Intelligent Robot Control

Lecture 7: Admittance control and virtual guides

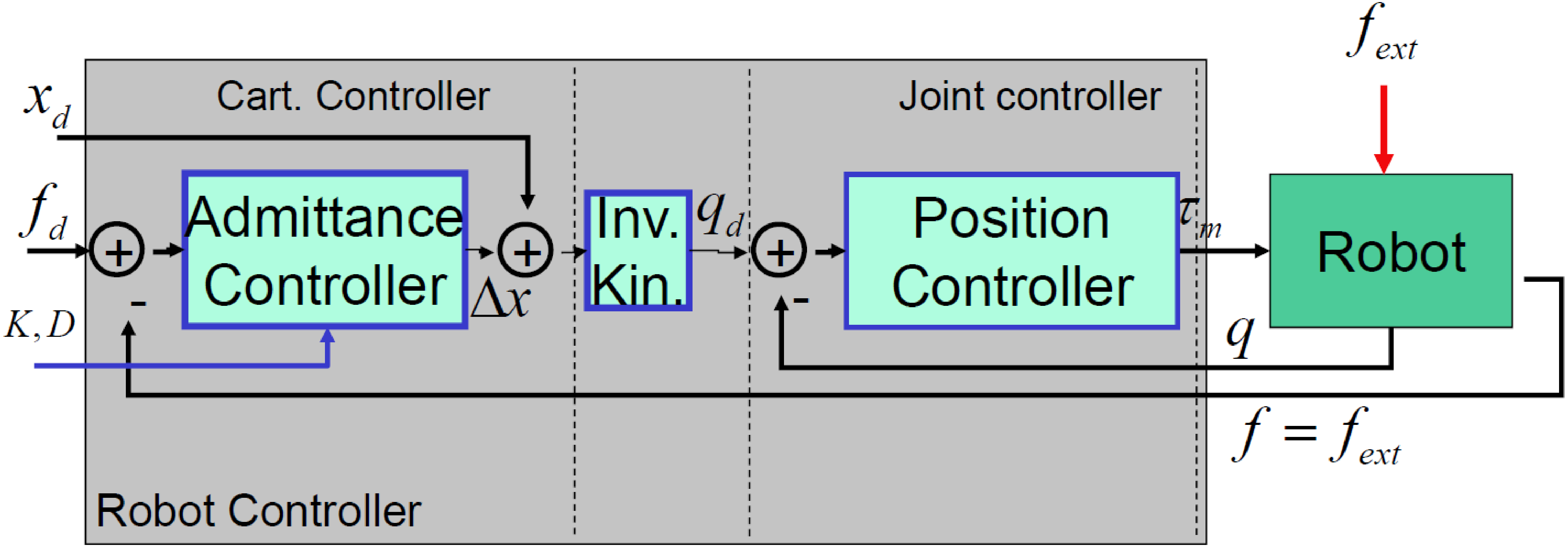
Tadej Petrič

tadej.petric@ijs.si

Institut
"Jožef Stefan"
Ljubljana, Slovenija

Admittance control

Principle Admittance Control



- Inner loop: high stiffness
- Outer loop: compliant

- Good behaviour for large and medium stiffness
- Good positioning accuracy
- Limitations for small stiffness (large compliance)
- Stability problems in contact

Institut "Jožef Stefan" Ljubljana, Slovenija

Admittance control and stability

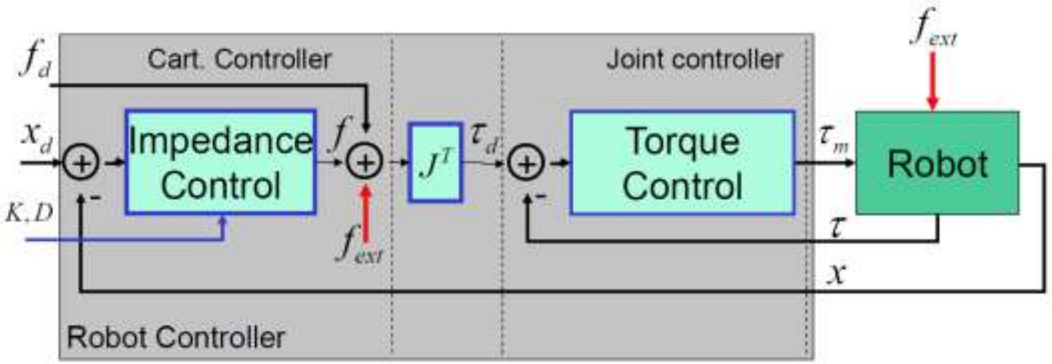
- Admittance control dynamics is bounded by the robot dynamics.

- Admittance control provides high level of accuracy in non-contact tasks but can result in instability during dynamic interaction with stiff environments.

