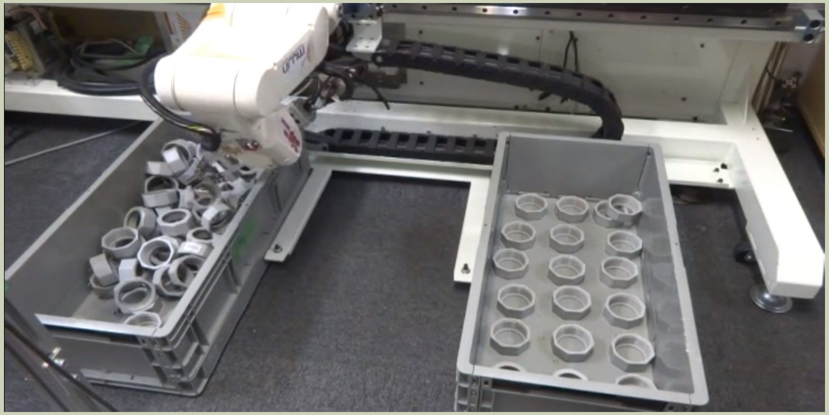
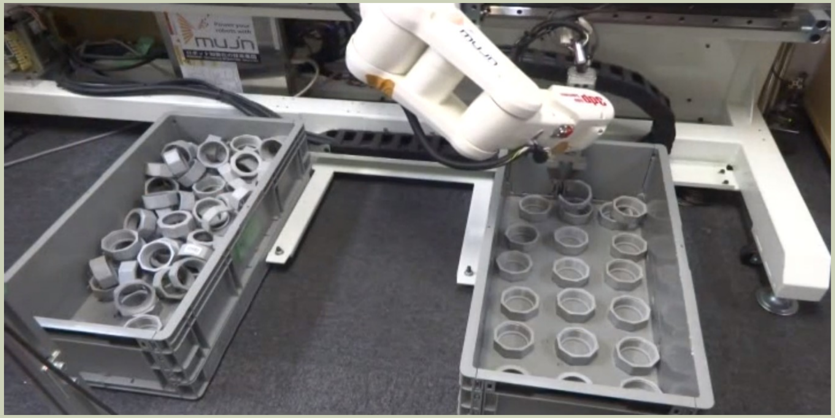
Integrating Dynamics into Industrial Motion Planning

Q.-C. Pham
NTU, Singapore

Path planning problem

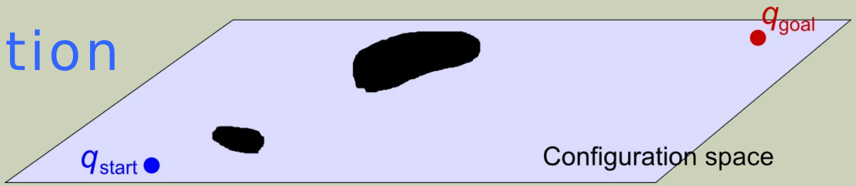
Find a collision-free path between q_{start} and q_{goal}



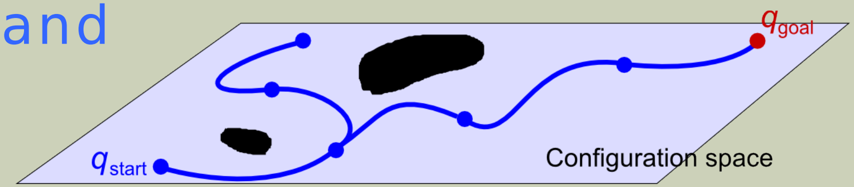


From academic breakthroughs…

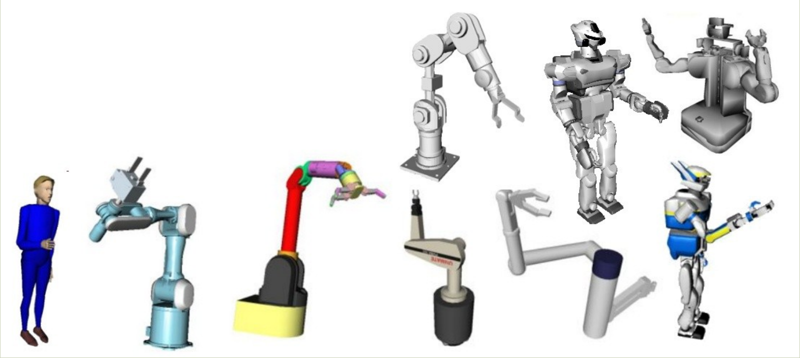
- Configuration space formulation (Lozano-Perez 1983)



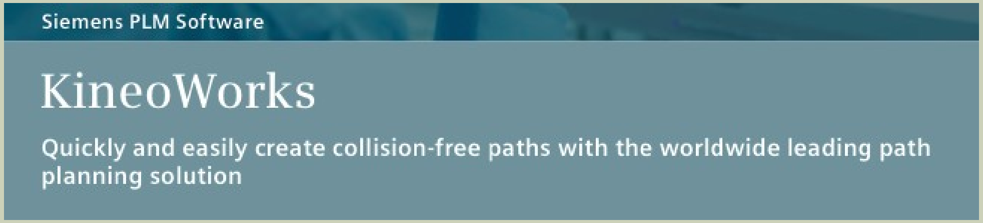
- Sampling-based planning (Kavraki et al 1996, Lavalle and Kuffner 2000)



