Control Software

Autonomous Underwater Vehicle Team
Virginia Tech

David Gayman
2 December, 2011

System Dynamics

Laplace transform:

- Transforms functions from time-domain to frequency-domain
- Useful for solving variety of differential equations applicable to controls problems
- Definition:

$$L[f(t)] = F(s) \equiv \int_0^\infty f(t) e^{-st} dt$$

System Dynamics

Table of useful transforms:

	$f(t)$	$F(s)$
18	$\frac{1}{a^2}(1 - e^{-at} - ate^{-at})$	$\frac{1}{s(s+a)^2}$
19	$\frac{1}{a^2}(at - 1 + e^{-at})$	$\frac{1}{s^2(s+a)}$
20	$e^{-at} \sin \omega t$	$\frac{\omega}{(s+a)^2 + \omega^2}$
21	$e^{-at} \cos \omega t$	$\frac{s+a}{(s+a)^2 + \omega^2}$
22	$\frac{\omega_n}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n t} \sin \omega_n \sqrt{1-\zeta^2}\, t$	$\frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$
23	$-\frac{1}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n t} \sin(\omega_n \sqrt{1-\zeta^2}\, t - \phi)$ $\phi = \tan^{-1}\frac{\sqrt{1-\zeta^2}}{\zeta}$	$\frac{s}{s^2 + 2\zeta\omega_n s + \omega_n^2}$
24	$1 - \frac{1}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n t} \sin(\omega_n \sqrt{1-\zeta^2}\, t + \phi)$ $\phi = \tan^{-1}\frac{\sqrt{1-\zeta^2}}{\zeta}$	$\frac{\omega_n^2}{s(s^2 + 2\zeta\omega_n s + \omega_n^2)}$

System Dynamics

Generic second-order dynamic system:

$$m\ddot{x}+b\dot{x}+kx=0$$

$$m\left[s^2X(s)-sx(0)-\dot{x}(0)\right]+b\left[sX(s)-x(0)\right]+kX(s)=0$$

$$X(s)=\frac{(ms+b)x(0)+m\dot{x}(0)}{ms^2+bs+k}$$

$$\ddot{x}+2\zeta\omega_n\dot{x}+\omega_n^2=0$$

$\zeta=$ damping ratio

$\omega_n=$ undamped natural frequency

$\omega_d=$ damped natural frequency

$=\omega_n\sqrt{\zeta^2-1}$

System Dynamics

Generic second-order dynamic system:

$$x(t)=\frac{x(0)}{\sqrt{1-\zeta^2}}e^{-\zeta\omega_n t}\sin\left(\omega_d t+\tan^{-1}\frac{\sqrt{1-\zeta^2}}{\zeta}\right)$$

Underdamped

$$x(t)=\left[\frac{\left(-\zeta+\sqrt{\zeta^2-1}\right)x(0)}{2\sqrt{\zeta^2-1}}-\frac{\dot{x}(0)}{2\omega_n\sqrt{\zeta^2-1}}\right]e^{-\left(\zeta\omega_n+\omega_n\sqrt{\zeta^2-1}\right)t}$$
$$+\left[\frac{\left(\zeta+\sqrt{\zeta^2-1}\right)x(0)}{2\sqrt{\zeta^2-1}}+\frac{\dot{x}(0)}{2\omega_n\sqrt{\zeta^2-1}}\right]e^{-\left(\zeta\omega_n-\omega_n\sqrt{\zeta^2-1}\right)t}$$

