



Programmation quadratique pour les robots humanoïdes

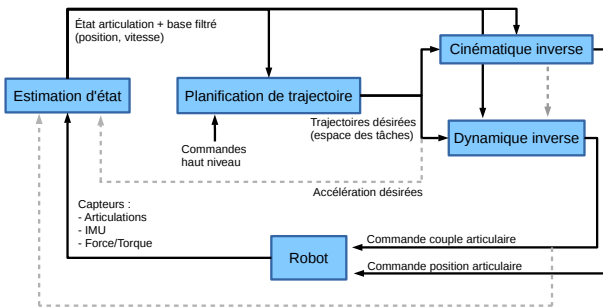
Quentin Rouxel

16 octobre 2018

“Quand on n’a qu’un marteau, tous les problèmes deviennent des clous”

Abraham Maslow

Introduction



- 1 Formulation
- 2 Cinématique inverse (IK)
- 3 Dynamique inverse (ID)
- 4 Conditions KKT et différentiation
- 5 Cinématique inverse avec contraintes de force
- 6 Contrôle à commande prédictive (MPC)
- 7 Estimation de l'état

Formulation

Problème de programmation quadratique
(QP) :

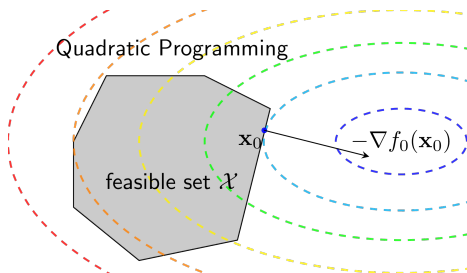
$$\underset{\mathbf{x}}{\operatorname{argmin}} \frac{1}{2} \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{q}^T \mathbf{x}$$

tel que :

$$\begin{cases} \mathbf{A} \mathbf{x} + \mathbf{b} = \mathbf{0} \\ \mathbf{C} \mathbf{x} + \mathbf{d} \geq \mathbf{0} \end{cases}$$

- \mathbf{x} : variable d'optimisation
- \mathbf{Q}, \mathbf{q} : coût quadratique
- \mathbf{A}, \mathbf{b} : contraintes d'égalités
- \mathbf{C}, \mathbf{d} : contraintes d'inégalités

- Espace de recherche : polytope convexe
- \mathbf{Q} symétrique définie positive



Coût et priorités

Souvent, coût : distance aux moindres carrés (pondérée) :

$$\|\Lambda \mathbf{x} - \beta\|_{\mathbf{W}}^2 \Rightarrow \begin{cases} \mathbf{Q} = \Lambda^T \mathbf{W} \Lambda \\ \mathbf{q} = -\Lambda^T \mathbf{W} \beta \end{cases}$$

avec

$$\mathbf{W} = \begin{bmatrix} w_1 & & & \\ & \ddots & & \\ & & \ddots & \\ & & & w_n \end{bmatrix}$$

matrice diagonale positive

Différentes tâches \Rightarrow 2 approches :

- Contraintes dures (*hard*)
 - Hiérarchie de QP (*stack-of-tasks*)
 - Résolution dans le kernel (*nullspace*)
 - Définition des priorités
- Contraintes souples (*soft*)
 - Un seul QP
 - Pondération relative
 - Choix des poids

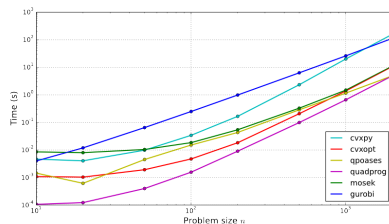
\Rightarrow Contraintes souples

Résolution

- Dense ou creux (*sparse*)
- Libre ou commercial

- CVXOPT
- QuadProg++
- Gurobi
- MOSEK
- qpOASES
- OSQP

Benchmarks :



QuadProg++ pour problème dense

⇒ EiQuadProg++ (modifié)

