Bearing

Using a sphere with center O $(0, 0, Z_0)$

$\Rightarrow$ Image of the sphere = centered circle

$$\begin{array}{ll}
\mathbf{L}_{x_c}^T & = [\ -1/Z_c \quad 0 \quad 0 \quad 0 \quad -1-r^2 \quad 0\ ] \\
\mathbf{L}_{y_c}^T & = [\ 0 \quad -1/Z_c \quad 0 \quad 1+r^2 \quad 0 \quad 0\ ] \\
\mathbf{L}_{\mu}^T & = [\ 0 \quad 0 \quad 2r^2/Z_c \quad 0 \quad 0 \quad 0\ ]
\end{array}$$

with $Z_c = (Z_0^2 - R^2)/Z_0$ and $r^2 = R^2/(Z_0^2 - R^2)$.

$$\Rightarrow S^* = \begin{pmatrix} 0 & -z_0 & 0 \\ z_0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}_{|C} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}_{|O}$$

1) Modeling issues

2) Classification of the vision-based tasks

$\Rightarrow$ 3) Control issues

4) Use of other exteroceptive sensors

Control

Regulation of a task function: $\mathbf{e}(\mathbf{r}(t)) = \mathbf{C}\ (\mathbf{s}(\mathbf{r}(t)) - \mathbf{s}^*)$

Numerous solutions:

- P, PI, PID controller [Weiss 87]
- Non linear control law [Hashimoto 93, Reyes 98]
- Optimal control (LQ, LQG) [Papanikilopoulos 93, Hashimoto 96]
- Predictive controller [Gangloff 98]
- Robust controller H_∞ [Khadraoui 96]

Visual task function

With k visual features $\mathbf{s}$, one constraints m $(= n - N \leq k)$ camera dof

using the **visual task function** $\mathbf{e}_1(\mathbf{r}(t)) = \mathbf{C}\,(\mathbf{s}(\mathbf{r}(t)) - \mathbf{s}^*)$

where $\mathbf{C}$ is a $m \times k$ combination matrix of full rank m.

$\Rightarrow$ It is possible to consider a supplementary task (trajectory following, joint limits avoidance, etc.).

Problem : How to combine both tasks ?

- $\mathbf{e}_1$: primary task
- $\mathbf{e}_2$: secondary task, expressed as a cost function to be minimized under the constraint that $\mathbf{e}_1$ is satisfied.

Global task function

A task function $\mathbf{e}$ minimizing the objective function h_s under the constraint $\mathbf{e}_1 = 0$ is given by :

$$\begin{aligned} \mathbf{e} &= \mathbf{W}^+\mathbf{e}_1 + (\mathbb{I}_n - \mathbf{W}^+\mathbf{W})\, \mathbf{g}_s^T \\ &= \mathbf{W}^+\mathbf{C}\,(\mathbf{s} - \mathbf{s}^*) + (\mathbb{I}_n - \mathbf{W}^+\mathbf{W})\, \mathbf{g}_s^T \end{aligned}$$

where :

- $\mathbf{g}_s$ = gradient of h_s ($\mathbf{g}_s = \frac{\partial h_s}{\partial \mathbf{r}}$)
- $\mathbf{W}$ is a $m \times n$ matrix of full rank m such that

  $\text{Ker } \mathbf{W} = \text{Ker } \mathbf{J}_1 = \text{Ker } \mathbf{L}_\mathbf{s}^T$

$$\Rightarrow (\mathbb{I}_n - \mathbf{W}^+\mathbf{W})\, \mathbf{g}_s^T \in \text{Ker } \mathbf{L}_\mathbf{s}^T \;,\; \forall \mathbf{g}_s^T$$

- if $m = n$, $\mathbf{W} = \mathbb{I}_n$, $\mathbf{e} = \mathbf{e}_1 = \mathbf{C}\,(\mathbf{s} - \mathbf{s}^*)$

