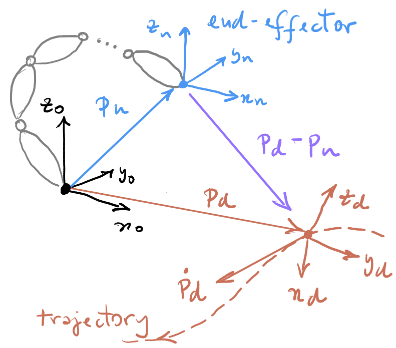
CLIK

• Position Control

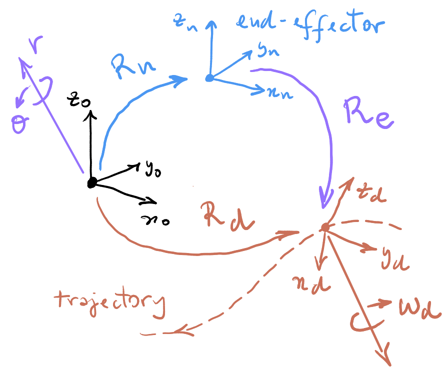


The end-effector's velocity is specified according to the desired velocity, $\dot{p}_d$, and the position error, $e_p = p_d - p_n$,

$$\dot{p}_n = \dot{p}_d + K_p (p_d - p_n) . \qquad (1)$$

Here, K_p is a positive definite matrix.

• Orientation Control



The orientation error is measured through the rotation matrix R_e,

$$R_e = R_d R_n^T . \qquad (2)$$

An angle-axis meaning can be attributed to R_e to obtain an error vector, $e_o = e_o(\theta, r)$.

Some typical choices are $e_o = \theta r$, $e_o = \sin(\theta)\, r$ and $e_o = \sin(\frac{\theta}{2})\, r$, which are computed by different means. Then the orientation control follows,

$$w_n = w_d + K_o e_o(\theta, r) . \qquad (3)$$

• CLIK Controller

Gathering the previous results, $(\dot{p}_n^T \, w_n^T)^T = \mathcal{J} \dot{q}$ becomes

$$\dot{q}(t) = \mathcal{J}^{-1} \begin{pmatrix} \dot{p}_d + K_p (p_d - p_n) \\ w_d + K_o e_o(\theta, r) \end{pmatrix} , \quad q(t) = \int_0^t \dot{q}(u)\, du + q(0) , \qquad (4)$$

where $q(0)$ denotes the initial configuration of the manipulator.

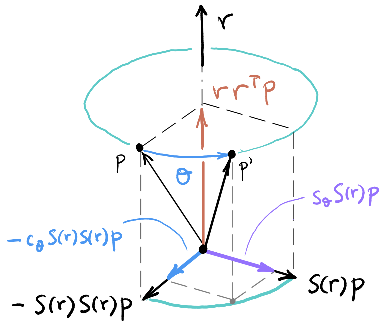
Orientation Error

The orientation error vector employed here is

$$e_o = \theta\, r \,. \tag{5}$$

The components $\{\theta, r\}$ are extracted from Rodrigues' rotation formula.

- Rodrigues' rotation formula



This formulation can be derived geometrically, by observing how a point, p, rotates about a unitary axis, r with $\|r\|=1$, by an angle θ,

$$p' = rr^Tp + s_\theta S(r)\,p - c_\theta S(r)S(r)\,p. \tag{6}$$

Here, S(a) denotes the skew-symmetric form of vector a,

$$a \times b = S(a)\,b = \begin{pmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{pmatrix}\begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix}, \quad \mathrm{vect}\big(S(a)\big) = a\,. \tag{7}$$

The rotation formula is therefore given by,

$$R(\theta, r) = rr^T + s_\theta S(r) - c_\theta S(r)S(r), \text{ with } \|r\|=1,\ \theta \in \mathbb{R}\,. \tag{8}$$

Its expansion is also usefull,

$$R(\theta,r) = \begin{pmatrix} r_x^2 & r_xr_y & r_xr_z \\ r_xr_y & r_y^2 & r_yr_z \\ r_xr_z & r_yr_z & r_z^2 \end{pmatrix} + \sin(\theta)\begin{pmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{pmatrix} - \cos(\theta)\begin{pmatrix} -r_y^2-r_z^2 & r_xr_y & r_xr_z \\ r_xr_y & -r_x^2-r_z^2 & r_yr_z \\ r_xr_z & r_yr_z & -r_x^2-r_y^2 \end{pmatrix}. \tag{9}$$

- Angle-axis extraction

The angle and axis of rotation are now extracted from a rotation matrix.

