Principles of Mechatronic Systems Design

or

"Striking a Balance is Making Everybody Equally Unhappy"

Paul Pounds

5 March 2013

University of Queensland

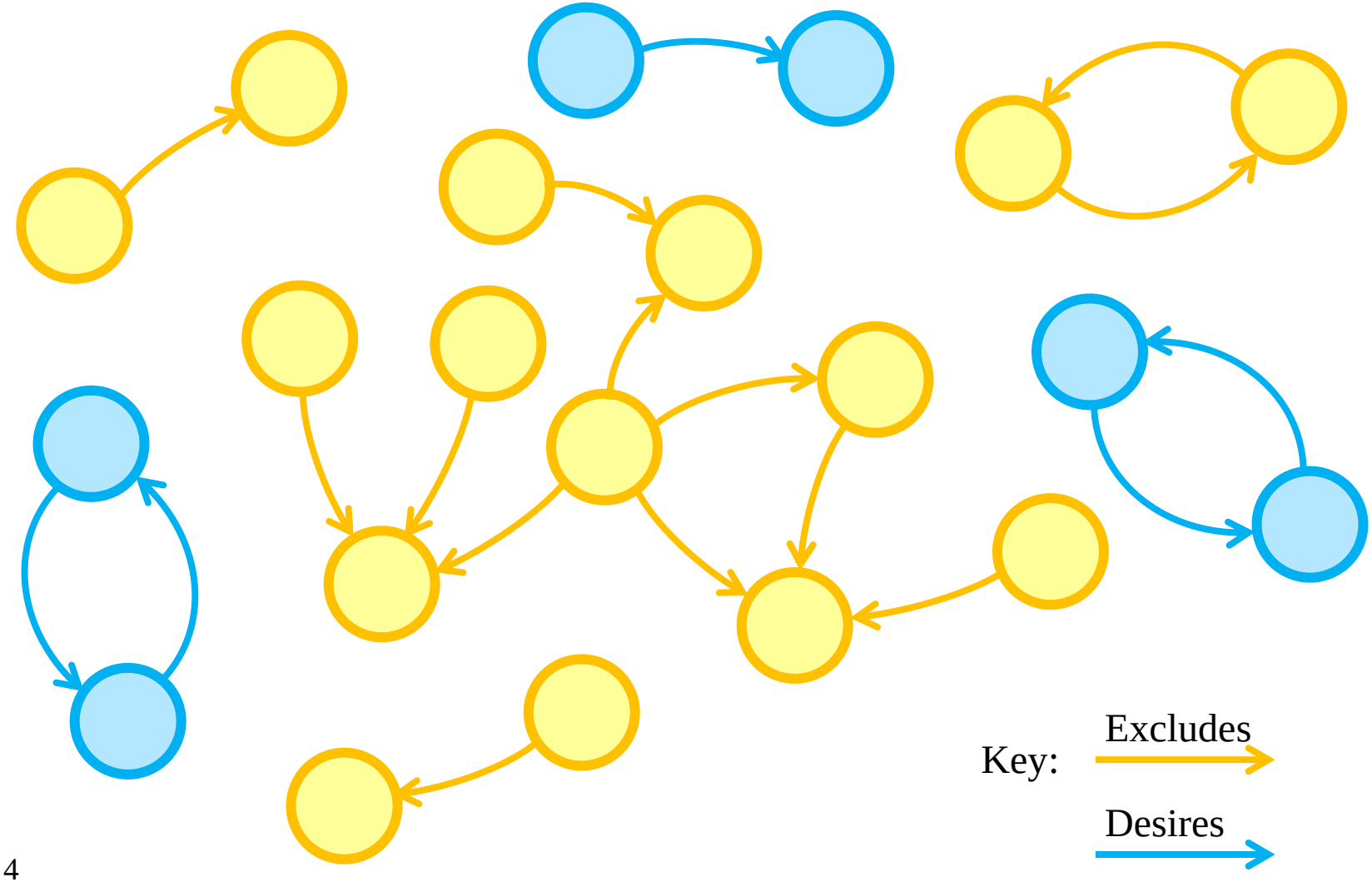
But first…

Some house keeping

House keeping

- The magical team sort algorithm satisfied all exclusion requests without tweaking
  - Things got complex *fast*
- Friend requests were not included
  - You'll thank me when you wish to throttle your teammates at 4 am the night before it's due
- Teams are now posted on Blackboard

METR3800 Directed Graph of Woe



House keeping

- Lab inductions and toolbox handout are tomorrow, starting 1 pm in Hawken c404
  - Great time/place to meet up with your team!
  - Must complete the inductions to work in the lab
  - Wear appropriate footwear
- The rules and spec have been updated
  - Make sure you're on the latest version

House keeping

- Lab sessions:
  - Tutors will be in the labs from 1 to 3 pm on Wednesdays and Thursdays
  - Morning contact sessions are times for holding meetings in, and for seminars and reviews
  - The lab will be accessible throughout the week, so you will have lots of opportunities to work

House keeping

IMPORTANT CLARIFICATION

- Our tank liner supplier has *FAILED* us
  - Ordered: 2.1m x 4m pool liner
  - Received: 2m x 3.7m pool liner
  - Outcome: Maps and terrain must be changed
  - Paul status: Cantankerous

ETA on tank: end of March

Calendar at a glance

You are here →

Week	Dates	Lecture	Reviews	Demos	Assessment submissions
1	25/2 – 1/3	Introduction			
2	4/3 – 8/3	Principles of Mechatronic Systems design			
3	11/3 – 15/3	By request			Design brief
4	18/3 – 22/3	By request	Progress review 1		
5	25/3 -29/3	By request			
Break	1/4 – 5/4				
6	8/4 – 12/4	By request	Progress seminar		
7	15/4 – 19/4	By request		25% demo	
8	22/4 – 26/4	By request			
9	29/4 – 3/5	By request	Progress review	50% demo	
10	6/5 – 10/5				
11	13/5 – 17/5			75% demo	Preliminary report
12	20/5 – 24/5				
13	27/5 – 31/5	Closing lecture		Final testing	Final report and addendum

← Next deliverable

Begin the terror

- Design Brief is due *next week*
- First progress review is the week after that

DON'T PANIC

But do get started!

Frequently Asked Questions

Lots of questions so far

FAQ Roundup

- **What is the maximum/minimum depth of the tank?**
  - 150 mm deep, 0 mm at the shoreline (obviously)

- **Are progress reviews group assessments or individual?**
  - Individual assessments, outlining your own personal contributions. During the progress reviews, I expect each student will talk for ~5 minutes describing what they set out to achieve, how it fits into the group's design strategy, work they have done so far and what their results were. Ideally they should also describe what they aim to have done by the next review.

- **Is there a limit to the height of the vessel?**
  - Nope, but you'd want it to be under 1.5 m to avoid hitting the LED panels
    (Wind generators will be of variable height, approximately at tank-level)

FAQ Roundup

- **Does the vessel have to have a certain draft?**
  - Nope – you can build a deep-keeled schooner or a shallow raft or whatever works.
- **Do stabilising flaps/fins on the hull count as part of the hull?**
  - So long as they fit in the bounding box <20mm above the waterline, sure
- **Would retractable water brakes or anchors be considered part of the hull? Can these be powered by a pre-charged battery?**
  - Sure, provided their maximum extension is within the bounding box <20mm above the water line. Assuming they are not used for propulsion, Spider-man style, sure
- **Must mast actuators be wind powered or can they use a battery?**
  - If they are not providing propulsion, they can use a pre-charged battery

FAQ Roundup

- **Can rudders, actuators for extending and retracting fins, and other gadgets such as rotating mounts for sensors (that do not propel the vessel) be powered from a pre-charged battery?**
  - Sure – only things that provide propulsion have to be wind-powered
- **How would parts machined from a small portion of the purchased raw material be billed?**
  - Cost the material at the fraction used – eg. charge half a $5 block of wood as $2.50
- **Can we make a submarine?**
  - Sure, so long as it's water bouyant
- **Can we propel our craft with exploding electrolytic capacitors?**
  - So long as they are charged by wind power, sure

FAQ Roundup

- **Are we allowed to release oil or sludge contained in our vessel to hinder the other teams' efforts, use it for "buoyancy"?**
  - 1. That's impolite
  - 2. You would be disqualified for damaging the tank and its contents
  - 3. It's not a competition; everyone can win. Hurting others won't actually help you any, and will only make them less inclined to give you assistance if you need it.

