Compositional Design of Society-Critical Systems: From Autonomy to Future Mobility

Invited Seminar
Singapore Maritime Institute (SMI)

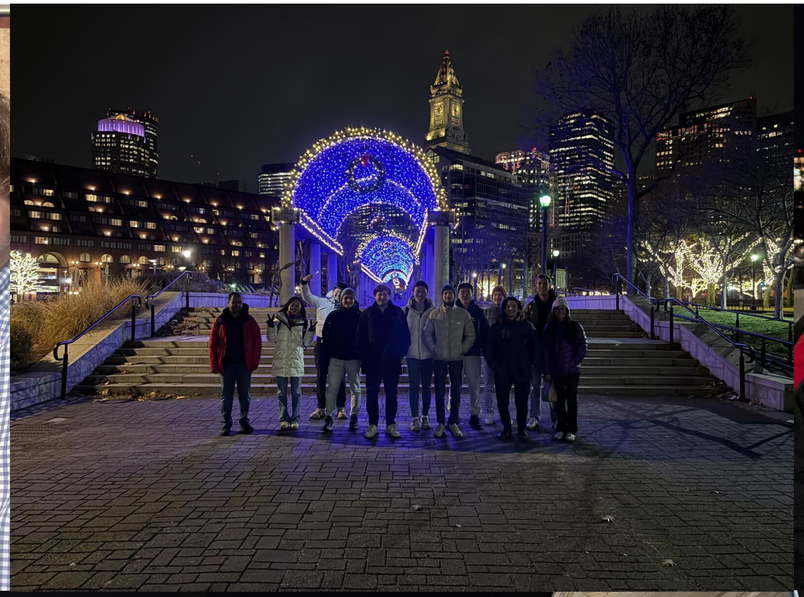
May 23, 2025

Gioele Zardini

LIDS, IDSS, CEE
Massachusetts Institute of Technology

gzardini@mit.edu - **https://zardini.mit.edu**

My lab… a bunch of amazing people that eat… and do exciting research



Need new tools to model and solve complex systems design optimization problems

Intermodal mobility systems (+ energy)

Sidara

U.S. DEPARTMENT OF ENERGY

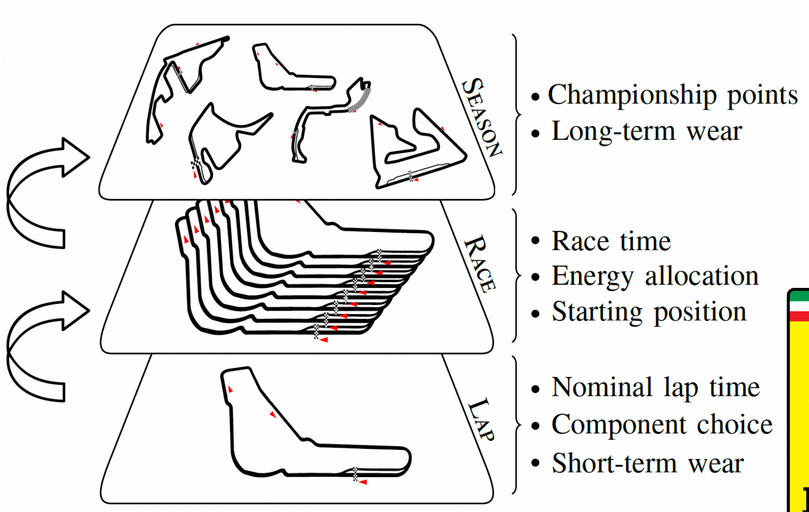
Railway + energy + built environment

Maritime Shipping

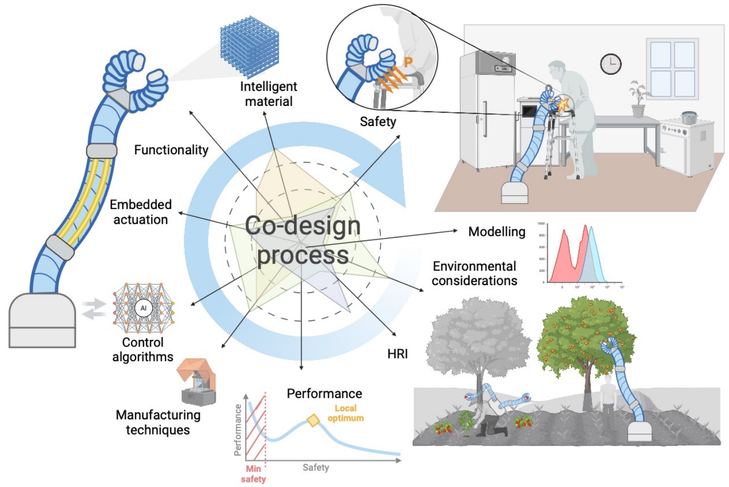
(Multi-agent) autonomy

amazonrobotics

Automotive



Ferrari



Soft robotics

Need new tools to model and solve complex systems design optimization problems

- **Societal impact** of new technologies depends on their **joint design** with **existing infrastructure and systems**

Intermodal mobility networks

Networks of ships

*Example - Autonomy: **Heaven** or **hell**?*

__30%__ of the cars would be enough

__First__- and __last-mile__ mobility could make __public transit__ more __convenient__ and __attractive__

More __affordable__, __sustainable__

Single components are slowly well understood, but we still lack a (*formal* and *practical*) theory for the **task-driven co-design** of **complex systems**

Agenda

- **Motivation**
  - *New challenges of engineering design*
  - *Motivation from autonomy and mobility*
  - *Desiderata for co-design*
- **Monotone Co-Design**
  - *Modeling design problems*
  - *Examples across domains*
  - *Design queries and optimization*
  - *From autonomy to mobility systems*
- **Strategic interactions**
  - *Game theory to deal with strategic interactions*
- **Outlook on future research**

Driven by **societal challenges**, I develop **efficient computational tools** to **automate** the **formulation** and **solution** of **large**, **complex system design problems**

Website containing all papers and more pointers:

https://zardini.mit.edu

Agenda

- **Motivation**
  - *New challenges of engineering design*
  - *Motivation from autonomy and mobility*
  - *Desiderata for co-design*
- **Monotone Co-Design**
  - *Modeling design problems*
  - *Examples across domains*
  - *Design queries and optimization*
  - *From autonomy to mobility systems*
- **Strategic interactions**
  - *Game theory to deal with strategic interactions*
- **Outlook on future research**

Driven by **societal challenges**, I develop **efficient computational tools** to **automate** the **formulation** and **solution** of **large**, **complex system design problems**

Website containing all papers and more pointers:

https://zardini.mit.edu

The vision of automated co-design

minimize
(resources usage)
subject to
(functionality constraints)

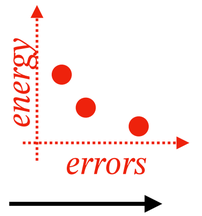
task specification
multi-domain knowledge
design options

“automated designer”

optimal design(s)

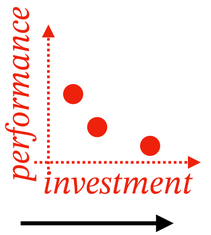
Autonomy co-design

task
robot autonomy, physics
components, algorithms



Mobility co-design

demand
networks, operations, infrastructure
mobility services, policies



Autonomy as the frontier of complexity for the co-design of complex systems

A fleet of autonomous vehicles

=

software **behavior** **coordination**
hardware
actuation **localization** **planning** **invasivity**
sensing
control **interaction** **learning** **liability**
computation
perception **mapping** **regulations**
energetics **communication** **infrastructure**

WAH LAU!!!

So many **components** (hardware, software, …), **objectives** and **choices** to make!

Nobody understands the **whole** thing!

We forget why we made **choices**, and we are afraid to make **changes** (high failure cost).

We need **faster** design cycles, **nimbler** execution.

anthropomorphization of 21st century engineering malaise

"My dear, it's simple: you lack a *practical* theory of **co-design**!"

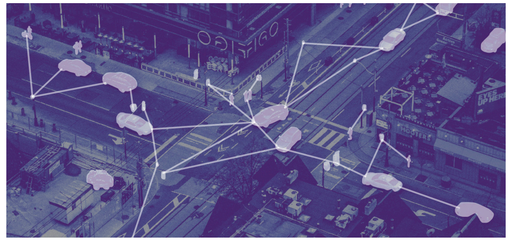
Formal,
Quantitative
Intellectually tractable

Your system is just a component in another person's system

Infrastructure level

Infrastructure choices and coordination of services

Service level



Fleet deployment

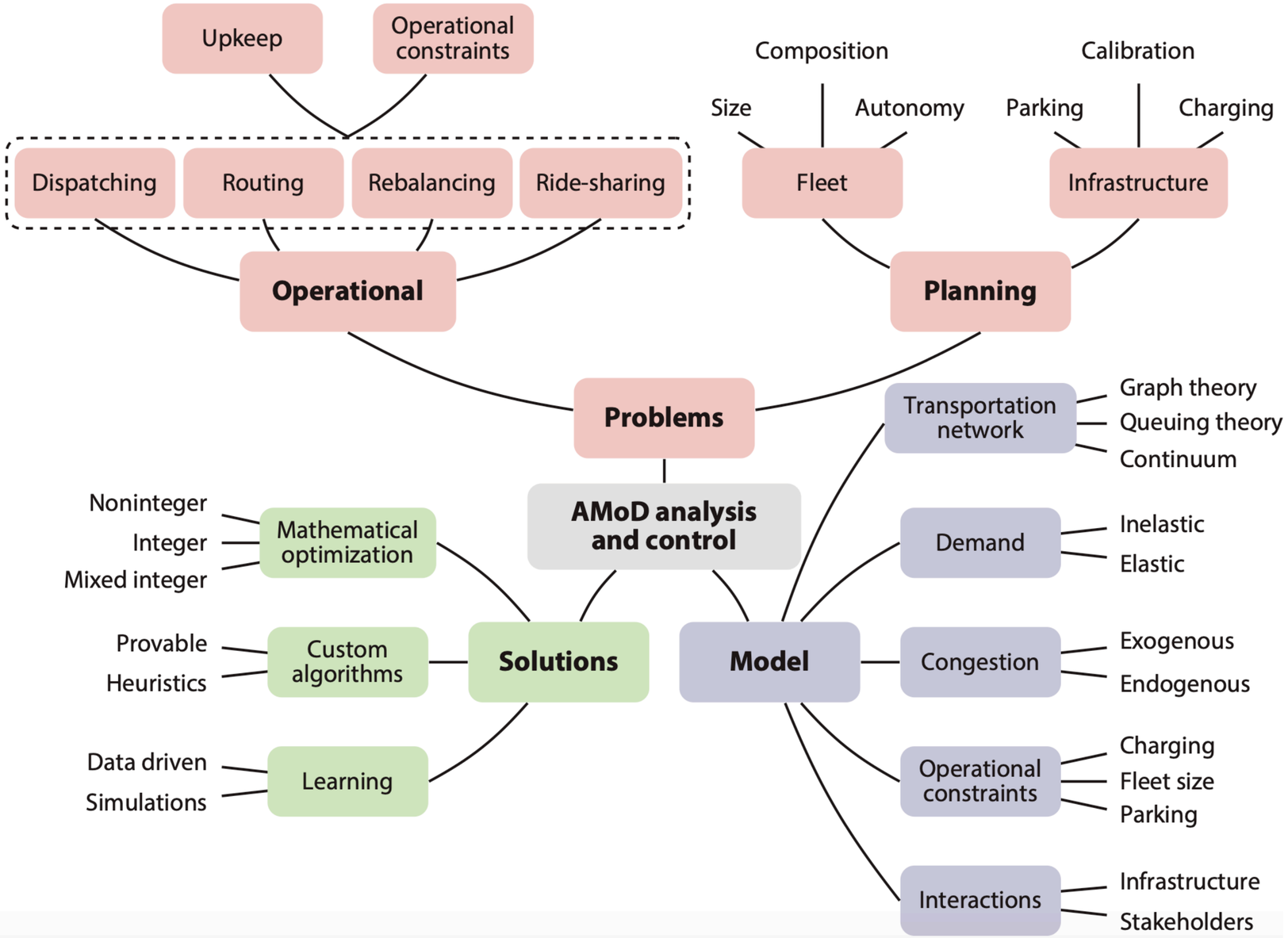
Platform level

Choice of components and algorithms

Subsystem level

Single component design

Your system is just a component in another person's system



[AnnuRev'22]

Complex systems typically feature multi-stakeholders interactions

Mobility providers (Jamey Tesler, CEO MassDOT)

Liability (Christian Mumenthaler, CEO SwissRe)

Policy makers (Michelle Wu, Mayor Boston)

Academia Tech Developers (Sally Kornbluth, MIT president)

Industry (Laura Major, CTO Motional)

Challenges for automated co-design of complex systems

Complexity when designing complex systems

Large systems

- Many components, scales
- Heterogeneous natures
- Multiple objectives

Strategic interactions

- Many agents
- Heterogeneous interactions
- Conflicts/collaborations

A fleet of autonomous vehicles

=

software, behavior, coordination, hardware, actuation, localization, planning, invasivity, sensing, control, interaction, learning, computation, perception, mapping, regulations, energetics, communication, infrastructure

Desiderata for the automation of complex systems co-design

- **Formal, domain-independent**
- **Computationally tractable**
  - Need to compute solutions efficiently
- **Compositional, hierarchical**
  - My system is a component of somebody else's system
- **Collaborative**
  - Pooling knowledge from experts across fields.
- **Intellectually tractable**
  - Not exclusively accessible to system architects
- **Continuous/adaptive**
  - Design is not static: it should be reactive to changes in goals and contexts

Agenda

- **Motivation**
  - *New challenges of engineering design*
  - *Motivation from autonomy and mobility*
  - *Desiderata for co-design*
- **Monotone Co-Design**
  - *Modeling design problems*
  - *Examples across domains*
  - *Design queries and optimization*
  - *From autonomy to mobility systems*
- **Strategic interactions**
  - *Game theory to deal with strategic interactions*
- **Outlook on future research**

Website containing all papers and more pointers:

https://zardini.mit.edu

Complexity when designing complex systems

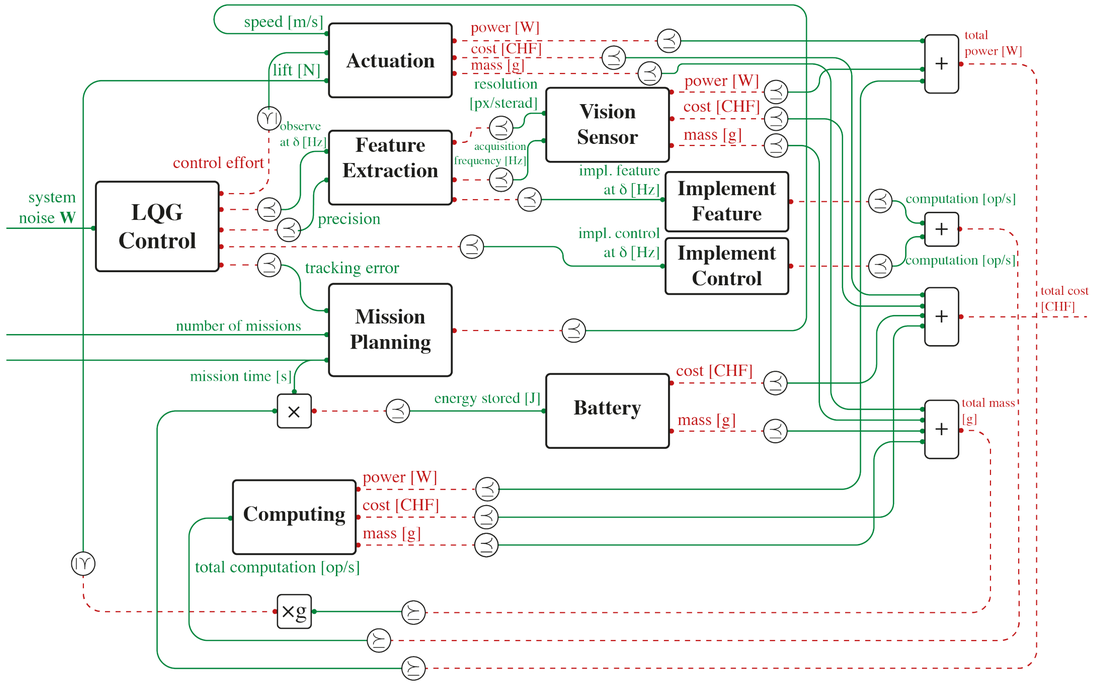
Large systems
- Many components
- Heterogeneous natures
- Multiple objectives

Strategic interactions
- Many agents
- Heterogeneous interactions
- Conflicts/collaborations

A new approach to multi-disciplinary engineering “co”-design

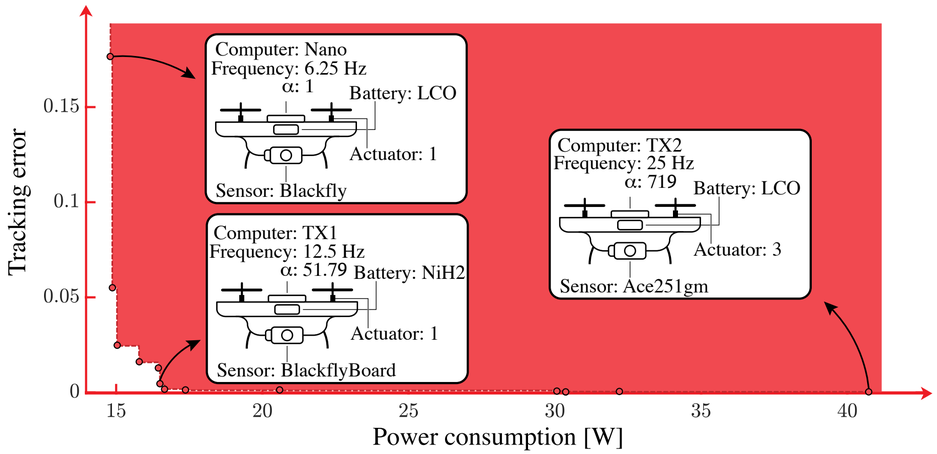
- A new approach to **collaborative**, **computational**, **compositional**, **continuous** design designed to work **across fields** and **across scales**.
- Leverages **domain theory**, **applied category theory**, and **optimization**
- Roadmap:
  - Defining **“design problems”** for **components.**
  - Modeling **co-design constraints** in a complex **system**.
  - **Efficient** solution to design queries.

