Time-optimal trajectory planning under dynamics constraints
Old algorithm, new applications

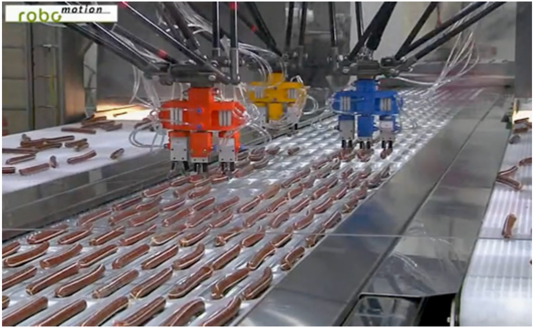
Quang-Cuong Pham

Department of Mechano-Informatics
University of Tokyo

November 29th, 2012
Workshop on "Generating Optimal Paths" at Humanoids'2012
Osaka, Japan

Time-optimal motion planning

- In the literature : minimum energy, minimum torque, maximum smoothness. . . planning algorithms
- But what is the most important in industry is time



Dynamics constraints

- In the literature : time-optimal motion planning under velocity and acceleration limits (e.g. Hauser and Ng-Thow-Hing 2010)
- These are kinematics constraints
- But what physically constraints the performance of the robot is the torque limits (= dynamics constraints)
- This case is much harder because of the nonlinearity !

Outline

Time-optimal path parameterization algorithm

Implementation in OpenRAVE

Some applications

Outline

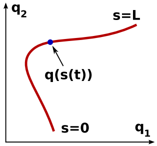
Time-optimal path parameterization algorithm

Implementation in OpenRAVE

Some applications

Time-optimal path parameterization algorithm under torque limits

- If the path is fixed, very nice algorithm developed in the 80's and 90's by Bobrow, Dubowsky, Gibson, Shin, McKay, Pfeiffer, Johanni, Slotine, Shiller, Zlajpah, Kunz, Stilman... and others



- Why this algorithn is great:
  - Fast ($\neq$ iterative optimization)
  - Exact: true time-optimal solution is attained ($\neq$ iterative optimization)
  - No need to recheck collisions
- Extension to the non-fixed path case: ongoing topic of research

Path parameterization algorithm

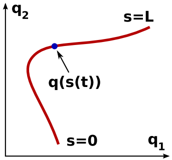
- Inputs :
  - Manipulator equation

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

  - Torque limits for each joint i

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

  - A given path $\mathbf{q}(s)_{s\in[0,L]}$ (set of points in the joint space)



- 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

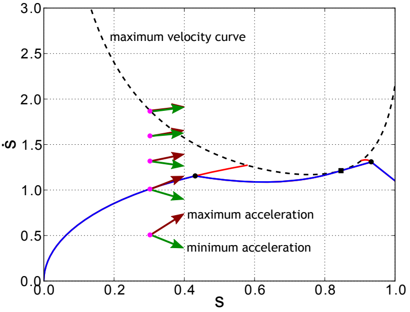
Outline of the algorithm

- Time minimal $\Leftrightarrow$ highest possible $\dot{s}$ (without violating the torque limits)
- Express the manipulator equations in terms of $s, \dot{s}, \ddot{s}$
- The torque limits become

$$\alpha(s, \dot{s}) \leq \ddot{s} \leq \beta(s, \dot{s}),$$

  where
  - $\alpha(s, \dot{s})$ is the minimum acceleration at $(s, \dot{s})$
  - $\beta(s, \dot{s})$ is the maximum acceleration at $(s, \dot{s})$
- If $\alpha(s, \dot{s}) > \beta(s, \dot{s})$: no possible acceleration $\ddot{s}$
- Maximum velocity curve defined by $\alpha(s, \dot{s}) = \beta(s, \dot{s})$

