Interactive Robotic Systems

Basics of modeling and control

Mathieu GROSSARD

Interactive Robotics Unit
CEA LIST - Nano-Innov
`mathieu.grossard@cea.fr`

Course content

Sessions on basics of modeling and control

- Hourly volume of 21h ;
- Lectures 1 to 5 (15h) and 2 tutorials (6h) using MATLAB®.

Main objectives for the lectures 1 to 5

- Master the terminology and fundamental concepts of robotics ;
- Master the formalism of representation for polyarticulated chains ;
- Master the methodology for computing the kinematic and dynamic models ;
- Master the methodology for identifying the dynamic model ;
- Master the most usual trajectory generation algorithms ;
- Master the main strategies for controlling robots.

Course organization

Syllabus

- **Session 1** Terminology, definitions and constituent elements of robots. Reminders on rigid-body motions (coordinates representation, rotation motions, homogeneous transformations, kinematic screw, static wrench).
- **Session 2** Direct geometric model, forward kinematic model and computation of the Jacobian matrix. Analysis of the robot workspace and the task space. Identification of singularity phenomena. Transmission of velocities and forces between joint and task spaces.
- **Session 3** Inverse geometric model and inverse kinematics model. Focus on inversion method for regular, singular and redundant robots.
- **Session 4** Tutorial #1 - Geometric and Kinematic models of manipulator arms.
- **Session 5** Direct and inverse dynamic models of rigid robots. Techniques for experimental identifying the dynamic model of robot. Extrapolations to the case of robots with flexible joints.
- **Session 6** Trajectory planning in the joint and task spaces. Overview of control techniques in position in the joint and Cartesian space. Overview of force control techniques.
- **Session 7** Tutorial #2 - Dynamic model and control of manipulator arms.

A few examples of robots


[Curiosity rover, NASA's Jet Propulsion Laboratory]


[Robot NAO, SoftBank Robotics]

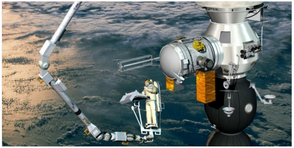

[Robot hand, Shadow Company]


[Robot Justin, DLR]

[Robot YuMi, ABB Robotics]


[Robot KUKA in a foundry]

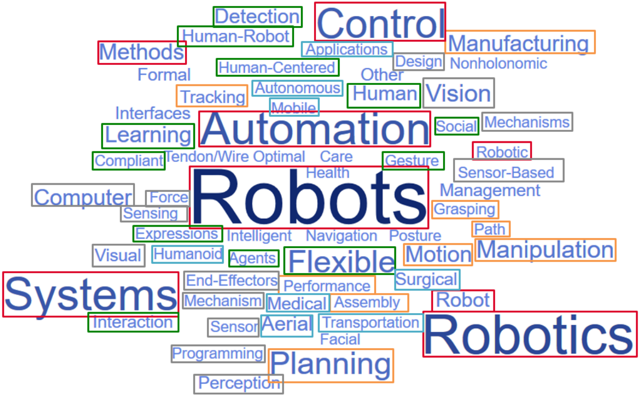

[Quadrupedal BigDog, Boston Dynamics]


[European Robotic Arm de l'ISS, Airbus Defence and Space]

[Robots Adept Quattro, Adept]

Issues in robotics

Main keywords associated with scientific articles presented at IROS in 2017
(*International Conference on Intelligent Robots and Systems, http ://www.iros2017.org*)



Predicted impact of robotics in the coming years

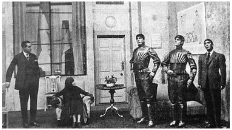
***Advanced robotics* and autonomous (or near-autonomous) vehicles will have *a potential annual economic impact by 2025* on a par with e.g. *mobile Internet*, *advanced materials* or *energy markets*.**

["Disruptive technologies: Advances that will transform life, business, and the global economy", McKinsey Global Institute May 2013]

A first approach of robotics

Robot and *Robotics*

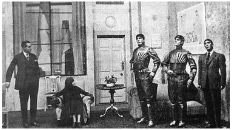
- Robot - "*R.U.R (Rossum's Universal Robots)*" from *Karel Capek* in 1920
  Science fiction play by the Czech writer *Karel Capek*
  - introduction of the word *robot* to the English language and to science fiction as a whole ;
  - magazine described the play as "thought-provoking" and a highly original thriller".



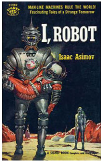
A first approach of robotics

Robot and *Robotics*

- Robot - "*R.U.R (Rossum's Universal Robots)*" from *Karel Capek* in 1920
  Science fiction play by the Czech writer *Karel Capek*
  - introduction of the word *robot* to the English language and to science fiction as a whole ;
  - magazine described the play as "thought-provoking" and a highly original thriller".



- Robotics - "*Runaround*" from *Isaac Asimov* in 1942
  Three Laws of Robotics : set of rules devised by the science fiction author *Isaac Asimov* [2]
  - "A robot may not injure a human being or, through inaction, allow a human being to come to harm"
  - "A robot must obey the orders given it by human beings except where such orders would conflict with the First Law"
  - "A robot must protect its own existence as long as such protection does not conflict with the First or Second Law"



Definition

Industrial robot (ISO - International Organization for Standardization - n° 8373)
automatically controlled, reprogrammable, multipurpose manipulator, programmable in three or more axes, which can be either fixed in place or mobile for use in industrial automation applications

The industrial robot includes : 1) the manipulator, including actuators, 2) the controller, including teach pendant and any communication interface (hardware and software).

Definition

Industrial robot (ISO - International Organization for Standardization - n° 8373)
automatically controlled, reprogrammable, multipurpose manipulator, programmable in three or more axes, which can be either fixed in place or mobile for use in industrial automation applications

The industrial robot includes : 1) the manipulator, including actuators, 2) the controller, including teach pendant and any communication interface (hardware and software).

Manipulator
multi-dofs mechanism consisting of a series of segments, jointed or sliding relative to one another

Mechanical structure

Kinematic/dynamic models

Multi-application
capable of being adapted to different applications

Task-related adaptation of the robot

Control objectives

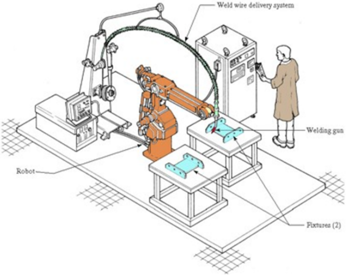
Automatically controlled, reprogrammable
changing programmed motions

Controller system

Control algorithms

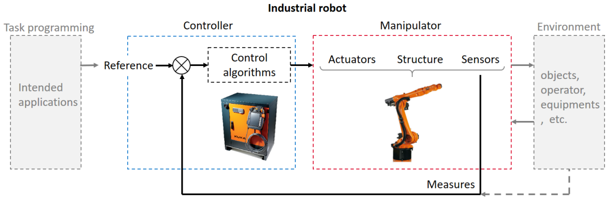
Components of a robotic cell

- Mechanism as articulated and actuated structure to act on the environment ;
- Perception capabilities help the robot to adapt to disturbances and unpredictable changes in its environment. ;
- Controller realizes the desired task objectives (generation of the input signals for the actuators as a function of the user's instructions and the sensor outputs) ;
- Communication interface through this the user programs the tasks that the robot must carry our ;
- Workcell and peripheral devices constitute the environment in which the robot works.

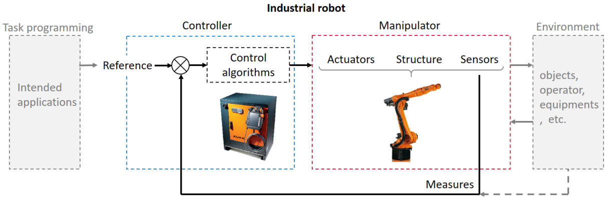


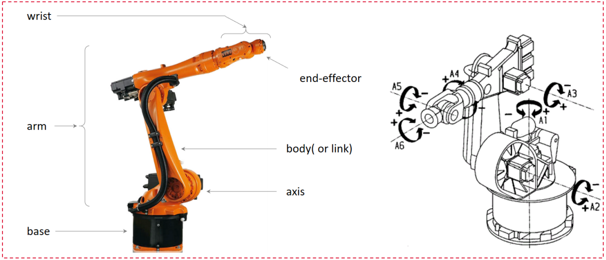
[Example of a robotic cell for arc welding, Kuka]

Terminology

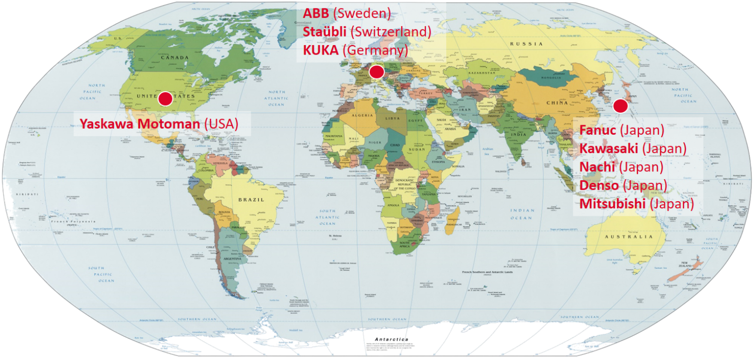


Terminology

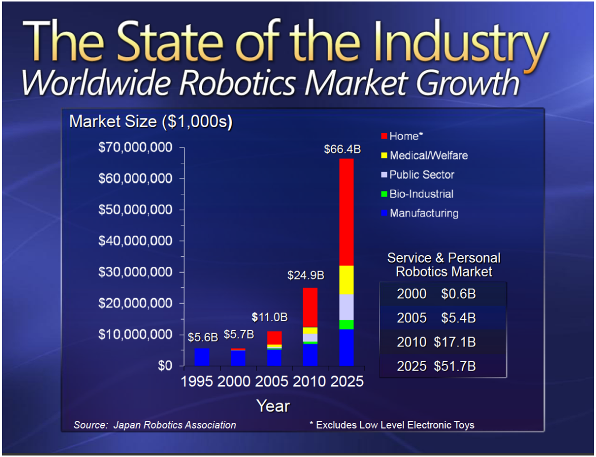




Some manufacturers of industrial robots

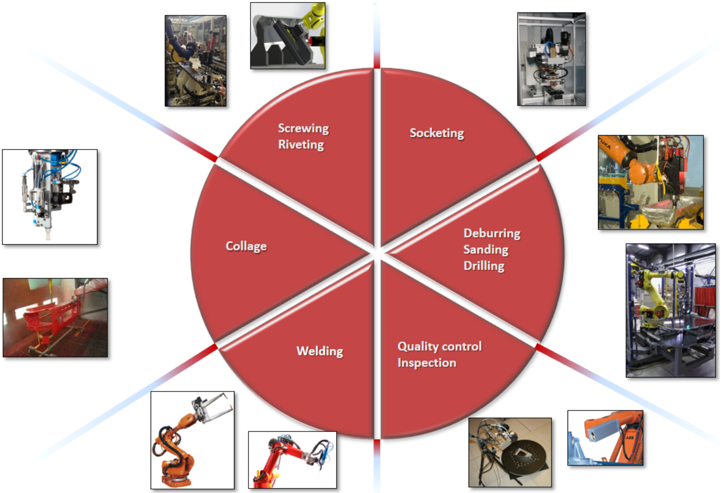


Forecast market for industrial robotics



(Cost of a 6-dof industrial robot in the order of several tens of k€)

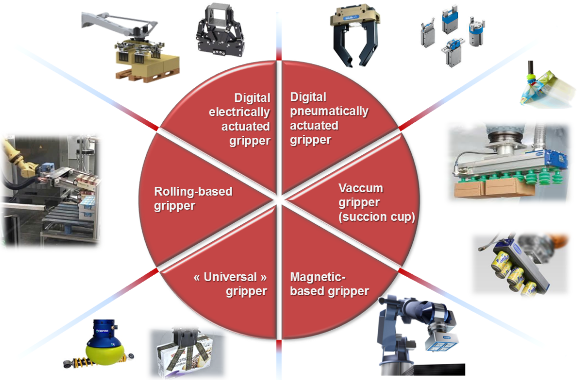
Some use-cases of robots in the manufacturing industry



Robot components

End-effector : device for handling objects (clamping devices, magnetic devices, vacuum devices, . . .) or processing them (tools, welding torch, spray gun, . . .).

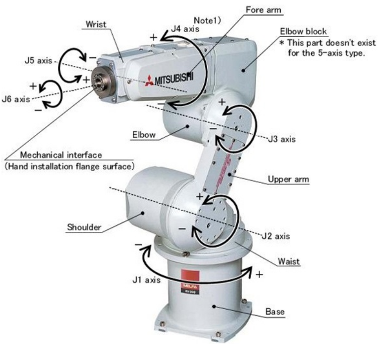
- Examples of some physical principles used for grasping objects

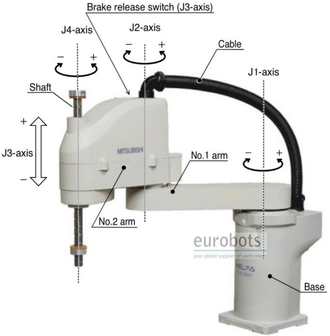


Robot components

Articulated mechanical structure or manipulator whose purpose is to bring the end-effector into a given situation (position and orientation), according to given speed and acceleration characteristics.

- Architecture described by a sequence of bodies called links, generally assumed to be rigidly connected by means of articulations called joints.

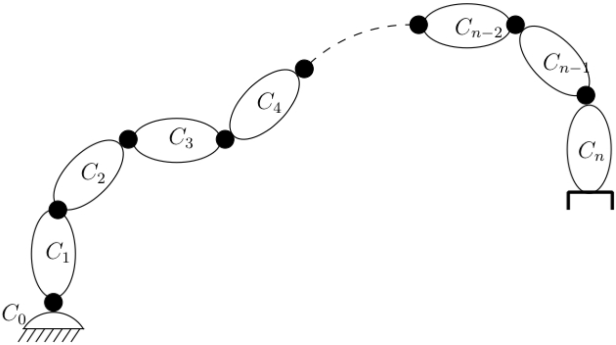




Typology of robot kinematic chains [13]

Simple open-tree structure :

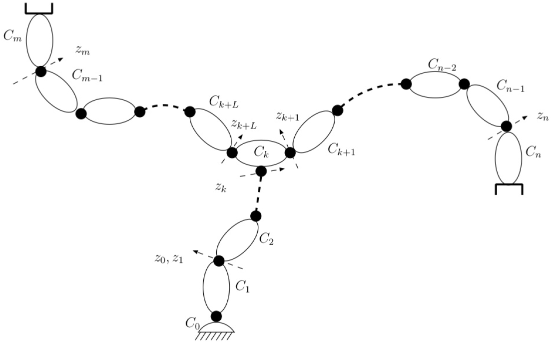
- Each link has at most two joints ;
- Purely sequential structure ;
- Most common architecture of manipulators.



Typology of robot kinematic chains [13]

Multiple open-tree structure :

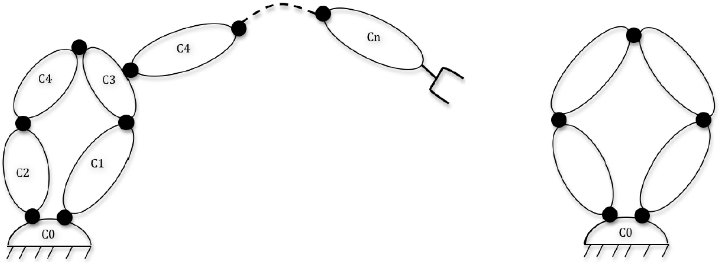
- Some links are connected to more than two other links ;
- No more possibility to number the links sequentially ;
- Possibility of having several end-effectors (like human body).



Typology of robot kinematic chains [13]

Closed chains containing kinematic loop(s) :

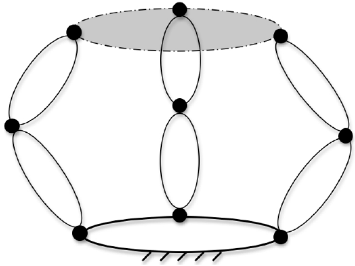
- Class of complex kinematic chains in which at least one of the links has more than two joints ;
- Class of simple closed chains such that all links have at most two joints.



Typology of robot kinematic chains [13]

Parallel robots in which the end-effector is connected to the base by several parallel chains :

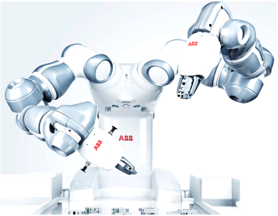
- Greater rigidity and precision ;
- Assembly tasks at high speeds ;
- High payload capacity compared to other robots.



Typology of robot kinematic chains [13]

Examples of robots from *ABB*^{TM} with different kinematic chains :

- Robot *IRB 2600* (simple open-tree structure) ;
- Robot *YuMi* (bimanual open-tree structure) ;
- Robot *IRB 4400* (complex closed chains) ;
- Robot *FlexPicker* (parallel robots).

Concept of Configuration (or Joint) Space

Degrees of freedom of a rigid body

- The minimum number of real-valued coordinates needed to represent the position and orientation of a rigid body is its number of degrees of freedom (*dof*).
  - General rule for determining the number of *dof* a rigid body :

$$dof = (\text{Sum of freedoms of the bodies}) - (\text{Number of independent constraints}) .$$

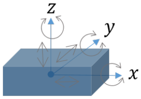
  - Alternative expression in terms of number of variables and independent equations that describe the system :

$$dof = (\text{Number of variables}) - (\text{Number of independent equations}) .$$

  - In summary :
    - a rigid body moving freely in 3-dimensional space has 6 degrees of freedom ;
    - a rigid body moving in a 2-dimensional plane has 3 degrees of freedom.

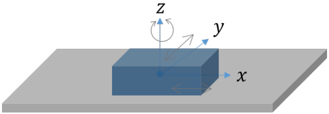
Free rigid body in 3D space – 6 dof

- 3 translational along x, y and z
- 3 rotational around x, y and z



Rigid body on a plane – 3 dof

- 2 translational along x and y
- 1 rotational around z



Concept of Configuration (or Joint) Space

Robot configuration

- The configuration of a robot is a complete specification of the position of all robot points relative to a fixed coordinate system.
  - robot links assumed to be made of rigid and known shape links, only a few variables needed to describe its configuration ;
  - variables aiming at describing the position and orientation of the links w.r.t. a frame of reference.
- The minimum number of real-valued coordinates needed to represent the configuration is the number of degrees of freedom of the robot.
  - since robots are constructed of rigid bodies :

$$dof = (\text{Sum of freedoms of the bodies}) - (\text{Number of independent constraints}) \, .$$

  - in practice, the joints constrain the motion of the rigid bodies, thus reducing the overall degrees of freedom of the robot.
- The space containing all possible configurations of the robot is called the Configuration (or Joint) space (C-space).

Robot joint

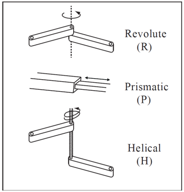
Definition

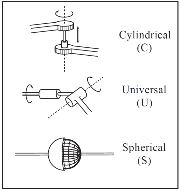
A joint connects two successive bodies by limiting the number of degrees of freedom of movement one from the other. Let f_i be the number of resulting degrees of freedom of the i^{th} joint, such that :

$$0 \leq f_i \leq 6.$$

- a joint can be viewed as providing freedoms f_i to allow one rigid body to move relative to another ;
- it can also be viewed as providing constraints c_i on the possible motions of the two rigid bodies it connects.

Symbolic representation of joints





Joint type	f_i	c_i (spatial case)
Revolute (R)	1	5
Prismatic (P)	1	5
Helical (H)	1	5
Cylindrical (C)	2	4
Universal (U)	2	4
Spherical (S)	3	3

When $f_i = 1$ (often the case in robotics), the i^{th} joint is said to be *simple*, either *revolute* or *prismatic* :

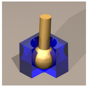
- Revolute joint It is a pivot joint, labeled R, reducing the relative motion between two bodies to a pure rotation around a common axis. The relative configuration between the two bodies is given by the angle around this axis.
- Prismatic joint It is a sliding joint, labeled P, reducing the relative motion between two bodies to a translation along a common axis. The relative configuration between the two bodies is measured by the distance along this axis.

Symbolic representation of joints

- Revolute joint (R)
  - Degree of freedom equal to 1 (one single rotation), number of constraints of 5 (3 translations, 2 rotations)
  - Articular value designating an *angle [rad]*

- Prismatic joint (P)
  - Degree of freedom equal to 1 (one single translation), number of constraints of 5 (2 translations, 3 rotations)
  - Articular value designating a *length [m]*

- More complex joints ($m \geq 2$)
  - Seen as a composition of *prismatic* and *revolute* joints ;
  - Example of a spherical (S) joint obtained with three *revolute* joints whose axes are concurrent.



Number of dof of a kinematical chain

Grübler's criterion

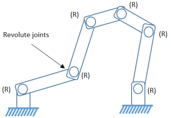
Consider a mechanism consisting of n_c rigid links (without the base body, which is assumed to be fixed) connected by j joints (let f_i and c_i be the number of degrees of freedoms and constraints provided by joint $i \in 1, \ldots, j$, such that $f_i + c_i = d$ with $d = 6$ for spatial mechanism and $d = 3$ for planar mechanism).
Then *Grübler*'s formula for the number of degrees of freedom of the robot is :

$$dof = dn_c - \sum_{i=1}^{j} c_i = d(n_c - j) + \sum_{i=1}^{j} f_i$$

- This index is obtained by counting the number of parameters defining the positions of a set of rigid bodies, reduced by the number of constraints imposed by the joints connecting these bodies ;
- This formula holds only if all joint constraints are independent. If they are not independent then the formula provides a lower bound on the number of dof.

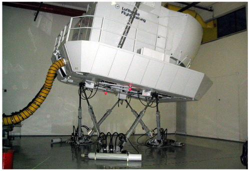
Number of dof of kinematical chains

Example n°1 : In-plane bipedal walking

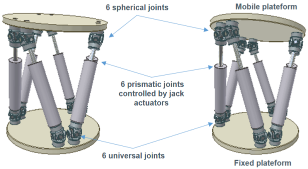


Example n°1

Example n°2 : *Stewart* platform



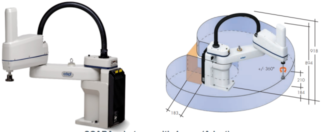
Stewart platform used as flight simulator by *Luftansa* company



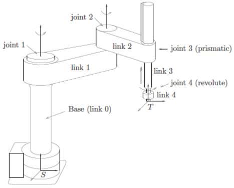
Example n°2

Number of dof of a simple open-tree structure

Example n°3 : simple open-tree structure



SCARA robot arm with 4 axes (Adept)



▸ Example n°3

If we limit ourselves to a simple open-tree structure made of kinematic pairs of types *R* and *P* for which $f_i = 1$ for all i, then the *Grübler* formula reduces further to

$$dof = n_c = j$$

- This formula means that the dof of the manipulator is then equal to its number of joints.
- Therefore, since 6 parameters are necessary to describe an arbitrary position and orientation in space for the end-effector, a simple open-tree robot manipulator possesses normally (at least) 6 joints of either revolute or prismatic type.

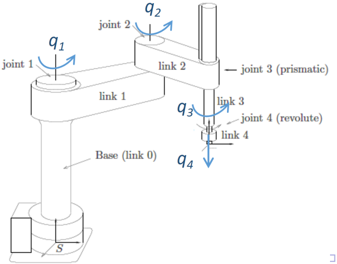
Joint space

Definition

To represent the configuration of the manipulator and the position of all its links, the most obvious solution is to use the *joint variables* or *joint coordinates*, which are the degrees of freedom of the kinematical chain. Therefore they are sometimes known as *configuration variables* of the robot :

$$q = [q_1, \ldots, q_N]^T$$

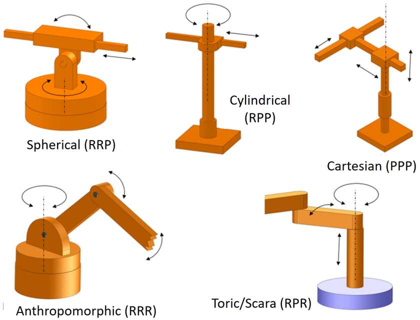
For real robots, these degrees of freedom are motorized. The Joint Space is defined by the hyper-space of joint variables, denoted $\mathbb{R}^q$.



General structure of a robot manipulator

Morphology of a simple open-tree manipulator :

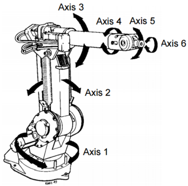
- Arm : usually the 3 first degrees of freedom (either of revolute or prismatic type) ;
  - mechanical role of the arm : to position the effector in space ;
  - 12 architectures really different and non redundant (for instance, elimination of cases made of 3 parallel prismatic joints or 3 parallel revolute joints) ;
  - most useful combination of joints for building a 3-dof arm (5 common configurations).

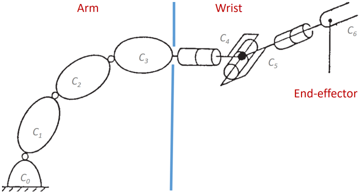


General structure of a robot manipulator

Morphology of a simple open-tree manipulator :

- Wrist : last degrees of freedom of the manipulator, caracterized by links with smaller dimensions and mass
  - in charge of giving the prescribed orientation to the tool ;
  - structure RRR made of three revolute joints with intersecting axes and orthogonal two by two ;
  - structure giving a spherical behavior, generally used as a wrist ;
  - manipulator obtained in associating a *wrist* with a 3-dof *arm* :
    - most common 6-dof structure ;
    - kinematic decoupling between effector orientation and position : position of the intersecting axes between the three last joint axes (wrist center) only depending on the configuration of the links 1, 2 and 3.





Task space

Definition

The task space (or *operational*/*cartesian* space), noted as $\mathbb{R}^X$, is a space in which the configuration of the end effector of a robot is represented, not the configuration of the entire robot. We consider as many operational spaces as there are end-effectors.

- Full specification of the end-effector pose through $\mathbb{R}^3 \times \mathbb{SO}(3)$:
  - specification of the end-effector position through the cartesian coordinates in $\mathbb{R}^3$.
  - specification of the end-effector orientation through $\mathbb{SO}(3) \subset \mathbb{R}^{3\times 3}$, special orthogonal group (also known as the group of rotation matrices in $\mathbb{R}^{3\times 3}$).
- Dimension M of $\mathbb{R}^X$ equal to the maximum number of dof that the end-effector can have. Its operational coordinates are noted :

$$X = [x_1, \ldots, x_M]^T$$

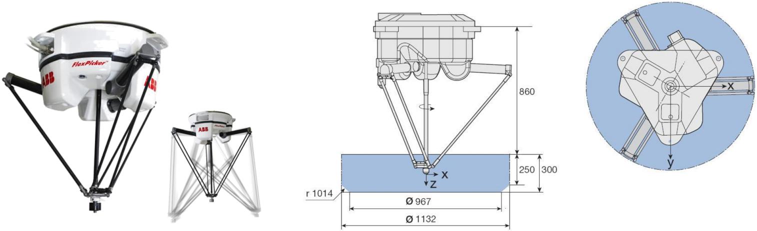
  - in 3-D space, $M = 6$ (3 are required to place a point of the body at any point in 3-D space and 3 to orientate this body in any way)
  - generally, $M \leq 6$ et $M \leq N$

Workspace

Definition

The workspace is a specification of the configurations that the end-effector of the robot can reach.

- The definition of the workspace is primarily driven by the robot's structure, independently of the task ;
- The lack of a configuration solution q to reach X means that the manipulator cannot attain the desired pose (potentially, because it may lie outside of the manipulator's workspace).

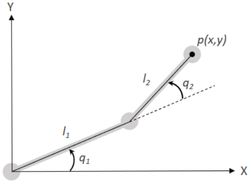


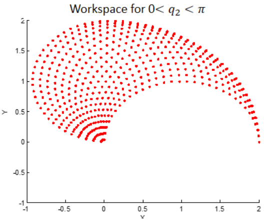
Workspace of the *IRB FlexPicker* robot from ABB^{TM} ($M = 4$)

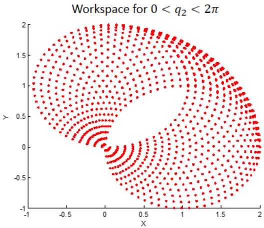
Workspace

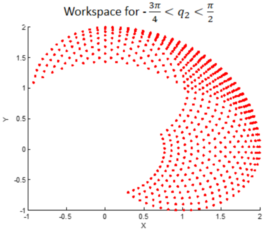
Example n°4 : planar RR mechanism with two links

- Maximum elongation : $L = l_1 + l_2$
- Joints limits : $q1_{\min} \leq q1 \leq q1_{\max}$ and $q2_{\min} \leq q2 \leq q2_{\max}$
- Search for the covered workspace according to the variation range of q_2







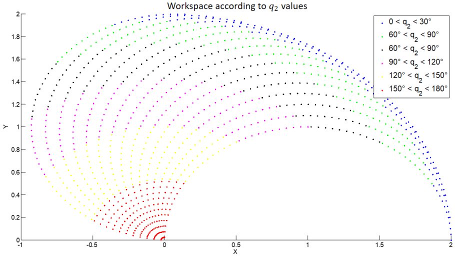


Workspace

Example n°4 : planar RR mechanism with two links

- Calculating the workspace area covered by $p(x, y)$;
- Searching for the optimal ratio of lengths $\lambda = \frac{l_1}{l_2}$;
- Searching for the optimal joint angle q_2 maximizing the variation rate of the workspace.

▸ Example n°4



$q_{2_{\min}} \leq q_2 \leq q_{2_{\max}}$	Area
$0^\circ < q_2 < 30^\circ$	0,21
$30^\circ < q_2 < 60^\circ$	0,58
$60^\circ < q_2 < 90^\circ$	0,79
$90^\circ < q_2 < 120^\circ$	0,79
$120^\circ < q_2 < 150^\circ$	0,58
$150^\circ < q_2 < 180^\circ$	0,21

Kinematic redundancy

Definition

A robot is kinematically redundant when the dof number of its task space is lower than the number of dof of its joint space :

$$M < N$$

Such robot can have an infinite number of configurations to locate the end effector at a desired location.

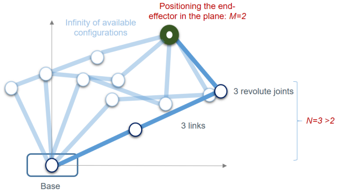
Combination of joints in a simple open-tree structure implying redundancy :

- More than 6 joints ;
- More than 3 revolute joints with intersecting axes ;
- More than 3 revolute joints with parallel axes ;
- More than 3 prismatic joints ;
- Configuration with 2 prismatic joints along parallel axes ;
- Configuration with 2 revolute joints around coincident axes.

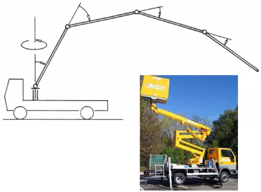
Kinematic redundancy

Problem of inverse kinematics for the realization of the operational task :

- Find $q(t) \in \mathbb{R}^N$ such that $X(t) = f\,(q(t)) \in \mathbb{R}^M$ (for every t)
- Infinity of joint configurations involving the same end-effector pose ;
- Internal motions in the joint space not observable in the operating space :
  - Choice of the $(N - M)$ joint motions coming from optmization process ;
  - Reconfiguration of the arm in its joint space without affecting the operational task.



Redundant 3R planar robot for positioning



Lift truck

Kinematic redundancy

Interests of having additional dof :

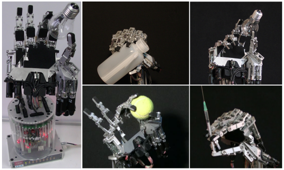
- collision avoidance ;
- increasing the volume of the accessible domain ;
- optimizing the manipulator motions in relation to a cost function ;
- avoiding singular configurations.

Drawbacks of having additional dof :

- Increasing design complexity : mechanics (many bodies and transmissions), instrumentation (more actuators and sensors), cost, . . .
- Complexifying algorithms for computing the inverse kinematics and motion control.

Technical features	
Size	154 cm
Width	62 cm
Weight	58 kg
Motion speed	0 - 2 km/h
Dof	30
Power supply	Batteries NiMH 48V 18Ah
Design	National Institute of Advanced Industrial Science and Technology, AIST

Japanese HRP-2 humanoïd robot, AIST



CEA dexterous robot hand [20]

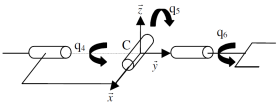
Singularity

Definition

For all robots, whether they are redundant or not, it is possible that in certain configurations, called *singularities*, the number of dof of the end-effector may be less than the size of the operational space in which it usually operates : the robot is then *locally redundant*.

For example, this case arises when :

- two axes of prismatic joints are becoming parallel to each other ;
- two axes of revolute joints are becoming coincident.



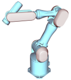
Loss of mobility in this particular wrist configuration :

- Mobility index of the Poly-articulated chain : $N_{\text{mob}} = 3$;
- Number of dof of the end-effector : 2.

Remark :

- Analytical method for determining the number of dof of the operational space of a mechanism, as well as its singularity configurations, expanded upon for the rest of the course.

A few cases of 7-dof redundant *lightweight* (collaborative) manipulators [8]

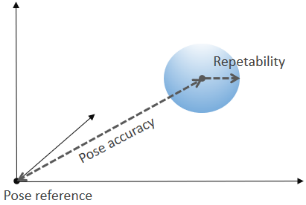
Robot	Specifications*	Actuation and instrumentation technologies
 KUKA-DLR LWR	$m_r = 16$ kg $m_c = 7$ kg $l^p = 0,936$ m	• Harmonic drive® gearboxes • Integrated strain gauges, motor and joint position sensors
 Barrett WAM	$m_r^{\text{total}} = 27$ kg $m_c = 3$ kg $l^p = 1$ m	• Gearless cable transmissions • Motor (and joint) position sensors
 Mitsubishi PA10-7CE	$m_r = 38$ kg $m_c = 10$ kg $l^a = 0,93$ m	• Servomotors, Harmonic drive® gearboxes • Motor position sensors
 ASSIST manipulator (CEA-LIST)	$m_r = 9,3$ kg $m_c = 3$ kg $l^a = 0,8$ m	• Cable drives, motor gearboxes • Motor position sensors

(*) : m_r robot mass ; m_c payload ; l characteristic dimension (distance between joints l^a or reachable l^p)

Useful features in robotics

Specification of the characteristics according to the ISO 9946 standard :

- Payload maximum load that can be handled by the robot while respecting nominal performances ;
- Workspace all the situations in space that the robot's end-effector can reach (as defined by its limits, essentially imposed by the number of degrees of freedom, the length of the links and joint limits) ;
- Maximum speeds and accelerations conditioning the cycle times ;
- Repeatability dispersion of the situations reached when successively controlling the same situation (maximum radius of the sphere centred at the barycentre of the cloud of reached points and containing all the points) ;
- Resolution smallest change in robot configuration that can be observed and controlled by the control system ;
- Pose accuracy difference between a requested pose (a situation) and the average of the reached poses.



For a given task, other characteristics can be taken into account :

- *technical* (energy, control, . . .) ;
- *economic* (cost, maintenance, . . .).

Example of technical data provided by a robot manufacturer

Data sheet (case of *IRB 120* robot from ABB® company)

Movement

Axis movement	Working range	Velocity IRB 120
Axis 1 rotation	+165° to -165°	250°/s
Axis 2 arm	+110° to -110°	250°/s
Axis 3 arm	+70° to -110°	250°/s
Axis 4 wrist	+160° to -160°	320°/s
Axis 5 bend	+120° to -120°	320°/s
Axis 6 turn	Default: +400° to -400° Max. rev: +242 to -242	420°/s

Performance (according to ISO 9283)

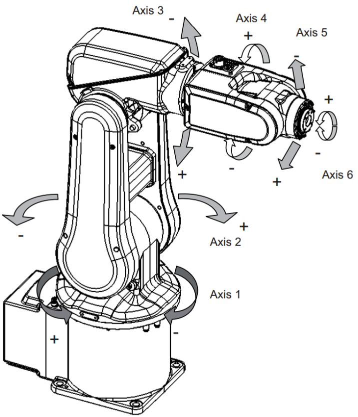
	IRB 120
1 kg picking cycle	
25 x 300 x 25 mm	0.58 s
25 x 300 x 25 with 180° axis 6 reorientation	0.92 s
Acceleration time 0-1 m/s	0.07 s
Position repeatability	0.01 mm

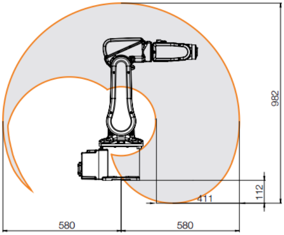
Technical information

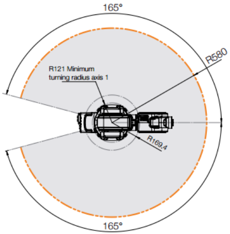
Electrical Connections	
Supply voltage	200-600 V, 50/60 Hz
Rated power transformer rating	3.0 kVA
Power consumption	0.24 kW
Physical	
Robot base	180 x 180 mm
Robot height	700 mm
Robot weight	25 kg

Example of technical data provided by a robot manufacturer

Workspace (case of *IRB 120* robot from ABB® company)

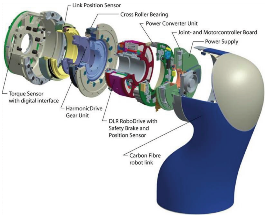






Instrumentation in robotics and perirobotics

- Actuators
  - electric (Direct current motor, brushless,. . .) ;
  - others (pneumatic, hydraulic,. . .).
- Sensors
  - Proprioceptive (incremental encoders or joint tachometers) ;
  - Exteroceptive (force, vision, . . .).
- Control unit
  - motor drive (power amplifiers, torque control – potentially speed and position –, low-level safety devices) ;
  - controller (high-level control with real-time constraints, task scheduling, artificial intelligence) ;
  - Human Machine Interface (console, joystick, . . .).

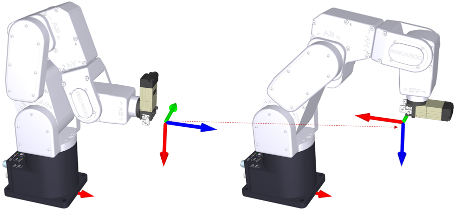


View of a DLR-KUKA robot joint [9]

Usefulness of frames in robotics

Fundamental notion of frames transformation in robotics for :

1. describing the motion of the robot bodies over the time ;



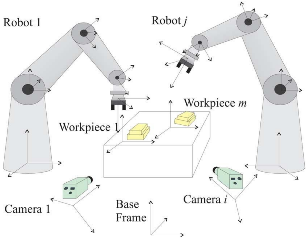
Initial and final configurations of one rigid body of a manipulator.

- Tracking the motion of the rigid body of a manipulator is done by tracking a particle belonging to the body and the rotation of the body around this point.

Usefulness of frames in robotics

Fundamental notion of frames transformation in robotics for :

2. specifying the situations that the reference frame associated with the robot's terminal device must take to perform a given task ;
3. integrating perception information from sensors, each with its own reference system, into the control system.



Sketch of a multi-arm robotic cell and frames assignment.

- Each element of the workcell is associated with one or more Cartesian frames to define its position and orientation relative to the other elements.

Usefulness of frames in robotics

Fundamental notion of frames transformation in robotics for :

4. expressing the situations of the different robot bodies in relation to each other.

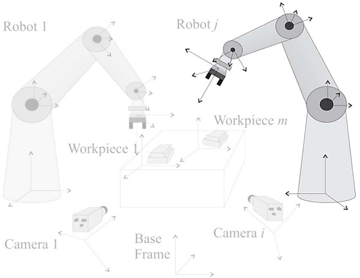
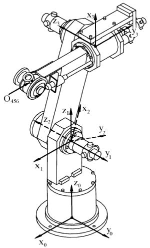


Illustration of reference frames associated with the rigid bodies of robot j.

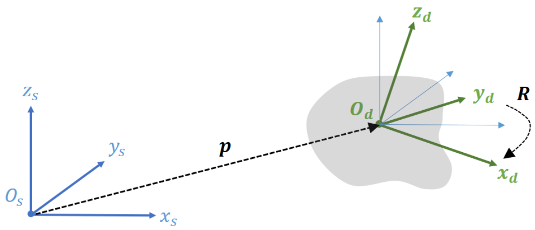


Another example of frames associated with the first three links of a 6 dof-robot.

- The robot configuration will be defined by the relative position and orientation of the reference frames associated with each of the rigid bodies.

Rigid body motion

- A systematic way to describe a rigid body motion relies on attaching a frame to the body and to parametrize the frame configuration in position and orientation during the time.
- If a direct orthonormal base $\mathscr{B}_s = (x_s, y_s, z_s)$ is attached to the body at point O_s, the configuration of the body after the action of a rigid body transformation g is given by the right-handed base $\mathscr{B}_d = (x_d, y_d, z_d)$ attached to O_d.



Initial and final configurations of a solid following a rigid body transformation.

Rigid body motion

Definition

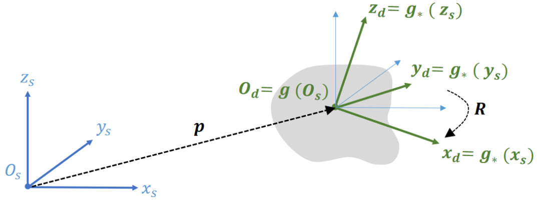
A mapping $g : \mathbb{R}^3 \mapsto \mathbb{R}^3$ is a *rigid body transformation* if it satisfies the following properties :

1. Length is preserved : $\|g(a) - g(b)\| = \|a - b\|$ for all points $a, b \in \mathbb{R}^3$
2. Cross-product is preserved : $g_*(v \times w) = g_*(v) \times g_*(w)$ for all vectors $v, w \in \mathbb{R}^3$ (where $g_*(v) = g(a) - g(b)$ if v is the vector that links a to b).

- A rigid body transformation is the resulting motion that brings one solid from an initial configuration to a final one, while preserving distance and orientation.
  - Assumption of rigid body : for all pairs of particles $a, b \in \mathbb{R}^3$, $\|a(t) - b(t)\| = \|a(0) - b(0)\| = C^{ste}$ whatever the body motion and the forces acting on it are.
  - A direct orthonormal frame will remain direct and orthonormal after the application of a *rigid motion*.

Rigid body motion

- The action of the previous rigid body transformation g is given by the frame $\mathscr{R}_d = (O_d, x_d, y_d, z_d)$ formed by the vectors $g_*(x_s), g_*(y_s), g_*(z_s)$ attached to the point $g(O_s)$.
- Let define :
  1. the vector position p that links the origins of $\mathscr{R}_s$ and $\mathscr{R}_d$;
  2. the rotation matrix R that represents the body-frame orientation from $\mathscr{R}_s$ to $\mathscr{R}_d$.
- Together, the pair (p, R) can designate :
  - a description of the position and orientation of $\mathscr{R}_d$ relative to $\mathscr{R}_s$;
  - a description of the rigid body motion from $\mathscr{R}_s$ to $\mathscr{R}_d$.



The frame $\mathscr{R}_d$ in $\mathscr{R}_s$ is given by (p, R).

Rotation matrices

Rotational motion in $\mathbb{R}^3$

The *Special Orthogonal group* $SO(3)$ in the Euclidian space $\mathbb{R}^3$, also known as the group of rotation matrices, is the set of all 3×3 real matrices R that satisfy (i) $det(R) = +1$ and (ii) $RR^t = R^t R = Id$:

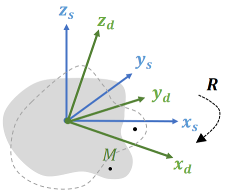
$$SO(3) = \{R \in \mathbb{R}^{3\times 3} : RR^t = R^t R = Id, det(R) = +1\}.$$

Some properties inherited from $SO(3)$

- Recall that the three columns of R_{sd} correspond to the coordinates of the frame unit axes $\mathscr{B}_d$ expressed in $\mathscr{B}_s$, also known as the direction cosines :

$$R_{sd} = \begin{bmatrix} x_d & y_d & z_d \end{bmatrix} = \begin{bmatrix} x_d \cdot x_s & y_d \cdot x_s & z_d \cdot x_s \\ x_d \cdot y_s & y_d \cdot y_s & z_d \cdot y_s \\ x_d \cdot z_s & y_d \cdot z_s & z_d \cdot z_s \end{bmatrix}$$

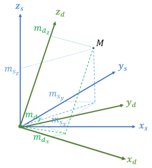
- 1^{st} geometrical meaning of the rotation matrix : $R_{sd} \in SO(3)$ is the matrix that describes the finite rotation allowing to move from $\mathscr{R}_s$ to the frame $\mathscr{R}_d$.



Rotation matrices

- 2^{nd} geometrical meaning of the rotation matrix : change of vector coordinates
  - Representation of a vector :
    - point M in space can be represented either as $m_d = \begin{bmatrix} m_{x_d} & m_{y_d} & m_{z_d} \end{bmatrix}^t$ w.r.t. frame $\mathscr{R}_d$ or $m_s = \begin{bmatrix} m_{x_s} & m_{y_s} & m_{z_s} \end{bmatrix}^t$ w.r.t. frame $\mathscr{R}_s$.
    - since m_d and m_s are representations of the same point M, it is

$$m_s = m_{x_d} x_d + m_{y_d} y_d + m_{z_d} z_d$$
$$= \underbrace{\begin{bmatrix} x_d & y_d & z_d \end{bmatrix}}_{R_{sd}} m_d$$



Representation of point M in two different frames.

  - Transformation matrix of the vector coordinates through frame change :

$$m_s = R_{sd} m_d \text{ and } m_d = R_{sd}^t m_s$$

- To sum up, the matrix R represents not only the orientation of a frame with respect to another frame, but it also describes the transformation of a vector from a frame to another frame with the same origin.

Properties of rotation matrices

Number of independent parameters

Let the rotation matrix

$$R = \begin{bmatrix} r_1 & r_2 & r_3 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

Only 3 parameters among the 9 components of R can be chosen independently :

- Orthogonality and unit norm conditions between columns (due to orthogonality property of R) imposing 6 constraints between r_i :

$$R^t R = Id \Leftrightarrow r_i^t r_j = \delta_{ij}$$

(inner product $r_i^t r_j$ null for $i \neq j$ and $r_i^t r_i = 1$)

- Only 3 independent parameters for describing entirely R :

$$R = R(\alpha, \beta, \gamma)$$

There exists different possible parametrizations for this transformation (*Euler* angles, Roll/Pitch/Yaw - RPY -, exponential coordinates, quaternions, etc.).

Properties of rotation matrices

Composition of rotation matrices :

- Let consider the vectors x_i (with $i = 1, 2, 3$) describing the positions of a generic 3D point that is expressed in the frames $\mathscr{R}_{s_1}$. Consider two successive rotations of the frame $\mathscr{R}_{s_0} = (O_{s_0}, x_{s_0}, y_{s_0}, z_{s_0})$:
  - R_{01} represents the orientation of $\mathscr{R}_{s_1} = (O_{s_1}, x_{s_1}, y_{s_1}, z_{s_1})$ in $\mathscr{R}_{s_0}$:
  $$x_0 = R_{01} x_1$$
  - R_2 represents the orientation of $\mathscr{R}_{s_2} = (O_{s_2}, x_{s_2}, y_{s_2}, z_{s_2})$ in $\mathscr{R}_{s_1}$:
  $$x_1 = R_{12} x_2$$
- The transformation matrix R from $\mathscr{R}_{s_0}$ to $\mathscr{R}_{s_2}$ is given by :
$$x_0 = R_{01} x_1 = x_0 = R_{01}(R_{12} x_2) = (R_{01} R_{12}) x_2$$
- Rule for composition of rotations :
$$x_0 = R x_2 \qquad \text{where} \qquad R = R_{01} R_{12}$$

Particular case of consecutive rotations around the same axis u

$$R_{u,q_1} R_{u,q_2} = R_{u,(q_1+q_2)}$$

Inverse transformation

- $R_{u,q}^{-1} = R_{u,q}^t$ (inherited from the group property of $SO(3)$)
- $R_{u,q}^{-1} = R_{u,-q} = R_{-u,q}$

Properties of rotation matrices

Non-commutative composition

$$R_{u,q_1} R_{v,q_2} \neq R_{v,q_2} R_{u,q_1}$$

Example

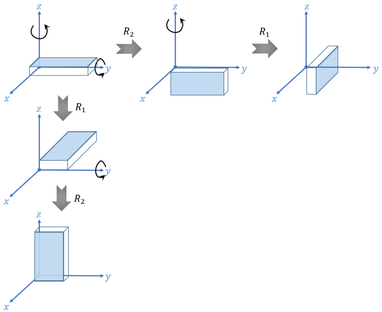
- Let consider a rigid-body following two successive rotations of angle $+\frac{\pi}{2}$, such that :

$$R_1 = R_{z,\frac{\pi}{2}}$$
$$R_2 = R_{y,\frac{\pi}{2}}$$

- The matrix product being non-commutative :

$$R_1 R_2 \neq R_2 R_1,$$

  inverting the order of rotations leads to different final configurations.

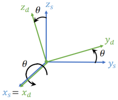


Angle-axis representation for rotation

Elementary rotations about coordinate frame axes

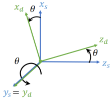
- Rotation matrices :
  - Rotation of angle θ around x_s axis :

$$R_{x,\theta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\theta) & -sin(\theta) \\ 0 & sin(\theta) & cos(\theta) \end{bmatrix}$$



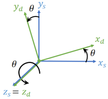
  - Rotation of angle θ around y_s axis :

$$R_{y,\theta} = \begin{bmatrix} cos(\theta) & 0 & sin(\theta) \\ 0 & 1 & 0 \\ -sin(\theta) & 0 & cos(\theta) \end{bmatrix}$$



  - Rotation of angle θ around z_s axis :

$$R_{z,\theta} = \begin{bmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



- These matrices will be useful to describe rotations about an arbitrary axis in space.
- As a geometrical meaning, the matrix R describes the rotation about an axis in space needed to align the axes of the reference frame with the corresponding axes of body frame.

Angle-axis representation for rotation

Rotation about an arbitrary axis : general case

If the *unit* vector ω which defines the axis of rotation is chosen arbitrary w.r.t. the direct orthonormal base $\mathscr{B}_s$, the computation is less direct.

Rodrigues formula

This formula describes any finite rotation as a rotation $\theta \in \mathbb{R}$ around an arbitrary axis defined by the *unit* vector $w \in \mathbb{R}^3$ ($\|w\| = 1$). The image v of any vector u by the rotation operation $R_{w,\theta}$ is unique :

$$v = cos(\theta)u + (1 - cos(\theta))\,(u \cdot w)\,w + sin(\theta)\,(w \times u)$$

The same result can be presented in the equivalent matrix form as follows :

$$R_{w,\theta} = cos(\theta)I_{3\times 3} + (1 - cos(\theta))\,ww^t + sin(\theta)\widehat{w} \;\; or \;\; R_{w,\theta} = I_{3\times 3} + (1 - cos(\theta))\,\widehat{w}^2 + sin(\theta)\widehat{w}$$

where $\widehat{w}$ notation means the skew-symmetric matrix associated to the cross product of vectors w and u as follows $w \times u = \widehat{w}u$:

$$\widehat{w} = \begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{bmatrix}$$

The expanded expression for $R_{w,\theta}$ is :

$$R_{w,\theta} = \begin{bmatrix} w_x^2\,(1 - cos(\theta)) + cos(\theta) & w_x w_y\,(1 - cos(\theta)) - w_z sin(\theta) & w_x w_z\,(1 - cos(\theta)) + w_y sin(\theta) \\ w_x w_y\,(1 - cos(\theta)) + w_z sin(\theta) & w_y^2\,(1 - cos(\theta)) + cos(\theta) & w_y w_z\,(1 - cos(\theta)) - w_x sin(\theta) \\ w_x w_z\,(1 - cos(\theta)) - w_y sin(\theta) & w_y w_z\,(1 - cos(\theta)) + w_x sin(\theta) & w_z^2\,(1 - cos(\theta)) + cos(\theta) \end{bmatrix}$$

Demonstration

Angle-axis representation for rotation

Retrieving the axis and angle parameters from a given rotation matrix

Inverse relationship

Let consider the rotation matrix $R \in SO(3)$ given by :

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

The identification of the quantities w and θ $(0 \leq \theta \leq \pi)$ such that $R = R_{w,\theta}$ is given by :

$$\theta = atan2\left(\frac{1}{2}\sqrt{(r_{32} - r_{23})^2 + (r_{13} - r_{31})^2 + (r_{21} - r_{12})^2}, \frac{1}{2}(r_{11} + r_{22} + r_{33} - 1)\right)$$

and

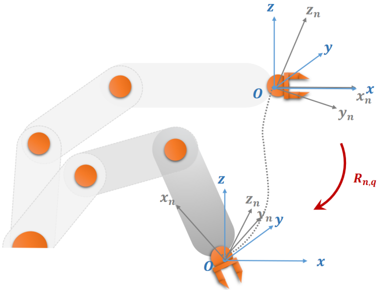
$$w = \begin{bmatrix} \frac{r_{32} - r_{23}}{2S\theta} & \frac{r_{13} - r_{31}}{2S\theta} & \frac{r_{21} - r_{12}}{2S\theta} \end{bmatrix}^t \quad \text{if } S\theta \neq 0$$

▸ Demonstration

Angle-axis representation for rotation

Retrieving the axis and angle parameters from a given rotation matrix

Example : Let assume that the frame $\mathscr{R}_n$ associated to the end-effector of a robot is in a situation w.r.t. a reference frame, that is defined by the rotation $R_{x,-45^\circ}$. Find the unit axis w and the angle θ of the rotation operation that brings the frame $\mathscr{R}_n$ in the following desired configuration : $R_{y,45^\circ} R_{z,90^\circ}$.



Example

Exponential coordinate representation

Exponential coordinates for rotation

Let $\theta \in \mathbb{R}$ be the angle of rotation in radians and let $w \in \mathbb{R}^3$ be a *unit* vector that specifies the direction of rotation. We associate to the vector w the matrix $\widehat{w} \in so\,(3)$, where :

$$so\,(3) = \left\{ S \in \mathbb{R}^{3\times 3} : S^t = -S \right\}.$$

is the vector space over the reals of all 3×3 skew-symmetric matrices. The matrix exponential of $\widehat{w}\theta \in so\,(3)$ with $\|w\| = 1$ is orthogonal

$$e^{\widehat{w}\theta} \in SO\,(3)$$

and defines the rotation $R_{w,\theta}$:

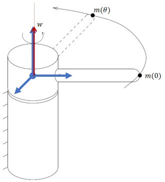
$$R_{w,\theta} = e^{\widehat{w}\theta} = I_{3\times 3} + (1 - cos(\theta))\,\widehat{w}^2 + sin(\theta)\widehat{w}$$

Demonstration

- The vector $w\theta \in \mathbb{R}^3$ serves as the parameters exponential coordinate representation of the rotation, while writing w and θ individually is the axis-angle representation of a rotation.
- Let note that both representations lead to *Rodrigues* formula.

Exponential coordinate representation

Physical meaning for rotating body :



- Considering the velocity of a point M attached to a rotating body at constant unit velocity about the axis w :

$$\dot{m}(t) = w \times m(t) = \widehat{w} m(t)$$

- Integrating the time-invariant linear differential equation

$$m(t) = e^{\widehat{w}t} m(t)$$

where $e^{\widehat{w}t}$ is the matrix exponential defined by

$$e^{\widehat{w}t} = I_{3\times 3} + \widehat{w}t + \widehat{w}^2 \frac{t^2}{2!} + \widehat{w}^3 \frac{t^3}{3!} + \dots$$

- Giving the net rotation if the body is rotating about the axis w at unit velocity for θ units of time :

$$R_{w,\theta} = e^{\widehat{w}\theta}$$

imaging that $m(0)$ rotates at a constant rate of 1 rad/s (since w has unit magnitude) from time $t = 0$ to $t = \theta$ (t and θ becoming interchangeable) as follows :

$$m(\theta) = e^{\widehat{w}\theta} m(0).$$

Euler angles representation for rotation

System of 3 independent angles to constitute a minimal representation of orientation

- Rotation from (O, x, y, z) to (O, X, Y, Z) obtained by composing elementary rotations expressed w.r.t. axes of *current frames* :
  - rotation of angle α around z :

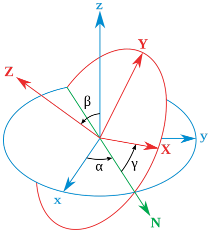
$$R_{z,\alpha} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

  - rotation of angle β around (new) N :

$$R_{N,\beta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\beta) & -\sin(\beta) \\ 0 & \sin(\beta) & \cos(\beta) \end{bmatrix}$$

  - rotation of angle γ around (new) Z :

$$R_{Z,\gamma} = \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



- The triple of *Euler* angle (α, β, γ) is used to represent the rotation R.
- The resulting rotation is given by :

$$R = R_{z,\alpha} R_{N,\beta} R_{Z,\gamma}$$

Euler angles representation for rotation

- Explicit expression of rotation matrix (in condensed writing [1])

$$R = \begin{bmatrix} C\alpha C\gamma - S\alpha C\beta S\gamma & -C\alpha S\gamma - S\alpha C\beta C\gamma & S\alpha S\beta \\ S\alpha C\gamma + C\alpha C\beta S\gamma & -S\alpha S\gamma + C\alpha C\beta C\gamma & -C\alpha S\beta \\ S\beta S\gamma & S\beta C\gamma & C\beta \end{bmatrix}$$

- Inverse relationships :
  - General case :
    1. Relationship on angle γ :

$$\gamma = arctg\left(\frac{S\gamma}{C\gamma}\right) = arctg\left(\frac{r_{31}}{r_{32}}\right) = atan2\left(r_{31}, r_{32}\right)$$

    2. Relationship on angle β :

$$\beta = atan2\left(r_{31}S\gamma + r_{32}C\gamma, r_{33}\right)$$

    3. Relationship on angle α :

$$\alpha = atan2\left(r_{21}C\gamma - r_{22}S\gamma, r_{11}C\gamma - r_{12}S\gamma\right)$$

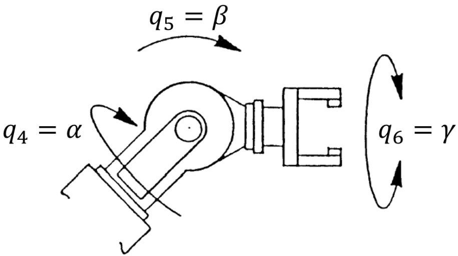
  - Existence of singular configurations for $\beta = k\pi$ $(k \in \mathbb{N})$:
    - physical case where the axes z and Z are confounded ;
    - no more possible distinctions between angles α and γ, for example if $\beta = 2k\pi$ $(k \in \mathbb{N})$:

$$R = R_{z,\alpha+\gamma} = \begin{bmatrix} C\alpha\gamma & -S\alpha\gamma & 0 \\ S\alpha\gamma & C\alpha\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

1. In robotics, we often use the notations C and S to designate the quantities $cos(\cdot)$ and $sin(\cdot)$

Euler angles representation for rotation

- Usefulness of *Euler* angles for the wrist configuration : correspondence with the joint variables of the mechanical structure



- Guaranteeing the uniqueness of the orientation representation by imposing :

$$\begin{cases} -\pi \leq \alpha < \pi \\ -\frac{\pi}{2} \leq \beta < \frac{\pi}{2} \\ -\pi \leq \gamma < \pi \end{cases}$$

$atan2(u, v)$ function

Software function returning the value of the arc tangent of $\frac{u}{v}$ and using the signs of u and v for determining the solution quadrant.

- Guaranteeing the uniqueness in providing a value in the interval $]-\pi, \pi]$:

$$-\pi < atan2\,(u, v) \leq \pi$$

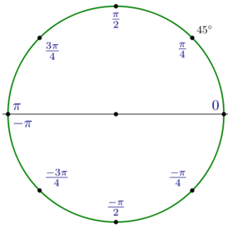
- Definition :

$$atan2\,(u, v) = \begin{cases} tan^{-1}\left(\frac{u}{v}\right) & if \quad v > 0 \\ tan^{-1}\left(\frac{u}{v}\right) + \pi sign\,(u) & if \quad v < 0 \\ \frac{\pi}{2} sign\,(u) & if \quad v = 0 \end{cases}$$

- Function $atan2(u, v)$ available in FORTRAN or MATLAB® for example

Angle in *rad* made between the positive part of the x axis and the point coordinates (v, u) in this plane :

$$atan2\,(1, 0) = \frac{\pi}{2}$$
$$atan2\,(0, -1) = \pi$$
$$atan2\,(-1, -1) = -\frac{3\pi}{4}$$
$$atan2\,(0, 1) = 0$$

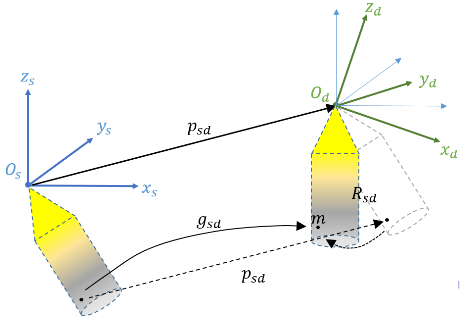


Rigid body motion involving both translation and rotation

Configuration of a rigid body in $\mathbb{R}^3$ [21]

Let $p_{sd} \in \mathbb{R}^3$ be the position vector of the origin of frame $\mathscr{R}_s$ to the origin of frame $\mathscr{R}_d$ ($p_{sd} = O_s O_d$) and $R_{sd} \in SO(3)$ the orientation of frame $\mathscr{R}_d$ relative to frame $\mathscr{R}_s$. A configuration of the system consists of the pair (p_{sd}, R_{sd}), and the configuration space of the system is the product space of $\mathbb{R}^3$ with $SO(3)$, which shall be denoted as $SE(3)$ (for *Special Euclidian group*) :

$$SE(3) = \{(p, R) : p \in \mathbb{R}^3, R \in SO(3)\} = \mathbb{R}^3 \times SO(3).$$



Rigid body motion involving both translation and rotation

Analogous to the rotational case, an element $(p, R) \in SE(3)$ serves as both a specification of the configuration of a rigid body in space and a transformation taking the coordinates of a point from one frame to another.

- Let m_s et $m_d \in \mathbb{R}^3$ be the coordinates of M relative to frame $\mathscr{R}_s$ and $\mathscr{R}_d$ respectively. Given m_d, the coordinates transformation is as foolows :

$$m_s = p_{sd} + R_{sd} m_d$$

  où

  - $R_{sd} \in SO(3)$ is the rotation matrix given the orientation of frame $\mathscr{R}_d$ relative to frame $\mathscr{R}_s$:

$$R_{sd} = \begin{bmatrix} x_d \cdot x_s & y_d \cdot x_s & z_d \cdot x_s \\ x_d \cdot y_s & y_d \cdot y_s & z_d \cdot y_s \\ x_d \cdot z_s & y_d \cdot z_s & z_d \cdot z_s \end{bmatrix}$$

    The components of R_{sd} are the direction cosines that represent the coordinates of the three vectors of $\mathscr{B}_d$ expressed in $\mathscr{B}_s$.
  - $p_{sd} \in \mathbb{R}^3$ est le vecteur de translation selon :

$$p_{sd} = \begin{bmatrix} O_s O_d \cdot x_s \\ O_s O_d \cdot y_s \\ O_s O_d \cdot z_s \end{bmatrix}$$

- By an abuse of notation, we write $g(m)$ the action of a rigid transformation on a point :

$$\boxed{g(m) = p + Rm},$$

  so that $m_s = g_{sd}(m_d)$.

Notion of homogeneous representation [23]

The transformation $m_s = g_{sd}(m_d)$ is an affine transformation. To represent the rigid transformation under a linear form, we introduce the homogeneous coordinates in $\mathbb{R}^4$ for point M et vector v (seen as a difference between two points).

- Let m be a point with cartesian coordinates m_x, m_y, m_z. We append 1 to the coordinates of a point to yield a vector in $\mathbb{R}^4$:

$$\overline{m} = \begin{bmatrix} m \\ 1 \end{bmatrix} = \begin{bmatrix} m_x \\ m_y \\ m_z \\ 1 \end{bmatrix}$$

  Thus, the origin has the form $\overline{O} = [0, 0, 0, 1]^t$.

- Let v_x, v_y, v_z be the cartesian coordinates of vector v. In homogeneous coordinates, it results in :

$$\overline{v} = \begin{bmatrix} v \\ 0 \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ v_z \\ 0 \end{bmatrix}$$

Remarks : The presence of 0 or 1 (respectively at the 4^{th} components of vectors or points) alows distinguishing between vectors and points, and enforce a few rules of syntax :

1. Sums and differences of vectors are vectors ;
2. The sum of a vector and a point is a point ;
3. The difference between two points is a vector ;
4. The sum of two points is meaningless !

Homogeneous transformation matrices

Definition

The 4×4 real matrix $\overline{g}_{sd}$ of the form :

$$\overline{g}_{sd} = \begin{bmatrix} R_{sd} & p_{sd} \\ 0_{1\times 3} & 1 \end{bmatrix}$$

is called the *homogeneous matrix* of transformation g_{sd} and defines the group of *rigid-body motions* $g = (r, R) \in SE(3)$ where

- R_{sd} is the 3×3 rotation matrix that can can be interpreted as representing the frame $\mathscr{R}_d = (O_d x_d y_d z_d)$ w.r.t. frame $\mathscr{R}_s = (O_s x_s y_s z_s)$ (or can be seen as representing the transformation from frame $\mathscr{R}_s$ to frame $\mathscr{R}_d$) ;
- p_{sd} is the column vector giving the translation from O_s to O_d.

Advantages of transformations in homogeneous coordinates :

- Unification of translations and rotations of rigid bodies in one unique matrix multiplication ;
  - Transformation of usual spatial coordinates : one vector addition and one matrix multiplication

$$m_s = p_{sd} + R_{sd} m_d$$

  - Transformation in homogeneous coordinates : one simple matrix multiplication

$$\overline{m}_s = \overline{g}_{sd}\overline{m}_d \qquad \text{soit} \qquad \begin{bmatrix} m_s \\ 1 \end{bmatrix} = \begin{bmatrix} R_{sd} & p_{sd} \\ 0_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} m_d \\ 1 \end{bmatrix}$$

where $\overline{g}_{sd}$ is the homogeneous transformation matrix associated to (p, R).

Properties in $SE(3)$

Elements of $SE(3)$ represent rigid motions

Any $g \in SE(3)$ is a rigid body transformation :

1. g preserves distance between points : $\|g(a) - g(b)\| = \|a - b\|$ for all points $a, b \in \mathbb{R}^3$
2. g preserves orientation betwen vectors : $g_*(v \times w) = g_*(v) \times g_*(w)$ for all vectors $v, w \in \mathbb{R}^3$

Demonstration

Properties in $SE(3)$

Elements of $SE(3)$ represent rigid motions

Any $g \in SE(3)$ is a rigid body transformation :

1. g preserves distance between points : $\|g(a) - g(b)\| = \|a - b\|$ for all points $a, b \in \mathbb{R}^3$
2. g preserves orientation betwen vectors : $g_*(v \times w) = g_*(v) \times g_*(w)$ for all vectors $v, w \in \mathbb{R}^3$

▸ Demonstration

Properties inherited from the algebraic group structure of $SE(3)$

- Rules for components : let $g_{bc} \in SE(3)$ (configuration of $\mathscr{R}_c$ relative to $\mathscr{R}_b$) and $g_{ab} \in SE(3)$ (configuration of $\mathscr{R}_b$ relative to $\mathscr{R}_a$). The configuration of $\mathscr{R}_c$ relative to $\mathscr{R}_a$ is :

$$\overline{g}_{ac} = \overline{g}_{ab}\overline{g}_{bc} = \begin{bmatrix} R_{ab}R_{bc} & R_{ab}p_{bc} + p_{ab} \\ 0_{1\times 3} & 1 \end{bmatrix} \in SE(3)$$

- Inverse transformation : $\overline{g}^{-1} = \left(-R^t p, R^t\right)$

$$\overline{g}_{ab}^{-1} = \overline{g}_{ba} = \begin{bmatrix} R_{ab}^t & -R_{ab}^t p_{ab} \\ 0_{1\times 3} & 1 \end{bmatrix} \in SE(3)$$

▸ Demonstration

Matrix for transformation of frames

Transformation matrix from $\mathscr{R}_i$ to $\mathscr{R}_j$:

Let any transformation, combining or not translation and rotation, from frame $\mathscr{R}_i$ to frame $\mathscr{R}_j$. Such transformation is then defined by the homogeneous transformation matrix $\overline{g}_{ij}$ of size 4×4, defined by :

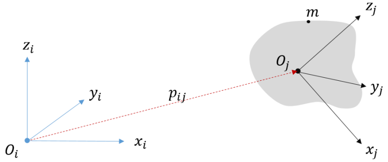
$$\overline{g}_{ij} = \left[\overline{x}_{ij}\overline{y}_{ij}\overline{z}_{ij}\overline{p}_{ij}\right] = \begin{bmatrix} R_{ij} & p_{ij} \\ 0_{1\times 3} & 1 \end{bmatrix} = \begin{bmatrix} x_x & y_x & z_x & p_x \\ x_y & y_y & z_y & p_y \\ x_z & y_z & z_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where

- $\overline{x}_{ij}, \overline{y}_{ij}, \overline{z}_{ij} \in \mathbb{R}^3$ respectively designate the unit vectors along the axes x_j, y_j, z_j of frame $\mathscr{R}_j$ expressed in frame $\mathscr{R}_i$,
- $\overline{p}_{ij}$ designates the origin of frame $\mathscr{R}_j$ expressed in $\mathscr{R}_i$.

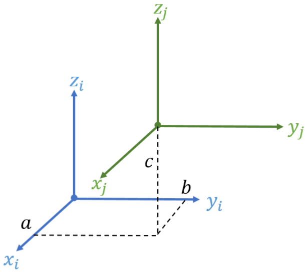
Remarks :

- We also say that the homogeneous matrix $\overline{g}_{ij}$ defines the frame $\mathscr{R}_j$ in frame $\mathscr{R}_i$.
- A parametrization of a spatial transformation $g_{ij} \in SE(3)$ is based on only 6 independent parameters (3 for R_{ij} and 3 for p_{ij}) and only 3 for a planar transformation.
- In certain book [13, 23], the homoegenous matrix $\overline{g}_{ij}$ is refered as T_{ij}, or ${}^iT_j$.



Computing $\overline{g}_{ij}$ for frame changes

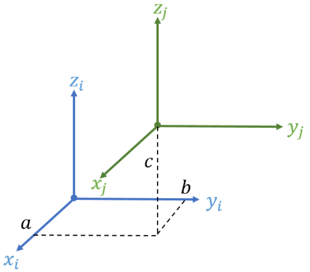
Pure translation



$\overline{g}_{ij} = Trans\,(a, b, c)$: example

Computing $\overline{g}_{ij}$ for frame changes

Pure translation



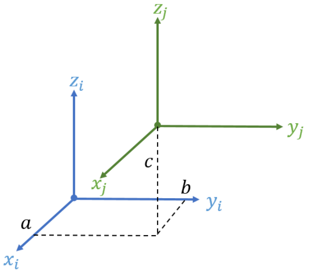
$\overline{g}_{ij} = Trans\,(a,\, b,\, c)$: example

Properties :

1. Decomposition of $Trans\,(a, b, c)$:
   $Trans\,(a, b, c) = Trans\,(x, a)\; Trans\,(y, b)\; Trans\,(z, c)$,
   the order of multiplications being arbitrary ;
2. $Trans^{-1}\,(u, d) = Trans\,(-u, d) = Trans\,(u, -d)$
   with $Trans\,(u, d)$ the translation of a value d along the axis defined by direction u.

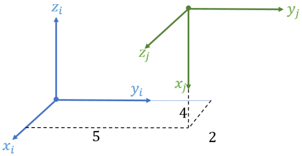
Computing $\overline{g}_{ij}$ for frame changes

Pure translation



▸ $\overline{g}_{ij} = Trans\,(a, b, c)$: example

Combining rotation and translation



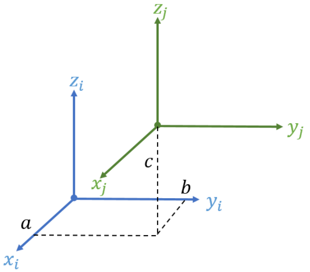
▸ $\overline{g}_{ij}$ and $\overline{g}_{ji}$: example

Properties :

1. Decomposition of $Trans\,(a, b, c)$:
   $Trans\,(a, b, c) = Trans\,(x, a)\; Trans\,(y, b)\; Trans\,(z, c)$,
   the order of multiplications being arbitrary ;
2. $Trans^{-1}\,(u, d) = Trans\,(-u, d) = Trans\,(u, -d)$
   with $Trans\,(u, d)$ the translation of a value d along the axis defined by direction u.

Computing $\overline{g}_{ij}$ for frame changes

Pure translation

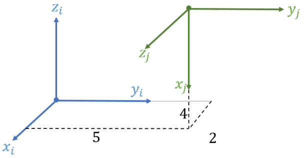


▸ $\overline{g}_{ij} = Trans\,(a, b, c)$: example

Properties :

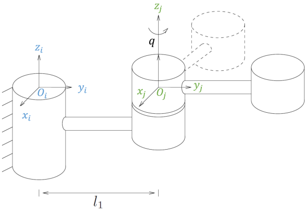
1. Decomposition of $Trans\,(a, b, c)$:
   $Trans\,(a, b, c) = Trans\,(x, a)\; Trans\,(y, b)\; Trans\,(z, c)$,
   the order of multiplications being arbitrary ;
2. $Trans^{-1}\,(u, d) = Trans\,(-u, d) = Trans\,(u, -d)$
   with $Trans\,(u, d)$ the translation of a value d along the axis defined by direction u.

Combining rotation and translation



▸ $\overline{g}_{ij}$ and $\overline{g}_{ji}$: example

Rotation around a fixed and eccentric axis



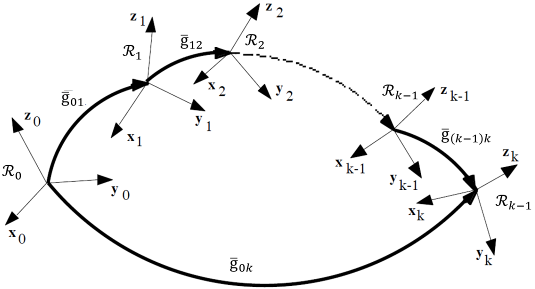
▸ $\overline{g}_{ij}$ and $\overline{g}_{ji}$: example

Properties of homogeneous transformation matrix

Successive transformations

Let a frame $\mathscr{R}_0$ undergoing k consecutive transformations. If the i^{th} transformation ($i = 1, \ldots, k$) is defined w.r.t to the current frame $\mathscr{R}_{i-1}$, then the transformation $\overline{g}_{0k}$ can be computed by the right-handed multiplication composition of the successive transformations :

$$\overline{g}_{0k}(q_1, q_2, \ldots, q_k) = \overline{g}_{01}(q_1)\,\overline{g}_{12}(q_2)\ldots\overline{g}_{(k-1)k}(q_k)$$



Properties of homogeneous transformation matrix

Non-commutative product

The product of two transformation matrices is not commutative :

$$\overline{g}_1\overline{g}_2 \neq \overline{g}_2\overline{g}_1$$

Consecutive transformation around a same axis

The vector n being unchanged through the rotation $R_{n,\theta}$, the translation to the left or to the right can be multiplied in any way :

$$R_{n,q}\,Trans\,(n,d) = Trans\,(n,d)\,R_{n,q}$$

Decomposition of the transformation matrix

A transformation matrix can be decomposed into two transformation matrices, one representing a pure translation, the other a pure rotation :

$$\overline{g} = \begin{bmatrix} R & p \\ 0_{1\times 3} & 1 \end{bmatrix} = \begin{bmatrix} I_{3\times 3} & p \\ 0_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} R & 0_{3\times 1} \\ 0_{1\times 3} & 1 \end{bmatrix}$$

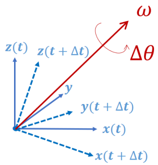
Angular velocity

Definition

From the angle-axis representation for rotation, the *angular velocity* ω of a rotating body relative to the fixed frame can be defined as the time derivative of the angular position θ of the body about some *unit* axis w (recall that $\|w\| = 1$) passing through the origin :

$$\omega = \dot{\theta} w.$$

- It is fully described through the angular velocity vector ω, which defines both the axis around which the object rotates and its speed of rotation ;
- it is normal to the plane of rotation ;
- it is oriented so that the motion is in the positive direction (usually given by the right hand rule) ;
- its norm is given by $\left\|\dot{\theta}\right\|$.