- **What does a Mexican salamander have to do with a game of marbles let alone treasure?**
  - Salamanders are awesome

Back to that design thing…

Mechatronic Systems Design

Lolwut?

What is design, anyway?

- Design:
  - n. A goal or intention
  - v. The process or creating a plan

- In engineering contexts, design is both the process and the end product of formulating a technological solution to a problem
  - Engineering design is the application of scientific knowledge to satisfy a goals

Things that are designed

- Devices/structures
- Materials/chemicals/substances
- Processes/formulae
- Documentation/procedures/policies
- Specifications/guidelines/standards
- etc.

The common thread:

"Things that make stuff work"

Design is purposeful

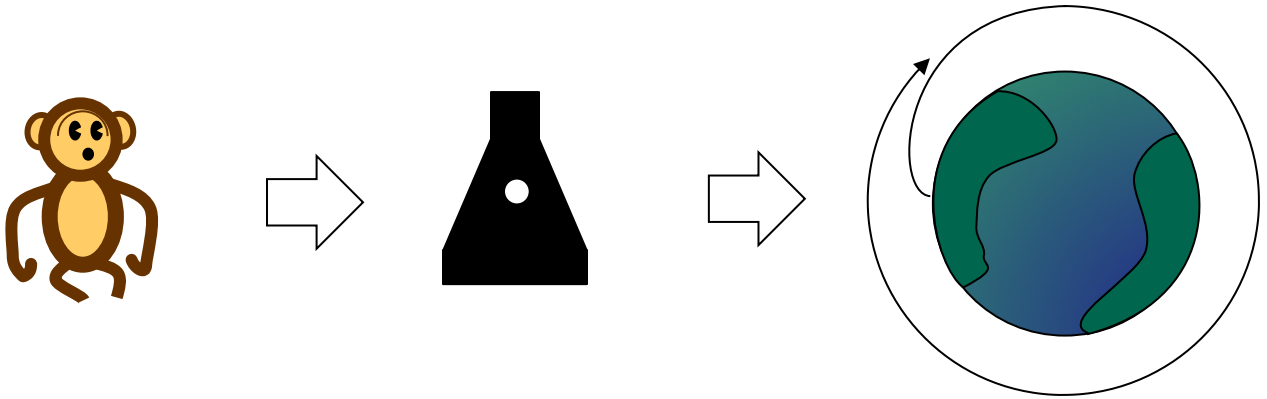
- All design has some end goal

  eg.

  – Providing power to 100,000 homes

  – Moving 1 Megahuman across a city twice a day

  – Putting a catarrhine into orbit



Design is constrained

- All design is constrained
  - With no constraints – no limits – anything is achievable without need for planning
  - Design is needed when failure is possible

    contraque: no need to ‘design’ trivial things
- Common constraints:
  - Time
  - Budget
  - Labour
  - Materials
  - Energy
  - Logistics
  - Machine access
  - Technology
  - Political capital

A way of thinking about design

Design is the dual of critique:

Analysis		Synthesis
• Specification	⟷	Implementation
• Deconstruction	⟷	Constitution
• Parameterisation	⟷	Optimisation

These are tools and philosophies for thinking about design, not a cookbook or an excuse not to use your brain

Specifications

- The precise statement of exactly what the system will do
  - Often worked out collaboratively with the engineering team
- Precision is key
  - Reduce uncertainty as much as possible
  - Avoid “feature creep”
  - Clients often don’t know what they want (and sometimes change their minds halfway!)

Design brief

- Design briefs communicate the specification of an engineering problem to the engineer
- Describes what must be done
  - Provides precise requirements and constraints
  - Specify metrics to assess success
- Preliminary analysis of the problem
  - Theoretical design implications
  - Possible solutions, their risks, benefits, issues

Specifications are important

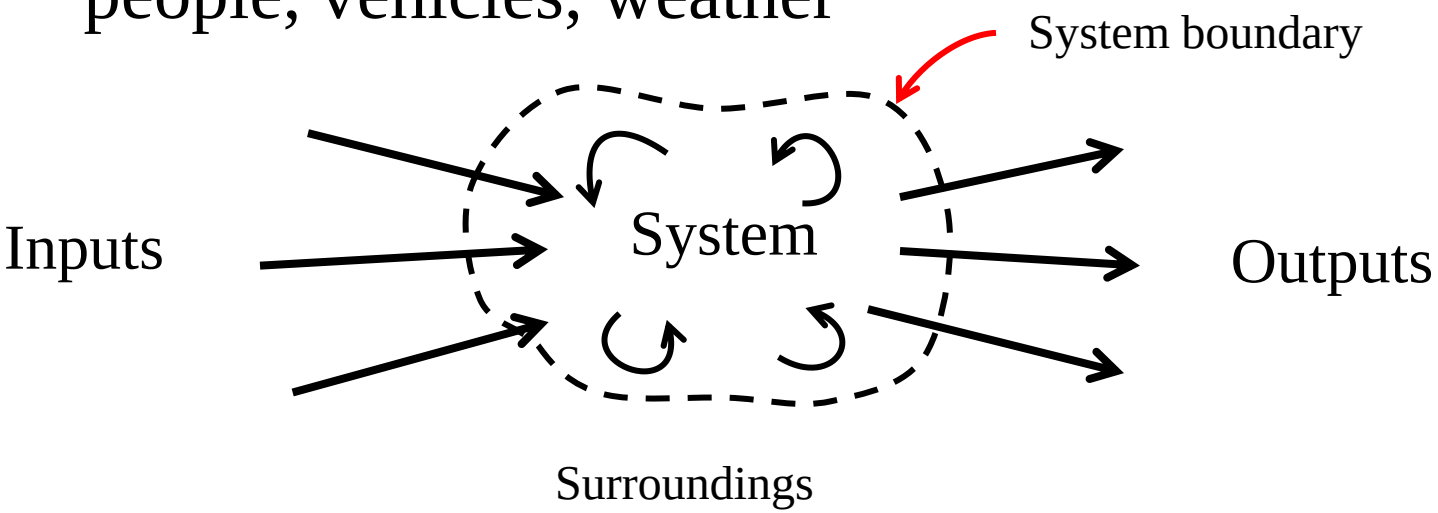
- All of your design effort is geared to meeting the specification
  - Avoids putting effort into unnecessary areas
  - Clear, complete specs' lead to better designs
  - Doubles as a performance claim to customers
- In legal disputes, meeting the specification can be a critical defence against breach of contract

Deconstruction

- Specifications provide an end-point for the design process
  - Work backwards to find a starting point
- Reductionism: break complex things down into understandable pieces
  - Find the essential parts of a system

So what about systems?

