Basics of Control Systems

Linear Time Invariant Systems (LTI)

Dr. Ron Desmarais

LTI

Process of Building a Control System

Linear Time Invariant Systems

LTI - Control System Construction

1. Identify the plant (i.e. web service)
    - identify the output will measure (i.e. cpu load)
    - identify what inputs effect the output (i.e. #requests)
2. Model the plant (you need a model!)
    - Identify the relationship between inputs and outputs
        - Find Impulse response g(t), set u(t)=delta [hint: use laplace here to avoid time domain conv.]
        - Determine any arbitrary input to the system (workload model) u(t)
        - Use convolution of u(t) * g(t) to find output response y(t) ---- oh oh ... this is very hard again, so use laplace!

Hint - Key to Solution - if system is LTI, use Laplace Transforms and avoid the convolution mathematics done in the time domain .. its too hard ... really

LTI - Control System Construction

to iterate again !!!!!!

1. Change from time domain to laplace so we can avoid convolution in time domain. We NEED to do convolution to see the output of the system from our selected inputs!
2. Using laplace turns the time domain integral into a the S-domain multiplication .... very nice!

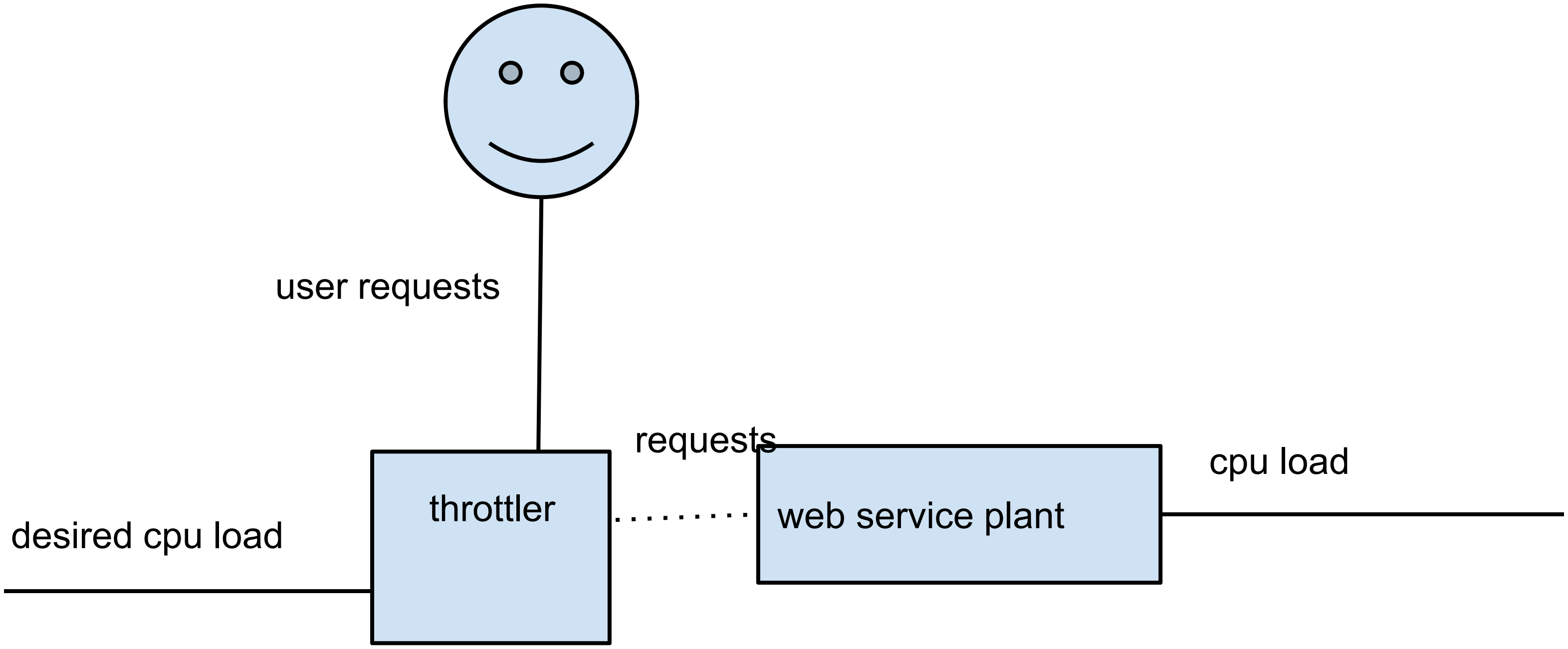
Key - Hint: once you have the impulse response of a system, and the system is LTI, then you can get the output of the system from any arbitrary input

