Geometric Discretization and Motion Planning of Nonholonomic Systems with Symmetries

Marin Kobilarov,
California Institute of Technology

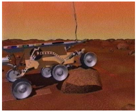
December 16, 2008

A robotic aerial vehicle example

Autonomous helicopter flying among buildings

Motivation: autonomous vehicles in natural environments

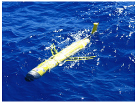
DARPA Challenges



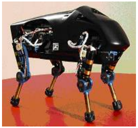
JPL Rover

BigDog

USC RESL Boat

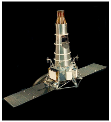


SLOCUM glider

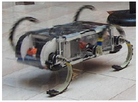


LittleDog

USC RESL Heli



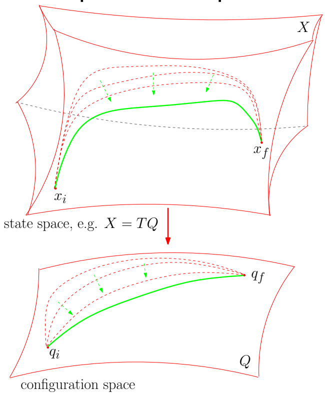
Satellite



RHex Robot

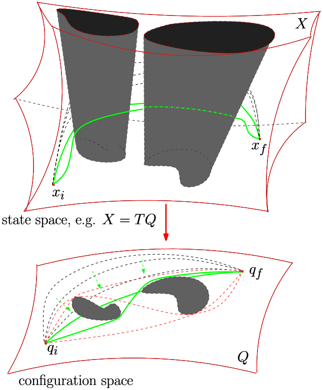
A more abstract view

Optimizing a trajectory in a complex state space



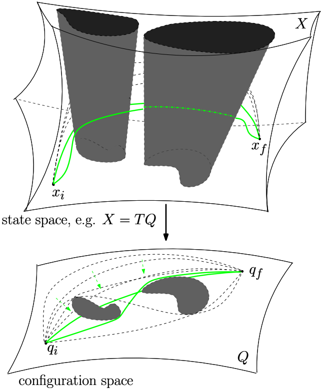
A more abstract view

Optimizing a trajectory in a complex state space



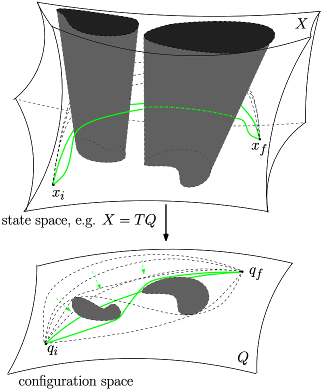
A more abstract view

Optimizing a trajectory in a complex state space



A more abstract view

Optimizing a trajectory in a complex state space

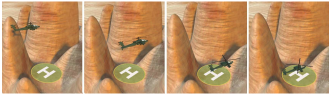


Key Points

- Trajectory numerical representation: accuracy and efficiency
  - geometric discretization
  - variational integrators
- Optimal control
  - discrete necessary conditions
  - local optimality
- Global solution among multiple homotopy classes
  - global state-space exploration
  - optimal motion primitives
  - dynamic programming

Framework for integration and control of vehicles

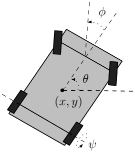
- Preview of some results: examples of computed motions



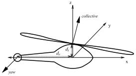
Helicopter - optimal landing

Multiple vehicles in an urban canyon

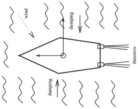
- Example developed models:



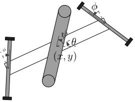
Car



Helicopter



Boat



Snakeboard

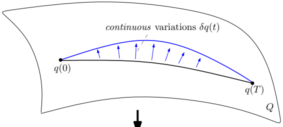
Outline

Outline

Outline

Continuous vs. Discrete Mechanics

Continuous Mechanics



continuous variational principle

$$\delta \int_0^T L(q,\dot{q})dt + \int_0^T f \cdot \delta q dt = 0$$

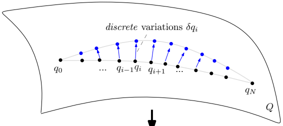
Euler-Lagrange equations

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = f$$

discretization

finite differences
stadard ODE integrators

Discrete Mechanics



discrete variational principle

$$\delta \sum_{k=0}^{N-1} L_d(q_k, q_{k+1}) + \sum_{k=0}^{N} f_k \cdot \delta q_k dt = 0$$

$$L_d(q_k, q_{k+1}) = hL\left(\frac{q_k + q_{k+1}}{2}\ \frac{q_{k+1} - q_k}{h}\right)$$

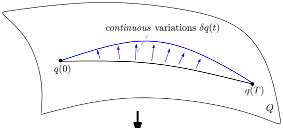
discrete Euler-Lagrange equations

$$\frac{\partial L_d}{\partial q_k}(q_{k-1}, q_k) - \frac{\partial L_d}{\partial q_k}(q_k, q_{k+1}) = h^2 f_k$$

variational integrator
used directly for computation

Continuous vs. Discrete Mechanics

Continuous Mechanics



continuous variational principle

$$\delta \int_0^T L(q,\dot q)dt + \int_0^T f\cdot \delta q dt = 0$$

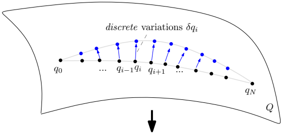
Euler-Lagrange equations

$$\frac{d}{dt}\frac{\partial L}{\partial \dot q} - \frac{\partial L}{\partial q} = f$$

discretization

finite differences
stadard ODE integrators

Discrete Mechanics



discrete variational principle

$$\delta \sum_{k=0}^{N-1} L_d(q_k,q_{k+1}) + \sum_{k=0}^{N} f_k \cdot \delta q_k dt = 0$$

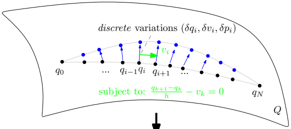
$$L_d(q_k,q_{k+1}) = hL(\tfrac{q_k+q_{k+1}}{2}\, \tfrac{q_{k+1}-q_k}{h})$$

discrete Euler-Lagrange equations

$$\frac{\partial L_d}{\partial q_k}(q_{k-1},q_k) - \frac{\partial L_d}{\partial q_k}(q_k,q_{k+1}) = h^2 f_k$$

variational integrator
used directly for computation

Discrete Mechanics
(Pontryagin-D'Alembert)



discrete variational principle

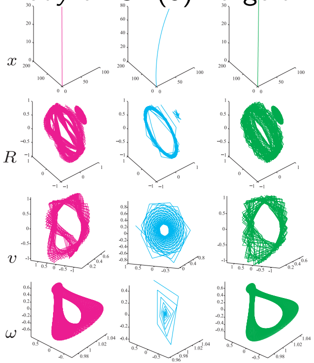
$$\delta \sum_{k=0}^{N-1} L(q_k,v_k) + \left\langle p_k, \frac{q_{k+1}-q_k}{h} - v_k \right\rangle + \sum_{k=0}^{N} f_k\cdot \delta q_k dt = 0$$

discrete Euler-Lagrange equations

$q_{k+1} = q_k + hv_k$	reconstruction
$p_k = \frac{\partial L}{\partial v}(q_k,v_k)$	Legendre transform
$p_k - p_{k-1} = h\frac{\partial L}{\partial q}(q_k,v_k) + hf_k$	

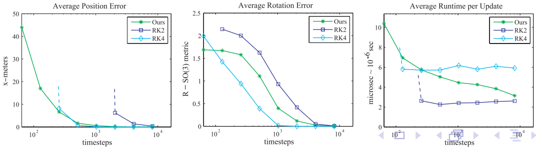
Superior numerics of discrete geometric integrators