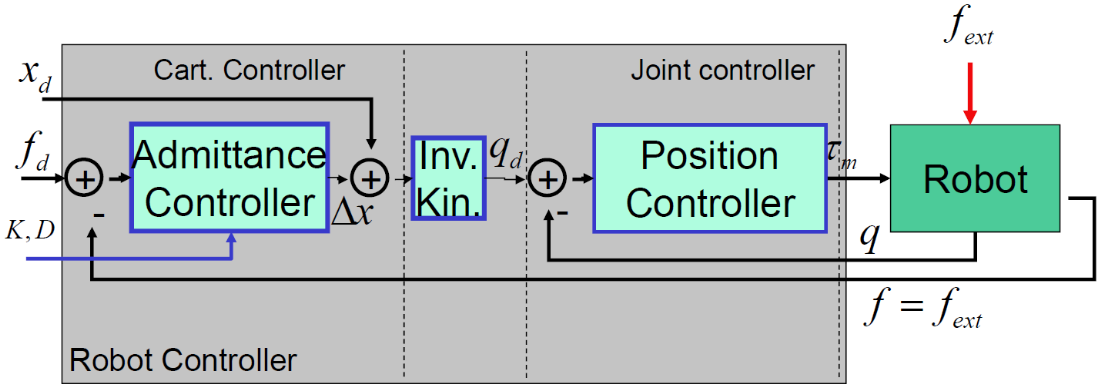
Institut
"Jožef Stefan"
Ljubljana, Slovenija

Impedance vs Admittance control

Impedance



Admittance



Very good performance when the environment is stiff

Poor accuracy when the environment is soft

Very good performance for soft environments

Instability for stiff environments

Impedance versus Admittance

Impedance:

- The motion produces forces, which are then felt by the operator.
- More suitable when the robot is in contact with stiff environment.
- Impedance based constraints generate forces to nullify the motion that violates the constraint.
- Must be highly backdriveable so that the environment (or human user) is able to move them freely when there is no controller force

Admittance:

- The forces applied to the robot by the operator contribute to the robot motion.
- Advantages when the environment is more compliant.
- Admittance based constraint prevent motion induced by external forces, which would violate the constraint.
- Highly stiff in response to environmental forces and only move when commanded by the controller.

Towards physical human-robot interaction

Physical human-robot interaction

- Advanced control system allow robots to operate fast, accurate and with persisting strength.
- They are fully autonomous in performing programmed tasks.
- Equipped with adequate sensory systems they can also operate in unstructured environments.
- There are limits in the autonomy of robots due the limitations in their perception of unpredicted situations.

$\Downarrow$

Human-robot setups, have definitely an advantage

Capabilities of modern robots + skills and judgement of a human

$\Downarrow$

Physical human-robot interaction (pHRI)

Institut
"Jožef Stefan"
Ljubljana, Slovenija

Human – robot cooperation

One type of human-robot interaction is **cooperation**, where we exploit mechanical capabilities of robotic devices and combine them with perception and cognitive capabilities of humans to achieve an overall goal.

- reduce human effort
- preserve high quality of task execution

Examples of human-robot collaboration:

Cobots (direct contact): a robot or an another intelligent device which assists a human to manipulate an object or supports a human in performing motion.

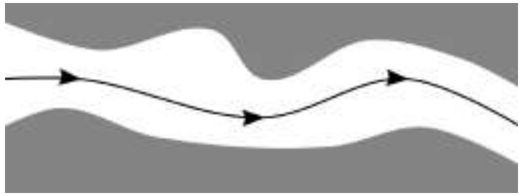
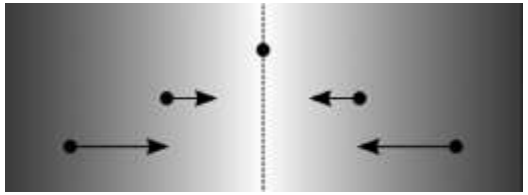
Telemanipulators (indirect contact): remotely controlled by the human operator in order to manipulate objects.

The operator is responsible for the task execution and the robot provides only the assistance by constraining the motion or guiding the human during the motion.

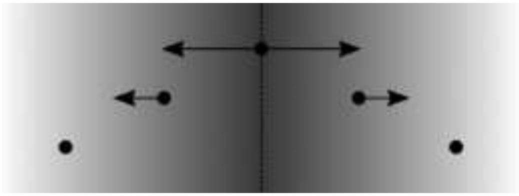
Virtual guides

Virtual guides (virtual fixtures): virtual constraints which influence the motion of a cooperative robotic system:

- **Guidance** virtual guides: force the operator to move toward a specific target point or to follow a predefined path

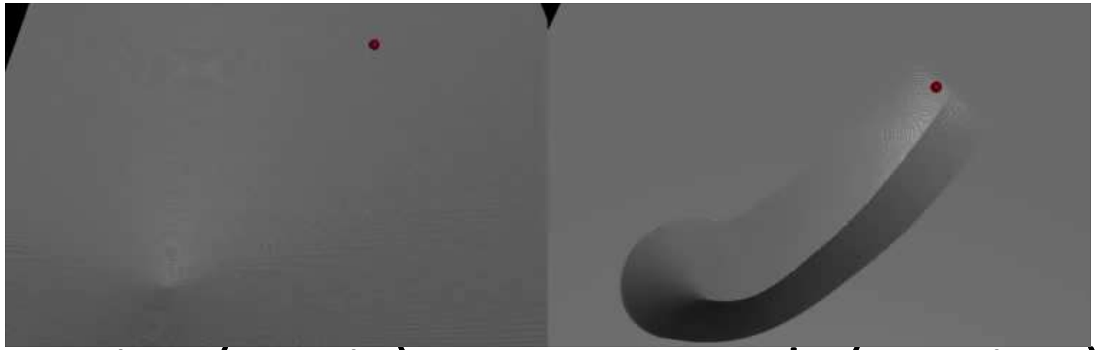
- **Forbidden-Region** virtual guides: prevent the operator to move into specific region

Similar function as real mechanical constraints (provide a surface that confines and/or guides a motion) except that they are implemented in the control algorithms.

