Impact-aware humanoid robot motion generation with a quadratic optimization controller

Yuquan Wang

CNRS-UM LIRMM

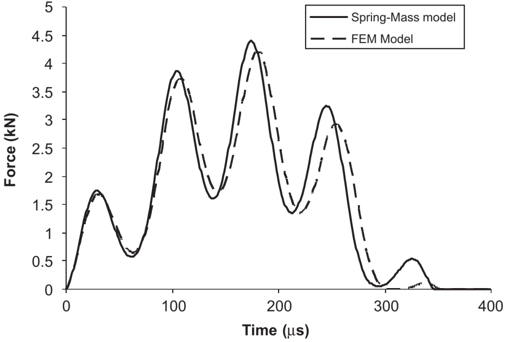
25/June/2020

Outline I

1. Why does a robot fear impacts?
2. Impact-awareness of a QP controller
3. Impact-aware multi-contact motion generation
4. Experiments

Impact can lead to instant failure

Impact is too fast to be compensated !



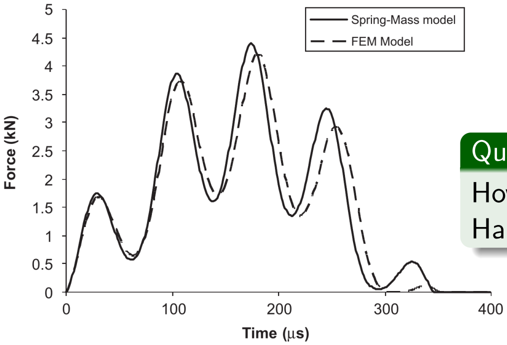
- Contact velocity $1.17m/s$.
- Plastic material
- Peak force: $4.5kN$.

Pashah, Sulaman, Michel Massenzio, and Eric Jacquelin. "Prediction of structural response for low velocity impact."

International Journal of Impact Engineering 35.2 (2008): 119-132.

Impact can lead to instant failure

Impact is too fast to be compensated !



- Contact velocity $1.17m/s$.
- Plastic material
- Peak force: $4.5kN$.

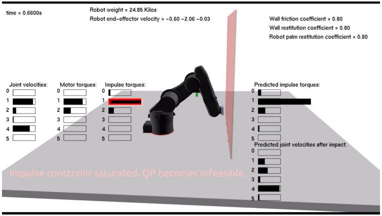
Question:

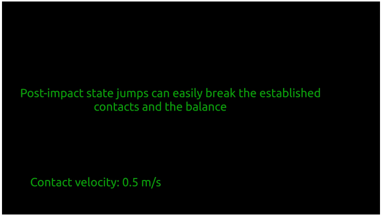
How should we compensate?
Hardware or software?

Pashah, Sulaman, Michel Massenzio, and Eric Jacquelin. "Prediction of structural response for low velocity impact."

International Journal of Impact Engineering 35.2 (2008): 119-132.

Why does a robot fear impact?





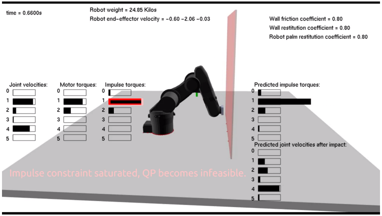
For all the robots:

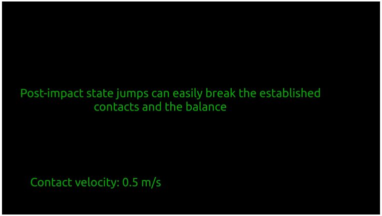
- Damage hardware, e.g. motors, harmonics.
- Unclear post-impact status: Rebounce? Sticking? Or slip?
- Equations of motions differ

Plus:

- Rigid environment and robot
- Unknown contact location

Why does a robot fear impact?





For all the robots:

- Damage hardware, e.g. motors, harmonics.
- Unclear post-impact status: Rebounce? Sticking? Or slip?
- Equations of motions differ

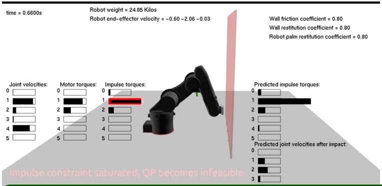
Plus:

- Rigid environment and robot
- Unknown contact location

For floating-based robots:

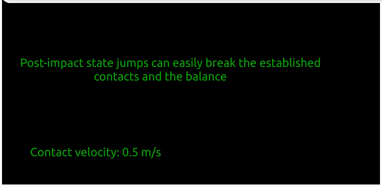
- Disturbing the standing stability.
- Breaking established contacts.

Why does a robot fear impact?



For all the robots:

Avoid impacts or use close-to-zero contact velocity.



For all the robots:

- Damage hardware, e.g. motors, harmonics.
- Unclear post-impact status: Rebounce? Sticking? Or slip?
- Equations of motions differ
- Unknown contact location

For floating-based robots:

- Disturbing the standing stability.
- Breaking established contacts.