- A system is a set of interrelated elements that interact as a whole
  - eg. transport networks, computers, duct tape, people, vehicles, weather



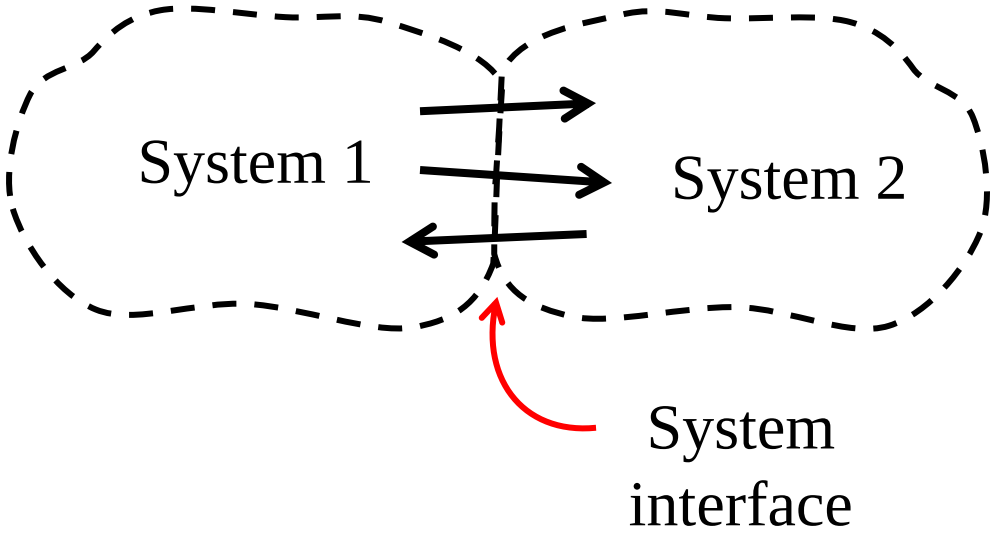
Systems engineer maxim: “Everything is a system”

Systems engineering

- Engineering the whole, rather than the parts
  - Structured way of handling complexity
  - Defines the interfaces between major components of the system
  - Abstracts performance and robustness from individual parts towards the gestalt
- Design uses systems to modify the state and behaviour of other systems

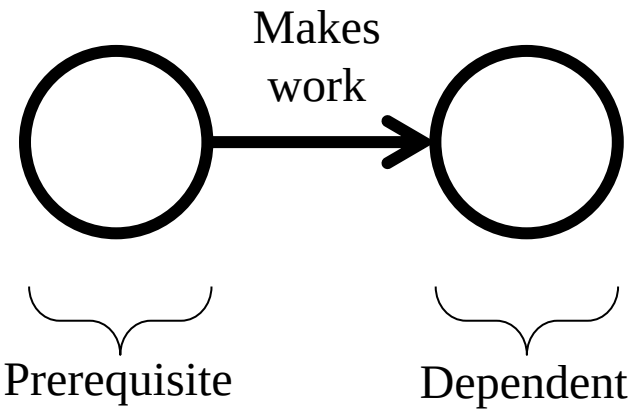
The systemic approach

- Systems decompose into networks of subsystems with touching boundaries
  - Information crosses the system interface



