Multi-Contact Planning

Karim Bouyarmane

CNRS-AIST JRL (Joint Robotics Laboratory)
National Institute of AIST, Tsukuba, Japan

Disney Research Pittsburgh, August 19, 2011

Outline

Outline

Targeted applications

- legged locomotion
- dexterous manipulation
- dual arm manipulation
- collaborative tasks
- ...

System

We consider systems made of N *entities*:

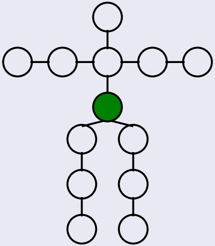
- humanoid robots
- dexterous fingers
- anthropomorphic arms
- manipulated objects
- and the environment

$$r \in \{1, \ldots, N\}$$

each entity $r \in \{1, \ldots, N\}$ is represented as a kinematic tree

each entity $r \in \{1, \dots, N\}$ is represented as a kinematic tree

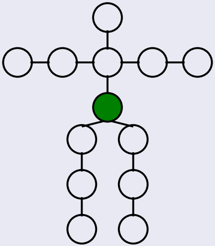
Humanoid robot



each entity $r \in \{1, \ldots, N\}$ is represented as a kinematic tree

Regular link Free-flying link Fixed link Joint

Humanoid robot

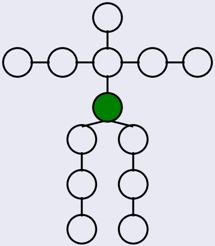


Manipulated object

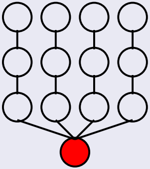
each entity $r \in \{1, \ldots, N\}$ is represented as a kinematic tree

○ Regular link ● Free-flying link ● Fixed link — Joint

Humanoid robot



Dexterous hand

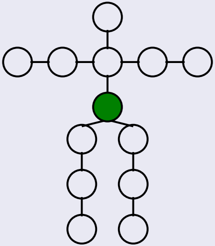


Manipulated object

each entity $r \in \{1, \ldots, N\}$ is represented as a kinematic tree

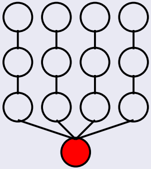
Regular link   Free-flying link   Fixed link   Joint

Humanoid robot



Manipulated object

Dexterous hand

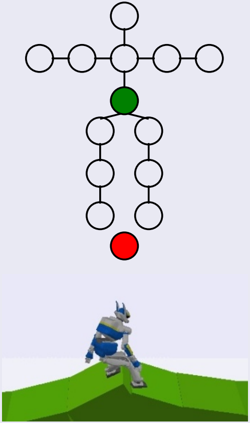


Surrounding environment

The system is an assembling of an arbitrary number N of entities

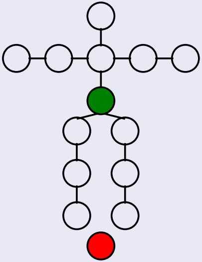
The system is an assembling of an arbitrary number N of entities

Locomotion

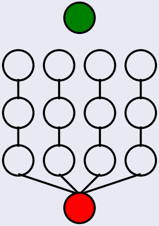


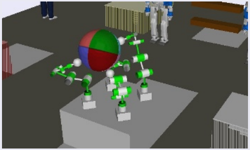
The system is an assembling of an arbitrary number N of entities

Locomotion



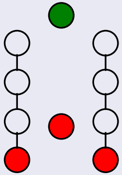
Manipulation

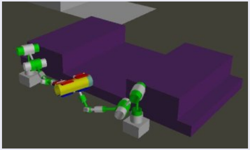




The system is an assembling of an arbitrary number N of entities

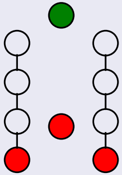
Dual arm

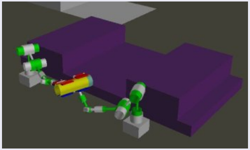




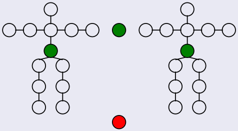
The system is an assembling of an arbitrary number N of entities