- Efficient implementations
  - ROS / MoveIt !
  - OpenRAVE



… to industrial successes



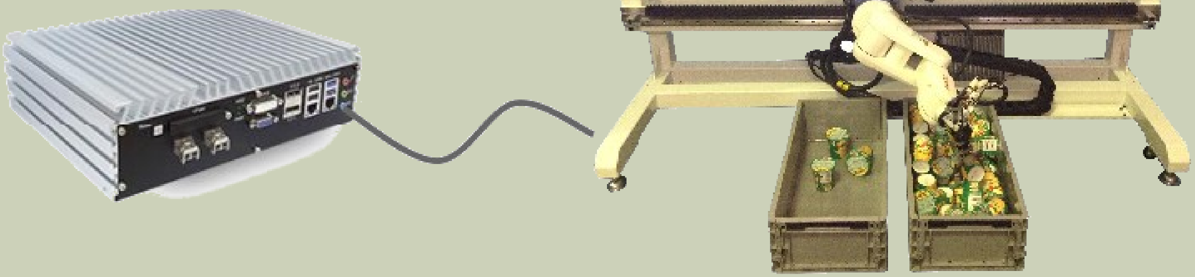
Siemens PLM Software

KineoWorks

Quickly and easily create collision-free paths with the worldwide leading path planning solution

ROS® industrial

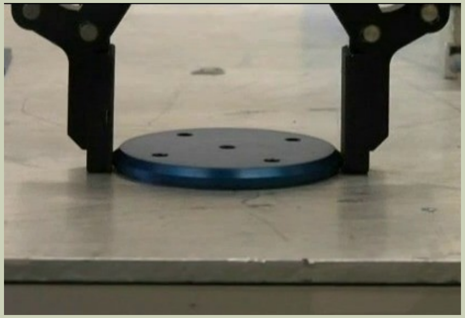
mujin



How about dynamics ?

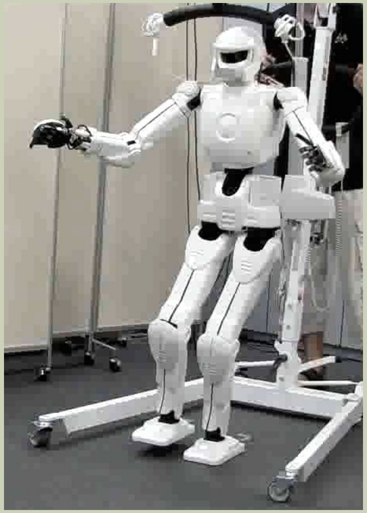
Torque constraints

Friction constraints



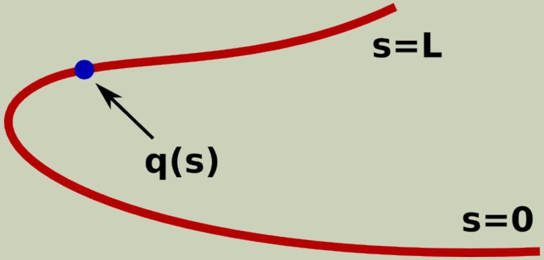
Fluid constraints

ZMP constraints



Planning with dynamics ?

- Planning in the state space ?
  - More dimensions (2n)
  - Obstacle avoidance difficult to guarantee
  - Less intuitive
- Trajectory decoupling (path + parameterization)
  - Cluttered environments
  - Can use regular PRM/RRT + many heuristics
  - Optimal time parameterization (Bobrow 1985 and many others)

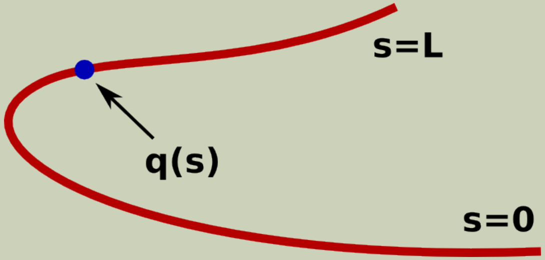


Time-Optimal Path Parameterization (TOPP)

- Developed by Bobrow (and many others)

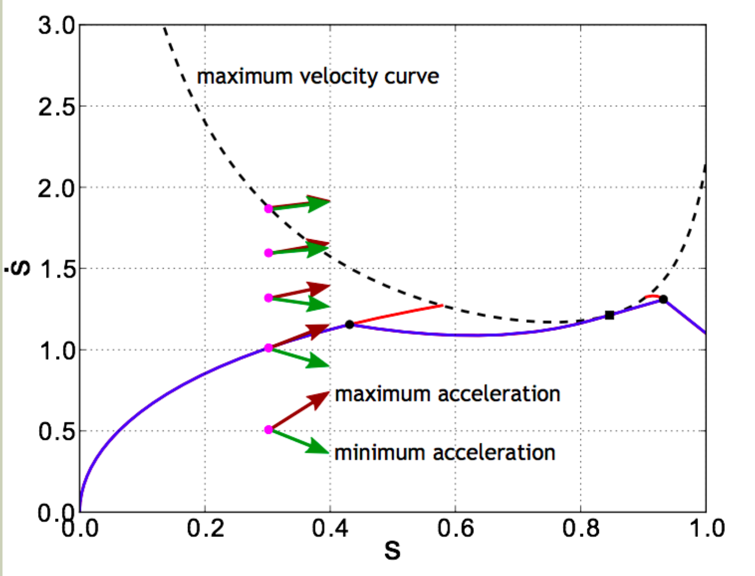
$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{q}}^\top \mathbf{C}(\mathbf{q})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \tau$$

