Chapter 7
Control

Part 3

7.3 Optimal and Model Predictive Control

Prediction and Optimality

- Prediction enables search
  - creates the capacity to elaborate alternatives.
- Optimality
  - creates the capacity to decide what to do.

Optimal Control

- Mobile robots are intelligent (=perceptive and deliberative):
  - perceive the environment around them. } Perceptive
  - predict environmental interactions for candidate motions.
  - rank alternative actions.
  - execute a chosen action.

  } Deliberative

- The intelligent control of mobile robots is an optimal control problem

Receding Horizon MPC

- Perceptive horizon is intrinsically limited.
- So, new information arrives all the time.
- Have to keep changing the plan.
- Need models to do adequate prediction for planning.
- The intelligent control of mobile robots is a receding horizon MPC problem

Oskar Bolza

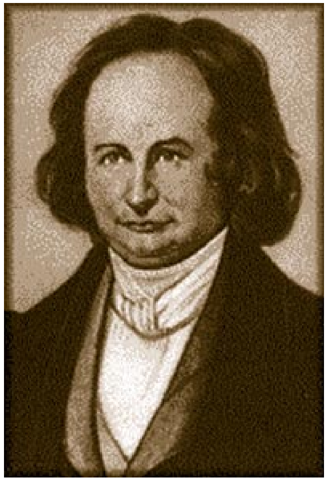
- Attended U Berlin 1875.
  - Taught by Helmholtz and Kirchoff.
  - Felt he had no talent for research.
- Attended Weierstrass's 1879 lecture on Calculus of Variations.
- Switch to Klein as advisor. Received his doctorate in 1886 after many course corrections.
- 1914: Wrote the optimal control paper on what is now called 'the problem of Bolza'.
- Thereafter left public life for 15 years in response to World War I.

Felix Klein on American professors of math of the time

"I doubt one half of them could tell what a determinant is."

Carl Jacobi

- Initially educated by an uncle
  - Who did a good job!
- Moved from first to last grade of high school in one year.
  - Qualified to enter university at age 12.
  - Had to stay in high school 4 more years til 16.
- Entered Berlin U in 1821. Joined Neumann and Bessel in 1826.
- Reputation as excellent teacher.
- Clarified the nature of the Jacobian.
- Honored in naming the Hamilton-Jacobi-Bellman equation.
- Died of smallpox around 1842.



Carl Gustav Jacob **Jacobi**
1804-1851

Lev Pontryagin

- Too poor to go to good schools.
- Blinded by an accident at age 14.
  - His mother was his devoted secretary for the rest of his life.
- Entered University of Moscow in 1925.
  - Took no notes!
  - Remembers derivations in his head!
- Appointed to Faculty of Mathematics 1934.
- Best known for the Pontryagin Maximum Principle → one of the most general theorems in all of optimization.

Lev Semenovich Pontryagin
1908-1988

Outline

- 7.3 Optimal and Model Predictive Control
  – 7.3.1 Calculus of Variations
  – 7.3.2 Optimal Control
  – 7.3.3 Model Predictive Control
  – 7.3.4 Techniques for Solving Optimal Control Problems
  – 7.3.5 Parametric Optimal Control

  Summary

7.3.1 Calculus of Variations

(Variational Optimization)

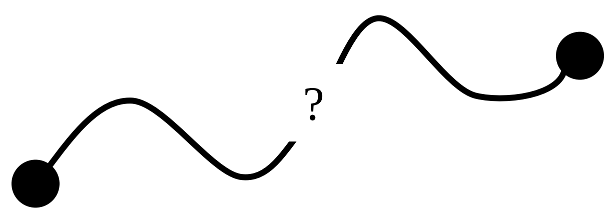
- A mathematical formulation of a quest for an unknown function.
- Replace
  - dx (a differential) with
  - $\delta x(t)$ a function of time (called a variation).

7.3.1 Calculus of Variations

- Consider this optimization problem.

minimize	$J[\underline{x}, t_f] = \phi(\underline{x}(t_f)) + \int_{t_0}^{t_f} L(\underline{x}, \dot{\underline{x}}, t)dt \quad t_f$ free
subject to:	$\underline{x}(t_0) = \underline{x}_0 \quad ; \quad \underline{x}(t_f) = \underline{x}_f$ (when $\phi(\underline{x}(t_f))$ is absent)

- E.g. shortest path between two points.



7.3.1 Calculus of Variations

- Consider this optimization problem.

minimize	$J[\underline{x}, t_f] = \phi(\underline{x}(t_f)) + \int_{t_0}^{t_f} L(\underline{x}, \dot{\underline{x}}, t)dt \quad t_f$ free
subject to:	$\underline{x}(t_0) = \underline{x}_0 \quad ; \quad \underline{x}(t_f) = \underline{x}_f$ (when $\phi(\underline{x}(t_f))$ is absent)

- $J[\underline{x},t_f]$ is a <u>functional</u> – a function of a function.
  - Square brackets notation $J[\underline{x}]$
  - J[sin(t)] = 6.2
  - J[at+b] = 12.9

7.3.1.1 Euler Lagrange Equations

$$\text{minimize} \quad J[\underline{x}, t_f] = \phi(\underline{x}(t_f)) + \int_{t_0}^{t_f} L(\underline{x}, \dot{\underline{x}}, t)dt \quad t_f \quad \text{free}$$

$$\text{subject to:} \quad \underline{x}(t_0) = \underline{x}_0 \quad ; \quad \underline{x}(t_f) = \underline{x}_f \ (\text{when } \phi(\underline{x}(t_f)) \text{ is absent})$$

(Necessary Conditions)

- Suppose a solution $\underline{x}^*(t), t_f^*$ has been found...
- Consider adding a small variation $\delta\underline{x}(t), \delta t_f$ to the solution.
- What happens to J? Substitute:

$$J[\underline{x}^* + \delta\underline{x}] = \phi(\underline{x}^*(t_f) + \delta\underline{x}(t_f)) + \int_{t_0}^{(t_f + \delta t)} L(\underline{x}^* + \delta\underline{x}, \dot{\underline{x}}^* + \delta\dot{\underline{x}}, t)dt$$

- Boundary conditions:

$$\delta\underline{x}(t_0) = \underline{0} \qquad \delta\underline{x}(t_f + \delta t) = \underline{0}$$

- Approximate L by its Taylor series:

$$L(\underline{x}^* + \delta\underline{x}, \dot{\underline{x}}^* + \delta\dot{\underline{x}}, t) \approx L(\underline{x}^*, \dot{\underline{x}}^*, t) + L_{\underline{x}}(\underline{x}^*, \dot{\underline{x}}^*, t)\delta\underline{x} + L_{\dot{\underline{x}}}(\underline{x}^*, \dot{\underline{x}}^*, t)\delta\dot{\underline{x}}$$

7.3.1.1 Euler Lagrange Equations

minimize	$J[\underline{x}, t_f] = \phi(\underline{x}(t_f)) + \int_{t_0}^{t_f} L(\underline{x}, \dot{\underline{x}}, t)dt \quad t_f \quad \text{free}$
subject to:	$\underline{x}(t_0) = \underline{x}_0 \quad ; \quad \underline{x}(t_f) = \underline{x}_f$ (when $\phi(\underline{x}(t_f))$ is absent)

(Necessary Conditions)

- Now, the perturbed objective is:

$$J[\underline{x}^* + \delta\underline{x}] = \phi(\underline{x}^*(t_f)) + \int_{t_0}^{(t_f + \delta t)} (L(.) + L_{\underline{x}}(.)\delta\underline{x} + L_{\dot{\underline{x}}}(.)\delta\dot{\underline{x}})dt$$

- Third term inside can be integrated by parts:

$$\int_{t_0}^{t_f} (L_{\dot{\underline{x}}}(.)\delta\dot{\underline{x}})dt = L_{\dot{\underline{x}}}(.)\delta\underline{x}\Big|_{t_0}^{t_f} - \int_{t_0}^{t_f} \left(\frac{\mathrm{d}}{\mathrm{d}t} L_{\dot{\underline{x}}}(.)\delta\underline{x}\right) dt$$

- Based on the boundary conditions the first part vanishes.