Unifying virtual guides — Guidance

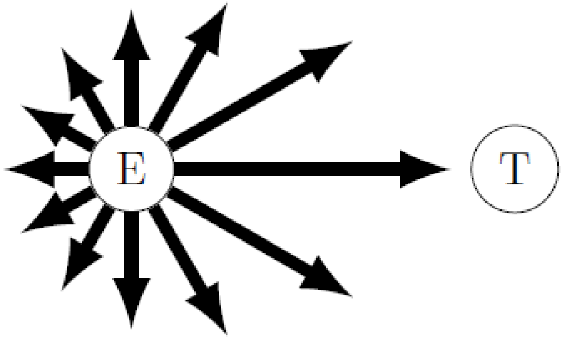
Target:



Point (static) Path (motion)

Strategies:

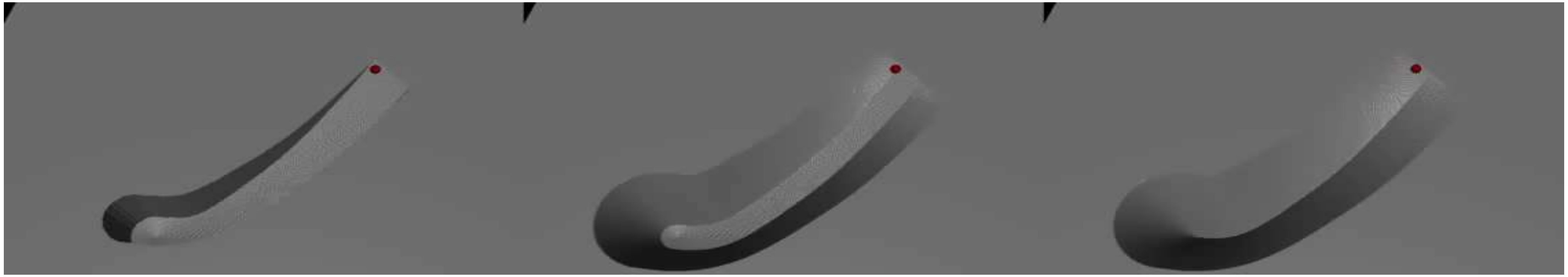
- **Pulling to the target**: stiffness, controlled position or velocity
- **Preferred direction**: damping



Institut
"Jožef Stefan"
Ljubljana, Slovenija

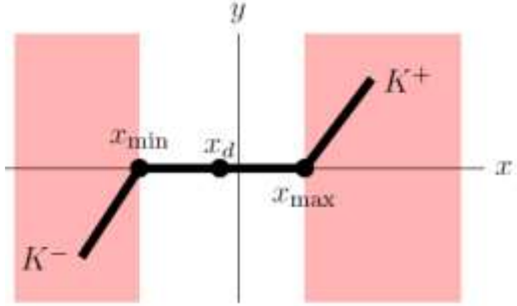
Unifying virtual guides — constraints

Preventing to move into restricted region



Hard constraints Soft constraints No deviation

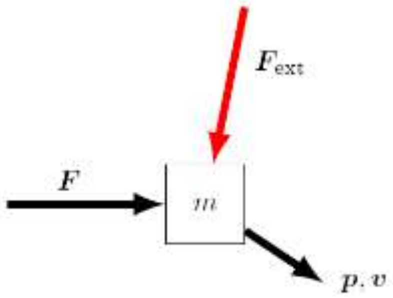
- **Hard guidance**: leaving the user with no or little freedom to deviate from the preferred (planned) path
- **Soft guidance**: give the user the freedom to move away from the path by allowing some motion in the non-preferred directions



$$y = \begin{cases} K^{+}(x - x_{\max}) & \text{for } x > x_{\max} \\ 0 & \text{otherwise} \\ K^{-}(x - x_{\min}) & \text{for } x < x_{\min} \end{cases}$$

Physical interaction

The physical human-robot comanipulation relies on **interaction forces** between the operator holding the robot tool and the robot.