LTI - Control System Construction

Once we have the impulse response g(t) [time domain] which is equal to transfer function G(s) [s domain]. Now we have enough information about the plant to build a control system. Or do we? Maybe ... Maybe not!

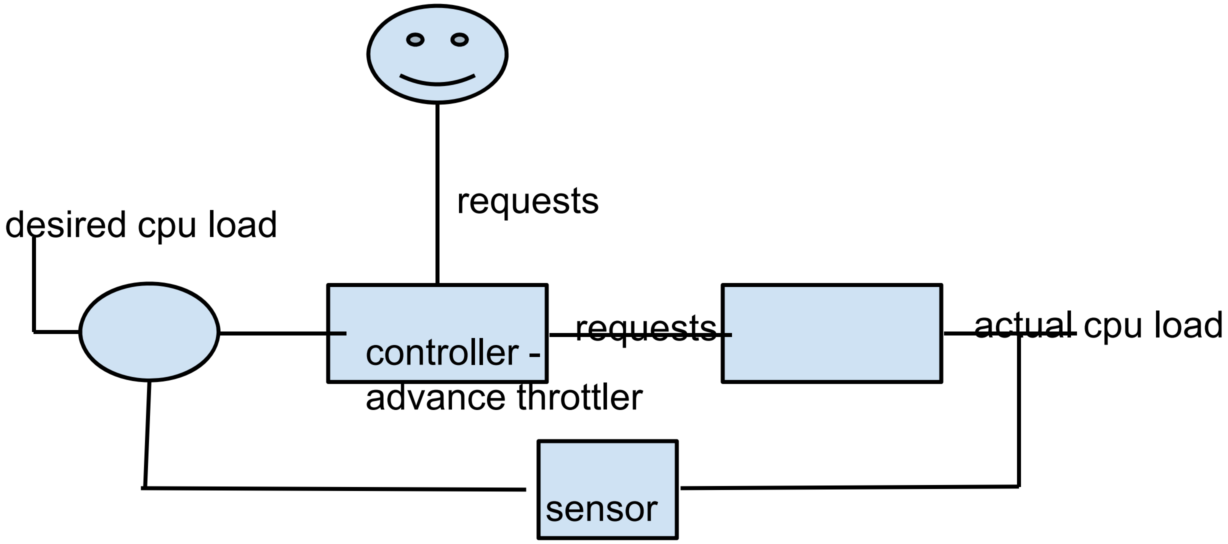
LTI - Control System Construction

Open Loop - open loop control could suffice! If the plant is stable with no disturbances, then ....



LTI - Control System Construction

Closed Loop - closed loop control is a little more complicated in three ways. The first is the sensor used to measure the plant will have to be modeled, the second is that the feedback (whether negative or positive) changes the overall system, and third is the controller is now in the loop!



LTI

So why use LTI?

http://www.youtube.com/watch?v=3eDDTFcSC_Y

Control Systems

Proportional Integral Derivative (PID)