7.3.1.1 Euler Lagrange Equations

minimize $\quad J[\underline{x}, t_f] = \phi(\underline{x}(t_f)) + \int_{t_0}^{t_f} L(\underline{x}, \dot{\underline{x}}, t)dt \quad t_f \quad$ free

subject to: $\quad \underline{x}(t_0) = \underline{x}_0 \quad ; \quad \underline{x}(t_f) = \underline{x}_f$ (when $\phi(\underline{x}(t_f))$ is absent)

(Necessary Conditions)

- The perturbed objective is now:

$$J[\underline{x}^* + \delta\underline{x}] = \phi(\underline{x}^*(t_f)) + \int_{t_0}^{t_f} L(.)dt + \int_{t_0}^{(t_f + \delta t)} \left(L_{\underline{x}}(.) - \frac{\mathrm{d}}{\mathrm{d}t}L_{\dot{\underline{x}}}(.)\right)\delta\underline{x}\,dt$$

Ignoring H.O.T.

- This is the same as:

$$J[\underline{x}^* + \delta\underline{x}] = J[\underline{x}^*] + \int_{t_0}^{(t_f + \delta t)} \left(L_{\underline{x}}(.) - \frac{\mathrm{d}}{\mathrm{d}t}L_{\dot{\underline{x}}}(.)\right)\delta\underline{x}\,dt$$

- The integrand must vanish to first order for a local minimum.

7.3.1.1 Euler Lagrange Equations (Necessary Conditions)

minimize	$J[\underline{x}, t_f] = \phi(\underline{x}(t_f)) + \int_{t_0}^{t_f} L(\underline{x}, \dot{\underline{x}}, t)dt$	t_f free
subject to:	$\underline{x}(t_0) = \underline{x}_0 \;\; ; \;\; \underline{x}(t_f) = \underline{x}_f$ (when $\phi(\underline{x}(t_f))$ is absent)	

- This must be zero:

$$\int_{t_0}^{(t_f + \delta t)} \left(L_{\underline{x}}(.) - \frac{d}{dt} L_{\dot{\underline{x}}}(.) \right) \delta \underline{x} dt$$

- But $\delta \underline{x}(t)$ is arbitrary, so the stuff in ( ) must be zero.

$$L_{\underline{x}}(.) - \frac{d}{dt} L_{\dot{\underline{x}}}(.) = 0$$

- These are the Euler-Lagrange Equations
- Second order differential equations.
  – Solves a lot of important problems in physics.

7.3.1.2 Transversality Conditions

minimize $J[\underline{x}, t_f] = \phi(\underline{x}(t_f)) + \int_{t_0}^{t_f} L(\underline{x}, \dot{\underline{x}}, t)dt \quad t_f$ free

subject to: $\underline{x}(t_0) = \underline{x}_0 \quad ; \quad \underline{x}(t_f) = \underline{x}_f$ (when $\phi(\underline{x}(t_f))$ is absent)

- Recall

minimize $$J[\underline{x}, t_f] = \phi(\underline{x}(t_f)) + \int_{t_0}^{t_f} L(\underline{x}, \dot{\underline{x}}, t)dt \quad t_f \quad \text{free}$$

subject to: $\underline{x}(t_0) = \underline{x}_0 \quad ; \quad \underline{x}(t_f) = \underline{x}_f$ (when $\phi(\underline{x}(t_f))$ is absent)

- When t_f is free, J must be stationary with respect to it. Thus:

$$\frac{d}{dt_f} J[\underline{x}, t_f] = [\dot{\phi}(\underline{x}(t)) + L(\underline{x}, \dot{\underline{x}}, t)]_{t = t_f} = 0$$

Outline

- 7.3 Optimal and Model Predictive Control
  - 7.3.1 Calculus of Variations
  - 7.3.2 Optimal Control
  - 7.3.3 Model Predictive Control
  - 7.3.4 Techniques for Solving Optimal Control Problems
  - 7.3.5 Parametric Optimal Control

  Summary

7.3.2 Optimal Control

- Consider this optimization problem.

minimize	$J[\underline{x}, \underline{u}, t_f] = \phi(\underline{x}(t_f)) + \int_{t_0}^{t_f} L(\underline{x}, \underline{u})dt$	t_f free
subject to:	$\dot{\underline{x}} = f(\underline{x}, \underline{u})$; $\underline{u} \in U$	
	$\underline{x}(t_0) = \underline{x}_0$; $\underline{x}(t_f) = \underline{x}_f$ (when $\phi(\underline{x}(t_f))$ is absent)	

Boltza Form

- Similar to calculus of variations but with $\dot{\underline{x}}$ (n-vector) replaced by $\underline{u}$ (m-vector).
- Now, you are in charge….

7.3.2 Optimal Control

(View as Constrained Optimization over Functionals)

- Problem has two main components:
  - UTILITY: doing something useful (probably to get somewhere, maybe in some best fashion).
  - CONSTRAINT: while respecting some constraints.

7.3.2 Optimal Control

(Utility)

- In Bolza form, want to optimize some functional representing "cost" or "utility":

$$J = \phi[x(t_f)] + \int_{t_0}^{t_f} L(\underline{x}, \underline{u}, t)\,dt$$

- Where:
  - $\phi[x(t_f)]$ (endpoint cost function) may be used to represent the desire to reach some particular terminal state.
  - the integral term can be used to, for example, express the cost of driving at high curvature.

7.3.2.1 "The" Minimum Principle

Or "the" Maximum Principle

- To solve the optimal control problem, define the Hamiltonian.

$$H(\underline{\lambda}, \underline{x}, \underline{u}) = L(\underline{x}, \underline{u}) + \underline{\lambda}^T f(\underline{x}, \underline{u})$$

- Time varying $\underline{\lambda}(t)$ is known as the co-state vector. Analogous to Lagrange multipliers.
- Maximum principle states u must minimize H.

$$H(\underline{\lambda}^*, \underline{x}^*, \underline{u}^*) \le H(\underline{\lambda}^*, \underline{x}^*, \underline{u}) \quad : \quad \underline{u} \in U$$

7.3.2.1 “The” Minimum Principle

(First Order Conditions)

- Derived just like Euler Lagrange Equations:

$$\dot{\underline{x}} = \frac{\partial H}{\partial \underline{\lambda}} = f(\underline{x}, \underline{u})$$ x satisfies system dynamics

$$\dot{\underline{\lambda}}^T = -\frac{\partial H}{\partial \underline{x}} = -L_{\underline{x}}(\underline{x}, \underline{u}) - \underline{\lambda}^T f_{\underline{x}}(\underline{x}, \underline{u})$$ co-state ODE

$$\frac{\partial}{\partial \underline{u}} H(\underline{\lambda}, \underline{x}, \underline{u}) = \underline{0}$$ H is stationary wrt u

$$\underline{x}(t_0) = \underline{x}_0 \quad \underline{x}(t_f) = \underline{x}_f \quad \underline{\lambda}(t_f) = \phi_{\underline{x}}(\underline{x}(t_f))$$ boundary conditions

$$\frac{d}{dt_f} J[\underline{x}, t_f] = [\dot{\phi}(\underline{x}(t)) + \underline{\lambda}^T f(\underline{x}, \underline{u}) + L(\underline{x}, \dot{\underline{x}})]_{t = t_f} = 0$$ transversality condition

- This 2-point boundary value problem is also known as the Euler-Lagrange equations.

7.3.2.2 Dynamic Programming

- A different view of optimal control…
- Define the value function V (aka optimal return function or optimal cost to go) as the cost of the optimal path.

$$V[\underline{x}(t_0), t_0] = J^*[\underline{x}, \underline{u}] = min_{\underline{u}}\{J[\underline{x}, \underline{u}]\} = min_{\underline{u}}\left\{\phi[x(t_f)] + \int_{t_0}^{t_f} L(\underline{x}, \underline{u}, t)dt\right\}$$

(EQ 6.53)

- Given J*, the optimal control can be computed from it.