$$m\dot{v} = F + F_{ext}$$

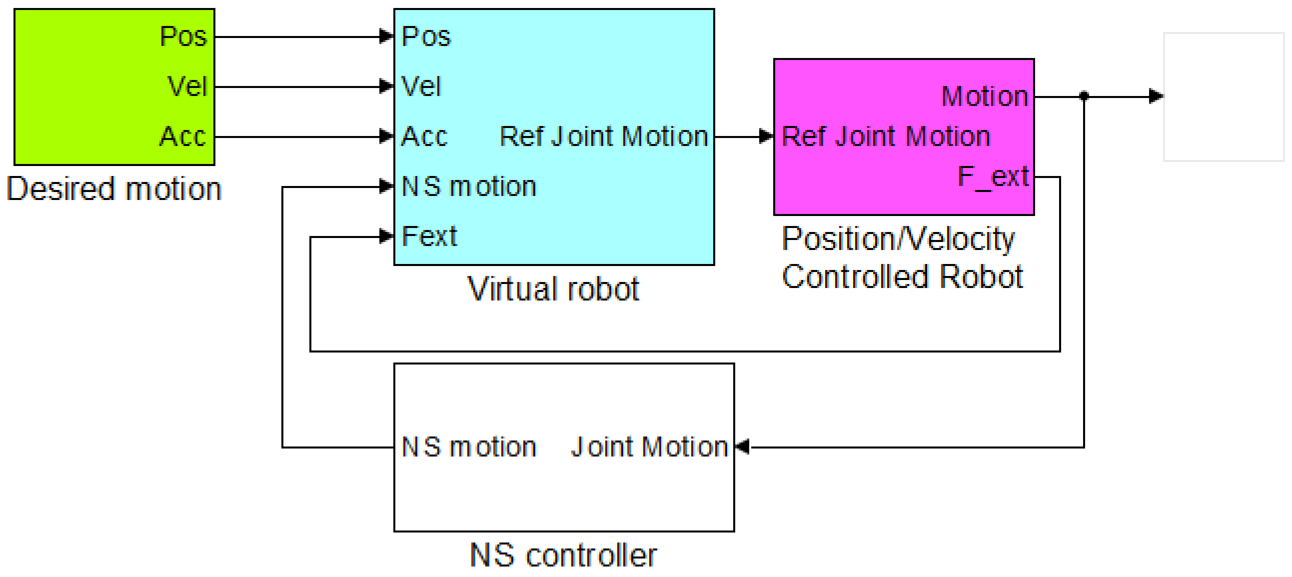
Two basic force/motion relationship types:

- Admittance: $A(s) = \dfrac{v(s)}{F(s)}$
- Impedance: $I(s) = \dfrac{F(s)}{v(s)}$

The admittance A(s) or impedance I(s) can be chosen differently in different directions.

Institut
"Jožef Stefan"
Ljubljana, Slovenija

Admittance top level control



$$\boldsymbol{\tau} = \mathbf{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) - \boldsymbol{\tau}_F$$

$$\ddot{e}_q + \mathbf{K}_v \dot{e}_q + \mathbf{K}_p e_q = 0$$

$$\boldsymbol{q} \approx \boldsymbol{q}_d$$

Virtual robot

Virtual robot dynamics is a double integrator $\dot{v} = \dot{v}_c$.

Admittance control:

- Acceleration level: $\dot{v}_c = \dot{v}_d + \mathbf{M}^{-1}(\mathbf{D}\dot{e}_x + \mathbf{K}e_x + F_{\mathrm{ext}})$

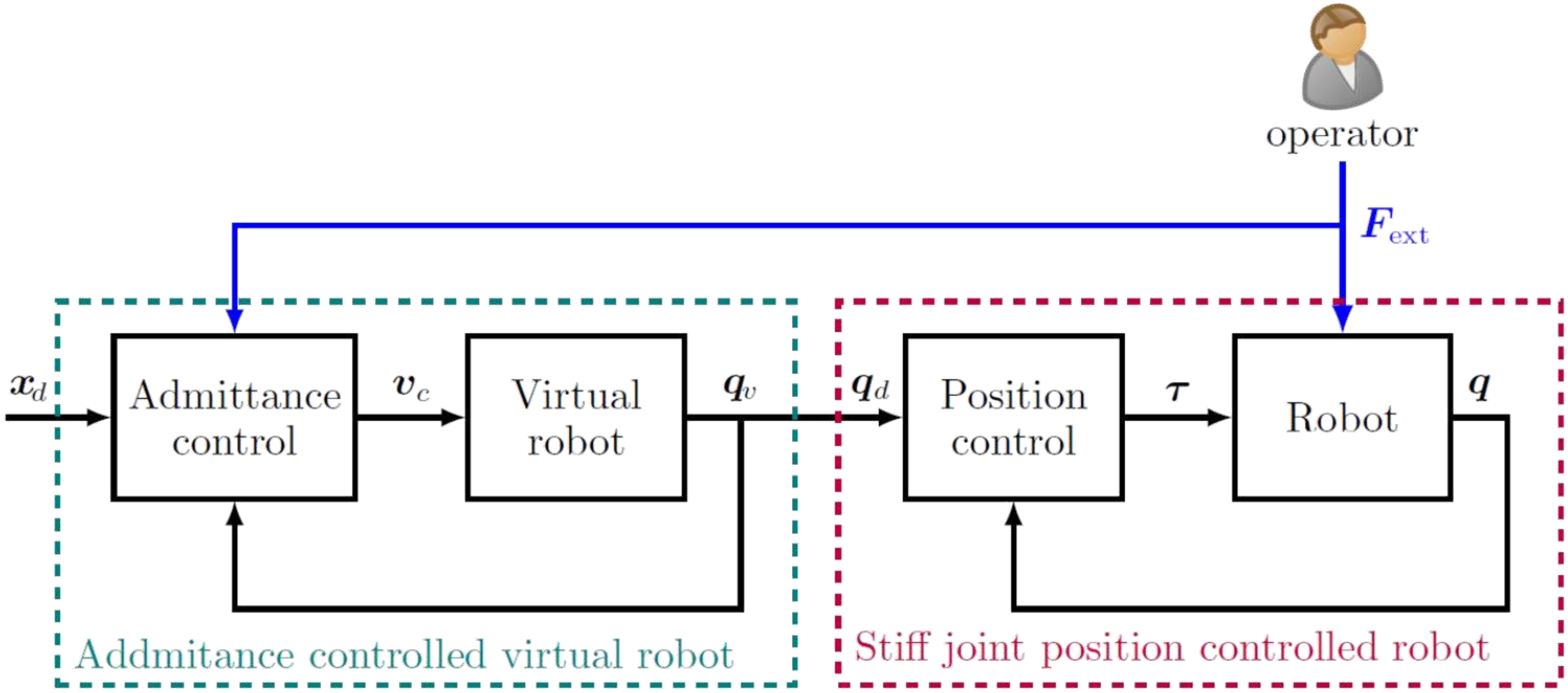
$$e_x = \begin{bmatrix} p_d - p \\ 2\log(\mathcal{Q}_d * \mathcal{Q}^{-1}) \end{bmatrix}$$

