Formulary – Robotics II [A.De Luca]

(2020-2021)

8.Regulation

Pd Control Law $u = K_p(q_d - q) - K_D\dot{q}$

Goal: asymptotic stabilization (= regulation) of the closed-loop equilibrium state

This control law achieves asymptotic stabilization of a desired state $(q,\dot{q}) = (q_d, 0)$ also in the presence of gravity if: i) $g(q_d) = 0$ (the desired configuration is an unforced equilibrium for the open-loop system); ii) KP is symmetric and positive definite, and its minimum eigenvalue $K_{p,m} > \alpha$, where α > 0 is a global upper bound on the norm of the Hessian of the potential energy $U_g(q)$ due to gravity; iii) KD is symmetric and positive definite. In general, these are only sufficient conditions.

***Pd* Control Law with Gravity Cancellation** $u = K_p(q_d - q) - K_D\dot{q} + g(q)$

Pd Control Law with Gravity Compensation $u = K_p(q_d - q) - K_D\dot{q} + g(q_d)$,
If $K_{p,m} > \alpha$, the state $(q_d, 0)$ of the robot under joint-space PD control + constant gravity compensation at qd is globally asymptotically stable.

PID control law: $u(t) = K_p(q_d - q)) - K_I \int_0^t (q_d - q(\tau))d\tau - K_D\dot{q}(t)$, in robots, a PID may be used to recover such a position error due to an incomplete (or absent) gravity compensation/cancellation.

9.Iterative Learning

[5/02/2018 ex.3; α computation in ex.3 5/06/20 and 6/06/17 ex.3]

Starting from the robot dynamic model

$$M(q)\dot{q} + c(q,\dot{q}) + g(q) = u$$

We have available a bound on the gradient of the gravity term

$$\left\| \frac{\delta g(q)}{\delta q} \right\| \le \alpha$$

Note that we can compute the norm of a matrix using the formula

$$\|A\| = \sqrt{\lambda_{max}(A^T A)}$$

To compute the maximum eigenvalue, solve the second-grade equation given by the characteristic poly:

$$\det(\lambda * I - A^T A) = 0$$

The iterative control law at i-th iteration is $u = \gamma K_p(q_d - q) - K_D\dot{q} + u_{i-1}$. The sequence $\{q_0, q_1, q_2 \dots\}$ converges to q_d (and $\dot{q} = 0$) from any initial value q_0, i.e. globally if

- $\lambda_{minn}(K_p) > \alpha$
- $\gamma \geq 2$

Combining them, the final condition is $\lambda_{min}(\widehat{K_p}) > 2\alpha$, given that $\widehat{K_p} = \gamma K_p$. We can choose the highest value for γ such that the condition $\lambda_{min}(K_p) > \alpha$ still holds.

The error decreases as: $\|e_i\| = \frac{\|e_{i-1}\|}{\gamma - 1}$

10. Trajectory Control

[11/06/21 ex.2]

Feedback Linearization $u = \widehat{M(q)}\left[\ddot{q_d} + K_p(q_d - q) + K_d(\dot{q}_d - \dot{q})\right] + \hat{n}(q, \dot{q})$

$\ddot{q_d} + K_p(q_d - q) + K_d(\dot{q}_d - \dot{q})$ is named a with $a = \ddot{q}$.

This law guarantees an exponential decay of the error transient and a decoupling among each **joint** coordinate.

$$\ddot{e} + K_D\,\dot{e} + K_p e = 0 \leftrightarrow \ddot{e_i} + K_{Di}\,\dot{e_i} + K_{pi} e_i = 0$$

Alternative global trajectory controller

$$u = M(q)\,\ddot{q_d} + S(q, \dot{q})\dot{q_d} + g(q) + F_v\dot{q_d} + K_p e + K_D\dot{e}$$

Guarantees asymptotic stability of $(e, \dot{e}) = (0, 0)$, but does not produce a complete cancellation of nonlinearities.

PID control law: $u(t) = K_p(q_d - q)) - K_I\int_0^t(q_d - q(\tau))d\tau - K_D\dot{q}(t)$, more robust to uncertainties, but also more complex to implement in real time.

