EECE 5550
Mobile Robotics

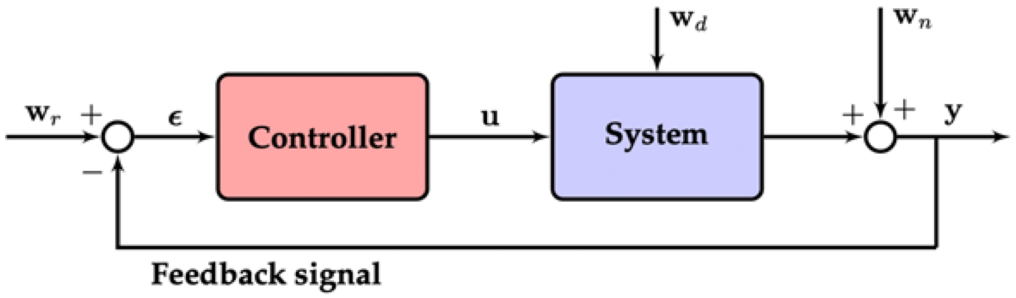
Lecture 21: Introduction to Optimal and Model Predictive Control

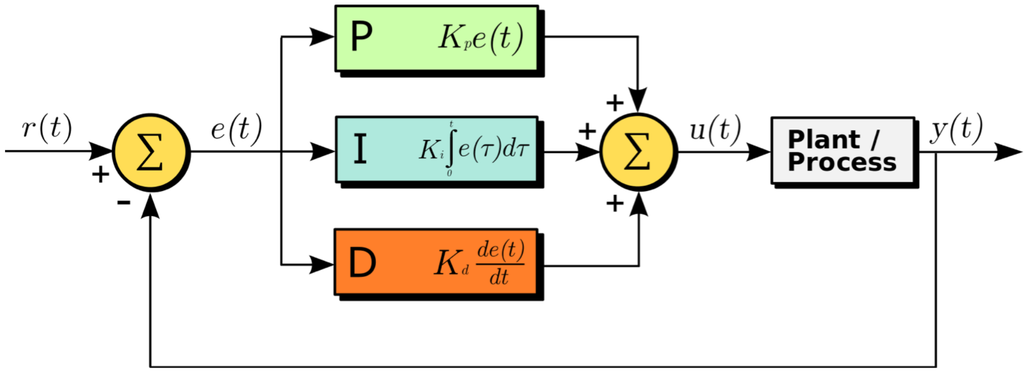
Derya Aksaray
Assistant Professor
Department of Electrical and Computer Engineering
Northeastern University

Recap

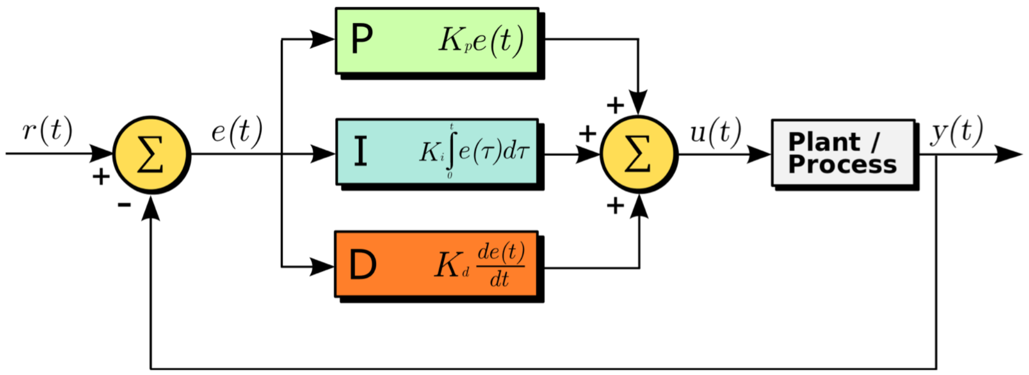
Last time: Basic feedback control

- Control system formulation
- State feedback control
- Proportional-Integral-Derivative (PID) control
- Stability analysis





PID Control Recap



Benefits of PID control

- Conceptually simple
- Relatively straightforward tuning procedures

But:

No natural way of including state or control constraints

- *State constraint:* Should have altitude > 0 for a drone
- *Control constraint:* Turtlebots can't move with 90 mph

PID formulation doesn't support *costs* or *preferences* for controls or trajectories.

- *Trajectory preference:* "faster is better"
- *Control cost:* More aggressive controls typically require more energy, fuel, etc.

Punchline: We might like to have a more expressive framework for modeling these constraints / preferences, and designing controllers that account for them