Control law

Since we have

$$\dot{\mathbf{e}} = \frac{\partial \mathbf{e}}{\partial \mathbf{r}}\, \mathcal{T}_q + \frac{\partial \mathbf{e}}{\partial t} \quad \text{where} \quad \begin{cases} \mathcal{T}_q = \mathcal{T}_c \text{ for eye-in-hand system} \\ \mathcal{T}_q = -\mathcal{T}_o \text{ for eye-to-hand system} \end{cases}$$

we obtain ideally for an exponential decrease of $\mathbf{e}$ ($\dot{\mathbf{e}} = -\lambda\, \mathbf{e}$)

$$\mathcal{T}_q = \left(\frac{\partial \mathbf{e}}{\partial \mathbf{r}}\right)^{-1} (-\lambda\, \mathbf{e} - \frac{\partial \mathbf{e}}{\partial t})$$

$$\text{with } \frac{\partial \mathbf{e}}{\partial \mathbf{r}} = \mathbf{W}^{+}\mathbf{C}\mathbf{L}_{\mathbf{s}}^{T} + (\mathbb{I}_n - \mathbf{W}^{+}\mathbf{W})\, \frac{\partial \mathbf{g}_s^T}{\partial \mathbf{r}}$$

Since $\mathbf{L}_{\mathbf{s}}^{T}$ and $\frac{\partial \mathbf{e}_1}{\partial t}$ are not perfectly known, one uses

$$\mathcal{T}_q = \left(\frac{\widehat{\partial \mathbf{e}}}{\partial \mathbf{r}}\right)^{-1} (-\lambda\, \mathbf{e} - \frac{\widehat{\partial \mathbf{e}}}{\partial t})$$

Stability analysis

Behavior of the closed-loop system :

$$\dot{\mathbf{e}} = -\lambda \frac{\partial \mathbf{e}}{\partial \mathbf{r}} \left(\widehat{\frac{\partial e}{\partial r}} \right)^{-1} \mathbf{e} \; - \; \frac{\partial \mathbf{e}}{\partial \mathbf{r}} \left(\widehat{\frac{\partial \mathbf{e}}{\partial \mathbf{r}}} \right)^{-1} \widehat{\frac{\partial \mathbf{e}}{\partial t}} + \frac{\partial \mathbf{e}}{\partial t}$$

If $\frac{\partial \mathbf{e}}{\partial t} = \widehat{\frac{\partial \mathbf{e}}{\partial t}} = 0$, $\|\mathbf{e}\|$ always decreases (global stability) if

$$\frac{\partial \mathbf{e}}{\partial \mathbf{r}} \left(\widehat{\frac{\partial \mathbf{e}}{\partial \mathbf{r}}} \right)^{-1} > 0$$

To suppress tracking errors and obtain the desired behavior $\dot{\mathbf{e}} = -\lambda \mathbf{e}$:

$$\widehat{\frac{\partial \mathbf{e}}{\partial \mathbf{r}}} = \frac{\partial \mathbf{e}}{\partial \mathbf{r}} \quad \text{and} \quad \widehat{\frac{\partial \mathbf{e}}{\partial t}} = \frac{\partial \mathbf{e}}{\partial t}$$

In practice

- If $k = m = n$, $\mathbf{W} = \mathbf{C} = \mathbb{I}_m$, $\mathbf{e} = \mathbf{e}_1 = \mathbf{s} - \mathbf{s}^* \Rightarrow \frac{\partial \mathbf{e}}{\partial \mathbf{r}} = \mathbf{L}_{\mathbf{s}}^T, \widehat{\frac{\partial \mathbf{e}}{\partial \mathbf{r}}} = \widehat{\mathbf{L}_{\mathbf{s}}^T}$

$$\mathcal{T}_q = -\lambda\, \widehat{\mathbf{L}_{\mathbf{s}}^T}^{-1} (\mathbf{s} - \mathbf{s}^*) - \widehat{\mathbf{L}_{\mathbf{s}}^T}^{-1} \frac{\widehat{\partial \mathbf{s}}}{\partial t} \quad \text{stable if} \quad \mathbf{L}_{\mathbf{s}}^T\, \widehat{\mathbf{L}_{\mathbf{s}}^T}^{-1} > 0$$

