Path length optimization

Florent Lamiraux

August 16, 2015

1 Definitions and Notation

1.1 Robot

We consider a robot composed of p joints. Each joint i has n_i degrees of freedom.

1.2 Path

A path is defined by a sequence of $wp+2$ waypoints:

$$P = (\mathbf{q}_0, \mathbf{q}_1, \cdots, \mathbf{q}_{wp+1}) \tag{1}$$

The path is the concatenation of straight interpolations between consecutive waypoints. The first and last waypoints are fixed. The optimization state variable is therefore defined by

$$\mathbf{x} = (\mathbf{q}_1, \cdots, \mathbf{q}_{wp}) \tag{2}$$

2 Cost

The optimization cost is defined by the following sum

$$C(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^{wp+1} \lambda_{i-1}(\mathbf{q}_i - \mathbf{q}_{i-1})^T W^2 (\mathbf{q}_i - \mathbf{q}_{i-1}) \tag{3}$$

where

$$\lambda_{i-1} = \frac{1}{\sqrt{(\mathbf{q}_{i\,0} - \mathbf{q}_{i-1\,0})^T W^2 (\mathbf{q}_{i\,0} - \mathbf{q}_{i-1\,0})}}$$

are constant coefficient aiming at keeping the same ratio between path segment lengths at minimum as at initial path. W is a block-diagonal matrix of weights:

$$W = \begin{pmatrix} w_1 I_{n_1} & 0 & \cdots & \cdots & 0 \\ 0 & w_2 I_{n_2} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \cdots \\ 0 & \cdots & 0 & w_{p-1} I_{n_{p-1}} & 0 \\ 0 & \cdots & \ddots & 0 & w_p I_{n_p} \end{pmatrix} \tag{4}$$

where each w_i, $1 \leq i \leq p$ is the weight associated to joint i.

2.1 Gradient

$$
\begin{aligned}
dC(\mathbf{x}) &= \sum_{i=1}^{wp+1} \lambda_{i-1}(\mathbf{q}_i - \mathbf{q}_{i-1})^T W^2 (d\mathbf{q}_i - d\mathbf{q}_{i-1}) \\
&= \lambda_0(\mathbf{q}_1 - \mathbf{q}_0)^T W^2 (d\mathbf{q}_1 - d\mathbf{q}_0) \\
&\quad +\lambda_1(\mathbf{q}_2 - \mathbf{q}_1)^T W^2 (d\mathbf{q}_2 - d\mathbf{q}_1) \\
&\quad \vdots \\
&\quad +\lambda_{wp}(\mathbf{q}_{wp+1} - \mathbf{q}_{wp})^T W^2 (d\mathbf{q}_{wp+1} - d\mathbf{q}_{wp}) \\
&= +\left(\lambda_0(\mathbf{q}_1 - \mathbf{q}_0) - \lambda_1(\mathbf{q}_2 - \mathbf{q}_1)\right)^T W^2 d\mathbf{q}_1 \\
&\quad +\left(\lambda_1(\mathbf{q}_2 - \mathbf{q}_1) - \lambda_2(\mathbf{q}_3 - \mathbf{q}_2)\right)^T W^2 d\mathbf{q}_2 \\
&\quad +\vdots \\
&\quad +\left(\lambda_{wp-1}(\mathbf{q}_{wp} - \mathbf{q}_{wp-1}) - \lambda_{wp}(\mathbf{q}_{wp+1} - \mathbf{q}_{wp})\right)^T W^2 d\mathbf{q}_{wp}
\end{aligned}
$$

$$
\nabla C(\mathbf{x}) = \left((\lambda_i(\mathbf{q}_{i+1} - \mathbf{q}_i)^T - \lambda_{i+1}(\mathbf{q}_{i+2} - \mathbf{q}_{i+1})^T) W^2\right)_{i=0\cdots wp-1} \tag{5}
$$

2.1.1 Computation

---

procedure ComputeGradient(P)
  $u_1 \leftarrow (\mathbf{q}_1 - \mathbf{q}_0)^T W^2$
  for $i = 0$ to $wp - 2$ **do**
    $u_2 \leftarrow (\mathbf{q}_{i+2} - \mathbf{q}_{i+1})^T W^2$
    $gradient[i\, n_{dof} : (i+1)\, n_{dof}] \leftarrow \lambda_i u_1 - \lambda_{i+1} u_2$
    $u_1 \leftarrow u_2$
  end for
  $u_2 \leftarrow (\mathbf{q}_{wp+1} - \mathbf{q}_{wp})^T W^2$
  $gradient[(wp-1)\, n_{dof} : wp\, n_{dof}] \leftarrow \lambda_{wp-1} u_1 - \lambda_{wp} u_2$
end procedure

2.2 Hessian

$$
\text{Hess } C(\mathbf{x}) = \begin{pmatrix}
(\lambda_0 + \lambda_1)W^2 & -\lambda_1 W^2 & 0 & \cdots & & 0 \\
-\lambda_1 W^2 & (\lambda_1 + \lambda_2)W^2 & -\lambda_2 W^2 & 0 & \cdots & 0 \\
0 & -\lambda_2 W^2 & (\lambda_2 + \lambda_3)W^2 & -\lambda_3 W^2 & 0 & \vdots \\
\vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\
\vdots & & \ddots & \ddots & \ddots & \vdots \\
0 & \cdots & 0 & -\lambda_{wp-2}W^2 & (\lambda_{wp-2} + \lambda_{wp-1})W^2 & -\lambda_{wp-1}W^2 \\
0 & \cdots & \cdots & 0 & -\lambda_{wp-1}W^2 & (\lambda_{wp-1} + \lambda_{wp})W^2
\end{pmatrix} \tag{6}
$$