Agile drone flight

https://youtu.be/meSItatXQ7M

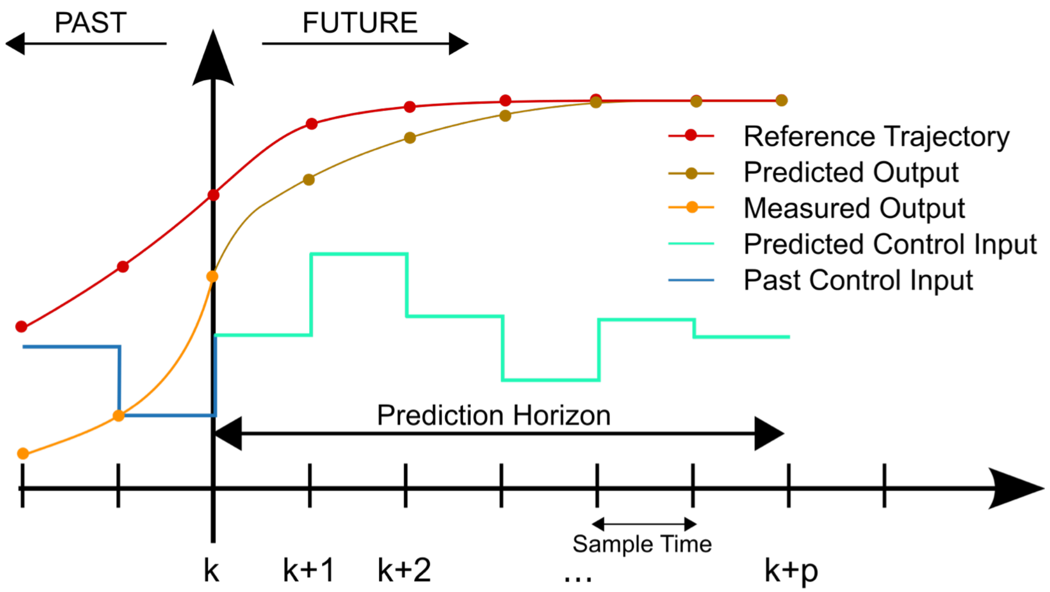
Plan of the day

Optimal control

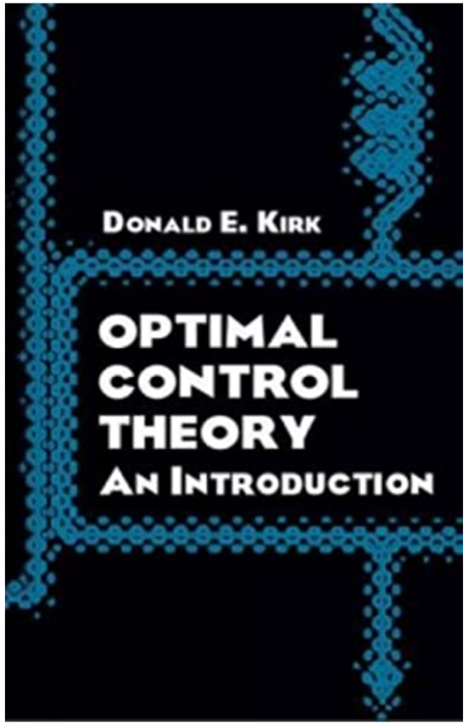
- Problem formulation
- Bellman's optimality principle

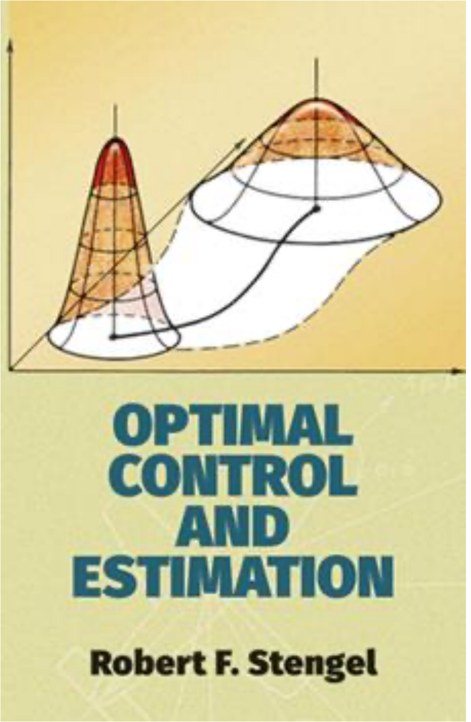
Solving the optimal control problem

- Solving the Bellman-Jacobi-Hamilton equation
- Dynamic programming
- Model predictive control



References





Survey paper: "A Survey of Numerical Methods for Optimal Control", by A.V. Rao

Optimal control problems

- Minimum time problem

$$J = t_f - t_0 = \int_{t_0}^{t_f} dt$$

- Terminal control problem

$$J = \sum_{i=1}^{n} h_i(x_i(t_f) - r_i(t_f))^2 = \left(x(t_f) - r(t_f)\right)^T H(x(t_f) - r(t_f))$$

- Minimum control effort problem

$$J = \int_{t_0}^{t_f} u^T(t) R u(t) dt$$

- Tracking problem

$$J = \int_{t_0}^{t_f} \left(x(t) - r(t)\right)^T Q(x(t) - r(t)) dt$$

Optimal control problem formulation

Performance index Terminal cost Running cost

Minimize:

$$J[x,u,t_f] \triangleq E\left(x(t_f),t_f\right) + \int_{t_0}^{t_f} L[x(t),u(t),t]\,dt$$

$t_f \in \mathbb{R},\ u:[t_0,t_f] \to \mathbb{R}^m,\ x:[t_0,t_f] \to \mathbb{R}^n$

subject to:

$$\dot{x}(t) = f[x(t),u(t),t]$$ state equation (system dynamics)

$$h(x(t),u(t),t) \le 0$$ path (state/control) constraints

$$e\left(x(t_f),t_f\right) = 0$$ endpoint (boundary) conditions

Optimal control problem formulation

Very general formulation! Supports:

- Expressing *preferences* for trajectories & controls (via performance index *J*)
- Complex (nonlinear) system dynamics *f*
- General state and control constraints (via path constraints *h*)
- Partially-constrained boundary condition (via endpoint condition *e*)

But: Decision variables include *functions* $x(t)$ and $u(t)$

- These are *infinite-dimensional objects*
- OCP is ***very*** hard to solve in general …

Optimal control problem (OCP)

Minimize:

$$J[x,u,t_f] \triangleq E\big(x(t_f),t_f\big) + \int_{t_0}^{t_f} L[x(t),u(t),t]\,dt$$

subject to:

$$\dot{x}(t) = f[x(t),u(t),t]$$

$$h(x(t),u(t),t) \leq 0$$

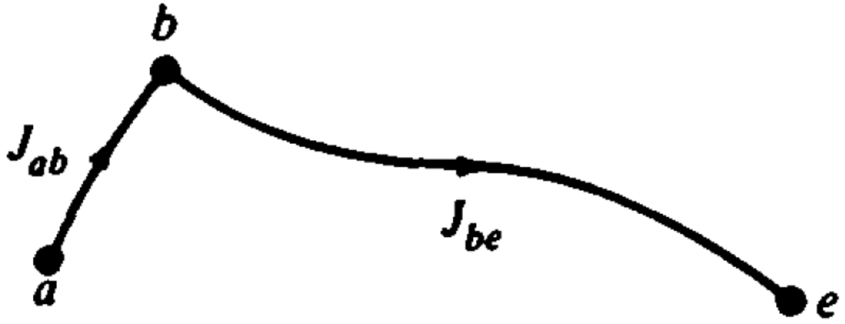
$$e\big(\,x(t_f),t_f\big) = 0$$

Bellman's optimality principle

In order to *search* for an optimal solution $u^*(t), x^*(t)$ of the control problem, we at least need some way of characterizing what optimal solutions look like.

Bellman's *principle of optimality* describes a simple yet very useful property of optimal solutions.

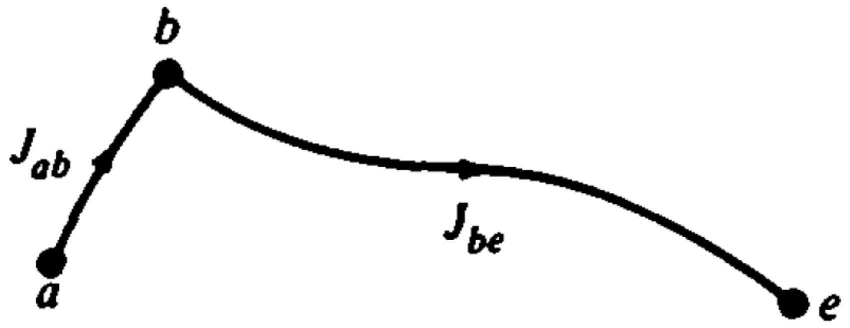
Let $u: [a, e] \to \mathbb{R}^m$ be an optimal control for OCP with state $x: [a, e] \to \mathbb{R}^n$.
Suppose that we follow u over the interval [a,b] for some $b \in (a, e)$, arriving at state $x(b)$.