“Co-design diagram”



optimization for a task

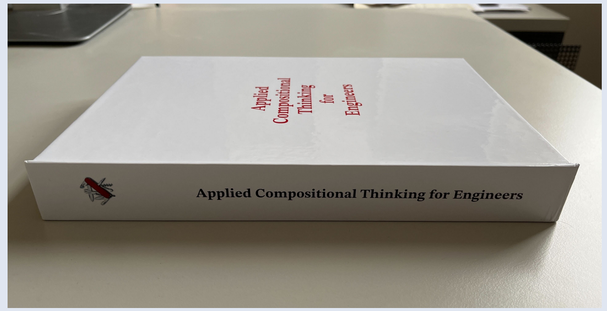
Pareto front of optimal designs



A new approach to multi-disciplinary engineering “co”-design

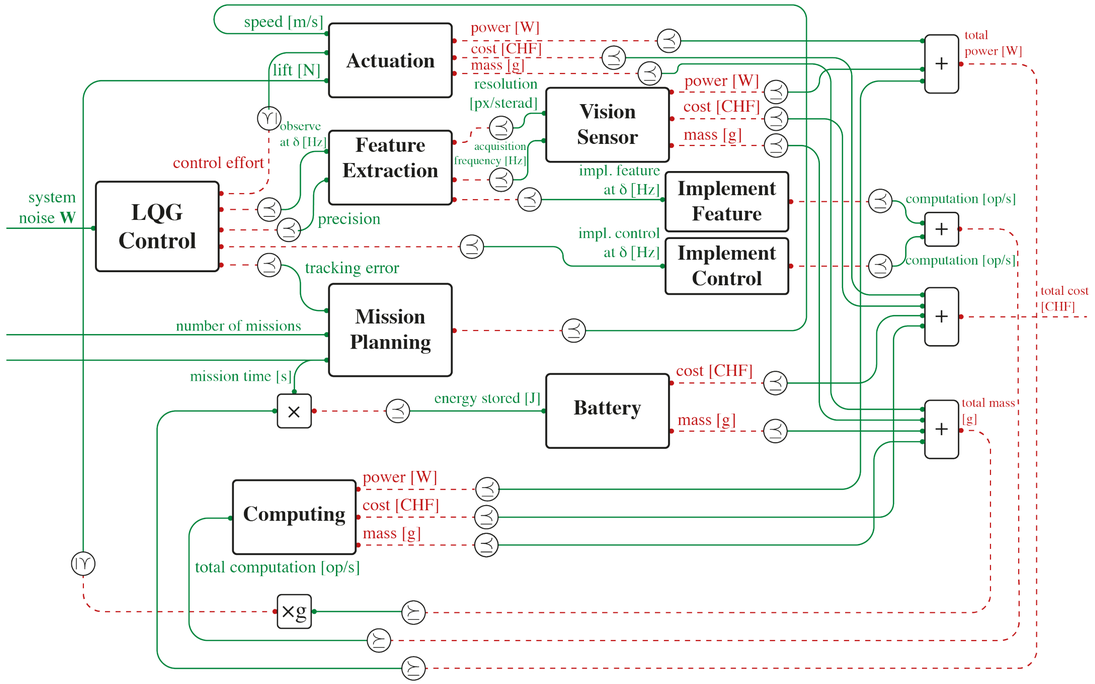
- A new approach to **collaborative**, **computational**, **compositional**, **continuous** design designed to work **across fields** and **across scales**.
- Leverages **domain theory**, **applied category theory**, and **optimization**
- Roadmap:
  - Defining **“design problems”** for **components.**
  - Modeling **co-design constraints** in a complex **system**.
  - **Efficient** solution to design queries.

a “pro” box



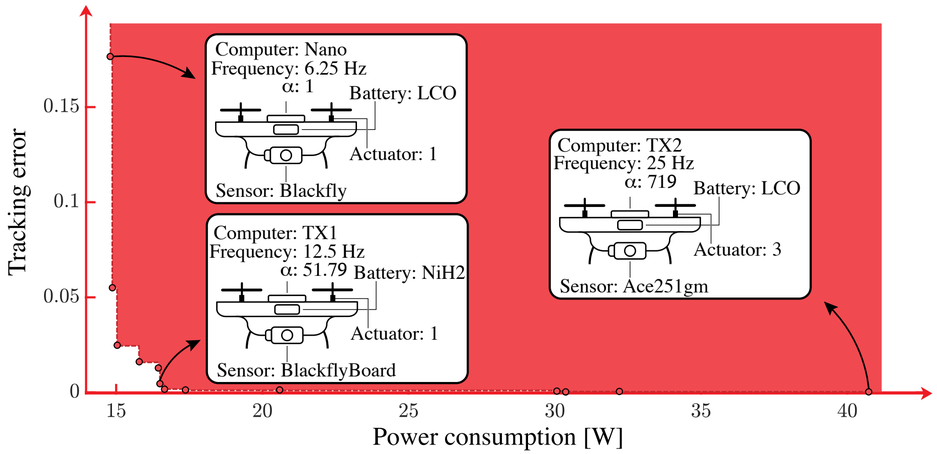
To be published by Cambridge University Press
https://bit.ly/3qQNrdR

“Co-design diagram”



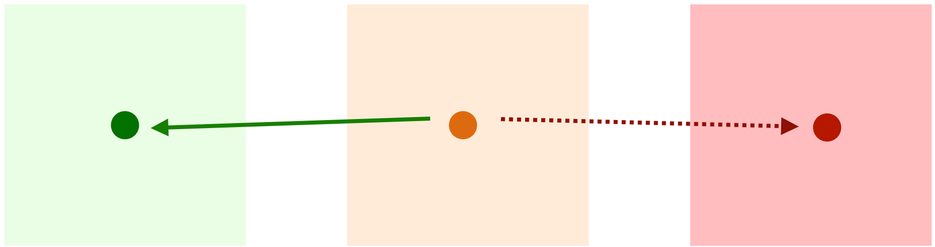
optimization for a task

Pareto front of optimal designs



An abstract view of design problems

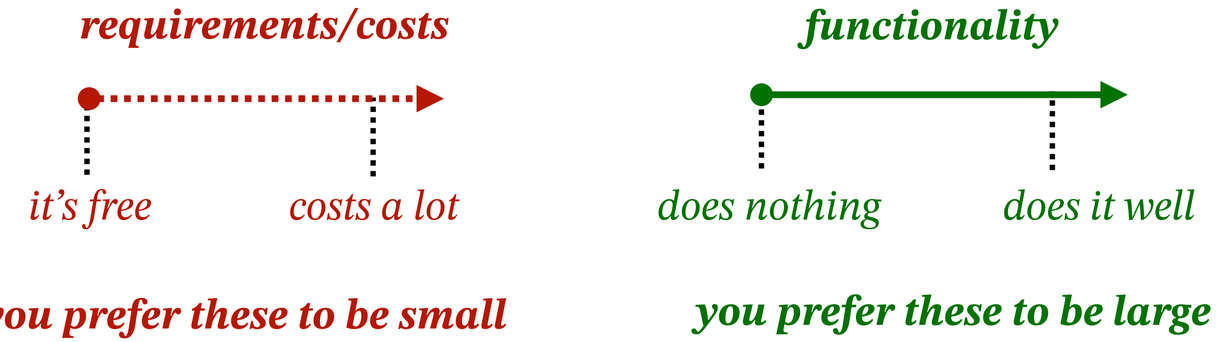
- Across fields, design or synthesis problems are defined with **three spaces**:
  - **implementation space**: the **options** we can choose from;
  - **functionality space**: what we need to **provide**/**achieve**;
  - **requirements/costs space**: the **resources** we need to have available;



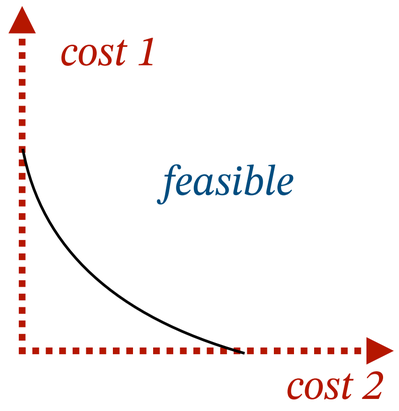
functionality *(provided)*	***implementations***	***costs, resources*** *(required)*
desired behavior	*choices*	*budget*
specifications	*plans*	*requirements*
objectives	*blueprints*	*dependencies*
guarantees	*decision variables*	*assumptions*
conclusions		
"function"	*"form"*	*"function"*

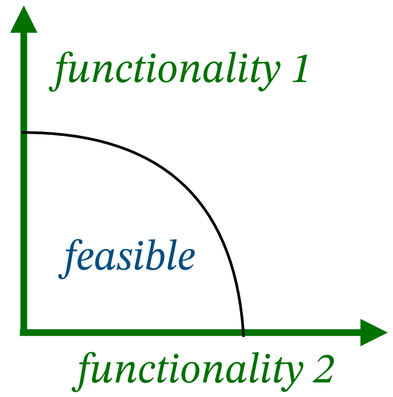
Trade-offs are everywhere: embrace them!

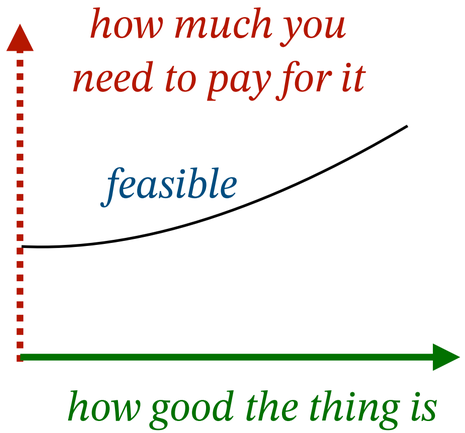
- We distinguish (semantically) between **functionality** and **requirements/costs.**



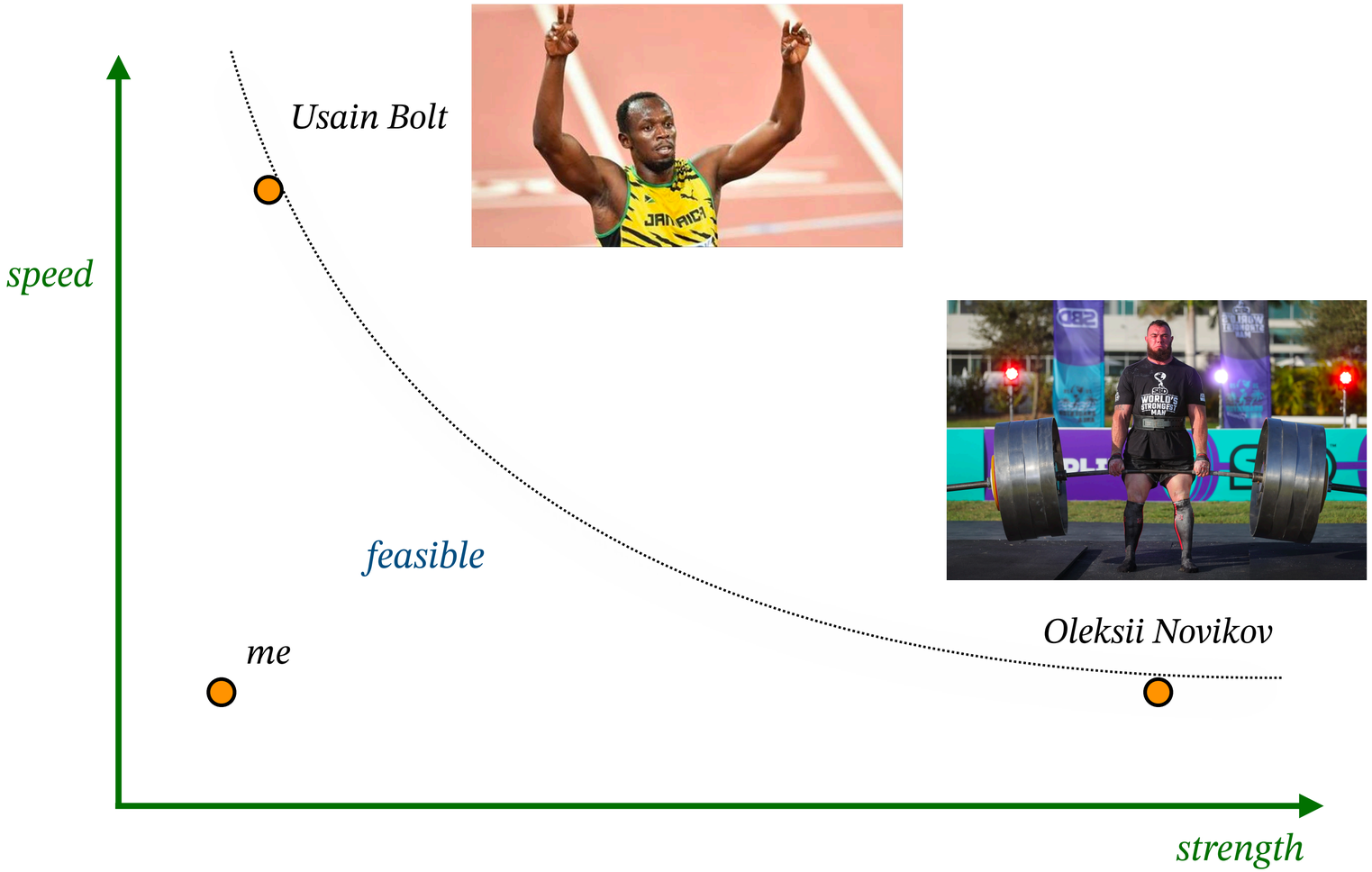
- Open an engineering book. Find the graphs talking about "achievable performance". What colors should the axes be? Classify into one of these:







Trade-offs for the human body



Multiple functionalities and costs

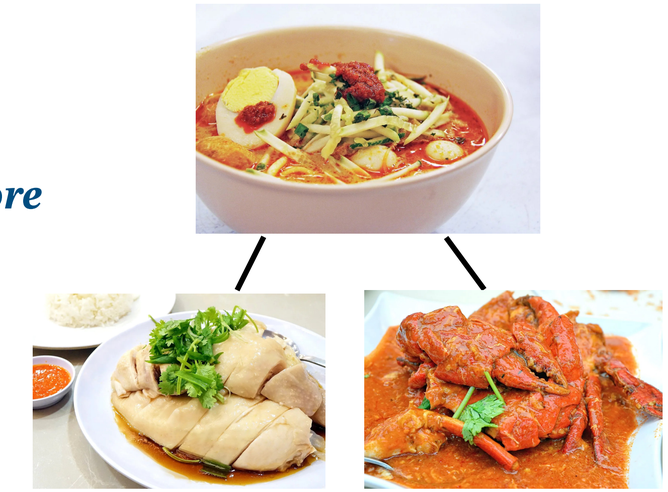
Multiple functionalities and costs

Multiple functionalities and costs

Partially ordered sets model trade-offs, across fields

- Posets model standard costs in engineering $\langle \mathbb{R}_{\geq 0}, \leq \rangle$, $\langle \mathbb{N}, \leq \rangle$
- … but also enable **richer** cost structures, with **incomparable** elements

A poset of food preferences in Singapore

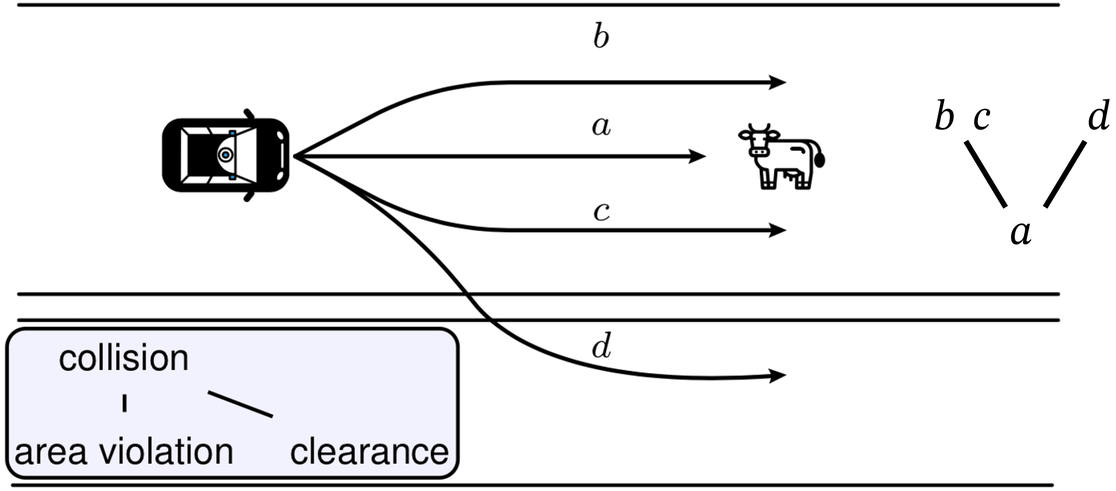


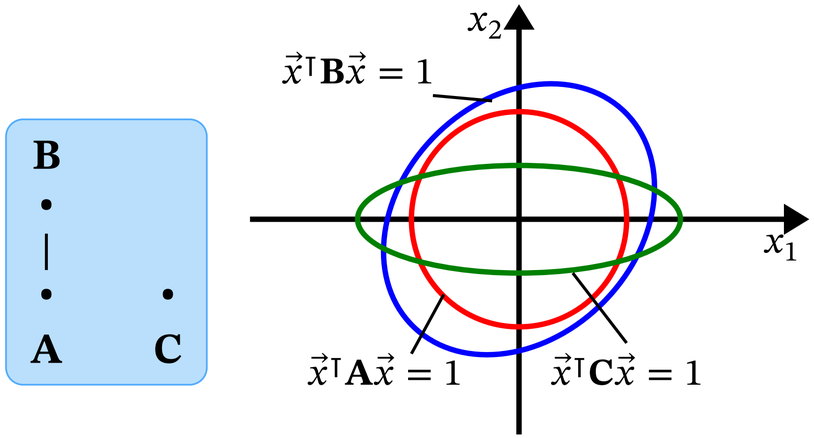
A poset of positive-definite matrices

$$\frac{\mathbf{A} \preceq_{\mathrm{PDM}(n)} \mathbf{B}}{\vec{x}^\intercal \mathbf{A} \vec{x} \leq \vec{x}^\intercal \mathbf{B} \vec{x} \quad \forall \vec{x} \in \mathbb{R}^n}$$

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 3/4 & -1/8 \\ -1/8 & 3/4 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1/2 & 0 \\ 0 & 2 \end{bmatrix}$$

Posets of rules, which induce priorities over behaviors

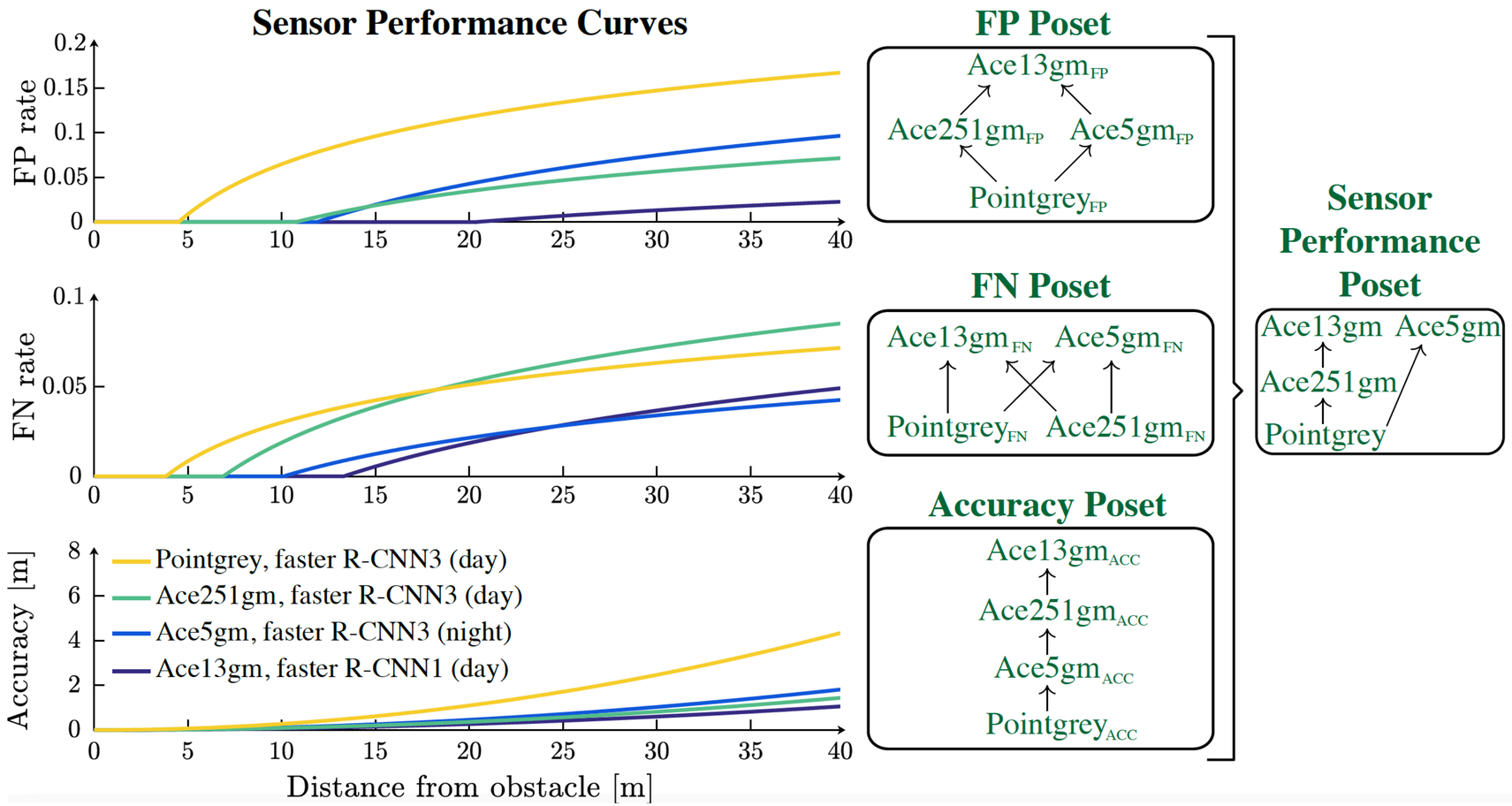




[RAL'22, CDC'22, IROS'21, ECC'21]