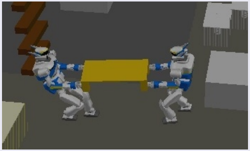
Dual arm





Collaborative





Configuration space

The configuration space of the system in the Cartesian product of the respective configuration spaces of the entities

$$\mathscr{C} = \prod_{r=1}^{N} \mathscr{C}_r$$

An entity's configuration space $\mathscr{C}_r$ can take different forms

$$\begin{cases} \mathscr{C}_r = \mathbb{R}^k & \text{, for a fixed-base robot} \\ \mathscr{C}_r = SE(3) \times \mathbb{R}^k & \text{, for a non-fixed-base robot} \\ \mathscr{C}_r = SE(3) & \text{, for a manipulated object} \\ \mathscr{C}_r = \{0\} & \text{, for a fixed environment object} \end{cases}$$

Configuration

$$q \in \mathscr{C} = \prod_{r=1}^{N} \mathscr{C}_r$$

$$q = (q_1, \ldots, q_r)$$

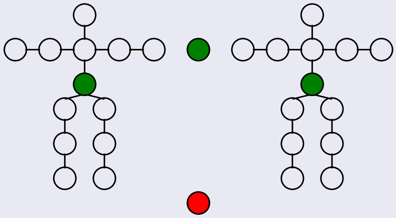
Contact

A contact can be defined between any two nodes of the system (even within same entity). Contacts can be unilateral or bilateral.

Contact

A contact can be defined between any two nodes of the system (even within same entity). Contacts can be unilateral or bilateral.

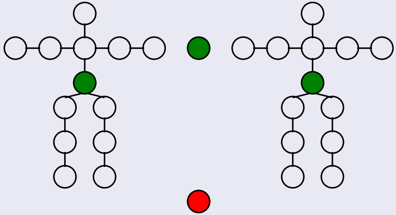
System without contacts



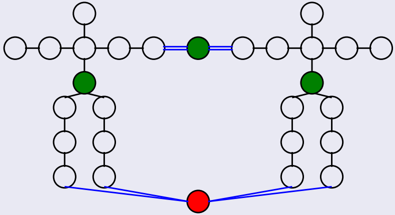
Contact

A contact can be defined between any two nodes of the system (even within same entity). Contacts can be unilateral or bilateral.

System without contacts



System with contacts



Contact

A contact c is a 7-tuple

$$c = (r_1, s_1, r_2, s_2, \quad x, y, \theta) \in \mathbb{N}^4 \times SE(2)$$

The set of all contacts is denoted E_{contacts}

Stance

A stance σ is a set of contacts

$$\sigma = \{c_1, \ldots, c_n\}$$

The set of all stances is denoted $\Sigma \subset 2^{E_{\text{contacts}}}$

Forward kinematics mapping

$$p_\Sigma : \mathscr{C} \rightarrow \Sigma$$

Forward kinematics mapping

$$p_{\Sigma} : \mathscr{C} \rightarrow \Sigma$$

Inverse kinematics mapping

$$\mathscr{Q}_{\sigma} = p_{\Sigma}^{-1}(\sigma)$$

Forward kinematics mapping

$$p_\Sigma : \mathscr{C} \rightarrow \Sigma$$

Inverse kinematics mapping

$$\mathscr{Q}_\sigma = p_\Sigma^{-1}(\sigma)$$

Feasible space

$$\mathscr{F}_\sigma \subset \mathscr{Q}_\sigma$$

such that $\forall q \in \mathscr{F}_\sigma$, q is in static equilibrium.

Adjacent stances

$\sigma' \in \mathrm{Adj}^+(\sigma)$ if $\sigma' \supset \sigma$ and $\mathrm{card}(\sigma') = \mathrm{card}(\sigma) + 1$.
$\sigma' \in \mathrm{Adj}^-(\sigma)$ if $\sigma' \subset \sigma$ and $\mathrm{card}(\sigma') = \mathrm{card}(\sigma) - 1$.

$$\mathrm{Adj}(\sigma) = \mathrm{Adj}^+(\sigma) \cup \mathrm{Adj}^-(\sigma)$$

Adjacent stances

$\sigma' \in \mathrm{Adj}^+(\sigma)$ if $\sigma' \supset \sigma$ and $\mathrm{card}(\sigma') = \mathrm{card}(\sigma) + 1$.
$\sigma' \in \mathrm{Adj}^-(\sigma)$ if $\sigma' \subset \sigma$ and $\mathrm{card}(\sigma') = \mathrm{card}(\sigma) - 1$.

$$\mathrm{Adj}(\sigma) = \mathrm{Adj}^+(\sigma) \cup \mathrm{Adj}^-(\sigma)$$

On $\mathrm{Adj}^+(\sigma)$ we define an equivalence relation $\sim_\sigma$ which leaves the position (x, y, θ) of the added contact undetermined.