Bellman's optimality principle

Let $u: [a, e] \to \mathbb{R}^m$ be an optimal control for OCP with state $x: [a, e] \to \mathbb{R}^n$.
Suppose that we follow u over the interval [a,b] for some $b \in (a, e)$, arriving at state *x*(*b*).

What can we say about the control $u: [b, e] \to \mathbb{R}^m$
that sends the state from *x*(*b*) to *x*(*e*)?



Key observation: $u: [b, e] \to \mathbb{R}^m$ must be an optimal control for moving from *x*(*b*) to *x*(*e*)!

Proof (by contradiction): Suppose that there were some other control $v: [b, e] \to \mathbb{R}^m$ such that $J_{be}(v) < J_{be}(u)$
Define $w: [a, e] \to \mathbb{R}^m$ to be the control that applies u on [a,b] and *v* on [b,e].
Then:

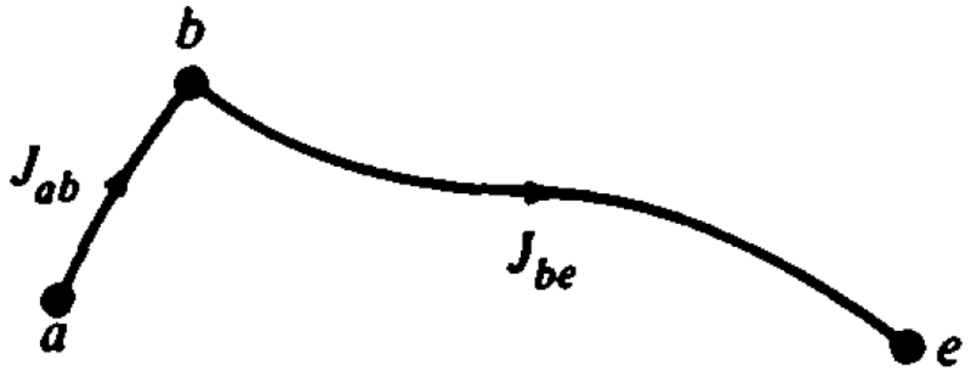
$$J_{ae}(w) = J_{ab}(u) + J_{be}(v) < J_{ab}(u) + J_{be}(u) = J_{ae}(u)$$

But this contradicts optimality of *u*. **QED**

Bellman's optimality principle

"An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision"

$$J_{ae}^* = J_{ab}^* + J_{be}^*$$



The optimal value function

Define the *value function* (or *cost-to-go function*) $V: \mathbb{R}^n \times [0, T] \rightarrow \mathbb{R}$ by:

$$V(x, t) \triangleq \min_{u \in U(x)} E(x(T), T) + \int_t^{t_f} L[x(\tau), u(\tau), \tau]\, d\tau$$

where here the set $U(x)$ denotes the set of admissible controls starting at state *x* at time *t*. Note that $V(x, t)$ reports the cost of the optimal control policy *u*, *starting at state x at time t*.

$$V(x_0, t_0) = \min_{u \in U(x_0)} E(x(T), T) + \int_{t_0}^{t_0 + \Delta t} L[x(t), u(t), t]\, dt + \int_{t_0 + \Delta t}^{t_f} L[x(t), u(t), t]\, dt$$

The optimal value function

- By Bellman's principle, given any $\Delta t > 0$, we have:

$$V(x_0, t_0) = \min_{u \in U(x_0)} E(x(T), T) + \int_{t_0}^{t_0+\Delta t} L[x(t), u(t), t]\, dt + \int_{t_0+\Delta t}^{t_f} L[x(t), u(t), t]\, dt$$

$$V(x_0, t_0) = \min_{u \in U(x_0)} \underbrace{\int_{t_0}^{t_0+\Delta t} L[x(t), u(t), t]\, dt}_{\text{(Optimal) cost to move from } x(t_0) \text{ to } x(t_0+\Delta t)} + \underbrace{V(x(t_0+\Delta t), t_0+\Delta t)}_{\text{Cost-to-go from } x(t_0+\Delta t)}$$

The Hamilton-Jacobi-Bellman (HJB) equation

$$V(x_0, t_0) = \min_{u \in U(x_0)} \int_{t_0}^{t_0+\Delta t} L[x(t), u(t), t]\, dt + V(x(t_0 + \Delta t), t_0 + \Delta t)$$

Taylor expansion of the right-hand side for small Δt:

$$\min_{u \in U(x_0)} \underbrace{L(x_0, u, t_0)\Delta t}_{\text{Differential of integral term}} + V(x_0, t_0) + \underbrace{\frac{\partial V(x_0, t_0)}{\partial x} \cdot \dot{x}(t_0)\Delta t + \frac{\partial V(x_0, t_0)}{\partial t}\Delta t + O(\Delta t^2)}_{\text{Total differential of optimal value function}}$$

$$= \min_{u \in U(x_0)} L(x_0, u, t_0)\Delta t + V(x_0, t_0) + \frac{\partial V(x_0, t_0)}{\partial x} \cdot \underbrace{f[x_0, u, t_0]}_{\text{System dynamics}}\Delta t + \frac{\partial V(x_0, t_0)}{\partial t}\Delta t + O(\Delta t^2)$$

The Hamilton-Jacobi-Bellman (HJB) equation

$$\cancel{V(x_0, t_0)} = \min_{u \in U(x_0)} L(x_0, u, t_0)\cancel{\Delta t} + \cancel{V(x_0, t_0)} + \frac{\partial V(x_0, t_0)}{\partial x} \cdot f[x_0, u, t_0]\cancel{\Delta t} + \frac{\partial V(x_0, t_0)}{\partial t}\cancel{\Delta t} + O(\cancel{\Delta} t^2)$$

Now subtract $V(x_0, t_0)$ from both sides, divide it to Δt , and take limit as $\Delta t \to 0$ to obtain the Hamilton-Jacobi-Bellman eq:

Since it has to be true for any x:

$$\boxed{\frac{\partial V(x,t)}{\partial t} + \min_{u \in U(x)} \left\{ \frac{\partial V(x,t)}{\partial x} \cdot f(x,u,t) + L(x,u,t) \right\} = 0}$$

