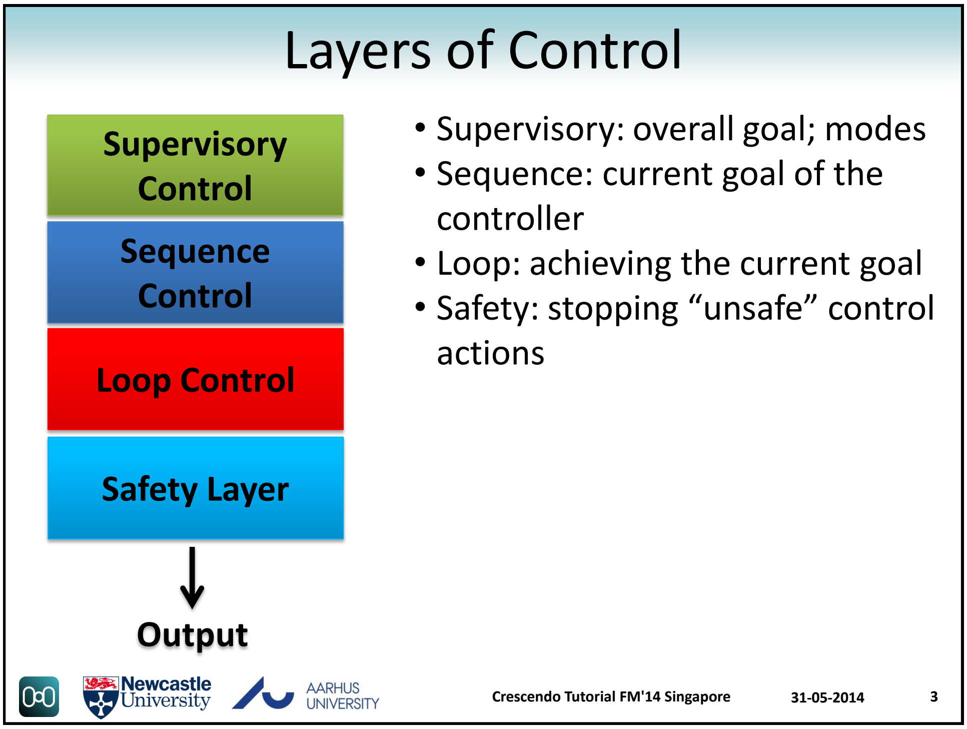
Part 2:
Real-time Control and 20-sim

John Fitzgerald
Peter Gorm Larsen
Ken Pierce

Overview

- Layers of control
  - Supervisory, sequence and loop
- Loop control
  - Open vs. closed
  - PID control
  - Sampling
  - PWM