- Velocity level: $v_c = v_d + \frac{1}{\mathbf{D}}(\mathbf{K}e_x + F_{\mathrm{ext}})$

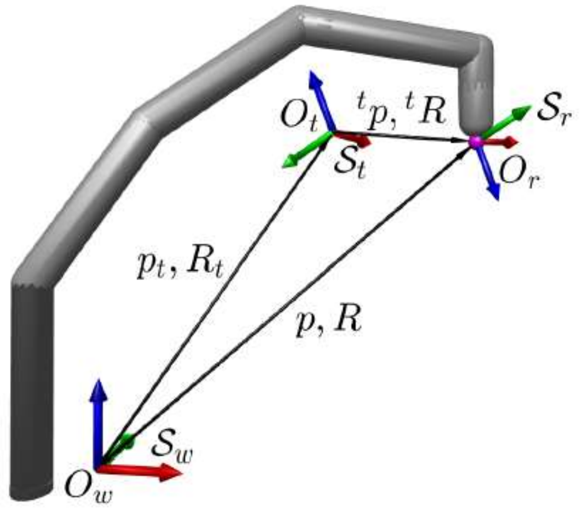
Inverse kinematics: $\dot{q}_d = \mathbf{J}^{\#}v_c + (\mathbf{I} - \mathbf{J}^{\#}\mathbf{J})\dot{q}_n$

$$\ddot{q}_d = \mathbf{J}^{\#}(\dot{v}_c - \dot{\mathbf{J}}\dot{q}) + (\mathbf{I} - \mathbf{J}^{\#}\mathbf{J})\ddot{q}_n$$

Admittance control on virtual robot



Control in task space



$$ {}^t\mathbf{T} = \begin{bmatrix} {}^t\mathbf{R} & {}^t\boldsymbol{p} \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_t^T & -\mathbf{R}_t^T \boldsymbol{p}_t \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \boldsymbol{p} \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_t^T \mathbf{R} & \mathbf{R}_t^T(\boldsymbol{p} - \boldsymbol{p}_t) \\ \mathbf{0} & 1 \end{bmatrix} $$

$$ {}^t\boldsymbol{v} = \widetilde{\mathbf{R}}_t^T \boldsymbol{v} - \widetilde{\mathbf{R}}_t^T \mathbf{J}_t \boldsymbol{v}_t $$

$$ \widetilde{\mathbf{R}}_t = \begin{bmatrix} \mathbf{R}_t & 0_{3\times3} \\ 0_{3\times3} & \mathbf{R}_t \end{bmatrix} \qquad \mathbf{J}_t = \begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{S}(\boldsymbol{p} - \boldsymbol{p}_t) \\ 0_{3\times3} & \mathbf{I}_{3\times3} \end{bmatrix} $$

$$ \dot{\boldsymbol{q}}_c = (\widetilde{\mathbf{R}}^T \mathbf{J})^{\#}\, {}^t\dot{\boldsymbol{x}}_c + (\mathbf{I} - (\widetilde{\mathbf{R}}^T \mathbf{J})^{\#} \widetilde{\mathbf{R}}^T \mathbf{J}) \dot{\boldsymbol{q}}_n $$

$$ {}^t\dot{\boldsymbol{x}}_c = \widetilde{\mathbf{R}}_t^T \mathbf{J}_t \boldsymbol{v}_d + \mathbf{K}_{p,t} \mathbf{R}_t^T \boldsymbol{e} $$

Unified constraints framework

Two sets of control design parameters and variables:

- **Dynamic parameters** (virtual stiffness, damping and inertia parameters): influence the feeling of the operator when moving the tool.
- **Desired motion** (position, velocity and acceleration): define the active constraints.