State jumps and constraints

Hardware feasibility:

- Limited joint velocity
- Limited joint torque

Under-actuated robots:

- Established contacts
- Standing stability criteria

State jumps and constraints

Hardware feasibility:

- Limited joint velocity
- Limited joint torque

Under-actuated robots:

- Established contacts
- Standing stability criteria

Controller feasibility:

- Joint velocity constraints
- Joint torque constraints

Force-dependent constraints:

- Fulfilling the contact wrench cone
- ZMP constraint

Constraining the post-impact state

We propose:

$$\text{Post-impact state} \leq \text{Bounds}$$

Constraining the post-impact state

We propose:

$$\text{Post-impact state} \leq \text{Bounds}$$

Re-write the above:

$$(\text{Pre-impact state}(\ddot{q}) + \text{State jump}) \leq \text{Bounds}$$

Constraining the post-impact state

We propose:

$$\text{Post-impact state} \leq \text{Bounds}$$

Re-write the above:

$$(\text{Pre-impact state}(\ddot{q}) + \text{State jump}) \leq \text{Bounds}$$

If we can derive the linear dependence:

$$\text{State jump} = D \underbrace{\ddot{q}}_{\text{decision variable}}$$

Constraining the post-impact state

We propose:

$$\text{Post-impact state} \leq \text{Bounds}$$

Re-write the above:

$$(\text{Pre-impact state}(\ddot{q}) + \text{State jump}) \leq \text{Bounds}$$

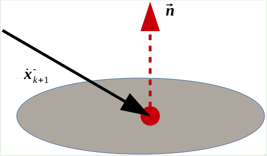
If we can derive the linear dependence:

$$\text{State jump} = D \underbrace{\ddot{q}}_{\text{decision variable}}$$

The task-space QP is impact aware!

Predict state jumps as symbolic expressions of $\ddot{q}$

Analytical impact dynamics model



- Kinematic impact dynamics model:

$$\Delta \dot{\boldsymbol{x}}_{k+1} = \dot{\boldsymbol{x}}^{+}_{k+1} - \dot{\boldsymbol{x}}^{-}_{k+1} = \underbrace{-(1 + c_{\mathrm{r}})\mathbf{P}_{\boldsymbol{n}}}_{\mathbf{P}_{\Delta}} \dot{\boldsymbol{x}}^{-}_{k+1}.$$

- Symbolic expression (linear function of $\ddot{q}$):

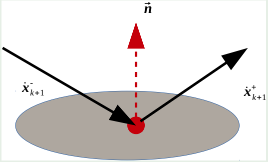
$$\Delta \dot{\boldsymbol{x}}_{k+1} = \mathbf{P}_{\Delta}(\boldsymbol{J}_k \dot{q}_k + \boldsymbol{J}_k \ddot{q}_k \Delta t + \dot{\boldsymbol{J}}_k \dot{q}_k \Delta t),$$

Zheng, Yuanfang, and Hooshang Hemami. "Mathematical modeling of a robot collision with its environment."

Journal of Robotic Systems 2.3 (1985): 289-307.

Predict state jumps as symbolic expressions of $\ddot{q}$

Analytical impact dynamics model



- Kinematic impact dynamics model:

$$\Delta\dot{\boldsymbol{x}}_{k+1} = \dot{\boldsymbol{x}}^{+}_{k+1} - \dot{\boldsymbol{x}}^{-}_{k+1} = \underbrace{-(1+c_{\mathrm{r}})\mathbf{P}_{\boldsymbol{n}}}_{\mathbf{P}_{\Delta}}\dot{\boldsymbol{x}}^{-}_{k+1}.$$

- Symbolic expression (linear function of $\ddot{q}$):

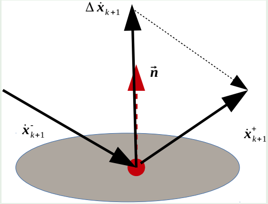
$$\Delta\dot{\boldsymbol{x}}_{k+1} = \mathbf{P}_{\Delta}(\boldsymbol{J}_k\dot{q}_k + \boldsymbol{J}_k\ddot{q}_k\Delta t + \dot{\boldsymbol{J}}_k\dot{q}_k\Delta t),$$

Zheng, Yuanfang, and Hooshang Hemami. "Mathematical modeling of a robot collision with its environment."

Journal of Robotic Systems 2.3 (1985): 289-307.

Predict state jumps as symbolic expressions of $\ddot{q}$

Analytical impact dynamics model



- Kinematic impact dynamics model:

$$\Delta \dot{\boldsymbol{x}}_{k+1} = \dot{\boldsymbol{x}}^{+}_{k+1} - \dot{\boldsymbol{x}}^{-}_{k+1} = \underbrace{-(1 + c_{\mathrm{r}})\mathbf{P}_{\boldsymbol{n}}}_{\mathbf{P}_{\Delta}} \dot{\boldsymbol{x}}^{-}_{k+1}.$$

- Symbolic expression (linear function of $\ddot{q}$):

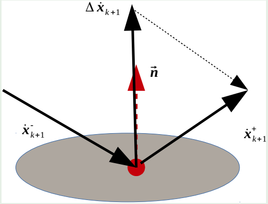
$$\Delta \dot{\boldsymbol{x}}_{k+1} = \mathbf{P}_{\Delta}(\boldsymbol{J}_k \dot{q}_k + \boldsymbol{J}_k \ddot{q}_k \Delta t + \dot{\boldsymbol{J}}_k \dot{q}_k \Delta t),$$

Zheng, Yuanfang, and Hooshang Hemami. "Mathematical modeling of a robot collision with its environment."

Journal of Robotic Systems 2.3 (1985): 289-307.