Layers of Control

Supervisory Control

Sequence Control

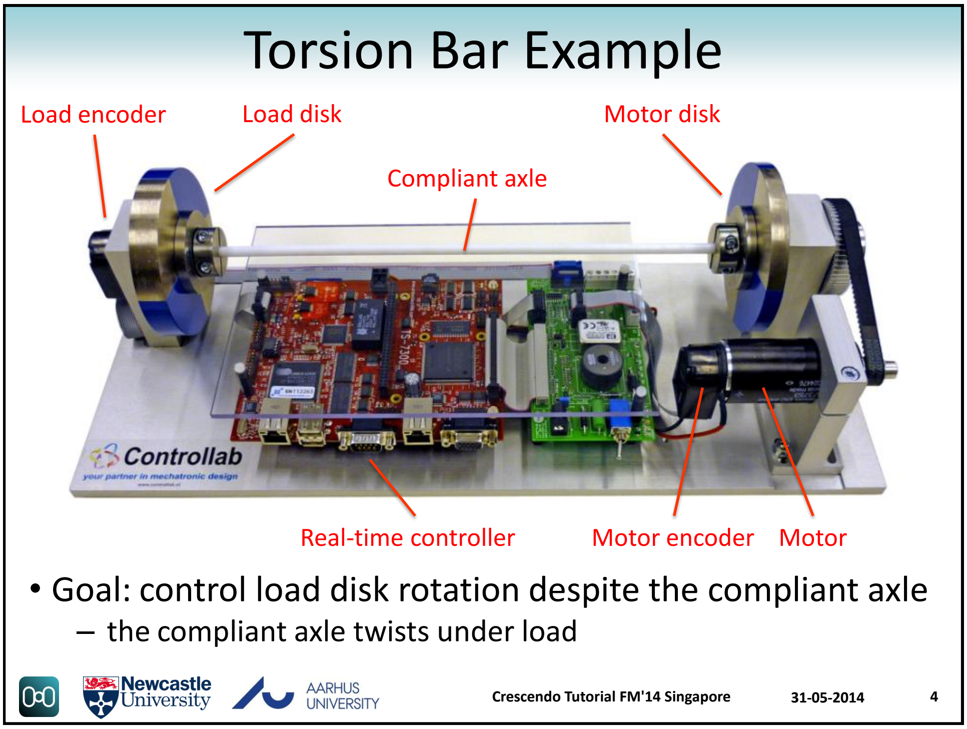
Loop Control

Safety Layer

Output

- Supervisory: overall goal; modes
- Sequence: current goal of the controller
- Loop: achieving the current goal
- Safety: stopping "unsafe" control actions

Crescendo Tutorial FM'14 Singapore 31-05-2014 3

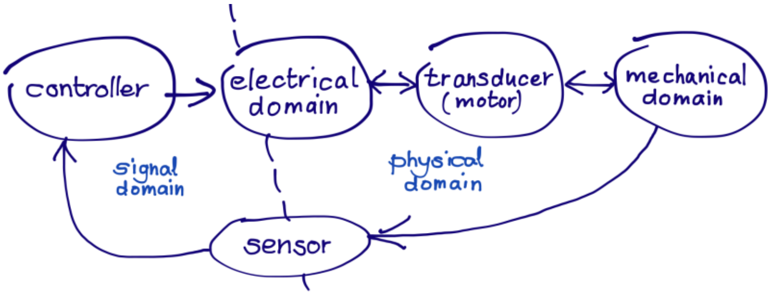


Torsion Bar Example

- Goal: control load disk rotation despite the compliant axle
  - the compliant axle twists under load

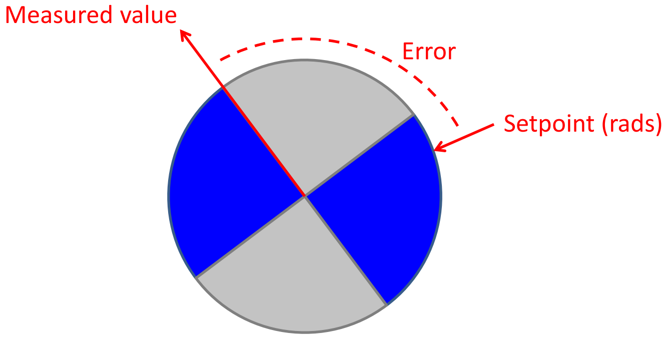
Crescendo Tutorial FM'14 Singapore 31-05-2014 4