7.3.2.5 Example LQR Control

- Control a linear system to optimize a quadratic objective.
- System: $\dot{\underline{x}} = F(t)\underline{x} + G(t)\underline{u}$
- Objective:

$$J[\underline{x}, \underline{u}, t_f] = \frac{1}{2}\underline{x}^T(t_f)S_f\underline{x}(t_f) + \frac{1}{2}\int_{t_0}^{t_f}(\underline{x}^T A(t)\underline{x} + \underline{u}^T B(t)\underline{u})dt$$

- The Hamiltonian is:

$$H = \frac{1}{2}(\underline{x}^T A\underline{x} + \underline{u}^T B\underline{u}) + \underline{\lambda}^T(F\underline{x} + G\underline{u})$$

7.3.2.5 Example LQR Control

- Kalman proved that the optimal control is:

$$\underline{u} = -B^{-1}G^T\underline{\lambda} = B^{-1}G^TS\underline{x}$$

- Where S satisfies the Ricatti equation:

$$\dot{S} = SF + F^TS - SGB^{-1}G^TS + A$$

- S is easy to get by integrating backward in time from the terminal constraint

$$S(t_f) = S_f$$

Outline

- 7.3 Optimal and Model Predictive Control
  - 7.3.1 Calculus of Variations
  - 7.3.2 Optimal Control
  - 7.3.3 Model Predictive Control
  - 7.3.4 Techniques for Solving Optimal Control Problems
  - 7.3.5 Parametric Optimal Control

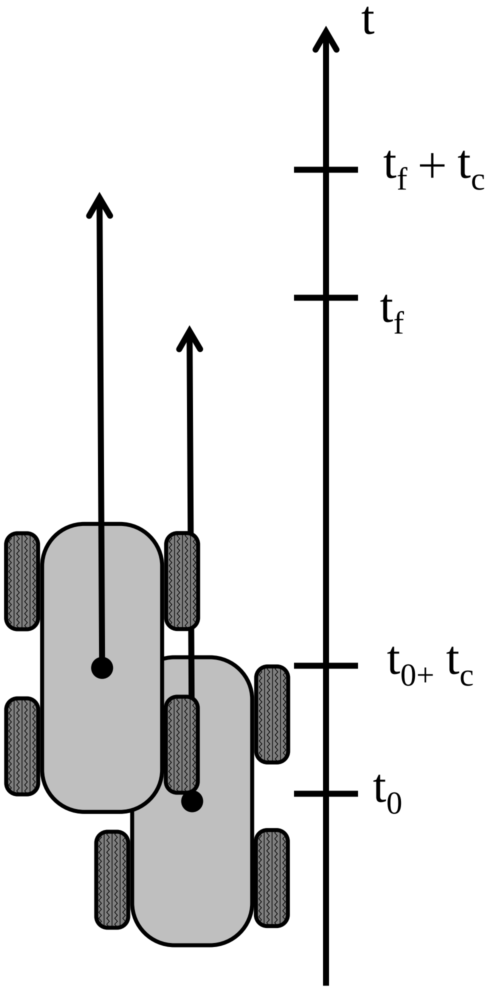
  Summary

7.3.3.1 Receding Horizon Control

- Solve the following problem for some finite prediction horizon t_f :

$$J = \phi[x(t_f)] + \int_{t_0}^{t_f} L(\underline{x}, \underline{u}, t)dt$$

- Execute the optimal control $u^*(t)$ for a control horizon t_c.
- Do it all over again for $t_0 + t_c$ and $t_f + t_c$



Issues

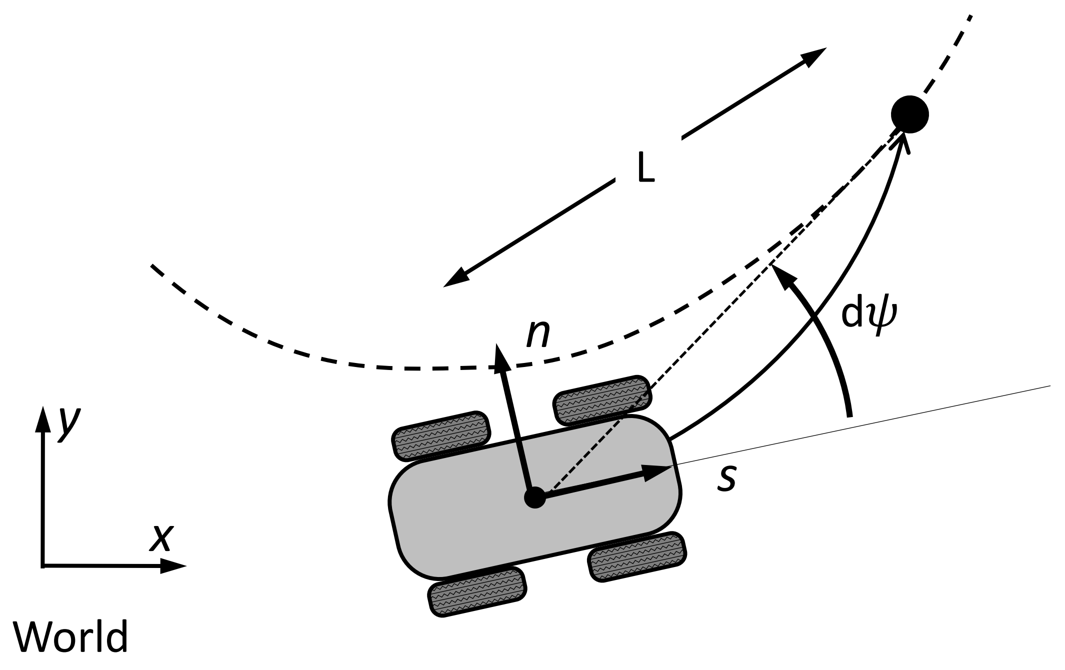
- Stability:
  - Hard to maintain stability with finite horizon.
- Feasibility:
  - Terminal state constraint $x(t_f)$ may not be satisfied.
  - Have no terminal state constraints (e.g. OA)

7.3.3.2.1 Example: Pure Pursuit

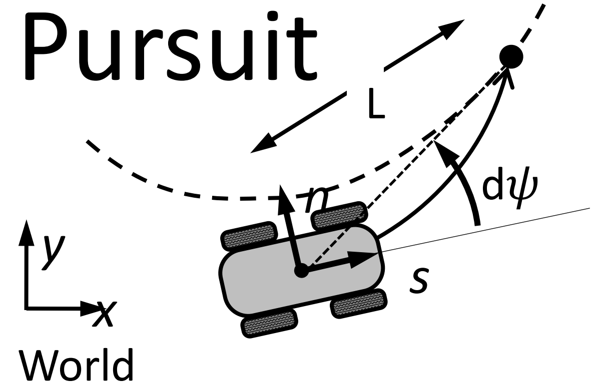
- Simplest objective is a single point t_f

$$J[\underline{x}, \underline{u}, t_f] = V(\underline{x}_f, \underline{u}) = (x(t_f) - x_f)^2 + (y(t_f) - y_f)^2$$

$$t_f = L / v$$



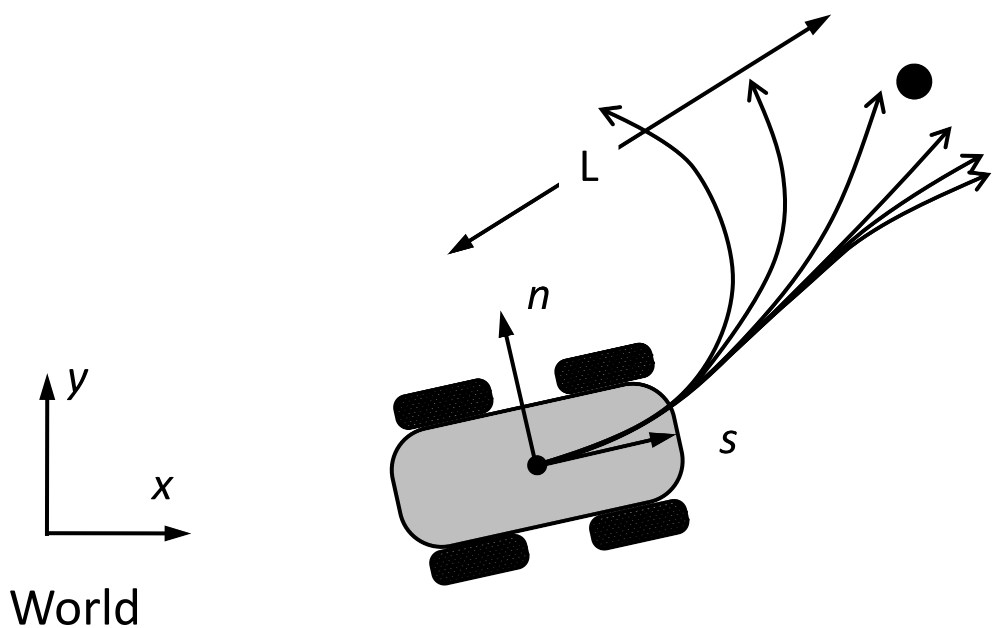
7.3.3.2.1 Example: Pure Pursuit



- Controller: $$u_{\kappa}^{*} = \delta\psi / L$$
- Inverse lookahead (1/L) distance acts like a proportional gain.
- Issues:
  - Stability depends critically on lookahead.
  - Misbehaves for infeasible paths.
  - Commands themselves are infeasible (instantaneous curvature change).