A numerically stable dual method for solving strictly convex quadratic programs

D. Goldfarb, A. Idnani

Mathematical programming 1983

Au niveau de la “vitesse” :

$$\mathbf{x} = [d\mathbf{q}]$$

$$\mathbf{J}(\mathbf{q}) \cdot \dot{\mathbf{q}} \stackrel{\text{def}}{=} [\omega_x \quad \omega_y \quad \omega_z \quad v_x \quad v_y \quad v_z]^T$$

Coût :

$$\mathbf{p} = \text{FK}(\mathbf{q})$$

$$\min_{d\mathbf{q}} \left\| (\mathbf{p} + \mathbf{J}(\mathbf{q})d\mathbf{q}) - \mathbf{p}^d \right\|_{\mathbf{W}}^2 + \|d\mathbf{q}\|_{\mathbf{W}}^2$$

$$\Lambda_1 = \mathbf{J}(\mathbf{q}), \quad \beta_1 = \mathbf{p}^d - \mathbf{p}$$

$$\Lambda_2 = \mathbb{I}, \quad \beta_2 = \mathbf{0}$$

(gestion spéciale des rotations)

Abus de notation (DOFs base flottante) :

$$\dot{\mathbf{q}} = [\boldsymbol{\omega}^T \quad \mathbf{v}^T \quad \dot{\boldsymbol{\theta}}^T]^T$$

Contraintes :

$$\bar{\mathbf{q}}^- \leq \mathbf{q} + d\mathbf{q} \leq \bar{\mathbf{q}}^+$$

$$-\overline{\Delta\mathbf{q}} \leq d\mathbf{q} \leq \overline{\Delta\mathbf{q}}$$

$$\overline{\mathbf{CoM}^-} \leq \mathbf{p}_{\text{CoM}} + \mathbf{J}_{\text{CoM}}(\mathbf{q}) \cdot d\mathbf{q} \leq \overline{\mathbf{CoM}^+}$$

(polytope)

Intégration de la position :

$$\mathbf{q}_{t+1} = \mathbf{q}_t + d\mathbf{q}$$

(gestion spéciale des quaternions si base flottante)

#variables= 29, #égalités= 0, #inégalités= 44, Temps \approx 0.2 ms

Dynamique inverse (1/2)

Problème :

- Couples articulaires
- Problème du double support (multi-contacts)
- Contraintes de stabilités
- Conversion espace cartésien \rightarrow articulaire
- Temps réel

Équation de la dynamique (espace articulaire) :

$$\begin{cases} \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{S}\boldsymbol{\tau} + \mathbf{J}(\mathbf{q})^T \boldsymbol{\lambda} \\ \mathbf{J}(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{J}}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = \mathbf{a} \end{cases}$$

Optimization based full body control for the atlas robot

S. Feng, E. Whitman, X. Xinjilefu, C. G. Atkeson

Humanoids 2014

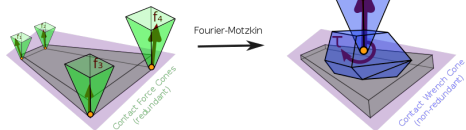
Contraintes de faisabilité et stabilité

Couples articulaires :

$$-\bar{\tau} \leq \tau \leq \bar{\tau}$$

Torseur (*wrench*) de contact :

$$\lambda = [m_x \quad m_y \quad m_z \quad f_x \quad f_y \quad f_z]^T$$



Stability of surface contacts for humanoid robots : Closed-form formulae of the contact wrench cone for rectangular support areas

S. Caron, Q. C. Pham, Y. Nakamura
ICRA 2015

Unilatéralité :

$$f_z \geq 0$$

Centre de pression (si contact plan) :

$$\overline{CoP}^- \leq \begin{bmatrix} -m_y \\ \frac{f_x}{f_z} \\ \frac{m_x}{f_z} \end{bmatrix} \leq \overline{CoP}^+$$

Cone de friction linéarisé (pyramide) :

$$-\mu \begin{bmatrix} f_x \\ f_y \end{bmatrix} \leq \begin{bmatrix} f_x \\ f_y \end{bmatrix} \leq \mu \begin{bmatrix} f_x \\ f_y \end{bmatrix}$$

Moment en rotation (Stéphane Caron) :

...