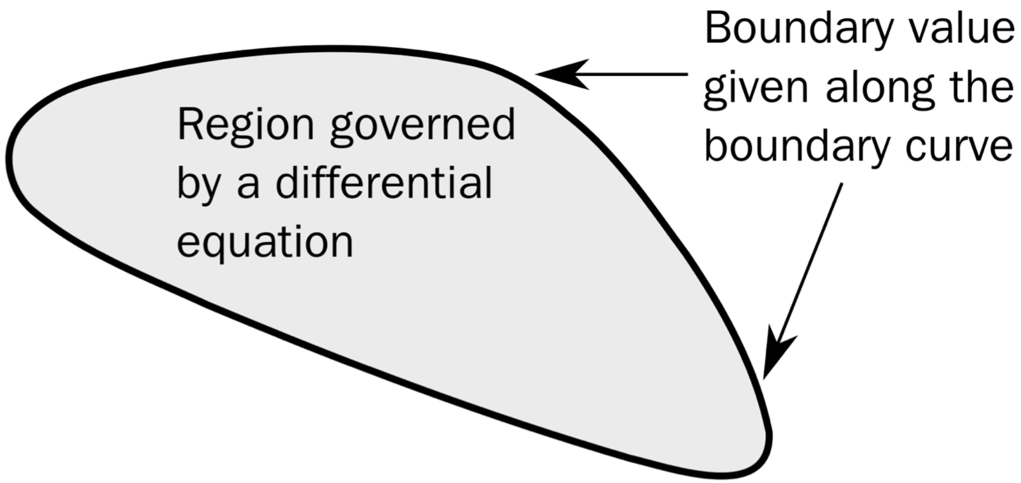
The characterization of the optimal solution

The Hamilton-Jacobi-Bellman Boundary Value Problem

Main Takeaway: The value function $V(x,t)$ is the solution of the following boundary value problem:

$$\frac{\partial V(x,t)}{\partial t} + \min_{u \in U(x)} \left\{ \frac{\partial V(x,t)}{\partial x} \cdot f(x,u,t) + L(x,u,t) \right\} = 0$$

subject to: $$V(x,T) = E(x,T)$$ (endpoint cost)



This is a very well-studied problem so we can solve for $V(x,t)$

Optimal control via the value function

Suppose that we have the value function $V(x, t)$ in hand. How can we use this to solve our control problem?

Recall: The value function $V(x, t)$ gives the optimal *cost-to-go* from state *x* at time *t*.
⇒If we are state *x* at time *t*, we should pick the control *u* towards the *lowest-cost reachable state*

From previous slides, the time derivative of *V* along trajectory $x(t)$ determined by *u* is:

$$\frac{d}{dt}[V(x(t), t)] = \frac{\partial V(x,t)}{\partial t} + \frac{\partial V(x,t)}{\partial x} \cdot f(x, u, t) + L(x, u, t)$$

$$\boxed{u^*(x, t) = argmin_{u \in U(x)} \left\{ \frac{\partial V(x, t)}{\partial x} \cdot f(x, u, t) + L(x, u, t) \right\}}$$

Key point: This shows that knowing the value function gives us an *optimal control policy*: a *pointwise mapping* from the state *x* and time *t* to the optimal control action *u*.

The discrete case

The same approach applies to discrete-time control problems. Consider:

Minimize:

$$J[x,u] \triangleq E\left(x_{t_f}\right) + \sum_{t=0}^{t_f} L(x_t, a_t)$$

subject to:

$$x_{t+1} = f(x_t, a_t)$$

$$a_t \in \Gamma(x_t)$$

$$e\left(x(t_f)\right) = 0$$

where:

- *E* is the terminal cost function
- *L* is running cost function
- *f* is the state transition function
- $\Gamma(x)$ is the set of *admissible actions* in state *x*
- *e* is the endpoint constraint

The discrete case: Bellman's equation

Applying Bellman's principle, we find that the value function $V(x)$ satisfies Bellman's equation:

$$V(x) = \min_{a \in \Gamma(x)} \underbrace{L(x,a)}_{\text{Cost to move from } x \text{ to next state } y = f(x,a)} + \underbrace{V(f(x,a))}_{\text{Cost-to-go from } y}$$

Based on the principle of optimality,

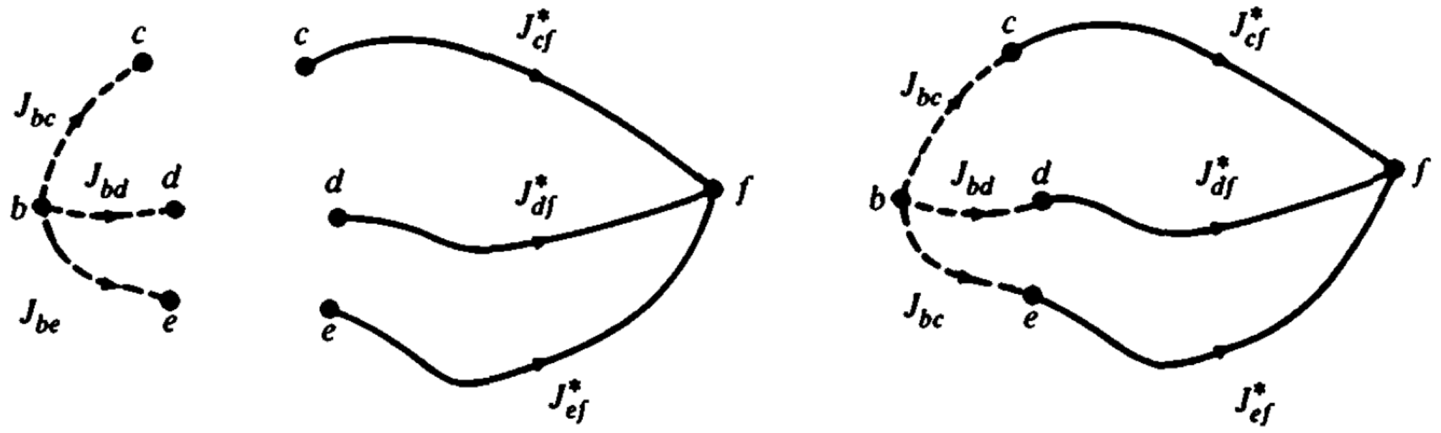
$$C^*_{bcf} = J_{bc} + J^*_{cf}$$

$$C^*_{bdf} = J_{bd} + J^*_{df}$$

$$C^*_{bef} = J_{be} + J^*_{ef}$$

The min of these costs must be the one associated with the optimal decision at b.

The paths resulting from all allowable decisions at b.



The optimal paths from c, d, and e to the terminal point f.