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



- "Bang-bang" behavior, switch points can be found very efficiently
- Computation time $O(n^2 N)$ ($\neq$ iterative optimization)
  - n : number of dofs
  - N : number of time-discretization steps
- Example: $n = 4$, $N = 500$ takes $\sim 2s$ in Python

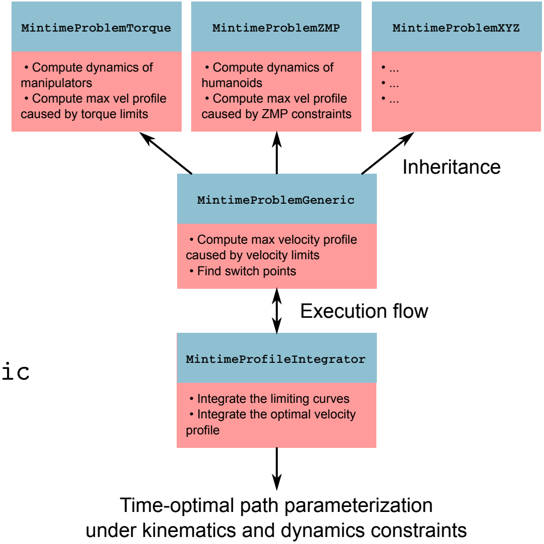
Outline

OpenRAVE

- OpenRAVE is "*an environment for testing, developing, and deploying robotics motion planning algorithms*", mainly developped by Rosen Diankov
- Very fast (C++), analytical inverse kinematics solver (IK-fast)
- Very easy to use (Python bindings layer)
- You can start developping meaningful robotics applications after 1 or 2 days!
- Industrially deployed at Mujin Inc.

Implementation architecture

- Open-source (`http://www.normalesup.org/~pham/code.html`)
- Modular design intended for re-usability
- New problems simply inherit from `MintimeProblemGeneric` and override robots- and constraints-specific methods



Other features of our implementation

- Consideration of velocity limits (correction from Kunz and Stilman 2012 based on Zlajpah 1996)
- Correct accelerations at the zero-inertia points (correction from Shiller and Lu 1992 and generalization of Kunz and Stilman 2012)
- Features under development
  - Motor dynamics
  - Third-order limits (to avoid torque jumps)

Outline

Time-optimal path parameterization algorithm

Implementation in OpenRAVE

Some applications

Global time-optimal algorithm

- Find the time-optimal trajectory between given initial and final configurations
- Generate paths by grid search and apply the path parameterization algorithm on each path

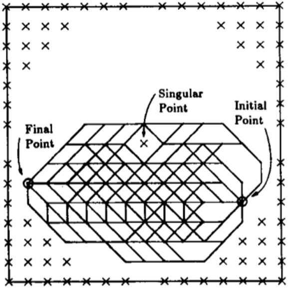
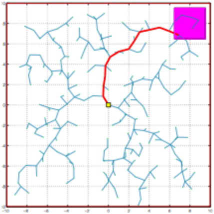


Fig. 15. Paths explored by the branch and bound search.

Shiller and Dubowsky, 1991

Trajectory smoothing using time-optimal shortcuts

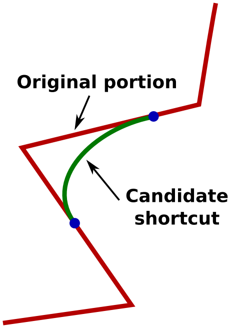
- Grid search does not work in higher dimensions (dof > 3)
- RRT works well in high-dof, cluttered spaces, but produces non optimal trajectories