Torsion Bar Domains

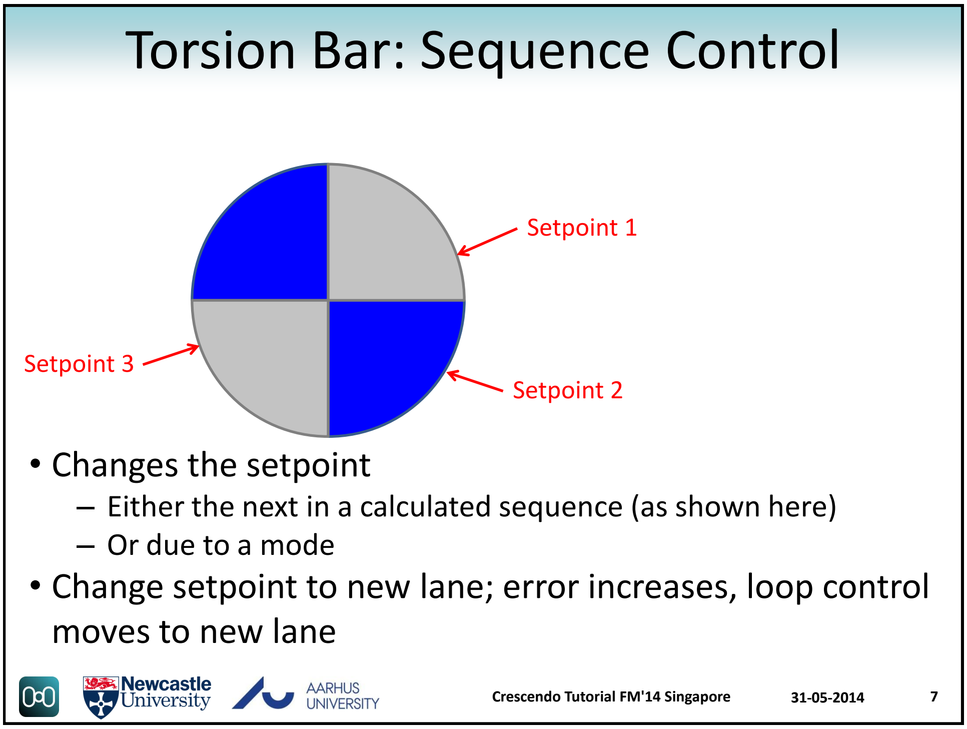


- Both domains can be modelled in DE or CT with effort
- Physical domain suited to 20-sim
- Signal domain suited to VDM when supervisory control and fault tolerance included

Torsion Bar: Loop Control

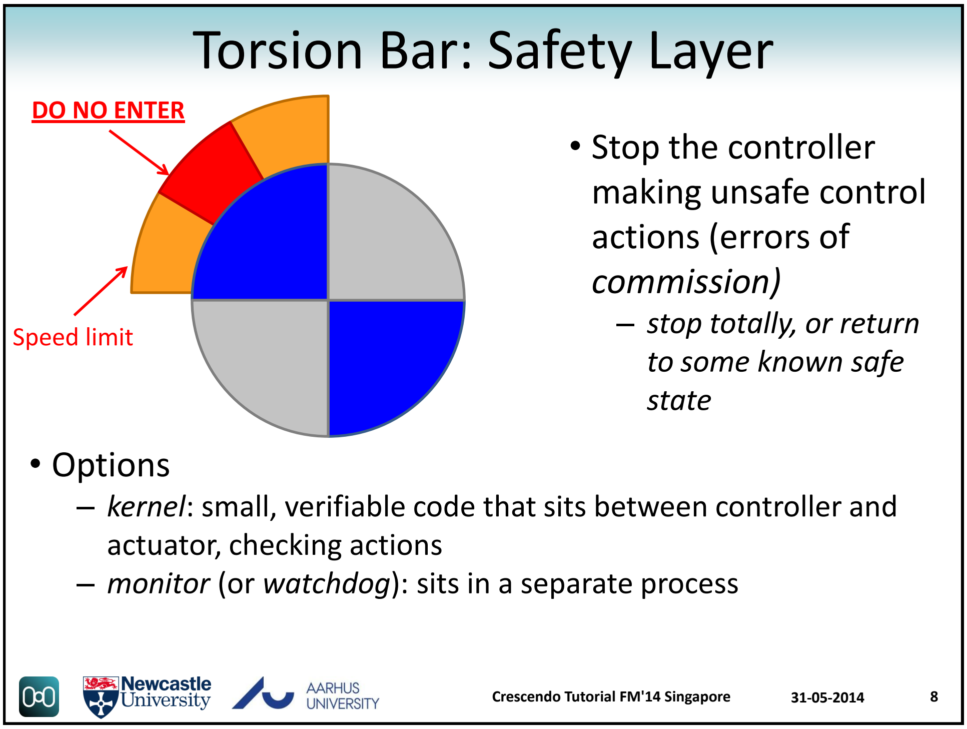


- *Setpoint*: desired position (angle in radians) of load disk
- *Measured value*: current position of load disk (encoder)
- *Error*: difference between measured and setpoint
- Control output: motor power to move load disk



