Some old and new algorithms for robot motion planning

Quang-Cuong Pham

Department of Mechano-Informatics
University of Tokyo

October 16th, 2012
Robotics Laboratory
Seoul National University

Outline

Time-optimal control under dynamics constraints

- Time-optimal path parameterization algorithm
- Trajectory smoothing using time-optimal shortcuts
- Critically dynamic motion planning for humanoid robots

Affine trajectory deformation

- Motivation
- Proposed algorithm
- Examples of application

Outline

Time-optimal control under dynamics constraints
- Time-optimal path parameterization algorithm
- Trajectory smoothing using time-optimal shortcuts
- Critically dynamic motion planning for humanoid robots

Affine trajectory deformation
- Motivation
- Proposed algorithm
- Examples of application

Path parameterization algorithm

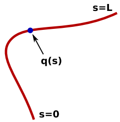
- Time-optimal path parameterization under torque limits algorithm (Bobrow et al 1985, and many others)
- Inputs :
  - Manipulator equation

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{q}}^\top \mathbf{C}(\mathbf{q})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \tau,$$

  - Torque limits for each joint i

$$\tau_i^{\min} \leq \tau_i(t) \leq \tau_i^{\max}$$

  - A given path $\mathbf{q}(s)$, $s \in [0, L]$



- Output : the time parameterization

$$\begin{array}{rccc} s: & [0,T] & \longrightarrow & [0,L] \\ & t & \longmapsto & s(t) \end{array}$$

that minimizes the traversal time T

Transforming the manipulator equations

- Let's express the manipulator equations in terms of $s, \dot{s}, \ddot{s}$
- Differentiate $\mathbf{q}$ with respect to s

$$\dot{\mathbf{q}} = \mathbf{q}_s \dot{s}$$

$$\ddot{\mathbf{q}} = \mathbf{q}_s \ddot{s} + \mathbf{q}_{ss} \dot{s}^2$$

- Substitute in the manipulator equation to obtain

$$\mathbf{M}(\mathbf{q}(s))\mathbf{q}_s(s)\ddot{s} + \left(\mathbf{M}(\mathbf{q}(s))\mathbf{q}_{ss}(s) + \mathbf{q}_s(s)^\top \mathbf{C}(\mathbf{q}(s))\mathbf{q}_s(s)\right) \dot{s} + \mathbf{g}(\mathbf{q}(s)) = \tau(s)$$

which can be rewritten as

$$\mathbf{a}(s)\ddot{s} + \mathbf{b}(s)\dot{s}^2 + \mathbf{c}(s) = \tau(s)$$

Transforming the torque constraints

- The torque constraints become

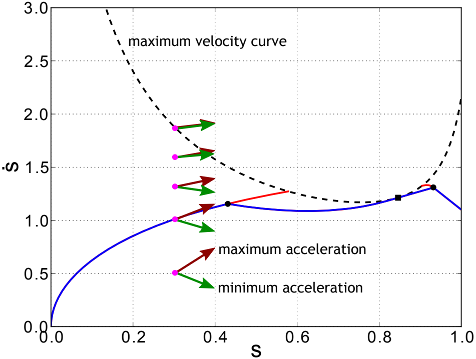
$$\tau_i^{\min} \leq a_i(s)\ddot{s} + b_i(s)\dot{s}^2 + c_i(s) \leq \tau_i^{\max}$$

- Set of $2n$ inequalities

$$\frac{\tau_i^{\min} - b_i(s)\dot{s}^2 - c_i(s)}{a_i(s)} \leq \ddot{s} \leq \frac{\tau_i^{\max} - b_i(s)\dot{s}^2 - c_i(s)}{a_i(s)}$$

- Minimal time = high $\dot{s}$
- Let's go to the phase plane $(s, \dot{s})$ ...