Constructing the discrete-time value function

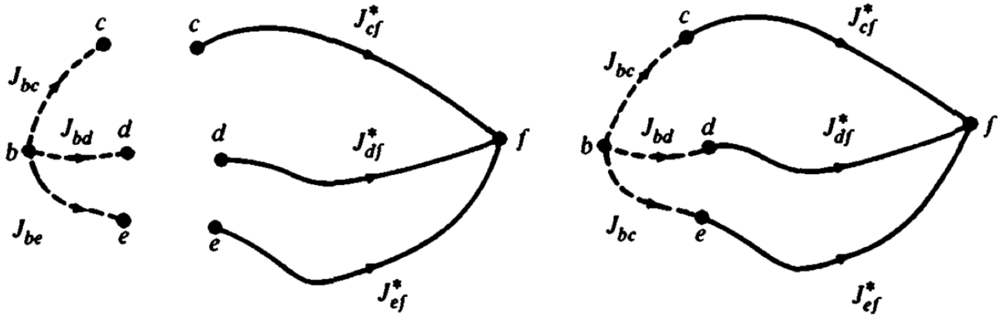
The discrete-time Bellman equation suggests a natural algorithm for constructing the value function via *dynamic programming* (specifically: *backwards induction*)

Base case: For each *terminal state* x, the cost-to-go is simply the terminal cost:

$$V(x) = E(x)$$

Recursive step: Working *backwards* (from the terminal states), calculate the value of *previous* states using the Bellman equation:

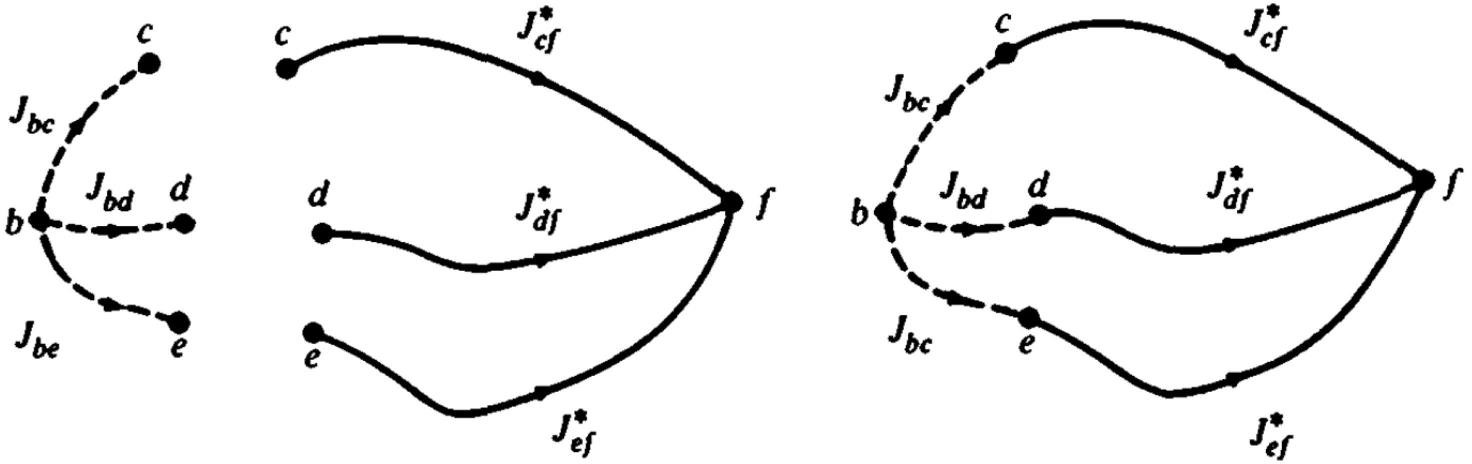
$$V(x) = \min_{a \in \Gamma(x)} L(x,a) + V(f(x,a))$$



Discrete-time optimal control via the value function

As in the continuous-time case, we can recover an *optimal control policy* *u* directly from the value function *V*(*x*):

$$u(x) = argmin_{a \in \Gamma(x)} L(x, a) + V(f(x, a))$$



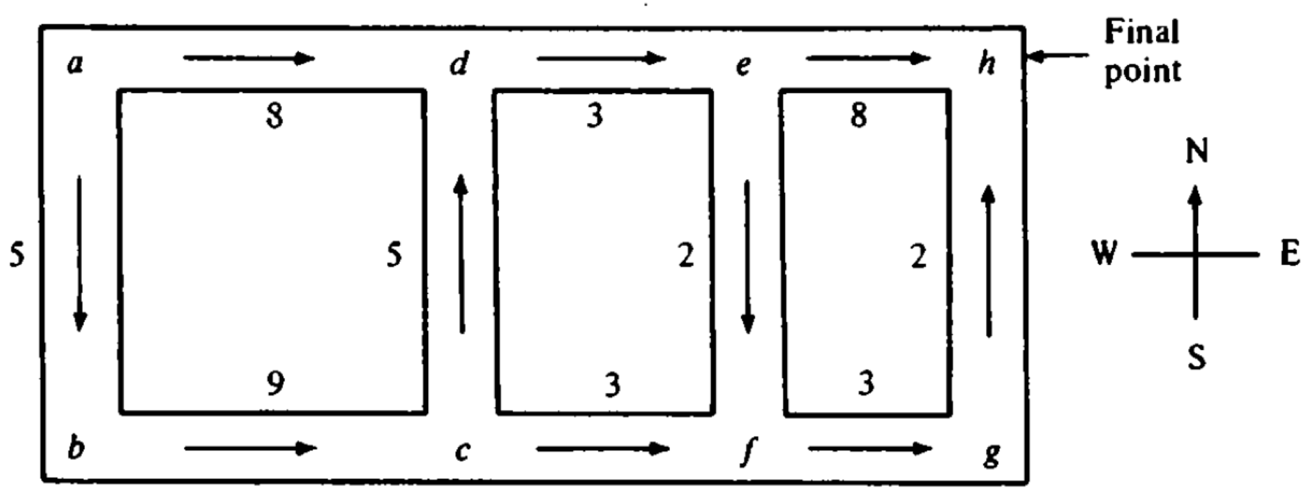
Exercise

Given: The following road network, with travel costs as indicated on the edges and goal state *h*

Find:
- Optimal value function *V*(*x*)
- Optimal control policy *u* that assigns to each state *x* its optimal *heading* (N,S,E,W)