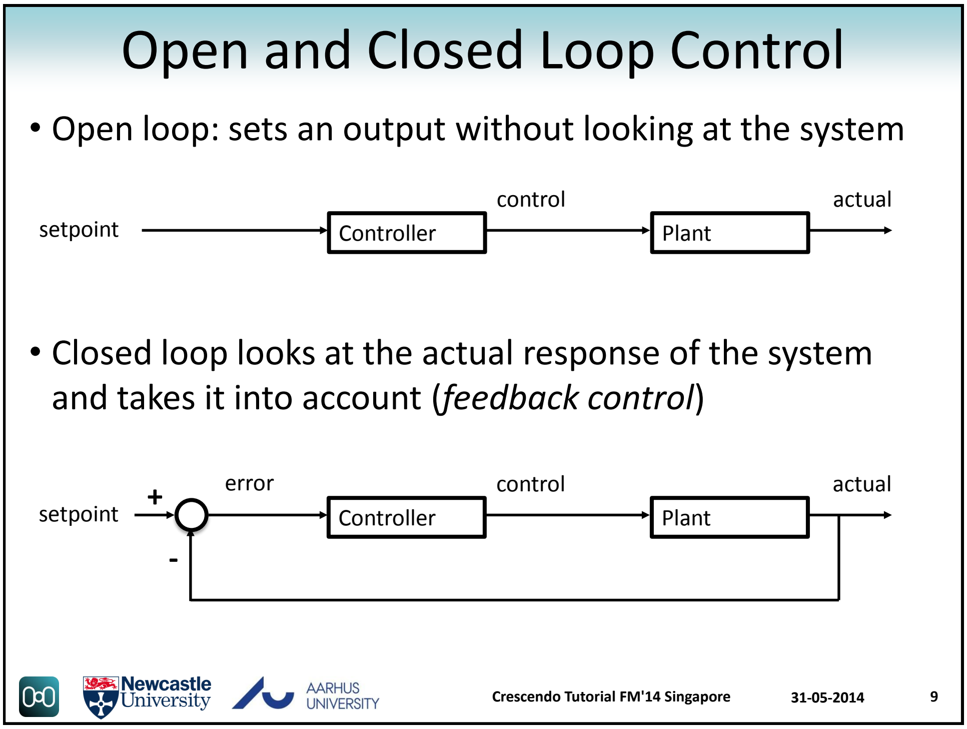
Torsion Bar: Sequence Control

- Changes the setpoint
  - Either the next in a calculated sequence (as shown here)
  - Or due to a mode
- Change setpoint to new lane; error increases, loop control moves to new lane



Torsion Bar: Safety Layer

- Stop the controller making unsafe control actions (errors of *commission)*
  - *stop totally, or return to some known safe state*
- Options
  - *kernel*: small, verifiable code that sits between controller and actuator, checking actions
  - *monitor* (or *watchdog*): sits in a separate process



Open and Closed Loop Control

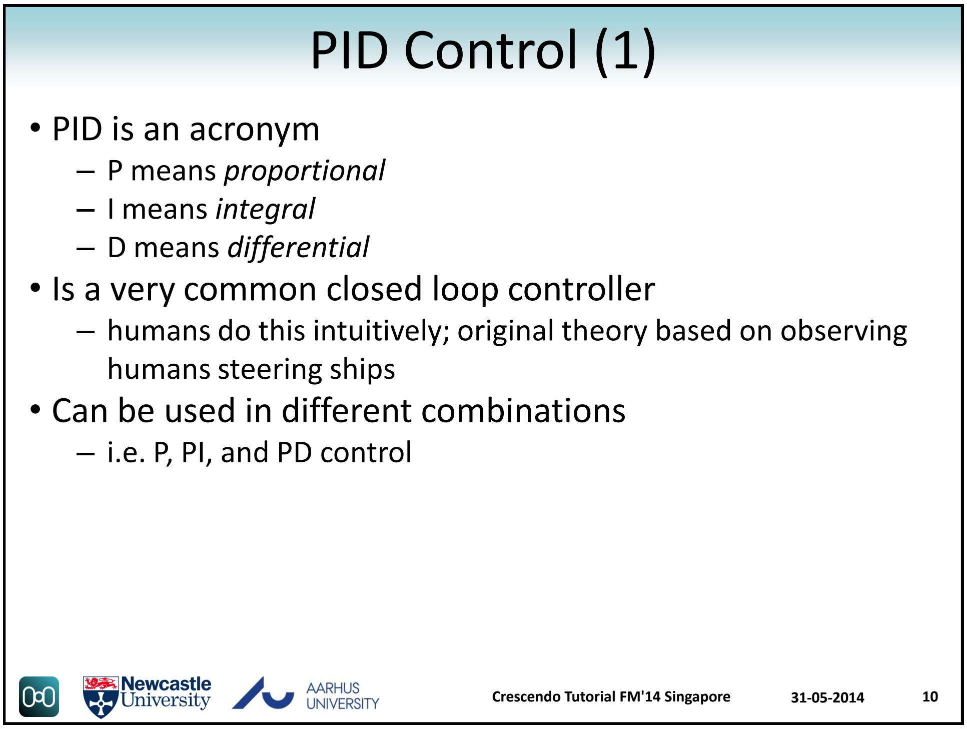
- Open loop: sets an output without looking at the system