Dr. Ron Desmarais

Cruise Control Example

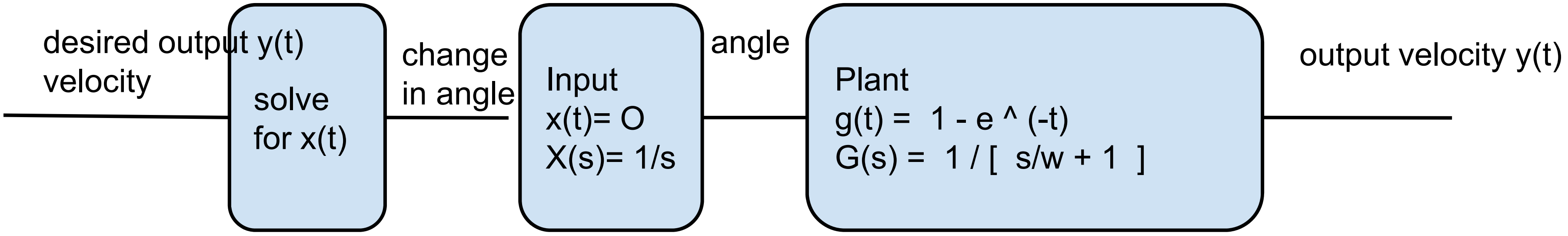
Understand Your Plant

Plant: Car
Input: Gas pedal angle
Output: Vehicle Velocity

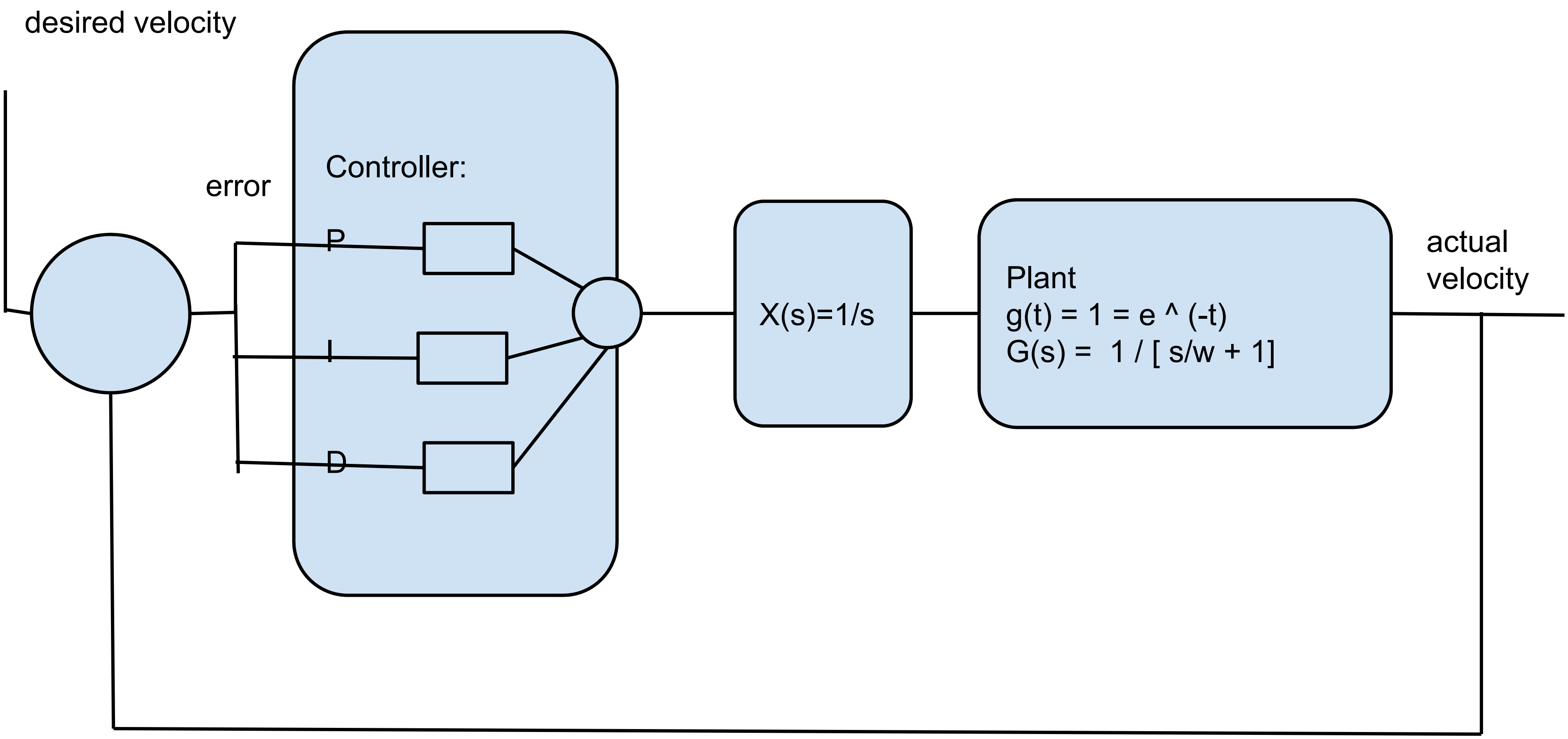
Input Model : x(t) = adjusted angle
Plant Model: g(t) = 1-e^(-t) : vehicle response
LPF

