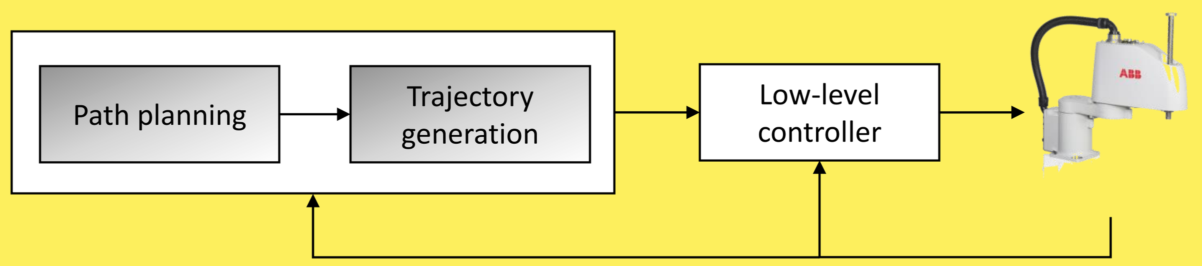
Sensor-Based Trajectory Planning in Dynamic Environments

Andreas Westerlund[1,2], Giacomo Spampinati[2], Johan Löfberg[1], Hamed Hagshenas[1], Mikael Norrlöf[1,2]
[1]Linköpings Universitet - Dept. of E.E., [2]ABB AB - Division of Robotics and Motion

Abstract

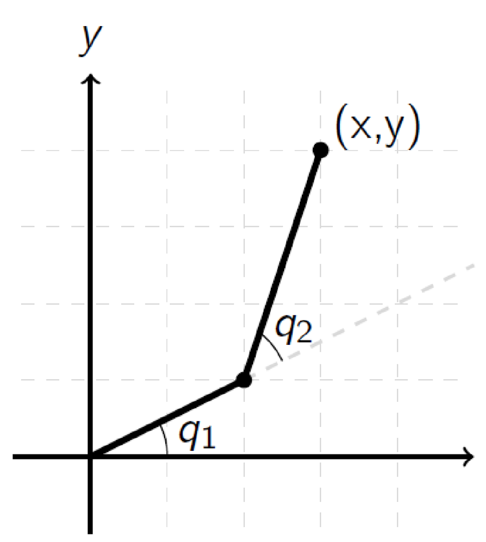
Traditional path planning of robotic manipulators uses the following structure:



In this work the Path planning and Trajectory generation steps are combined and an optimization solution is used. As a result, the path and the trajectory can be adopted to dynamic environment, for example tracking moving targets or avoiding moving obstacles. The solution is implemented and testes with a model of a SCARA robot.

Robot Model

The robot is a SCARA type manipultor with 2-DOF.



$$\mathbf{p} = \begin{pmatrix} x \\ y \end{pmatrix}$$

$$\mathbf{q} = \begin{pmatrix} q_1 \\ q_2 \end{pmatrix}$$

The dynamic model is given in the following form

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

where M is the inertia matrix, C the coriolis and centripetal terms and g the gravity vector.

An evaluation of the resulting algorithm is also performed using a dynamic model of a full 4-DOF SCARA robot to evaluate the actual computation time with a more complex dynamic model.

Optimal control problem

The combined path and trajectory planning problem can be solved using the following optimization formulation:

$$\min_{\boldsymbol{\tau}(t)} \quad T$$

$$s.t. \quad \begin{cases} M(\mathbf{q}(t))\ddot{\mathbf{q}}(t) + C(\mathbf{q}(t),\dot{\mathbf{q}}(t))\dot{\mathbf{q}}(t) + G(\mathbf{q}(t)) = \boldsymbol{\tau}(t) \\ \boldsymbol{\tau}_- \leq \boldsymbol{\tau}(t) \leq \boldsymbol{\tau}_+ \\ \dot{\mathbf{q}}_- \leq \dot{\mathbf{q}}(t) \leq \dot{\mathbf{q}}_+ \\ \ddot{\mathbf{q}}_- \leq \ddot{\mathbf{q}}(t) \leq \ddot{\mathbf{q}}_+ \\ \mathbf{p}(0) = \mathbf{p}_i, \quad \dot{\mathbf{p}}(0) = \dot{\mathbf{p}}_i \\ \mathbf{p}(T) = \mathbf{p}_T, \quad \dot{\mathbf{p}}(T) = \dot{\mathbf{p}}_T \end{cases}$$

Timed Elastic Nodes approach

To solve the optimal control problem partition the time interval into n subintervals

$$\begin{aligned} 0 = t_1 \leq t_1 + \Delta T_1 &= t_2, \\ t_2 \leq t_2 + \Delta T_2 &= t_3, \\ &\vdots \\ t_{n-1} \leq t_{n-1} + \Delta T_{n-1} &= t_n. \end{aligned}$$