Dynamique inverse (2/2)

Variable d'optimisation :

$$\mathbf{x} = [\ddot{\mathbf{q}}^T \quad \boldsymbol{\tau}^T \quad \boldsymbol{\lambda}^T]^T$$

Coût :

$$\min_{\ddot{\mathbf{q}}} \left\| \mathbf{J}(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{J}}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{q} - \mathbf{a}^* \right\|_{\mathbf{W}}^2$$

$$\boldsymbol{\Lambda} = [\mathbf{J}(\mathbf{q}) \quad \mathbf{0} \quad \mathbf{0}], \quad \boldsymbol{\beta} = \mathbf{a}^* - \dot{\mathbf{J}}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{q}$$

Autres objectifs :

- Régularisation
- Contrôle des centres de pression
- Contrôle de la répartition des forces de contacts (f_z)
- Contrôle de la quantité de mouvement du centre de masse

Contraintes d'égalités :

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{S}\boldsymbol{\tau} + \mathbf{J}(\mathbf{q})^T \boldsymbol{\lambda}$$

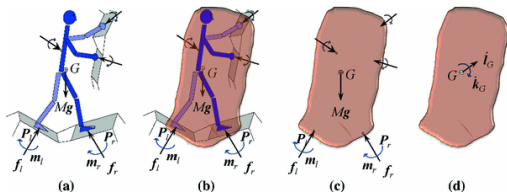
Contraintes d'inégalités :

- Bornes sur couples articulaires
- Critères de stabilité sur les torseurs de contact

Contrôle en accélération :

$$\mathbf{a}^* = \mathbf{K}_p(\mathbf{p}^d - \mathbf{p}) + \mathbf{K}_d(\mathbf{v}^d - \mathbf{v}) + \mathbf{a}^d$$

Dynamique centroïdale



$$h_G(\mathbf{q}) \stackrel{\text{def}}{=} \mathbf{A}_G(\mathbf{q})\dot{\mathbf{q}}$$

$$\stackrel{\text{def}}{=} \mathbf{I}_G(\mathbf{q}) \begin{bmatrix} \omega_x^G \\ \omega_y^G \\ \omega_z^G \\ v_x^G \\ v_y^G \\ v_z^G \end{bmatrix}$$

Improved computation of the humanoid centroidal dynamics and application for whole-body control

P. M. Wensing, P. M., D. E. Orin

International Journal of Humanoid Robotics 2016

$$\Rightarrow \mathbf{A}_G(\mathbf{q}), \dot{\mathbf{A}}_G(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$$

$$\dot{h}_G(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{A}_G(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{A}}_G(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$$

$$= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -mg \end{bmatrix} + \sum_{i=0}^N {}^G \mathbf{X}_i(\mathbf{q})\lambda_i$$

#variables= 35, #égalités= 29, #inégalités= 80, Temps \approx 0.6 ms

Remarques :

- État cinématique complet : position et vitesse articulations et base
- Stabilité instantanée
- \Rightarrow Accélérations cibles doivent être réalisables

Télé-opération pour la manipulation

Projet en cours → télé-opération :

- Robot humanoïde ⇒ contraintes de stabilité
- Terrain non plat
- Multi-contacts
- Robuste aux erreurs de l'opérateur

⇒ Cinématique inverse avec contraintes sur les forces



Architecture : IK → ID → robot

Conditions de Karush–Kuhn–Tucker (KKT) (1/2)

QP :

$$\underset{x}{\operatorname{argmin}} \frac{1}{2} \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{q}^T \mathbf{x}$$

$$\begin{cases} \mathbf{A} \mathbf{x} + \mathbf{b} = \mathbf{0} \\ \mathbf{C} \mathbf{x} + \mathbf{d} \geq \mathbf{0} \end{cases}$$

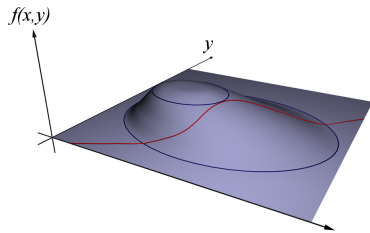
Lagrangien :

$$L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = \frac{1}{2} \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{q}^T \mathbf{x} - \boldsymbol{\lambda} (\mathbf{A} \mathbf{x} + \mathbf{b}) - \boldsymbol{\nu} (\mathbf{C} \mathbf{x} + \mathbf{d})$$