Adjacent stances

$\sigma' \in \mathrm{Adj}^+(\sigma)$ if $\sigma' \supset \sigma$ and $\mathrm{card}(\sigma') = \mathrm{card}(\sigma) + 1$.
$\sigma' \in \mathrm{Adj}^-(\sigma)$ if $\sigma' \subset \sigma$ and $\mathrm{card}(\sigma') = \mathrm{card}(\sigma) - 1$.

$$\mathrm{Adj}(\sigma) = \mathrm{Adj}^+(\sigma) \cup \mathrm{Adj}^-(\sigma)$$

On $\mathrm{Adj}^+(\sigma)$ we define an equivalence relation $\sim_\sigma$ which leaves the position (x, y, θ) of the added contact undetermined.

- each equivalence class $[\sigma']$ is isomorphic to $\mathbb{R}^2 \times \mathbb{S}^1$

Adjacent stances

$\sigma' \in \mathrm{Adj}^{+}(\sigma)$ if $\sigma' \supset \sigma$ and $\mathrm{card}(\sigma') = \mathrm{card}(\sigma) + 1$.
$\sigma' \in \mathrm{Adj}^{-}(\sigma)$ if $\sigma' \subset \sigma$ and $\mathrm{card}(\sigma') = \mathrm{card}(\sigma) - 1$.

$$\mathrm{Adj}(\sigma) = \mathrm{Adj}^{+}(\sigma) \cup \mathrm{Adj}^{-}(\sigma)$$

On $\mathrm{Adj}^{+}(\sigma)$ we define an equivalence relation $\sim_\sigma$ which leaves the position (x, y, θ) of the added contact undetermined.

- each equivalence class $[\sigma']$ is isomorphic to $\mathbb{R}^2 \times \mathbb{S}^1$
- the quotient space is denoted $\mathrm{Adj}^{+}(\sigma)_{/\sim_\sigma}$

Adjacent stances

$\sigma' \in \mathrm{Adj}^+(\sigma)$ if $\sigma' \supset \sigma$ and $\mathrm{card}(\sigma') = \mathrm{card}(\sigma) + 1$.
$\sigma' \in \mathrm{Adj}^-(\sigma)$ if $\sigma' \subset \sigma$ and $\mathrm{card}(\sigma') = \mathrm{card}(\sigma) - 1$.

$$\mathrm{Adj}(\sigma) = \mathrm{Adj}^+(\sigma) \cup \mathrm{Adj}^-(\sigma)$$

On $\mathrm{Adj}^+(\sigma)$ we define an equivalence relation $\sim_\sigma$ which leaves the position (x, y, θ) of the added contact undetermined.

- each equivalence class $[\sigma']$ is isomorphic to $\mathbb{R}^2 \times \mathbb{S}^1$
- the quotient space is denoted $\mathrm{Adj}^+(\sigma)_{/\sim_\sigma}$

Feasible sequence of stances

$$(\sigma_1, \dots, \sigma_n) \in \Sigma^n$$

such that

$$\forall i \in \{1, \dots, n-1\} \quad \sigma_{i+1} \in \mathrm{Adj}(\sigma_i) \text{ and } \mathscr{F}_{\sigma_i} \cap \mathscr{F}_{\sigma_{i+1}} \neq \varnothing$$

Two problems

Two problems

IK problem

Given a stance, find a feasible configuration.

Two problems

IK problem

Given a stance, find a feasible configuration.

sequence-of-stances planning problem

Given start and goal stances, find a feasible sequence of stances.

Outline

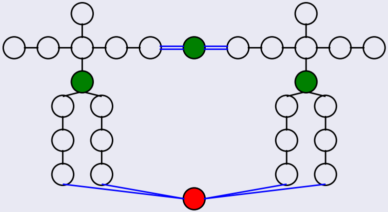
Problem

Given a set of contacts between multiple entities (robots, objects to manipulate, surrounding environment), find a configuration of the whole system satisfying static equilibrium conditions for all the entities.