Phase plane $(s, \dot{s})$ integration



- "Bang-bang" behavior
- Switch points can be found very efficiently (Pfeiffer and Johanni 1987, Slotine and Yang 1989, Shiller and Lu 1992)

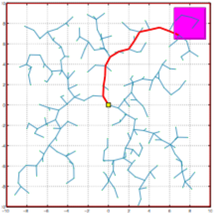
Global time-optimal algorithm

- Generate paths by grid search and apply the path parameterization algorithm

Shiller and Dubowsky, 1991

Trajectory smoothing using time-optimal shortcuts

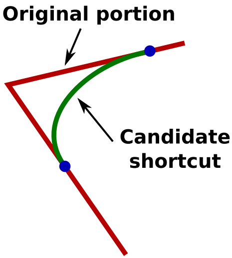
- Grid search does not work in higher dimensions (dof $>$ 3)
- RRT works well in high-dof, cluttered spaces, but produces non optimal trajectories

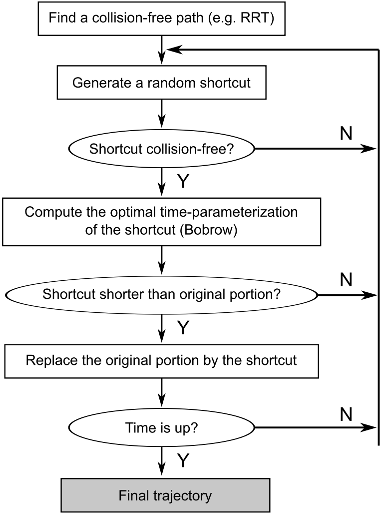


Karaman and Frazzoli, 2011

- Post-process with shortcuts, e.g. Hauser and Ng-Thow-Hing 2010 (acceleration and velocity limits)
- Here we propose to use time-optimal shortcuts with torque and velocity limits

Time-optimal shortcuts

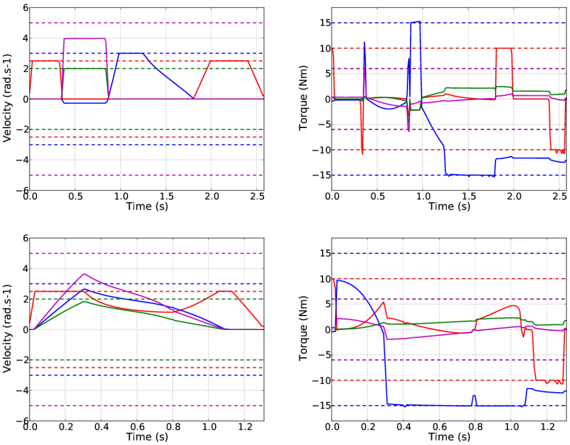




Pham, *Asian MMS*, 2012

Simulation results

Velocity profiles | Torque profiles



Pham, *Asian MMS*, 2012

Critically dynamic motion planning for humanoid robots

- *"ZMP is defined as that point on the ground at which the net moment of the inertial forces and the gravity forces has no component along the horizontal axes"* (Vukobratovic, 1969)
- Condition for dynamic balance: ZMP contained in the support area

Vukobratovic and Borovac, *IJHR*, 2004

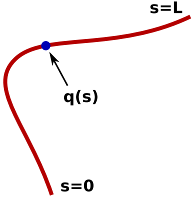
Optimal time parameterization

- ZMP equation

$$x_{\mathbf{ZMP}} = \frac{\sum_i m_i(\ddot{z}_i + g)x_i - \sum_i m_i \ddot{x}_i z_i - \sum_i (\mathbf{M}_i)_y}{\sum_i m_i(\ddot{z}_i + g)},$$

- Express as a function of joint angles $x_i = r(\mathbf{q})$
- Differentiating yields

$$x_i = r_{\mathbf{q}}\dot{\mathbf{q}} \qquad x_i = r_{\mathbf{q}}\ddot{\mathbf{q}} + \dot{\mathbf{q}}^\top r_{\mathbf{qq}}\dot{\mathbf{q}}$$