Open Loop Control

If system is perfect, and you modelled your plant and input as LTI, then you can solve for what angle to set accelerator at



Closed Loop Control



desired velocity

error

Controller:

P

I

D

X(s)=1/s

Plant
g(t) = 1 = e ^ (-t)
G(s) = 1 / [s/w + 1]

actual velocity

The PID Controller

why choose the PID Controller rather than making a perfectly optimized controller?
the PID controller is very easy to design, build and maintain ... it is often used for industrial control systems

P: Proportional Control - Kp * error

I: Integral Control - Ki * SUM error

D: Derivative Control - Kd * Change error

Effects of Proportional Control

-large error means larger input signal

Effects of Derivative Control

-quickly changing error means larger input signal

Effects of Integral Control

- small error over long perods of time cause larger input sigal

Analysis Tools to find Kp, Ki and Kd

We want system stability!

Three Important Measures of PID

1. overshoot
2. risetime
3. settling time

Tools to analyze PID's

Bode Plots - frequency response of system - tells me which frequecies give me a high gain which may cause system instability - 1930's Bell labs

Root Locus - relates to finding overshoot,risetime and settling time to determine conditions of dampning ratio and natural frequency of transfer function

-these can be difficult to understand for a control novice ... so lets do something else for now!

Rules of thumb to finding Kp, Ki and Kd (Ziegle-Nichols (ZN) tuning rules)

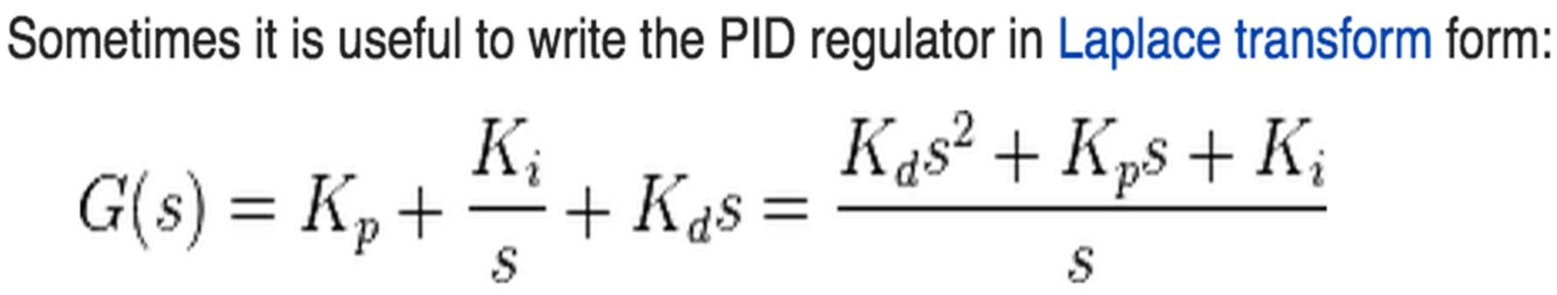
1. find ultimate gain Ku - to do this you need to evaluate your system and record responses