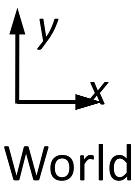
7.3.3.3 Example: Model Predictive

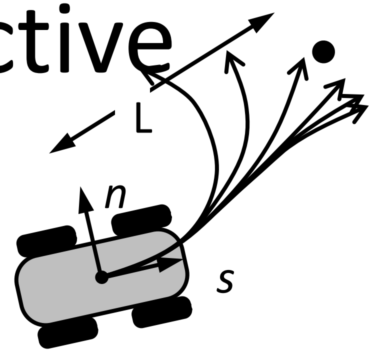
- Model steering response in terms of latency and max rates.
- Sample alternatives, simulate, and pick best.



7.3.3.3 Example: Model Predictive

- Same crosstrack objective:





$$J[\underline{x}, \underline{u}, t_f] = V(\underline{x}_f, \underline{u}) = (x(t_f) - x_f)^2 + (y(t_f) - y_f)^2$$

$$t_f = L / v$$

- Issues:
  - Like pure pursuit, does not acquire path at correct heading or curvature.
    - Guarantees errors beyond lookahead.

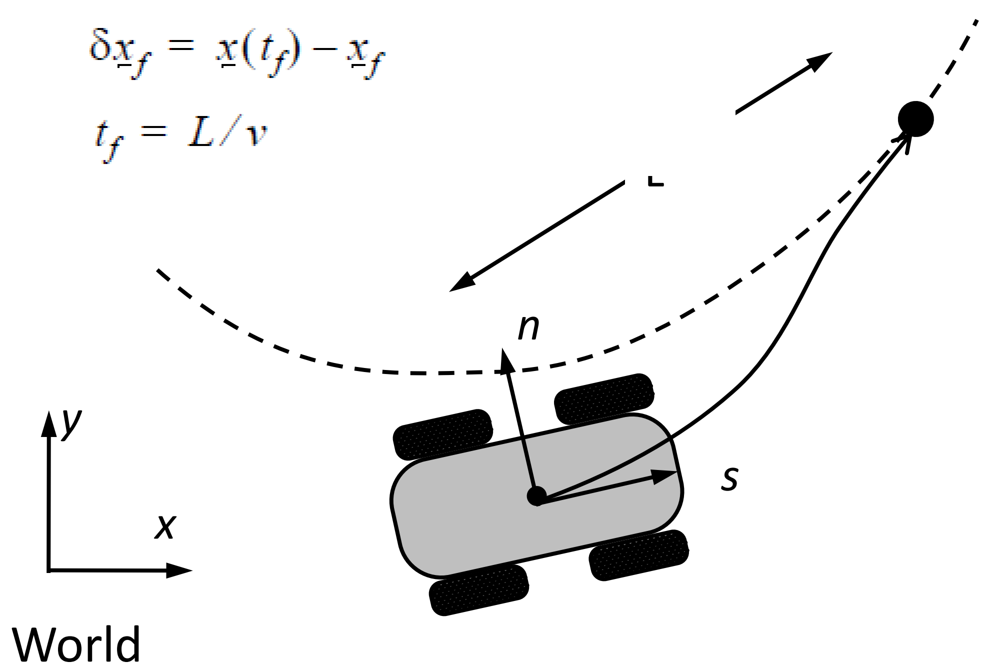
7.3.3.4 Example: Trajectory Gen

- Case 2: Minimize an endpoint cost based on total pose error.

$$J[\underline{x}, \underline{u}, t_f] = V(\underline{x}_f, \underline{u}) = \delta\underline{x}_f^T S \delta\underline{x}_f$$

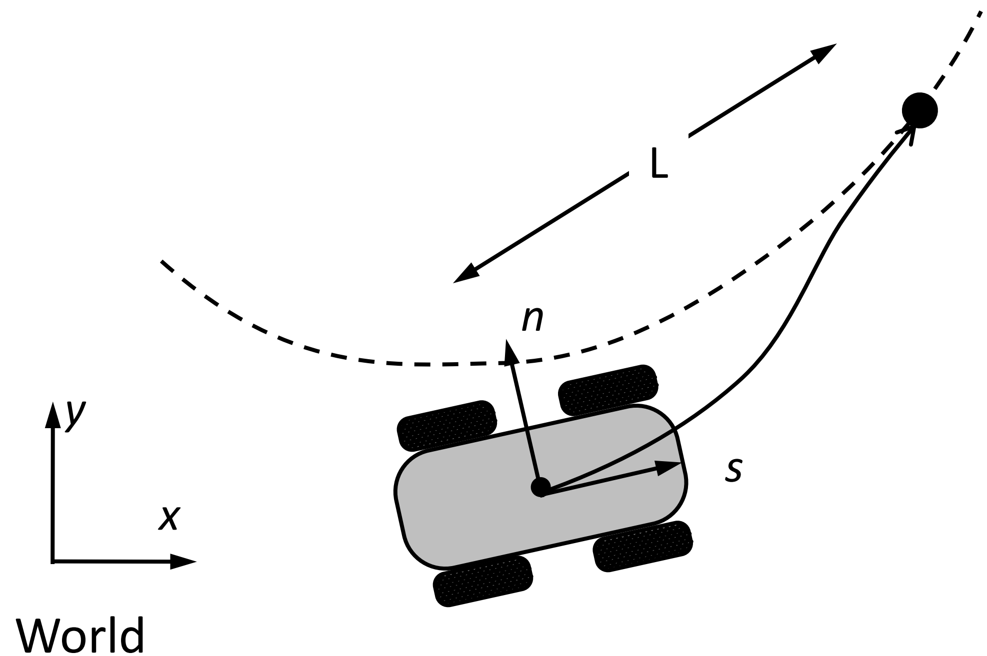
$$\delta\underline{x}_f = \underline{x}(t_f) - \underline{x}_f$$

$$t_f = L / v$$



7.3.3.4 Example: Trajectory Gen

- Case 1: Invert dynamics in an exact trajectory generator .
- Issue: Lookahead point may not be feasible.