Conditions stationnaires de KKT :

$$\begin{cases} \mathbf{Q} \mathbf{x}^* + \mathbf{q} + \mathbf{A}^T \boldsymbol{\lambda}^* + \mathbf{C}^T \boldsymbol{\nu}^* = \mathbf{0} \\ \mathbf{A} \mathbf{x}^* + \mathbf{b} = \mathbf{0} \\ \operatorname{diag}(\boldsymbol{\nu}^*) (\mathbf{C} \mathbf{x}^* + \mathbf{d}) = \mathbf{0} \\ \boldsymbol{\nu}^* \geq \mathbf{0} \end{cases}$$

- $\boldsymbol{\lambda}$: lagrangien des contraintes d'égalités
- $\boldsymbol{\nu}$: lagrangien des contraintes d'inégalités
- $\nu_i \neq 0$ pour contrainte active



Conditions de Karush–Kuhn–Tucker (KKT) (2/2)

Différentiation puis résolution d'un système linéaire :

$$\Rightarrow \partial \mathbf{x}^*$$

sachant :

$$(\mathbf{Q}, \mathbf{A}, \mathbf{C}, \mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\nu}^*, \partial \mathbf{Q}, \partial \mathbf{q}, \partial \mathbf{A}, \partial \mathbf{b}, \partial \mathbf{C}, \partial \mathbf{d})$$

$\Rightarrow \boldsymbol{\lambda}^*$ et $\boldsymbol{\nu}^*$ fournis par EiQuadProg++ (après modification)

Optnet : Differentiable optimization as a layer in neural networks

B. Amos, J. Z. Kolter

International Conference on Machine Learning 2017

Dynamique inverse statique

QP : Coût et variables :

$$\min_x \|\boldsymbol{\tau}\|_{\mathbf{W}}^2 \quad \text{avec} \quad \mathbf{x} = \begin{bmatrix} \boldsymbol{\tau} \\ \boldsymbol{\lambda}_l \\ \boldsymbol{\lambda}_r \end{bmatrix}$$

Contraintes d'égalités :

$$\mathbf{g}(\mathbf{q}) = \mathbf{S}\boldsymbol{\tau} + \mathbf{J}_l(\mathbf{q})^\top \boldsymbol{\lambda}_l + \mathbf{J}_r(\mathbf{q})^\top \boldsymbol{\lambda}_r$$

Contraintes d'inégalité :

- Bornes sur couples articulaires
- Critères de stabilité sur les torseurs de contact

Différentiation des conditions KKT \Rightarrow

$$\frac{\partial \begin{bmatrix} \boldsymbol{\tau} \\ \boldsymbol{\lambda}_l \\ \boldsymbol{\lambda}_r \end{bmatrix}}{\partial \mathbf{q}}(\mathbf{q})$$

Besoin des dérivées partielles (matrice et tenseurs) :

$$\frac{\partial \mathbf{g}}{\partial \mathbf{q}}(\mathbf{q}), \quad \frac{\partial \mathbf{J}_l}{\partial \mathbf{q}}(\mathbf{q}), \quad \frac{\partial \mathbf{J}_r}{\partial \mathbf{q}}(\mathbf{q}),$$

Analytical derivatives of rigid body dynamics algorithms

J. Carpentier, N. Mansard
RSS 2018

IK et contraintes statique de faisabilité

- 1 Calcul torques/forces statiques avec ID statique QP
- 2 Calcul des dérivées partielles (gravité articulaire et hessiennes)
- 3 Différentiation de la solution avec KKT
- 4 IK avec QP incluant la linéarisation local des contraintes de stabilité

Contraintes d'inégalités :