- If $k > m$, $\mathbf{C} = \mathbf{W}\widehat{\mathbf{L}_{\mathbf{s}}^T}^{+}_{|\mathbf{s}=\mathbf{s}^*}$, $\widehat{\frac{\partial \mathbf{e}}{\partial \mathbf{r}}} = \mathbb{I}_n$,

$$\mathcal{T}_q = -\lambda\, \mathbf{e} - \frac{\widehat{\partial \mathbf{e}}}{\partial t} \quad \text{stable if} \quad \mathbf{W}\widehat{\mathbf{L}_{\mathbf{s}}^T}^{+}_{|\mathbf{s}=\mathbf{s}^*} \mathbf{L}_{\mathbf{s}}^T \mathbf{W}^+ > 0 \ \ (\text{only around } \mathbf{s}^*)$$

If translational dof are controlled and 2D visual features are used, an estimation $\widehat{\mathbf{P}}$ or $\widehat{\mathbf{P}^*}$ is necessary to compute $\widehat{\mathbf{L}_{\mathbf{s}}^T}$

A simple case $k = m = n = 2$

Case of a pan-tilt camera observing a point : $\mathbf{s} = (x\ y)^T \quad \mathbf{s}^* = (0\ 0)^T$

$$\dot{\mathbf{e}} = \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} xy & -(1+x^2) \\ 1+y^2 & -xy \end{pmatrix} \begin{pmatrix} \omega_x \\ \omega_y \end{pmatrix}$$

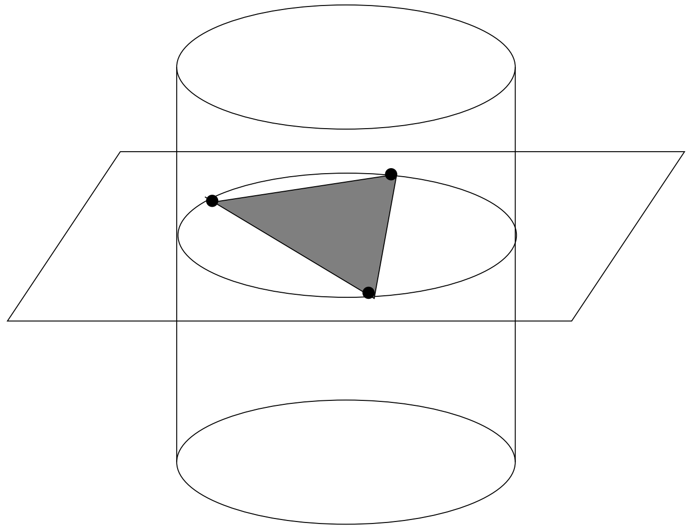
$$\mathcal{T}_c = -\lambda\, \widehat{\mathbf{L}_{\mathbf{s}}^T}^{-1} (\mathbf{s} - \mathbf{s}^*) \Leftrightarrow \begin{pmatrix} \omega_x \\ \omega_y \end{pmatrix} = -\frac{\lambda}{1+x^2+y^2} \begin{pmatrix} y \\ -x \end{pmatrix}$$

If no error occurs, $\dot{\mathbf{s}} = -\lambda \mathbf{s}$: trajectory = straight line in the image

Control when $k = m = n = 6$

$$\mathbf{C} = \mathbf{W} = \mathbb{I}_6 \quad \Rightarrow \quad \mathcal{T}_q = -\lambda\, \widehat{\mathbf{L}_{\mathbf{s}}^T}^{-1} (\mathbf{s} - \mathbf{s}^*) \text{ stable if } \mathbf{L}_{\mathbf{s}}^T\, \widehat{\mathbf{L}_{\mathbf{s}}^T}^{-1} > 0$$

- **Impossible with only 2D visual features**



Using 3 points ($\mathbf{L}_{\mathbf{s}}^T$ is 6×6)

- pose ambiguity (4 solutions)

- possible singularity of $\mathbf{L}_{\mathbf{s}}^T$

Control when $k = m = n = 6$

- **Possible with 3D visual features**

For instance, if $\mathbf{s} = \begin{pmatrix} {}^{c^*}\mathbf{P}_c \\ \theta\mathbf{u} \end{pmatrix}, \mathcal{T}_c = \begin{pmatrix} \mathbf{V} \\ \omega \end{pmatrix} = -\lambda \begin{pmatrix} {}^{c}\mathbf{A}_{c^*}\,{}^{c^*}\mathbf{P}_c \\ \theta\mathbf{u} \end{pmatrix}$