- Orbits of the *rigid Body* on *SE*(3)- large time-steps



Ground truth   Runge-Kutta 4   variational integrator

- Accuracy and efficiency vs. resolution



Nonholonimic Integrators

- Comparisons: Accuracy and efficiency vs. resolution

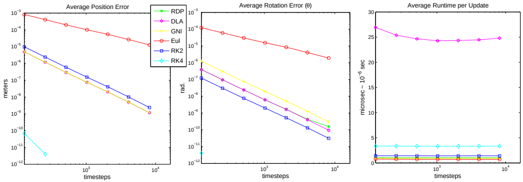


Figure: Snakeboard 10 second trajectories

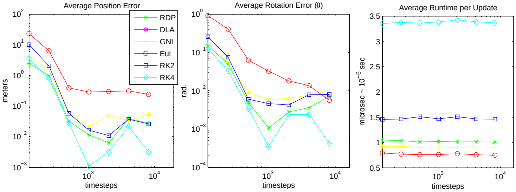


Figure: Snakeboard 10 minute trajectories

- *DLA*: Discrete Lagrange-d'Alembert (Cortez, 2002)
- *GNI*: Geometric Nonholonomic Integrator (Ferraro, Iglesias, De Diego, 2007)
- *RDP*: Reduced d'Alembert-Pontryagin (Kobilarov, 2007)

A typical system setup

- configuration space $Q = M \times G$, configuration $q \in Q$
- M – *shape space*, e.g. joint angles
- G – *Lie group*, e.g. $SE(3)$ denoting the system *pose*
- *nonholonomic constraints* $\dot{q} \in \mathcal{D}$, distribution $\mathcal{D} \subset TQ$
- *symmetries* associated with group transformations
- *external forces*, e.g. gravity, friction

State Space Structure

Principle bundle $\pi : Q \to Q/G$; distribution $\mathcal{D}_q \subset T_qQ$, $q \in Q$.

$$\mathcal{V}_q = T_q \,\mathrm{Orb}(q), \quad \mathcal{S}_q = \mathcal{D}_q \cap \mathcal{V}_q, \quad \mathcal{D}_q = \mathcal{S}_q \oplus \mathcal{H}_q.$$

- $\mathcal{V}_q$: space of tangent vectors parallel to symmetry directions, i.e. the *vertical space*
- $\mathcal{S}_q$: space of symmetry directions that satisfy the constraints (generated by $\mathfrak{s}_q = \{\xi \in \mathfrak{g} \mid \xi_Q(q) \in \mathcal{S}_q\} \subset T_qQ/G$)
- $\mathcal{H}_q$: space of tangent vectors that satisfy the constraints but are not aligned with any directions of symmetry, i.e. the *horizontal space*

Nonholonomic Connection

A principle connection $\mathcal{A} : TQ \to \mathfrak{g}$ with horizontal distribution $\mathcal{H}_q$.

Nonholonomic Connection (*Bloch* 2003; *Cendra, Marsden*, 2001)

Constructed as $\mathcal{A} = \mathcal{A}^{\mathrm{kin}} + \mathcal{A}^{\mathrm{sym}}$,
$\mathcal{A}^{\mathrm{kin}}$ is the kinematic, $\mathcal{A}^{\mathrm{sym}}$ is the mechanical connection

$$g^{-1}\dot{g} + \mathcal{A}(r)\dot{r} = \Omega,$$

defining vertical and horizontal velocity components

$$\dot{q} = \mathrm{ver}_r\, \dot{q} + \mathrm{hor}_r\, \dot{q} \Leftrightarrow (\dot{r}, g^{-1}\dot{g})_r = (0, \Omega) + (\dot{r}, -\mathcal{A}(r)\dot{r}),$$

where $\Omega \in \mathfrak{s}_r$ is the *locked angular velocity*.

Vertical Variations $(\delta r, \delta g)$

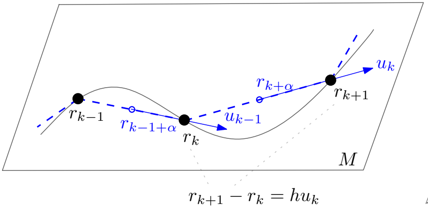
Variations such that $\delta r = 0$ and $\delta g g^{-1} = \mathcal{A}(r, g) \cdot (\delta r, \delta g) \in \mathfrak{s}_r$

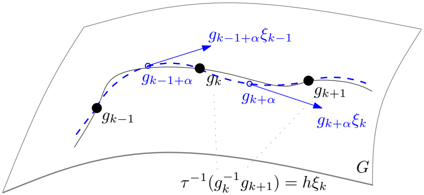
Horizontal Variations $(\delta r, \delta g)$

Variations such that $\mathcal{A}(r, g) \cdot (\delta r, \delta g) = 0$, or $(\delta r, g^{-1}\delta g) = (\delta r, -A(r)\delta r) \in (TM \times \mathfrak{g})_r$

Discrete Trajectory

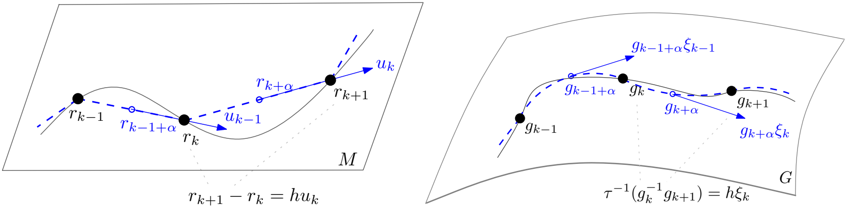
Pick coordinates $(r, g) \in M \times G$





Discrete Trajectory

Pick coordinates $(r, g) \in M \times G$



Discrete path
$(r, u, p, g, \Omega, \mu)_d : \{t_k\}_{k=0}^N \longrightarrow (TM \oplus T^*M) \times G \times \mathfrak{s} \times \mathfrak{g}^*$
subject to the constraints

$$r_{k+1} - r_k = hu_k, \qquad \tau^{-1}(g_k^{-1}g_{k+1}) = h\xi_k,$$

where $\xi_k = \Omega_k - \mathcal{A}(r_{k+\alpha})u_k$, with $r_{k+\alpha} := (1-\alpha)r_k + \alpha r_{k+1}$ for a chosen $\alpha \in [0, 1]$ and the map $\tau : \mathfrak{g} \to G$ represents the *difference* between two configurations in the group

Lagrange-D'Alembert-Pontryagin Nonholonomic Principle

Discrete Reduced LDAP Principle

Denoting $\xi_k := \Omega_k - \mathcal{A}(r_{k+\alpha})u_k$:

$$\delta \sum_{k=0}^{N-1} h\,[\ell(r_{k+\alpha}, u_k, \xi_k) + \langle p_k, (r_{k+1} - r_k)/h - u_k\rangle$$

$$+\langle \mu_k, \tau^{-1}(g_k^{-1} g_{k+1})/h - \xi_k\rangle] + \sum_{k=0}^{N-1} [h\langle f_{k+\alpha}, \delta r_{k+\alpha}\rangle] = 0,$$

subject to:

vertical variations $(\delta r_k, g_k^{-1}\delta g_k) = (0, \eta_k), \eta_k \in \mathfrak{s}_{r_k}$

horizontal variations $(\delta r_k, g_k^{-1}\delta g_k) = (\delta r_k, -\mathcal{A}(r_k)\delta r_k)$,