Problem

Given a set of contacts between multiple entities (robots, objects to manipulate, surrounding environment), find a configuration of the whole system satisfying static equilibrium conditions for all the entities.

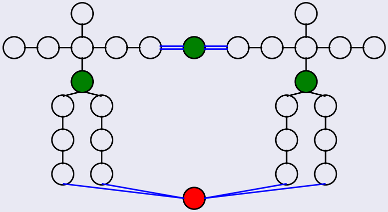
Input



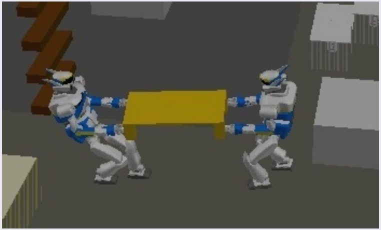
Problem

Given a set of contacts between multiple entities (robots, objects to manipulate, surrounding environment), find a configuration of the whole system satisfying static equilibrium conditions for all the entities.

Input



Output



Related work

Indirectly related work (more suitable to rejection sampling)

- Closed kinematic chains sampling (LaValle99, Han00, Cortes02)
- Static equilibrium tests (Bretl08, Rimon08)
- Contact forces optimization (Boyd07)

Related work

Indirectly related work (more suitable to rejection sampling)

- Closed kinematic chains sampling (LaValle99, Han00, Cortes02)
- Static equilibrium tests (Bretl08, Rimon08)
- Contact forces optimization (Boyd07)

Directly related work

- Posture generation (Escande06)
- Iterative constraints enforcement (Hauser05)

Related work

Indirectly related work (more suitable to rejection sampling)

- Closed kinematic chains sampling (LaValle99, Han00, Cortes02)
- Static equilibrium tests (Bretl08, Rimon08)
- Contact forces optimization (Boyd07)

Directly related work

- Posture generation (Escande06)
- Iterative constraints enforcement (Hauser05)

Our contributions

- More general contact configurations (non-horizontal, non-coplanar, frictional contacts)
- Torque limits as constraints (versus post-rejection in the case of Hauser05)
- Multi-agent systems, non-fixed contact supports

Problem variables

The variables of the problem are made of

- the configuration variables q
- the contact forces variables λ

Total variables

$$(q, \lambda)$$

Configuration variables q

Configuration variables

$$q \in \mathscr{C} = \prod_{r=1}^{N} \mathscr{C}_r$$

$$q = (q_1, \ldots, q_r)$$

Contact forces variables λ

Let us suppose we have n contact points $c \in \{1, \ldots, n\}$

Contact forces variables λ

Let us suppose we have n contact points $c \in \{1, \ldots, n\}$

Contact model

At each contact point c we consider a discretized friction cone with generators K_c. The contact force is then

$$f_c = \lambda_c^T K_c$$

with λ_c a vector of positive real numbers

$$\lambda_c \geq 0$$

Contact forces variables λ

Contact forces variables

$$\lambda = (\lambda_c)_{c \in \{1,\ldots,n\}}$$

$$\lambda = (\lambda_1, \ldots, \lambda_n)$$

Optimization formulation

$$\min_{q,\lambda} \mathrm{obj}(q, \lambda)$$

Optimization formulation

$$\min_{q,\lambda} \mathrm{obj}(q, \lambda)$$

The objective function

$$\mathrm{obj}(q, \lambda) = \alpha \left\| q - q_{\mathrm{ref}} \right\|^2 + \beta \left\| \lambda \right\|^2 + \gamma \left\| \tau \right\|^2$$

Optimization formulation

$$\min_{q,\lambda} \mathrm{obj}(q, \lambda)$$

The objective function

$$\mathrm{obj}(q, \lambda) = \alpha \left\| q - q_{\mathrm{ref}} \right\|^2 + \beta \left\| \lambda \right\|^2 + \gamma \left\| \tau \right\|^2$$

Constraints

- Static equilibrium for all non-fixed-base entities
- Joint limits for all joints
- Torque limits for all joint actuators
- Newton's third law
- $\lambda_c \geq 0$ for all unilateral contacts c
- Collision-free

Solver

We use IPOPT as a black-box non-linear optimization solver.
Requires methods for computing the gradient.