12. Adaptive Control

[29/05/2016-17 ex.3; 5/06/20 ex.4;11/06/18 ex.2]

Goal of adaptive control: given a twice differentiable desired joint trajectory $q_d(t)$ we want to execute it under large dynamic uncertainties, with a trajectory tracking error vanishing asymptotically and guaranteeing global stability, no matter how big is the initial trajectory error.

Idea: on-line modification with a reference velocity

$$\dot{q_r} = \dot{q_d} + \Lambda(q_d - q)$$

Typically, $\Lambda = K_D^{-1}K_p$. Moreover $\ddot{q_r} = \ddot{q_d} + \Lambda\dot{e}$

Thus, $u = \widehat{M}(q)\,\ddot{q}_r + \hat{S}(q,\dot{q})\dot{q}_r + \hat{g}(q) + \hat{F}_v\dot{q}_r + K_p e + K_D\dot{e} = Y(q,\dot{q},\dot{q}_r,\ddot{q}_r)\,\hat{a} + K_p e + K_D\dot{e}$

Where the update law for the estimates of the dynamic coefficients is $\dot{\hat{a}} = \Gamma Y^T(q,\dot{q},\dot{q}_r,\ddot{q}_r)\,(\dot{q}_r - \dot{q})$, $\Gamma > 0$ diagonal

13.Cartesian Control

[17/06/19 ex.4, 12/01/21 ex.2]

Starting from the robot dynamic model:

$$u = M(q)a + c(q,\dot{q}) + g(q)$$

Imposing $a = \ddot{q} = J^{\#}(\ddot{p} - \dot{J}\dot{q}) = J^{\#}(\ddot{p}_d + K_D(\dot{p}_d - \dot{p}) + K_P(p_d - p) - \dot{J}\dot{q})$

Because $(\ddot{p}_d - \ddot{p}) + K_D\,(\dot{p}_d - \dot{p}) + K_p(p_d - p) = 0$

This law guarantees an exponential decay of the error transient and a decoupling among each **cartesian** coordinate.

If we have to design a control law for the robot such that the trajectory tracking error dynamics is exponentially stable, linear, and decoupled along the **normal and tangential directions to the path** [10/09/09 ex.1, 11/06/21 ex.5]:

$$t = R^T(\alpha)p$$

We want to obtain a decoupled and exponentially decaying error in the task frame: $\ddot{e}_t + K_D\,\dot{e}_t + K_p e_t = 0$ (1)

We have:

$$e_t = R^T e, \qquad with\; e = (p_d - p)$$

$$\dot{e}_t = R^T\dot{e} + \dot{R}^T e, \quad with\; \dot{e} = (\dot{p}_d - \dot{p})$$

$$\ddot{e}_t = R^T\ddot{e} + 2\dot{R}^T\dot{e} + \ddot{R}^T e, \quad with\; \ddot{e} = (\ddot{p}_d - \ddot{p}) \quad (2)$$

We now have to compute $\ddot{p}$ to complete the control law (plugging the (2) in the (1) and by making explicit the term $\ddot{p}$ knowing that $\ddot{e} = (\ddot{p}_d - \ddot{p})$):

$$\ddot{p} = \ddot{p}_d + R_t\big(K_{d,t}\dot{e}_t + K_{p,t}\,e_t + 2\dot{R}^T\dot{e} + \ddot{R}^T e\big)$$

Recalling that $a = \ddot{q} = J^{\#}(\ddot{p} - \dot{J}\dot{q})$ in the dynamic model:

$$u = M(q)a + c(q,\dot{q}) + g(q)$$

14.Environment Interaction

Constrained dynamics [09/01/13 ex.2]

The **constrained dynamic model** is:

$$M(q)\,\ddot{q} = \left[I - A^T(q)\left(A_M^{\#}(q)\right)^T\right]\left(u - c(q,\dot{q}) - g(q) - M(q)A_M^{\#}(q)\dot{A}(q)\dot{q}\right)$$

Where:

- A is the constrained Jacobian.
- $\left[I - A^T(q)\left(A_M^{\#}(q)\right)^T\right]$ is the dynamically consistent projection matrix.
- $A_M^{\#}(q) = M^{-1}(q)A^T(q)\left(A(q)M^{-1}(q)A^T(q)\right)^{-1}$
- $\lambda = \left(A_M^{\#}\right)^T(c(q,\dot{q}) + g(q) - u) - (AM^{-1}A^T)^{-1}\dot{A}\dot{q}$

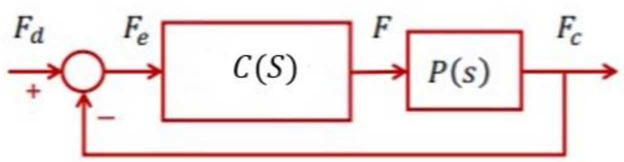
Reduced dynamics [11/07/18 ex.4; 11/09/20 ex.4; 11/07/17 ex.2; 6/02/13]

Steps for the **reduced dynamic model**:

1. Dynamic model → $M(q)\ddot{q} + c(q,\dot{q}) + g(q) = u$
2. Imposing the M dimensional constraints → $h(q) = 0$
3. Compute the Jacobian of the constraints → $\mathrm{A(q)} = \frac{\partial \mathrm{h(q)}}{\partial \mathrm{q}}$
4. Choose the matrix $D(q)$ such that → $\begin{matrix} A(q) \\ D(q) \end{matrix}$ is a non-singular ($det \neq 0$) NxN matrix. (reduceDynamics on Matlab)
5. Compute the inverse in order to obtain $E(q)$ and $F(q)$ → $\begin{matrix} A(q) \\ D(q) \end{matrix}^{-1} = [E(q)\ \ F(q)]$.
6. Define the (N-M)- dimensional vector of pseudo-velocities v as the linear combination of the robot generalized velocities → $\mathrm{v = D(q)\dot{q}}$ and $\mathrm{\dot{v} = D(q)\ddot{q} + \dot{D}(q)\dot{q}}$.
   Remembering that: $N = degrees\ of\ freedom\ of\ the\ robot$ and $M = \text{number of constraint}$.
7. Inverse relationship (from pseudo to generalized velocities and acceleration) → $\dot{q} = F(q)v$ and $\ddot{q} = F(q)\dot{v} + \dot{F}(q)v\ = F(q)v + \left(E(q)\dot{A}(q) + F(q)\dot{D}(q)\right)F(q)v$
8. The new dynamic model with the Jacobian of constraints → $M(q)\ddot{q} + c(q,\dot{q}) + g(q) = u + A^T(q)\lambda$
9. Reduced (N-M)- dimensional dynamic model → $(F^T MF)\dot{v} = F^T\left(u - c - g + M\dot{F}v\right)$ where $F^T MF$ is the reduced inertia matrix.
10. If requested, the force multipliers →
$$\lambda = E^T\left(MF\dot{v} - M\dot{F}v + c + g - u\right)$$

15. Laplace domain (masses with Fc control)

[07-07-10 ex.1; 04-02-21 ex.4; 11-06-18 ex.3; 28/10/16 ex.2]



$P(s)$ is the plant of the open loop system and it is computed as the ratio between output and input:

$$P(S) = \frac{output}{input} = \frac{F_c(s)}{F(s)}$$

$C(s)$ is the controller:

- For a simple feedback controller $F = K_p(F_d - F_c) = K_p F_e$ we have $C(s) = \frac{F(s)}{F_e(s)} = K_p > 0$
- For a proportional-integral (PI) controller on the force error $F(t) = K_p F_e(t) + K_I \int_0^t F_e(\tau)d\tau$ we have $C(s) = K_p + \frac{K_I}{s}$