setpoint → Controller → control → Plant → actual

- Closed loop looks at the actual response of the system and takes it into account (*feedback control*)

setpoint + → ○ → error → Controller → control → Plant → actual

-

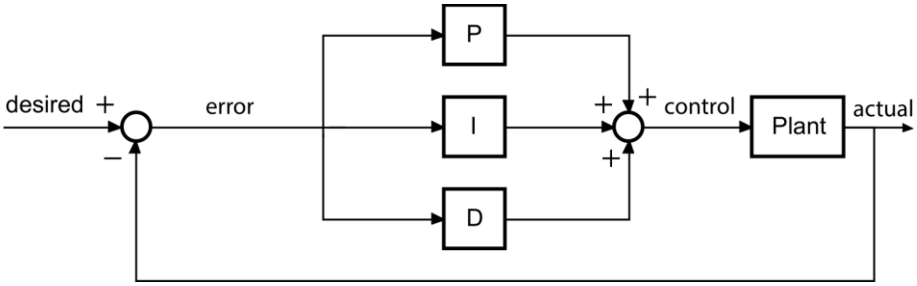


PID Control (1)

- PID is an acronym
  - P means *proportional*
  - I means *integral*
  - D means *differential*
- Is a very common closed loop controller
  - humans do this intuitively; original theory based on observing humans steering ships
- Can be used in different combinations
  - i.e. P, PI, and PD control

PID Control (2)

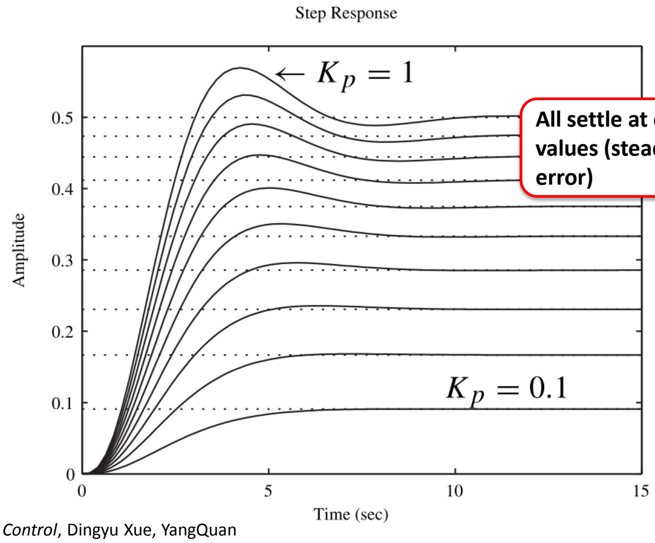
- All three elements look at the *error* (between *measured* value and *setpoint*)
  - each one influences the control output



Proportional Element (1)

- Output is proportional to error
- Multiplies the error (*ε*) by a gain (*P*)
- i.e. output = ε × *P*
- *P* is dimensionless (just a number), since actual / measured value have the same units
- Doesn’t rely on time (instantaneous error)
- On its own, proportional control tends to overshoot
- Does not settle on the setpoint (a *steady-state error*)

Proportional Element (2)



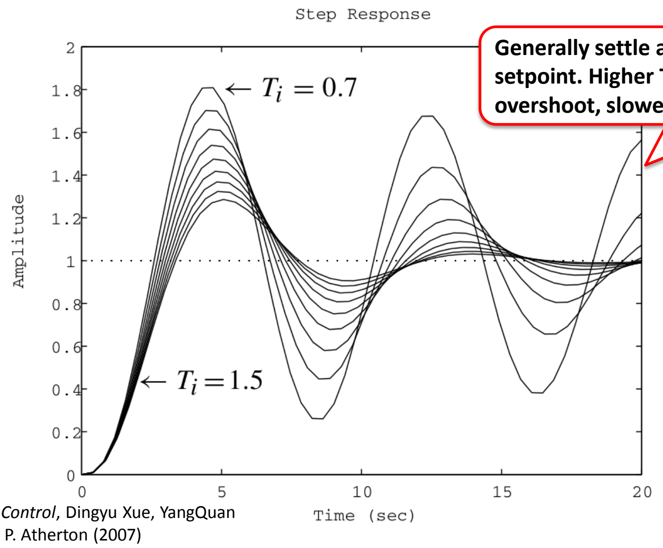
All settle at different values (steady-state error)

