SE1CY15 Cybernetics
Professor Richard Mitchell

Fundamental concepts and applications of Cybernetics –

Feedback theory, Artificial Intelligence and Robotics are covered in SE1CY15 and SE1FC15.

SE1CY15 also covers relevant Calculus which is needed to model Cybernetic systems. Other Maths is in SE1MA15.

SE1CY15 : 20 lectures on Feedback and 10 on Calculus by RJM

+ 10 lectures on Calculus / Differential Eqns by WH

Assessment:

3 hour exam + Tutorial Work for Feedback in PC Lab (from week 4)

Note material used to be in modules SE1CC11 and SE1EM11

On Blackboard, Books, etc.

Lecture Notes on Blackboard and directly on link from

http://www.personal.reading.ac.uk/~shsmchlr/teach.htm

We dont provide printed handouts – we expect you to download file with lecture notes and then add your own comments.

There are some web pages available to help with course at

http://www. reading.ac.uk/~shsmchlr/jscyber/index.htm

One hour PC lab sessions will use these and MatLab.

"Custom book" 'Cybernetics, Circuits and Computing', Mitchell, Harwin, Cadenas, Gong, Potter and Warwick, Pearson, ISBN 978-1-78016-067-2

I have set up a 'discussion board' on Blackboard

Feel free to ask a question or answer someone else's

Bring calculator to lectures

Feedback Syllabus

Introduction to systems with feedback - show variety

Block diagram analysis - show what feedback does

Positive and Negative Feedback - and consequences

Systems with Limits - applications of inevitable limits

Relating feedback and electronics

Frequency Response Analysis of Dynamic Systems

Time Domain Analysis and stability of Dynamic Systems

Computer Modeling of Feedback (including Biological) Systems

See also operational amplifiers and circuit theory in SE1EE15.

Feedback also features in Artificial Intelligence and Robotics

Why the Maths

Dynamic Systems are ones that change – but how do they change?

Differentiation is mathematical process for change

Integration is the reverse process

To work out how a feedback system changes we use these techniques

Often we form a so called differential equation

This 'models' the system

We solve this equation to find out how it actually changes

We also model systems using 'complex numbers'

Which actually makes analysis less complex!

More is said in the Maths lectures in SE1CY15 and in SE1MA15

Cybernetics – a Different Perspective

Kybernetes (Steersman)

Robots

Neural Nets

Learning Robot

VR

Gaia

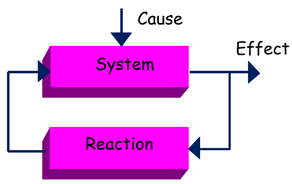
All different + use feedback

Standard and Cybernetic Approaches

Standard View: Aristotelian (Greek) - Cause & Effect

The Open Loop View

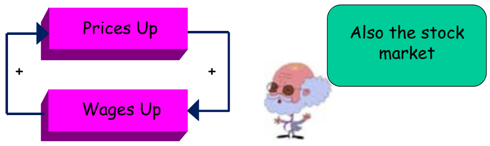
The (practical) **Cybernetic view** : Closed Loop – Has Feeback



Irony: Cybernetics comes from a Greek Word!

Feedback Problems

System more complicated - can lead to run away disasters



Note, signs – net sign round the loop is positive

Arms race is similar – between countries and animals

Not good – but can be useful for quick changes

However …

Feedback for Control

One very useful application for feedback is control

Kybernetes means steersman, so

Aim – to keep boat following a particular course

Despite external factors – winds and tides – pushing boat off course.

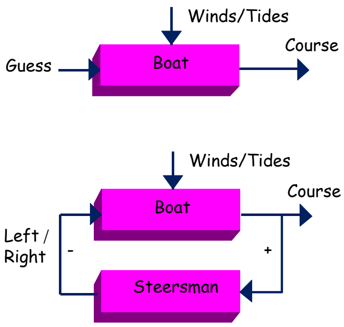
How does he do it?

Feedback Advantages - for Control

No control
– like steering
With eyes closed

With control
– looking
where you
are going

Net sign -ve



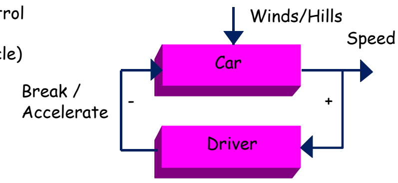
As Applied To Other Systems

Easy to see that also applies to driving car in right direction

But also for travelling at the right speed

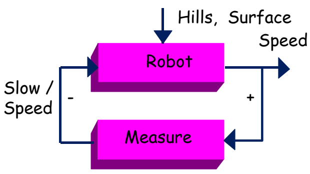
Now detect how fast you are going …

Speed Control
of Car (or
other vehicle)



And to Mobile Robot

Need to be able to control its speed when on different surfaces or on hills..



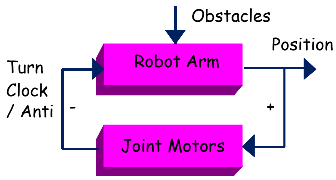
Robot measures its speed – if it is slowing down, then robot speeds up.

Robot Manipulator

Another type of robot – a manipulator -

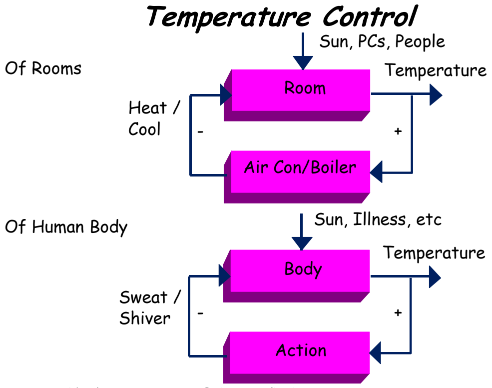
Aim : gripper to be in right position to pick object

Again use feedback control



In fact, for gripper to be in right place, each joint (shoulder, elbow, wrist) must be at the correct angle

Temperature Control



Learning – Another Feedback Process

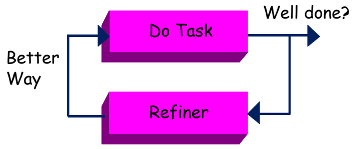
A steersman is not born knowing how to control a boat – he learns

Complex systems need a feedback controller that adapts (learns)

Learning is a feedback process:

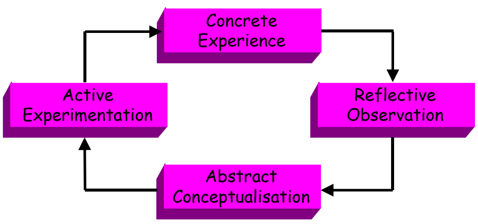
'You learn by your mistakes'

Trial and Error – used by our Robots / Babies



KOLB Learning Cycle

Applies to how we learn ... Note learning requires doing ...



