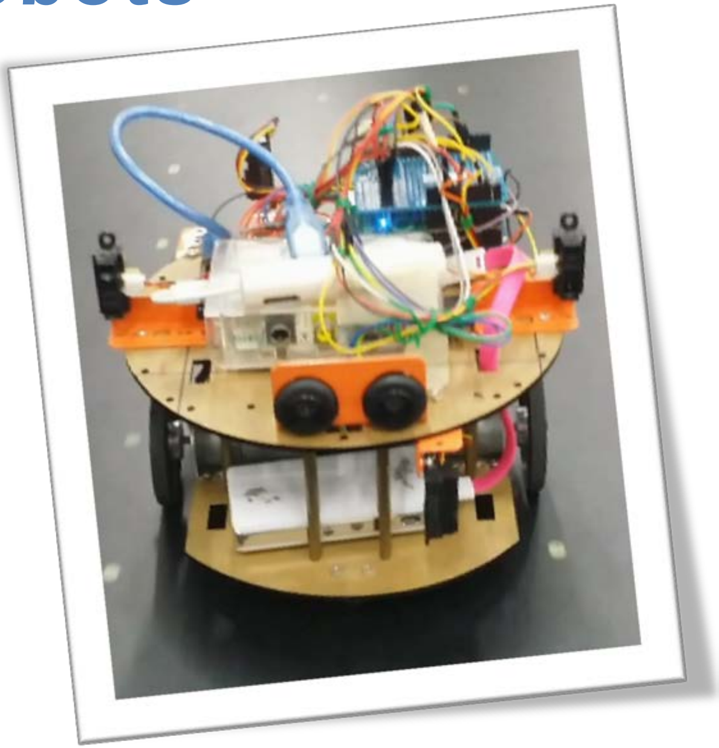
Implementing PID Controller for 2-wheel drive robots