Scenario 1 – robot guides the operator to the target:

- Desired position = target; rop_d: actively influences the motion

Scenario 2 – operator is encouraged to move to the target:

- No position control (gain K=0); anisotropic damping (gain D)

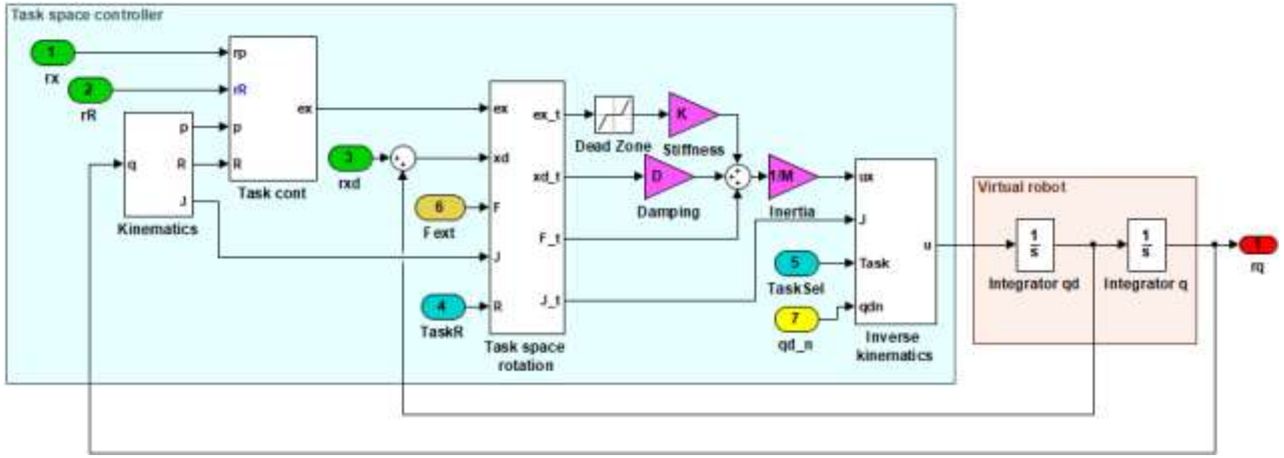
Scenario 3 – constraint should not influence the motion in a certain region:

- Dead-zones should be used for the relevant variables.

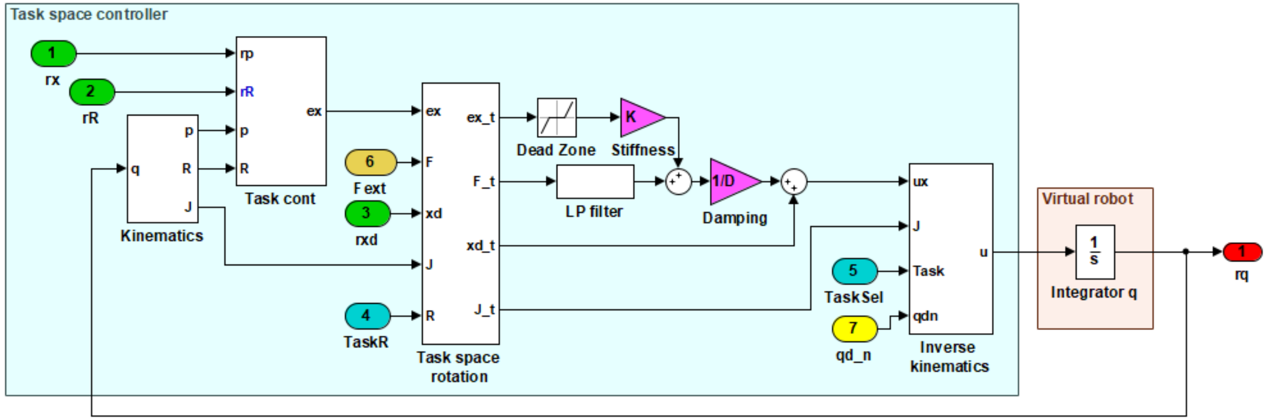
Note: Constraints can require a special behavior in some directions, which are not aligned with the Cartesian space.

Simulink – admittance control

Acceleration level:



Velocity level:



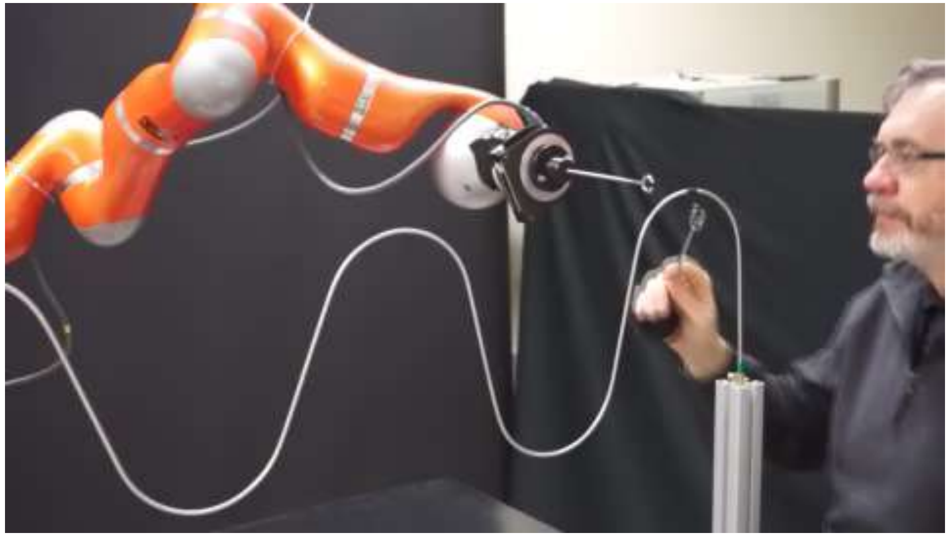
Institut "Jožef Stefan" Ljubljana, Slovenija

Unified framwork design steps

1. The space relevant for the task has to be identified (the DOFs needed to perform the task.
2. Define the states of the virtual robot which require different constraining actions.
3. Define the configuration of the constraints in each state. For most constraints this is the geometry (it can be also a velocity profile if the goal is to follow a velocity) and the control parameters (gains K, D, M), dead-zone parameters, and desired motion.
4. Map the control and constraints from the Cartesian into the suitable task space.