Angular velocity

Relationship between angular velocity and rotation matrix

Let a frame $\mathscr{R}_i$ be attached to a body, whose rotation with respect to the frame $\mathscr{R}_0$ is parametrized by the rotation matrix $R_{0i} \in SO(3)$. The *angular velocity* ${}^0\omega_{0i} \in \mathbb{R}^3$ (${}^0\omega_{0i} = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T$) of the rotating body relative to the fixed frame is given by :

$$ {}^0\widehat{\omega}_{0i} = \dot{R}_{0i} R_{0i}^T $$

with the 3×3 skew-symmetric matrix representation of vector ω as follows :

$$ \widehat{\omega} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} $$

Explanations :

- Considering the velocity of the point M attached to the rotating body in $\mathscr{R}_0$:

$$ {}^0\dot{m}(t) = {}^0\omega_{0i} \times {}^0m(t) = \boxed{{}^0\widehat{\omega}_{0i}}\,{}^0m(t) $$

- Comparing with the frame change coordinates relationship :

$$ {}^0m(t) = R_{0i}(t)\,{}^im $$

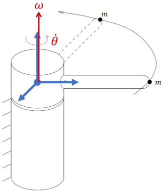
  where ${}^im$ are the coordinates of the point M fixed in the body frame $\mathscr{R}_i$ (i.e. ${}^i\dot{m} = 0$). Taking its time derivative :

$$ {}^0\dot{m}(t) = \dot{R}_{0i}(t)\,{}^im = \boxed{\dot{R}_{0i}(t)\,R_{0i}^T(t)}\,{}^0m $$

Angular velocity

Applications

1. Given $R_{\theta(t),z}$, computation of ${}^0\omega$.

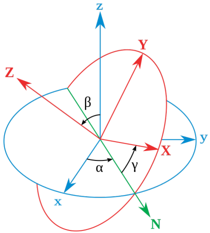


Example

2. Explicit expression for angular velocities in terms of *Euler* angles

$$\begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} = \Omega_{r_{\text{Euler}}} {}^0\omega$$

Find the 3×3 matrix $\Omega_{r_{\text{Euler}}}$



Example

Definition of a screw

Screw theory is the algebraic calculation of pairs of vectors that arise in the kinematics and dynamics of rigid bodies.

- A vector field $\mathscr{H}$ on $\mathbb{R}^3$ is a screw if there exists a point M and a vector Θ such that for all points P in $\mathbb{R}^3$:

$$H(P) = H(M) + \Theta \times MP$$

(this relationship is also called *Varignon* formula)

where
  - $H(P)$ is the moment of $\mathscr{H}$ at P ;
  - Θ is called the resultant of the *screw* $\mathscr{H}$ (independent of the considered point M) ;
  - the symbol $\times$ indicates the vector product.

Robert Ball, author of treatises on screw theory in 1876 and 1900.

Definition of a screw

- Thus, the screw at a point M is well defined by an ordered pair $(H(M), \Theta)$, where $H(M)$ and Θ are three-dimensional real vectors ;
- In the following, a *screw* is a six-dimensional vector that concatenates H and Θ, such as :
  - linear V & angular ω velocities (also called *twist*) ;
  - torque m & force f (also called *wrench*).

Screw type	**Twist**	**Wrench**
Moment $H(M)$ Resultant Θ	linear velocity $V(M)$ angular velocity ω	torque $m(M)$ forces f
Notation used for $\mathscr{H}$	$\mathscr{V}$	$\mathscr{F}$

- Duality
  - Twists and wrenches are in two dual spaces : the motion space and the force space ;
  - When $\mathscr{V}$ and $\mathscr{F}$ are the twist and wrench acting on a single rigid body, their scalar product represents the instantaneous power of the motion :

$$\mathscr{V} \cdot \mathscr{F} \overset{\text{def}}{=} V(M) \cdot f + \omega \cdot m(M)$$

  - From *Varignon* formula, this number is invariant w.r.t. the point M where vectors are taken.

Representation of twist

Twist or Kinematic screw : definition

The velocity field of a rigid body $\mathscr{C}_j$, to which is attached a frame $\mathscr{R}_j$, moving w.r.t. a rigid body $\mathscr{C}_i$ frame, to which is attached a frame $\mathscr{R}_i$, is completely defined by a *kinematic screw* or *twist*.
At one particular point (taken here as the origin O_j of frame $\mathscr{R}_j$), the velocity field is defined by :

$$ {}^i\mathscr{V}_{ij} = \begin{bmatrix} {}^iV_{ij}\left(O_j\right) \\ {}^i\omega_{ij} \end{bmatrix} \in \mathbb{R}^6. $$

- A frame is attached to each rigid body, so that notations relative to bodies are equivalent to notations relative to frames : ${}^iV_{O_j \in \mathscr{C}_j / \mathscr{C}_i} = {}^iV_{O_j \in \mathscr{R}_j / \mathscr{R}_i}$, or in a contacted notation, ${}^iV_{ij}\left(O_j\right)$
- The *twist* is an ordered pair of two three-dimensional real vectors :
  - ${}^iV_{ij}\left(O_j\right) \in \mathbb{R}^3$: representing the linear velocity of point O_j in the motion of $\mathscr{R}_j$ w.r.t. $\mathscr{R}_i$ with coordinates given in frame $\mathscr{R}_i$ (sometimes written as ${}^iV_{O_j \in j/i}$), such that :

    $$ {}^iV_{ij}\left(O_j\right) = \left[\frac{d}{dt} O_i O_j\right]_{\mathscr{R}_i} $$

    (coordinates reference frame, frame for computing/observing velocity, frame in motion)
  - ${}^i\omega_{ij} \in \mathbb{R}^3$: representing the angular velocity of the body in the motion of $\mathscr{R}_j$ w.r.t. $\mathscr{R}_i$ with coordinates given in frame $\mathscr{R}_i$ (sometimes written as ${}^i\omega_{j/i}$).

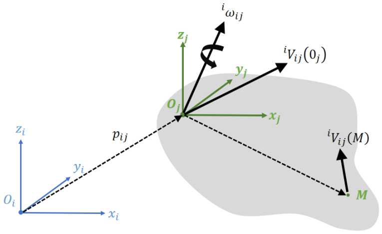
Representation of twist

Change of reference point on the moving body

The relationship between velocities of two points on the same rigid body $\mathscr{C}_j$ w.r.t. frame $\mathscr{R}_i$ are given by *Varignon* formula :

$$ {}^iV_{ij}(M) = {}^iV_{ij}(O_j) + {}^i\omega_{ij} \times {}^iO_jM \Rightarrow \boxed{\begin{pmatrix} {}^iV_{ij}(M) \\ {}^i\omega_{ij} \end{pmatrix} = \begin{bmatrix} \mathbb{I}_{3\times 3} & -{}^i\widehat{O_jM} \\ \mathbb{O}_{3\times 3} & \mathbb{I}_{3\times 3} \end{bmatrix} \begin{pmatrix} {}^iV_{ij}(O_j) \\ {}^i\omega_{ij} \end{pmatrix}} $$

- All vectors in the formula are meant to be referred to the coordinate frame $\mathscr{R}_i$;
- Recall that the resultant ${}^i\omega_{ij}$ remains independent of the considered point of $\mathscr{C}_j$.



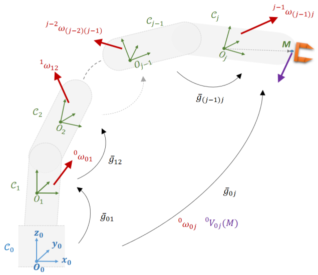
Representation of linear velocities at points O_j and M on the same rigid body $\mathscr{C}_j$.

Composition of twists

Finding the resultant velocity due to the relative motion of several body frames

Let M be a point of body $\mathscr{C}_j$ in motion w.r.t. to body $\mathscr{C}_{j-1}$, itself in motion w.r.t. to body $\mathscr{C}_{j-2}$, ..., itself in motion w.r.t. to body $\mathscr{C}_1$, itself in motion w.r.t. to body $\mathscr{C}_0$. The velocity field is computed as follows :

$$^0\mathscr{V}_{0j} = {}^0\mathscr{V}_{01} + {}^0\mathscr{V}_{12} + \ldots + {}^0\mathscr{V}_{(j-2)(j-1)} + {}^0\mathscr{V}_{(j-1)j}$$



Composition of twists

Finding the resultant velocity due to the relative motion of several body frames

Let M be a point of body $\mathscr{C}_j$ in motion w.r.t. to body $\mathscr{C}_{j-1}$, itself in motion w.r.t. to body $\mathscr{C}_{j-2}$, ..., itself in motion w.r.t. to body $\mathscr{C}_1$, itself in motion w.r.t. to body $\mathscr{C}_0$. The velocity field is computed as follows :

$$^0\mathscr{V}_{0j} = ^0\mathscr{V}_{01} + ^0\mathscr{V}_{12} + \ldots + ^0\mathscr{V}_{(j-2)(j-1)} + ^0\mathscr{V}_{(j-1)j}$$

- Absolute linear velocity at a particular point as the sum of relative linear velocities :

$$^0V_{0j}(M) = {}^0V_{01}(M) + {}^0V_{12}(M) + \ldots + {}^0V_{(j-2)(j-1)}(M) + {}^0V_{(j-1)j}(M)$$

  - Absolute velocity ${}^0V_{0j}(M)$: velocity of M from body $\mathscr{C}_j$ in motion w.r.t. to frame $\mathscr{R}_0$.
  - Relative velocity ${}^0V_{(j-k)(j-k+1)}(M)$: velocity of M supposed being fixed to body $\mathscr{C}_{j-k+1}$ in the motion w.r.t. to frame $\mathscr{R}_{j-k}$ with coordinates given in frame $\mathscr{R}_0$.
- Rule for composition of instantaneous angular velocities :

$$\begin{aligned} {}^0\omega_{0j} &= {}^0\omega_{01} + {}^0\omega_{12} + \ldots + {}^0\omega_{(j-2)(j-1)} + {}^0\omega_{(j-1)j} \\ &= {}^0\omega_{01} + R_{01}\,{}^1\omega_{12} + \ldots + R_{0(j-2)}\,{}^{(j-2)}\omega_{(j-2)(j-1)} + R_{0(j-1)}\,{}^{(j-1)}\omega_{(j-1)j} \end{aligned}$$

Representation of wrench or static screw

Wrench or Static screw : definition

A generalized force acting on a rigid body $\mathscr{C}_j$, to which is attached a frame $\mathscr{R}_j$, due to its physical interaction with a rigid body $\mathscr{C}_i$, to which is attached a frame $\mathscr{R}_i$, consists of a linear component (pure force) and an angular component (pure moment) given at one particular point. This generalized force can be represented as a vector in $\mathbb{R}^6$:

$$ {}^i\mathscr{F}_{ij} = \begin{bmatrix} {}^i m_{ij}\left(O_j\right) \\ {}^i f_{ij} \end{bmatrix} $$

- Note that the vector field of the *moment* m constitutes a screw where the *resultant* of the screw is f :

$$ {}^i m_{ij}(M) = {}^i m_{ij}\left(O_j\right) + {}^i f_{ij} \times {}^i O_j M $$

  for point $M \in \mathscr{C}_j$.
- It is often more practical to permute the order of f and m. In the following, we will use that re-ordering. In this case, the wrench representation becomes :

$$ {}^i\mathscr{F}_{ij} = \begin{bmatrix} {}^i f_{ij} \\ {}^i m_{ij}\left(O_j\right) \end{bmatrix} $$

Transformation of screws

Transforming screw from one frame to another

The 6×6 *screw transformation matrix* X allows to transform screws among coordinate frames in considering both a base change and a point change. Let $g_{ab} = (p_{ab}, R_{ab}) \in SE(3)$ be the mapping from one coordinate system into another. The *screw transformation matrix* is given as follows :

$$^{a}\mathscr{V}_{ij} = X_{ab}\,{}^{b}\mathscr{V}_{ij} \quad \text{with} \quad X_{ab} = \begin{bmatrix} R_{ab} & \widehat{p}_{ab} R_{ab} \\ \mathbb{O}_{3\times 3} & R_{ab} \end{bmatrix}$$

and

$$^{a}\mathscr{F}_{ij} = (X_{ab})^{-T}\,{}^{b}\mathscr{F}_{ij} \quad \text{with} \quad X_{ab}^{-T} = X_{ba}^{T} = \begin{bmatrix} R_{ba}^{T} & \mathbb{O}_{3\times 3} \\ -R_{ba}^{T}\widehat{p}_{ba} & R_{ba}^{T} \end{bmatrix}$$

Demonstration

The transformation matrix between screws has the following properties :

- Equivalent expressions :

$$X_{ab} = \begin{bmatrix} R_{ba}^{T} & -R_{ba}^{T}\widehat{p}_{ba} \\ \mathbb{O}_{3\times 3} & R_{ba}^{T} \end{bmatrix} \text{ and } (X_{ab})^{-T} = \begin{bmatrix} R_{ab} & \mathbb{O}_{3\times 3} \\ \widehat{p}_{ab} R_{ab} & R_{ab} \end{bmatrix}$$

- Inverse :
  - its determinant is equal to one : $det\left(X_{ab}\right) = +1$;
  - expression of its inverse :

$$X_{ab}^{-1} = X_{ba} = \begin{bmatrix} R_{ab}^{T} & -R_{ab}^{T}\widehat{p}_{ab} \\ \mathbb{O}_{3\times 3} & R_{ab}^{T} \end{bmatrix}$$

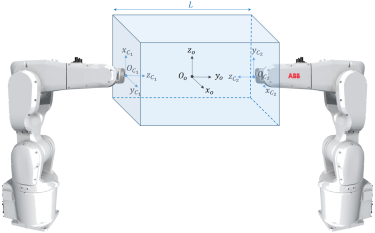
- Product :

$$X_{0n} = X_{01} X_{12} \dots X_{(n-1)n}$$

Transformation of screws

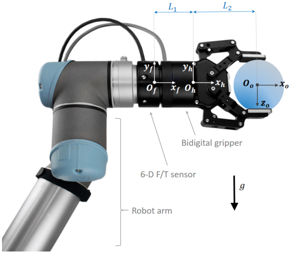
Applications

1. Multirobot arm grasping
   - Several contacts where ${}^{C_i}\mathscr{F}_{oi}$ is the wrench applied by the i^{th} robot body on object, given in the contact frame.
   - Computation of the net object wrench resulting from the collection of forces f_{oi} and moments m_{oi} acting on the object.



Example

2. Robot gripper holding an object
   - Gripper of mass m_h holding an object of mass m_o in a gravitational field g.
   - Computation of the net object wrench measured by the six-axis force/torque sensor mounted between the gripper and the robot arm.



Example

Transformation models for robotics

Computation of some mathematical models for design and control of robots, such as :

- Transformation models between the joint space (in which the configuration of the robot is defined) and the task space (in which the location of the end-effector is specified) :
  - direct and inverse geometric models
    giving the location of the end-effector X as a function of the joint variables of the mechanism q and vice versa ;
  - direct and inverse kinematic models
    giving the velocity of the end-effector $\dot{X}$ as a function of the joint velocities $\dot{q}$ and vice versa.
- dynamic models giving the relations between the input torques or forces of the actuators Γ and the positions q, velocities $\dot{q}$ and accelerations $\ddot{q}$ of the joints.

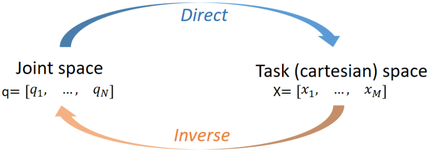
Transformation models for robotics

Computation of some mathematical models for design and control of robots, such as :

- Transformation models between the joint space (in which the configuration of the robot is defined) and the task space (in which the location of the end-effector is specified) :
  - direct and inverse geometric models
    giving the location of the end-effector X as a function of the joint variables of the mechanism q and vice versa ;
  - direct and inverse kinematic models
    giving the velocity of the end-effector $\dot{X}$ as a function of the joint velocities $\dot{q}$ and vice versa.
- dynamic models giving the relations between the input torques or forces of the actuators Γ and the positions q, velocities $\dot{q}$ and accelerations $\ddot{q}$ of the joints.

Remark :
In this course of introduction to robotics, only the case of simple open-tree structures will be studies. For example, in [13] and [21], the cases of parallel robots and multiple open-tree structures are studied.

Formulation of geometric models



- Case of the direct geometric model of the manipulator $X = f(q)$:
  - Expression of the position and orientation X of the end-effector as a function of joint variables q ;
  - Methodology :
    1. assign frames $\mathscr{R}_i$ to each rigid body $\mathscr{C}_i$ of the chain ;
    2. description of their relative position/orientation using one dedicated setting (convention used in this course : Modified *Denavit-HartenbeR* - MDH -, called *Khalil-Kleinfinger* [14]) ;
    3. use of homogeneous transformation matrices to describe change transformation $\overline{g}_{(i-1)i}$ between two adjacent bodies $\mathscr{C}_{i-1}$ and $\mathscr{C}_i$;
    4. forward kinematic of the complete kinematic chain from $\mathscr{C}_0$ to $\mathscr{C}_n$ obtained recursively.
- Case of the inverse geometric description of the manipulator $q = f^{-1}(X)$:
  - Joint coordinates q needed to bring the end-effector in a prescribed position and orientation X ;
  - Nature and number of equations raising the issues about *existence* and *multiplicity* of the solution.

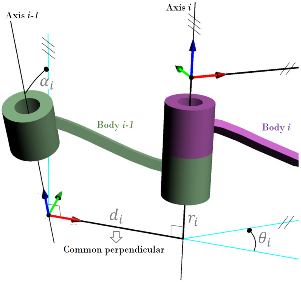
Khalil-Kleinfinger convention

Rigid body given by the relative position of its axes $i-1$ and i :

- distance d_i (length of the common perpendicular to the two axes) ;
- twisting angle α_i (angle of the rotation around the common perpendicular that brings the two axes parallel) ;

Relative position of two successive bodies thanks to two parameters :

- angle θ_i (angle of the needed rotation around axis i to bring the common perpendicular of body $i-1$ parallel to the one of body i) ;
- distance r_i which must then be translated along the i axis to bring them into coincidence..



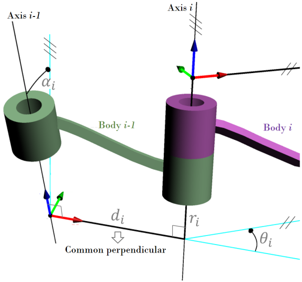
Khalil-Kleinfinger convention

Rigid body given by the relative position of its axes $i-1$ and i :

- distance d_i (length of the common perpendicular to the two axes) ;
- twisting angle α_i (angle of the rotation around the common perpendicular that brings the two axes parallel) ;

Relative position of two successive bodies thanks to two parameters :

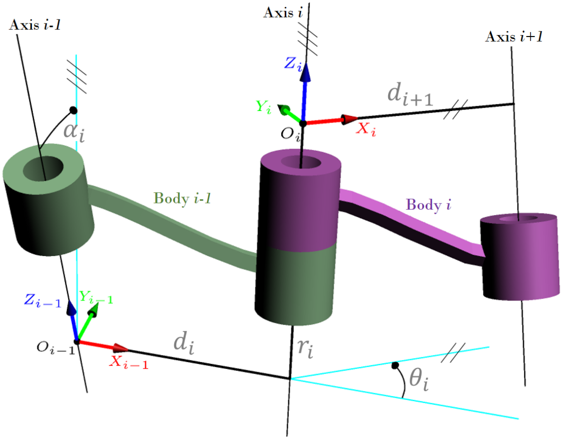
- angle θ_i (angle of the needed rotation around axis i to bring the common perpendicular of body $i-1$ parallel to the one of body i) ;
- distance r_i which must then be translated along the i axis to bring them into coincidence..



Remarks :

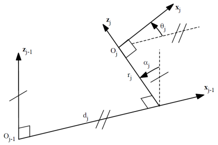
1. θ_i or r_i : joint variables
   - $q_i = \theta_i$ for a revolute joint ;
   - $q_i = r_i$ for a prismatic joint.
2. $(\alpha_i, d_i, \theta_i, r_i)$: four parameters constituting a minimal geometrical characterization of a body with the following body.

Khalil-Kleinfinger convention



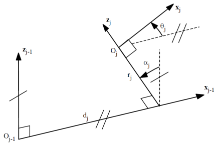
Methodology for computing the DGM

1. Assignation of orthonormal frame $\mathscr{R}_i$ to each body $\mathscr{C}_i$:
   - the axis Z_i is along the axis of joint connecting body $\mathscr{C}_{i-1}$ to body $\mathscr{C}_i$;
   - the axis X_i is aligned with the common normal between Z_i and Z_{i+1} (i.e. $X_i = Z_i \wedge Z_{i+1}$) ;
     - if Z_i and Z_{i+1} are collinear, then X_i is not unique and can be taken in any plane perpendicular to them ;
     - if Z_i and Z_{i+1} are parallel, then X_i is not unique and is in the plane defined by them ;
     - in the case of intersecting joint axes, X_i is normal to the plane defined by them and passing through their intersection point ;
   - the Y_i axis is formed by the right-hand rule to complete the coordinate system (X_i, Y_i, Z_i) (i.e. $Y_i = Z_i \wedge X_i$).



Methodology for computing the DGM

1. Assignation of orthonormal frame $\mathscr{R}_i$ to each body $\mathscr{C}_i$:
   - the axis Z_i is along the axis of joint connecting body $\mathscr{C}_{i-1}$ to body $\mathscr{C}_i$;
   - the axis X_i is aligned with the common normal between Z_i and Z_{i+1} (i.e. $X_i = Z_i \wedge Z_{i+1}$) ;
     - if Z_i and Z_{i+1} are collinear, then X_i is not unique and can be taken in any plane perpendicular to them ;
     - if Z_i and Z_{i+1} are parallel, then X_i is not unique and is in the plane defined by them ;
     - in the case of intersecting joint axes, X_i is normal to the plane defined by them and passing through their intersection point ;
   - the Y_i axis is formed by the right-hand rule to complete the coordinate system (X_i, Y_i, Z_i) (i.e. $Y_i = Z_i \wedge X_i$).



2. Transformation between succesive bodies $\mathscr{C}_{i-1}$ and $\mathscr{C}_i$ given by the following four parameters :
   - α_i : angle between axes Z_{i-1} and Z_i around axis X_{i-1} ;
   - d_i : distance between axes Z_{i-1} and Z_i along axis X_{i-1} ;
   - θ_i : angle between axes X_{i-1} and X_i around axis Z_i ;
   - r_i : distance between axes X_{i-1} and X_i along axis Z_i ;

Methodology for computing the DGM

3. Change transformation between two adjacent bodies :
   - Transformation matrix defining the frame $\mathscr{R}_i$ relative to frame $\mathscr{R}_{i-1}$

$$
\begin{aligned}
\overline{g}_{(i-1)i} &= R_{x,\alpha_i}\, Trans\,(x, d_i)\, R_{z,\theta_i}\, Trans\,(z, r_i) \\
&= \begin{bmatrix} cos\,(\theta_i) & -sin\,(\theta_i) & 0 & d_i \\ cos\,(\alpha_i)\, sin\,(\theta_i) & cos\,(\alpha_i)\, cos\,(\theta_i) & -sin\,(\alpha_i) & -r_i sin\,(\alpha_i) \\ sin\,(\alpha_i)\, sin\,(\theta_i) & sin\,(\alpha_i)\, cos\,(\theta_i) & cos\,(\alpha_i) & r_i cos\,(\alpha_i) \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} R_{(i-1)i} & p_{(i-1)i} \\ 0_{1\times 3} & 1 \end{bmatrix}
\end{aligned}
$$

Methodology for computing the DGM

3. Change transformation between two adjacent bodies :
   - Transformation matrix defining the frame $\mathscr{R}_i$ relative to frame $\mathscr{R}_{i-1}$

$$
\begin{aligned}
\overline{g}_{(i-1)i} =& R_{x,\alpha_i}\, Trans\,(x, d_i)\, R_{z,\theta_i}\, Trans\,(z, r_i) \\
=& \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & d_i \\ \cos(\alpha_i)\sin(\theta_i) & \cos(\alpha_i)\cos(\theta_i) & -\sin(\alpha_i) & -r_i \sin(\alpha_i) \\ \sin(\alpha_i)\sin(\theta_i) & \sin(\alpha_i)\cos(\theta_i) & \cos(\alpha_i) & r_i \cos(\alpha_i) \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
=& \begin{bmatrix} R_{(i-1)i} & p_{(i-1)i} \\ 0_{1\times3} & 1 \end{bmatrix}
\end{aligned}
$$

   - Inverse transformation matrix defining the frame $\mathscr{R}_{i-1}$ relative to frame $\mathscr{R}_i$

$$
\begin{aligned}
\overline{g}^{-1}_{(i-1)i} =& \overline{g}_{i(i-1)} \\
=& Trans\,(z, -r_i)\, R_{z,-\theta_i}\, Trans\,(x, -d_i)\, R_{x,-\alpha_i} \\
=& \begin{bmatrix} R^t_{(i-1)i} & -R^t_{(i-1)i} p_{(i-1)i} \\ 0_{1\times3} & 1 \end{bmatrix} \\
=& \begin{bmatrix} & R^t_{(i-1)i} & & \begin{matrix} -d_i \cos(\theta_i) \\ d_i \sin(\theta_i) \\ -r_i \end{matrix} \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
$$

Methodology for computing the DGM

4. DGM of the whole kinematic chain :
Representation of DGM thanks to the transformation matrix $\overline{g}_{0n}$ in the case of a simple open-tree chain

$$X = f(q) \qquad \text{where} \qquad f = \overline{g}_{0N}$$

- Computation of $\overline{g}_{0n}$ obtained recursively

$$\overline{g}_{0N}(q) = \overline{g}_{01}(q_1)\dots\overline{g}_{(i-1)i}(q_i)\dots\overline{g}_{(N-1)N}(q_N)$$

- $q = [q_1 \dots q_N]^t$ being the vector of joint variables ;
- $X = [x_1 \dots x_M]^t$ being the vector of cartesian variables.
  - Several possibilities exist for parameterizing the orientation from vector X (using, for example, *Euler* angles computed from the direction cosines of R_{0n} to obtain $X = [p^x \, p^y \, p^z \, \alpha \beta \gamma]$).

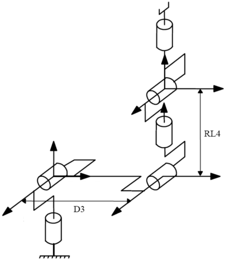
Recommendations for choosing the frames :

- $\mathscr{R}_0$ confounded with $\mathscr{R}_1$ when $q_1 = 0$ (cancellation of parameters α_1, d_1) ;
- X_N (free since Z_{N+1} does not exist) chosen colinear to X_{N-1} when $q_N = 0$;
- X_i chosen such that $r_i = 0$ or $r_{i+1} = 0$ when Z_i is parallel to Z_{i+1}.

DGM of serial robot

Example : case of robot RX130L from *Staubli*^{TM}

- Assignation of frames ;
- Parameterization according to *Khalil-Kleinfinger* convention ;
- Computation of $\overline{g}_{(i-1)i}$ for $i = 1, \ldots, 4$.



i	α_i	d_i	θ_i	r_i
1	?	?	?	?
2	?	?	?	?
3	?	?	?	?
4	?	?	?	?
5	?	?	?	?
6	?	?	?	?

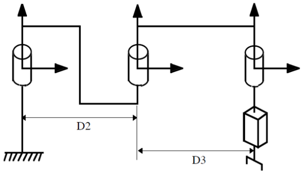
Example

DGM of serial robot

Example : case of robot SCARA IRB 910SC from *ABB*^{TM}

- Assignation of frames ;
- Parameterization according to *Khalil-Kleinfinger* convention.

i	α_i	d_i	θ_i	r_i
1	?	?	?	?
2	?	?	?	?
3	?	?	?	?
4	?	?	?	?



Example

Introduction

Definition

The direct kinematic model (DKM) of a robot manipulator gives the velocity of the end-effector $\dot{X}$ as a function of joint velocities $\dot{q}$:

$$\dot{X} = J(q)\,\dot{q}$$

where $J(q)$ denotes the Jacobian matrix of dimensions $M \times N$ and given by $\frac{\partial X}{\partial q}$.

- The same Jacobian matrix also appears in the direct differential model, which provides the differential displacement of the end-effector dX in terms of the differential variation of the joint variables dq :

$$dX = J(q)\,dq$$

- Interests of the Jacobian matrix for robots :
  - usefulness for singularities analysis and dimension of the reachable workspace of the robot ;
  - usefulness for numerically computing the solutions to the problem of Inverse Geometric Model (IGM) ;
  - usefulness for establishing in static the relationship between wrench exerted by the end-effector and forces/torques at the actuators level.
- Main methods to compute the Jacobian matrix :
  1. through direct derivation of DGM ;
  2. through velocities composition.

1- Jacobian matrix as derivative of the DGM

- The Jacobian matrix can be obtained by differentiating the DGM $X = f(q)$ using its partial derivatives, such that :

$$^{0}J(q) = \begin{bmatrix} \frac{\partial f_1}{\partial q_1} & \cdots & \frac{\partial f_1}{\partial q_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_M}{\partial q_1} & \cdots & \frac{\partial f_M}{\partial q_N} \end{bmatrix}$$

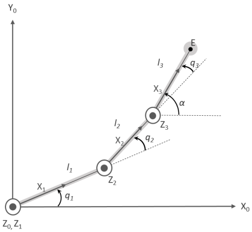
  - let note that the DGM f is given in frame $\mathscr{R}_0$ in this case ;
  - functions f_i are differentiable since it is formed by affines and trigonometric functions.
- let note that the Jacobian matrix J is configuration-dependent.
- Specificities of this approach :
  - convenient for simple robots having a reduced number of degrees of freedom ;
  - less practical for a general N degree-of-freedom robot.

1- Jacobian matrix as derivative of the DGM

Examples :

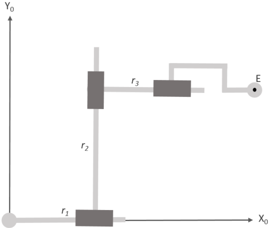
- Computation of homogeneous transformation matrix of end-effector in $\mathscr{R}_0$;
- Computation of ${}^0J$ through the derivation of the DGM.

Case of planar RRR robot



Example

Case of planar PPP robot



Example

1- Jacobian matrix as derivative of the DGM

Main steps for computing ${}^0J\left(q\right)$

$$ {}^0\mathscr{V}_{0N}\left(0_n\right) = \left[\begin{array}{c} {}^0V_{0N}\left(O_N\right) \\ {}^0\omega_{0N} \end{array} \right] = \left[\begin{array}{c} {}^0J_v\left(q\right) \\ {}^0J_\omega\left(q\right) \end{array} \right] \dot{q} = {}^0J\left(q\right)\dot{q} $$

where ${}^0V_{0N}\left(O_N\right)$ and ${}^0\omega_{0N}$ are the twist components of the end-effector velocity given in base frame.

- Linear velocity part :
  1. Computation of the homogeneous transformation matrix $\overline{g}_{0N}$ (used for DGM computation) ;
  2. Extraction of the position vector $p_{0N}\left(q\right)$ from the last column of $\overline{g}_{0N}$;
  3. Derivative of $p_{0N}\left(q\right)$ providing the upper submatrix of ${}^0J\left(q\right)$:

$$ {}^0V_{0N}\left(O_N\right) = \frac{d}{dt} p_{0N}\left(q\right) = \frac{d}{dt} O_0O_N = {}^0J_v\left(q\right)\dot{q}. $$

- Angular velocity part :
  1. Extraction of the matrix $R_{0N}\left(q\right)$ from $\overline{g}_{0N}$;
  2. Computation of ${}^0\widehat{\omega}_{0N}$ according to ${}^0\widehat{\omega}_{0N} = \dot{R}_{0N}R_{N0} = \left(\frac{\partial R_{0N}}{\partial q_1}\dot{q}_1 + \ldots + \frac{\partial R_{0N}}{\partial q_N}\dot{q}_N\right) R_{N0}$
  3. Term-by-term identification of ${}^0\omega_{0N} \in \mathbb{R}^3$ from the preproduct matrix ${}^0\widehat{\omega}_{0N}$ to compute the lower submatrix of ${}^0J\left(q\right)$:

$$ {}^0\omega_{0N} = {}^0J_\omega\left(q\right)\dot{q} $$

2- Jacobian matrix using velocity composition rule

Effect of the i^{th} joint on the end-effector velocity

- Expression of the cartesian velocity ${}^{i}\mathscr{V}_{iN}\left(O_N\right) = \begin{bmatrix} {}^{i}V_{iN}\left(O_N\right) \\ {}^{i}\omega_{iN} \end{bmatrix} \in \mathbb{R}^6$ of end-effector $\mathscr{R}_N$ due to the i^{th} joint of the poly-articulated chain :
  - ${}^{i}V_{iN}\left(O_N\right)$: velocity of origin O_N relative to $\mathscr{R}_i$, such that ${}^{i}V_{iN}\left(O_N\right) = \frac{dp_{iN}}{dt}$ with $p_{iN} = O_i O_N$, element of translation in the homogeneous transformation $\overline{g}_{iN}$;
  - ${}^{i}\omega_{iN}$: angular velocity of frame $\mathscr{R}_N$ relative to $\mathscr{R}_i$.
- Type of joint :
  - Case of a prismatic joint ($q_i = r_i$)

$$ {}^{i}\mathscr{V}_{iN}\left(O_N\right) = \begin{bmatrix} {}^{i}V_{iN}\left(O_N\right) \\ {}^{i}\omega_{iN} \end{bmatrix} = \begin{bmatrix} Z_i \\ 0_{3\times 1} \end{bmatrix} \dot{q}_i $$

  - Case of a revolute joint ($q_i = \theta_i$)

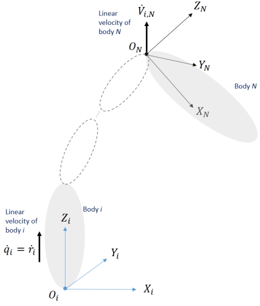
$$ {}^{i}\mathscr{V}_{iN}\left(O_N\right) = \begin{bmatrix} {}^{i}V_{iN}\left(O_N\right) \\ {}^{i}\omega_{iN} \end{bmatrix} = \begin{bmatrix} Z_i \times p_{iN} \\ Z_i \end{bmatrix} \dot{q}_i $$

- Remarks :
  - According to *Khalil-Kleinfinger* convention, the joint axis is along the vector $Z_i = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^t$.
  - All the chain between joint i and extremity N is supposed to constitute one single rigid body.

2- Jacobian matrix using velocity composition rule

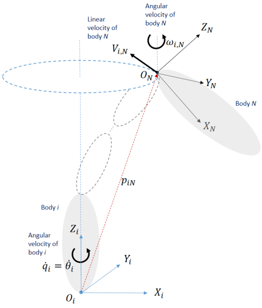
Effect of the i^{th} joint on the end-effector velocity

Case of a prismatic joint



$$\begin{bmatrix} {}^iV_{iN}(O_N) \\ {}^i\omega_{iN} \end{bmatrix} = \underbrace{\begin{bmatrix} Z_i \\ 0_{3\times 1} \end{bmatrix}}_{={}^iJ_i(q)} \dot{q}_i$$

Case of a revolute joint



$$\begin{bmatrix} {}^iV_{iN}(O_N) \\ {}^i\omega_{iN} \end{bmatrix} = \underbrace{\begin{bmatrix} Z_i \times p_{iN} \\ Z_i \end{bmatrix}}_{={}^iJ_i(q)} \dot{q}_i$$

2- Jacobian matrix using velocity composition rule

Effect of the i^{th} joint on the end-effector velocity

- Projection of cartesian velocity in base frame $\mathscr{R}_0$
  - Linear velocity ${}^0V_{iN}$ of the origin O_N w.r.t. $\mathscr{R}_i$ given in $\mathscr{R}_0$:

$$ {}^0V_{iN}(O_N) = R_{0i}{}^iV_{iN}(O_N) $$

  - Angular velocity ${}^0\omega_{iN}$ of $\mathscr{R}_N$ w.r.t. $\mathscr{R}_i$ given in $\mathscr{R}_0$:

$$ {}^0\omega_{iN}(O_N) = R_{0i}{}^i\omega_{iN}(O_N) $$

i.e.

$$ {}^0\mathscr{V}_{iN}(O_N) = \begin{bmatrix} {}^0V_{iN}(O_N) \\ {}^0\omega_{iN} \end{bmatrix} = {}^0J_i(q)\,\dot{q}_i $$

with

$$ {}^0J_i(q) = \begin{bmatrix} R_{0i} & 0_{3\times3} \\ 0_{3\times3} & R_{0i} \end{bmatrix} {}^iJ_i(q) $$

Remarks about dependency of ${}^0J_i$:

- with the reference frame in which the velocities are given (possibility to project the cartesian velocity in another frame if required) ;
- with the configuration q of manipulator.

2- Jacobian matrix using velocity composition rule

Methodology for the practical computation of ${}^0J\left(q\right)$

Using composition of the differents twists ${}^0\mathscr{V}_{iN}\left(O_N\right)$ for $i = 1, \ldots, N$, the Jacobian matrix that maps joint velocities $\dot{q}$ of the robot to the cartesian velocities $\dot{X}$ given in inertial reference frame $\mathscr{R}_0$ is given by :

$$^0J\left(q\right) = \left[\begin{array}{ccccc} ^0J_1\left(q\right) & \ldots & ^0J_i\left(q\right) & \ldots & ^0J_N\left(q\right) \end{array}\right]$$

where ${}^0J_i\left(q\right)$ defines the cartesian velocity due to the action of the i^{th} joint given in frame $\mathscr{R}_0$, according to the type of joint :

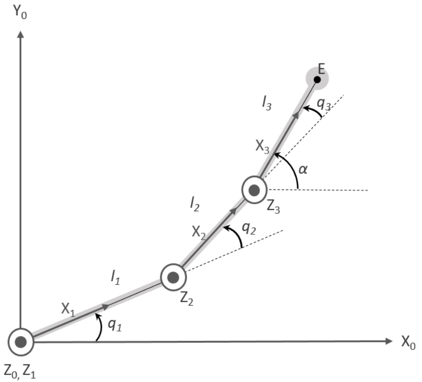
$$^0J_i\left(q\right) = \begin{cases} \left[\begin{array}{c} R_{0i}Z_i \\ 0_{3\times1} \end{array}\right] & \text{if the } i^{th} \text{ joint is prismatic} \\ \\ \left[\begin{array}{c} R_{0i}\left(Z_i \times p_{iN}\right) \\ R_{0i}Z_i \end{array}\right] & \text{if the } i^{th} \text{ joint is revolute} \end{cases}$$

with :

- R_{0i} : rotation matrix of frame $\mathscr{R}_i$ w.r.t. $\mathscr{R}_0$;
- Z_i : unit vector of the i^{th} joint given in frame $\mathscr{R}_i$;
- p_{iN} : position vector from origin of frame $\mathscr{R}_i$ to origin of frame $\mathscr{R}_N$.

2- Jacobian matrix using velocity composition rule

Example : returning to the previous case of the planar RRR robot
Computation of the Jacobian matrix in base frame ${}^{0}J$



Example

Kinematic model associated with the task coordinates representation

Definition of analytical Jacobian

Let the end-effector position X_p and orientation X_r be specified in terms of a minimal number of parameters in the operational space (for example using *Euler* angles, RPY, etc.) and are given by $X = \begin{bmatrix} X_p & X_r \end{bmatrix}^T$ for representing the pose of frame $\mathscr{R}_N$ relative to frame $\mathscr{R}_0$, such that :

$$\begin{bmatrix} X_p \\ X_r \end{bmatrix} = \underbrace{\begin{bmatrix} \Omega_p & \mathbb{O}_{3\times 3} \\ \mathbb{O}_{3\times 3} & \Omega_r \end{bmatrix}}_{\Omega} \begin{bmatrix} {}^0V_{0N}(O_N) \\ {}^0\omega_{0N} \end{bmatrix}.$$

The *analytical Jacobian* $J_X(q)$ can be related to the previously computed Jacobian ${}^0J(q)$ (referred as *geometric* Jacobian) as follows :

$$\begin{bmatrix} X_p \\ X_r \end{bmatrix} = \underbrace{\Omega^0 J(q)}_{J_X(q)} \dot{q}.$$

- This relationship shows that $J_X(q)$ and ${}^0J(q)$, in general, differ. However, when the dof cause rotations of the end-effector all about the same fixed axis in space (as it is the case for the previously studied RRR planar robot), the two Jacobian are the same.
- The matrix Ω_p is equal to $\mathbb{I}_{3\times 3}$ when the position of frame $\mathscr{R}_N$ is described by the Cartesian coordinates.
- In the example given in the previous chapter, let recall that we have already calculated $\Omega_{r_{\text{Euler}}}$ for *Euler* orientation representation :

$$\Omega_r = \begin{bmatrix} -sin(\alpha)cotg(\beta) & cos(\alpha)cotg(\beta) & 1 \\ cos(\alpha) & sin(\alpha) & 0 \\ \frac{sin(\alpha)}{sin(\beta)} & -\frac{cos(\alpha)}{sin(\beta)} & 0 \end{bmatrix}$$

Dimension of the task space for simply open-tree robot

Redundancy

For a given joint configuration q, the rank r of the Jacobian matrix $J \in \mathbb{R}^{M \times N}$ corresponds to the degrees-of-freedom of the robot task space (associated to the end-effector frame), i.e. to the dimension of the reachable task space in this *particular configuration* q.

The maximum rank r_{max} that takes the Jacobian matrix in all its possible configurations is the number of degrees-of-freedom M of the task space of a robot.

- If $M = N$ (N being the number of degrees-of-freedom of the robot, equal to the number of motorized joint in the case of a simply open-tree robot), then the robot is said to be *non-redundant*.
  - However, let note that a robot which is not redundant ($M = N$) may be *locally redundant* or *redundant w.r.t. a particular task* whose number of degrees of freedom, r, is less than the number of degrees of freedom of the robot N.
- If $M < N$, then the robot is *redundant* of order $N - M$.

Dimension of the task space for simply open-tree robot

Analysis of the range and null of the Jacobian matrix J

- The Jacobian describes the linear mapping from the joint velocity space to the end-effector velocity space.
  - the *range* of J is the subspace $\mathscr{R}(J)$ in $\mathbb{R}^M$ of the end-effector velocities that can be generated by the joint velocities, in the given manipulator posture ;
  - the *null* of J is the subspace $\mathscr{N}(J)$ in $\mathbb{R}^N$ of the joint velocities that do not produce any end-effector velocity, in the given manipulator posture.
- The following relation holds independently of the rank r of the matrix J :

$$\boxed{\dim(\mathscr{R}(J)) + \dim(\mathscr{N}(J)) = N}$$

and the dimension of the null of J, equal to $N - \dim(\mathscr{R}(J))$, is an indicator of the order of redundancy.

  - if the Jacobian has *full* rank, one has :

$$\dim(\mathscr{R}(J)) = M, \quad \dim(\mathscr{N}(J)) = N - M$$

and the range of J spans the entire space $\mathbb{R}^M$.
  - Instead, if the Jacobian degenerates at a *singularity*, the dimension of the range space decreases ($\dim(\mathscr{R}(J)) = r < M$) while the dimension of the null space increases.

Example : returning to the previous case of planar PPP robot
Analysis of the rank and the null space of the Jacobian matrix ${}^0J$

Example

Dimension of the task space for simply open-tree robot

Singularity

Singularities are solutions of :

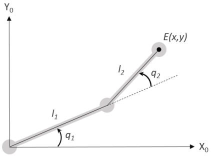
$$\begin{cases} det\left(J\right) = 0 & \text{for the non-redundant case (square matrix } J\text{),} \\ det\left(JJ^t\right) = 0 & \text{for the redundant case (non-square matrix } J\text{)} \end{cases}$$

where $det\left(\bullet\right)$ denotes the determinant of matrix $\bullet$.

- For certain joint configurations, it may happen that the rank r is inferior to M (the robot being redundant or not) : the number of degrees of freedom of the end-effector becomes less than the dimension of the task space. We say that the robot posseses a singularity or a local redundancy of order $M - r$.
  - this rank loss means that it is impossible to generate velocities along or around certain directions ;
  - close to this singularity, small velocities in the task space may imply significant (infinite) velocities in joint space.
- Types of singularities :
  1. Boundary singularity corresponding to points located at the border of the reachable workspace, i.e. robot in *extended* or *folded* configuration (possibility to avoid easily these configurations in bringing the manipulator away to the border of the reachable workspace)
  2. Internal singularity appearing inside the reachable workspace and generally caused by an alignment of two or more axes (more tricky problem as singular configurations can be reached in the workspace for a planned trajectory in the operational space)

Dimension of the task space for simply open-tree robot

Case study of the workspace of the planar RR robot (see previous example)

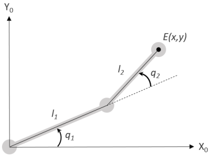


- Representation of its workspace considering no mechanical stops (assuming $L_1 > L_2$)
- Representation of singularity branches

Example

Dimension of the task space for simply open-tree robot

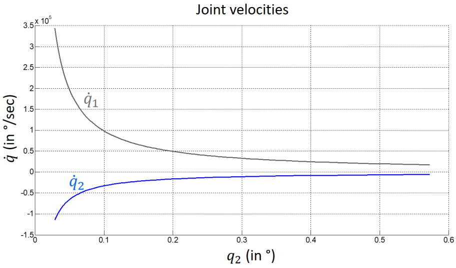
Case study of the workspace of the planar RR robot (see previous example)



- Representation of its workspace considering no mechanical stops (assuming $L_1 > L_2$)
- Representation of singularity branches

Example

- Influence of the closeness of singular configuration on the joint velocity



Evolution of joint velocity $\dot{q} = {}^0J^{-1}(q)\,\dot{X}_d$ enabling to reach the cartesian velocity $\dot{X}_d = \begin{bmatrix} -1 & -1 \end{bmatrix}^t$ around the joint configuration $q_1 = 0$ and $0,03^\circ < q_2 < 0,55^\circ$

Transmission of velocities between joint and task spaces

Recall on decomposition in singular values (function *SVD* in *Matlab*^{TM})
Let consider J of dimensions $m \times n$ and of rank r for a given configuration. There exists orthogonal matrices $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$, such that

$$J = U\Sigma V^t$$

The matrix $\Sigma \in \mathbb{R}^{m\times n}$ takes the following form :

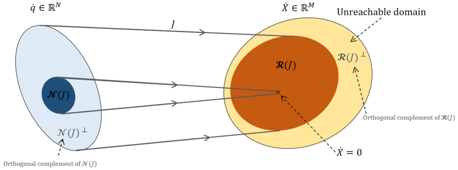
$$\Sigma = \begin{bmatrix} S & 0_{r\times(n-r)} \\ 0_{(m-r)\times r} & 0_{(m-r)\times(n-r)} \end{bmatrix}$$

$S \in \mathbb{R}^{r\times r}$ is the diagonal matrix computed with the non-null singular values σ_i of J, ordered in a decreasing way : $\sigma_1 \geq \ldots \geq \sigma_r$.

- The singular values of J are equal to the square roots of the eigenvalues λ_i of the product J^tJ : $\sigma_i = \sqrt{\lambda_i\left(J^tJ\right)}$;
- The matrix V is constituted by the eigenvectors of J^tJ ;
- The matrix U is constituted by the eigenvectors of JJ^t ;
- Since $\sigma_i = 0$ for $i > r$, the direct kinematic model becomes : $\dot{X} = \sum_{i=1}^{r} \sigma_i U_i V_i^t \dot{q}$.

Transmission of velocities between joint and task spaces

Null and range spaces of J



- Orthonormal basis for the subspace of $\dot{q}$ generating an end-effector velocity : $(V_1, \ldots, V_r)$;
- Orthonormal basis of null space of J giving $\dot{X} = 0$: $(V_{r+1}, \ldots, V_n)$;
- Orthonormal basis for the set of the achievable end-effector velocities $\dot{X}$, defining the range space of J : $(U_1, \ldots, U_r)$;
- Orthonormal basis for the subspace composed of the set of $\dot{X}$ that cannot be generated by the robot, defining the complement of the range space : $(U_{r+1}, \ldots, U_m)$.

Transmission of velocities between joint and task spaces

Velocity transmission performance

Assuming the joint velocity norm defined by $\dot{q}^t\dot{q} \leq 1$ (unit hyper-sphere), the resulting velocity in task space are given by the quadratic form (representing an ellipsoid in $\mathbb{R}^{\dot{X}}$)[30] :

$$\dot{X}^t \left(JJ^t\right)^{-1} \dot{X} \leq 1$$

The ellipsoid has a form and an orientation defined through JJ^t :

- the principal axes of the ellipsoid are given by the vectors $U_1, \ldots, U_m$ (i.e. eigenvectors of de JJ^t) ;
- the lengths of the principal axes are determined by the singular values $\sigma_1 \geq \ldots \geq \sigma_m$ (i.e. root square of the eigenvalues of JJ^t) ;
- its volume is an indicator of the robot capacity to generate velocity (velocity manipulability), given by : $\mathscr{W} = \sqrt{det\left(J\left(q\right)J^t\left(q\right)\right)}$ (or $\left|det\left(J\left(q\right)\right)\right|$ in the non-redundant case), thus $\mathscr{W} = \prod_{i=1}^{r} \sigma_i \geq 0$.

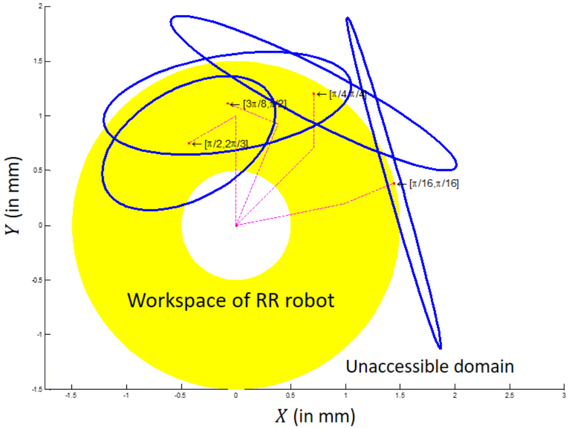
From the SVD decomposition of J, it follows that :

$$J = U\Sigma V^t \Rightarrow JJ^t = U\Sigma^2 U^t \Rightarrow \left(JJ^t\right)^{-1} = U\Sigma^{-2}U^t$$

Transmission of velocities between joint and task spaces

Velocity transmission performance

Case study of the RR robot workspace



Transmission of forces/torques between joint and task spaces

Mapping of an external wrench into joint torques

The interaction forces between the robot and its environment can be calculated directly using the kinematic Jacobian matrix. The joint forces/torques Γ_e resulting from the application of the wrench ${}^0\mathscr{F}_e$ giving the forces applied at the end of the poly-articulated chain is equal to

$$\Gamma_e = {}^0J(q)^t\,{}^0\mathscr{F}_e$$

where ${}^0\mathscr{F}_e$ denotes the static wrench applied by the robot end-effector on the environment given in coordinates frame $\mathscr{R}_0$:

$${}^0\mathscr{F}_e = [f_x, f_y, f_z, m_x, m_y, m_z]^t$$

and ${}^0J$ the Jacobian matrix computed at the application point of the external wrench on the robot.

Demonstration

- ${}^0\mathscr{F}_e$ is a vector of dimension $M \times 1$ ($M = 6$ in general) composed by forces f and torques m applied to the manipulator ;
- Γ_e is a vector of dimension $M \times 1$ composed by forces and/or torques applied to the n joints.

Transmission of forces/torques between joint and task spaces

Performance of forces/torques transmission [28]
Analogously, we can study the force transmission performance using a force manipulability ellipsoid, which corresponds to the set of achievable wrench in the task space $\mathbb{R}^M$ corresponding to the constraint $\Gamma^t\Gamma \leq 1$. The force ellipsoid is defined by :

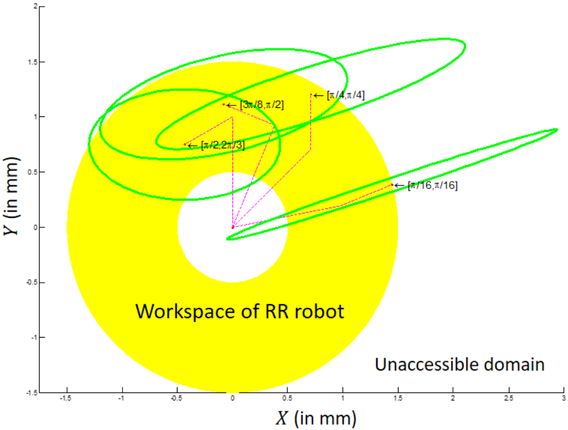
$$\mathscr{F}^t \left(JJ^t\right) \mathscr{F} \leq 1.$$

- Characteristics of the ellipsoid :
  - Same principal axes of the ellipsoids in velocity and force : $U_1, \ldots, U_m$ (eigenvectors of JJ^t) ;
  - Length of axes inversely proportional to singular values σ_i of J
    The velocity is controlled most accurately in the direction where the robot can resist large force disturbances, and force is most accurately controlled in the direction where the robot can rapidly adapt its motion.
- Let note that the singular configurations for applying cartesian velocity are identical to those for applying cartesian force (since $Rank\,(J) = Rank\,(J^t)$).

Transmission of forces/torques between joint and task spaces

Performance of forces/torques transmission

Case study of the RR robot workspace



Transmission of forces/torques between joint and task spaces

Duality force/velocity

- Subspaces analysis
  - The torques of the actuators are uniquely determined for an arbitrary wrench ${}^0\mathscr{F}_e$; the range space of J^T, denoted as $\mathscr{R}\left(J^T\right)$, is the set of Γ_e balancing the static wrench ${}^0\mathscr{F}_e$ according to $\Gamma_e = {}^0J\left(q\right)^t\,{}^0\mathscr{F}_e$;
  - For a zero Γ_e, the corresponding static wrench can be non-zero ; we thus define the null space of J^T, $\mathscr{N}\left(J^T\right)$, as the set of static wrenches that do not require actuator torques in order to be balanced.
- Following the previous Singular Values Decomposition (SVD) of matrix J for velocity analysis, the SVD of matrix J^T for force analysis leads to :

$$J^T = V\Sigma U^T$$

so that,

$$\mathbb{R}^M = \mathscr{R}\left(J\right) + \mathscr{R}\left(J\right)^{\perp} = \mathscr{R}\left(J\right) + \mathscr{N}\left(J^T\right),$$

$$\mathbb{R}^N = \mathscr{R}\left(J^T\right) + \mathscr{R}\left(J^T\right)^{\perp} = \mathscr{R}\left(J^T\right) + \mathscr{N}\left(J\right).$$

Physical insights

- $\mathscr{N}\left(J^T\right) = \mathscr{R}\left(J\right)^{\perp}$: in this configuration, the endpoint wrench is borne by the structure of the robot without requiring balancing actuator torques, while it also refers to the set of directions along which the robot cannot generate velocity.
- $\mathscr{R}\left(J^T\right)^{\perp} = \mathscr{N}\left(J\right)$: in this configuration, some joint torques cannot be compensated by the external wrench, while it also refers to the joint velocities that cannot generate cartesian velocities.

Inverse Geometric model (IGM)

Usefulness of the inverse kinematic models for manipulator : $q = f^{-1}(X)$

- Finding the joint coordinates q needed to bring the robot tool in a desired position and orientation X ;
- Transforming the coordinates for computer control algorithms from the desired task coordinates to references to joints coordinates.

Tricky problem : a general approach for finding its solution does not exist !

Inverse Geometric model (IGM)

Usefulness of the inverse kinematic models for manipulator : $q = f^{-1}(X)$

- Finding the joint coordinates q needed to bring the robot tool in a desired position and orientation X ;
- Transforming the coordinates for computer control algorithms from the desired task coordinates to references to joints coordinates.

Tricky problem : a general approach for finding its solution does not exist !

- Type and number of equations raising the problem of the *existence* and *multiplicity* of the solution(s), in the general case :
  - No solution (ex. specification of a targeted position X out of the robot workspace) ;
  - A finite set of solutions ;
  - Infinite numbers of solutions.

Inverse Geometric model (IGM)

Usefulness of the inverse kinematic models for manipulator : $q = f^{-1}(X)$

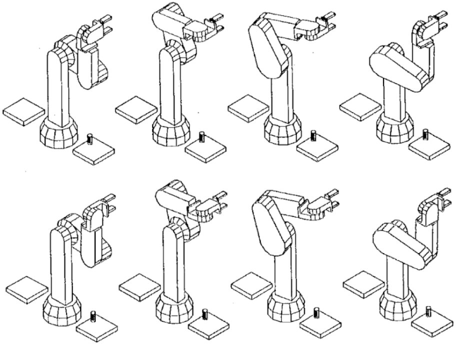
- Finding the joint coordinates q needed to bring the robot tool in a desired position and orientation X ;
- Transforming the coordinates for computer control algorithms from the desired task coordinates to references to joints coordinates.

Tricky problem : a general approach for finding its solution does not exist !

- Type and number of equations raising the problem of the *existence* and *multiplicity* of the solution(s), in the general case :
  - No solution (ex. specification of a targeted position X out of the robot workspace) ;
  - A finite set of solutions ;
  - Infinite numbers of solutions.
- Two main classes of approaches for solving the IGM :
  1. explicit solutions (true for decoupled six-dof robot, e.g. 6-dof robots with 3-dof spherical wrist mounted on a 3-dof arm [25], or robots with relatively simple geometry that have many zero distances and parallel or perpendicular joint axes [23])
  2. iterative numerical methods when no explicit form exists (mainly exploiting the inverse differential model).

Admissible solutions for the case $N = M$

Set of admissible solutions to the IGM problem : illustration for the case of a serial 6R robot (case where $M = N = 6$)



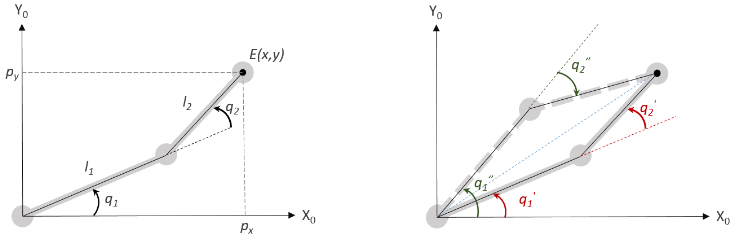
- 4 solutions out of singularities (for the only positioning of the wrist center) ;
- 8 solutions when considering the complete pose of the end-effector (spherical wrist : 2 alternative solutions for the last 3 joints).

Admissible solutions for the case $N = M$

Explicit solution : example of the 2R robot ($M = N = 2$)
Direct Geometric model :

$$\begin{cases} p_x & = l_1 cos\,(q_1) + l_2 cos\,(q_1 + q_2) \\ p_y & = l_1 sin\,(q_1) + l_2 sin\,(q_1 + q_2) \end{cases}$$

In the regular case, the IGM problem has two pairs of solutions : $\left(q_1^{'}, q_2^{'}\right)$ and $\left(q_1^{''}, q_2^{''}\right)$.



Example

Algorithms for numerical computation of IGM

Newton-Raphson method (for $M = N$)

- Usefulness when an analytical solution to the problem $X_d = f(q)$ does not exist or is diffcult to obtain ;
- When considering a first-order *Taylor* series approximation of the function f giving the DGM,

$$X_d = f(q_k) + \underbrace{\frac{\partial f(q_k)}{\partial q}}_{J(q_k)} (q - q_k) + o\left((q - q_k)^2\right)$$

we propose the iteration of joint variables at next step as follows :

$$q_{k+1} = q_k + J^{-1}(q_k)\left[X_d - f(q_k)\right]$$

- convergence if q^0 (initial conditions) *relatively* close to the solution $q^* : X_d = f(q^*)$;
- quadratic convergence rate in the neighbourhood of the solution ;
- problems *close to* singularities of the Jacobian matrix $J(q)$ in the redundant case ($M < N$).

Algorithms for numerical computation of IGM

Gradient-based method

- When considering the minimisation of the following objective-function,

$$H(q) = \frac{1}{2}\left\|X_d - f(q)\right\|^2 = \frac{1}{2}\left(X_d - f(q)\right)^t\left(X_d - f(q)\right)$$

the iteration joint variables at next step is made in the opposite direction of the gradient, so as to decrease the function H :

$$q_{k+1} = q_k - \alpha\nabla H(q_k) = q_k + \alpha J^t\left(q_k\right)\left[X_d - f(q_k)\right]$$

  - simpler on the computational point (transpose of the Jacobian matrix, and not its inverse) ;
  - direct usefulness for the case of task-redundant robot ;
  - searching for the amplification gain step α to guaranty decreasing of the error function H at each iteration (*linesearch* technique) ;
  - linear convergence rate ;

Algorithms for numerical computation of IGM

Gradient-based method

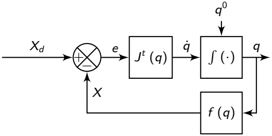
- When considering the minimisation of the following objective-function,

$$H(q) = \frac{1}{2}\left\|X_d - f(q)\right\|^2 = \frac{1}{2}\left(X_d - f(q)\right)^t\left(X_d - f(q)\right)$$

the iteration joint variables at next step is made in the opposite direction of the gradient, so as to decrease the function H :

$$q_{k+1} = q_k - \alpha \nabla H(q_k) = q_k + \alpha J^t\left(q_k\right)\left[X_d - f(q_k)\right]$$

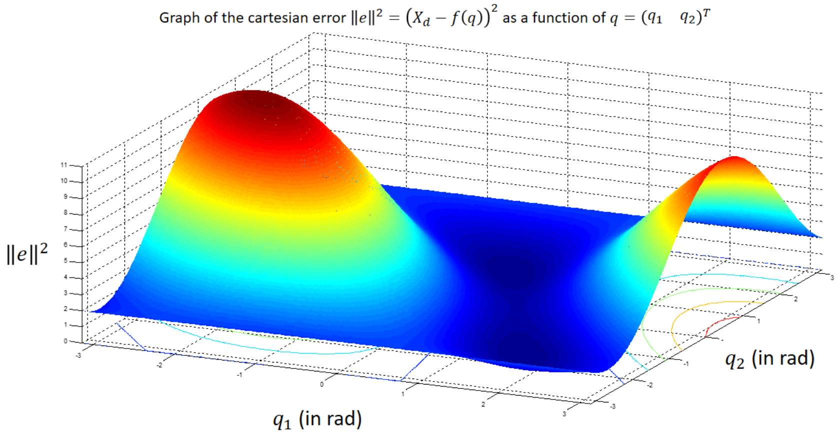
  - simpler on the computational point (transpose of the Jacobian matrix, and not its inverse) ;
  - direct usefulness for the case of task-redundant robot ;
  - searching for the amplification gain step α to guaranty decreasing of the error function H at each iteration (*linesearch* technique) ;
  - linear convergence rate ;
  - algorithm revisited as a feedback scheme ($\alpha = 1$).



Demonstration for asymptotic stability of the algorithm
Demonstration

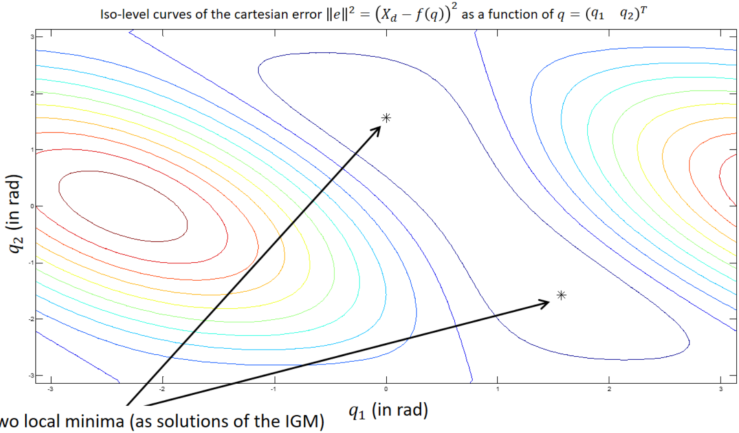
Algorithms for numerical computation of IGM

Case study of robot 2R when considering $X_d = (1, 1)$ and $l_1 = l_2 = 1$.



Algorithms for numerical computation of IGM

Case study of robot 2R when considering $X_d = (1,1)$ and $l_1 = l_2 = 1$.



Pair of solutions coming from the explicit formulation : $\left(q_1', q_2'\right) = \left(0, \frac{\pi}{2}\right)$ and $\left(q_1'', q_2''\right) = \left(\frac{\pi}{2}, -\frac{\pi}{2}\right)$

Algorithms for numerical computation of IGM

- Iterative optimisation procedure

$k \leftarrow 0$
while $\left\| J^t\left(q_k\right) \right\| > \epsilon$ **do**
  $k \leftarrow k+1$
  $\triangleright$ *Case of the Gradient-based method*
  $q_k \leftarrow q_{k-1} + \alpha J^t\left(q_{k-1}\right)\left[X_d - f(q_{k-1})\right]$
  $\triangleright$ *Case of the Newton-Raphson-based method*
  $q_k \leftarrow q_{k-1} + J^{-1}\left(q_{k-1}\right)\left[X_d - f(q_{k-1})\right]$
end while
$q^* \leftarrow q_{k+1}$
return (q^*)

Algorithms for numerical computation of IGM

- Iterative optimisation procedure

  $k \leftarrow 0$
  while $\left\| J^t\left(q_k\right) \right\| > \epsilon$ **do**
  $\quad k \leftarrow k+1$
  $\quad$ ▷ *Case of the Gradient-based method*
  $\quad q_k \leftarrow q_{k-1} + \alpha J^t\left(q_{k-1}\right)\left[X_d - f(q_{k-1})\right]$
  $\quad$ ▷ *Case of the Newton-Raphson-based method*
  $\quad q_k \leftarrow q_{k-1} + J^{-1}\left(q_{k-1}\right)\left[X_d - f(q_{k-1})\right]$
  end while
  $q^* \leftarrow q_{k+1}$
  return (q^*)

- Difficulties coming from these methods
  - Lack of convergence when the error $e = X_d - f(q_{k-1})$ is in the null of J^t or in the singular configuration cases ;
  - Multiple initialisations with different q_0 to avoid local minima ;
  - Search for an adaptive step to choose the best α at each iteration ;
  - Consideration of joint limits only when the algorithm is ended.

Algorithms for numerical computation of IGM

- Iterative optimisation procedure

```
k ← 0
while ‖J^t (q_k)‖ > ε do
    k ← k + 1
                ▷ Case of the Gradient-based method
    q_k ← q_{k−1} + α J^t (q_{k−1}) [X_d − f(q_{k−1})]
    ▷ Case of the Newton-Raphson-based method
    q_k ← q_{k−1} + J^{−1} (q_{k−1}) [X_d − f(q_{k−1})]
end while
q* ← q_{k+1}
return (q*)
```

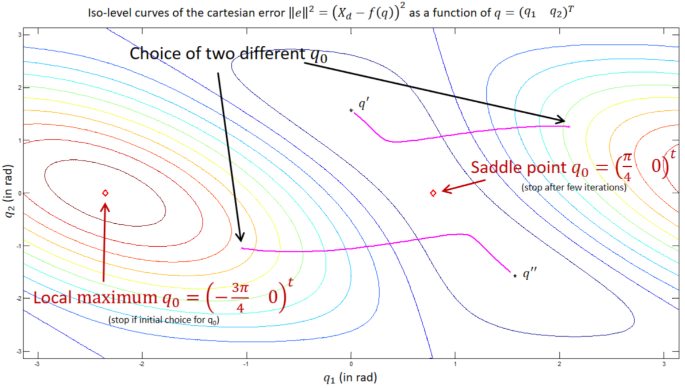
- Difficulties coming from these methods
  - Lack of convergence when the error $e = X_d - f(q_{k-1})$ is in the null of J^t or in the singular configuration cases ;
  - Multiple initialisations with different q_0 to avoid local minima ;
  - Search for an adaptive step to choose the best α at each iteration ;
  - Consideration of joint limits only when the algorithm is ended.

- Remarks
  - Possibilities to combine Gradient-based method for the first iterations (guaranteed convergence but with low convergence rate) and the *Newton-Raphson*-based method for the last iterations (quadratic convergence rate) ;
  - Other possible stop criteria ;
    - cartesian error $\|X_d - f(q_k)\| < \epsilon_x$
    - joint error $\|q_k - q_{k-1}\| < \epsilon_q$

Algorithms for numerical computation of IGM

Case study of robot 2R when considering $X_d = (1, 1)$ and $l_1 = l_2 = 1$.

- Analysis of the algorithm convergence



- Case of no convergence when the Jacobian matrix $J(q)$ becomes singular in q and the error e belongs to the null of $J^t(q)$.

Example

Inversion of the kinematic model

Usefulness of the inverse kinematic model

1. Search for a joint velocity $\dot{q}$ enabling to achieve the cartesian velocity $\dot{X}$ according to $\dot{X} = J(q)\dot{q}$: search for inversion technique of $J \in \mathbb{R}^{m \times n}$ in the general case.
2. Inverse differential kinematic model also used for kinematic control along a continuous time scaling end-effector trajectory $X_d(t)$ in task space :
   - Tracking the trajectory $X_d\,(t)$ of the robot end-effector consisting in providing to the robot controller a succession of values X_{d_k} (index k corresponding to the sampling time $t_k = kT_e$) ;
   - Coordinates transformation needed to provide to the axes controllers the series of reference values q_{d_k} corresponding to X_{d_k} ;
     - execution of the previous iterative algorithm at each sample $t_0, \ldots, t_k, \ldots, t_f$: $q \leftarrow q + J^{-1}(q)\left(X_{d_k} - f\,(q)\right)$ (in general, 1 or 2 iterations being sufficient) ;
     - "reasonable" choice from q_{0_k} at t_k being the solution to the previous problem at t_{k-1} ;
   - Eventual problems requiring the search for robust inversion techniques :
     - crossing a singular configuration (case where $J(q)$ non-invertible) ;
     - redundant or under-determined robots (case where $J(q)$ not square).

Inversion of the kinematic model in the redundant case ($M < N$)

- Infinity of solutions to the inverse kinematic problem, from which one can choose the one that is *the closest possible* of a particular or preferred configuration $\dot{q}_0$
- Use of this redundancy for :
  - determining a motion out of singularity to avoid the previous case ;
  - avoiding obstacles or increasing dexterity.
- Inverse kinematics as a convex quadratic optimization problem with equality constraint :

$$\min_{\dot{q}} \frac{1}{2} (\dot{q} - \dot{q}_0)^t \, W \, (\dot{q} - \dot{q}_0) \, ,$$
$$\text{s.c.} \quad J(q) \, \dot{q} = \dot{X}_d$$

  - Positive-definite matrix $W \in \mathbb{R}^{n \times n}$ used for norm weighting purpose $\|\dot{q} - \dot{q}_0\|_W$:
    - to give *more or less importance* to the position or the orientation ;
    - to optimize the involved kinetic energy during the motion : $\frac{1}{2} \dot{q}^t A(q) \, \dot{q}$, etc.
  - Let note that in this problem the kinematic constraints are totally satisfied, which means that the desired pose in the task space is strictly reached (provided that $\dot{X}_d \in \mathscr{R}(J)$).

Inversion of the kinematic model in the redundant case ($M < N$)

Solution of the quadratic optimisation

The general solution to the problem of inverse kinematics in the redundant case is written as follows

$$\dot{q}^* = J_W^{\sharp} \dot{X}_d + \left(I_n - J_W^{\sharp} J \right) \dot{q}_0$$

- Particular solution to the problem $J_W^{\sharp} \dot{X}_d$, that of minimum weighted norm (for $\dot{q}_0 = 0$), with :

$$J_W^{\sharp} = W^{-1} J^t \left(J W^{-1} J^t \right)^{-1}$$

- Set of homogeneous solutions ($J\dot{q} = 0$) being determined through the orthogonal projection of $\dot{q}_0$ in the null of J given by $\left(I_n - J_W^{\sharp} J \right)$

Demonstration

Inversion of the kinematic model in the redundant case $(M < N)$

Particular case where $W = I_n$:

- General solution of the non-weighted norm problem :

$$\dot{q}^* = \underbrace{J^\sharp \dot{X}_d}_{\substack{\text{Particular solution} \\ \text{(here the pseudo-inverse)}}} + \underbrace{\left(I_n - J^\sharp J\right)\dot{q}_0}_{\substack{\text{Orthogonal projection} \\ \text{of } \dot{q}_0 \text{ on } \mathcal{N}(J)}}$$

- right pseudo-inverse matrix defined by $J^\sharp = J^t\left(JJ^t\right)^{-1}$:
  - *Moore-Penrose* inverse of $\dot{q}^* = J^\sharp \dot{X}_d$ (*pinv()* function of *Matlab*^{TM}) defined as the unique solution of

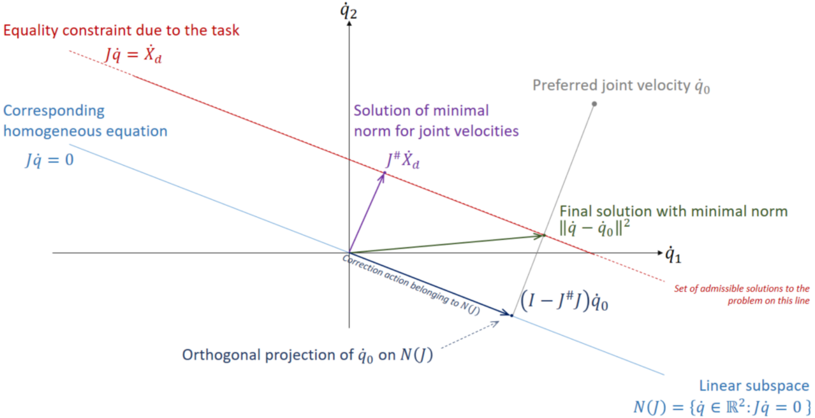
$$\min_{\dot{q}} \frac{1}{2}\|\dot{q}\|^2$$

    for desired $\dot{X}_d$ (if $Rank\,(J) = M$, i.e. full row rank).
  - $J^\sharp$ said *pseudo-inverse* of J in the sense that it is not a real inverse (indeed, $JJ^\sharp = I_m$ - *right pseudo-inverse* - but $J^\sharp J \neq I_n$) ;
- Properties of the projector $N_J = \left(I_n - J^\sharp J\right)$:
  1. Symmetry : $\left(I_n - J^\sharp J\right)^t = \left(I_n - J^\sharp J\right)$;
  2. Idempotent : $\left(I_n - J^\sharp J\right)^2 = \left(I_n - J^\sharp J\right)$;
  3. Orthogonality between $J^\sharp \dot{X}_d$ and $\left(I_n - J^\sharp J\right)\dot{q}_0$.
  4. Invariance through pseudo-inversion : $\left(I_n - J^\sharp J\right)^\sharp = \left(I_n - J^\sharp J\right)$.

Inversion of the kinematic model in the redundant case $(M < N)$

Graphical illustration of joint velocity for the case where $N = 2$ and $M = 1$ in a given configuration $\overline{q} = (\overline{q}_1, \overline{q}_2)$:

$$J(\overline{q})\,\dot{q} = \dot{X}_d \Leftrightarrow \begin{bmatrix} J_1 & J_2 \end{bmatrix} \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \end{pmatrix} = \dot{X}_d$$



Inversion of the kinematic model in the redundant case ($M < N$)

Secondary task in joint space

- Term $\left(I_n - J^\sharp J\right)\dot{q}_0$ belonging to the null space of J :
  - No influence on the value of $\dot{X}_d$;
  - Physical description of some internal motions of the robot ;
  - Use for satisfying additional optimisation constraints by projecting a secondary task on the null space of the Jacobian matrix.
- Search for preferred configurations using the *projected gradient* technique
  - Choice of a differentiable, scalar and positive-definite objective function H :

$$\dot{q}_0 = -\alpha \nabla_q H(q) = -\alpha \begin{pmatrix} \frac{\partial H}{\partial q_1} \\ \vdots \\ \frac{\partial H}{\partial q_n} \end{pmatrix}$$

    where the term $\alpha > 0$ denotes the tradeoff between the minimisation objectives of $\frac{1}{2}\|\dot{q}\|^2$ and $H(q)$.
  - Decrease of the values taken by $H(q)$ at each iteration during the execution of the task $\dot{X}_d(t)$.

Inversion of the kinematic model in the redundant case $(M < N)$

Some usual cases for the choice of $H(q)$:

- Avoiding joint limits $(q_i \in [q_{min}, q_{max}])$ [7] :

$$H_{lim.}(q) = \sum_{i=1}^{n} \left(\frac{q_i - \overline{q}_i}{q_{max} - q_{min}} \right)^2 \quad where \quad \overline{q}_i = \frac{q_{max} - q_{min}}{2}$$

- Increasing manipulability (recall the velocity ellipsoids) :

$$H_{man.}(q) = \sqrt{\det\left(J(q)\, J^t(q)\right)}$$

(maximisation of the distance to singularities)

- Avoiding or searching for particular joint configurations through some force fields deriving from attractive or repulsive potential functions [15]
- Avoiding obstacles through the maximisation of the minimal caretsian distance of the robot to obstacles :

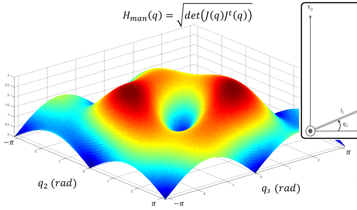
$$H_{obs.}(q) = \min_{\substack{\text{a : robot} \\ \text{b : obstacles}}} \|a(q) - b\|_2$$

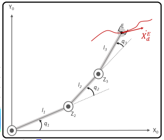
(difficulties arising from the potential non-differentiability of the function)

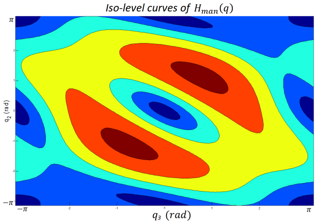
Inversion of the kinematic model in the redundant case ($M < N$)

Study of the manipulability : positioning of the end-effector E with the planar robot RRR (body lengths chosen to be unitary)

$$H_{man.}(q) = \sqrt{\det\left(J(q)\,J^t(q)\right)}$$



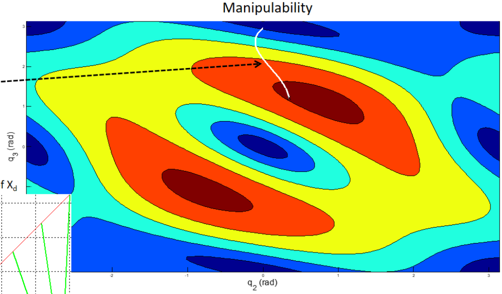




- Redundancy of order 1 ($M = 2$ and $N = 3$) ;
- Potential function independent of q_1 ;
- Minima of H_{man} for q_2 and q_3 belonging to $\{-\pi; 0; \pi\}$.