Solver

We use IPOPT as a black-box non-linear optimization solver.
Requires methods for computing the gradient.

Gradient computations

All the gradient computations are based on the algorithm providing the *Kinematic Jacobian* J

$$\dot{x} = J\dot{\theta}$$

cf. paper for details, especially concerning the gradient of the torque limit constraint based on the symbolic differentiation of the Kinematic Jacobian J (Bruynincks96)

Example videos (external player)

Outline

Problem

Given $\sigma_{\mathrm{start}}, \sigma_{\mathrm{goal}} \in \Sigma$ find a feasible sequence of stances $(\sigma_1, \ldots, \sigma_n)$ such that $\sigma_1 = \sigma_{\mathrm{start}}$ and $\sigma_n = \sigma_{\mathrm{goal}}$. i.e. such that

- $\sigma_1 = \sigma_{\mathrm{start}}$
- $\sigma_n = \sigma_{\mathrm{goal}}$
- $\forall i \in \{1, \ldots, n-1\} \quad \sigma_{i+1} \in \mathrm{Adj}(\sigma_i)$
- $\forall i \in \{1, \ldots, n-1\} \quad \mathscr{F}_{\sigma_i} \cap \mathscr{F}_{\sigma_{i+1}} \neq \varnothing$

Related work

- (Escande 2006) locomotion for a single humanoid robot.
- (Hauser 2005) discretization of candidate contact locations on the environment.

Related work

- (Escande 2006) locomotion for a single humanoid robot.
- (Hauser 2005) discretization of candidate contact locations on the environment.

Our contributions

- More general systems
- More general contact configurations (non-horizontal, non-coplanar, frictional contacts, bilateral contacts, non-fixed contact locations)
- Unified framework for locomotion and manipulation

Best-First algorithm

Best-First algorithm

- initialize priority queue $Q \leftarrow \{\sigma_{\text{start}}\}$

Best-First algorithm

- initialize priority queue $Q \leftarrow \{\sigma_{\text{start}}\}$
- loop

Best-First algorithm

- initialize priority queue $Q \leftarrow \{\sigma_{\text{start}}\}$
- loop
  - pop best stance σ from Q

Best-First algorithm

- initialize priority queue $Q \leftarrow \{\sigma_{\text{start}}\}$
- loop
  - pop best stance σ from Q
  - for all $[\sigma'] \in \text{Adj}^{+}(\sigma)_{/\sim_\sigma}$

Best-First algorithm

- initialize priority queue $Q \leftarrow \{\sigma_{\text{start}}\}$
- loop
  - pop best stance σ from Q
  - for all $[\sigma'] \in \text{Adj}^{+}(\sigma)_{/\sim_\sigma}$
    - call the IK solver to find q in $\mathscr{F}_\sigma \cap \mathscr{F}_{[\sigma']}$

Best-First algorithm

- initialize priority queue $Q \leftarrow \{\sigma_{\text{start}}\}$
- loop
  - pop best stance σ from Q
  - for all $[\sigma'] \in \text{Adj}^{+}(\sigma)_{/\sim_\sigma}$
    - call the IK solver to find q in $\mathscr{F}_\sigma \cap \mathscr{F}_{[\sigma']}$
    - upon success push $\sigma' = p_\Sigma(q)$ into Q

Best-First algorithm

- initialize priority queue $Q \leftarrow \{\sigma_{\text{start}}\}$
- loop
  - pop best stance σ from Q
  - for all $[\sigma'] \in \text{Adj}^{+}(\sigma)_{/\sim_\sigma}$
    - call the IK solver to find q in $\mathscr{F}_\sigma \cap \mathscr{F}_{[\sigma']}$
    - upon success push $\sigma' = p_\Sigma(q)$ into Q
  - for all $\sigma' \in \text{Adj}^{-}(\sigma)$

Best-First algorithm

- initialize priority queue $Q \leftarrow \{\sigma_{\text{start}}\}$
- loop
  - pop best stance σ from Q
  - for all $[\sigma'] \in \text{Adj}^{+}(\sigma)_{/\sim_\sigma}$
    - call the IK solver to find q in $\mathscr{F}_\sigma \cap \mathscr{F}_{[\sigma']}$
    - upon success push $\sigma' = p_\Sigma(q)$ into Q
  - for all $\sigma' \in \text{Adj}^{-}(\sigma)$
    - call the IK solver to find q in $\mathscr{F}_\sigma \cap \mathscr{F}_{\sigma'}$