We can compute the transfer function of the system as:

$$W(s) = \frac{F_c(s)}{F_d(s)} = \frac{P(s)C(s)}{1 + P(s)C(s)}$$

Starting from the transfer function we can study the system behavior: for example, if it has all negative poles, it is asymptotically stable. However, note that even if it is asymptotically stable if the gain, computed as $W(0)$ is not unitary we have an error (we can delete or reduce it by using the **PI controller shown above**). We can compute the value of this error starting from the input-error transfer function

$$W_e(s) = \frac{F_e(s)}{F_d(s)} = \frac{F_d(s) - F_c(s)}{F_d(s)} = 1 - W(s)$$

Then, using the final value theorem, the steady-state error for a constant F_d is computed as:

$$F_{e,\infty} = \lim_{t\to\infty} F_e(t) = \lim_{s\to 0} F_e(s) = \lim_{s\to 0} W_e(s)F_d(s) = W_e(0)F_d$$

Or if we are in an equilibrium state $e_F = F_d - F_{c,e}$

11/06/18

venerdì 9 luglio 2021 10:18

$$F = m\ddot{x} + d\dot{x} + K_s x$$

$$F = \alpha K_f (F_d - F_c) + \beta F_d$$

$$F_c = K_s x$$

$$P(s) = \frac{K_s}{ms^2 + ds + K_s} = \frac{F_c}{F}$$

Combined

$$W(s) = \frac{F_c}{F_d} = \frac{K_s(1+K_f)}{ms^2+ds+K_s+K_f K_s} = \frac{K_s(1+K_f)}{ms^2+ds+K_s(1+K_f)} \quad \alpha=1 \quad \beta=1 \quad F_d = \frac{F + K_f F_c}{1+K_f}$$

Pure feedback

$$W(s) = \frac{P(s)C(s)}{1+P(s)C(s)} = \frac{K_s K_f}{ms+ds+K_s(1+K_f)} \quad \alpha=1 \quad \beta=0 \quad C(s) = K_f = \frac{F \to K_f(F_d-F_c)}{F_e \to F_d - F_c}$$

Pure feedforward

$$W(s) = \frac{F_c}{\underbrace{F_d}_{=F}} = \frac{K_s}{ms^2+ds+K_s} = P(s) \quad \alpha=0 \quad \beta=1$$

Impedance control

[11/06/12 ex.2; 11/01/18]

Dynamic model: $M(q)\ddot{q} + c(q,\dot{q}) + g(q) = \tau + J_c^T(q)F_c$

Regulation control law = $\tau = g(q) - J_c^T(q)F_c + K_p(q_d - q) - K_D\dot{q}$

16.Hybrid Control

[17/06/19 ex.5]

Natural constraint:

- **End-effector motion** $\left(\frac{v}{w}\right)$ **<u>is prohibited</u>** along/around $6-K$ directions (since the environment reacts there with forces/torques).
- **Reaction forces/torques** $\left(\frac{f}{m}\right)$ **<u>are absent</u>** along/around K *directions* (where the environment does not are absent prevent end-effector motions).

Artificial constraint:

- **End-effector velocities** $\left(\frac{v}{w}\right)$ along/around K *directions* where feasible motions can occur.

- **Contact forces/torques** $\left(\frac{f}{m}\right)$ along/around $6-K$ directions where admissible reactions of the environment can occur.

Steps for the **hybrid control**:

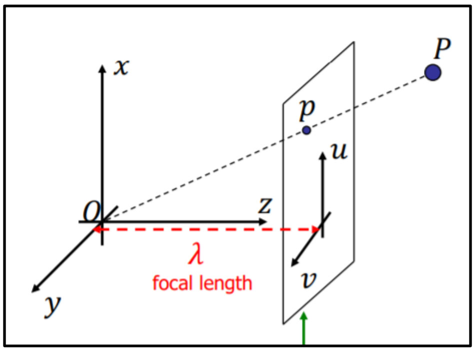
1. Choose the reference frame by drawing on the figure.
2.

NATURAL CONSTRAINT	ARTIFICIAL CONSTRAINT
$v_i = 0$	$f_i = f_i, des\ \ (= 0)$
$v_i \neq 0, f_i = 0$	$v_i = v_i, des$
$\varpi_i = 0$	$m_i = m_i, des\ \ (= 0)$
$\varpi_i \neq 0, m_i = 0$	$\varpi_i = \varpi_i, des(= 0)$

3. Compute $\boldsymbol{K}$ (generalized force components) and $\boldsymbol{6-K}$ (planar motion components) if requested.