Linear Feedback Control, Dingyu Xue, YangQuan Chen, and Derek P. Atherton (2007)

Integral Element (1)

- Looks at the past
- Integrates previous error over time, multiplies by a gain (*I*)
  - Proportional to magnitude and duration of error
- *I* is also dimensionless
- Relies on time, hence fixed period important
  - variable instability can cause instability
- Gets to the setpoint faster, but liable to *overshoot*

Integral Element (2)



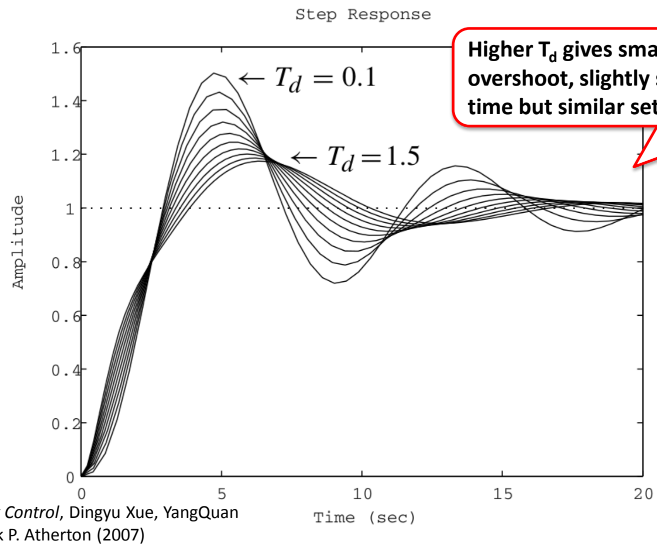
Generally settle at the setpoint. Higher T_i gives less overshoot, slower response

Linear Feedback Control, Dingyu Xue, YangQuan Chen, and Derek P. Atherton (2007)

Differential Element (1)

- Predicts system behaviour
- Looks at the rate of change of error
- Differentiates the error over time, multiplies by a gain (D)
- *D* is also dimensionless
- Also relies on fixed time
- Can help reduce overshoot, but sudden changes in error cause a *derivative kick* (since predicted behaviour was wrong). Low-pass filter can help smooth input.

Differential Element (2)



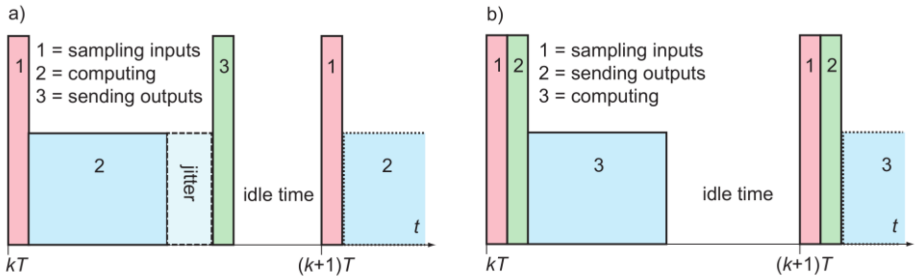
Higher T_d gives smaller overshoot, slightly slower rise time but similar settling time

Linear Feedback Control, Dingyu Xue, YangQuan Chen, and Derek P. Atherton (2007)

Sampling (1)

- Real-time controllers must:
  - Observer the environment / plant, i.e. read (sample) sensors
  - Perform control computations (e.g. PID control)
  - Send output, i.e. write actuators
- Fixed time intervals are important
  - e.g. PID performs calculations assuming a constant sample time