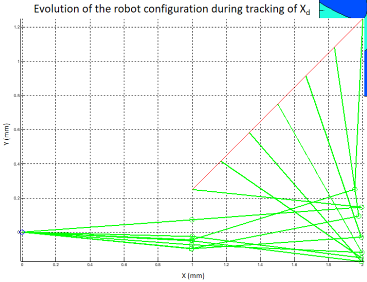
Inversion of the kinematic model in the redundant case ($M < N$)

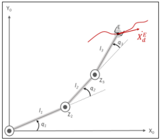
Study of the manipulability : positionning of the end-effector E with the planar robot RRR (body lengths chosen to be unitary)



Followed path in joint space to maximize manipulability

Direction of gradient $\nabla H_{man.}(q)$





Planar RRR robot with tracking of the desired cartesian velocity

$${}^{0}\dot{X}_{d}^{E} = [1 \quad 1]^{t}$$

Inversion of the kinematic model in the redundant case ($M < N$)

Secondary task in the operational space

Taking into account priorities through projections in the subspaces of the successive null spaces of the different tasks :

- Description of the tasks to be realized according to their order of priority :
  - First-priority subtask $\dot{X}_{d_1}$ corresponding to $J_1(q)$:

$$\dot{X}_{d_1} = J_1(q)\dot{q}$$

  - Second-priority subtask $\dot{X}_{d_2}$ corresponding to $J_2(q)$:

$$\dot{X}_{d_2} = J_2(q)\dot{q}$$

- General formulation of the solution :

$$\boxed{\dot{q}^* = J_1^\sharp \dot{X}_{d_1} + (J_2 N_{J_1})^\sharp \left(\dot{X}_{d_2} - J_2 J_1^\sharp \dot{X}_{d_1}\right)}$$

where we use the contracted notation N_{J_1} to denote the projector in the null space of J_1 : $N_{J_1} = \left(I - J_1^\sharp J_1\right)$.

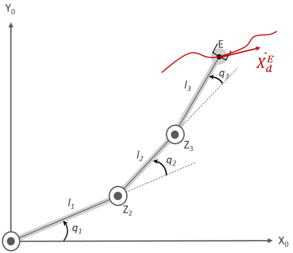
Demonstration

Inversion of the kinematic model in the redundant case $(M < N)$

Secondary task in the operational space

Case of the planar RRR robot with specification of the linear velocity of the end-effector $(M = 2)$:

Coming back to example 10 :



(unitary body lengths)

1. Given a desired velocity of the end-effector ${}^0\dot{X}_d^E(t) = (1, 1)^t$, searching for the joint velocity $\dot{q}_1^*(t)$ (numerical evaluation when the robot is in configuration $\overline{q} = \left(\left(\frac{\pi}{3}, \frac{2\pi}{3}, \frac{\pi}{3}\right)^t\right).$;
2. Searching for $\dot{q}_2^*(t)$ respecting the previous objective while adding $\dot{q}_1$ and $\dot{q}_2$ to be null (evaluation in $\overline{q}$).

Example

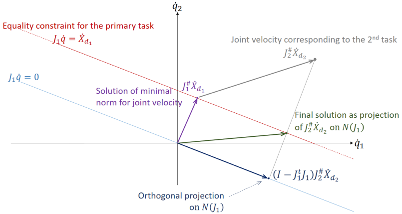
Inversion of the kinematic model in the redundant case ($M < N$)

Secondary task in the operational space

The previous method may lead to an algorithmic singularity when the null spaces of J_1 and J_2 are neighbours (inducing too high joint velocities).

- To avoid algorithmic singularities, an alternative formulation leads to determine the solution for the secondary task and then to project it on the null space of J_1 :

$$\dot{q}^* = \underbrace{J_1^\sharp \dot{X}_{d_1}}_{\text{Primary task}} + N_{J_1} \underbrace{\left(J_2^\sharp \dot{X}_{d_2} \right)}_{\text{Secondary task}}$$



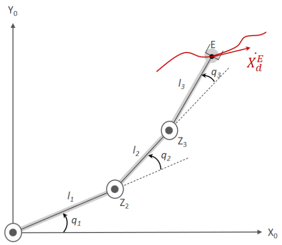
- For highly redundant system, possibility to put in series several tasks according to :

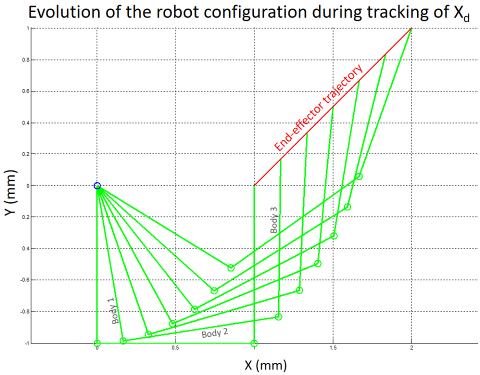
$$\dot{q}^* = J_1^\sharp \dot{X}_{d_1} + N_{J_1} \left(J_2^\sharp \dot{X}_{d_2} + (N_{J_1} \cap N_{J_2}) \left(J_3^\sharp \dot{X}_{d_3} + \dots \right) \right)$$

the vector $J_3^\sharp \dot{X}_{d_3}$ being projected on the intersection of null spaces of both J_1 and J_2 so as to avoid perturbing both tasks.

Inversion of the kinematic model in the redundant case ($M < N$)

Study for avoiding obstacles : trajectory tracking of the end-effector with imposed linear velocity $\dot{X}_d^E(t) = (1, 1)^t$ using a planar RRR robot





Inversion of the kinematic model in the redundant case $(M < N)$

Study for avoiding obstacles : trajectory tracking of the end-effector with imposed linear velocity $\dot{X}_d^E(t) = (1,1)^t$ using a planar RRR robot

- Searching for $\dot{q}_2$ for avoiding ponctual obstacles, as solution of

$$\min_{\dot{q}_2} \left\| \dot{X}_2(t) - J_2 \dot{q}_2 \right\|_2$$

where

  - $\dot{X}_2(t)$ follows the repulsive law defined by :

$$\dot{X}_2(t) = \alpha(t)\, v_0 n_0(t)$$

  with :

    - the repulsion coefficient α given by :

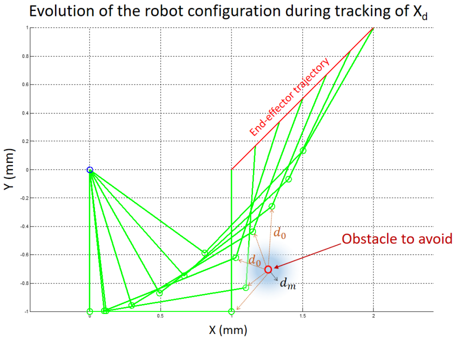
$$\alpha(t) = \begin{cases} \left(\frac{d_m}{\|d_0\|}\right)^2 - 1 & \text{if} \quad \|d_0\| < d_m \\ 0 & \text{if} \quad \|d_0\| \geq d_m \end{cases}$$

    - v_0 a nominal velocity arbitrary chosen ;
    - d_0 denotes the distance between the obstacle point and the extremity of the 2^{nd} body according to the unitary vector n_0 ;
    - d_m the distance corresponding to the influence radius ;
  - J_2 the Jacobian matrix associated to the extremity of the 2^{nd} segment.
- Final joint configuration as follows :

$$\dot{q}^* = J^\sharp \dot{X}_d^E(t) + \left(I_n - J^\sharp J\right) J_2^\sharp \dot{X}_2$$

Inversion of the kinematic model in the redundant case ($M < N$)

Study for avoiding obstacles : trajectory tracking of the end-effector with imposed linear velocity $\dot{X}_d^E(t) = (1,1)^t$ using a planar RRR robot



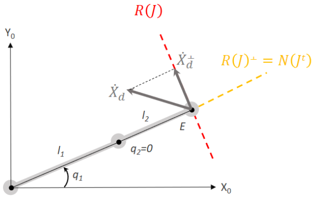
Inversion of the kinematic model in the singular case ($M = N$)

Pseudo-inverse method : $\dot{q}^* = J^\sharp \dot{X}_d$

- Singular configuration q limiting the capacity to generate arbitrary motions, and formalized by a loss of rank for $J(q)$:

$$Rank(J) = r < M$$

- Several possible cases in a singular configuration :
  1. If $\dot{X}_d \in \mathscr{R}(J)$ exclusively, then the constraint $J(q)\,\dot{q} = \dot{X}_d$ is satisfied (the velocity is achievable even if the inverse J^{-1} does not exist) ;
  2. Otherwise, the constraint $J(q)\,\dot{q} = \dot{X}_d$ is not fulfilled : $J\dot{q}^* = \dot{X}_d^\perp$ where $\dot{X}_d^\perp$ is the orthogonal projection of $\dot{X}_d$ on $Im(J)$ (subspace of achievable velocities), so that the error $\left\| J(q)\,\dot{q}^* - \dot{X}_d \right\|$ is minimum.
     - Let note that, in the particular case where $\dot{X}_d \in Im(J)^\perp$ exclusively (i.e. $\dot{X}_d \in N(J^t)$), then the solution according to the pseudo-inverse method returns $\dot{q} = 0$ (unachievable velocity, since $\dot{X}_d^\perp = 0$).



$\dot{q} = J^\# \dot{X}_d$
Joint velocity vector of minimal norm generating $\dot{X}_d^\perp$

Inversion of the kinematic model in the singular case ($M = N$)

Pseudo-inverse method : $\dot{q}^* = J^\sharp \dot{X}_d$

- To sum-up, insights in the returned *pseudo-inverse* solution for kinematic inverse problem
  - Regular case : *exact and unique solution out of singular configurations* (since $J^\sharp = J^{-1}$ in this case), but solution not acceptable in the neighbourhood of singular configurations (use of the inverse kinematic model with a bad conditioning of J which can lead to high joint velocities, incompatible with the actuators capabilities) ;
  - Singular case : approximated solution *on a singular configuration* allowing nevertheless to calculate the joint velocity $\dot{q}$ with minimal norm, while minimizing the error $\left\| J(q)\,\dot{q}^* - \dot{X}_d \right\|$ (constraint fully satisfied if $\dot{X}_d \in \mathscr{R}(J)$).
- Discontinuous behavior when passing from *regular* to *singular* case :

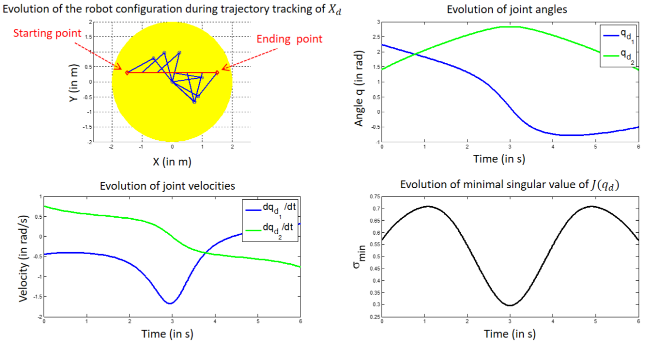
$$\dot{q}^* = \sum_{i=1}^{m} \frac{1}{\sigma_i} V_i U_i^t \dot{X}_d$$

When coming close to a singularity $\sigma_{min} \to 0$ ($\|\dot{q}\|$ high), then $\sigma_{min} = 0$ (sum being stopped at $m-1$) involving a discontinuity for crossing the singularity.

Inversion of the kinematic model in the regular case ($M = N$)

Case study of robot 2R

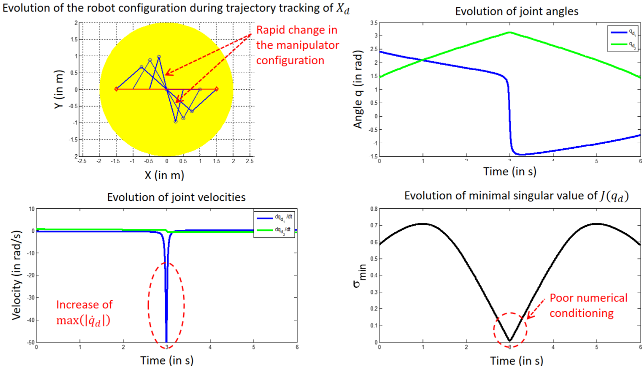
1. Behaviour analysis of $\dot{q} = J^{-1}(q)\dot{X}_d$ in the regular case out of singularities
   - Tracking a straight path $X_d(t)$ at constant speed $v = 0.5 m.s^{-1}$ during $T = 6$s.



Inversion of the kinematic model in the quasi-singular case $(M = N)$

Case study of robot 2R

2. Behaviour analysis of $\dot{q} = J^{-1}(q)\dot{X}_d$ in the regular case but close to singularity
   - new trajectory reference $X_d(t)$ close to singular case $(\min_{t_k} \{\sigma_{min}(J(q_d))\} \approx 0)$;
   - increase of $max\, |\dot{q}_i|$ in the neighbourhood of singular configurations.



Inversion of the kinematic model in the singular case $(M = N)$

Inversion using the damped least-squares method
To increase the excessive joint velocities close to singularities, we can think about tolerating an *error on the trajectory tracking*, replacing the inversion problem $\dot{x}_d = J(q_d)\,\dot{q}$ by the following minimisation problem [29] :

$$\min_{\dot{q}} \left\{ \frac{1}{2} \left\| J(q)\,\dot{q} - \dot{X}_d \right\|^2 + \frac{\lambda^2}{2} \|\dot{q}\|^2 \right\}, \quad \lambda \geq 0.$$

- Inverse kinematic seen as an optimisation problem :
  - 1^{st} term of the objective function representative of the norm of the trajectory error ;
  - 2^{nd} term of the objective function representative of the norm of the joint velocity ;
- Role of the coefficient λ :
  - weighting coefficient named as *damping ratio* ;
  - choice of $\lambda = 0$ when far away from singular configurations, then $\lambda > 0$ when $\sigma_{min}(J(q))$ close to 0) ;
  - when $\lambda > 0$, decrease (*damping*) of the amplitude of the joint velocity $max\,|\dot{q}_i|$ obtained to the detrimental to velocity trajectory error $\epsilon_{\dot{X}} = \lambda^2 \left(\lambda^2 I_m + JJ^t\right)^{-1}$

Inversion of the kinematic model in the singular case ($M = N$)

Inversion using the damped least-squares method

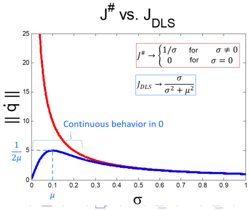
- Solution to the inversion problem using the damped least-squares method :

$$\boxed{\dot{q}^* = \left(\lambda^2 I_n + J^t J\right)^{-1} J^t \dot{X}_d = \underbrace{J^t\left(\lambda^2 I_m + JJ^t\right)^{-1}}_{J_{DLS}(q)} \dot{X}_d}$$

- Possible use of the Jacobian matrix J_{DLS} both for the case $m = n$ as well as for the redundant case $m < n$ (in this case, we will prefer the expression $J^t\left(\lambda^2 I_m + JJ^t\right)^{-1}$ for computing J_{DLS}).
- Rewriting J_{DLS} from its decomposition in singular values

$$\boxed{\dot{q}^* = \sum_{i=1}^{m} \frac{\sigma_i}{\sigma_i^2 + \lambda^2} V_i U_i^t \dot{X}_d}$$

- damping ratio limiting the maximal amplitude in the detrimental to the precision when $\sigma_i << \lambda$;
- damping ratio with few impact when $\sigma_i >> \lambda$.

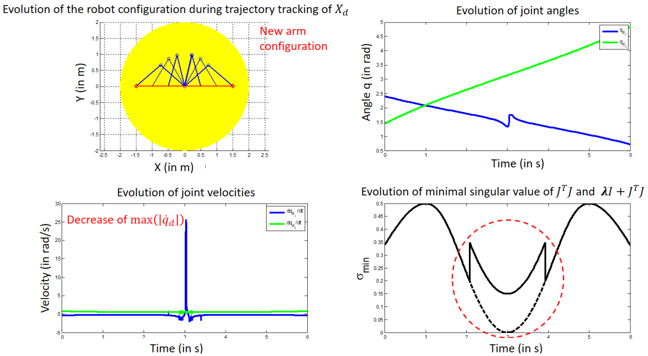


Demonstration

Inversion of the kinematic model in the quasi-singular case ($M = N$)

Case study of robot 2R

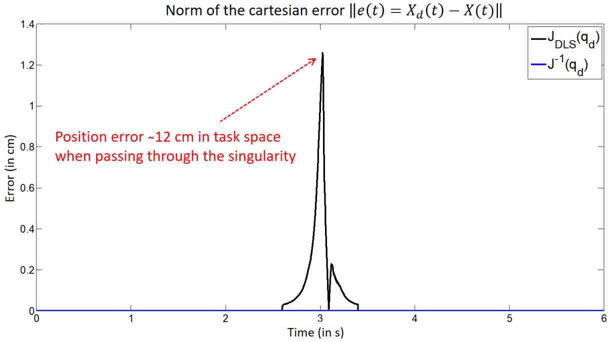
2. Behaviour analysis of $\dot{q} = J_{DLS}(q)\dot{X}_d$ in the regular case but close to singularity
   - Physically, high-dynamic reconfiguration of the arm avoided ;
   - Use of the damped pseudo-inverse matrix when $\sigma_{min}(J)$ becomes close to 0 ;
   - Damping velocities $\dot{q}_i$.



Inversion of the kinematic model in the quasi-singular case ($M = N$)

Case study of robot 2R

- Comparing the behaviour of $\dot{q} = J_{DLS}(q)\dot{X}_d$ and $\dot{q} = J^{-1}(q)\dot{X}_d$ on the cartesian error in the case close to singularity.



Inversion of the kinematic model in the over-determined case ($M > N$)

Constrained system : no exact solution to the inverse kinematic problem (system said *incompatible*)

- Situation corresponding to the case where the desired velocity cannot be exactly obtained ;
- Search for an approximated solution that minimize the Euclidian norm of the cartesian error :

$$\min_{\dot{q}} \frac{1}{2} \left\| J(q)\,\dot{q} - \dot{X}_d \right\|$$

where, if $Rank(J) = N$ (full column rank), the solution, said *the least-squares solution*, equals $\dot{q}^* = J^\sharp(q)\,\dot{x}_d$ with

$$J^\sharp(q) = \left(J^t J\right)^{-1} J^t$$

- Property of left-inversion : $J(q)^\sharp J = I_n$ (but $JJ(q)^\sharp \neq I_m$) ;

Inversion of kinematic model

Sum-up about *Moore-Penrose* pseudo-inverse Jacobian matrix :

1. If $Rank\,(J) = r = M < N$ (full row rank), then $J^{\sharp} = J^t \left(JJ^t\right)^{-1}$ (right pseudo-inverse - redundant case) ;
2. If $Rank\,(J) = r = M = N$, then $J^{\sharp} = J^{-1}$ (regular case) ;
3. If $Rank\,(J) = r = N < M$ (full column rank), then $J^{\sharp} = \left(J^t J\right)^{-1} J^t$ (left pseudo-inverse - over-determined case).

Properties of the pseudo-inverse matrix (valid irrespective of the conditions on the rank of J) :

- $JJ^{\sharp}J = J$, $J^{\sharp}JJ^{\sharp} = J^{\sharp}$, $\left(JJ^{\sharp}\right)^t = JJ^{\sharp}$ and $\left(J^{\sharp}J\right)^t = J^{\sharp}J$ (Hermitian matrices) ;
- Algorithmically, the pseudo-inverse always exists and is obtained from the decomposition into singular values of $J = U\Sigma V^t$:

$$J^{\sharp} = V\Sigma^{+}U^t \Rightarrow \dot{q} = \sum_{i=1}^{r} \frac{1}{\sigma_i} V_i U_i^t \dot{X}_d$$

where $\Sigma^{\sharp}$, pseudo-inverse of the diagonal matrix Σ, is a diagonal matrix whose non-zero elements are obtained by inverting the non-zero elements (of the diagonal) of Σ :

$$\Sigma^{+} = \left[\begin{array}{ccc} \frac{1}{\sigma_1} & & \\ & \ddots & \\ & & \frac{1}{\sigma_r} \quad 0 \\ \hline \multicolumn{3}{c}{0_{(N-M)\times(N)}} \end{array}\right]$$

Introduction

- Study of the dynamic relations describing the motions of an articulated structure made up of rigid bodies $\mathscr{C}_i$ connected to each other ;
- Considerations of two complementary points of view :

1. *Direct* dynamic model
   Expression of joint accelerations from the joint efforts, velocities and positions that are assumed to be known

$$\ddot{q} = n\left(\Gamma, \dot{q}, q\right)$$

   then, by successive integrations,

$$\dot{q} = \int \ddot{q} dt \quad \text{et} \quad q = \int \dot{q} dt$$

   - Usefulness for the simulation of the robot behaviour ;
   - Computation from the *Newton-Euler* equations or by inversion of the *inverse* dynamic model.

2. *Inverse* dynamic model
   Expression of joint efforts from the accelerations, velocities and joint positions that are assumed to be known

$$\Gamma = h\left(\ddot{q}, \dot{q}, q\right)$$

   - Usefulness for dimensioning the actuators and for synthesizing control laws ;
   - Computation from the *Newton-Euler* or *Lagrange* equations.

Lagrangian mechanics applied to manipulators

- Replacing the *vectorial* quantities by *scalar* values :
  - kinetic energy E_c in the case of n rigid bodies $\mathscr{C}_i$ connected to each other :

$$E_c = \sum_{i=1}^{n} E_{c_i}$$

  - potential energy E_p in the case of n rigid bodies $\mathscr{C}_i$ connected to each other :

$$E_p = \sum_{i=1}^{n} E_{p_i}$$

  - Work of external mechanical forces applied to the system.
- System considered *globally* rather than considering *separately* each component of the mechanism (offering the advantage of automatically eliminating the interaction forces associated with kinematic constraints).

Lagrangian mechanics applied to manipulators

- *Lagrange* equations of a N-dof manipulator described by n generalized coordinates q_i :

$$\psi_i = \frac{d}{dt}\left(\frac{\partial \mathscr{L}}{\partial \dot{q}_i}\right) - \frac{\partial \mathscr{L}}{\partial q_i} \quad \text{for} \quad i = 1, \ldots, n$$

where $\mathscr{L}$ defines the *lagrangian* of the system :

$$\mathscr{L} = E_c - E_p$$

- Recall that ψ_i denotes the non-conservative generalized forces (external or dissipative) producing a mechanical work on q_i.
In robotics, we consider :
    - τ_i : i^{th} joint torque coming from the mechanical work provided b ythe actuators ;
    - $\left[J(q)^t F_c\right]_i$: term coming from the external contact forces with the environment ;
    - $-\tau_{f_i}(\dot{q}_i)$: joint friction torque.

Finally,

$$\psi_i = \tau_i + \left[J(q)^t F_c\right]_i - \tau_{f_i}(\dot{q}_i)$$

Inetial properties of the rigid body $\mathscr{C}_i$

Let consider the rigid body $\mathscr{C}_i$ of volume $\mathscr{V}_i$ whose center of mass G_i is chosen to be the origin of the frame attached to the body $\mathscr{R}_{G_i} = (G_i, x_i, y_i, z_i)$.

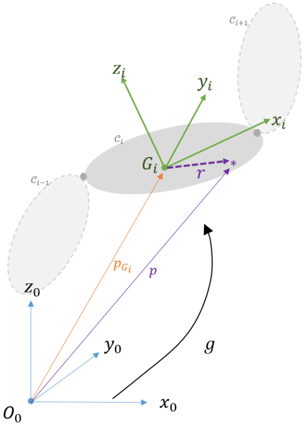
Mass properties of the body :

- $r \in \mathbb{R}^3$ defining the coordinates of a material point p attached to $\mathscr{V}_i$ relatively to $\mathscr{R}_{G_i}$ $(r = p - p_{G_i})$;
- Mass defined by :

$$m_i = \int_{\mathscr{V}_i} \rho(x, y, z)\, d\mathscr{V}_i$$

  where the body is supposed to be made of a homogeneous material $(\rho(x, y, z) = \rho)$;
- Center of mass of the body $\mathscr{C}_i$ defined by the weighting mean :

$$p_{G_i} = \frac{1}{m_i} \int_{\mathscr{V}_i} \rho(x, y, z)\, p(x, y, z)\, d\mathscr{V}_i.$$



Inetial properties of the rigid body $\mathscr{C}_i$

Let consider the rigid body $\mathscr{C}_i$ of volume $\mathscr{V}_i$ whose center of mass G_i is chosen to be the origin of the frame attached to the body $\mathscr{R}_{G_i} = (G_i, x_i, y_i, z_i)$.

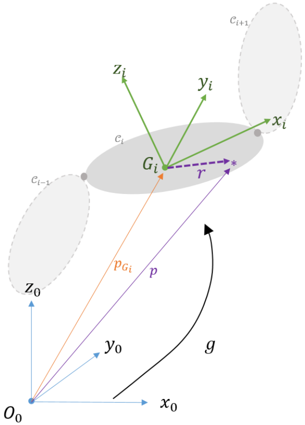
Kinetics of material points :

- Transformation of body $\mathscr{C}_i$ relatively to the inertial frame $\mathscr{R}_0$ given by $g = \left(p_{G_i}, R\right) \in SE(3)$;
- Velocity of a generic material point attached to $\mathscr{C}_i$ relatively to the inertial frame :

$$\dot{p} = \dot{p}_{G_i} + \dot{R}r$$

- Kinetic energy of the body evaluated by intergation of the kinetic energy ol all the material points defining the volume :

$$E_{c_i} = \frac{1}{2}\int_{\mathscr{V}_i} \rho(x, y, z) \left\| \dot{p}_{G_i} + \dot{R}r \right\|^2 d\mathscr{V}_i.$$

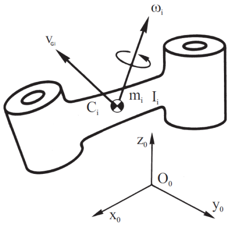


Kinetic energy of the rigid body $\mathscr{C}_i$

Konig theorem

The kinetic energy of a rigid body $\mathscr{C}_i$ of center of mass G_i and mass m_i in motion w.r.t. a Galilean frame $\mathscr{R}_0$ is computed as follows :

$$E_{c_i} = \frac{1}{2} m_i V_{G_i}^t V_{G_i} + \frac{1}{2} \omega_i^t I_i \omega_i$$



Demonstration

Kinetic energy of the rigid body $\mathscr{C}_i$

Kinetic energy $E_{c_i} = \frac{1}{2} m_i V_{G_i}^t V_{G_i} + \frac{1}{2} \omega_i^t I_i \omega_i$ seen as the sum of :

- the *translational kinetic energy* of the body assuming that the mass is concentrated at the center of mass G_i (equivalence to the material point) ;
- the *rotational kinetic energy* of the body, where $I_i \in \mathbb{R}^{3\times3}$ is the skew-symmetric matrix, called *inertial tensor* of the object, that is constant in the referential $\mathscr{R}_{G_i}$ attached to the center of mass of $\mathscr{C}_i$:

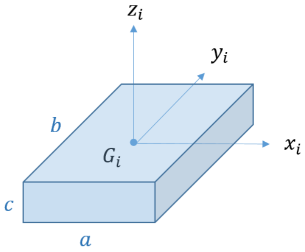
$$I_i = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} = \int_{\mathscr{V}_i} \rho \hat{r}^2 d\mathscr{V}_i$$

Remark :
Each of the two terms of kinetic energy E_{c_i} is invariant w.r.t. the choice of the reference frame used to express the velocities (but ω_i and I_i must be expressed in the same reference frame, preferably $\mathscr{R}_{G_i}$).

Kinetic energy of the rigid body $\mathscr{C}_i$

Inertia tensor : Computation of the inertia tensor evaluated at the center of mass :

$$I_i = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} = \begin{bmatrix} \int_{\mathscr{V}_i} \left(y^2 + z^2\right) dm_i & -\int_{\mathscr{V}_i} xydm_i & -\int_{\mathscr{V}_i} xzdm_i \\ & \int_{\mathscr{V}_i} \left(x^2 + z^2\right) dm_i & -\int_{\mathscr{V}_i} yzdm_i \\ (sym.) & & \int_{\mathscr{V}_i} \left(x^2 + y^2\right) dm_i \end{bmatrix}$$



Demonstration

Kinetic energy of the rigid body $\mathscr{C}_i$

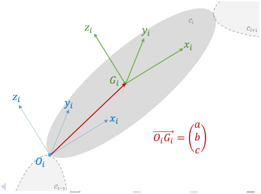
Inertia tensor :

Generalized *Huygens* theorem

For the body $\mathscr{C}_i$ of mass m_i, the inertia tensor defined at the origin O_i of the frame $\mathscr{R}_i = (O_i, x_i, y_i, z_i)$ is expressed as a function of the inertia tensor evaluated at its center of mass G_i in the frame $\mathscr{R}_{G_i} = (G_i, x_i, y_i, z_i)$ according to the following *transport formula.* :

$$I_{O_i} = I_{G_i} + m_i \begin{bmatrix} b^2+c^2 & -ab & -ac \\ (\text{sym.}) & a^2+c^2 & -bc \\ (\text{sym.}) & (\text{sym.}) & a^2+b^2 \end{bmatrix} = \begin{bmatrix} XX_i & XY_i & XZ_i \\ (\text{sym.}) & YY_i & YZ_i \\ (\text{sym.}) & (\text{sym.}) & ZZ_i \end{bmatrix}$$

$$I_{O_i} = \begin{bmatrix} I_{xx_i} & I_{xy_i} & I_{xz_i} \\ (\text{sym.}) & I_{yy_i} & I_{yz_i} \\ (\text{sym.}) & (\text{sym.}) & I_{zz_i} \end{bmatrix} + m_i \begin{bmatrix} b^2+c^2 & -ab & -ac \\ (\text{sym.}) & a^2+c^2 & -bc \\ (\text{sym.}) & (\text{sym.}) & a^2+b^2 \end{bmatrix} = \begin{bmatrix} XX_i & XY_i & XZ_i \\ (\text{sym.}) & YY_i & YZ_i \\ (\text{sym.}) & (\text{sym.}) & ZZ_i \end{bmatrix}$$

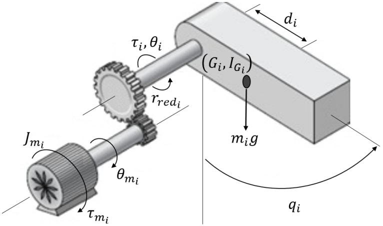


Kinetic energy of the actuators

Motor-to-joint mechanical transmission of the corresponding rigid body $\mathscr{C}_i$

- Torque provided by the motor τ_{m_i} ;
- Joint torque τ_i, equivalent to the motor torque after the reduction stage : $\tau_i = r_{red_i}\tau_{m_i}$;
- Angular motor position θ_{m_i} ;
- Angular motor position after the reduction stage θ_i
  - $\theta_i = \frac{1}{r_{red_i}}\theta_{m_i}$
  - θ_i equal to the joint angle in the case of the mechanically rigid transmission, i.e. $\theta_i = q_i$;

Study of one revolute robotic axis :



- Taking into account the moment of inertia J_{m_i} of the actuator rotor

$$I_{m_i} = r_{red_i}^2 J_{m_i}$$

with I_{m_i} the load-side inertia of the motor.

- Rotational kinetic energy of the actuator rotor :

$$E_{cm_i} = \frac{1}{2} I_{m_i}\dot{\theta}_i^2$$

Example computation of the kinetic energy of the robotic axis

Kinetic energy of the rigid body $\mathscr{C}_i$

Dependence of E_{c_i} to q and $\dot{q}$

- Computation of V_{G_i} from the Jacobian matrix providing the translation velocity of the center of mass G_i of body $\mathscr{C}_i$ as a function of joint velocities $\dot{q}$:

$$V_{G_i} = J_{v_{G_i}}(q)\,\dot{q}$$

where the Jacobian matrix $J_{v_{G_i}}$ takes the following form :

$$J_{v_{G_i}} = \begin{bmatrix} J^1_{v_{G_i}}(q) & \dots & J^i_{v_{G_i}}(q) & 0 & \dots & 0 \end{bmatrix}$$

- Computation of ω_i from the Jacobian matrix providing the rotational velocity of body $\mathscr{C}$ as a function of joint velocities $\dot{q}$:

$$\omega_i = J_{\omega_i}(q)\,\dot{q}$$

where the Jacobian matrix J_{ω_i} takes the following form :

$$J_{\omega_i} = \begin{bmatrix} J^1_{\omega_i}(q) & \dots & J^i_{\omega_i}(q) & 0 & \dots & 0 \end{bmatrix}$$

This Jacobian must be expressed in the same frame as the corresponding inertia tensor matrix :

  - Inertia tensor I_i being constant if it is given in the frame $\mathscr{R}_{G_i}$;
  - Otherwise, ${}^0I_i = R_{0i}\,{}^iI_i R^t_{0i}$ in the reference base frame $\mathscr{R}_0$ for example (frame generally used to express Jacobian matrices).

Remark : The approach to obtain the matrices $J_{v_{G_i}}$ and J_{ω_i} is identical to the one seen previously, except that they express the velocity of the center of mass of the body $\mathscr{C}$ and no longer the velocity of the origin of the robot's terminal frame.
Useful use of the *Varignon* formula : $V(G_i) = V(O_i) + \omega_i \times O_iG_i$.

Kinetic energy of a poly-articulated chain

- Kinetic energy of a poly-articulated chain made of N bodies $\mathscr{C}_i$:

$$E_c(q,\dot{q}) = \sum_{i=1}^{N} E_{c_i}(q,\dot{q}) = \frac{1}{2}\dot{q}^t \underbrace{\sum_{i=1}^{N}\left(m_i {}^0J_{v_{G_i}}^t(q)\,{}^0J_{v_{G_i}}(q) + {}^0J_{\omega_i}^t(q)\,{}^0I_i\,{}^0J_{\omega_i}(q)\right)}_{A(q)}\dot{q}$$

- Inertia matrix of the manipulator $A(q) \in \mathbb{R}^{n\times n}$

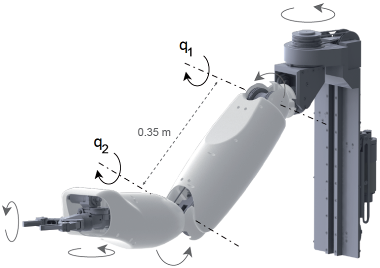
$$A(q) = \left(a_{ij}(q)\right)_{1\leq i,j\leq n}$$

  - Matrix dependent to the configuration q of the robotic arm ;
  - Symmetric matrix ;
  - Definite-positive matrix (thus always invertible regardless of joint configuration) ;
- Adding the influence of the actuators
  - Hypothesis :
    1. Infinitely rigid mechanical transmissions (no flexible phenomena) ;
    2. Centre of mass of the rotor located on its axis of rotation ;
    3. Gyroscopic effects of the rotor neglected (only the moment of inertia of the own rotation is considered) ;
    4. No couplings in mechanical transmissions (actuator τ_{m_i} driving only q_i).
  - Addition of the contributions of the body-side actuators inertia on the diagonal of $A(q)$ (augmentation of the term $a_{ii}(q)$ with $r_{red_i}^2 J_{m_i}$).

Kinetic energy of a poly-articulated chain

Case study of a planar 2-axes robot under the influence of gravity

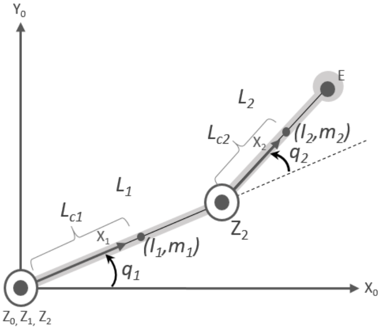
- CEA lightweight robot prototype [18] :
  - Anthropomorph arm (kinematic architecture of 7 dofs) ;
  - Mass of the robotic arm of about 15 kg for a payload of 3 kg ;
  - Actuation using DC-motors.



Kinetic energy of a poly-articulated chain

Case study of a planar 2-axes robot under the influence of gravity

- Restriction of the robot to its sagittal motion :
  - Main segments of "bras" and "forearm" ;
  - 2-dof mecanism : q_1 (shoulder) and q_2 (elbow) ;
  - Restricted motion in the vertical plane under the influence of gravity.



- Inertia tensor I_i of body i at its center of mass ;
- Mass m_i of segment i ;
- Length of segment i parameterized by L_i ;
- Position of center of mass L_{c_i} along the segment i ;
- Reduction ratio r_{red_i} of joint i ;
- Inertia moment J_{m_i} of motor actuating joint i.

Example computation of the robot kinetic energy.

Potential energy of a polyarticulated chain

General formulation for the potential energy

The potential energy due to the action of gravitational forces on the rigid body $\mathscr{C}_i$ of center of mass G_i and mass m_i in motion w.r.t. a Galilean frame $\mathscr{R}_0$ is given by :

$$E_{p_i}(q) = -\int_{\mathscr{V}_i} g^t p dm_i = -g^t \int_{\mathscr{V}_i} \rho p d\mathscr{V}_i = -m_i g^{t\,0} p_{G_i}(q)$$

where g is the gravity acceleration vector in $\mathscr{R}_0$ and ${}^0p_{G_i}$ is the position vector of the center of mass G_i in $\mathscr{R}_0$, obtained by the homogeneous transformations according to :

$$\begin{bmatrix} {}^0p_{G_i} \\ 1 \end{bmatrix} = \overline{g}_{01}(q_1)\ldots\overline{g}_{(i-1)i}(q_i)\begin{bmatrix} {}^ip_{G_i} \\ 1 \end{bmatrix}$$

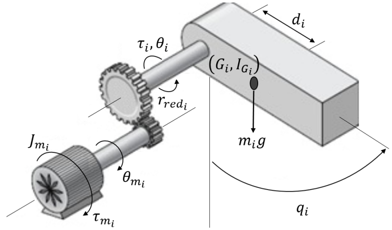
The potential energy of gravity of a manipulator made of N rigid bodies is obtained by summing the contribution of potential energies of all the bodies :

$$E_p(q) = \sum_{i=1}^{N} E_{p_i}(q) = -g^t\left(\sum_{i=1}^{N} m_i {}^0p_{G_i}(q)\right) \quad \text{with} \quad g = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}.$$

This is equivalent to assuming all the mass of $\mathscr{C}_i$ concentrated at the center of mass (equivalent to a material point) and calculating the component of G_i in the opposite direction to the gravity vector g in the Galilean coordinate system.

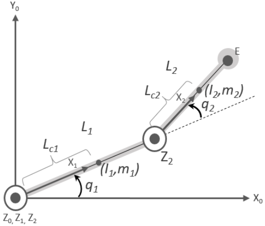
Potential energy of a polyarticulated chain

Study of one revolute robotic axis



Example Computation of the potential energy.

Case study of the planar 2-dof robot



Example computation of the potential energy

Dynamics of motion for a serial manipulator

Expanded form of the *inverse* dynamic model

Analytical formulation deduced from the *Lagrange* equations :

$$\sum_{j=1}^{N} a_{kj}(q)\,\ddot{q}_j + \sum_{j=1}^{N}\sum_{i=1}^{N} c_{ijk}(q)\,\dot{q}_i\dot{q}_j + g_k(q) = \psi_k \quad \text{for} \quad k = 1, \ldots, n$$

- a_{kj} term of the inertia matrix $A(q)$ defined by :

$$\frac{\partial E_c(q,\dot{q})}{\partial \dot{q}_k} = \sum_j a_{kj}(q)\,\dot{q}_j$$

- c_{ijk} *Christoffel* symbol, such that :

$$c_{ijk}(q) = \frac{1}{2}\left(\frac{\partial a_{kj}}{\partial q_i} + \frac{\partial a_{ki}}{\partial q_j} - \frac{\partial a_{ij}}{\partial q_k}\right)$$

- g_k gravitational term defined by $g_k(q) = \frac{\partial E_p(q)}{\partial q_k}$ according to :

$$G(q) = \begin{bmatrix} g_1(q) & \ldots & g_N(q) \end{bmatrix}^t = \frac{\partial E_p(q)}{\partial q} = -\sum_{i=1}^{N} m_i J_{v_{G_i}}^t \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}$$

Demonstration

Dynamics of motion for a serial manipulator

Elements a_{kj}, c_{ijk} and g_k :

- dependency to the joint position uniquely ;
- computation relatively simple once the configuration of the manipulator is known.

Physical insigths :

- Acceleration terms
  - a_{kk} : *inertia moment* around the k-th joint axis, for a given configuration and considering all the other joints blocked in position ($a_{kk}(q) > 0$) ;
  - a_{kj} : *inertial coupling*, taking into account the effects of joint acceleration j on joint k ;
- Quadratic velocity terms
  - The terms grouped in $c_{iik}\dot{q}_i^2$ correspond to the *centrifugal terms* induced on the joint k by the velocity of the joint j (note that $c_{kkk} = \frac{1}{2}\frac{\partial a_{kk}}{\partial q_k} = 0$) ;
  - The terms grouped in $c_{ijk}\dot{q}_j q_i$ represent the *Coriolis terms* induced on the k-th joint by the velocity of the i-th and j-th joints. ;
  - Properties of symmetry for *Christoffel* symbols : $c_{ijk} = c_{jik}$.
- Terms depending of the configuration
  - g_k epresents the torques generated on the k-th joint by the force of gravity acting on the manipulator in its current configuration.

Dynamics of motion for a serial manipulator

Matricial form of the *inverse* dynamic model

Analytical formulation deduced from the *Lagrange* equations :

$$A(q)\,\ddot{q} + C(q,\dot{q})\,\dot{q} + G(q) + \Gamma_f(\dot{q}) = \Gamma + J(q)^t\,F_c$$

- $A(q) \in \mathbb{R}^{N\times N}$ symmetric and definite-positive inertia matrix ;
- $C(q,\dot{q})\,\dot{q} \in \mathbb{R}^N$ vector of *Coriolis* and centrifugal forces, where the elements of the matrix $C(q,\dot{q})$ are calulated as follows :

$$[C(q,\dot{q})]_{k,j} = \sum_{i=1}^{N} c_{ijk}(q)\,\dot{q}_i;$$

- $G(q) \in \mathbb{R}^N$ vector of joint torques due to gravity effects :

$$G(q) = \begin{bmatrix} g_1(q) & \dots & g_N(q) \end{bmatrix}^t;$$

- $\Gamma_f(\dot{q}) = \begin{bmatrix} \tau_{f_1} & \dots & \tau_{f_n} \end{bmatrix}^t \in \mathbb{R}^N$ vector of joint friction effects ;
- $\Gamma = \begin{bmatrix} \tau_1 & \dots & \tau_n \end{bmatrix}^t \in \mathbb{R}^N$ vector of the actuators torques.

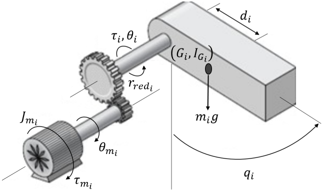
- Generally, we denote $H(q,\dot{q})$ the quantity $C(q,\dot{q})\,\dot{q} + G(q) + \Gamma_f(\dot{q})$.

Dynamics of motion for a serial manipulator

Equations of the dynamic motion

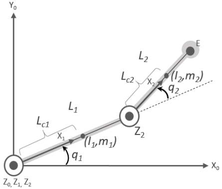
Analytical formulation deduced from the *Lagrange* equations :

Study of one revolute robotic axis



Exemple Dynamics of the motion.

Study of the planar RR robot



Exemple Dynamics of the motion.

Properties of the dynamic model

Properties of the matrix $A(q)$ [11]

1. $A(q) \in \mathbb{R}^{N \times N}$ symmetric, definite-positive and dependent to the joint configuration q of the manipulator ;
2. $A(q)$ is upper and lower bounded :

$$\mu_1 \mathbb{I} \preceq A(q) \preceq \mu_2 \mathbb{I}$$

   i.e.

$$x^t (A(q) - \mu_1 \mathbb{I}) x \geq 0 \quad et \quad x^t (\mu_2 \mathbb{I} - A(q)) x \geq 0;$$

3. Thus, $A^{-1}(q)$ is definite-positive and bounded :

$$\frac{1}{\mu_2} \mathbb{I} \preceq A^{-1}(q) \preceq \frac{1}{\mu_1} \mathbb{I};$$

4. In the case of revolute joints, μ_1 and μ_2 are constant (no dependence on joint configuration q, since the components of $A(q)$ are functions of $cos(q)$ and $sin(q)$) ;
5. In the case of prismatic joints, μ_1 and μ_2 are functions linearly dependent of q ;
6. Since $A(q)$ is bounded,

$$\alpha_1 \leq \|A(q)\|_X \leq \alpha_2$$

   for certain norms ($X = 1, 2, p, \infty$).

Properties of the dynamic model

Properties of the vector $c(q,\dot{q}) = C(q,\dot{q})\,\dot{q}$ [11]

1. For a given manipulator, the matrix $C(q,\dot{q})$ is not unique, but the vector $c(q,\dot{q})$ is unique ;
2. $C(q,\dot{q})\,\dot{q}$ is a quadratic function of joint velocity $\dot{q}$;
3. The k-th generic component of vector $c(q,\dot{q}) = C(q,\dot{q})\,\dot{q}$ takes the form :
$$C(q,\dot{q}_a)\,\dot{q}_b = \dot{q}_a^t S_k(q)\,\dot{q}_b$$
where $S_k(q) = \frac{1}{2}\left(\frac{\partial a_k}{\partial q} + \left(\frac{\partial a_k}{\partial q}\right)^t - \frac{\partial A(q)}{\partial q_k}\right)$ is a symmetric matrix of size $N \times N$ for which a_k is the k-th column of $A(q)$;
4. It results that :
$$\|C(q,\dot{q})\,\dot{q}\| \leq \beta\,\|\dot{q}\|^2 \,;$$
5. β is constant in the case of revolute joints, and linearly dependent of q in the case of prismatic joints ;
6. the matrix defined by
$$\dot{A}(q) - 2C(q,\dot{q})$$
where $C(q,\dot{q})$ is computed from the *Christoffel* symbols is skew-symmtric. It results that :
$$x^t\left(\dot{A}(q) - 2C(q,\dot{q})\right)x = 0.$$

Demonstration

Properties of the dynamic model

Properties of vector $G(q)$ [11]

1. the term of joint torques induced by the gravity $G(q)$ is bounded :

$$\|G(q)\| \leq g_b;$$

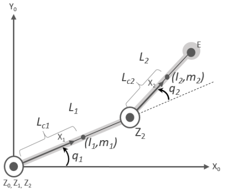
2. g_b is constant in the case of revolute joints, and linearly dependent of q in the case of prismatic joints ;
3. In the case of revolute joints, the vectorial function $G(q)$ is *Lipschitzian* :

$$\forall q_b, q_a \in \mathbb{R}^n, \|G(q_b) - G(q_a)\| \leq k_g \|q_b - q_a\|;$$

Study of planar RR robot $\left(-\frac{\pi}{2} \leq q_1, q_2 \leq \frac{\pi}{2}\right)$:

- Search for bounds α_1 and α_2 of the inertia matrix $A(q)$ for the 1-norm ;
- Search for bound β of vector $c(q, \dot{q})$ for the 1-norm ;
- Verify that $\dot{A}(q) - 2C(q, \dot{q})$ is indeed skew-symmetric ;
- Search for bound g_b of $G(q)$ for the 1-norm.

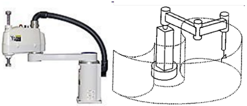
Example



Analysis of the gravity torques effects

- Static balancing of the masses involved ; ;
- Mechanical compensation mechanism ;
- Absence of gravitational terms :
  - planar robots with horizontal motions ($E_p = C^{ste}$) ;
  - spatial applications.

(a) Weight compensation structure



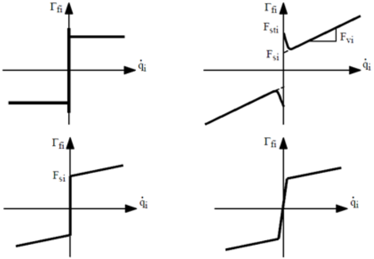
(b) Robot SCARA

(c) Long-range articulated robot

Friction phenomena

Dissipative phenomena due to friction Γ_f at the joint-, transmissions- and/or motors-levels

- Complexity of representation [12] : viscous, dry, *Coulomb*, *Stribeck*...



- Usual representation of *Coulomb*-like friction in robotics :
  - Consideration of the i-th joint :

$$\tau_{f_i}(\dot{q}_i) = sign(\dot{q}_i)\, F_{s_i} + diag(\dot{q}_i)\, F_{v_i}$$

  - Joint friction torques :

$$\Gamma_f(\dot{q}) = diag(sign(\dot{q}))\, F_s + diag(\dot{q})\, F_v$$

where $\Gamma_f = \left[\tau_{f_1} \ldots \tau_{f_N}\right]^t$.

Direct dynamic model

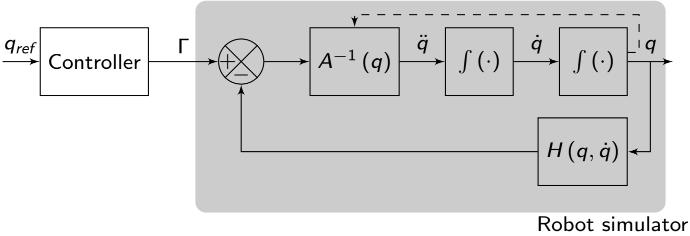
Formulation of joint accelerations from the efforts, velocities and positions that are assumed to be known

$$\ddot{q} = n\left(\Gamma, \dot{q}, q\right)$$

then, by successive integrations, $\dot{q} = \int \ddot{q}dt$ and $q = \int \dot{q}dt$.

- Usefulness for the simulation of the robot behaviour ;
- Computation from the *Newton-Euler* equations or by inversion of the *inverse* dynamic model :

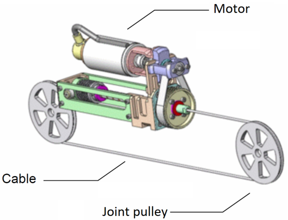
$$\ddot{q} = A\left(q\right)^{-1}\left(\Gamma - C\left(q, \dot{q}\right)\dot{q} - G\left(q\right) - \Gamma_f\right) = A\left(q\right)^{-1}\left(\Gamma - H\left(q, \dot{q}\right)\right)$$



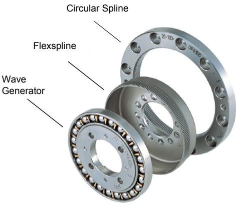
Joint elasticity

Origins of joint compliance :

- Industrial robotics : cable-based transmissions, no infinitely rigid gears ratio, heavy load support, . . .



(d) Actuator with cable-based transmission (CEA)



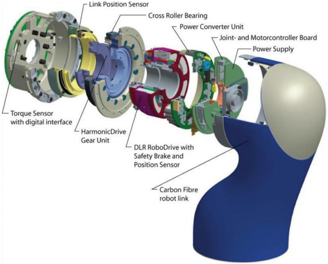
(e) Harmonic Drive gears®

- Robotic dedicated to physical Human-Robot interaction : *compliance* in the transmission introduced for safety (remote actuartors thanks to cables and pulleys, design of lightweight structures, soft elastic actuators,. . .)

Flexibilities between the actuators (θ) and the to-be-controlled outputs (q) assumed to be concentrated at the joint-level under the hypothesis of small deformations

Example of flexible-joint robots

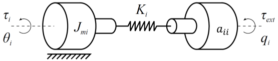
(f) Prototype of a CEA lightweight robot amr using cable-based transmissions



(g) LWR III DLR robot using Harmonic Drive® components

Taking into account joint flexibilities in the robot dynamics

Simplified model of a flexible joint



Modification of the energy-based description of rigid systems

- Adding the alstic potential energy E_{p_e} to the gravity potential energy E_{p_g} according to $E_p = E_{p_g} + E_{p_e}$:

$$E_{p_e} = \sum_{i=1}^{N} E_{p_{e_i}} = \sum_{i=1}^{N} \frac{1}{2} k_i \left(q_i - \theta_i\right)^2 = \frac{1}{2} \left(q - \theta\right)^t K \left(q - \theta\right)$$

  with K the diagonal definite-positive matrix of elements k_i that defines the joint stiffness.

- Adding the kinetic energy of motors E_{c_m} by dissociating it from that of the rigid bodies E_{c_c} according to $E_c = E_{c_c} + E_{c_m}$:

$$E_{c_m} = \sum_{i=1}^{N} E_{c_{m_i}} = \sum_{i=1}^{n} \frac{1}{2} I_{m_i} \dot{\theta}_i^2 = \frac{1}{2} \dot{\theta}^t I_m \dot{\theta}$$

  with I_m the diagonal definite-positive matrix of the actuators inertia expressed at the joint side.

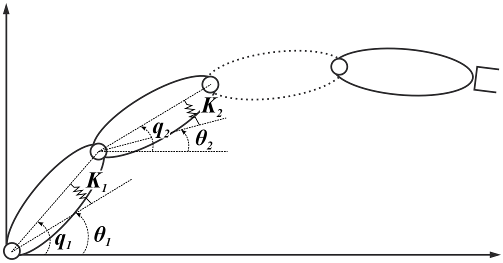
Taking into account joint flexibility in the robot dynamics

Inverse dynamic model of flexible-joint robots

- $2n$ generalized coordinates
  - q vector of the n joint angles
  - θ vector of the n motor angles expressed at the joint side according to $\theta = R_{red}^{-1}\theta_m$ where

$$R_{red} = diag\left(r_{red_1}, \ldots, r_{red_N}\right)$$

denotes the reduction ratio matrix ($\Gamma_m R_{red}^t = \Gamma$).



Taking into account joint flexibility in the robot dynamics

Inverse dynamic model of flexible-joint robots

- Computation from the *Lagrangian* function :

$$
\begin{aligned}
\mathscr{L} &= E_c - E_p \\
&= \left(E_{c_c} + E_{c_m}\right) - \left(E_{p_g} + E_{p_e}\right) \\
&= \left(\frac{1}{2}\dot{q}^t A\left(q\right)\dot{q} + \frac{1}{2}\dot{\theta}^t I_m \dot{\theta}\right) - \left(g^t \left(\sum_{i=1}^{N} -m_i{}^0 p_{G_i}\left(q\right)\right) + \frac{1}{2}\left(q-\theta\right)^t K\left(q-\theta\right)\right)
\end{aligned}
$$

thus,

$$
\begin{aligned}
A\left(q\right)\ddot{q} + C\left(q,\dot{q}\right)\dot{q} + G\left(q\right) + \Gamma_{f_c} + K\left(q-\theta\right) &= 0 \\
I_m\ddot{\theta} + \Gamma_{f_m} - K\left(q-\theta\right) &= \Gamma
\end{aligned}
$$

  - $2n$ second-order differential equations that are coupled through the elastic torque $K\left(q-\theta\right)$;
  - n first equations giving the joint dynamics and the n last equations for the motor dynamics;
- Dissociation of frictional effects (Γ_{f_m} at the motor-level and Γ_{f_c} at the joint-level).

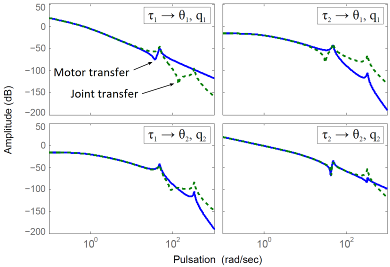
Dynamic effects of an elastic motor-to-joint coupling

Example of a *simplified* RR planar robot [17]

- Approximation with local validity :
  - Neglecting the *Coriolis* and centrifugal effects ;
  - Inertia matrix assumed to be constant ;

$$A\ddot{q} + F_v\dot{q} + K(q-\theta) = 0$$
$$J_m\ddot{\theta} + F_{v_m}\dot{\theta} - K(q-\theta) = \tau$$

- Frequency responses of the simplified model :
  - Influence due to the inertial coupling of non-null extra-diagonal terms of A ;
  - Alternating anti-resonance/resonance pattern for the frequency response of τ/θ ;
  - Coupling of the resonants modes due to joint flexibility.



Inverse dynamic model in the task space

Dynamics of the end-effector [16]

Let $\ddot{X} = \left[\dot{V}, \dot{\omega}\right] \in \mathbb{R}^6$ be the end-effector acceleration and $\mathscr{F} \in \mathbb{R}^6$ the cartesian force induced by the joint torques at the end-effector level $\Gamma = J(q)^T \mathscr{F}$. In the non-singular case, the dynamics of the end-effector motion is as follows :

$$\Lambda(q)\ddot{X} + \mu(q, \dot{q}) + p(q) = \mathscr{F}$$

with :

- $\Lambda(q)$ the inertia seen from the end-effector level :

$$\Lambda(q) = \left(J(q) A^{-1}(q) J^t(q)\right)^{-1}$$

- $\mu(q, \dot{q})$ the *Coriolis*/centrifugal effects in the cartesian space :

$$\mu(q, \dot{q}) = \Lambda(q) J(q) A^{-1}(q) C(q, \dot{q}) \dot{q} - \Lambda \dot{J}(q) \dot{q}$$

- $p(q)$ the gravity effects in the cartesian space :

$$p(q) = \Lambda(q) J(q) A^{-1}(q) G(q)$$

Demonstration

Usefulness of the dynamic model identification

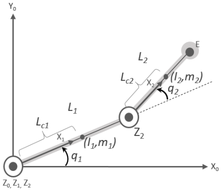
Need for a knowledge of the numerical values of the dynamic parameters of the robot

- Simulation of the dynamic model ;
- Model-based control.

Presence in the dynamic model of a set of minimal dynamic parameters

- Dynamic model fully parameterized by these terms ;
- Combination of terms characteristic of each body and actuator.

Coming back on the planar RR robot



Minimal dynamic parameters :

- 5 unknown body parameters : $I_{zz_1} + m_2 L_1^2 + J_{m_1}$, I_{zz_2}, MZ_2, $MZ_1 + m_2 L_1$ et J_{m_2}
  ($MZ_1 = m_1 L_{c_1}$ et $MZ_2 = m_2 L_{c_2}$ are the first moments of inertia of the bodies 1 and 2 around z_0) ;
- 4 additional terms related to joint friction effects : F_{s_1}, F_{v_1}, F_{s_2} et F_{v_2}.

In total, 9 minimum dynamic parameters are sufficient to set up the dynamic model of the robot.

Usefulness of the dynamic model identification

Set of minimum dynamic parameters

- Importance of a minimum number of parameters for experimental identification and simplicity of the embedded dynamic model in adaptive control strategies ;
- Combination of terms relating to. . .
  - inertial effects of each body defined by its inertia tensor given in its frame $\mathscr{R}_i = (O_i, x_i, y_i, z_i)$ using the *Huygens transport formula* ;
  - mass effect through its mass m_i ;
  - the effect due to the first moment of inertia of the body $\mathscr{C}_i$ around the origin of the $\mathscr{R}_i$ frame defined by :

$$^{i}mS_i = [MX_i, MY_i, MZ_i]^t$$

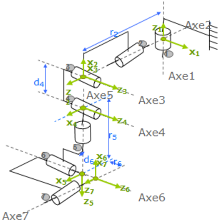
    where S_i is the vector having for origin O_i and for extremity the center of mass G_i of the body $\mathscr{C}_i$, equal to O_iG_i ;
  - the inertial effects of each actuator J_{m_i} ;
  - the frictional effect F_{v_i} and F_{s_i}.
- Set of unknowns to be determined for the body $\mathscr{C}_i$ of the poly-articulated chain :

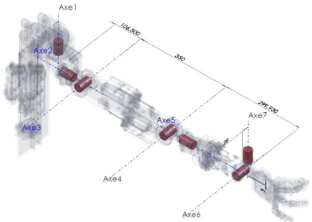
$$\underbrace{XX_i \quad YY_i \quad ZZ_i}_{\text{Inertial moments}} \quad \underbrace{XY_i \quad XZ_i \quad YZ_i}_{\text{Inertial products}} \quad \underbrace{MX_i \quad MY_i \quad MZ_i}_{\text{First inertial moment}}$$

$$\underbrace{M_i}_{\text{Mass}} \quad \underbrace{J_{m_i}}_{\text{Actuator inertial}} \quad \underbrace{F_{v_i} \quad F_{s_i}}_{\textit{Coulomb}\text{ friction}}$$

Methods for estimating the inertial parameters of a robot

- Physical experiments on each of the individual isolated bodies (realized by the manufacturer before assembling the robot)
  - Mass m_i : directly weighted ;
  - Coordinates of the center-of-mass L_{c_i} : determining counterbalanced points of the link ;
  - Principal moments of inertia XX_i, YY_i et ZZ_i : pendular motions method ;
  - Products of intertia : difficult !
- Functionnalities of certain CAD software ;
- Identification using techniques coming from the control field (frequential, temporal responses, . . .).
  - The approach reported in the following is based on the analysis of the "input/output" behavior of the robot on some planned motion and on estimating the parameter values by minimizing the difference between a function of the real robot variables and its mathematical model.





Linearity property of the robot model w.r.t. dyanmic parameters

Rewritting the dynamic model of rigid-link robot

Existence of a linear regression w.r.t. a set of parameters

$$A(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + \Gamma_f(q,\dot{q}) = \boxed{\mathrm{W}(q,\dot{q},\ddot{q})\chi = \Gamma}$$

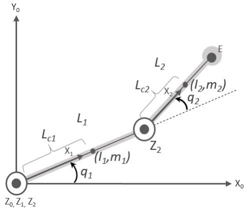
with

- $W(q,\dot{q},\ddot{q})$ observation matrix, or regressor, of dimensions $n \times p$;
- χ vector of dimension $p \times 1$ containing the to-be-identified p parameters involved in the dynamic model.

Case study of the planar RR robot

Format the linear formulation of the dynamic model w.r.t a set of dynamic parameters χ to be determined :

$$W(q,\dot{q},\ddot{q})\chi = \begin{bmatrix}\tau_1\\ \tau_2\end{bmatrix}$$



Example

Exploiting the linear structure of the model w.r.t. parameters

Methodology for identification

1. Use of the linear model w.r.t.the set of unknown minimal parameters.

$$W\left(q,\dot{q},\ddot{q}\right)\chi=\Gamma$$

2. Choosing an exciting motion trajectory w.r.t. parameters χ_i ;
3. Trajectory execution for the position-controlled robot ;
4. Recording of position data q and torque data Γ at each sampling time of the trajectory ;
5. Building a overdetermined linear system by sampling the model at different instants during motion

$$\tilde{W}\left(q,\dot{q},\ddot{q}\right)\chi+\rho=\tilde{\Gamma}$$

   with
   - $\tilde{W}\left(q,\dot{q},\ddot{q}\right)$ regressor matrix of dimension $c\times p$ with $c >> p$;
   - χ vector of dimension $p\times 1$ containing the p unknown dyanmic parameters of the model ;
   - ρ vector of dimension $c\times 1$ of residues or errors.
6. Estimation of the parameters using linear regression techniques

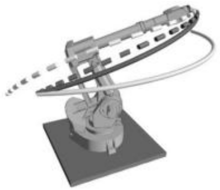
Exploiting the linear structure of the model w.r.t. parameters

Choosing an exciting motion trajectory w.r.t. parameters

- Exploration of all the robot workspace
- Excitation of all the components of the robot dynamic model
- Number of samples sufficiently important to guaranty the condition of full column rank

$$rank\left(\tilde{W}\left(q,\dot{q},\ddot{q}\right)\right) = p$$

- Searching for optimal movement resulting in a numerical conditioning close to one for $\tilde{W}$
- Possibility of identifying the parameters by sub-assembly with simple movements involving only certain joints, the others being blocked (sequential movements).



Exploiting the linear structure of the model w.r.t. parameters

- Building an overdetermined linear system
  - Concatenation in rows of the experimental data

$$c = n \times n_c \updownarrow \quad \begin{bmatrix} W\left(q\left(t_1\right), \dot{q}\left(t_1\right), \ddot{q}\left(t_1\right)\right) \\ \vdots \\ W\left(q\left(t_k\right), \dot{q}\left(t_k\right), \ddot{q}\left(t_k\right)\right) \\ \vdots \\ W\left(q\left(t_{n_c}\right), \dot{q}\left(t_{n_c}\right), \ddot{q}\left(t_{n_c}\right)\right) \end{bmatrix} \chi = \begin{bmatrix} \tau\left(t_1\right) \\ \vdots \\ \tau\left(t_k\right) \\ \vdots \\ \tau\left(t_{n_c}\right) \end{bmatrix} \quad \Leftrightarrow \tilde{W}\left(q, \dot{q}, \ddot{q}\right) \chi = \tilde{\Gamma}$$

- Estimation of the parameters using linear regression techniques
  - Formulation of the optimisation problem :

$$\hat{\chi} = \mathrm{Min}_{\chi} \left\|\rho\right\|_2$$

  - Least-squares solution :

$$\hat{\chi} = \left(\tilde{W}^t \tilde{W}\right)^{-1} \tilde{W}^t \tilde{\Gamma} = \tilde{W}^+ \tilde{\Gamma}$$

with $\tilde{W}$ matrix of full row rank. $\tilde{W}^+$ denotes the pseudo-inverse matrix of $\tilde{W}$.

Demonstration

Exploiting the linear structure of the model w.r.t. parameters

Variance-covariance matrix of the estimation error

Assuming that the vector of errors ρ est zero mean additive independent noise with standard deviation σ_ρ, its variance-covariance matrix is given by

$$C_\rho = E\left[\rho\rho^t\right] = \sigma_\rho^2 \mathbb{I}$$

The variance-covariance matrix of the estimation error is given by

$$C_{\hat{\chi}} = E\left[(\chi - \hat{\chi})(\chi - \hat{\chi})^t\right] = \sigma_\rho^2 \left(\tilde{W}^t \tilde{W}\right)^{-1}$$

▸ Demonstration

Relative standard deviation of the identified parameters

The standard deviation on the j-th parameter is obtained from the diagonal elements of $C_{\hat{\chi}}$, as follows :

$$\sigma_{\hat{\chi}_j\%} = 100 \frac{\sigma_{\hat{\chi}_j}}{|\hat{\chi}_j|} \quad \text{où} \quad \sigma_{\hat{\chi}_j}^2 = C_{\hat{\chi}}(j, j)$$

Practical considerations

Computing the regression matrix $\tilde{W}$

- Evaluation of the joint positions $q\left(t_k\right)$ from the encoder measurements
- Evaluation of the joint velocities $\dot{q}\left(t_k\right)$
  - either by by tachometric measurement
  - or by estimation : low-pass filtering techniques (because of quantization noise) and numerical derivation using a central difference algorithm

$$\dot{q}\left(t_k\right) = \frac{q\left(t_{k+1}\right) - q\left(t_{k-1}\right)}{2T_e}$$

- Evaluation of the joint acceleration $\ddot{q}\left(t_k\right)$ through estimation from velocities signals

Calculation of the joint torques $\tilde{\Gamma}$

- either thanks to the use of joint torque sensors
- or through the estimation from measures/references of motors current

Validation of the identification procedure

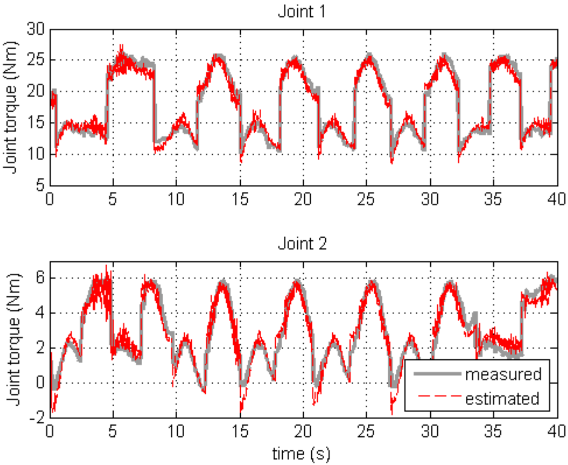
- Choice of the ratio Nber equations/Nber parameters > 500
- Relative standard deviation for each parameter $< 10\%$
- Verification : inertia matrix still definite-positive
- Cross-tests

Experimental study of the 2-dof robot

TABLE – Identified dynamic parameters of the 2-axes rigid robot [19].

Parameters	CAD values	Estimated values	Standard deviations (%)
ZZR_1	1,5321	1,0259	0,3393
MXR_1	1,7347	1,9755	0,0553
$F_{s_{1+}}$	-	1,9570	0,4962
$F_{s_{1-}}$	-	7,4936	0,1295
ZZ_2	0,0884	0,0728	0,9379
MY_2	0,4372	0,3861	0,1092
J_{m2}	0,1228	0,0932	0,7831
$F_{s_{2+}}$	-	1,7525	0,1843
$F_{s_{2-}}$	-	1,2280	0,2643

Experimental study of the 2-dof robot

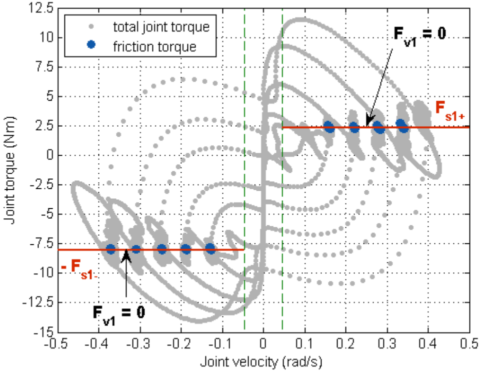


Inverse dynamic model validation - experimental and simulated joint torques for a sinusoidal trajectory [19].

Experimental study of the 2-dof robot

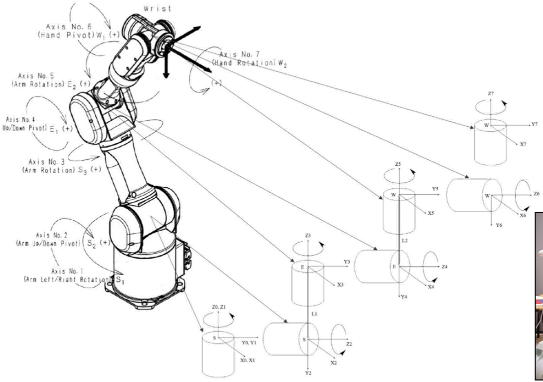
Exciting trajectories for the identification of the frictional torques

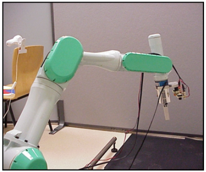
- Velocity-controlled robot ;
- Axis-by-axis movement at constant speed ;
- Elimination of the effect of gravity by choosing an appropriate robot configuration



Frictional torques - experimental data for the axis 1 (references in velocity steps of variable amplitudes). The axis 2 is characterized by similar curves.

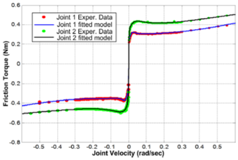
Friction phenomena : the case of the Mitsubishi PA-10 robot [3]



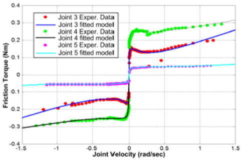


Mitsubishi PA-10 robot with 7 motorized dofs (2 dofs for the shoulder, 2 dofs for the elbow and 3 dofs for the wrist). Aggregation of 21 final identifiable inertial parameters

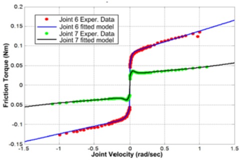
Friction phenomena : the case of the Mitsubishi PA-10 robot [3]



Joints 1, 2: Friction experimental data and fitted curve.



Joints 3, 4, 5: Friction experimental data and fitted curve.



Joint 6, 7: Friction experimental data and fitted curve.

Frictional torques - axis-by-axis experimental data

Search for an augmented *Stribeck* model, as follows :

$$\Gamma_f(\dot{q}) = f_1\dot{q} + f_2 sign(\dot{q}) - f_3 sign(\dot{q})\, e^{-\frac{|\dot{q}|}{f_4}} - f_5 sign(\dot{q})\, e^{-\frac{1}{f_6|\dot{q}|}}$$

- Dealing with asymmetries
- Parameters f_i computed using the nonlinear least-squares method

Extension to the case of flexible-joint robot

Linearity of the dynamic model [8]

In the flexible case, the dynamic model of the robot

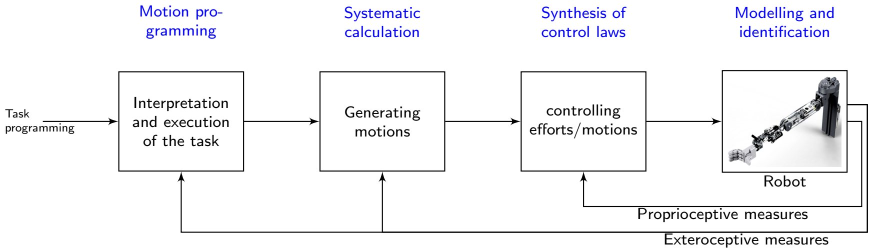
$$\begin{aligned} A(q)\,\ddot{q} + C(q,\dot{q})\,\dot{q} + G(q) + \Gamma_{f_c} + K(q-\theta) &= 0 \\ J_m\ddot{\theta} + \Gamma_{f_m} - K(q-\theta) &= \Gamma \end{aligned}$$

can also be rewritten under a linear form w.r.t. parameters, including here the motor-side θ and joint-side q variables. The dynamic model is rewritten as follows :

$$\underbrace{\begin{pmatrix} 0 \\ \Gamma \end{pmatrix}}_{\Gamma_{flex.}} = \underbrace{\begin{pmatrix} W_{rig.}(q,\dot{q},\ddot{q}) & W_{q-\theta} & 0 & 0 & 0 \\ 0 & -W_{q-\theta} & W_{\ddot{\theta}} & W_{\dot{\theta}} & W_{sign(\dot{\theta})} \end{pmatrix}}_{W_{flex.}(q,\dot{q},\ddot{q},\theta,\dot{\theta},\ddot{\theta})} \underbrace{\begin{pmatrix} \chi_{rig.} \\ \chi_K \\ \chi_{J_m} \\ \chi_{F_{vm}} \\ \chi_{F_{sm}} \end{pmatrix}}_{\chi_{flex.}}$$

noting that $A(q)\ddot{q} + H(q,\dot{q}) = W_{rig.}(q,\dot{q},\ddot{q})\chi_{rig.}$ is the linear regressor corresponding to the rigid model but limited to the joint contribution.

Approch to motion control

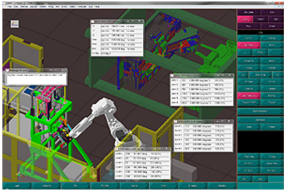


- Task programming :
  - description using a program made of a sequence of elementary instructions ;
- Generating motions :
  - computing the references in position, velocity or acceleration sent to each axis controller to perform the needed trajectories defining the task ;
- Robotic control laws :
  - controlling all the axes with the aim of controlling the joint position $q(t)$ on the previously calculated trajectory $q_d(t)$.

Methods for programming the task motions

Programming the motions of an industrial robot through two main methods :

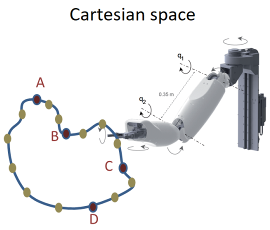
- Learning
  - creation of trajectories by storing the points corresponding to Cartesian coordinates that will determine the robot's position.
  - specification directly on the robot using the teach pendant.
- Off-line programming
  - programming of the task by an operator via an off-line programming software (on a dedicated computer) by importing a CAD model thanks to which he will be able to generate the movements.
  - visualization of the programming result thanks to an integrated simulator which is a virtual representation of the robot's working environment.

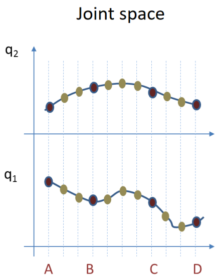


Basics of trajectory generation

Function for calculating the desired trajectories (either in the joint space or in the Cartesian space) describing the equations of motion to be followed from spatial and temporal constraints related to the task :

- Description of the sequence of successive positions to be adopted by the robot and a time history along them ;
- Main movements considered in the following :
  - *Point-to-Point* motion : only the initial and final points on the path and the traveling time are specified ;
  - *Path* motion : generalization to the case when also intermediate points along the path are specified.