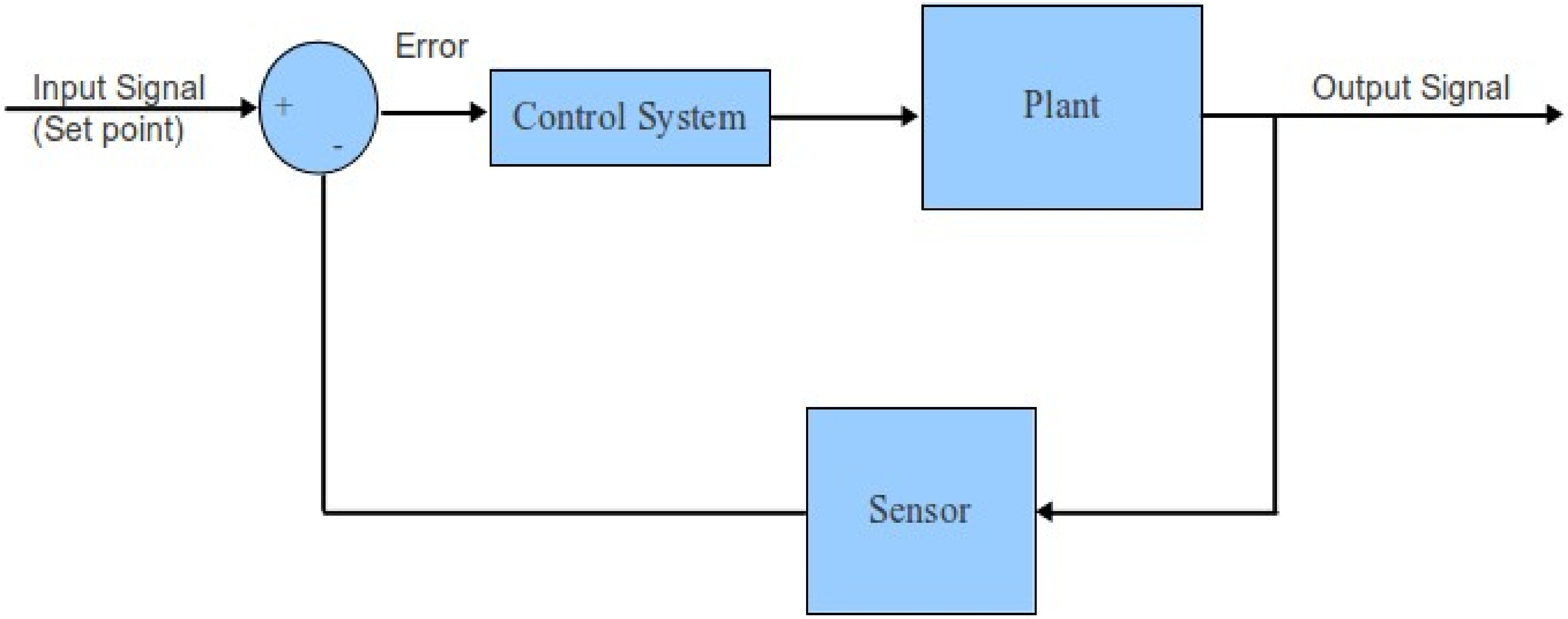
Overdamped

$$x(t)=x(0)\,e^{-\omega_n t}+\left[\omega_n x(0)+\dot{x}(0)\right]te^{-\omega_n t}$$

Critically damped

System Dynamics

Block diagram representation:



Control System – Action

- Effect of a control system determined analytically by transfer function and inverse Laplace transform
- Five types of automatic control systems:

1. Two-position controller
2. Proportional (P)
3. Integral (I)
4. Derivative (D)
5. P/I/D Combination

Control System – Action

1. Two-position controller

- Simplistic, turns a plant on or off to keep error within reasonable band
- Common in climate control systems

Control System – Action

2. Proportional control

- Forces system to correct the error ("drives" the control action)
- Inherent steady-state error
- Basic control system
- Interesting case: Undamped system with inertia *cannot be made stable* with proportional gain alone

Control System – Action

3. Integral (I)

- Corrects steady-state error
- Gain can be chosen to make system stable (even with inertia)

Control System – Action

4. Derivative (D)

- Can add high sensitivity to control system
- Artificially adds damping
- Must be used in combination with P or I