- Bornes sur les positions et vitesses articulaires
- Bornes sur couples articulaires
- Critères de stabilité sur les torseurs de contact

$$\dots \leq \begin{bmatrix} \tau \\ \lambda_l \\ \lambda_r \end{bmatrix}_{t+1} = \begin{bmatrix} \tau \\ \lambda_l \\ \lambda_r \end{bmatrix}_t + \frac{\partial \begin{bmatrix} \tau \\ \lambda_l \\ \lambda_r \end{bmatrix}}{\partial \mathbf{q}}(\mathbf{q}) \cdot d\mathbf{q}_{t+1} \leq \dots$$

Cinématique inverse et contraintes – simulation géométrique

Temps : ≤ 2 ms

Contrôle à commande prédictive (MPC)

Système dynamique linéaire (discret) :

$$\mathbf{s}_{k+1} = \mathbf{A}\mathbf{s}_k + \mathbf{B}\mathbf{u}_k$$

$$\begin{bmatrix} \mathbf{s}_1 \\ \vdots \\ \mathbf{s}_n \end{bmatrix} = \mathbf{S} \begin{bmatrix} \mathbf{u}_0 \\ \vdots \\ \mathbf{u}_{n-1} \end{bmatrix} + \mathbf{T}\mathbf{s}_0$$

$$\mathbf{u} = [\mathbf{u}_0^T \quad \dots \quad \mathbf{u}_{n-1}^T]^T$$

$$\min_{\mathbf{u}} \left\| \begin{bmatrix} \mathbf{s}_1 \\ \vdots \\ \mathbf{s}_n \end{bmatrix} - \begin{bmatrix} \mathbf{s}_1^d \\ \vdots \\ \mathbf{s}_n^d \end{bmatrix} \right\|_{\mathbf{W}}^2 + \|\mathbf{u}\|_{\mathbf{W}}^2$$

Si pas de contrainte \Rightarrow LQR

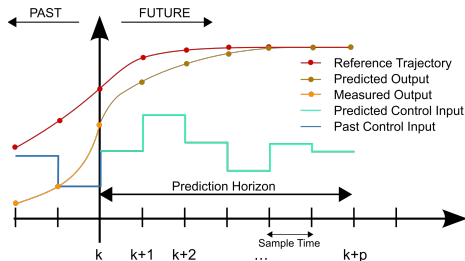
Contraintes (convexes) :

$$\bar{\mathbf{u}}^- \leq \mathbf{u} \leq \bar{\mathbf{u}}^+$$

$$\bar{\Delta \mathbf{s}}^- \leq (\mathbf{s}_k - \mathbf{s}_{k-1}) \leq \bar{\Delta \mathbf{s}}^+$$

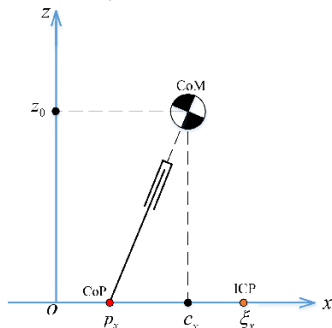
$$\bar{\Delta \mathbf{u}}^- \leq (\mathbf{u}_k - \mathbf{u}_{k-1}) \leq \bar{\Delta \mathbf{u}}^+$$

...



Application : contrôle du centre de mass

Modèle du pendule inversé linéaire



$$\ddot{x} = \frac{g}{z_0}(x - p_x)$$

$$\frac{d}{dt} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \\ p \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \\ p \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -\frac{z_0}{g} \\ 1 \end{bmatrix} \dot{p}$$

An improved formulation for model predictive control of legged robots for gait planning and feedback control

control

K. Yuan, Z. Li
IROS 2018

Estimation de l'état

Dynamic state estimation using quadratic programming

X. Xinjilefu, S. Feng, C. G. Atkeson
IROS 2014

- Alternative au filtre de Kalman étendu (EKF)
- Contraintes cinématiques de contacts

$$z = \begin{bmatrix} \mathbf{x} \\ \mathbf{w} \end{bmatrix}$$

$$\min_z \underbrace{\|\mathbf{w}\|_{Q_e}^2}_{\text{erreur de modélisation}} + \underbrace{\|\mathbf{Y}\mathbf{x} - \mathbf{y}^*\|_{Q_m}^2}_{\text{erreur de mesure}}$$

avec :

$$\text{modèle}(\mathbf{x}) = \mathbf{w}$$