17. Visual Servoing

[29/05/2017 ex.2 polar]



$u = \lambda\frac{X}{Z}$

$v = \lambda\frac{Y}{Z}$

$P = (X, Y, Z)$ Cartesian Point (camera frame).

$p = (u, v, \lambda)$ Representative point on the image plane.

Interaction Matrix:

$$\begin{bmatrix} u^{\cdot} \\ v^{\cdot} \end{bmatrix} = \begin{bmatrix} -\frac{\lambda}{Z} & 0 & \frac{u}{Z} & \frac{uv}{\lambda} & -\left(\lambda + \frac{u^2}{\lambda}\right) & v \\ 0 & -\frac{\lambda}{Z} & \frac{v}{Z} & \left(\lambda + \frac{u^2}{\lambda}\right) & -\frac{uv}{\lambda} & u \end{bmatrix} \begin{bmatrix} V \\ \Omega \end{bmatrix} = J_P(u, v, Z) \begin{bmatrix} V \\ \Omega \end{bmatrix}$$

$$\begin{bmatrix} u^{\cdot} \\ v^{\cdot} \end{bmatrix} = J_P(u, v, Z) \begin{bmatrix} V \\ \Omega \end{bmatrix}$$

Formule utili

Exercises with null space: [12/01/21 ex. 1-2]

velocity $\dot{r} = J_r(q)\dot{q}$

acceleration $\ddot{r} = J_r(q)\ddot{q} + \dot{J}_r(q)\dot{q}$

$J(q)\ddot{q} = \ddot{r} - \dot{J}(q)\dot{q} \triangleq \ddot{x}$

$$\ddot{q} = J^{\#}(q)\ddot{x} + \underbrace{\left(I - J^{\#}(q)J(q)\right)}_{\text{proiezione nel NULL SPACE}}\ddot{q}_0 \to \underbrace{\nabla_q H}_{\text{minimising error}} \underbrace{- K_D\dot{q}}_{\text{damping velocity term}}$$

$\dot{q} = J^{\#}\dot{r} + (I - J^{\#}J)\dot{q}_0$

$J(q)$ passaggio da joint a cartesian space

$J^{\#}(q)$ passaggio da cartesian a joint space

Exercise with jerk 06/06/17 ex.1

$\dot{p} = J\dot{q}$ $\quad \dot{q} = J^{-1}\dot{p}$

$\ddot{p} = J\ddot{q} + \dot{J}\dot{q}$ $\quad \ddot{q} = J^{-1}(\ddot{p} - \dot{J}\dot{q})$

$\dddot{p} = J\dddot{q} + \dot{J}\ddot{q} + \ddot{J}\dot{q} + \dot{J}\ddot{q} = J\dddot{q} + 2\dot{J}\ddot{q} + \ddot{J}\dot{q}$

$\dddot{q} = J^{-1}(\dddot{p} - 2\dot{J}\ddot{q} - \ddot{J}\dot{q})$ ← DIFF. INVERSION SCHEME

$\dddot{q} = J^{-1}\left(\dddot{p} - 2\dot{J}\left(J^{-1}\ddot{p} - \dot{J}J^{-1}\dot{p}\right) - \ddot{J}J^{-1}\dot{p}\right)$

Useful exercises:

- 27/1014 ex.2 maximum norm contact force with torque bounds
- 28/10/16 ex.2 Lagrangian approach and stability proof with Lyapunov
- 15/07/20 ex.4 torque trajectory time scaling
- 15/07/20 ex.5 different laws for different stabilities conditions (first case use Laplace to analyse transient)

Basic control law $\quad u = K_P(q_d - q) + K_D(\dot{q}_d - \dot{q})$

Gravity cancellation $\quad u = K_P(q_d - q) + K_D(\dot{q}_d - \dot{q}) + g(q)$

Gravity compensation $\quad u = K_P(q_d - q) + K_D(\dot{q}_d - \dot{q}) + g(q_d)$

Feedback linearization, $a = \ddot{q}$:

Joint space

$u = M(q)\,a + n(q, \dot{q}) \qquad (\ddot{q}_d - \ddot{q}) + K_D(\dot{q}_d - \dot{q}) + K_P(q_d - q) = 0$

