The problem - The vision

Problem 1: Variability according to the customers

Problem 2: Variability in the realization (humans)

Problem 3: Variability in the delivery

Problem 4: Collaboration with humans

Problem 5: Certification

Behavior - Game theory

Rules of the game to constrain the actions

The game score

$$\begin{cases} \min f(\mathbf{u}(t), \mathbf{v}(t)) \\ \mathbf{g}(\mathbf{u}(t), \mathbf{v}(t)) < 0 \\ \mathbf{h}(\mathbf{u}(t), \mathbf{v}(t)) = 0 \end{cases}$$

Actions

Game and other agent representation

Behavior - Game theory

What are the necessary and meaningful constraints ?

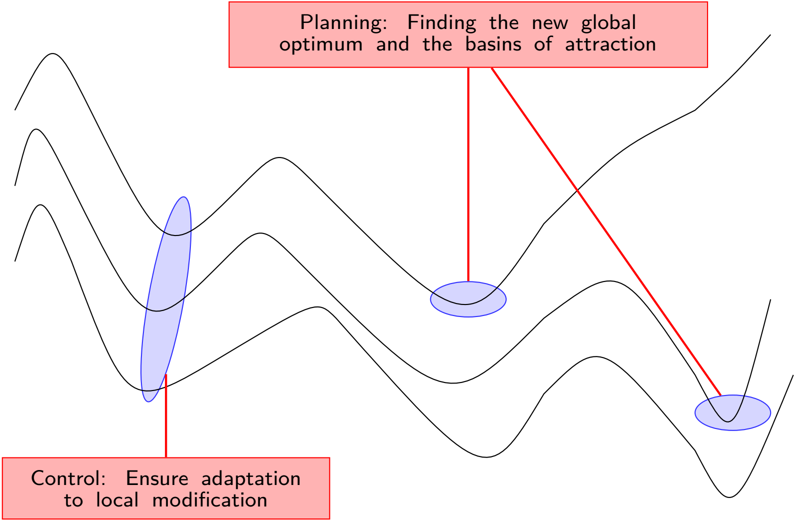
How to build the cost function for the behavior of interest ?

$$
\begin{cases}
\min f(\mathbf{u}(t), \mathbf{v}(t)) \\
\mathbf{g}(\mathbf{u}(t), \mathbf{v}(t)) < 0 \\
\mathbf{h}(\mathbf{u}(t), \mathbf{v}(t)) = 0
\end{cases}
$$

How to deal with such a huge search space ?

How to build an efficient world representation ?

Approach


Planning: Finding the new global optimum and the basins of attraction
Control: Ensure adaptation to local modification

Autonomous Motion generation: the problem

$$\begin{cases} \min f(\mathbf{u}(t), \mathbf{v}(t)) \\ \mathbf{g}(\mathbf{u}(t), \mathbf{v}(t)) < 0 \\ \mathbf{h}(\mathbf{u}(t), \mathbf{v}(t)) = 0 \end{cases}$$

[Saidi,IROS 2007]

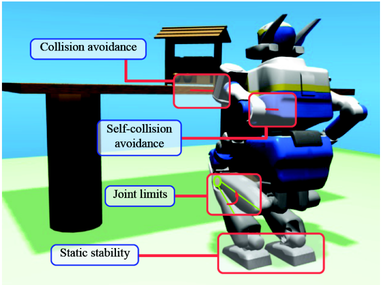
$\mathbf{f}(t)$: The cost function

$\mathbf{u}(t)$: The control vector

$\mathbf{g}(t)$: The inequality constraints

$\mathbf{h}(t)$: The equality constraints

$\mathbf{v}(t)$: The environment representation



Autonomous Motion generation: the problem

$$\begin{cases} \min_{\mathbf{q}(\cdot),\mathbf{u}(\cdot)} \quad \int_0^T L(t,\mathbf{q}(t),\mathbf{u}(t))dt \\ \dot{\mathbf{q}}(t) = f(\mathbf{q}(t),\mathbf{u}(t)), \quad t \in [0,T] \\ \mathbf{q}(0) = \mathbf{q}_0, \ T > 0 \\ 0 \leq \mathbf{gh}(\mathbf{q}(t),\mathbf{u}(t)), \quad t \in [0,T] \end{cases}$$