Partially ordered sets model trade-offs, across fields

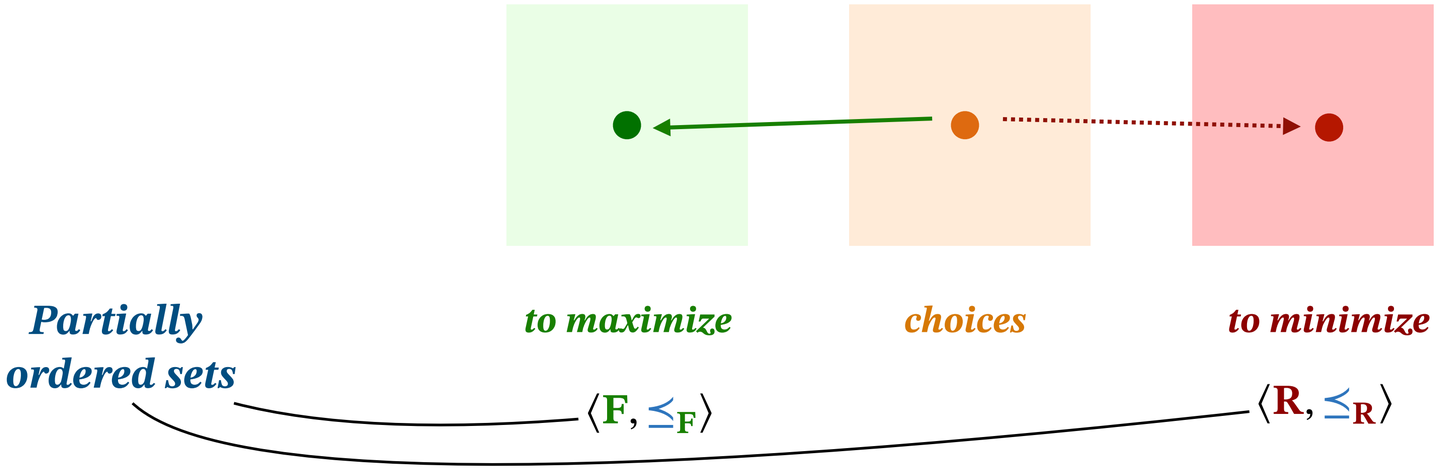
A poset of sensor/algorithm pairs



[TRO'25, CDC'22, IROS'21]

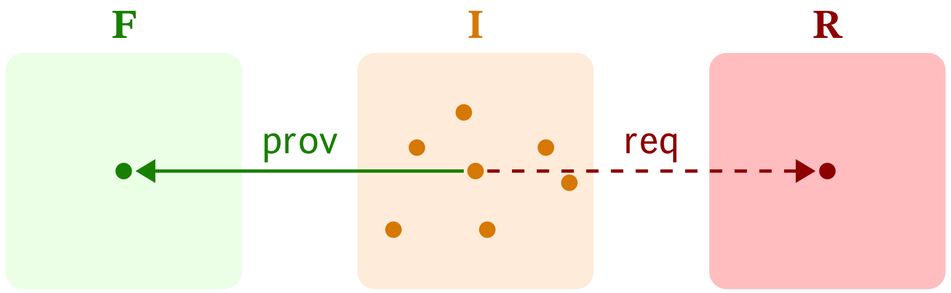
An abstract view of design problems

- Across fields, design or synthesis problems are defined with **three spaces**:
  - **implementation space:** the **options** we can choose from;
  - **functionality space**: what we need to **provide**/**achieve**;
  - **requirements/costs space**: the **resources** we need to have available;

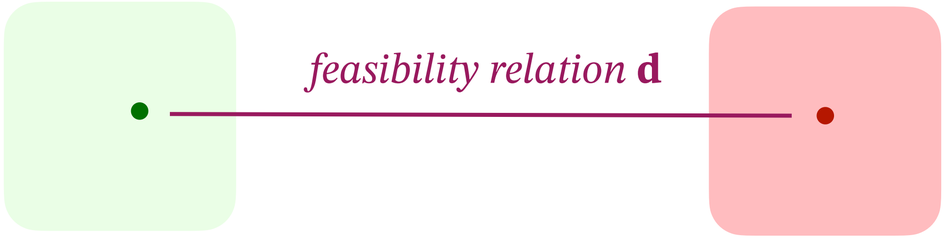


Transparent vs black-box models

- The "**Design Problems with Implementations" model** is a "**transparent" model:**



- **DP** model: **direct feasibility relation** between functionality and resources ("black box") as a monotone map:



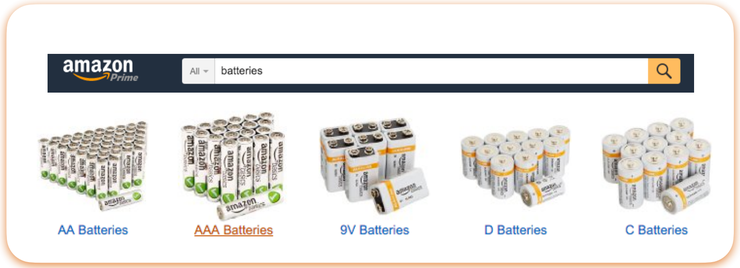
$$\mathbf{d} : \mathbf{F}^{\mathrm{op}} \times \mathbf{R} \to_{\mathbf{Pos}} \mathbf{Bool}$$
$$\langle f^*, r \rangle \mapsto \exists i \in \mathbf{I} : (f \leq_{\mathbf{F}} \mathsf{prov}(i)) \wedge (\mathsf{req}(i) \leq_{\mathbf{R}} r)$$

... a "boolean profunctor"

- **Monotonicity**:
  - **Lower functionality** does **not** require **more resources**;
  - More **resources** do not provide **less functionality.**

Co-design enables a rich class of model population techniques

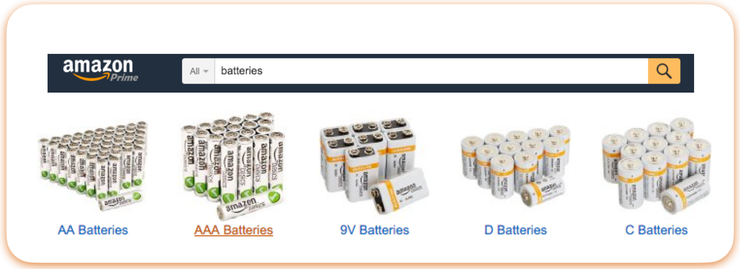
- “**Catalogues**”: off-the-shelf designs.



	Spark	Phantom 3 Std	Phantom 4 Adv	Phantom 4 Pro	Mavic	Inspire
						
Flight time	16 mins	25 mins	30 mins	30 mins	27 mins	27 mins
Top Speed	31 mph (50 km/h)	36 mph (58 km/h)	45 mph (72 km/h)	45 mph (72 km/h)	40 mph (65 km/h)	58 mph (94 km/h)
Range	1.2 miles (2 km)	0.6 miles (1 km)	4.3 miles (7 km)	4.3 miles (7 km)	4.3 miles (7 km)	4.3 miles (7 km)
Camera	12-MP stills 1080p video	12-MP stills 2704 x 1520p video	20-MP stills 4K 60fps video	20-MP stills 4K 60fps video	12-MP stills 4K video	20.8-MP stills 4K/5K video
Size	5.6 x 5.6 x 2.1 in (14.3 x 14.3 x 5.5 cm)	13.8 in diagonal (350 mm)	13.8 in diagonal (350 mm)	13.8 in diagonal (350 mm)	13.2 in diagonal (350 mm)	16.8 x 12.5 x 16.7 in (42.7 x 31.7 x 42.5 cm)
Takeoff weight	11.6 oz (330 g)	2.6 lb (1.2 kg)	3 lb (1.4 kg)	3 lb (1.4 kg)	1.6 lb (743 kg)	8.8 lb (4 kg)
Other features	Follow me, Return home, Obstacle avoidance, FPV	Follow me, Return home	Follow me, Return home, Obstacle avoidance	Follow me, Return home, 3 Direction Obstacle avoidance	Follow me, Return home, Obstacle avoid-ance, folding arms	Obstacle avoidance, Spotlight Pro/Broadcast/ Composition mode
Price	US$499	US$499	US$1,349	US$1,499	US$999	US$2,999 ($6,198 with camera/gimbal)

Co-design enables a rich class of model population techniques

- “**Catalogues**”: off-the-shelf designs.



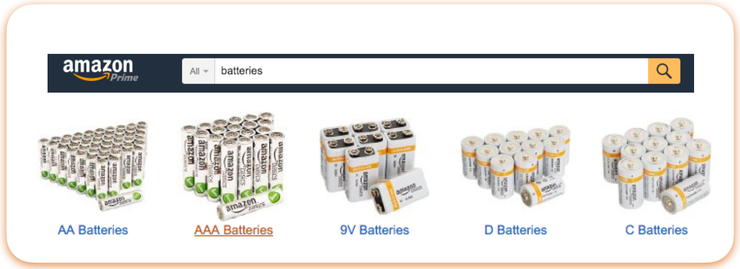
	Spark	Phantom 3 Std	Phantom 4 Adv	Phantom 4 Pro	Mavic	Inspire
						
Flight time	16 mins	25 mins	30 mins	30 mins	27 mins	27 mins
Top Speed	31 mph (50 km/h)	36 mph (58 km/h)	45 mph (72 km/h)	45 mph (72 km/h)	40 mph (65 km/h)	58 mph (94 km/h)
Range	1.2 miles (2 km)	0.6 miles (1 km)	4.3 miles (7 km)	4.3 miles (7 km)	4.3 miles (7 km)	4.3 miles (7 km)
Camera	12-MP stills 1080p video	12-MP stills 2704 x 1520p video	20-MP stills 4K 60fps video	20-MP stills 4K 60fps video	12-MP stills 4K video	20.8-MP stills 4K/5K video
Size	5.6 x 5.6 x 2.1 in (14.3 x 14.3 x 5.5 cm)	13.8 in diagonal (350 mm)	13.8 in diagonal (350 mm)	13.8 in diagonal (350 mm)	13.2 in diagonal (350 mm)	16.8 x 12.5 x 16.7 in (42.7 x 31.7 x 42.5 cm)
Takeoff weight	11.6 oz (330 g)	2.6 lb (1.2 kg)	3 lb (1.4 kg)	3 lb (1.4 kg)	1.6 lb (743 kg)	8.8 lb (4 kg)
Other features	Follow me, Return home, Obstacle avoidance, FPV	Follow me, Return home	Follow me, Return home, Obstacle avoidance	Follow me, Return home, 3 Direction Obstacle avoidance	Follow me, Return home, Obstacle avoid-ance, folding arms	Obstacle avoidance, Spotlight Pro/Broadcast/ Composition mode
Price	US$499	US$499	US$1,349	US$1,499	US$999	US$2,999 ($6,198 with camera/gimbal)

- “**First-principles**”: analytical relations.

mission energy ≥ *mission duration* x *power consumption*

Co-design enables a rich class of model population techniques

- “**Catalogues**”: off-the-shelf designs.



	Spark	Phantom 3 Std	Phantom 4 Adv	Phantom 4 Pro	Mavic	Inspire
Flight time	16 mins	25 mins	30 mins	30 mins	27 mins	27 mins
Top Speed	31 mph (50 km/h)	36 mph (58 km/h)	45 mph (72 km/h)	45 mph (72 km/h)	40 mph (65 km/h)	58 mph (94 km/h)
Range	1.2 miles (2 km)	0.6 miles (1 km)	4.3 miles (7 km)	4.3 miles (7 km)	4.3 miles (7 km)	4.3 miles (7 km)
Camera	12-MP stills 1080p video	12-MP stills 2704 x 1520p video	20-MP stills 4K 60fps video	20-MP stills 4K 60fps video	12-MP stills 4K video	20.8-MP stills 4K/5K video
Size	5.6 x 5.6 x 2.1 in (14.3 x 14.3 x 5.5 cm)	13.8 in diagonal (350 mm)	13.8 in diagonal (350 mm)	13.8 in diagonal (350 mm)	13.2 in diagonal (350 mm)	16.8 x 12.5 x 16.7 in (42.7 x 31.7 x 42.5 cm)
Takeoff weight	11.6 oz (330 g)	2.6 lb (1.2 kg)	3 lb (1.4 kg)	3 lb (1.4 kg)	1.6 lb (743 kg)	8.8 lb (4 kg)
Other features	Follow me, Return home, Obstacle avoidance, FPV	Follow me, Return home	Follow me, Return home, Obstacle avoidance	Follow me, Return home, 3 Direction Obstacle avoidance	Follow me, Return home, Obstacle avoid-ance, folding arms	Obstacle avoidance, Spotlight Pro/Broadcast/ Composition mode
Price	US$499	US$499	US$1,349	US$1,499	US$999	US$2,999 ($6,198 with camera/gimbal)

- “**First-principles**”: analytical relations.

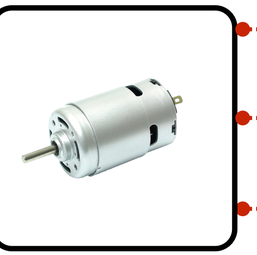
mission energy ≥ *mission duration* x *power consumption*

- “**Data-driven**”, “**on-demand**”
  - The optimization will only ask for a **sequence** of data points. The model is constructed **incrementally**.
  - Opens the door to **experiments**, black-box **simulations**, solutions of **optimization problems**.

Design problems arise naturally in many domains, across scales


Co-Design to Enable User-Friendly Tools to Assess the Impact of Future Mobility Solutions

Gioele Zardini[1], Nicolas Lanzetti[2], Andrea Censi[1], Emilio Frazzoli[1], and Marco Pavone[3]



cost [$]

speed [rad/s]

mass [kg]

torque [Nm]

current [A]

speed [m/s]

fix cost

task

operational cost

agents' speed [m/s]

area to cover [m2]

agents

time [s]

total cost [$]

demand

investments

externalities

obs. noise

LQG

tracking error

syst. noise

effort

guarantees

assumptions

accuracy

GMapping

nr. particles

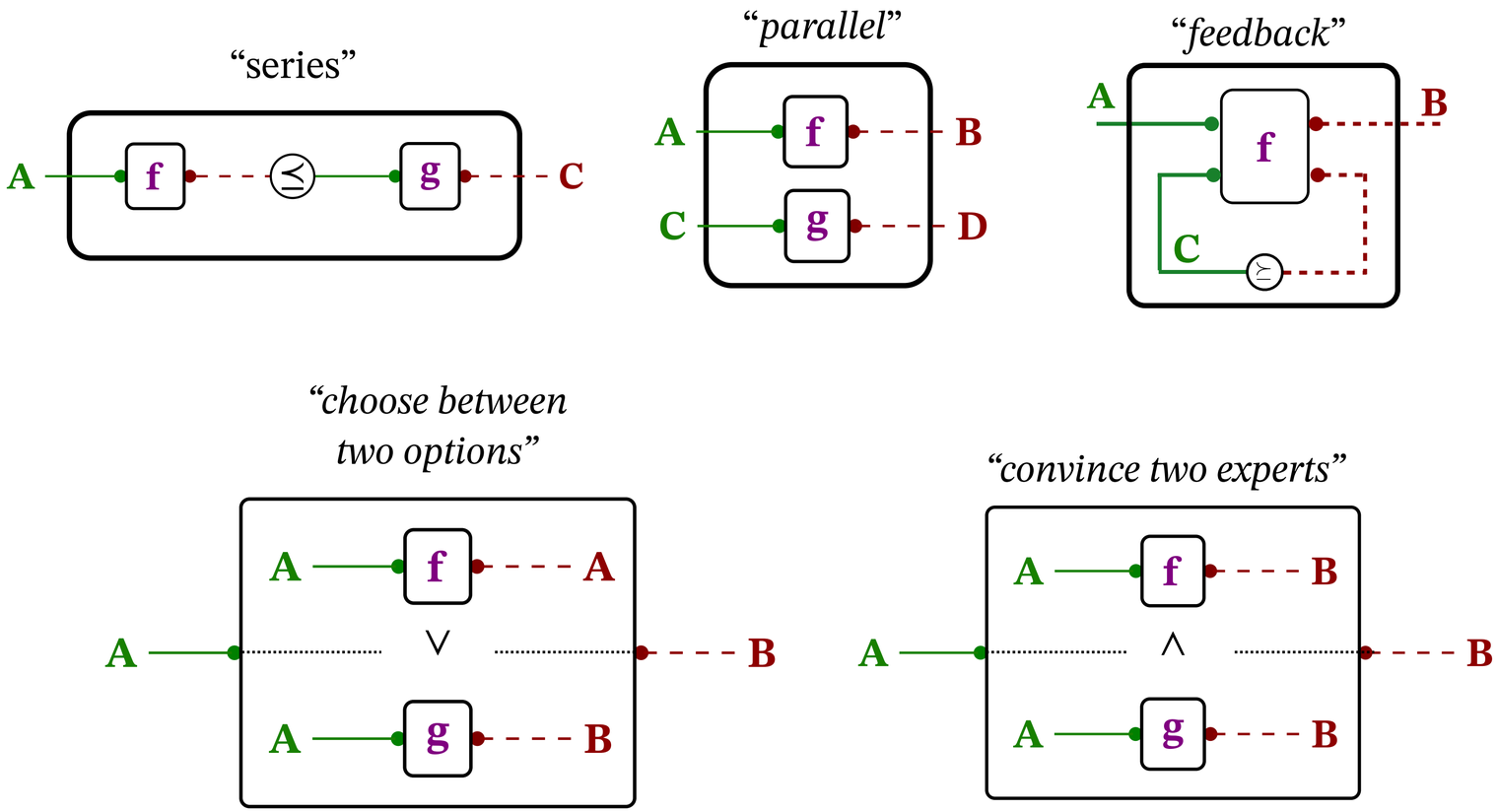
robustness

Bin packing

content shapes

container shape

Design problems can be composed in various ways, preserving properties



- The **composition** of any two **DPs** returns a **DP** (closure)
- Very practical tool to **decompose** large **problems** into **subproblems**