Experience something ... Assess it ... Interpret / compare with other knowledge .... Then put what learnt into practice

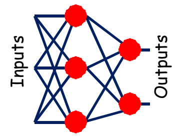
Machine Learning – Neural Network

Like brain: have network of neurons

Each neuron sums products of each input and weight of associated connection

Given inputs, network calculates outputs

But must learn weights

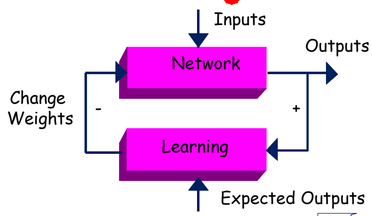


So have training data

Keep applying Inputs

Find Outputs

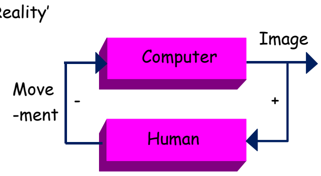
Use error to change weights : Feedback



Human Computer Interaction

Just positioning a mouse is a feedback process

Ultimate HCI is 'Virtual Reality'



Also, Augmented Reality - mixed real and virtual world,

Tele-operation - remote control where operator given input to suggest he/she at remote location.

Needs force feedback and touch – 'haptics'

Haptics

Haptics – feel artificial world

Finger tips hold robot like device

It moves subtly, so you can feel textures

Can push back .. Force feedback

Touch is important .. Our fingers very sensitive

Consider picking up an egg .. press with fingers .. not too much

Press just enough that does not quite slip – feedback ...

Other Interaction

Mobile robots interacting with their environment

- Robots can say sense if there is a wall
- Move around so avoids walls
- At any instance, behaviour set by what sensors see
- eg If no wall, robot goes forward
  - If wall on left, robot steers to right

Or if robots can sense lights

- Behaviour is to steer towards the light

Or have multiple robots, they interact with each other

- In all cases there are feedback loops

Interaction and Learning

Baxter robot

Designed for safe interaction

Can teach it tasks rather than programming it

Eg move one arm by hand, other arm moves with it

In this way you teach it the moves so Baxter can pick up balls.

Summary

We have introduced the module

Key point we have demonstrated the principle of feedback

And that it can be applied to many types of system

- technological as well as animal and (briefly) economic

We have shown feedback can be used for

- control, learning and interaction

Next week we will look at more examples of feedback systems

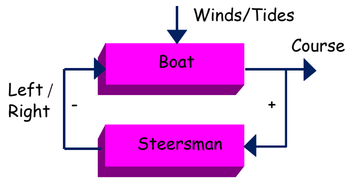
2: More Feedback Systems

We have introduced the cybernetic principle of feedback

And seen it in technological, animal & environmental systems

"control and communication in the animal and the machine", Wiener

One example, a control system : the steersman



We will consider more feedback systems

Rein Control

If the boat is going too far to the left, the steersman steers right

Or if going right, steersman steers to the left.

- Two opposing actions
- Called rein control – after the way in which rider steers horse

For air conditioning system

- If too cold, blow hot air; if too hot, blow cold air – rein control

For body

- If too hot, sweat; if too cold, shiver – rein control

For driving at right speed

- If driving too fast, break; if too slow, accelerate – rein control

Grimblebot – self balancing robot

Robot balances on 2 wheels

If leans forward could fall

- So it moves forward

If leans back

- So it moves back

Again rein control

Achieved by sensor measuring distance to floor

Echo Location using ultrasonics

Also used for robot navigation

Some Other Examples Of Feedback

Pressure Cooker - control pressure

If too hot, valve pushed up, air escape, pressure down

Cooking

Taste stew, if too bland, add flavour, taste, …

League tables

Government say seeks to improve quality of education

Measures schools – how many grades A-C at GCSE

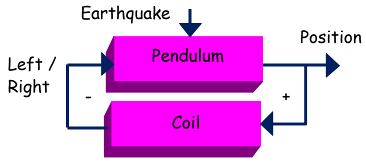
BUT – schools put effort at those at C/D level to get to C

Result not enough support for top students …

Feedback Seismometer

Feedback Seismometer – for measuring earthquake

Small pendulum in coil, feedback stops pendulum moving



Coil output is force stopping pendulum moving equals Earthquake

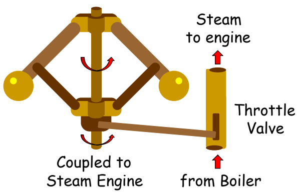
So system output is not the position which is being controlled

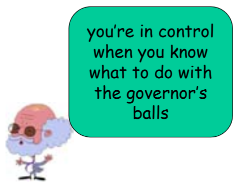
Cybernetics Is In Fact Not New

Watt Steam Engine Governor – to control speed of engine

Affects amount of steam from boiler to engine

In fact governor borrowed from Wind Mills





http://www.reading.ac.uk/~shsmchlr/jscyber/demoGovernor.html

First Man Made Feedback System

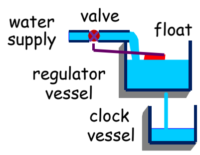
250 BC Water Clock:

Constant Flow into vessel starting at Sun up

Height of water indicates time

But need to ensure constant flow

So have second vessel, with hole, which keep full ….