Buzz-Wire experiment

Buzz-Wire task: a ring has to be moved along curving wire without touching it. This is a typical path following task, which requires a certain level of mental effort from the human for achieving the goal.



To reduce the effort, a KUKA LWR robot arm with 7DOF should assist a human to do the task.

The path was encoded using radial basis functions (RBF).

Problem statement for tracking tasks

A possible sequence could be as follows:

- Human operator grasps the tool,
- The tool is moved closer to the path
- When is on the target path, the tool has to stay on that path.
- After the task is completed, the robot can be moved away.

During these steps the aim of the robot is to support and guide the human operator in each step.

As there might be some regions in the robot workspace which have to be avoided, the robot should prevent the tool to enter them.

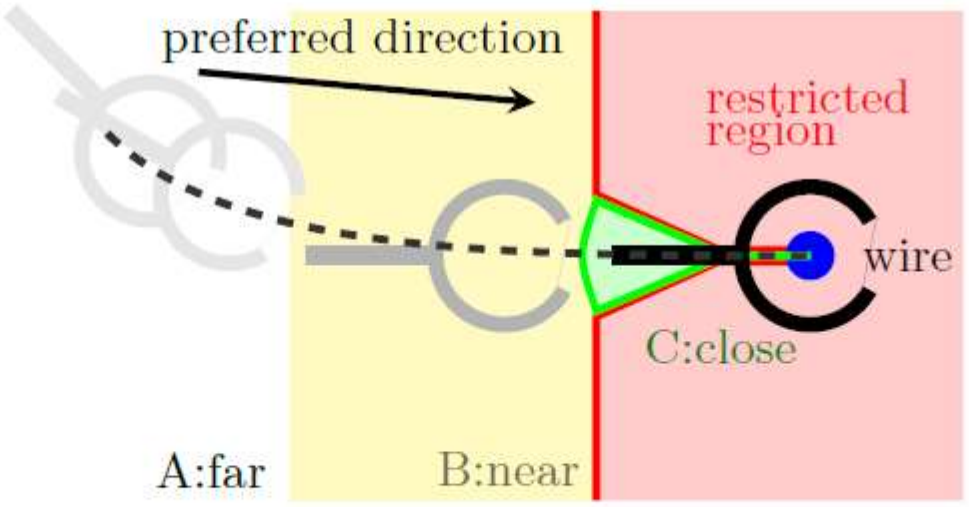
In general, it is not necessary to exploit all available DOFs of the robot in each situation. For example, when the tool is far from the target position, the orientation of the tool might not be important and the unused orientation DOFs can be utilized for some lower priority tasks like a pose optimization or obstacle avoidance.

Selection of virtual guides

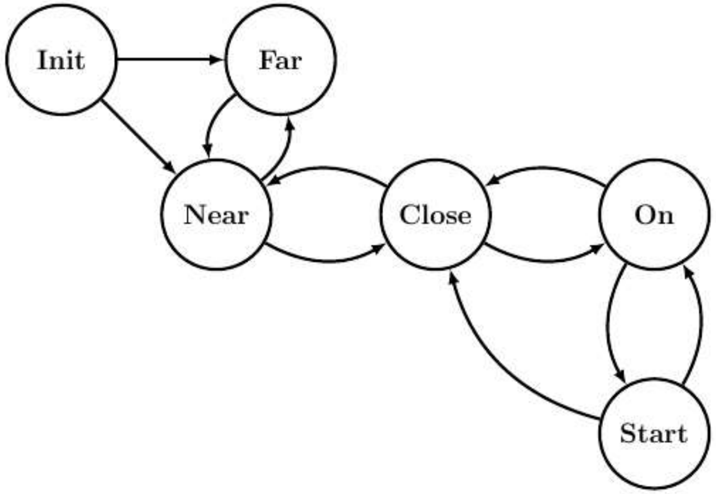
Different zones (states) depending on the motion complexity and requirements:

- Far (3DOF used)
- Near (6 DOF used)
- Close (6DOF used)
- On path (5 or 6 DOF used;

  task space is changing)



State machine:



Institut
"Jožef Stefan"
Ljubljana, Slovenija

Selection of virtual guides

Guidance:

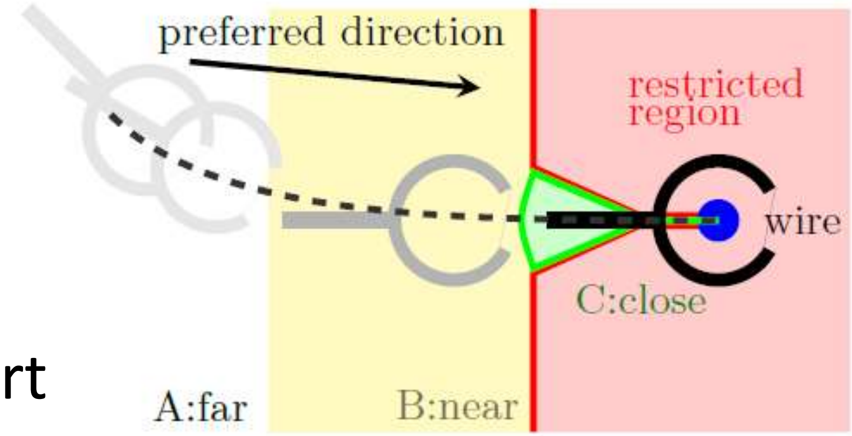
- Preferred direction toward the wire in zone Far, Near and Close (lower damping in this direction)

Restricted regions:

- Motion allowed only to reach the wire and not beyond (dead-zone for K) in zone Near
- Narrow passage in zone Close
- No deviation from path in zone On
- Moving off path only in zone Start

Dynamics:

- Low damping D to reduce human effort



Institut
"Jožef Stefan"
Ljubljana, Slovenija

Selection of task space

Task space:

- Cartesian for zone: Far, Near, Close.
- Path dependent for zone: On, Start.

DOF used in task space:

- 3 positions in zone: Far
- 3 positions and 3 rotations in zone: Near, Close
- 3 positions and 2/3 orientations in zone: On, Start

Null-space:

$$\dot{q}_n = \mathbf{K}_1(q_{\mathrm{opt}} - q) - \mathbf{K}_2\,\mathbf{J}^T F_{\mathrm{body}}$$

Experimental results – Motion far from wire

Task space:

- Cartesian, only positions

Null-space:

- 4 DOF

$$\dot{q}_n = \mathrm{K}_1(q_{\mathrm{opt}} - q) - \mathrm{K}_2\,\mathrm{J}^T F_{\mathrm{body}}$$

Guidance:

- Preferred direction toward the wire

Restricted regions:

- Practically no (border in another region)

Institut "Jožef Stefan" Ljubljana, Slovenija

Experimental results – Motion near wire

Task space:

- Cartesian, all 6DOF

Null-space:

- 1 DOF

$$\dot{q}_n = \mathbf{K}_1(q_{\text{opt}} - q) - \mathbf{K}_2\, \mathbf{J}^T \boldsymbol{F}_{\text{body}}$$

Guidance:

- Preferred direction toward the wire (lower damping in this direction)
- Preferred orientation depends on the closes point on path -hard constraint

Restricted regions:

- Tool can not move beyond the wire plane, except on wire.

Institut
"Jožef Stefan"
Ljubljana, Slovenija

Experimental results – Motion close to wire

Task space:

- Cartesian, all 6DOF

Null-space:

- 1 DOF

$$\dot{q}_n = \mathbf{K}_1(q_{\text{opt}} - q) - \mathbf{K}_2\, \mathbf{J}^T \boldsymbol{F}_{\text{body}}$$

Guidance:

- Preferred direction toward the wire (lower damping in this direction)
- Preferred orientation depends on the closes point on path -hard constraint

Restricted regions:

- Allowed region narrower closer to the wire.

Experimental results – Moving along the path

Task space:

- Path dependent, all positions
- Axis symmetric tool (1 RDOF)

Null-space:

- 1 or 2 DOF

$$\dot{q}_n = \mathbf{K}_1(q_{\mathrm{opt}} - q) - \mathbf{K}_2\,\mathbf{J}^T \boldsymbol{F}_{\mathrm{body}}$$

Guidance:

- Allowed position along the wire (low damping in this direction)
- Allowed orientation perpendicular to the path

Restricted regions:

- Moving off the wire (Hard constraint). Only at the start of path motion also in the gap direction allowed (to be able to move off the wire).

Using proxy

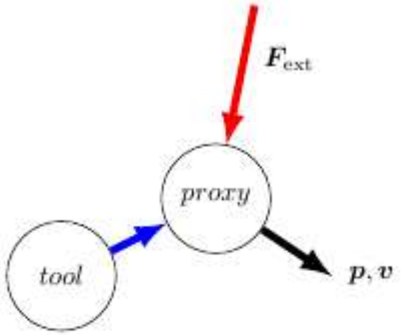
A proxy is an entirely simulated virtual object, which can have any dynamic properties required by the controller and constraints.

The robot tool is virtually connected to the proxy by some form of virtual linkage. The linkage model is typically elastic or viscoelastic, although it can take any form to give the constraint properties desired.

Operator is moving the proxy and the proxy moves the robot tool.

Path tracking:

- Proxy can move only in one dimension (on a path)
- Tool follows the proxy.



Experimental results – Using proxy for path tracking

Task space:

- 1 DOF, other task space directions are constraint to the path

Guidance:

- Proxy can move only along the wire

Behavior:

- Different dynamic properties of the proxy.

Experimental results – Using proxy for path tracking

Task space:

- 1 DOF, other task space directions are constraint to the path

Guidance:

- Proxy can move only along the wire
- Nominal path velocity defined

Operator can increase or decrease the path motion velocity.