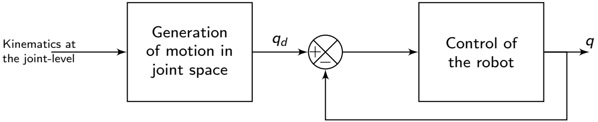




Basics of trajectory generations

Generation of trajectories in the joint space : appropriate for rapid movement in open space

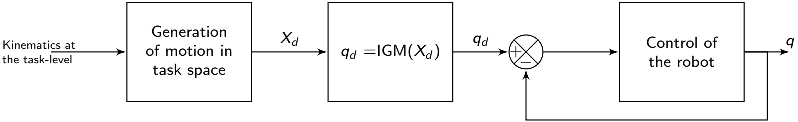
+ Need for less online calculations (no calls to the geometric or inverse kinematic model)
+ Movement unaffected by passing through singular configurations
+ Constraints of velocity and maximum accelerations directly interpretable
– Uncertainty about the tool trajectory in Cartesian space



Basics of trajectory generations

Generation of trajectories in the Cartesian space : appropriate for precise motions in the task space

- \+ Precise control of the tool path in Cartesian space
- – Need to transform each point of the trajectory into joint coordinates
- – Taking into account constraints related to the passage through the singular configurations and the reconfiguration of the robot



Basics of trajectory generations

Notations :

- q_0 vector of joint coordinates corresponding to the initial configuration ;
- q_f vector of joint coordinates corresponding to the final configuration ;
- X_0 vector of Cartesian coordinates corresponding to the initial configuration ;
- X_f vector of Cartesian coordinates corresponding to the final configuration ;
- k_v vector of maximal joint velocities (k_{v_j} is given through the actuators features) ;
- k_a vector of maximal joint accelerations (k_{a_j} being computed from the ratio between the maximal motor torques τ_j and the maximal inertia seen by the joints - sup bound of terms a_{ii} -).

Generalities

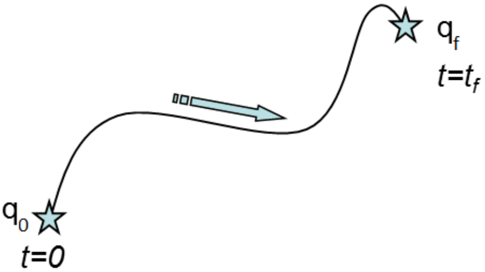
General formulation of the problem

It is a question of setting the motion between the configurations q_0 and q_f as a function of time t for $0 \leq t \leq t_f$ according to

$$q(t) = q_0 + r(t)D$$

where $D = q_f - q_0$.

- Limit values of the interpolation function $r(t)$ ($r(0) = 0$ et $r(t_f) = 1$);
- Joint velocities given by $\dot{q}(t) = \dot{r}(t)D$.



Main methods for generating *Point-to-Point* joint trajectory

Different cases encountered in the generation of *point-to-point* joint trajectory in minimal time :

- Study of the third-order polynomial interpolation :
  - taking into account initial $\dot{q}_0$ and final $\dot{q}_f$ velocities.
- Study of the fifth-order polynomial interpolation :
  - Continuity in position, velocity and acceleration :
  - Motions of class $\mathscr{C}^2$.
- Study of trapeze velocity profile (said *bang-bang*) :
  - Continuous motion in velocity ;
  - Guaranteed minimum time by saturating velocity and acceleration simultaneously.

Main methods for generating *Point-to-Point* joint trajectory

Different cases encountered in the generation of *point-to-point* joint trajectory in minimal time :

- Study of the third-order polynomial interpolation :
  - taking into account initial $\dot{q}_0$ and final $\dot{q}_f$ velocities.
- Study of the fifth-order polynomial interpolation :
  - Continuity in position, velocity and acceleration :
  - Motions of class $\mathscr{C}^2$.
- Study of trapeze velocity profile (said *bang-bang*) :
  - Continuous motion in velocity ;
  - Guaranteed minimum time by saturating velocity and acceleration simultaneously.

Example : calculation of the third-order polynomial interpolation :
Determine the polynomial time law taking into account the initial $(\dot{q}(t_0) = \dot{q}_0)$ and terminal $(\dot{q}(t_f) = \dot{q}_f)$ constraints in velocities.

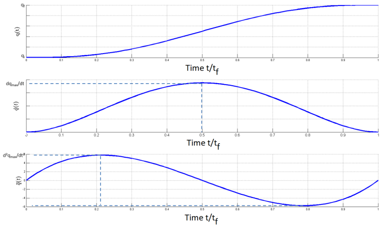
Example

Main methods for generating *Point-to-Point* joint trajectory

Study of the fifth-order polynomial interpolation [12]

- Boundary conditions :
  - $q(0) = q_0$ et $q(t_f) = q_f$
  - $\dot{q}(0) = 0$ et $\dot{q}(t_f) = 0$
  - $\ddot{q}(0) = 0$ et $\ddot{q}(t_f) = 0$
- Expression of the time law :

$$r(t) = 10\left(\frac{t}{t_f}\right)^3 - 15\left(\frac{t}{t_f}\right)^4 + 6\left(\frac{t}{t_f}\right)^5$$



- Maximal velocity and acceleration :

$$\begin{cases} \dot{q}_{max} = \frac{15}{8}\left(\frac{q_f - q_0}{t_f}\right) \\ \ddot{q}_{max} = \frac{10}{\sqrt{3}}\left(\frac{q_f - q_0}{t_f^2}\right) \end{cases}$$

- Minimal final time for the j-th joint while saturating velocity and acceleration

$$t_{f_j} = \max\left[\frac{15\,|D_j|}{8k_{v_j}}, \sqrt{\frac{10\,|D_j|}{\sqrt{3}k_{a_j}}}\right]$$

- Minimal final global time t_f while coordinating all the joints

$$t_f = \max\left(t_{f_1}, \ldots, t_{f_n}\right)$$

Main methods for generating *Point-to-Point* joint trajectory

Bang-bang profile with constant velocity phase [12]

- Continuous velocity trajectory guaranteeing a minimal time :
  1. saturation in acceleration for $0 \leq t \leq T_i$ until reaching the maximal velocity ;
  2. saturation in velocity for $T_i \leq t \leq t_{f_i} - T_i$;
  3. saturation in deceleration for $t_{f_i} - T_i \leq t \leq t_{f_i}$.
- Condition for the existence of a constant velocity phase equal to the maximal velocity :

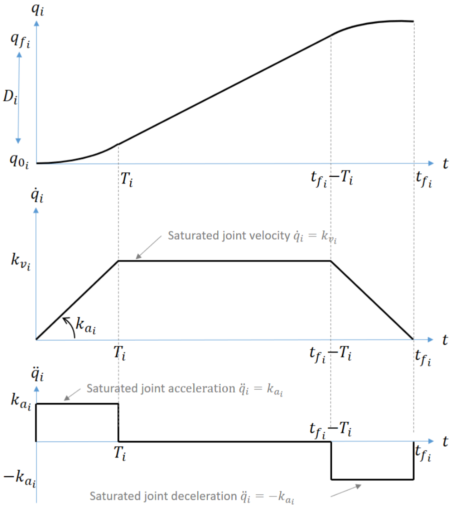
$$|D_i| > \frac{k_{v_i}^2}{k_{a_i}}$$

Demonstration

- Travelling time :

$$t_{f_i} = \frac{|D_i|}{k_{v_i}} + T_i = \frac{|D_i|}{k_{v_i}} + \frac{k_{v_i}}{k_{a_i}}$$

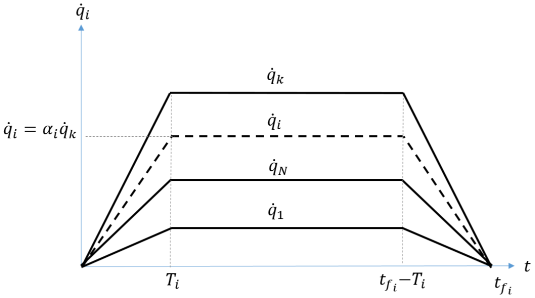
Demonstration



Main methods for generating *Point-to-Point* joint trajectory

Bang-bang profile with constant velocity phase [22]

- Synchronisation of all the robot axes :
  - Setting all axes to the slowest axis (i.e. slowing down the fastest axes so that they finish their movement at the same time as the slowest) ;
  - Reduced sollicitation of the actuators without degrading the travelling time of the effector from one point to the next ;
  - Synchronization of acceleration and deceleration phases by a homothetic law ;



Generalities

General formulation of the problem

Let consider the following homogeneous matrices giving respectively :

- the initial desired pose of the end-effector :

$$^{0}\overline{g}_{0E} = \begin{bmatrix} ^{0}R_{0E} & ^{0}p_{0E} \\ 0_{1\times 3} & 1 \end{bmatrix}$$

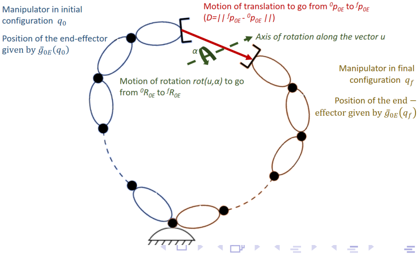
- the final desired pose of the end-effector :

$$^{f}\overline{g}_{0E} = \begin{bmatrix} ^{f}R_{0E} & ^{f}p_{0E} \\ 0_{1\times 3} & 1 \end{bmatrix}$$

Searching for a rectilinear trajectory of the tool point by decomposing the movement according to :

- a motion of translation in straight path between the origins of $^{0}\overline{g}_{0E}$ and $^{f}\overline{g}_{0E}$;
- a motion of rotation α around an axis u of the end-effector enabling to align $^{0}R_{0E}$ and $^{f}R_{0E}$.

Synchronous termination of both motions.



Calculation of the interpolation

Evolution of the desired situation (1/2)

- Motion of translation :
  - distance to travel, such that :

$$D = \left\| {}^{f}p_{0E} - {}^{0}p_{0E} \right\| = \sqrt{({}^{f}p_x - {}^{0}p_x)^2 + ({}^{f}p_y - {}^{0}p_y)^2 + ({}^{f}p_z - {}^{0}p_z)^2}$$

- Motion of rotation :
  - Calculation of the parameters for rotation (u and α) as solutions of :

$${}^{0}R_{0E} R_{(u,\alpha)} = {}^{f}R_{0E}$$

where $R_{(u,\alpha)}$ is the 3×3 rotation matrix corresponding to a rotation of angle α around the vector u (given in the terminal frame) :

$$R_{(u,\alpha)} = {}^{0}R_{0E}^{t}\, {}^{f}R_{0E} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

The kinematic inversion theorem (seen in chapter 2) allows to deduce u and α ($0 \leq \alpha \leq \pi$) according to :

$$\alpha = atan2\left(\frac{1}{2}\sqrt{(r_{32} - r_{23})^2 + (r_{13} - r_{31})^2 + (r_{21} - r_{12})^2}, \frac{1}{2}(r_{11} + r_{22} + r_{33} - 1) \right)$$

and

$$u = \begin{bmatrix} \frac{r_{32} - r_{23}}{2S\alpha} & \frac{r_{13} - r_{31}}{2S\alpha} & \frac{r_{21} - r_{12}}{2S\alpha} \end{bmatrix}^t \quad \text{if } S\alpha \neq 0$$

Calculation of the interpolation

Evolution of the desired situation (2/2)

- Timing law

$$\overline{g}_{0E}(t) = \begin{bmatrix} R(t) & p(t) \\ 0_{1\times 3} & 1 \end{bmatrix}$$

with

- evolution pf the translation motion :

$$p(t) = {}^{0}p_{0E} + r(t)\left({}^{f}p_{0E} - {}^{0}p_{0E}\right) = {}^{0}p_{0E} + \frac{s(t)}{D}\left({}^{f}p_{0E} - {}^{0}p_{0E}\right)$$

- evolution of the rotation motion :

$$R(t) = {}^{0}R_{0E}R_{(u,r(t)\alpha)}$$

where $r(t)$ is the interpolation function and $s(t)$ is the travelled distance at the moment t.

- Remarks
  - Expression of the rotation about a fixed axis w.r.t. the base frame $\mathscr{R}_0$: calculation of the rotation $R_{(u,\alpha)}$ as solution of $R_{(u,\alpha)}{}^{0}R_{0E} = {}^{f}R_{0E}$;
  - Interpolation methods presented in the case of trajectory generation in joint space that can be used to generate the synchronized motion for the two variables D and α in time ;
  - Necessity to make sure, with the help of the kinematic model, that the trajectory is achievable by the robot from the point of view of joint velocities and accelerations ;
  - Use of the IGM can pose several problems (already discussed in chapter *Inverse Kinematics*) :
    - need to ensure that the desired operational position is achievable for the IGM to accept at least one solution ;
    - possible crossing of a singular configuration (loss of degrees of freedom).

Path motion

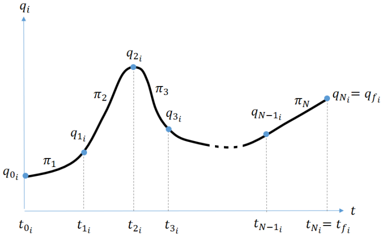
Path motion seen as trajectory generation with via points

- Insertion of via points between the initial and final points in order to :
  - guarantee better monitoring on the executed trajectories for more complex applications ;
  - avoid collisions between the robot and its environment (specifying points more densely in those segments of the path where obstacles have to avoided).
- Problem seen as generating a trajectory when $N+1$ points, termed *path points*, are specified and have to be reached by the manipulator at certain instants of time.
  - Use of one single polynomial passing through all the points and satisfying the boundary conditions is difficult to exploit with increasing the number of points and may lead to oscillatory trajectory behaviour as the order of the polynomial increases.
  - Splitting the trajectory in low degree polynomials between the path points provides an elegant way of overcoming this problem and reduces the computational burden of trajectory generation

Path motion

Brief overview for generating path motions with constraints at path points

- Searching for a function $q_i(t)$ formed by a sequence of N polynomials $\pi_k(t)$ for $k = 1, \dots, N$ continuous with continuous derivatives (e.g. continuous first derivatives for the case of cubic polynomials) ;
- Searching for low-order polynomials $\pi_k(t)$ for generating *Point-to-Point* trajectory between t_{k-1} and t_k using previous method (e.g. lowest order being cubic polynomials for imposing continuity of velocities at the path points), while imposing :
  - conditions for the polynomials $\pi_k(t)$ (e.g. $\pi_k(t_{k-1}) = q_{k-1}$, $\pi_k(t_k) = q_k$, $\dot{\pi}_k(t_{k-1}) = \dot{q}_{k-1}$ and $\dot{\pi}_k(t_k) = \dot{q}_k$),
  - as well as conditions for continuity at the path points (e.g. $\dot{\pi}_k(t_{k-1}) = \dot{\pi}_{k-1}(t_{k-1})$).



Other methods exist for path motion : interpolating polynomials with continuous accelerations at path points (splines), interpolating linear polynomials with parabolic blends, and so on.
Similarly, in the task space, the above interpolation techniques can also been applied to propose smooth Cartesian polynomials from the assignment of $N + 1$ points specifying directly the end-effector location.

Prioritization of the control

Task level

Task objective (as specified by the user) : analysis and decomposition into actions (based on knowledge models of the robot and the system environment)

Action level

Symbolic commands converted to intermediate configuration sequences

Primitive level

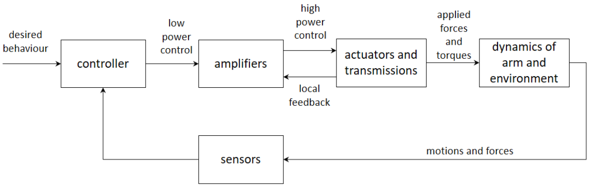
Choice of a control strategy and references in trajectory generation for servo drives

Low-level control

Implementation of control algorithms, real-time calculation of low-level commands for actuator/sensor pairs

Typical control robot system

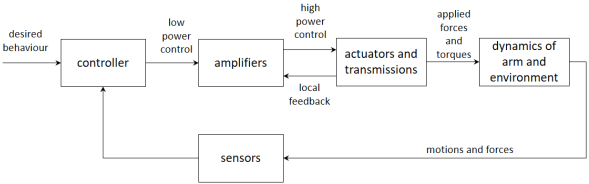
Control system overview (1/2)



- Typical embedded sensors :
  - encoders, resolvers for joint position and angle sensing ;
  - joint force-torque sensors and/or multi-axis force-torque sensors at the wrist between the end of the arm and the end-effector.
- Different technologies for creating mechanical power transforming the speeds and forces, and transmitting to the robot joints :
  - many types of actuators : mainly electric, hydraulic or pneumatic ;
  - actuators located at the joint themselves (e.g. gearheads) or remotely (e.g. with mechanical power transmitted by cables or timing belts).

Typical control robot system

Control system overview (2/2)



- Amplifiers :
  - an inner control loop is used to help the amplifier and the actuator to achieve the desired force or torque (e.g. a DC motor amplifier in torque control mode may sense the current actually flowing through the motor and implement a local controller to better match the desired current, since the current is proportional to the torque produced by the motor) ;
  - in the following, we consider each joint's amplifier and actuators as ideal and treat them as a transformer from controller output reference signals to applied forces and torques.
- Controller :
  - The rest of this section is devoted to the control algorithms that go *inside* the *controller* box.

Specifications for the control algorithms

Expected performances and pre-requisites for the controller

- Performances under nominal conditions
  - velocity for the task realization (i.e. cycle time)
  - precision and repetability (in static and dynamic operations)
  - energy needs

$\Rightarrow$ Interests for advanced model-based control approaches

- Robustness in perturbed conditions
  - adaptation to changing environments
  - stability and repeatability despite disturbances, parameter changes, uncertainties and modelling errors

$\Rightarrow$ Use of sensor measurements and feedback loops

Preferred control approaches

- Control in closed loop (insensitive to disturbances and small parameter variations)
- Robust control (tolerance to a relatively wide range of uncertainties)
- Adaptive control (on-line performance improvement, adaptation to parametric variations)
- Learning-based approach control (particularly relevant for high-level control strategies)

Practical aspects for robot controller architecture

Synchronous tasks related to low-level control

- Operation running under a real-time software architecture
  - VxWorks (Wind River)
  - RT Linux, Xenomaï, Linux RTAI, ...
- Guarantee of a constant sampling time period T_e (of about a few ms)
- Functionalities
  - measurement sampling
  - computation of the torque references from the control loops
  - low-level safety management

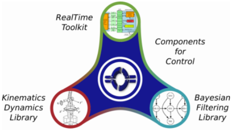
Software supervision

- Asynchronous operation w.r.t. the low-level controller
- Functionalities
  - Viewing robot state
  - Switching command for the robot operating mode (on/off, path generation, ...).
  - Programming robot tasks using a macro language

WIND RIVER

Linux

XENOMAI



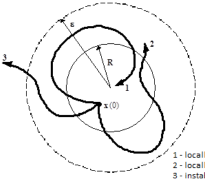
Reminders on the stability of dynamic systems

Physical insights

Consider a dynamical system which satisfies

$$\dot{x} = f(x, t), \qquad x(t_0) = x_0, \qquad x \in \mathbb{R}^n$$

- A point x_e is an equilibrium point of the above system if $f(x_e, t) = 0$.
- Intuitively, we say an equilibrium point is :
  - locally stable if all solutions which start near x_e (meaning that the initial conditions are in a neighbourhood of x_e) remain near x_e for all time.
  - locally asymptotically stable if x_e is locally stable and, furthermore, all solutions starting near x_e tends towards x_e as $t \longrightarrow \infty$.



1 - locally asymptotically stable
2 - locally stable
3 - instable

- These definitions are *local* : they describe the behaviour of a system *near* an equilibrium point. We say that an equilibrium point x_e is globally stable if it is stable for all initial conditions $x_0 \in \mathbb{R}^n$.

Reminders on the stability of dynamic systems

Basic definitions

For the dynamical system $\dot{x} = f(x, t)$, the equilibrium point x_e, such that, if $x(t_0) = x_e$ then $x(t) = x_e$ for all $t \geq t_0$, is said :

- stable if $\forall t \geq t_0$

$$\forall \epsilon > 0, \quad \exists R > 0 \quad \text{such that} \quad \|x(t_0) - x_e\| < R \quad \Rightarrow \quad \|x(t) - x_e\| < \epsilon$$

- asymptotically stable if it is stable and if

$$\exists \alpha > 0 \quad \text{such that} \quad \|x(t_0) - x_e\| < \alpha \quad \Rightarrow \quad \lim_{t \to +\infty} x(t) = x_e$$

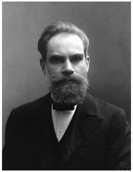
- globally asymptotically stable if it is asymptotically stable for all initial conditions $x_0 \in \mathbb{R}^n$.

Reminders on the stability of dynamic systems

The Direct method of *Lyapunov*

Lyapunov's direct method (also called the second method of *Lyapunov*) allows us to determine the stability of a system without explicitly integrating the differential equation $\dot{x} = f(x, t)$.

- The method is a generalization of the idea that if there is some *measure of energy* in a system, then we can study the rate of change of the energy of the system to ascertain the stability.
- The problem is to search for the existence of a candidate function that verifies certain specific conditions related to the function f of the differential equation and to x_e.



Aleksandr Liapounov (1857-1918)

Reminders on the stability of dynamic systems

Candidate function of *Lyapunov* The function $V(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ is positive definite (PD) if $V(x) > 0$ except in 0 where $V(0) = 0$.

- If $V(x) \geq 0$ then the function is positive semidefinite (PSD)
- Typically, the forms of V can be chosen as quadratic ($\frac{1}{2}x^t Px$ where $P \succ 0$)

Reminders on the stability of dynamic systems

Candidate function of *Lyapunov* The function $V(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ is positive definite (PD) if $V(x) > 0$ except in 0 where $V(0) = 0$.

- If $V(x) \geq 0$ then the function is positive semidefinite (PSD)
- Typically, the forms of V can be chosen as quadratic ($\frac{1}{2}x^t Px$ where $P \succ 0$)

Sufficient condition for stability/instability in the sense of *Lyapunov*

- Stability
  - Let V be a candidate function such that its time-derivative along the trajectory of the system $\dot{x} = f(x)$ verifies $\dot{V}(x) \leq 0$, then 0 is a stable equilibrium state.
  - Let V be a candidate function such that its time-derivative along the trajectory of the system $\dot{x} = f(x)$ verifies $\dot{V}(x) < 0$, then 0 is an asymptotically stable equilibrium state.
- Instability
  - If there exists V candidate function such that $\dot{V}(x) > 0$ along the trajectory of the system, then 0 is an instable equilibrium state.

Reminders on the stability of dynamic systems

Remarks about the method

- Basic procedure based on the generation of an energy-type scalar function for the dynamic system and examine its time derivative. We can thus conclude about the stability without resorting to the explicit solution of non-linear differential equations ;
- Problem with this method is finding a Lyapunov function for the system in the absence of a methodology. In the nonlinear case, there is no systematic method to choose a suitable *Lyapunov* function, hence the use of experience, intuition and physical considerations.

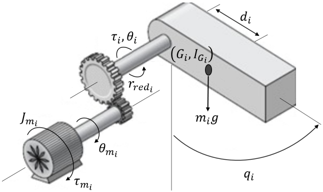
Reminders on the stability of dynamic systems

Remarks about the method

- Basic procedure based on the generation of an energy-type scalar function for the dynamic system and examine its time derivative. We can thus conclude about the stability without resorting to the explicit solution of non-linear differential equations ;
- Problem with this method is finding a Lyapunov function for the system in the absence of a methodology. In the nonlinear case, there is no systematic method to choose a suitable *Lyapunov* function, hence the use of experience, intuition and physical considerations.

Example

Study of one revolute robotic axis



- Considered angular variation $q_i \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$;
- Searching for the equilibrium state of the autonomous system ;
- Studying stability of the equilibrium points of the autonomous system.

Example

Reminders on the stability of dynamic systems

Stability in the case of Linear Temporally Invariant system

Given the following linear system :

$$\dot{x} = Ax$$

we consider a quadratic candidate function of *Lyapunov*, $V(x) = x^t P x$, thus

$$\dot{V}(x) = \dot{x}^t P x + x^t P \dot{x} = x^t A^t P x + x^t P A x = x^t \left(A^t P + PA\right) x = -x^t Q x$$

A necessary and sufficient condition for a system $\dot{x} = Ax$ to be asymptotically stable is that $\forall Q = Q^t \succ 0$, the matrix P, unique solution of the following *Lyapunov* equation, is positive definite :

$$A^t P + PA + Q = 0$$

Reminders on the stability of dynamic systems

Stability in the case of Linear Temporally Invariant system

Given the following linear system :

$$\dot{x} = Ax$$

we consider a quadratic candidate function of *Lyapunov*, $V(x) = x^t P x$, thus

$$\dot{V}(x) = \dot{x}^t P x + x^t P \dot{x} = x^t A^t P x + x^t P A x = x^t \left(A^t P + PA\right) x = -x^t Q x$$

A necessary and sufficient condition for a system $\dot{x} = Ax$ to be asymptotically stable is that $\forall Q = Q^t \succ 0$, the matrix P, unique solution of the following *Lyapunov* equation, is positive definite :

$$A^t P + PA + Q = 0$$

In linear control theory, we show that :

- the system is asymptotically (and even exponentially) stable iif all eigenvalues of A have negative real part (belonging to the left complex half-plane) ; ;
- the system is unstable if there is at least an eigenvalue of A with a positive real part.

Reminders on the stability of dynamic systems

Stability in the case of Linear Temporally Invariant system

Given the following linear system :

$$\dot{x} = Ax$$

we consider a quadratic candidate function of *Lyapunov*, $V(x) = x^t P x$, thus

$$\dot{V}(x) = \dot{x}^t P x + x^t P \dot{x} = x^t A^t P x + x^t P A x = x^t \left(A^t P + PA\right) x = -x^t Q x$$

A necessary and sufficient condition for a system $\dot{x} = Ax$ to be asymptotically stable is that $\forall Q = Q^t \succ 0$, the matrix P, unique solution of the following *Lyapunov* equation, is positive definite :

$$A^t P + PA + Q = 0$$

In linear control theory, we show that :

- the system is asymptotically (and even exponentially) stable iif all eigenvalues of A have negative real part (belonging to the left complex half-plane) ; ;
- the system is unstable if there is at least an eigenvalue of A with a positive real part.

This result can be exploited locally in the non-linear case ($\dot{x} = f(x)$), in considering $A = \frac{\partial f}{\partial x}$ evaluated at the equilibrium point.

Reminders on the stability of dynamic systems

LaSalle's theorem enables one to conclude asymptotic stability of an equilibrium point even when $\dot{V}(x)$ is NSD.

LaSalle's invariance principle

If there is a V candidate such as $\dot{V}(x) \leq 0$ along the trajectories of $\dot{x} = f(x)$, then the trajectories of the system converge asymptotically to the largest invariant set $M \subseteq S = \left\{ x \in \mathbb{R}^n : \dot{V}(x) = 0 \right\}$.

- The set M is invariant for a dynamic system when a trajectory starting from M remains for all time in M

$$x(t_0) \in M \quad \Rightarrow \quad x(t) \in M, \quad \forall t \geq t_0$$

- From *LaSalle*'s theorem, each solution starting from S converges to M when $t \to +\infty$;
- Thus, if $M \equiv \{0\}$, then 0 is an asymptotically stable equilibrium state.

Stabilization of a manipulator arm under the action of joint control

Regulation objectives

The objective of asymptotic stabilization fo the robot dyanmics

$$\Gamma = A(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q)$$

is achieved by the regulation of the state $x = [q, \dot{q}]^t$ towards its equilibrium state defined by $x_e = [q_e = q_d, \dot{q}_e = 0]$ where q_d is the desired position of the robot.

- Nonlinear state-space representation :

$$\dot{x} = \begin{pmatrix} \dot{q} \\ -A^{-1}(q)\left(C(q,\dot{q})\dot{q} + G(q)\right) \end{pmatrix} + \begin{pmatrix} 0 \\ A^{-1}(q) \end{pmatrix}\Gamma$$

  Note that $A^{-1}(q)$ is well defined.

- Equilibrium state of the robot (under the control action Γ_e) :

$$\begin{cases} \dot{q}_e = 0 \\ -A^{-1}(q_e)\left(C(q_e,\dot{q}_e)\dot{q}_e + G(q_e)\right) + A^{-1}(q_e)\Gamma_e = 0 \end{cases} \Rightarrow \begin{cases} \dot{q}_e = 0 \\ \Gamma_e = G(q_d) \end{cases}$$

  At the equilibrium, the joint velocities are null and the joint torque control balance the gravitational torques.

Decentralized P.D. joint controller

P.D. joint control law

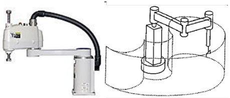
Let consider the following linear P.D. control law

$$\Gamma = u = K_p\left(q_d - q\right) - K_d\dot{q}$$

realized axis-per-axis at the joint level, for which $K_p \succ 0$ and $K_d \succ 0$ are positive definite and diagonal matrices ($K_p = diag\left(k_{p_i}\right)$ and $K_d = diag\left(k_{d_i}\right)$).
In absence of terms due to gravity effects (i.e. $G\left(q\right) = 0$), the state of the robot $\left[q_d, 0\right]^t$ under the previous control action is asympotically stable.

Demonstration of stability

1. Propose a *Lyapunov* candidate funtion V ;
2. Calculate $\dot{V}$ and conclude about stability using the equation of the system dynamics in closed-loop and the fact that $\left(\dot{A} - 2C\right)$ is skew-symmetric ;
3. Exploiting the *LaSalle*'s theorem, conclude about the asymptotic convergence of the system



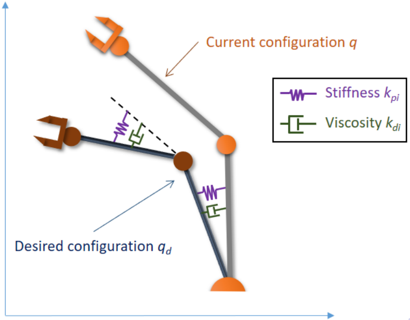
Demonstration

Decentralized P.D. joint controller

Physical insight

The *Proportional* and *Derivative* gains of the control law $\Gamma = u = K_p (q_d - q) - K_d \dot{q}$ realized at the joint level, for which $K_p \succ 0$ and $K_d \succ 0$ are positive definite and diagonal matrices ($K_p = diag\,(k_{p_i})$ and $K_d = diag\,(k_{d_i})$), can be seen as stiffness and viscosity that are programmed by the controller.

Sometimes, we speak about virtual stiffness and viscosity.



Decentralized P.D. joint controller

Practical implementation within the controller of the derivative action when only motor (or joint) position is measured (e.g. by means of encoders)

Continuous time control law

$$u(t) = K_p e(t) + K_d \dot{e}(t)$$

$e = q_d - q$, $\dot{e} = -\dot{q}$

Representation in *Laplace* domain

$$u(s) = (K_p + K_d s)\, e(s)$$

Not proper transfer function

$$u(s) = \left(K_p + K_d \frac{s}{1+\tau s}\right) e(s)$$

Derivative action limited to pulsations $\omega \leq 1/\tau$

Z-transform (sampling period T_e)

$$u(z) = \left(K_p + K_d \frac{1-z^{-1}}{T_e}\right) e(z)$$

$$u(z) = \left(K_p + K_d \frac{\frac{1-z^{-1}}{T_e}}{1+\tau\frac{1-z^{-1}}{T_e}}\right) e(z)$$

Discrete-time implantation

$$u_k = K_p e_k + K_d \frac{e_k - e_{k-1}}{T_e}$$

$$u_k = K_p e_k + \frac{\tau}{\tau+T_e} u_{k-1} + \frac{K_d}{\tau+T_e}\left(e_k - e_{k-1}\right)$$

Two possible realisations

Decentralized P.D. joint controller with gravity compensation

- For robots motions submitted to gravity influence, modified control law as follows :

$$u = \Gamma = K_p (q_d - q) - K_d \dot{q} + \hat{G}(q)$$

with $K_p \succ 0$ et $K_d \succ 0$ and a nonlinear compensation of gravity torques, providing the equilibrium state $(q_d, 0)$ asymptotically stable (as demonstration as previously).
- In the case where the gravity compensation is not realized (or not perfectly realized, i.e. $\hat{G}(q) \neq G(q)$) :
  - system converging towards the equilibrium state $[q^* \neq q_d, 0]$;
  - static position error $\epsilon_q = q^* - q_d$ due to the influence of joint perturbations that are not rejected by the controller ;
  - For position-controlled robot with P.D. controller with gravity compensation, very precise knowledge of the terms of gravity required to ensure high static positioning accuracy of the system.
- In linear control theory, the integral action I in the control law generally required to eliminate those errors in steady-state for step response.

Decentralized P.D. joint controller with integral term [12]

Discussions about the role of the integral term

1. Deactivation of the integral component when the position error is very large, the proportional term being sufficient (avoidance of actuator saturation)

$$u_1(t) = K_p(q_d - q) - K_d\dot{q} + \hat{G}(q)$$

2. Activation of the integral action when the static error is due to the force of gravity and friction around the final position

$$u_2(t) = K_p(q_d - q) - K_d\dot{q} + K_i \int_{t_0}^{t} (q_d - q)\, d\lambda$$

Advantages for using decentralized joint PID

- Ease of implantation
- Low computational cost

Drawbacks of this methodology

- Reference overshoots (varying time response of the robot according to its configuration)
- Poor tracking accuracy during high-dynamics motions

Decentralized P.I.D. joint controller

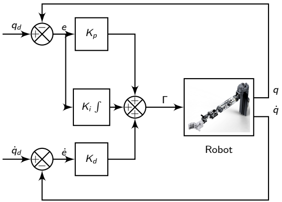
Robotic P.I.D. control law at joint level

The following linear control law of type *Proportional Integral Derivative*

$$\Gamma(t) = u(t) = \underbrace{K_p(q_d - q) + K_d(\dot{q}_d - \dot{q})}_{u_1(t)} + \underbrace{K_i \int_{t_0}^{t} (q_d - q)\, d\lambda}_{u_2(t)}$$

with $K_p \succ 0$, $K_d \succ 0$ and $K_i \succ 0$, ensures the cancellation of static errors due to incomplete compensation (or failure to take into account) of gravity or friction phenomena.

Generally, the term $\dot{q}_d$ is not taken into account in regulation objectives.



Decentralized P.I.D. joint controller

A (simple) methodology for synthesizing the P.I.D. controller gains (1/2)

- Assumptions considered about the robotic mechanism for the controller gains synthesis
  - Robot considered as a linear system (at least locally) whereas it is modeled as a non-linear and coupled dynamic system
  - Controls independent of joint couplings (SISO controller)
  - Constant gains of the P.I.D. controller
- Simplified model for joint i : second-order linear system with constant coefficients

$$\tau_i = a_{ii}\ddot{q}_i + f_{v_i}\dot{q}_i + \gamma_i$$

  where a_{ii} designates the maximum value of the element of the inertia matrix A and γ_i the disturbance torque due to coupling and other phenomena neglected in the approximation
- losed-loop system transfer function (with zero disturbance $\gamma_i = 0$)

$$\frac{q_i(s)}{q_{d_i}(s)} = \frac{k_{d_i}s^2 + k_{p_i}s + k_{i_i}}{a_{ii}s^3 + \left(k_{d_i} + f_{v_i}\right)s^2 + k_{p_i}s + k_{i_i}}$$

Demonstration

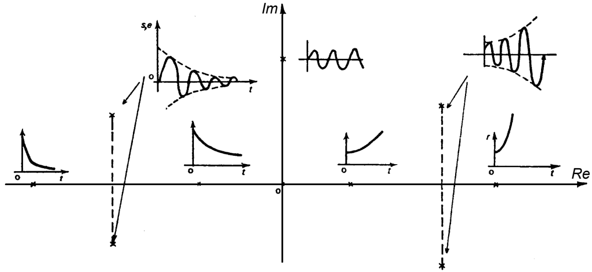
Decentralized P.I.D. joint controller

A (simple) methodology for synthesizing the P.I.D. controller gains (2/2)

- Searching for the poles of the closed-loop system
  - Characteristic polynomial : $\Delta(s) = a_{ii}s^3 + \left(k_{d_i} + f_{v_i}\right)s^2 + k_{p_i}s + k_{i_i}$
  - Tuning of the P.I.D. gains :
    - Common solution in robotics : fast response without oscillations
    - Choice of gains in order to obtain a negative real triple pole : $\Delta^*(s) = a_{ii}(s + \omega_i)^3$:

$$\begin{cases} k_{p_i} = 3a_{ii}\omega_i^2 \\ k_{d_i} = 3a_{ii}\omega_i - f_{v_i} \\ k_{i_i} = a_{ii}\omega_i^3 \end{cases}$$

  - Compromise on the choice of ω_i (fast dynamics vs. stability) : pulsation lower than the resonance pulsation (mechanical vibratory modes, used value : $\omega_i = \omega_{r_i}/2$)



P.I.D. regulation in Cartesian mode

Advantages of the approach of controlling in the Cartesian space

- Easier control of the performances actually achieved in the Cartesian space by the control law (in particular overshoots and oscillations), since the robot's temporal response varies according to its joint configuration. ;
- Improved tracking accuracy for highly-dynamic Cartesian motions.

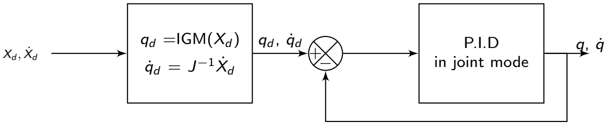
P.I.D. regulation in Cartesian mode

Advantages of the approach of controlling in the Cartesian space

- Easier control of the performances actually achieved in the Cartesian space by the control law (in particular overshoots and oscillations), since the robot's temporal response varies according to its joint configuration. ;
- Improved tracking accuracy for highly-dynamic Cartesian motions.

1^{st} practice : transposition of the joint control problem

- Previous linear control law of type *Proportional Integral Derivative* used for implementation in the joint space ;
- Generation of the setpoints q_d and $\dot{q}_d$ resulting from the transformation of the setpoints motions from the operational space X_d and $\dot{X}_d$ to the joint space ;
- Nevertheless, position error still expressed in the joint space.



P.I.D. regulation in Cartesian mode

2nd practice : formulation of the control problem in the Cartesian space (1/2)

Direct specification in the Cartesian space

The following linear control law of the Proportional Integral Derivative type

$$\Gamma(t) = u(t) = J(q)^t \left[K_p (X_d - X) + K_d \left(\dot{X}_d - \dot{X} \right) + K_i \int_{t_0}^{t} (X_d - X)\, d\lambda \right]$$

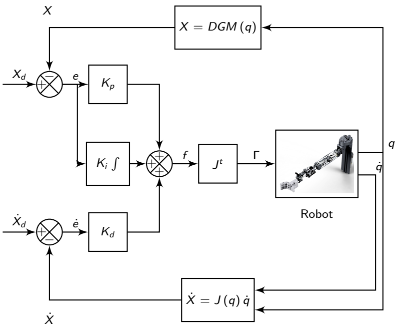
allows the regulation of the end-effector directly in the Cartesian (or operational) space..

- Structural similarity with the P.I.D. command in joint mode (replacement of q by X in the joint control law equation) ;
- Transformation of the error obtained in the operating space to the joint space by multiplication by J^t.

P.I.D. regulation in Cartesian mode

2^{nd} practice : formulation of the control problem in the Cartesian space (2/2)

- Block diagram of the control law



- Practical aspects for the estimations of the cartesian position and velocity :
  - Estimation of X from the DGM ;
  - Estimation of $\dot{X}$:
    - Numerical differentiation of the joint position q (in the case of absence of tachometric measurements)
    - Use of the forward kinematic model for estimation $\dot{X}$ from $\dot{q}$

P.I.D. regulation in Cartesian mode

Stability of a variant of the P.D. controller in Cartesian mode

The following control law of type *Proportional Derivative*

$$\Gamma(t) = J(q)^t \left[K_p (X_d - X)\right] - K_d \dot{q} + \widehat{G}(q),$$

with $K_p \succ 0$ et $K_d \succ 0$, converge asympotically towards the set

$$S = \{\dot{q} = 0, q / K_p (X_d - MGD(q)) \in N_{J^t}\}$$

Demonstration

- Corollary : For a given initial state $[q(t_0), \dot{q}(t_0)]$, in the case where the robot does not encounter any singular configurations (i.e. configurations for which $\operatorname{rank}(J) = r < M \leq N$) during its trajectory, then the system converge asymptotically towards the unique (in the non-redundant case $M = N$) or towards one of the many (in the redundant case $M < N$) state, such that :

$$q \in \mathbb{R}^n / MGD(q) = X_d \quad and \quad \dot{q} = 0.$$

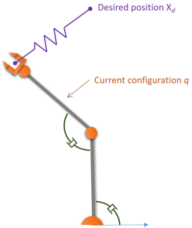
Remind that $\operatorname{rank}\left(J^t\right) = \operatorname{rank}(J)$.

P.I.D. regulation in Cartesian mode

Physical insights

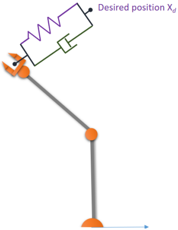
1- P.D. regulation in cartesian mode with gravity compensation

$$\Gamma(t) = J(q)^t \left[K_p (X_d - X)\right] - K_d \dot{q} + \widehat{G}(q).$$



2- Possible variant for the regulation with gravity compensation

$$\Gamma(t) = J(q)^t \left[K_p (X_d - X) - K_d \dot{X}\right] + \widehat{G}(q).$$



Transformation of *elastic* (or *visco-elastic* in the second case) forces/torques acting on the end-effector via J^t to control torque at the joint level.

Linearizing and decoupling techniques

Class of applicable solutions for the trajectory tracking problem with C^2 continuous trajectory functions

$$q_d(t),\ \dot{q}_d(t) \text{ and } \ddot{q}_d(t)$$

- Favourable context for the use of this approach :
  - Need for high dynamic precision despite the fact that the robot evolves according to sometimes large dynamics defined by the trajectory ;
  - Insufficiency of the previous decentralized time-invariant controller with feedback actions to control multivariable, nonlinear and coupled dynamical system..

Computed torque control approach

- Linearizing and decoupling techniques consisting of transforming a nonlinear control problem into a linear one by using an appropriate feedback law ;
- Uniform response regardless of robot configuration ;
- Requiring the online calculation of the dynamic model of the robot.

$\Rightarrow$ Control strategy based on the flatness system theory (the equations of the robot define a so-called *flat* system whose *flat outputs* are the joint variables q [6])

Basics of flatness system theory

Definition [31]

A dyanmical system defined its equation $\dot{x}(t) = f(x(t), u(t))$ where $x(t)$ is the state and $u(t)$ the control action is said *flat* if there exists a vector $z(t)$ such that :

$$z(t) = h\left(x(t), u(t), u^{(1)}(t), \ldots, u^{(r)}(t)\right)$$

et two functions $\phi(.)$ et $\psi(.)$ such that

$$x(t) = \phi\left(z(t), z^{(1)}(t), \ldots, z^{(p)}(t)\right)$$
$$u(t) = \psi\left(z(t), z^{(1)}(t), \ldots, z^{(m)}(t)\right)$$

where f, h, ψ, ϕ are regular smooth functions and r, p, m are three integers.

- z si called the *flat output* of the system ;
- All the dynamic behaviour of the system is then described from the behaviour of its flat output ;
- There is no uniqueness of the flat outputs, so the parameterization is not unique.

Basics of flatness system theory

Property of flatness for rigid robots

The dynamics of the rigid robot

$$\Gamma = A(q)\,\ddot{q} + \underbrace{C(q,\dot{q})\,\dot{q} + G(q) + \Gamma_f(q,\dot{q})}_{H(q,\dot{q})}$$

rewritten under a nonlinear state-space representation ($x_1 = q$, $x_2 = \dot{x}_1$ and $u = \Gamma$)

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = A(x_1)^{-1}\left(u - H(x_1, x_2)\right) \end{cases}$$

has the vector $z = x_1$ as flat output.

Indeed, the state and the control vectors of the system is expressed entirely from z according to :

- $x_2 = \dot{z}$ (first condition fulfilled)
- $u = A(z)\,\ddot{z} + H(z,\dot{z})$ (second condition fulfilled)

Note that $A(x_1)^{-1}$ is well defined.

Basics of flatness system theory

Closed-loop control is required for the desired tracking problem and the relation

$$u(t) = \psi\left(z(t), z^{(1)}(t), \ldots, z^{(m)}(t)\right)$$

is the one that will allow us to build the control algorithm.

Commande par platitude [31]

The flatness control is defined by :

$$u(t) = \psi\left(z(t), z^{(1)}(t), \ldots, z^{(m-1)}(t), v(t)\right)$$

where $v(t)$ is the new control input. When $\frac{\partial\psi(\cdot)}{\partial z^{(m)}}$ is locally invertible, it leads to the decoupled system :

$$z^{(m)}(t) = v(t)$$

The initial control problem is then reduced to that of a linear, invariant, decoupled system formed by chains of pure integrators of order m from z.

Application of the flatness control theory to the case of rigid robots

- **Expression of the control law**
  By applying the previous principle of flatness control to the case of a rigid robot, the expression of the control law is written :

$$u = \hat{A}(q)\, v + \hat{H}(q, \dot{q})$$

  where $z^{(2)}(t)$ has been replaced by $v(t)$. We note $\hat{A}$ and $\hat{H}$ the estimates of the dynamic model contributions of the robot.

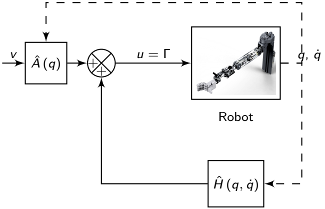
- **Expression of the system after the action of the linearizing feedback**
  in the ideal case where the model is perfectly known and identified ($\hat{A} = A$ and $\hat{H} = H$), the closed-loop system becomes :

$$\begin{aligned} A(q)\,\ddot{q} + H(q,\dot{q}) = u \Leftrightarrow & A(q)\,\ddot{q} + H(q,\dot{q}) = \hat{A}(q)\, v + \hat{H}(q,\dot{q}) \\ \Leftrightarrow & \ddot{q} = \left(A^{-1}(q)\,\hat{A}(q)\right) v \\ \Leftrightarrow & \ddot{q} = v \end{aligned}$$

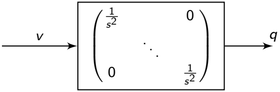
  Let note that $\frac{\partial \psi(\cdot)}{\partial z^{(m)}} = A(q)$ is well invertible.

Application of the flatness control theory to the case of rigid robots

- Under the action of the flatness control input $u = \hat{A}(q)\, v + \hat{H}(q, \dot{q})$, the rigid robot is brought back to the case of a decoupled linear system made up of double pure integrators whose command input is the vector v.
  - Linearizing and decoupling algorithm



  - Equivalent block diagram



Linearized and decoupled robot

Stabilizing corrector

Stabilisation [31]

After the decoupling and linearization step, an additional stabilization loop is required. The control input :

$$v(t) = z_d^{(m)}(t) + \sum_{j=0}^{m-1} K_j \left(z_d^{(j)}(t) - z^{(j)}(t) \right)$$

where K_j is a positive definite and diagonal matrix, leads to :

$$u(t) = \psi \left(z(t), z^{(1)}(t), \ldots, z^{(m-1)}(t), z_d^{(m)}(t) + \sum_{j=0}^{m-1} K_j \left(z_d^{(j)}(t) - z^{(j)}(t) \right) \right)$$

and ensures an asymptotically stable trajectory tracking

$$\lim_{t \to +\infty} (z_d(t) - z(t)) = 0$$

- In the case of rigid robot, the stabilizing control law becomes
$$v(t) = \ddot{q}_d(t) + K_d(\dot{q}_d(t) - \dot{q}(t)) + K_p(q_d(t) - q(t))$$
- It is a Proportional Derivative corrector with a feedforward action on the acceleration term.

Stabilizing corrector

- Error dynamics in closed-loop
  In the case of a decoupled and linearized rigid robot, a Proportional Derivative corrector with a feedforward action on the acceleration term

$$v(t) = \ddot{q}_d(t) + K_d(\dot{q}_d(t) - \dot{q}(t)) + K_p(q_d(t) - q(t))$$

  results in a closed-loop dyanmic system in joint space that is :

$$\begin{aligned} & v(t) = \ddot{q}_d(t) + K_d(\dot{q}_d(t) - \dot{q}(t)) + K_p(q_d(t) - q(t)) \\ \Leftrightarrow & \ddot{q}(t) = \ddot{q}_d(t) + K_d(\dot{q}_d(t) - \dot{q}(t)) + K_p(q_d(t) - q(t)) \\ \Leftrightarrow & (\ddot{q}_d(t) - \ddot{q}(t)) + K_d(\dot{q}_d(t) - \dot{q}(t)) + K_p(q_d(t) - q(t)) = 0 \\ \Leftrightarrow & \ddot{e}(t) + K_d\dot{e}(t) + K_p e(t) = 0 \end{aligned}$$

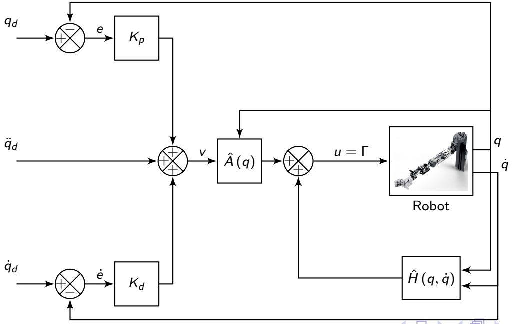
  where $e = q_d - q$ is the joint position error.
- Remarks
  - The dynamic of the error is globally exponentially stable ;
  - The choice for the gains K_{p_j} and K_{d_j} is realized according to the desired damping ratio ξ_j and pulsation ω_{n_j} as follows

$$\begin{cases} K_{p_j} = \omega_{n_j}^2 \\ K_{p_j} = 2\xi_j\omega_{n_j} \end{cases}$$

Stabilizing corrector

Summary of the general procedure for the *Computed Torque Control* (CTC)

1. Linearisation/decoupling of the model (through compensation of nonlinear *Coriolis*, centrifugal, gravity and friction torques) ;
2. Feedback action for guaranteeing the stability in closed-loop (correction of position and velocity with variable gains represented respectively by $\hat{A}(q)\,K_p$ and $\hat{A}(q)\,K_d$) ;
3. Anticipation of the reference in acceleration (inertial torques via $\hat{A}(q)\,\ddot{q}_d$).

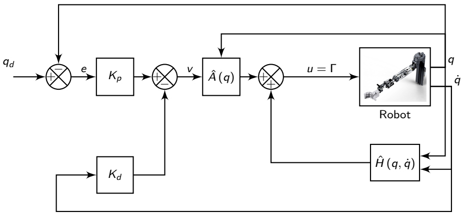


Stabilizing corrector

Variant of the approach considering only the specification of the final position

- Simplification of the general control law :

$$
\begin{aligned}
& u(t) = \hat{A}(q)\left[K_p(q_d - q) + K_d(\dot{q}_d - \dot{q}) + \ddot{q}_d\right] + \hat{C}(q,\dot{q}) + \hat{G}(q) \\
\Rightarrow \quad & u(t) = \hat{A}(q)\left[K_p(q_d - q) - K_d\dot{q}\right] + \hat{C}(q,\dot{q}) + \hat{G}(q)
\end{aligned}
$$



- Another variant of the control scheme :
  - Gravity compensation alone in the overall contribution of the nonlinear terms in $H(q,\dot{q})$:

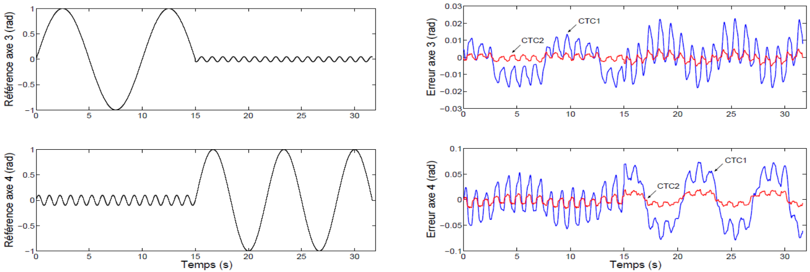
$$u(t) = K_p(q_d - q) - K_d\dot{q} + \hat{G}(q)$$

  - analogy with the P.D. corrector with gravity compensation at the joint level seen previously.

Experimental study - case of the regulation for the alone final position q_d

Monovariable P.D. correctors on the ASSIST robot previsouly decoupled and linearized (2-axes) [17]

- Tuning of two Computed Torque Controllers (CTC1 and CTC2) for trajectory tracking



(a) *Références moteur.* (b) *Erreurs moteurs.*

Influence of modelling errors on the *Computed Torque Control* performances

- Uncertainties introduced by the linearizing and decoupling strategy
  - Imperfect decoupling due to modeling or identification errors ;
  - Equation of the closed-loop system

$$\hat{A}(q)(\ddot{q}_d + K_d\dot{e} + K_p e) + \hat{H}(q,\dot{q}) = A(q)\ddot{q} + H(q,\dot{q})$$

  - Equation of the error dynamics

$$\ddot{e} + K_d\dot{e} + K_p e = \hat{A}^{-1}(q)\left[\left(A(q) - \hat{A}(q)\right)\ddot{q} + H(q,\dot{q}) - \hat{H}(q,\dot{q})\right]$$

    Excitation of the error by the uncertainties !
- Variant appropriated to the *Computed Torque Control* :
  - computation of the decoupling according to the variables corresponding to the desired motion :

$$u = \hat{A}(q_d)\, v + \hat{H}(q_d, \dot{q}_d)$$

    advantage of such decoupling computation uncontaminated by measurement noise ;
  - approaches of robust control [18].

Basics of the flatness theory for flexible-joint robots

Flat output for flexible-joint robots [8]

The reduced model for robots with flexible joints

$$\begin{aligned} A(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + \Gamma_{f_c} + K(q-\theta) &= 0 \\ J_m\ddot{\theta} + \Gamma_{f_m} - K(q-\theta) &= \Gamma \end{aligned}$$

is linearizable through feedback loop and the joint variable $z = q$ is a flat output.

- By isolating θ in the 1st equation and by derivating twice, it follows that :

  $$\ddot{\theta} = \ddot{q} + K^{-1}\left[A(q)q^{(4)} + 2\dot{A}(q,\dot{q})q^{(3)} + \ddot{A}(q,\dot{q},\ddot{q})\ddot{q} + \ddot{H}(q,\dot{q},\ddot{q},q^{(3)})\right]$$

  Adding the two equations of the dynamic model, and replacing $\ddot{\theta}$ by its expression as a function of q and its derivatives, lead to the expression of the motor torque as a function of q and its sucessive derivative until order 4 :

  $$\Gamma = J_m\ddot{\theta} + A(q)\ddot{q} + H(q,\dot{q}) \;\Rightarrow\; u = \psi(q,\dot{q},\ddot{q},q^{(3)},q^{(4)})$$

- Choosing the state $x = [x_1\ x_2\ x_3\ x_4]^T = [q\ \dot{q}\ \ddot{q}\ q^{(3)}]^T$ and $y = x_1 = q$, a state-space representation is : $\dot{x_1} = x_2$, $\dot{x_2} = x_3$, $\dot{x_3} = x_4$ et $\dot{x_4} = f_4(x) + g_4(x)\Gamma$ with

  $$\begin{aligned} f_4(x) = &-A(x_1)^{-1}KJ^{-1}\left[A(x_1)x_3 + H(x_1,x_2)\right] - A(x_1)^{-1}\left[K + \ddot{A}(x_1,x_2,x_3)\right]x_3 \\ &- 2A(x_1)^{-1}\dot{A}(x_1,x_2)x_4 - A(x_1)^{-1}\ddot{H}(x_1,x_2,x_3,x_4) \\ g_4(x) = &A(x_1)^{-1}KJ^{-1} \end{aligned}$$

Basics of the flatness theory for flexible-joint robots

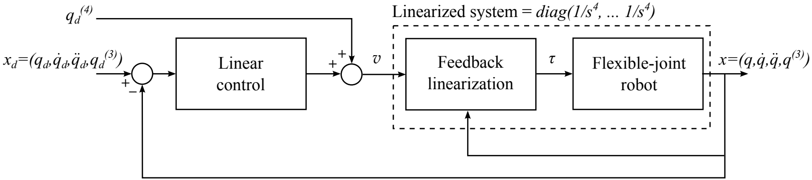
Linearizing and decoupling control

The joint reference trajectory q must be at least four time derivable to guarantee the existence of a continuous nominal torque. In this case, the linearizing feedback control for this system is :

$$\Gamma = g_4(x)^{-1}(v - f_4(x))$$

where v is the new control input for the linearized and decoupled system which consists of n quadruple independent integrators [8] :

$$q^{(4)} = v$$

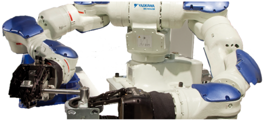


Controlling the robot/environment interaction

Many industrial applications requiring the contact of the robot end-effector with its environment.

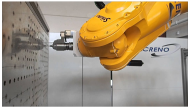
- quantity describing the state of interaction more effectively is the contact force at the manipulator's end-effector ;
- robot following the desired path while providing the force necessary either to overcome the resistance from the environment or to comply with it.

Polishing task



Bimanual assembling task

Grinding task



Drilling task

Existing approaches for controlling the robot/environment interaction

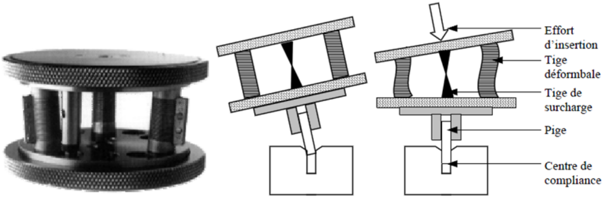
Objective of controlling the contact force between the robot end-effector and the environment

- Limitation of purely position-controlled robots to control the contact force :
  - need for accurate robot modelling and for an accurate knowledge of the stiffness of the environment in all its directions.
- Short overview of the main force-based control strategies :
  1. Indirect or *implicit* force control : control strategy via a position control outer-loop (i.e. input reference in position)
     - compliance control (methods involving the relation between position and applied force) [27] ;
     - impedance control (methods using the relation between velocity and applied force) [10].
  2. Direct or *explicit* force control : control strategy via a force control outer-loop (i.e. input reference in force)
     - parallel hybrid position/force control (method involving two position and force control outer-loops) [26] ;
     - external hybrid control (method involving a force control outer-loop and a position control inner-loop) [5] [24].

Compliant motion-based control approach

Passive stiffness control [1]

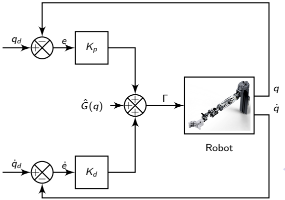
- Use of a deformable mechanical structure ;
- Correction of robot positioning errors (geometrical variation of structures) ;
- Disadvantages of adapting the tool to the task :
  - fixed stiffness ;
  - fixed compliance center.



Compliant motion-based control approach

Active stiffness control [27]

- Method actively controlling the apparent stiffness of the robot end-effector and allowing simultaneous position and force control
  - Position-controlled manipulator subjected to a disturbing force $\mathscr{F}_e$ which induces a position error $dX = X_d - X$; basic stiffness formulation being given by :
  $$\mathscr{F}_e = K_c dX$$
  where K_c is the desired 6×6 stiffness matrix.
  - Using relationships between joint and task spaces for static wrench and for twist :
  $$\Gamma = J^T K_c J dq = K_q dq.$$
  The matrix K_q is called the *joint stiffness matrix* and is not diagonal but symmetric.
  - It determines the proportional gains of the servo loops in the joint space and the joint torque vector is given by :
  $$\Gamma = K_q(q_d - q) + K_d(\dot{q}_d - q) + \hat{G}(q)$$
  where $\hat{G}(q)$ represents gravity torque compensation and K_d can be interpreted as a damping matrix.



Impedance control

Assigning a prescribed dynamic behaviour for the robot while its effector is interacting with the environment [10] :

- Desired performance specified by a generalized dynamic impedance representing a *mass-spring-damper* system :

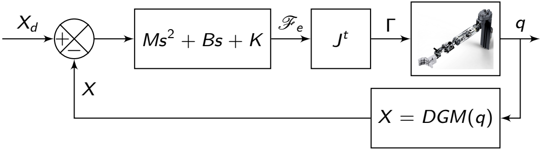
$$\mathscr{F}_e(s) = Z(s)\dot{X}(s) \qquad \text{where} \qquad sZ(s) = Ms^2 + Bs + K$$

  - M : desired inertia matrix
  - B : desired damping matrix
  - K : desired stiffness matrix
- The values of these matrices are chosen to obtain the desired performances :
  - $M \nearrow$ in the directions where a contact is expected in order to limit the dynamics of the robot ;
  - $B \nearrow$ in the directions where it is necessary to dissipate the kinetic energy and therefore to damp the response ;
  - Tuning of K : along the position controlled directions, the user should set a high stiffness to obtain an accurate positioning of the end-effector / along the interaction directions, the stiffness should be small enough to limit the contact forces ;

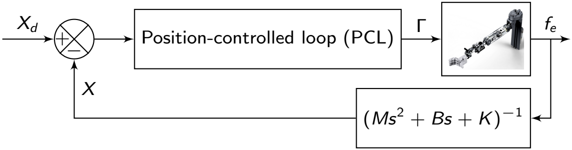
Impedance control

Possible schemes depending on whether or not a force sensor is available

1. Impedance control scheme without explicit force feedback

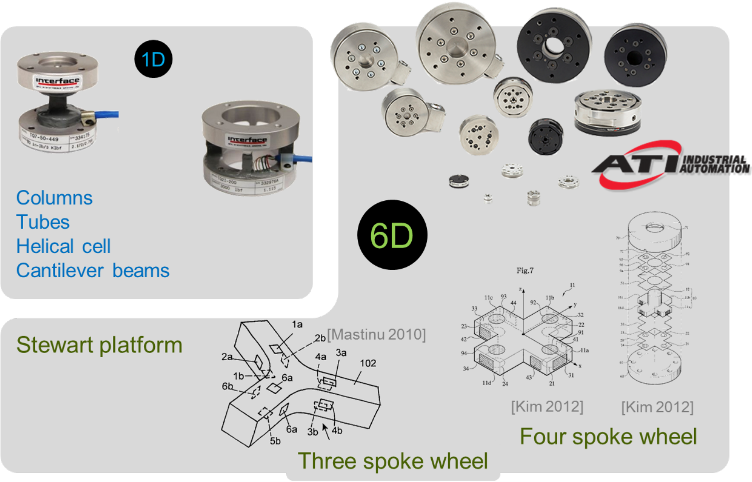


2. Impedance control scheme with explicit force feedback



Force/torque sensors mostly used in robotics

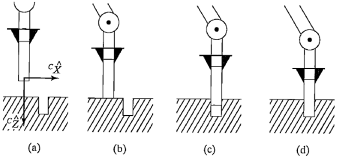
Instrumentation for measuring F/T at robot/environment interface [4]



Parallel hybrid position/force control

Situation of control satisfying simultaneously the desired position and force constraints of the task (1/2) [26]

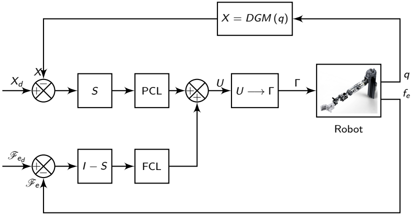
- Force/position duality for the description of the task according to Mason formulation :
  - directions that are constrained in position are force controlled ;
  - directions that are constrained in force (zero force) are position or velocity controlled.
- Robot controlled by two complementary feedback loops, one for the position, the other for the force (each having its own sensory system and control law) ;
- Example of a rivet insertion task involving hybrid force/position control :



Parallel hybrid position/force control

Situation of control satisfying simultaneously the desired position and force constraints of the task (2/2) [26]

- Diagonal matrices of selection S and $S - I$ made of 1 or 0 :
  - switching between position and force control mode according to each degree of freedom of the task space :
    - $S = diag(s_1, \ldots, s_6)$;
    - $s_i = 1$ direction controlled in force, 0 otherwise.
- Orthogonality between the position or velocity controlled directions and the force controlled directions in the compliance frame.

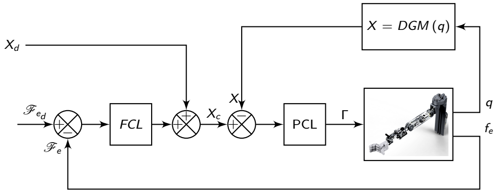


Since both loops act cooperatively, each robot joint contributes to the realization of both the position control and the force control.

External hybrid control scheme

Two embedded control loops [5] [24]

- Outer Force Control Loop (FCL) :
  - force reference $\mathscr{F}_{e_d}$ as input of the outer loop ;
  - output of the outer loop transformed into a desired position input for the inner loop.
- Inner Position Control Loop (PCL) (always active) :
  - in the case of a force applied after the force sensor at the end-effector : motion of the robot in the direction of the applied force to regulate it ;
  - in the case of a disturbance applied to the robot's kinematic chain upstream of the force sensor) : disturbance compensated by the position loop.
- Thanks to the integral force action, the wrench error $\mathscr{F}_{e_d} - \mathscr{F}_e$ is allowed to prevail over the position error $X_d - X$ at steady state.



References I

[1] D. Ardayfio.
Fundamentals of Robotics.
Mechanical Engineering. Taylor & Francis, 1987.

[2] I. Asimov.
I, Robot.
1950.

[3] N.A. Bompos, P.K. Artemiadis, A.S. Oikonomopoulos, and K.J. Kyriakopoulos.
Modeling, full identification and control of the mitsubishi pa-10 robot arm.
In *Advanced intelligent mechatronics, 2007 IEEE/ASME international conference on*, pages 1–6, 2007.

[4] Arnaud Hubert Davinson Castano-Cano, Mathieu Grossard.
Next force sensing technology for robots : multi-axis resonant sensors.
volume 9859, pages 9859 – 9859 – 9, 2016.

[5] J. De Schutter and H. van Brussel.
Compliant robot motion ii. a control approach based on external control loops.
Int. J. Rob. Res., 7(4) :18–33, 1988.

[6] Michel Fliess, Jean Lévine, Philippe Martin, and Pierre Rouchon.
Flatness and defect of nonlinear systems : introductory theory and examples.
International Journal of Control, 61 :13–27, 06 1995.

[7] A. Fournier.
Generation de mouvements en robotique : application des inverses généralisées et des pseudo-inverses.
PhD thesis. USTL, Montpellier, 1980.

[8] M. Grossard, Regnier S., and N. Chaillet.
Flexible Robotics : Applications to Multiscale Manipulations.
Wiley, 2013.

[9] B. Henze, A. Werner, M. A. Roa, G. Garofalo, J. Englsberger, and C. Ott.
Control applications of toro — a torque controlled humanoid robot.
In *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 841–841, Nov 2014.

References II

[10] N. Hogan.
Impedance control : An approach to manipulation : Part i—theory, part ii : Implementation, part iii : Applications.
Journal of Dynamic Systems, Measurement, and Control, 107 :1–24, 1985.

[11] Rafael Kelly.
Control of robot manipulators in joint space.
Advanced textbooks in control and signal processing. Springer, London, 2005.

[12] W. Khalil.
Polycopié de cours de l'Ecole Centrale de Nantes.
2005.

[13] W. Khalil and E. Dombre.
Modélisation, identification et commande des robots.
Collection robotique. Hermes Science Publications, 1999.

[14] W. Khalil and J. Kleinfinger.
A new geometric notation for open and closed-loop robots.
In *Proceedings. 1986 IEEE International Conference on Robotics and Automation*, volume 3, pages 1174–1179, Apr 1986.

[15] O. Khatib.
Real-time obstacle avoidance for manipulators and mobile robots.
In *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, volume 2, pages 500–505, Mar 1985.

[16] O. Khatib.
A unified approach for motion and force control of robot manipulators : The operational space formulation.
IEEE Journal on Robotics and Automation, 3(1) :43–53, February 1987.

[17] M. Makarov.
Contribution à la modélisation et la commande robuste de robots manipulateurs à articulations flexibles - Applications à la robotique interactive.
Thèse de Doctorat de l'Ecole Supélec, 2013.

[18] M. Makarov, M. Grossard, P. Rodríguez-Ayerbe, and D. Dumur.
Modeling and preview h infinity control design for motion control of elastic-joint robots with uncertainties.
IEEE Transactions on Industrial Electronics, 63(10) :6429–6438, Oct 2016.

References III

[19] M. Makarov, M. Grossard, P. Rodriguez-Ayerbe, and D. Dumur.
Generalized predictive control of an anthropomorphic robot arm for trajectory tracking.
In *Advanced Intelligent Mechatronics (AIM), 2011 IEEE/ASME International Conference on*, pages 948–953, 2011.

[20] J. Martin and M. Grossard.
Design of a fully modular and backdrivable dexterous hand.
The International Journal of Robotics Research, 33(5) :783–798, 2014.

[21] Richard M. Murray, S. Shankar Sastry, and Li Zexiang.
A Mathematical Introduction to Robotic Manipulation.
CRC Press, Inc., Boca Raton, FL, USA, 1st edition, 1994.

[22] Chaillet N. and Laurent G.
Robotique industrielle, polycopié de cours de l'UBFC et ENSMM.
2005.

[23] R.P. Paul.
Robot Manipulators : Mathematics, Programming, and Control : the Computer Control of Robot Manipulators.
Artificial Intelligence Series. MIT Press, 1981.

[24] V. Perdereau and M. Drouin.
A new scheme for hybrid force-position control.
Robotica, 11 :453–464, 1993.

[25] Lee Donald Pieper.
The kinematics of manipulators under computer control.
09 2017.

[26] M.H. Raibert and J.J. Craig.
Hybrid position/force control of manipulators.
Journal of Dynamic Systems, Measurement and Control, 103(2) :126–133, 1981.

[27] J. K. Salisbury.
Active stiffness control of a manipulator in cartesian coordinates.
In *1980 19th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes*, pages 95–100, 1980.

References IV

[28] Jean-Jacques E. Slotine and H. Asada.
Robot Analysis and Control.
John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1992.

[29] C. W. Wampler.
Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods.
IEEE Transactions on Systems, Man, and Cybernetics, 16(1) :93–101, Jan 1986.

[30] Tsuneo Yoshikawa.
Manipulability of robotic mechanisms.
The International Journal of Robotics Research, 4(2) :3–9, 1985.

[31] I. Zambettakis and F. Rotella.
Commande des systèmes par platitude.
Dossier Techniques de l'Ingénieur [S 7450], 2007.

Mobility Index and number of dof of a kinematical chain

- Number of links : $n_c = 5$

Mobility Index and number of dof of a kinematical chain

- Number of links : $n_c = 5$
- Number of joints : $j = \underbrace{6}_{\text{Revolute joints}}$

Mobility Index and number of dof of a kinematical chain

- Number of links : $n_c = 5$
- Number of joints : $j = \underbrace{6}_{\text{Revolute joints}}$
- Dof of joints : $\sum_{i=1}^{j} f_i = 6 \times 1 = 6$

Mobility Index and number of dof of a kinematical chain

- Number of links : $n_c = 5$
- Number of joints : $j = \underbrace{6}_{\text{Revolute joints}}$
- Dof of joints : $\sum_{i=1}^{j} f_i = 6 \times 1 = 6$

In total, $dof = 3 \times (5 - 6) + 6 = 3.$

Mobility Index and number of dof of a kinematical chain

- Number of links : $n_c = 5$
- Number of joints : $j = \underbrace{6}_{\text{Revolute joints}}$
- Dof of joints : $\sum_{i=1}^{j} f_i = 6 \times 1 = 6$

In total, $dof = 3 \times (5 - 6) + 6 = 3$.
During the double support phase of the bipedal walking, it is possible for human with these assumptions to reach specific position and orientation of its body in the plane.

Course

Mobility Index and number of dof of a kinematical chain

- Number of links : $n_c = \underbrace{(6 \times 2)}_{\text{6 actuators (cylinders)}} + \underbrace{(1)}_{\text{Mobile plateform}} = 13$

Mobility Index and number of dof of a kinematical chain

- Number of links : $n_c = \underbrace{(6 \times 2)}_{\text{6 actuators (cylinders)}} + \underbrace{(1)}_{\text{Mobile plateform}} = 13$
- Number of joints : $j = \underbrace{(6)}_{\text{Spherical joints}} + \underbrace{(6)}_{\text{Prismatic joints}} + \underbrace{(6)}_{\text{Universal (said "cardan") joints}} = 18$

Mobility Index and number of dof of a kinematical chain

- Number of links : $n_c = \underbrace{(6 \times 2)}_{\text{6 actuators (cylinders)}} + \underbrace{(1)}_{\text{Mobile plateform}} = 13$
- Number of joints : $j = \underbrace{(6)}_{\text{Spherical joints}} + \underbrace{(6)}_{\text{Prismatic joints}} + \underbrace{(6)}_{\text{Universal (said "cardan") joints}} = 18$
- Dof of joints : $\sum_{i=1}^{j} f_i = \underbrace{(6 \times 3)}_{\text{6 Spherical joints}} + \underbrace{(6 \times 1)}_{\text{6 Prismatic joints}} + \underbrace{(6 \times 2)}_{\text{6 Universal joints}} = 36$

Mobility Index and number of dof of a kinematical chain

- Number of links : $n_c = \underbrace{(6 \times 2)}_{\text{6 actuators (cylinders)}} + \underbrace{(1)}_{\text{Mobile plateform}} = 13$
- Number of joints : $j = \underbrace{(6)}_{\text{Spherical joints}} + \underbrace{(6)}_{\text{Prismatic joints}} + \underbrace{(6)}_{\text{Universal (said "cardan") joints}} = 18$
- Dof of joints : $\sum_{i=1}^{j} f_i = \underbrace{(6 \times 3)}_{\text{6 Spherical joints}} + \underbrace{(6 \times 1)}_{\text{6 Prismatic joints}} + \underbrace{(6 \times 2)}_{\text{6 Universal joints}} = 36$

In total, $dof = 6 \times (13 - 18) + 36 = 6$ needed independent variables to describe and control the kinematical chain.

Mobility Index and number of dof of a kinematical chain

- Number of links : $n_c = \underbrace{(6 \times 2)}_{\text{6 actuators (cylinders)}} + \underbrace{(1)}_{\text{Mobile plateform}} = 13$
- Number of joints : $j = \underbrace{(6)}_{\text{Spherical joints}} + \underbrace{(6)}_{\text{Prismatic joints}} + \underbrace{(6)}_{\text{Universal (said "cardan") joints}} = 18$
- Dof of joints : $\sum_{i=1}^{j} f_i = \underbrace{(6 \times 3)}_{\text{6 Spherical joints}} + \underbrace{(6 \times 1)}_{\text{6 Prismatic joints}} + \underbrace{(6 \times 2)}_{\text{6 Universal joints}} = 36$

In total, $dof = 6 \times (13 - 18) + 36 = 6$ needed independent variables to describe and control the kinematical chain.
In practice, it corresponds to the controlled elongations of the 6 cylindrical actuators.

Course

Mobility Index and number of dof of a kinematical chain

- Number of links : $n_c = 4$

Mobility Index and number of dof of a kinematical chain

- Number of links : $n_c = 4$
- Number of joints : $j = \underbrace{3}_{\text{Revolute joints}} + \underbrace{1}_{\text{Prismatic joints}} = 4$

Mobility Index and number of dof of a kinematical chain

- Number of links : $n_c = 4$
- Number of joints : $j = \underbrace{3}_{\text{Revolute joints}} + \underbrace{1}_{\text{Prismatic joints}} = 4$
- Dof of joints : $\sum_{i=1}^{j} f_i = 4 \times 1 = 4$

Mobility Index and number of dof of a kinematical chain

- Number of links : $n_c = 4$
- Number of joints : $j = \underbrace{3}_{\text{Revolute joints}} + \underbrace{1}_{\text{Prismatic joints}} = 4$
- Dof of joints : $\sum_{i=1}^{j} f_i = 4 \times 1 = 4$

In total, $dof = 6 \times (4 - 4) + 4 = 4$ needed independent variables to describe and control the kinematical chain.

Mobility Index and number of dof of a kinematical chain

- Number of links : $n_c = 4$
- Number of joints : $j = \underbrace{3}_{\text{Revolute joints}} + \underbrace{1}_{\text{Prismatic joints}} = 4$
- Dof of joints : $\sum_{i=1}^{j} f_i = 4 \times 1 = 4$

In total, $dof = 6 \times (4 - 4) + 4 = 4$ needed independent variables to describe and control the kinematical chain.
In practice, in a simple open-tree structure $dof = j$, the number of joints (assumed to be *simple*, i.e. $f_i = 1$) constitutes the number of degrees of freedom of the kinematic chain : $dof = \sum_{i=1}^{j} f_i$.

Course

Workspace of a planar RR mechanism with two links

- Evaluation of swept area :

$$\mathscr{A} = \iint dx dy$$

Workspace of a planar RR mechanism with two links

- Evaluation of swept area :

$$\mathscr{A} = \iint dxdy$$

  - Variables change :

$$\mathscr{A} = \iint dxdy = \iint |det\,(J)|\; dq_1 dq_2$$

where $J\,(q_1\,, q_2)$ is the Jacobian coordinate change matrix.