$$\tau_i^{\min} \le \tau_i(u) \le \tau_i^{\max}$$



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

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



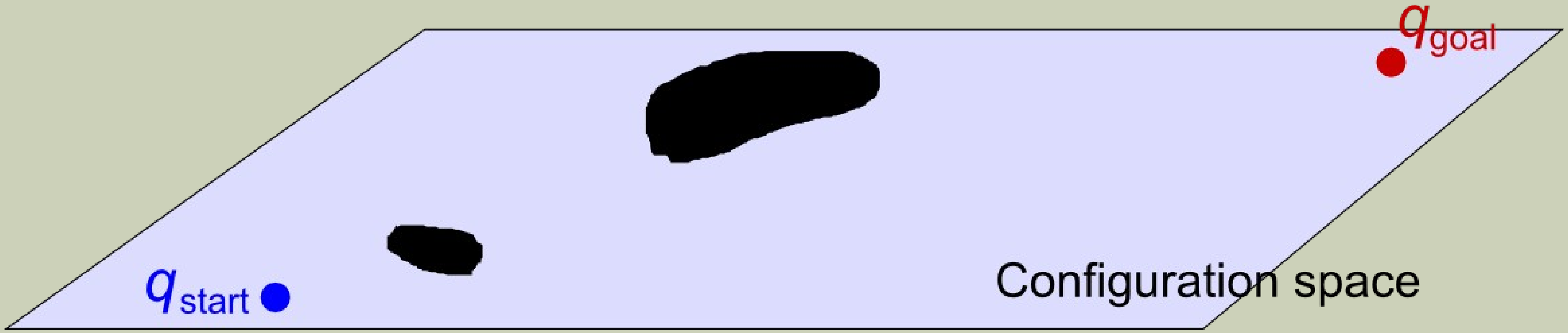
- Applicable to many types of problems
  - Velocity / acceleration / torque bounds
  - Grip stability / friction constraints
  - ZMP constraints

Time-Optimal Path Parameterization (TOPP)

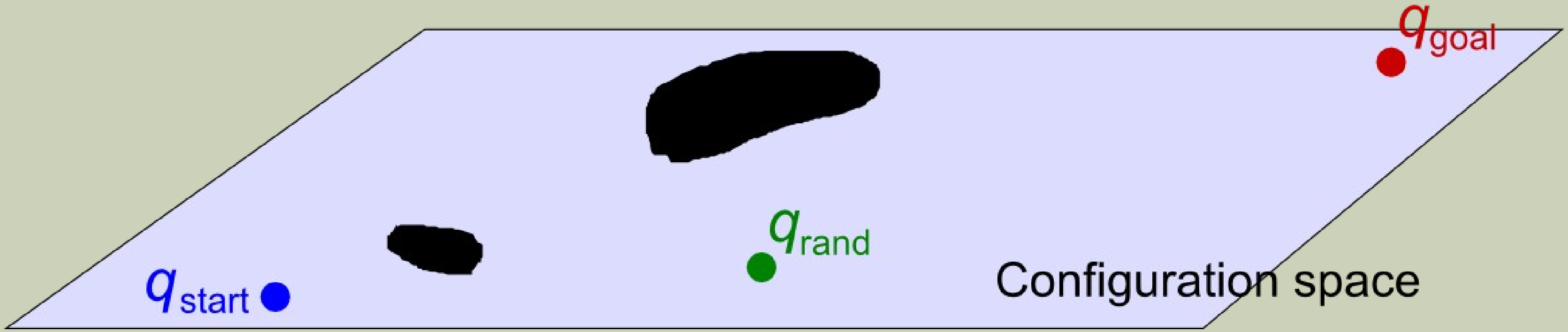
- Our implementation of Bobrow algorithm
  - https://github.com/quangounet/TOPP
  - Fast (torque constraints 7 DOF, 1s, 100 points : 6ms)
  - Integrated with OpenRAVE

- Currently supported constraints
  - Velocity / acceleration / torque bounds
  - Friction constraints
  - ZMP constraints