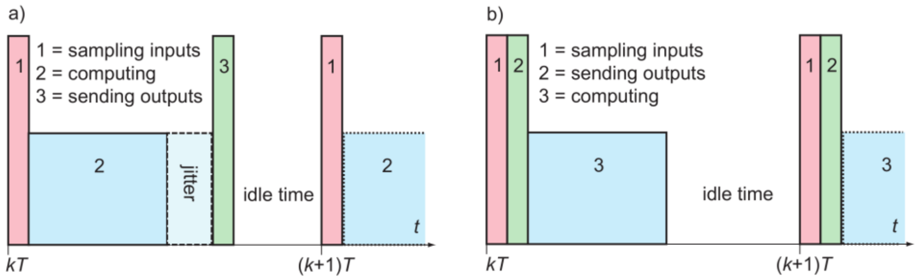
Sampling (2)



a) sample, compute, send outputs

- natural solution
- **but** causes *jitter* (time difference each time)
- acceptable if jitter is small enough for the controlled system, e.g. if system is relatively stable

Sampling (3)



a) sample, compute, send outputs

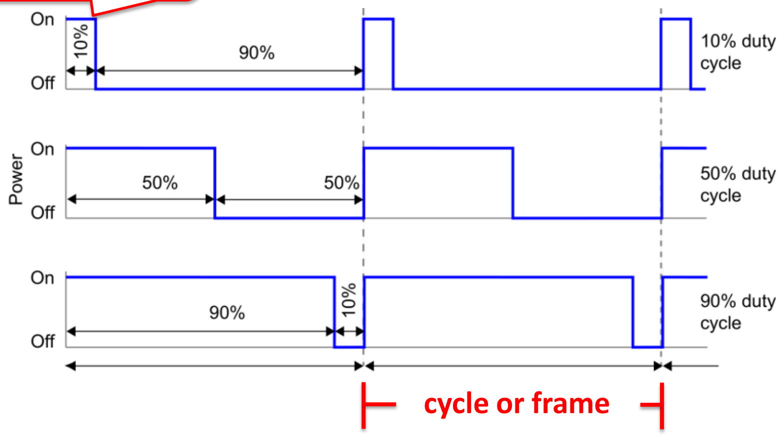
- natural solution
- **but** causes *jitter* (time difference each time)
- acceptable if jitter is small enough for the controlled system, e.g. if system is relatively stable

PWM (Pulse Width Modulation)

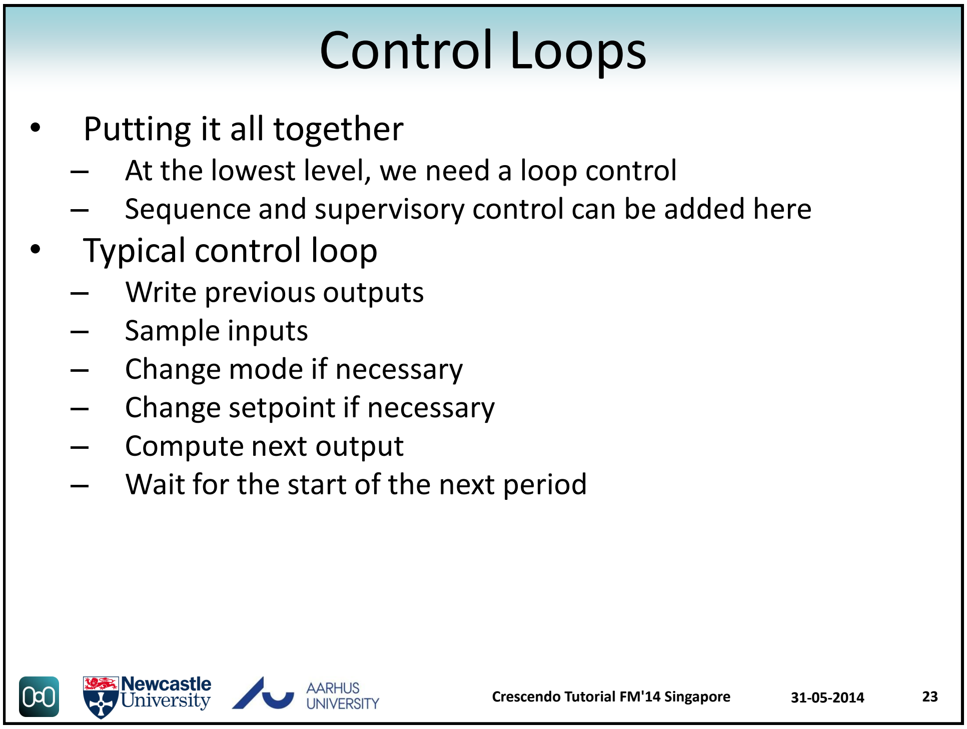
- Digital or binary output are either on or off
  - e.g. digital input to a high-voltage motor
- Variable power can be achieved by turning the output on and off
- The percentage of time the power is switched on is called the *duty cycle*