Workspace of a planar RR mechanism with two links

- Evaluation of swept area :

$$\mathscr{A} = \iint dxdy$$

  - Variables change :

$$\mathscr{A} = \iint dxdy = \iint |det\,(J)|\; dq_1 dq_2$$

    where $J\,(q_1, q_2)$ is the Jacobian coordinate change matrix.
  - Direct geometric model :

$$\begin{cases} x = l_1 cos\,(q_1) + l_2 cos\,(q_1 + q_2) \\ y = l_1 sin\,(q_1) + l_2 sin\,(q_1 + q_2) \end{cases}$$

Workspace of a planar RR mechanism with two links

- Evaluation of swept area :

$$\mathscr{A} = \iint dxdy$$

  - Variables change :

$$\mathscr{A} = \iint dxdy = \iint \left|det\left(J\right)\right| dq_1 dq_2$$

    where $J\left(q_1, q_2\right)$ is the Jacobian coordinate change matrix.
  - Direct geometric model :

$$\begin{cases} x = l_1 cos\left(q_1\right) + l_2 cos\left(q_1 + q_2\right) \\ y = l_1 sin\left(q_1\right) + l_2 sin\left(q_1 + q_2\right) \end{cases}$$

  - Calculation of the Jacobian matrix :

$$J\left(q_1, q_2\right) = \begin{bmatrix} \frac{\partial x}{\partial q_1} & \frac{\partial x}{\partial q_2} \\ \frac{\partial y}{\partial q_1} & \frac{\partial y}{\partial q_2} \end{bmatrix} = \begin{bmatrix} -l_1 sin\left(q_1\right) - l_2 sin\left(q_1 + q_2\right) & -l_2 sin\left(q_1 + q_2\right) \\ l_1 cos\left(q_1\right) + l_2 cos\left(q_1 + q_2\right) & l_2 cos\left(q_1 + q_2\right) \end{bmatrix}$$

Workspace of a planar RR mechanism with two links

- Evaluation of swept area :

$$\mathscr{A} = \iint dxdy$$

  - Variables change :

$$\mathscr{A} = \iint dxdy = \iint |det\left(J\right)| \, dq_1 dq_2$$

  where $J\left(q_1, q_2\right)$ is the Jacobian coordinate change matrix.

  - Direct geometric model :

$$\begin{cases} x = l_1 cos\left(q_1\right) + l_2 cos\left(q_1 + q_2\right) \\ y = l_1 sin\left(q_1\right) + l_2 sin\left(q_1 + q_2\right) \end{cases}$$

  - Calculation of the Jacobian matrix :

$$J\left(q_1, q_2\right) = \begin{bmatrix} \frac{\partial x}{\partial q_1} & \frac{\partial x}{\partial q_2} \\ \frac{\partial y}{\partial q_1} & \frac{\partial y}{\partial q_2} \end{bmatrix} = \begin{bmatrix} -l_1 sin\left(q_1\right) - l_2 sin\left(q_1 + q_2\right) & -l_2 sin\left(q_1 + q_2\right) \\ l_1 cos\left(q_1\right) + l_2 cos\left(q_1 + q_2\right) & l_2 cos\left(q_1 + q_2\right) \end{bmatrix}$$

  - Calculation of the Jacobian matrix determinant :

$$det\left(J\right) = l_1 l_2 sin\left(q_2\right)$$

Workspace of a planar RR mechanism with two links

- Evaluation of swept area :

$$\mathscr{A} = \iint dxdy$$

- Variables change :

$$\mathscr{A} = \iint dxdy = \iint \left|det\left(J\right)\right| dq_1 dq_2$$

where $J\left(q_1, q_2\right)$ is the Jacobian coordinate change matrix.

- Direct geometric model :

$$\begin{cases} x = l_1 cos\left(q_1\right) + l_2 cos\left(q_1 + q_2\right) \\ y = l_1 sin\left(q_1\right) + l_2 sin\left(q_1 + q_2\right) \end{cases}$$

- Calculation of the Jacobian matrix :

$$J\left(q_1, q_2\right) = \begin{bmatrix} \frac{\partial x}{\partial q_1} & \frac{\partial x}{\partial q_2} \\ \frac{\partial y}{\partial q_1} & \frac{\partial y}{\partial q_2} \end{bmatrix} = \begin{bmatrix} -l_1 sin\left(q_1\right) - l_2 sin\left(q_1 + q_2\right) & -l_2 sin\left(q_1 + q_2\right) \\ l_1 cos\left(q_1\right) + l_2 cos\left(q_1 + q_2\right) & l_2 cos\left(q_1 + q_2\right) \end{bmatrix}$$

- Calculation of the Jacobian matrix determinant :

$$det\left(J\right) = l_1 l_2 sin\left(q_2\right)$$

- Expression of swept area :

$$\mathscr{A} = l_1 l_2 \left({q_1}_{max} - {q_1}_{min}\right) \int_{{q_2}_{min}}^{{q_2}_{max}} \left|sin\left(q_2\right)\right| dq_2$$

Workspace of a planar RR mechanism with two links

Optimization of the lengths ratio

Workspace of a planar RR mechanism with two links

Optimization of the lengths ratio

- Maximum elongation $L = l_1 + l_2$ being specified, the lengths ratio $\lambda = \frac{l_1}{l_2}$ will maximize the domain

$$\mathscr{A} = \frac{\lambda}{(1+\lambda)^2} L^2 \left(q_{1_{\max}} - q_{1_{\min}}\right) \int_{q_{2_{\min}}}^{q_{2_{\max}}} \left|sin\left(q_2\right)\right| dq_2$$

Workspace of a planar RR mechanism with two links

Optimization of the lengths ratio

- Maximum elongation $L = l_1 + l_2$ being specified, the lengths ratio $\lambda = \frac{l_1}{l_2}$ will maximize the domain

$$\mathscr{A} = \frac{\lambda}{(1+\lambda)^2} L^2 \left(q_{1_{max}} - q_{1_{min}}\right) \int_{q_{2_{min}}}^{q_{2_{max}}} |sin\left(q_2\right)| dq_2$$

iff

$$\frac{\partial \mathscr{A}}{\partial \lambda} = 0 \quad \text{et} \quad \frac{\partial^2 \mathscr{A}}{\partial \lambda^2} < 0$$

Workspace of a planar RR mechanism with two links

Optimization of the lengths ratio

- Maximum elongation $L = l_1 + l_2$ being specified, the lengths ratio $\lambda = \frac{l_1}{l_2}$ will maximize the domain

$$\mathscr{A} = \frac{\lambda}{(1+\lambda)^2} L^2 \left(q_{1_{max}} - q_{1_{min}}\right) \int_{q_{2_{min}}}^{q_{2_{max}}} |sin\,(q_2)\,|dq_2$$

iff

$$\frac{\partial \mathscr{A}}{\partial \lambda} = 0 \qquad \text{et} \qquad \frac{\partial^2 \mathscr{A}}{\partial \lambda^2} < 0$$

- Now, it appears that :

$$\frac{\partial \mathscr{A}}{\partial \lambda} = \frac{1-\lambda}{(1+\lambda)^3} L^2 \left(q_{1_{max}} - q_{1_{min}}\right) \int_{q_{2_{min}}}^{q_{2_{max}}} |sin\,(q_2)\,|dq_2 = 0 \qquad \text{pour} \qquad \lambda = 1$$

et

$$\frac{\partial^2 \mathscr{A}}{\partial \lambda^2} = \frac{2\,(\lambda-2)}{(1+\lambda)^4} L^2 \left(q_{1_{max}} - q_{1_{min}}\right) \int_{q_{2_{min}}}^{q_{2_{max}}} |sin\,(q_2)\,|dq_2 < 0 \qquad \text{pour} \qquad \lambda = 1$$

Workspace of a planar RR mechanism with two links

Optimization of the lengths ratio

- Maximum elongation $L = l_1 + l_2$ being specified, the lengths ratio $\lambda = \frac{l_1}{l_2}$ will maximize the domain

$$\mathscr{A} = \frac{\lambda}{(1+\lambda)^2} L^2 \left(q_{1_{\max}} - q_{1_{\min}}\right) \int_{q_{2_{\min}}}^{q_{2_{\max}}} |sin\,(q_2)\,| dq_2$$

iff

$$\frac{\partial \mathscr{A}}{\partial \lambda} = 0 \quad \text{et} \quad \frac{\partial^2 \mathscr{A}}{\partial \lambda^2} < 0$$

- Now, it appears that :

$$\frac{\partial \mathscr{A}}{\partial \lambda} = \frac{1-\lambda}{(1+\lambda)^3} L^2 \left(q_{1_{\max}} - q_{1_{\min}}\right) \int_{q_{2_{\min}}}^{q_{2_{\max}}} |sin\,(q_2)\,| dq_2 = 0 \quad \text{pour} \quad \lambda = 1$$

et

$$\frac{\partial^2 \mathscr{A}}{\partial \lambda^2} = \frac{2\,(\lambda-2)}{(1+\lambda)^4} L^2 \left(q_{1_{\max}} - q_{1_{\min}}\right) \int_{q_{2_{\min}}}^{q_{2_{\max}}} |sin\,(q_2)\,| dq_2 < 0 \quad \text{pour} \quad \lambda = 1$$

- Therefore, the workspace is maximal when the two links have identical lengths.

Workspace of a planar RR mechanism with two links

Searching for the optimal joint angle q_2 maximizing the variation rate of the workspace :

Workspace of a planar RR mechanism with two links

Searching for the optimal joint angle q_2 maximizing the variation rate of the workspace :
The integrande $l_1 l_2 \left(q_{1_{\max}} - q_{1_{\min}}\right) |sin\left(q_2\right)|$ represents the variation rate of the workspace with respect to q_2.

- it is zero if $q_2 = 0$;
- it reaches its maximum when $q_2 = \pm\frac{\pi}{2}$.

Workspace of a planar RR mechanism with two links

Searching for the optimal joint angle q_2 maximizing the variation rate of the workspace :
The integrande $l_1 l_2 \left(q_{1_{\max}} - q_{1_{\min}}\right) |sin\left(q_2\right)|$ represents the variation rate of the workspace with respect to q_2.

- it is zero if $q_2 = 0$;
- it reaches its maximum when $q_2 = \pm\frac{\pi}{2}$.

Conclusion :

- This means that an angular variation of q_2 centered around $\pm\frac{\pi}{2}$ allows maximum scanning of the workspace.
- This is confirmed by the table given in the course for the special case where $q_{1_{\max}} - q_{1_{\min}} = 90^\circ$ and $l_2 = l_1 = 1$.
- It is interesting to observe that the case of the human arm meets these conditions.

Course

Rotational motion in $\mathbb{R}^3$

Properties of rotation matrices inherited from groupe structure $SO(3)$ endowed with the matrix multiplication operator :

- closure : if $R_1,\ R_2 \in SO\,(3)$, then $R_1 R_2$ is also a rotation matrix, since :
  - $R_1 R_2 \left(R_1 R_2\right)^t = R_1 R_2 R_2^t R_1^t = R_1 R_1^t = Id$;
  - $det\left(R_1 R_2\right) = det\left(R_1\right) det\left(R_2\right) = +1$.
- identity element existence given by the identity matrix.
- inverse element existence : the inverse of $R \in SO\,(3)$ is unique and equal to $R^t \in SO\,(3)$.
- associativity : it naturally comes from the associativity of matrix multiplications : $\left(R_1 R_2\right) R_3 = R_1 \left(R_2 R_3\right)$.

Course

Rotation about an arbitrary axis : general case

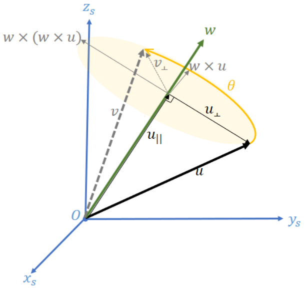
Let $w = [w_x, w_y, w_z]^t$ the unit vector defining the rotation of angle θ.

- The vector u can be decomposed along its parallel and its perpendicular components w.r.t. w : $u = u_{||} + u_{\perp}$
  where the parallel component is the orthogonal projection of u on w :

$$u_{||} = (u \cdot w)\, w$$

  and the perpendicular component to w is :

$$u_{\perp} = u - u_{||} = u - (u \cdot w)\, w = -w \times (w \times u)$$



Rotation about an arbitrary axis : general case

- The parallel component $u_{||}$ is invariant by the rotation operation :

$$v_{||} = u_{||}$$

and only the perpendicular component $u_\perp$ has direction change (but with constant amplitude), according to :

$$\begin{aligned} |v_\perp| &= |u_\perp| \\ v_\perp &= cos(\theta)u_\perp + sin(\theta)w \times u \end{aligned}$$

Let note that $w \times u_\perp = w \times \left(u - u_{||}\right) = w \times u$.

- Finally,

$$\begin{aligned} v &= v_{||} + v_\perp \\ &= u_{||} + cos(\theta)u_\perp + sin(\theta)w \times u \\ &= u_{||} + cos(\theta)\left(u - u_{||}\right) + sin(\theta)w \times u \\ &= cos(\theta)u + (1 - cos(\theta))\, u_{||} + sin(\theta)w \times u \\ &= cos(\theta)u + (1 - cos(\theta))\,(u \cdot w)\, w + sin(\theta)w \times u \end{aligned}$$

Rotation about an arbitrary axis : general case

- Substituting the cross product $w \times u$ by the associated skew-symmetric matrix $\widehat{w}u$ into the *Rodrigues* formula :

$$
\begin{aligned}
v &= cos(\theta)u + (1 - cos(\theta))\,(u \cdot w)\,w + sin(\theta)w \times u \\
&= cos(\theta)u + (1 - cos(\theta))\,(u + w \times (w \times u)) + sin(\theta)w \times u \\
&= cos(\theta)u + (1 - cos(\theta))\left(I_{3\times 3} + \widehat{w}^2\right) u + sin(\theta)\widehat{w}u \\
&= \underbrace{\left[cos(\theta)I_{3\times 3} + (1 - cos(\theta))\left(I_{3\times 3} + \widehat{w}^2\right) + sin(\theta)\widehat{w}\right]}_{=R_{w,\theta}} u
\end{aligned}
$$

- Let note that :

$$
\begin{aligned}
I_{3\times 3} + \widehat{w}^2 &= \begin{bmatrix} 1 - w_z^2 - w_y^2 & w_x w_y & w_x w_z \\ w_y w_x & 1 - w_x^2 - w_z^2 & w_y w_z \\ w_z w_x & w_z w_y & 1 - w_x^2 - w_y^2 \end{bmatrix} \\
&= \begin{bmatrix} w_x^2 & w_x w_y & w_x w_z \\ w_y w_x & w_y^2 & w_y w_z \\ w_z w_x & w_z w_y & w_z^2 \end{bmatrix} \quad \text{since vector } w \text{ is unitary} \\
&= ww^t
\end{aligned}
$$

- We find both matrix forms given by *Rodrigues* formula :

$R_{w,\theta} = cos(\theta)I_{3\times 3} + (1 - cos(\theta))\,ww^t + sin(\theta)\widehat{w}$ *or* $R_{w,\theta} = I_{3\times 3} + (1 - cos(\theta))\,\widehat{w}^2 + sin(\theta)\widehat{w}$

Inverse relationship : demonstration (1/2)

- The unit vector w itself is not affected by the rotation operation, neither in orientation nor in length. It follows that :

  $$Rw = w$$

  This relationship indicates that the vector w is an eigenvector of matrix R, associated with the eigenvalue equal to $\lambda_1 = 1$.

Inverse relationship : demonstration (1/2)

- The unit vector w itself is not affected by the rotation operation, neither in orientation nor in length. It follows that :

$$Rw = w$$

This relationship indicates that the vector w is an eigenvector of matrix R, associated with the eigenvalue equal to $\lambda_1 = 1$.

- The matrix R being of size 3×3, it possesses two another eigenvalues λ_2 and λ_3. Based on the invariance of the determinant, the matrix determinant is equal to the product of its eigenvalues :

$$det\,(R) = \lambda_1 \lambda_2 \lambda_3 = +1 \qquad (\text{since } R \in SO\,(3))$$

Inverse relationship : demonstration (1/2)

- The unit vector w itself is not affected by the rotation operation, neither in orientation nor in length. It follows that :

$$Rw = w$$

  This relationship indicates that the vector w is an eigenvector of matrix R, associated with the eigenvalue equal to $\lambda_1 = 1$.
- The matrix R being of size 3×3, it possesses two another eigenvalues λ_2 and λ_3. Based on the invariance of the determinant, the matrix determinant is equal to the product of its eigenvalues :

$$det\,(R) = \lambda_1 \lambda_2 \lambda_3 = +1 \qquad (\text{since } R \in SO\,(3))$$

  Since $\lambda_1 = 1$, the two other eigenvalues must be complex conjugates and with unitary modules ; thus, they are of the following forms $\lambda_2 = e^{i\phi}$ and $\lambda_3 = e^{-i\phi}$.

Inverse relationship : demonstration (1/2)

- The unit vector w itself is not affected by the rotation operation, neither in orientation nor in length. It follows that :

$$Rw = w$$

  This relationship indicates that the vector w is an eigenvector of matrix R, associated with the eigenvalue equal to $\lambda_1 = 1$.
- The matrix R being of size 3×3, it possesses two another eigenvalues λ_2 and λ_3. Based on the invariance of the determinant, the matrix determinant is equal to the product of its eigenvalues :

$$det\,(R) = \lambda_1\lambda_2\lambda_3 = +1 \qquad (\text{since } R \in SO\,(3))$$

  Since $\lambda_1 = 1$, the two other eigenvalues must be complex conjugates and with unitary modules ; thus, they are of the following forms $\lambda_2 = e^{i\phi}$ and $\lambda_3 = e^{-i\phi}$.
- On the one hand, the trace of a matrix being equal to the sum of its eigenvalues, it follows :

$$Tr\,(R) = \lambda_1 + \lambda_2 + \lambda_3 = 1 + 2C\phi$$

Inverse relationship : demonstration (1/2)

- The unit vector w itself is not affected by the rotation operation, neither in orientation nor in length. It follows that :

$$Rw = w$$

This relationship indicates that the vector w is an eigenvector of matrix R, associated with the eigenvalue equal to $\lambda_1 = 1$.

- The matrix R being of size 3×3, it possesses two another eigenvalues λ_2 and λ_3. Based on the invariance of the determinant, the matrix determinant is equal to the product of its eigenvalues :

$$det\,(R) = \lambda_1 \lambda_2 \lambda_3 = +1 \qquad (\text{since } R \in SO\,(3))$$

Since $\lambda_1 = 1$, the two other eigenvalues must be complex conjugates and with unitary modules ; thus, they are of the following forms $\lambda_2 = e^{i\phi}$ and $\lambda_3 = e^{-i\phi}$.

- On the one hand, the trace of a matrix being equal to the sum of its eigenvalues, it follows :

$$Tr\,(R) = \lambda_1 + \lambda_2 + \lambda_3 = 1 + 2C\phi$$

On the other hand, the value of the angle ϕ that appears in λ_2 and λ_3 can be easily determined in observing in the *Rodrigues* formula that the trace of R is equal to :

$$Tr\,(R) = 1 + 2C\theta \qquad (\text{since } w \text{ is unitary})$$

Inverse relationship : demonstration (2/2)

- It comes that the angle ϕ that defines the eigenvalues λ_2 and λ_3 is exactly the rotation angle θ. To sum up, the eigenvalues of R are

$$\lambda_1 = 1, \quad \lambda_2 = e^{i\theta}, \quad \lambda_3 = e^{-i\theta}.$$

where θ is the rotation angle around a direction that is solution of $Rw = w$.

Inverse relationship : demonstration (2/2)

- It comes that the angle ϕ that defines the eigenvalues λ_2 and λ_3 is exactly the rotation angle θ. To sum up, the eigenvalues of R are

$$\lambda_1 = 1, \quad \lambda_2 = e^{i\theta}, \quad \lambda_3 = e^{-i\theta}.$$

  where θ is the rotation angle around a direction that is solution of $Rw = w$.
- In practice, some explicit relationships can be deduced in comparing the components of R with the form given by *Rodrigues* formula :
  - Computation of the trace :

$$r_{11} + r_{22} + r_{33} = 1 + 2C\theta$$

Inverse relationship : demonstration (2/2)

- It comes that the angle ϕ that defines the eigenvalues λ_2 and λ_3 is exactly the rotation angle θ. To sum up, the eigenvalues of R are

$$\lambda_1 = 1, \quad \lambda_2 = e^{i\theta}, \quad \lambda_3 = e^{-i\theta}.$$

  where θ is the rotation angle around a direction that is solution of $Rw = w$.
- In practice, some explicit relationships can be deduced in comparing the components of R with the form given by *Rodrigues* formula :
  - Computation of the trace :

$$r_{11} + r_{22} + r_{33} = 1 + 2C\theta$$

  - Computation of the extra diagonal terms :

$$\begin{aligned} r_{32} - r_{23} &= 2w_x S\theta \\ r_{13} - r_{31} &= 2w_y S\theta \\ r_{21} - r_{12} &= 2w_z S\theta \end{aligned}$$

Inverse relationship : demonstration (2/2)

- It comes that the angle ϕ that defines the eigenvalues λ_2 and λ_3 is exactly the rotation angle θ. To sum up, the eigenvalues of R are
$$\lambda_1 = 1, \quad \lambda_2 = e^{i\theta}, \quad \lambda_3 = e^{-i\theta}.$$
where θ is the rotation angle around a direction that is solution of $Rw = w$.
- In practice, some explicit relationships can be deduced in comparing the components of R with the form given by *Rodrigues* formula :
  - Computation of the trace :
  $$r_{11} + r_{22} + r_{33} = 1 + 2C\theta$$
  - Computation of the extra diagonal terms :
  $$\begin{aligned} r_{32} - r_{23} &= 2w_x S\theta \\ r_{13} - r_{31} &= 2w_y S\theta \\ r_{21} - r_{12} &= 2w_z S\theta \end{aligned}$$
- Since w is unitary, the angle θ is obtained thanks to the previous equations (in this case $\theta \neq 0$) :
$$\begin{aligned} \theta &= atan2\left(S\theta, C\theta\right) \\ &= atan2\left(\frac{1}{2}\sqrt{(r_{32} - r_{23})^2 + (r_{13} - r_{31})^2 + (r_{21} - r_{12})^2}, \frac{1}{2}\left(r_{11} + r_{22} + r_{33} - 1\right)\right) \end{aligned}$$

Inverse relationship : demonstration (2/2)

- It comes that the angle ϕ that defines the eigenvalues λ_2 and λ_3 is exactly the rotation angle θ. To sum up, the eigenvalues of R are

$$\lambda_1 = 1, \quad \lambda_2 = e^{i\theta}, \quad \lambda_3 = e^{-i\theta}.$$

where θ is the rotation angle around a direction that is solution of $Rw = w$.
- In practice, some explicit relationships can be deduced in comparing the components of R with the form given by *Rodrigues* formula :
  - Computation of the trace :

$$r_{11} + r_{22} + r_{33} = 1 + 2C\theta$$

  - Computation of the extra diagonal terms :

$$\begin{aligned} r_{32} - r_{23} &= 2w_x S\theta \\ r_{13} - r_{31} &= 2w_y S\theta \\ r_{21} - r_{12} &= 2w_z S\theta \end{aligned}$$

- Since w is unitary, the angle θ is obtained thanks to the previous equations (in this case $\theta \neq 0$) :

$$\begin{aligned} \theta &= atan2\,(S\theta, C\theta) \\ &= atan2\left(\frac{1}{2}\sqrt{(r_{32} - r_{23})^2 + (r_{13} - r_{31})^2 + (r_{21} - r_{12})^2}, \frac{1}{2}\,(r_{11} + r_{22} + r_{33} - 1)\right) \end{aligned}$$

and the vector w is given by :

$$w = \begin{bmatrix} \frac{r_{32} - r_{23}}{2S\theta} & \frac{r_{13} - r_{31}}{2S\theta} & \frac{r_{21} - r_{12}}{2S\theta} \end{bmatrix}^t \quad \text{if } S\theta \neq 0$$

Retrieving the axis and angle parameters from a given rotation matrix

- We know that : $R_{x,-45^\circ} R_{w,\theta} = R_{y,45^\circ} R_{z,90^\circ}$

Retrieving the axis and angle parameters from a given rotation matrix

- We know that : $R_{x,-45^\circ} R_{w,\theta} = R_{y,45^\circ} R_{z,90^\circ}$, thus :

$$R_{w,\theta} = R_{x,45^\circ} R_{y,45^\circ} R_{z,90^\circ}$$

Retrieving the axis and angle parameters from a given rotation matrix

- We know that : $R_{x,-45^\circ} R_{w,\theta} = R_{y,45^\circ} R_{z,90^\circ}$, thus :

$$R_{w,\theta} = R_{x,45^\circ} R_{y,45^\circ} R_{z,90^\circ}$$

In more details,

$$R_{w,\theta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\left(\frac{\pi}{4}\right) & -sin\left(\frac{\pi}{4}\right) \\ 0 & sin\left(\frac{\pi}{4}\right) & cos\left(\frac{\pi}{4}\right) \end{bmatrix} \begin{bmatrix} cos\left(\frac{\pi}{4}\right) & 0 & sin\left(\frac{\pi}{4}\right) \\ 0 & 1 & 0 \\ -sin\left(\frac{\pi}{4}\right) & 0 & cos\left(\frac{\pi}{4}\right) \end{bmatrix} \begin{bmatrix} cos\left(\frac{\pi}{2}\right) & -sin\left(\frac{\pi}{2}\right) & 0 \\ sin\left(\frac{\pi}{2}\right) & cos\left(\frac{\pi}{2}\right) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{\sqrt{2}} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

Retrieving the axis and angle parameters from a given rotation matrix

- We know that : $R_{x,-45^\circ} R_{w,\theta} = R_{y,45^\circ} R_{z,90^\circ}$, thus :

$$R_{w,\theta} = R_{x,45^\circ} R_{y,45^\circ} R_{z,90^\circ}$$

In more details,

$$R_{w,\theta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\left(\frac{\pi}{4}\right) & -sin\left(\frac{\pi}{4}\right) \\ 0 & sin\left(\frac{\pi}{4}\right) & cos\left(\frac{\pi}{4}\right) \end{bmatrix} \begin{bmatrix} cos\left(\frac{\pi}{4}\right) & 0 & sin\left(\frac{\pi}{4}\right) \\ 0 & 1 & 0 \\ -sin\left(\frac{\pi}{4}\right) & 0 & cos\left(\frac{\pi}{4}\right) \end{bmatrix} \begin{bmatrix} cos\left(\frac{\pi}{2}\right) & -sin\left(\frac{\pi}{2}\right) & 0 \\ sin\left(\frac{\pi}{2}\right) & cos\left(\frac{\pi}{2}\right) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{\sqrt{2}} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

- Searching for the angle based on the formula given in the course :

Retrieving the axis and angle parameters from a given rotation matrix

- We know that : $R_{x,-45^\circ} R_{w,\theta} = R_{y,45^\circ} R_{z,90^\circ}$, thus :

$$R_{w,\theta} = R_{x,45^\circ} R_{y,45^\circ} R_{z,90^\circ}$$

In more details,

$$R_{w,\theta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\left(\frac{\pi}{4}\right) & -sin\left(\frac{\pi}{4}\right) \\ 0 & sin\left(\frac{\pi}{4}\right) & cos\left(\frac{\pi}{4}\right) \end{bmatrix} \begin{bmatrix} cos\left(\frac{\pi}{4}\right) & 0 & sin\left(\frac{\pi}{4}\right) \\ 0 & 1 & 0 \\ -sin\left(\frac{\pi}{4}\right) & 0 & cos\left(\frac{\pi}{4}\right) \end{bmatrix} \begin{bmatrix} cos\left(\frac{\pi}{2}\right) & -sin\left(\frac{\pi}{2}\right) & 0 \\ sin\left(\frac{\pi}{2}\right) & cos\left(\frac{\pi}{2}\right) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{\sqrt{2}} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

- Searching for the angle based on the formula given in the course :

$$\theta = atan2\,(S\theta,\, C\theta) = atan2\left(\frac{\sqrt{3}}{2}, -\frac{1}{2}\right) = 120^\circ$$

Retrieving the axis and angle parameters from a given rotation matrix

- We know that : $R_{x,-45^\circ} R_{w,\theta} = R_{y,45^\circ} R_{z,90^\circ}$, thus :

$$R_{w,\theta} = R_{x,45^\circ} R_{y,45^\circ} R_{z,90^\circ}$$

In more details,

$$R_{w,\theta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\left(\frac{\pi}{4}\right) & -sin\left(\frac{\pi}{4}\right) \\ 0 & sin\left(\frac{\pi}{4}\right) & cos\left(\frac{\pi}{4}\right) \end{bmatrix} \begin{bmatrix} cos\left(\frac{\pi}{4}\right) & 0 & sin\left(\frac{\pi}{4}\right) \\ 0 & 1 & 0 \\ -sin\left(\frac{\pi}{4}\right) & 0 & cos\left(\frac{\pi}{4}\right) \end{bmatrix} \begin{bmatrix} cos\left(\frac{\pi}{2}\right) & -sin\left(\frac{\pi}{2}\right) & 0 \\ sin\left(\frac{\pi}{2}\right) & cos\left(\frac{\pi}{2}\right) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{\sqrt{2}} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

- Searching for the angle based on the formula given in the course :

$$\theta = atan2\left(S\theta, C\theta\right) = atan2\left(\frac{\sqrt{3}}{2}, -\frac{1}{2}\right) = 120^\circ$$

We could also have searched for the eigenvalues of $R_{w,\theta}$: $\lambda_1 = 1$, $\lambda_2 = -\frac{1}{2} + i\frac{\sqrt{3}}{2} = e^{i\frac{2\pi}{3}}$ and $\lambda_3 = -\frac{1}{2} - i\frac{\sqrt{3}}{2} = e^{-i\frac{2\pi}{3}}$.

Retrieving the axis and angle parameters from a given rotation matrix

- We know that : $R_{x,-45^\circ} R_{w,\theta} = R_{y,45^\circ} R_{z,90^\circ}$, thus :

$$R_{w,\theta} = R_{x,45^\circ} R_{y,45^\circ} R_{z,90^\circ}$$

In more details,

$$R_{w,\theta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\left(\frac{\pi}{4}\right) & -sin\left(\frac{\pi}{4}\right) \\ 0 & sin\left(\frac{\pi}{4}\right) & cos\left(\frac{\pi}{4}\right) \end{bmatrix} \begin{bmatrix} cos\left(\frac{\pi}{4}\right) & 0 & sin\left(\frac{\pi}{4}\right) \\ 0 & 1 & 0 \\ -sin\left(\frac{\pi}{4}\right) & 0 & cos\left(\frac{\pi}{4}\right) \end{bmatrix} \begin{bmatrix} cos\left(\frac{\pi}{2}\right) & -sin\left(\frac{\pi}{2}\right) & 0 \\ sin\left(\frac{\pi}{2}\right) & cos\left(\frac{\pi}{2}\right) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{\sqrt{2}} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

- Searching for the angle based on the formula given in the course :

$$\theta = atan2\,(S\theta,\, C\theta) = atan2\left(\frac{\sqrt{3}}{2}, -\frac{1}{2}\right) = 120^\circ$$

We could also have searched for the eigenvalues of $R_{w,\theta}$: $\lambda_1 = 1$, $\lambda_2 = -\frac{1}{2} + i\frac{\sqrt{3}}{2} = e^{i\frac{2\pi}{3}}$ and $\lambda_3 = -\frac{1}{2} - i\frac{\sqrt{3}}{2} = e^{-i\frac{2\pi}{3}}$.

- Searching for the vector of rotation based on the formula given in the course :

Retrieving the axis and angle parameters from a given rotation matrix

- We know that : $R_{x,-45^\circ} R_{w,\theta} = R_{y,45^\circ} R_{z,90^\circ}$, thus :

$$R_{w,\theta} = R_{x,45^\circ} R_{y,45^\circ} R_{z,90^\circ}$$

In more details,

$$R_{w,\theta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\left(\frac{\pi}{4}\right) & -sin\left(\frac{\pi}{4}\right) \\ 0 & sin\left(\frac{\pi}{4}\right) & cos\left(\frac{\pi}{4}\right) \end{bmatrix} \begin{bmatrix} cos\left(\frac{\pi}{4}\right) & 0 & sin\left(\frac{\pi}{4}\right) \\ 0 & 1 & 0 \\ -sin\left(\frac{\pi}{4}\right) & 0 & cos\left(\frac{\pi}{4}\right) \end{bmatrix} \begin{bmatrix} cos\left(\frac{\pi}{2}\right) & -sin\left(\frac{\pi}{2}\right) & 0 \\ sin\left(\frac{\pi}{2}\right) & cos\left(\frac{\pi}{2}\right) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{\sqrt{2}} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

- Searching for the angle based on the formula given in the course :

$$\theta = atan2\left(S\theta, C\theta\right) = atan2\left(\frac{\sqrt{3}}{2}, -\frac{1}{2}\right) = 120^\circ$$

We could also have searched for the eigenvalues of $R_{w,\theta}$: $\lambda_1 = 1$, $\lambda_2 = -\frac{1}{2} + i\frac{\sqrt{3}}{2} = e^{i\frac{2\pi}{3}}$ and $\lambda_3 = -\frac{1}{2} - i\frac{\sqrt{3}}{2} = e^{-i\frac{2\pi}{3}}$.

- Searching for the vector of rotation based on the formula given in the course :

$$w = \begin{bmatrix} \frac{1}{\sqrt{3}} & 0 & \frac{\sqrt{2}}{\sqrt{3}} \end{bmatrix}^t$$

Retrieving the axis and angle parameters from a given rotation matrix

- We know that : $R_{x,-45^\circ} R_{w,\theta} = R_{y,45^\circ} R_{z,90^\circ}$, thus :

$$R_{w,\theta} = R_{x,45^\circ} R_{y,45^\circ} R_{z,90^\circ}$$

In more details,

$$R_{w,\theta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\left(\frac{\pi}{4}\right) & -sin\left(\frac{\pi}{4}\right) \\ 0 & sin\left(\frac{\pi}{4}\right) & cos\left(\frac{\pi}{4}\right) \end{bmatrix} \begin{bmatrix} cos\left(\frac{\pi}{4}\right) & 0 & sin\left(\frac{\pi}{4}\right) \\ 0 & 1 & 0 \\ -sin\left(\frac{\pi}{4}\right) & 0 & cos\left(\frac{\pi}{4}\right) \end{bmatrix} \begin{bmatrix} cos\left(\frac{\pi}{2}\right) & -sin\left(\frac{\pi}{2}\right) & 0 \\ sin\left(\frac{\pi}{2}\right) & cos\left(\frac{\pi}{2}\right) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{\sqrt{2}} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

- Searching for the angle based on the formula given in the course :

$$\theta = atan2\left(S\theta, C\theta\right) = atan2\left(\frac{\sqrt{3}}{2}, -\frac{1}{2}\right) = 120^\circ$$

We could also have searched for the eigenvalues of $R_{w,\theta}$: $\lambda_1 = 1$, $\lambda_2 = -\frac{1}{2} + i\frac{\sqrt{3}}{2} = e^{i\frac{2\pi}{3}}$ and $\lambda_3 = -\frac{1}{2} - i\frac{\sqrt{3}}{2} = e^{-i\frac{2\pi}{3}}$.

- Searching for the vector of rotation based on the formula given in the course :

$$w = \begin{bmatrix} \frac{1}{\sqrt{3}} & 0 & \frac{\sqrt{2}}{\sqrt{3}} \end{bmatrix}^t$$

We could also have directly searched for the eigenvector associated to the eigenvalue 1 (i.e. $Rw = w$) to determine the invariant vector w through $R_{w,\theta}$.

Exponential coordinates of rotation

Let us expand the matrix exponential $e^{\widehat{w}\theta}$ in series form.
A straightforward calculation shows that $\widehat{w}^3 = -\widehat{w}$, and therefore we can replace $\widehat{w}^3$ by $-\widehat{w}$, $\widehat{w}^4$ by $-\widehat{w}^2$, $\widehat{w}^5$ by $-\widehat{w}^3 = \widehat{w}$, and so on, obtaining :

$$\begin{aligned} e^{\widehat{w}\theta} &= I_{3\times 3} + \widehat{w}\theta + \widehat{w}^2\frac{\theta^2}{2!} + \widehat{w}^3\frac{\theta^3}{3!} + \dots \\ &= I_{3\times 3} + \left(\theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} + \dots\right)\widehat{w} + \left(\frac{\theta^2}{2!} - \frac{\theta^4}{4!} + \frac{\theta^6}{6!} + \dots\right)\widehat{w}^2 \end{aligned}$$

Recall the series expansions for $sin(\theta)$ and $cos(\theta)$:

$$\begin{aligned} cos(\theta) &= 1 - \frac{\theta^2}{2!} + \frac{\theta^4}{4!} - \dots \\ sin(\theta) &= \theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} - \dots \end{aligned}$$

The exponential $e^{\widehat{w}\theta}$ therefore simplifies to the following :

$$e^{\widehat{w}\theta} = I_{3\times 3} + (1 - cos(\theta))\,\widehat{w}^2 + sin(\theta)\widehat{w}$$

Course

Properties in $SE(3)$

1. Keeping distance between points :
   $\forall a, b \in \mathbb{R}^3$, $\|g(a) - g(b)\| = \|R(a-b)\| = \left[(a-b)^t R^t R (a-b)\right]^{\frac{1}{2}} = \|a-b\|$.

Properties in $SE(3)$

1. Keeping distance between points :
   $\forall a, b \in \mathbb{R}^3$, $\|g(a) - g(b)\| = \|R(a-b)\| = \left[(a-b)^t R^t R (a-b)\right]^{\frac{1}{2}} = \|a-b\|$.
2. Keeping orientation between vectors :
   Let note that only the action of the rotation from the rigid transformation $g = (p, R)$ applied to a vector $v = s - r$ change the coordinates of vector :

$$g_*(v) = g(s) - g(r) = R(s-r) = Rv$$

   It results that : $\forall v, w \in \mathbb{R}^3$,
   $g_*(v \times w) = R(v \times w) = (Rv) \times (Rw) = g_*(v) \times g_*(w)$, since R preserves orthogonality, orientation and distance.

Course

Properties in $SE(3)$

$SE(3)$ forms a group :

Properties in $SE(3)$

$SE(3)$ forms a group :

1. Composition for two elements of $SE(3)$:

$$\overline{g}_{ac} = \overline{g}_{ab}\overline{g}_{bc} = \begin{bmatrix} R_{ab}R_{bc} & R_{ab}p_{bc} + p_{ab} \\ 0_{1\times 3} & 1 \end{bmatrix}$$

As $SO(3)$ is a group, $R_{ab}R_{bc}$ is also the rotation of frame $\mathscr{R}_c$ relative to $\mathscr{R}_a$.
Besides, we have $R_{ab}p_{bc} + p_{ab} \in \mathbb{R}^3$.
Finally, $\overline{g}_{ac} \in SE(3)$.

Properties in $SE(3)$

$SE(3)$ forms a group :

1. Composition for two elements of $SE(3)$:

$$\overline{g}_{ac} = \overline{g}_{ab}\overline{g}_{bc} = \begin{bmatrix} R_{ab}R_{bc} & R_{ab}p_{bc} + p_{ab} \\ 0_{1\times 3} & 1 \end{bmatrix}$$

   As $SO(3)$ is a group, $R_{ab}R_{bc}$ is also the rotation of frame $\mathscr{R}_c$ relative to $\mathscr{R}_a$.
   Besides, we have $R_{ab}p_{bc} + p_{ab} \in \mathbb{R}^3$.
   Finally, $\overline{g}_{ac} \in SE(3)$.

2. The identity matrix of size 4×4 belongs to $SE(3)$.

Properties in $SE(3)$

$SE(3)$ forms a group :

1. Composition for two elements of $SE(3)$:

$$\overline{g}_{ac} = \overline{g}_{ab}\overline{g}_{bc} = \begin{bmatrix} R_{ab}R_{bc} & R_{ab}p_{bc} + p_{ab} \\ 0_{1\times 3} & 1 \end{bmatrix}$$

   As $SO(3)$ is a group, $R_{ab}R_{bc}$ is also the rotation of frame $\mathscr{R}_c$ relative to $\mathscr{R}_a$.
   Besides, we have $R_{ab}p_{bc} + p_{ab} \in \mathbb{R}^3$.
   Finally, $\overline{g}_{ac} \in SE(3)$.

2. The identity matrix of size 4×4 belongs to $SE(3)$.

3. If $\overline{g} \in SE(3)$, then $\overline{g}^{-1}$ is directly determined by matrix inversion :
   $\overline{g}^{-1} = (-R^t p, R^t) \in SE(3)$.

Properties in $SE(3)$

$SE(3)$ forms a group :

1. Composition for two elements of $SE(3)$:

$$\bar{g}_{ac} = \bar{g}_{ab}\bar{g}_{bc} = \begin{bmatrix} R_{ab}R_{bc} & R_{ab}p_{bc} + p_{ab} \\ 0_{1\times 3} & 1 \end{bmatrix}$$

   As $SO(3)$ is a group, $R_{ab}R_{bc}$ is also the rotation of frame $\mathscr{R}_c$ relative to $\mathscr{R}_a$.
   Besides, we have $R_{ab}p_{bc} + p_{ab} \in \mathbb{R}^3$.
   Finally, $\bar{g}_{ac} \in SE(3)$.

2. The identity matrix of size 4×4 belongs to $SE(3)$.
3. If $\bar{g} \in SE(3)$, then $\bar{g}^{-1}$ is directly determined by matrix inversion :
   $\bar{g}^{-1} = (-R^t p, R^t) \in SE(3)$.
4. The rule for composition of rigid body transformations is associative (due to the associativity property of matrix product).

Course

Computing $\overline{g}_{ij}$ for frame changes

Let $Trans\,(a, b, c)$ be such transformation, with a, b et c the components of translation along the axes x_i, y_i, and z_i :

$$p_{ij} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

Computing $\overline{g}_{ij}$ for frame changes

Let $Trans\,(a, b, c)$ be such transformation, with a, b et c the components of translation along the axes x_i, y_i, and z_i :

$$p_{ij} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

The orientation being preserved during this transformation (i.e. rotation of null angle), $Trans\,(a, b, c)$ gets the form :

$$\overline{g}_{ij} = Trans\,(a, b, c) = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Course

Computing $\overline{g}_{ij}$ for frame changes

1. Computation of $\overline{g}_{ij}$:
   - Frame translation :

$$p_{ij} = \begin{bmatrix} 2 \\ 5 \\ 4 \end{bmatrix}$$

Computing $\overline{g}_{ij}$ for frame changes

1. Computation of $\overline{g}_{ij}$:
   - Frame translation :

$$p_{ij} = \begin{bmatrix} 2 \\ 5 \\ 4 \end{bmatrix}$$

   - Frame rotation :

$$R_{ij} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix}$$

Computing $\overline{g}_{ij}$ for frame changes

1. Computation of $\overline{g}_{ij}$:
   - Frame translation :

$$p_{ij} = \begin{bmatrix} 2 \\ 5 \\ 4 \end{bmatrix}$$

   - Frame rotation :

$$R_{ij} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix}$$

   - Finally,

$$\overline{g}_{ij} = \begin{bmatrix} 0 & 0 & 1 & 2 \\ 0 & 1 & 0 & 5 \\ -1 & 0 & 0 & 4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Computing $\overline{g}_{ij}$ for frame changes

1. Computation of $\overline{g}_{ij}$:
   - Frame translation :
$$p_{ij} = \begin{bmatrix} 2 \\ 5 \\ 4 \end{bmatrix}$$
   - Frame rotation :
$$R_{ij} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix}$$
   - Finally,
$$\overline{g}_{ij} = \begin{bmatrix} 0 & 0 & 1 & 2 \\ 0 & 1 & 0 & 5 \\ -1 & 0 & 0 & 4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
2. Computing $\overline{g}_{ji}$ in inverting $\overline{g}_{ij}$:
$$\overline{g}_{ji} = \overline{g}_{ij}^{-1} = \begin{bmatrix} R_{ij}^t & -R_{ij}^t p_{ij} \\ 0_{1\times 3} & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 & 4 \\ 0 & 1 & 0 & -5 \\ 1 & 0 & 0 & -2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Computing $\overline{g}_{ij}$ for frame changes

1. Computing $\overline{g}_{ij}$:
   - Frame translation :

$$p_{ij} = \begin{bmatrix} 0 \\ l_1 \\ 0 \end{bmatrix}$$

Computing $\overline{g}_{ij}$ for frame changes

1. Computing $\overline{g}_{ij}$:
   - Frame translation :

$$p_{ij} = \begin{bmatrix} 0 \\ l_1 \\ 0 \end{bmatrix}$$

   - Frame rotation :

$$R_{ij} = R_{z_j,q} = \begin{bmatrix} cos\left(q\right) & -sin\left(q\right) & 0 \\ sin\left(q\right) & cos\left(q\right) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Computing $\overline{g}_{ij}$ for frame changes

1. Computing $\overline{g}_{ij}$:
   - Frame translation :

$$p_{ij} = \begin{bmatrix} 0 \\ l_1 \\ 0 \end{bmatrix}$$

   - Frame rotation :

$$R_{ij} = R_{z_j,q} = \begin{bmatrix} cos\,(q) & -sin\,(q) & 0 \\ sin\,(q) & cos\,(q) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

   - Finally,

$$\overline{g}_{ij} = \begin{bmatrix} cos\,(q) & -sin\,(q) & 0 & 0 \\ sin\,(q) & cos\,(q) & 0 & l_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Computing $\overline{g}_{ij}$ for frame changes

1. Computing $\overline{g}_{ij}$:
   - Frame translation :

$$p_{ij} = \begin{bmatrix} 0 \\ l_1 \\ 0 \end{bmatrix}$$

   - Frame rotation :

$$R_{ij} = R_{z_j,q} = \begin{bmatrix} cos\left(q\right) & -sin\left(q\right) & 0 \\ sin\left(q\right) & cos\left(q\right) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

   - Finally,

$$\overline{g}_{ij} = \begin{bmatrix} cos\left(q\right) & -sin\left(q\right) & 0 & 0 \\ sin\left(q\right) & cos\left(q\right) & 0 & l_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2. Computing $\overline{g}_{ji}$ in inverting $\overline{g}_{ij}$:

$$\overline{g}_{ji} = \overline{g}_{ij}^{-1} = \begin{bmatrix} R_{ij}^t & -R_{ij}^t p_{ij} \\ 0_{1\times 3} & 1 \end{bmatrix} = \begin{bmatrix} cos\left(q\right) & sin\left(q\right) & 0 & -l_1 sin\left(q\right) \\ -sin\left(q\right) & cos\left(q\right) & 0 & -l_1 cos\left(q\right) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Course

Angular velocity

Elementary rotation about z axis :

$$R_{z,\theta(t)} = \begin{bmatrix} cos\left(\theta(t)\right) & -sin\left(\theta(t)\right) & 0 \\ sin\left(\theta(t)\right) & cos\left(\theta(t)\right) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Since θ is time-dependent,

$$\begin{aligned} {}^{0}\widehat{\omega}\left(t\right) &= \dot{R}R^{-1} \\ &= \begin{bmatrix} -\dot{\theta}(t)sin\left(\theta(t)\right) & -\dot{\theta}(t)cos\left(\theta(t)\right) & 0 \\ \dot{\theta}(t)cos\left(\theta(t)\right) & -\dot{\theta}(t)sin\left(\theta(t)\right) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos\left(\theta(t)\right) & sin\left(\theta(t)\right) & 0 \\ -sin\left(\theta(t)\right) & cos\left(\theta(t)\right) & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 0 & -\dot{\theta}(t) & 0 \\ \dot{\theta}(t) & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

Thus, by term-to-term identification with the antisymmetric matrix :

$$\widehat{\omega} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

it results that :

$${}^{0}\omega\left(t\right) = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}(t) \end{bmatrix}$$

is the angular velocity about z axis in the fixed frame. ▸ Course

Angular velocity

- We deduce from *Euler* representation that :
  - α is the rotation angle about $z = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$;
  - β is the rotation angle about the current N axis (after applying $R_{z,\alpha}$) whose unit vector with respect to initial frame is $\begin{bmatrix} cos(\alpha) & sin(\alpha) & 0 \end{bmatrix}^T$;
  - γ is the rotation angle about the current Z axis (after applying $R_{z,\alpha}R_{N,\beta}$) whose unit vector components with respect to initial frame are $\begin{bmatrix} sin(\alpha)sin(\beta) & -cos(\alpha)sin(\beta) & cos(\beta) \end{bmatrix}^T$.
- Thus, the angular velocity relative to the fixed frame is given by :

$$^0\omega = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \dot{\alpha} + \begin{bmatrix} cos(\alpha) \\ sin(\alpha) \\ 0 \end{bmatrix} \dot{\beta} + \begin{bmatrix} sin(\alpha)sin(\beta) \\ -cos(\alpha)sin(\beta) \\ cos(\beta) \end{bmatrix} \dot{\gamma}$$

  thus,

$$^0\omega = \underbrace{\begin{bmatrix} 0 & cos(\alpha) & sin(\alpha)sin(\beta) \\ 0 & sin(\alpha) & -cos(\alpha)sin(\beta) \\ 1 & 0 & cos(\beta) \end{bmatrix}}_{\Omega^{-1}_{r_{\text{Euler}}}} \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix}$$

- By taking the inverse of $\Omega^{-1}_{r_{\text{Euler}}}$ (singular matrix when $sin(\beta) = 0$ as already obtained), we have :

$$\Omega_{r_{\text{Euler}}} = \begin{bmatrix} -sin(\alpha)cotg(\beta) & cos(\alpha)cotg(\beta) & 1 \\ cos(\alpha) & sin(\alpha) & 0 \\ \frac{sin(\alpha)}{sin(\beta)} & -\frac{cos(\alpha)}{sin(\beta)} & 0 \end{bmatrix}$$

Transformation of screws

Transforming velocity from one frame to another

- Previous *Varignon* relationship are meant to be referred to frame $\mathscr{R}_b$:

$$\begin{pmatrix} {}^{b}V_{ij}\left(O_a\right) \\ {}^{b}\omega_{ij} \end{pmatrix} = \begin{bmatrix} \mathbb{I}_{3\times 3} & -{}^{b}\widehat{O_b O_a} \\ \mathbb{O}_{3\times 3} & \mathbb{I}_{3\times 3} \end{bmatrix} \begin{pmatrix} {}^{b}V_{ij}\left(O_b\right) \\ {}^{b}\omega_{ij} \end{pmatrix}$$

- Projecting velocities in frame $\mathscr{R}_a$, it is :

$$\begin{aligned} {}^{a}V_{ij}\left(O_a\right) &= R_{ab}\,{}^{b}V_{ij}\left(O_a\right) \\ {}^{a}\omega_{ij} &= R_{ab}\,{}^{b}\omega_{ij} \end{aligned}$$

i.e.

$$\begin{pmatrix} {}^{a}V_{ij}\left(O_a\right) \\ {}^{a}\omega_{ij} \end{pmatrix} = \begin{bmatrix} R_{ab} & \mathbb{O}_{3\times 3} \\ \mathbb{O}_{3\times 3} & R_{ab} \end{bmatrix} \begin{pmatrix} {}^{b}V_{ij}\left(O_a\right) \\ {}^{b}\omega_{ij} \end{pmatrix}$$

- Combining previous relationships together leads to :

$$\begin{pmatrix} {}^{a}V_{ij}\left(O_a\right) \\ {}^{a}\omega_{ij} \end{pmatrix} = \begin{bmatrix} R_{ab} & -R_{ab}\,{}^{b}\widehat{O_b O_a} \\ \mathbb{O}_{3\times 3} & R_{ab} \end{bmatrix} \begin{pmatrix} {}^{b}V_{ij}\left(O_b\right) \\ {}^{b}\omega_{ij} \end{pmatrix}$$

Transformation of screws

- Using contracted notation, the previous expression becomes :

$$
{}^{a}\mathscr{V}_{ij} = \begin{bmatrix} R_{ab} & -R_{ab}\widehat{p_{ba}} \\ \mathbb{O}_{3\times 3} & R_{ab} \end{bmatrix} {}^{b}\mathscr{V}_{ij}
$$

- Using following lemma :

$$
-R_{ab}\widehat{p_{ba}} = \widehat{p_{ab}}R_{ab}
$$

Elements of proof :

- recall that $p_{ba} = -R_{ab}^{T}p_{ab}$ (coming from the expressions of g_{ab} and its inverse g_{ba} and comparing together their respective fourth column),
- combined with the following property : $\forall p \in \mathbb{R}^3$ and $\forall R \in SO(3)$, $\widehat{Rp} = R\hat{p}R^T$,
- it comes the following simplification : $-R_{ab}\widehat{p_{ba}} = R_{ab}\widehat{R_{ab}^{t}p_{ab}} = \underbrace{R_{ab}R_{ab}^{T}}_{\mathbb{I}}\widehat{p_{ab}}R_{ab} = \widehat{p_{ab}}R_{ab}$.

- The previous contracted notation is equivalently equal to :

$$
{}^{a}\mathscr{V}_{ij} = \underbrace{\begin{bmatrix} R_{ab} & \widehat{p_{ab}}R_{ab} \\ \mathbb{O}_{3\times 3} & R_{ab} \end{bmatrix}}_{X_{ab}} {}^{b}\mathscr{V}_{ij}
$$

Course

Transformation of screws

Inverse of screw transformation matrix

- The inverse of X_{ab} is given by matrix

$$\begin{bmatrix} R_{ab}^T & -R_{ab}^T \widehat{p}_{ab} \\ \mathbb{O}_{3\times 3} & R_{ab}^T \end{bmatrix}$$

Let observe that, in accordance with the above lemma, this matrix corresponds to a permutation of indices a and b, and, thus, is equal to X_{ba}. We can easily verify that $X_{ab}X_{ba} = X_{ba}X_{ab} = \mathbb{I}$, thus :

$$X_{ab}^{-1} = X_{ba}$$

Transformation matrix for wrench

- An insightful way to derive the relationship between ${}^a\mathscr{F}_{ij}$ and ${}^b\mathscr{F}_{ij}$ is to use the fact that the power generated by an $(\mathscr{F}, \mathscr{V})$ pair must be the same regardless of the frame in which it is represented.
- Recall that the dot product of a force and velocity is a power, and power is coordinate-independent quantity, thus :

$${}^b\mathscr{V}_{ij}^{T\,b}\mathscr{F}_{ij} = {}^a\mathscr{V}_{ij}^{T\,a}\mathscr{F}_{ij}$$

Transformation of screws

- As we know that ${}^{a}\mathscr{V}_{ij} = X_{ab}\,{}^{b}\mathscr{V}_{ij}$, the previous equation can be rewritten as

$$
\begin{aligned}
{}^{b}\mathscr{V}_{ij}^{T}\,{}^{b}\mathscr{F}_{ij} &= \left(X_{ab}\,{}^{b}\mathscr{V}_{ij}\right)^{T}\,{}^{a}\mathscr{F}_{ij} \\
&= {}^{b}\mathscr{V}_{ij}^{T} X_{ab}^{T}\,{}^{a}\mathscr{F}_{ij}
\end{aligned}
$$

Since this must hold for all ${}^{b}\mathscr{V}_{ij}$, this simplifies to

$$
{}^{b}\mathscr{F}_{ij} = X_{ab}^{T}\,{}^{a}\mathscr{F}_{ij}
$$

Similarly,

$$
{}^{a}\mathscr{F}_{ij} = X_{ab}^{-T}\,{}^{b}\mathscr{F}_{ij} = X_{ba}^{T}\,{}^{b}\mathscr{F}_{ij}
$$

▸ Course

Multirobot arm grasping

Computation of the net object wrench ${}^{o}\mathscr{F}_{o} = {}^{o}\mathscr{F}_{o1} + {}^{o}\mathscr{F}_{o2}$.

- Firstly, the position and orientation of each contact frames w.r.t. the object frame are computed :
  - considering frame $\mathscr{R}_1$:

$$R_{oC_1} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

$$p_{oC_1} = \begin{pmatrix} 0 \\ -\frac{L}{2} \\ 0 \end{pmatrix}$$

  - considering frame $\mathscr{R}_2$:

$$R_{oC_2} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$p_{oC_2} = \begin{pmatrix} 0 \\ \frac{L}{2} \\ 0 \end{pmatrix}$$

Multirobot arm grasping

- The total object wrench given in the object frame is given by :

$$
\begin{aligned}
{}^{o}\mathscr{F}_{o} &= {}^{o}\mathscr{F}_{o1} + {}^{o}\mathscr{F}_{o2} \\
&= X_{oC_1}^{-T}\,{}^{C_1}\mathscr{F}_{o1} + X_{oC_2}^{-T}\,{}^{C_2}\mathscr{F}_{o2} \\
&= \begin{bmatrix} R_{oC_1} & \mathbb{O}_{3\times 3} \\ \widehat{p}_{oC_1} R_{oC_1} & R_{oC_1} \end{bmatrix} {}^{C_1}\mathscr{F}_{o1} + \begin{bmatrix} R_{oC_2} & \mathbb{O}_{3\times 3} \\ \widehat{p}_{oC_2} R_{oC_2} & R_{oC_2} \end{bmatrix} {}^{C_2}\mathscr{F}_{o2} \\
&= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ -\frac{L}{2} & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & \frac{L}{2} & 0 & 1 & 0 & 0 \end{bmatrix} {}^{C_1}\mathscr{F}_{o1} + \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & \frac{L}{2} & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ -\frac{L}{2} & 0 & 0 & 0 & 1 & 0 \end{bmatrix} {}^{C_2}\mathscr{F}_{o2} \\
&= \begin{pmatrix} f_1^y + f_2^x \\ f_1^z - f_2^z \\ f_1^x + f_2^y \\ m_1^y + m_2^x + \frac{L}{2}\left(f_2^y - f_1^x\right) \\ m_1^z - m_2^z \\ m_1^x + m_2^y + \frac{L}{2}\left(f_1^y - f_2^x\right) \end{pmatrix}
\end{aligned}
$$

Course

Robot gripper holding an object

- We define frames $\mathscr{R}_f$ at the F/T torque sensor, $\mathscr{R}_h$ at the center of mass of the gripper and $\mathscr{R}_o$ at the center of mass of the apple.
- According to the coordinate axes in the figure, the gravitational wrench applied on the gripper in $\mathscr{R}_h$ is given by the column vector :

$$ {}^{h}\mathscr{F}_{hg} = \begin{bmatrix} 0 & -m_h g & 0 & 0 & 0 & 0 \end{bmatrix}^T $$

  and the gravitational wrench on the object in $\mathscr{R}_o$ is :

$$ {}^{o}\mathscr{F}_{og} = \begin{bmatrix} 0 & 0 & m_o g & 0 & 0 & 0 \end{bmatrix}^T . $$

- The orientation and position of each contact frames w.r.t. the sensor frame are :
  - considering frame $\mathscr{R}_h$:

$$ R_{fh} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad p_{fh} = \begin{pmatrix} L_1 \\ 0 \\ 0 \end{pmatrix} $$

  - considering frame $\mathscr{R}_o$:

$$ R_{fo} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}, \quad p_{fo} = \begin{pmatrix} L_1 + L_2 \\ 0 \\ 0 \end{pmatrix} $$

Robot gripper holding an object

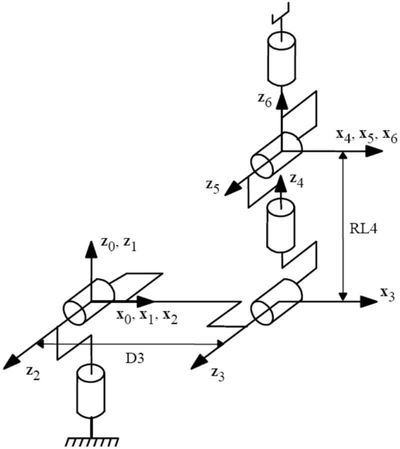
- The total wrench given in the F/T sensor frame is given by :

$$
\begin{aligned}
{}^{f}\mathscr{F}_{(h+o)g} &= {}^{f}\mathscr{F}_{hg} + {}^{f}\mathscr{F}_{og} \\
&= \left(X_{fh}^{-T}\right) {}^{h}\mathscr{F}_{hg} + \left(X_{fo}^{-T}\right) {}^{o}\mathscr{F}_{og} \\
&= \begin{bmatrix} R_{fh} & \mathbb{O}_{3\times 3} \\ \widehat{p}_{fh} R_{fh} & R_{fh} \end{bmatrix} {}^{h}\mathscr{F}_{hg} + \begin{bmatrix} R_{fo} & \mathbb{O}_{3\times 3} \\ \widehat{p}_{fo} R_{fo} & R_{fo} \end{bmatrix} {}^{o}\mathscr{F}_{og} \\
&= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -L_1 & 0 & 1 & 0 \\ 0 & L_1 & 0 & 0 & 0 & 1 \end{bmatrix} {}^{h}\mathscr{F}_{hg} + \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -L_1 - L_2 & 0 & 0 & 0 & -1 \\ 0 & 0 & -L_1 - L_2 & 0 & 1 & 0 \end{bmatrix} {}^{o}\mathscr{F}_{og} \\
&= \begin{pmatrix} 0 \\ -(m_o + m_h)g \\ 0 \\ 0 \\ 0 \\ (m_o(L_1 + L_2) + m_h L_1)\, g \end{pmatrix}
\end{aligned}
$$

Course

DGM of robot RX130L from *Staübli*^{TM}

- Assignation of frames



Course

DGM of robot RX130L from *Staübli*^{TM}

- Table for parameters $(\alpha_i, d_i, \theta_i, r_i)$

i	α_i	d_i	θ_i	r_i
1	0	0	θ_1	0
2	$\pi/2$	0	θ_2	0
3	0	$D_3 = 0,625m$	θ_3	0
4	-$\pi/2$	0	θ_4	RL4=0,925m
5	$\pi/2$	0	θ_5	0
6	-$\pi/2$	0	θ_6	0

Course

DGM of robot RX130L from $Staübli^{TM}$

- Table for parameters $(\alpha_i, d_i, \theta_i, r_i)$

i	α_i	d_i	θ_i	r_i
1	0	0	θ_1	0
2	$\pi/2$	0	θ_2	0
3	0	$D_3 = 0,625m$	θ_3	0
4	-$\pi/2$	0	θ_4	RL4=0,925m
5	$\pi/2$	0	θ_5	0
6	-$\pi/2$	0	θ_6	0

Course

- Computation of elementary transformation matrices $\overline{g}_{(i-1)i}$ from the general theoretical expression given in the course :

$$\overline{g}_{(i-1)i} = R_{x,\alpha_i}\, Trans\,(x, d_i)\, R_{z,\theta_i}\, Trans\,(z, r_i)$$

$$= \begin{bmatrix} cos\,(\theta_i) & -sin\,(\theta_i) & 0 & d_i \\ cos\,(\alpha_i)\, sin\,(\theta_i) & cos\,(\alpha_i)\, cos\,(\theta_i) & -sin\,(\alpha_i) & -r_i sin\,(\alpha_i) \\ sin\,(\alpha_i)\, sin\,(\theta_i) & sin\,(\alpha_i)\, cos\,(\theta_i) & cos\,(\alpha_i) & r_i cos\,(\alpha_i) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

DGM of robot RX130L from *Staübli*^{TM}

- Transformation matrix $\overline{g}_{01}$

$$\overline{g}_{01} = R_{z,\theta_1} = \begin{bmatrix} cos\left(q_1\right) & -sin\left(q_1\right) & 0 & 0 \\ sin\left(q_1\right) & cos\left(q_1\right) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

DGM of robot RX130L from *Staübli*^{TM}

- Transformation matrix $\overline{g}_{01}$

$$\overline{g}_{01} = R_{z,\theta_1} = \begin{bmatrix} cos\left(q_1\right) & -sin\left(q_1\right) & 0 & 0 \\ sin\left(q_1\right) & cos\left(q_1\right) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Transformation matrix $\overline{g}_{12}$

$$\overline{g}_{12} = R_{x,\pi/2} R_{z,\theta_2} = \begin{bmatrix} cos\left(q_2\right) & -sin\left(q_2\right) & 0 & 0 \\ 0 & 0 & -1 & 0 \\ sin\left(q_2\right) & cos\left(q_2\right) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

DGM of robot RX130L from *Staübli*^{TM}

- Transformation matrix $\overline{g}_{01}$

$$\overline{g}_{01} = R_{z,\theta_1} = \begin{bmatrix} cos\,(q_1) & -sin\,(q_1) & 0 & 0 \\ sin\,(q_1) & cos\,(q_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Transformation matrix $\overline{g}_{12}$

$$\overline{g}_{12} = R_{x,\pi/2} R_{z,\theta_2} = \begin{bmatrix} cos\,(q_2) & -sin\,(q_2) & 0 & 0 \\ 0 & 0 & -1 & 0 \\ sin\,(q_2) & cos\,(q_2) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Transformation matrix $\overline{g}_{23}$

$$\overline{g}_{23} = Trans\,(x, D_3)\,R_{z,\theta_3} = \begin{bmatrix} cos\,(q_3) & -sin\,(q_3) & 0 & D_3 \\ sin\,(q_3) & cos\,(q_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

DGM of robot RX130L from *Staübli*^{TM}

- Transformation matrix $\overline{g}_{01}$

$$\overline{g}_{01} = R_{z,\theta_1} = \begin{bmatrix} cos\left(q_1\right) & -sin\left(q_1\right) & 0 & 0 \\ sin\left(q_1\right) & cos\left(q_1\right) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Transformation matrix $\overline{g}_{12}$

$$\overline{g}_{12} = R_{x,\pi/2}R_{z,\theta_2} = \begin{bmatrix} cos\left(q_2\right) & -sin\left(q_2\right) & 0 & 0 \\ 0 & 0 & -1 & 0 \\ sin\left(q_2\right) & cos\left(q_2\right) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Transformation matrix $\overline{g}_{23}$

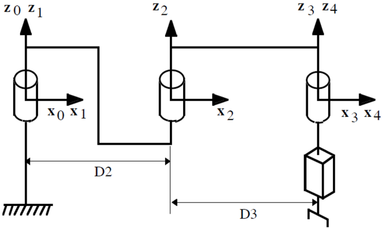
$$\overline{g}_{23} = Trans\left(x, D_3\right) R_{z,\theta_3} = \begin{bmatrix} cos\left(q_3\right) & -sin\left(q_3\right) & 0 & D_3 \\ sin\left(q_3\right) & cos\left(q_3\right) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Transformation matrix $\overline{g}_{34}$

$$\overline{g}_{34} = R_{x,-\pi/2}R_{z,\theta_4}Trans\left(z, RL_4\right) = \begin{bmatrix} cos\left(q_4\right) & -sin\left(q_4\right) & 0 & 0 \\ 0 & 0 & 1 & RL_4 \\ -sin\left(q_4\right) & -cos\left(q_4\right) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

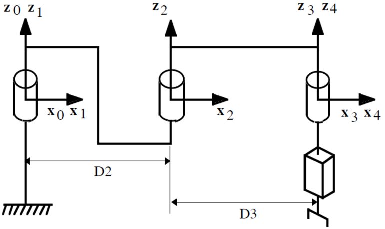
DGM of robot SCARA

- Assignation of frames



DGM of robot SCARA

- Assignation of frames



- Table for parameters $(\alpha_i, d_i, \theta_i, r_i)$

i	α_i	d_i	θ_i	r_i
1	0	0	θ_1	0
2	0	D_2	θ_2	0
3	0	D_3	θ_3	0
4	0	0	0	r_4

Course

Case of planar RRR robot

- Homogeneous transformation matrix $\overline{g}_{0E}$:

$$\overline{g}_{0E} = \overline{g}_{03}\overline{g}_{3E} = \overline{g}_{01}\overline{g}_{12}\overline{g}_{23}\overline{g}_{3E}$$

Case of planar RRR robot

- Homogeneous transformation matrix $\overline{g}_{0E}$:
$$\overline{g}_{0E} = \overline{g}_{03}\overline{g}_{3E} = \overline{g}_{01}\overline{g}_{12}\overline{g}_{23}\overline{g}_{3E}$$
  - Homogeneous transformation matrix $\overline{g}_{01}$ (parameters $(\alpha_i, d_i, \theta_i, r_i) = (0, 0, q_1, 0)$)

Case of planar RRR robot

- Homogeneous transformation matrix $\overline{g}_{0E}$:

$$\overline{g}_{0E} = \overline{g}_{03}\overline{g}_{3E} = \overline{g}_{01}\overline{g}_{12}\overline{g}_{23}\overline{g}_{3E}$$

- Homogeneous transformation matrix $\overline{g}_{01}$ (parameters $(\alpha_i, d_i, \theta_i, r_i) = (0, 0, q_1, 0)$)

$$\overline{g}_{01} = R_{z_1, q_1} = \begin{bmatrix} cos\,(q_1) & -sin\,(q_1) & 0 & 0 \\ sin\,(q_1) & cos\,(q_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Case of planar RRR robot

- Homogeneous transformation matrix $\overline{g}_{0E}$:

$$\overline{g}_{0E} = \overline{g}_{03}\overline{g}_{3E} = \overline{g}_{01}\overline{g}_{12}\overline{g}_{23}\overline{g}_{3E}$$

  - Homogeneous transformation matrix $\overline{g}_{01}$ (parameters $(\alpha_i, d_i, \theta_i, r_i) = (0, 0, q_1, 0)$)

$$\overline{g}_{01} = R_{z_1, q_1} = \begin{bmatrix} cos\,(q_1) & -sin\,(q_1) & 0 & 0 \\ sin\,(q_1) & cos\,(q_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

  - Homogeneous transformation matrix $\overline{g}_{12}$ (parameters $(\alpha_i, d_i, \theta_i, r_i) = (0, l_1, q_2, 0)$)

Case of planar RRR robot

- Homogeneous transformation matrix $\overline{g}_{0E}$:

$$\overline{g}_{0E} = \overline{g}_{03}\overline{g}_{3E} = \overline{g}_{01}\overline{g}_{12}\overline{g}_{23}\overline{g}_{3E}$$

- Homogeneous transformation matrix $\overline{g}_{01}$ (parameters $(\alpha_i, d_i, \theta_i, r_i) = (0, 0, q_1, 0)$)

$$\overline{g}_{01} = R_{z_1,q_1} = \begin{bmatrix} cos\,(q_1) & -sin\,(q_1) & 0 & 0 \\ sin\,(q_1) & cos\,(q_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Homogeneous transformation matrix $\overline{g}_{12}$ (parameters $(\alpha_i, d_i, \theta_i, r_i) = (0, l_1, q_2, 0)$)

$$\overline{g}_{12} = Trans(x_1, l_1)R_{z_2,q_2} = \begin{bmatrix} cos\,(q_2) & -sin\,(q_2) & 0 & l_1 \\ sin\,(q_2) & cos\,(q_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Case of planar RRR robot

- Homogeneous transformation matrix $\overline{g}_{0E}$:

$$\overline{g}_{0E} = \overline{g}_{03}\overline{g}_{3E} = \overline{g}_{01}\overline{g}_{12}\overline{g}_{23}\overline{g}_{3E}$$

  - Homogeneous transformation matrix $\overline{g}_{01}$ (parameters $(\alpha_i, d_i, \theta_i, r_i) = (0, 0, q_1, 0)$)

$$\overline{g}_{01} = R_{z_1, q_1} = \begin{bmatrix} cos\,(q_1) & -sin\,(q_1) & 0 & 0 \\ sin\,(q_1) & cos\,(q_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

  - Homogeneous transformation matrix $\overline{g}_{12}$ (parameters $(\alpha_i, d_i, \theta_i, r_i) = (0, l_1, q_2, 0)$)

$$\overline{g}_{12} = Trans(x_1, l_1)R_{z_2, q_2} = \begin{bmatrix} cos\,(q_2) & -sin\,(q_2) & 0 & l_1 \\ sin\,(q_2) & cos\,(q_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

  - Homogeneous transformation matrix $\overline{g}_{23}$ (parameters $(\alpha_i, d_i, \theta_i, r_i) = (0, l_2, q_3, 0)$)

Case of planar RRR robot

- Homogeneous transformation matrix $\overline{g}_{0E}$:

$$\overline{g}_{0E} = \overline{g}_{03}\overline{g}_{3E} = \overline{g}_{01}\overline{g}_{12}\overline{g}_{23}\overline{g}_{3E}$$

- Homogeneous transformation matrix $\overline{g}_{01}$ (parameters $(\alpha_i, d_i, \theta_i, r_i) = (0, 0, q_1, 0)$)

$$\overline{g}_{01} = R_{z_1,q_1} = \begin{bmatrix} cos\,(q_1) & -sin\,(q_1) & 0 & 0 \\ sin\,(q_1) & cos\,(q_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Homogeneous transformation matrix $\overline{g}_{12}$ (parameters $(\alpha_i, d_i, \theta_i, r_i) = (0, l_1, q_2, 0)$)

$$\overline{g}_{12} = Trans(x_1, l_1)R_{z_2,q_2} = \begin{bmatrix} cos\,(q_2) & -sin\,(q_2) & 0 & l_1 \\ sin\,(q_2) & cos\,(q_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Homogeneous transformation matrix $\overline{g}_{23}$ (parameters $(\alpha_i, d_i, \theta_i, r_i) = (0, l_2, q_3, 0)$)

$$\overline{g}_{23} = Trans(x_2, l_2)R_{z_3,q_3} = \begin{bmatrix} cos\,(q_3) & -sin\,(q_3) & 0 & l_2 \\ sin\,(q_3) & cos\,(q_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Case of planar RRR robot

- Homogeneous transformation matrix $\overline{g}_{0E}$:

$$\overline{g}_{0E} = \overline{g}_{03}\overline{g}_{3E} = \overline{g}_{01}\overline{g}_{12}\overline{g}_{23}\overline{g}_{3E}$$

  - Homogeneous transformation matrix $\overline{g}_{01}$ (parameters $(\alpha_i, d_i, \theta_i, r_i) = (0, 0, q_1, 0)$)

$$\overline{g}_{01} = R_{z_1,q_1} = \begin{bmatrix} cos\,(q_1) & -sin\,(q_1) & 0 & 0 \\ sin\,(q_1) & cos\,(q_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

  - Homogeneous transformation matrix $\overline{g}_{12}$ (parameters $(\alpha_i, d_i, \theta_i, r_i) = (0, l_1, q_2, 0)$)

$$\overline{g}_{12} = Trans(x_1, l_1)R_{z_2,q_2} = \begin{bmatrix} cos\,(q_2) & -sin\,(q_2) & 0 & l_1 \\ sin\,(q_2) & cos\,(q_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

  - Homogeneous transformation matrix $\overline{g}_{23}$ (parameters $(\alpha_i, d_i, \theta_i, r_i) = (0, l_2, q_3, 0)$)

$$\overline{g}_{23} = Trans(x_2, l_2)R_{z_3,q_3} = \begin{bmatrix} cos\,(q_3) & -sin\,(q_3) & 0 & l_2 \\ sin\,(q_3) & cos\,(q_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

  - Homogeneous transformation matrix $\overline{g}_{3E}$ (parameters $(\alpha_i, d_i, \theta_i, r_i) = (0, l_3, 0, 0)$)

Case of planar RRR robot

- Homogeneous transformation matrix $\overline{g}_{0E}$:

$$\overline{g}_{0E} = \overline{g}_{03}\overline{g}_{3E} = \overline{g}_{01}\overline{g}_{12}\overline{g}_{23}\overline{g}_{3E}$$

  - Homogeneous transformation matrix $\overline{g}_{01}$ (parameters $(\alpha_i, d_i, \theta_i, r_i) = (0, 0, q_1, 0)$)

$$\overline{g}_{01} = R_{z_1,q_1} = \begin{bmatrix} cos(q_1) & -sin(q_1) & 0 & 0 \\ sin(q_1) & cos(q_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

  - Homogeneous transformation matrix $\overline{g}_{12}$ (parameters $(\alpha_i, d_i, \theta_i, r_i) = (0, l_1, q_2, 0)$)

$$\overline{g}_{12} = Trans(x_1, l_1)R_{z_2,q_2} = \begin{bmatrix} cos(q_2) & -sin(q_2) & 0 & l_1 \\ sin(q_2) & cos(q_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

  - Homogeneous transformation matrix $\overline{g}_{23}$ (parameters $(\alpha_i, d_i, \theta_i, r_i) = (0, l_2, q_3, 0)$)

$$\overline{g}_{23} = Trans(x_2, l_2)R_{z_3,q_3} = \begin{bmatrix} cos(q_3) & -sin(q_3) & 0 & l_2 \\ sin(q_3) & cos(q_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

  - Homogeneous transformation matrix $\overline{g}_{3E}$ (parameters $(\alpha_i, d_i, \theta_i, r_i) = (0, l_3, 0, 0)$)

$$\overline{g}_{3E} = Trans(x_3, l_3) = \begin{bmatrix} 1 & 0 & 0 & l_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Case of planar RRR robot

- Homogeneous transformation matrix $\overline{g}_{0E}$:

$$\begin{aligned}\overline{g}_{0E} &= \overline{g}_{01}\overline{g}_{12}\overline{g}_{23}\overline{g}_{3E}\\ &= \dots\\ &= \begin{bmatrix} C123 & -S123 & 0 & l_1 C1 + l_2 C12 + l_3 C123 \\ S123 & C123 & 0 & l_1 S12 + l_2 S12 + l_3 S123 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\end{aligned}$$

Case of planar RRR robot

- Homogeneous transformation matrix $\overline{g}_{0E}$:

$$
\begin{aligned}
\overline{g}_{0E} &= \overline{g}_{01}\overline{g}_{12}\overline{g}_{23}\overline{g}_{3E} \\
&= \dots \\
&= \begin{bmatrix} C123 & -S123 & 0 & l_1 C1 + l_2 C12 + l_3 C123 \\ S123 & C123 & 0 & l_1 S12 + l_2 S12 + l_3 S123 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
$$

- Geometric model $X = f(q)$

Case of planar RRR robot

- Homogeneous transformation matrix $\overline{g}_{0E}$:

$$\begin{aligned}\overline{g}_{0E} &= \overline{g}_{01}\overline{g}_{12}\overline{g}_{23}\overline{g}_{3E} \\ &= \dots \\ &= \begin{bmatrix} C123 & -S123 & 0 & l_1 C1 + l_2 C12 + l_3 C123 \\ S123 & C123 & 0 & l_1 S12 + l_2 S12 + l_3 S123 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\end{aligned}$$

- Geometric model $X = f(q)$
  - Identification from the form of $\overline{g}_{0E}$

$$\overline{g}_{0E} = \begin{bmatrix} & R_{(q_1+q_2+q_3),z_0} & & \begin{matrix} p_x \\ p_y \\ p_z \end{matrix} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Case of planar RRR robot

- Homogeneous transformation matrix $\overline{g}_{0E}$:

$$\begin{aligned}\overline{g}_{0E} &= \overline{g}_{01}\overline{g}_{12}\overline{g}_{23}\overline{g}_{3E} \\ &= \dots \\ &= \begin{bmatrix} C123 & -S123 & 0 & l_1C1 + l_2C12 + l_3C123 \\ S123 & C123 & 0 & l_1S12 + l_2S12 + l_3S123 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\end{aligned}$$

- Geometric model $X = f(q)$
  - Identification from the form of $\overline{g}_{0E}$

$$\overline{g}_{0E} = \begin{bmatrix} & R_{(q_1+q_2+q_3),z_0} & & \begin{matrix} p_x \\ p_y \\ p_z \end{matrix} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

  - Choice of cartesian coordinates X

Case of planar RRR robot

- Homogeneous transformation matrix $\overline{g}_{0E}$:

$$\begin{aligned}\overline{g}_{0E} &= \overline{g}_{01}\overline{g}_{12}\overline{g}_{23}\overline{g}_{3E} \\ &= \dots \\ &= \begin{bmatrix} C123 & -S123 & 0 & l_1C1 + l_2C12 + l_3C123 \\ S123 & C123 & 0 & l_1S12 + l_2S12 + l_3S123 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\end{aligned}$$

- Geometric model $X = f\left(q\right)$
  - Identification from the form of $\overline{g}_{0E}$

$$\overline{g}_{0E} = \begin{bmatrix} & & & p_x \\ & R_{\left(q_1+q_2+q_3\right),z_0} & & p_y \\ & & & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

  - Choice of cartesian coordinates X
    - Coordinates $\left(p_x, p_y\right)$ of point E in $\mathscr{R}_0$ (let note that $p_z = 0$, i.e. planar motion of the robot)
    - Angle α of the last body around axis x_0 (also equal to the 1^{st} *Euler* angle, R_{α,z_0} due to the planar motion of the robot)

Case of planar RRR robot

- Homogeneous transformation matrix $\overline{g}_{0E}$:

$$\begin{aligned}\overline{g}_{0E} &= \overline{g}_{01}\overline{g}_{12}\overline{g}_{23}\overline{g}_{3E}\\ &= \dots\\ &= \begin{bmatrix} C123 & -S123 & 0 & l_1C1 + l_2C12 + l_3C123 \\ S123 & C123 & 0 & l_1S12 + l_2S12 + l_3S123 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\end{aligned}$$

- Geometric model $X = f\left(q\right)$
  - Identification from the form of $\overline{g}_{0E}$

$$\overline{g}_{0E} = \begin{bmatrix} & & & p_x \\ & R_{\left(q_1+q_2+q_3\right),z_0} & & p_y \\ & & & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

  - Choice of cartesian coordinates X
    - Coordinates $\left(p_x, p_y\right)$ of point E in $\mathscr{R}_0$ (let note that $p_z = 0$, i.e. planar motion of the robot)
    - Angle α of the last body around axis x_0 (also equal to the 1^{st} *Euler* angle, R_{α,z_0} due to the planar motion of the robot)
  - Relationships of the sought DGM :

$$\begin{aligned} p_x &= l_1C1 + l_2C12 + l_3C123 \\ p_y &= l_1S12 + l_2S12 + l_3S123 \\ \alpha &= q_1 + q_2 + q_3 \end{aligned}$$

Case of planar RRR robot

- Computation of ${}^0J$ through the derivation of the DGM $X = f(q)$ with $X = \begin{pmatrix} p_x \\ p_y \\ \alpha \end{pmatrix}$ and $q = \begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix}$:

Case of planar RRR robot

- Computation of ${}^{0}J$ through the derivation of the DGM $X = f(q)$ with $X = \begin{pmatrix} p_x \\ p_y \\ \alpha \end{pmatrix}$ and $q = \begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix}$:

$$\begin{pmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{\alpha} \end{pmatrix} = {}^{0}J(q) \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{pmatrix}$$

Case of planar RRR robot

- Computation of ${}^0J$ through the derivation of the DGM $X = f(q)$ with $X = \begin{pmatrix} p_x \\ p_y \\ \alpha \end{pmatrix}$ and $q = \begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix}$:

$$\begin{pmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{\alpha} \end{pmatrix} = {}^0J(q) \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{pmatrix}$$

where

$${}^0J(q) = \begin{bmatrix} -l_1 S1 - l_2 S12 - l_3 S123 & -l_2 S12 - l_3 S123 & -l_3 S123 \\ l_1 C1 + l_2 C12 + l_3 C123 & l_2 C12 + l_3 C123 & l_3 C123 \\ 1 & 1 & 1 \end{bmatrix}$$

Course

Case of planar PPP robot

Velocities of the cartesien end-effector positions E, $\dot{X} = \begin{pmatrix} \dot{p}_x & \dot{p}_y \end{pmatrix}^t$, as a function of the joint velocities $\dot{q} = \begin{pmatrix} \dot{r}_1 & \dot{r}_2 & \dot{r}_3 \end{pmatrix}^t$:

Case of planar PPP robot

Velocities of the cartesien end-effector positions E, $\dot{X} = \begin{pmatrix} \dot{p}_x & \dot{p}_y \end{pmatrix}^t$, as a function of the joint velocities $\dot{q} = \begin{pmatrix} \dot{r}_1 & \dot{r}_2 & \dot{r}_3 \end{pmatrix}^t$:

$$\begin{pmatrix} \dot{p}_x \\ \dot{p}_y \end{pmatrix} = {}^0J(q) \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{pmatrix}$$

Case of planar PPP robot

Velocities of the cartesien end-effector positions E, $\dot{X} = \begin{pmatrix} \dot{p}_x & \dot{p}_y \end{pmatrix}^t$, as a function of the joint velocities $\dot{q} = \begin{pmatrix} \dot{r}_1 & \dot{r}_2 & \dot{r}_3 \end{pmatrix}^t$:

$$\begin{pmatrix} \dot{p}_x \\ \dot{p}_y \end{pmatrix} = {}^0J(q) \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{pmatrix}$$

where

$${}^0J(q) = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Course

Jacobian matrix using composition rule

Expression of the Jacobian matrix for *RRR* robot :

$$^{0}J(q) = \left[\begin{array}{ccc} {}^{0}J_1(q) & {}^{0}J_2(q) & {}^{0}J_3(q) \end{array}\right]$$

giving the cartesian velocity of point E as a function of joint variables $q = \left[\begin{array}{ccc} q_1 & q_2 & q_3 \end{array}\right]^t$.