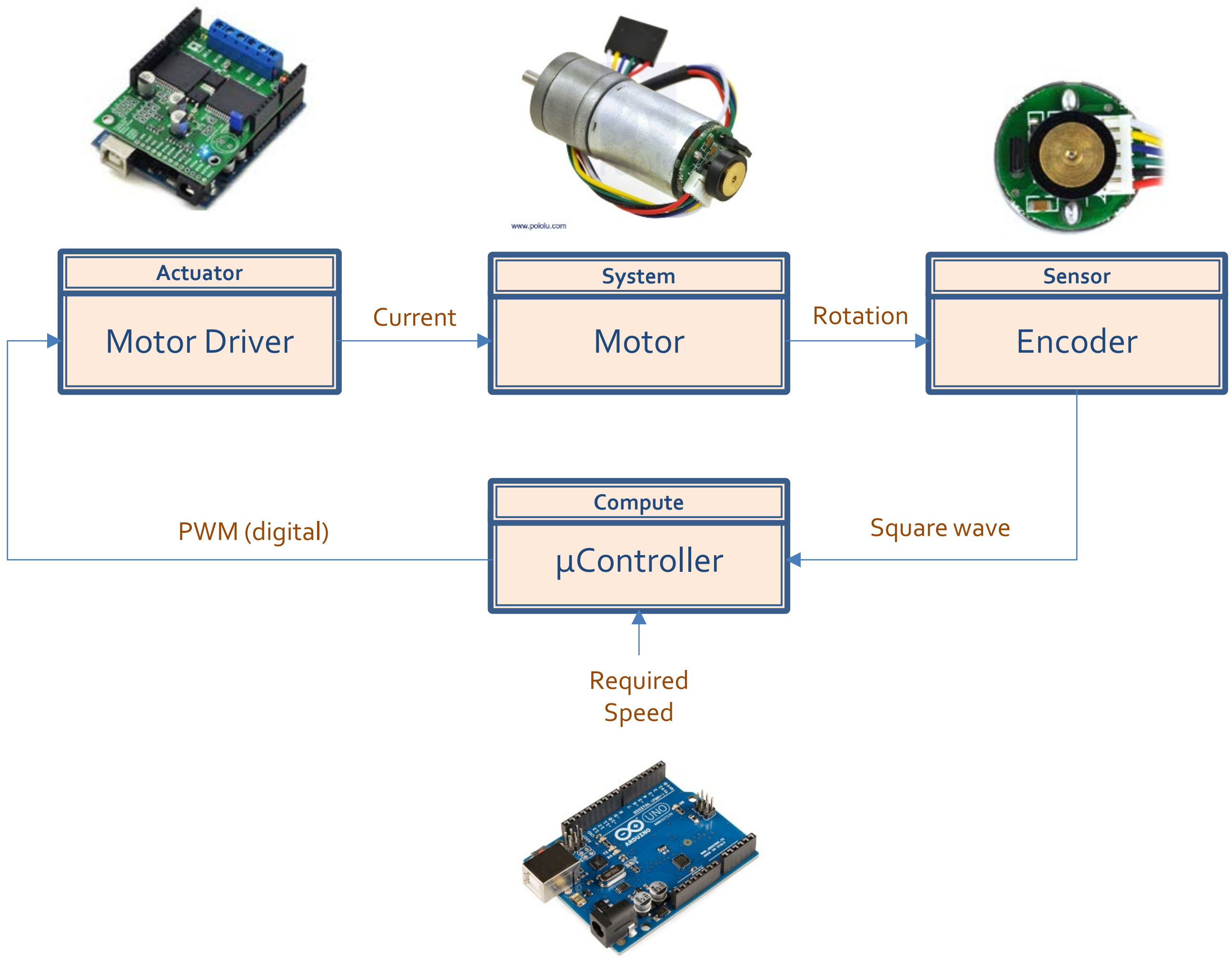


presented by

Suresh Sundaram
Assistant Professor
School of Computer Engineering

August 2015

Overview of Speed Control System



Understanding of Actuator

- Microcontroller board drives the motor through PWM.
- In the board library, required Motor Speed is specified by (u) and is in the range of -400 to +400.
- Driver sends corresponding current to motor to drive it
- Overall relationship -

  *subject to battery charge, surface friction, etc

Table – Input Speed vs Measured RPM

Speed	RPM
400	131.863
350	122.2386
300	103.6323
250	84.4372
200	65.01241
150	46.05117
100	27.73403
50	9.67545
0	0