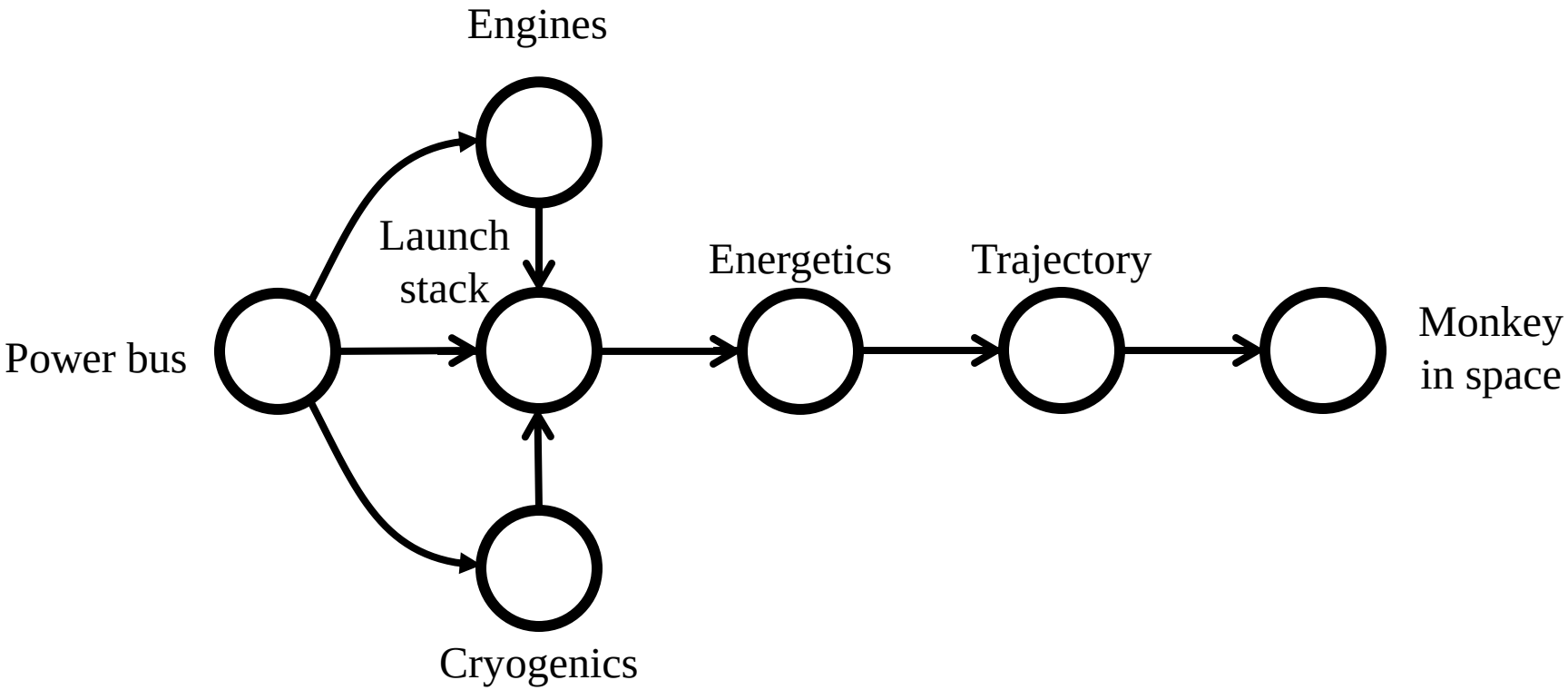
Causal system dependency

- Systems can be thought of as a cascaded series of enabling functions



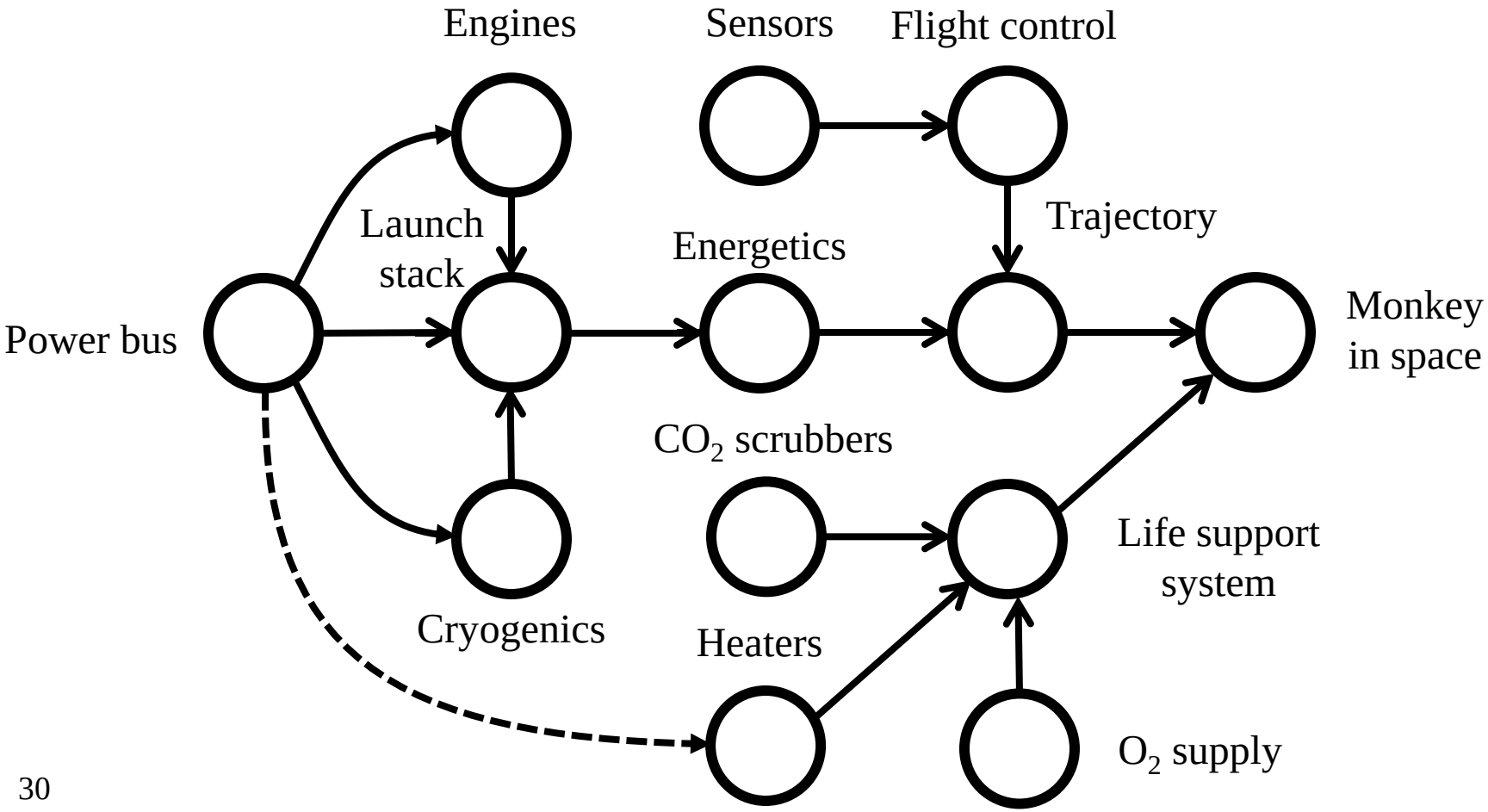
Causal system dependency

- For example:



Causal system dependency

- For example:



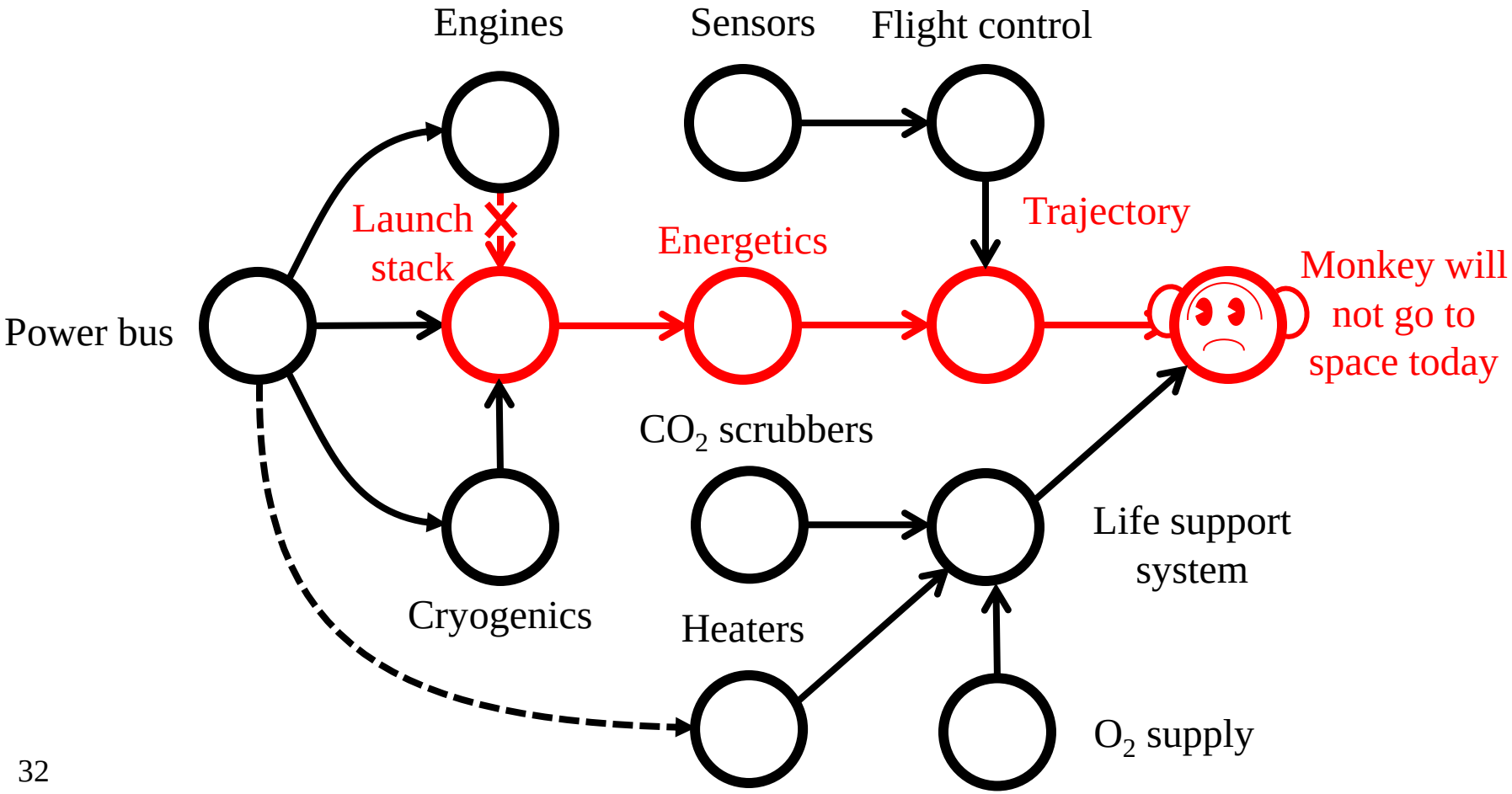
Aid to understanding failure

- We can also use causal system decomposition to understand failure

- Faults in the system propagate through directed graphs
  - Find the consequences of a failure
  - Work upstream to find the causes of a failure
  - Verify the “causal chain” to prove the system

Aid to understanding failure

- Consider a system defect:



Functional* design

- System decomposition tells us what "functions" are required by the design

- A successful design must satisfy all functions and their prerequisites

*Function as in "Thing that make things work", not as in

```
void main(void);
```

Constitutive design

- 'Constitution' turns functions into features
  - Tells you how the big picture fits together
  - The "broad strokes" of defining an approach
  - Functional requirements are a guide at best
- Designers have the most flexibility during design constitution – also the highest risk!
  - Bad structural decisions effect everything else
  - Constitution determines ~90% of system costs

Putting the pieces together

- Eg. Monkey capsule must provide oxygen, remove CO_2, and regulate temperature.
  - How big do the oxygen tanks need to be?
  - Can we use standard gas tanks or do they have to be custom-built?
  - How much separation is needed between the O_2 and the heater elements?
  - How do you get the monkey to stay in there, anyway?

A jigsaw puzzle

- Engineering is just like solving a jigsaw puzzle with many pieces, except each piece costs $1000 and can be one of a dozen shapes or completely custom-made, and if you don't solve the puzzle right, people die.

- In real life, engineering is often a process of try-and-see iteration
  - Sometimes, there is no "right" way.