$\ell(r, \dot{r}, \xi) = L(r, \dot{r}, e, g^{-1}\dot{g})$: the *reduced Lagrangian*

Discrete Equations of Motion

$$
\begin{aligned}
&g_k^{-1} g_{k+1} = \tau(h(\Omega_k - \mathcal{A}(r_{k+\alpha})u_k)), \\
&r_{k+1} - r_k = h u_k, \\
&\mu_k = \frac{\partial \ell_{k+\alpha}}{\partial \Omega}, \\
&\langle \mathcal{DEP}_\tau(k), e_b(r_k) \rangle = 0, \\
&\left(\frac{\partial \ell_{k+\alpha}}{\partial u} - \frac{\partial \ell_{k-1+\alpha}}{\partial u} \right) - h \left(\alpha \frac{\partial \ell_{k-1+\alpha}}{\partial r} + (1-\alpha) \frac{\partial \ell_{k+\alpha}}{\partial r} \right) \\
&= \mathcal{A}(r_k)^* \, \mathcal{DEP}_\tau(k) + h \left(\alpha f_{k-1+\alpha} + (1-\alpha) f_{k+\alpha} \right),
\end{aligned}
$$

where the *discrete Euler-Poincaré* operator $\mathcal{DEP}_\tau$ is defined as

$$
\mathcal{DEP}_\tau(k) := (\mathrm{d}\tau^{-1}_{h(\Omega_k - \mathcal{A}(r_{k+\alpha})u_k)})^* \mu_k - (\mathrm{d}\tau^{-1}_{-h(\Omega_{k-1} - \mathcal{A}(r_{k-1+\alpha})u_{k-1})})^* \mu_{k-1}
$$

Disrete Euler-Poincare equations

- the unconstrained case $Q = G$

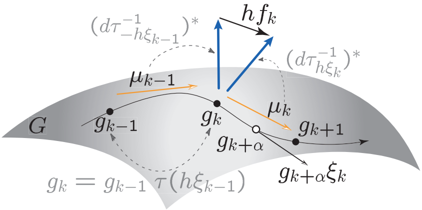


Figure: Tangent maps $d\tau^{-1}$ transforming momenta

Continuous	Discrete
$\dot{\mu} = \mathrm{ad}^*_\xi\,\mu + f$	$(\mathrm{d}\tau^{-1}_{h\xi_k})^*\mu_k - (\mathrm{d}\tau^{-1}_{-h\xi_{k-1}})^*\mu_{k-1} = hf_k$

Implementation

Simple matrix operations. Example: $G = SE(2)$, $\tau = \mathrm{cay}$

$$\mathrm{cay}(\widehat{v}) = \begin{bmatrix} \frac{1}{4+(v^1)^2} \begin{bmatrix} (v^1)^2 - 4 & -4v^1 & -2v^1v^3 + 4v^2 \\ 4v^1 & (v^1)^2 - 4 & 2v^1v^2 + 4v^3 \end{bmatrix} \\ \begin{matrix} 0 & \qquad\quad 0 & \qquad\quad 1 \end{matrix} \end{bmatrix}$$

The maps $[\mathrm{d}\tau_{\xi}^{-1}]$ can be expressed as the 3×3 matrices:

$$[\mathrm{dcay}_{\widehat{v}}^{-1}] = \mathbf{I}_3 - \frac{1}{2}[\mathrm{ad}_v] + \frac{1}{4}\begin{bmatrix} v^1 \cdot v & \mathbf{0}_{3\times 2} \end{bmatrix}$$

where

$$[\mathrm{ad}_v] = \begin{bmatrix} 0 & 0 & 0 \\ v^3 & 0 & -v^1 \\ -v^2 & v^1 & 0 \end{bmatrix}.$$

Note: a general method for any matrix group is also available

Discrete Nonholonomic Momentum Map

- Define the *local* discrete momentum map $\mathsf{J}^{\mathrm{loc}} : TM \times \mathfrak{g} \to \mathfrak{g}^*$

$$\mathsf{J}^{\mathrm{loc}}(r_k, u_k, \xi_k) = (\mathrm{d}\tau^{-1}_{h\xi_k})^* \mu_k, \qquad \text{where } \mu_k = \frac{\partial \ell}{\partial \xi}(r_k + \alpha u_k, u_k, \xi_k),$$

and the *spatial* discrete momentum map $\mathsf{J} : TQ \to \mathfrak{g}^*$ through

$$\mathsf{J}(r_k, u_k, g_k, v_k) := \mathrm{Ad}^*_{g_k^{-1}} \mathsf{J}^{\mathrm{loc}}(r_k, u_k, g_k^{-1} v_k),$$

where $(r_k, u_k) \in TM$ and $(g_k, v_k) \in TG$.

Discrete Nonholonomic Momentum Map

- Define the *local* discrete momentum map $\mathsf{J}^{\mathrm{loc}} : TM \times \mathfrak{g} \to \mathfrak{g}^*$

  $$\mathsf{J}^{\mathrm{loc}}(r_k, u_k, \xi_k) = (\mathrm{d}\tau^{-1}_{h\xi_k})^* \mu_k, \qquad \text{where } \mu_k = \frac{\partial \ell}{\partial \xi}(r_k + \alpha u_k, u_k, \xi_k),$$

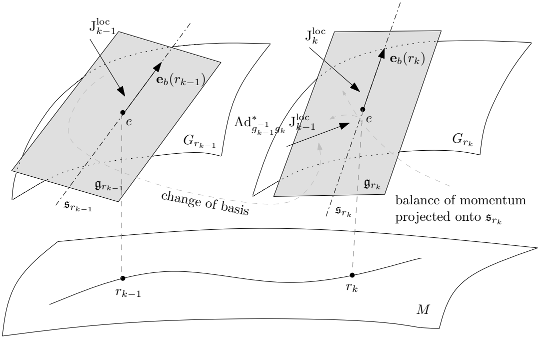
  and the *spatial* discrete momentum map $\mathsf{J} : TQ \to \mathfrak{g}^*$ through

  $$\mathsf{J}(r_k, u_k, g_k, v_k) := \mathrm{Ad}^*_{g_k^{-1}} \mathsf{J}^{\mathrm{loc}}(r_k, u_k, g_k^{-1} v_k),$$

  where $(r_k, u_k) \in TM$ and $(g_k, v_k) \in TG$.
- The *momentum components* $\mathsf{J}^{\mathrm{nh}}_b(r_k, u_k, g_k, v_k)$ at point k along the basis elements $e_b : Q \to \mathfrak{s}$ are

  $$\begin{aligned} \mathsf{J}^{\mathrm{nh}}_b(r_k, u_k, g_k, v_k) &= \langle \mathsf{J}(r_k, u_k, g_k, v_k), e_b(r_k, g_k) \rangle \\ &= \langle \mathsf{J}^{\mathrm{loc}}(r_k, u_k, g_k^{-1} v_k), e_b(r_k) \rangle. \end{aligned}$$

Discrete Momentum Map Evolution



Discrete Momentum Map Change

The momentum components J^{nh}_b evolve along discrete LDAP solution trajectories according to (denote $J(k) := J(r_k, u_k, g_k, v_k)$)

$$J^{nh}_b(k) - J^{nh}_b(k-1) = \langle J(k-1), e_b(r_k, g_k) - e_b(r_{k-1}, g_{k-1}) \rangle.$$

* consistent with previous results, e.g. *Cortes, 2001; Ferraro et. al. 2007*

Outline

Optimal Control

Goal: find an optimal trajectory to a desired state

- Compute the forces $f(t)$ such that the systems moves from $(q(0), \dot{q}(0))$ to $(q(T), \dot{q}(T))$ during a time interval $[0, T]$
- minimizing the cost function

$$J(q, f) = \int_0^T C(q(t), f(t))\mathrm{dt}, \tag{1}$$

e.g. minimum control effort: $C = \frac{1}{2}\|f\|^2$;
min. time: $C = 1$.