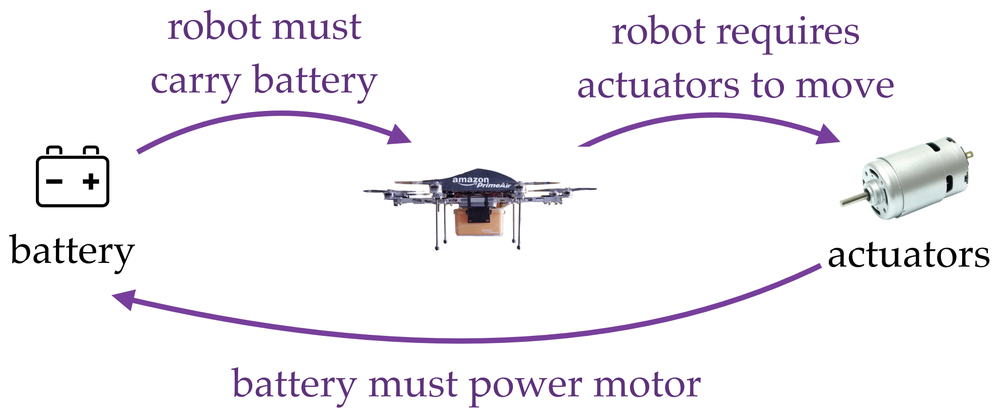
> **"A system is composed of components; a component is something you understand."**
>
> — Howard Aiken (1900-1973)

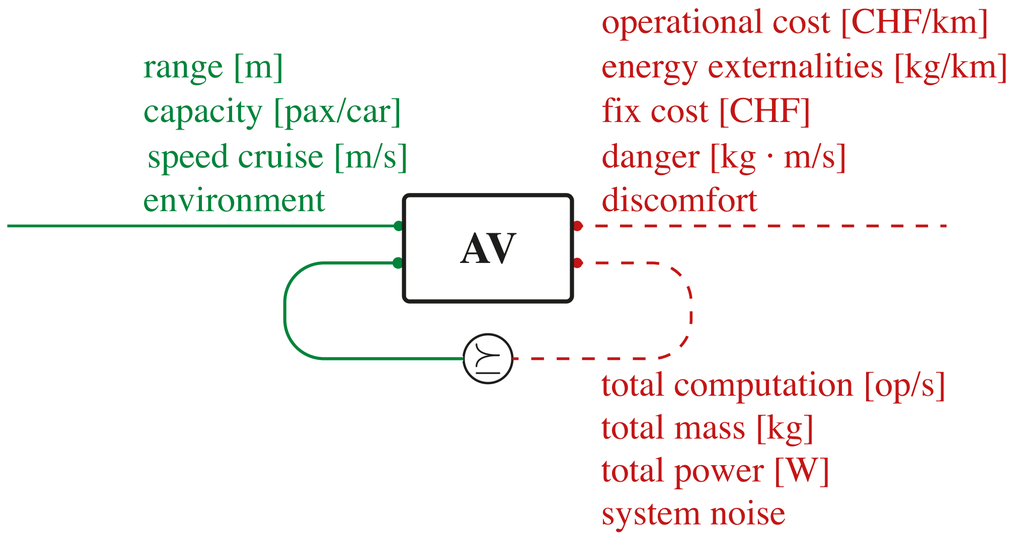
*There is a category **DP** which is traced monoidal, and locally posetal*

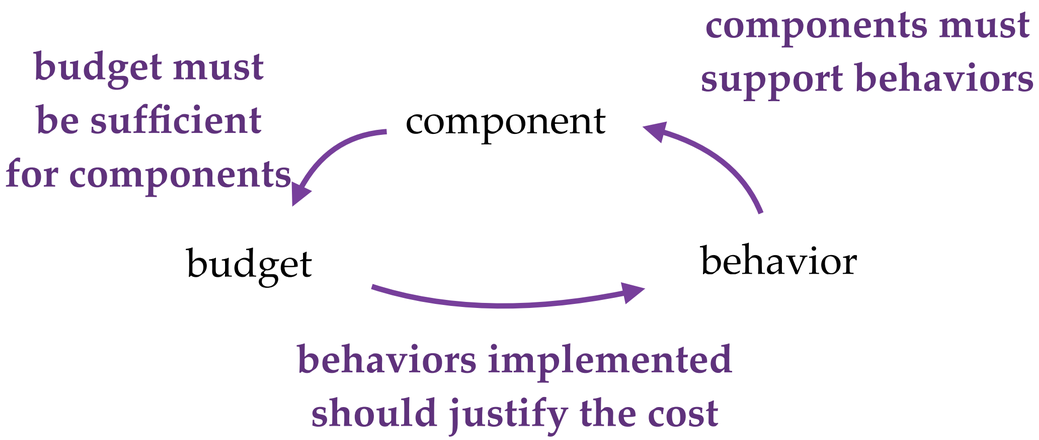
✓ ***Formal***
Compositional/hierarchical

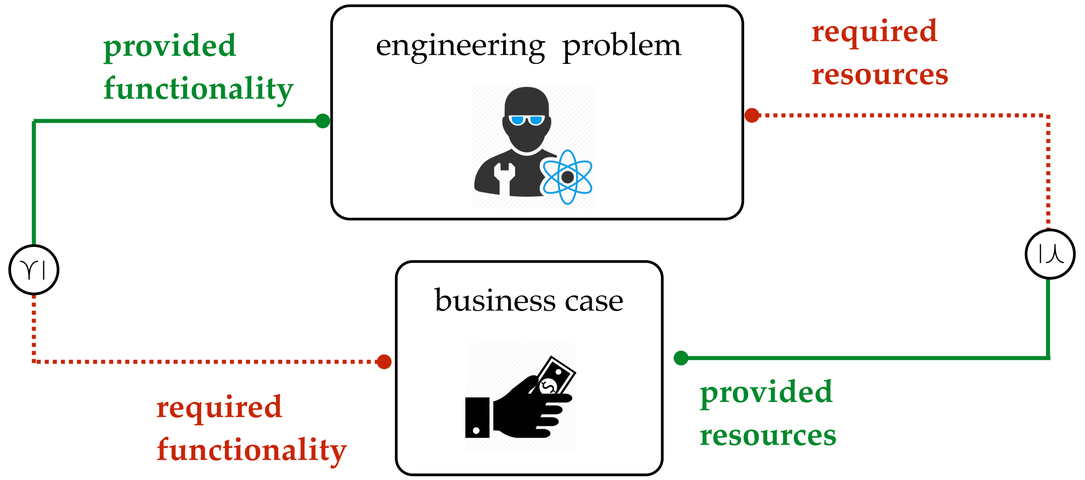
Feedback as the irreducible complexity of system design

- Where is **feedback**? In the **co-design constraints**









What Applied Category Theory (ACT) is for us

- **You have developed very good mental models** of systems and components in your domain.
  - Some models are formal, some models are “informal”.
  - Some models are computational, some are not.

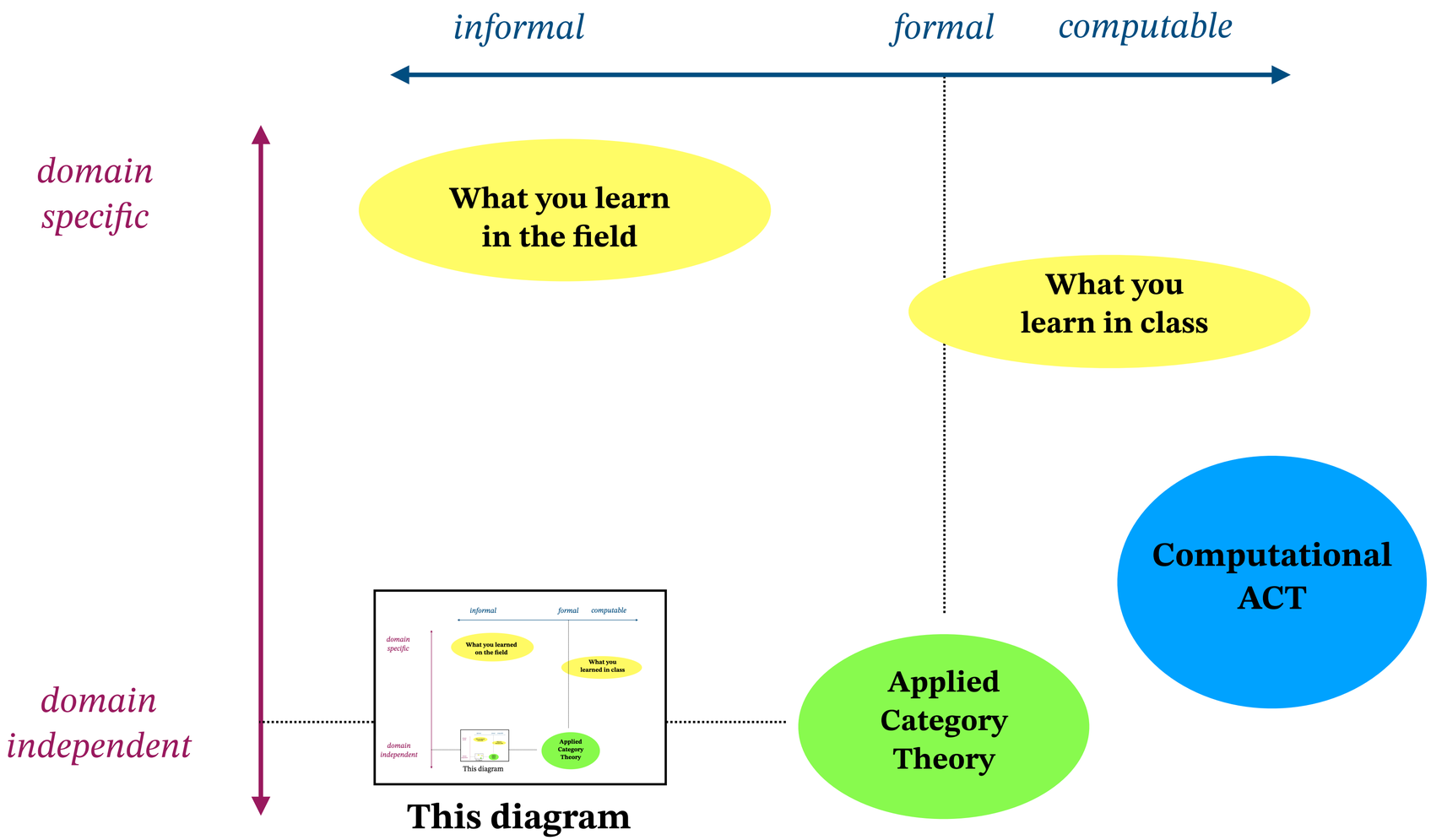
- **You use the more formal models to design, simulate, and analyze your systems**.
  - For example, PDEs, etc.
- The formal models also **help you communicate and share your ideas** (i.e., CDC)

- Think of the **mathematics** you already know as **a set of patterns** to describe the world.

- Now, unfortunately, you are an engineer **in the 21st century:**
  - The engineers before us solved a lot of problems!
  - The new problems to solve **have to deal with very complex systems, architectures, and interactions!**
- **ACT is an updated set of mathematical patterns that is particularly suited to describe large systems** with complex interaction/interconnection/decomposition features.

The case for Applied Category Theory

- Applied Category Theory is **formal** but **domain-independent** and **intellectually tractable***

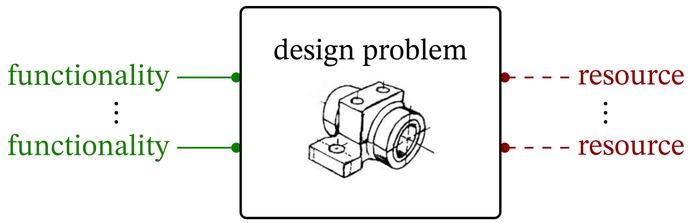


Multiple queries from the same design problem

- Two basic design queries are:

Given the functionality to be provided,
what are the **minimal resources** required?

FixFunMinReq



FixReqMaxFun

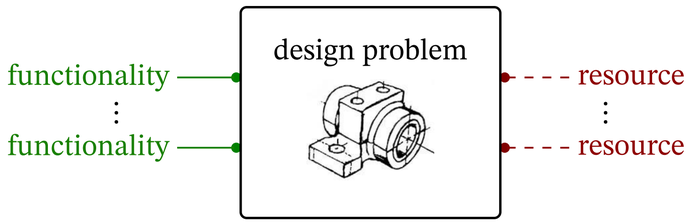
Given the resources that are available, what is
the **maximal functionality** that can be provided?

Multiple queries from the same design problem

- Two basic design queries are:

Given the functionality to be provided,
what are the **minimal resources** required?

FixFunMinReq



FixReqMaxFun

Given the resources that are available, what is
the **maximal functionality** that can be provided?

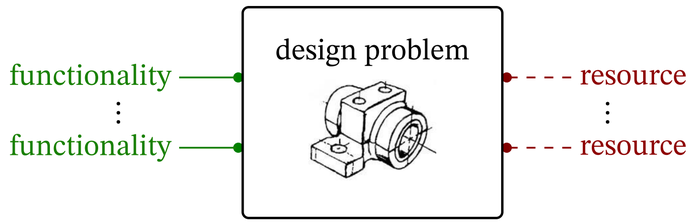
- The two problems are **dual**

Multiple queries from the same design problem

- Two basic design queries are:

Given the functionality to be provided,
what are the **minimal resources** required?

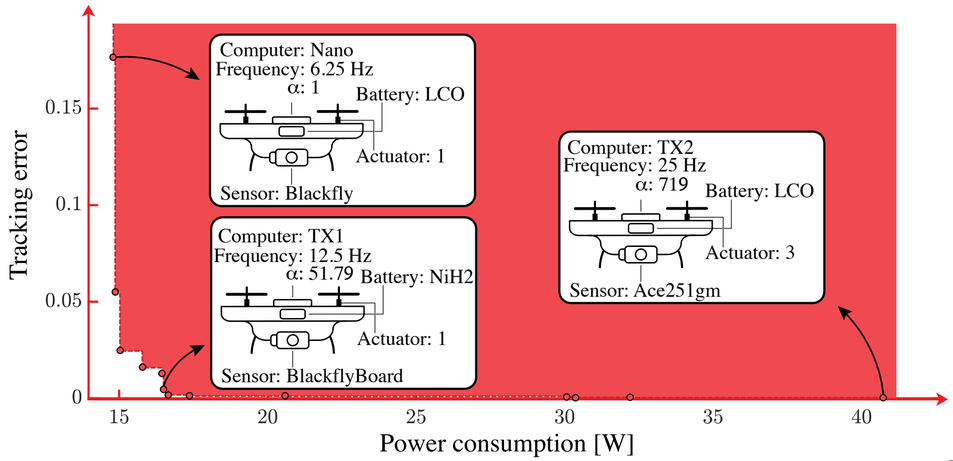
FixFunMinReq



FixReqMaxFun

Given the resources that are available, what is
the **maximal functionality** that can be provided?

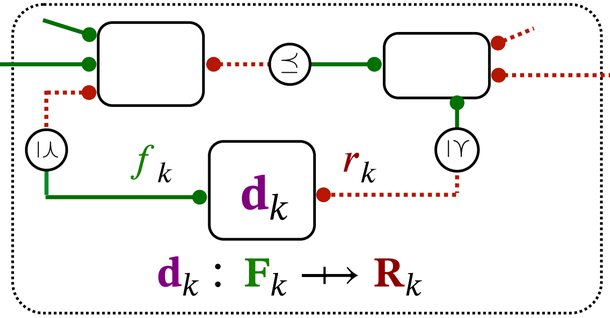
- We are looking for:
  - A map from functionality to **upper sets** of feasible resources: $h : \mathbf{F} \to \mathcal{U}\mathbf{R}$
  - A map from functionality to **antichains** of minimal resources: $h : \mathbf{F} \to \mathcal{A}\mathbf{R}$



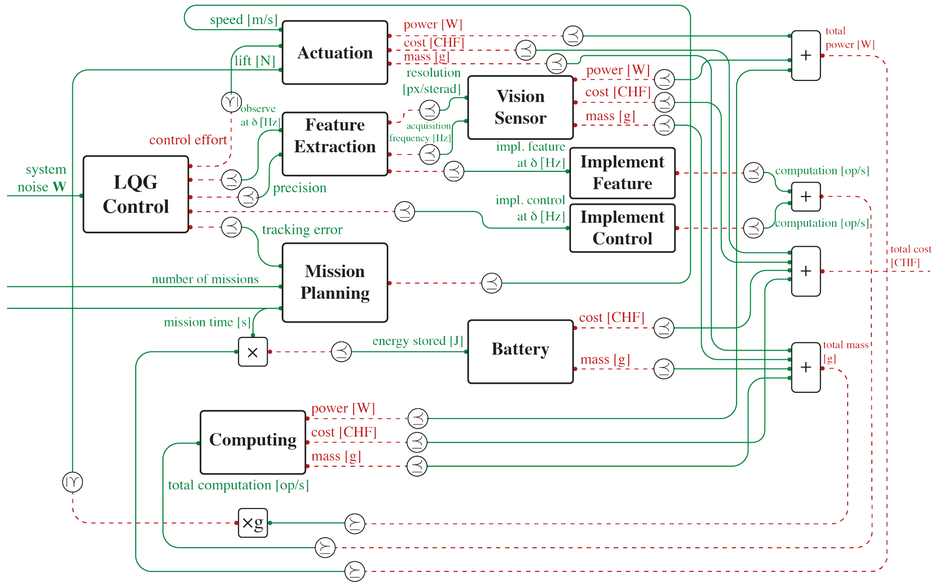
A new family of compositional optimization problems

- This is the semantics of **FixFunMinReq** as a **family of optimization problems.**

chosen by user $\overline{f}$

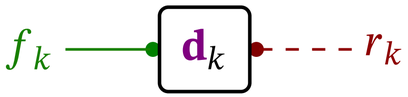


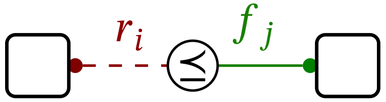
$\overline{r}$ **to minimize**



variables $f_k \in \langle \mathbf{F}_k, \preceq_{\mathbf{F}_k} \rangle$ $r_k \in \langle \mathbf{R}_k, \preceq_{\mathbf{R}_k} \rangle$

constraints *for **each node**:* *for **each edge:***





$\mathbf{d}_k(f_k^*, r_k) = \top$ $r_i \preceq f_j$

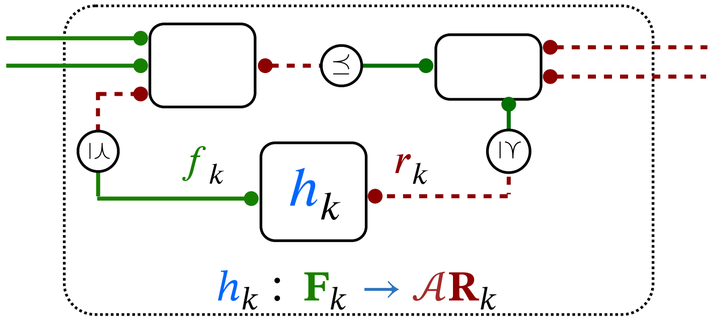
component feasibility *co-design constraint*

! **not convex**
! **not differentiable**
! **not continuous**
! **not even defined on continuous spaces**

objective $\underset{\preceq}{\text{Min}}\, \bar{r}$

Compositional solution of design problem queries

- Suppose that we are given the map $h_k : \mathbf{F}_k \to \mathcal{A}\mathbf{R}_k$ for all nodes in the co-design graph



- Can we find the map $h : \mathbf{F} \to \mathcal{A}\mathbf{R}$ for the entire graph?