Advantages

- $\mathbf{L}_{\mathbf{s}}^T$ block-diagonal and never singular
- Translational and rotational motions decoupled
- Camera trajectory : straight line in 3D space

Drawbacks

- No control in the image (the target may get out of the image)

Control when $k = m = n = 6$: 2 1/2 D visual servoing

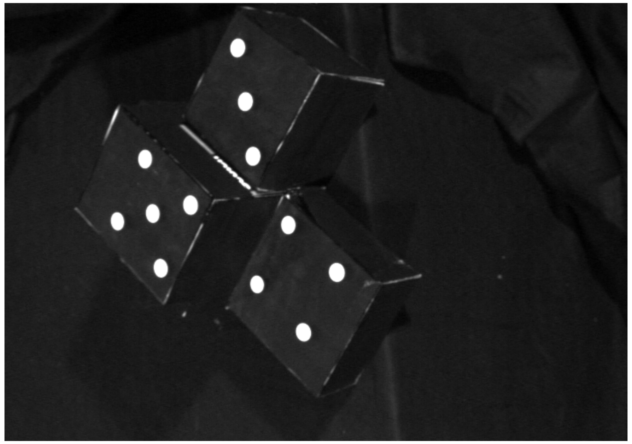
Idea : Combine 2D image data and 3D data

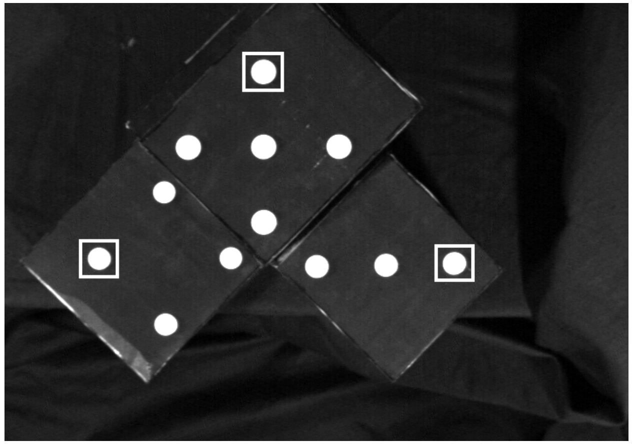
$$\mathbf{s} = \begin{pmatrix} x \\ y \\ \log Z \\ \theta u_x \\ \theta u_y \\ \theta u_z \end{pmatrix} \begin{matrix} \left.\begin{matrix} \\ \\ \end{matrix}\right\} \text{image point coordinates} \\ \to \text{rel. depth} \\ \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} \text{rotation to realize} \end{matrix} \Rightarrow \mathbf{L}_{\mathbf{s}}^{T} \ \textbf{triangular} \ \text{and never singular}$$

Advantages :

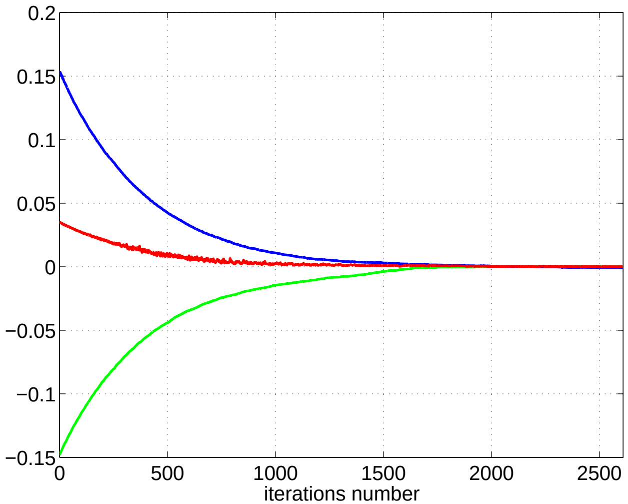
- Decoupled control scheme (image point trajectory : straight line)
- Analytical conditions for global stability possible in presence of calibration errors
- No 3D CAD model needed, only one scalar unknown