- subject to discrete equations of motion
- other constraints such as joint limits, obstacles, etc...

For clarity, consider the simpler case $Q = G$

Nonholonomic distribution $\mathfrak{h} \subset \mathfrak{g}$ (the *sub-Riemannian* case):

$$\text{velocity } \xi \in \mathfrak{h} = \text{span}\{X_1, ..., X_m\}, m < n, \quad \langle\langle X_i, X_j \rangle\rangle = \delta_{ij}$$

The dynamics satisfies

$$\begin{aligned}
&\langle (\mathrm{d}\tau^{-1}_{h\xi_k})^* \mu_k - (\mathrm{d}\tau^{-1}_{-h\xi_{k-1}})^* \mu_{k-1} - hf_k, X_i \rangle = 0, \quad i = 1, ..., m, \\
&\langle\langle \xi_k, X_i \rangle\rangle = 0, \quad\quad i = m+1, ..., n, \\
&\langle \mu_k, X_i \rangle = \begin{cases} \langle \mathbb{I}\, \xi_k, X_i \rangle, & i = 1, ..., m \\ 0, & i = m+1, ..., n \end{cases}, \\
&g_k^{-1} g_{k+1} = \tau(h\xi_k).
\end{aligned}$$

Necessary Conditions for Optimality

Define the Lagrangian multipliers $\eta_k \in \mathfrak{h}$, $\rho_k \in \mathfrak{h}^{\perp *}$, $\lambda_k \in \mathfrak{g}^*$ and and the *Hamiltonian* function

$$H_k := H(\xi_{k-1}, \xi_k, f_k, \eta_k) =$$

$$\langle (\mathrm{d}\tau^{-1}_{h\xi_k})^* \mathbb{I}\,\xi_k - (\mathrm{d}\tau^{-1}_{-h\xi_{k-1}})^* \mathbb{I}\,\xi_{k-1} - hf_k, \eta_k \rangle + \frac{h}{2}\|f_k\|^2,$$

and the *augmented* discrete cost function

$$J'_d(\xi_{0:N-1}, f_{0:N}, \zeta_{0:N}, \rho_{0:N-1}, \lambda_{0:N-1})$$

$$= \sum_{k=0}^{N} H_k + \sum_{k=0}^{N-1} \left(h\langle \rho_k, \xi_k \rangle + \langle \lambda_k, \tau^{-1}(g_k^{-1} g_{k+1}) - h\xi_k \rangle \right),$$

An optimal solution must satisfy

$$(\mathrm{d}\tau^{-1}_{h\xi_k})^* \lambda_k - (\mathrm{d}\tau^{-1}_{-h\xi_{k-1}})^* \lambda_{k-1} = 0,$$

$$\text{where } \lambda_k = \frac{\partial (H_k + H_{k+1})}{\partial \xi_k} + \rho_k = \frac{\partial \widetilde{H}_k}{\partial \xi_k} + \rho_k,$$

$$\widetilde{H}_k := -\langle (\mathrm{d}\tau^{-1}_{h\xi_k})^* \mathbb{I}\,\xi_k, \mathrm{Ad}_{\tau(h\xi_k)} \widetilde{f}^{\sharp}_{k+1} - \widetilde{f}^{\sharp}_k \rangle.$$

Indirect Optimal Control Formulation

An optimal trajectory (minimizing the control effort $\frac{h}{2}\sum_{k=0}^{N}\|\widetilde{f}_k\|^2$) satisfies

$$(\mathrm{d}\tau^{-1}_{h\xi_k})^*\lambda_k - (\mathrm{d}\tau^{-1}_{-h\xi_{k-1}})^*\lambda_{k-1} = 0, \quad k = 1, ..., N-1 \tag{2}$$

$$\tau^{-1}(\tau(h\xi_0)\cdots\tau(h\xi_{N-1})\cdot(g(0)^{-1}g(T))^{-1}) = 0, \tag{3}$$

where $\lambda_k \in \mathfrak{g}^*$is computed through

$$(\lambda_k)_i = \left\langle \mathbb{I}(\mathrm{d}\tau^{-1}_{h\xi_k}(\nu_k)) - h(\mathrm{d}\tau_{h\xi_k})^*\,\mathrm{ad}^*_{\left(\mathrm{Ad}_{\tau(h\xi_k)}\widetilde{f}^{\sharp}_{k+1}\right)}(\mathrm{d}\tau^{-1}_{h\xi_k})^*\,\mathbb{I}(\xi_k) + \rho_k, e^i \right\rangle$$
$$+ \left\langle \mathbb{I}(\xi_k), h\left(\mathrm{D}\,\mathrm{d}\tau^{-1}_{h\xi_k}\cdot e^i\right)(\nu_k)\right\rangle, \text{ where } \{e^i\} \text{ is the basis for } \mathfrak{g}$$

$$\nu_k = \mathrm{Ad}_{\tau(h\xi_k)}\widetilde{f}^{\sharp}_{k+1} - \widetilde{f}_k^{\sharp},$$

$$\xi_k \in \mathfrak{h},\ \rho_k \in \mathfrak{h}^{\perp *}.$$

Nn equations (2)-(3) in the Nn unknowns $\xi_{0:N-1}, \rho_{0:N-1}$ solved with standard root-finding

Outline

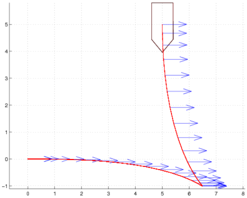
Simple boat model

- Group $G = SE(2)$ with coordinates $q = (\theta, x, y)$
- body fixed velocity $\xi \in \mathfrak{se}(2)$ defined by $\xi = (\omega, v, v^{\perp})$
- forces $f : SE(2) \times \mathfrak{se}(2) \rightarrow \mathfrak{se}(2)^*$ in the form

$$f(g, \xi) = -R(g, \xi)\xi + f_{ext}(g, \xi) + Bu,$$

where R is a damping matrix, f_{ext} are external forces due to wind or current, and $u = (u_r, u_l)$ are the thruster control inputs and B is

$$B = \begin{bmatrix} -c & c \\ 1 & 1 \\ 0 & 0 \end{bmatrix}.$$



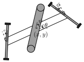
Boat station-keeping

RESL boat

Snakeboard

- $Q = SE(2) \times S \times S$, shape $r = (\psi, \phi)$, $G = SE(2)$ with coordinates (θ, x, y); distance l center-to-wheels, mass m, moments of inertia I and J.
- Constraint distribution:

$$\mathcal{D}_q = \text{span}\left\{\frac{\partial}{\partial\psi}, \frac{\partial}{\partial\phi}, c\frac{\partial}{\partial\theta} + a\frac{\partial}{\partial x} + b\frac{\partial}{\partial y}\right\},$$



where $a = -2l\cos\theta\cos^2\phi$, $b = -2l\sin\theta\cos^2\phi$, $c = \sin 2\phi$.