$$V(x) = \min_{a \in \Gamma(x)} L(x,a) + V(f(x,a))$$

$$u(x) = argmin_{a \in \Gamma(x)} L(x,a) + V(f(x,a))$$



Solution

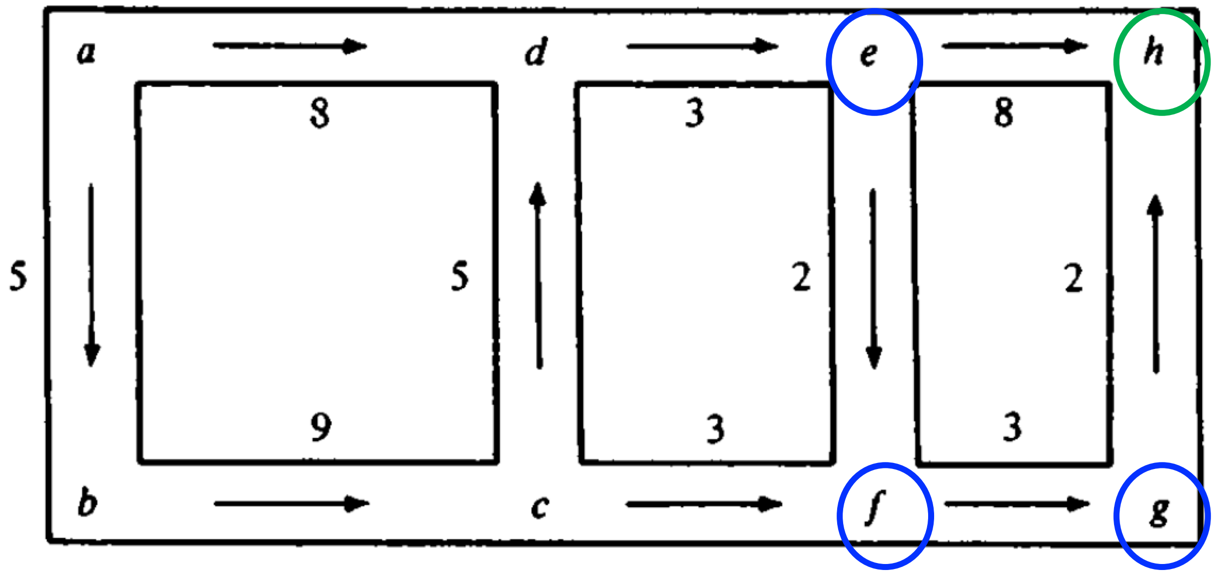
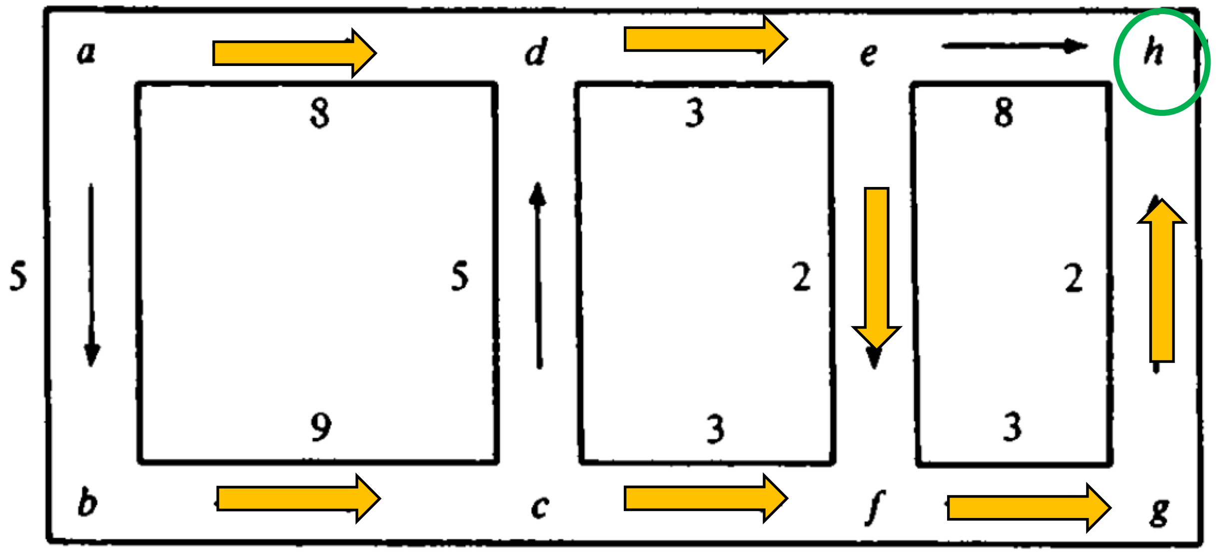


Table 3-1 CALCULATION OF OPTIMAL HEADINGS BY DYNAMIC PROGRAMMING

Current intersection α	Heading u_i	Next intersection x_i	Minimum cost from α to h via x_i $J_{\alpha x_i} + J^*_{x_i h} = C^*_{\alpha x_i h}$	Minimum cost to reach h from α $J^*_{\alpha h}$	Optimal heading at α $u^*(\alpha)$
g	N	h	$2 + 0 = 2$	2	N
f	E	g	$3 + 2 = 5$	5	E
e	E	h	$8 + 0 = 8$		
	S	f	$2 + 5 = 7$	7	S
d	E	e	$3 + 7 = 10$	10	E
c	N	d	$5 + 10 = 15$		
	E	f	$3 + 5 = 8$	8	E
b	E	c	$9 + 8 = 17$	17	E
a	E	d	$8 + 10 = 18$	18	E
	S	b	$5 + 17 = 22$		

Solution



Optimal policy: A mapping from states to actions

Table 3-1 CALCULATION OF OPTIMAL HEADINGS BY DYNAMIC PROGRAMMING

Current intersection α	*Heading* u_i	*Next intersection* x_i	*Minimum cost from* α *to* h *via* x_i $J_{\alpha x_i} + J^*_{x_i h} = C^*_{\alpha x_i h}$	*Minimum cost to reach* h *from* α $J^*_{\alpha h}$	*Optimal heading at* α $u^*(\alpha)$
g	N	h	$2 + 0 = 2$	2	N
f	E	g	$3 + 2 = 5$	5	E
e	E S	h f	$8 + 0 = 8$ $2 + 5 = 7$	7	S
d	E	e	$3 + 7 = 10$	10	E
c	N E	d f	$5 + 10 = 15$ $3 + 5 = 8$	8	E
b	E	c	$9 + 8 = 17$	17	E
a	E S	d b	$8 + 10 = 18$ $5 + 17 = 22$	18	E

Optimal control: recap

The optimal control formulation provides a very expressive framework for modeling problems with nonlinear dynamics and state and control constraints

BUT: Optimal control problems are *very* hard to solve exactly

- Continuous case: HJB boundary-value problem
- Discrete-time case: Dynamic programming

⇒Both of these suffer from the curse of dimensionality

Key question: Can we do something more tractable?

Optimal control problem (OCP)

Minimize:

$$J[x, u, t_f] \triangleq E\left(x(t_f), t_f\right) + \int_{t_0}^{t_f} L[x(t), u(t), t]\, dt$$

subject to:

$$\dot{x}(t) = f[x(t), u(t), t]$$

$$h(x(t), u(t), t) \leq 0$$

$$e\left(x(t_f), t_f\right) = 0$$

Model Predictive Control (MPC)

Motivation: (Global) optimality is an *extremely strong* requirement. We can probably make due with a "pretty good" (but cheaper) control solution that satisfies the problem constraints.