Overview

Open Loop Control

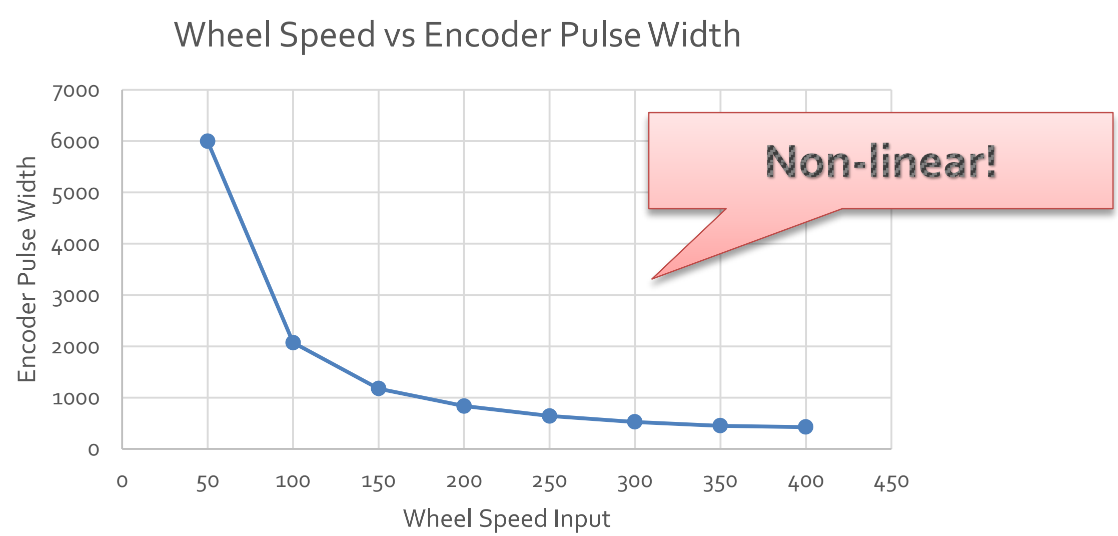
PID Control

2 WD Speed Control

Issues

Understanding the Sensor

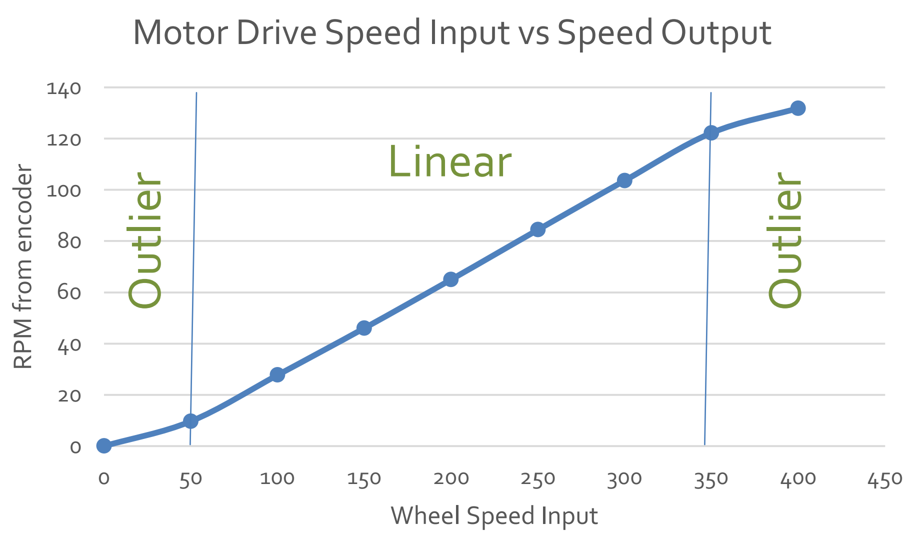
- Encoder is used to measure the speed of motor
- Need to convert square wave from motor encoder to a meaningful speed!!
- Using time-width of pulse is one way -> faster the wheel speed, longer the time-width :



Suresh Sundaram and Venugopalan TK, School of Computer Engineering, Nanyang Technological University, Singapore (2015).

Understanding the Sensor

- Convert time-width to rpm of wheel -> (note 562.25 square waves for every revolution of wheel)



Overview

Open Loop Control

PID Control

2 WD Speed Control

Issues

Suresh Sundaram and Venugopalan TK, School of Computer Engineering, Nanyang Technological University, Singapore (2015).

Open Loop Control

- Assume: perfect system. We can achieve straight line motion using table look-up!
- Setting equal speed on both wheel (100 rpm)!!!

Overview

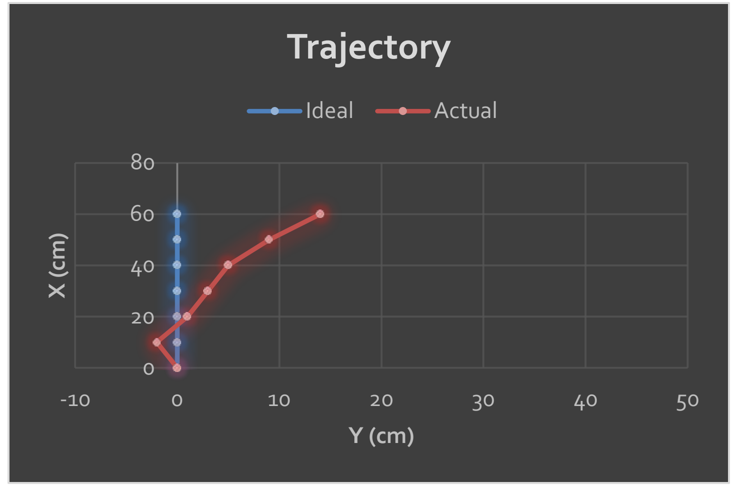
Open Loop Control

PID Control

2 WD Speed Control

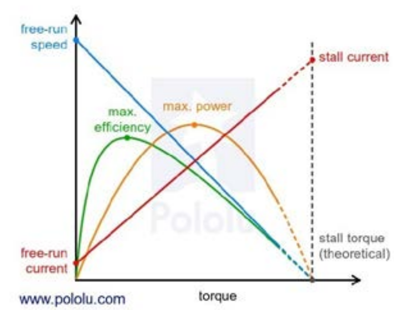
Issues

Open Loop Control



- Open loop control system will not work due to various reasons!