✔ ***Computationally tractable***

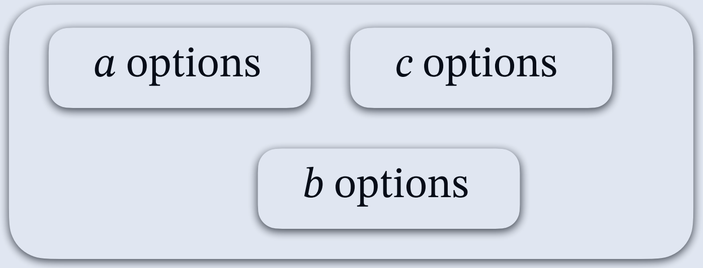
- **Compositional approach**: just need to work out the composition formulas for all operations

solution(**composition**(a, b)) = **composition**(**solution**(a), **solution**(b))

... a functor between a category of problems and one of solutions

- The set of **minimal** feasible resources can be obtained as the **least fixed point** of a monotone function in the space of anti-chain

- We have a **complete solution**: guaranteed to find the set of **all** optimal solutions (if empty, **certificate of infeasibility**)

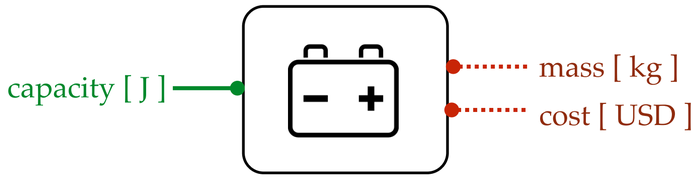


- The complexity is **not combinatorial in the number of options** for each component
- The complexity depends on the **complexity of the interactions**: the co-design **constraints**

$O(a + b + c)$

User-friendly interfaces

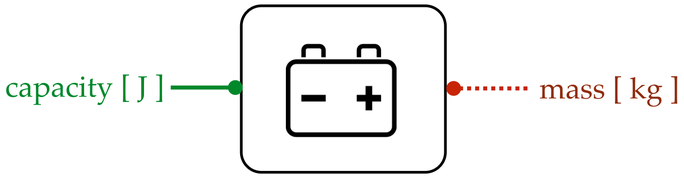
- "**Catalogues**": already available designs




```
catalogue {
    provides capacity [J]
    requires mass [g]
    requires cost [USD]

    500 kWh ⟷ model1 ⟼ 100 g, 10 USD
    600 kWh ⟷ model2 ⟼ 200 g, 200 USD
    600 kWh ⟷ model3 ⟼ 250 g, 150 USD
    700 kWh ⟷ model4 ⟼ 400 g, 400 USD
}
```

- "**First-principles**": analytical relations.




```
mcdp {
    provides capacity [J]
    requires mass [kg]

    specific_energy_Li_Ion = 500 Wh / kg

    required mass >= provided capacity / specific_energy_Li_Ion
}
```


```
mcdp {
    provides lift [N]
    requires power [W]
    c = 10.0 W/N²
    required power ≽ c · provided lift²
}
```

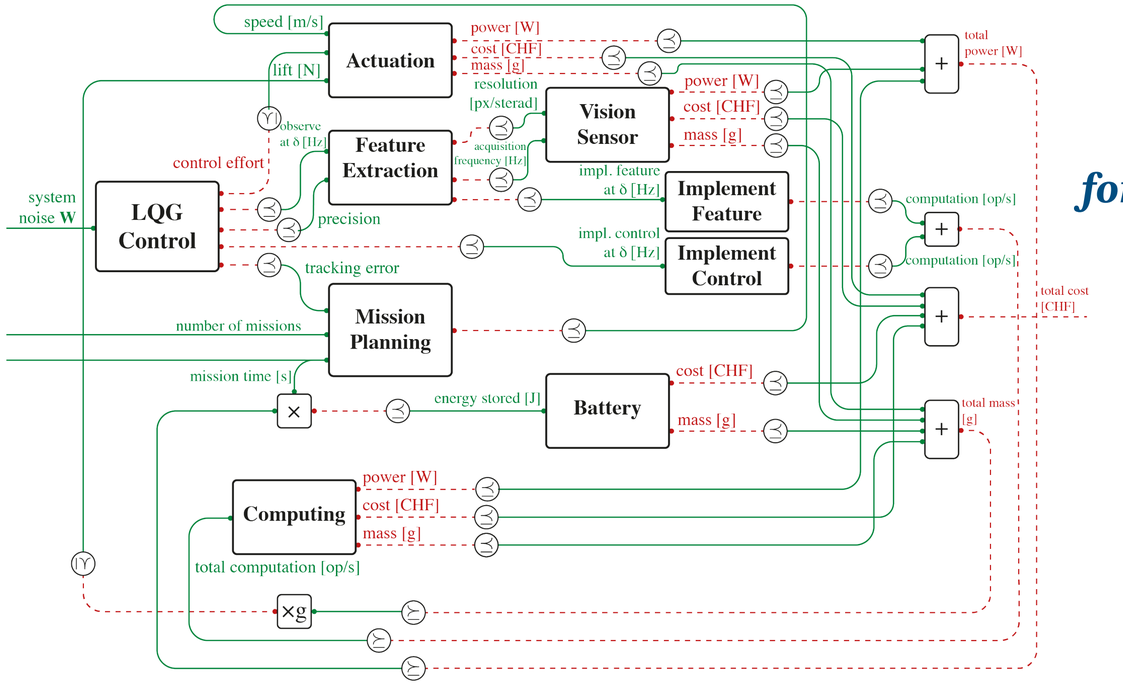
... and a solver

A systematic process for task-driven co-design of complex systems

- A new approach to **co-design** designed to work **across fields** and **across scales**.
- What we have seen:
  - Defining **"design problems"** for **components.**
  - Modeling **co-design constraints** in a complex **system**.
  - **Efficient** solution to design queries.

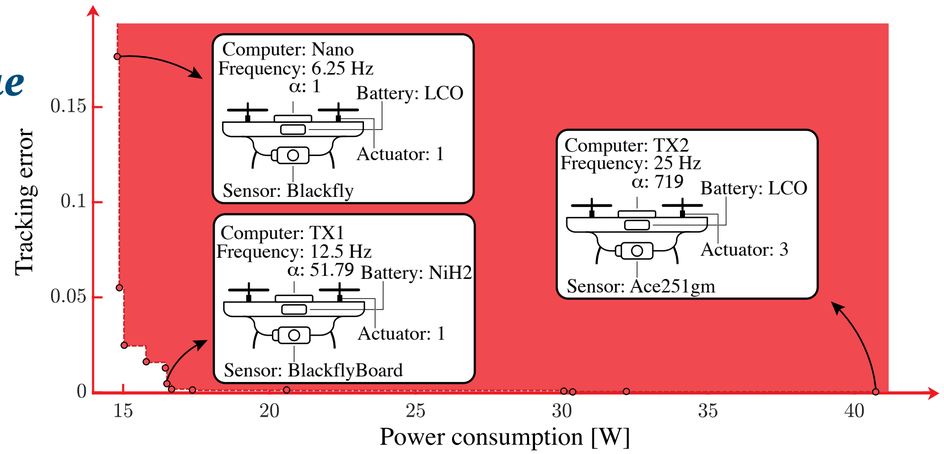
- **Modeling** approach
- Actual **implementation**:
  - Coming up with the skeleton/diagram
  - Populating the models

"Co-design diagram for a drone"



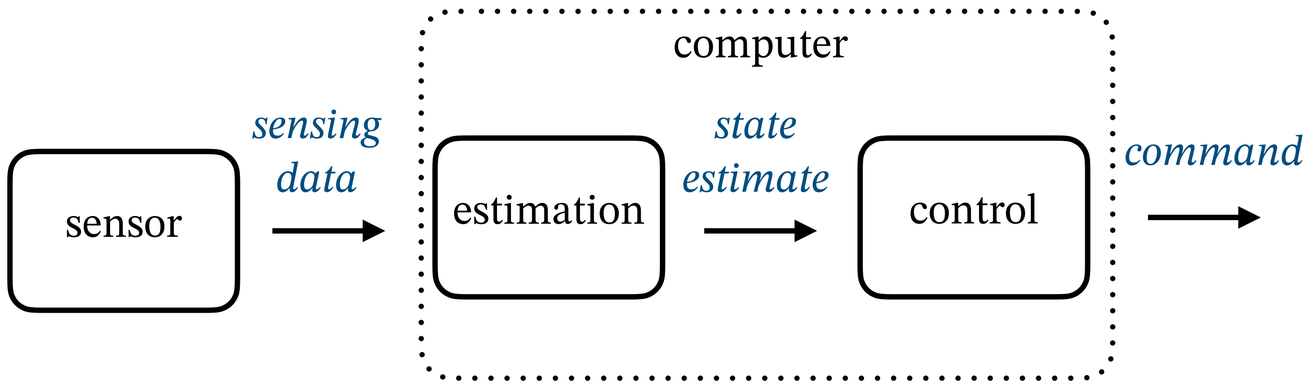
optimization for a search-and-rescue task

Pareto front of optimal designs

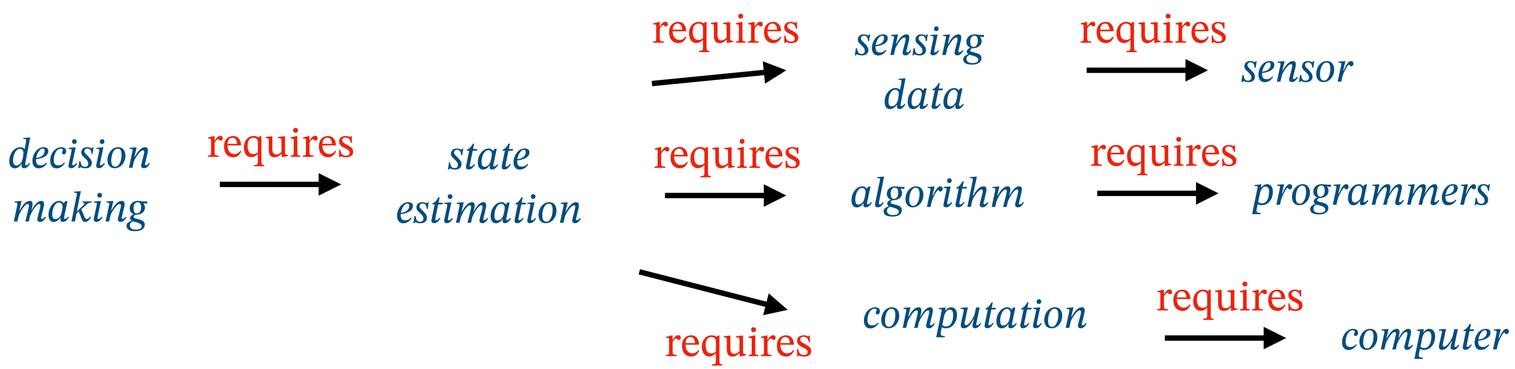


A systematic process for task-driven co-design of complex systems

- A **systematic modeling approach**:
  - **Define the task** - *what do we need to do?*
  - **Functional decomposition** - *how to decompose the functionality?*
  - **Find components** - *decompose until you find components* (hardware and software)



Data/Information flow

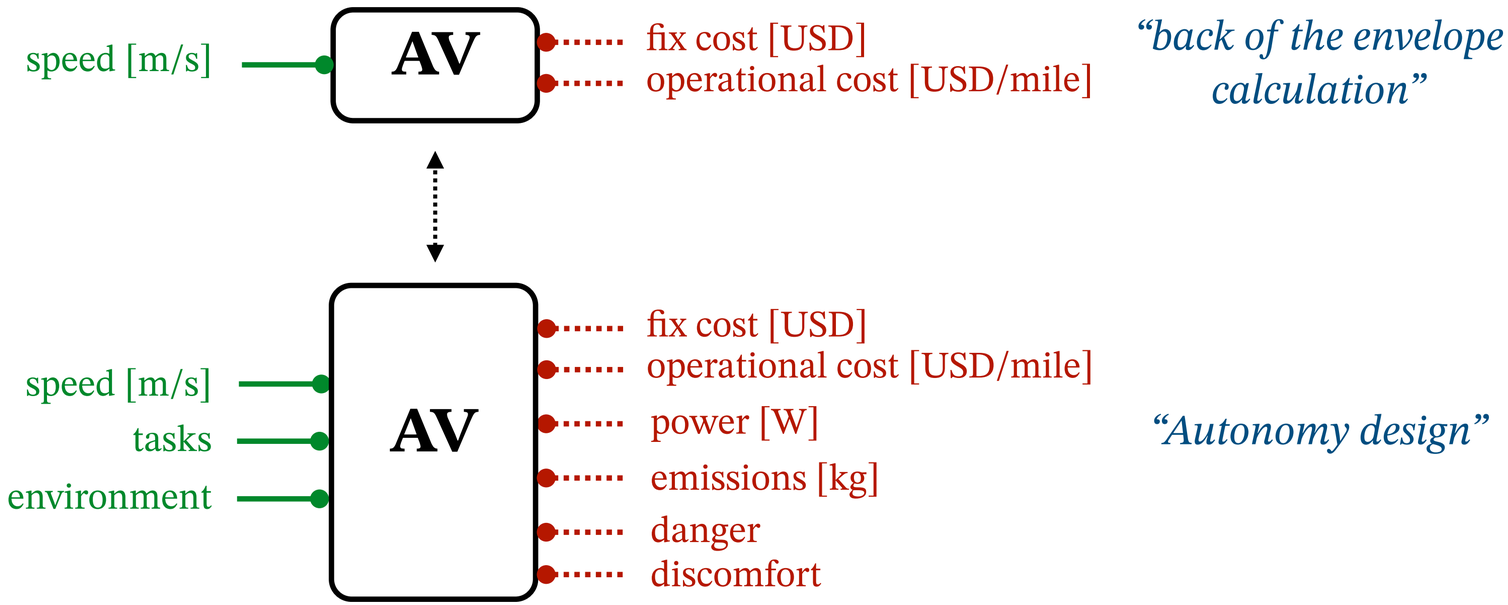


Logical dependencies

A systematic process for task-driven co-design of complex systems

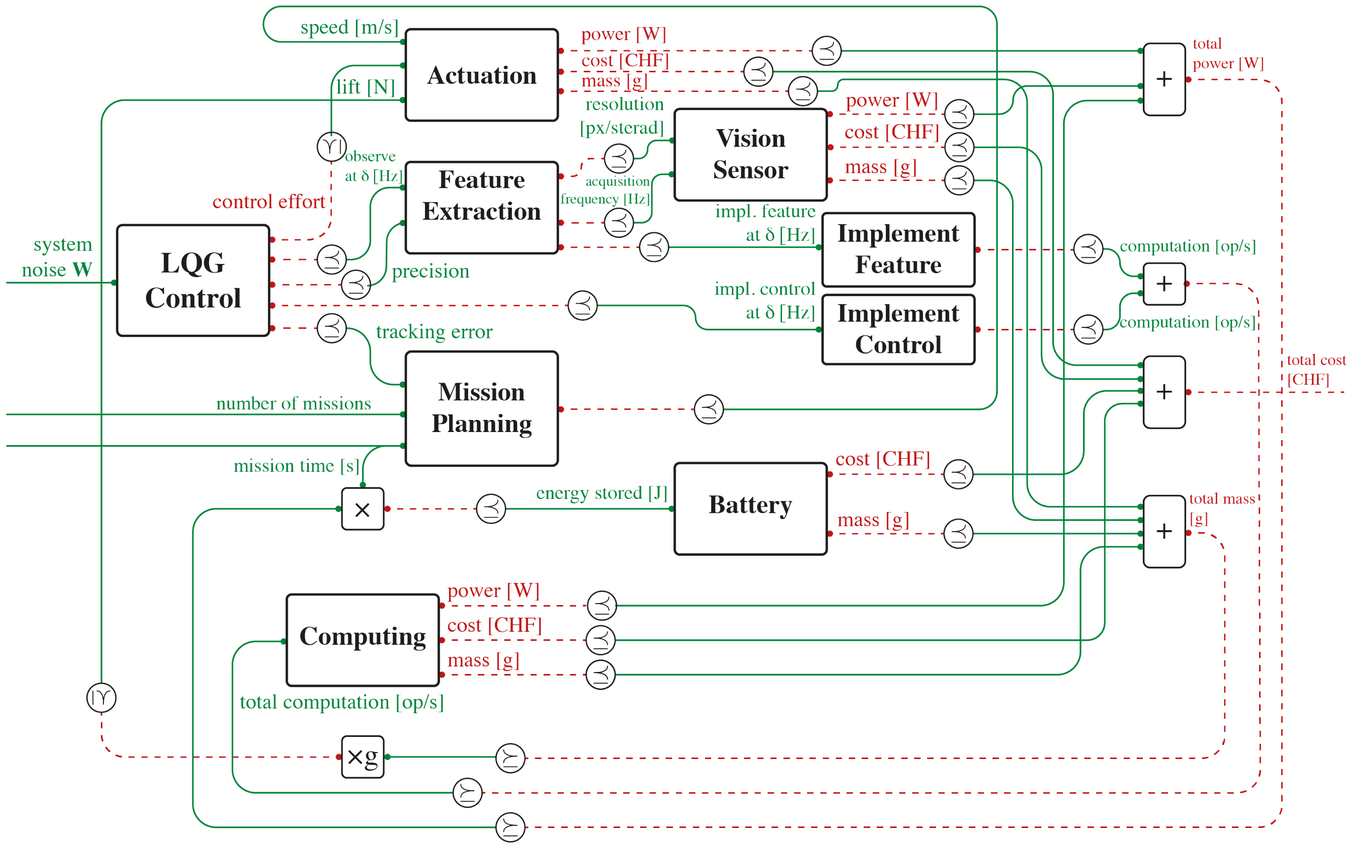
- Actual **implementation**:
  - **Write a skeleton** - *write the structure using the formal language and the **logical** dependencies.*

Context informs **level of detail**:



A systematic process for task-driven co-design of complex systems

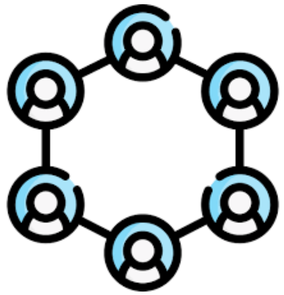
- Actual **implementation**:
  - **Populate the models:**
    catalogues, analytic models, data-driven



Continuous
Collaborative
Intellectually tractable

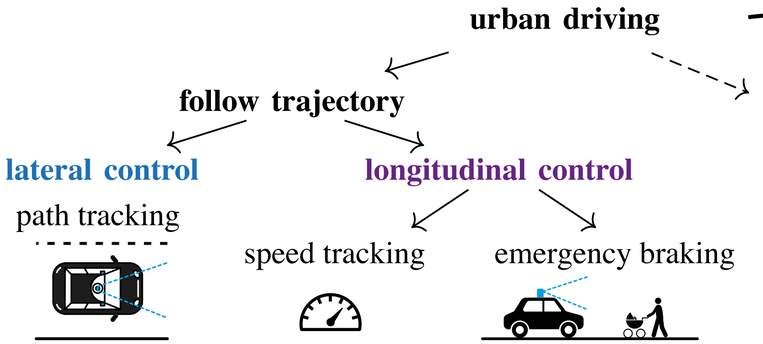
*If unsure about a component, easy to embed **assumptions***