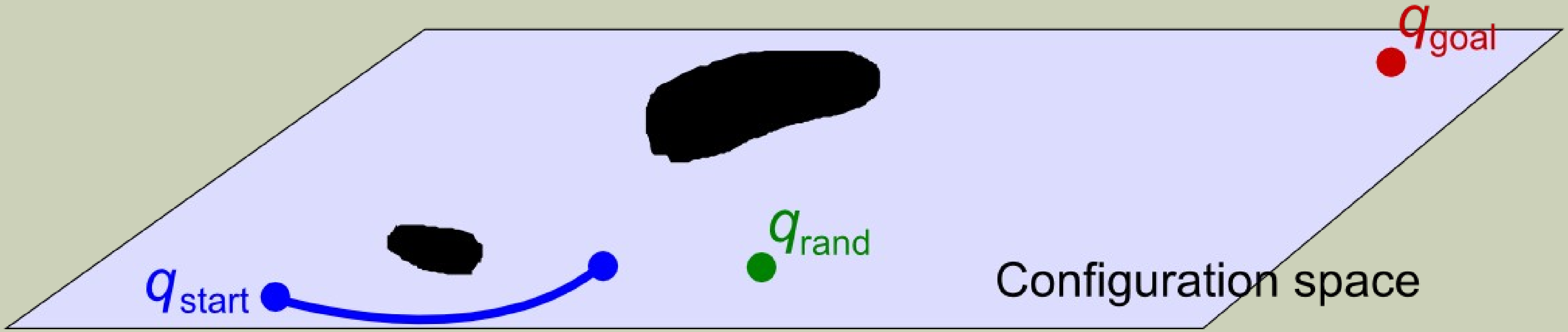
Sampling-based algorithm



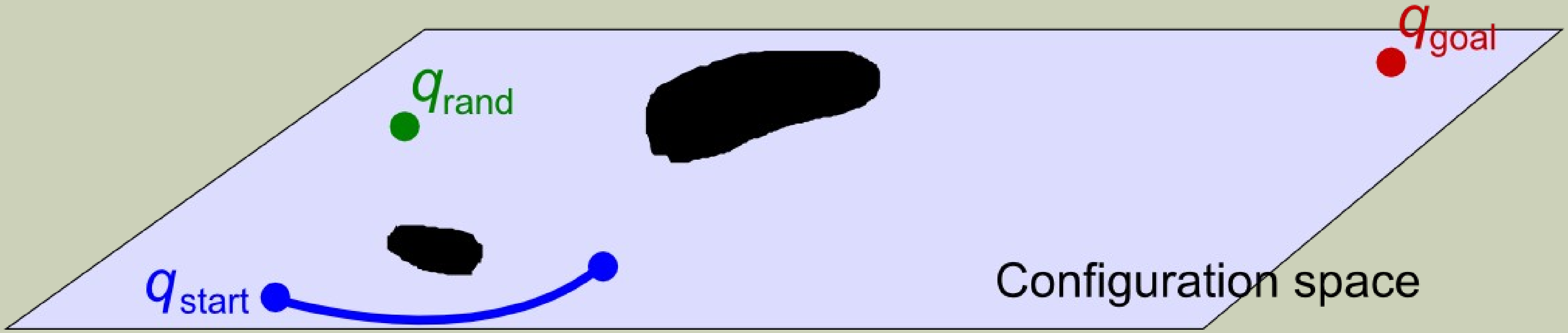
Sampling-based algorithm



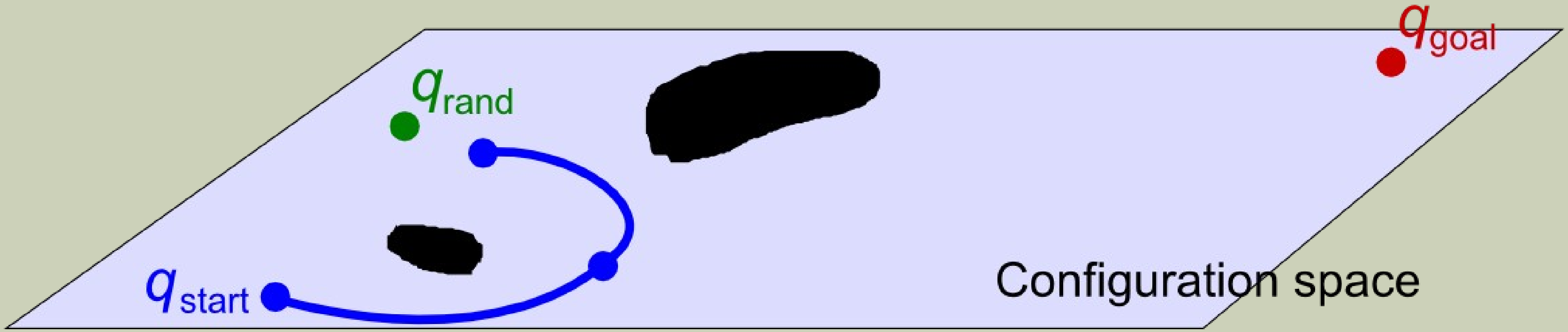
Sampling-based algorithm



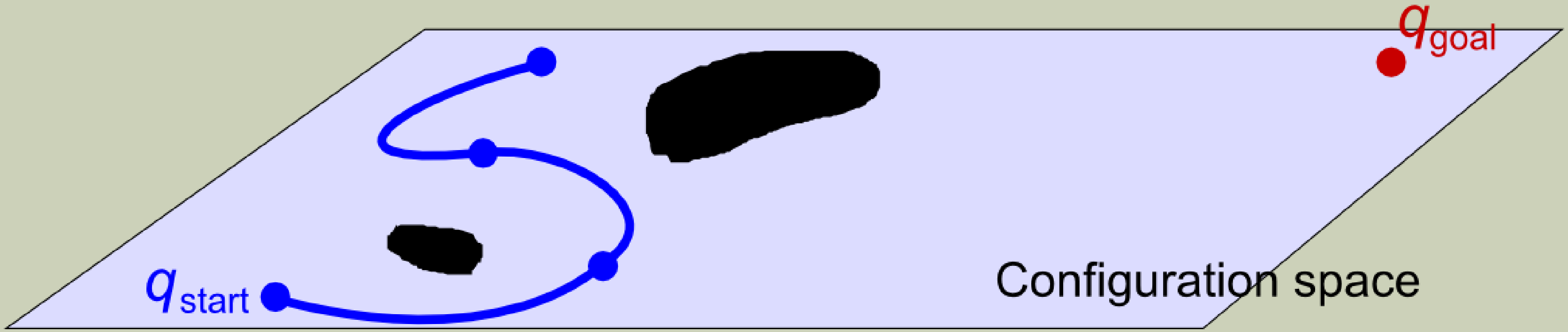
Sampling-based algorithm



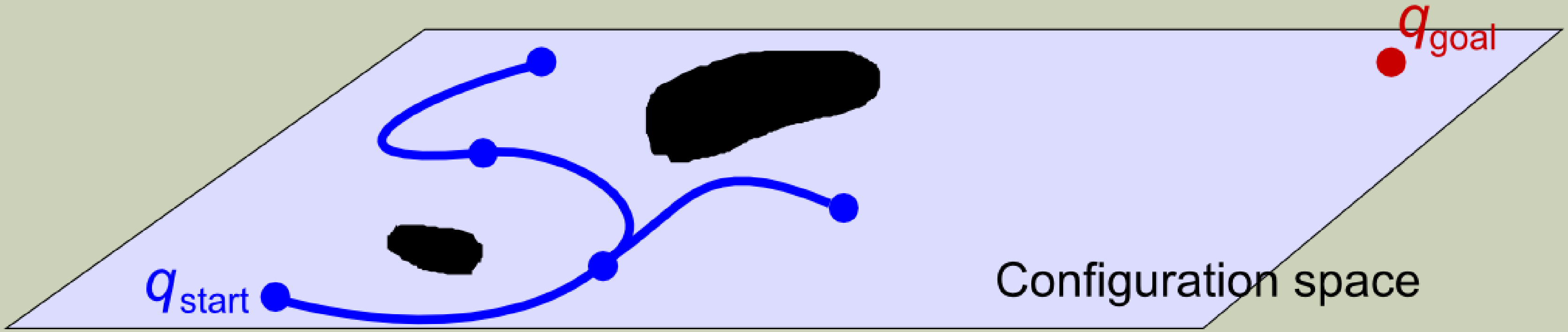
Sampling-based algorithm



