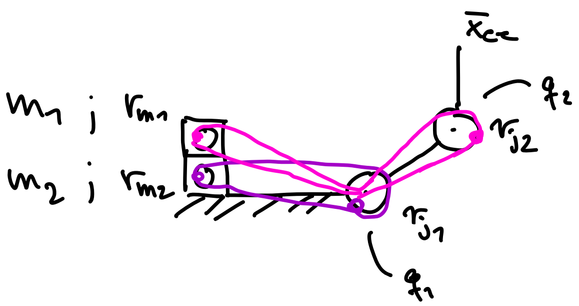
Recap: $\bar{x}_{ee} = f(\bar{q})$ Forward Kinematic Function

Problems
- difficult to obtain inverse
- inverse is not unique

Strategy:
- use local linear approxim. as proxy: $\left[\frac{\partial \bar{x}_i}{\partial q_j}\right]$
  - → good enough
  - → easy to invert
  - → we operate with steps $(\delta \bar{q}, \delta \bar{x}_{ee})$ and no longer with absolute values

Another Example:

(Purpose is to demonstrate that the same strategy can also be applied to other problems)



(robotic finger)

Slightly more complicated, because we do not control the joints directly, but the motors (connected via tendons).

We can easily obtain the forward model, predicting the joint positions, based on the motor positions.

$$\bar{q} = g(\bar{m}) = \begin{bmatrix} \left(\frac{r_{j1}}{r_{m1}}\right) m_1 \\ \left(\frac{r_{j2}}{r_{m2}}\right) m_2 \end{bmatrix}$$

We can combine the forward "tendon model" g and the forward kinematics model f, to obtain a new model, predicting the end effector pose $\bar{x}_{ee}$ based on the motor positions:

$$\bar{x}_{ee} = f(\overbrace{g(\bar{m})}^{\bar{q}}) = f \circ g\,(\bar{m})$$

If we wanted to know the motor values $\bar{m}$ that realize a desired pose $\bar{x}^*_{ee}$, we would need to invert this combined model:

$$\bar{m} = (f \circ g)^{-1}\,(m)$$

But we have the same problems:

- difficult to obtain
- potentially not unique

We apply the same strategy:

- use local linear approximation as a proxy: $\left[\dfrac{\partial \bar{x}_i}{\partial m_j}\right]$

Similar to before, we now operate with steps:

$$\delta \bar{x}_{ee} = \frac{\partial \bar{x}_{ee}}{\partial \bar{m}} \, \delta m$$

We can easily invert this:

$$\delta \bar{m} = \left(\frac{\partial \bar{x}_{ee}}{\partial \bar{m}} \right)^{-1} \delta \bar{x}_{ee}$$

This allows us to control the fingertip by adjusting the motor positions

We can easily decompose the combined problem into more managable sub-problems, using the chain rule:

$$\frac{\partial \bar{x}_{ee}}{\partial \bar{m}} = \frac{\partial \bar{x}_{ee}}{\partial \bar{q}} \, \frac{\partial \bar{q}}{\partial \bar{m}}$$

$$\Updownarrow$$

$$\delta \bar{x}_{ee} = \underbrace{\frac{\partial \bar{x}_{ee}}{\partial \bar{q}}}_{J} \, \underbrace{\underbrace{\frac{\partial \bar{q}}{\partial \bar{m}}}_{J_R} \, \delta \bar{m}}_{\delta \bar{q}}$$

$$\delta \bar{x}_{ee} = J \, \delta \bar{q}$$

Looks familiar!

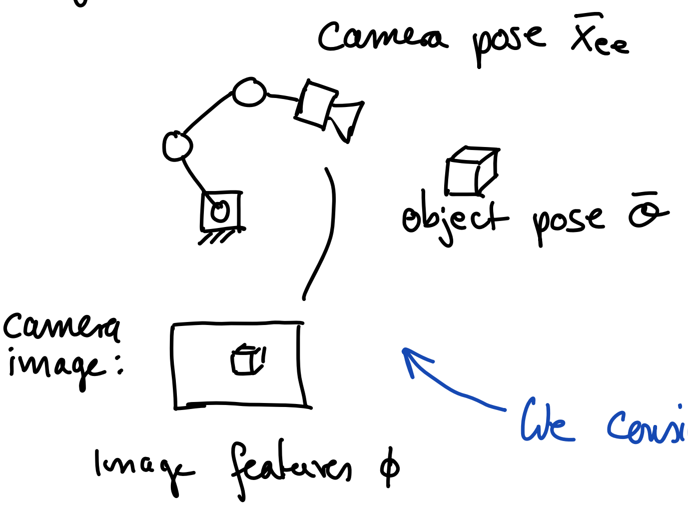
We only need to solve the sub-problem J_R ("tendon Jacobian")