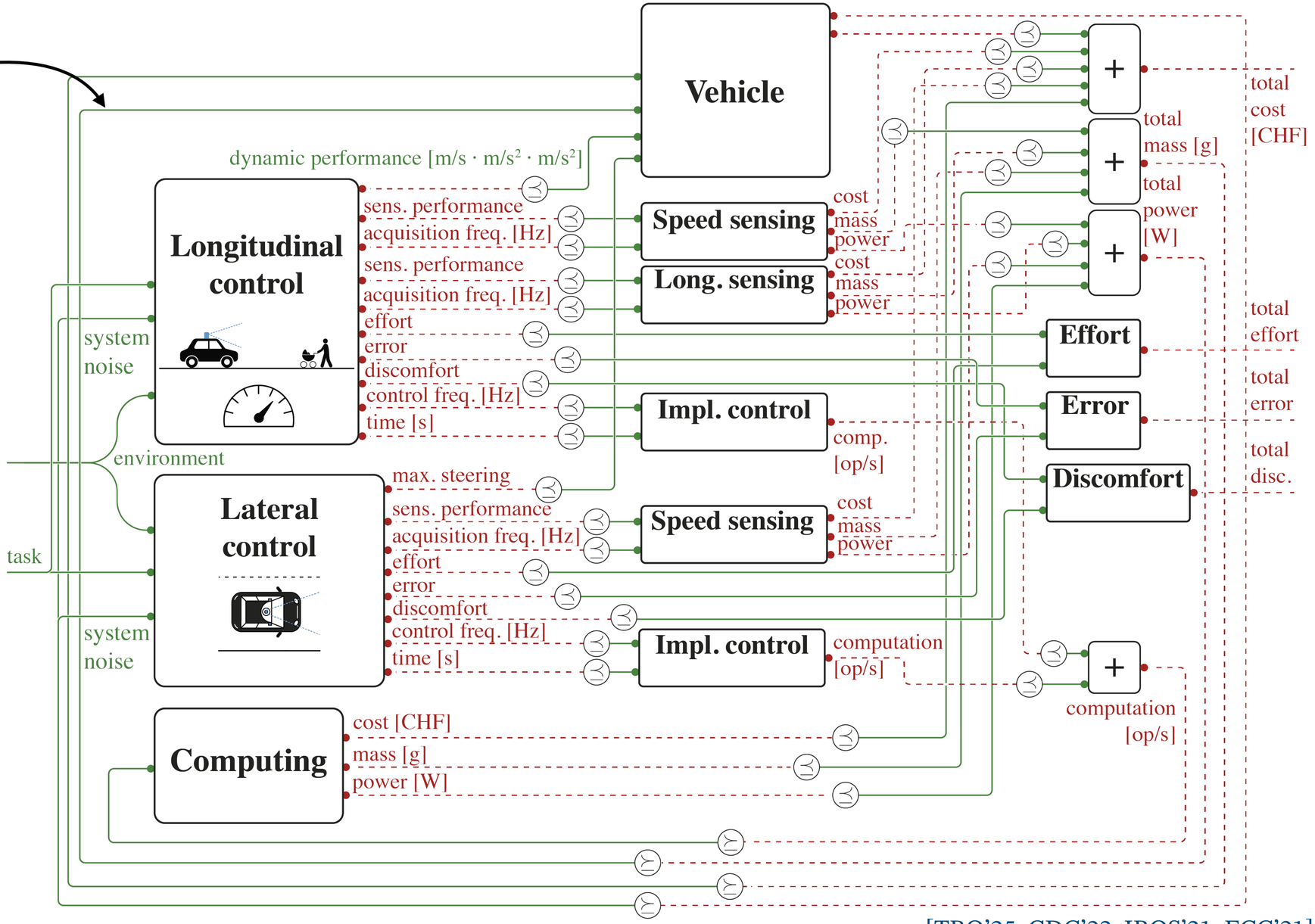
*Technologies don't need to exist already - parametric with **time***



*Decentralized - **humans** in the loop*

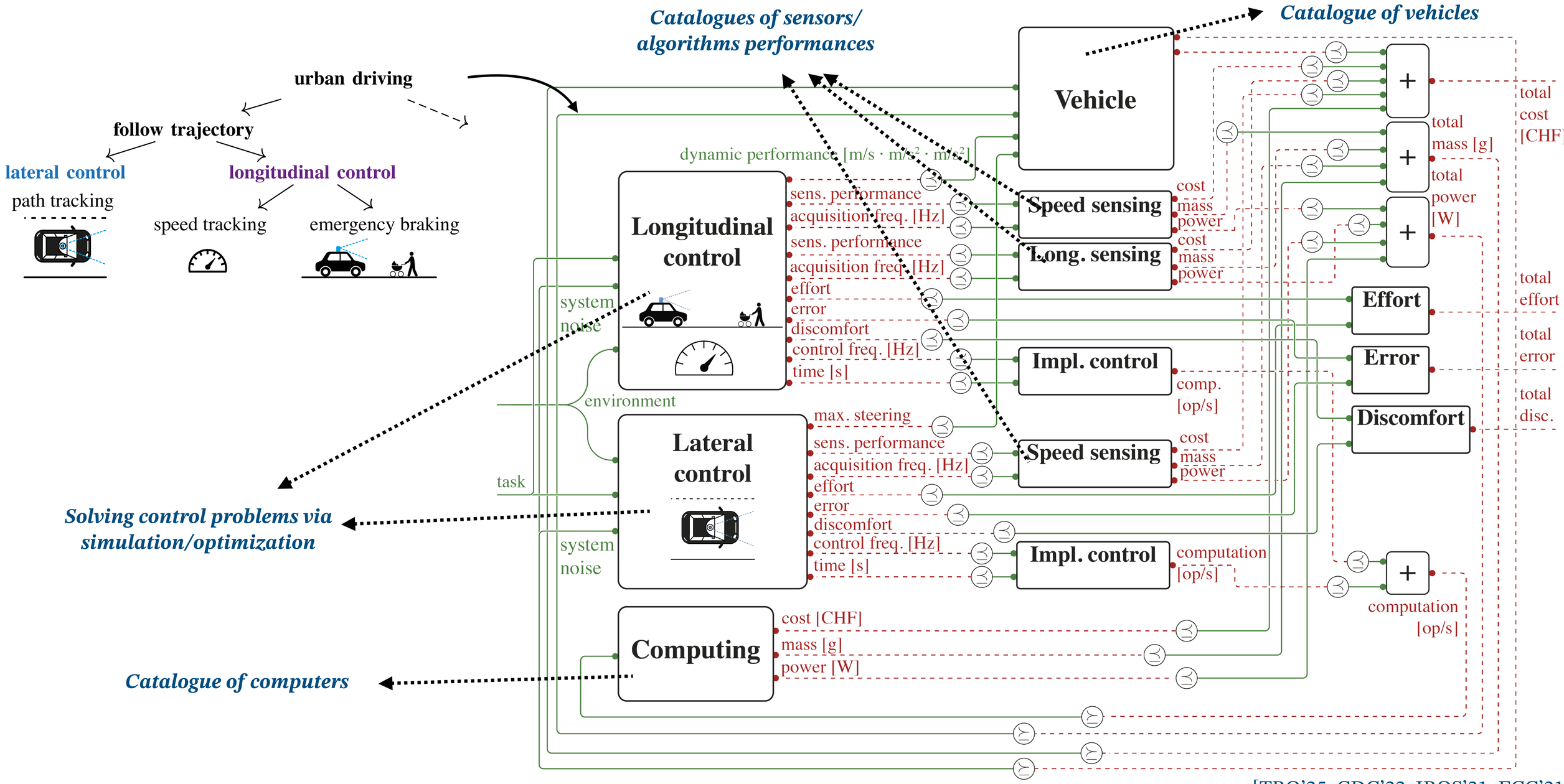
Task-driven co-design of an autonomous vehicle





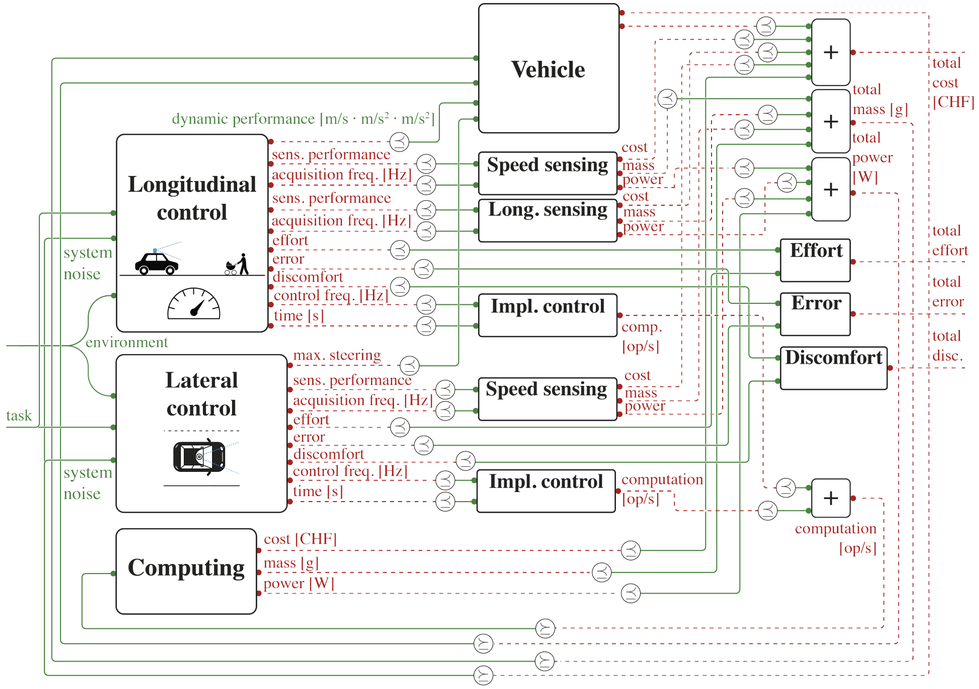
[TRO'25, CDC'22, IROS'21, ECC'21]

Task-driven co-design of an autonomous vehicle

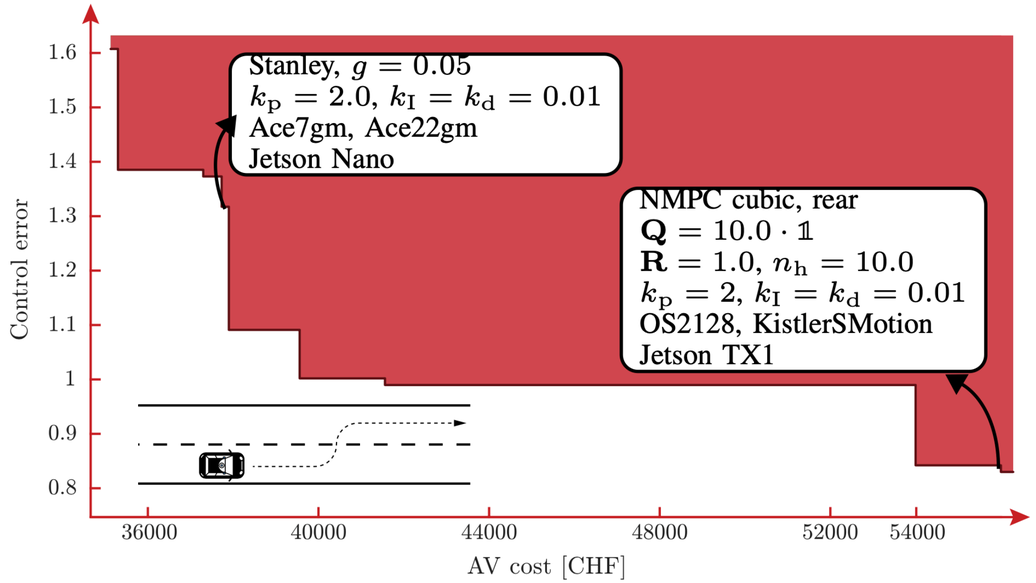


[TRO'25, CDC'22, IROS'21, ECC'21]

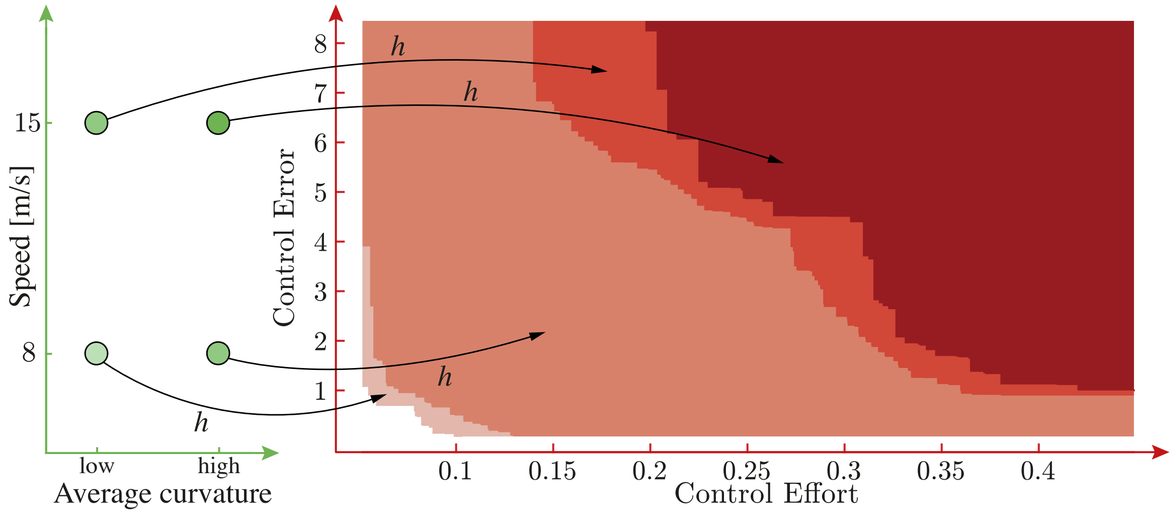
We can find optimal designs, with insights at heterogeneous abstraction levels

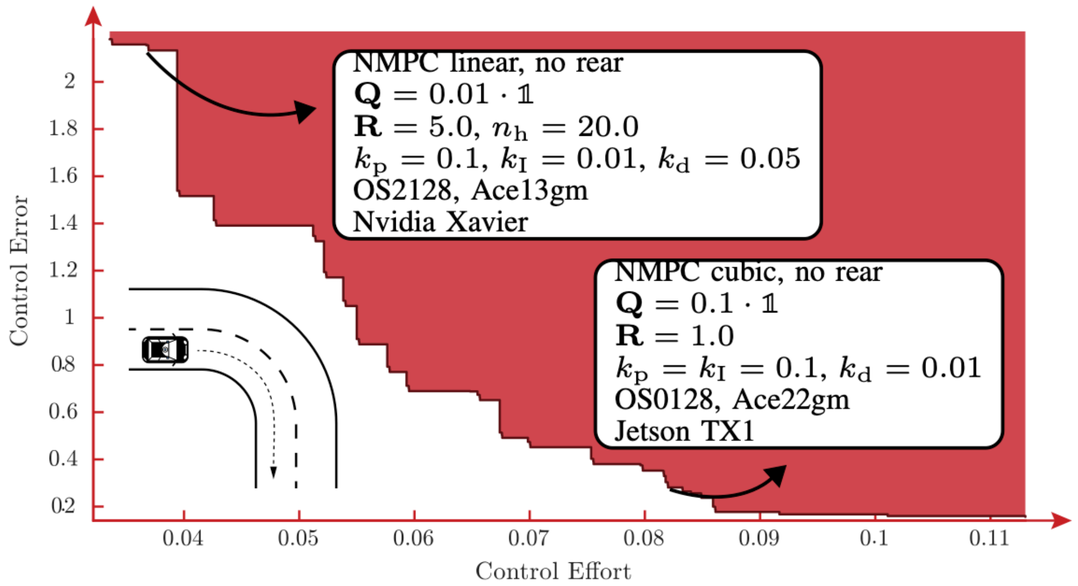


Fix an environment
Fix a task



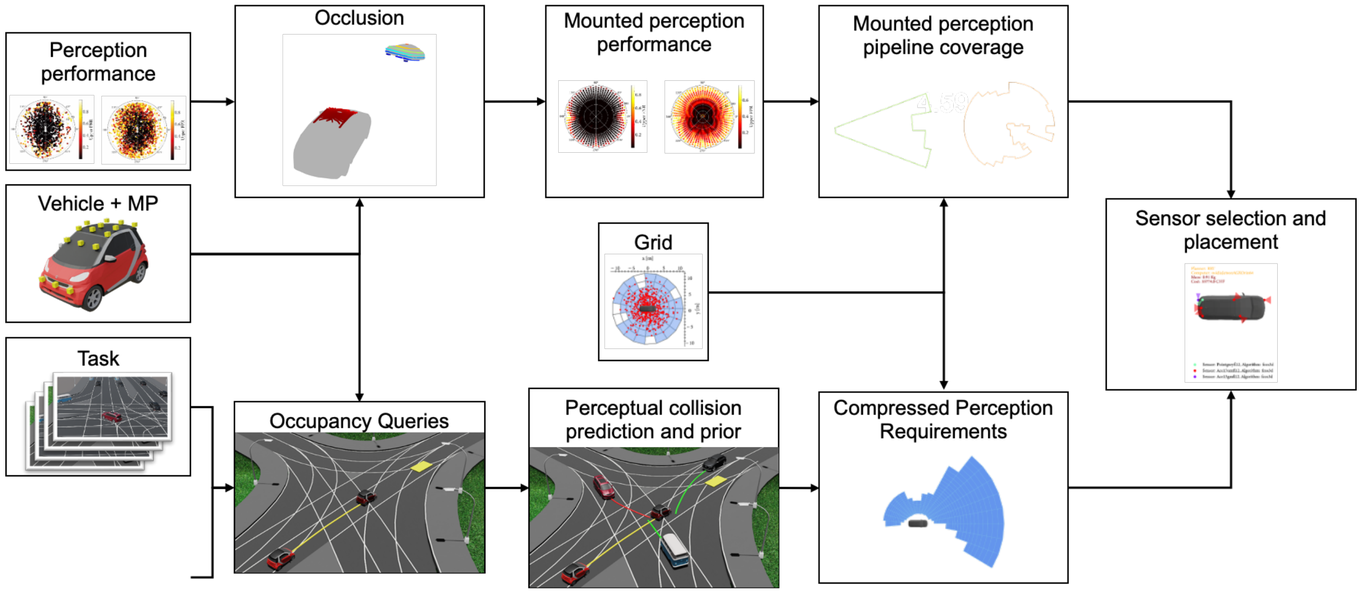
Monotonicity in task complexity

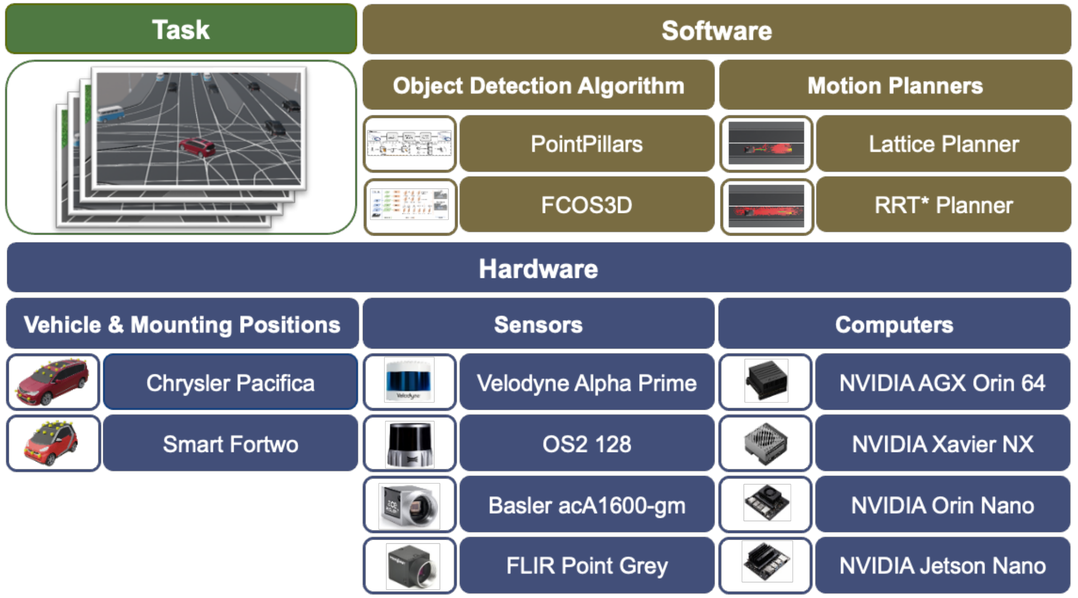


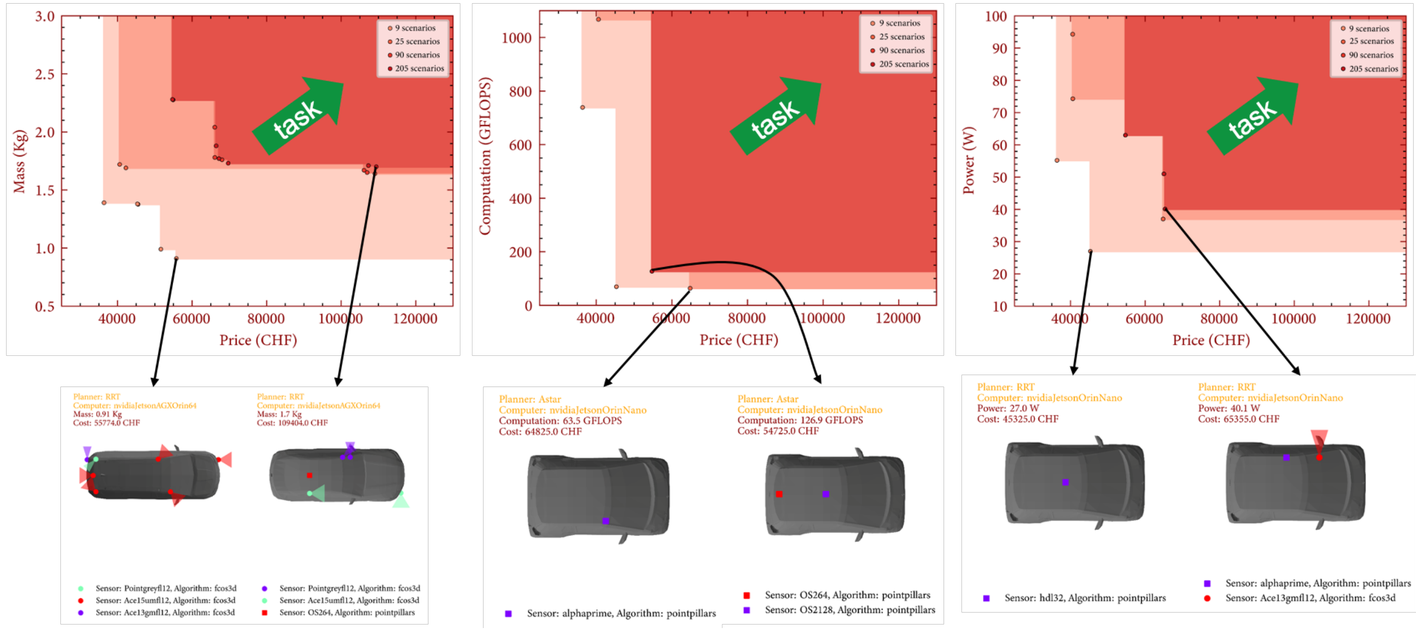


[TRO'25, CDC'22, IROS'21, ECC'21]