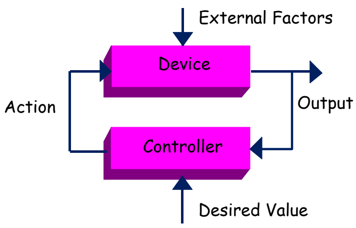


Generalising Feedback System

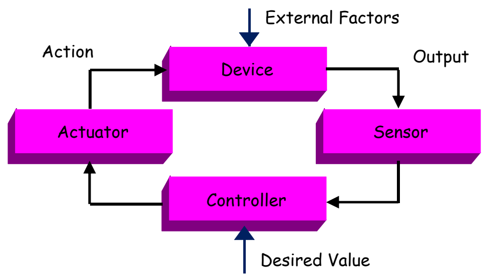
The steersman system and others can be generalised …

System has a device affected by external factors

Whose output is controlled – to equal a desired value



More Realistic Feedback Model



Reflects the fact that the output must be measured - by a sensor

The control action must be effected – by an actuator

Comments

For steersman – he is sensor (or uses GPS, etc) and controller

His actions must somehow move the rudder to change direction

For Grimblebot – sensor is ultrasonics measuring height

Actuator is the motor which turns the wheels so move

For (old style) central heating system,

sensor and controller is a bimetallic strip – if too cold, connect circuit, so boiler pumps hot water to radiators.

For driving at right speed – sensor is speedometer

Important that sensor is accurate – else drive at wrong speed

League tables are poor sensors!

Negative and Positive Feedback

Negative Feedback produces 'regulation' –

If output moves from desired

feedback moves it back

Net sign round loop –ve (as per steersman)

Positive Feedback produces change

If output moves,

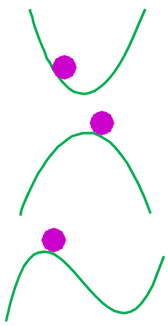
feedback moves it further

Poss run away - eg inflation, arms race

A system with +ve and -ve can be good

Move quicker from one state to another state

We will give an example … from Gaia Theory



Gaia Theory

Feedback can control temperature of a room or our body

James Lovelock's Gaia Theory considers how it is used to control temperature (and other factors) of the Earth

Life on Earth and the planet work together using feedback.

He argued Life influences the environment, whereas biologists argued Life adapts to the environment.

To demonstrate this, Lovelock and Andrew Watson developed a model of an imaginary planet, whose life were two species of daisy

It's called Daisyworld and like Earth, it orbits a sun whose output increases over time and hence heats the planet.

Daisyworld shows how life keeps the temperature constant(ish).

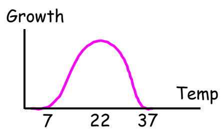
Using positive and negative feedback…

Daisyworld

Black/white daisy seeds in grey soil

Daisies grow best at 22^{o}C

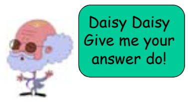
No grow if < 7^{o}C or > 37^{o}C



Daisyworld's Sun is heating up

- like Earth's

What happens to planet's temperature?



Remember

Black objects absorb heat and so heat the local area

White objects reflect heat away and so cool the local area

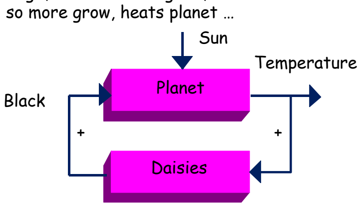
Planet albedo (black..white) sets how much sun's heat absorbed

Initially

Initially, sun too cold, planet too cold for life to grow

When just warm enough, black daisies grow, so have areas of black, heat planet, so more grow, heats planet …

Positive Feedback



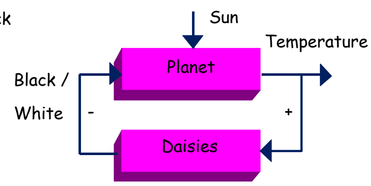
Rapidly temperature rises til at 22^{O}C – the optimum.

Daisyworld Control

Now at right temp … Daisies keep it there

Numbers of Black and White vary heating/cooling – rein control

Negative feedback



In fact the areas of black and white daisies vary .. Change albedo

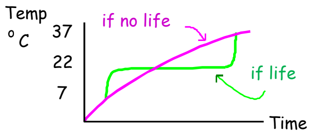
Affects how much heat is absorbed (warming) or reflected (cooling)

The End … and Overall

Ultimately, whole planet is covered in white daisies

Cant cool any more, so if sun heats, daisies die, planet warms …

Positive feedback

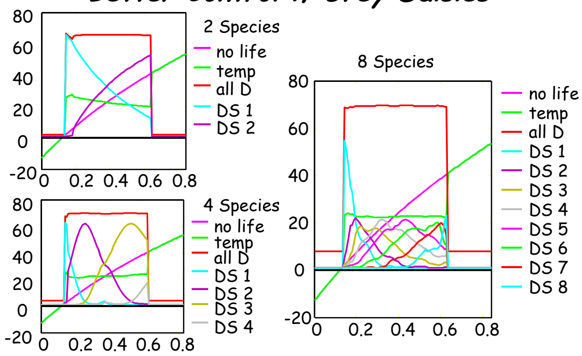


Note, for long period, temp constant –

Above is idealised figure … let's look at actual results .. See also

http://www.reading.ac.uk/~shsmchlr/jscyber/demoDaisyWorldCalc.html

Better Control if Grey Daisies



Daisyworld and Earth

Better control with more daisies .. argument for biodiversity

Daisyworld is imaginary – but has relevance to Earth – eg ice ages

Polar ice caps can grow/shrink – 'snowball earth' positive fb

These caps are white – act like white daisies

Large forest areas and sea are dark – act like black daisies

Clouds can form over tropical rain forests / or over sea

Clouds are white from above – act as white daisies

So amount of cloud can affect how black/white the planet looks

It turns out that life affects amount of cloud ..

Summary

We have looked more at feedback

Seen some other examples of feedback systems

not necessarily new ones!

We have considered rein control

We have introduced concepts of sensors and actuators

We have also looked at positive and negative feedback

And illustrated this with Daisyworld,

thereby introducing Gaia Theory – a cybernetic view of Earth

Next Week

Start to consider how to analyse diagrams …

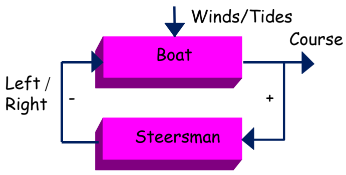
3: Modeling Feedback Systems

We have introduced the cybernetic principle of feedback

And seen it in technological, animal & environmental systems

"control and communication in the animal and the machine", Wiener

One example, a control system : the steersman

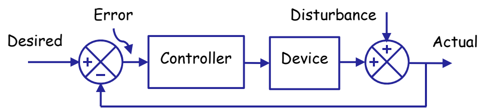


There is, however, a more instructive/useful form of block diagram

In this lecture we introduce this and how to develop such diagrams

Classical Feedback Control System

Often draw block diagram as follows



Output ('actual'), fedback, compared with input ('desired').