In each sub-interval the joint variables can be paramterized as cubic splines

$$\mathbf{q}(t) = \mathbf{s}_k(t) \quad \text{for} \quad t_k \leq t < t_{k+1}$$

The state vector is defined as,

$$\mathbf{x}(t) = \begin{pmatrix} \mathbf{q}(t) \\ \dot{\mathbf{q}}(t) \\ \ddot{\mathbf{q}}(t) \end{pmatrix}, \quad \mathbf{x}(t;\mathbf{s}_k) = \begin{pmatrix} \mathbf{s}_k(t) \\ \dot{\mathbf{s}}_k(t) \\ \ddot{\mathbf{s}}_k(t) \end{pmatrix}$$

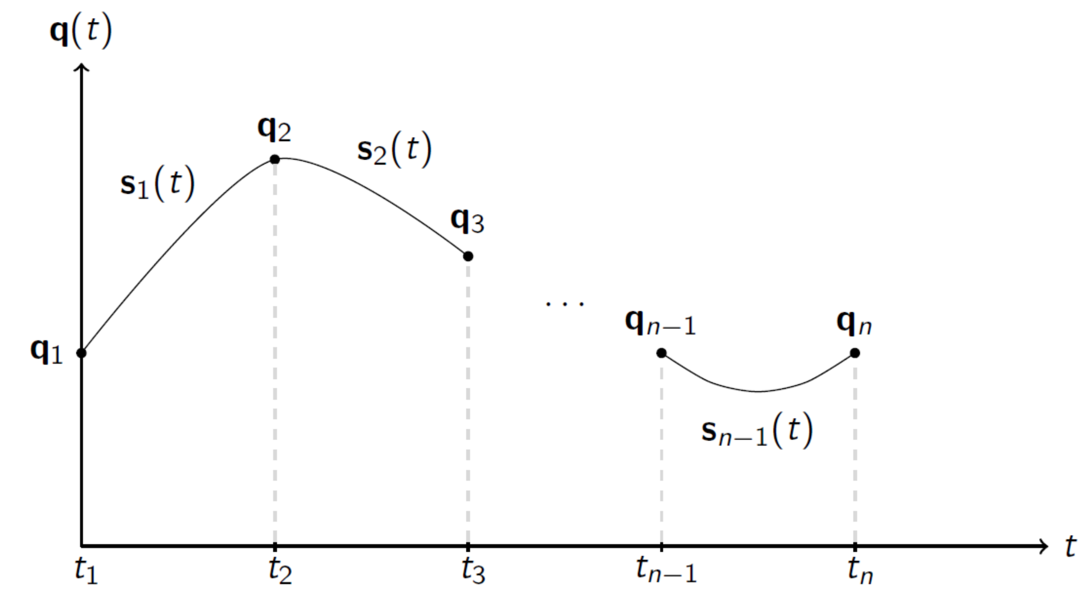
in addition the nodes and time intervals are combined into the *timed elastic nodes,*

$$\mathbf{b}_l = \{\mathbf{x}_1, \Delta T_1, \mathbf{x}_2, \Delta T_2, \ldots, \mathbf{x}_{n-1}, \Delta T_{n-1}, \mathbf{x}_n\}$$

This leads to the discretized optimal control formulation,

$$\min_{\mathbf{b}_l} \quad \sum_{k=1}^{n-1} \Delta T_k$$

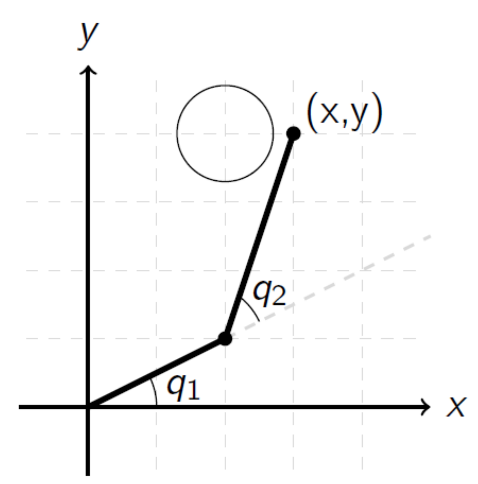
$$s.t. \quad \begin{cases} \mathbf{x}_1 = \mathbf{x}_i, \ \mathbf{x}_n = \mathbf{x}_f, \ \Delta T_k > 0 \\ \mathbf{x}_{k+1} = \mathbf{x}(t_k + \Delta T_k; \mathbf{s}_k) \\ g(\mathbf{x}_k, \Delta T_k) \geq \mathbf{0} \end{cases} \qquad (k = 1, 2, \ldots, n-1).$$