Main idea: We will make two simplifying assumptions:

- We will only plan out to a (fixed) *planning horizon T*
- We will consider a *parametric family* of controls $\hat{u}(\cdot;\alpha)\colon [0,T] \to \mathbb{R}^m$ with *finite-dimensional parameter* $\alpha \in \mathbb{R}^p$. [Ex: polynomials up to order *k*, piecewise constant functions with *k* segments, etc.]

Now observe:

- Control $\hat{u}(t;\alpha)$ is determined by choice of α.
- Having fixed α, system dynamics for *x* then reduce to:

$$\dot{x} = f(x, \hat{u}(t;\alpha), t) = f(x, t; \alpha), \qquad x(0) = x_0$$

This is a first-order ODE for *x*(*t*). We can easily solve this using e.g. Runge-Kutta.
⇒The *trajectory* $x(t;\alpha)$ is likewise parameterized by α.

Payoff: Under this modeling assumption for the controls, the *entire control problem* reduces to a (*finite-dimensional*) *optimization problem over* α**.**

MPC: Problem transcription

Optimal control problem (OCP)

$$\min_{u,x} E\big(x(t_f), t_f\big) + \int_{t_0}^{t_f} L[x(t), u(t), t]\, dt$$

subject to: $\dot{x}(t) = f[x(t), u(t), t]$

$$h(x(t), u(t), t) \le 0$$

$$e\big(\, x(t_f), t_f\big) = 0$$

MPC subproblem (NLP)

$$\min_{\alpha \in \mathbb{R}^p} E(x(t), T) + \int_0^T L[x(t), \hat{u}(t;\alpha), t]\, dt$$

subject to: $\dot{x}(t) = f[x(t), \hat{u}(t;\alpha), t]$

$$h(x(t), \hat{u}(t;\alpha), t) \le 0$$

$$e(x(T)) = 0$$

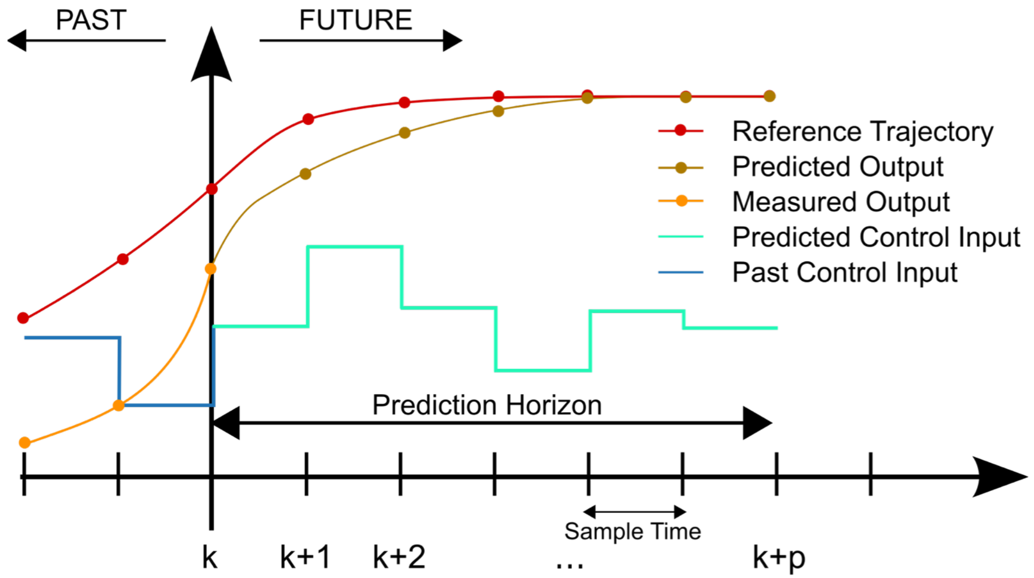
Key point: Assuming a finite horizon T and finite-dimensional parameterization $u(\cdot;\alpha)\colon [0, T] \to \mathbb{R}^m$ for controls reduces the optimal control problem to a standard (finite-dimensional) nonlinear program

Payoff: We can (approximately) solve sparse NLPs *very fast*

Model Predictive Control

Repeat:

1. Given current state $x(0) = x_0$, solve MPC subproblem to generate (locally) optimal control $\hat{u}(\cdot;\alpha^*): [0,T] \to \mathbb{R}^m$ out to planning horizon
2. Apply first stage of control $\hat{u}$
3. Go to step 1



MPC subproblem (NLP)

$$\min_{\alpha \in \mathbb{R}^p} E(x(t), T) + \int_0^T L[x(t), \hat{u}(t;\alpha), t]\, dt$$

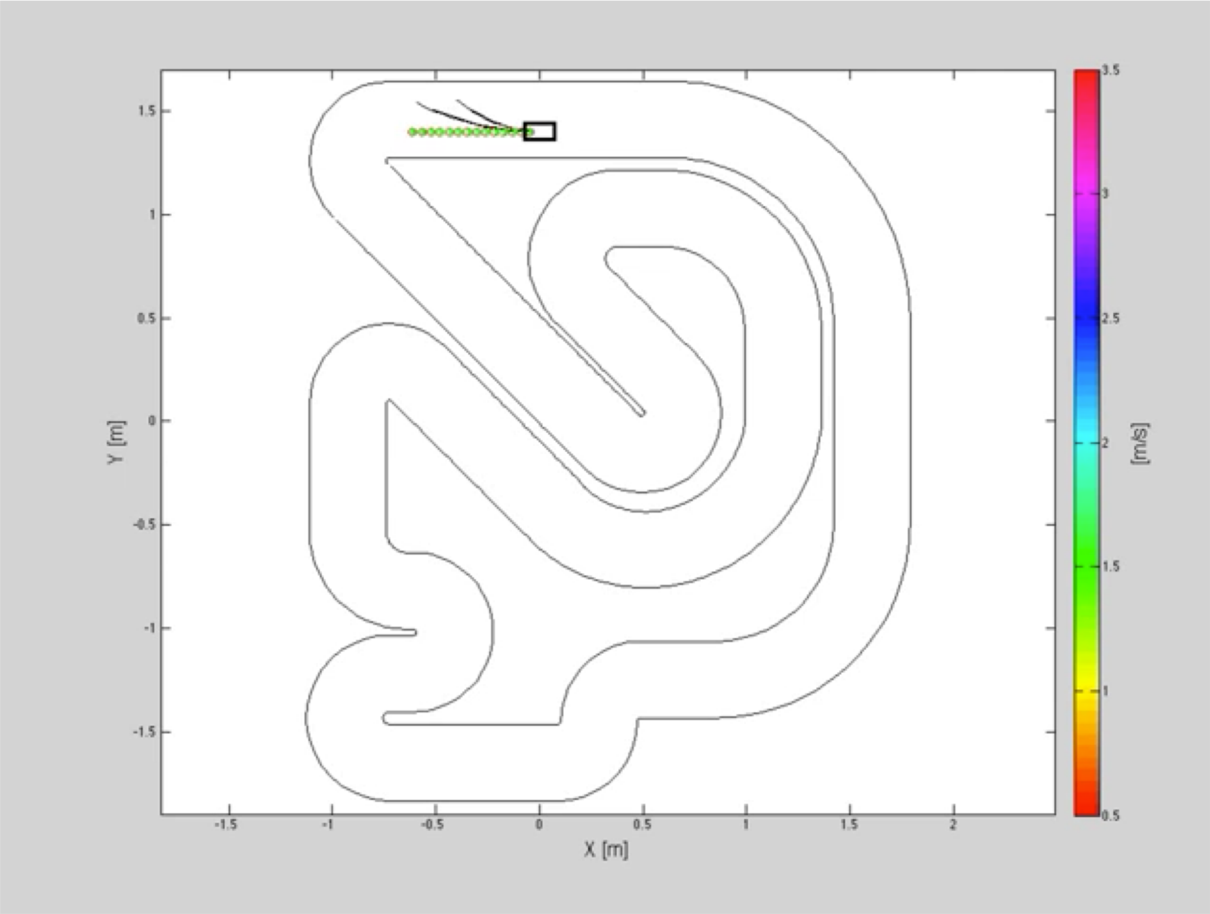
subject to:

$$\dot{x}(t) = f[x(t), \hat{u}(t;\alpha), t]$$

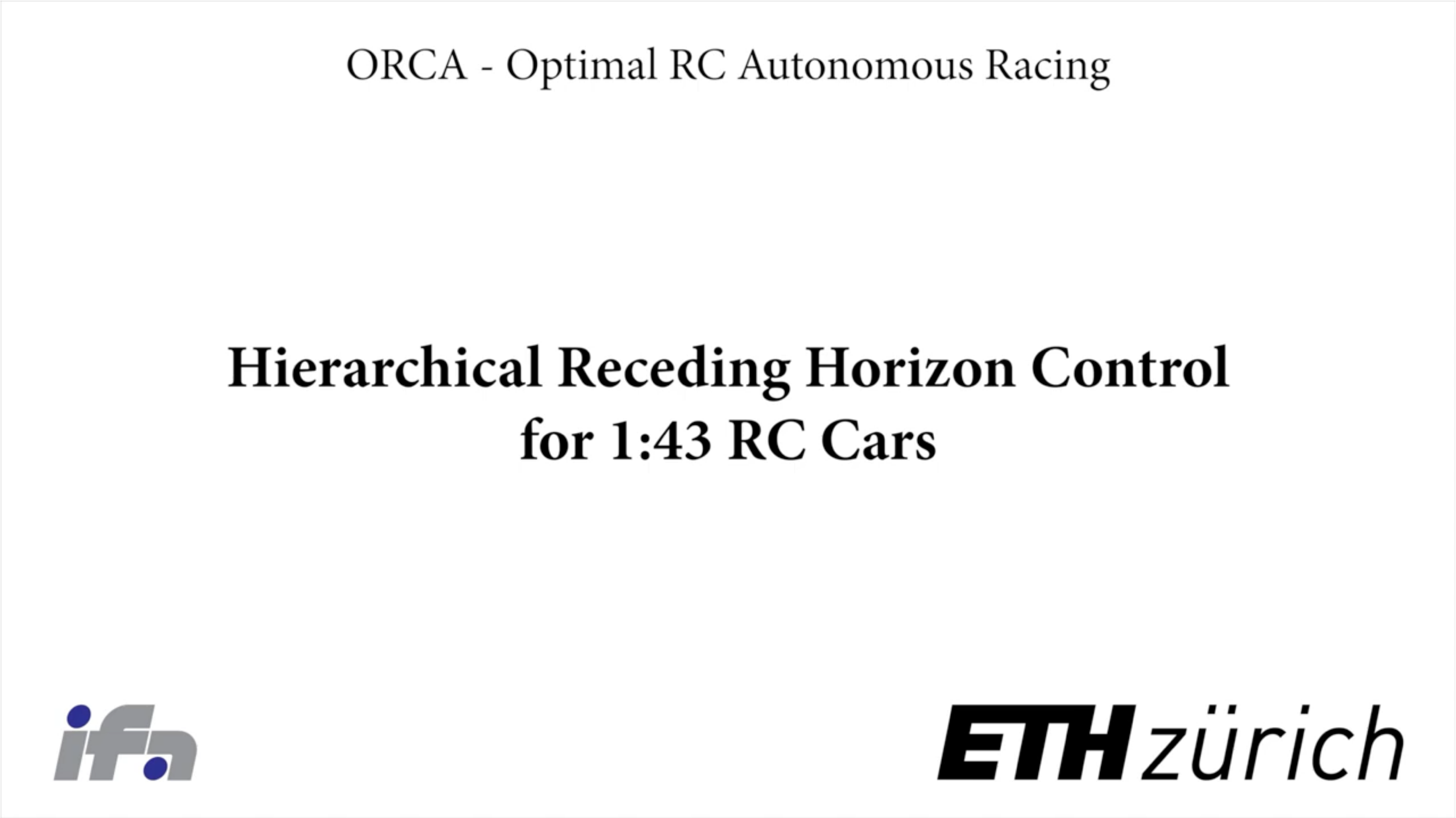
$$h(x(t), \hat{u}(t;\alpha), t) \leq 0$$

$$e(x(T)) = 0$$

Hierarchical Receding Horizon Control Simulation



Hierarchical Receding Horizon Control RC Car



Nonlinear MPC for trajectory optimization and tracking

https://youtu.be/Y7-1CBqs4x4

MPC – Drone racing

https://youtu.be/mHDQcckqdg4

Optimal and Model Predictive Control: Summary

Very general formulation! Supports:

- Expressing *preferences* for trajectories & controls (via performance index *J*)
- Complex (nonlinear) system dynamics f
- General state and control constraints
- Partially-constrained boundary conditions

But: Decision variables include *functions* $x(t)$ and $u(t)$
⇒ VERY hard to solve in general

Model Predictive Control (MPC):

Approximate optimal control strategy:

- Uses (fixed) finite planning horizon *T*
- Assume parametric family of controls $\hat{u}(\cdot;\alpha^*) : [0, T] \to \mathbb{R}^m$ w/ parameter α

Optimal control problem (OCP)

$$\min_{u,x} E\big(x(t_f), t_f\big) + \int_{t_0}^{t_f} L[x(t), u(t), t]\, dt$$

subject to: $\dot{x}(t) = f[x(t), u(t), t]$

$$h(x(t), u(t), t) \leq 0$$

$$e\big(\, x(t_f), t_f\big) = 0$$

MPC: finite-dimensional nonlinear program that can be solved very fast