Predict state jumps as symbolic expressions of $\ddot{q}$

Analytical impact dynamics model



- Kinematic impact dynamics model:

$$\Delta \dot{\boldsymbol{x}}_{k+1} = \dot{\boldsymbol{x}}^{+}_{k+1} - \dot{\boldsymbol{x}}^{-}_{k+1} = \underbrace{-(1+c_{\mathrm{r}})\mathbf{P}_{\boldsymbol{n}}}_{\mathbf{P}_{\Delta}} \dot{\boldsymbol{x}}^{-}_{k+1}.$$

- Symbolic expression (linear function of $\ddot{q}$):

$$\Delta \dot{\boldsymbol{x}}_{k+1} = \mathbf{P}_{\Delta}(\boldsymbol{J}_k \dot{q}_k + \boldsymbol{J}_k \ddot{q}_k \Delta t + \dot{\boldsymbol{J}}_k \dot{q}_k \Delta t),$$

Zheng, Yuanfang, and Hooshang Hemami. "Mathematical modeling of a robot collision with its environment."

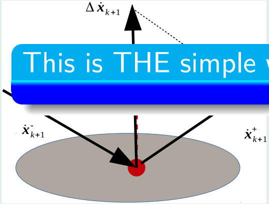
Journal of Robotic Systems 2.3 (1985): 289-307.

The analytical solution of a 3D Impact is UNKNOWN

- Stewart, David E. "Rigid-body dynamics with friction and impact." SIAM review 42.1 (2000): 3-39.
- Featherstone, Roy. Rigid body dynamics algorithms. Springer, 2014.
- Jia, Yan-Bin, and Feifei Wang. "Analysis and computation of two body impact in three dimensions." Journal of Computational and Nonlinear Dynamics 12.4 (2017): 041012.
- Jia, Yan-Bin, Matthew Gardner, and Xiaoqian Mu. "Batting an in-flight object to the target." The International Journal of Robotics Research (2019): 0278364918817116.

Predict state jumps as symbolic expressions of $\ddot{q}$

Analytical impact dynamics model



- Kinematic impact dynamics model:

This is THE simple way to predict the 1D state jumps:

$\dot{x}^{-}_{k+1}$.

- Symbolic expression (linear function of $\ddot{q}$):

$$\Delta\dot{x}_{k+1} = \mathbf{P}_{\Delta}(J_k\dot{q}_k + J_k\ddot{q}_k\Delta t + \dot{J}_k\dot{q}_k\Delta t),$$

Zheng, Yuanfang, and Hooshang Hemami. "Mathematical modeling of a robot collision with its environment."

Journal of Robotic Systems 2.3 (1985): 289-307.

Actively looking for an alternative solution now.

T

- Stewart, David E. "Rigid-body dynamics with friction and impact." SIAM review 42.1 (2000): 3-39.
- Featherstone, Roy. Rigid body dynamics algorithms. Springer, 2014.
- Jia, Yan-Bin, and Feifei Wang. "Analysis and computation of two body impact in three dimensions." Journal of Computational and Nonlinear Dynamics 12.4 (2017): 041012.
- Jia, Yan-Bin, Matthew Gardner, and Xiaoqian Mu. "Batting an in-flight object to the target." The International Journal of Robotics Research (2019): 0278364918817116.

2DOF Fixed-base manipulator toy example

Joint velocity constraints:

$$\dot{q} \leq \text{Bounds}.$$

Recall:

$$(\text{Pre-impact state}(\ddot{q}) + \text{State jump}) \leq \text{Bounds}$$

2DOF Fixed-base manipulator toy example

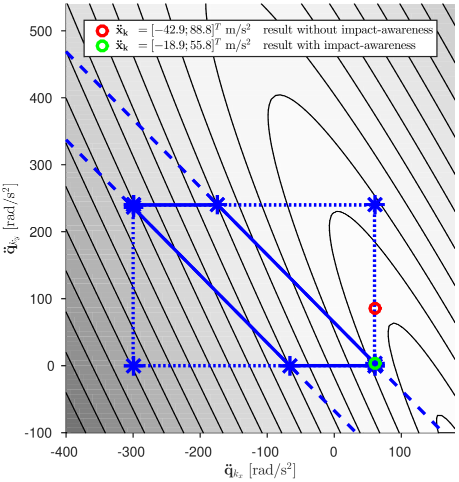
Joint velocity constraints:

$$\dot{q} \leq \text{Bounds}.$$

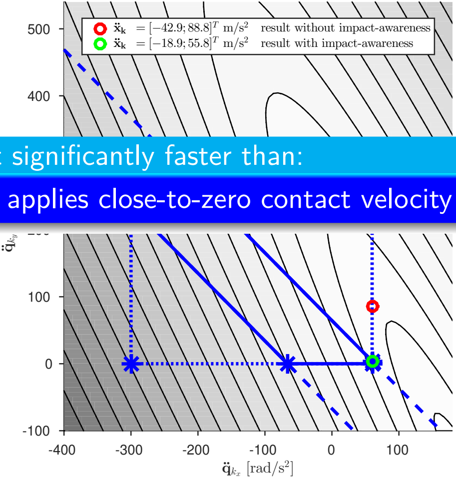
Impact-aware joint velocity constraints:

$$\dot{q}^{-}_{k+1} + \underbrace{\Delta\dot{q}_{k+1}}_{\text{Linear w.r.t. } \ddot{q}} \leq \text{Bounds}$$