Best-First algorithm

- initialize priority queue $Q \leftarrow \{\sigma_{\text{start}}\}$
- loop
  - pop best stance σ from Q
  - for all $[\sigma'] \in \mathrm{Adj}^{+}(\sigma)_{/\sim_\sigma}$
    - call the IK solver to find q in $\mathscr{F}_\sigma \cap \mathscr{F}_{[\sigma']}$
    - upon success push $\sigma' = p_\Sigma(q)$ into Q
  - for all $\sigma' \in \mathrm{Adj}^{-}(\sigma)$
    - call the IK solver to find q in $\mathscr{F}_\sigma \cap \mathscr{F}_{\sigma'}$
    - upon success push σ' into Q

Best-First algorithm

- initialize priority queue $Q \leftarrow \{\sigma_{\mathrm{start}}\}$
- loop
  - pop best stance σ from Q
  - for all $[\sigma'] \in \mathrm{Adj}^{+}(\sigma)_{/\sim_\sigma}$
    - call the IK solver to find q in $\mathscr{F}_\sigma \cap \mathscr{F}_{[\sigma']}$
    - upon success push $\sigma' = p_\Sigma(q)$ into Q
  - for all $\sigma' \in \mathrm{Adj}^{-}(\sigma)$
    - call the IK solver to find q in $\mathscr{F}_\sigma \cap \mathscr{F}_{\sigma'}$
    - upon success push σ' into Q
- until σ is close enough to σ_{goal}

Example videos (External player)

Outline

Problem

Given a feasible sequence of stances, synthesize a physically valid motion for the robot that realizes the steps.

Related work

- (Abe, da Silva, Popovic 2007) Multi-objective control with frictional contacts
- (de Lasa, Mordatch, Hertzmann 2010) Feature-based locomotion controller

Related work

- (Abe, da Silva, Popovic 2007) Multi-objective control with frictional contacts
- (de Lasa, Mordatch, Hertzmann 2010) Feature-based locomotion controller

Our contributions

- Changing contact configurations
- Acyclic non-gaited motion

Related work

- (Abe, da Silva, Popovic 2007) Multi-objective control with frictional contacts
- (de Lasa, Mordatch, Hertzmann 2010) Feature-based locomotion controller

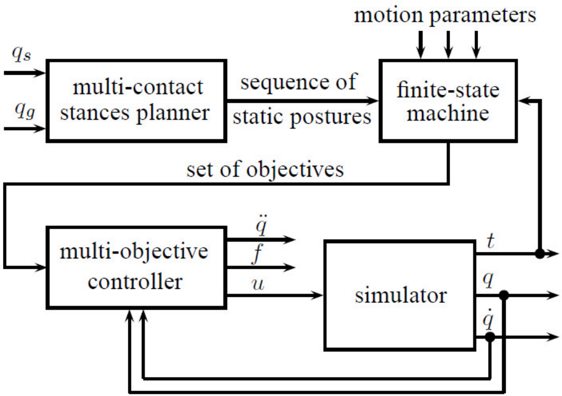
Our contributions

- Changing contact configurations
- Acyclic non-gaited motion

Restrictions

- Locomotion only

Overview


motion parameters
q_s
q_g
multi-contact stances planner
sequence of static postures
finite-state machine
set of objectives
multi-objective controller
$\ddot{q}$
f
u
simulator
t
q
$\dot{q}$

Multi-objective Control

Quadratic objective

$$\min_{\ddot{q},\lambda,\tau} \sum_i w_i \, ||\ddot{g}_i - \ddot{g}_i^{\text{desired}}||^2$$

Multi-objective Control

Quadratic objective

$$\min_{\ddot{q},\lambda,\tau} \sum_i w_i \, ||\ddot{g}_i - \ddot{g}_i^{\text{desired}}||^2$$

Linear constraints

- Dynamics equation

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

- Maintaining Contacts

$$J\ddot{q} + \dot{J}\dot{q} = 0$$

Finite state machine

Kinds of objectives

- regulation around a fixed point
- steering to a target position/velocity in a given time

applied to

- CoM (when removing a contact)
- Link going to make a contact
- posture to solve redundancy

