Numerical Experiments (Piecewise-Linear Motion Planning amidst Static, Moving, or Morphing Obstacles)

1 MMP vs NLP vs RRT

Setup. In each dimension $n \in \{2, \ldots, 4\}$, we generate 10 motion planning problems where the path is constrained to live in the unit box $B = [-1, 1]^n$ and must avoid 10 static or dynamic spherical obstacles. More precisely, each motion planning problem is given by data $\mathcal{D} = (\boldsymbol{x}_0, \boldsymbol{x}_T, \{g_1, \ldots, g_m\})$, where $\boldsymbol{x}_0 = (-1, \ldots, -1) \in \mathbb{R}^n, \boldsymbol{x}_T = (1, \ldots, 1) \in \mathbb{R}^n$, $i \in [n]$, $g_i(\boldsymbol{x}) = 1 - x_i$, $g_{i+n}(t, \boldsymbol{x}) = 1 + x_i$ for $i \in [n]$, $g_{2n+k}(t, \boldsymbol{x}) = \|\boldsymbol{x} - (\boldsymbol{c}_k + t\boldsymbol{v}_k)\|^2 - \left(\frac{2}{10}\right)^2$. The centers $\boldsymbol{c}_k$ are sampled uniformly at random from B, the velocities $\boldsymbol{v}_k$ are either identically zero in the static case, or sampled uniformly from B in the dynamic case. See figure 2 for an example of this setup in dimension $n = 2$.

Comparison. In table 1, we compare our MMP solver (with $r = 2$, $N = 20$, and $\lambda = 0.1$) against a classical sampling-based technique (RRT) and basic nonlinear programming baseline (NLP) implemented with the help of the KNITRO.jl package [**?**]. MMP consistently achieves higher success rates, significantly shorter and smoother trajectories (the smoothness of a path $\boldsymbol{x}(t)$ is given by $\int_0^T \left(\dot{\boldsymbol{x}}(t) - \int_0^T \dot{\boldsymbol{x}}(s)\mathrm{d}s\right)^2 \mathrm{d}t$). The solve times are higher but remain highly practical.

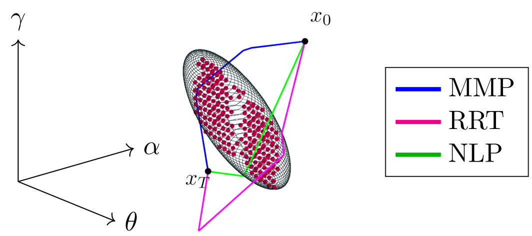

Figure 1: Plot of the paths found by RRT, NLP, and MMP (proposed) for the bimanual planning manipulation task of figure 3. The red dots depict values of the joint angles (α, β, γ) that would make the two arms collide. The black wireframe depicts the smallest enclosing ellipsoid containing the red dots.

2 Bimanual Manipulation

As a proof of concept, we also consider a bimanual manipulation task requiring two two-link arms working collaboratively to go from an initial to goal configuration without colliding (see figure 3). For visualization, we restrict attention to planar manipulation problems involving planning in a configuration space of 3 joint angles. First, we lift the obstacle set in cartesian space to joint angle space by evaluating all collision configurations on a grid over the 3 joint angles. We then fit an enclosing ellipsoid (see figure 1) to obtain a semialgebraic description of the obstacle region in configuraton space. RRT finds a feasible path, but produces a jerky path involving an unnecessary recoiling of one of the arms. NLP fails to find a feasible path at all. MMP, with 3 segments, succeeds in finding the shortest and smoothest path.

n	Methods	Static Obstacles				Dynamic Obstacles			
		success rate	length	smoothness	solve time	success rate	length	smoothness	solve time
2	RRT	40%	3.73	0.06	0.02	50%	3.59	0.12	0.06
	NLP	0%	nan	nan	nan	20%	3.4	0.06	0.06
	MMP	**60%**	**3.0**	**0.03**	0.43	**50 %**	**2.85**	**0.03**	0.43
3	RRT	50%	5.74	0.13	0.12	40%	4.82	0.1	0.23
	NLP	70%	3.44	0.05	0.12	60%	3.48	0.06	0.17
	MMP	**100%**	3.5	**0.04**	0.47	**100%**	3.55	**0.04**	0.47
4	RRT	60%	7.67	0.15	1.43	0%	nan	nan	nan
	NLP	80%	3.99	0.08	0.25	90%	4.34	0.1	0.2
	MMP	**100%**	4.11	**0.05**	0.55	**100%**	**4.1**	**0.05**	0.55

Table 1: Average success, smoothness, and solve-time comparison of RRT, NLP and MMP (proposed) methods over 10 static and dynamic motion planning problems.