Outline

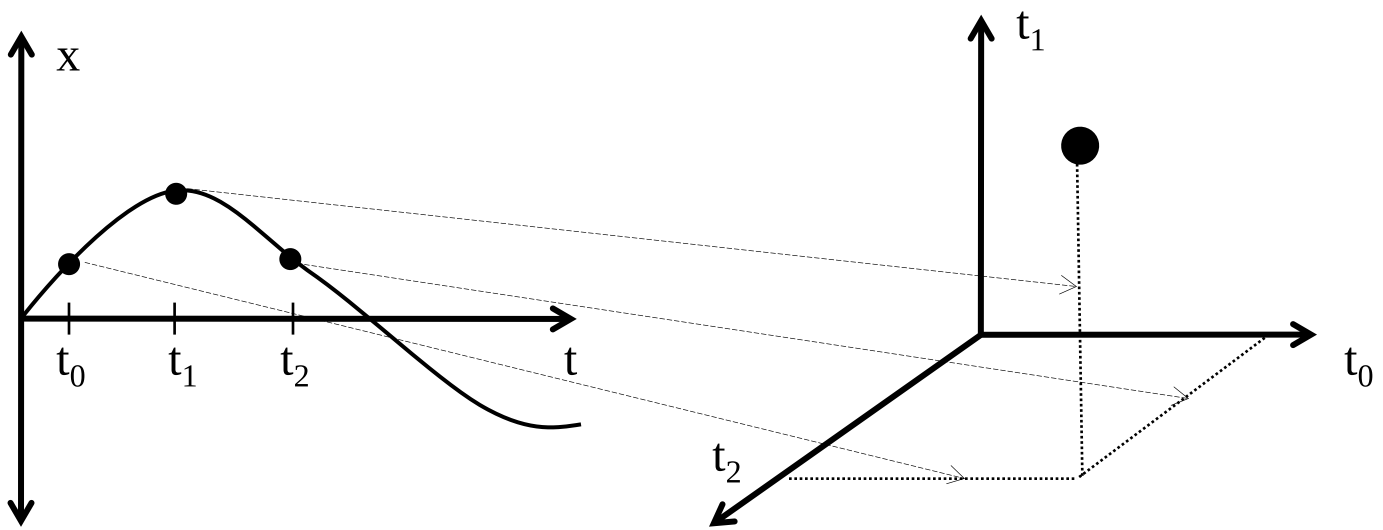
- 7.3 Optimal and Model Predictive Control
  - 7.3.1 Calculus of Variations
  - 7.3.2 Optimal Control
  - 7.3.3 Model Predictive Control
  - <u>7.3.4 Techniques for Solving Optimal Control Problems</u>
  - 7.3.5 Parametric Optimal Control

  Summary

Approaches

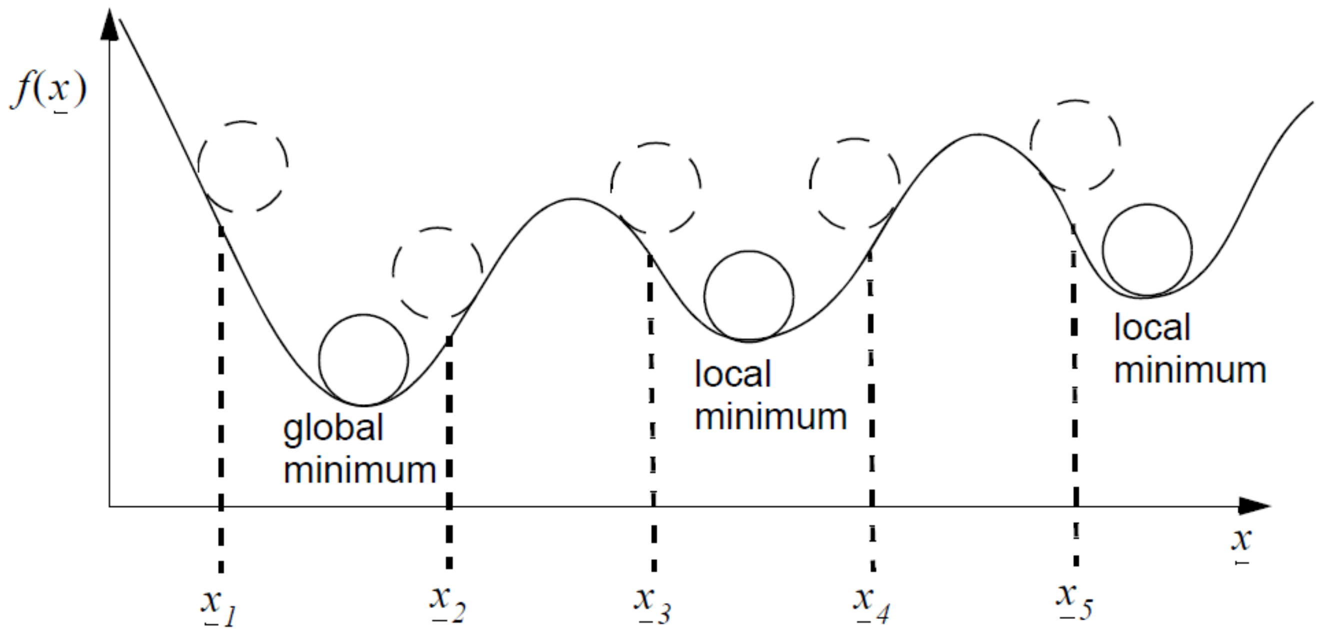
- Dynamic programming
  - Construct V and follow its gradient
  - Later: Most of motion planning is based on this.
- Direct methods
  - Minimize the objective
- Indirect methods
  - Satisfy the necessary conditions
- Parameterization
  - Next Section

7.3.4.1.1 Hilbert Space



- An arbitrary function can be thought of as a point in R^∞.
- Hence, unknown functions are like infinite parameter vectors.

7.3.4.1.2 Convexity and Sampling



- Convexity is an issue for <u>functional</u> objectives too.

7.3.4.1.3 Continuum vs Sampled Methods

- Continuum
  - (+) Solutions are arbitrarily dense.
  - (+) Gradient information exploited for efficiency.
  - (-) Local minima / need good initial guess.
- Sampled
  - (+) immune to local minima.
  - (-) less efficient
  - (-) performs poorly in high spatial frequency cost fields.
- Both
  - best of both worlds.

Solution Methods

- Dynamic programming
  - Derives a control that is valid for all initial conditions.
  - Often intractible.
- Calculus of Variations
  - Poses a boundary value problem
  - Derives control for one initial condition.
- Finite Parameterization
  - Converts problem to nonlinear programming.

7.3.4.2 Direct Methods: Finite Differences

- Discretize the dynamic model:

$$\underline{x}(k+1) = \underline{x}(k) + f(\underline{x}(k), \underline{u}(k))\Delta t$$

- Discretize the objective:

$$J = \phi(\underline{x}(n)) + \sum_{k=0}^{N-1} L(\underline{x}(k), \underline{u}(k), k)\Delta t$$

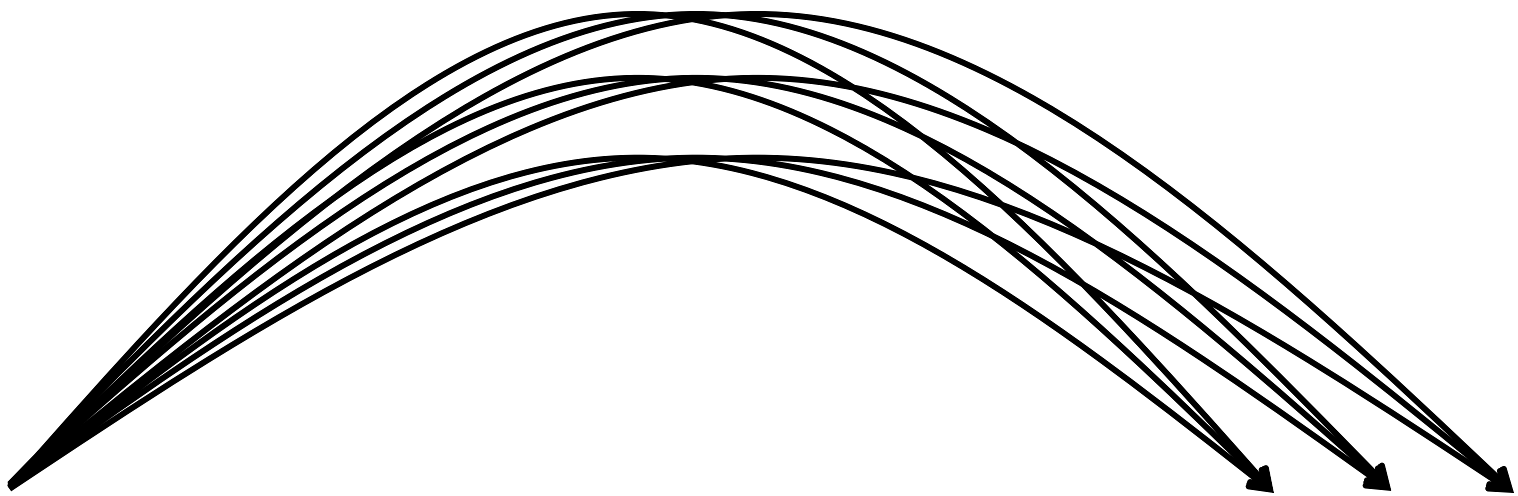
- This is a constrained optimization problem with linear constraints.
- There are Nm unknowns in $\underline{u}$( ) and Nn dof in $\underline{x}$( ) so there are N(n-m) dof left for optimization.