- Parameterize $\mathbf{q}$ by a path parameter s as $\mathbf{q} = \mathbf{q}(s)$
- This gives

$$\dot{\mathbf{q}} = \mathbf{q}_s \dot{s} \qquad \ddot{\mathbf{q}} = \mathbf{q}_s \ddot{s} + \mathbf{q}_{ss}\dot{s}^2$$

Optimal time parameterization (continued)

- Replacing yields

$$x_i = (r_{\mathbf{q}}\mathbf{q}_s)\ddot{s} + (r_{\mathbf{q}}\mathbf{q}_{ss} + \mathbf{q}_s^\top r_{\mathbf{qq}}\mathbf{q}_s)\dot{s}^2$$

- Thus $\ddot{x}_i$ can be expressed as

$$\ddot{x}_i = a_{x_i}(s)\ddot{s} + b_{x_i}(s)\dot{s}^2,$$

- Finally one can express

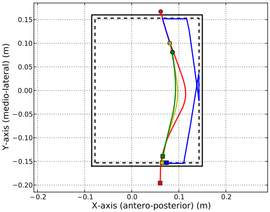
$$x_{\mathbf{ZMP}} = \frac{a(s)\ddot{s} + b(s)\dot{s}^2 + c(s)}{d(s)\ddot{s} + e(s)\dot{s}^2 + mg}$$

- This last expression can be treated by a Bobrow-like algorithm
- One can run the Bobrow algorithm to find the optimal parameterization s and which verifies

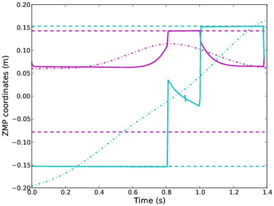
$$\begin{array}{ccccc} x_{\mathbf{min}} & \leq & x_{\mathbf{ZMP}} & \leq & x_{\mathbf{max}} \\ y_{\mathbf{min}} & \leq & y_{\mathbf{ZMP}} & \leq & y_{\mathbf{max}} \end{array}$$

Simulation results

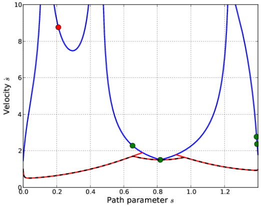
ZMP in space



ZMP as function of time



Phase-space $(s, \dot{s})$



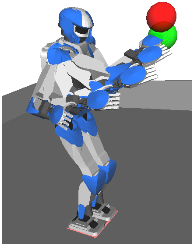
Pham and Nakamura, *Humanoids*, 2012

Outline

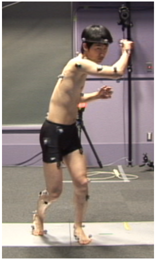
Time-optimal control under dynamics constraints
- Time-optimal path parameterization algorithm
- Trajectory smoothing using time-optimal shortcuts
- Critically dynamic motion planning for humanoid robots

Affine trajectory deformation
- Motivation
- Proposed algorithm
- Examples of application

Why deform trajectories ?



- To deal with perturbations
- To retarget motion-captured motions
- Advantages
  - save time
  - natural-looking motions



Existing approaches

- Spline-based (e.g. Lee and Shin, SIGGRAPH, 1999)

$$\mathbf{q} \to \mathbf{q} + \sum a_i \mathbf{s}_i$$

- Dynamic-system-based (e.g. Ijspeert et al, ICRA, 2002)

$$\mathbf{q} \text{ solution of } \quad \dot{\mathbf{q}} = \mathbf{f}(\mathbf{q}, \sum a_i \psi_i),$$

$$a_i \to a_i'$$

Existing approaches (continued)

Drawback: extrinsic basis functions ($\mathbf{s}_i$, ψ_i) $\Rightarrow$ artefacts

- loss of smoothness (splines that undulate too much,...)
- undesirable frequencies, phase-shift
- loss of invariance
- $\Rightarrow$ computational effort to reduce artefacts, preserve invariance
- need of reintegration

Affine geometry and invariance

- Affine transformation :