1. Motor characteristics are different
2. Friction of surface
3. Uncertainties
4. PWM calculation errors



Overview

Open Loop Control

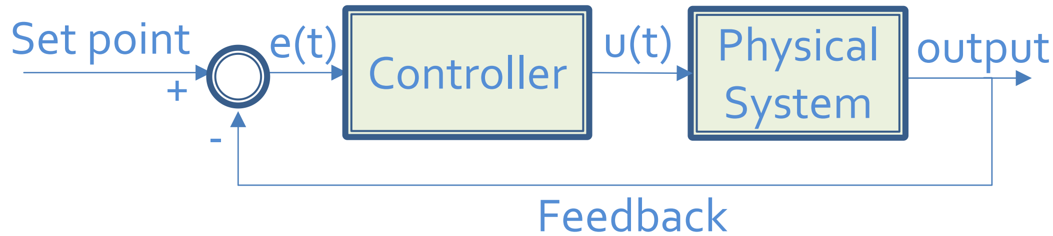
PID Control

2 WD Speed Control

Issues

Closed-Loop Control

Basic PID Control System



- Physical System – Motor
- Input u(t) – Motor current (0-12 A)
- Output – Motor speed (0-130 rpm)
- Set point – Required rpm
- Error e(t) = (Set point – Feedback)
- **PID Controller tries to minimize the error and handle uncertainties.**

PID Control

PID – Proportional, Integral, Derivative Control

- PID is one type of control law!
- Analog PID Implementation:
  - $u(t) = K_P * e(t) + K_I * \int_0^t e(\tau) d\tau + K_D * \frac{de(t)}{dt}$

Reduce the rise time

Reduce a steady offset from set point

Reduce the overshoot

Seborg, et al, Process Dynamics and Control, Second Edition

Overview
Open Loop Control
PID Control
2 WD Speed Control
Issues

Digital PID

Analog Control Law → Z-Transform → Digital-PID

- Digital PID Implementation*:

$$u[k] = u[k-1] + K_1 * e[k] + K_2 * e[k-1] + K_3 * e[k-2]$$

Where

$K_1 = K_p + K_i + K_d$

$K_2 = -K_p - 2K_d$

$K_3 = K_d$

*K.J. Astrom and T.Hagglund. PID Controllers, 2nd ed., Instrument Society of America, 1995.

- Difference equation -> No need to integrate/differentiate in the code!
- The method to find K1,K2,K3 will be discussed later!!

Digital vs Analog

Comparing Digital and Analog Controllers

Digital PID in micro controller	Analog PID
More economical because of cheap components and the simple design algorithm	Comparatively expensive due to the complexity of the design algorithm
Fully integrated and compact	A large number of operational amplifiers and other components are needed
High noise immunity	Noise susceptibility is high
More flexibility because of the ability to program and reprogram the chip	Redesigning is required for any change in the system parameters
High accuracy with faster processing and low power consumption	Less accurate with more processing time and power consumption is higher

Abdul Rasool et al (2009), Sudan Engineering Society Journal

Pseudo code for Digital PID

Algorithm to implement PID control in any micro processor

1. Fix required set point
2. Calculate k1,k2,k3
3. Read feedback signal into 'y' (Data Acquisition)
4. Calculate error between set point and feedback
5. Compute 'u' using digital PID control law
6. Send 'u' to output for control
7. **Wait for 'xx' msec (sampling time) before continuing**
8. Repeat 3-7