$a = \ddot{q}_d + K_D(\dot{q}_d - \dot{q}) + K_P(q_d - q)$

Cartesian space

$u = M(q)\,a + n(q, \dot{q}) \qquad (\ddot{p}_d - \ddot{p}) + K_D(\dot{p}_d - \dot{p}) + K_P(p_d - p) = 0$

non avendo errore in joint space:

$a = J^{\#}(\ddot{p} - \dot{J}\dot{q}) \qquad$ since $\ddot{p} = Ja + \dot{J}\dot{q}$

$= J^{\#}(\ddot{p}_d + K_D(\dot{p}_d - \dot{p}) + K_P(p_d - p) - \dot{J}\dot{q}) \quad$ with $\ddot{p} = \ddot{p}_d + K_D(\dot{p}_d - \dot{p}) + K_P(p_d - p)$

Alternative global trajectory $\quad u = M(q)\ddot{q}_d + S(q, \dot{q})\dot{q}_d + g(q) + F_v\dot{q}_d + K_P e + K_D \dot{e}$

Adaptive control law $\quad u = \hat{M}(q)\ddot{q}_r + \hat{S}(q, \dot{q})\dot{q}_r + \hat{g}(q) + \hat{F}_v\dot{q}_r + K_P e + K_D\dot{e}$

$= Y(q, \dot{q}, \dot{q}_r, \ddot{q}_r)\,\hat{a} + K_P e + K_D \dot{e}$

$\dot{\hat{a}} = \Gamma Y^T(q, \dot{q}, \dot{q}_r, \ddot{q}_r)(\dot{q}_r - \dot{q})$

ROBOT 2R:

The requested symbolic form of the terms in (5) are easily obtained for a 2R planar robot (see lecture slides). The kinematic terms are

$$\boldsymbol{p}(\boldsymbol{q}) = \begin{pmatrix} l_1 \cos q_1 + l_2 \cos(q_1 + q_2) \\ l_1 \sin q_1 + l_2 \sin(q_1 + q_2) \end{pmatrix},$$

$$\boldsymbol{J}(\boldsymbol{q}) = \frac{\partial \boldsymbol{p}(\boldsymbol{q})}{\partial \boldsymbol{q}} = \begin{pmatrix} -(l_1 \sin q_1 + l_2 \sin(q_1 + q_2)) & -l_2 \sin(q_1 + q_2) \\ l_1 \cos q_1 + l_2 \cos(q_1 + q_2) & l_2 \cos(q_1 + q_2) \end{pmatrix},$$

$$\dot{\boldsymbol{J}}(\boldsymbol{q}) = -\begin{pmatrix} l_1 \cos q_1 \dot{q}_1 + l_2 \cos(q_1 + q_2)(\dot{q}_1 + \dot{q}_2) & l_2 \cos(q_1 + q_2)(\dot{q}_1 + \dot{q}_2) \\ l_1 \sin q_1 \dot{q}_1 + l_2 \sin(q_1 + q_2)(\dot{q}_1 + \dot{q}_2) & l_2 \sin(q_1 + q_2)(\dot{q}_1 + \dot{q}_2) \end{pmatrix}.$$

The dynamic terms are

$$\boldsymbol{M}(\boldsymbol{q}) = \begin{pmatrix} a_1 + 2a_2 \cos q_2 & a_3 + a_2 \cos q_2 \\ a_3 + a_2 \cos q_2 & a_3 \end{pmatrix},$$

$$\boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \begin{pmatrix} -a_2 \sin q_2 \, (2\dot{q}_1 + \dot{q}_2)\, \dot{q}_2 \\ a_2 \sin q_2 \, \dot{q}_1^2 \end{pmatrix},$$

with dynamic coefficients $a_1 = I_{c1,zz} + m_1 d_{c1}^2 + I_{c2,zz} + m_2 d_{c2}^2 + m_2 l_1^2 > 0$, $a_2 = m_2 l_1 d_{c2}$ and $a_3 = I_{c2,zz} + m_2 d_{c2}^2 > 0$.