[Saidi,IROS 2007]

f: The system dynamics

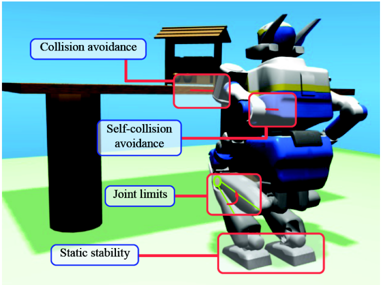
u: The control vector

q: The system state

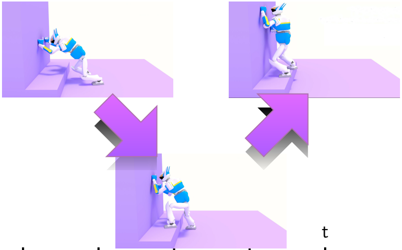
gh: The (in)equality constraints

v: The environment representation

L: The cost



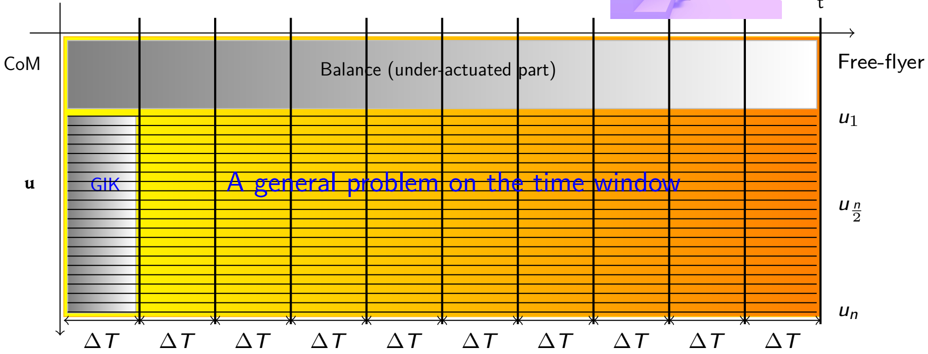
Motion generation: Model Predictive Control



- Size of the problem

$$1.6 \times 200 \times 30 = 9600 \text{ variables}$$

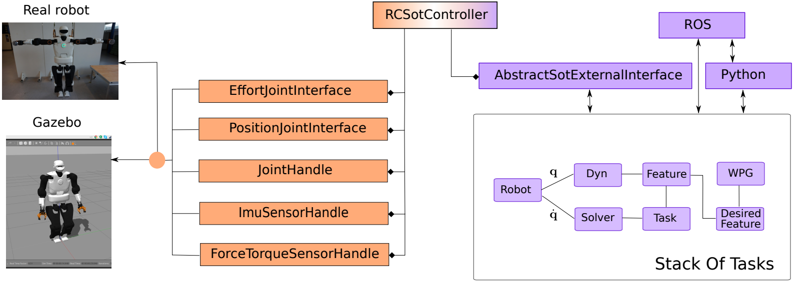
- Non linear constraints
- Discrete nature due to contacts



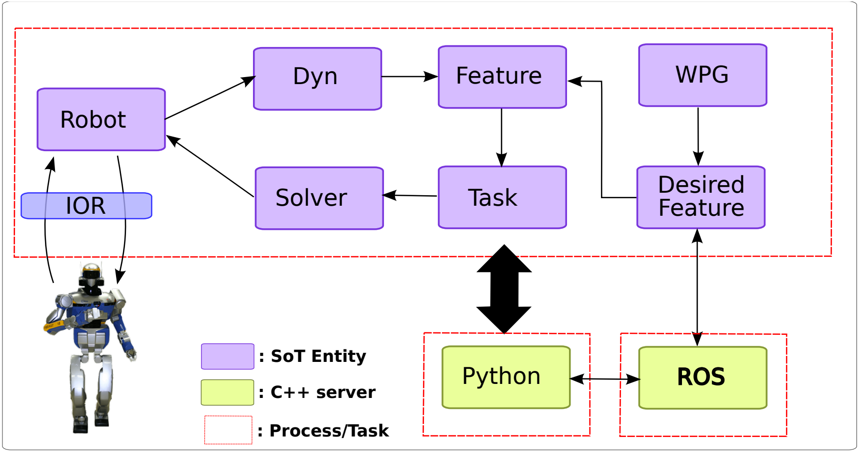
MuJoCo [Koenemann, IROS 2015]