Comparison : subtract actual from desired (see + / - in 'summer')

If Actual ≠ Desired have Error processed by Control block

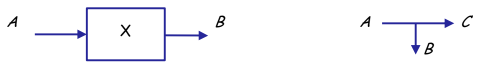
makes 'Device' change its 'actual' state.

i.e. See what have got, if not what want, do something!

Device under control affected by external disturbance

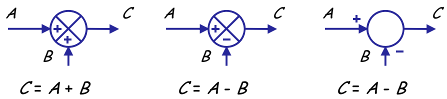
Block Diagram Components

By way of explanation : these are components of block diagrams



Block : *X defines how A becomes B: B = A*X*   Take off : *B = C = A*
X is Transfer Function

Summing Junctions :

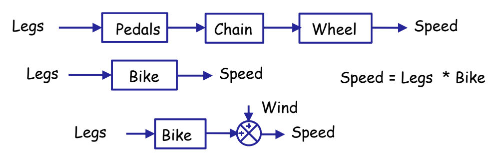


Note *A, B, C* are signals (voltages, positions, speeds, etc)
In Control System, Error = Desired – Actual; for instance

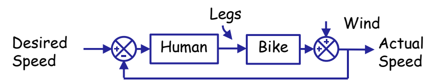
p43 RJM 14/09/15   SE1CY15 Feedback - Part A

Developing System Models

To develop model: divide system into manageable parts
Then bring together to form whole model. e.g. A bike.

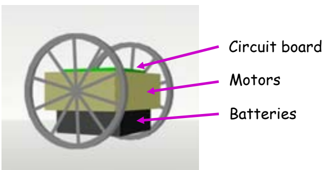


Human can try to ride at constant speed



p44 RJM 14/09/15   SE1CY15 Feedback - Part A

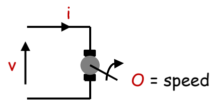
Another Example – a robot



ERIC – has two wheels each powered by a motor
Computer on circuit board outputs signal to each motor
At what speed do the wheels turn?
We consider being on different surfaces, going on hills…

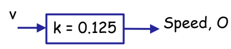
p45 RJM 14/09/15   SE1CY15 Feedback - Part A

First, consider the motor



Here voltage applied is v
It causes current i to flow
Generating a torque so motor turns
O, the motor speed, depends on v
As v becomes bigger so does speed O

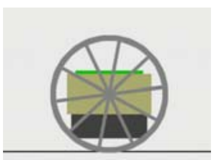
Model O = k * v
k is property of motor
Assume here k = 0.125



If apply 8V, robot speed = 8*0.125 = 1 unit/s clockwise
If v = -24V, motor speed is 3 unit/s anticlockwise, but..

p46 RJM 14/09/15   SE1CY15 Feedback - Part A

What about friction

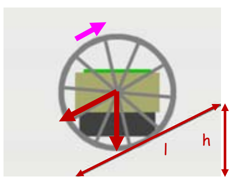


The motor is connected to a wheel
If hold robot in the air, speed O is k*v
But when put on surface, friction between it and wheel slows the wheel
Friction depends on surface
Carpet slows wheel more than wood
Faster the speed, bigger the friction

Our mathematical model now becomes more sophisticated
O = k * v - F * v

p47 RJM 14/09/15   SE1CY15 – Feedback – Part A

What about hills?



Suppose the robot is going up hill
Now gravity also slows the robot
The robot's weight means there is a vertical force = robot mass * g
(g is gravitational constant, 9.8m/s^2)
Part of force is parallel to the hill
This slows the robot

The steeper the slope, the greater the force
In fact if slope is length l and rises by amount h
slowing weight = mass * g * h / l

p48 RJM 14/09/15   SE1CY15 – Feedback – Part A

So model becomes

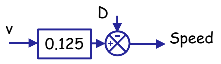
Suppose robot is on a slope with friction F, where it rises by an amount h whilst travelling a distance l and the robot mass is m

Then the speed O when you apply voltage v is

$O = k * v - F * v - m*g*h/l$

If the robot is going down hill, its weight speeds it up

To simplify slightly, let's have a net disturbance D being that due to friction and that due to hills



Some Numbers In

If v = 24, k = 0.125 and there is no friction

O = 24 * 0.125 = 3 // this is the desired speed

Suppose friction F = 2 and slope is flat

O = 24 * 0.125 – 2*0.125 = 2.75 less than desired speed 3

Suppose friction F = 2 and slope rises by h = 1 when travelling a distance l = 2, and m = 0.1 and g = 9.8

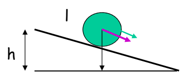
O = 24 * 0.125 – 2*0.125 – 0.1*9.8*1/2 = 2.26 // too slow

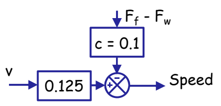
If the robot is going down hill, the speed is

O = 24 * 0.125 – 2*0.125 + 0.1*9.8*1/2 = 3.24 // too fast

The speed varies a lot …

Lecture 3 – In Class Exercise





Robot downhill – weight speeds up

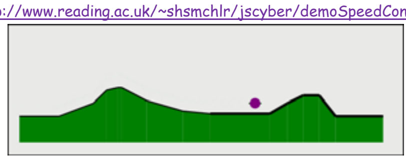
Speed is k * v – c * (F + m * g * h / l)

What is speed if v = 8, h = 1, l = 2, m = 0.2, F is 1 and c = 0.1

Speed is

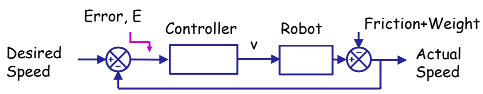
Let's see this in action

http://www.reading.ac.uk/~shsmchlr/jscyber/demoSpeedControl.html



Robot moves .. Note hills/ friction .. V is 24 … O varies

We need control …



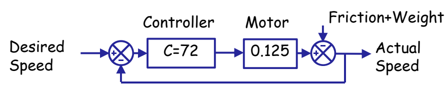
The difference between the desired speed and the actual speed is the 'error', E

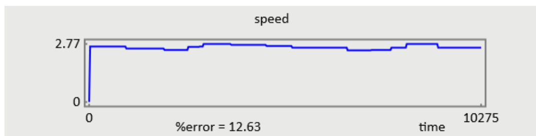
The controller calculates a value v (fed to the robot) based on E

But how do we set v?