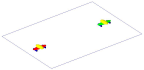
- Vertical space:

$$\mathcal{V}_q = \text{span}\left\{\frac{\partial}{\partial\theta}, \frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right\},$$

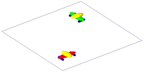
- Constrained symmetry space:

$$\mathcal{S}_q = \mathcal{V}_q \cap \mathcal{D}_q = \text{span}\left\{c\frac{\partial}{\partial\theta} + a\frac{\partial}{\partial x} + b\frac{\partial}{\partial y}\right\}.$$

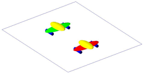
Optimal trajectories:



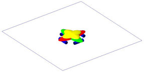
forward motion



90^o turn



parallel parking

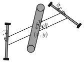


axis turn

Snakeboard

- $Q = SE(2) \times S \times S$, shape $r = (\psi, \phi)$, $G = SE(2)$ with coordinates (θ, x, y); distance l center-to-wheels, mass m, moments of inertia I and J.
- Constraint distribution:

$$\mathcal{D}_q = \text{span}\left\{\frac{\partial}{\partial\psi}, \frac{\partial}{\partial\phi}, c\frac{\partial}{\partial\theta} + a\frac{\partial}{\partial x} + b\frac{\partial}{\partial y}\right\},$$



where $a = -2l\cos\theta\cos^2\phi$, $b = -2l\sin\theta\cos^2\phi$, $c = \sin 2\phi$.

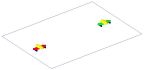
- Vertical space:

$$\mathcal{V}_q = \text{span}\left\{\frac{\partial}{\partial\theta}, \frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right\},$$

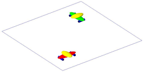
- Constrained symmetry space:

$$\mathcal{S}_q = \mathcal{V}_q \cap \mathcal{D}_q = \text{span}\left\{c\frac{\partial}{\partial\theta} + a\frac{\partial}{\partial x} + b\frac{\partial}{\partial y}\right\}.$$

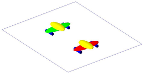
Optimal trajectories:



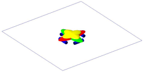
forward motion



90^o turn



parallel parking



axis turn

Outline

Challenges

- So far we have considered:
  Optimal Control based on *variational geometric integrators*
  (approach termed DMOC: *Discrete Mechanics and Optimal Control*)

Challenges

- So far we have considered:
  Optimal Control based on *variational geometric integrators*
  (approach termed DMOC: *Discrete Mechanics and Optimal Control*)
- Benefits:
  - a principled way to construct mechanical integrators
  - respects the geometric structure of the state-space
  - numerical stability, accuracy
  - suitable for discrete optimal control

Challenges

- So far we have considered:
  Optimal Control based on *variational geometric integrators*
  (approach termed DMOC: *Discrete Mechanics and Optimal Control*)
- Benefits:
  - a principled way to construct mechanical integrators
  - respects the geometric structure of the state-space
  - numerical stability, accuracy
  - suitable for discrete optimal control
- Limitations (as with any optimal control method):
  - lots of complex constraint $\Rightarrow$ expensive or even impossible
  - locally optimal $\Rightarrow$ solution might be in a "bad" homotopy class

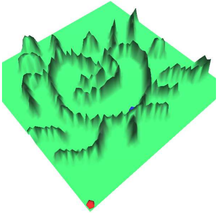
Challenges

- So far we have considered:
  Optimal Control based on *variational geometric integrators*
  (approach termed DMOC: *Discrete Mechanics and Optimal Control*)
- Benefits:
  - a principled way to construct mechanical integrators
  - respects the geometric structure of the state-space
  - numerical stability, accuracy
  - suitable for discrete optimal control
- Limitations (as with any optimal control method):
  - lots of complex constraint $\Rightarrow$ expensive or even impossible
  - locally optimal $\Rightarrow$ solution might be in a "bad" homotopy class
- Goal of this part:
  - extend DMOC to complex state-spaces cluttered with obstacles
  - find near globally optimal solution
  - guarantee efficiency

Outline

Finding global solution

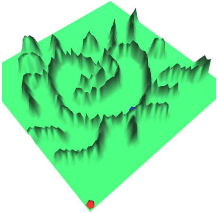
- Example - what is the optimal motion in this complex terrain?



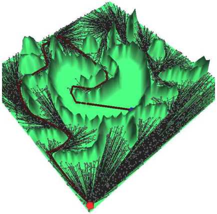
optimal motion?

Finding global solution

- Example - what is the optimal motion in this complex terrain?



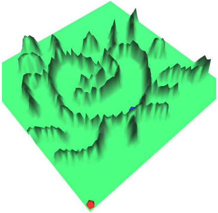
optimal motion?



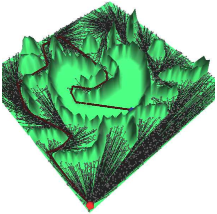
sampling-based roadmap

Finding global solution

- Example - what is the optimal motion in this complex terrain?



optimal motion?

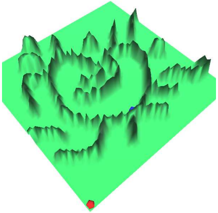


sampling-based roadmap

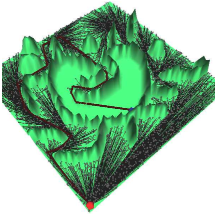
- Approach: sampling-based roadmaps
  (Kavraki; Latombe; LaValle; Amato; etc... 1996-present)
  - approximate free space as a tree/graph of reachable nodes

Finding global solution

- Example - what is the optimal motion in this complex terrain?



optimal motion?

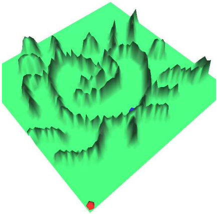


sampling-based roadmap

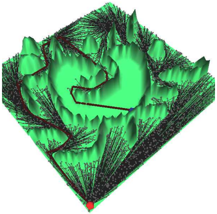
- Approach: sampling-based roadmaps
  (Kavraki; Latombe; LaValle; Amato; etc... 1996-present)
  - approximate free space as a tree/graph of reachable nodes
  - nodes are *sampled* in order to *explore* the state-space

Finding global solution

- Example - what is the optimal motion in this complex terrain?



optimal motion?

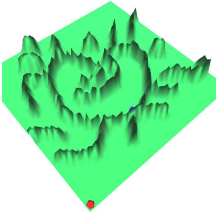


sampling-based roadmap

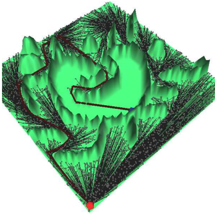
- Approach: sampling-based roadmaps
  (Kavraki; Latombe; LaValle; Amato; etc... 1996-present)
  - approximate free space as a tree/graph of reachable nodes
  - nodes are *sampled* in order to *explore* the state-space
  - edges correspond to motions satisfying the dynamics

Finding global solution

- Example - what is the optimal motion in this complex terrain?



optimal motion?

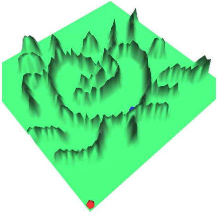


sampling-based roadmap

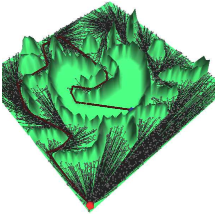
- Approach: sampling-based roadmaps
  (Kavraki; Latombe; LaValle; Amato; etc... 1996-present)
  - approximate free space as a tree/graph of reachable nodes
  - nodes are *sampled* in order to *explore* the state-space
  - edges correspond to motions satisfying the dynamics
  - optimal control path = shortest path on the graph

Finding global solution

- Example - what is the optimal motion in this complex terrain?