2DOF Fixed-base manipulator toy example



2DOF Fixed-base manipulator toy example

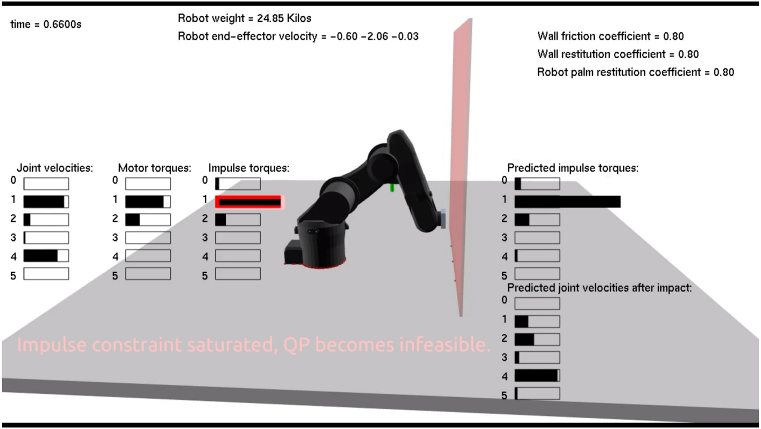


Conservative, yet significantly faster than:

the SOTA which applies close-to-zero contact velocity

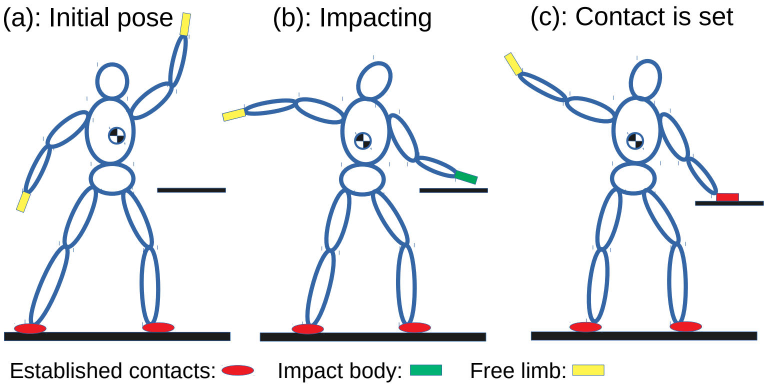
Wang Y and Kheddar A (2019) Impact-friendly robust control design with task-space quadratic optimization. In: Proceedings of Robotics: Science and Systems, volume 15. Freiburg, Germany.

Open source: `https://github.com/wyqsnddd/pyQpController`

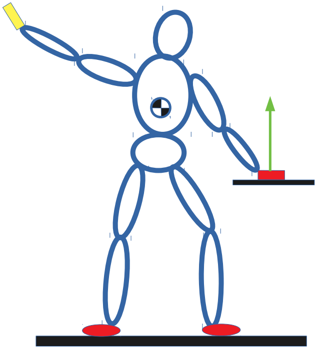


- Safely maximize the contact velocity / impulse.
- Hardware limits are respected.
- No reset map, impact timing or location, offline trajectory generation.
- Multiple (simultaneous) contact.
- Increase the size of the QP.
- We need to know the contact surface normal.

Whole-body state jumps



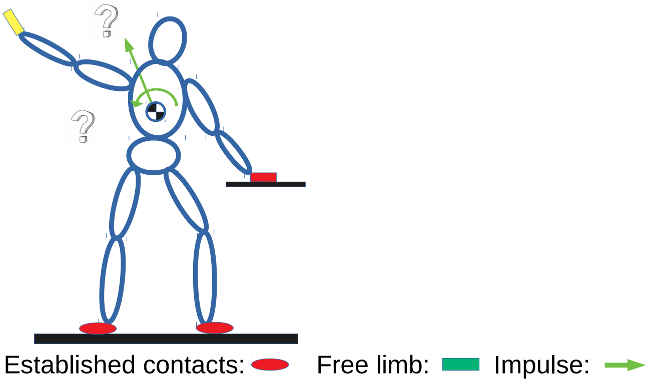
Whole-body state jumps



Established contacts: Free limb: Impulse:

Predictable impulse: $\boldsymbol{I}$

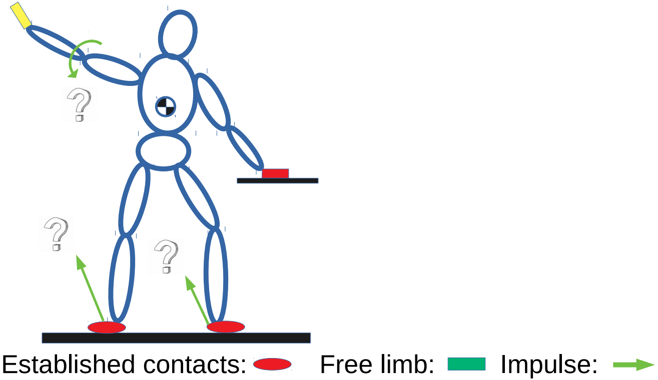
Whole-body state jumps



Centroidal momentum jump:

$$\begin{bmatrix} \Delta \mathcal{L}(\ddot{q}) \\ \Delta P(\ddot{q}) \end{bmatrix} = \begin{bmatrix} ? \\ ? \end{bmatrix}.$$

Whole-body state jumps



External impulses from established contacts: $\boldsymbol{I}_l(\ddot{q}) = ?$ $\boldsymbol{I}_r(\ddot{q}) = ?$.