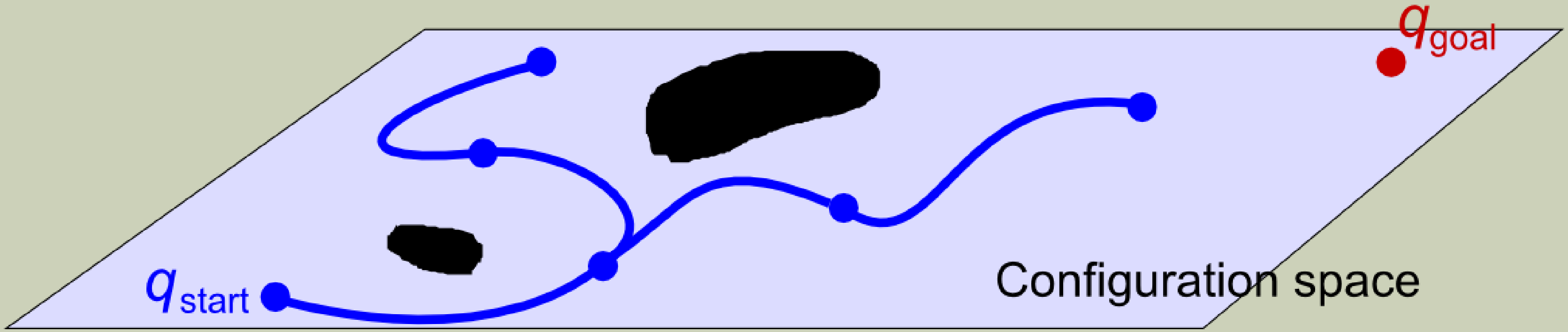
Sampling-based algorithm



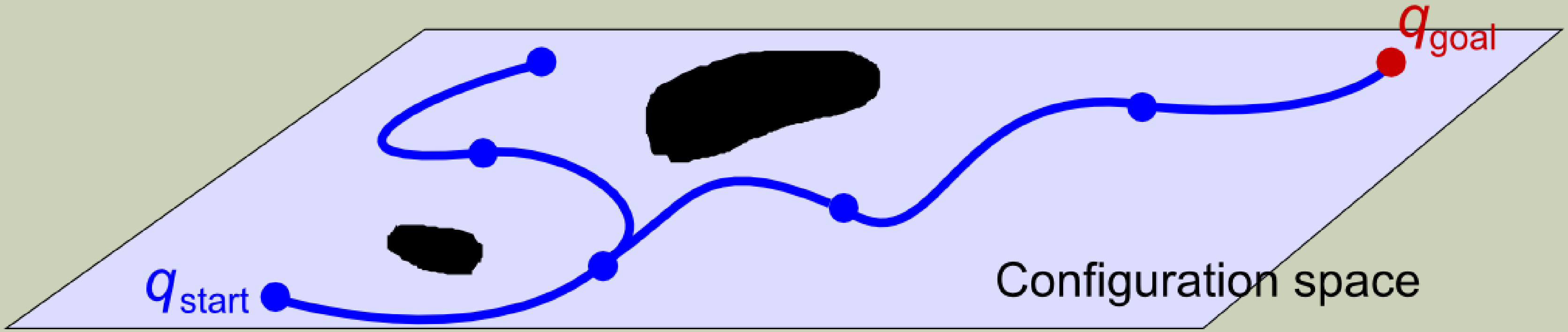
Sampling-based algorithm



Sampling-based algorithm

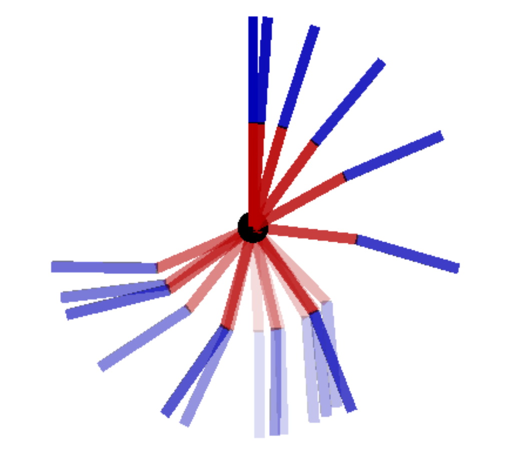


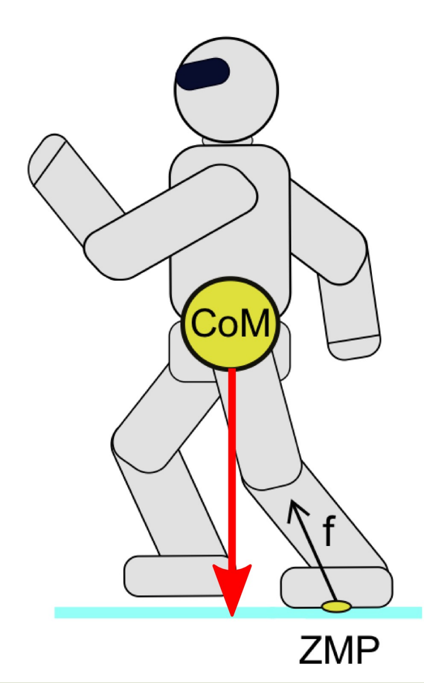
Sampling-based algorithm



Quasi-static planning

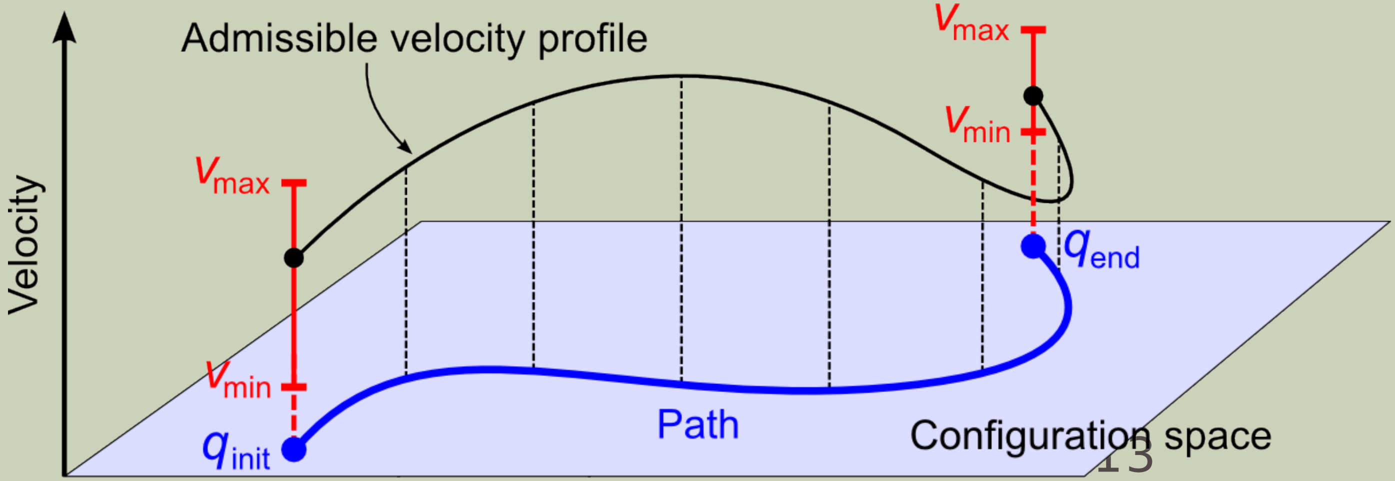