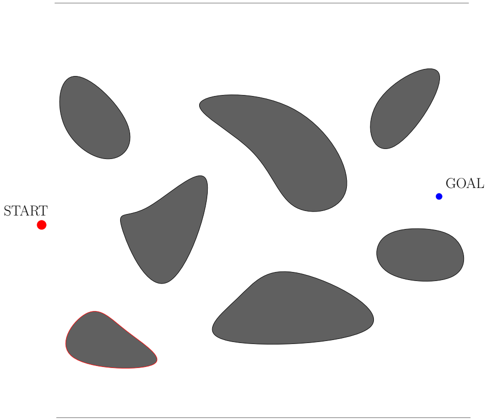
optimal motion?



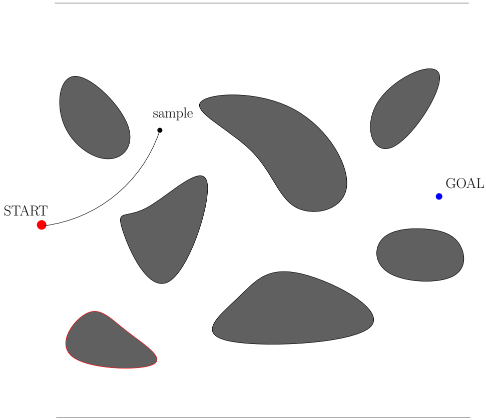
sampling-based roadmap

- Approach: sampling-based roadmaps
  (Kavraki; Latombe; LaValle; Amato; etc... 1996-present)
  - approximate free space as a tree/graph of reachable nodes
  - nodes are *sampled* in order to *explore* the state-space
  - edges correspond to motions satisfying the dynamics
  - optimal control path = shortest path on the graph
  - *global solution*, optimal with respect to the approximation

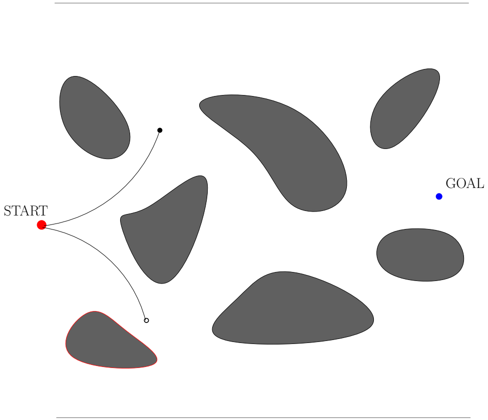
Combining DMOC with incremental roadmaps



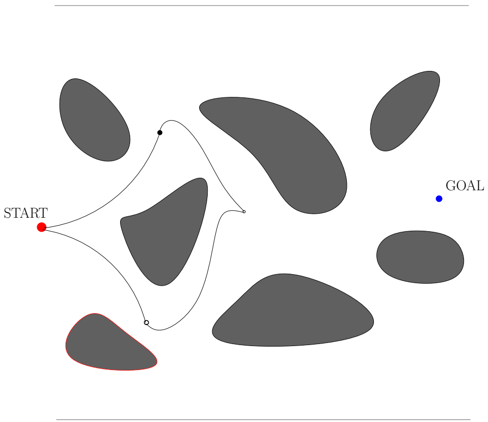
Combining DMOC with incremental roadmaps



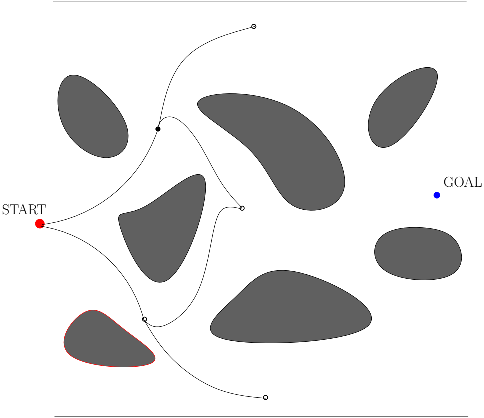
Combining DMOC with incremental roadmaps



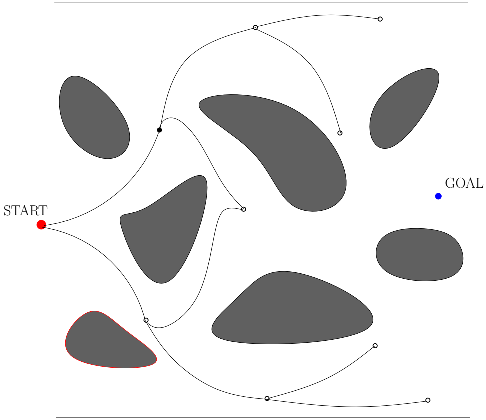
Combining DMOC with incremental roadmaps



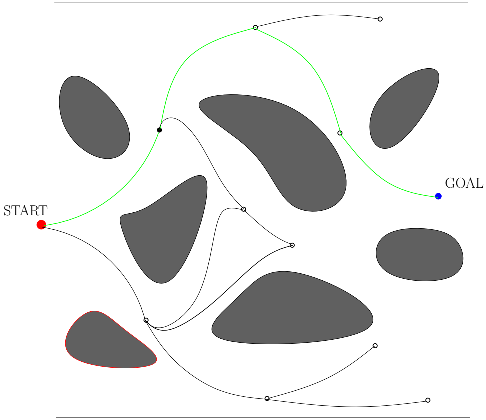
Combining DMOC with incremental roadmaps



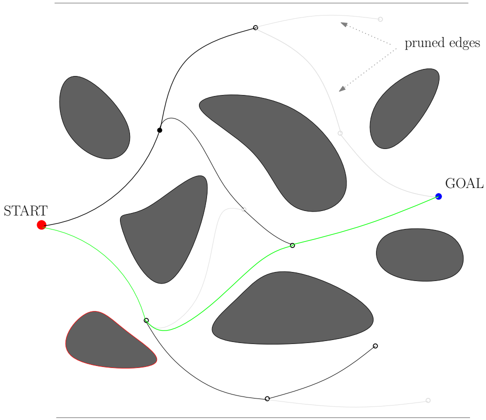
Combining DMOC with incremental roadmaps



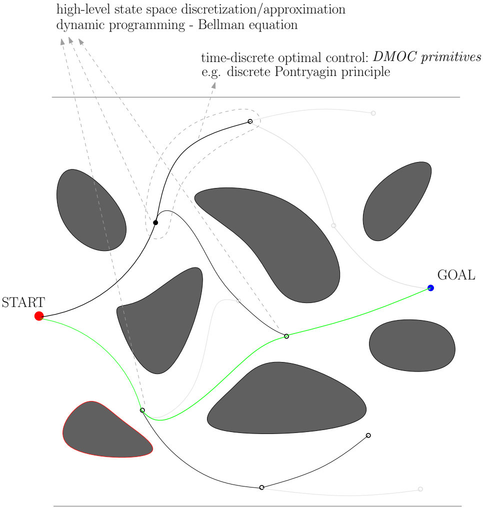
Combining DMOC with incremental roadmaps



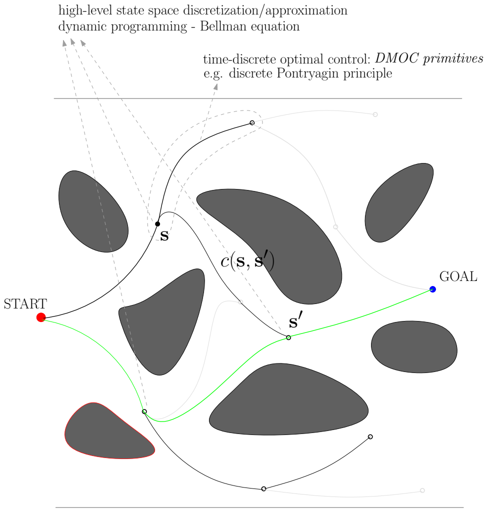
Combining DMOC with incremental roadmaps



Combining DMOC with incremental roadmaps



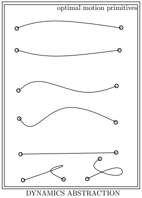
Combining DMOC with incremental roadmaps

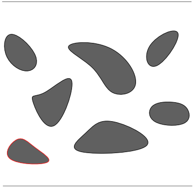


Bellman's principle: optimal cost $J^*(\mathbf{s}) = \min_{\mathbf{s}'}[J^*(\mathbf{s}') + c(\mathbf{s}, \mathbf{s}')]$,
where $J(\mathbf{s})$ - cost-to-go from $\mathbf{s}$ to the goal; $c(\mathbf{s}, \mathbf{s}')$ - cost b/n $\mathbf{s}$ and $\mathbf{s}'$.