PWM (2)

Control loop frequency ideally faster than the smallest duty cycle

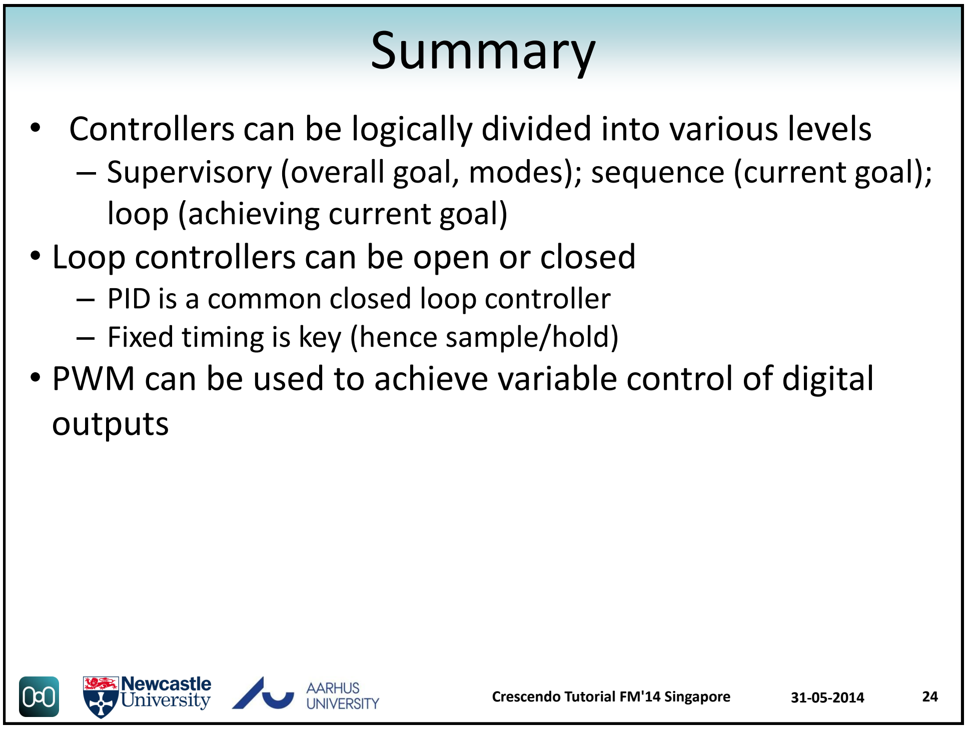


- PWM output with three different duty cycles (giving 10%, 50% and 90% power)



Control Loops

- Putting it all together
  - At the lowest level, we need a loop control
  - Sequence and supervisory control can be added here
- Typical control loop
  - Write previous outputs
  - Sample inputs
  - Change mode if necessary
  - Change setpoint if necessary
  - Compute next output
  - Wait for the start of the next period



Summary

- Controllers can be logically divided into various levels
  - Supervisory (overall goal, modes); sequence (current goal); loop (achieving current goal)
- Loop controllers can be open or closed
  - PID is a common closed loop controller
  - Fixed timing is key (hence sample/hold)
- PWM can be used to achieve variable control of digital outputs

Practical 1:
Tool Installation and 20-sim

John Fitzgerald
Peter Gorm Larsen
Ken Pierce

Newcastle University

AARHUS UNIVERSITY

Instructions

- Install the tools from the memory stick. Select 64-bit or 32-bit depending on your **Java** version:
  - Tools\64-bit
    - Crescendo-2.0.2-x86_64.exe (Crescendo for 64-bit Java and 20-sim)
    - jre-7u55-windows-x64.exe (Java 7 64-bit, if required)
  - Tools\32-bit
    - Crescendo-2.0.2-x86.exe (Crescendo for 32-bit Jaba and 20-sim)
    - jre-7u55-windows-x64.exe (Java 7 32-bit, if required)
- Follow the instructions in *Tools\Activating 20-sim.pdf*
- Extract *Practical\Practical1.zip* from the memory stick
  - this will place a TorsionBar folder on your hard drive
- Navigate to the extracted folder and follow the instructions in *Practical1-Instructions.pdf*