Finite state machine

Kinds of objectives

- regulation around a fixed point
- steering to a target position/velocity in a given time

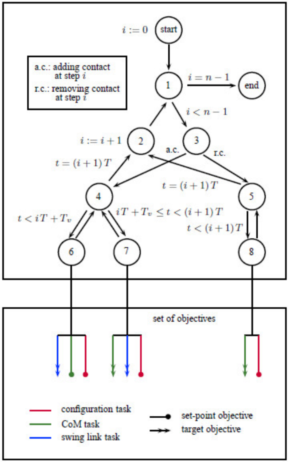
applied to

- CoM (when removing a contact)
- Link going to make a contact
- posture to solve redundancy

Finite state machine

At every time step of the simulation a finite state machine feeds the controller with the objectives depending on the current state along the sequence of stances

the FSM


start
$i := 0$
a.c.: adding contact at step i
r.c.: removing contact at step i
1
end
$i = n - 1$
$i < n - 1$
2
3
$i := i + 1$
a.c.
r.c.
$t = (i + 1)T$
$t = (i + 1)T$
4
5
$t < iT + T_v$
$iT + T_v \leq t < (i + 1)T$
$t < (i + 1)T$
6
7
8
set of objectives
configuration task
CoM task
swing link task
set-point objective
target objective

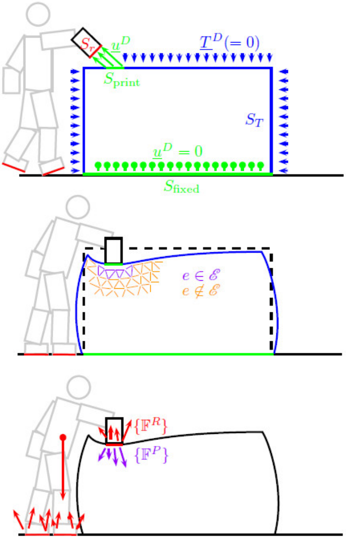
Example videos (external player)

Outline

Problem

solve the IK problem in case one of the contacts is made on a deformable support.

Overview


S_r
$\underline{u}^D$
$\underline{T}^D(=0)$
S_{print}
S_T
$\underline{u}^D = 0$
S_{fixed}
$e \in \mathscr{E}$
$e \notin \mathscr{E}$
$\{\mathbb{F}^R\}$
$\{\mathbb{F}^P\}$

Linear elasticity equations

$$\underline{\underline{\varepsilon}}(\underline{x}) = \frac{1}{2}\left(\nabla \underline{u} + \nabla^T \underline{u}\right)(\underline{x}),$$

$$\operatorname{div} \underline{\underline{\sigma}}(\underline{x}) + \rho\, \underline{f}(\underline{x}) = 0,$$

$$\underline{\underline{\sigma}}(\underline{x}) = \underline{\underline{\underline{\underline{A}}}} : \underline{\underline{\varepsilon}}(\underline{x}),$$

Linear elasticity equations

$$\underline{\underline{\varepsilon}}(\underline{x}) = \frac{1}{2}(\nabla \underline{u} + \nabla^T \underline{u})(\underline{x}),$$
$$\mathrm{div}\, \underline{\underline{\sigma}}(\underline{x}) + \rho\, \underline{f}(\underline{x}) = 0,$$
$$\underline{\underline{\sigma}}(\underline{x}) = \underline{\underline{\underline{\underline{A}}}} : \underline{\underline{\varepsilon}}(\underline{x}),$$

Boundary conditions

$$\underline{\underline{\sigma}}(\underline{x}).\underline{n}(\underline{x}) = \underline{T}^D(\underline{x}) \quad (\underline{x} \in S_T),$$
$$\underline{u}(\underline{x}) = \underline{u}^D(\underline{x}) \quad (\underline{x} \in S_u),$$

Weak formulation

$$\int_{\Omega} \underline{\varepsilon}[\underline{u}] : \underline{\underline{\underline{\underline{\mathcal{A}}}}} : \underline{\varepsilon}[\underline{w}] \, dV = \int_{\Omega} \rho \underline{f}.\underline{w} \, dV + \int_{S_T} \underline{T}^D.\underline{w} \, dS \, ,$$