Simplest is so called Proportional Control : v = E * constant

Simple Proportional control

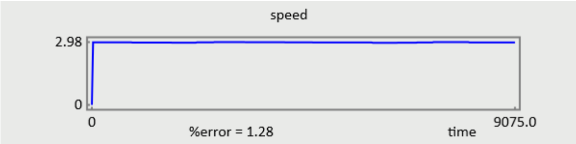




Desired Speed 3 – actual speed pretty constant … but varies and not 3!

Happens when controller output is proportional the Error (here * 72)

Have C bigger : 792

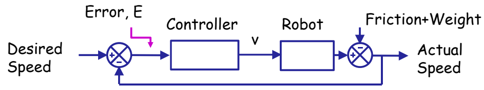


Note, C is almost constant …

And much nearer to 3

Later we will explain why having bigger C helps

But if you were the controller?



You would need a 'speedometer'

Then you would tend to increase v if robot too slow, decrease if too fast

So you are not 'computing' v as E * constant

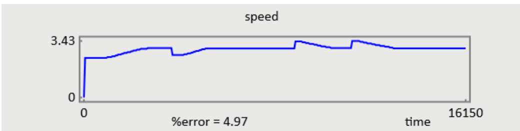
You are doing something like v = v + amount due to E

This is what is known as integral control ..

V is proportional to the integral of the error

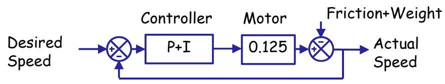
Program Allows this

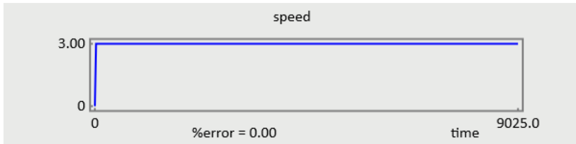
In this case, speedo indicates too slow … so press U



Not perfect, as human is slow to respond …

Advanced Control does work..





Here controller output has two parts added

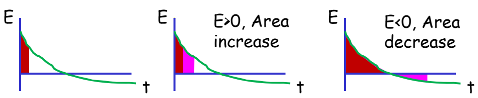
one proportional to the Error, the other to the integral of the Error

This is so called P+I control

Why Does this Work

Remember integration is area under function

Area at one time is the area last time plus an amount due to Error now



So if error is high, integral of it increases a lot – bigger signal to robot

Speeds robot so error is smaller – so signal to robot now changes less

If E negative, area decreased, robot slows

When E is 0, area unchanged, so speed constant – (when Desired = Actual)

I control only ok, but machines often use P+I control …

Some actually use P+I+D control – D is differential of error!

Summary

We have looked at a formal block diagram of control systems

Summing Junction plus blocks with 'transfer functions'

We have considered how to build a model of a system

First of a bike, then of a robot

We then looked at the robot and saw that, when taking account of friction and hills, its speed varies unless we use control

We saw simple control was better – and improved when C was bigger

We then introduced integral and then P+I control

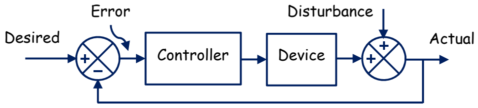
Next Week

We will look at how we extend the transfer function concept – including showing why bigger C helps…

4 : Analyzing Feedback Systems

We have introduced the cybernetic principle of feedback and modeled the Classical Feedback Control System

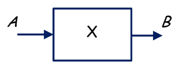
aim - actual (output) equals desired (input)



To verify this (and to find out it's not true, as shown with robot!) we need to be able to analyse such feedback systems.

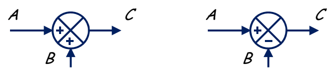
We have models (transfer functions) of the components, but what is the overall model (or transfer function)? Let's find out.

Reminder of Component Types



X is transfer function
How signal A transferred to B

$B = A*X$ (or $X*A$) or $\frac{B}{A} = X$



$C = A + B$ $C = A - B$

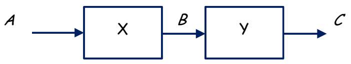
If B = 0, C = A
Then TF = 1
If A = 0, C = -B
Then TF = -1

But systems have combinations of such components.

Can we combine for overall model or transfer function

Analysis of Two in Series

First consider: two blocks in series



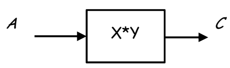
From what we have already said

$B = A * X$

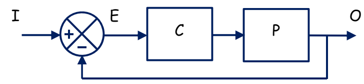
and $C = B * Y$

So $C = A * X * Y$

ie X and Y combined single block model Transfer Function X * Y



What of Feedback Control System ?



$E = I - O$ ← *From summing junction*

$O = E * C * P$

$O = (I - O) * C * P$ ← *Now substitute for E*

$= I * C * P - O * C * P$

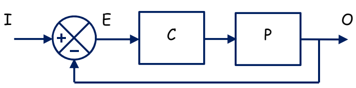
$O + O * C * P = I * C * P$ ← *Gathering 'like' terms*

$O * (1 + C * P) = I * C * P$ ← *Now divide by 1 + C * P,*

$O = I * \frac{C * P}{1 + C*P}$ or $\frac{O}{I} = \frac{C * P}{1 + C*P}$

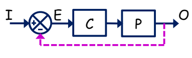
But needed 7 lines of algebra ...

Better : Forward over 1 Minus Loop Rule



Forward transfer function, I to O,

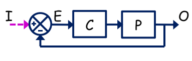
ignore (0) feedback signals: $C * P$



Loop transfer function, ignore (set to 0) signals entering loop

E to E (or O to O) LTF = $- C * P$

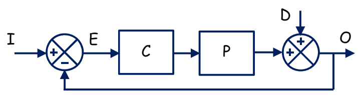
(here I = 0, so O to E is -1)



Thus, $\frac{O}{I} = \frac{\text{Forward}}{1 - \text{Loop}} = \frac{C*P}{1 - -C*P} = \frac{C*P}{1 + C*P}$

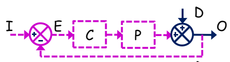
'closed loop transfer function'

What about Disturbances?