- Final path not parameterizable ?
- Check quasi-static feasibility at each step
- Loss of completeness / optimality





Admissible Velocity Propagation (AVP)

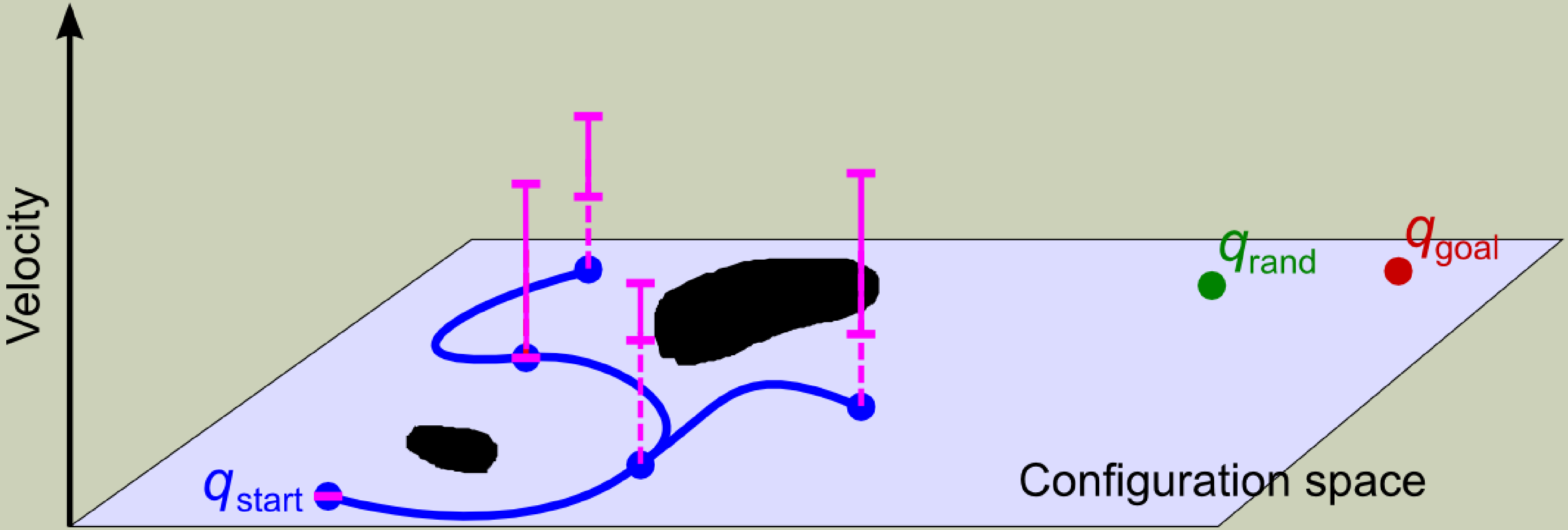
- Inputs
  - Path in configuration space
  - (*v*min,*v*max) at the beginning of the path
- Output
  - Admissible (*v*min,*v*max) at the end of the path