P control : Kc = 0:5Ku

PI control : Kc = 0:45Ku; I = Pu/1.2

PID control (cascade) : Kc = 0:6Ku; I = Pu/2; D = Pu/8

Laplace in Matlab

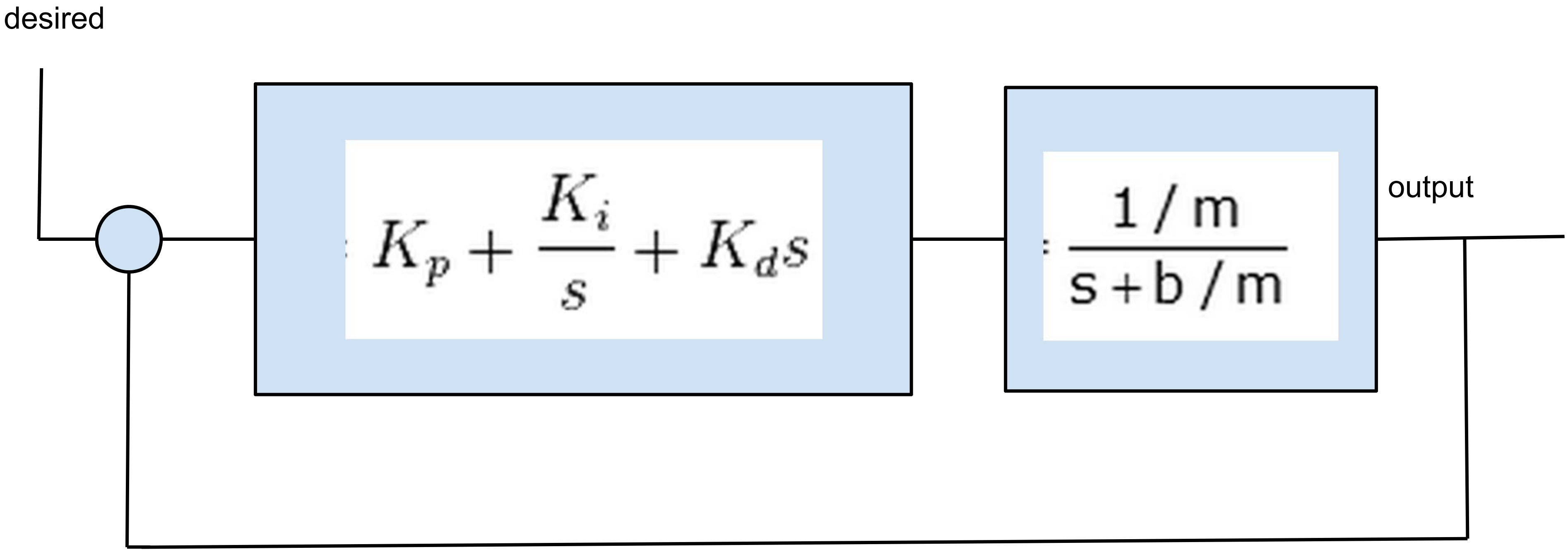


Sometimes it is useful to write the PID regulator in Laplace transform form:

$$G(s) = K_p + \frac{K_i}{s} + K_d s = \frac{K_d s^2 + K_p s + K_i}{s}$$

http://en.wikipedia.org/wiki/PID_controller

Matlab - Laplace



Procedure

Step 1:

- Generate Step Response of Plant from its transfer function, where b=m=1

Step 2:

- Create a simple controller P, PI, PID … and $\frac{1/m}{s+b/m}$ s transfer function, start with transfer function of a proportional controller:

: Kp

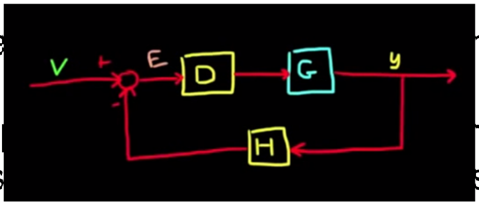
Step 3:

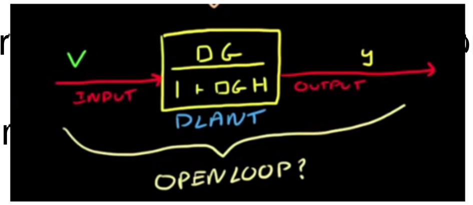
- Determine transfer function when used in a feedback loop with no delay

- de nd compare to response fr mpare?

Step 4:

- re rs, experiment with differer
- us s to create a PID




V
DG
1 + DGH
y
INPUT
OUTPUT
PLANT
OPEN LOOP?