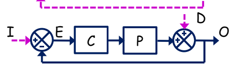


O = output of P + D; If assume I = 0,

Forward = 1

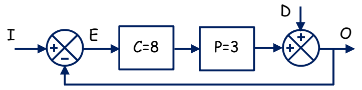


Loop same, so -CP



Thus $\frac{O}{D} = \frac{\text{Forward}}{1\text{-Loop}} = \frac{1}{1 -- C*P} = \frac{1}{1 + C*P}$

In Class Exercise



For O/I : Forward = 3*8 = 24

Loop = -3*8 = -24

$\frac{O}{I} = \frac{Forward}{1 - Loop} = \frac{24}{1 - -24} = \frac{24}{25}$

For O/D : Forward = 1

Loop = -3*8 = -24

$\frac{O}{I} = \frac{Forward}{1 - Loop} = \frac{1}{1 - -24} = \frac{1}{25}$

See http://www.reading.ac.uk/~shsmchlr/jscyber/tfstatic.html

Complete Response

If assume D = 0, $\frac{O}{I} = \frac{C*P}{1 + C*P}$ i.e. $O = \frac{C*P}{1 + C*P} * I$

If assume I = 0, $\frac{O}{D} = \frac{1}{1 + C*P}$ i.e. $O = \frac{1}{1 + C*P} * D$

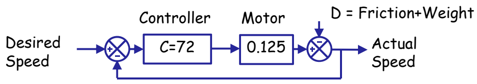
In general, I and D wont be 0, so we combine both

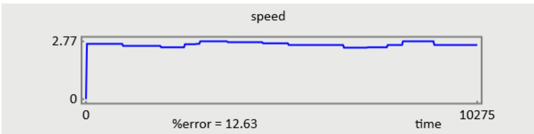
This is the **principle of superposition** (also used in electronics), so

$$O = \frac{C*P}{1 + C*P} * I + \frac{1}{1 + C*P} * D$$

In example, $O = \frac{24}{25} * I + \frac{1}{25} * D = 0.96* I + 0.04*D$

Relating to Robot Last Week





Ignoring Friction, O is about 2.7 +/- about 0.1

Do the numbers confirm this?

Doing analysis

Here I = 3, C = 72, P = 1/8 and let's ignore D;

O = 3 * (72*1/8) / (1 + 72*1/8) = 3 * 9/10 = 2.7

So ignoring D, O is approx 2.7 (as shown in the figure)

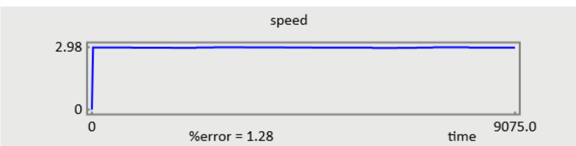
Ignoring I, and assume D = 1 (about right)

O = 1 / (1 + 72/8) = 0.1

Thus the model suggests O should be 2.7 +/- 0.1

Again consistent with figure

In class Exercise – C = 792



Here I = 3, C = 792, P = 1/8 and let's ignore D;

O = 3 * (792*1/8) / (1 + 792*1/8) = 3 * 99/100 = 2.97

So ignoring D, O is approx 2.97 (as shown in the figure)

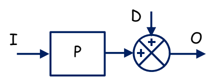
Ignoring I, and assume D = 1

O = 1 / (1 + 792/8) = 0.01

Thus the model suggests O should be 2.97 +/- 0.01

On Assessing System

Without feedback (open loop) …



Knowing P, assuming D = 0, if want O, set I to O/P

eg if P = 0.125, want O = 3, set I to 24

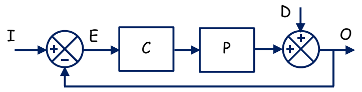
But if D = 1, O = 0.125 *24 + 1 = 4

So any disturbance affects O directly by that amount

Also, during operation values change (eg resistor warms)

So if P changes by 10% to 0.1375, O changes (by 10%) to 3.3

With Feedback – Is it what we want



If D = 0, $O = \frac{C*P}{1+C*P} * I$ Is O = I i.e. $\frac{O}{I} = 1$?

NO, as 1+C*P ≠ C*P

If I = 0, want: O = 0 * D   Got: $O = \frac{1}{1+C*P} * D$

BUT, 1/1+CP ≠ 0

When systems operate, their values change

If P changes, want $\frac{O}{I} = \frac{C*P}{1+C*P}$ to not change

BUT, it will

But, we can get close to what we want, and better than open loop ..

Let's Put Some Numbers In

Suppose C = 5 and P = 2 … let's investigate

If D = 0, $\frac{O}{I} = \frac{5*2}{1+5*2} = \frac{10}{11} = 0.909$

If I = 0, $\frac{O}{D} = \frac{1}{1+5*2} = \frac{1}{11} = 0.0909$

If P changes by 10% to 2.2

If D = 0, $\frac{O}{I} = \frac{5*2.2}{1+5*2.2} = \frac{11}{12} = 0.9167$

% change in $\frac{O}{I} = 100*\frac{0.9167 - 0.909}{0.909} = 0.84\%$

O quite close to 1; One unit of D results in O changing by < 0.1;

10% change in P results in smaller 0.84% change … ok, can do better

Exercise : Try with C = 50 and 500 ?

$\frac{O}{I} = \frac{50*2}{1+50*2} =$ $\frac{O}{I} = \frac{500*2}{1+500*2} =$

$\frac{O}{D} = \frac{1}{1+50*2} =$ $\frac{O}{D} = \frac{1}{1+500*2} =$

$\frac{O}{I} = \frac{50*2.2}{1+50*2.2} =$ $\frac{O}{I} = \frac{500*2.2}{1+500*2.2} =$

%change: 100*   %change:

Conclusion – much better when C * P is big

Consistent with bigger C in robot example

Numbers for complete model

$$O = \frac{C*P}{1+C*P} * I + \frac{1}{1+C*P} * D$$

If C = 11, P = 9, I = 2 and D = 5;

O = 99/100 * 2 + 1/100 * 5 = 2.03

If C = 500, P = 2, I = 1 and D = 4

O = 1000/1001 * 1 + 4 / 1001 ~ 1

If (minus) loop gain (CP) high, response close to desired as

1+CP ≈ CP, so $\frac{C*P}{1+C*P} \approx 1$ and 1+CP large so $\frac{1}{1+C*P} \approx 0$