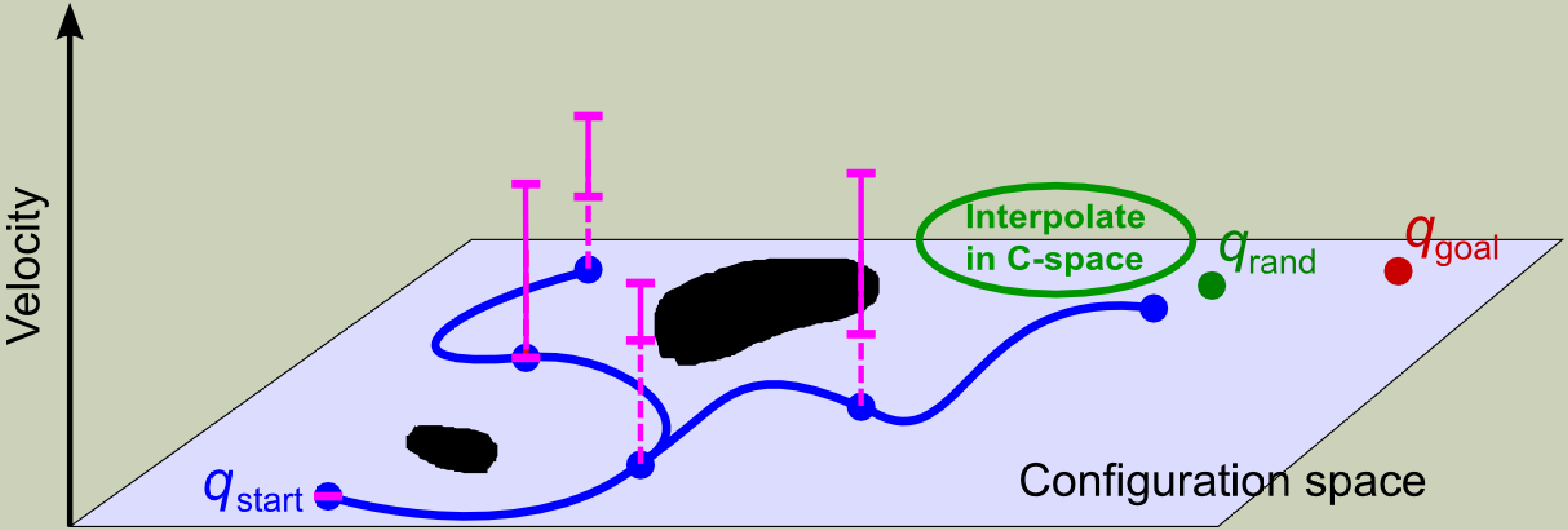
Admissible Velocity Propagation (AVP)