Jacobian matrix using composition rule

Expression of the Jacobian matrix for RRR robot :

$$^{0}J\left(q\right)=\left[\begin{array}{ccc} ^{0}J_{1}\left(q\right) & ^{0}J_{2}\left(q\right) & ^{0}J_{3}\left(q\right)\end{array}\right]$$

giving the cartesian velocity of point E as a function of joint variables $q=\left[\begin{array}{ccc} q_{1} & q_{2} & q_{3}\end{array}\right]^{t}$.
Considering all revolute-type joint (robot type $3R$), it comes that :

$$^{0}J\left(q\right)=\left[\begin{array}{ccc} R_{01}\left(Z_{1}\times p_{1E}\right) & R_{02}\left(Z_{2}\times p_{2E}\right) & R_{03}\left(Z_{3}\times p_{3E}\right)\\ R_{01}Z_{1} & R_{02}Z_{2} & R_{03}Z_{3}\end{array}\right]$$

Jacobian matrix using composition rule

Expression of the Jacobian matrix for *RRR* robot :

$$^{0}J(q) = \left[\begin{array}{ccc} ^{0}J_1(q) & ^{0}J_2(q) & ^{0}J_3(q) \end{array} \right]$$

giving the cartesian velocity of point E as a function of joint variables $q = \left[\begin{array}{ccc} q_1 & q_2 & q_3 \end{array} \right]^t$.
Considering all revolute-type joint (robot type $3R$), it comes that :

$$^{0}J(q) = \left[\begin{array}{ccc} R_{01}(Z_1 \times p_{1E}) & R_{02}(Z_2 \times p_{2E}) & R_{03}(Z_3 \times p_{3E}) \\ R_{01}Z_1 & R_{02}Z_2 & R_{03}Z_3 \end{array} \right]$$

with :

$$Z_1 = Z_2 = Z_3 = \left[\begin{array}{c} 0 \\ 0 \\ 1 \end{array} \right]$$

Jacobian matrix using composition rule

Computation of translation elements :

Jacobian matrix using composition rule

Computation of translation elements :

- p_{3E} : translation vector extracted from the homogeneous transformation $\overline{g}_{3E}$ (already computed in previous example)

Jacobian matrix using composition rule

Computation of translation elements :

- p_{3E} : translation vector extracted from the homogeneous transformation $\overline{g}_{3E}$ (already computed in previous example)

$$p_{3E} = \begin{bmatrix} l_3 \\ 0 \\ 0 \end{bmatrix}$$

Jacobian matrix using composition rule

Computation of translation elements :

- p_{3E} : translation vector extracted from the homogeneous transformation $\overline{g}_{3E}$ (already computed in previous example)

$$p_{3E} = \begin{bmatrix} l_3 \\ 0 \\ 0 \end{bmatrix}$$

- p_{2E} : translation vector extracted from the homogeneous transformation $\overline{g}_{2E} = \overline{g}_{23}\overline{g}_{3E}$

Jacobian matrix using composition rule

Computation of translation elements :

- p_{3E} : translation vector extracted from the homogeneous transformation $\overline{g}_{3E}$ (already computed in previous example)

$$p_{3E} = \begin{bmatrix} l_3 \\ 0 \\ 0 \end{bmatrix}$$

- p_{2E} : translation vector extracted from the homogeneous transformation $\overline{g}_{2E} = \overline{g}_{23}\overline{g}_{3E}$

$$\begin{aligned} p_{2E} =& R_{23} p_{3E} + p_{23} \quad \text{(formula given in previous chapter)} \\ =& \begin{bmatrix} cos\left(q_3\right) & -sin\left(q_3\right) & 0 \\ sin\left(q_3\right) & cos\left(q_3\right) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} l_3 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} l_2 \\ 0 \\ 0 \end{bmatrix} \quad (R_{23} \text{ and } p_{23} \text{ extracted from } \overline{g}_{23}) \\ =& \begin{bmatrix} l_2 + l_3 cos(q_3) \\ l_3 sin(q_3) \\ 0 \end{bmatrix} \end{aligned}$$

Jacobian matrix using composition rule

Computation of translation elements :

- p_{3E} : translation vector extracted from the homogeneous transformation $\overline{g}_{3E}$ (already computed in previous example)

$$p_{3E} = \begin{bmatrix} l_3 \\ 0 \\ 0 \end{bmatrix}$$

- p_{2E} : translation vector extracted from the homogeneous transformation $\overline{g}_{2E} = \overline{g}_{23}\overline{g}_{3E}$

$$\begin{aligned} p_{2E} =& R_{23}p_{3E} + p_{23} \quad \text{(formula given in previous chapter)} \\ =& \begin{bmatrix} cos\left(q_3\right) & -sin\left(q_3\right) & 0 \\ sin\left(q_3\right) & cos\left(q_3\right) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} l_3 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} l_2 \\ 0 \\ 0 \end{bmatrix} \quad (R_{23} \text{ and } p_{23} \text{ extracted from } \overline{g}_{23}) \\ =& \begin{bmatrix} l_2 + l_3 cos(q_3) \\ l_3 sin(q_3) \\ 0 \end{bmatrix} \end{aligned}$$

Jacobian matrix using composition rule

Computation of translation elements :

Jacobian matrix using composition rule

Computation of translation elements :

- p_{1E} : translation vector extracted from the homogeneous transformation $\overline{g}_{1E} = \overline{g}_{12}\overline{g}_{2E}$

Jacobian matrix using composition rule

Computation of translation elements :

- p_{1E} : translation vector extracted from the homogeneous transformation $\overline{g}_{1E} = \overline{g}_{12}\overline{g}_{2E}$

$$
\begin{aligned}
p_{1E} =& R_{12}p_{2E} + p_{12} \quad \text{(formula given in previous chapter)} \\
=& \begin{bmatrix} cos\left(q_2\right) & -sin\left(q_2\right) & 0 \\ sin\left(q_2\right) & cos\left(q_2\right) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} l_2 + l_3 cos(q_3) \\ l_3 sin(q_3) \\ 0 \end{bmatrix} + \begin{bmatrix} l_1 \\ 0 \\ 0 \end{bmatrix} \\
=& \begin{bmatrix} l1 + l_2 cos(q_2) + l_3 cos(q_2 + q_3) \\ l_2 sin(q_2) + l_3 sin(q_2 + q_3) \\ 0 \end{bmatrix}
\end{aligned}
$$

Jacobian matrix using composition rule

Computation of translation elements :

- p_{1E} : translation vector extracted from the homogeneous transformation $\overline{g}_{1E} = \overline{g}_{12}\overline{g}_{2E}$

$$
\begin{aligned}
p_{1E} =& R_{12}p_{2E} + p_{12} \quad \text{(formula given in previous chapter)} \\
=& \begin{bmatrix} cos\left(q_2\right) & -sin\left(q_2\right) & 0 \\ sin\left(q_2\right) & cos\left(q_2\right) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} l_2 + l_3 cos(q_3) \\ l_3 sin(q_3) \\ 0 \end{bmatrix} + \begin{bmatrix} l_1 \\ 0 \\ 0 \end{bmatrix} \\
=& \begin{bmatrix} l1 + l_2 cos(q_2) + l_3 cos(q_2 + q_3) \\ l_2 sin(q_2) + l_3 sin(q_2 + q_3) \\ 0 \end{bmatrix}
\end{aligned}
$$

Jacobian matrix using composition rule

Computation of rotation elements :

Jacobian matrix using composition rule

Computation of rotation elements :

- R_{01} : rotation matrix of the homogeneous transformation $\overline{g}_{01}$ (already computed in previous example)

Jacobian matrix using composition rule

Computation of rotation elements :

- R_{01} : rotation matrix of the homogeneous transformation $\overline{g}_{01}$ (already computed in previous example)

$$R_{01} = R_{z_1, q_1} = \begin{bmatrix} cos\,(q_1) & -sin\,(q_1) & 0 \\ sin\,(q_1) & cos\,(q_1) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Jacobian matrix using composition rule

Computation of rotation elements :

- R_{01} : rotation matrix of the homogeneous transformation $\overline{g}_{01}$ (already computed in previous example)

$$R_{01} = R_{z_1,q_1} = \begin{bmatrix} cos\,(q_1) & -sin\,(q_1) & 0 \\ sin\,(q_1) & cos\,(q_1) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- R_{02} : rotation matrix of the homogeneous transformation $\overline{g}_{02} = \overline{g}_{01}\overline{g}_{12}$

Jacobian matrix using composition rule

Computation of rotation elements :

- R_{01} : rotation matrix of the homogeneous transformation $\overline{g}_{01}$ (already computed in previous example)

$$R_{01} = R_{z_1,q_1} = \begin{bmatrix} cos\,(q_1) & -sin\,(q_1) & 0 \\ sin\,(q_1) & cos\,(q_1) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- R_{02} : rotation matrix of the homogeneous transformation $\overline{g}_{02} = \overline{g}_{01}\overline{g}_{12}$

$$\begin{aligned} R_{02} =& R_{01} R_{12} \\ =& R_{z_1,q_1} R_{z_1=z_2,q_2} \\ =& R_{z_1,q_1+q_2} \\ =& \begin{bmatrix} cos\,(q_1+q_2) & -sin\,(q_1+q_2) & 0 \\ sin\,(q_1+q_2) & cos\,(q_1+q_2) & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

Jacobian matrix using composition rule

Computation of rotation elements :

- R_{01} : rotation matrix of the homogeneous transformation $\overline{g}_{01}$ (already computed in previous example)

$$R_{01} = R_{z_1,q_1} = \begin{bmatrix} cos\,(q_1) & -sin\,(q_1) & 0 \\ sin\,(q_1) & cos\,(q_1) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- R_{02} : rotation matrix of the homogeneous transformation $\overline{g}_{02} = \overline{g}_{01}\overline{g}_{12}$

$$\begin{aligned} R_{02} =& R_{01} R_{12} \\ =& R_{z_1,q_1} R_{z_1=z_2,q_2} \\ =& R_{z_1,q_1+q_2} \\ =& \begin{bmatrix} cos\,(q_1+q_2) & -sin\,(q_1+q_2) & 0 \\ sin\,(q_1+q_2) & cos\,(q_1+q_2) & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

- R_{03} : rotation matrix of the homogeneous transformation $\overline{g}_{03} = \overline{g}_{01}\overline{g}_{12}\overline{g}_{23}$

Jacobian matrix using composition rule

Computation of rotation elements :

- R_{01} : rotation matrix of the homogeneous transformation $\overline{g}_{01}$ (already computed in previous example)

$$R_{01} = R_{z_1,q_1} = \begin{bmatrix} cos\,(q_1) & -sin\,(q_1) & 0 \\ sin\,(q_1) & cos\,(q_1) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- R_{02} : rotation matrix of the homogeneous transformation $\overline{g}_{02} = \overline{g}_{01}\overline{g}_{12}$

$$\begin{aligned} R_{02} =& R_{01} R_{12} \\ =& R_{z_1,q_1} R_{z_1=z_2,q_2} \\ =& R_{z_1,q_1+q_2} \\ =& \begin{bmatrix} cos\,(q_1+q_2) & -sin\,(q_1+q_2) & 0 \\ sin\,(q_1+q_2) & cos\,(q_1+q_2) & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

- R_{03} : rotation matrix of the homogeneous transformation $\overline{g}_{03} = \overline{g}_{01}\overline{g}_{12}\overline{g}_{23}$

$$\begin{aligned} R_{03} =& R_{01} R_{12} R_{23} \\ =& R_{z_1,q_1+q_2+q_3} \\ =& \begin{bmatrix} cos\,(q_1+q_2+q_3) & -sin\,(q_1+q_2+q_3) & 0 \\ sin\,(q_1+q_2+q_3) & cos\,(q_1+q_2+q_3) & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

Jacobian matrix using composition rule

- Computation of ${}^{0}J_1$:

Jacobian matrix using composition rule

- Computation of ${}^{0}J_{1}$:

$$
{}^{0}J_{1}(q) = \begin{bmatrix} R_{01}\left(Z_{1} \times p_{1E}\right) \\ R_{01}Z_{1} \end{bmatrix}
= \begin{bmatrix} R_{01}\begin{pmatrix} -l_{2}sin(q_{2}) - l_{3}sin(q_{2}+q_{3}) \\ l_{1} + l_{2}cos(q_{2}) + l_{3}cos(q_{2}+q_{3}) \\ 0 \end{pmatrix} \\ R_{01}\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \end{bmatrix}
= \begin{pmatrix} -l_{1}S1 - l_{2}S12 - l_{3}S123 \\ l_{1}C1 + l_{2}C12 + l_{3}C123 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}
$$

Jacobian matrix using composition rule

- Computation of ${}^0J_2$:

Jacobian matrix using composition rule

- Computation of ${}^0J_2$:

$$
{}^0J_2(q) = \begin{bmatrix} R_{02}\left(Z_2 \times p_{2E}\right) \\ R_{02}Z_2 \end{bmatrix}
= \begin{bmatrix} R_{02}\begin{pmatrix} -l_3 sin(q_3) \\ l_2 + l_3 cos(q_3) \\ 0 \end{pmatrix} \\ R_{02}\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \end{bmatrix}
= \begin{pmatrix} -l_2 S12 - l_3 S123 \\ l_2 C12 + l_3 C123 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}
$$

Jacobian matrix using composition rule

- Computation of ${}^0J_3$:

Jacobian matrix using composition rule

- Computation of ${}^{0}J_3$:

$$
{}^{0}J_3(q) = \begin{bmatrix} R_{03}\left(Z_3 \times p_{3E}\right) \\ R_{03} Z_3 \end{bmatrix}
= \begin{bmatrix} R_{03} \begin{pmatrix} 0 \\ l_3 \\ 0 \end{pmatrix} \\ R_{03} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \end{bmatrix}
= \begin{pmatrix} -l_3 S123 \\ l_3 C123 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}
$$

Jacobian matrix using composition rule

- Computation of ${}^{0}J$:

Jacobian matrix using composition rule

- Computation of ${}^0J$:

$$
{}^0J(q) = \left[\begin{array}{ccc} {}^0J_1(q) & {}^0J_2(q) & {}^0J_E(q) \end{array} \right]
$$

$$
= \begin{bmatrix} -l_1S1 - l_2S12 - l_3S123 & -l_2S12 - l_3S123 & -l_3S123 \\ l_1C1 + l_2C12 + l_3C123 & l_2C12 + l_3C123 & l_3C123 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}
$$

Course

Analysis of the Jacobian matrix ${}^0J$ of the PPP robot

Rank of ${}^0J$

- Rank of ${}^0J$: equal to the dimension of the subspace corresponding to the cartesian velocities caused by the robot, image of the joint velocity subspace through its Jacobian matrix ${}^0J$:

Analysis of the Jacobian matrix ${}^0J$ of the PPP robot

Rank of ${}^0J$

- Rank of ${}^0J$: equal to the dimension of the subspace corresponding to the cartesian velocities caused by the robot, image of the joint velocity subspace through its Jacobian matrix ${}^0J$:

$$\begin{aligned} \text{Rank}\left({}^0J(q)\right) &= \text{Rank}\left(\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}\right) \\ &= 2 \qquad \forall q \\ &= r_{max} \\ &= M \\ &< 3 \qquad \left(= N = \dim\left(\mathbb{R}^{\dot{q}}\right)\right) \end{aligned}$$

Analysis of the Jacobian matrix ${}^0J$ of the PPP robot

Rank of ${}^0J$

- Rank of ${}^0J$: equal to the dimension of the subspace corresponding to the cartesian velocities caused by the robot, image of the joint velocity subspace through its Jacobian matrix ${}^0J$:

$$\begin{aligned}
\text{Rank}\left({}^0J\,(q)\right) &= \text{Rank}\left(\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}\right) \\
&= 2 \qquad \forall q \\
&= r_{max} \\
&= M \\
&< 3 \qquad \left(= N = \dim\left(\mathbb{R}^{\dot{q}}\right)\right)
\end{aligned}$$

- Redundancy of order 1.

Analysis of the Jacobian matrix ${}^0J$ of the PPP robot

Rank of ${}^0J$

- Rank of ${}^0J$: equal to the dimension of the subspace corresponding to the cartesian velocities caused by the robot, image of the joint velocity subspace through its Jacobian matrix ${}^0J$:

$$\begin{aligned}\text{Rank}\left({}^0J\left(q\right)\right) &= \text{Rank}\left(\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}\right) \\ &= 2 \qquad \forall q \\ &= r_{max} \\ &= M \\ &< 3 \qquad \left(= N = \dim\left(\mathbb{R}^{\dot{q}}\right)\right)\end{aligned}$$

- Redundancy of order 1.

Analysis of the Jacobian matrix ${}^0J$ of the PPP robot

Analysis of null of ${}^0J$

- The redundant robots (for which $\dim\left(\mathcal{N}\left({}^0J\right)\right) > 0$) have an infinite number of joint velocities that produce the same velocity of the end-effector.

Analysis of the Jacobian matrix ${}^0J$ of the PPP robot

Analysis of null of ${}^0J$

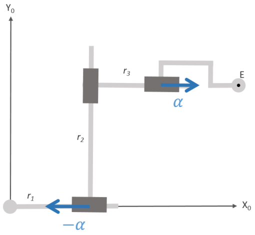
- The redundant robots (for which $\dim\left(\mathcal{N}\left({}^0J\right)\right) > 0$) have an infinite number of joint velocities that produce the same velocity of the end-effector.
  - A basis for the null space of ${}^0J$ is generated by the vector $\begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}$.

Analysis of the Jacobian matrix ${}^0J$ of the PPP robot

Analysis of null of ${}^0J$

- The redundant robots (for which $\dim\left(\mathcal{N}\left({}^0J\right)\right) > 0$) have an infinite number of joint velocities that produce the same velocity of the end-effector.
  - A basis for the null space of ${}^0J$ is generated by the vector $\begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}$.
  - Subspace of the joint velocities producing the same task velocity :

$$\dot{X}_d = {}^0J\left(\dot{q}_{Im} + \alpha\begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}\right) \text{ with } \alpha \in \mathbb{R}$$



Course

Singularity of RR robot

- Recall on the expression of ${}^0J$:

$$^0J\,(q_1\,,\,q_2) = \begin{bmatrix} -l_1S1 - l_2S12 & -l_2S12 \\ l_1C1 + l_2C12 & l_2C12 \end{bmatrix}$$

- Calculation of determinant of ${}^0J$:

Singularity of RR robot

- Recall on the expression of ${}^{0}J$:

$$ {}^{0}J\left(q_1, q_2\right) = \begin{bmatrix} -l_1 S1 - l_2 S12 & -l_2 S12 \\ l_1 C1 + l_2 C12 & l_2 C12 \end{bmatrix} $$

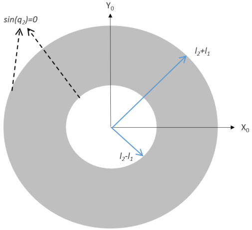
- Calculation of determinant of ${}^{0}J$:

$$ \begin{aligned} det\left({}^{0}J\right) =& l_1 l_2 \left(cos\left(q_1\right) sin\left(q_1 + q_2\right) - sin\left(q_1\right) cos\left(q_1 + q_2\right)\right) \\ =& l_1 l_2 \left[cos\left(q_1\right)\left(sin\left(q_1\right) cos\left(q_2\right) + cos\left(q_1\right) sin\left(q_2\right)\right)\right. \\ & \left. \ldots - sin\left(q_1\right)\left(cos\left(q_1\right) cos\left(q_2\right) - sin\left(q_1\right) sin\left(q_2\right)\right)\right] \\ =& l_1 l_2 \left(cos^2\left(q_1\right) + sin^2\left(q_1\right)\right) sin\left(q_2\right) \\ =& l_1 l_2 sin\left(q_2\right) \end{aligned} $$

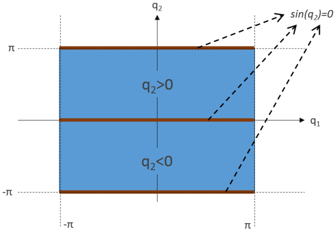
- Singularities in $q_2 = 0$ and $q_2 = \pm\pi$ (solutions of $det\left({}^{0}J\right) = 0$).

Singularity of RR robot

- Workspace of the robot ($L_1 > L_2$)



- Singularity branches of the planar robot



Transmission of forces/torques between joint and task spaces

The operator $\delta\,(.)$ represents a virtual displacement.

Transmission of forces/torques between joint and task spaces

The operator $\delta\,(.)$ represents a virtual displacement.
A virtual displacement produced by the end-effector in the task space is given by $\delta X = J\delta q$.

Transmission of forces/torques between joint and task spaces

The operator $\delta(.)$ represents a virtual displacement.
A virtual displacement produced by the end-effector in the task space is given by $\delta X = J\delta q$.
In the task space, the virtual mechanical work resulting from the static wrench ${}^0\mathscr{F}_e$ is given by ${}^0\mathscr{F}_e^t \delta X$.

Transmission of forces/torques between joint and task spaces

The operator $\delta\,(.)$ represents a virtual displacement.
A virtual displacement produced by the end-effector in the task space is given by $\delta X = J\delta q$.
In the task space, the virtual mechanical work resulting from the static wrench ${}^{0}\mathscr{F}_e$ is given by ${}^{0}\mathscr{F}_e^{t}\delta X$.
To generate this virtual mechanical work, the robot provides a virtual work due to its joint torques according to $\Gamma_e^t\delta q$.

Transmission of forces/torques between joint and task spaces

The operator $\delta(.)$ represents a virtual displacement.
A virtual displacement produced by the end-effector in the task space is given by $\delta X = J\delta q$.
In the task space, the virtual mechanical work resulting from the static wrench ${}^0\mathscr{F}_e$ is given by ${}^0\mathscr{F}_e^t \delta X$.
To generate this virtual mechanical work, the robot provides a virtual work due to its joint torques according to $\Gamma_e^t \delta q$.
The variation of the total mechanical work that is generated by the system

Transmission of forces/torques between joint and task spaces

The operator $\delta(.)$ represents a virtual displacement.
A virtual displacement produced by the end-effector in the task space is given by $\delta X = J\delta q$.
In the task space, the virtual mechanical work resulting from the static wrench ${}^0\mathscr{F}_e$ is given by ${}^0\mathscr{F}_e^t \delta X$.
To generate this virtual mechanical work, the robot provides a virtual work due to its joint torques according to $\Gamma_e^t \delta q$.
The variation of the total mechanical work that is generated by the system

$$\begin{aligned} \delta\mathscr{W} &= {}^0\mathscr{F}_e^t \delta X - \Gamma_e^t \delta q \\ &= \left({}^0\mathscr{F}_e^t J - \Gamma_e^t\right) \delta q \end{aligned}$$

Transmission of forces/torques between joint and task spaces

The operator $\delta(.)$ represents a virtual displacement.
A virtual displacement produced by the end-effector in the task space is given by $\delta X = J\delta q$.
In the task space, the virtual mechanical work resulting from the static wrench ${}^0\mathscr{F}_e$ is given by ${}^0\mathscr{F}_e^t \delta X$.
To generate this virtual mechanical work, the robot provides a virtual work due to its joint torques according to $\Gamma_e^t \delta q$.
The variation of the total mechanical work that is generated by the system

$$\begin{aligned}
\delta\mathscr{W} &= {}^0\mathscr{F}_e^t \delta X - \Gamma_e^t \delta q \\
&= \left({}^0\mathscr{F}_e^t J - \Gamma_e^t\right)\delta q
\end{aligned}$$

must be null at the equilibrium state, thus :

Transmission of forces/torques between joint and task spaces

The operator $\delta(.)$ represents a virtual displacement.
A virtual displacement produced by the end-effector in the task space is given by $\delta X = J\delta q$.
In the task space, the virtual mechanical work resulting from the static wrench ${}^{0}\mathscr{F}_e$ is given by ${}^{0}\mathscr{F}_e^t \delta X$.
To generate this virtual mechanical work, the robot provides a virtual work due to its joint torques according to $\Gamma_e^t \delta q$.
The variation of the total mechanical work that is generated by the system

$$\begin{aligned} \delta\mathscr{W} &= {}^{0}\mathscr{F}_e^t \delta X - \Gamma_e^t \delta q \\ &= \left({}^{0}\mathscr{F}_e^t J - \Gamma_e^t\right) \delta q \end{aligned}$$

must be null at the equilibrium state, thus :

$$\delta\mathscr{W} = 0 \Leftrightarrow \boxed{\Gamma_e = J^{t\,0}\mathscr{F}_e}$$

Course

IGM for the 2R robot

- Searching for the joint variable q_1
  When putting to square and when summing the two equations coming from the DGM :

$$p_x^2 + p_y^2 - (l_1 + l_2)^2 = 2 l_1 l_2 \left(cos\left(q_1\right) cos\left(q_1 + q_2\right) + sin\left(q_1\right) sin\left(q_1 + q_2\right)\right)$$
$$= 2 l_1 l_2 cos\left(q_2\right)$$

which we deduce that :

$$cos\left(q_2\right) = \frac{p_x^2 + p_y^2 - (l_1 + l_2)^2}{2 l_1 l_2}$$

(must belong to $[-1, 1]$, otherwise E is out of the workspace)

IGM for the 2R robot

- Searching for the joint variable q_1
  When putting to square and when summing the two equations coming from the DGM :

$$\begin{aligned} p_x^2 + p_y^2 - (l_1 + l_2)^2 &= 2 l_1 l_2 \left(cos\left(q_1\right) cos\left(q_1 + q_2\right) + sin\left(q_1\right) sin\left(q_1 + q_2\right)\right) \\ &= 2 l_1 l_2 cos\left(q_2\right) \end{aligned}$$

  which we deduce that :

$$cos\left(q_2\right) = \frac{p_x^2 + p_y^2 - (l_1 + l_2)^2}{2 l_1 l_2}$$

  (must belong to $[-1, 1]$, otherwise E is out of the workspace)
  and

$$sin\left(q_2\right) = \pm\sqrt{1 - cos^2\left(q_2\right)}$$

  (2 solutions according to the sign).

IGM for the 2R robot

- Searching for the joint variable q_1
  When putting to square and when summing the two equations coming from the DGM :

$$p_x^2 + p_y^2 - (l_1 + l_2)^2 = 2 l_1 l_2 \left(cos\left(q_1\right) cos\left(q_1 + q_2\right) + sin\left(q_1\right) sin\left(q_1 + q_2\right)\right)$$
$$= 2 l_1 l_2 cos\left(q_2\right)$$

  which we deduce that :

$$cos\left(q_2\right) = \frac{p_x^2 + p_y^2 - (l_1 + l_2)^2}{2 l_1 l_2}$$

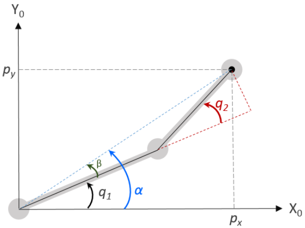
  (must belong to $[-1, 1]$, otherwise E is out of the workspace)
  and

$$sin\left(q_2\right) = \pm\sqrt{1 - cos^2\left(q_2\right)}$$

  (2 solutions according to the sign).
  Finally, $\boxed{q_2 = atan2\left(sin\left(q_2\right), cos\left(q_2\right)\right)}$ under analytical form.

IGM for the 2R robot

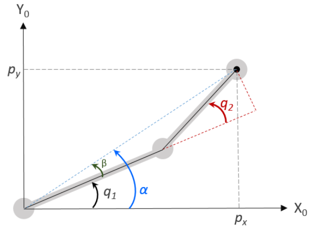
- Searching for the joint variable q_1



Two solutions have to be searched for q_1 (one for each value of $sin(q_2)$).

IGM for the 2R robot

- Searching for the joint variable q_1

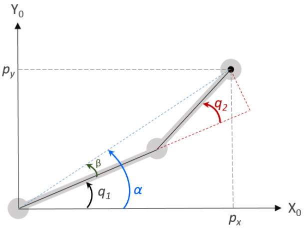


Two solutions have to be searched for q_1 (one for each value of $sin\left(q_2\right)$).
By geometrical inspection :

$$\boxed{\mathrm{q_1}} = \alpha - \beta \boxed{= \mathrm{atan2}(p_y, p_x) - atan2\left(l_2 sin\left(q_2\right), l_1 + l_2 cos\left(q_2\right)\right)}$$

IGM for the 2R robot

- Searching for the joint variable q_1



Two solutions have to be searched for q_1 (one for each value of $sin\,(q_2)$).
By geometrical inspection :

$$\boxed{\mathrm{q}_1} = \alpha - \beta \boxed{= \mathrm{atan2}(p_y, p_x) - atan2\,(l_2 sin\,(q_2)\,, l_1 + l_2 cos\,(q_2))}$$

Remarks :

- Note : the difference of $atan2$ functions above must be re-calculated in $(-\pi, \pi]$;
- Two configuration solutions : elbow up $\left(q_1^{''}, q_2^{''}\right)$ and elbow down $\left(q_1^{'}, q_2^{'}\right)$.

Course

Stability of the gradient-based algorithm

- Let e be the tracking error, such that $e = X_d - f(q)$.
  $e \rightarrow 0$ iff the equilibrium point of the closed loop $e = 0$ is asymptotically stable.
- Choice about the *Lyapunov* candidate function : $V = \frac{1}{2} e^t e \geq 0$.
- Derivative w.r.t. time of the *Lyapunov* function :

$$\begin{aligned}\dot{V} &= e^t \dot{e} \\ &= e^t \frac{d}{dt} \left(X_d - f(q)\right) \\ &= -e^t J \dot{q} \\ &= -e^t J J^t e \\ &\leq 0\end{aligned}$$

Thus,

$$\dot{V} = 0 \Leftrightarrow e \in N\left(J^t\right)$$

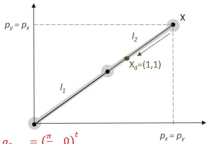
in particular $e = 0$.

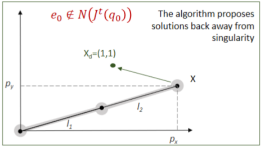
Course

Algorithm for the numerical computation of IGM in the regular case

Case study of robot 2R when considering $X_d = (1, 1)$ and $l_1 = l_2 = 1$.

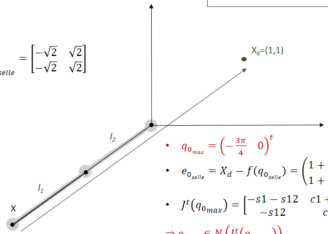
- Case of non-convergence when the Jacobian matrix $J(q)$ is singular in q and when the error e belongs to the null of $J^t(q)$ (recall that $\mathscr{N}(J^t) = \mathscr{R}(J)^\perp$: set of unfeasible velocities).





- $q_{0_{selle}} = \left(\frac{\pi}{4} \quad 0\right)^t$
- $e_{0_{selle}} = X_d - f(q_{0_{selle}}) = \begin{pmatrix} 1-\sqrt{2} \\ 1-\sqrt{2} \end{pmatrix}$
- $J^t(q_{0_{selle}}) = \begin{bmatrix} -s1-s12 & c1+c12 \\ -s12 & c12 \end{bmatrix}_{q=q_{0selle}} = \begin{bmatrix} -\sqrt{2} & \sqrt{2} \\ -\sqrt{2} & \sqrt{2} \end{bmatrix}$

$\Rightarrow e_{0_{selle}} \in N\left(J^t(q_{0_{selle}})\right)$



- $q_{0_{max}} = \left(-\frac{3\pi}{4} \quad 0\right)^t$
- $e_{0_{selle}} = X_d - f(q_{0_{selle}}) = \begin{pmatrix} 1+\sqrt{2} \\ 1+\sqrt{2} \end{pmatrix}$
- $J^t(q_{0_{max}}) = \begin{bmatrix} -s1-s12 & c1+c12 \\ -s12 & c12 \end{bmatrix}_{q=q_{0max}} = \begin{bmatrix} \sqrt{2} & -\sqrt{2} \\ \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \end{bmatrix}$

$\Rightarrow e_{0_{max}} \in N\left(J^t(q_{0_{max}})\right)$

Course

Inversion of the kinematic model in the redundant case

Problem that can be seen as a minimisation problem without constraints if we use $\lambda \in \mathbb{R}^m$, whose coordinates λ_i are the *Lagrange* multipliers.

The Lagrangian $\mathscr{L}$ associated to the optimisation problem becomes :

$$\mathscr{L}(\dot{q}, \lambda) = \frac{1}{2} (\dot{q} - \dot{q}_0)^t W (\dot{q} - \dot{q}_0) + \lambda^t \left(J(q)\, \dot{q} - \dot{X}_d \right)$$

Inversion of the kinematic model in the redundant case

Problem that can be seen as a minimisation problem without constraints if we use $\lambda \in \mathbb{R}^m$, whose coordinates λ_i are the *Lagrange* multipliers.
The Lagrangian $\mathscr{L}$ associated to the optimisation problem becomes :

$$\mathscr{L}(\dot{q}, \lambda) = \frac{1}{2}(\dot{q} - \dot{q}_0)^t W (\dot{q} - \dot{q}_0) + \lambda^t \left(J(q)\dot{q} - \dot{X}_d \right)$$

- Necessary condition
  If $\dot{q}^*$ is a searched solution, then it exists a vector λ^*, such that the function $\mathscr{L}$ admits a differential that is null at the stationary point $(\dot{q}^*, \lambda^*)$:

$$\nabla_{\dot{q},\lambda}\mathscr{L}|_{(\dot{q}^*,\lambda^*)} = 0 \Leftrightarrow \begin{cases} \nabla_{\dot{q}}\mathscr{L}|_{(\dot{q}^{*t},\lambda^*)} = W(\dot{q}^* - \dot{q}_0) + J^t\lambda^* = 0 \\ \nabla_{\lambda}\mathscr{L}|_{(\dot{q}^{*t},\lambda^*)} = J\dot{q}^* - \dot{X}_d = 0 \end{cases}$$

Inversion of the kinematic model in the redundant case

Problem that can be seen as a minimisation problem without constraints if we use $\lambda \in \mathbb{R}^m$, whose coordinates λ_i are the *Lagrange* multipliers.
The Lagrangian $\mathscr{L}$ associated to the optimisation problem becomes :

$$\mathscr{L}(\dot{q}, \lambda) = \frac{1}{2}(\dot{q} - \dot{q}_0)^t W (\dot{q} - \dot{q}_0) + \lambda^t \left(J(q)\dot{q} - \dot{X}_d\right)$$

- Necessary condition
  If $\dot{q}^*$ is a searched solution, then it exists a vector λ^*, such that the function $\mathscr{L}$ admits a differential that is null at the stationary point $(\dot{q}^*, \lambda^*)$:

$$\nabla_{\dot{q},\lambda}\mathscr{L}|_{(\dot{q}^*,\lambda^*)} = 0 \Leftrightarrow \begin{cases} \nabla_{\dot{q}}\mathscr{L}|_{(\dot{q}^{*t},\lambda^*)} = W(\dot{q}^* - \dot{q}_0) + J^t\lambda^* = 0 \\ \nabla_{\lambda}\mathscr{L}|_{(\dot{q}^{*t},\lambda^*)} = J\dot{q}^* - \dot{X}_d = 0 \end{cases}$$

  from which we deduce, from the 1^{st} equation, that $\dot{q}^* = \dot{q}_0 - W^{-1}J^t\lambda^*$.

Inversion of the kinematic model in the redundant case

Problem that can be seen as a minimisation problem without constraints if we use $\lambda \in \mathbb{R}^m$, whose coordinates λ_i are the *Lagrange* multipliers.
The Lagrangian $\mathscr{L}$ associated to the optimisation problem becomes :

$$\mathscr{L}(\dot{q}, \lambda) = \frac{1}{2}(\dot{q} - \dot{q}_0)^t W (\dot{q} - \dot{q}_0) + \lambda^t \left(J(q)\dot{q} - \dot{X}_d \right)$$

- Necessary condition
  If $\dot{q}^*$ is a searched solution, then it exists a vector λ^*, such that the function $\mathscr{L}$ admits a differential that is null at the stationary point $(\dot{q}^*, \lambda^*)$:

$$\nabla_{\dot{q},\lambda}\mathscr{L}|_{(\dot{q}^*,\lambda^*)} = 0 \Leftrightarrow \begin{cases} \nabla_{\dot{q}}\mathscr{L}|_{(\dot{q}^{*t},\lambda^*)} = W(\dot{q}^* - \dot{q}_0) + J^t\lambda^* = 0 \\ \nabla_{\lambda}\mathscr{L}|_{(\dot{q}^{*t},\lambda^*)} = J\dot{q}^* - \dot{X}_d = 0 \end{cases}$$

  from which we deduce, from the 1^{st} equation, that $\dot{q}^* = \dot{q}_0 - W^{-1}J^t\lambda^*$.
  When combining with the 2^{nd} equation, it folows that $J\dot{q}_0 - JW^{-1}J^t\lambda^* - \dot{X}_d = 0$, thus $\lambda^* = \left(JW^{-1}J^t\right)^{-1}\left(J\dot{q}_0 - \dot{X}_d\right)$, since the matrix $\left(JW^{-1}J^t\right) \in R^{m\times m}$ is well inversible (if J is full raw rank).

Inversion of the kinematic model in the redundant case

Problem that can be seen as a minimisation problem without constraints if we use $\lambda \in \mathbb{R}^m$, whose coordinates λ_i are the *Lagrange* multipliers.
The Lagrangian $\mathscr{L}$ associated to the optimisation problem becomes :

$$\mathscr{L}(\dot{q}, \lambda) = \frac{1}{2}(\dot{q} - \dot{q}_0)^t W (\dot{q} - \dot{q}_0) + \lambda^t \left(J(q)\dot{q} - \dot{X}_d \right)$$

- Necessary condition
  If $\dot{q}^*$ is a searched solution, then it exists a vector λ^*, such that the function $\mathscr{L}$ admits a differential that is null at the stationary point $(\dot{q}^*, \lambda^*)$:

$$\nabla_{\dot{q},\lambda}\mathscr{L}|_{(\dot{q}^*,\lambda^*)} = 0 \Leftrightarrow \begin{cases} \nabla_{\dot{q}}\mathscr{L}|_{(\dot{q}^{*t},\lambda^*)} = W(\dot{q}^* - \dot{q}_0) + J^t\lambda^* = 0 \\ \nabla_{\lambda}\mathscr{L}|_{(\dot{q}^{*t},\lambda^*)} = J\dot{q}^* - \dot{X}_d = 0 \end{cases}$$

from which we deduce, from the 1^{st} equation, that $\dot{q}^* = \dot{q}_0 - W^{-1}J^t\lambda^*$.
When combining with the 2^{nd} equation, it folows that $J\dot{q}_0 - JW^{-1}J^t\lambda^* - \dot{X}_d = 0$, thus $\lambda^* = \left(JW^{-1}J^t\right)^{-1}\left(J\dot{q}_0 - \dot{X}_d\right)$, since the matrix $\left(JW^{-1}J^t\right) \in R^{m\times m}$ is well inversible (if J is full raw rank).

$$\dot{q}^* = \dot{q}_0 - W^{-1}J^t\left(JW^{-1}J^t\right)^{-1}\left(J\dot{q}_0 - \dot{X}_d\right)$$

$$= \underbrace{W^{-1}J^t\left(JW^{-1}J^t\right)^{-1}}_{J_W^\sharp}\dot{X}_d + \left(I_n - \underbrace{W^{-1}J^t\left(JW^{-1}J^t\right)^{-1}}_{J_W^\sharp}J\right)\dot{q}_0$$

Inversion of the kinematic model in the redundant case

- Sufficient condition :
  The stationary point $(\dot{q}^*, \lambda^*)$ is a local extremum (local minimum in our case) if the Hessian matrix of $\mathscr{L}$ is positive-definite :

$$\nabla^2_{\dot{q}\dot{q}}\, \mathscr{L}|_{(\dot{q}^*, \lambda^*)} \succ 0$$

Inversion of the kinematic model in the redundant case

- Sufficient condition :
  The stationary point $(\dot{q}^*, \lambda^*)$ is a local extremum (local minimum in our case) if the Hessian matrix of $\mathscr{L}$ is positive-definite :

  $$\nabla^2_{\dot{q}\dot{q}} \mathscr{L}|_{(\dot{q}^*,\lambda^*)} \succ 0$$

  This condition is verified since $\nabla^2_{\dot{q}\dot{q}} \mathscr{L}|_{(\dot{q}^*,\lambda^*)} = W$ with W chosen as being definite-positive by hypothesis.

Inversion of the kinematic model in the redundant case

- Sufficient condition :
  The stationary point $(\dot{q}^*, \lambda^*)$ is a local extremum (local minimum in our case) if the Hessian matrix of $\mathscr{L}$ is positive-definite :

$$\nabla^2_{\dot{q}\dot{q}} \mathscr{L}|_{(\dot{q}^*, \lambda^*)} \succ 0$$

  This condition is verified since $\nabla^2_{\dot{q}\dot{q}} \mathscr{L}|_{(\dot{q}^*, \lambda^*)} = W$ with W chosen as being definite-positive by hypothesis.

In fact, the point $(\dot{q}^*, \lambda^*)$ is even the **unique and global** extremum of the optimisation problem, since the quadratic objective function is strictly convex (since W is definite positive).

Course

Secondary task in operational space

- Joint configuration as solution of the task 1 : $\dot{q}^* = J_1^\sharp \dot{X}_{d_1} + N_{J_1} \dot{q}_{0_1}$

Secondary task in operational space

- Joint configuration as solution of the task 1 : $\dot{q}^* = J_1^\sharp \dot{X}_{d_1} + N_{J_1} \dot{q}_{0_1}$
- Searching for the joint configuration as solution of the task 2 :

Secondary task in operational space

- Joint configuration as solution of the task 1 : $\dot{q}^* = J_1^\sharp \dot{X}_{d_1} + N_{J_1} \dot{q}_{0_1}$
- Searching for the joint configuration as solution of the task 2 :
  - While injecting the previous solution (so as to respect the 1^{st} objective) in the expression coming from the task 2 :

$$\dot{X}_{d_2} = J_2 \dot{q}^* = J_2 \left(J_1^\sharp \dot{X}_{d_1} + N_{J_1} \dot{q}_{0_1} \right)$$

Secondary task in operational space

- Joint configuration as solution of the task 1 : $\dot{q}^* = J_1^\sharp \dot{X}_{d_1} + N_{J_1} \dot{q}_{0_1}$
- Searching for the joint configuration as solution of the task 2 :
  - While injecting the previous solution (so as to respect the 1^{st} objective) in the expression coming from the task 2 :

$$\dot{X}_{d_2} = J_2 \dot{q}^* = J_2 \left(J_1^\sharp \dot{X}_{d_1} + N_{J_1} \dot{q}_{0_1} \right)$$

  - The vector $\dot{q}_{0_1}$ is deduced as solution of the previous equation (in which we choose the *internal motions* null, i.e. $\dot{q}_{0_2}$) :

$$\dot{q}_{0_1} = \left(J_2 N_{J_1} \right)^\sharp \left(\dot{X}_{d_2} - J_2 J_1^\sharp \dot{X}_{d_1} \right)$$

Secondary task in operational space

- Joint configuration as solution of the task 1 : $\dot{q}^* = J_1^\sharp \dot{X}_{d_1} + N_{J_1} \dot{q}_{0_1}$
- Searching for the joint configuration as solution of the task 2 :
  - While injecting the previous solution (so as to respect the 1^{st} objective) in the expression coming from the task 2 :

$$\dot{X}_{d_2} = J_2 \dot{q}^* = J_2 \left(J_1^\sharp \dot{X}_{d_1} + N_{J_1} \dot{q}_{0_1} \right)$$

  - The vector $\dot{q}_{0_1}$ is deduced as solution of the previous equation (in which we choose the *internal motions* null, i.e. $\dot{q}_{0_2}$) :

$$\dot{q}_{0_1} = \left(J_2 N_{J_1} \right)^\sharp \left(\dot{X}_{d_2} - J_2 J_1^\sharp \dot{X}_{d_1} \right)$$

  - While substituting the found expression of $\dot{q}_{0_1}$ in the expression for task 1 :

$$\dot{q}^* = J_1^\sharp \dot{X}_{d_1} + N_{J_1} \left(J_2 N_{J_1} \right)^\sharp \left(\dot{X}_{d_2} - J_2 J_1^\sharp \dot{X}_{d_1} \right)$$

Secondary task in operational space

- Joint configuration as solution of the task 1 : $\dot{q}^* = J_1^\sharp \dot{X}_{d_1} + N_{J_1} \dot{q}_{0_1}$
- Searching for the joint configuration as solution of the task 2 :
  - While injecting the previous solution (so as to respect the 1^{st} objective) in the expression coming from the task 2 :

$$\dot{X}_{d_2} = J_2 \dot{q}^* = J_2 \left(J_1^\sharp \dot{X}_{d_1} + N_{J_1} \dot{q}_{0_1} \right)$$

  - The vector $\dot{q}_{0_1}$ is deduced as solution of the previous equation (in which we choose the *internal motions* null, i.e. $\dot{q}_{0_2}$) :

$$\dot{q}_{0_1} = \left(J_2 N_{J_1} \right)^\sharp \left(\dot{X}_{d_2} - J_2 J_1^\sharp \dot{X}_{d_1} \right)$$

  - While substituting the found expression of $\dot{q}_{0_1}$ in the expression for task 1 :

$$\dot{q}^* = J_1^\sharp \dot{X}_{d_1} + N_{J_1} \left(J_2 N_{J_1} \right)^\sharp \left(\dot{X}_{d_2} - J_2 J_1^\sharp \dot{X}_{d_1} \right)$$

- Taking into account the properties of the pseudo-inverse :

$$N_{J_1} \left(J_2 N_{J_1} \right)^\sharp = \left(J_2 N_{J_1} \right)^\sharp,$$

Secondary task in operational space

- Joint configuration as solution of the task 1 : $\dot{q}^* = J_1^\sharp \dot{X}_{d_1} + N_{J_1} \dot{q}_{0_1}$
- Searching for the joint configuration as solution of the task 2 :
  - While injecting the previous solution (so as to respect the 1^{st} objective) in the expression coming from the task 2 :

$$\dot{X}_{d_2} = J_2 \dot{q}^* = J_2 \left(J_1^\sharp \dot{X}_{d_1} + N_{J_1} \dot{q}_{0_1} \right)$$

  - The vector $\dot{q}_{0_1}$ is deduced as solution of the previous equation (in which we choose the *internal motions* null, i.e. $\dot{q}_{0_2}$) :

$$\dot{q}_{0_1} = \left(J_2 N_{J_1} \right)^\sharp \left(\dot{X}_{d_2} - J_2 J_1^\sharp \dot{X}_{d_1} \right)$$

  - While substituting the found expression of $\dot{q}_{0_1}$ in the expression for task 1 :

$$\dot{q}^* = J_1^\sharp \dot{X}_{d_1} + N_{J_1} \left(J_2 N_{J_1} \right)^\sharp \left(\dot{X}_{d_2} - J_2 J_1^\sharp \dot{X}_{d_1} \right)$$

- Taking into account the properties of the pseudo-inverse :

$$N_{J_1} \left(J_2 N_{J_1} \right)^\sharp = \left(J_2 N_{J_1} \right)^\sharp,$$

Finally,

$$\dot{q}^* = J_1^\sharp \dot{X}_{d_1} + \left(J_2 N_{J_1} \right)^\sharp \left(\dot{X}_{d_2} - J_2 J_1^\sharp \dot{X}_{d_1} \right)$$

Secondary task in the operational space

1 - Search for the joint velocity $\dot{q}_1^*(t)$ respecting the 1^{st} task :

Secondary task in the operational space

1 - Search for the joint velocity $\dot{q}_1^*(t)$ respecting the 1^{st} task :

- Coming back to the expression of matrix ${}^0J$ for planar RRR robot :

$$\begin{pmatrix} {}^0\dot{x}_d^E \\ {}^0\dot{y}_d^E \end{pmatrix} = {}^0J(q) \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{pmatrix}$$

where

$${}^0J(q) = \begin{bmatrix} -l_1S1 - l_2S12 - l_3S123 & -l_2S12 - l_3S123 & -l_3S123 \\ l_1C1 + l_2C12 + l_3C123 & l_2C12 + l_3C123 & l_3C123 \end{bmatrix}$$

Secondary task in the operational space

1 - Search for the joint velocity $\dot{q}_1^*(t)$ respecting the 1^{st} task :

- Coming back to the expression of matrix ${}^0J$ for planar RRR robot :

$$\begin{pmatrix} {}^0\dot{x}_d^E \\ {}^0\dot{y}_d^E \end{pmatrix} = {}^0J(q) \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{pmatrix}$$

where

$${}^0J(q) = \begin{bmatrix} -l_1S1 - l_2S12 - l_3S123 & -l_2S12 - l_3S123 & -l_3S123 \\ l_1C1 + l_2C12 + l_3C123 & l_2C12 + l_3C123 & l_3C123 \end{bmatrix}$$

Numerical evaluation at the configuration $q = \left(\frac{\pi}{3}, \frac{2\pi}{3}, \frac{\pi}{3}\right)^t$:

$${}^0J(q) = \frac{1}{2}\begin{bmatrix} 0 & \sqrt{3} & \sqrt{3} \\ -2 & -3 & -1 \end{bmatrix}$$

Secondary task in the operational space

1 - Search for the joint velocity $\dot{q}_1^*(t)$ respecting the 1^{st} task :

- Coming back to the expression of matrix ${}^0J$ for planar RRR robot :

$$\begin{pmatrix} {}^0\dot{x}_d^E \\ {}^0\dot{y}_d^E \end{pmatrix} = {}^0J(q) \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{pmatrix}$$

where

$${}^0J(q) = \begin{bmatrix} -l_1S1 - l_2S12 - l_3S123 & -l_2S12 - l_3S123 & -l_3S123 \\ l_1C1 + l_2C12 + l_3C123 & l_2C12 + l_3C123 & l_3C123 \end{bmatrix}$$

Numerical evaluation at the configuration $q = \left(\frac{\pi}{3}, \frac{2\pi}{3}, \frac{\pi}{3}\right)^t$:

$${}^0J(q) = \frac{1}{2}\begin{bmatrix} 0 & \sqrt{3} & \sqrt{3} \\ -2 & -3 & -1 \end{bmatrix}$$

- After computing the *Moore-Penrose* pseudo-inverse :

$$\dot{q}_1^* = {}^0J^{\sharp\,0}\dot{X}_d^E = \frac{1}{3}\begin{bmatrix} -\frac{4\sqrt{3}}{3} & -2 \\ \frac{\sqrt{3}}{3} & -1 \\ \frac{5\sqrt{3}}{3} & 1 \end{bmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{3}\begin{pmatrix} -\frac{4\sqrt{3}}{3} - 2 \\ \frac{\sqrt{3}}{3} - 1 \\ \frac{5\sqrt{3}}{3} + 1 \end{pmatrix} \approx \begin{pmatrix} -1,44 \\ -0,14 \\ 1,30 \end{pmatrix}$$

Secondary task in the operational space

1 - Search for the joint velocity $\dot{q}_1^*(t)$ respecting the 1^{st} task :

- Coming back to the expression of matrix ${}^0J$ for planar RRR robot :

$$\begin{pmatrix} {}^0\dot{x}_d^E \\ {}^0\dot{y}_d^E \end{pmatrix} = {}^0J(q) \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{pmatrix}$$

where

$${}^0J(q) = \begin{bmatrix} -l_1S1 - l_2S12 - l_3S123 & -l_2S12 - l_3S123 & -l_3S123 \\ l_1C1 + l_2C12 + l_3C123 & l_2C12 + l_3C123 & l_3C123 \end{bmatrix}$$

Numerical evaluation at the configuration $q = \left(\frac{\pi}{3}, \frac{2\pi}{3}, \frac{\pi}{3}\right)^t$:

$${}^0J(q) = \frac{1}{2} \begin{bmatrix} 0 & \sqrt{3} & \sqrt{3} \\ -2 & -3 & -1 \end{bmatrix}$$

- After computing the *Moore-Penrose* pseudo-inverse :

$$\dot{q}_1^* = {}^0J^{\sharp}\,{}^0\dot{X}_d^E = \frac{1}{3} \begin{bmatrix} -\frac{4\sqrt{3}}{3} & -2 \\ \frac{\sqrt{3}}{3} & -1 \\ \frac{5\sqrt{3}}{3} & 1 \end{bmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{3} \begin{pmatrix} -\frac{4\sqrt{3}}{3} - 2 \\ \frac{\sqrt{3}}{3} - 1 \\ \frac{5\sqrt{3}}{3} + 1 \end{pmatrix} \approx \begin{pmatrix} -1,44 \\ -0,14 \\ 1,30 \end{pmatrix}$$

- Checking the velocity error to be null in cartesian space :

$$\begin{pmatrix} {}^0\dot{x}^E \\ {}^0\dot{y}^E \end{pmatrix} = {}^0J\dot{q}_1^* = \begin{pmatrix} 1 \\ 1 \end{pmatrix} =^0 \dot{X}_d^E(t) \Leftrightarrow \left\| {}^0\dot{X}_d^E - {}^0J\dot{q}_1^* \right\| = 0$$

Secondary task in the operational space

2 - Searching for the joint velocity $\dot{q}_2^*(t)$ respecting the 1^{st} and the 2^{nd} tasks :

Secondary task in the operational space

2 - Searching for the joint velocity $\dot{q}_2^*(t)$ respecting the 1^{st} and the 2^{nd} tasks :

- General solution :

$$\dot{q}_2^* = J_1^\sharp \dot{X}_{d_1} + \left(J_2 N_{J_1}\right)^\sharp \left(\dot{X}_{d_2} - J_2 J_1^\sharp \dot{X}_{d_1}\right)$$

Secondary task in the operational space

2 - Searching for the joint velocity $\dot{q}_2^*(t)$ respecting the 1^{st} and the 2^{nd} tasks :

- General solution :

$$\dot{q}_2^* = J_1^\sharp \dot{X}_{d_1} + (J_2 N_{J_1})^\sharp \left(\dot{X}_{d_2} - J_2 J_1^\sharp \dot{X}_{d_1} \right)$$

- Additional task :

$$J_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad \dot{X}_{d_2} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Secondary task in the operational space

2 - Searching for the joint velocity $\dot{q}_2^*(t)$ respecting the 1^{st} and the 2^{nd} tasks :

- General solution :

$$\dot{q}_2^* = J_1^\sharp \dot{X}_{d_1} + (J_2 N_{J_1})^\sharp \left(\dot{X}_{d_2} - J_2 J_1^\sharp \dot{X}_{d_1}\right)$$

- Additional task :

$$J_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad \dot{X}_{d_2} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

- Matrix for the projector in the null space of J_1 :

$$N_1 = I_3 - J_1^\sharp J_1 = \frac{1}{3} \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{bmatrix}$$

Secondary task in the operational space

2 - Searching for the joint velocity $\dot{q}_2^*(t)$ respecting the 1^{st} and the 2^{nd} tasks :

- General solution :

$$\dot{q}_2^* = J_1^\sharp \dot{X}_{d_1} + (J_2 N_{J_1})^\sharp \left(\dot{X}_{d_2} - J_2 J_1^\sharp \dot{X}_{d_1} \right)$$

- Additional task :

$$J_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad \dot{X}_{d_2} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

- Matrix for the projector in the null space of J_1 :

$$N_1 = I_3 - J_1^\sharp J_1 = \frac{1}{3} \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{bmatrix}$$

- From the computation of N_1, whose rank is equal to 1, we find the joint velocities :

$$\dot{q}_2^* = \frac{1}{2} \begin{pmatrix} -\frac{\sqrt{3}}{3} - 1 \\ -\frac{\sqrt{3}}{3} - 1 \\ \frac{5\sqrt{3}}{3} + 1 \end{pmatrix} \approx \begin{pmatrix} -0,79 \\ -0,79 \\ 1,94 \end{pmatrix}$$

Secondary task in the operational space

2 - Searching for the joint velocity $\dot{q}_2^*(t)$ respecting the 1^{st} and the 2^{nd} tasks :

- General solution :

$$\dot{q}_2^* = J_1^\sharp \dot{X}_{d_1} + (J_2 N_{J_1})^\sharp \left(\dot{X}_{d_2} - J_2 J_1^\sharp \dot{X}_{d_1}\right)$$

- Additional task :

$$J_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad \dot{X}_{d_2} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

- Matrix for the projector in the null space of J_1 :

$$N_1 = I_3 - J_1^\sharp J_1 = \frac{1}{3} \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{bmatrix}$$

- From the computation of N_1, whose rank is equal to 1, we find the joint velocities :

$$\dot{q}_2^* = \frac{1}{2} \begin{pmatrix} -\frac{\sqrt{3}}{3} - 1 \\ -\frac{\sqrt{3}}{3} - 1 \\ \frac{5\sqrt{3}}{3} + 1 \end{pmatrix} \approx \begin{pmatrix} -0,79 \\ -0,79 \\ 1,94 \end{pmatrix}$$

- Analysis of the velocities errors in the cartesian space :

$$\left\| {}^0\dot{X}_d^E - {}^0J\dot{q}_2^* \right\| = 0,$$

$$\left\| \dot{X}_{d_2} - J_2\dot{q}_2^* \right\| \approx 1,12$$

Secondary task in the operational space

- Analysis of the velocities errors in the cartesian space :
  - Respect of the imposed constraint for task 2 (cartesian trajectory respected) and minimisation of the norm of joint velocity, since

$$\left\|\dot{X}_{d_2} - J_2\dot{q}_2^*\right\| < \left\|\dot{X}_{d_2} - J_2\dot{q}_1^*\right\| \approx 1,44$$

  - Indeed, the null space of J_1 of dimension 1 only, is exploited optimally to tend to the achievement of both objectives (without completely fulfilling them), while keeping the trajectory tracking performances of the end-effector.

Course

Inversion using the damped least-squares method

Expression of J_{DLS}

- The minimisation of $H = \frac{1}{2}\left\| J(q)\,\dot{q} - \dot{X}_d \right\|^2 + \frac{\lambda^2}{2}\|\dot{q}\|^2$ is equivalent to the minimisation of

$$H = \frac{1}{2}\left\| \begin{bmatrix} J \\ \lambda I \end{bmatrix} \dot{q} - \begin{pmatrix} \dot{X}_d \\ 0 \end{pmatrix} \right\|^2$$

- Computation of the first derivative of H

$$\frac{\partial H}{\partial q} = \frac{\partial}{\partial q}\left\{ \frac{1}{2}\dot{q}^t \underbrace{\begin{bmatrix} J \\ \lambda I \end{bmatrix}^t \begin{bmatrix} J \\ \lambda I \end{bmatrix}}_{\left[J^t J + \lambda^2 I\right]} \dot{q} + \frac{1}{2}\begin{pmatrix} \dot{X}_d \\ 0 \end{pmatrix}^t \begin{pmatrix} \dot{X}_d \\ 0 \end{pmatrix} \dots \right.$$
$$\left. -\frac{1}{2}\dot{q}^t \begin{bmatrix} J \\ \lambda I \end{bmatrix}^t \begin{pmatrix} \dot{X}_d \\ 0 \end{pmatrix} - \frac{1}{2}\begin{pmatrix} \dot{X}_d \\ 0 \end{pmatrix}^t \begin{bmatrix} J \\ \lambda I \end{bmatrix} \dot{q} \right\}$$

Inversion using the damped least-squares method

$$\begin{aligned}\frac{\partial H}{\partial q} =&\frac{1}{2}\dot{q}^t\left(\left[J^tJ+\lambda^2 I\right]^t+\left[J^tJ+\lambda^2 I\right]\right)\dots\\ &-\frac{1}{2}\begin{pmatrix}\dot{X}_d\\0\end{pmatrix}^t\begin{bmatrix}J\\ \lambda I\end{bmatrix}-\frac{1}{2}\begin{pmatrix}\dot{X}_d\\0\end{pmatrix}^t\begin{bmatrix}J\\ \lambda I\end{bmatrix}\\ =&\dot{q}^t\left[J^tJ+\lambda^2 I\right]-\dot{X}_d^tJ\end{aligned}$$

Thus, the partial derivative cancels when

$$\dot{q}=\left(J^tJ+\lambda^2 I\right)^{-1}J^t\dot{X}_d$$

and we easily show that :

$$\left(\lambda^2 I+J^tJ\right)^{-1}J^t=J^t\left(\lambda^2 I+JJ^t\right)^{-1}$$

Formula for deriative of linear functions

$$\frac{\partial a^t x}{\partial x}=\frac{\partial x^t a}{\partial x}=a^t$$

$$\frac{\partial Ax}{\partial x}=\frac{\partial x^t A}{\partial x^t}=A$$

Derivative of quadratic functions

$$\frac{\partial x^tAx}{\partial x}=x^t\left(A^t+A\right)$$

We could easily check that the matrix $\left(J^tJ+\lambda^2 I\right)$ is inversible, from its SVD decomposition :

$$\sigma_i^{J_{DLS}}=\frac{1}{\sigma_i^2+\lambda^2}$$

Course

Konig theorem

ÉKinetic energy of the body evaluated by intergation of the kinetic energy ol all the material points defining the volume :

$$E_{c_i} = \frac{1}{2}\int_{\mathscr{V}_i} \rho \left\| \dot{p}_{G_i} + \dot{R}r \right\|^2 d\mathscr{V}_i$$
$$= \underbrace{\frac{1}{2}\int_{\mathscr{V}_i} \rho \left\| \dot{p}_{G_i} \right\|^2 d\mathscr{V}_i}_{E^1_{c_i}} + \underbrace{\int_{\mathscr{V}_i} \rho \dot{p}^t_{G_i} \dot{R} r d\mathscr{V}_i}_{E^2_{c_i}} + \underbrace{\frac{1}{2}\int_{\mathscr{V}_i} \rho \left\| \dot{R}r \right\|^2 d\mathscr{V}_i}_{E^3_{c_i}}$$

Konig theorem

ÉKinetic energy of the body evaluated by intergation of the kinetic energy ol all the material points defining the volume :

$$E_{c_i} = \frac{1}{2}\int_{\mathscr{V}_i} \rho \left\| \dot{p}_{G_i} + \dot{R}r \right\|^2 d\mathscr{V}_i$$
$$= \underbrace{\frac{1}{2}\int_{\mathscr{V}_i} \rho \left\| \dot{p}_{G_i} \right\|^2 d\mathscr{V}_i}_{E^1_{c_i}} + \underbrace{\int_{\mathscr{V}_i} \rho \dot{p}^t_{G_i} \dot{R} r d\mathscr{V}_i}_{E^2_{c_i}} + \underbrace{\frac{1}{2}\int_{\mathscr{V}_i} \rho \left\| \dot{R}r \right\|^2 d\mathscr{V}_i}_{E^3_{c_i}}$$

- The term $E^1_{c_i}$ is the *translation kinetic energy* and is equal to :

$$\frac{1}{2}\int_{\mathscr{V}_i} \left\| \dot{p}_{G_i} \right\|^2 dm_i = \frac{1}{2} m_i \dot{p}^t_{G_i} \dot{p}_{G_i} = \frac{1}{2} m_i {}^0V^t_{0,G_i} {}^0V_{0,G_i}$$

Konig theorem

ÉKinetic energy of the body evaluated by intergation of the kinetic energy ol all the material points defining the volume :

$$E_{c_i} = \frac{1}{2}\int_{\mathscr{V}_i} \rho \left\| \dot{p}_{G_i} + \dot{R}r \right\|^2 d\mathscr{V}_i$$
$$= \underbrace{\frac{1}{2}\int_{\mathscr{V}_i} \rho \left\| \dot{p}_{G_i} \right\|^2 d\mathscr{V}_i}_{E_{c_i}^1} + \underbrace{\int_{\mathscr{V}_i} \rho \dot{p}_{G_i}^t \dot{R} r d\mathscr{V}_i}_{E_{c_i}^2} + \underbrace{\frac{1}{2}\int_{\mathscr{V}_i} \rho \left\| \dot{R}r \right\|^2 d\mathscr{V}_i}_{E_{c_i}^3}$$

- The term $E_{c_i}^1$ is the *translation kinetic energy* and is equal to :

$$\frac{1}{2}\int_{\mathscr{V}_i} \left\| \dot{p}_{G_i} \right\|^2 dm_i = \frac{1}{2} m_i \dot{p}_{G_i}^t \dot{p}_{G_i} = \frac{1}{2} m_i {}^0V_{0,G_i}^t \, {}^0V_{0,G_i}$$

- the *mutual* term $E_{c_i}^2$

$$\int_{\mathscr{V}_i} \rho \dot{p}_{G_i}^t \dot{R} r d\mathscr{V}_i = \left(\dot{p}_{G_i}^t \dot{R} \right) \int_{\mathscr{V}_i} \rho r d\mathscr{V}_i$$

is null, since $\int_{\mathscr{V}_i} \rho r d\mathscr{V}_i = \int_{\mathscr{V}_i} \rho p d\mathscr{V}_i - \int_{\mathscr{V}_i} \rho p_{G_i} d\mathscr{V}_i = m_i p_{G_i} - m_i p_{G_i} = 0$ (from the definition of the center of mass).

Konig theorem

- The term $E_{c_i}^3$ is the *rotational kinetic energy* and is equal to :

$$\begin{aligned}
\frac{1}{2}\int_{\mathscr{V}_i}\rho\left\|\dot{R}r\right\|^2 d\mathscr{V}_i &= \frac{1}{2}\int_{\mathscr{V}_i}\rho\left(\dot{R}r\right)^t\left(\dot{R}r\right)d\mathscr{V}_i \\
&= \frac{1}{2}\int_{\mathscr{V}_i}\rho\left(R\widehat{\omega}r\right)^t\left(R\widehat{\omega}r\right)d\mathscr{V}_i \\
&= \frac{1}{2}\int_{\mathscr{V}_i}\rho\left(\widehat{r}\omega\right)^t\left(R^tR\right)\left(\widehat{r}\omega\right)d\mathscr{V}_i \\
&= \frac{1}{2}\omega^t\left(\int_{\mathscr{V}_i}\rho\widehat{r}^t\widehat{r}d\mathscr{V}_i\right)\omega \\
&:= \frac{1}{2}\omega^t I_i\omega
\end{aligned}$$

where $\omega \in \mathbb{R}^3$ is the angular velocity of $\mathscr{C}_i$.

Konig theorem

- The term $E^3_{c_i}$ is the *rotational kinetic energy* and is equal to :

$$\begin{aligned}
\frac{1}{2}\int_{\mathscr{V}_i} \rho \left\|\dot{R}r\right\|^2 d\mathscr{V}_i &= \frac{1}{2}\int_{\mathscr{V}_i} \rho \left(\dot{R}r\right)^t \left(\dot{R}r\right) d\mathscr{V}_i \\
&= \frac{1}{2}\int_{\mathscr{V}_i} \rho \left(R\widehat{\omega}r\right)^t \left(R\widehat{\omega}r\right) d\mathscr{V}_i \\
&= \frac{1}{2}\int_{\mathscr{V}_i} \rho \left(\widehat{r}\omega\right)^t \left(R^t R\right) \left(\widehat{r}\omega\right) d\mathscr{V}_i \\
&= \frac{1}{2}\omega^t \left(\int_{\mathscr{V}_i} \rho \widehat{r}^t \widehat{r} d\mathscr{V}_i\right) \omega \\
&:= \frac{1}{2}\omega^t I_i \omega
\end{aligned}$$

where $\omega \in \mathbb{R}^3$ is the angular velocity of $\mathscr{C}_i$.
Note that, according to the properties seen in the chapter "transformation", the change of $\dot{R}$ by the quantity $R\widehat{\omega}$ above implies that the angular velocity vector of the body ω is expressed in the reference frame related to the body (and not in the inertial reference frame).

inertia tensor

Rectangular bar of homogeneous material (density $\rho = \frac{m_i}{abc}$) :

inertia tensor

Rectangular bar of homogeneous material (density $\rho = \frac{m_i}{abc}$) :

- $I_{xx} = \rho \int_{\mathscr{V}_i} \left(y^2 + z^2\right) dxdydz = \frac{m_i}{abc} \int_{-c/2}^{c/2} \int_{-b/2}^{b/2} \int_{-a/2}^{a/2} \left(y^2 + z^2\right) dxdydz = \frac{m_i}{12} \left(b^2 + c^2\right)$;

inertia tensor

Rectangular bar of homogeneous material (density $\rho = \frac{m_i}{abc}$) :

- $I_{xx} = \rho \int_{\mathscr{V}_i} \left(y^2 + z^2\right) dxdydz = \frac{m_i}{abc} \int_{-c/2}^{c/2} \int_{-b/2}^{b/2} \int_{-a/2}^{a/2} \left(y^2 + z^2\right) dxdydz = \frac{m_i}{12} \left(b^2 + c^2\right)$;
- $I_{xy} = -\rho \int_{\mathscr{V}_i} (xy)\, dxdydz = -\frac{m_i}{abc} \int_{-c/2}^{c/2} \int_{-b/2}^{b/2} \int_{-a/2}^{a/2} (xy)\, dxdydz = 0$;

inertia tensor

Rectangular bar of homogeneous material (density $\rho = \frac{m_i}{abc}$) :

- $I_{xx} = \rho \int_{\mathscr{V}_i} \left(y^2 + z^2\right) dxdydz = \frac{m_i}{abc} \int_{-c/2}^{c/2} \int_{-b/2}^{b/2} \int_{-a/2}^{a/2} \left(y^2 + z^2\right) dxdydz = \frac{m_i}{12} \left(b^2 + c^2\right)$;
- $I_{xy} = -\rho \int_{\mathscr{V}_i} (xy)\, dxdydz = -\frac{m_i}{abc} \int_{-c/2}^{c/2} \int_{-b/2}^{b/2} \int_{-a/2}^{a/2} (xy)\, dxdydz = 0$;
- The other components are calculated in the same way and we obtain :

$$I_i = \begin{bmatrix} \frac{m_i}{12}\left(b^2 + c^2\right) & 0 & 0 \\ 0 & \frac{m_i}{12}\left(a^2 + c^2\right) & 0 \\ 0 & 0 & \frac{m_i}{12}\left(a^2 + b^2\right) \end{bmatrix}$$

inertia tensor

Rectangular bar of homogeneous material (density $\rho = \frac{m_i}{abc}$) :

- $I_{xx} = \rho \int_{\mathscr{V}_i} \left(y^2 + z^2\right) dxdydz = \frac{m_i}{abc} \int_{-c/2}^{c/2} \int_{-b/2}^{b/2} \int_{-a/2}^{a/2} \left(y^2 + z^2\right) dxdydz = \frac{m_i}{12} \left(b^2 + c^2\right)$;
- $I_{xy} = -\rho \int_{\mathscr{V}_i} (xy)\, dxdydz = -\frac{m_i}{abc} \int_{-c/2}^{c/2} \int_{-b/2}^{b/2} \int_{-a/2}^{a/2} (xy)\, dxdydz = 0$;
- The other components are calculated in the same way and we obtain :

$$I_i = \begin{bmatrix} \frac{m_i}{12}\left(b^2 + c^2\right) & 0 & 0 \\ 0 & \frac{m_i}{12}\left(a^2 + c^2\right) & 0 \\ 0 & 0 & \frac{m_i}{12}\left(a^2 + b^2\right) \end{bmatrix}$$

Note that the products of inertia are zero when the main axes of the body coincide with the central axes of inertia. Thus, the matrix corresponding to the inertia tensor is diagonal in this case.