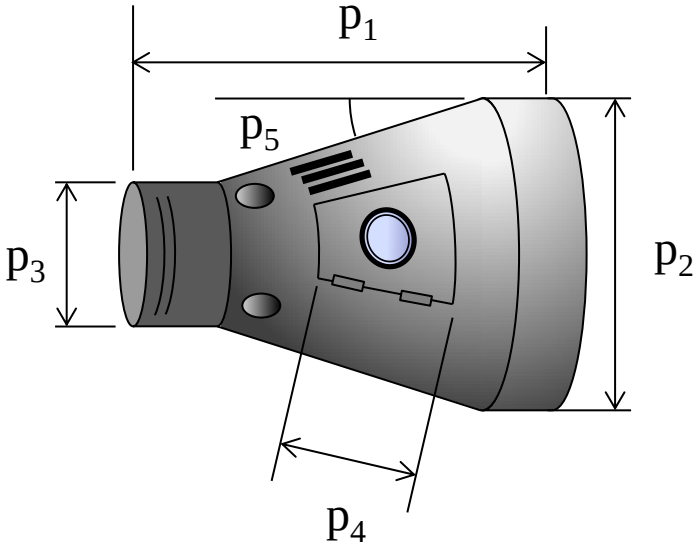
Parametric design

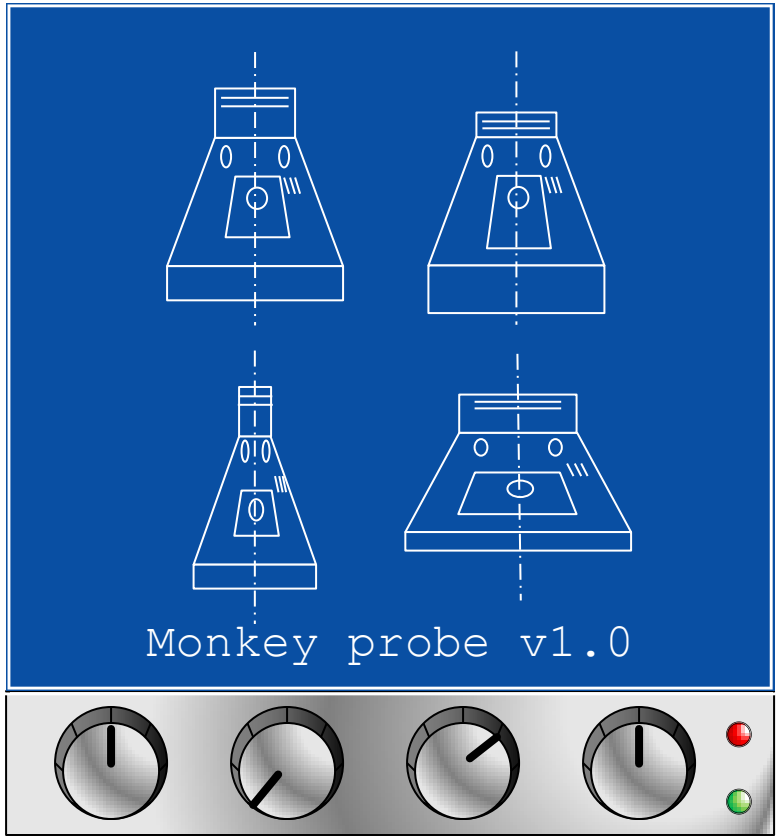
- When you *do* have a clear high-level structural concept of your solution, there are usually many unconstrained variables

- The key dimensions, values and settings that describe a design are called the “design parameters”

Parametric design

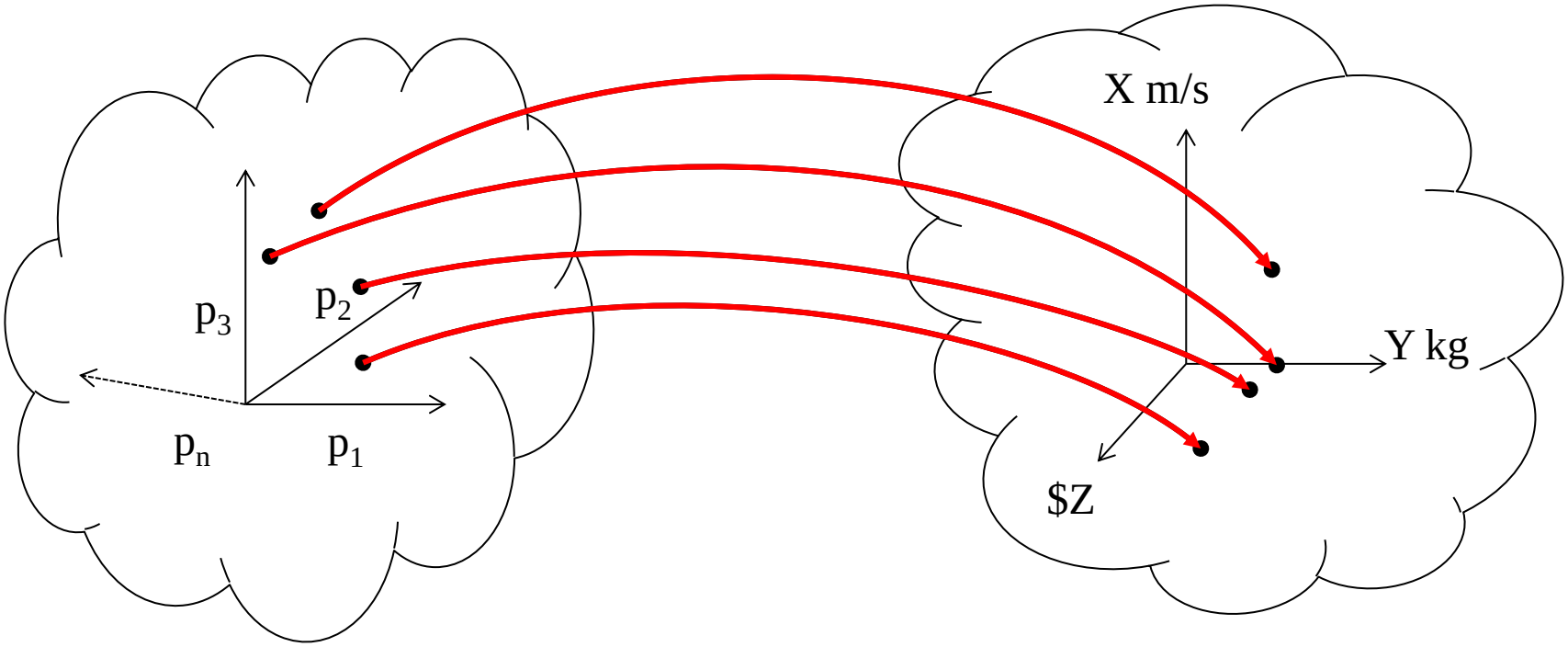
- Parameters can be thought of as a series of twist knobs that adjust the design





Design space

- Combinations of design parameters map to some realisable performance of the system



"Design Space"
(Parameter space)

"Performance Space"

The fundamental problem

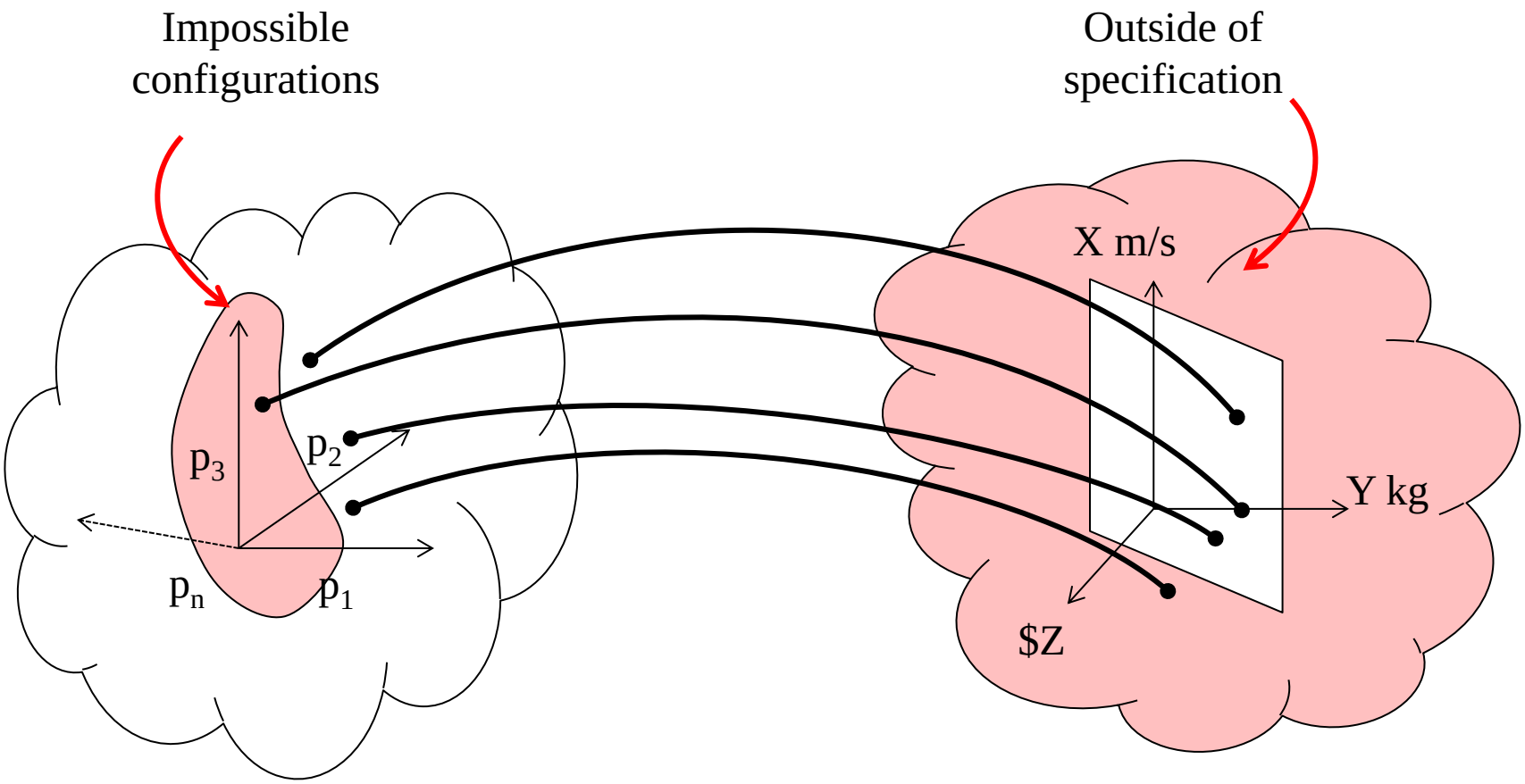
- Knowing how to set those knobs is difficult
  - Complex interactions between parameters
  - Competing design goals
  - Constraints on parameter space
- You will rarely satisfy all of your goals
  - You will NEVER meet all your ambitions