7.3.4.2 Direct Methods: Finite Differences

- Process:
  - Start with a guess of the inputs <u>u</u>( ) for every k.
  - Integrate the system model to determine <u>x</u>( ) for every k.
  - Compute J(<u>x</u>,<u>u</u>).
  - Compute its gradient w.r.t. <u>u</u>.
  - Line search the descent direction.
  - Repeat until convergence.

7.3.4.3 Indirect Methods: Shooting Method

- Necessary conditions are a 2 point BVP.
- Shooting: Analogous to aiming a canon by trial and error.



7.3.4.4 Penalty Function Approach

- An approach that converts a cost to a constraints:
  - (+) reduces order of problem
  - (+) simpler formulation
  - (+) great for constraints that cannot be satisfied.
  - (-) constraints no longer satisfied exactly.
- In optimal control, this means use f(xf,tf) rather than a terminal boundary condition.

Outline

- 7.3 Optimal and Model Predictive Control
  - 7.3.1 Calculus of Variations
  - 7.3.2 Optimal Control
  - 7.3.3 Model Predictive Control
  - 7.3.4 Techniques for Solving Optimal Control Problems
  - 7.3.5 Parametric Optimal Control

  Summary

7.3.5.1 Conversion to Constrained Optimization

- Ancient technique:
  - “method of undetermined coefficients”.
- Assume inputs of parameterized form:

$$\underline{u}(t) \rightarrow \tilde{\underline{u}}(\underline{p}, t)$$

- Parameters determine inputs …
- Inputs determine state …
- So we can write …

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

Any $\underline{x}$ or $\underline{u}$ (or both) can now be replaced by p.

7.3.5.1 Conversion to Constrained Optimization

- The boundary conditions become:

Integrals are suppressed notationally – but they are still there.

$$\underline{g}(\underline{p}, t_0, t_f) = \underline{x}(t_0) + \int_{t_0}^{t_f} \underline{\tilde{f}}(\underline{p}, t)dt = \underline{x}_b$$

- This is conventionally written as:

$$\underline{c}(\underline{p}, t_0, t_f) = \underline{g}(\underline{p}, t_0, t_f) - \underline{x}_b = 0$$

- The performance index is now:

$$\tilde{J}(\underline{p}, t_f) = \tilde{\phi}(\underline{p}, t_f) + \int_{t_0}^{t_f} \tilde{L}(\underline{p}, t)dt$$

7.3.5.1 Conversion to Constrained Optimization

- Minimize:

$$\tilde{J}(\underline{p}, t_f) = \tilde{\phi}(\underline{p}, t_f) + \int_{t_0}^{t_f} \tilde{L}(\underline{p}, t)dt \quad t_f \quad \text{free}$$

- Subject to:

$$\underline{c}(\underline{p}, t_0, t_f) = 0$$

7.3.5.2 First Order Response to Parameter Variation

- Practical solutions to most nonlinear problems involve linearization.
- How do we linearize an integral of a differential equation with respect to some parameters in the inputs?
  - Use Leibnitz' rule:
- Recall first, a property of partial derivatives:

$$\frac{\partial}{\partial \underline{p}}(\dot{\underline{x}}) = \frac{\partial}{\partial \underline{p}}\left(\frac{\partial \underline{x}}{\partial t}\right) = \frac{\partial}{\partial t}\left(\frac{\partial \underline{x}}{\partial \underline{p}}\right)$$

Parameter Jacobian Of Time Derivative == Time Derivative Of Parameter Jacobian

7.3.5.2 First Order Response to Parameter Variation

$$\frac{\partial}{\partial \underline{p}}(\dot{\underline{x}}) = \frac{\partial}{\partial \underline{p}}\left(\frac{\partial \underline{x}}{\partial t}\right) = \frac{\partial}{\partial t}\left(\frac{\partial \underline{x}}{\partial \underline{p}}\right)$$

- Use last result to linearize the dynamics:

$$F = \frac{\partial \dot{\underline{x}}}{\partial \underline{x}} = \frac{\partial \dot{f}}{\partial \underline{x}} \qquad G = \frac{\partial \dot{\underline{\underline{x}}}}{\partial \underline{u}} = \frac{\partial \dot{f}}{\partial \underline{u}}$$

- Hence, Jacobian of the dynamics wrt the parameters can be had by integrating an auxiliary differential equation.

$$\frac{\partial \underline{x}}{\partial \underline{p}} = \int_{t_0}^{t_f} \left[F(\underline{p}, t)\frac{\partial \underline{x}}{\partial \underline{p}} + G(\underline{p}, t)\frac{\partial \underline{u}}{\partial \underline{p}} \right] dt$$

Was that progress? Replace one DE with two?

7.3.5.2 First Order Response to Parameter Variation

$$\frac{\partial}{\partial \underline{p}}(\dot{\underline{x}}) = \frac{\partial}{\partial \underline{p}}\left(\frac{\partial \underline{x}}{\partial t}\right) = \frac{\partial}{\partial t}\left(\frac{\partial \underline{x}}{\partial \underline{p}}\right)$$

- Finally the performance index can be differentiated to enable parameter search.

$$\frac{\partial}{\partial \underline{p}}\tilde{J}(\underline{p}) = \frac{\partial}{\partial \underline{p}}\tilde{\phi}(\underline{p}, t_f) + \int_{t_0}^{t_f}\left\{\frac{\partial}{\partial \underline{x}}L(\underline{p}, t)\frac{\partial \underline{x}}{\partial \underline{p}} + \frac{\partial}{\partial \underline{u}}L(\underline{p}, t)\frac{\partial \underline{u}}{\partial \underline{p}}\right\}dt$$

- These results are different forms of Leibnitz rule - the derivative of the integral is the integral of the derivative (unless the two variables involved are the same).

Parametric Optimal Control – For Steering

- Change to distance with: $\underline{q} = [\underline{p}^T, s_f]^T$
- Minimize:

$$\text{minimize:}_{\underline{q}} \quad J(\underline{q}) = \phi(\underline{q}) + \int_0^{s_f} L(\underline{q})\,ds$$

- Subject to:

$$\text{subject to:} \quad \underline{c}(\underline{q}) = 0 \quad ; \quad s_f \quad \textit{free}$$

- This is now a problem in nonlinear programming, also called constrained optimization.
- Such problems can be solved using the technique of Lagrange Multipliers → next.

7.3.5.3 Necessary Conditions

- Define the Hamiltonian (aka Lagrangian):

$$H(\underline{q}, \underline{\lambda}) = J(\underline{q}) + \underline{\lambda}^T \underline{c}(\underline{q})$$

- The necessary conditions for a constrained optimum are:

$$\frac{\partial}{\partial \underline{q}} H(\underline{q}, \underline{\lambda}) = \frac{\partial}{\partial \underline{q}} J(\underline{q}) + \underline{\lambda}^T \frac{\partial}{\partial \underline{q}} \underline{c}(\underline{q}) = \underline{0}^T \qquad \text{p + 1 equations}$$

$$\frac{\partial}{\partial \underline{\lambda}} H(\underline{q}) = \underline{c}(\underline{q}) = \underline{0} \qquad \text{n equations}$$

- A set of n+p+1 equations in as many unknowns.

7.3.5.3 Necessary Conditions
(Newton's Method)

- Three steps:
  - Transpose first set of equations.
  - Linearize about a point where they are not satisfied.
  - Insist on satisfaction to first order after perturbation.

$$\begin{bmatrix} \frac{\partial^2 H}{\partial \underline{q}^2}(\underline{q}, \underline{\lambda}) & \frac{\partial}{\partial \underline{q}} \underline{g}(\underline{q})^T \\ \frac{\partial}{\partial \underline{q}} \underline{g}(\underline{q}) & 0 \end{bmatrix} \begin{bmatrix} \Delta \underline{q} \\ \Delta \underline{\lambda} \end{bmatrix} = \begin{bmatrix} -\frac{\partial}{\partial \underline{q}} H(\underline{q}, \underline{\lambda})^T \\ -\underline{g}(\underline{q}) \end{bmatrix}$$

- Each iteration produces a descent direction for line search. Iterate and update $\underline{q}$ and $\underline{\lambda}$ until convergence.
- No constraints → use only 1st set
- No performance index → use only 2nd set.

