rightarrow 0 \text{ and } \beta \rightarrow 0 \]

(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:

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

\[ \omega = \frac{\alpha}{2\alpha - \beta} \left( \lambda_\alpha \alpha + \frac{v_t}{\rho} \left( \frac{\sin 2\alpha}{2} \cos \beta - \sin \beta \right) + \frac{\sin 2\alpha}{2} \lambda_v \right) + \frac{\alpha - \beta}{2\alpha - \beta} \phi_t \]

which can also achieve the convergence of the tracking errors, but with less control efforts
Comparison: control efforts of Controllers 1 and Controller 2

\[ \omega_1 = \lambda_\alpha \alpha + \frac{\alpha + \beta}{\rho} \left( \frac{\sin 2\alpha}{2\alpha} \cos \beta - \frac{\sin \beta}{\alpha} \right) v_t - \frac{\beta}{\alpha} \phi_t + \frac{\sin 2\alpha}{2\alpha} \lambda_v (\alpha + \beta) \]

\[ \omega_2 = \frac{\alpha}{2\alpha - \beta} \left( \lambda_\alpha \alpha + \frac{v_t}{\rho} \left( \frac{\sin 2\alpha}{2} \cos \beta - \sin \beta \right) + \frac{\sin 2\alpha}{2} \lambda_v \right) + \frac{\alpha - \beta}{2\alpha - \beta} \phi_t \]

or

\[ \omega_1 = \xi - \gamma_1 \eta - \lambda_\alpha \beta \]

\[ \omega_2 = \xi - \gamma_2 \eta \]

\[ \xi = \lambda_\alpha \alpha + \frac{v_t}{\rho} \left( \frac{\sin 2\alpha}{2} \cos \beta - \sin \beta \right) + \frac{\sin 2\alpha}{2} \lambda_v \]

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

\[ k = \frac{\beta}{\alpha} \]
• By observation, the magnitude of controller 2 is less than that of controller 1.

• Analysing the factors affecting the controller magnitude, it is obvious that, except for the region near \( k = \frac{\beta}{\alpha} = 2 \), the magnitude affecting Controller 1 is larger in magnitude than that affecting Controller 2.
Simulation Results (tracking a target Moving along a circle)

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

\[ \lambda_v = 0.075, \quad \lambda_\alpha = 0.15 \]
Linear velocity

Angular velocity
Experiments

Robot trajectory under Controller 1

Robot trajectory under Controller 2
Under Controller 1:

Tracking errors

Velocities

Under Controller 2:

Tracking errors

Velocities
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:

\[ p_{rt} = \begin{bmatrix} x_{rt} \\ y_{rt} \end{bmatrix}^T \]

\[ \dot{x}_{rt} = v_{tar} \cos \theta_{tar} - v \cos \theta \]

\[ \dot{y}_{tar} = v_{tar} \sin \theta_{tar} - v \sin \theta \]

Potential fields:

\[ U = U_{att} + U_{rep} \]

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

\[ U_{rep} = \begin{cases} 
1/2 \xi_2 (\rho^{-1} - \rho_0^{-1})^2, & \text{if } \rho \leq \rho_0 \\
0, & \text{else}
\end{cases} \]
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

Minimisation of

\[ \overline{U}_{att} = \nabla_{\dot{p}_n} U_{att} \frac{\ddot{p}_n}{\| \dot{p}_n \|} \]

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

\[ \theta = \psi + \sigma \]

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

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

\[ v \geq \| v_{\text{tar}} \sin(\theta_{\text{tar}} - \psi) \| \]

Intuitively

\[ v = (v_{\text{tar}}^2 + 2 \lambda_1 v_{\text{tar}} \| p_{rt} \| \cos(\theta_{\text{tar}} - \psi) + \lambda_1^2 \| p_{rt} \|^2)^{\frac{1}{2}} \]

It leads to:

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

\[ \| p_{rt} \| \rightarrow 0 \]

The speed determined by the relative linear distance, the target velocity and their directional relationship.
Comparison of the robot and target speeds:

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

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.