We can increase the granularity of our co-design models





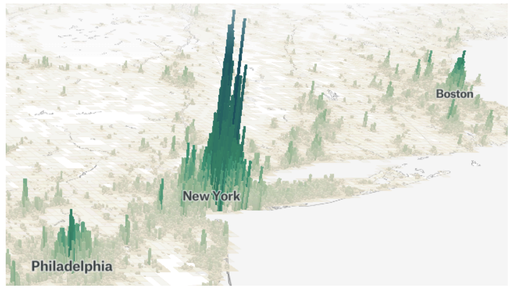


[T-RO'25]

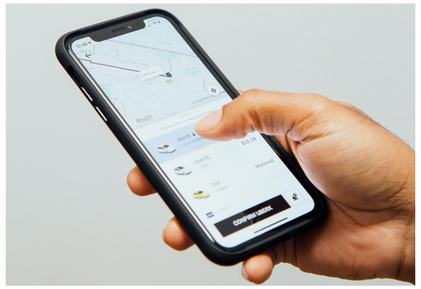
Co-design across scales: from autonomy to mobility systems

- Mobility systems are **under pressure**

Travel demand is changing
By 2050, 68% of population in cities



Need for **service design** and **regulations**
Over 1,000% ride-hailing increase in 2012-22



Need to meet **sustainability goals**
Cities cause 60% of GHGs, 30% from mobility

- We look at the problem from the perspective of **municipalities** and **policy makers**

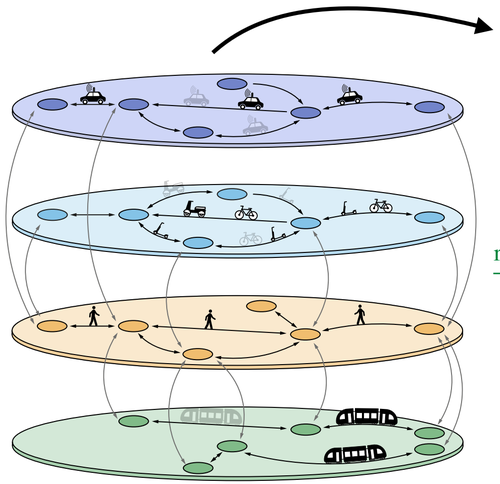
How many vehicles should we allow?
Which infrastructure investments?

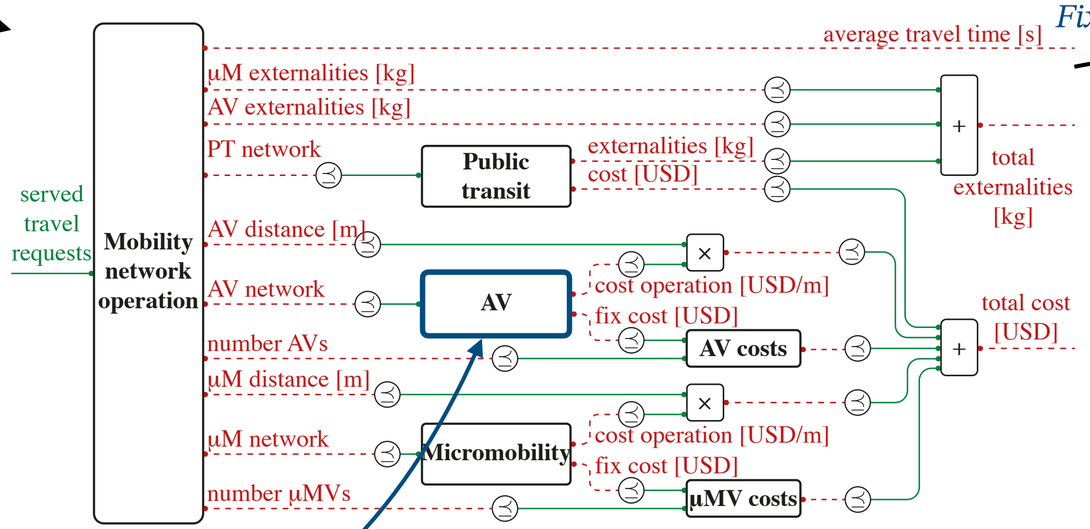
How performant?
Which services to encourage?

- Need for **demand-driven** co-design of **mobility solutions** and the **intermodal network** they enable

- Several **disciplines** involved (transportation science, autonomy, economics, policy-making)

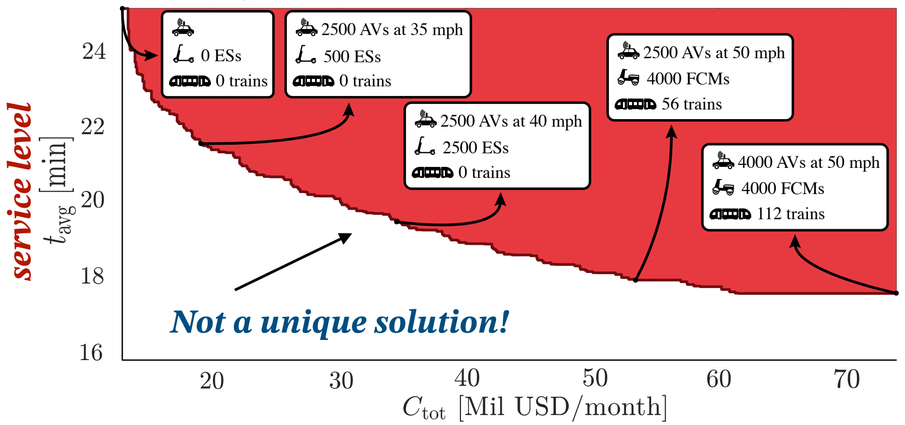
Co-design to enable user-friendly tools to assess the impact of future mobility solutions





Fix demand, minimize costs

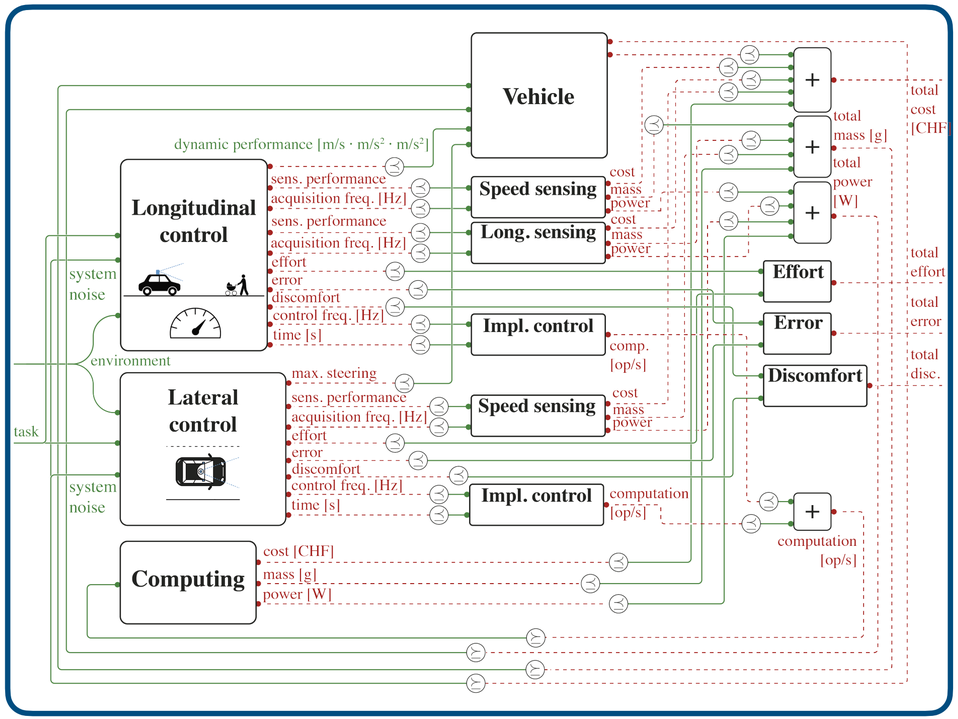
Washington D.C. in 2030



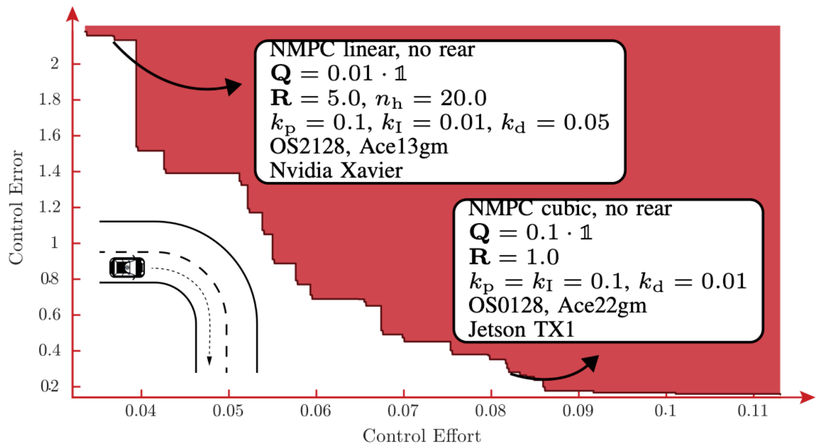
Not a unique solution!

investments

compositionality (vertical composition)



Fix environment, task



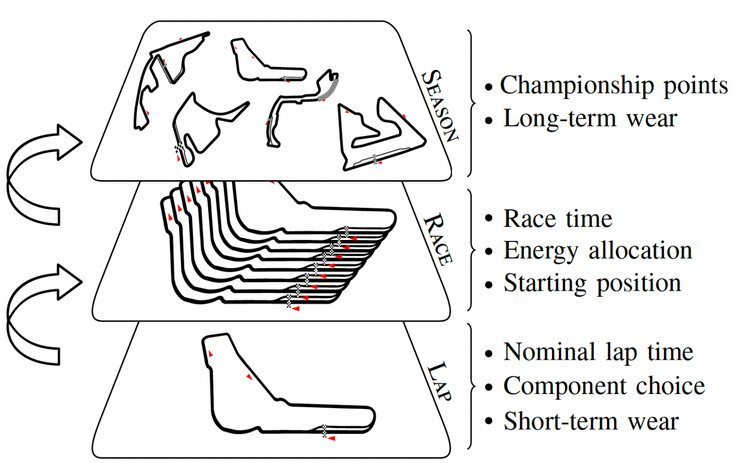
Details about software and hardware implementations, in a way that was not possible before

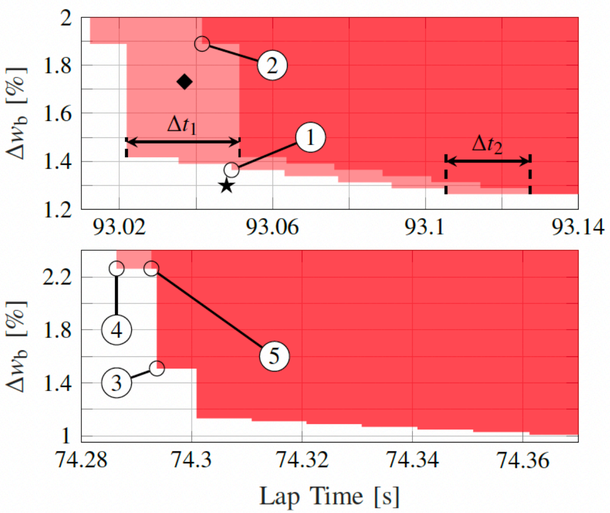
Sidara

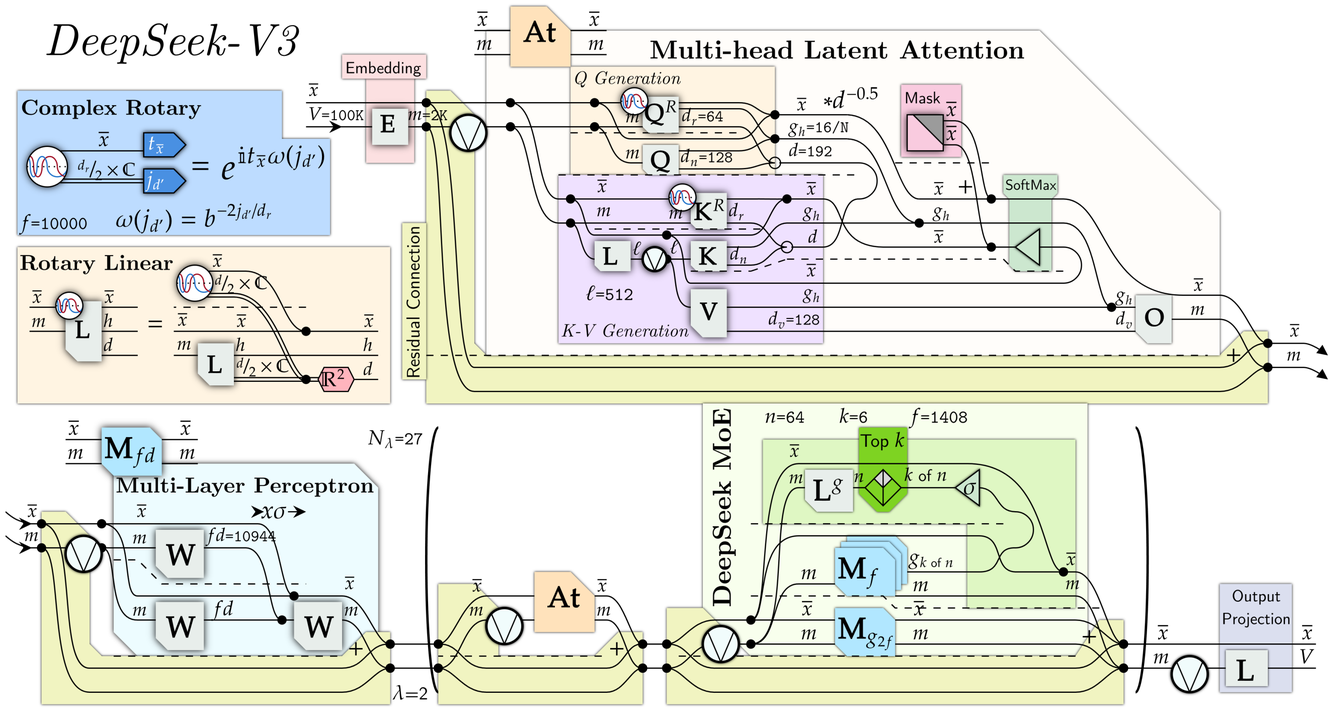
U.S. DEPARTMENT OF ENERGY

[TRO'25, TNSE'23, CDC'22, IROS'21, ECC'21, ITSC'20]

Co-design to enable user-friendly tools to assess the impact of future mobility solutions







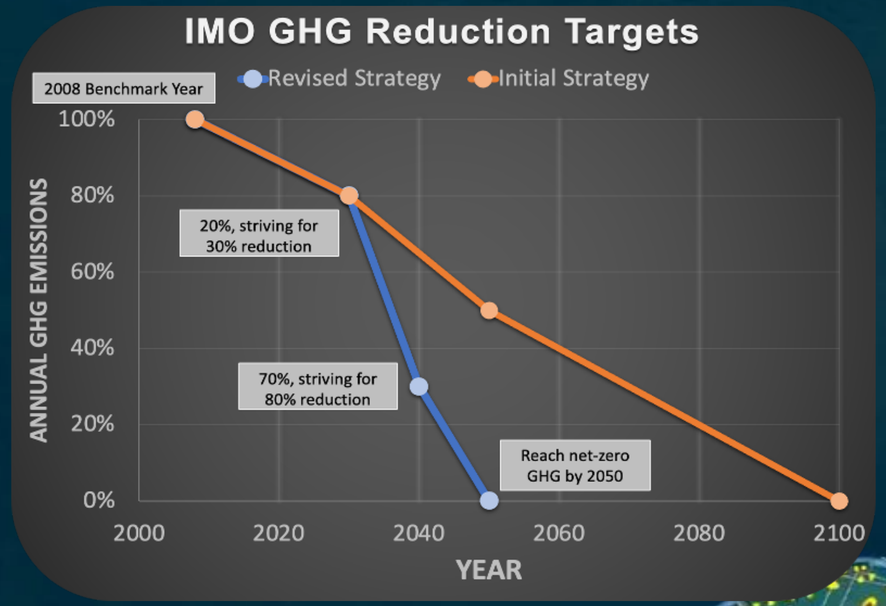
and many more...