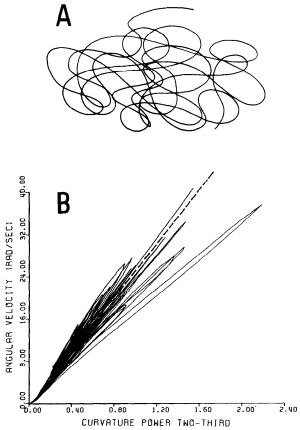
$$\mathbb{R}^n \to \mathbb{R}^n$$

$$\mathbf{x} \mapsto \mathbf{u} + \mathcal{M}(\mathbf{x})$$

- Affine transformations : group of dimension $n + n^2$
- Affine invariance : properties preserved by this group
  - non-affine-invariant : euclidean distance, angle, circles, euclidean velocity,. . .
  - affine-invariant : straight lines, parallelism, midpoints, ellipses, affine distance, affine velocity,. . .

Two-thirds power law

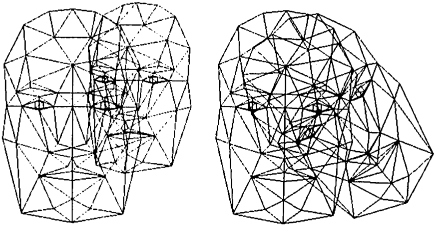
- Inverse relationship between velocity and curvature
- Formalization : $v = c\kappa^{-1/3}$ $(\omega = c\kappa^{2/3})$
- Very robust in hand drawing (Lacquaniti et al, *Acta Psy*, 1983, Pham and Bennequin, *in revision*, 2010)
- Exists in locomotion (Hicheur et al, *Exp Brain Res*, 2005, Bennequin et al, *PLoS Comp Biol*, 2010)



Lacquaniti et al, *Acta Psy*, 1983

Two-thirds power law and affine invariance

- Pollick and Sapiro, *Vision Res*, 1996 :
  two-thirds power law = constant affine velocity
- Possible explanation : affine invariance in vision, perception/action coupling



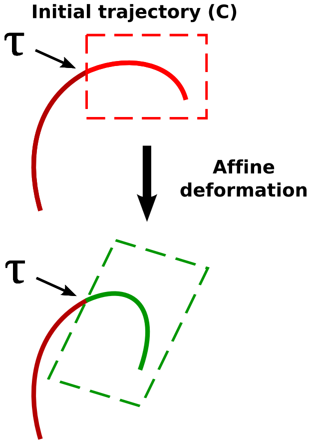
Koenderink and van Doorn, *JOSA*, 1991

Affine deformation of a trajectory

- Deformation of a trajectory $\mathbf{q} = (q_1(t), \ldots, q_n(t))_{t\in[0,T]}$ at a time instant τ by $\mathcal{F}$:

$$\begin{aligned} \forall t < \tau \qquad & \mathbf{q}'(t) = \mathbf{q}(t) \\ \forall t \geq \tau \qquad & \mathbf{q}'(t) = \mathcal{F}(\mathbf{q}(t)) \end{aligned}$$

- Affine deformation : $\mathcal{F}(\mathbf{x}) = \mathbf{u} + \mathcal{M}(\mathbf{x})$
- Use $\mathbf{u}$ and $\mathcal{M}$ to achieve various objectives
- Euclidian deformation : cf. Seiler et al, WAFR, 2010



Pham and Nakamura, *RSS*, 2012

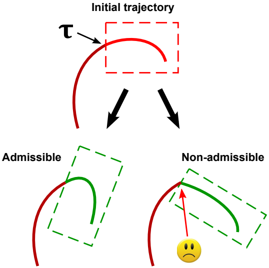
Advantages of affine deformations

- Purely intrinsic function basis $q_1, \ldots, q_n$
- $\Rightarrow$ Naturally preserves motion properties
  - smoothness for $t > \tau$
  - synchronization (e.g. in synergies)
  - periodicity, frequency spectrum, phase-shift
  - piece-wise parabolicity (e.g. industrial robots)
  - all affine-invariant properties (straight lines, affine velocity,...)