The most general case for the axis follows from $R - R^T = 2 s_\theta S(r)$,

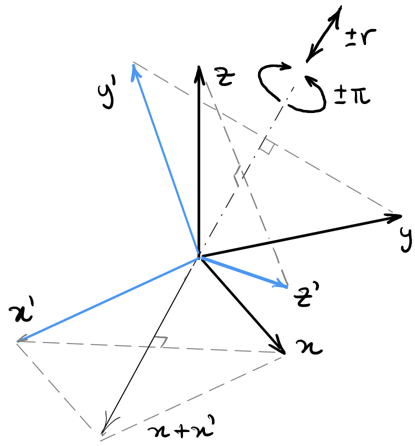
$$r = \frac{1}{2 s_\theta}\,\mathrm{vect}(R - R^T)\,, \quad \forall\, s_\theta \neq 0\,, \tag{10}$$

and the most general case for the angle follows from trace(R) =

$= r_x^2 + r_y^2 + r_z^2 + 2c_\theta(r_x^2+r_y^2+r_z^2) = 1 + 2\cos(\theta) \in [-1, 3]$. Allowing to write

$$\theta = \arccos\left(\frac{\mathrm{trace}(R) - 1}{2}\right), \quad \forall\ -1 \le \mathrm{trace}(R) \le 3. \tag{11}$$

Special cases arise when $\sin(\theta)=0$ and r cannot be determined from (10).
If $\theta = 0$, R becomes $R(0,r) = I_3$, and the direction may be arbitrary.
If $\theta = \pm\pi$, R becomes $R(\pm\pi, \pm r) = rr^T - S(r)S(r)$, which is equivalent to a reflection along the r axis, as illustrated.



In principle, any direction $\{x, y, z\}$ may be used to determine r. For instance,

$$r = \frac{x+x'}{\|x+x'\|} = \frac{\hat{i}+R\hat{i}}{\sqrt{(\hat{i}+R\hat{i})^T(\hat{i}+R\hat{i})}} = \frac{\hat{i}+R\hat{i}}{\sqrt{2+2\,\hat{i}^T R\hat{i}}} \,. \quad (12)$$

However, if $x' = -x$ then $\|x+x'\| = 0$. Thus a voting scheme may be employed to determine the largest possible denominator ($\|x+x'\|$, $\|y+y'\|$ or $\|z+z'\|$).
This results in

$$r = \pm \frac{\hat{e}_i + R\hat{e}_i}{\sqrt{2+2\hat{e}_i^T R\hat{e}_i}} \quad \text{with} \quad i : \hat{e}_i^T R\hat{e}_i \in \max(\mathrm{diag}(R)) \,, \quad (13)$$

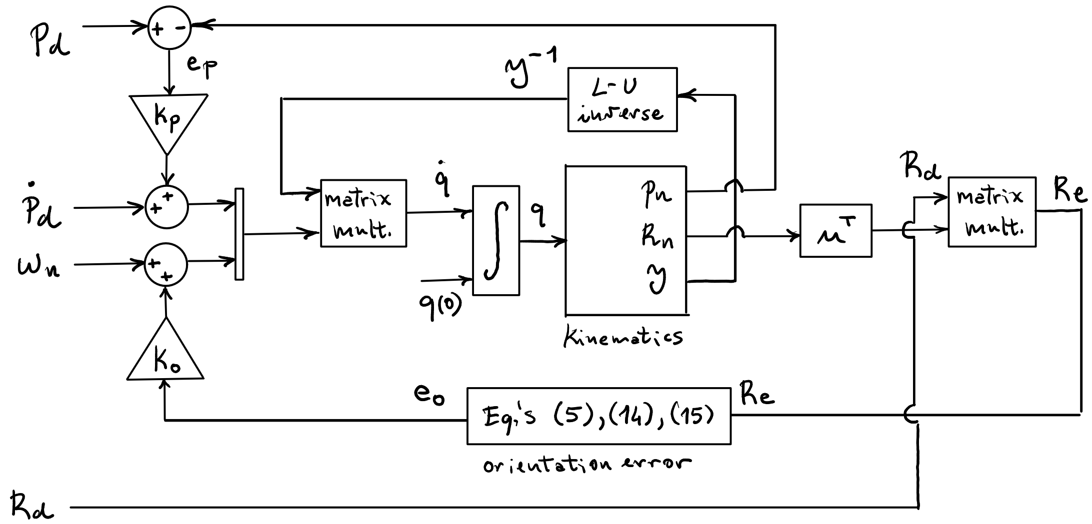
where $\hat{e}_1 = \hat{i}$, $\hat{e}_2 = \hat{j}$ and $\hat{e}_3 = \hat{k}$. Summarizing, we have