Control System – Action

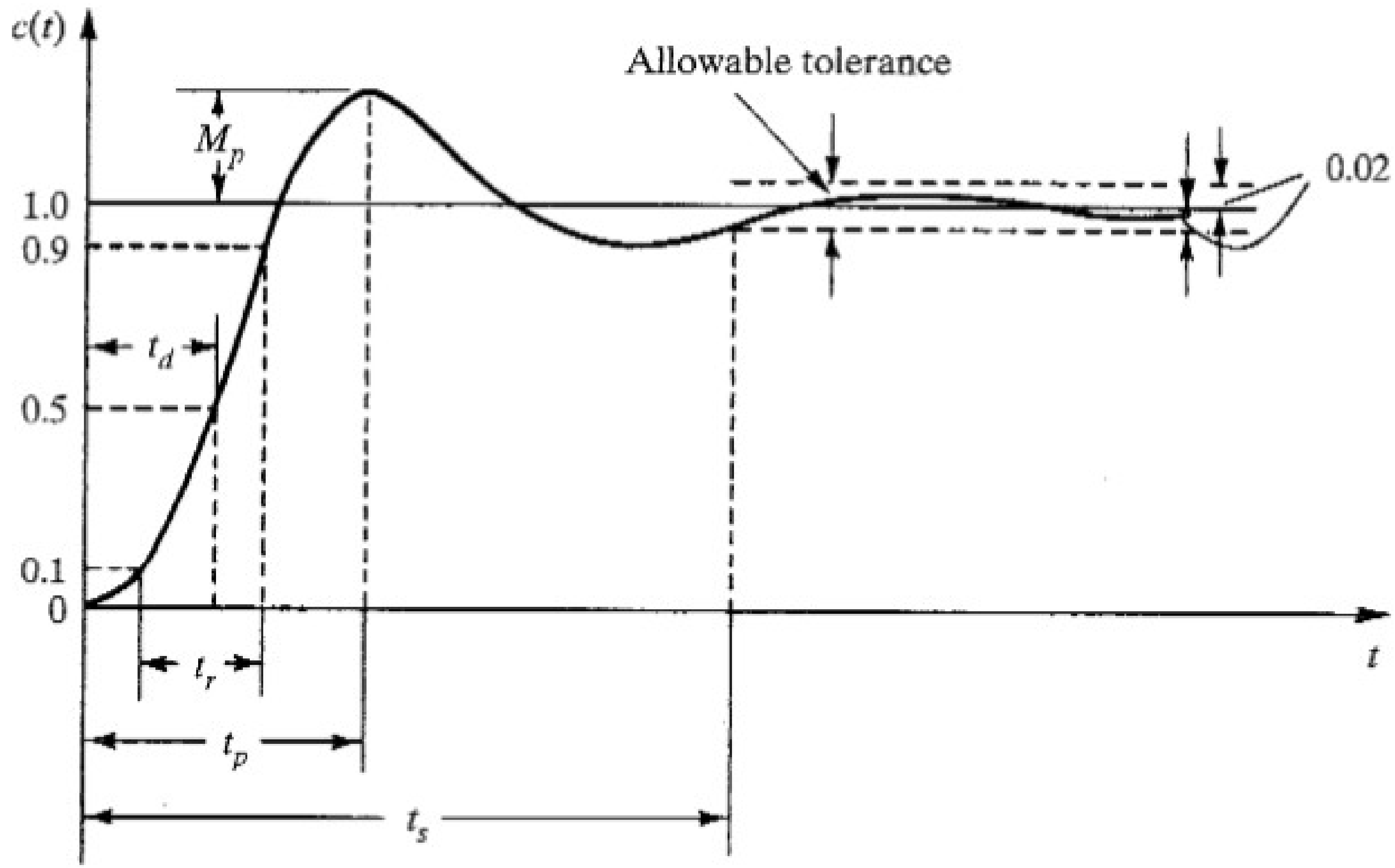
5. P/I/D Combination

- PI, PD, PID
- Uses advantages of all three basic types

$$G_c(s) = K_p + K_i \frac{1}{s} + K_d s$$

Control System – Action

Time-domain response with control system:



Control System – Action

Transient-response specifications:

Specification	Name	Definition
t_d	Delay time	Time to reach half steady-state value (1^{st} time)
t_r	Rise time	Time to rise from 10% to 90% of final value
t_p	Peak time	Time to reach 1^{st} peak of overshoot
M_p	Maximum overshoot	$100\% \times \frac{x(t_p) - x(\infty)}{c_\infty}$
t_s	Settling time	Time to reach 2% of final value

Control System – Action

Demonstration

Control System – Design

Qualitative goals:

- Control the plant to cause stable output signal,
- Reasonable damping (damp out instabilities),
- Relatively fast response to change in set point

Control System – Design

Quantitative goals:

- Can specify transient response parameters and tolerances (fully constrained problem)
- Specify or a subset of these parameters or equivalents (under-constrained problem)

Control System – Design

Stability:

- Can assess system stability from closed-loop transfer function
- Routh's criteria
- Root-locus plot
- Both methods can provide analytical constraints on the control system parameters

Control System – Design

Damping:

- If sufficient response characteristics specified, can design for damping analytically
- Often qualitative or pseudo-analytical methods are used

Control System – Design

Plant transfer function not known:

- For simple systems (ex. 2nd-order LTI mass-spring-dashpot), the transfer function can be approximated
- Values for period and amplitude at multiple locations can be determined in response to an input
- Damping ratio and natural frequency can be calculated
- Thus if the form of the transfer function is assumed, dynamic characteristics can be determined experimentally

Control System – Design

Plant transfer function not known:

- Alternatively for complicated systems, plant transfer function form can be *ignored*
- PID tuning procedures can determine nearly optimal controller parameters

Control System – Design

Starting point: Zieger-Nichols for nonoscillatory systems

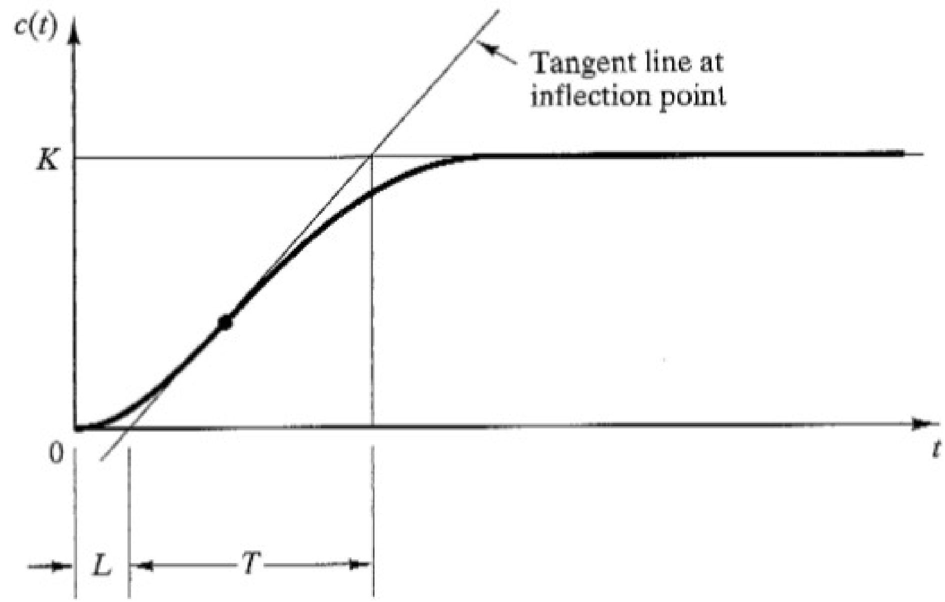
- Obtain plant response to step input
- Extract delay time L and time constant T
- Use tabulated values for PID gains
- Method derived by approximating transfer function by first-order system

Control System – Design

Starting point: Zieger-Nichols for nonoscillatory systems

$$G_c(s) = K_p\left(1 + \frac{1}{T_i s} + T_d s\right)$$

Type of Controller	Kp	Ti	Td
P	T/L	Infinity	0
PI	0.9 T/L	L/0.3	0
PID	1.2 T/L	2 L	0.5 L