End-effector velocity jump of a free limb: $\Delta\dot{\boldsymbol{x}}(\ddot{q}) = ?$.

Impulse Distribution Quadratic Programming (IDQP)

$\min_{\Delta \dot{q}, \boldsymbol{I}}$ Sum of momentum jumps

s.t. Task space: End-effector Momentum jump = external impulse

Centroidal frame: Momentum jump = Sum of external impulses

Controlled impacts: Generalized momentum jump = Known impulse

Impulse Distribution Quadratic Programming (IDQP)

$$\min_{\Delta \dot{q}, \boldsymbol{I}} \quad \text{Sum of momentum jumps}$$

s.t. Task space: End-effector Momentum jump = external impulse

$$\begin{bmatrix} \Delta\dot{\boldsymbol{x}}_{\sigma_{\text{contact}}} \\ \Delta\dot{\boldsymbol{x}}_{\sigma_{\text{impact}}} \\ \Delta\dot{\boldsymbol{x}}_{\sigma_{\text{free}}} \end{bmatrix} = \begin{bmatrix} \Upsilon_{\sigma_{\text{contact}},\sigma_{\text{contact}}}, & \Upsilon_{\sigma_{\text{contact}},\sigma_{\text{impact}}}, & \Upsilon_{\sigma_{\text{contact}}\sigma_{\text{free}}} \\ \Upsilon_{\sigma_{\text{impact}},\sigma_{\text{contact}}}, & \Upsilon_{\sigma_{\text{impact}},\sigma_{\text{impact}}}, & \Upsilon_{\sigma_{\text{impact}}\sigma_{\text{free}}} \\ \Upsilon_{\sigma_{\text{free}},\sigma_{\text{contact}}}, & \Upsilon_{\sigma_{\text{free}},\sigma_{\text{impact}}}, & \Upsilon_{\sigma_{\text{free}}\sigma_{\text{free}}} \end{bmatrix} \begin{bmatrix} \boldsymbol{I}_{\sigma_{\text{contact}}} \\ \boldsymbol{I}_{\sigma_{\text{impact}}} \\ \boldsymbol{0} \end{bmatrix}$$

Centroidal frame: Momentum jump = Sum of external impulses

Controlled impacts: Generalized momentum jump = Known impulse

Impulse Distribution Quadratic Programming (IDQP)

$\min_{\Delta \dot{q}, \boldsymbol{I}}$ Sum of momentum jumps

s.t. Task space: End-effector Momentum jump = external impulse

$$\begin{bmatrix} \Delta \dot{\boldsymbol{x}}_{\sigma_{\text{contact}}} \\ \Delta \dot{\boldsymbol{x}}_{\sigma_{\text{impact}}} \\ \Delta \dot{\boldsymbol{x}}_{\sigma_{\text{free}}} \end{bmatrix} = \begin{bmatrix} \Upsilon_{\sigma_{\text{contact}},\sigma_{\text{contact}}}, & \Upsilon_{\sigma_{\text{contact}},\sigma_{\text{impact}}}, & \Upsilon_{\sigma_{\text{contact}}\sigma_{\text{free}}} \\ \Upsilon_{\sigma_{\text{impact}},\sigma_{\text{contact}}}, & \Upsilon_{\sigma_{\text{impact}},\sigma_{\text{impact}}}, & \Upsilon_{\sigma_{\text{impact}}\sigma_{\text{free}}} \\ \Upsilon_{\sigma_{\text{free}},\sigma_{\text{contact}}}, & \Upsilon_{\sigma_{\text{free}},\sigma_{\text{impact}}}, & \Upsilon_{\sigma_{\text{free}}\sigma_{\text{free}}} \end{bmatrix} \begin{bmatrix} \boldsymbol{I}_{\sigma_{\text{contact}}} \\ \boldsymbol{I}_{\sigma_{\text{impact}}} \\ \boldsymbol{0} \end{bmatrix}$$

Centroidal frame: Momentum jump = Sum of external impulses

$$A_G(\boldsymbol{\theta})\Delta \dot{q} = \sum \boldsymbol{I}_{\sigma_{\text{contact}}} + \sum \boldsymbol{I}_{\sigma_{\text{impact}}}$$

Controlled impacts: Generalized momentum jump = Known impulse

Impulse Distribution Quadratic Programming (IDQP)

$\min_{\Delta \dot{q}, \boldsymbol{I}}$ Sum of momentum jumps

s.t. Task space: End-effector Momentum jump = external impulse

$$\begin{bmatrix} \Delta \dot{\boldsymbol{x}}_{\sigma_{\text{contact}}} \\ \Delta \dot{\boldsymbol{x}}_{\sigma_{\text{impact}}} \\ \Delta \dot{\boldsymbol{x}}_{\sigma_{\text{free}}} \end{bmatrix} = \begin{bmatrix} \Upsilon_{\sigma_{\text{contact}}, \sigma_{\text{contact}}}, & \Upsilon_{\sigma_{\text{contact}}, \sigma_{\text{impact}}}, & \Upsilon_{\sigma_{\text{contact}} \sigma_{\text{free}}} \\ \Upsilon_{\sigma_{\text{impact}}, \sigma_{\text{contact}}}, & \Upsilon_{\sigma_{\text{impact}}, \sigma_{\text{impact}}}, & \Upsilon_{\sigma_{\text{impact}} \sigma_{\text{free}}} \\ \Upsilon_{\sigma_{\text{free}}, \sigma_{\text{contact}}}, & \Upsilon_{\sigma_{\text{free}}, \sigma_{\text{impact}}}, & \Upsilon_{\sigma_{\text{free}} \sigma_{\text{free}}} \end{bmatrix} \begin{bmatrix} \boldsymbol{I}_{\sigma_{\text{contact}}} \\ \boldsymbol{I}_{\sigma_{\text{impact}}} \\ \boldsymbol{0} \end{bmatrix}$$