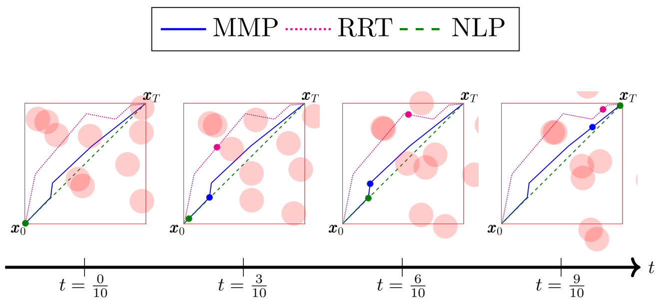

Figure 2: Typical paths obtained for a motion planning problem in 2D. The path is constrained to be in the black box and should avoid the moving obstacles. Each obstacle is a sphere that moves with constant velocity (depicted by a red arrow).

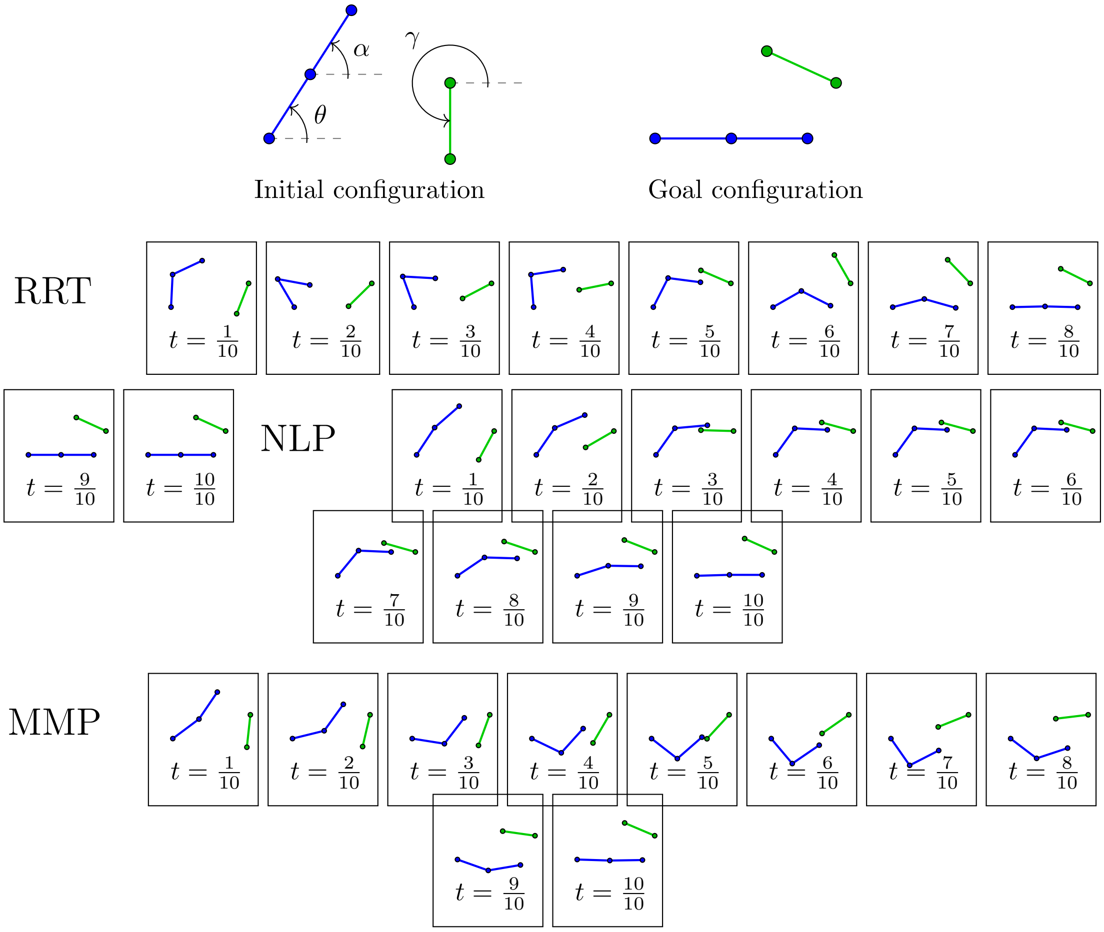

Figure 3: Performance comparison of RRT, NLP, and MMP (proposed) methods on a bimanual planar manipulation task.