Outline

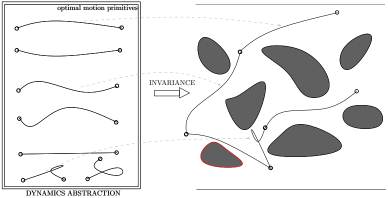
Combining DMOC with incremental roadmaps





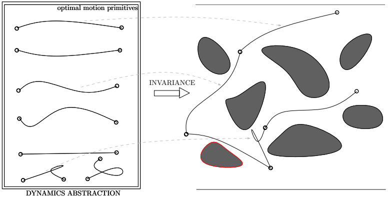
- *DMOC primitives*
  - simple optimized motions (precomputed)
  - invariant to group transformations
  - can be *concatenated* to produce complex trajectories

Combining DMOC with incremental roadmaps



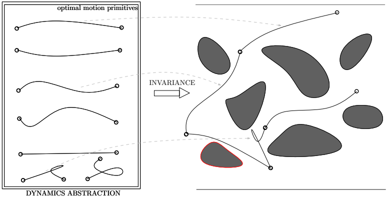
- *DMOC primitives*
  - simple optimized motions (precomputed)
  - invariant to group transformations
  - can be *concatenated* to produce complex trajectories

Combining DMOC with incremental roadmaps



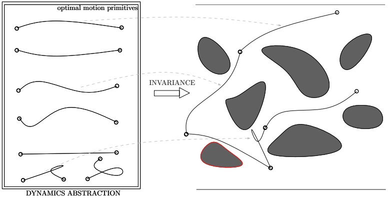
- *DMOC primitives*
  - simple optimized motions (precomputed)
  - invariant to group transformations
  - can be *concatenated* to produce complex trajectories

Combining DMOC with incremental roadmaps



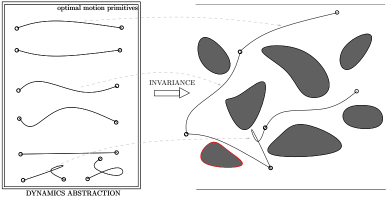
- *DMOC primitives*
  - simple optimized motions (precomputed)
  - invariant to group transformations
  - can be *concatenated* to produce complex trajectories
  - paths constructed from sequences of *compatible* primitives [1]

Combining DMOC with incremental roadmaps



- *DMOC primitives*
  - simple optimized motions (precomputed)
  - invariant to group transformations
  - can be *concatenated* to produce complex trajectories
  - paths constructed from sequences of *compatible* primitives [1]
- Main benefits:
  - control problem decomposed into two simpler sub-problems:
    1. how to sequence primitives to create roadmap edges
    2. find optimal trajectory through dynamic programming

Combining DMOC with incremental roadmaps



- *DMOC primitives*
  - simple optimized motions (precomputed)
  - invariant to group transformations
  - can be *concatenated* to produce complex trajectories
  - paths constructed from sequences of *compatible* primitives [1]
- Main benefits:
  - control problem decomposed into two simpler sub-problems:
    1. how to sequence primitives to create roadmap edges
    2. find optimal trajectory through dynamic programming
  - the complex differential control problem reduced to a lower dimensional algebraic one

Frazzoli, Dahleh, Feron, *IEEE Trans. on Robotics and Automation, 2005.*

Primitive Invariance

Denote state-space by X, state $x \in X$, and control set U, $u \in U$, e.g. $X = TQ$, $x = (q, v)$

- The flow $\varphi : X \times \mathbb{R} \to X$ of a primitive is G-*invariant* i.e.

$$\Phi_g(\varphi(x_0, t)) = \varphi(\Phi_g(x_0), t), \quad \Phi_g \text{ is the group action with } g \in G$$

- Two primitives π_1 and π_2 are *equivalent*, if $\exists g, T$ s.t.

$$(x_1(t), u_1(t)) = (\Phi_g(x_2(t - T)), u_2(t - T)), \forall t \in [t_{i,1}, t_{f,1}]$$

- Two primitives π_1 and π_2 are *compatible* if $\exists g_{12} \in G$ s.t.

$$x_1(T_1) = \Phi(g_{12}, x_2(0))$$

For discrete DMOC trajectories use discrete flow $\varphi_d : X \times \mathbb{N} \to X$

$$\varphi_d(x_k, i) \approx \varphi(x(kh), ih), \qquad x_k \approx x(kh)$$

Types of Primitives

- *Trim Primitives*: continuously parametrized steady-state motions

$$\alpha : t \in [0, T] \to (x_\alpha(t), u_a(t))$$

along left invariant vector field $\xi_\alpha \in \mathfrak{g}$ with constant control inputs

$$x_\alpha(t) = \Phi(\exp(t\xi_\alpha), x_\alpha(0)), u_\alpha(t) = u_\alpha, \forall t \in [0, T].$$

The trim primitives are denoted

$$\alpha(\tau) : t \in [0, T] \to (\Phi(\exp(t\xi_\alpha), x_\alpha(0)), u_\alpha),$$

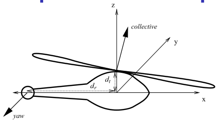
where τ is called coasting time, and the set of all such primitives defined by $\mathcal{T}_\alpha = \{\alpha(\tau), \tau \geq 0\}$.
The displacement of a trim primitive α with coasting time τ is simply $g_\alpha = \exp(\tau\xi_\alpha)$.

- *Maneuvers*: switches b/n steady-state motions. Therefore they are defined to be compatible form left and right with trim primitives. The set of maneuvers is denoted $\mathcal{M}(\mathcal{S}, G) \subseteq \mathcal{P}(\mathcal{S}, G)$. Formally, a maneuver π satisfies

$$\pi \in \mathcal{M}(\mathcal{S}, G) \Leftrightarrow \exists \alpha, \beta \in \mathcal{T}(\mathcal{S}, G) : \alpha\pi\beta \in \mathcal{P}(\mathcal{S}, G).$$

The displacement as a result of executing the maneuver is denoted g_π.

Example: helicopter



- Model – underactuated rigid body
  - *State*: orientatation $R \in SO(3)$, position $x \in \mathbb{R}^3$, angular velocity $\omega \in \mathbb{R}^3$, linear velocity $v \in \mathbb{R}^3$
  - *Controls*: collective u_c, yaw u_ψ, rotor forward pitch γ_p, rotor sideways roll γ_r with control input covectors

$$f^1(\gamma) = (d_t \sin\gamma_r, d_t \sin\gamma_p \cos\gamma_r, 0, \sin\gamma_p, \cos\gamma_p \cos\gamma_r),$$
$$f^2(\gamma) = (0, 0, d_r, 0, -1, 0).$$

  - gravity; bounds on velocity and controls
- Primitives: invariant under $G' = \mathrm{SO}(2) \times \mathbb{R}^3$

trim vector $\xi_\alpha \in \mathfrak{se}(3)$:

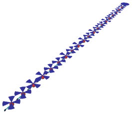
$$\xi_\alpha = \begin{bmatrix} 0 & -\omega_z & 0 & v_x \\ \omega_z & 0 & 0 & v_y \\ 0 & 0 & 0 & v_z \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

invariance conditions ($\dot{\xi} = 0$), with θ-pitch, ϕ-roll:

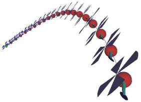
$$\gamma_p = 0, \gamma_r = 0, u_\psi = 0,$$
$$v_y \omega_z = -g \sin\theta,$$
$$-v_x \omega_z = g \cos\theta \sin\phi,$$
$$u_c = g \cos\theta \cos\phi.$$

Example: helicopter (cont.)