Centroidal frame: Momentum jump = Sum of external impulses

$$A_G(\boldsymbol{\theta}) \Delta \dot{q} = \sum \boldsymbol{I}_{\sigma_{\text{contact}}} + \sum \boldsymbol{I}_{\sigma_{\text{impact}}}$$

Controlled impacts: Generalized momentum jump = Known impulse

$$\boldsymbol{J}_i \Delta \dot{q} = \Delta \dot{\boldsymbol{x}}_i \text{ for } i \in \sigma_{\text{impact}}.$$

Impulse Distribution Quadratic Programming (IDQP)

$\min_{\Delta \dot{q}, \boldsymbol{I}}$ Sum of momentum jumps

s.t. Task space: End-effector Momentum jump = external impulse

External impulses from established contacts: $\boldsymbol{I}_l(\ddot{q})$ $\boldsymbol{I}_r(\ddot{q})$.

$$\begin{bmatrix} \Delta \dot{\boldsymbol{x}}_{\sigma_{\text{impact}}} \\ \Delta \dot{\boldsymbol{x}}_{\sigma_{\text{free}}} \end{bmatrix} = \begin{bmatrix} \Upsilon_{\sigma_{\text{impact}},\sigma_{\text{contact}}}, & \Upsilon_{\sigma_{\text{impact}},\sigma_{\text{impact}}}, & \Upsilon_{\sigma_{\text{impact}}\sigma_{\text{free}}} \\ \Upsilon_{\sigma_{\text{free}},\sigma_{\text{contact}}}, & \Upsilon_{\sigma_{\text{free}},\sigma_{\text{impact}}}, & \Upsilon_{\sigma_{\text{free}}\sigma_{\text{free}}} \end{bmatrix} \begin{bmatrix} \boldsymbol{I}_{\sigma_{\text{impact}}} \\ \boldsymbol{0} \end{bmatrix}$$

Centroidal frame: Momentum jump = Sum of external impulses

Centroidal momentum jump: $\Delta P(\ddot{q})$, $\Delta \mathcal{L}(\ddot{q})$.

Controlled impacts: Generalized momentum jump = Known impulse

End-effector velocity jump of a free limb: $\Delta \dot{\boldsymbol{x}}(\ddot{q})$.

Impulse Distribution Quadratic Programming (IDQP)

min$_{\Delta\dot{q}, \boldsymbol{I}}$ Sum of momentum jumps

s.t. Task space: End-effector Momentum jump = external impulse

External impulses from established contacts: $\boldsymbol{I}_l(\ddot{q})$ $\boldsymbol{I}_r(\ddot{q})$.

contact

σ_{impact}

$\mathbf{0}$

Other state jump: $\Delta\boldsymbol{z}(\ddot{q})$, $\Delta\xi(\ddot{q})$, $\Delta\dot{\boldsymbol{c}}(\ddot{q})$.

Centroidal frame: Momentum jump = Sum of external impulses

Centroidal momentum jump: $\Delta P(\ddot{q})$, $\Delta\mathcal{L}(\ddot{q})$.

Controlled impacts: Generalized momentum jump = Known impulse

End-effector velocity jump of a free limb: $\Delta\dot{\boldsymbol{x}}(\ddot{q})$.

Impact-aware constraints formulation

Constrained ZMP:

$$\boldsymbol{z} \leq \text{Bounds} \quad \Rightarrow \quad \boldsymbol{z} + \Delta\boldsymbol{z}(\ddot{q}) \leq \text{Bounds}$$

Impact-aware constraints formulation

Constrained ZMP:

$$\boldsymbol{z} \leq \text{Bounds} \quad \Rightarrow \quad \boldsymbol{z} + \Delta \boldsymbol{z}(\ddot{q}) \leq \text{Bounds}$$

Constrained DCM:

$$\boldsymbol{\xi} \leq \text{Bounds} \quad \Rightarrow \quad \boldsymbol{\xi} + \Delta \boldsymbol{\xi}(\ddot{q}) \leq \text{Bounds}$$

Impact-aware constraints formulation

Constrained ZMP:

$$\boldsymbol{z} \leq \text{Bounds} \quad \Rightarrow \quad \boldsymbol{z} + \Delta \boldsymbol{z}(\ddot{q}) \leq \text{Bounds}$$

Constrained DCM:

$$\boldsymbol{\xi} \leq \text{Bounds} \quad \Rightarrow \quad \boldsymbol{\xi} + \Delta \boldsymbol{\xi}(\ddot{q}) \leq \text{Bounds}$$

Constrained COM velocity:

$$\dot{\boldsymbol{c}} \leq \text{Bounds} \quad \Rightarrow \quad \dot{\boldsymbol{c}} + \Delta \dot{\boldsymbol{c}}(\ddot{q}) \leq \text{Bounds}$$