Weak formulation

$$\int_{\Omega} \underline{\varepsilon}[\underline{u}] : \underline{\underline{\underline{\underline{\mathcal{A}}}}} : \underline{\varepsilon}[\underline{w}]\, dV = \int_{\Omega} \rho \underline{f}.\underline{w}\, dV + \int_{S_T} \underline{T}^D.\underline{w}\, dS\,,$$

Approximation using the Galerkin method

$$[\mathbb{K}^F]\{\mathbb{U}^F\} + [\mathbb{K}^D]\{\mathbb{U}^D\} = 0\,,$$

which can be rewritten as

$$\{\mathbb{U}\} = [\mathbb{K}]\{\mathbb{U}^D\}\,.$$

Weak formulation

$$\int_{\Omega} \underline{\varepsilon}[\underline{u}] : \underline{\underline{\underline{\underline{\mathcal{A}}}}} : \underline{\varepsilon}[\underline{w}] \, dV = \int_{\Omega} \rho \underline{f} . \underline{w} \, dV + \int_{S_T} \underline{T}^D . \underline{w} \, dS \,,$$

Approximation using the Galerkin method

$$[\mathbb{K}^F]\{\mathbb{U}^F\} + [\mathbb{K}^D]\{\mathbb{U}^D\} = 0 \,,$$

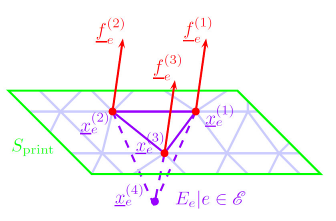
which can be rewritten as

$$\{\mathbb{U}\} = [\mathbb{K}]\{\mathbb{U}^D\} \,.$$

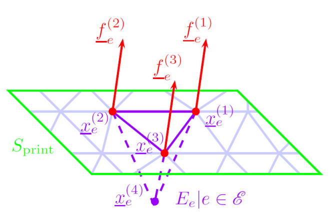
the FEM method

the FEM method allows to compute $[\mathbb{K}]$ through computing some other intermediate matrices $[D], [A], [B_e], [\mathbb{K}_e]$ (please see paper for the details)

Nodal Reaction Forces



Nodal Reaction Forces



Problem

relate the nodal reaction forces f_e to the deformation $\{\mathbb{U}^D\}$ in order to compute the gradient of the new static-equilibrium constraint.

The analytic relation

$$\underline{f}_e^{(j)}(q) = -\frac{\alpha_e}{3}\left(\underline{n}_e(q)^T \otimes I_{3\times 3}\right)[D][A][B_e][\mathbb{K}_e]\left\{\mathbb{U}^D(q)\right\}.$$

The analytic relation

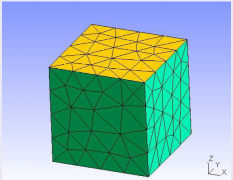
$$\underline{f}_e^{(j)}(q) = -\frac{\alpha_e}{3}\left(\underline{n}_e(q)^T \otimes I_{3\times 3}\right)[D][A][B_e][\mathbb{K}_e]\{\mathbb{U}^D(q)\}\,.$$

Its gradient

$$\frac{\partial \underline{f}_e^j}{\partial q} = \\ -\frac{\alpha_e}{3}\left[\left(\left[\frac{\partial \underline{n}_e}{\partial q_i}\right]^T \otimes I_{3\times 3}\right)[D][A][B_e][\mathbb{K}_e]\{\mathbb{U}^D\}\right]_{i=1}^{\mathrm{dim(q)}} \\ -\frac{\alpha_e}{3}\left(\underline{n}_e^T \otimes I_{3\times 3}\right)[D][A][B_e][\mathbb{K}_e]\frac{\partial\{\mathbb{U}^D\}}{\partial q}\,.$$

Example

the deformable support



physical properties

Young's modulus E	$10^6\,\mathrm{Pa}$
Poisson's ratio ν	0.4
Mass density ρ	$10^3\,\mathrm{kg/m^3}$

Example videos (external player)

conclusion

next

- More dynamic motion (reduced-model planning phase)
- Generalize motion generation to manipulation problems
- plan sequence of steps on linear elastic deformable support
- generate motion on deformable support
- ...

End of the presentation

Any questions?