\[
\theta = \bar{\psi} + \arcsin \frac{v_{\text{tar}} \sin(\theta_{\text{tar}} - \bar{\psi})}{v}
\]

\[
v = \sqrt{v_{\text{tar}} \cos(\theta_{\text{tar}} - \psi) - \sum_{i=1}^{n} \beta_i v_{\text{obsi}} \cos(\theta_{\text{obsi}} - \theta_{\text{roi}}) + \lambda_1 \| p_r \|^2 + v_{\text{tar}}^2 \sin^2(\theta_{\text{tar}} - \bar{\psi})}
\]

\[
\bar{\psi} = \arctan \left( \frac{\sin \psi - \sum_{i=1}^{n} \beta_i \sin \theta_{\text{roi}}}{\cos \psi - \sum_{i=1}^{n} \beta_i \cos \theta_{\text{roi}}} \right)
\]

\[
\beta_i = \frac{\eta_i \| p_{\text{roi}} \|}{\xi_1 \| p_r \|}, \quad \eta_i = \xi_2 \rho_i^{-2} \| p_{\text{roi}} \|^{-1} (\rho_i^{-1} - \rho_0^{-1})
\]
Simulation Results: \[ x_t = 3.0 + \sin t, \quad y_t = 2.0 + \cos t \]
\[ \theta_{tar} = -t, \quad v_{tar} = 1.0, \quad \lambda_1 = 1 \]

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

![Diagram showing actual and desired trajectory with skid in the transient period]
Model based adaptive control

Dynamic model: \[ M\ddot{\omega} + \beta C(\omega) = \tau \]

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

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

\[ C(\omega) = \begin{bmatrix} 0 & \omega_r - \omega_l \\ \omega_l - \omega_r & 0 \end{bmatrix} \]

\[ \beta = \frac{m_c dr^3}{4b^2}, \quad m = m_c + 2m_w \]

\[ I = m_c d^2 + 2m_w b^2 + I_c + 2I_m \]

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

\[ b, d, r \] are the geometric parameters
Introducing new variables

\[ \omega' = [\omega'_1 \quad \omega'_2]^T, \quad \tau' = [\tau'_1 \quad \tau'_2]^T \]

\[ \omega'_1 = \omega_r + \omega_l, \quad \omega'_2 = \omega_r - \omega_l \]

\[ \tau'_1 = \tau'_r + \tau_l, \quad \tau'_2 = \tau'_r - \tau_l \]

then

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

Dynamic model is transformed to a more compact form:

\[ M'\dot{\omega'} + \beta \omega'_2 C'\omega' = \tau' \]

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

\[ \alpha_1 = \frac{mr^2}{2} + I_w, \quad \alpha_2 = \frac{Ir^2}{2b^2} + I_w \]
Based on the transformed dynamic model, the adaptive speed controllers are derived:

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

\[
\tau_i = \frac{k_1 + k_2}{2} (\omega_{ld} - \omega_i) + \frac{k_1 - k_2}{2} (\omega_{rd} - \omega_r) + \frac{1}{2} (\hat{\alpha}_1 \dot{\omega}_{ld} - \hat{\alpha}_2 \dot{\omega}_{2d}) - \hat{\beta} \omega'_2 \omega_{rd}
\]

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

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

Modified to reduce the amplitudes of the control outputs:

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

\[
\tau_i = \frac{k_1 + k_2}{2} (\omega_{ld} - \omega_i) + \frac{k_1 - k_2}{2} (\omega_{rd} - \omega_r) + \frac{1}{2} (\hat{\alpha}_1 \dot{\omega}_{ld} - \hat{\alpha}_2 \dot{\omega}_{2d}) - k \hat{\beta} \omega_r
\]

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

\[
k = \omega'_2 - \gamma_k (\omega'_2 e_1 - \omega'_1 e_2)
\]
Simulation results

\[ \omega_r - \omega_l \]
rad/s

\[ t(\text{sec}) \]

\[ \tau_l \]
Nm

\[ t(\text{sec}) \]
Model free PID control

A loop for the coupling of the wheels’ speeds is added.
When $G_l(s) = G_r(s) = G(s)$, $K_l(s) = K_r(s) = K(s)$

Transfer functions:

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

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

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

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

- 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)
Inverse kinematic model:

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

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

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

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

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

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

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

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

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

\[ v_{ry} = -v_x \sin \theta + v_y \cos \theta \]
- Chooped fed motors with drivers to drive the wheels
- PID controller implemented with one 80296 microcontrollers (three PWM outputs)
- Encoder resolution 512 ppr
- Sampling time 1 ms
- Control loop completed within 0.5 ms

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, substraction, 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 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.