Impact-aware constraints formulation

Constrained ZMP:

$$\boldsymbol{z} \leq \text{Bounds} \quad \Rightarrow \quad \boldsymbol{z} + \Delta \boldsymbol{z}(\ddot{q}) \leq \text{Bounds}$$

Constrained DCM:

$$\boldsymbol{\xi} \leq \text{Bounds} \quad \Rightarrow \quad \boldsymbol{\xi} + \Delta \boldsymbol{\xi}(\ddot{q}) \leq \text{Bounds}$$

Constrained COM velocity:

$$\dot{\boldsymbol{c}} \leq \text{Bounds} \quad \Rightarrow \quad \dot{\boldsymbol{c}} + \Delta \dot{\boldsymbol{c}}(\ddot{q}) \leq \text{Bounds}$$

Constrained angular momentum:

$$\mathcal{L} \leq \text{Bounds} \quad \Rightarrow \quad \mathcal{L} + \Delta \mathcal{L}(\ddot{q}) \leq \text{Bounds}$$

Impact-aware constraints formulation

Constrained ZMP:

$$\boldsymbol{z} \leq \text{Bounds} \quad \Rightarrow \quad \boldsymbol{z} + \Delta \boldsymbol{z}(\ddot{q}) \leq \text{Bounds}$$

Constrained DCM:

$$\boldsymbol{\xi} \leq \text{Bounds} \quad \Rightarrow \quad \boldsymbol{\xi} + \Delta \boldsymbol{\xi}(\ddot{q}) \leq \text{Bounds}$$

Seamless integration to the whole-body QP controller.

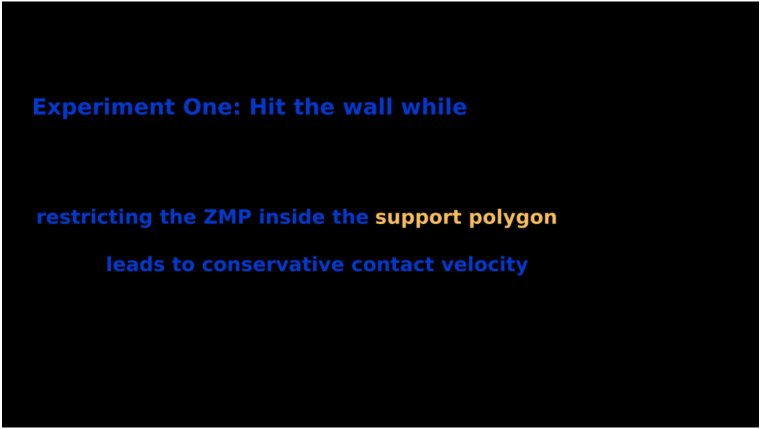
C

$$\dot{\boldsymbol{c}} \leq \text{Bounds} \quad \Rightarrow \quad \dot{\boldsymbol{c}} + \Delta \dot{\boldsymbol{c}}(\ddot{q}) \leq \text{Bounds}$$

Constrained angular momentum:

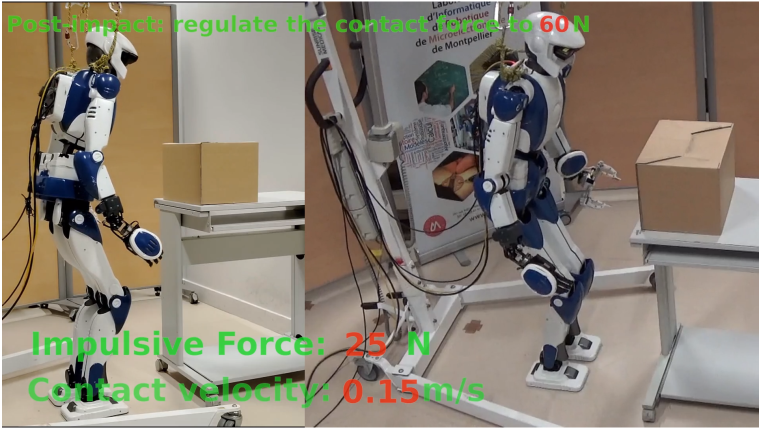
$$\mathcal{L} \leq \text{Bounds} \quad \Rightarrow \quad \mathcal{L} + \Delta \mathcal{L}(\ddot{q}) \leq \text{Bounds}$$

Restrict ZMP strictly inside the support polygon



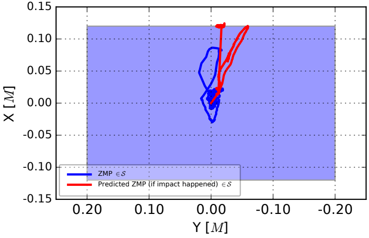
Reference contact velocity: 0.8 m/s.

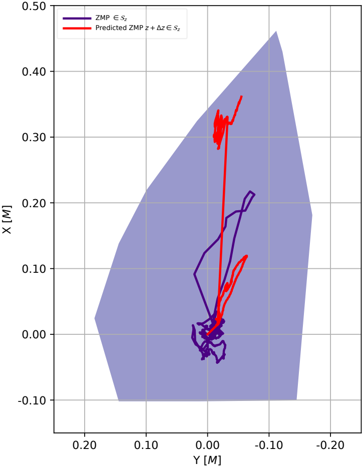
Box-grabbing controller



Reference contact velocity: 0.15 m/s.