7.3.5.4 Parametric Optimal Control

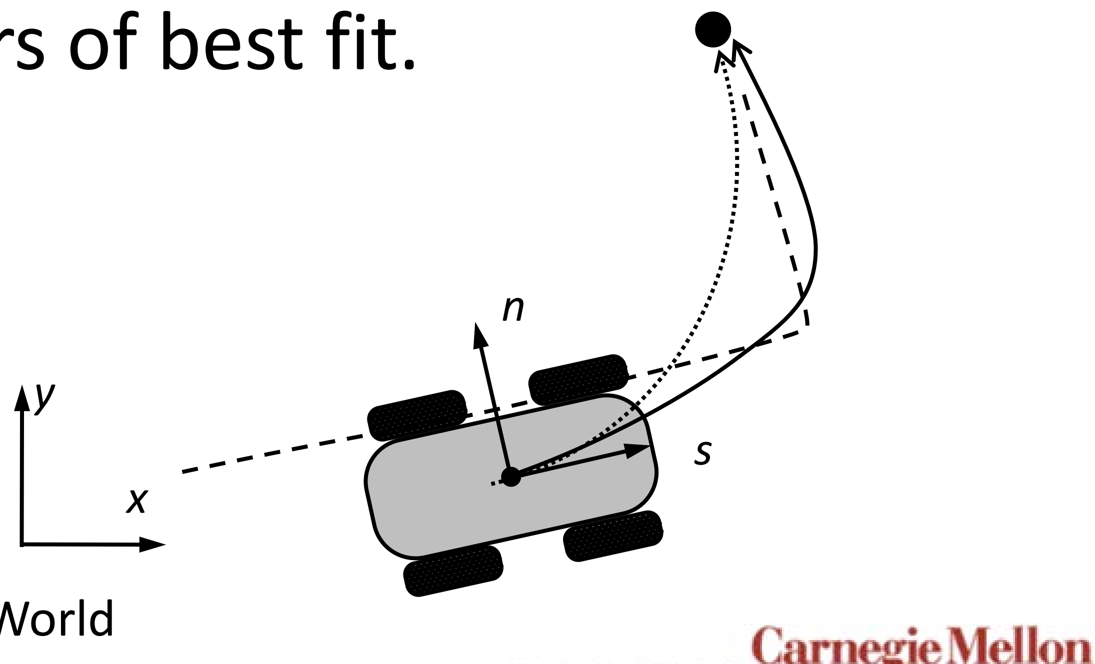
- Use full state error along the entire path:

$$J[\underline{x}, \underline{u}, t_f] = \delta\underline{x}_f^T S \delta\underline{x}_f + \int_{t_0}^{t_f} \delta\underline{x}^T(t) A \delta\underline{x}(t) dt$$

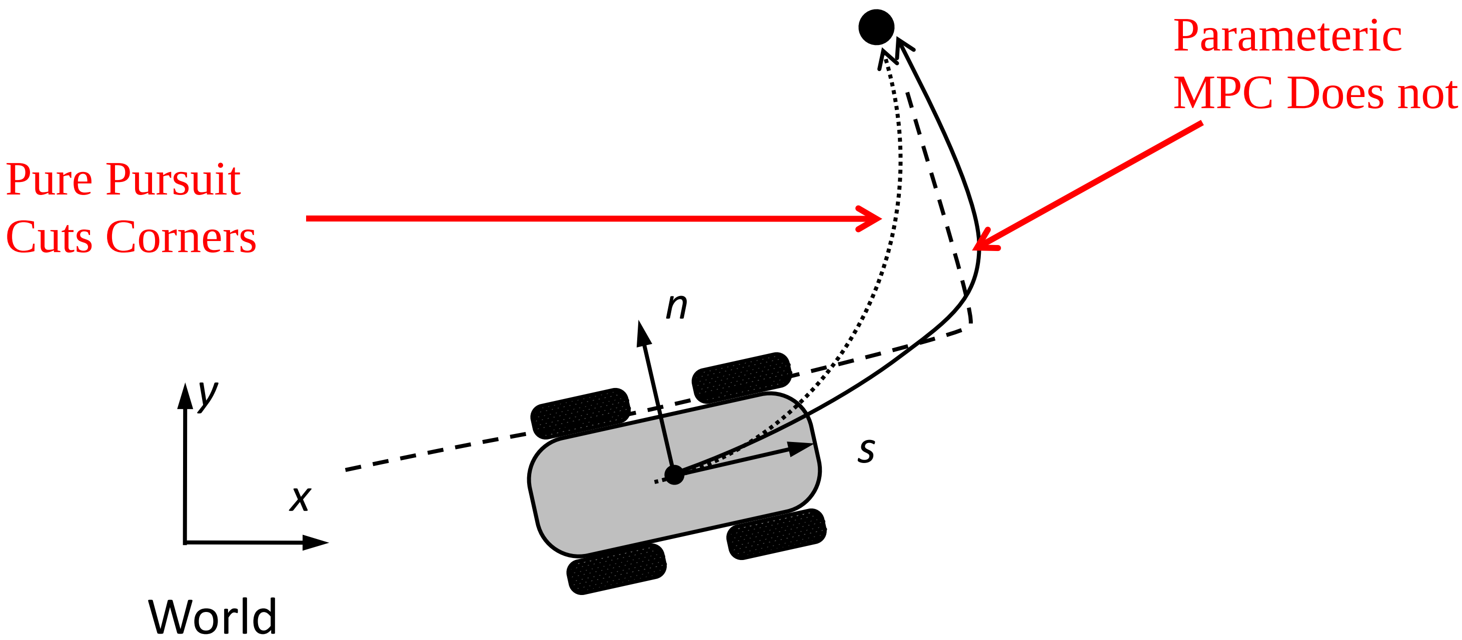
$$\delta\underline{x}(t) = \underline{x}(t) - \underline{x}_{path}(t)$$

$$t_f = L/v$$

- Solve for parameters of best fit.



7.3.5.4 Parametric Optimal Control



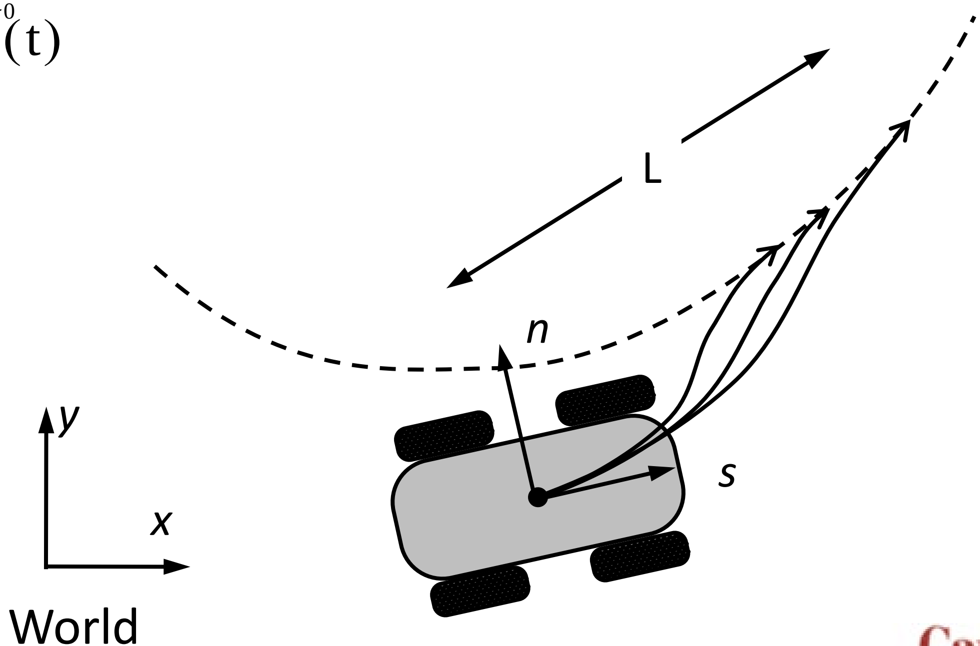
- It is straightforward to add predictive velocity control too.