Collision Avoidance

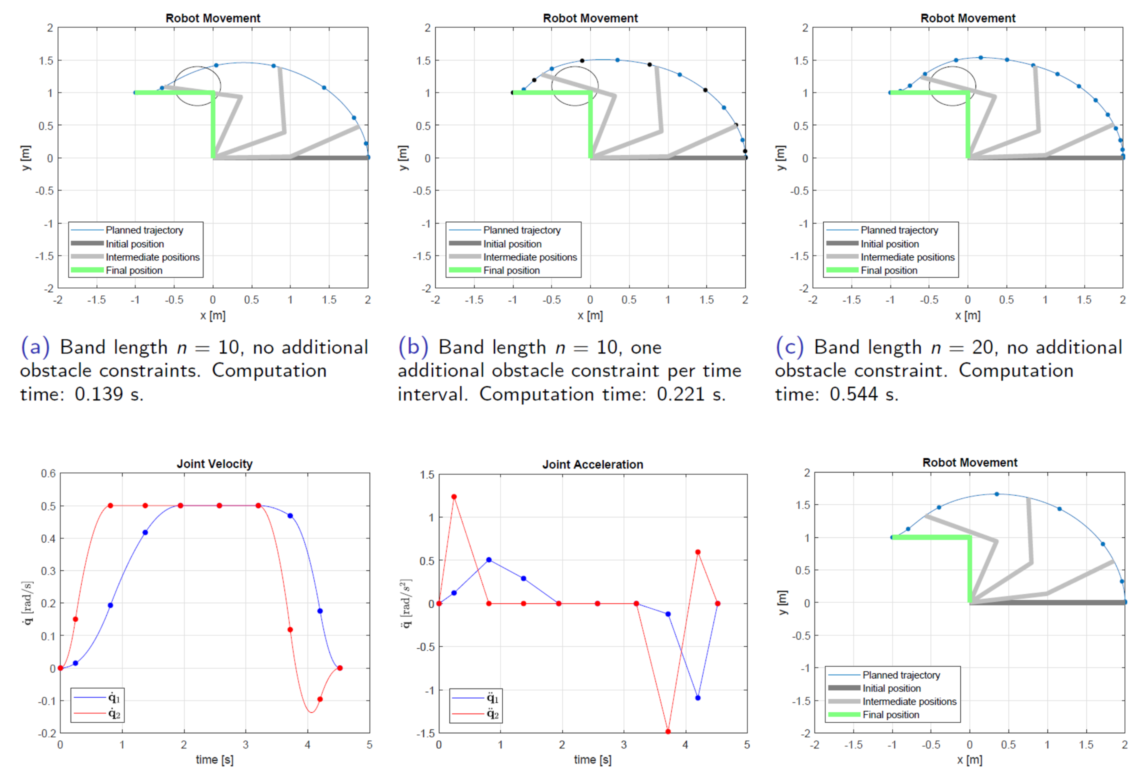
Assume the obstacle is below the robot arm and that only the end-effector needs to avoid the obstacle. Obstacles are modelled as spherical bodies,

$$||\chi(\mathbf{q}_k) - \mathcal{O}_j||^2 \geq (s + r_j)^2$$



Results

The approach is evaluated in a test case with one obstacle. Computational times are provided for different cases, the cycle time was 4.5 sec.



(a) Band length $n = 10$, no additional obstacle constraints. Computation time: 0.139 s.

(b) Band length $n = 10$, one additional obstacle constraint per time interval. Computation time: 0.221 s.

(c) Band length $n = 20$, no additional obstacle constraint. Computation time: 0.544 s.

Conclusions

- The planned trajectories are close to the optimal solution
- The spline-based trajectory planner yield time continuous trajectories
- Violations of the constraints between spline nodes
- Current implementation is not real-time capable, with 4-DOF scara model 3.5 sec cycle was planned in 8.5 sec.

Future work

- Investigate alternative methods to the interior-point method
- Alternative parametrisations of the splines
- Improve the computational efficiency

References

1. A. Westerlund, "Sensor-Based Trajectory Planning in Dynamic Environments", LiTH-ISY-EX--18/5164--SE.
2. M. Biel, M. Norrlöf, "Efficient Trajectory Reshaping in a Dynamic Environment", 2018 IEEE 15TH Intenational Workshop on Advanced Motion Control (AMC), International Workshop on Advanced Motion Control, 54-59, 2018.

LINKÖPING UNIVERSITY

LINKÖPING UNIVERSITY
ELLIIT / LINK-SIC