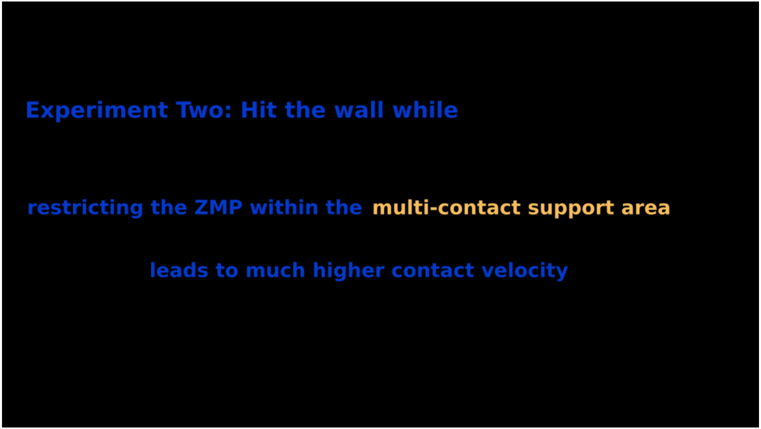
From support polygon to multi-contact COM area





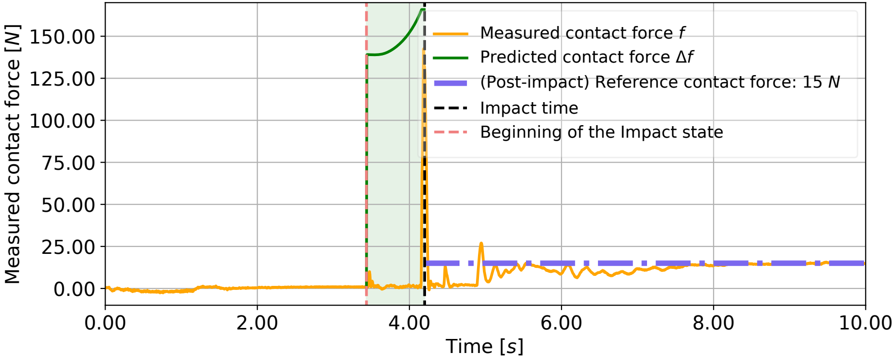
Caron S, Pham QC and Nakamura Y (2017) Zmp support areas for multi-contact mobility under frictional constraints. IEEE Transactions on Robotics 33(1): 67-80.

Push an unknown wall with energetic impacts

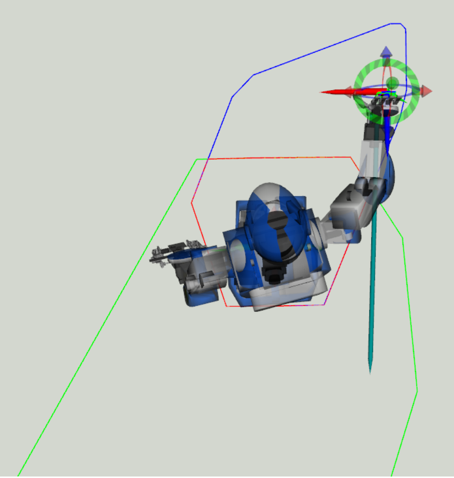


Reference contact velocity: 0.8 m/s.

Impact force stabilization



Real-time updating the multi-contact areas



Summary:

Contributions:

- Controlled state jumps: $\Delta \boldsymbol{d}(\ddot{q})$.
- Autonomously *filters* the contact velocity: $\dot{\boldsymbol{x}} = \text{feasible}(\dot{\boldsymbol{x}}^{\text{ref}})$.
- Hardware limits and standing stability margins are respected.
- Robust to impact timing or location
- No reset maps or offline trajectory generation.

Summary:

Contributions:

- Controlled state jumps: $\Delta \boldsymbol{d}(\ddot{q})$.
- Autonomously *filters* the contact velocity: $\dot{\boldsymbol{x}} = \text{feasible}(\dot{\boldsymbol{x}}^{\text{ref}})$.
- Hardware limits and standing stability margins are respected.
- Robust to impact timing or location
- No reset maps or offline trajectory generation.

Future work:

- Frictional impact dynamics model in 3D.
- Refined impulse propagation.
- Off-line and on-line parameter identification, e.g., coefficient of restitution.
- Investigate the tangential impulse.

Summary:

Contributions:

- Controlled state jumps: $\Delta \boldsymbol{d}(\ddot{q})$.
- Autonomously *filters* the contact velocity: $\dot{\boldsymbol{x}} = \text{feasible}(\dot{\boldsymbol{x}}^{\text{ref}})$.
- Hardware limits and standing stability margins are respected.
- Robust to impact timing or location

Thank you very much!

Fu

- Frictional impact dynamics model in 3D.
- Refined impulse propagation.
- Off-line and on-line parameter identification, e.g., coefficient of restitution.
- Investigate the tangential impulse.

Summary:

Contributions:

- Controlled state jumps: $\Delta \boldsymbol{d}(\ddot{q})$.
- Autonomously *filters* the contact velocity: $\dot{\boldsymbol{x}} = \text{feasible}(\dot{\boldsymbol{x}}^{\text{ref}})$.
- Hardware limits and standing stability margins are respected.
- Robust to impact timing or location

Thank you very much!

Fu...

- Frictional impact dynamics model in 3D.

Pre-print:
Impact-Aware Task-Space Quadratic-Programming Control
https://hal.archives-ouvertes.fr/hal-02741682/

wyqsnddd@gmail.com