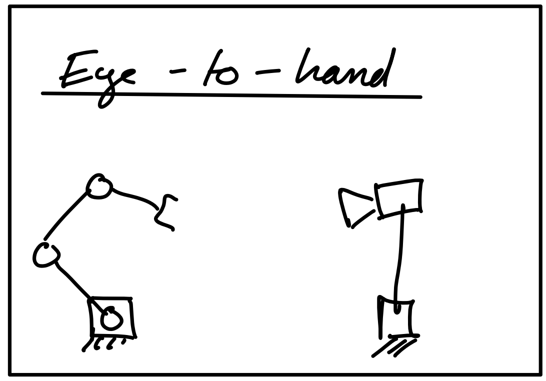
$$J_R = \frac{\partial \bar{q}}{\partial \bar{m}} = \begin{bmatrix} \frac{\partial q_1}{\partial m_1} & \frac{\partial q_1}{\partial m_2} \\ \frac{\partial q_2}{\partial m_1} & \frac{\partial q_2}{\partial m_2} \end{bmatrix} = \begin{bmatrix} \frac{r_{j1}}{r_{m1}} & 0 \\ 0 & \frac{r_{j2}}{r_{m2}} \end{bmatrix}$$

Visual Servoing: (A complex problem ...)

We want to make the robot move (this concerns $\bar{q}$) depending on what it sees (this concerns ϕ).

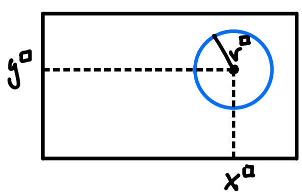
Eye-in-hand





We consider the eye-in-hand case.

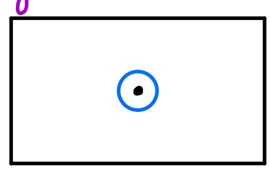
Camera Image



$$\phi = \begin{bmatrix} x^o \\ y^o \\ r^o \end{bmatrix}$$

ϕ : image features

Goal:



ϕ^* : desired image features

We want to find a configuration $\bar{q}$ that realizes the desired image features ϕ^*.
Thus, we want the camera to have a specific position relative to the object.

The space of image features ϕ and the space of configurations $\bar{q}$ are completely different things $\triangledown$

We will now create a relationship between these two

The camera pose $\bar{x}_{ee}$ depends on the configuration $\bar{q}$:

$$f(\bar{q}) = \bar{x}_{ee}$$

The image features ϕ depend on the camera pose $\bar{x}_{ee}$ (assuming that the object pose $\bar{o}$ is static):

$$c(\bar{x}_{ee}) = \phi$$

We can now combine the forward kinematic model f and the forward image model c to obtain a new model, predicting the image features ϕ based on the configuration $\bar{q}$:

$$\phi = c(f(\bar{q})) = c \circ f(\bar{q})$$

If we now wanted to know the configuration that realizes desired image features ϕ^*, we would need to invert this:

$$\bar{q} = (c \circ f)^{-1}(\phi^*)$$

We would encounter the <u>same problems</u>:

- difficult to obtain
- potentially not unique

We apply the <u>same strategy</u>:

- use local linear approximation as proxy $\frac{\partial \phi}{\partial \bar{q}}$
  - $\longrightarrow$ easy to obtain
  - $\longrightarrow$ easy to invert

Similar to before, we now <u>operate with steps</u>:

$$\delta \phi = \frac{\partial \phi}{\partial \bar{q}} \delta \bar{q}$$

We can easily <u>invert</u> this:

$$\delta \bar{q} = \left(\frac{\partial \phi}{\partial \bar{q}} \right)^{-1} \delta \phi$$

This allows us to control the image features by adjusting the joint positions

We can easily <u>decompose</u> the combined problem into more managable <u>sub-problems</u>, using the chain rule:

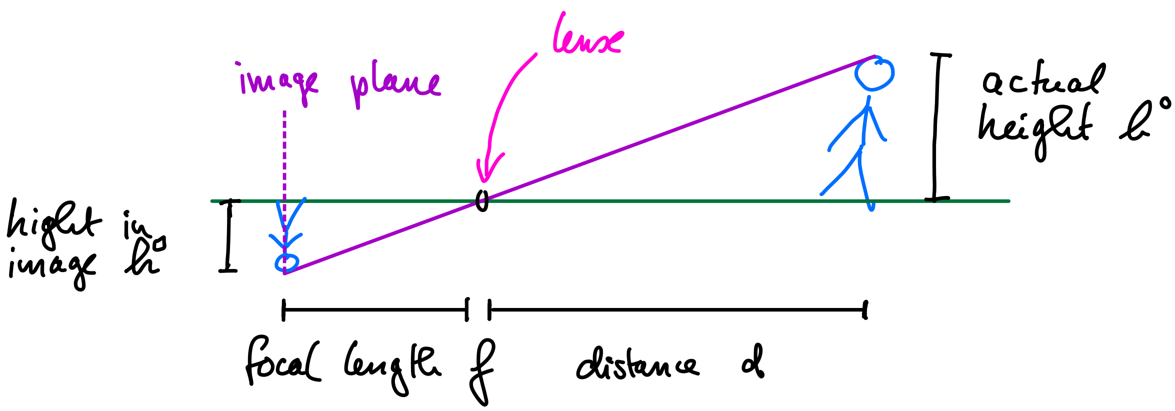
$$\frac{\partial \phi}{\partial \bar{q}} = \frac{\partial \phi}{\partial \bar{x}_{ee}} \frac{\partial \bar{x}_{ee}}{\partial \bar{q}}$$

$$\Updownarrow$$

$$\delta \phi = \underbrace{\frac{\partial \phi}{\partial \bar{x}_{ee}}}_{J_I} \underbrace{\underbrace{\frac{\partial \bar{x}_{ee}}{\partial \bar{q}}}_{J} \delta \bar{q}}_{\delta \bar{x}_{ee}}$$

We only need to solve the sub-problem J_I ("Image Jacobian")

First, we create a forward image model, based on the camera obscura. (This model is not perfect, but good enough for our purposes).



forward image model

$$l^\square = c(l^\circ)$$

ϕ image feature

object feature $\bar{o}$

Simple proportional relationship between the side lengths of the two triangles:

$$\frac{l^\square}{f} = \frac{l^\circ}{d}$$

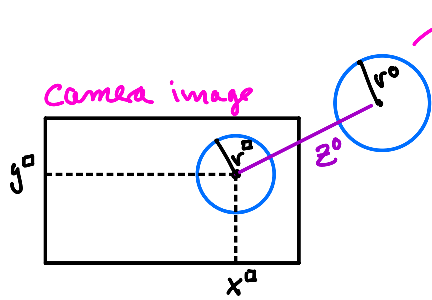
$$\Downarrow \quad \boxed{l^\square = \frac{f}{d} l^\circ}$$

forward image model

We can apply the same logic to the image features of a circle (i.e., we also use the simple proportional relationship, found in the camera obsura model)

$$\phi = \begin{bmatrix} x^\square \\ y^\square \\ r^\square \end{bmatrix}$$

image features of circle



$$\bar{o} = \begin{bmatrix} x^\circ \\ y^\circ \\ z^\circ \end{bmatrix}$$

object pose wrt. to camera frame

$$\phi = c(\bar{o})$$

$$\boxed{\begin{bmatrix} x^\square \\ y^\square \\ r^\square \end{bmatrix} = \frac{f}{z^\circ} \begin{bmatrix} x^\circ \\ y^\circ \\ r^\circ \end{bmatrix}}$$

forward image model

Using this forward image model, we can already compute the following:

- focal length:

$$r^\square = \frac{f r^\circ}{z^\circ} \longrightarrow \boxed{f = \frac{r^\square z^\circ}{r^\circ}}$$

- distance betw. camera and object:

$$\boxed{z^\circ = \frac{f r^\circ}{r^\square}}$$

We now want to linearize the model.
It might help to imagine, that we split up the forward image model into separate functions, one for each image feature dimension, and compute the partial derivatives for each of them individually:

$$x^a = C_{x^a}(\bar{o}) = \frac{f x^o}{z^o}$$

$$y^a = C_{y^a}(\bar{o}) = f\frac{y^o}{z^o}$$

$$r^a = C_{r^a}(\bar{o}) = f\frac{r^o}{z^o}$$

$$J_I = \left[\frac{\partial c_i}{\partial \bar{o}_j}\right] = \begin{bmatrix} f/z^o & 0 & -(f x^o)/z^{o2} \\ 0 & f/z^o & -(f y^o)/z^{o2} \\ 0 & 0 & -(f r^o)/z^{o2} \end{bmatrix}$$