So O ≈ 1 * I + 0 * D ≈ I

Summary

We have started to analyse feedback systems

We have formed block diagrams,

where each block has a transfer function

We have then seen how to combine the blocks

So the overall system has a transfer function

We have seen that for the control system, Output <> Input

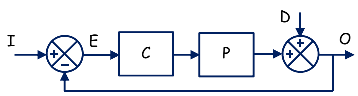
However, if loop gain is high then Output ~ Input

And not affected by disturbances, or changes in parameters

Next week, we explore another form of feedback system

5 : More Feedback Systems Analysis

We extend our analysis; relate to electronics, and introduce MatLab



For this feedback control system: use forward/1-loop

If D = 0, $\frac{O}{I} = \frac{C*P}{1+C*P}$ If I = 0, $\frac{O}{D} = \frac{1}{1+C*P}$

Overall, $O = \frac{C*P}{1+C*P} * I + \frac{1}{1+C*P} * D$

If C*P big, O ~ 1*I + 0*D - what control engineers want

See again http://www.reading.ac.uk/~shsmchlr/jscyber/tfstatic.html

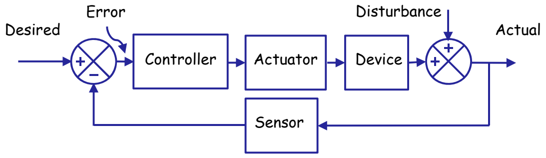
More Complete Diagram

In that diagram C represents Controller, P device under control

Earlier we commented that in practice also need

Actuator - turn control signal to suitable form to drive device.

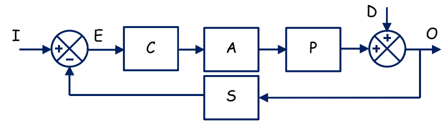
Sensor - measure Actual, converts to form so can compare.



So let's add these extra blocks and analyse…

'Fuller' Control System

Here A is the actuator and S the sensor



Now Forward = C * A * P and Loop is - C * A * P * S

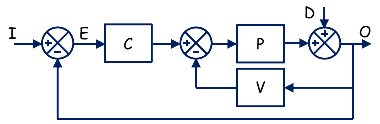
If D = 0, $\frac{O}{I} = \frac{C*A*P}{1 + C*A*P*S}$ If I = 0, $\frac{O}{D} = \frac{1}{1 + C*A*P*S}$

Overall, $O = \frac{C*A*P}{1 + C*A*P*S} * I + \frac{1}{1 + C*A*P*S} * D$

If C*A*P*S big, O ~ I/S + 0*D – want S = 1 : true measurement

If more than one 'loop' ...

Better control if feedback both position and velocity :



TF = Forward over 1 minus sum of each loop (-CP and –PV)

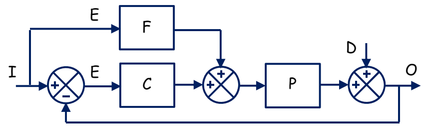
If D = 0, $\frac{O}{I} = \frac{C*P}{1 - -C*P - -P*V} = \frac{C*P}{1 + C*P + P*V}$

If I = 0, $\frac{O}{D} = \frac{1}{1 + C*P + P*V}$

For more complex systems, rule needs refining, but ok in Part 1

If more than one 'Forward'

Put in an 'anticipatory' amount – so called feedforward control



Now 'Forward' has two paths F*P and C*P, sum each Forward

Hence if D = 0, $\frac{O}{I} = \frac{\text{Each Forward}}{1 - \text{Loop}} = \frac{F*P + C*P}{1 - -C*P} = \frac{F*P + C*P}{1 + C*P}$

If F = 1/P, then F*P = 1, so $\frac{O}{I} = \frac{1 + C*P}{1 + C*P} = 1$

If I = 0, Forward is still 1, so $\frac{O}{D} = \frac{1}{1 + C*P}$

Lecture 5 - In Class Exercise

For system with feed-forward control :

$\frac{O}{I} = \frac{F*P + C*P}{1 + C*P}$ $\frac{O}{D} = \frac{1}{1 + C*P}$

Assuming, C = 6, P = 4 and F = 0.25, calculate $\frac{O}{I}$ and $\frac{O}{D}$.

Then, find $\frac{O}{I}$ if P changes to 4.4

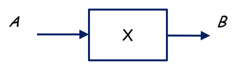
$\frac{O}{I} = \frac{0.25*4 + 6*4}{1 + 24} = \frac{25}{25} = 1$ $\frac{O}{D} = \frac{1}{1 + 24} = 0.04$

$\frac{O}{I} = \frac{1.1 + 26.4}{1 + 26.4} = 1.004$

Relating Blocks to Real World

Simple Block

X is Transfer Function



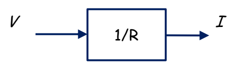
Defines how A becomes B $B = A*X$ (or $X*A$) or $\frac{B}{A} = X$

Sometimes call X the 'gain' : eg X = 10

B is 10 times A, B and A are signals of same type

But X can change signal types – eg Resistor

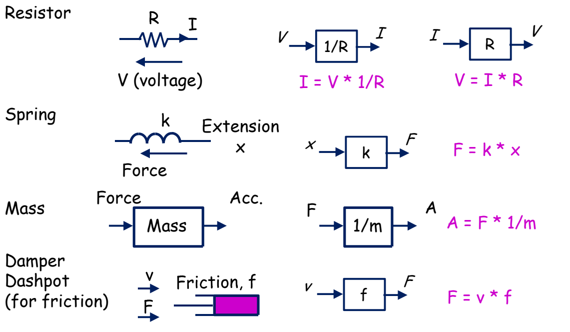
B (Current) = A (Voltage) * X (1/Resistance)



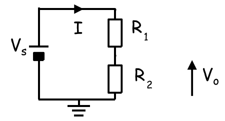
Units Current Amps, Voltage Volts Resistance Ohms (Ω) 1/R in Siemens

V = 2V, R = 4Ω, then I = 2 * ¼ = 0.5A

Some Other Examples



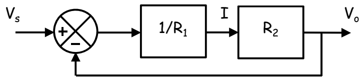
Potential Divider is Feedback System!