Other considerations – (a) Need to know kp,ki,kd

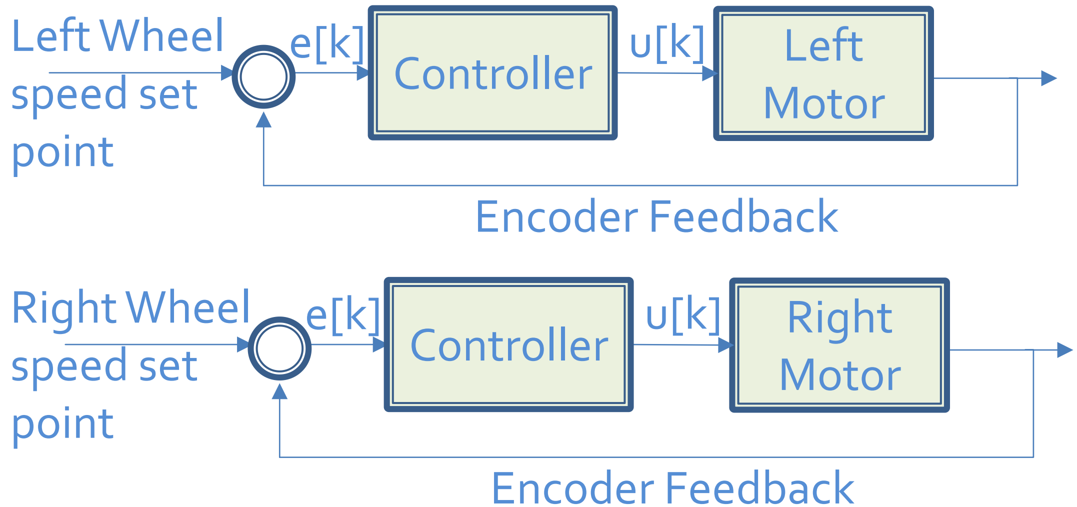
(b) Sampling Time

(c) Input/output saturation, etc.

-> Will be covered in the slides to come

2-wheel drive PID control

2-wheel drive car



- 2 independent control loops
  - Micro-controller implements the two above control loops

2-wheel drive PID control

- Control Loop:
  - Set point – Determined by scenario, eg:
    - Go-forward => left and right wheel SP = 'xx' rpm)
    - Turn-left -> left wheel SP = '-xx' rpm, right = 'xx' rpm for a particular duration
  - Controller –
    - Need to determine tuning parameters to achieve the set point
  - Output –
    - Need to convert the controller output to PWM – Motor driver library can be used
  - Feedback –
    - Need to convert the encoder feedback to speed – needs understanding of the encoder specification

Tuning Parameters

Next logical question – How to choose the K_p, K_i, and K_d ?

- Various Methods :
  - *Ziegler-Nichol's Open Loop Tuning Method* –

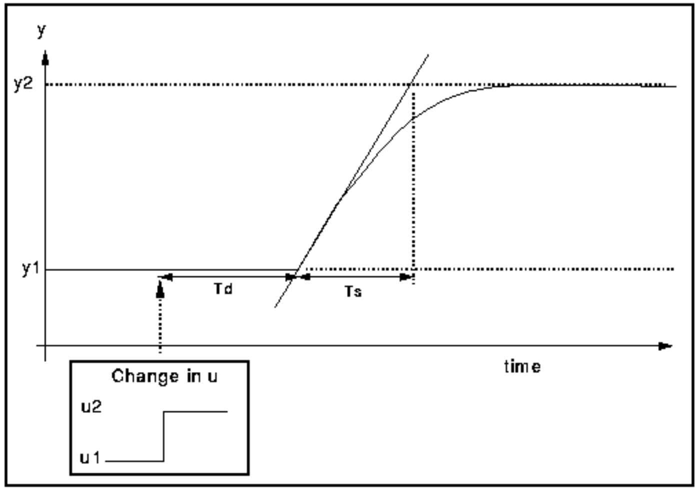
Controller	Parameters		
	K_c	T_i	T_d
P	$\frac{\tau_s}{K\tau_d}$		
PI	$\frac{0.9\tau_s}{K\tau_d}$	$\frac{\tau_d}{0.3}$	
PID	$\frac{1.2\tau_s}{K\tau_d}$	$2\tau_d$	$0.5\tau_d$

Seborg, et al, Process Dynamics and Control, Second Edition..

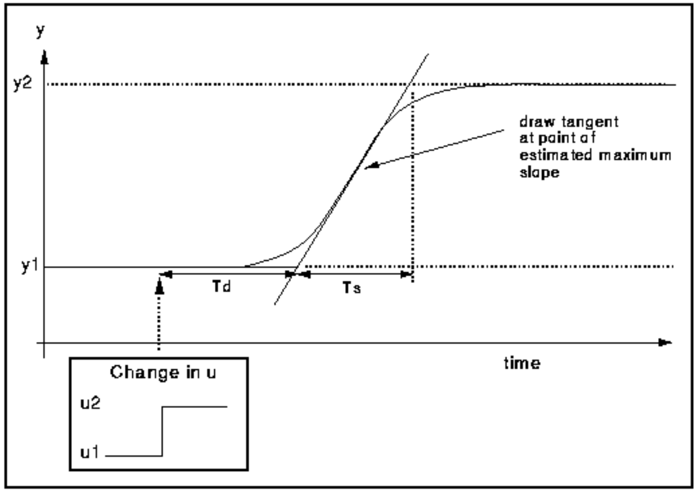
- Overview
- Open Loop Control
- PID Control
- **2 WD Speed Control**
- Issues