Advantages of affine deformations (continued)

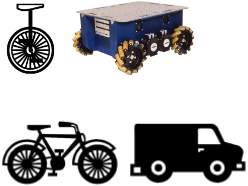
- Yet versatile enough : $n + n^2 + 1$ free parameters $(\mathbf{u}, \mathcal{M}, \tau)$
  - C^p-ness at τ
  - Desired final position, velocity, acceleration, ...
  - Optimization
  - Inequality constraints

C^p-ness at τ



Examples :

- Planar wheeled robots of type I : C^1
- Planar wheeled robots of type II : C^1 and curvature continuous (Pham, *RSS*, 2011 ; Pham and Nakamura, *ICRA*, 2012)
- Joint trajectories of manipulators : C^1



C^p-ness at τ (continued)

- Enforcement :
  - $C^0 \rightarrow$ fix $\mathbf{u}$
  - $C^p \rightarrow$ fix $\mathbf{u}$ and pn coefficients of $\mathcal{M}$

$$\mathbf{q}' = \mathbf{u} + \mathcal{M}(\mathbf{q})$$
$$\dot{\mathbf{q}}(\tau) = \mathcal{M}(\dot{\mathbf{q}}(\tau))$$
$$\ddot{\mathbf{q}}(\tau) = \mathcal{M}(\ddot{\mathbf{q}}(\tau))$$

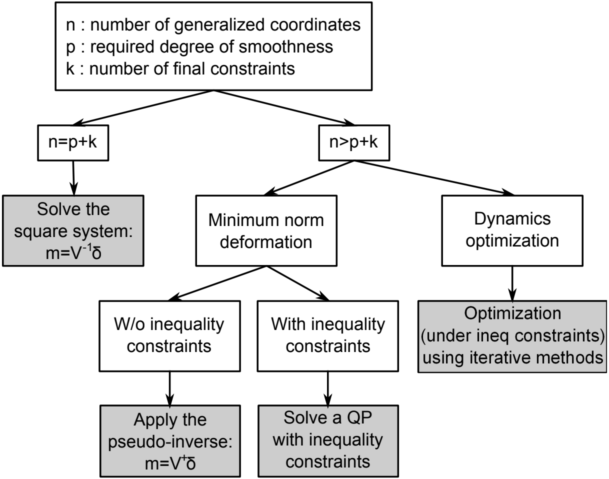
$$\begin{array}{cc} & \begin{array}{cccccc} \dot{q}(\tau) & \ddot{q}(\tau) & q^{(p)}(\tau) & & & \end{array} \\ \mathbf{M} = & \begin{pmatrix} 1 & & & m_{1,p+1} & \cdots & m_{1,n} \\ & \ddots & & \vdots & \vdots & \vdots \\ & & 1 & m_{p,p+1} & \cdots & m_{p,n} \\ 0 & \dots & 0 & 1+m_{p+1,p+1} & \cdots & m_{p+1,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & m_{n,p+1} & \cdots & 1+m_{n,n} \end{pmatrix} & n \\ & \begin{array}{cc} \text{p} & \text{n-p} \end{array} \end{array}$$

- Examples :
  - Type I wheeled robot : $n = 2$, $C^1 \Rightarrow$ 2 coefficients of $\mathcal{M}$ and τ left
  - Type II wheeled robot : 1 coefficients of $\mathcal{M}$ and τ left

Final configuration

- Reach a new desired final position : fix n coefficients of $\mathcal{M}$
- Desired final position and final velocity : fix $2n$ coefficients of $\mathcal{M}$
- k final constraints : fix kn coefficients of $\mathcal{M}$
- Recap : C^p at τ and k final constraints : fix $\mathbf{u}$ and $n(p+k)$
- Remaining : $n^2 - n(p+k)$ coefficients

Flowchart of the algorithm



Pham and Nakamura, *RSS*, 2012

Optimization

- Minimization of trajectory change (minimum norm deformation)

$$\text{minimize } \|\mathbf{q} - \mathbf{q}'\|$$
$$\downarrow$$
$$\text{minimize } \|\mathcal{M} - \mathcal{I}\|$$
$$\downarrow$$

exact formula using pseudo-inverse

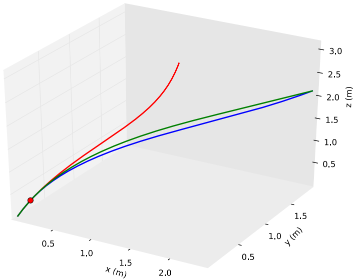
- Minimization of torques, energy, ...