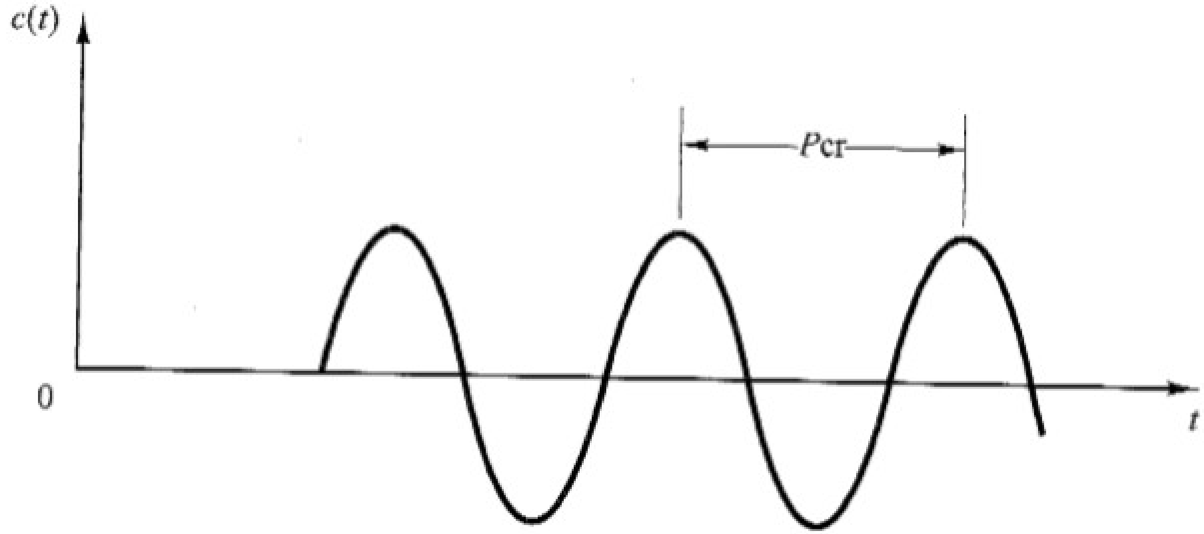


Control System – Design

Starting point: Zieger-Nichols for oscillatory systems

- Set I and D gains to zero
- Apply initial conditions
- Adjust P gain until marginal stability is observed
- Record Kcr, critical proportional gain, and Pcr, critical period
- Using tabulated expressions compute PID gains

Control System – Design



Type of Controller	Kp	Ti	Td
P	0.5 Kcr	Infinity	0
PI	0.45 Kcr	Pcr/1.2	0
PID	0.6 Kcr	0.5 Pcr	0.125 Pcr

Control System – Design

Procedural PID tuning

1. Set Ki, Kd to zero
2. Applying a step input, adjust Kp to observe some overshoot, very small undershoot
3. Set Ti = 1.5 Tau, where Tau is time between over- and under-shoot
4. Reduce Kp to ~0.8 of its value from step 2
5. Set Td = Ti/4

Control System – Design

6. Observe system response
7. Adjust to desired response by following guidelines in table

Effects of *increasing* a parameter independently

Parameter	Rise time	Overshoot	Settling time	Steady-state error	Stability[3]
K_p	Decrease	Increase	Small change	Decrease	Degrade
K_i	Decrease[4]	Increase	Increase	Decrease significantly	Degrade
K_d	Minor decrease	Minor decrease	Minor decrease	No effect in theory	Improve if K_d small

Control System – Design

Inputs:

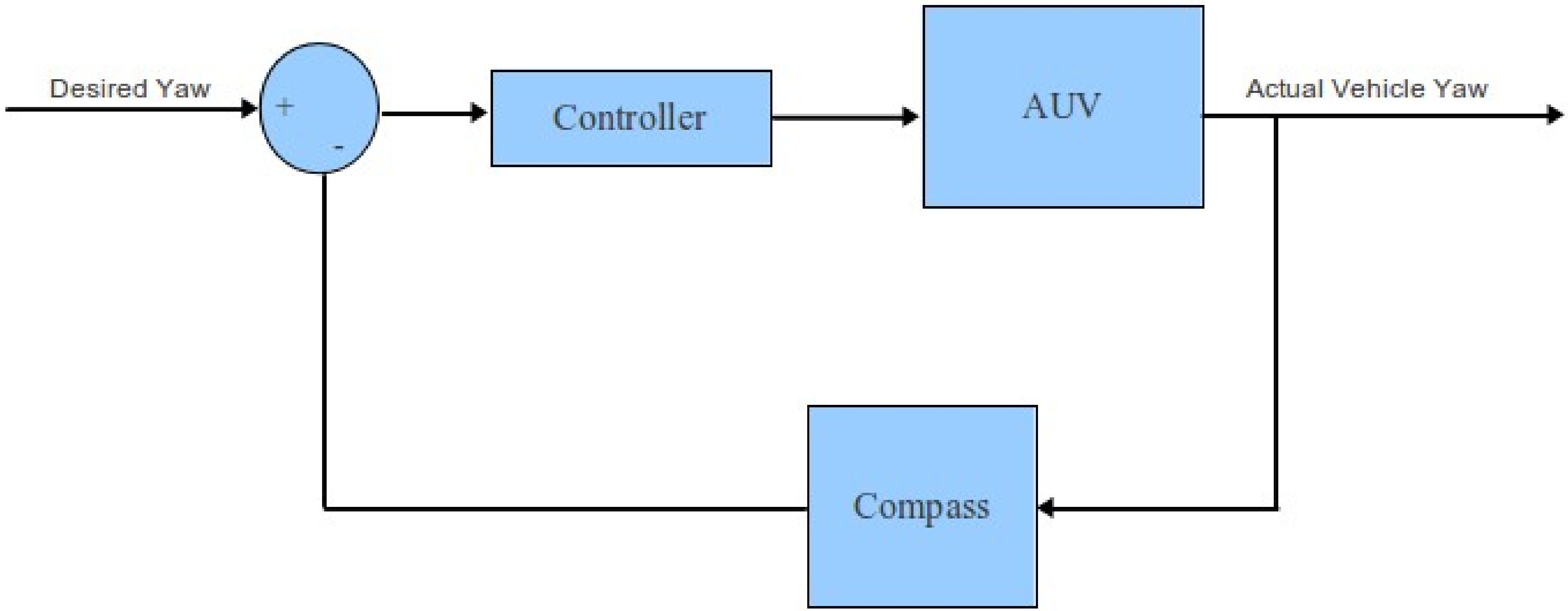
- Set 1:
  1. Desired speed in x, y, z
  2. Desired yaw rate
- Set 2:
  1. Desired relative location in x, y, z
  2. Desired relative location in yaw

Output:

- Time-domain position curves

Control System – Design

AUV control system:



Testing Procedures

Goals:

- Make vehicle highly reliable
- Control system should respond to commands precisely

Testing Procedures

Simulation:

- Simulink and Labview have quality simulation capabilities
- Lower-level code could also be tested

Testing Procedures

Real-world tests:

- System response can be determined in any body of water
- Mathematical model of AUV could be verified experimentally
- Compartmentalized testing/tuning of each degree of freedom
  - Ex. Yaw tuning, minimizing lateral motion in surfacing maneuver
- “Ball-tracking” tests are the goal – instruct AUV to target a moving buoy, and ensure the control system can accurately position itself in front (4 DOF)

Software Implementation

- LCM passes messages between software modules
- C++ front-end function communicates with LCM
- C++ function interfaces with compiled Simulink code
- Simulink model designed in graphical form and compiled into .zip file

Software Implementation

Demonstration

Tasks

1. Plan overhead logic (communicate with Software team about this)
2. Specify tolerances
   - Numerical performance requirements
   - Response curve constraints (optional)
   - Consider how coupled motion will affect these
3. Predict control constants
4. Optimize response with simulation
5. Design tests and schedule
6. Design the software to interface with C++ code, which will interface with LCM
7. Implement software and tune performance

Notes

- Address two biggest issues from last year:

1. Technical approach
    - Last year's method was to tune each parameter during full system tests
    - Proven to work, but could be improved
    - Excessive number of parameters (~36)
    - No measures of precision availabe
    - Better to estimate system, predict controller TF, test, evaluate, present numerical tolerances

Notes

2. Project management

- Last year time not managed well during pool tests
- Necessary to plan rigorous schedule *ahead of time*
- Use additional resources (ie, the tub)

Questions?

References

1. System Dynamics, Katsuhiko Ogata, Fourth Edition
2. Vehicle Vibration and Control, AOE 3034, Professor: Dr. Philen, Virginia Tech
3. Notes from Ogata, with important observations
4. Wikipedia.com, http://www.wikipedia.com/
5. "Tuning of PID Controllers," *techteach.no/publications/books*