Design constraints

- University engineering problems are typically tightly constrained – they have only one “right” answer

- In the real world, engineering problems are either under-constrained (many solutions) or over-constrained (no solutions)

Design constraints



Design metrics

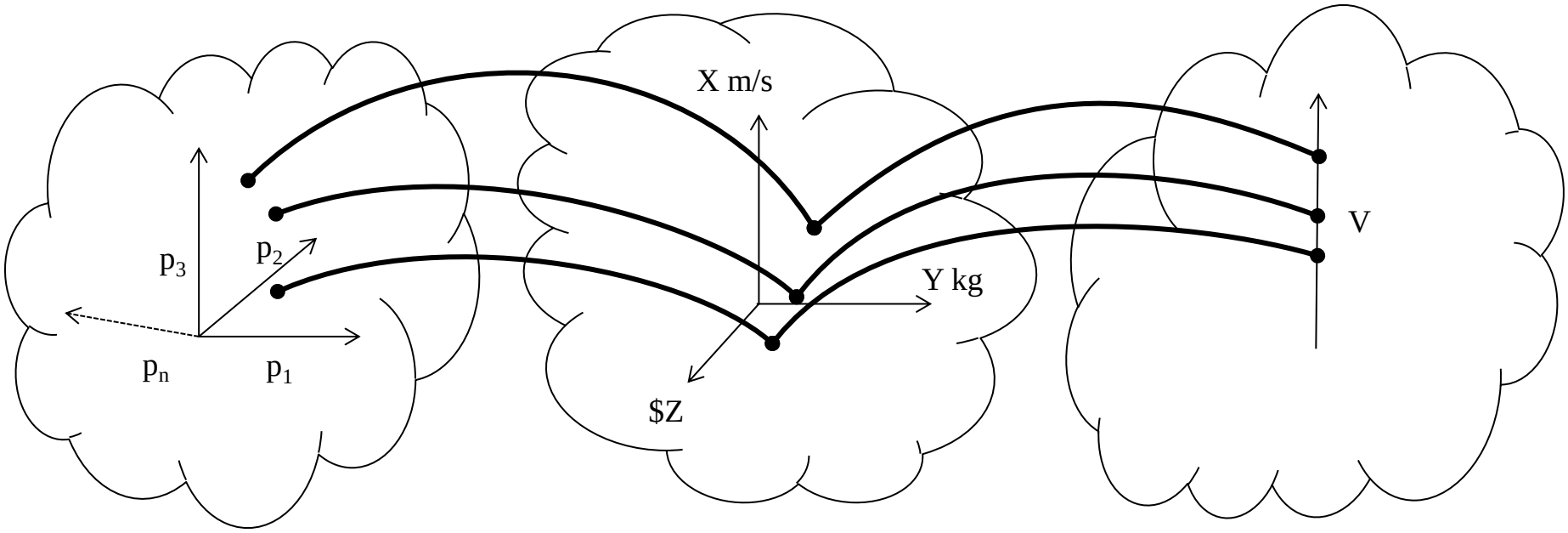
- We can make cost and value functions to encode how "good" a candidate design is
  - eg. Aim to maximise propellant bang for buck, given parameters 1 to *n*, we might use:

$$Value = \frac{explosive\ force}{unit\ cost}$$

$$V = \frac{X(p_1, p_2 \ldots pn)}{C(p_1, p_2 \ldots pn)} = f(p_1, p_2 \ldots pn)$$

$$V_{max} = \sup(f(p_1, p_2 \ldots pn))$$

Metric space*



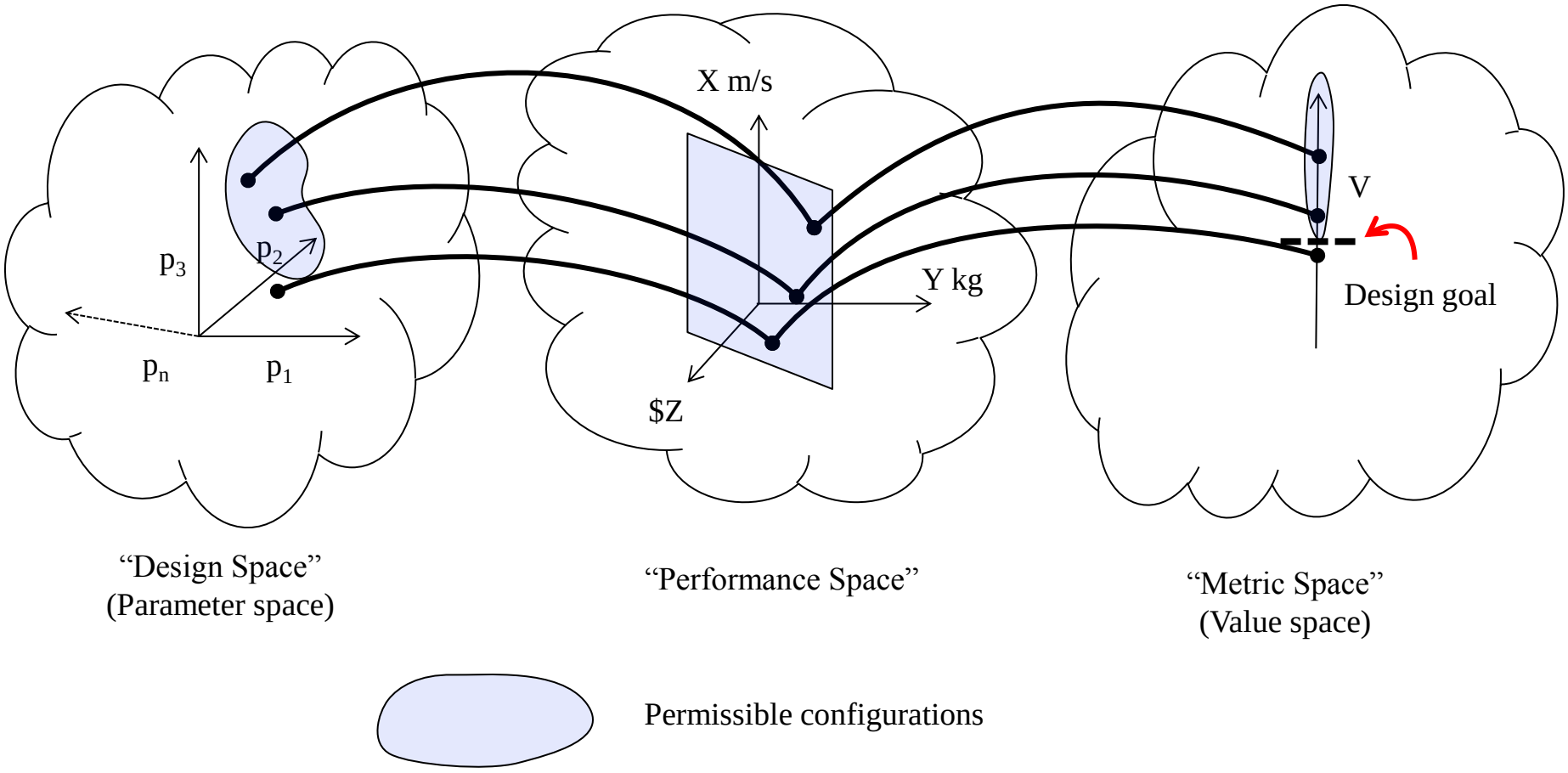
"Design Space"
(Parameter space)