- Based on Bobrow algorithm
- Implemented in TOPP

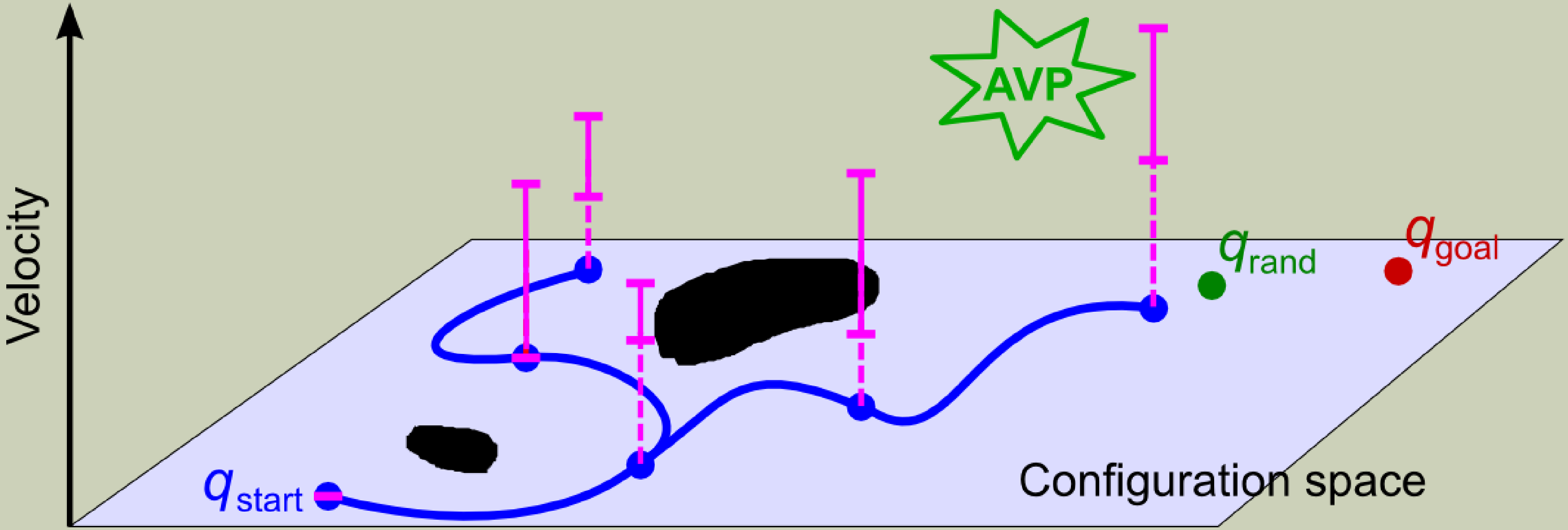
Planning using AVP



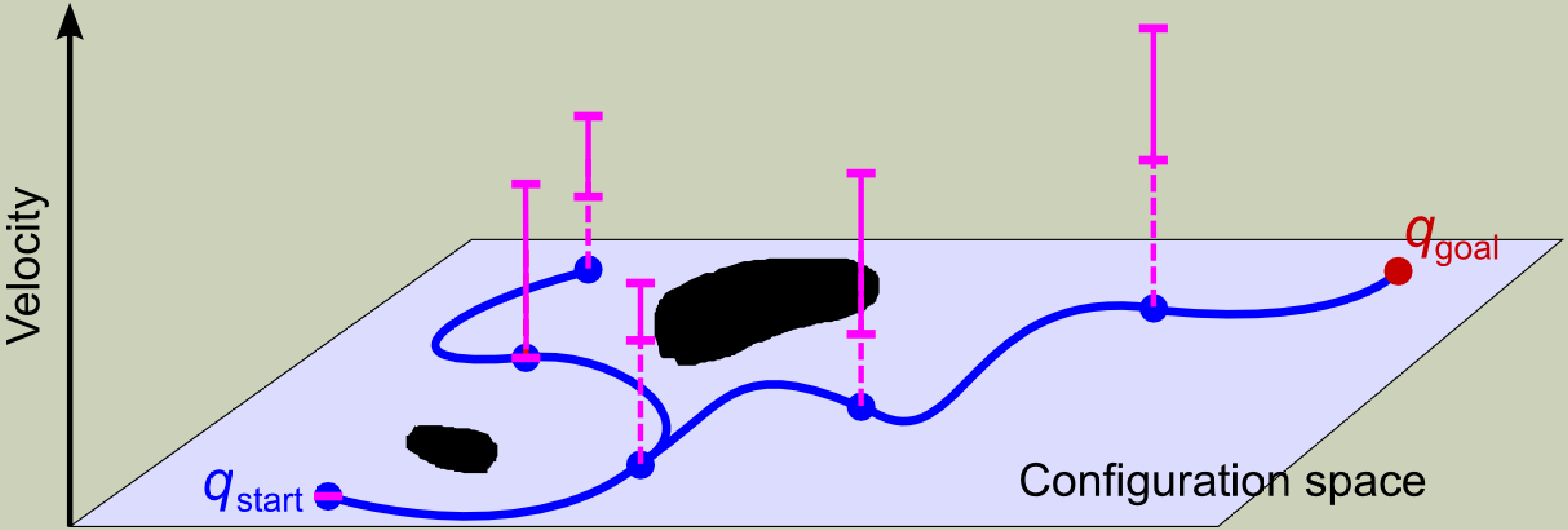
Planning using AVP



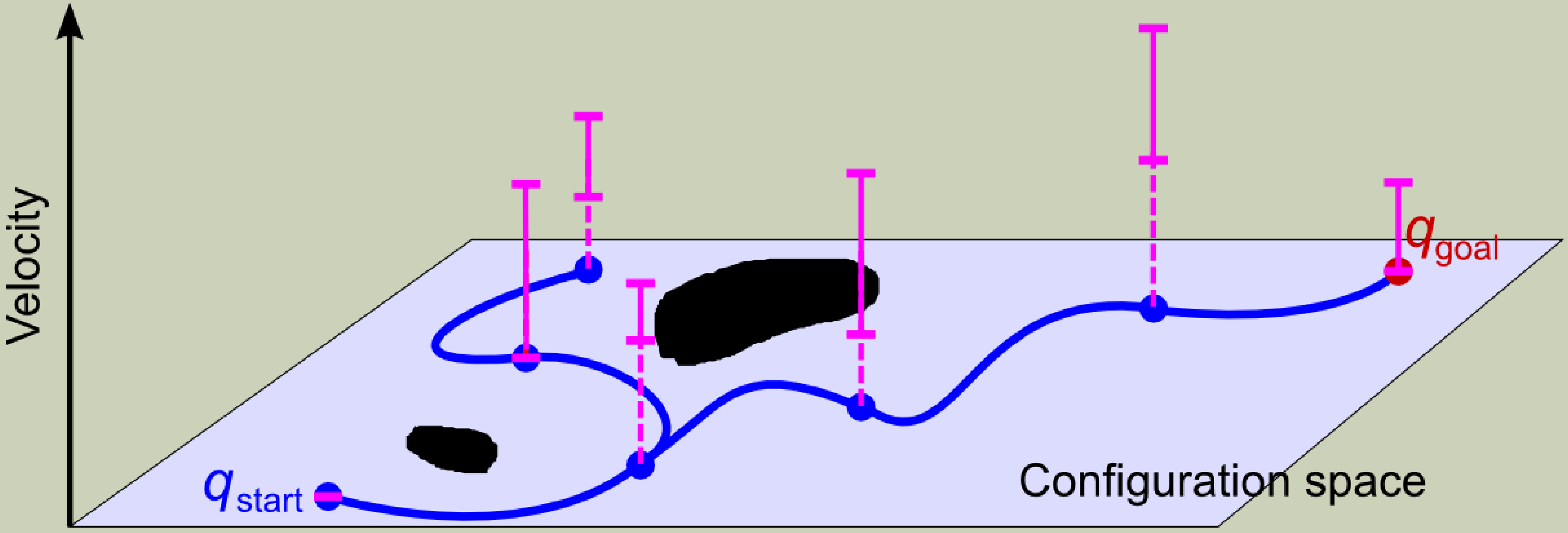
Planning using AVP



Planning using AVP

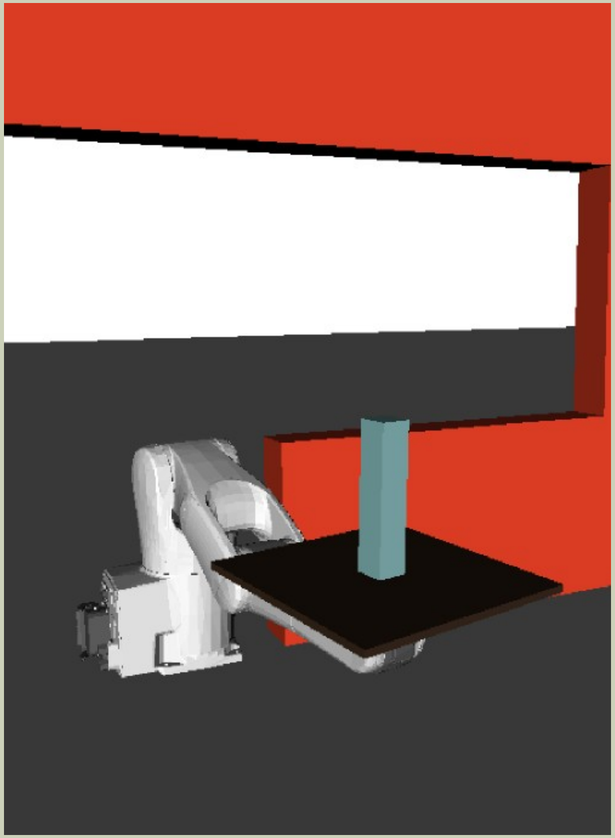


Planning using AVP



Example : Non-prehensile transportation

- No need to design specific grippers
- Save time on grasp/release
- Use friction



Conclusion

- Approach to integrate dynamics into motion planning
- Can be built upon existing sampling-based planners
- Negligible overhead over quasi-static planning
- Source code available https://github.com/quangounet/TOPP
- Current work
  - Liquid transportation
  - Humanoid robot
  - Integrate with other platforms (ROS/MoveIt!...)