Important insight about obj. movement and camera movement:

$$\frac{\partial \phi}{\partial \bar{o}} = - \frac{\partial \phi}{\partial \bar{x}_{ec}}$$

Example:
obj moves right; cam remains still is the same as cam moves left, while object remains still.

$$J_I = \begin{bmatrix} -f/z^o & 0 & f x^o/z^{o2} \\ 0 & -f/z^o & f y^o/z^{o2} \\ 0 & 0 & f r/z^{o2} \end{bmatrix}$$

Problem:
This depends on infos about the object (e.g. z^o) ↯

We use the forward image model to replace unknown terms by known terms.

We already computed z° above:

$$z^{\circ} = \frac{f r^{\circ}}{r^{\square}}$$

We can compute x° in a similar way:

$$x^{\square} = \frac{f x^{\circ}}{z^{\circ}} \quad \rightarrow \quad x^{\circ} = \frac{x^{\square} z^{\circ}}{f}$$

(from the model)

We now replace z° by the expression above:

$$x^{\circ} = \frac{x^{\square}\left(\frac{f r^{\circ}}{r^{\square}}\right)}{f} \quad \Leftrightarrow \quad x^{\circ} = \frac{x^{\square} r^{\circ}}{r^{\square}}$$

Similar to how we did it with x°, we can do this also with y° and r°

$$y^{\circ} = \frac{y^{\square} r^{\circ}}{r^{\square}} \quad , \quad r^{\circ} = \frac{r^{\circ 2}}{r^{\square}}$$

We now insert these terms in the elements of the image Jacobian:

$$\underset{\text{(from Jacobian)}}{-\frac{f}{z^{\circ}}} = -f \cdot \frac{r^{\square}}{\underbrace{r^{\circ} f}_{(z^{\circ}\text{ from above})}} = -\frac{r^{\square}}{r^{\circ}}$$

$$\frac{f x^\circ}{z^{\circ 2}} = f x^\circ \left(\frac{r^\square}{r^\circ f}\right)^2 = f \, \frac{x^\square r^\circ}{r^\square} \, \frac{r^{\square 2}}{r^{\circ 2} f^2} = \frac{x^\square r^\square}{r^\circ f}$$

(from Jacobian) (z° from above) (x° from above)

The same way, we obtain these other terms:

$$\frac{f y^\circ}{z^{\circ 2}} = \frac{y^\square r^\square}{r^\circ f} \qquad\qquad \frac{f r^\circ}{z^{\circ 2}} = \frac{r^{\square 2}}{r^\circ f}$$

Now we replace all these terms in the image Jacobian:

$$J_I = \begin{bmatrix} -\frac{r^\square}{r^\circ} & 0 & \frac{x^\square r^\square}{r^\circ f} \\ 0 & -\frac{r^\square}{r^\circ} & \frac{y^\square r^\square}{r^\circ f} \\ 0 & 0 & \frac{r^{\square 2}}{r^\circ f} \end{bmatrix}$$

We only need to know the real circle radius r°, the focal length f, and the image features ϕ.

Now we can put it all back together:

$$\delta\phi = \frac{\partial \phi}{\partial \bar{q}} \delta \bar{q} = \underbrace{\frac{\partial \phi}{\partial \bar{x}_{ee}}}_{J_I} \underbrace{\frac{\partial \bar{x}_{ee}}{\partial \bar{q}}}_{J} \delta \bar{q}$$

$$\Downarrow \quad \delta\phi = J_I J \, \delta\bar{q}$$

$$\Downarrow \quad \delta\bar{q} = (J_I J)^{-1} \delta\phi$$

We can now use this to take steps in configuration space $\delta \bar{q}$ that minimize errors in image feature space $e_\phi = (\phi^* - \phi)$:

$$\delta \phi = e_\phi = (\underbrace{\phi^*}_{\text{desired features}} - \underbrace{\phi)}_{\text{actual features}}$$

$$\boxed{\delta \bar{q} = \gamma \, (J_I J)^{-1} \, \delta \phi}$$

... schön!