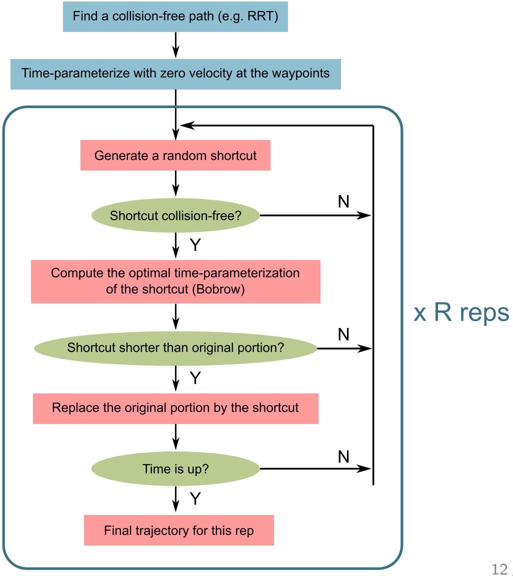
Karaman and Frazzoli, 2011

- 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 limits

Time-optimal shortcuts

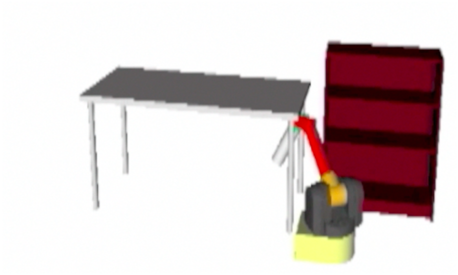


Pham, Asian-MMS 2012



Simulation results

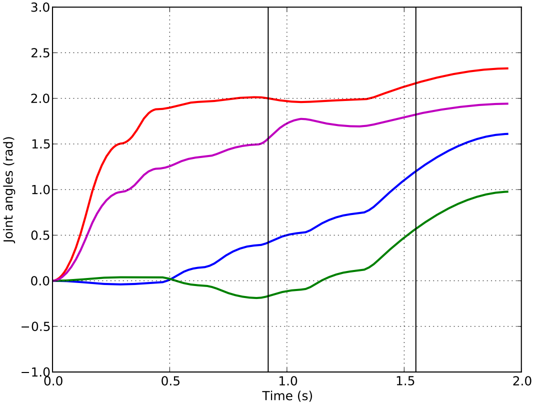
See video at `http://www.normalesup.org/~pham/videos/manipulator.m4v`



Simulation results

Joint angles profiles after ...

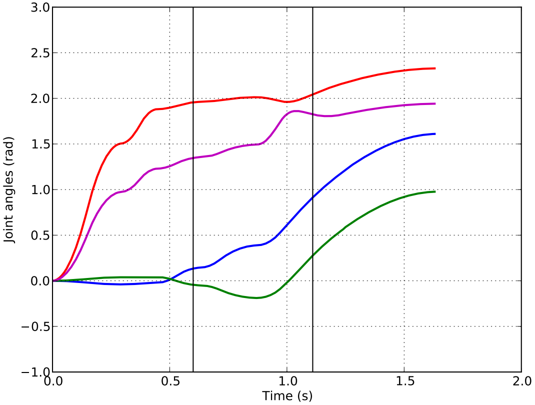
0 shortcut



Simulation results

Joint angles profiles after ...

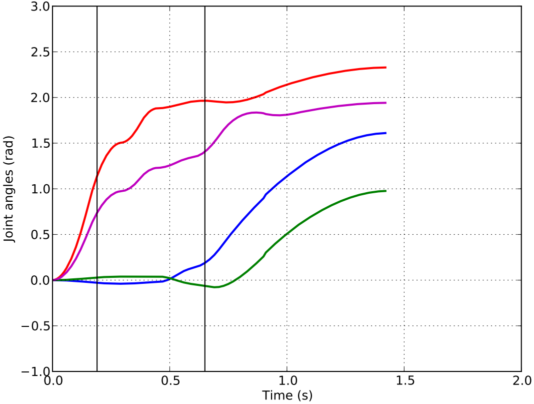
1 shortcut



Simulation results

Joint angles profiles after ...