▸ Course

Study of one revolute robotic axis

- Hypothesis :
  1. Center of mass of the rotor located on its rotation axis ;
  2. Neglected rotor gyroscopic effects (only the moment of inertia of the proper rotation is considered) ;

Study of one revolute robotic axis

- Hypothesis :
  1. Center of mass of the rotor located on its rotation axis ;
  2. Neglected rotor gyroscopic effects (only the moment of inertia of the proper rotation is considered) ;
- Kinetic energy of the driven body $\mathscr{C}_i$ and the driving actuator :

$$\begin{aligned} E_c &= E_{c_i} + E_{c_{m_i}} \\ &= \left(\frac{1}{2} m_i V_{G_i}^2 + \frac{1}{2} I_{G_i} \dot{q}_i^2\right) + \left(\frac{1}{2} r_{red_i}^2 J_{m_i} \dot{\theta}_i^2\right) \\ &= \left(\frac{1}{2} m_i V_{G_i}^2 + \frac{1}{2} I_{G_i} \dot{q}_i^2\right) + \left(\frac{1}{2} r_{red_i}^2 J_{m_i} \dot{q}_i^2\right) \end{aligned}$$

where the velocity of the center of mass V_{G_i} equals to $V_{G_i} = d_i \dot{q}$.

Study of one revolute robotic axis

- Hypothesis :
  1. Center of mass of the rotor located on its rotation axis ;
  2. Neglected rotor gyroscopic effects (only the moment of inertia of the proper rotation is considered) ;
- Kinetic energy of the driven body $\mathscr{C}_i$ and the driving actuator :

$$\begin{aligned} E_c &= E_{c_i} + E_{c_{m_i}} \\ &= \left(\frac{1}{2} m_i V_{G_i}^2 + \frac{1}{2} I_{G_i} \dot{q}_i^2\right) + \left(\frac{1}{2} r_{red_i}^2 J_{m_i} \dot{\theta}_i^2\right) \\ &= \left(\frac{1}{2} m_i V_{G_i}^2 + \frac{1}{2} I_{G_i} \dot{q}_i^2\right) + \left(\frac{1}{2} r_{red_i}^2 J_{m_i} \dot{q}_i^2\right) \end{aligned}$$

where the velocity of the center of mass V_{G_i} equals to $V_{G_i} = d_i \dot{q}$.
Thus,

$$E_c = \frac{1}{2} \underbrace{\left(m_i d_i^2 + I_{G_i} + r_{red_i}^2 J_{m_i}\right)}_{I_{eq_i}} \dot{q}_i^2 .$$

Study of planar 2-axes robot

Kinetic energy of the robot :

$$E_c = \frac{1}{2}\dot{q}^t \underbrace{\sum_{i=1}^{2}\left(m_i\,{}^0J_{v_{G_i}}^t(q)\,{}^0J_{v_{G_i}}(q) + {}^0J_{\omega_i}^t(q)\,{}^0I_i\,{}^0J_{\omega_i}(q)\right)}_{A(q)}\dot{q}$$

Study of planar 2-axes robot

Kinetic energy of the robot :

$$E_c = \frac{1}{2}\dot{q}^t \underbrace{\sum_{i=1}^{2}\left(m_i {}^0J^t_{v_{G_i}}(q)\, {}^0J_{v_{G_i}}(q) + {}^0J^t_{\omega_i}(q)\, {}^0I_i {}^0J_{\omega_i}(q)\right)}_{A(q)} \dot{q}$$

- Kinetic energy of the body $\mathscr{C}_1$:
  - Translational kinetic energy $E_{1_{Tran.}} = \frac{1}{2} m_1 \dot{q}^{t\,0} J^t_{v_{G_1}}(q)\, {}^0J_{v_{G_1}}(q)\, \dot{q}$

Study of planar 2-axes robot

Kinetic energy of the robot :

$$E_c = \frac{1}{2}\dot{q}^t \underbrace{\sum_{i=1}^{2}\left(m_i {}^0J^t_{v_{G_i}}(q)\,{}^0J_{v_{G_i}}(q) + {}^0J^t_{\omega_i}(q)\,{}^0I_i\,{}^0J_{\omega_i}(q)\right)}_{A(q)}\dot{q}$$

- Kinetic energy of the body $\mathscr{C}_1$:
  - Translational kinetic energy $E_{1_{Tran.}} = \frac{1}{2} m_1 \dot{q}^{t\,0} J^t_{v_{G_1}}(q)\,{}^0J_{v_{G_1}}(q)\,\dot{q}$
    - Jacobian matrix giving the velocity of center of mass G_1 :

$${}^0V_{G_1} = {}^0J_{v_{G_1}}\dot{q}$$

where the Jacobian matrix expressing the velocity of translation of the center of mass V_{G_1} in the $\mathscr{R}_0$ frame is easily deduced from the example n$^\circ$12 according to :

$${}^0J_{v_{G_1}} = \begin{bmatrix} -L_{c_1}S1 & 0 \\ L_{c_1}C1 & 0 \\ 0 & 0 \end{bmatrix}$$

Study of planar 2-axes robot

Kinetic energy of the robot :

$$E_c = \frac{1}{2}\dot{q}^t \underbrace{\sum_{i=1}^{2}\left(m_i {}^0J^t_{v_{G_i}}(q)\, {}^0J_{v_{G_i}}(q) + {}^0J^t_{\omega_i}(q)\, {}^0I_i\, {}^0J_{\omega_i}(q)\right)}_{A(q)}\dot{q}$$

- Kinetic energy of the body $\mathscr{C}_1$:
  - Translational kinetic energy $E_{1_{Tran.}} = \frac{1}{2} m_1 \dot{q}^{t\,0} J^t_{v_{G_1}}(q)\, {}^0J_{v_{G_1}}(q)\, \dot{q}$
    - Jacobian matrix giving the velocity of center of mass G_1 :

$${}^0V_{G_1} = {}^0J_{v_{G_1}}\dot{q}$$

where the Jacobian matrix expressing the velocity of translation of the center of mass V_{G_1} in the $\mathscr{R}_0$ frame is easily deduced from the example n$^\circ$12 according to :

$${}^0J_{v_{G_1}} = \begin{bmatrix} -L_{c_1}S1 & 0 \\ L_{c_1}C1 & 0 \\ 0 & 0 \end{bmatrix}$$

Another method (more systematic) :
The vector ${}^0V_{G_1}$ can also be deduced from ${}^0V_{0_1}$ (velocity of the origin of the frame given by the Jacobian matrix ${}^0J_1$) from the *Varignon* formula.

Study of planar 2-axes robot

Kinetic energy of the robot :

$$E_c = \frac{1}{2}\dot{q}^t \underbrace{\sum_{i=1}^{2}\left(m_i {}^0J^t_{v_{G_i}}(q)\,{}^0J_{v_{G_i}}(q) + {}^0J^t_{\omega_i}(q)\,{}^0I_i\,{}^0J_{\omega_i}(q)\right)}_{A(q)}\dot{q}$$

- Kinetic energy of the body $\mathscr{C}_1$:
  - Translational kinetic energy $E_{1_{Tran.}} = \frac{1}{2} m_1 \dot{q}^{t\,0} J^t_{v_{G_1}}(q)\,{}^0J_{v_{G_1}}(q)\,\dot{q}$
    - Jacobian matrix giving the velocity of center of mass G_1 :

      $${}^0V_{G_1} = {}^0J_{v_{G_1}}\dot{q}$$

      where the Jacobian matrix expressing the velocity of translation of the center of mass V_{G_1} in the $\mathscr{R}_0$ frame is easily deduced from the example n$^\circ$12 according to :

      $${}^0J_{v_{G_1}} = \begin{bmatrix} -L_{c_1}S1 & 0 \\ L_{c_1}C1 & 0 \\ 0 & 0 \end{bmatrix}$$

      Another method (more systematic) :
      The vector ${}^0V_{G_1}$ can also be deduced from ${}^0V_{0_1}$ (velocity of the origin of the frame given by the Jacobian matrix ${}^0J_1$) from the *Varignon* formula.
    - $E_{1_{Tran.}} = \frac{1}{2}\dot{q}^t \begin{bmatrix} m_1 L^2_{c_1} & 0 \\ 0 & 0 \end{bmatrix}\dot{q}$

Study of planar 2-axes robot

Kinetic energy of the robot :

- Rotational kinetic energy $E_{1_{Rot.}} = \frac{1}{2}\dot{q}\,{}^{0}J^{t}_{\omega_1}(q)\,{}^{0}I_1\,{}^{0}J_{\omega_1}(q)\,\dot{q}^{t}$

Study of planar 2-axes robot

Kinetic energy of the robot :

- Rotational kinetic energy $E_{1_{Rot.}} = \frac{1}{2}\dot{q}\,{}^{0}J^{t}_{\omega_1}(q)\,{}^{0}I_1\,{}^{0}J_{\omega_1}(q)\,\dot{q}^{t}$
  - Jacobian matrix giving the instataneous rotation velocity of $\mathscr{C}_1$:

$$ {}^{0}\omega_1 = {}^{0}J_{\omega_1}\,\dot{q} $$

where the Jacobian matrix expressing the angular velocity of $\mathscr{C}_1$ in the frame $\mathscr{R}_0$ is given in the example n$^\circ$12 according to : :

$$ {}^{0}J_{\omega_1} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} $$

Study of planar 2-axes robot

Kinetic energy of the robot :

- Rotational kinetic energy $E_{1_{Rot.}} = \frac{1}{2} \dot{q}^0 J^t_{\omega_1}(q)\, {}^0I_1 {}^0J_{\omega_1}(q)\, \dot{q}^t$
  - Jacobian matrix giving the instataneous rotation velocity of $\mathscr{C}_1$:

$$^0\omega_1 = {}^0J_{\omega_1}\dot{q}$$

where the Jacobian matrix expressing the angular velocity of $\mathscr{C}_1$ in the frame $\mathscr{R}_0$ is given in the example n$^\circ$12 according to : :

$$^0J_{\omega_1} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}$$

  - Besides, the inertia tensor of $\mathscr{C}_1$ given in the frame $\mathscr{R}_0$ equals

$$^0I_1 = R_{01} I_1 R^t_{01}$$

where R_{10} is deduced from the homogeneous transformation $\overline{g}_{10}$ already calculated in the previous examples.

Study of planar 2-axes robot

Kinetic energy of the robot :

- Rotational kinetic energy $E_{1_{Rot.}} = \frac{1}{2}\dot{q}^{0}J_{\omega_1}^{t}(q)\,^{0}I_1\,^{0}J_{\omega_1}(q)\,\dot{q}^{t}$
  - Jacobian matrix giving the instataneous rotation velocity of $\mathscr{C}_1$:

    $$^{0}\omega_1 = {^{0}J_{\omega_1}}\,\dot{q}$$

    where the Jacobian matrix expressing the angular velocity of $\mathscr{C}_1$ in the frame $\mathscr{R}_0$ is given in the example n$^\circ$12 according to : :

    $$^{0}J_{\omega_1} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}$$

  - Besides, the inertia tensor of $\mathscr{C}_1$ given in the frame $\mathscr{R}_0$ equals

    $$^{0}I_1 = R_{01}\,I_1\,R_{01}^{t}$$

    where R_{10} is deduced from the homogeneous transformation $\overline{g}_{10}$ already calculated in the previous examples.
  - $E_{1_{Rot.}} = \frac{1}{2}\dot{q}^{t}\begin{bmatrix} I_{zz1} & 0 \\ 0 & 0 \end{bmatrix}\dot{q}$ with I_{zz_1} the inertia moment of $\mathscr{C}_1$ in its center of mass w.r.t. the axis z_0.

Study of planar 2-axes robot

Kinetic energy of the robot :

- Rotational kinetic energy $E_{1_{Rot.}} = \frac{1}{2}\dot{q}\,{}^0J^t_{\omega_1}(q)\,{}^0I_1\,{}^0J_{\omega_1}(q)\,\dot{q}^t$
  - Jacobian matrix giving the instataneous rotation velocity of $\mathscr{C}_1$:

    $${}^0\omega_1 = {}^0J_{\omega_1}\dot{q}$$

    where the Jacobian matrix expressing the angular velocity of $\mathscr{C}_1$ in the frame $\mathscr{R}_0$ is given in the example n$^\circ$12 according to : :

    $${}^0J_{\omega_1} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}$$

  - Besides, the inertia tensor of $\mathscr{C}_1$ given in the frame $\mathscr{R}_0$ equals

    $${}^0I_1 = R_{01}I_1R^t_{01}$$

    where R_{10} is deduced from the homogeneous transformation $\overline{g}_{10}$ already calculated in the previous examples.
  - $E_{1_{Rot.}} = \frac{1}{2}\dot{q}^t\begin{bmatrix} I_{zz1} & 0 \\ 0 & 0 \end{bmatrix}\dot{q}$ with I_{zz_1} the inertia moment of $\mathscr{C}_1$ in its center of mass w.r.t. the axis z_0.
    In this example, the angular velocity vector ω_1 equals to $\dot{q}_1 z_0 = \dot{q}_1 z_1$ and thus the inertia moment I_{zz_1} remains unchanged through the rotation R_{10}.

Study of planar 2-axes robot

Kinetic energy of the robot :

- Rotational kinetic energy $E_{1_{Rot.}} = \frac{1}{2}\dot{q}^0 J^t_{\omega_1}(q)\,{}^0I_1{}^0J_{\omega_1}(q)\,\dot{q}^t$
  - Jacobian matrix giving the instataneous rotation velocity of $\mathscr{C}_1$:

$$^0\omega_1 = {}^0J_{\omega_1}\dot{q}$$

    where the Jacobian matrix expressing the angular velocity of $\mathscr{C}_1$ in the frame $\mathscr{R}_0$ is given in the example n$^\circ$12 according to : :

$$^0J_{\omega_1} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}$$

  - Besides, the inertia tensor of $\mathscr{C}_1$ given in the frame $\mathscr{R}_0$ equals

$$^0I_1 = R_{01}I_1R^t_{01}$$

    where R_{10} is deduced from the homogeneous transformation $\overline{g}_{10}$ already calculated in the previous examples.
  - $E_{1_{Rot.}} = \frac{1}{2}\dot{q}^t \begin{bmatrix} I_{zz1} & 0 \\ 0 & 0 \end{bmatrix} \dot{q}$ with I_{zz_1} the inertia moment of $\mathscr{C}_1$ in its center of mass w.r.t. the axis z_0.
    In this example, the angular velocity vector ω_1 equals to $\dot{q}_1 z_0 = \dot{q}_1 z_1$ and thus the inertia moment I_{zz_1} remains unchanged through the rotation R_{10}.
- Finally the kinetic energy of $\mathscr{C}_1$ equals :

$$E_{c_1} = \frac{1}{2}\dot{q}^t \begin{bmatrix} m_1L^2_{c_1} + I_{zz1} & 0 \\ 0 & 0 \end{bmatrix} \dot{q}$$

Study of planar 2-axes robot

- Kinetic energy of $\mathscr{C}_2$:
  - Translational kinetic energy $E_{2_{Tran.}} = \frac{1}{2} m_2 \dot{q}\, {}^0J^t_{v_{G_2}}(q)\, {}^0J_{v_{G_2}}(q)\, \dot{q}^t$

Study of planar 2-axes robot

- Kinetic energy of $\mathscr{C}_2$:
  - Translational kinetic energy $E_{2_{Tran.}} = \frac{1}{2} m_2 \dot{q}^0 J^t_{v_{G_2}}(q)\, {}^0J_{v_{G_2}}(q)\, \dot{q}^t$
    - Jacobian matrix giving the velocity of the center of mass G_2 :

$$ {}^0V_{G_2} = {}^0J_{v_{G_2}} \dot{q} $$

where the Jacobian matrix expressing the translational velocity of center of mass V_{G_2} in the frame $\mathscr{R}_0$ equals :

$$ {}^0J_{v_{G_2}} = \begin{bmatrix} -L_1 S1 - L_{c_2} S12 & -L_{c_2} S12 \\ L_1 C1 + L_{c_2} C12 & L_{c_2} C12 \\ 0 & 0 \end{bmatrix} $$

Study of planar 2-axes robot

- Kinetic energy of $\mathscr{C}_2$:
  - Translational kinetic energy $E_{2_{Tran.}} = \frac{1}{2} m_2 \dot{q}\, {}^0J^t_{v_{G_2}}(q)\, {}^0J_{v_{G_2}}(q)\, \dot{q}^t$
    - Jacobian matrix giving the velocity of the center of mass G_2 :

$$ {}^0V_{G_2} = {}^0J_{v_{G_2}}\, \dot{q} $$

where the Jacobian matrix expressing the translational velocity of center of mass V_{G_2} in the frame $\mathscr{R}_0$ equals :

$$ {}^0J_{v_{G_2}} = \begin{bmatrix} -L_1 S1 - L_{c_2} S12 & -L_{c_2} S12 \\ L_1 C1 + L_{c_2} C12 & L_{c_2} C12 \\ 0 & 0 \end{bmatrix} $$

  - $E_{2_{Tran.}} = \frac{1}{2} \dot{q}^t \begin{bmatrix} m_2 \left(L_1^2 + L_{c_2}^2 + 2 L_1 L_{c_2} cos\,(q_2) \right) & m_2 \left(L_{c_2}^2 + L_1 L_{c_2} cos\,(q_2) \right) \\ m_2 \left(L_{c_2}^2 + L_1 L_{c_2} cos\,(q_2) \right) & m_2 L_{c_2}^2 \end{bmatrix} \dot{q}$

Study of planar 2-axes robot

- Kinetic energy of $\mathscr{C}_2$:
  - Translational kinetic energy $E_{2_{Tran.}} = \frac{1}{2} m_2 \dot{q}^0 J_{v_{G_2}}^t (q)\, {}^0 J_{v_{G_2}} (q)\, \dot{q}^t$
    - Jacobian matrix giving the velocity of the center of mass G_2 :

$$ {}^0 V_{G_2} = {}^0 J_{v_{G_2}} \dot{q} $$

where the Jacobian matrix expressing the translational velocity of center of mass V_{G_2} in the frame $\mathscr{R}_0$ equals :

$$ {}^0 J_{v_{G_2}} = \begin{bmatrix} -L_1 S1 - L_{c_2} S12 & -L_{c_2} S12 \\ L_1 C1 + L_{c_2} C12 & L_{c_2} C12 \\ 0 & 0 \end{bmatrix} $$

    - $E_{2_{Tran.}} = \frac{1}{2} \dot{q}^t \begin{bmatrix} m_2 \left(L_1^2 + L_{c_2}^2 + 2 L_1 L_{c_2} \cos(q_2) \right) & m_2 \left(L_{c_2}^2 + L_1 L_{c_2} \cos(q_2) \right) \\ m_2 \left(L_{c_2}^2 + L_1 L_{c_2} \cos(q_2) \right) & m_2 L_{c_2}^2 \end{bmatrix} \dot{q}$
  - Rotational kinetic energy $E_{2_{Rot.}} = \frac{1}{2} \dot{q}^0 J_{\omega_2}^t (q)\, {}^0 I_2 {}^0 J_{\omega_2} (q)\, \dot{q}^t$

Study of planar 2-axes robot

- Kinetic energy of $\mathscr{C}_2$:
  - Translational kinetic energy $E_{2_{Tran.}} = \frac{1}{2} m_2 \dot{q}\,{}^0 J^t_{v_{G_2}}(q)\,{}^0 J_{v_{G_2}}(q)\,\dot{q}^t$
    - Jacobian matrix giving the velocity of the center of mass G_2 :

$$ {}^0 V_{G_2} = {}^0 J_{v_{G_2}}\,\dot{q} $$

where the Jacobian matrix expressing the translational velocity of center of mass V_{G_2} in the frame $\mathscr{R}_0$ equals :

$$ {}^0 J_{v_{G_2}} = \begin{bmatrix} -L_1 S1 - L_{c_2} S12 & -L_{c_2} S12 \\ L_1 C1 + L_{c_2} C12 & L_{c_2} C12 \\ 0 & 0 \end{bmatrix} $$

  - $E_{2_{Tran.}} = \frac{1}{2}\dot{q}^t \begin{bmatrix} m_2\left(L_1^2 + L_{c_2}^2 + 2L_1 L_{c_2}\cos(q_2)\right) & m_2\left(L_{c_2}^2 + L_1 L_{c_2}\cos(q_2)\right) \\ m_2\left(L_{c_2}^2 + L_1 L_{c_2}\cos(q_2)\right) & m_2 L_{c_2}^2 \end{bmatrix} \dot{q}$
  - Rotational kinetic energy $E_{2_{Rot.}} = \frac{1}{2}\dot{q}\,{}^0 J^t_{\omega_2}(q)\,{}^0 I_2\,{}^0 J_{\omega_2}(q)\,\dot{q}^t$
    - Jacobian matrix expressing the instantaneous rotational velocity of $\mathscr{C}_2$:

$$ {}^0 J_{\omega_2} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix} $$

Study of planar 2-axes robot

- Kinetic energy of $\mathscr{C}_2$:
  - Translational kinetic energy $E_{2_{Tran.}} = \frac{1}{2} m_2 \dot{q}^0 J^t_{v_{G_2}}(q)\, {}^0 J_{v_{G_2}}(q)\, \dot{q}^t$
    - Jacobian matrix giving the velocity of the center of mass G_2 :

      $$ {}^0 V_{G_2} = {}^0 J_{v_{G_2}} \dot{q} $$

      where the Jacobian matrix expressing the translational velocity of center of mass V_{G_2} in the frame $\mathscr{R}_0$ equals :

      $$ {}^0 J_{v_{G_2}} = \begin{bmatrix} -L_1 S1 - L_{c_2} S12 & -L_{c_2} S12 \\ L_1 C1 + L_{c_2} C12 & L_{c_2} C12 \\ 0 & 0 \end{bmatrix} $$

    - $E_{2_{Tran.}} = \frac{1}{2} \dot{q}^t \begin{bmatrix} m_2 \left(L_1^2 + L_{c_2}^2 + 2 L_1 L_{c_2} cos(q_2) \right) & m_2 \left(L_{c_2}^2 + L_1 L_{c_2} cos(q_2) \right) \\ m_2 \left(L_{c_2}^2 + L_1 L_{c_2} cos(q_2) \right) & m_2 L_{c_2}^2 \end{bmatrix} \dot{q}$
  - Rotational kinetic energy $E_{2_{Rot.}} = \frac{1}{2} \dot{q}^0 J^t_{\omega_2}(q)\, {}^0 I_2 {}^0 J_{\omega_2}(q)\, \dot{q}^t$
    - Jacobian matrix expressing the instantaneous rotational velocity of $\mathscr{C}_2$:

      $$ {}^0 J_{\omega_2} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix} $$

    - $E_{2_{Rot.}} = \frac{1}{2} \dot{q}^t \begin{bmatrix} I_{zz_2} & I_{zz_2} \\ I_{zz_2} & I_{zz_2} \end{bmatrix} \dot{q}$ with I_{zz_2} the inertia moment of $\mathscr{C}_2$ at its center of mass w.r.t. the axis z_0 (here, the vector ω_2 equals $\dot{q}_2 z_0 = \dot{q}_2 z_2$).

Study of planar 2-axes robot

- Kinetic energy of the polyarticulated chain :

$$\frac{1}{2}\dot{q}^t \begin{bmatrix} m_1 L_{c_1}^2 + m_2\left(L_1^2 + L_{c_2}^2 + 2L_1 L_{c_2} cos\left(q_2\right)\right) + I_{zz_1} + I_{zz_2} & m_2\left(L_{c_2}^2 + L_1 L_{c_2} cos\left(q_2\right)\right) + I_{zz_2} \\ m_2\left(L_{c_2}^2 + L_1 L_{c_2} cos\left(q_2\right)\right) + I_{zz_2} & m_2 L_{c_2}^2 + I_{zz_2} \end{bmatrix} \dot{q}$$

Study of planar 2-axes robot

- Kinetic energy of the polyarticulated chain :

$$\frac{1}{2}\dot{q}^t \left[\begin{array}{cc} m_1 L_{c_1}^2 + m_2 \left(L_1^2 + L_{c_2}^2 + 2 L_1 L_{c_2} \cos(q_2) \right) + I_{zz_1} + I_{zz_2} & m_2 \left(L_{c_2}^2 + L_1 L_{c_2} \cos(q_2) \right) + I_{zz_2} \\ m_2 \left(L_{c_2}^2 + L_1 L_{c_2} \cos(q_2) \right) + I_{zz_2} & m_2 L_{c_2}^2 + I_{zz_2} \end{array} \right] \dot{q}$$

- Total kinetic energy of the robot taking into account the influence of the actuators inertia :

$$\frac{1}{2}\dot{q}^t \left[\begin{array}{cc} r_{red_1}^2 J_{m_1} & 0 \\ 0 & r_{red_2}^2 J_{m_2} \end{array} \right] \dot{q}$$

to the kinetic energy of the polyarticuled chain.

Study of planar 2-axes robot

- Kinetic energy of the polyarticulated chain :

$$\frac{1}{2}\dot{q}^t \begin{bmatrix} m_1 L_{c_1}^2 + m_2\left(L_1^2 + L_{c_2}^2 + 2L_1 L_{c_2}\cos(q_2)\right) + I_{zz_1} + I_{zz_2} & m_2\left(L_{c_2}^2 + L_1 L_{c_2}\cos(q_2)\right) + I_{zz_2} \\ m_2\left(L_{c_2}^2 + L_1 L_{c_2}\cos(q_2)\right) + I_{zz_2} & m_2 L_{c_2}^2 + I_{zz_2} \end{bmatrix} \dot{q}$$

- Total kinetic energy of the robot taking into account the influence of the actuators inertia :

$$\frac{1}{2}\dot{q}^t \begin{bmatrix} r_{red_1}^2 J_{m_1} & 0 \\ 0 & r_{red_2}^2 J_{m_2} \end{bmatrix} \dot{q}$$

to the kinetic energy of the polyarticuled chain.
Finally,

$$E_c = \frac{1}{2}\dot{q}^t A(q)\,\dot{q}$$

where the inertia matrix of the robot equals :
$A(q) =$

$$\begin{bmatrix} r_{red_1}^2 J_{m_1} + m_1 L_{c_1}^2 + m_2\left(L_1^2 + L_{c_2}^2 + 2L_1 L_{c_2}\cos(q_2)\right) + I_{zz_1} + I_{zz_2} & m_2\left(L_{c_2}^2 + L_1 L_{c_2}\cos(q_2)\right) + I_{zz_2} \\ m_2\left(L_{c_2}^2 + L_1 L_{c_2}\cos(q_2)\right) + I_{zz_2} & r_{red_2}^2 J_{m_2} + m_2 L_{c_2}^2 + I_{zz_2} \end{bmatrix}$$

Potential energy of a polyarticulated chain

Potential energy of gravity of one revolute robotic axis :

$$E_{p_i}(q) = -m_i \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} {}^0p_{G_i}(q)$$

with

Potential energy of a polyarticulated chain

Potential energy of gravity of one revolute robotic axis :

$$E_{p_i}(q) = -m_i \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} {}^0p_{G_i}(q)$$

with

- ${}^0p_{G_i}(q)$ the position of the center of mass of $\mathscr{C}_i$ w.r.t. $\mathscr{R}_0$:

$$ {}^0p_{G_i}(q_i) = {}^0O_0G_i = \begin{bmatrix} 0 \\ d_i\left(1 - cos\left(q_i\right)\right) \\ 0 \end{bmatrix}_{\mathscr{R}_0}$$

Potential energy of a polyarticulated chain

Potential energy of gravity of one revolute robotic axis :

$$E_{p_i}(q) = -m_i \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} {}^0p_{G_i}(q)$$

with

- ${}^0p_{G_i}(q)$ the position of the center of mass of $\mathscr{C}_i$ w.r.t. $\mathscr{R}_0$:

$$ {}^0p_{G_i}(q_i) = {}^0O_0G_i = \begin{bmatrix} 0 \\ d_i(1 - cos(q_i)) \\ 0 \end{bmatrix}_{\mathscr{R}_0}$$

- Final expression of $E_{p_i}(q)$:

$$E_{p_i}(q) = m_i g d_i (1 - cos(q_i))$$

Course

Potential energy of a polyarticulated chain

Potential energy of gravity of the planar robot :

$$E_p(q) = \sum_{i=1}^{2} E_{p_i}(q) = -\begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \left(\sum_{i=1}^{2} m_i \, {}^0p_{G_i}(q) \right).$$

with

Potential energy of a polyarticulated chain

Potential energy of gravity of the planar robot :

$$E_p(q) = \sum_{i=1}^{2} E_{p_i}(q) = -\begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \left(\sum_{i=1}^{2} m_i \, {}^0p_{G_i}(q) \right).$$

with

- ${}^0p_{G_i}(q)$ the position of the center of mass of $\mathscr{C}_i$ w.r.t. $\mathscr{R}_0$, so that :

Potential energy of a polyarticulated chain

Potential energy of gravity of the planar robot :

$$E_p(q) = \sum_{i=1}^{2} E_{p_i}(q) = -\begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \left(\sum_{i=1}^{2} m_i \, {}^0p_{G_i}(q) \right).$$

with

- ${}^0p_{G_i}(q)$ the position of the center of mass of $\mathscr{C}_i$ w.r.t. $\mathscr{R}_0$, so that :

$$\begin{bmatrix} {}^0p_{G_1}(q) \\ 1 \end{bmatrix} = \begin{bmatrix} {}^0O_0G_1 \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} C1 & -S1 & 0 & 0 \\ S1 & C1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\overline{g}_{01}} \begin{bmatrix} L_{c_1} \\ 0 \\ 0 \\ 1 \end{bmatrix}_{\mathscr{R}_1} = \begin{bmatrix} L_{c_1}C1 \\ L_{c_1}S1 \\ 0 \\ 1 \end{bmatrix}_{\mathscr{R}_0}$$

Potential energy of a polyarticulated chain

Potential energy of gravity of the planar robot :

$$E_p(q) = \sum_{i=1}^{2} E_{p_i}(q) = -\begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \left(\sum_{i=1}^{2} m_i \, {}^0p_{G_i}(q) \right).$$

with

- ${}^0p_{G_i}(q)$ the position of the center of mass of $\mathscr{C}_i$ w.r.t. $\mathscr{R}_0$, so that :

$$\begin{bmatrix} {}^0p_{G_1}(q) \\ 1 \end{bmatrix} = \begin{bmatrix} {}^0O_0G_1 \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} C1 & -S1 & 0 & 0 \\ S1 & C1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\overline{g}_{01}} \begin{bmatrix} L_{c_1} \\ 0 \\ 0 \\ 1 \end{bmatrix}_{\mathscr{R}_1} = \begin{bmatrix} L_{c_1}C1 \\ L_{c_1}S1 \\ 0 \\ 1 \end{bmatrix}_{\mathscr{R}_0}$$

and

$$\begin{bmatrix} {}^0p_{G_2}(q) \\ 1 \end{bmatrix} = \begin{bmatrix} {}^0O_0G_2 \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} C12 & -S12 & 0 & L_1C1 \\ S12 & C12 & 0 & L_1S1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\overline{g}_{02}} \begin{bmatrix} L_{c_2} \\ 0 \\ 0 \\ 1 \end{bmatrix}_{\mathscr{R}_2} = \begin{bmatrix} L_{c_2}C12 + L_1C1 \\ L_{c_2}S12 + L_1S1 \\ 0 \\ 1 \end{bmatrix}_{\mathscr{R}_0}$$

Potential energy of a polyarticulated chain

Potential energy of gravity of the planar robot :

$$E_p(q) = \sum_{i=1}^{2} E_{p_i}(q) = -\begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \left(\sum_{i=1}^{2} m_i {}^0p_{G_i}(q) \right).$$

with

- ${}^0p_{G_i}(q)$ the position of the center of mass of $\mathscr{C}_i$ w.r.t. $\mathscr{R}_0$, so that :

$$\begin{bmatrix} {}^0p_{G_1}(q) \\ 1 \end{bmatrix} = \begin{bmatrix} {}^0O_0G_1 \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} C1 & -S1 & 0 & 0 \\ S1 & C1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\bar{g}_{01}} \begin{bmatrix} L_{c_1} \\ 0 \\ 0 \\ 1 \end{bmatrix}_{\mathscr{R}_1} = \begin{bmatrix} L_{c_1}C1 \\ L_{c_1}S1 \\ 0 \\ 1 \end{bmatrix}_{\mathscr{R}_0}$$

  and

$$\begin{bmatrix} {}^0p_{G_2}(q) \\ 1 \end{bmatrix} = \begin{bmatrix} {}^0O_0G_2 \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} C12 & -S12 & 0 & L_1C1 \\ S12 & C12 & 0 & L_1S1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\bar{g}_{02}} \begin{bmatrix} L_{c_2} \\ 0 \\ 0 \\ 1 \end{bmatrix}_{\mathscr{R}_2} = \begin{bmatrix} L_{c_2}C12 + L_1C1 \\ L_{c_2}S12 + L_1S1 \\ 0 \\ 1 \end{bmatrix}_{\mathscr{R}_0}$$

- Final expression of $E_p(q)$:

$$E_p(q) = m_1 g L_{c_1} \sin(q_1) + m_2 g \left(L_1 \sin(q_1) + L_{c_2} \sin(q_1 + q_2) \right)$$

Dynamics of motion

Lagrange equations :

$$\frac{d}{dt}\left(\frac{\partial \mathscr{L}}{\partial \dot{q}_k}\right) - \frac{\partial \mathscr{L}}{\partial q_k} = \tau_k \quad \text{for} \quad k = 1, \ldots, n$$

Dynamics of motion

Lagrange equations :

$$\frac{d}{dt}\left(\frac{\partial \mathscr{L}}{\partial \dot{q}_k}\right) - \frac{\partial \mathscr{L}}{\partial q_k} = \tau_k \quad \text{for} \quad k = 1, \ldots, n$$

where the Lagrangian of the system $\mathscr{L}$ is written as :

$$\begin{aligned}
\mathscr{L} &= E_c\left(q, \dot{q}\right) - E_p\left(q\right) \\
&= \frac{1}{2}\dot{q}^t A\left(q\right)\dot{q} + g^t\left(\sum_{i=1}^{N} m_i{}^0 p_{G_i}\left(q\right)\right) \\
&= \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} a_{ij}\left(q\right)\dot{q}_i\dot{q}_j + g^t\left(\sum_{i=1}^{N} m_i{}^0 p_{G_i}\left(q\right)\right)
\end{aligned}$$

Dynamics of motion

Lagrange equations :

$$\frac{d}{dt}\left(\frac{\partial \mathscr{L}}{\partial \dot{q}_k}\right) - \frac{\partial \mathscr{L}}{\partial q_k} = \tau_k \quad \text{for} \quad k = 1, \ldots, n$$

where the Lagrangian of the system $\mathscr{L}$ is written as :

$$\begin{aligned}
\mathscr{L} &= E_c\left(q, \dot{q}\right) - E_p\left(q\right) \\
&= \frac{1}{2}\dot{q}^t A\left(q\right)\dot{q} + g^t\left(\sum_{i=1}^{N} m_i {}^0p_{G_i}\left(q\right)\right) \\
&= \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} a_{ij}\left(q\right)\dot{q}_i\dot{q}_j + g^t\left(\sum_{i=1}^{N} m_i {}^0p_{G_i}\left(q\right)\right)
\end{aligned}$$

Computation of the partial derivative of $\mathscr{L}$:

Dynamics of motion

Lagrange equations :

$$\frac{d}{dt}\left(\frac{\partial \mathscr{L}}{\partial \dot{q}_k}\right) - \frac{\partial \mathscr{L}}{\partial q_k} = \tau_k \quad \text{for} \quad k = 1, \ldots, n$$

where the Lagrangian of the system $\mathscr{L}$ is written as :

$$\begin{aligned}
\mathscr{L} &= E_c(q, \dot{q}) - E_p(q) \\
&= \frac{1}{2}\dot{q}^t A(q)\,\dot{q} + g^t\left(\sum_{i=1}^{N} m_i\, {}^0p_{G_i}(q)\right) \\
&= \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} a_{ij}(q)\,\dot{q}_i\dot{q}_j + g^t\left(\sum_{i=1}^{N} m_i\, {}^0p_{G_i}(q)\right)
\end{aligned}$$

Computation of the partial derivative of $\mathscr{L}$:

- $\frac{\partial \mathscr{L}}{\partial \dot{q}_k} = \frac{\partial E_c}{\partial \dot{q}_k} = \sum_{j=1}^{N} a_{kj}(q)\,\dot{q}_j$, from which we deduce :

$$\begin{aligned}
\frac{d}{dt}\left(\frac{\partial \mathscr{L}}{\partial \dot{q}_k}\right) &= \sum_{j=1}^{N} a_{kj}(q)\,\ddot{q}_j + \sum_{j=1}^{N} \frac{da_{kj}(q)}{dt}\dot{q}_j \\
&= \sum_{j=1}^{N} a_{kj}(q)\,\ddot{q}_j + \sum_{i=1}^{N}\sum_{j=1}^{N} \frac{\partial a_{kj}(q)}{\partial q_i}\dot{q}_i\dot{q}_j
\end{aligned}$$

Dynamics of motion

Lagrange equations :

$$\frac{d}{dt}\left(\frac{\partial \mathscr{L}}{\partial \dot{q}_k}\right) - \frac{\partial \mathscr{L}}{\partial q_k} = \tau_k \quad \text{for} \quad k = 1, \ldots, n$$

where the Lagrangian of the system $\mathscr{L}$ is written as :

$$\begin{aligned}
\mathscr{L} &= E_c(q, \dot{q}) - E_p(q) \\
&= \frac{1}{2}\dot{q}^t A(q)\,\dot{q} + g^t\left(\sum_{i=1}^{N} m_i{}^0 p_{G_i}(q)\right) \\
&= \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} a_{ij}(q)\,\dot{q}_i\dot{q}_j + g^t\left(\sum_{i=1}^{N} m_i{}^0 p_{G_i}(q)\right)
\end{aligned}$$

Computation of the partial derivative of $\mathscr{L}$:

- $\frac{\partial \mathscr{L}}{\partial \dot{q}_k} = \frac{\partial E_c}{\partial \dot{q}_k} = \sum_{j=1}^{N} a_{kj}(q)\,\dot{q}_j$, from which we deduce :

$$\begin{aligned}
\frac{d}{dt}\left(\frac{\partial \mathscr{L}}{\partial \dot{q}_k}\right) &= \sum_{j=1}^{N} a_{kj}(q)\,\ddot{q}_j + \sum_{j=1}^{N} \frac{da_{kj}(q)}{dt}\dot{q}_j \\
&= \sum_{j=1}^{N} a_{kj}(q)\,\ddot{q}_j + \sum_{i=1}^{N}\sum_{j=1}^{N} \frac{\partial a_{kj}(q)}{\partial q_i}\dot{q}_i\dot{q}_j
\end{aligned}$$

- $\frac{\partial \mathscr{L}}{\partial q_k} = \frac{\partial E_c - E_p}{\partial q_k} = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \frac{\partial a_{ij}}{\partial q_k}\dot{q}_i\dot{q}_j - \frac{\partial E_p}{\partial q_k}$.

Dynamics of motion

The *Lagrange* equations take the following form, for $k = 1, \ldots, n$:

$$\sum_{j=1}^{N} a_{kj}(q)\,\ddot{q}_j + \sum_{i=1}^{N}\sum_{j=1}^{N}\left(\frac{\partial a_{kj}(q)}{\partial q_i} - \frac{1}{2}\frac{\partial a_{ij}(q)}{\partial q_k}\right)\dot{q}_i\dot{q}_j + \frac{\partial E_p(q)}{\partial q_k} = \tau_k$$

Dynamics of motion

The *Lagrange* equations take the following form, for $k = 1, \ldots, n$:

$$\sum_{j=1}^{N} a_{kj}(q)\,\ddot{q}_j + \sum_{i=1}^{N}\sum_{j=1}^{N}\left(\frac{\partial a_{kj}(q)}{\partial q_i} - \frac{1}{2}\frac{\partial a_{ij}(q)}{\partial q_k}\right)\dot{q}_i\dot{q}_j + \frac{\partial E_p(q)}{\partial q_k} = \tau_k$$

By posing g_k the term derived from potential gravity energy :

$$g_k(q) = \frac{\partial E_p(q)}{\partial q_k}$$

Dynamics of motion

The *Lagrange* equations take the following form, for $k = 1, \ldots, n$:

$$\sum_{j=1}^{N} a_{kj}(q)\,\ddot{q}_j + \sum_{i=1}^{N}\sum_{j=1}^{N}\left(\frac{\partial a_{kj}(q)}{\partial q_i} - \frac{1}{2}\frac{\partial a_{ij}(q)}{\partial q_k}\right)\dot{q}_i\dot{q}_j + \frac{\partial E_p(q)}{\partial q_k} = \tau_k$$

By posing g_k the term derived from potential gravity energy :

$$g_k(q) = \frac{\partial E_p(q)}{\partial q_k}$$

and h_{kji} the quantity :

$$h_{kji}(q) = \frac{\partial a_{kj}(q)}{\partial q_i} - \frac{1}{2}\frac{\partial a_{ij}(q)}{\partial q_k}$$

Dynamics of motion

The *Lagrange* equations take the following form, for $k = 1, \ldots, n$:

$$\sum_{j=1}^{N} a_{kj}(q)\,\ddot{q}_j + \sum_{i=1}^{N}\sum_{j=1}^{N}\left(\frac{\partial a_{kj}(q)}{\partial q_i} - \frac{1}{2}\frac{\partial a_{ij}(q)}{\partial q_k}\right)\dot{q}_i\dot{q}_j + \frac{\partial E_p(q)}{\partial q_k} = \tau_k$$

By posing g_k the term derived from potential gravity energy :

$$g_k(q) = \frac{\partial E_p(q)}{\partial q_k}$$

and h_{kji} the quantity :

$$h_{kji}(q) = \frac{\partial a_{kj}(q)}{\partial q_i} - \frac{1}{2}\frac{\partial a_{ij}(q)}{\partial q_k}$$

it follows that :

$$\sum_{j=1}^{N} a_{kj}(q)\,\ddot{q}_j + \sum_{i=1}^{N}\sum_{j=1}^{N} h_{kji}(q)\,\dot{q}_i\dot{q}_j + g_k(q) = \tau_k$$

Dynamics of motion

The product

$$\sum_{i=1}^{N}\sum_{j=1}^{N} h_{kji}\left(q\right)\dot{q}_i\dot{q}_j = \sum_{i=1}^{N}\sum_{j=1}^{N}\left(\frac{\partial a_{kj}\left(q\right)}{\partial q_i} - \frac{1}{2}\frac{\partial a_{ij}\left(q\right)}{\partial q_k}\right)\dot{q}_i\dot{q}_j$$

is a vector of size $(N \times 1)$ whose elements are some quadratic functions of the joint velocities.

Dynamics of motion

The product

$$\sum_{i=1}^{N}\sum_{j=1}^{N} h_{kji}(q)\,\dot{q}_i\dot{q}_j = \sum_{i=1}^{N}\sum_{j=1}^{N}\left(\frac{\partial a_{kj}(q)}{\partial q_i} - \frac{1}{2}\frac{\partial a_{ij}(q)}{\partial q_k}\right)\dot{q}_i\dot{q}_j$$

is a vector of size $(N \times 1)$ whose elements are some quadratic functions of the joint velocities. By swapping the (i, j) indices and exploiting symmetry, it follows :

$$\sum_{i=1}^{N}\sum_{j=1}^{N}\left(\frac{\partial a_{kj}(q)}{\partial q_i} - \frac{1}{2}\frac{\partial a_{ij}(q)}{\partial q_k}\right)\dot{q}_i\dot{q}_j = \sum_{i=1}^{N}\sum_{j=1}^{N}\frac{1}{2}\left(\frac{\partial a_{kj}(q)}{\partial q_i} + \frac{\partial a_{ki}(q)}{\partial q_j} - \frac{\partial a_{ij}(q)}{\partial q_k}\right)\dot{q}_i\dot{q}_j$$

$$= \sum_{i=1}^{N}\sum_{j=1}^{N} c_{ijk}\,\dot{q}_i\dot{q}_j$$

where $c_{ijk} = \frac{1}{2}\left(\frac{\partial a_{kj}(q)}{\partial q_i} + \frac{\partial a_{ki}(q)}{\partial q_j} - \frac{\partial a_{ij}(q)}{\partial q_k}\right)$ are the *Christoffel* symboles.

Dynamics of motion

The product

$$\sum_{i=1}^{N}\sum_{j=1}^{N} h_{kji}(q)\,\dot{q}_i\dot{q}_j = \sum_{i=1}^{N}\sum_{j=1}^{N}\left(\frac{\partial a_{kj}(q)}{\partial q_i} - \frac{1}{2}\frac{\partial a_{ij}(q)}{\partial q_k}\right)\dot{q}_i\dot{q}_j$$

is a vector of size $(N \times 1)$ whose elements are some quadratic functions of the joint velocities.
By swapping the (i,j) indices and exploiting symmetry, it follows :

$$\sum_{i=1}^{N}\sum_{j=1}^{N}\left(\frac{\partial a_{kj}(q)}{\partial q_i} - \frac{1}{2}\frac{\partial a_{ij}(q)}{\partial q_k}\right)\dot{q}_i\dot{q}_j = \sum_{i=1}^{N}\sum_{j=1}^{N}\frac{1}{2}\left(\frac{\partial a_{kj}(q)}{\partial q_i} + \frac{\partial a_{ki}(q)}{\partial q_j} - \frac{\partial a_{ij}(q)}{\partial q_k}\right)\dot{q}_i\dot{q}_j$$

$$= \sum_{i=1}^{N}\sum_{j=1}^{N} c_{ijk}\,\dot{q}_i\dot{q}_j$$

where $c_{ijk} = \frac{1}{2}\left(\frac{\partial a_{kj}(q)}{\partial q_i} + \frac{\partial a_{ki}(q)}{\partial q_j} - \frac{\partial a_{ij}(q)}{\partial q_k}\right)$ are the *Christoffel* symboles.
Since the matrix $A(q)$ is symmetric, it follows that, for a given k : $c_{ijk} = c_{jik}$.

Course

Study of one revolute robotic axis

Function of the Lagrangian :

Study of one revolute robotic axis

Function of the Lagrangian :

$$\begin{aligned}\mathscr{L}(q,\dot{q}) &= E_{c_i}(q,\dot{q}) - E_{p_i}(q) \\ &= \frac{1}{2}\left(m_i d_i^2 + I_{G_i} + r_{red_i}^2 J_{m_i}\right)\dot{q}_i^2 - m_i g d_i\left(1 - cos\left(q_i\right)\right)\end{aligned}$$

Study of one revolute robotic axis

Function of the Lagrangian :

$$\begin{aligned}\mathscr{L}(q,\dot{q}) &= E_{c_i}(q,\dot{q}) - E_{p_i}(q)\\ &= \frac{1}{2}\left(m_i d_i^2 + I_{G_i} + r_{red_i}^2 J_{m_i}\right)\dot{q}_i^2 - m_i g d_i\left(1 - cos\left(q_i\right)\right)\end{aligned}$$

from which we deduce that :

$$\frac{\partial \mathscr{L}}{\partial \dot{q}} = \left(m_i d_i^2 + I_{G_i} + r_{red_i}^2 J_{m_i}\right)\dot{q}_i \quad \Rightarrow \quad \frac{d}{dt}\frac{\partial \mathscr{L}}{\partial \dot{q}} = \left(m_i d_i^2 + I_{G_i} + r_{red_i}^2 J_{m_i}\right)\ddot{q}_i,$$

Study of one revolute robotic axis

Function of the Lagrangian :

$$\begin{aligned}\mathscr{L}(q,\dot{q}) &= E_{c_i}(q,\dot{q}) - E_{p_i}(q) \\ &= \frac{1}{2}\left(m_i d_i^2 + I_{G_i} + r_{red_i}^2 J_{m_i}\right)\dot{q}_i^2 - m_i g d_i\left(1 - cos\left(q_i\right)\right)\end{aligned}$$

from which we deduce that :

$$\frac{\partial \mathscr{L}}{\partial \dot{q}} = \left(m_i d_i^2 + I_{G_i} + r_{red_i}^2 J_{m_i}\right)\dot{q}_i \quad \Rightarrow \quad \frac{d}{dt}\frac{\partial \mathscr{L}}{\partial \dot{q}} = \left(m_i d_i^2 + I_{G_i} + r_{red_i}^2 J_{m_i}\right)\ddot{q}_i,$$

and

$$\frac{\partial \mathscr{L}}{\partial q} = -m_i g d_i sin\left(q_i\right).$$

Study of one revolute robotic axis

Function of the Lagrangian :

$$\mathscr{L}(q,\dot{q}) = E_{c_i}(q,\dot{q}) - E_{p_i}(q)$$
$$= \frac{1}{2}\left(m_i d_i^2 + I_{G_i} + r_{red_i}^2 J_{m_i}\right)\dot{q}_i^2 - m_i g d_i \left(1 - cos\left(q_i\right)\right)$$

from which we deduce that :

$$\frac{\partial \mathscr{L}}{\partial \dot{q}} = \left(m_i d_i^2 + I_{G_i} + r_{red_i}^2 J_{m_i}\right)\dot{q}_i \quad \Rightarrow \quad \frac{d}{dt}\frac{\partial \mathscr{L}}{\partial \dot{q}} = \left(m_i d_i^2 + I_{G_i} + r_{red_i}^2 J_{m_i}\right)\ddot{q}_i,$$

and

$$\frac{\partial \mathscr{L}}{\partial q} = -m_i g d_i sin\left(q_i\right).$$

Generalized efforts taking into account the torque applied to the joint and the friction effects :

$$\psi_i = \tau_i - \tau_{f_i}.$$

Study of one revolute robotic axis

Function of the Lagrangian :

$$\begin{aligned}\mathscr{L}(q,\dot{q}) &= E_{c_i}(q,\dot{q}) - E_{p_i}(q)\\ &= \frac{1}{2}\left(m_i d_i^2 + I_{G_i} + r_{red_i}^2 J_{m_i}\right)\dot{q}_i^2 - m_i g d_i\left(1 - cos\left(q_i\right)\right)\end{aligned}$$

from which we deduce that :

$$\frac{\partial \mathscr{L}}{\partial \dot{q}} = \left(m_i d_i^2 + I_{G_i} + r_{red_i}^2 J_{m_i}\right)\dot{q}_i \quad \Rightarrow \quad \frac{d}{dt}\frac{\partial \mathscr{L}}{\partial \dot{q}} = \left(m_i d_i^2 + I_{G_i} + r_{red_i}^2 J_{m_i}\right)\ddot{q}_i,$$

and

$$\frac{\partial \mathscr{L}}{\partial q} = -m_i g d_i sin\left(q_i\right).$$

Generalized efforts taking into account the torque applied to the joint and the friction effects :

$$\psi_i = \tau_i - \tau_{f_i}.$$

2^{nd} order differential equation to describe the motion :

$$\left(m_i d_i^2 + I_{G_i} + r_{red_i}^2 J_{m_i}\right)\ddot{q}_i + f_{\dot{q}0}\dot{q}_i + m_i g d_i sin\left(q_i\right) = \tau_i,$$

choosing a joint visquous friction as follows $\tau_{f_i} = f_{\dot{q}0}\dot{q}_i$ where $f_{\dot{q}0} > 0$.

Course

Dynamics for the planar RR robot

Function of Lagrangian :

$$\begin{aligned}\mathscr{L}(q,\dot{q}) &= \sum_{i=1}^{2}\left(E_{c_i}(q,\dot{q}) - E_{p_i}(q)\right)\\ &= \frac{1}{2}\dot{q}^t A(q)\,\dot{q} - \left(m_1 g L_{c_1} sin(q_1) + m_2 g\left(L_1 sin(q_1) + L_{c_2} sin(q_1+q_2)\right)\right)\end{aligned}$$

Dynamics for the planar RR robot

Function of Lagrangian :

$$\begin{aligned}\mathscr{L}(q,\dot{q}) &= \sum_{i=1}^{2}\left(E_{c_i}(q,\dot{q}) - E_{p_i}(q)\right)\\ &= \frac{1}{2}\dot{q}^t A(q)\,\dot{q} - \left(m_1 g L_{c_1} sin(q_1) + m_2 g\left(L_1 sin(q_1) + L_{c_2} sin(q_1+q_2)\right)\right)\end{aligned}$$

where the already computed inertia matrix $A(q)$ is equal to :

$$A(q) = \begin{bmatrix} a_{11}(q) & a_{12}(q) \\ (sym.) & a_{22}(q) \end{bmatrix}$$

with

$$\begin{aligned}a_{11}(q) &= r_{red_1}^2 J_{m_1} + m_1 L_{c_1}^2 + m_2\left(L_1^2 + L_{c_2}^2 + 2L_1 L_{c_2} cos(q_2)\right) + I_{zz_1} + I_{zz_2}\\ a_{12}(q) &= m_2\left(L_{c_2}^2 + L_1 L_{c_2} cos(q_2)\right) + I_{zz_2}\\ a_{22}(q) &= r_{red_2}^2 J_{m_2} + m_2 L_{c_2}^2 + I_{zz_2}\end{aligned}$$

Dynamics for the planar RR robot

Computation of the *Christoffel* symbols : $c_{ijk}(q) = \frac{1}{2}\left(\frac{\partial a_{kj}}{\partial q_i} + \frac{\partial a_{ki}}{\partial q_j} - \frac{\partial a_{ij}}{\partial q_k}\right)$

$$c_{111}(q) = \frac{1}{2}\left(\frac{\partial a_{11}}{\partial q_1} + \frac{\partial a_{11}}{\partial q_1} - \frac{\partial a_{11}}{\partial q_1}\right) = \frac{1}{2}\frac{\partial a_{11}}{\partial q_1} = 0$$

$$c_{121}(q) = \frac{1}{2}\left(\frac{\partial a_{12}}{\partial q_1} + \frac{\partial a_{11}}{\partial q_2} - \frac{\partial a_{12}}{\partial q_1}\right) = \frac{1}{2}\frac{\partial a_{11}}{\partial q_2} = -m_2 L_1 L_{c_2} sin(q_2) = h$$

$$c_{211}(q) = c_{121}(q) = h \text{ (using symmetry with } c_{ijk})$$

$$c_{221}(q) = \frac{\partial a_{12}}{\partial q_2} - \frac{1}{2}\frac{\partial a_{22}}{\partial q_1} = \frac{\partial a_{12}}{\partial q_2} = h$$

$$c_{112}(q) = \frac{\partial a_{21}}{\partial q_1} - \frac{1}{2}\frac{\partial a_{11}}{\partial q_2} = -\frac{1}{2}\frac{\partial a_{11}}{\partial q_2} = -h$$

$$c_{122}(q) = \frac{1}{2}\frac{\partial a_{22}}{\partial q_1} = 0$$

$$c_{212}(q) = c_{122}(q) = 0 \text{ (using symmetry with } c_{ijk})$$

$$c_{222}(q) = \frac{1}{2}\frac{\partial a_{22}}{\partial q_2} = 0$$

Dynamics for the planar RR robot

Matrix $C(q, \dot{q})$ given by :

$$[C(q, \dot{q})]_{k,j} = \sum_{i=1}^{N} c_{ijk}(q)\, \dot{q}_i$$

Dynamics for the planar RR robot

Matrix $C(q, \dot{q})$ given by :

$$[C(q, \dot{q})]_{k,j} = \sum_{i=1}^{N} c_{ijk}(q)\, \dot{q}_i$$

thus

$$C(q, \dot{q}) = \begin{bmatrix} h\dot{q}_2 & h(\dot{q}_1 + \dot{q}_2) \\ -h\dot{q}_1 & 0 \end{bmatrix}$$

Dynamics for the planar RR robot

Matrix $C(q,\dot{q})$ given by :

$$[C(q,\dot{q})]_{k,j} = \sum_{i=1}^{N} c_{ijk}(q)\,\dot{q}_i$$

thus

$$C(q,\dot{q}) = \begin{bmatrix} h\dot{q}_2 & h(\dot{q}_1+\dot{q}_2) \\ -h\dot{q}_1 & 0 \end{bmatrix}$$

Vector of forces deriving from a gravity potential :

$$\begin{aligned} G(q) &= \begin{bmatrix} g_1(q) \\ g_2(q) \end{bmatrix} \\ &= \begin{bmatrix} \frac{\partial E_p(q)}{\partial q_1} \\ \frac{\partial E_p(q)}{\partial q_2} \end{bmatrix} \\ &= \begin{bmatrix} (m_1 L_{c_1} + m_2 L_1)\, g\cos(q_1) + m_2 L_{c_2}\, g\cos(q_1+q_2) \\ m_2 L_{c_2}\, g\cos(q_1+q_2) \end{bmatrix} \end{aligned}$$

Dynamics for the planar RR robot

Expanded form of $A(q)\,\ddot{q} + C(q,\dot{q})\,\dot{q} + G(q) = \Gamma$:

Dynamics for the planar RR robot

Expanded form of $A(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \Gamma$:

$$a_{11}(q)\ddot{q}_1 + a_{12}(q)\ddot{q}_2 + c_{121}(q)\dot{q}_1\dot{q}_2 + c_{211}(q)\dot{q}_2\dot{q}_1 + c_{221}(q)\dot{q}_2^2 + g_1(q) = \tau_1$$
$$a_{21}(q)\ddot{q}_1 + a_{22}(q)\ddot{q}_2 + c_{112}(q)\dot{q}_1^2 + g_2(q) = \tau_2$$

thus

$$\begin{aligned}
&\left(r_{red_1}^2 J_{m_1} + m_1 L_{c_1}^2 + m_2\left(L_1^2 + L_{c_2}^2 + 2L_1L_{c_2}cos(q_2)\right) + I_{zz_1} + I_{zz_2}\right)\ddot{q}_1 \\
&\quad + \left(m_2\left(L_{c_2}^2 + L_1L_{c_2}cos(q_2)\right) + I_{zz_2}\right)\ddot{q}_2 \\
&\quad - m_2L_1L_{c_2}sin(q_2)\dot{q}_1^2 - 2m_2L_1L_{c_2}sin(q_2)\dot{q}_1\dot{q}_2 \\
&\quad + \left(m_1L_{c_1} + m_2L_1\right)gcos(q_1) + m_2L_{c_2}gcos(q_1+q_2) = \tau_1
\end{aligned}$$

$$\begin{aligned}
&\left(m_2\left(L_{c_2}^2 + L_1L_{c_2}cos(q_2)\right) + I_{zz_2}\right)\ddot{q}_1 + \left(r_{red_2}^2 J_{m_2} + m_2L_{c_2}^2 + I_{zz_2}\right)\ddot{q}_2 \\
&\quad + m_2L_1L_{c_2}sin(q_2)\dot{q}_1^2 \\
&\quad + m_2L_{c_2}gcos(q_1+q_2) = \tau_2
\end{aligned}$$

▸ Course

Properties of the vector $c(q, \dot{q}) = C(q, \dot{q})\,\dot{q}$

Skew-symmetry of $\left(\dot{A}(q) - 2C(q, \dot{q})\right)$ of generic term (n_{kj}), where $C(q, \dot{q})$ is defined from the *Christoffel* symbols

Properties of the vector $c(q,\dot{q}) = C(q,\dot{q})\,\dot{q}$

Skew-symmetry of $\left(\dot{A}(q) - 2C(q,\dot{q})\right)$ of generic term (n_{kj}), where $C(q,\dot{q})$ is defined from the *Christoffel* symbols

$$
\begin{aligned}
n_{kj} =& \dot{a}_{kj} - 2\,[C]_{kj} \\
=& \sum_i \frac{\partial a_{kj}}{\partial q_i}\dot{q}_i - \sum_i \left(\frac{\partial a_{kj}}{\partial q_i} + \frac{\partial a_{ki}}{\partial q_j} - \frac{\partial a_{ij}}{\partial q_k}\right)\dot{q}_i \\
=& \sum_i \left(\frac{\partial a_{kj}}{\partial q_i} - \left(\frac{\partial a_{kj}}{\partial q_i} + \frac{\partial a_{ki}}{\partial q_j} - \frac{\partial a_{ij}}{\partial q_k}\right)\right)\dot{q}_i \\
=& \sum_i \left(\frac{\partial a_{ij}}{\partial q_k} - \frac{\partial a_{ki}}{\partial q_j}\right)\dot{q}_i
\end{aligned}
$$

Properties of the vector $c(q,\dot q) = C(q,\dot q)\,\dot q$

Skew-symmetry of $\left(\dot A(q) - 2C(q,\dot q)\right)$ of generic term (n_{kj}), where $C(q,\dot q)$ is defined from the *Christoffel* symbols

$$
\begin{aligned}
n_{kj} =& \dot a_{kj} - 2\left[C\right]_{kj} \\
=& \sum_i \frac{\partial a_{kj}}{\partial q_i}\dot q_i - \sum_i \left(\frac{\partial a_{kj}}{\partial q_i} + \frac{\partial a_{ki}}{\partial q_j} - \frac{\partial a_{ij}}{\partial q_k}\right)\dot q_i \\
=& \sum_i \left(\frac{\partial a_{kj}}{\partial q_i} - \left(\frac{\partial a_{kj}}{\partial q_i} + \frac{\partial a_{ki}}{\partial q_j} - \frac{\partial a_{ij}}{\partial q_k}\right)\right)\dot q_i \\
=& \sum_i \left(\frac{\partial a_{ij}}{\partial q_k} - \frac{\partial a_{ki}}{\partial q_j}\right)\dot q_i
\end{aligned}
$$

Besides,

$$
\begin{aligned}
n_{jk} =& \dot a_{jk} - 2\left[C\right]_{jk} \\
=& \dots \\
=& \sum_i \left(\frac{\partial a_{ik}}{\partial q_j} - \frac{\partial a_{ji}}{\partial q_k}\right)\dot q_i \\
=& -\sum_i \left(\frac{\partial a_{ji}}{\partial q_k} - \frac{\partial a_{ik}}{\partial q_j}\right)\dot q_i \\
=& -\sum_i \left(\frac{\partial a_{ij}}{\partial q_k} - \frac{\partial a_{ki}}{\partial q_j}\right)\dot q_i \qquad (A(q) \text{ sym., i.e. } a_{ji} = a_{ij} \text{ et } a_{ki} = a_{ik})
\end{aligned}
$$

Properties of the vector $c(q, \dot{q}) = C(q, \dot{q})\,\dot{q}$

Thus, $n_{kj} = -n_{jk}$ (skew-symmetry proven), thus

$$\forall x \in \mathbb{R}^n, \quad x^t \left(\dot{A}(q) - 2C(q, \dot{q})\right) x = 0.$$

Properties of the vector $c(q,\dot{q}) = C(q,\dot{q})\,\dot{q}$

Thus, $n_{kj} = -n_{jk}$ (skew-symmetry proven), thus

$$\forall x \in \mathbb{R}^n, \quad x^t \left(\dot{A}(q) - 2C(q,\dot{q})\right) x = 0.$$

By the way, the following poperty :

$$q^t \left(\dot{A}(q) - 2C(q,\dot{q})\right) \dot{q} = 0.$$

is still true, even if the matrix $C(q,\dot{q})$, not unique, is not defined from the *Christoffel* symbols (the proof lies on the fact that in the absence of control action on the robot $\Gamma = 0$, the variation of the total energy $\left(\dot{E}_c + \dot{E}_p\right)$ is null).

Course

Properties of the dynamic model

Recall : definition of the norm $\|\bullet\|_1$ in $\mathbb{R}^n$.
Let consider $c = (c_1, \ldots, c_n)^t \in \mathbb{R}^n$ and $A = (a_{i,j})_{1 \leq i,j \leq n}$.

$$\|c\|_1 = \sum_{i=1}^{n} |c_i| \quad \|A\|_1 = \max_j \left(\sum_{i=1}^{n} |a_{i,j}| \right)$$

Properties of the dynamic model

Recall : definition of the norm $\|\bullet\|_1$ in $\mathbb{R}^n$.
Let consider $c = (c_1, \ldots, c_n)^t \in \mathbb{R}^n$ and $A = (a_{i,j})_{1 \le i,j \le n}$.

$$\|c\|_1 = \sum_{i=1}^{n} |c_i| \quad \|A\|_1 = \max_j \left(\sum_{i=1}^{n} |a_{i,j}| \right)$$

Bounds of the inertia matrix :

$$\begin{aligned}
\|A(q)\|_1 &= \max \left\{ \sum_{i=1}^{n} |a_{i,1}(q)|, \sum_{i=1}^{n} |a_{i,2}(q)(q)| \right\} \\
&= \max \left\{ |a_{1,1}(q)| + |a_{2,1}(q)|, |a_{1,2}(q)| + |a_{2,2}(q)| \right\} \\
&= \max \left\{ \left| r_{red_1}^2 J_{m_1} + m_1 L_{c_1}^2 + m_2 \left(L_1^2 + L_{c_2}^2 + 2 L_1 L_{c_2} C2 \right) + I_{zz_1} + I_{zz_2} \right| + \left| m_2 \left(L_{c_2}^2 + L_1 L_{c_2} C2 \right) + I_{zz_2} \right|, \right. \\
&\quad \left. \left| m_2 \left(L_{c_2}^2 + L_1 L_{c_2} C2 \right) + I_{zz_2} \right| + \left| r_{red_2}^2 J_{m_2} + m_2 L_{c_2}^2 + I_{zz_2} \right| \right\}
\end{aligned}$$

Properties of the dynamic model

Recall : definition of the norm $\|\bullet\|_1$ in $\mathbb{R}^n$.
Let consider $c = (c_1, \dots, c_n)^t \in \mathbb{R}^n$ and $A = (a_{i,j})_{1 \le i,j \le n}$.

$$\|c\|_1 = \sum_{i=1}^{n} |c_i| \quad \|A\|_1 = \max_j \left(\sum_{i=1}^{n} |a_{i,j}| \right)$$

Bounds of the inertia matrix :

$$\begin{aligned}
\|A(q)\|_1 &= \max \left\{ \sum_{i=1}^{n} |a_{i,1}(q)|, \sum_{i=1}^{n} |a_{i,2}(q)(q)| \right\} \\
&= \max \left\{ |a_{1,1}(q)| + |a_{2,1}(q)|, |a_{1,2}(q)| + |a_{2,2}(q)| \right\} \\
&= \max \left\{ \left| r_{red_1}^2 J_{m_1} + m_1 L_{c_1}^2 + m_2 \left(L_1^2 + L_{c_2}^2 + 2 L_1 L_{c_2} C2 \right) + I_{zz_1} + I_{zz_2} \right| + \left| m_2 \left(L_{c_2}^2 + L_1 L_{c_2} C2 \right) + I_{zz_2} \right|, \right. \\
&\quad \left. \left| m_2 \left(L_{c_2}^2 + L_1 L_{c_2} C2 \right) + I_{zz_2} \right| + \left| r_{red_2}^2 J_{m_2} + m_2 L_{c_2}^2 + I_{zz_2} \right| \right\}
\end{aligned}$$

Assuming that q_1 et q_2 belong to $\left[-\frac{\pi}{2}, \frac{\pi}{2} \right]$,

$$\alpha_1 \le \|A(q)\|_1 \le \alpha_2$$

Properties of the dynamic model

Recall : definition of the norm $\|\bullet\|_1$ in $\mathbb{R}^n$.
Let consider $c = (c_1, \ldots, c_n)^t \in \mathbb{R}^n$ and $A = (a_{i,j})_{1 \leq i,j \leq n}$.

$$\|c\|_1 = \sum_{i=1}^{n} |c_i| \quad \|A\|_1 = \max_j \left(\sum_{i=1}^{n} |a_{i,j}| \right)$$

Bounds of the inertia matrix :

$$\begin{aligned}
\|A(q)\|_1 &= \max \left\{ \sum_{i=1}^{n} |a_{i,1}(q)|, \sum_{i=1}^{n} |a_{i,2}(q)(q)| \right\} \\
&= \max \left\{ |a_{1,1}(q)| + |a_{2,1}(q)|, |a_{1,2}(q)| + |a_{2,2}(q)| \right\} \\
&= \max \left\{ \left| r_{red_1}^2 J_{m_1} + m_1 L_{c_1}^2 + m_2 \left(L_1^2 + L_{c_2}^2 + 2 L_1 L_{c_2} C2 \right) + I_{zz_1} + I_{zz_2} \right| + \left| m_2 \left(L_{c_2}^2 + L_1 L_{c_2} C2 \right) + I_{zz_2} \right|, \right. \\
&\quad \left. \left| m_2 \left(L_{c_2}^2 + L_1 L_{c_2} C2 \right) + I_{zz_2} \right| + \left| r_{red_2}^2 J_{m_2} + m_2 L_{c_2}^2 + I_{zz_2} \right| \right\}
\end{aligned}$$

Assuming that q_1 et q_2 belong to $\left[-\frac{\pi}{2}, \frac{\pi}{2} \right]$,

$$\alpha_1 \leq \|A(q)\|_1 \leq \alpha_2$$

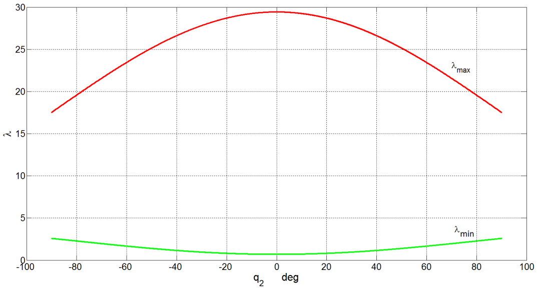
where

$$\alpha_1 = \max \left\{ r_{red_1}^2 J_{m_1} + m_1 L_{c_1}^2 + m_2 \left(L_1^2 + 2 L_{c_2}^2 \right) + I_{zz_1} + 2 I_{zz_2}, 2 m_2 L_{c_2}^2 + 2 I_{zz_2} + r_{red_2}^2 J_{m_2} \right\}$$

$$\alpha_2 = \max \left\{ r_{red_1}^2 J_{m_1} + m_1 L_{c_1}^2 + m_2 \left(L_1^2 + 2 L_{c_2}^2 + 3 L_1 L_{c_2} \right) + I_{zz_1} + 2 I_{zz_2}, m_2 \left(2 L_{c_2}^2 + L_1 L_{c_2} \right) + 2 I_{zz_2} + r_{red_2}^2 J_{m_2} \right\}$$

Properties of the dynamic model

Bounds of the inertia matrix :
the scalar quantities μ_1 and μ_2 can be defined as the minimal and maximal eigenvalues $(\lambda_{min}, \lambda_{max})$ of $A(q)$, for all q.



$$\lambda_{min} = 0,68$$
$$\lambda_{max} = 29,42$$

Numerical applications :
$m_1 = m_2 = 10\text{kg},\ L_1 = L_2 = 1\text{m},\ L_{c_1} = L_{c_2} = 0.5\text{m},\ I_{zz_1} = I_{zz_2} = 0.87\ kg.m^2$ et $J_{m_1} = J_{m_2} = 0\ kg.m^2$.

Properties of the dynamic model

Bounds of vector $c(q, \dot{q})$:

$$\begin{aligned} \|c(q, \dot{q})\|_1 &= \sum_{i=1}^{n} |c_i(q, \dot{q})| \\ &= |c_1(q, \dot{q})| + |c_2(q, \dot{q})| \\ &= \left| h(q) \left(2\dot{q}_1 \dot{q}_2 + \dot{q}_2^2 \right) \right| + \left| h(q)\, \dot{q}_1^2 \right| \end{aligned}$$

where $h(q) = -m_2 L_1 L_{c_2} sin(q_2)$.

Properties of the dynamic model

Bounds of vector $c(q, \dot{q})$:

$$\begin{aligned} \|c(q,\dot{q})\|_1 &= \sum_{i=1}^{n} |c_i(q,\dot{q})| \\ &= |c_1(q,\dot{q})| + |c_2(q,\dot{q})| \\ &= \left| h(q)\left(2\dot{q}_1\dot{q}_2 + \dot{q}_2^2\right)\right| + \left| h(q)\,\dot{q}_1^2 \right| \end{aligned}$$

where $h(q) = -m_2 L_1 L_{c_2} sin(q_2)$.

$$\|c(q,\dot{q})\|_1 \leq m_2 L_1 L_{c_2} \left| 2\dot{q}_1\dot{q}_2 + \dot{q}_2^2 + \dot{q}_1^2 \right| \leq m_2 L_1 L_{c_2} \underbrace{\left(|2\dot{q}_1\dot{q}_2| + \left|\dot{q}_2^2\right| + \left|\dot{q}_1^2\right| \right)}_{(|\dot{q}_1|+|\dot{q}_2|)^2} = \beta \|\dot{q}\|_1^2$$

Properties of the dynamic model

Bounds of vector $c(q,\dot{q})$:

$$
\begin{aligned}
\|c(q,\dot{q})\|_1 &= \sum_{i=1}^{n} |c_i(q,\dot{q})| \\
&= |c_1(q,\dot{q})| + |c_2(q,\dot{q})| \\
&= \left| h(q)\left(2\dot{q}_1\dot{q}_2 + \dot{q}_2^2\right)\right| + \left|h(q)\,\dot{q}_1^2\right|
\end{aligned}
$$

where $h(q) = -m_2 L_1 L_{c_2} sin(q_2)$.

$$
\|c(q,\dot{q})\|_1 \le m_2 L_1 L_{c_2}\left|2\dot{q}_1\dot{q}_2 + \dot{q}_2^2 + \dot{q}_1^2\right| \le m_2 L_1 L_{c_2}\underbrace{\left(|2\dot{q}_1\dot{q}_2| + \left|\dot{q}_2^2\right| + \left|\dot{q}_1^2\right|\right)}_{(|\dot{q}_1|+|\dot{q}_2|)^2} = \beta\,\|\dot{q}\|_1^2
$$

Skew-symmetry of matrix $\left(\dot{A}(q) - 2C(q,\dot{q})\right)$:

$$
\begin{aligned}
\dot{A}(q) - 2C(q,\dot{q}) &= \begin{bmatrix} 2h\dot{q}_2 & h\dot{q}_2 \\ h\dot{q}_2 & 0 \end{bmatrix} - 2\begin{bmatrix} h\dot{q}_2 & h(\dot{q}_1+\dot{q}_2) \\ -h\dot{q}_1 & 0 \end{bmatrix} \\
&= \begin{bmatrix} 0 & -h\dot{q}_2 - 2h\dot{q}_1 \\ h\dot{q}_2 + 2h\dot{q}_1 & 0 \end{bmatrix}
\end{aligned}
$$

Properties of the dynamic model

Bounds for the gravity vector $G(q)$:

$$
\begin{aligned}
\|G(q)\|_1 &= \sum_{i=1}^{2} |g_i(q)| \\
&= \left|\left(m_1 L_{c_1} + m_2 L_1\right) g\cos(q_1) + m_2 L_{c_2} g\cos(q_1+q_2)\right| + \left|m_2 L_{c_2} g\cos(q_1+q_2)\right| \\
&\leq \underbrace{(m_1 L_{c_1} + m_2 L_1 + 2 m_2 L_{c_2}) g}_{g_b}
\end{aligned}
$$

Course

Dynamics in the task space

Relationship between the cartesian effort and the actuating joint torques :

$$\Gamma = J(q)^t \mathscr{F}$$

Dynamics in the task space

Relationship between the cartesian effort and the actuating joint torques :

$$\Gamma = J(q)^t \mathscr{F}$$

and the cartesian and joint velocities :

$$\dot{X} = J(q)\,\dot{q}$$

Dynamics in the task space

Relationship between the cartesian effort and the actuating joint torques :

$$\Gamma = J\left(q\right)^t \mathscr{F}$$

and the cartesian and joint velocities :

$$\dot{X} = J\left(q\right)\dot{q}$$

Acceleration of the end-effector $\ddot{X} \in \mathbb{R}^6$ obtained by derivation :

$$\ddot{X} = J\left(q\right)\ddot{q} + \dot{J}\left(q\right)\dot{q}$$

Dynamics in the task space

Relationship between the cartesian effort and the actuating joint torques :

$$\Gamma = J(q)^t \mathscr{F}$$

and the cartesian and joint velocities :

$$\dot{X} = J(q)\,\dot{q}$$

Acceleration of the end-effector $\ddot{X} \in \mathbb{R}^6$ obtained by derivation :

$$\ddot{X} = J(q)\,\ddot{q} + \dot{J}(q)\,\dot{q}$$

Direct dynamic model given in the joint space :

$$\ddot{q} = A(q)^{-1}\left(\Gamma - C(q,\dot{q})\,\dot{q} - G(q)\right) = A(q)^{-1}\left(J(q)^t \mathscr{F} - C(q,\dot{q})\,\dot{q} - G(q)\right)$$

Dynamics in the task space

Relationship between the cartesian effort and the actuating joint torques :

$$\Gamma = J(q)^t \mathscr{F}$$

and the cartesian and joint velocities :

$$\dot{X} = J(q)\,\dot{q}$$

Acceleration of the end-effector $\ddot{X} \in \mathbb{R}^6$ obtained by derivation :

$$\ddot{X} = J(q)\,\ddot{q} + \dot{J}(q)\,\dot{q}$$

Direct dynamic model given in the joint space :

$$\ddot{q} = A(q)^{-1}\left(\Gamma - C(q,\dot{q})\,\dot{q} - G(q)\right) = A(q)^{-1}\left(J(q)^t \mathscr{F} - C(q,\dot{q})\,\dot{q} - G(q)\right)$$

by substituting $\ddot{q}$ in the expression of $\ddot{X}$:

$$\begin{aligned}\ddot{X} &= J(q)\,A(q)^{-1}\left(J(q)^t \mathscr{F} - C(q,\dot{q})\,\dot{q} - G(q)\right) + \dot{J}(q)^t\,\dot{q}\\ &= \left(J(q)\,A(q)^{-1}\,J(q)^t\right)\mathscr{F} - J(q)\,A(q)^{-1}\,C(q,\dot{q})\,\dot{q} - J(q)\,A(q)^{-1}\,G(q) + \dot{J}(q)\,\dot{q}\end{aligned}$$

Dynamics in the task space

Relationship between the cartesian effort and the actuating joint torques :

$$\Gamma = J(q)^t \mathscr{F}$$

and the cartesian and joint velocities :

$$\dot{X} = J(q)\,\dot{q}$$

Acceleration of the end-effector $\ddot{X} \in \mathbb{R}^6$ obtained by derivation :

$$\ddot{X} = J(q)\,\ddot{q} + \dot{J}(q)\,\dot{q}$$

Direct dynamic model given in the joint space :

$$\ddot{q} = A(q)^{-1}(\Gamma - C(q,\dot{q})\,\dot{q} - G(q)) = A(q)^{-1}\left(J(q)^t \mathscr{F} - C(q,\dot{q})\,\dot{q} - G(q)\right)$$

by substituting $\ddot{q}$ in the expression of $\ddot{X}$:

$$\begin{aligned}\ddot{X} &= J(q)\,A(q)^{-1}\left(J(q)^t \mathscr{F} - C(q,\dot{q})\,\dot{q} - G(q)\right) + \dot{J}(q)^t\,\dot{q} \\ &= \left(J(q)\,A(q)^{-1}\,J(q)^t\right)\mathscr{F} - J(q)\,A(q)^{-1}\,C(q,\dot{q})\,\dot{q} - J(q)\,A(q)^{-1}\,G(q) + \dot{J}(q)\,\dot{q}\end{aligned}$$

In the case where $J(q)$ is non-singular and full row rank, we conclude that :

$$\Lambda(q)\,\ddot{X} + \underbrace{\Lambda(q)\,J(q)\,A(q)^{-1}\,C(q,\dot{q})\,\dot{q} - \Lambda(q)\,\dot{J}(q)\,\dot{q}}_{\mu(q,\dot{q})} + \underbrace{\Lambda(q)\,J(q)\,A(q)^{-1}\,G(q)}_{p(q)} = \mathscr{F}$$

where we posed $\Lambda(q) = \left(J(q)\,A(q)^{-1}\,J(q)^t\right)^{-1}$.