Kirchhoff: $V_s = I * R_1 + V_o$, so

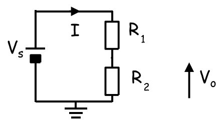
$I = \frac{V_s - V_o}{R_1}$

$V_o = R_2 * I$, so



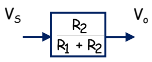
$$\frac{V_o}{V_s} = \frac{\frac{1}{R_1}R_2}{1 - -\frac{1}{R_1}R_2} = \frac{R_2}{R_2 + R_1}$$

Hence



$$\frac{V_o}{V_s} = \frac{R_2}{R_2 + R_1}$$

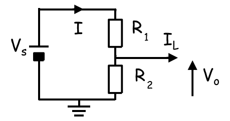
So can model as single block



If $R_1 = 9k\Omega$ and $R_2 = 1k\Omega$,

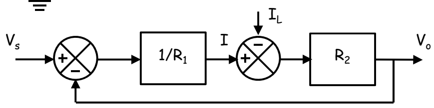
$V_o = V_s / 10$

But what if PD drives another circuit



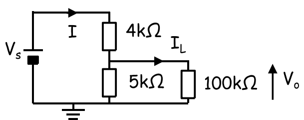
$I = \frac{V_s - V_o}{R_1}$

$V_o = R_2 * (I - I_L)$, so

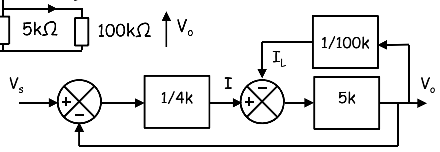


If $I_L = 0$, $\frac{V_o}{V_s} = \frac{\frac{1}{R_1}R_2}{1 - -\frac{1}{R_1}R_2} = \frac{R_2}{R_2 + R_1}$ If $V_s = 0$, $\frac{V_o}{I_L} = \frac{-R_2}{1 - -\frac{1}{R_1}R_2} = -\frac{R_1 R_2}{R_2 + R_1}$

Connect Resistor to get Load Current

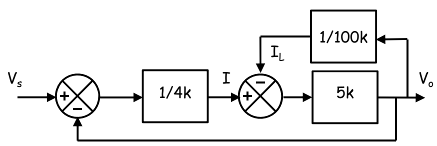


NB $I_L = Vo/100k$



We can find $\frac{V_o}{V_s}$ as this = $\frac{\text{Forward}}{1 - \text{Main Loop} - I_L \text{ Loop}}$

Lecture 5 – In Class Exercise



Find 3 TFs, and hence V_o/V_s. Find V_o if V_s = 9V with/without 100k

Forward = $\frac{1}{4k}5k$

Main Loop = $-\frac{1}{4k}5k$

I_L Loop = $-5k\frac{1}{100k}$

So $\frac{V_o}{V_s} = \frac{1.25}{1 + 1.25 + 0.05} = \frac{1.25}{2.30}$

If V_s = 9V, V_o = 4.89

Without 100K load, V_s =

Analysis Using MATLAB

MATLAB is an interactive package that you will use in labs, etc

It processes matrices, which have one or many numbers

At the prompt (>>) you can type commands to

assign values to variables, and/or to perform calculations

call functions provided by MATLAB or ones you write (like C)

plot graphs, etc

You can define a suitable graphical interface …

NB 35 is a number

[35, 8, 1, -8] single row vector (matrix) 4 columns

[35, 67; 89, 88; 3, -1*5] 3 row 2 column matrix (; for new row)

Using MATLAB on Control Systems

```
>> C = 11; P = 9; I = 2; D = 5;
>> O = I * (C * P) / (1 + C * P)
O =
   1.9800
>> O = D / (1 + C * P)
O =
   0.0500
>> O = I * (C * P) / (1 + C * P) + D / (1 + C * P)
O =
   2.0300
```

';' so no output shown

Do calc, assign to O display result (as no ';')

{ could extend easily to cope with V in multi loop system, or F for feed forward }

Or to see effect of change in P

```
>> C = 50; P = 2; Pa = P * 1.1;
>> OverI = C * P / (1 + C * P)
OverI =
   0.9901
>> OverD = 1 / (1 + C * P)
OverD =
   0.0099
>> OverIa = C * Pa / (1 + C * Pa)
OverIa =
   0.9910
>> PCchange = 100 * (OverIa - OverI) / OverI
PCchange =
   0.0901
```

Set Values

Calc O/I

Calc O/D

Calc O/I new P

Calc % change in O/I

Can Write MATLAB Function – and call it

```
function FBSys (C, P, pc);           % store in file fbsys.m
% FBSYS (C, P, pc)
% Analyse system with gains C and P noting effect if P change by pc%,
% Prof Richard Mitchell 12/7/05
den = (1 + C * P);                   % calculate values
OverI = C * P / den;
OverD = 1 / den;
Pa = P + P * pc/100;                 % find new P
OverIa = C * Pa / (1 + C * Pa);      % find new O/I
[OverI, OverD, OaoverI, 100*(OverIa-OverI)/OverI]
                                     % put values in matrix and print
>> fbsys(50, 2, 10)
   0.9901    0.0099    0.9910    0.0901
```


Summary

We have analysed feedback control system

We have briefly noted multiple loops

We have shown how this approach also applies to electronics

Next lecture we build on this …

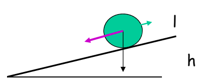
By looking at a feedback system aimed at amplifying the input

The next tutorial will start you using MatLab

Lecture 3 After Class Exercise

Here robot shown on a less steep hill.

Speed = k * v - c * (F + m * g * h / l)

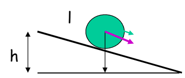


a) Suppose h is 0.5, l = 2, m = 0.1, F is 1, c = 0.1 and v = 8

What is robot speed ?

If going downhill, robot's weight will speed it up :

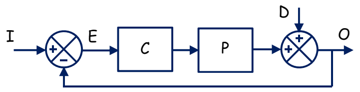
Speed = k * v - c * (F - m * g * h / l)



b) For same values as above, now what is speed downhill?

Go to Blackboard, module SE1CY15, find associated quiz and answer

Lecture 4 - After Class Exercise



Suppose

C = 27

P = 37

a) Find 1 minus Loop

b) Find O/I assuming D = 0

c) Find O/D assuming I = 0

d) Evaluate O if I = 10 and D = -5

e) Find O/I if P changed to 40