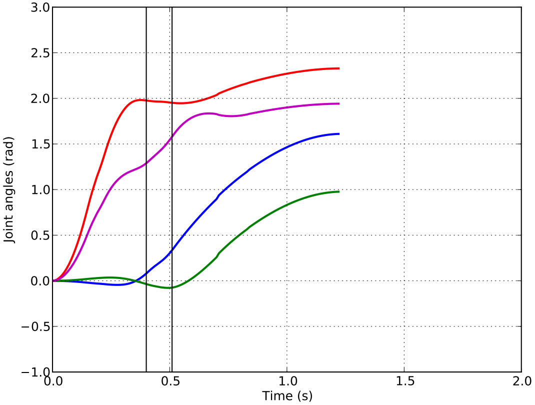
2 shortcuts



Simulation results

Joint angles profiles after ...

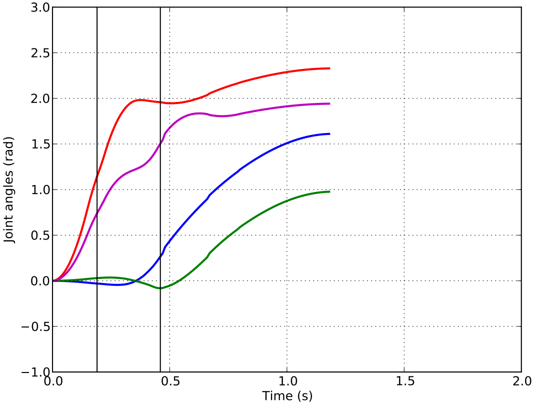
3 shortcuts



Simulation results

Joint angles profiles after ...

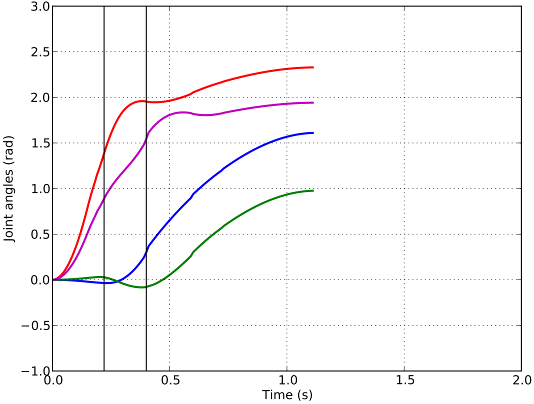
4 shortcuts



Simulation results

Joint angles profiles after ...

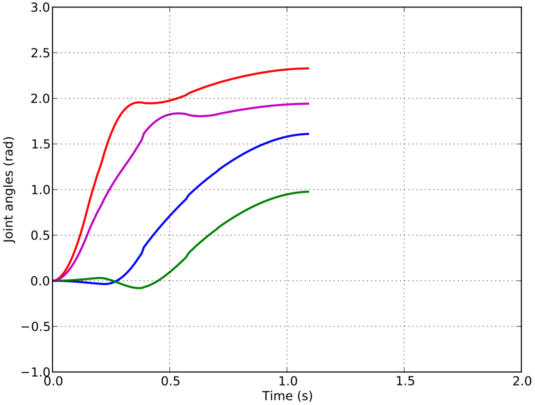
5 shortcuts



Simulation results

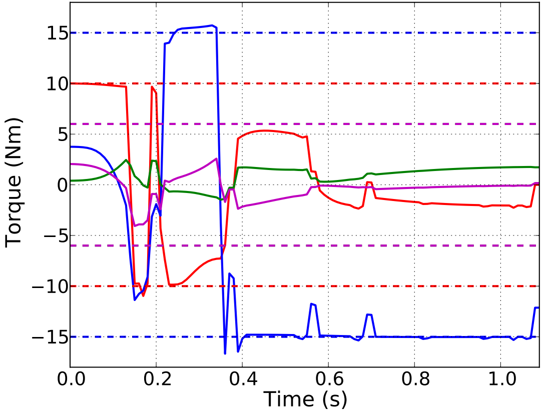
Joint angles profiles after ...