https://www.youtube.com/watch?v=WbsQBPzQakc

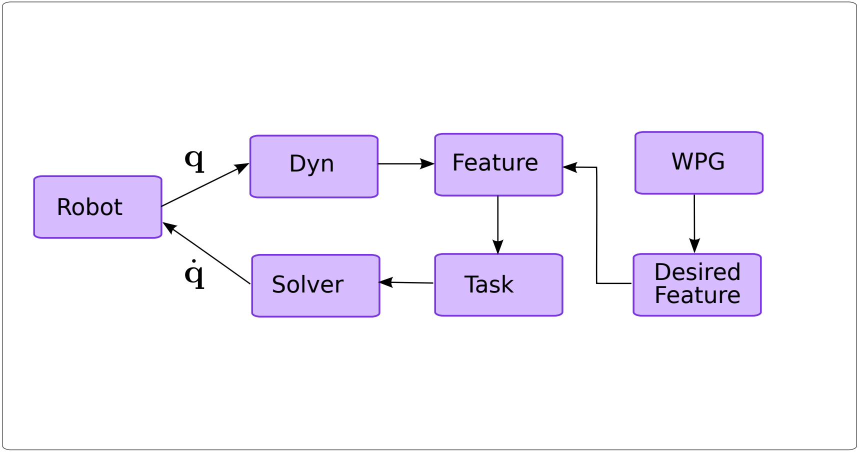
Structure de l'encapsulation



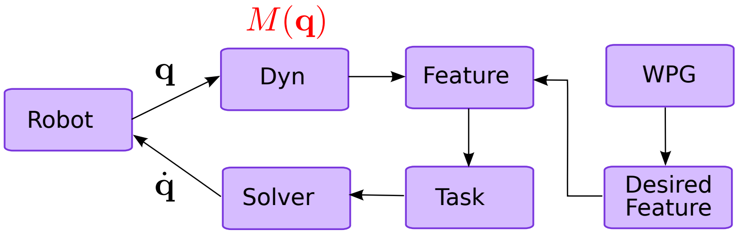
Software structure - Conceptual view



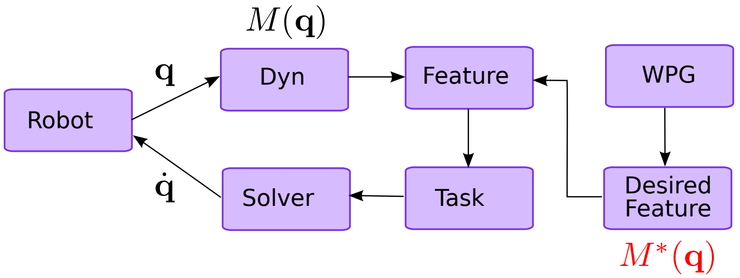
Software structure - Link with Model



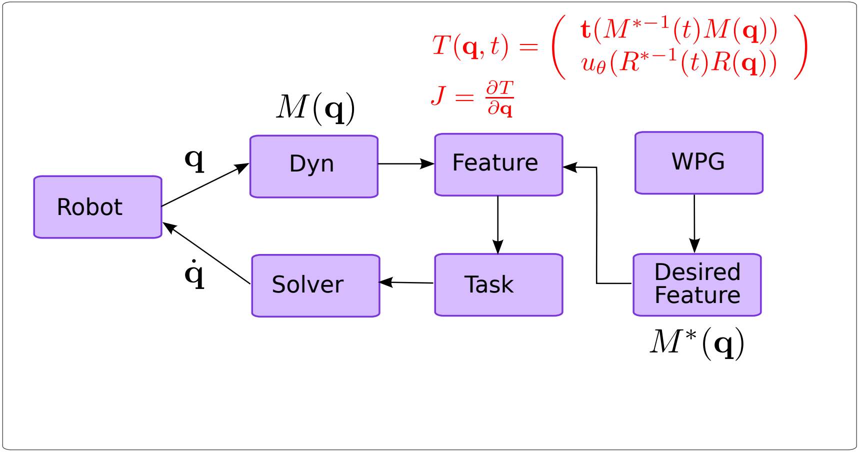
Software structure - Link with Model



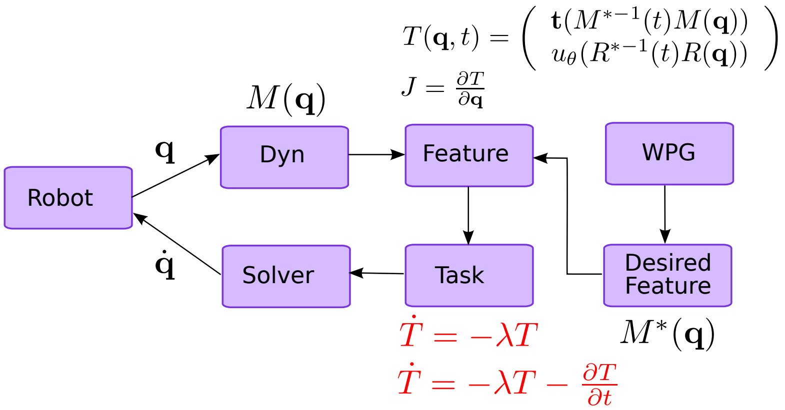
Software structure - Link with Model



Software structure - Link with Model



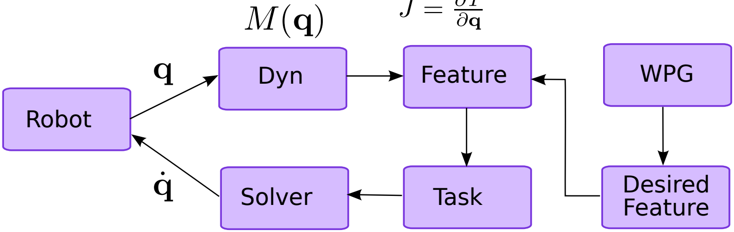
Software structure - Link with Model



Software structure - Link with Model

$$T(\mathbf{q},t) = \begin{pmatrix} \mathbf{t}(M^{*-1}(t)M(\mathbf{q})) \\ u_\theta(R^{*-1}(t)R(\mathbf{q})) \end{pmatrix}$$

$$J = \frac{\partial T}{\partial \mathbf{q}}$$

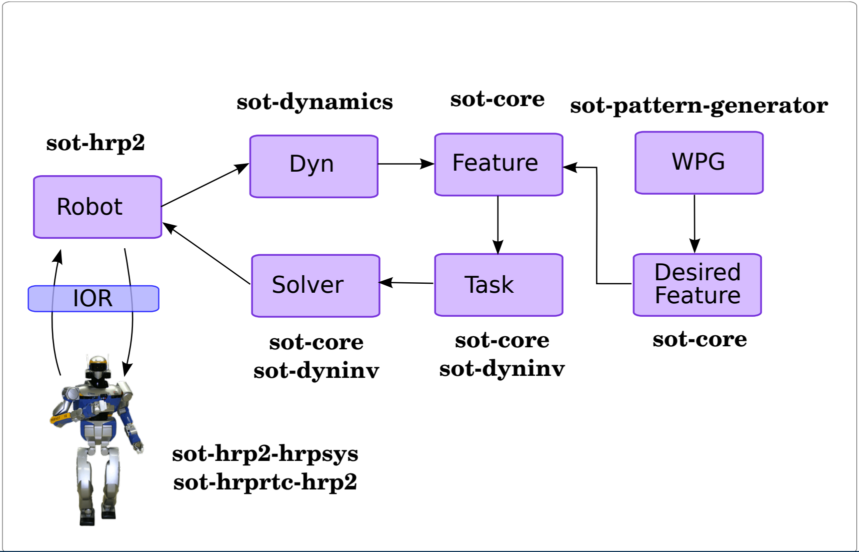


$$\dot{\mathbf{q}} \triangleq -J^{+}(\lambda T + \frac{\partial T}{\partial t})$$

$$\dot{T} = -\lambda T$$

$$\dot{T} = -\lambda T - \frac{\partial T}{\partial t}$$

Software structure - Repositories



Example

Solvers

Inverse Dynamics

Weighted Pseudo Inverse

- Faster (!?) computation
- Easier to formulate
- Do not guarantee convergence
- Difficulty to tune the weights
- Do not handle properly inequalities

Hierarchical Quadratic Program

- Slower (!?) computation time
- Warranty on priority
- Handle easily inequalities
- Difficult to formulate (here hidden in the solver)
- Known problems with cycles and singularities management

Conclusions

Pros

- Generic to put instantaneous controller together
- Allow code reusability
- Real-time performance
- Adapted to complex applications
- Binary packages support
- Eigen support

Cons

- Research code
- The learning curve seems to be yet steep