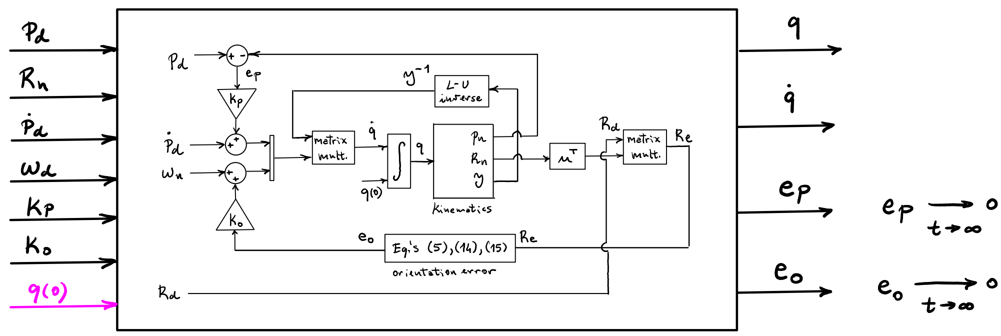
$$\{\theta, r\} = \begin{cases} \begin{cases} \theta = \arccos\left(\frac{\mathrm{trace}(R)-1}{2}\right) \\ r = \frac{1}{2S\theta}\begin{pmatrix} R_{32}-R_{23} \\ R_{13}-R_{31} \\ R_{21}-R_{12} \end{pmatrix} \end{cases} & , \text{if } -1 < \mathrm{trace}(R) < 3 \\ \begin{cases} \theta = \pm\pi \\ \begin{cases} r = \pm\frac{\hat{i}+R\hat{i}}{\sqrt{2+2R_{11}}} & , \text{if } R_{11} \in \max(\mathrm{diag}(R)) \\ r = \pm\frac{\hat{j}+R\hat{j}}{\sqrt{2+2R_{22}}} & , \text{if } R_{22} \in \max(\mathrm{diag}(R)) \\ r = \pm\frac{\hat{k}+R\hat{k}}{\sqrt{2+2R_{33}}} & , \text{if } R_{33} \in \max(\mathrm{diag}(R)) \end{cases} \end{cases} & , \text{if } \mathrm{trace}(R) = -1 \\ \begin{cases} \theta = 0 \\ r = (0\ 0\ 1)^T \end{cases} & , \text{otherwise} \end{cases} \quad (14)$$

Finally, a normalizing step is usefull to overcome numerical issues,

$$r = r / \|r\| \,. \quad (15)$$

CLIK - Block Diagram





CLIK

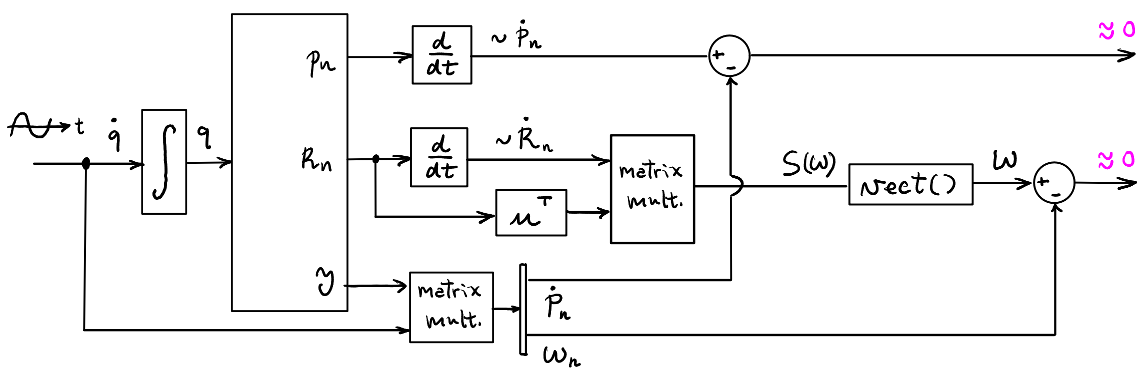
Note: $q(0)$ must not be set as a manipulator singularity.

Jacobian validation

Verify that $\dot p_n - J_p \dot q = 0$ and $w_n - J_o \dot q = 0$.

Note that $\dot R = (\dot x \;\; \dot y \;\; \dot z) = (\omega \times x \;\; \omega \times y \;\; \omega \times z)$, or

$$\dot R = S(\omega) R, \text{ where } S(\omega) = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix} \text{ and } vect(S(\omega)) = \omega. \qquad (16)$$



Note: The accuracy of the numerical derivatives can be improved by reducing the max step size in simulink's solver settings.