"Performance Space"

"Metric Space"
(Value space)

*Not quite the same as (but not completely unrelated to) *Sur quelques points du calcul fonctionnel*

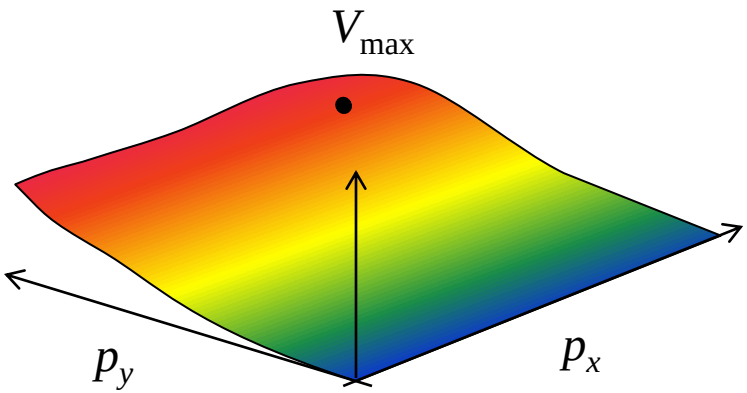
Metric space*

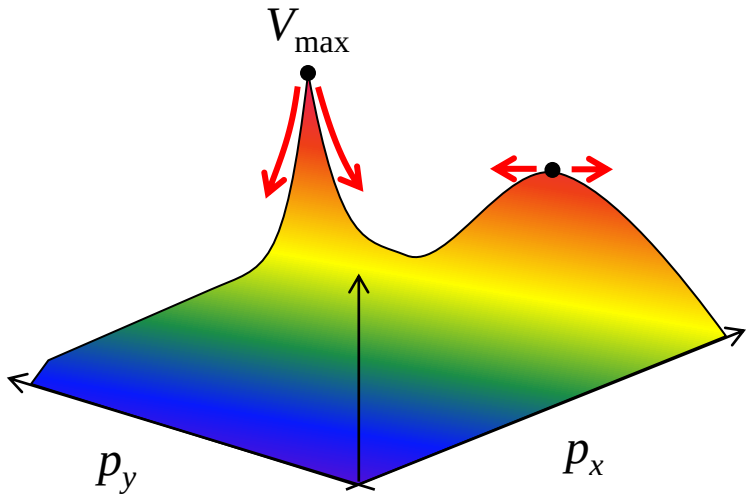


*Not quite the same as (but not completely unrelated to) *Sur quelques points du calcul fonctionnel*

Design optimisation

- For many systems, we can explicitly solve for the optimal design point
  - Estimate is only as good as your value function
  - Gradient of the value function is the design parameter "sensitivity"





Finding a value function

- How do we encode the utility of a design?
  - Highly subjective: what does “best” mean?
- Many tools for thinking about utility
  - Multi-criteria decision analysis
  - Pairwise comparison
  - Decision matrix method
  - Management by objectives

There is a whole field of “value engineering”

A quick example

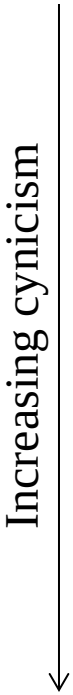
- Pair-wise decision matrix:

	Safe	Low cost	Reusable	Easy to build	Payload capacity	Score
Safe		X	X	X	X	4
Low cost			X	X		2
Reusable						0
Easy to build			X			1
Payload capacity		X	X	X		3

Prioritise by score

“D” for “X”

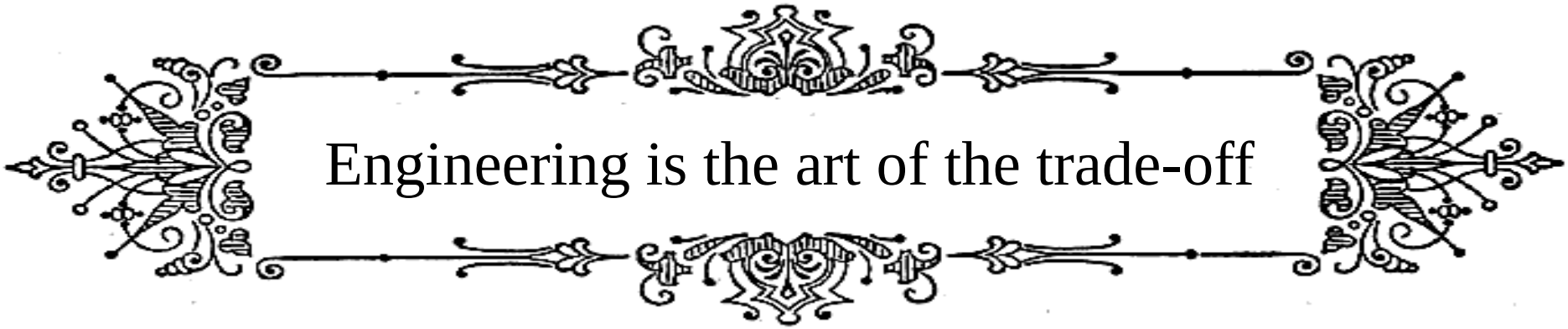
- How to choose a value function?
  - Design for performance
  - Design for manufacture
  - Design for reliability
  - Design for sustainability
  - Design for cost
  - Design for marketability
  - Design for obsolescence



Of course…

- It is relatively rare that a single value function can fully capture the complex give and take of a real-world design problem
  - Uncertain system constraints/assumptions
  - Uncertain system parameters
  - Uncertain system specifications (!)
  - Mutually exclusive goals
  - Conflicting agendas
  - Conflicting personalities