7.3.5.5 Example: Adaptive Horizon

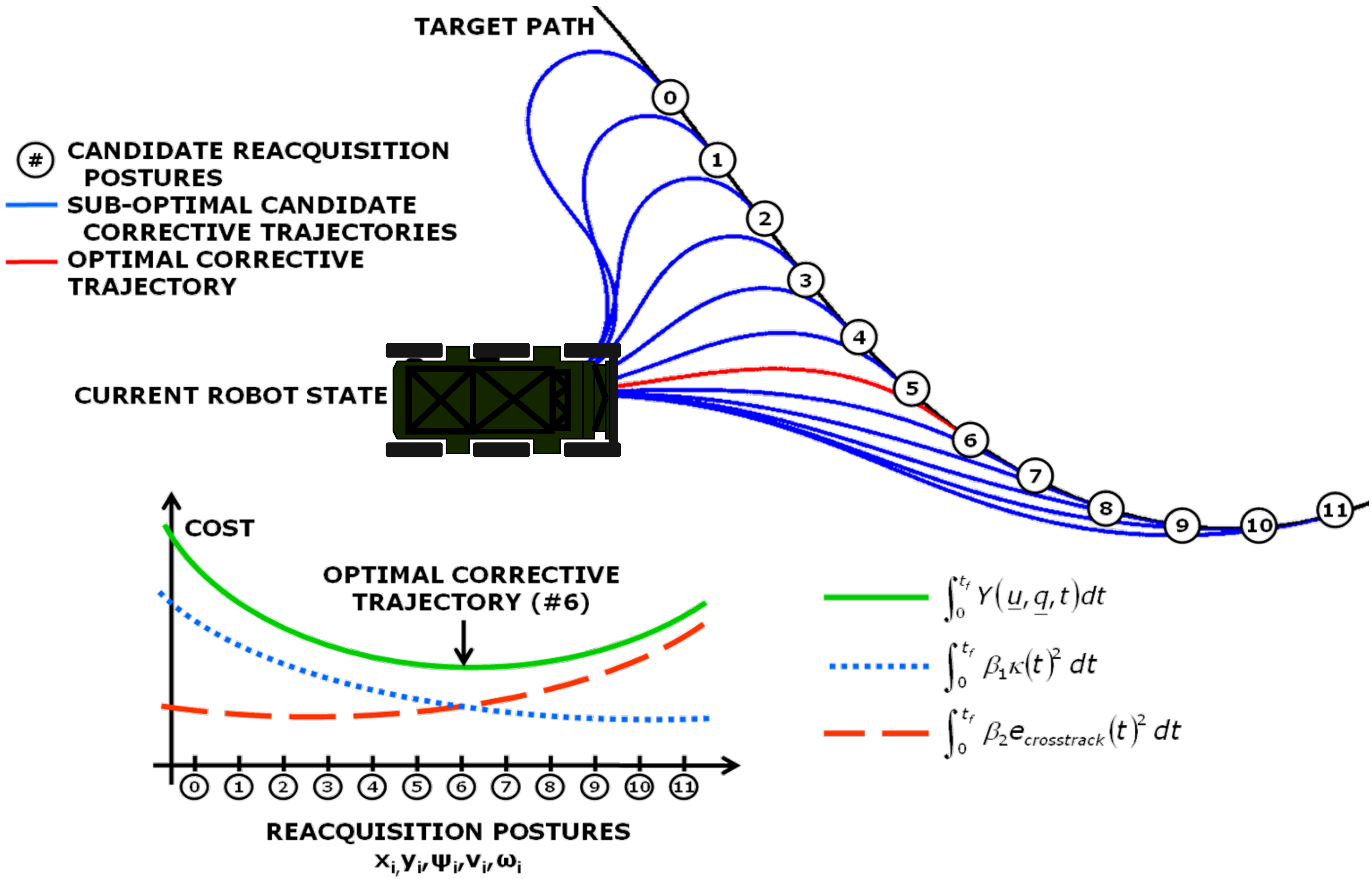
- Terminal time t_f is free.
- Objective includes both control effort and crosstrack error.

$$J[\underline{x},\underline{u},t_f] = \delta\underline{x}_f^T S \delta\underline{x}_f + \int_{t_0}^{t_f} (\delta\underline{x}^T(t) A \delta\underline{x}(t) + \underline{u}^T(t) B \underline{u}(t)) dt \qquad t_f = free$$

$$\delta\underline{x}(t) = \underline{x}(t) - \underline{x}_{path}(t)$$



7.3.5.5 Example: Adaptive Horizon



Video

CONSTRAINED OPTIMIZATION PATH TRACKING EXPERIMENTS
JUNE 28TH, 2006
Simulation View
Carnegie Mellon
THE ROBOTICS INSTITUTE

7.3.5.6 Intricate Path Following

- Plan right through the velocity reversals.

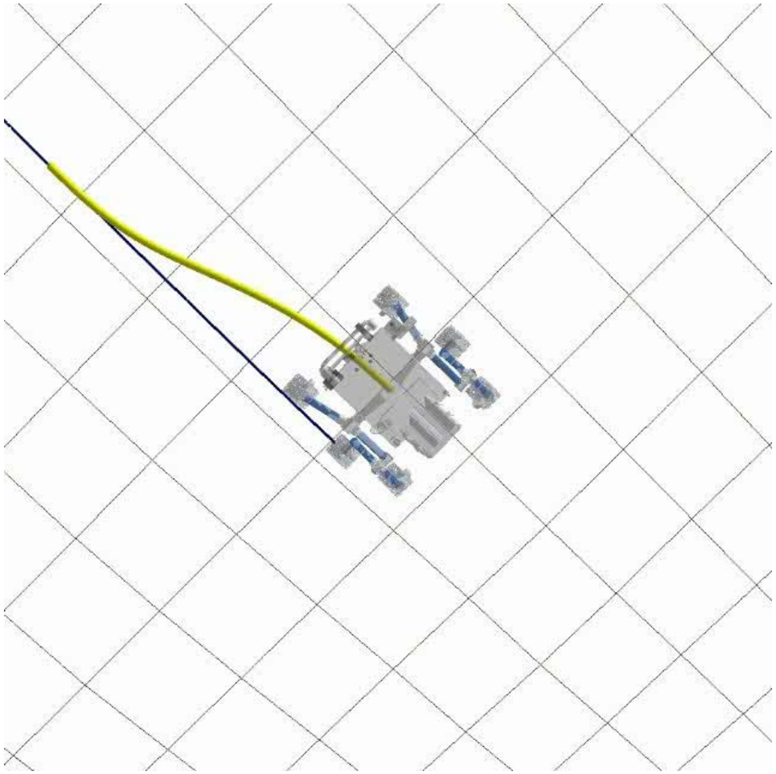
$$\underline{x}(t) = \underline{x}(t_0) + \int_0^{t_1} f(\underline{x}, \underline{u}(\underline{p}_1, t))dt \qquad (t_0 < t < t_1)$$

$$\underline{x}(t) = \underline{x}(t_1) + \int_{t_1}^{t_2} f(\underline{x}, \underline{u}(\underline{p}_2, t))dt \qquad (t_1 < t < t_2)$$

$$\underline{x}(t) = \underline{x}(t_2) + \int_{t_2}^{t_3} f(\underline{x}, \underline{u}(\underline{p}_3, t))dt \qquad (t_2 < t < t_3)$$

7.3.5.6 Intricate Path Following

- Receding Horizon Model Predictive Control (RHMPC) controller here.



Outline

- 7.3 Optimal and Model Predictive Control
  - 7.3.1 Calculus of Variations
  - 7.3.2 Optimal Control
  - 7.3.3 Model Predictive Control
  - 7.3.4 Techniques for Solving Optimal Control Problems
  - 7.3.5 Parametric Optimal Control

  Summary

Summary

- Optimal Control is a generalization of the Calculus of Variations which expresses the mobile robot control problem well.
- Trajectory generation fits very nicely into the standard form of an optimal control problem.
- Curvature polynomials of arbitrary order are a convenient representation of trajectories.
  - Cubic ones have just enough degrees of freedom to achieve an arbitrary terminal posture.
  - There is a pretty painless way to compute these.