



HAL
open science

Computational details for : "Differential Dynamic Programming for Maximum a Posteriori Centroidal State Estimation of Legged Robots"

François Bailly, Justin Carpentier, Philippe Souères

► To cite this version:

François Bailly, Justin Carpentier, Philippe Souères. Computational details for : "Differential Dynamic Programming for Maximum a Posteriori Centroidal State Estimation of Legged Robots". [Research Report] LAAS-CNRS; Université de Montréal. 2021. hal-03180052v1

HAL Id: hal-03180052

<https://hal.science/hal-03180052v1>

Submitted on 24 Mar 2021 (v1), last revised 25 Mar 2021 (v4)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Computational details for : “Differential Dynamic Programming for Maximum a Posteriori Centroidal State Estimation of Legged Robots”

François Bailly^{a,b,*}, Thomas Flayols^a, Justin Carpentier^c and Philippe Souères^a

This document complements the paper entitled “Differential Dynamic Programming for Maximum a Posteriori Centroidal State Estimation of Legged Robots” [1]. The purpose of this work was to estimate the centroidal dynamics of legged robots by formulating a maximum a posteriori problem and solving it thanks to differential dynamic programming (DDP). In the following, the computations of the partial derivatives of the unoptimized value function (Q_k) are provided for the DDP algorithm.

APPENDIX I

PARTIAL DERIVATIVES OF Q_k

- $Q_{xk} = \nabla_{\mathbf{x}_k} l_k + \nabla_{\mathbf{x}_k} \mathcal{V}_{k+1}(f(\mathbf{x}_k, \boldsymbol{\omega}_k)),$
 $\nabla_{\mathbf{x}_k} l_k = 2 \frac{\partial(g(\mathbf{x}_k) - \mathbf{y}_k)^T}{\partial \mathbf{x}_k} \Sigma_{\boldsymbol{\eta}_k}^{-1} (g(\mathbf{x}_k) - \mathbf{y}_k),$
 $\frac{\partial g(\mathbf{x}_k)}{\partial \mathbf{x}_k} = \frac{C(\mathbf{x}_k)}{\partial \mathbf{x}_k} \mathbf{x}_k + C(\mathbf{x}_k) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2c_k \times & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \hat{=} \tilde{C}(\mathbf{x}_k),$
 $Q_{xk} = 2\tilde{C}(\mathbf{x}_k)^T \Sigma_{\boldsymbol{\eta}_k}^{-1} (g(\mathbf{x}_k) - \mathbf{y}_k) + A^T \mathcal{V}'_x.$
- $Q_{\omega k} = \nabla_{\boldsymbol{\omega}_k} l_k + \nabla_{\boldsymbol{\omega}_k} \mathcal{V}_{i+1}(f(\mathbf{x}_k, \boldsymbol{\omega}_k)),$
 $\nabla_{\boldsymbol{\omega}_k} l_k = \frac{\partial \|\boldsymbol{\omega}_k\|_{\Sigma_{\boldsymbol{\omega}_k}^{-1}}^2}{\partial \boldsymbol{\omega}_k} = 2\Sigma_{\boldsymbol{\omega}_k}^{-1} \boldsymbol{\omega}_k,$
 $Q_{\omega k} = 2\Sigma_{\boldsymbol{\omega}_k}^{-1} \boldsymbol{\omega}_k + B^T \mathcal{V}'_x.$
- $Q_{xxk} = \nabla_{\mathbf{x}_k}^2 l_k + \nabla_{\mathbf{x}_k}^2 \mathcal{V}_{i+1}(f(\mathbf{x}_k, \boldsymbol{\omega}_k)),$

^a Laboratoire de Simulation et Modélisation du Mouvement, Faculté de Médecine, Université de Montréal, Laval, QC, Canada LAAS-CNRS, 7 Avenue du Colonel Roche, F-31400 Toulouse, France

^b LAAS-CNRS, 7 Avenue du Colonel Roche, F-31400 Toulouse, France

^c Inria, Département d’informatique de l’ENS, École normale supérieure, CNRS, PSL Research University, Paris, France

*corresponding author: francois.bailly@umontreal.ca

where, the element of $\nabla_{\mathbf{x}_k}^2 l_k$ at the i^{th} row and j^{th} column is denoted by:

$$[\nabla_{\mathbf{x}_k}^2 l_k]_{ij} = \sum_{m=1}^Y \sum_{n=1}^Y [\Sigma_{\boldsymbol{\eta}_k}^{-1}]_{mn} \left(\frac{\partial \tilde{C}_{im}^T}{\partial \mathbf{x}_j} [g(\mathbf{x}_k) - \mathbf{y}_k]_n + \tilde{C}_{im}^T \frac{\partial [g(\mathbf{x}_k)]_n}{\partial \mathbf{x}_j} \right),$$

$$[\nabla_{\mathbf{x}_k}^2 l_k]_{ij} = [\tilde{C}^T \Sigma_{\boldsymbol{\eta}_k}^{-1} \tilde{C}]_{ij}$$

$$+ \sum_{n=1}^Y \sum_{m=1}^Y \frac{\partial \tilde{C}_{km}^T}{\partial \mathbf{x}_l} [\Sigma_{\boldsymbol{\eta}_k}^{-1}]_{mn} [g(\mathbf{x}_k) - \mathbf{y}_k]_n,$$

$$[\nabla_{\mathbf{x}_k}^2 l_k]_{ij} = [\tilde{C}^T \Sigma_{\boldsymbol{\eta}_k}^{-1} \tilde{C}]_{ij} + \tilde{\mathbf{c}}_{ij}^T \Sigma_{\boldsymbol{\eta}_k}^{-1} (g(\mathbf{x}_k) - \mathbf{y}_k),$$

where, $\tilde{\mathbf{c}}_{ij} \in \mathbb{R}^Y$ is the stacked vector of $\frac{\partial \tilde{C}_{im}^T}{\partial \mathbf{x}_j}$ for $m \in [1..Y]$.

- $Q_{\omega \omega k} = \nabla_{\boldsymbol{\omega}_k}^2 l_k + \nabla_{\boldsymbol{\omega}_k}^2 \mathcal{V}_{i+1}(f(\mathbf{x}_k, \boldsymbol{\omega}_k)),$
 $\nabla_{\boldsymbol{\omega}_k}^2 l_k = 2\Sigma_{\boldsymbol{\omega}_k}^{-1} \frac{\partial \boldsymbol{\omega}_k}{\partial \boldsymbol{\omega}_k} = 2\Sigma_{\boldsymbol{\omega}_k}^{-1}$
 $Q_{\omega \omega k} = 2\Sigma_{\boldsymbol{\omega}_k}^{-1} + B^T \mathcal{V}_{xx} B$

REFERENCES

- [1] F. Bailly, T. Flayols, J. Carpentier, and P. Souères, “Differential dynamic programming for maximum a posteriori centroidal state estimation of legged robots,” in *Internat. Conf. on Robotics and Automation*. IEEE, 2021.