[TRO'25, TMLR'25, T-IV'25 (sub), NMI'25 (sub)]

Sustainability and Performance of the Maritime Shipping Ecosystem

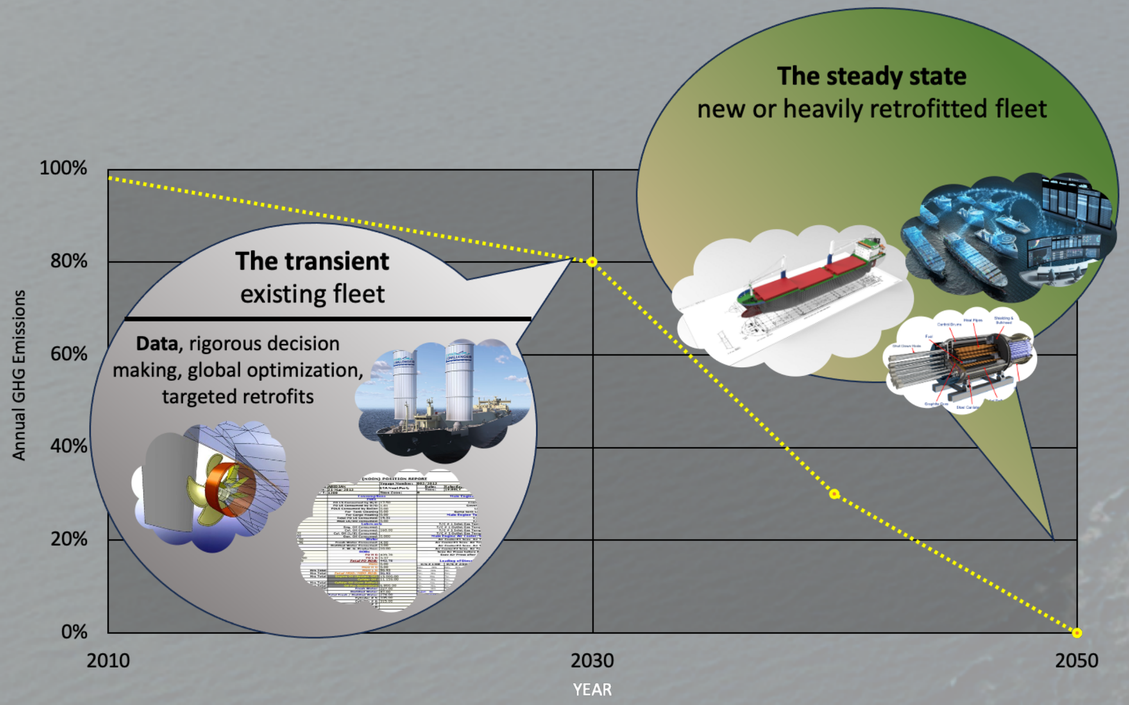
- Ships deliver **90% of the world's trade** by volume — 1.5 tons of goods/person/year
- Close to **110k vessels** around the globe (for reference, there are around 1.7 billion cars around the globe)
- Total shipping trade value was $**14.5 trillion** in 2019 (about the GDP of China)
- Increase of volume by a **factor of 4** in the last 10 years
- If shipping was a country, it would be the **7th emitter globally**

Rank	Country
1	China
2	United States
3	India
4	Russia
5	Japan
6	Germany
7	[redacted]
8	Iran
9	Saudi Arabia



MIT Maritime Shipping Consortium, with Themis Sapsis & Fotini Christia

The path toward decarbonization of the maritime shipping ecosystem



Challenges

- Ships currently ordered will live **30 years**
- Data is highly multimodal - **no standards/algorithms**
- Many **stakeholders**, **policies**, and **regulations**

Multiple active threads:

Large-scale fleet control & optimization

New ship designs and impact on markets

Regulations, incentives, dynamic teaming

Emissions, resource allocation, auctions

Agenda

- **Motivation**
  - *New challenges of engineering design*
  - *Motivation from autonomy and mobility*
  - *Desiderata for co-design*
- **Monotone Co-Design**
  - *Modeling design problems*
  - *Examples across domains*
  - *Design queries and optimization*
  - *From autonomy to mobility systems*
- **Strategic interactions**
  - *Game theory to deal with strategic interactions*
- **Outlook on future research**

Website containing all papers and more pointers:

https://zardini.mit.edu

Complexity when designing complex systems

Large systems
- Many components
- Heterogeneous natures
- Multiple objectives

Strategic interactions
- Many agents
- Heterogeneous interactions
- Conflicts/collaborations

Agenda

- **Motivation**
  - *New challenges of engineering design*
  - *Motivation from autonomy and mobility*
  - *Desiderata for co-design*
- **Monotone Co-Design**
  - *Modeling design problems*
  - *Examples across domains*
  - *Design queries and optimization*
  - *From autonomy to mobility systems*
- **Strategic interactions**
  - *Game theory to deal with strategic interactions*

- **Outlook on future research**

Website containing all papers and more pointers:
https://zardini.mit.edu

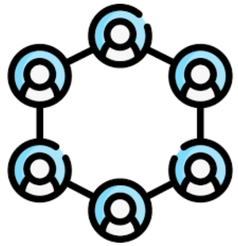
Modeling & Algorithmic Foundations

Societal Applications

User-friendly Tools

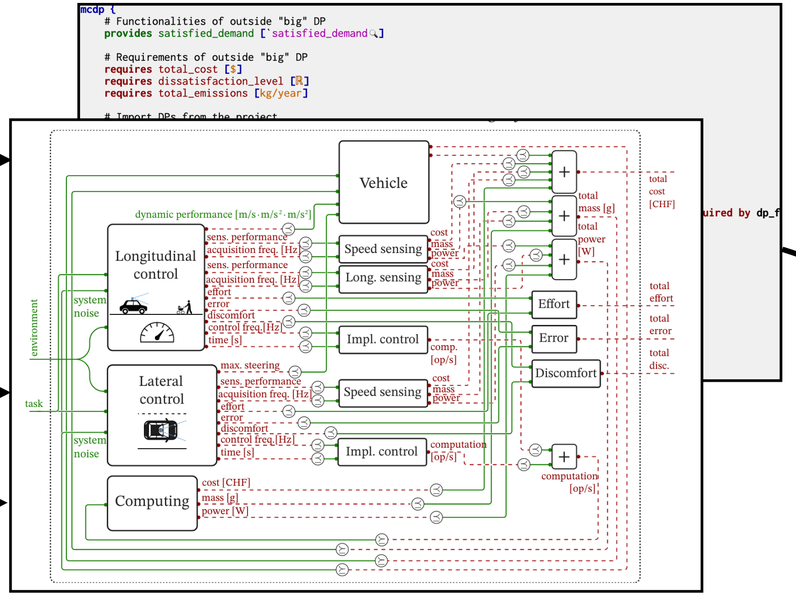
Distributed Collaborative Learning and Designing of Complex Systems

Distributed stakeholders across disciplines



Task specifications & objectives

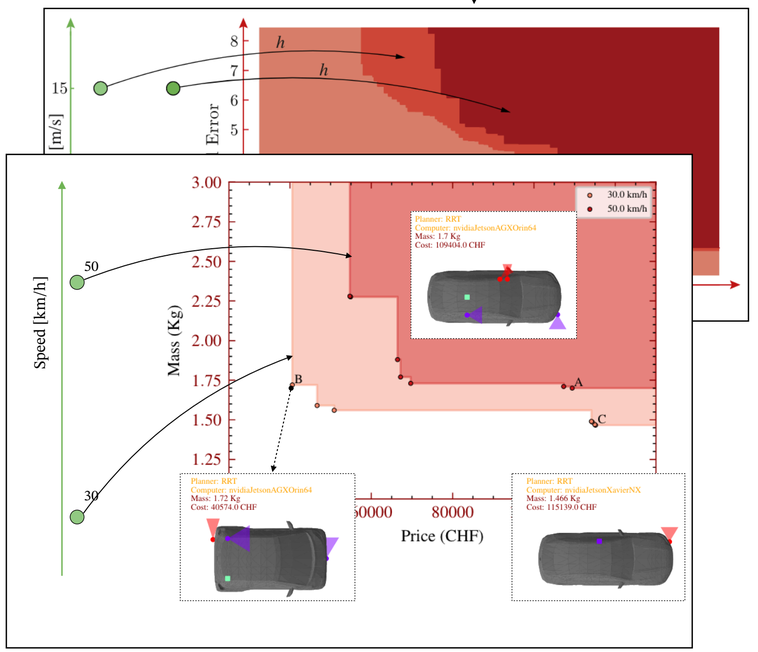
Models (data-driven & classic)



Optimization Queries

Datasets knowledge

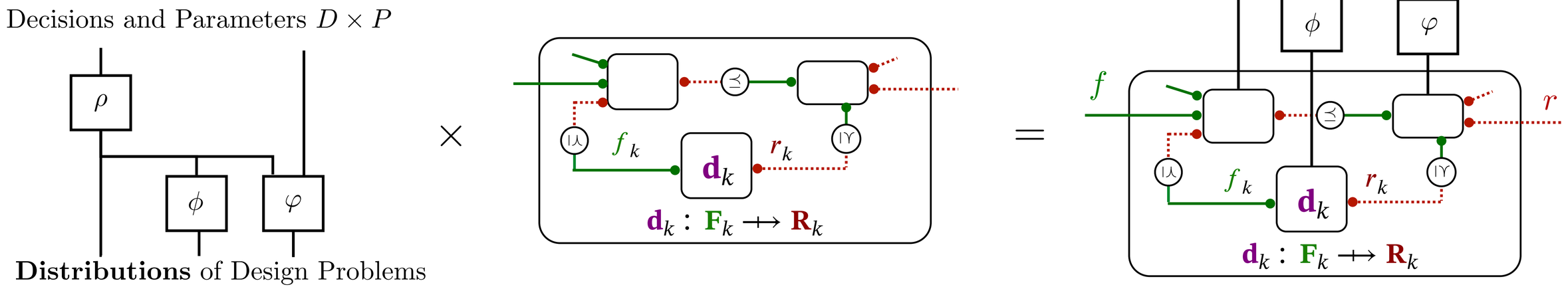
Trade-offs and optimal design solutions



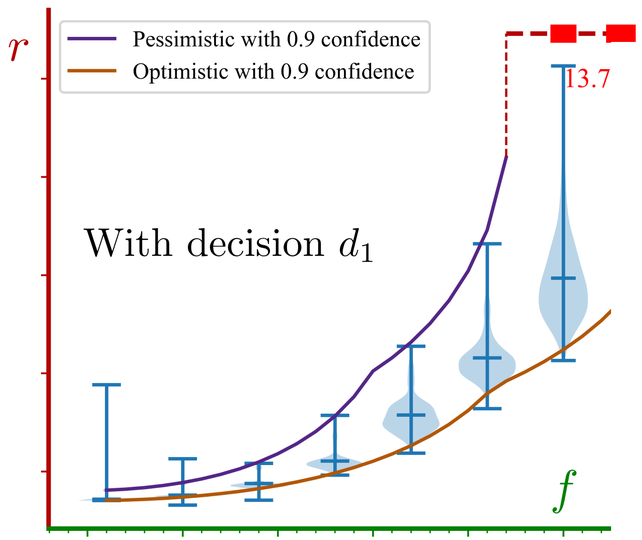
Design solution to implement

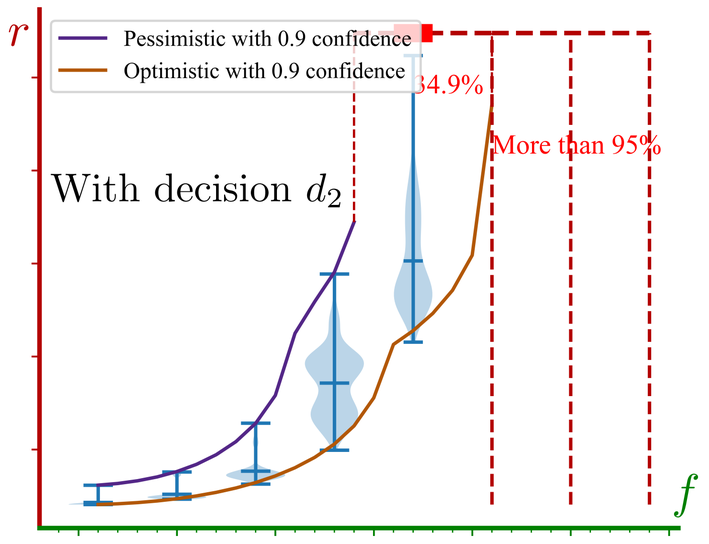
Compositional Modeling of Uncertainty

- Categorical model for uncertainty × Co-design = Composable parameterized uncertain co-design
  - Compositional properties again **guaranteed**



- Stochastic optimization: best decision with uncertainty
- Efficient and functorial queries
- Adaptive sampling of problems and parameters
- Learning parameters from experiments
- Exploration and exploitation trade-off
- ……





[ACT'25, CDC'25 (sub)]

My lab is building the next generation tools for systems design optimization

Modeling and Algorithmic Foundations

Leveraging **optimization**, **control theory, game theory, domain theory,** and **applied category theory**:

- Extend and improve current **modeling** & **solution algorithms** for **multi-objective** design optimization
- Promote **interdisciplinarity** by bridging the gap between **standard optimization** and **co-design**
- Explicitly account for **strategic interactions** of stakeholders, developing a theory of **co-design games**

Societal Applications

Mobility, networks, infrastructure
Strategic interactions at all levels

Mission-driven autonomy

Aerospace, automotive, production chains, energy and data networks

User-friendly Tools

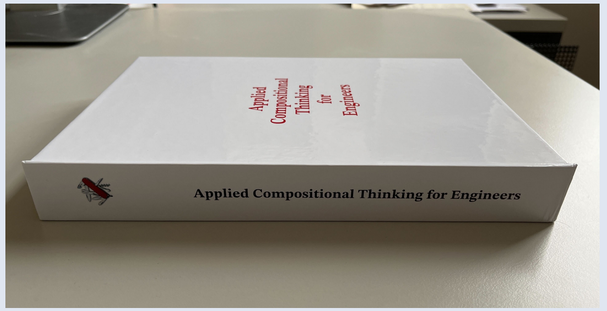
Collaborative, intellectually tractable

Authorities & Industry

Literature, workshops, classes

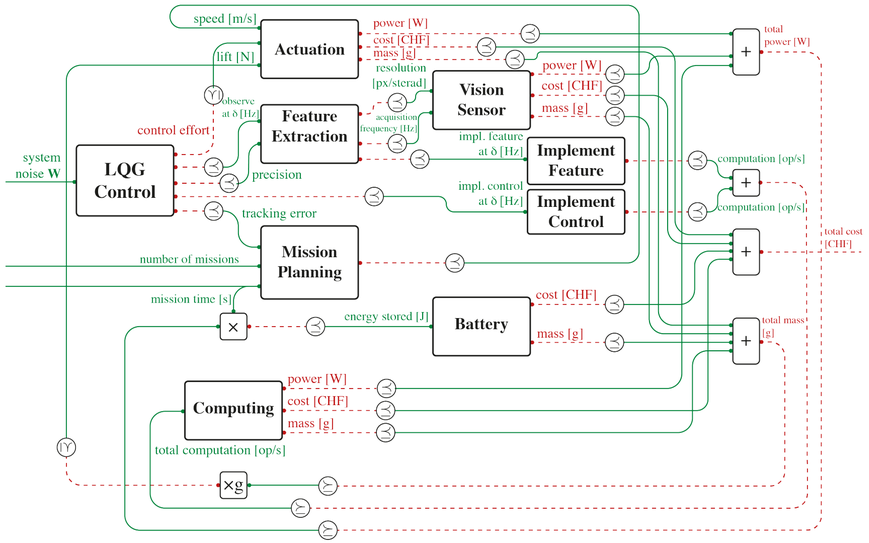
Take-aways

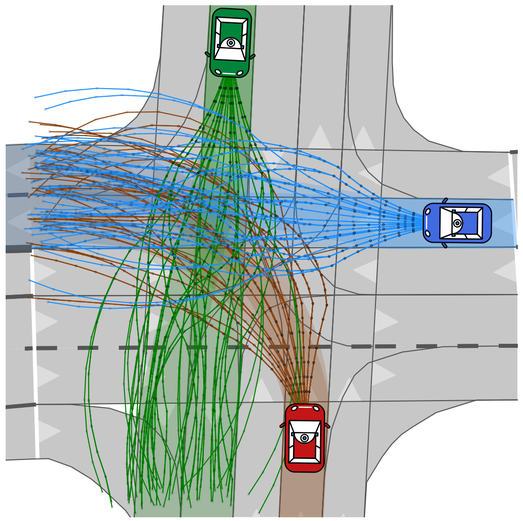
- A new approach to **co-design** designed to work **across fields** and **scales**.
- It is:
  - **Compositional** *horizontally* and *hierarchically*.
  - Supports both **data-driven** and **model-based** components.
  - **Computationally tractable**.
  - **Intellectually tractable.**
- Currently: extend **modeling** and **algorithmic** capabilities

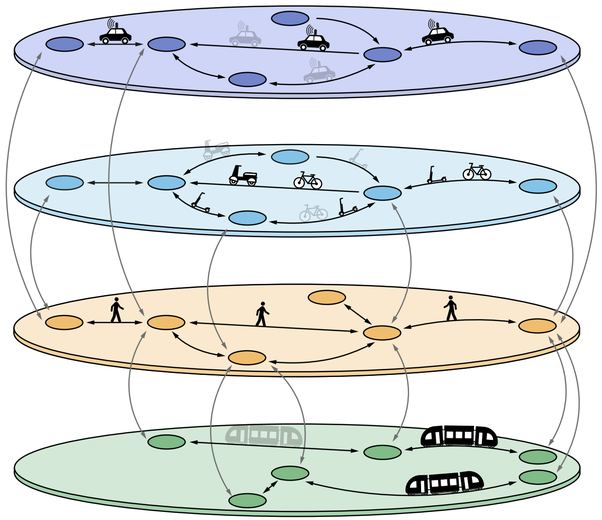


Access the book at:
https://bit.ly/3qQNrdR

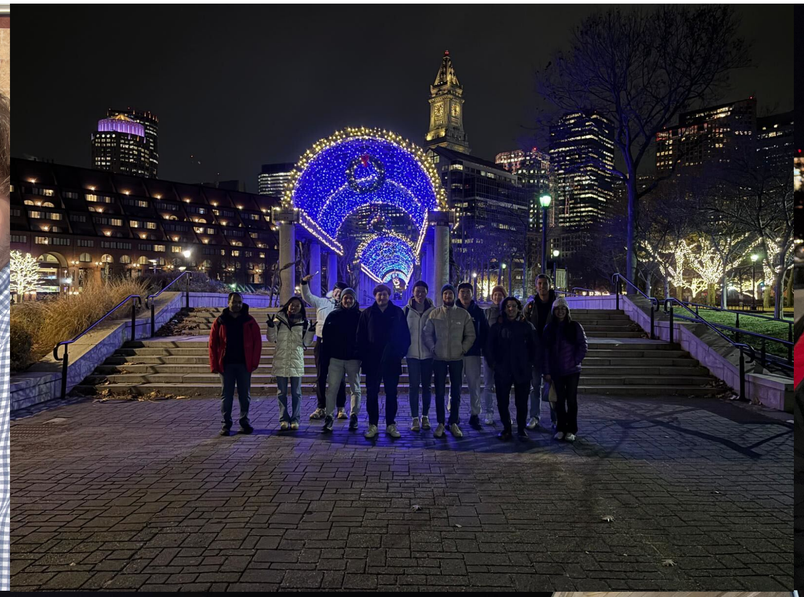
- We need to account for **strategic interactions** of **designers**:
  - **Posetal games**: A new class of games, where **utilities** are **posets**
- Currently: **uncertainty** and **new computational** schemes







My lab… a bunch of amazing people that eat… and do exciting research



Take-aways

- A new approach to **co-design** designed to work **across fields** and **scales**.
- It is:
  - **Compositional** horizontally and hierarchically.
  - Supports both **data-driven** and **model-based** components.
  - **Computationally tractable**.
  - **Intellectually tractable.**
- Future: extend **modeling** and **algorithmic** capabilities

- We need to account for **strategic interactions** of **designers**:
  - **Posetal games**: A new class of games, where **utilities** are **posets**
- Future: **uncertainty** and **computational** schemes

Questions?

I'm hiring/welcoming visitors

zardini.mit.edu