Course

Rewritting the dynamic model of rigid-link robot

Inverse dynamic model :

Rewritting the dynamic model of rigid-link robot

Inverse dynamic model :

$$\underbrace{\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}}_{\Gamma} = \underbrace{\begin{bmatrix} r_{red_1}^2 J_{m_1} + ZZ_1 + ZZ_2 + 2L_1 MZ_2 cos\left(q_2\right) + m_2 L_1^2 & ZZ_2 + L_1 MZ_2 cos\left(q_2\right) \\ (\text{sym.}) & r_{red_2}^2 J_{m_2} + ZZ_2 \end{bmatrix}}_{A(q)} \ddot{q} + \dots$$

$$\underbrace{\begin{bmatrix} -L_1 MZ_2 sin\left(q_2\right) \dot{q}_2 & -L_1 MZ_2 sin\left(q_2\right)\left(\dot{q}_1 + \dot{q}_2\right) \\ L_1 MZ_2 sin\left(q_2\right) \dot{q}_1 & 0 \end{bmatrix}}_{C(q,\dot{q})} \dot{q} + \dots$$

$$\underbrace{\begin{bmatrix} \left(MZ_1 + m_2 L_1\right) gcos\left(q_1\right) + MZ_2 gcos\left(q_1 + q_2\right) \\ MZ_2 gcos\left(q_1 + q_2\right) \end{bmatrix}}_{G(q)}$$

Rewritting the dynamic model of rigid-link robot

Inverse dynamic model :

$$\underbrace{\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}}_{\Gamma} = \underbrace{\begin{bmatrix} r_{red_1}^2 J_{m_1} + ZZ_1 + ZZ_2 + 2L_1 MZ_2 cos\left(q_2\right) + m_2 L_1^2 & ZZ_2 + L_1 MZ_2 cos\left(q_2\right) \\ \text{(sym.)} & r_{red_2}^2 J_{m_2} + ZZ_2 \end{bmatrix}}_{A(q)} \ddot{q} + \dots$$

$$\underbrace{\begin{bmatrix} -L_1 MZ_2 sin\left(q_2\right)\dot{q}_2 & -L_1 MZ_2 sin\left(q_2\right)\left(\dot{q}_1 + \dot{q}_2\right) \\ L_1 MZ_2 sin\left(q_2\right)\dot{q}_1 & 0 \end{bmatrix}}_{C(q,\dot{q})} \dot{q} + \dots$$

$$\underbrace{\begin{bmatrix} \left(MZ_1 + m_2 L_1\right) gcos\left(q_1\right) + MZ_2 gcos\left(q_1 + q_2\right) \\ MZ_2 gcos\left(q_1 + q_2\right) \end{bmatrix}}_{G(q)}$$

where we have defined the moments of inertia ZZ_1 and ZZ_2 according to *Huygens* theorem :

- $ZZ_1 = I_{zz_1} + m_1 L_{c_1}^2$;
- $ZZ_2 = I_{zz_2} + m_2 L_{c_2}^2$.

Rewritting the dynamic model of rigid-link robot

Factorization from the set of parameters : ZZ_1, ZZ_2, MZ_2, $m_2L_1^2$, MZ_1, m_2L_1, $r_{red_1}^2 J_{m_1}$ et $r_{red_2}^2 J_{m_2}$

Rewritting the dynamic model of rigid-link robot

Factorization from the set of parameters : ZZ_1, ZZ_2, MZ_2, $m_2L_1^2$, MZ_1, m_2L_1, $r_{red_1}^2 J_{m_1}$ et $r_{red_2}^2 J_{m_2}$

$$\underbrace{\begin{bmatrix} \ddot{q}_1 \\ 0 \end{bmatrix} \left(r_{red_1}^2 J_{m_1} + ZZ_1 + m_2 L_1^2\right) + \begin{bmatrix} \ddot{q}_1 + \ddot{q}_2 \\ \ddot{q}_1 + \ddot{q}_2 \end{bmatrix} ZZ_2 + \begin{bmatrix} 0 \\ \ddot{q}_2 \end{bmatrix} \left(r_{red_2}^2 J_{m_2}\right) + \begin{bmatrix} 2L_1 \cos\left(q_2\right) \ddot{q}_1 + L_1 \cos\left(q_2\right) \ddot{q}_2 \\ L_1 \cos\left(q_2\right) \ddot{q}_1 \end{bmatrix} MZ_2}_{A(q)\ddot{q}}$$

$$\ldots + \underbrace{\begin{bmatrix} -L_1 \sin\left(q_2\right) \left(2\dot{q}_2 \dot{q}_1 + \dot{q}_2^2\right) \\ L_1 \sin\left(q_2\right) \dot{q}_1^2 \end{bmatrix} MZ_2}_{C(q,\dot{q})\dot{q}}$$

$$\ldots + \underbrace{\begin{bmatrix} g\cos\left(q_1\right) \\ 0 \end{bmatrix} \left(MZ_1 + m_2 L_1\right) + \begin{bmatrix} g\cos\left(q_1 + q_2\right) \\ g\cos\left(q_1 + q_2\right) \end{bmatrix} MZ_2}_{G(q)} = \underbrace{\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}}_{\Gamma}$$

Rewritting the dynamic model of rigid-link robot

Factorization from the set of parameters : ZZ_1, ZZ_2, MZ_2, $m_2L_1^2$, MZ_1, m_2L_1, $r_{red_1}^2 J_{m_1}$ et $r_{red_2}^2 J_{m_2}$

$$\underbrace{\begin{bmatrix} \ddot{q}_1 \\ 0 \end{bmatrix} \left(r_{red_1}^2 J_{m_1} + ZZ_1 + m_2 L_1^2\right) + \begin{bmatrix} \ddot{q}_1 + \ddot{q}_2 \\ \ddot{q}_1 + \ddot{q}_2 \end{bmatrix} ZZ_2 + \begin{bmatrix} 0 \\ \ddot{q}_2 \end{bmatrix} \left(r_{red_2}^2 J_{m_2}\right) + \begin{bmatrix} 2L_1 \cos(q_2)\, \ddot{q}_1 + L_1 \cos(q_2)\, \ddot{q}_2 \\ L_1 \cos(q_2)\, \ddot{q}_1 \end{bmatrix} MZ_2}_{A(q)\ddot{q}}$$

$$\ldots + \underbrace{\begin{bmatrix} -L_1 \sin(q_2) \left(2\dot{q}_2\dot{q}_1 + \dot{q}_2^2\right) \\ L_1 \sin(q_2)\, \dot{q}_1^2 \end{bmatrix} MZ_2}_{C(q,\dot{q})\dot{q}}$$

$$\ldots + \underbrace{\begin{bmatrix} g\cos(q_1) \\ 0 \end{bmatrix} (MZ_1 + m_2 L_1) + \begin{bmatrix} g\cos(q_1 + q_2) \\ g\cos(q_1 + q_2) \end{bmatrix} MZ_2}_{G(q)} = \underbrace{\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}}_{\Gamma}$$

By grouping the terms post-multiplied by MZ_2, it comes :

Rewritting the dynamic model of rigid-link robot

Factorization from the set of parameters : ZZ_1, ZZ_2, MZ_2, $m_2 L_1^2$, MZ_1, $m_2 L_1$, $r_{red_1}^2 J_{m_1}$ et $r_{red_2}^2 J_{m_2}$

$$\underbrace{\begin{bmatrix} \ddot{q}_1 \\ 0 \end{bmatrix} \left(r_{red_1}^2 J_{m_1} + ZZ_1 + m_2 L_1^2 \right) + \begin{bmatrix} \ddot{q}_1 + \ddot{q}_2 \\ \ddot{q}_1 + \ddot{q}_2 \end{bmatrix} ZZ_2 + \begin{bmatrix} 0 \\ \ddot{q}_2 \end{bmatrix} \left(r_{red_2}^2 J_{m_2} \right) + \begin{bmatrix} 2L_1 \cos(q_2)\, \ddot{q}_1 + L_1 \cos(q_2)\, \ddot{q}_2 \\ L_1 \cos(q_2)\, \ddot{q}_1 \end{bmatrix} MZ_2}_{A(q)\ddot{q}}$$

$$\ldots + \underbrace{\begin{bmatrix} -L_1 \sin(q_2) \left(2\dot{q}_2 \dot{q}_1 + \dot{q}_2^2 \right) \\ L_1 \sin(q_2)\, \dot{q}_1^2 \end{bmatrix} MZ_2}_{C(q,\dot{q})\dot{q}}$$

$$\ldots + \underbrace{\begin{bmatrix} g\cos(q_1) \\ 0 \end{bmatrix} (MZ_1 + m_2 L_1) + \begin{bmatrix} g\cos(q_1 + q_2) \\ g\cos(q_1 + q_2) \end{bmatrix} MZ_2}_{G(q)} = \underbrace{\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}}_{\Gamma}$$

By grouping the terms post-multiplied by MZ_2, it comes :
$\Gamma = W(q, \dot{q}, \ddot{q})\, \chi$ avec

Rewritting the dynamic model of rigid-link robot

Factorization from the set of parameters : ZZ_1, ZZ_2, MZ_2, $m_2 L_1^2$, MZ_1, $m_2 L_1$, $r_{red_1}^2 J_{m_1}$ et $r_{red_2}^2 J_{m_2}$

$$\underbrace{\begin{bmatrix} \ddot{q}_1 \\ 0 \end{bmatrix} \left(r_{red_1}^2 J_{m_1} + ZZ_1 + m_2 L_1^2 \right) + \begin{bmatrix} \ddot{q}_1 + \ddot{q}_2 \\ \ddot{q}_1 + \ddot{q}_2 \end{bmatrix} ZZ_2 + \begin{bmatrix} 0 \\ \ddot{q}_2 \end{bmatrix} \left(r_{red_2}^2 J_{m_2} \right) + \begin{bmatrix} 2L_1 \cos\left(q_2\right) \ddot{q}_1 + L_1 \cos\left(q_2\right) \ddot{q}_2 \\ L_1 \cos\left(q_2\right) \ddot{q}_1 \end{bmatrix} MZ_2}_{A(q)\ddot{q}}$$

$$\ldots + \underbrace{\begin{bmatrix} -L_1 \sin\left(q_2\right) \left(2\dot{q}_2 \dot{q}_1 + \dot{q}_2^2\right) \\ L_1 \sin\left(q_2\right) \dot{q}_1^2 \end{bmatrix} MZ_2}_{C(q,\dot{q})\dot{q}}$$

$$\ldots + \underbrace{\begin{bmatrix} g\cos\left(q_1\right) \\ 0 \end{bmatrix} \left(MZ_1 + m_2 L_1\right) + \begin{bmatrix} g\cos\left(q_1 + q_2\right) \\ g\cos\left(q_1 + q_2\right) \end{bmatrix} MZ_2}_{G(q)} = \underbrace{\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}}_{\Gamma}$$

By grouping the terms post-multiplied by MZ_2, it comes :
$\Gamma = W\left(q, \dot{q}, \ddot{q}\right) \chi$ avec
$\chi = \begin{bmatrix} ZZ_1 + m_2 L_1^2 + r_{red_1}^2 J_{m_1} & ZZ2 & MZ2 & MZ_1 + m_2 L_1 & r_{red_2}^2 J_{m_2} \end{bmatrix}^t$

Rewritting the dynamic model of rigid-link robot

Factorization from the set of parameters : ZZ_1, ZZ_2, MZ_2, $m_2 L_1^2$, MZ_1, $m_2 L_1$, $r_{red_1}^2 J_{m_1}$ et $r_{red_2}^2 J_{m_2}$

$$\underbrace{\begin{bmatrix} \ddot{q}_1 \\ 0 \end{bmatrix} \left(r_{red_1}^2 J_{m_1} + ZZ_1 + m_2 L_1^2\right) + \begin{bmatrix} \ddot{q}_1 + \ddot{q}_2 \\ \ddot{q}_1 + \ddot{q}_2 \end{bmatrix} ZZ_2 + \begin{bmatrix} 0 \\ \ddot{q}_2 \end{bmatrix} \left(r_{red_2}^2 J_{m_2}\right) + \begin{bmatrix} 2L_1 \cos(q_2)\, \ddot{q}_1 + L_1 \cos(q_2)\, \ddot{q}_2 \\ L_1 \cos(q_2)\, \ddot{q}_1 \end{bmatrix} MZ_2}_{A(q)\ddot{q}}$$

$$\ldots + \underbrace{\begin{bmatrix} -L_1 \sin(q_2) \left(2\dot{q}_2 \dot{q}_1 + \dot{q}_2^2\right) \\ L_1 \sin(q_2)\, \dot{q}_1^2 \end{bmatrix} MZ_2}_{C(q,\dot{q})\dot{q}}$$

$$\ldots + \underbrace{\begin{bmatrix} g\cos(q_1) \\ 0 \end{bmatrix} (MZ_1 + m_2 L_1) + \begin{bmatrix} g\cos(q_1 + q_2) \\ g\cos(q_1 + q_2) \end{bmatrix} MZ_2}_{G(q)} = \underbrace{\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}}_{\Gamma}$$

By grouping the terms post-multiplied by MZ_2, it comes :
$\Gamma = W(q, \dot{q}, \ddot{q})\, \chi$ avec
$\chi = \begin{bmatrix} ZZ_1 + m_2 L_1^2 + r_{red_1}^2 J_{m_1} & ZZ2 & MZ2 & MZ_1 + m_2 L_1 & r_{red_2}^2 J_{m_2} \end{bmatrix}^t$
and

Rewritting the dynamic model of rigid-link robot

Factorization from the set of parameters : ZZ_1, ZZ_2, MZ_2, $m_2L_1^2$, MZ_1, m_2L_1, $r_{red_1}^2 J_{m_1}$ et $r_{red_2}^2 J_{m_2}$

$$\underbrace{\begin{bmatrix} \ddot{q}_1 \\ 0 \end{bmatrix}\left(r_{red_1}^2 J_{m_1} + ZZ_1 + m_2L_1^2\right) + \begin{bmatrix} \ddot{q}_1 + \ddot{q}_2 \\ \ddot{q}_1 + \ddot{q}_2 \end{bmatrix} ZZ_2 + \begin{bmatrix} 0 \\ \ddot{q}_2 \end{bmatrix}\left(r_{red_2}^2 J_{m_2}\right) + \begin{bmatrix} 2L_1 cos\,(q_2)\,\ddot{q}_1 + L_1 cos\,(q_2)\,\ddot{q}_2 \\ L_1 cos\,(q_2)\,\ddot{q}_1 \end{bmatrix} MZ_2}_{A(q)\ddot{q}}$$

$$\ldots + \underbrace{\begin{bmatrix} -L_1 sin\,(q_2)\left(2\dot{q}_2\dot{q}_1 + \dot{q}_2^2\right) \\ L_1 sin\,(q_2)\,\dot{q}_1^2 \end{bmatrix} MZ_2}_{C(q,\dot{q})\dot{q}}$$

$$\ldots + \underbrace{\begin{bmatrix} gcos\,(q_1) \\ 0 \end{bmatrix}(MZ_1 + m_2L_1) + \begin{bmatrix} gcos\,(q_1 + q_2) \\ gcos\,(q_1 + q_2) \end{bmatrix} MZ_2}_{G(q)} = \underbrace{\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}}_{\Gamma}$$

By grouping the terms post-multiplied by MZ_2, it comes :
$\Gamma = W\,(q, \dot{q}, \ddot{q})\,\chi$ avec

$$\chi = \begin{bmatrix} ZZ_1 + m_2L_1^2 + r_{red_1}^2 J_{m_1} & ZZ2 & MZ2 & MZ_1 + m_2L_1 & r_{red_2}^2 J_{m_2} \end{bmatrix}^t$$

and

$$W\,(q, \dot{q}, \ddot{q}) =$$

$$\begin{bmatrix} \ddot{q}_1 & \ddot{q}_1 + \ddot{q}_2 & L_1 cos\,(q_2)\,(2\ddot{q}_1 + \ddot{q}_2) - L_1 sin\,(q_2)\left(2\dot{q}_2\dot{q}_1 + \dot{q}_2^2\right) + gcos\,(q_1 + q_2) & gcos\,(q_1) & 0 \\ 0 & \ddot{q}_1 + \ddot{q}_2 & L_1 cos\,(q_2)\,\ddot{q}_1 + L_1 sin\,(q_2)\,\dot{q}_1^2 + gcos\,(q_1 + q_2) & 0 & \ddot{q}_2 \end{bmatrix}$$

▸ Course

Solution of the optimisation problem

$\hat{\chi}$ as solution of $\mathrm{Min}_{\chi} = \left\|\tilde{\Gamma} - \tilde{W}\chi\right\|_2$

Solution of the optimisation problem

$\hat{\chi}$ as solution of $\mathrm{Min}_{\chi} = \left\| \tilde{\Gamma} - \tilde{W}\chi \right\|_2$

Searching for $\left.\frac{\partial J}{\partial \chi}\right)_{\chi=\hat{\chi}} = 0$ where $J = \left(\tilde{\Gamma} - \tilde{W}\chi\right)^t \left(\tilde{\Gamma} - \tilde{W}\chi\right)$

Solution of the optimisation problem

$\hat{\chi}$ as solution of $\text{Min}_{\chi} = \left\| \tilde{\Gamma} - \tilde{W}\chi \right\|_2$

Searching for $\left.\frac{\partial J}{\partial \chi}\right)_{\chi=\hat{\chi}} = 0$ where $J = \left(\tilde{\Gamma} - \tilde{W}\chi\right)^t \left(\tilde{\Gamma} - \tilde{W}\chi\right)$

Developing the criterion expression :

$$\begin{aligned} J &= \left(\tilde{\Gamma} - \tilde{W}\chi\right)^t \left(\tilde{\Gamma} - \tilde{W}\chi\right) \\ &= \left(\tilde{\Gamma}^t - \chi^t \tilde{W}^t\right) \left(\tilde{\Gamma} - \tilde{W}\chi\right) \\ &= \tilde{\Gamma}^t\tilde{\Gamma} - \tilde{\Gamma}^t\tilde{W}\chi - \chi^t\tilde{W}^t\tilde{\Gamma} + \chi^t\tilde{W}^t\tilde{W}\chi \end{aligned}$$

Solution of the optimisation problem

$\hat{\chi}$ as solution of $\text{Min}_{\chi} = \left\| \tilde{\Gamma} - \tilde{W}\chi \right\|_2$

Searching for $\left.\frac{\partial J}{\partial \chi}\right)_{\chi=\hat{\chi}} = 0$ where $J = \left(\tilde{\Gamma} - \tilde{W}\chi\right)^t \left(\tilde{\Gamma} - \tilde{W}\chi\right)$

Developing the criterion expression :

$$\begin{aligned} J &= \left(\tilde{\Gamma} - \tilde{W}\chi\right)^t \left(\tilde{\Gamma} - \tilde{W}\chi\right) \\ &= \left(\tilde{\Gamma}^t - \chi^t \tilde{W}^t\right) \left(\tilde{\Gamma} - \tilde{W}\chi\right) \\ &= \tilde{\Gamma}^t\tilde{\Gamma} - \tilde{\Gamma}^t\tilde{W}\chi - \chi^t\tilde{W}^t\tilde{\Gamma} + \chi^t\tilde{W}^t\tilde{W}\chi \end{aligned}$$

Formulary for derivating linear functions

$$\frac{\partial a^t x}{\partial x} = \frac{\partial x^t a}{\partial x} = a^t$$
$$\frac{\partial A x}{\partial x} = \frac{\partial x^t A}{\partial x^t} = A$$

Derivative of quadratic functions

$$\frac{\partial x^t A x}{\partial x} = x^t \left(A^t + A\right)$$

Solution of the optimisation problem

$\hat{\chi}$ as solution of $\text{Min}_{\chi} = \left\|\tilde{\Gamma} - \tilde{W}\chi\right\|_2$

Searching for $\left.\frac{\partial J}{\partial \chi}\right)_{\chi=\hat{\chi}} = 0$ where $J = \left(\tilde{\Gamma} - \tilde{W}\chi\right)^t \left(\tilde{\Gamma} - \tilde{W}\chi\right)$

Developing the criterion expression :

$$\begin{aligned} J &= \left(\tilde{\Gamma} - \tilde{W}\chi\right)^t \left(\tilde{\Gamma} - \tilde{W}\chi\right) \\ &= \left(\tilde{\Gamma}^t - \chi^t\tilde{W}^t\right)\left(\tilde{\Gamma} - \tilde{W}\chi\right) \\ &= \tilde{\Gamma}^t\tilde{\Gamma} - \tilde{\Gamma}^t\tilde{W}\chi - \chi^t\tilde{W}^t\tilde{\Gamma} + \chi^t\tilde{W}^t\tilde{W}\chi \end{aligned}$$

Formulary for derivating linear functions

$$\frac{\partial a^t x}{\partial x} = \frac{\partial x^t a}{\partial x} = a^t$$

$$\frac{\partial Ax}{\partial x} = \frac{\partial x^t A}{\partial x^t} = A$$

Derivative of quadratic functions

$$\frac{\partial x^t A x}{\partial x} = x^t\left(A^t + A\right)$$

Calculation of the partial derivative

$$\begin{aligned} \frac{\partial J}{\partial \chi} &= -\tilde{\Gamma}^t\tilde{W} - \left(\tilde{W}^t\tilde{\Gamma}\right)^t + \chi^t\left(\left(\tilde{W}^t\tilde{W}\right)^t + \tilde{W}^t\tilde{W}\right) \\ &= -2\tilde{\Gamma}^t\tilde{W} + 2\chi^t\tilde{W}^t\tilde{W} \end{aligned}$$

Solution of the optimisation problem

$\hat{\chi}$ as solution of $\text{Min}_\chi = \left\| \tilde{\Gamma} - \tilde{W}\chi \right\|_2$

Searching for $\left.\frac{\partial J}{\partial \chi}\right)_{\chi=\hat{\chi}} = 0$ where $J = \left(\tilde{\Gamma} - \tilde{W}\chi\right)^t \left(\tilde{\Gamma} - \tilde{W}\chi\right)$

Developing the criterion expression :

$$\begin{aligned} J &= \left(\tilde{\Gamma} - \tilde{W}\chi\right)^t \left(\tilde{\Gamma} - \tilde{W}\chi\right) \\ &= \left(\tilde{\Gamma}^t - \chi^t\tilde{W}^t\right)\left(\tilde{\Gamma} - \tilde{W}\chi\right) \\ &= \tilde{\Gamma}^t\tilde{\Gamma} - \tilde{\Gamma}^t\tilde{W}\chi - \chi^t\tilde{W}^t\tilde{\Gamma} + \chi^t\tilde{W}^t\tilde{W}\chi \end{aligned}$$

Formulary for derivating linear functions

$$\frac{\partial a^t x}{\partial x} = \frac{\partial x^t a}{\partial x} = a^t$$

$$\frac{\partial Ax}{\partial x} = \frac{\partial x^t A}{\partial x^t} = A$$

Derivative of quadratic functions

$$\frac{\partial x^t A x}{\partial x} = x^t\left(A^t + A\right)$$

Calculation of the partial derivative

$$\begin{aligned} \frac{\partial J}{\partial \chi} &= -\tilde{\Gamma}^t\tilde{W} - \left(\tilde{W}^t\tilde{\Gamma}\right)^t + \chi^t\left(\left(\tilde{W}^t\tilde{W}\right)^t + \tilde{W}^t\tilde{W}\right) \\ &= -2\tilde{\Gamma}^t\tilde{W} + 2\chi^t\tilde{W}^t\tilde{W} \end{aligned}$$

$$\text{Thus } \left.\frac{\partial J}{\partial \chi}\right)_{\chi=\hat{\chi}} = 0 \Leftrightarrow \quad \hat{\chi}^t\tilde{W}^t\tilde{W} = \tilde{\Gamma}^t\tilde{W}$$

$$\Leftrightarrow \quad \tilde{W}^t\tilde{W}\hat{\chi} = \tilde{W}^t\tilde{\Gamma}$$

$$\Leftrightarrow \quad \hat{\chi} = \underbrace{\left(\tilde{W}^t\tilde{W}\right)^{-1}\tilde{W}^t}_{\tilde{W}^+}\tilde{\Gamma}$$

Variance-covariance matrix of the estimation error

$$
\begin{aligned}
C_{\hat{\chi}} &= E\left[(\chi - \hat{\chi})(\chi - \hat{\chi})^t\right] \\
&= E\left[\left(\chi - \tilde{W}^+ \tilde{\mathrm{r}}\right)\left(\chi - \tilde{W}^+ \tilde{\mathrm{r}}\right)^t\right] \\
&= E\left[\left(\chi - \tilde{W}^+\left(\tilde{W}\chi + \rho\right)\right)\left(\chi - \tilde{W}^+\left(\tilde{W}\chi + \rho\right)\right)^t\right] \\
&= E\left[\left(\chi - \underbrace{\tilde{W}^+\tilde{W}}_{\mathbb{I}}\chi - \tilde{W}^+\rho\right)\left(\chi - \underbrace{\tilde{W}^+\tilde{W}}_{\mathbb{I}}\chi - \tilde{W}^+\rho\right)^t\right] \\
&= E\left[\tilde{W}^+\rho\rho^t\tilde{W}^{+^t}\right] \\
&= \tilde{W}^+ E\left[\rho\rho^t\right]\tilde{W}^{+^t} \\
&= \tilde{W}^+ C_\rho \tilde{W}^{+^t} \\
&= \sigma_\rho^2 \tilde{W}^+ \tilde{W}^{+^t} \\
&= \sigma_\rho^2 \left[\left(\tilde{W}^t\tilde{W}\right)^{-1}\tilde{W}^t\right]\left[\left(\tilde{W}^t\tilde{W}\right)^{-1}\tilde{W}^t\right]^t \\
&= \sigma_\rho^2 \left(\tilde{W}^t\tilde{W}\right)^{-1}\left(\tilde{W}^t\tilde{W}\right)\left(\left(\tilde{W}^t\tilde{W}\right)^{-1}\right)^t
\end{aligned}
$$

Variance-covariance matrix of the estimation error

$$
\begin{aligned}
C_{\hat{\chi}} &= E\left[(\chi - \hat{\chi})(\chi - \hat{\chi})^t\right] \\
&= E\left[\left(\chi - \tilde{W}^+\tilde{\Gamma}\right)\left(\chi - \tilde{W}^+\tilde{\Gamma}\right)^t\right] \\
&= E\left[\left(\chi - \tilde{W}^+\left(\tilde{W}\chi + \rho\right)\right)\left(\chi - \tilde{W}^+\left(\tilde{W}\chi + \rho\right)\right)^t\right] \\
&= E\left[\left(\chi - \underbrace{\tilde{W}^+\tilde{W}}_{\mathbb{I}}\chi - \tilde{W}^+\rho\right)\left(\chi - \underbrace{\tilde{W}^+\tilde{W}}_{\mathbb{I}}\chi - \tilde{W}^+\rho\right)^t\right] \\
&= E\left[\tilde{W}^+\rho\rho^t\tilde{W}^{+^t}\right] \\
&= \tilde{W}^+E\left[\rho\rho^t\right]\tilde{W}^{+^t} \\
&= \tilde{W}^+C_\rho\tilde{W}^{+^t} \\
&= \sigma_\rho^2\tilde{W}^+\tilde{W}^{+^t} \\
&= \sigma_\rho^2\left[\left(\tilde{W}^t\tilde{W}\right)^{-1}\tilde{W}^t\right]\left[\left(\tilde{W}^t\tilde{W}\right)^{-1}\tilde{W}^t\right]^t \\
&= \sigma_\rho^2\left(\tilde{W}^t\tilde{W}\right)^{-1}\left(\tilde{W}^t\tilde{W}\right)\left(\left(\tilde{W}^t\tilde{W}\right)^{-1}\right)^t
\end{aligned}
$$

$$
\begin{aligned}
C_{\hat{\chi}} &= \dots \\
&= \sigma_\rho^2\left(\left(\tilde{W}^t\tilde{W}\right)^{-1}\right)^t \\
&= \sigma_\rho^2\left(\left(\tilde{W}^t\tilde{W}\right)^t\right)^{-1} \\
&= \sigma_\rho^2\left(\tilde{W}^t\tilde{W}\right)^{-1}
\end{aligned}
$$

▸ Course

Third-order polynomial interpolation

Timing law :

$$q(t) = q_0 + r(t)D = q_0 + \left[a\left(\frac{t}{t_f}\right) + b\left(\frac{t}{t_f}\right)^2 + c\left(\frac{t}{t_f}\right)^3\right](q_f - q_0)$$

Third-order polynomial interpolation

Timing law :

$$q(t) = q_0 + r(t)D = q_0 + \left[a\left(\frac{t}{t_f}\right) + b\left(\frac{t}{t_f}\right)^2 + c\left(\frac{t}{t_f}\right)^3\right](q_f - q_0)$$

(we check that r(0)=0).

Third-order polynomial interpolation

Timing law :

$$q(t) = q_0 + r(t)D = q_0 + \left[a\left(\frac{t}{t_f}\right) + b\left(\frac{t}{t_f}\right)^2 + c\left(\frac{t}{t_f}\right)^3 \right](q_f - q_0)$$

(we check that r(0)=0).
Searching for the coefficients a, b and c :

Third-order polynomial interpolation

Timing law :

$$q(t) = q_0 + r(t)D = q_0 + \left[a\left(\frac{t}{t_f}\right) + b\left(\frac{t}{t_f}\right)^2 + c\left(\frac{t}{t_f}\right)^3 \right](q_f - q_0)$$

(we check that r(0)=0).
Searching for the coefficients a, b and c :

- Condition 1 :

$$r(t_f) = 1 \Leftrightarrow a + b + c = 1$$

Third-order polynomial interpolation

Timing law :

$$q(t) = q_0 + r(t)D = q_0 + \left[a\left(\frac{t}{t_f}\right) + b\left(\frac{t}{t_f}\right)^2 + c\left(\frac{t}{t_f}\right)^3\right](q_f - q_0)$$

(we check that r(0)=0).
Searching for the coefficients a, b and c :

- Condition 1 :

$$r(t_f) = 1 \Leftrightarrow a + b + c = 1$$

- Condition 2 :

$$\begin{aligned}
\dot{q}(0) = \dot{q}_0 \Leftrightarrow & \quad \dot{r}(0)D = \dot{q}_0 \\
\Leftrightarrow & \quad \left[\frac{a}{t_f} + \frac{2b}{t_f^2}t + \frac{3c}{t_f^3}t^2\right]_{t=0}(q_f - q_0) = \dot{q}_0 \\
\Leftrightarrow & \quad a\frac{(q_f - q_0)}{t_f} = \dot{q}_0
\end{aligned}$$

Third-order polynomial interpolation

- Condition 3 :

Third-order polynomial interpolation

- Condition 3 :

$$\begin{aligned}\dot{q}(t_f) = \dot{q}_f \Leftrightarrow & \quad \dot{r}(t_f) D = \dot{q}_f \\ \Leftrightarrow & \quad a\frac{(q_f - q_0)}{t_f} + b\frac{2\,(q_f - q_0)}{t_f} + c\,\frac{3\,(q_f - q_0)}{t_f} = \dot{q}_f\end{aligned}$$

Third-order polynomial interpolation

- Condition 3 :

$$\begin{aligned}\dot{q}(t_f) = \dot{q}_f \Leftrightarrow && \dot{r}(t_f)D = \dot{q}_f \\ \Leftrightarrow && a\frac{(q_f - q_0)}{t_f} + b\frac{2\,(q_f - q_0)}{t_f} + c\frac{3\,(q_f - q_0)}{t_f} = \dot{q}_f\end{aligned}$$

Writting under the form of a linear system :

$$AX = B \Leftrightarrow \begin{bmatrix} 1 & 1 & 1 \\ \frac{1}{t_f} & 0 & 0 \\ \frac{1}{t_f} & \frac{2}{t_f} & \frac{3}{t_f} \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 1 \\ \frac{\dot{q}_0}{(q_f - q_0)} \\ \frac{\dot{q}_f}{(q_f - q_0)} \end{bmatrix}$$

Third-order polynomial interpolation

- Condition 3 :

$$\dot{q}(t_f) = \dot{q}_f \Leftrightarrow \qquad \dot{r}(t_f)D = \dot{q}_f$$

$$\Leftrightarrow \qquad a\frac{(q_f - q_0)}{t_f} + b\frac{2\,(q_f - q_0)}{t_f} + c\frac{3\,(q_f - q_0)}{t_f} = \dot{q}_f$$

Writting under the form of a linear system :

$$AX = B \Leftrightarrow \begin{bmatrix} 1 & 1 & 1 \\ \frac{1}{t_f} & 0 & 0 \\ \frac{1}{t_f} & \frac{2}{t_f} & \frac{3}{t_f} \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 1 \\ \frac{\dot{q}_0}{(q_f - q_0)} \\ \frac{\dot{q}_f}{(q_f - q_0)} \end{bmatrix}$$

We verify that the system is invertible ($det\,(A) = -\frac{1}{t_f^2}$) and the inverse of A is equal to :

$$A^{-1} = \begin{bmatrix} 0 & t_f & 0 \\ 3 & -2t_f & -t_f \\ -2 & t_f & t_f \end{bmatrix}$$

Third-order polynomial interpolation

- Condition 3 :

$$\dot{q}(t_f) = \dot{q}_f \Leftrightarrow \qquad \dot{r}(t_f)D = \dot{q}_f$$
$$\Leftrightarrow \qquad a\frac{(q_f - q_0)}{t_f} + b\frac{2\,(q_f - q_0)}{t_f} + c\frac{3\,(q_f - q_0)}{t_f} = \dot{q}_f$$

Writting under the form of a linear system :

$$AX = B \Leftrightarrow \begin{bmatrix} 1 & 1 & 1 \\ \frac{1}{t_f} & 0 & 0 \\ \frac{1}{t_f} & \frac{2}{t_f} & \frac{3}{t_f} \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 1 \\ \frac{\dot{q}_0}{(q_f - q_0)} \\ \frac{\dot{q}_f}{(q_f - q_0)} \end{bmatrix}$$

We verify that the system is invertible ($det\,(A) = -\frac{1}{t_f^2}$) and the inverse of A is equal to :

$$A^{-1} = \begin{bmatrix} 0 & t_f & 0 \\ 3 & -2t_f & -t_f \\ -2 & t_f & t_f \end{bmatrix}$$

Thus, the coefficients of the polynomial are given by :

$$\begin{cases} a = t_f\,\frac{\dot{q}_0}{(q_f - q_0)} \\ b = 3 - 2t_f\,\frac{\dot{q}_0}{(q_f - q_0)} - t_f\,\frac{\dot{q}_f}{(q_f - q_0)} \\ c = -2 + t_f\,\frac{\dot{q}_0}{(q_f - q_0)} + t_f\,\frac{\dot{q}_f}{(q_f - q_0)} \end{cases}$$

Bang-bang profile with constant velocity phase

Condition of the existence of a constant velocity phase :

$$\underbrace{\frac{|D_i|}{2}}_{\substack{\text{Distance to travel}\\ \text{during the acceleration phase}}} > \underbrace{|d_i|}_{\substack{\text{Distance already travelled}\\ \text{at the velocity } \dot{q}_i \\ \text{until } T_i \text{ before saturation}}}$$

Bang-bang profile with constant velocity phase

Condition of the existence of a constant velocity phase :

$$\underbrace{\frac{|D_i|}{2}}_{\substack{\text{Distance to travel} \\ \text{during the acceleration phase}}} > \underbrace{|d_i|}_{\substack{\text{Distance already travelled} \\ \text{at the velocity } \dot{q}_i \\ \text{until } T_i \text{ before saturation}}}$$

- Search for the instant T_i :

Bang-bang profile with constant velocity phase

Condition of the existence of a constant velocity phase :

$$\underbrace{\frac{|D_i|}{2}}_{\substack{\text{Distance to travel} \\ \text{during the acceleration phase}}} > \underbrace{|d_i|}_{\substack{\text{Distance already travelled} \\ \text{at the velocity } \dot{q}_i \\ \text{until } T_i \text{ before saturation}}}$$

- Search for the instant T_i :
  - on the one hand, for $0 \leq t \leq T_i$, the joint velocity follows the timing law :

$$\dot{q}_i = k_{a_i} t$$

Bang-bang profile with constant velocity phase

Condition of the existence of a constant velocity phase :

$$\underbrace{\frac{|D_i|}{2}}_{\substack{\text{Distance to travel}\\ \text{during the acceleration phase}}} > \underbrace{|d_i|}_{\substack{\text{Distance already travelled}\\ \text{at the velocity } \dot{q}_i \\ \text{until } T_i \text{ before saturation}}}$$

- Search for the instant T_i :
  - on the one hand, for $0 \leq t \leq T_i$, the joint velocity follows the timing law :

$$\dot{q}_i = k_{a_i} t$$

  - on the other hand, for $T_i \leq t \leq t_{f_i} - T_i$, the joint velocity is constant and equal to its saturation velocity :

$$\dot{q}_i = k_{v_i}$$

Bang-bang profile with constant velocity phase

Condition of the existence of a constant velocity phase :

$$\underbrace{\frac{|D_i|}{2}}_{\substack{\text{Distance to travel}\\ \text{during the acceleration phase}}} > \underbrace{|d_i|}_{\substack{\text{Distance already travelled}\\ \text{at the velocity } \dot{q}_i\\ \text{until } T_i \text{ before saturation}}}$$

- Search for the instant T_i :
  - on the one hand, for $0 \leq t \leq T_i$, the joint velocity follows the timing law :

    $$\dot{q}_i = k_{a_i} t$$

  - on the other hand, for $T_i \leq t \leq t_{f_i} - T_i$, the joint velocity is constant and equal to its saturation velocity :

    $$\dot{q}_i = k_{v_i}$$

  - Thus, the moment T_i is deduced from the above two equations that are evaluated at this same moment :

    $$T_i = \frac{k_{v_i}}{k_{a_i}}$$

Bang-bang profile with constant velocity phase

- Searching for the travelled distance until T_i :

Bang-bang profile with constant velocity phase

- Searching for the travelled distance until T_i :
  - the acceleration being constant and the velocity being null at the beginning, the travelled distance by joint i during the time T_i is :

$$\frac{1}{2} k_{a_i} T_i^2$$

Bang-bang profile with constant velocity phase

- Searching for the travelled distance until T_i :
  - the acceleration being constant and the velocity being null at the beginning, the travelled distance by joint i during the time T_i is :

$$\frac{1}{2} k_{a_i} T_i^2$$

  - or, by replacing T_i with the previous expression :

$$d_i = \int_0^{T_i} \dot{q}_i dt = \frac{1}{2} k_{a_i} T_i^2 = \frac{1}{2} k_{a_i} \left(\frac{k_{v_i}}{k_{a_i}} \right)^2 = \frac{k_{v_i}^2}{2 k_{a_i}}$$

Bang-bang profile with constant velocity phase

- Searching for the travelled distance until T_i :
  - the acceleration being constant and the velocity being null at the beginning, the travelled distance by joint i during the time T_i is :

$$\frac{1}{2} k_{a_i} T_i^2$$

  - or, by replacing T_i with the previous expression :

$$d_i = \int_0^{T_i} \dot{q}_i dt = \frac{1}{2} k_{a_i} T_i^2 = \frac{1}{2} k_{a_i} \left(\frac{k_{v_i}}{k_{a_i}} \right)^2 = \frac{k_{v_i}^2}{2k_{a_i}}$$

- Finally, the condition of the existence the constant velocity phase is :

$$\frac{|D_i|}{2} > \frac{k_{v_i}^2}{2k_{a_i}} \Leftrightarrow |D_i| > \frac{k_{v_i}^2}{k_{a_i}}$$

Course

Bang-bang profile with constant velocity phase

- Area of the trapeze under the velocity curve equals to :

Bang-bang profile with constant velocity phase

- Area of the trapeze under the velocity curve equals to :

$$\mathscr{A} = k_{v_i} \left(t_{f_i} - T_i \right)$$

where $T_i = \frac{k_{v_i}}{k_{a_i}}$ from the previous exercice.

Bang-bang profile with constant velocity phase

- Area of the trapeze under the velocity curve equals to :

$$\mathscr{A} = k_{v_i}\left(t_{f_i} - T_i\right)$$

where $T_i = \frac{k_{v_i}}{k_{a_i}}$ from the previous exercice.

- besides, the area of the trapeze under the velocity curve is equal to the travelled distance, since :

$$\mathscr{A} = \int_0^{t_{f_i}} \dot{q}_i dt = q_i\left(t_{f_i}\right) - q_i\left(t_0\right) = |D_i|$$

Bang-bang profile with constant velocity phase

- Area of the trapeze under the velocity curve equals to :

$$\mathscr{A} = k_{v_i}\left(t_{f_i} - T_i\right)$$

where $T_i = \frac{k_{v_i}}{k_{a_i}}$ from the previous exercice.

- besides, the area of the trapeze under the velocity curve is equal to the travelled distance, since :

$$\mathscr{A} = \int_0^{t_{f_i}} \dot{q}_i dt = q_i\left(t_{f_i}\right) - q_i\left(t_0\right) = |D_i|$$

- Finally, we deduce that :

$$k_{v_i}\left(t_{f_i} - T_i\right) = |D_i| \Leftrightarrow t_{f_i} = \frac{|D_i|}{k_{v_i}} + \underbrace{T_i}_{\frac{k_{v_i}}{k_{a_i}}}$$

Course

Stability in the sense of *Lyapunov*

Differential equation of the motion :

$$\underbrace{\left(m_i d_i^2 + I_{G_i} + r_{red_i}^2 J_{m_i}\right)}_{I_{eq}} \ddot{q}_i + f_{v_i}\dot{q}_i + m_i g d_i sin\left(q_i\right) = \tau_i,$$

Stability in the sense of *Lyapunov*

Differential equation of the motion :

$$\underbrace{\left(m_i d_i^2 + I_{G_i} + r_{red_i}^2 J_{m_i}\right)}_{I_{eq}} \ddot{q}_i + f_{v_i}\dot{q}_i + m_i g d_i sin\left(q_i\right) = \tau_i,$$

- Searching for the equilibrium state :
  At the equilibrium, $\ddot{q}_i = \dot{q}_i = 0$ and in the absence of control ($\tau_i = 0$ for the autonomous system) :

$$m_i g d_i sin\left(q_i\right) = 0.$$

Stability in the sense of *Lyapunov*

Differential equation of the motion :

$$\underbrace{\left(m_i d_i^2 + I_{G_i} + r_{red_i}^2 J_{m_i}\right)}_{I_{eq}} \ddot{q}_i + f_{v_i}\dot{q}_i + m_i g d_i sin\left(q_i\right) = \tau_i,$$

- Searching for the equilibrium state :
  At the equilibrium, $\ddot{q}_i = \dot{q}_i = 0$ and in the absence of control ($\tau_i = 0$ for the autonomous system) :

  $$m_i g d_i sin\left(q_i\right) = 0.$$

  The equilibrium points correspond to the cases : $q_{i_e} = n\pi$ with $n \in \mathbb{Z}$. In the considered interval, there is only one equilibrium point :

  $$q_{i_e} = 0.$$

Stability in the sense of *Lyapunov*

Differential equation of the motion :

$$\underbrace{\left(m_i d_i^2 + I_{G_i} + r_{red_i}^2 J_{m_i}\right)}_{I_{eq}} \ddot{q}_i + f_{v_i}\dot{q}_i + m_i g d_i sin\left(q_i\right) = \tau_i,$$

- Searching for the equilibrium state :
  At the equilibrium, $\ddot{q}_i = \dot{q}_i = 0$ and in the absence of control ($\tau_i = 0$ for the autonomous system) :

  $$m_i g d_i sin\left(q_i\right) = 0.$$

  The equilibrium points correspond to the cases : $q_{i_e} = n\pi$ with $n \in \mathbb{Z}$. In the considered interval, there is only one equilibrium point :

  $$q_{i_e} = 0.$$

- Proposition of a *Lyapunov* candidate function :

  $$V\left(x\right) = m_i g d_i \left(1 - cos\left(q_i\right)\right) + \frac{1}{2} I_{eq}\dot{q}_i^2,$$

  as the sum of the potential and the kinetic energies of the system (where we posed $x = \left[q_i, \dot{q}_i\right]^t$). Note that, locally (i.e. for $q_i \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$), $V\left(x\right) > 0$ except in $x = [0, 0]$ where $V\left(0\right) = 0$.

Stability in the sense of *Lyapunov*

- Calculation of the time-derivative of the proposed *Lyapunov* candidate function :

$$\begin{aligned}\dot{V}(x) &= m_i g d_i sin(q_i)\,\dot{q}_i + I_{eq}\dot{q}_i\ddot{q}_i\\ &= \dot{q}_i\left(m_i g d_i sin(q_i) + I_{eq}\ddot{q}_i\right)\\ &= -f_{v_i}\dot{q}_i^2\\ &\leq 0\end{aligned}$$

Thus, the equilibrium point $x_e = [q_{i_e} = 0, \dot{q}_{i_e} = 0]^t$ origin of the system is stable.

Course

Decentralized P.D. joint controller

- *Lyapunov* function :

Decentralized P.D. joint controller

- *Lyapunov* function : $V = \frac{1}{2}\dot{q}^t A(q)\,\dot{q} + \frac{1}{2}e^t K_p e$

Decentralized P.D. joint controller

- *Lyapunov* function : $V = \frac{1}{2}\dot{q}^t A(q)\,\dot{q} + \frac{1}{2}e^t K_p e$
  - let note that $e = q_d - q$ with q_d constant $\Rightarrow \dot{e} = -\dot{q}$
  - $V \geq 0$ and $V = 0 \Leftrightarrow e = \dot{e} = 0$

Decentralized P.D. joint controller

- *Lyapunov* function : $V = \frac{1}{2}\dot{q}^t A\left(q\right)\dot{q} + \frac{1}{2}e^t K_p e$
  - let note that $e = q_d - q$ with q_d constant $\Rightarrow \dot{e} = -\dot{q}$
  - $V \geq 0$ and $V = 0 \Leftrightarrow e = \dot{e} = 0$
- Stability proof

Decentralized P.D. joint controller

- *Lyapunov* function : $V = \frac{1}{2}\dot{q}^t A(q)\,\dot{q} + \frac{1}{2}e^t K_p e$
  - let note that $e = q_d - q$ with q_d constant $\Rightarrow \dot{e} = -\dot{q}$
  - $V \geq 0$ and $V = 0 \Leftrightarrow e = \dot{e} = 0$
- Stability proof

$$\begin{aligned}
\dot{V} &= \left(\frac{1}{2}\ddot{q}^t A(q)\,\dot{q} + \frac{1}{2}\dot{q}^t \dot{A}(q)\,\dot{q} + \frac{1}{2}\dot{q}^t A(q)\,\ddot{q}\right) + \left(\frac{1}{2}\dot{e}^t K_p e + \frac{1}{2}e^t K_p \dot{e}\right) \\
&= \left(\frac{1}{2}\dot{q}^t \dot{A}(q)\,\dot{q} + \dot{q}^t A(q)\,\ddot{q}\right) + \left(-\frac{1}{2}\dot{q}^t K_p e - \frac{1}{2}e^t K_p \dot{q}\right) \\
&= \left(\frac{1}{2}\dot{q}^t \dot{A}(q)\,\dot{q} + \dot{q}^t A(q)\,\ddot{q}\right) + \left(-\dot{q}^t K_p e\right) \\
&= \dot{q}^t \left(\frac{1}{2}\dot{A}(q)\,\dot{q} + \underbrace{A(q)\,\ddot{q}}_{=\Gamma - C(q,\dot{q})\dot{q}}\right) - \dot{q}^t K_p e \\
&= \dot{q}^t \left(\underbrace{\Gamma}_{=K_p e - K_d \dot{q}} - C(q,\dot{q})\,\dot{q} + \frac{1}{2}\dot{A}(q)\,\dot{q}\right) - \dot{q}^t K_p e \\
&= \left(\frac{1}{2}\dot{q}^t \left(\dot{A} - 2C\right)\dot{q} + \dot{q}^t K_p e - \dot{q}^t K_d \dot{q}\right) - \dot{q}^t K_p e
\end{aligned}$$

Decentralized P.D. joint controller

- Stability proof

$$
\begin{aligned}
&= \dots \\
&= \underbrace{\frac{1}{2}\dot{q}^t\left(\dot{A} - 2C\right)\dot{q}}_{=0 \text{ since } \left(\dot{A}-2C\right) \text{ is skew-sym..}} \qquad -\dot{q}^t K_d \dot{q} \\
&= -\dot{q}^t K_d \dot{q} \\
&\leq 0 \quad (\text{since } K_d = diag\left(k_{d_i}\right) > 0)
\end{aligned}
$$

Decentralized P.D. joint controller

- Stability proof

$$
\begin{aligned}
&= \dots \\
&= \underbrace{\frac{1}{2}\dot{q}^t\left(\dot{A}-2C\right)\dot{q}}_{=0 \text{ since } \left(\dot{A}-2C\right) \text{ is skew-sym..}} \quad -\dot{q}^t K_d \dot{q} \\
&= -\dot{q}^t K_d \dot{q} \\
&\leq 0 \ \ (\text{since } K_d = diag\left(k_{d_i}\right) > 0)
\end{aligned}
$$

  Thus, $\dot{V} \leq 0$, that proves the stability only.
- From LaSalle's theorem, the trajectories of the system converge asymptotically towards the largest invariant set of $S = \left\{x \in \mathbb{R}^n : \dot{V}\left(x\right) = 0\right\}$.

Decentralized P.D. joint controller

- Stability proof

$$\begin{aligned} &= \dots \\ &= \underbrace{\frac{1}{2}\dot{q}^t\left(\dot{A}-2C\right)\dot{q}}_{=0 \text{ since } \left(\dot{A}-2C\right) \text{ is skew-sym..}} \quad -\dot{q}^t K_d \dot{q} \\ &= -\dot{q}^t K_d \dot{q} \\ &\leq 0 \ \ (\text{since } K_d = diag\left(k_{d_i}\right) > 0) \end{aligned}$$

  Thus, $\dot{V} \leq 0$, that proves the stability only.
- From LaSalle's theorem, the trajectories of the system converge asymptotically towards the largest invariant set of $S = \left\{x \in \mathbb{R}^n : \dot{V}\left(x\right) = 0\right\}$.

  Yet $\dot{V}\left(x\right) = 0 \Leftrightarrow \dot{q} = 0$.

Decentralized P.D. joint controller

- Stability proof

$$\begin{aligned} &= \dots \\ &= \underbrace{\frac{1}{2}\dot{q}^t\left(\dot{A}-2C\right)\dot{q}}_{=0 \text{ since } \left(\dot{A}-2C\right) \text{ is skew-sym..}} \quad -\dot{q}^t K_d \dot{q} \\ &= -\dot{q}^t K_d \dot{q} \\ &\leq 0 \;\; (\text{since } K_d = diag\left(k_{d_i}\right) > 0) \end{aligned}$$

  Thus, $\dot{V} \leq 0$, that proves the stability only.

- From LaSalle's theorem, the trajectories of the system converge asymptotically towards the largest invariant set of $S = \left\{x \in \mathbb{R}^n : \dot{V}(x) = 0\right\}$.

  Yet $\dot{V}(x) = 0 \Leftrightarrow \dot{q} = 0$. The invariant set M is defined by $\dot{q} = 0$ (and $\ddot{q} = 0$ besides).

Decentralized P.D. joint controller

- Stability proof

$$
\begin{aligned}
&= \dots \\
&= \underbrace{\frac{1}{2}\dot{q}^t\left(\dot{A}-2C\right)\dot{q}}_{=0 \text{ since } \left(\dot{A}-2C\right) \text{ is skew-sym..}} \quad -\dot{q}^t K_d \dot{q} \\
&= -\dot{q}^t K_d \dot{q} \\
&\leq 0 \quad (\text{since } K_d = diag\left(k_{d_i}\right) > 0)
\end{aligned}
$$

  Thus, $\dot{V} \leq 0$, that proves the stability only.

- From LaSalle's theorem, the trajectories of the system converge asymptotically towards the largest invariant set of $S = \left\{x \in \mathbb{R}^n : \dot{V}(x) = 0\right\}$.

  Yet $\dot{V}(x) = 0 \Leftrightarrow \dot{q} = 0$. The invariant set M is defined by $\dot{q} = 0$ (and $\ddot{q} = 0$ besides). In this case , we note that the system dynamic in closed-loop is :

Decentralized P.D. joint controller

- Stability proof

$$
\begin{aligned}
&= \dots \\
&= \underbrace{\frac{1}{2}\dot{q}^t\left(\dot{A}-2C\right)\dot{q}}_{=0 \text{ since } \left(\dot{A}-2C\right) \text{ is skew-sym..}} \quad -\dot{q}^t K_d \dot{q} \\
&= -\dot{q}^t K_d \dot{q} \\
&\leq 0 \ \ (\text{since } K_d = diag\left(k_{d_i}\right) > 0)
\end{aligned}
$$

  Thus, $\dot{V} \leq 0$, that proves the stability only.

- From LaSalle's theorem, the trajectories of the system converge asymptotically towards the largest invariant set of $S = \left\{x \in \mathbb{R}^n : \dot{V}(x) = 0\right\}$.
  Yet $\dot{V}(x) = 0 \Leftrightarrow \dot{q} = 0$. The invariant set M is defined by $\dot{q} = 0$ (and $\ddot{q} = 0$ besides). In this case , we note that the system dynamic in closed-loop is :

$$
\dot{q} = 0 \Rightarrow \qquad A(q)\ddot{q} = K_p e \Rightarrow \qquad \ddot{q} = A^{-1}(q) K_p e
$$

Decentralized P.D. joint controller

- Stability proof

$$
\begin{aligned}
&= \dots \\
&= \underbrace{\frac{1}{2}\dot{q}^t\left(\dot{A}-2C\right)\dot{q}}_{=0 \text{ since } \left(\dot{A}-2C\right) \text{ is skew-sym..}} \quad -\dot{q}^t K_d \dot{q} \\
&= -\dot{q}^t K_d \dot{q} \\
&\leq 0 \quad (\text{since } K_d = diag\left(k_{d_i}\right) > 0)
\end{aligned}
$$

  Thus, $\dot{V} \leq 0$, that proves the stability only.

- From LaSalle's theorem, the trajectories of the system converge asymptotically towards the largest invariant set of $S = \left\{x \in \mathbb{R}^n : \dot{V}(x) = 0\right\}$.

  Yet $\dot{V}(x) = 0 \Leftrightarrow \dot{q} = 0$. The invariant set M is defined by $\dot{q} = 0$ (and $\ddot{q} = 0$ besides). In this case , we note that the system dynamic in closed-loop is :

$$
\dot{q} = 0 \Rightarrow \qquad A(q)\,\ddot{q} = K_p e \Rightarrow \qquad \ddot{q} = A^{-1}(q)\,K_p e
$$

  At the equilibrium, since $\ddot{q} = 0$, we deduce that $e = 0$.

Decentralized P.D. joint controller

- Stability proof

$$
\begin{aligned}
&= \dots \\
&= \underbrace{\frac{1}{2}\dot{q}^t\left(\dot{A}-2C\right)\dot{q}}_{=0 \text{ since } \left(\dot{A}-2C\right) \text{ is skew-sym..}} \quad -\dot{q}^t K_d \dot{q} \\
&= -\dot{q}^t K_d \dot{q} \\
&\leq 0 \quad (\text{since } K_d = diag\left(k_{d_i}\right) > 0)
\end{aligned}
$$

  Thus, $\dot{V} \leq 0$, that proves the stability only.

- From LaSalle's theorem, the trajectories of the system converge asymptotically towards the largest invariant set of $S = \left\{x \in \mathbb{R}^n : \dot{V}(x) = 0\right\}$.

  Yet $\dot{V}(x) = 0 \Leftrightarrow \dot{q} = 0$. The invariant set M is defined by $\dot{q} = 0$ (and $\ddot{q} = 0$ besides). In this case , we note that the system dynamic in closed-loop is :

$$
\dot{q} = 0 \Rightarrow \qquad A(q)\ddot{q} = K_p e \Rightarrow \qquad \ddot{q} = A^{-1}(q) K_p e
$$

  At the equilibrium, since $\ddot{q} = 0$, we deduce that $e = 0$. The system converges asymptotically towards the unique invariant state defined by $\dot{V}(x) = 0$, i.e. $q = q_d$ and $\dot{q} = 0$.

Course

Decentralized P.I.D. joint controller

Equation of the closed-loop linear system :

$$a_{ii}\ddot{q}_i + f_{v_i}\dot{q}_i + \gamma_i = \tau_i = k_{p_i}\left(q_{d_i} - q_i\right) + k_{d_i}\left(q_{d_i} - q_i\right) + k_{i_i}\int_{t_0}^{t}\left(q_{d_i} - q_i\right)d\lambda$$

Decentralized P.I.D. joint controller

Equation of the closed-loop linear system :

$$a_{ii}\ddot{q}_i + f_{v_i}\dot{q}_i + \gamma_i = \tau_i = k_{p_i}\left(q_{d_i} - q_i\right) + k_{d_i}\left(q_{d_i} - q_i\right) + k_{i_i}\int_{t_0}^{t}\left(q_{d_i} - q_i\right)d\lambda$$

When applying the *Laplace* trasnform $\mathscr{L}_a$ to this linear differential equation with constant coefficients,

$$X(s) = \mathscr{L}_a\left\{x(t)\right\} = \int_{0^-}^{+\infty} e^{-st}x(t)\,dt$$

Decentralized P.I.D. joint controller

Equation of the closed-loop linear system :

$$a_{ii}\ddot{q}_i + f_{v_i}\dot{q}_i + \gamma_i = \tau_i = k_{p_i}\left(q_{d_i} - q_i\right) + k_{d_i}\left(q_{d_i} - q_i\right) + k_{i_i}\int_{t_0}^{t}\left(q_{d_i} - q_i\right) d\lambda$$

When applying the *Laplace* trasnform $\mathscr{L}_a$ to this linear differential equation with constant coefficients,

$$X(s) = \mathscr{L}_a\{x(t)\} = \int_{0-}^{+\infty} e^{-st}x(t)\,dt$$

and using some remarkable properties of this transform (in this case, $\mathscr{L}_a\{\dot{x}(t)\} = sX(s)$ and $\mathscr{L}_a\{\int x(t)\} = \frac{X(s)}{s}$ with null initial conditions), it follows (with zero disturbing torque) :

$$\left(a_{ii}s^2 + f_{v_i}s\right) q_i(s) = \left(k_{p_i} + k_{d_i}s + \frac{k_{i_i}}{s}\right)\left(q_{d_i}(s) - q_i(s)\right)$$

Decentralized P.I.D. joint controller

Equation of the closed-loop linear system :

$$a_{ii}\ddot{q}_i + f_{v_i}\dot{q}_i + \gamma_i = \tau_i = k_{p_i}\left(q_{d_i} - q_i\right) + k_{d_i}\left(q_{d_i} - q_i\right) + k_{i_i}\int_{t_0}^{t}\left(q_{d_i} - q_i\right)d\lambda$$

When applying the *Laplace* trasnform $\mathscr{L}_a$ to this linear differential equation with constant coefficients,

$$X\left(s\right) = \mathscr{L}_a\left\{x\left(t\right)\right\} = \int_{0-}^{+\infty} e^{-st}x\left(t\right)dt$$

and using some remarkable properties of this transform (in this case, $\mathscr{L}_a\left\{\dot{x}\left(t\right)\right\} = sX\left(s\right)$ and $\mathscr{L}_a\left\{\int x\left(t\right)\right\} = \frac{X(s)}{s}$ with null initial conditions), it follows (with zero disturbing torque) :

$$\left(a_{ii}s^2 + f_{v_i}s\right)q_i\left(s\right) = \left(k_{p_i} + k_{d_i}s + \frac{k_{i_i}}{s}\right)\left(q_{d_i}\left(s\right) - q_i\left(s\right)\right)$$

The closed-loop transfer function can thus be deduced :

$$\frac{q_i\left(s\right)}{q_{d_i}\left(s\right)} = \frac{k_{d_i}s^2 + k_{p_i}s + k_{i_i}}{a_{ii}s^3 + \left(k_{d_i} + f_{v_i}\right)s^2 + k_{p_i}s + k_{i_i}}$$

Course

Stability of a variant of the P.D. controller

- Proposed *Lyapunov* function :

Stability of a variant of the P.D. controller

- Proposed *Lyapunov* function : $V = \frac{1}{2}\dot{q}^t A(q)\dot{q} + \frac{1}{2}e_x^t K_p e_x$

Stability of a variant of the P.D. controller

- Proposed *Lyapunov* function : $V = \frac{1}{2}\dot{q}^t A(q)\,\dot{q} + \frac{1}{2} e_x^t K_p e_x$
  - let note that $e_x = X_d - X$ the Cartesian error with X_d constant $\Rightarrow$ $\dot{e_X} = -\dot{X} = -DGM(q)$;
  - $V \geq 0$ and $V = 0 \Leftrightarrow e_x = 0$ and $\dot{q} = 0$, i.e.

$$V = 0 \Leftrightarrow [q, \dot{q}] \in \{\dot{q} = 0, q/DGM(q) = X_d\} \subseteq S$$

Stability of a variant of the P.D. controller

- Proposed *Lyapunov* function : $V = \frac{1}{2}\dot{q}^t A(q)\,\dot{q} + \frac{1}{2} e_x^t K_p e_x$
  - let note that $e_x = X_d - X$ the Cartesian error with X_d constant $\Rightarrow$ $\dot{e_X} = -\dot{X} = -DGM(q)$;
  - $V \geq 0$ and $V = 0 \Leftrightarrow e_x = 0$ and $\dot{q} = 0$, i.e.

$$V = 0 \Leftrightarrow [q, \dot{q}] \in \{\dot{q} = 0, q/DGM(q) = X_d\} \subseteq S$$

- Proof of stability

Stability of a variant of the P.D. controller

- Proposed *Lyapunov* function : $V = \frac{1}{2}\dot{q}^t A(q)\,\dot{q} + \frac{1}{2} e_x^t K_p e_x$
  - let note that $e_x = X_d - X$ the Cartesian error with X_d constant $\Rightarrow$ $\dot{e_X} = -\dot{X} = -DGM(q)$;
  - $V \geq 0$ and $V = 0 \Leftrightarrow e_x = 0$ and $\dot{q} = 0$, i.e.

$$V = 0 \Leftrightarrow [q, \dot{q}] \in \{\dot{q} = 0, q/DGM(q) = X_d\} \subseteq S$$

- Proof of stability

$$\begin{aligned}
\dot{V} &= \dot{q}^t \left(\dot{q}^t A(q)\,\ddot{q} + \frac{1}{2}\dot{A}(q)\,\dot{q}\right) - e_x^t K_p \dot{X} \\
&= \dot{q}^t \left(\Gamma - C(q,\dot{q})\,\dot{q} - G(q) + \frac{1}{2}\dot{A}(q)\,\dot{q}\right) - e_x^t K_p J\dot{q} \\
&= \dot{q}^t \left(\Gamma - G(q)\right) - e_x^t K_p J\dot{q} \\
&= \dot{q}^t \left(J(q)^t K_p e_x - K_d\dot{q} + \widehat{G}(q) - G(q)\right) - e_x^t K_p J\dot{q} \\
&= -\dot{q}^t K_d \dot{q} \\
&\leq 0
\end{aligned}$$

Stability of a variant of the P.D. controller

- Proposed *Lyapunov* function : $V = \frac{1}{2}\dot{q}^t A(q)\,\dot{q} + \frac{1}{2}e_x^t K_p e_x$
  - let note that $e_x = X_d - X$ the Cartesian error with X_d constant $\Rightarrow$ $\dot{e_X} = -\dot{X} = -DGM(q)$;
  - $V \geq 0$ and $V = 0 \Leftrightarrow e_x = 0$ and $\dot{q} = 0$, i.e.

$$V = 0 \Leftrightarrow [q, \dot{q}] \in \{\dot{q} = 0, q/DGM(q) = X_d\} \subseteq S$$

- Proof of stability

$$\begin{aligned}
\dot{V} &= \dot{q}^t\left(\dot{q}^t A(q)\,\ddot{q} + \frac{1}{2}\dot{A}(q)\,\dot{q}\right) - e_x^t K_p \dot{X} \\
&= \dot{q}^t\left(\Gamma - C(q,\dot{q})\,\dot{q} - G(q) + \frac{1}{2}\dot{A}(q)\,\dot{q}\right) - e_x^t K_p J\dot{q} \\
&= \dot{q}^t\left(\Gamma - G(q)\right) - e_x^t K_p J\dot{q} \\
&= \dot{q}^t\left(J(q)^t K_p e_x - K_d\dot{q} + \widehat{G}(q) - G(q)\right) - e_x^t K_p J\dot{q} \\
&= -\dot{q}^t K_d \dot{q} \\
&\leq 0
\end{aligned}$$

From LaSalle's theorem, the trajectories of the system converge asymptotically to the largest invariant set of $\left\{x \in \mathbb{R}^n : \dot{V}(x) = 0\right\}$.

Stability of a variant of the P.D. controller

- Proposed *Lyapunov* function : $V = \frac{1}{2}\dot{q}^t A(q)\,\dot{q} + \frac{1}{2}e_x^t K_p e_x$
  - let note that $e_x = X_d - X$ the Cartesian error with X_d constant $\Rightarrow$ $\dot{e_X} = -\dot{X} = -DGM(q)$;
  - $V \geq 0$ and $V = 0 \Leftrightarrow e_x = 0$ and $\dot{q} = 0$, i.e.

$$V = 0 \Leftrightarrow [q, \dot{q}] \in \{\dot{q} = 0, q / DGM(q) = X_d\} \subseteq S$$

- Proof of stability

$$\begin{aligned}
\dot{V} &= \dot{q}^t \left(\dot{q}^t A(q)\,\ddot{q} + \frac{1}{2}\dot{A}(q)\,\dot{q}\right) - e_x^t K_p \dot{X} \\
&= \dot{q}^t \left(\Gamma - C(q,\dot{q})\,\dot{q} - G(q) + \frac{1}{2}\dot{A}(q)\,\dot{q}\right) - e_x^t K_p J\dot{q} \\
&= \dot{q}^t \left(\Gamma - G(q)\right) - e_x^t K_p J\dot{q} \\
&= \dot{q}^t \left(J(q)^t K_p e_x - K_d\dot{q} + \widehat{G}(q) - G(q)\right) - e_x^t K_p J\dot{q} \\
&= -\dot{q}^t K_d \dot{q} \\
&\leq 0
\end{aligned}$$

From LaSalle's theorem, the trajectories of the system converge asymptotically to the largest invariant set of $\left\{x \in \mathbb{R}^n : \dot{V}(x) = 0\right\}$. Yet, $\dot{V}(x) = 0 \Leftrightarrow \dot{q} = 0$.

Stability of a variant of the P.D. controller

- Proposed *Lyapunov* function : $V = \frac{1}{2}\dot{q}^t A(q)\,\dot{q} + \frac{1}{2} e_x^t K_p e_x$
  - let note that $e_x = X_d - X$ the Cartesian error with X_d constant $\Rightarrow$ $\dot{e_X} = -\dot{X} = -DGM(q)$;
  - $V \geq 0$ and $V = 0 \Leftrightarrow e_x = 0$ and $\dot{q} = 0$, i.e.
$$V = 0 \Leftrightarrow [q, \dot{q}] \in \{\dot{q} = 0, q/DGM(q) = X_d\} \subseteq S$$
- Proof of stability
$$\begin{aligned}
\dot{V} &= \dot{q}^t \left(\dot{q}^t A(q)\,\ddot{q} + \frac{1}{2}\dot{A}(q)\,\dot{q}\right) - e_x^t K_p \dot{X} \\
&= \dot{q}^t \left(\Gamma - C(q,\dot{q})\,\dot{q} - G(q) + \frac{1}{2}\dot{A}(q)\,\dot{q}\right) - e_x^t K_p J \dot{q} \\
&= \dot{q}^t \left(\Gamma - G(q)\right) - e_x^t K_p J \dot{q} \\
&= \dot{q}^t \left(J(q)^t K_p e_x - K_d \dot{q} + \widehat{G}(q) - G(q)\right) - e_x^t K_p J \dot{q} \\
&= -\dot{q}^t K_d \dot{q} \\
&\leq 0
\end{aligned}$$

From LaSalle's theorem, the trajectories of the system converge asymptotically to the largest invariant set of $\left\{x \in \mathbb{R}^n : \dot{V}(x) = 0\right\}$. Yet, $\dot{V}(x) = 0 \Leftrightarrow \dot{q} = 0$. The invariant set is defined by $\dot{q} = 0$ (and $\ddot{q} = 0$ besides).

Stability of a variant of the P.D. controller

- Proposed *Lyapunov* function : $V = \frac{1}{2}\dot{q}^t A(q)\,\dot{q} + \frac{1}{2} e_x^t K_p e_x$
  - let note that $e_x = X_d - X$ the Cartesian error with X_d constant $\Rightarrow$ $\dot{e_X} = -\dot{X} = -DGM(q)$;
  - $V \geq 0$ and $V = 0 \Leftrightarrow e_x = 0$ and $\dot{q} = 0$, i.e.

$$V = 0 \Leftrightarrow [q, \dot{q}] \in \{\dot{q} = 0, q/DGM(q) = X_d\} \subseteq S$$

- Proof of stability

$$\begin{aligned}
\dot{V} &= \dot{q}^t \left(\dot{q}^t A(q)\,\ddot{q} + \frac{1}{2}\dot{A}(q)\,\dot{q}\right) - e_x^t K_p \dot{X} \\
&= \dot{q}^t \left(\Gamma - C(q,\dot{q})\,\dot{q} - G(q) + \frac{1}{2}\dot{A}(q)\,\dot{q}\right) - e_x^t K_p J\dot{q} \\
&= \dot{q}^t \left(\Gamma - G(q)\right) - e_x^t K_p J\dot{q} \\
&= \dot{q}^t \left(J(q)^t K_p e_x - K_d \dot{q} + \widehat{G}(q) - G(q)\right) - e_x^t K_p J\dot{q} \\
&= -\dot{q}^t K_d \dot{q} \\
&\leq 0
\end{aligned}$$

From LaSalle's theorem, the trajectories of the system converge asymptotically to the largest invariant set of $\left\{x \in \mathbb{R}^n : \dot{V}(x) = 0\right\}$. Yet, $\dot{V}(x) = 0 \Leftrightarrow \dot{q} = 0$. The invariant set is defined by $\dot{q} = 0$ (and $\ddot{q} = 0$ besides). In this case, we note that the system dynamics in closed-loop is as follows :

Stability of a variant of the P.D. controller

- Proposed *Lyapunov* function : $V = \frac{1}{2}\dot{q}^t A(q)\,\dot{q} + \frac{1}{2} e_x^t K_p e_x$
  - let note that $e_x = X_d - X$ the Cartesian error with X_d constant $\Rightarrow$ $\dot{e_X} = -\dot{X} = -DGM(q)$;
  - $V \geq 0$ and $V = 0 \Leftrightarrow e_x = 0$ and $\dot{q} = 0$, i.e.

$$V = 0 \Leftrightarrow [q, \dot{q}] \in \{\dot{q} = 0, q/DGM(q) = X_d\} \subseteq S$$

- Proof of stability

$$\begin{aligned}
\dot{V} &= \dot{q}^t \left(\dot{q}^t A(q)\,\ddot{q} + \frac{1}{2}\dot{A}(q)\,\dot{q} \right) - e_x^t K_p \dot{X} \\
&= \dot{q}^t \left(\Gamma - C(q, \dot{q})\,\dot{q} - G(q) + \frac{1}{2}\dot{A}(q)\,\dot{q} \right) - e_x^t K_p J \dot{q} \\
&= \dot{q}^t \left(\Gamma - G(q) \right) - e_x^t K_p J \dot{q} \\
&= \dot{q}^t \left(J(q)^t K_p e_x - K_d \dot{q} + \widehat{G}(q) - G(q) \right) - e_x^t K_p J \dot{q} \\
&= -\dot{q}^t K_d \dot{q} \\
&\leq 0
\end{aligned}$$

From LaSalle's theorem, the trajectories of the system converge asymptotically to the largest invariant set of $\left\{ x \in \mathbb{R}^n : \dot{V}(x) = 0 \right\}$. Yet, $\dot{V}(x) = 0 \Leftrightarrow \dot{q} = 0$. The invariant set is defined by $\dot{q} = 0$ (and $\ddot{q} = 0$ besides). In this case, we note that the system dynamics in closed-loop is as follows :

$$\ddot{q} = A^{-1}(q)\left(\Gamma - G(q)\right) \Rightarrow \ddot{q} = A^{-1}(q)\,J(q)^t K_p e_x$$

Stability of a variant of the P.D. controller

- Proposed *Lyapunov* function : $V = \frac{1}{2}\dot{q}^t A(q)\,\dot{q} + \frac{1}{2}e_x^t K_p e_x$
  - let note that $e_x = X_d - X$ the Cartesian error with X_d constant $\Rightarrow$ $\dot{e_X} = -\dot{X} = -DGM(q)$;
  - $V \geq 0$ and $V = 0 \Leftrightarrow e_x = 0$ and $\dot{q} = 0$, i.e.

$$V = 0 \Leftrightarrow [q, \dot{q}] \in \{\dot{q} = 0, q/DGM(q) = X_d\} \subseteq S$$

- Proof of stability

$$\begin{aligned}
\dot{V} &= \dot{q}^t \left(\dot{q}^t A(q)\,\ddot{q} + \frac{1}{2}\dot{A}(q)\,\dot{q}\right) - e_x^t K_p \dot{X} \\
&= \dot{q}^t \left(\Gamma - C(q,\dot{q})\,\dot{q} - G(q) + \frac{1}{2}\dot{A}(q)\,\dot{q}\right) - e_x^t K_p J\dot{q} \\
&= \dot{q}^t (\Gamma - G(q)) - e_x^t K_p J\dot{q} \\
&= \dot{q}^t \left(J(q)^t K_p e_x - K_d\dot{q} + \widehat{G}(q) - G(q)\right) - e_x^t K_p J\dot{q} \\
&= -\dot{q}^t K_d \dot{q} \\
&\leq 0
\end{aligned}$$

From LaSalle's theorem, the trajectories of the system converge asymptotically to the largest invariant set of $\left\{x \in \mathbb{R}^n : \dot{V}(x) = 0\right\}$. Yet, $\dot{V}(x) = 0 \Leftrightarrow \dot{q} = 0$. The invariant set is defined by $\dot{q} = 0$ (and $\ddot{q} = 0$ besides). In this case, we note that the system dynamics in closed-loop is as follows :

$$\ddot{q} = A^{-1}(q)\,(\Gamma - G(q)) \Rightarrow \ddot{q} = A^{-1}(q)\,J(q)^t K_p e_x$$

At the equilibrium, since $\ddot{q} = 0$, we deduce that $K_p e_x \in N_{J(q)^t}$.

Course