Tuning Parameters

- Step Test to determine K, τ_d and τ_s



Ideal Process



Real Process

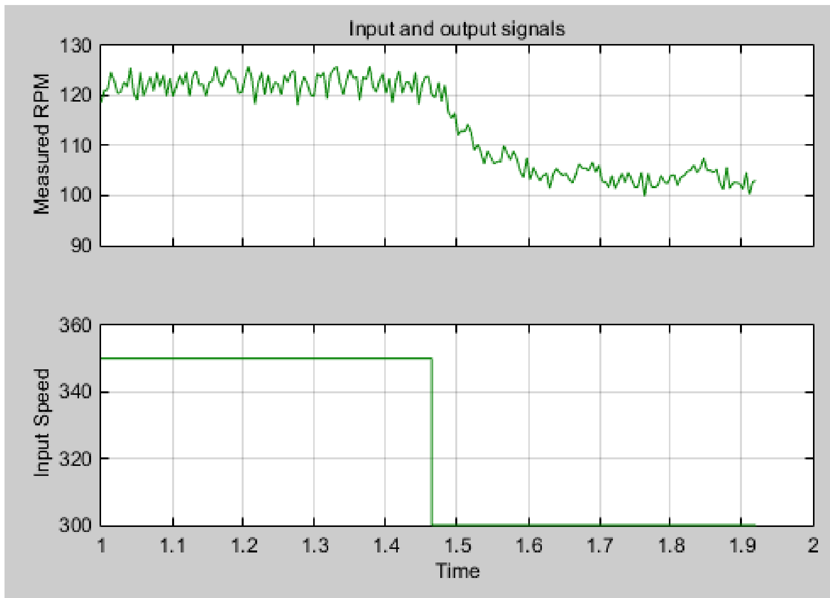
Response of output to an input change of 'M' (u2-u1) at t = 0 sec

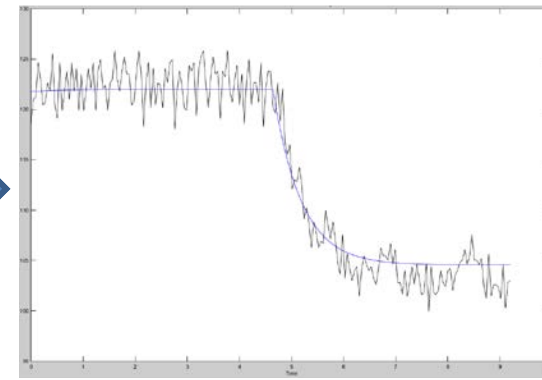
http://www.see.ed.ac.uk/~jwp/control06/controlcourse/course/map/ZN/opennotes.html

- Determine τ_d and τ_s
- Determine K = (y2-y1)/(u2-u1)

Tuning Parameters

- Actual Test data from motor step test –





Determine K, τ and θ

Determine K_p, K_i, K_d

Overview

Open Loop Control

PID Control

2 WD Speed Control

Issues

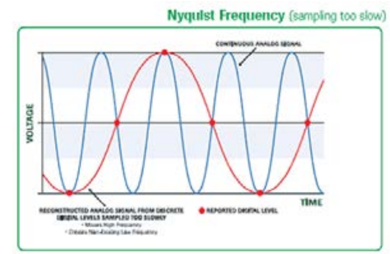
Closed-loop control

Implementation Issues

- **Excessive noise** in input signals
  - Consider filtering
- **Slow processor** – sluggish control
  - Ensure all computations are complete within sampling period
- **Saturation** – breaching limits
  - Impose validity limits for inputs, calculations, outputs.
- **Slipping –** during frequent start/stop
  - Avoid jerky control

Sampling Rate

Rate of data acquisition from feedback -> Important!



- **Too slow**
  - Controller acts slowly, feedback sampling violates nyquist criterion
- **Too fast**
  - Excess noise in differentiator, overflow in integrator
- **Ideal**
  - Rule of thumb -> Sampling Rate = (0.01 to 0.1)*Settling time of system

Beyond this ...

- We have discussed speed control of 2 wheel drive robot
- Other tasks in navigating maze -> obstacle avoidance, exploration will be taken care by top-level mission control

Good Luck!

Suresh Sundaram and Venugopalan TK, School of Computer Engineering, Nanyang Technological University, Singapore (2015).