6 shortcuts



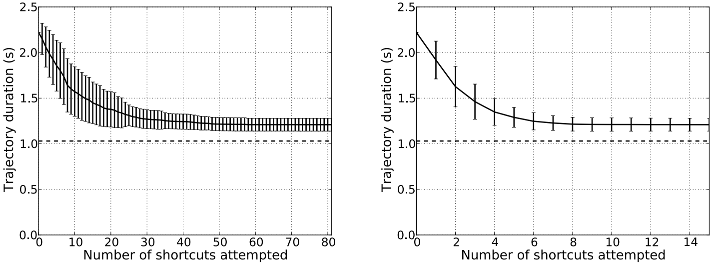
Simulation results

Torques profiles of the final trajectory



Computation time

- Time limit for one rep : 15s
  - ∼100 shortcuts attempts
  - ∼6 effective shortcuts
- No significant improvement after ∼7 effective shortcuts



- Choosing the best out of 10 reps (computation time: 2min30s) approaches the best out of 100 reps (computation time: 25min) by a margin of 9%

Discussion

- Current limitations
  - No guarantee of global optimality, but works well in practice
  - Torque jumps $\Rightarrow$ third-order optimization (motor model)
- Directions of research
  - Heuristic for choosing the endpoints of the random shortcuts?
  - Heuristic to choose the shortcut path between two given endpoints?
  - Third-order limits to avoid torque jumps

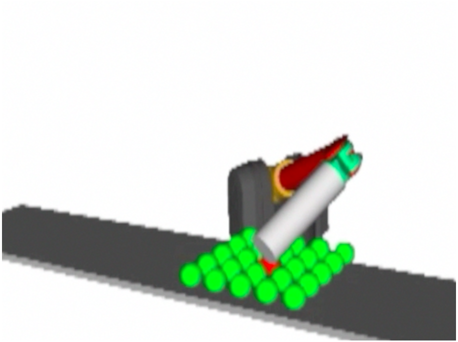
Pick and place on conveyor belts

- Need to generate very quickly a new trajectory (minimum time under torque constraints) for each new object arriving on the conveyor belt
- We propose to
  - Deform an optimal trajectory (using e.g. affine deformations, cf. Pham and Nakamura, RSS 2012) to reach the new target
  - Apply time-optimal path parameterization

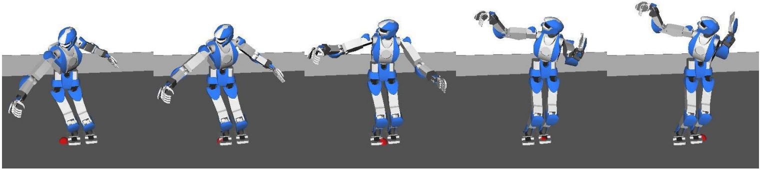
Simulation results

See video at

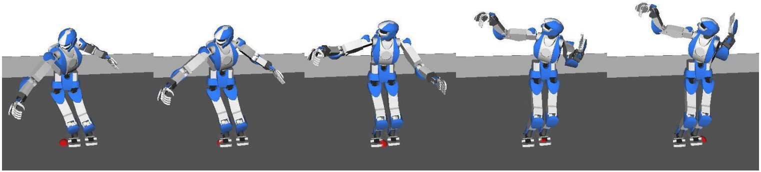
http://www.normalesup.org/~pham/videos/pick-and-place.m4v



Humanoids under ZMP constraints



Humanoids under ZMP constraints



- See you at `Interactive Session I` tomorrow 15:00-16:00!

Conclusion

- Open-source implementation of the Bobrow algorithm in OpenRAVE
- Modular design intended for easy re-usability
- Various new features (velocity limits, zero-inertia points)
- Applications to
  - trajectory smoothing using time-optimal shortcuts
  - fast trajectory deformations
- Thank you very much for your attention, questions and comments!