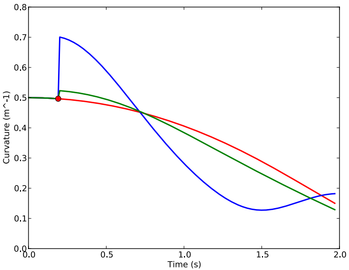
$$\text{minimize } \int_\tau^T c(\mathbf{q}'(t), \dot{\mathbf{q}}'(t), \ddot{\mathbf{q}}'(t))\,\mathrm{d}t$$
$$\downarrow$$
$$\text{minimize } \int_\tau^T c(\mathbf{q}(\tau) + \mathcal{M}(\mathbf{q}(t) - \mathbf{q}(\tau)), \mathcal{M}(\dot{\mathbf{q}}(t)), \mathcal{M}(\ddot{\mathbf{q}}(t)))\,\mathrm{d}t$$
$$\downarrow$$

iterative optimization in the space of $\mathcal{M}$

Optimization (bis)

- Maximization of rigidity : find $\mathcal{M}$ “closest” to an Euclidean transformation
- Use the polar decomposition $\mathbf{M} = \mathbf{QS}$ where $\mathbf{Q}$ is orthogonal and $\mathbf{S}$ is symmetric and minimize $\mathbf{S}^2 - \mathbf{I}$





Inequality constraints

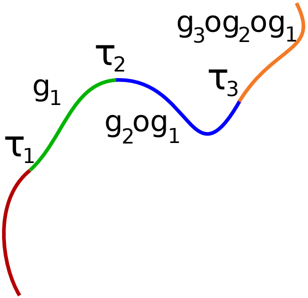
- Joint limits, velocity limits, obstacle avoidance,...
- Inequality on $\mathbf{q}'(t_i) \rightarrow$ inequality on coefficients of $\mathcal{M}$
- Hierarchy of
  - equality constraints (C^p-ness, final constraints)
  - inequality constraints (joint limits, obstacle avoidance)
- Optimization $\rightarrow$ Quadratic Programming

Subgroup constraints

- Equi-affine subgroup: dimension $n^2 + n - 1$ (e.g. hand movements, locomotion, etc. cf. Pollick and Sapiro 1997, Bennequin et al 2009)
- Euclidean subgroup: dimension $n(n+1)/2$ (e.g. needle steering, cf. Duindam et al 2010)

Group formulation

- Group property
  - Composition of deformations
  - Inverse
  - Lie group of dimension $n^2 - n(p+k) + 1$
- Group-based interpretation of trajectory redundancy
  - "Trajectory redundancy" : dimension of the deformation group
  - ("Configuration redundancy" : dimension $n - m$)
  - Adding a constraint = taking a subgroup
  - Sampling within the group



Examples

- Interactive real-time motion editing with OpenRAVE
- Motion transfer to a humanoid robot
- Combine affine deformations and time-optimal parameterization in pick-and-place tasks on conveyor belts

Conclusion

- Deformation algorithm inspired from human motor control
  - More likely to preserve human-like features
- Computational advantages
  - fast computation (one-step matrix computation)
  - naturally preserves relevant features
  - versatility :
    - C^p-ness at τ
    - final constraints
    - inequality constraints (joint limits, obstacles,. . .)
    - optimization (closeness, torques, energy,. . .)

General conclusions

- Two algorithms
  - Planning time-optimal trajectories under dynamics constraints using the path-parameterization algorithm
  - Deformation of trajectories using affine transformations
- Thank you very much for your attention and questions and comments !