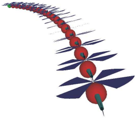
Trim Primitives:



α_1



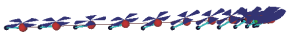
α_2



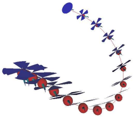
α_3

Maneuvers:

π_1



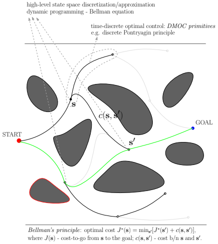
π_2



π_3

Outline

State space exploration

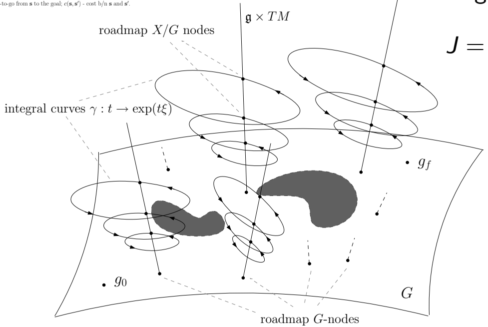


- State space approximation/discretization: sets of orbit of invariant vector fields $X \approx \{x_\alpha(t) \mid \Phi_g(\varphi(x_\alpha, t)) = \varphi(\Phi_g(x_\alpha), t)\}$
- Each graph node is a set of orbits attached at some $g \in G$.
- The Control Problem: Find $\{\pi_i, \alpha_i, \tau_i\}$ such that

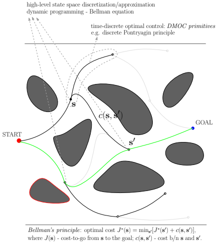
$$g_0^{-1} g_f = [\Pi_i \exp(\tau_i \xi_{\alpha_i}) g_{\pi_i}] \exp(\tau_N \xi_{\alpha_N})\}$$

minimizing total cost, for cost $c : \mathcal{C}_d \to \mathbb{R}$

$$J = \sum_i [(\tau_i c(\alpha_i) + c(\pi_i)] + \tau_N c(\alpha_N)$$



State space exploration

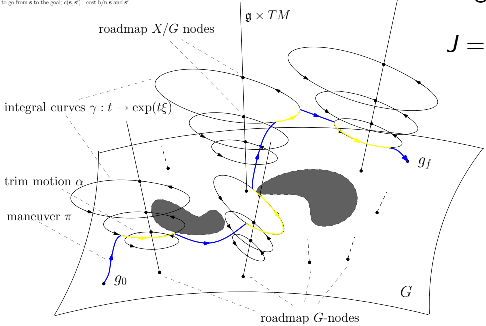


- State space approximation/discretization: sets of orbit of invariant vector fields $X \approx \{x_\alpha(t) \mid \Phi_g(\varphi(x_\alpha, t)) = \varphi(\Phi_g(x_\alpha), t)\}$
- Each graph node is a set of orbits attached at some $g \in G$.
- The Control Problem: Find $\{\pi_i, \alpha_i, \tau_i\}$ such that

$$g_0^{-1} g_f = [\Pi_i \exp(\tau_i \xi_{\alpha_i}) g_{\pi_i}] \exp(\tau_N \xi_{\alpha_N})\}$$

minimizing total cost, for cost $c : \mathcal{C}_d \longrightarrow \mathbb{R}$

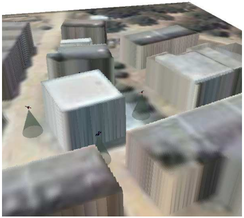
$$J = \sum_i [(\tau_i c(\alpha_i) + c(\pi_i)] + \tau_N c(\alpha_N)$$



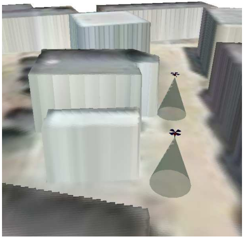
Example: helicopter

Part of the roadmap

topview

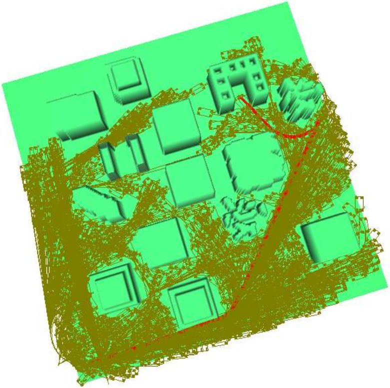


sideview



closeview

Example: car

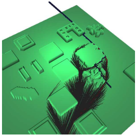


roadmap construction

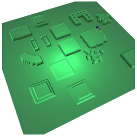
Outline

Roadmap extensions

- Goal with *time-dependent* dynamics



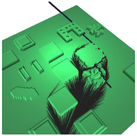
roadmap construction



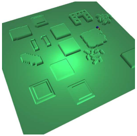
motion

Roadmap extensions

- Goal with *time-dependent* dynamics

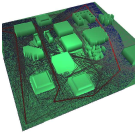


roadmap construction



motion

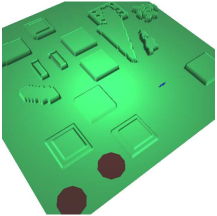
- Maximizing *coverage*



roadmap construction

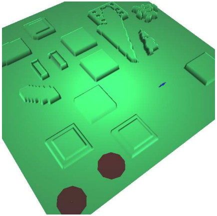
Goal with *uncertain dynamics*

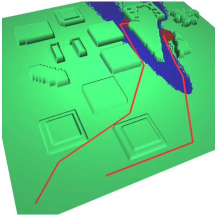
- goal distribution as particles
- goal heading north with uncertainty
- two vehicles with circular sensing radius



Goal with *uncertain dynamics*

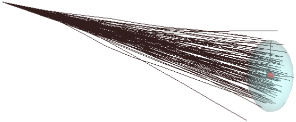
- goal distribution as particles
- goal heading north with uncertainty
- two vehicles with circular sensing radius
- vehicle controlled to gain maximum information about goal position



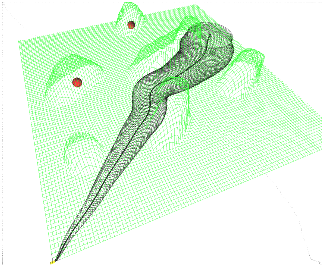


Control under uncertainty

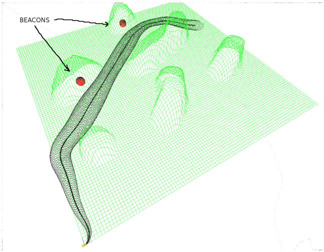
- Propagation of uncertainty



- Planning with Uncertainty: distance vs. uncertainty



Shortest Distance



Minimum Uncertainty

Issues / Directions

- Improve motion planning control in complex state spaces
- More insight into the structure / numerics of optimal control problems?
- Propagation of uncertainty / robustness to noise