The most important truth in your degree



Methodological approaches

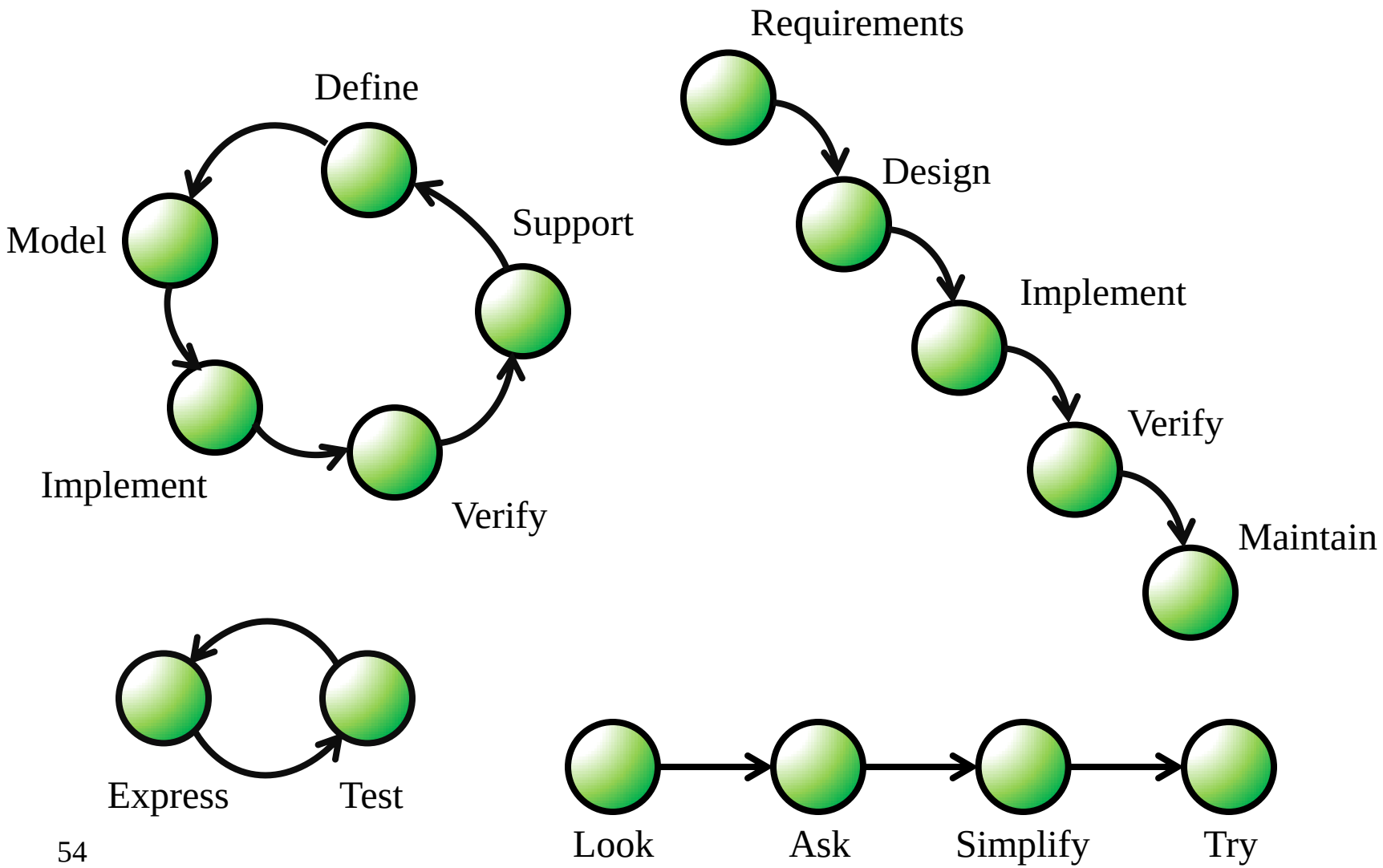
- Ok, that's great – but how do we do this trade-off thing, exactly?
  - Lots of different ways!
  - Quite likely as many design processes as there are design engineers
- Here are just a few popular design process methodologies…

Alphabet soup

- LAST: Look, Ask, Simplify, Try
- ETC: Express, Test, Cycle
- PDP: "Product Design Process"
  - Define, Model, Implement, Verify, Support
- "Waterfall Model"
  - Requirements, Design, Implement, Verify, Maintain

(And many, many, many more – each more buzzwordy, cliché and feel-good managerial-speak than the last)

Cyclic vs linear models



The common threads

1. Work out what to do
   - Specifications; the design brief – be precise
   - Understand the real constraints
2. Find a solution*
   - Iterate until you do
3. Make sure it works
   - Modelling, validation, testing
   - Good engineering practice

*Wasn't that the problem to begin with??

The synthesis step

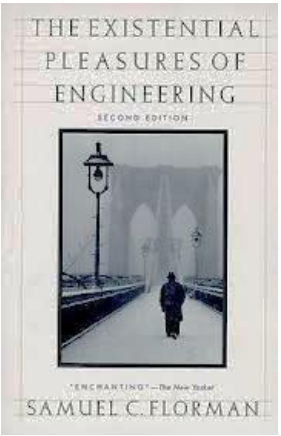
- Constitution + Optimisation = Synthesis

The messy, complicated, creative, intuitive, frustrating, marvellous, deep, ineffable, often iterative intellectual challenge that lies at the heart of all brilliant engineering solutions

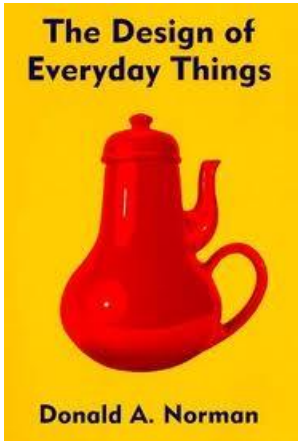
Art, *not* science: Anyone who claims they can teach you to do this is sadly misguided

Recommended reading

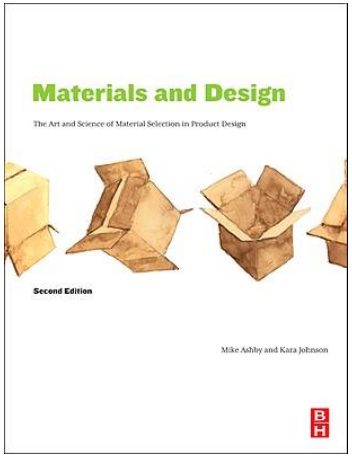
- "The Existential Pleasure of Engineering"
  - Samuel C. Florman



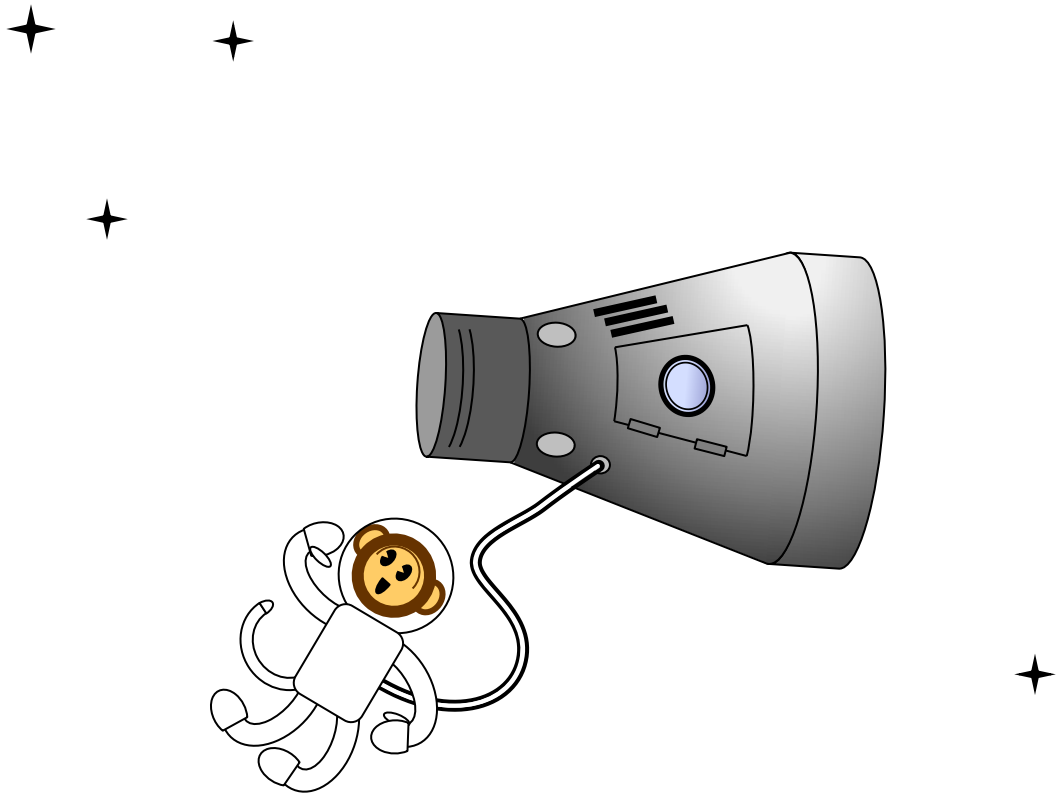
- "The Design of Everyday Things"
  - Donald A. Norman



- "Materials and Design"
  - M. Ashby and K. Johnson



Questions?



Tune-in next time for…

Principles of Sailing

or

"Main sheets, spinnakers and belaying pins ahoy!"

Fun fact: Thirty two monkeys have been launched as part of various space programs – most recently by Iran on 31st January.

Nineteen survived.