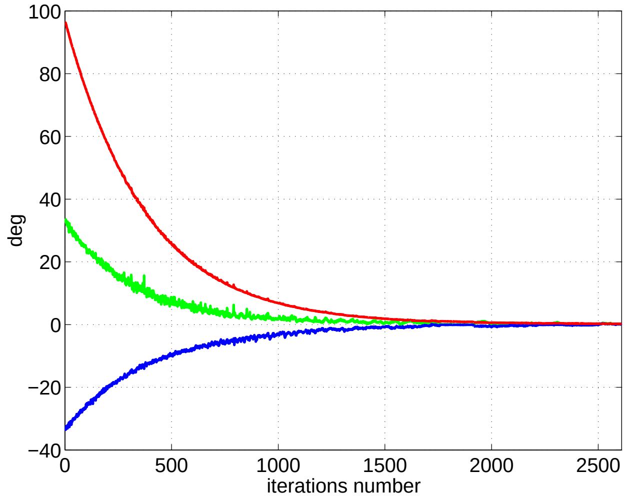
Results

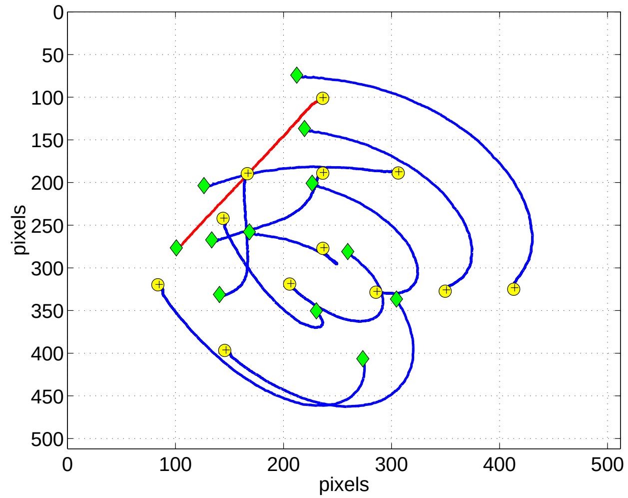




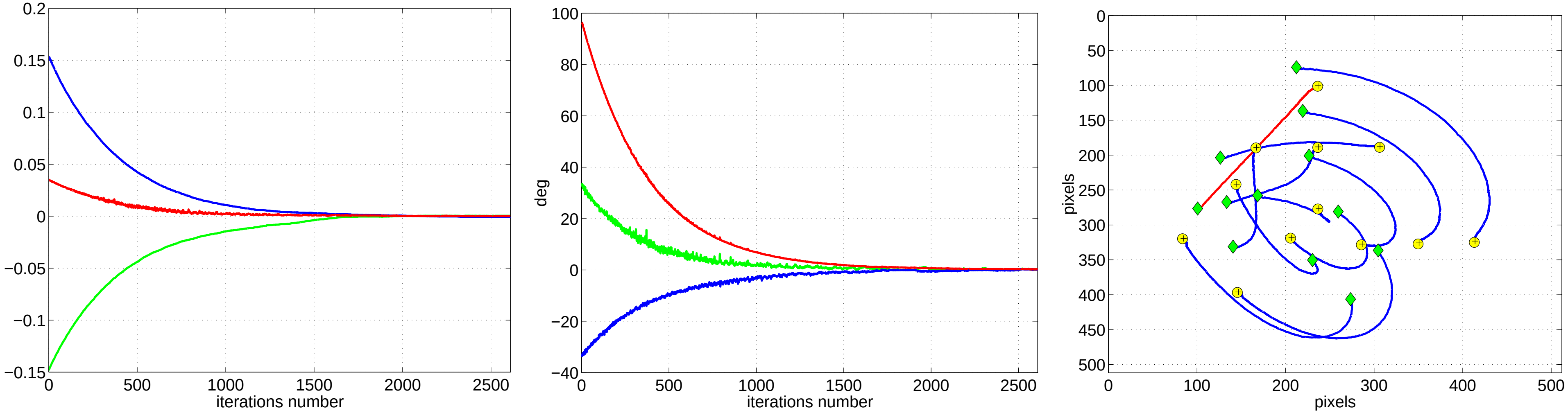
Coarse calibration



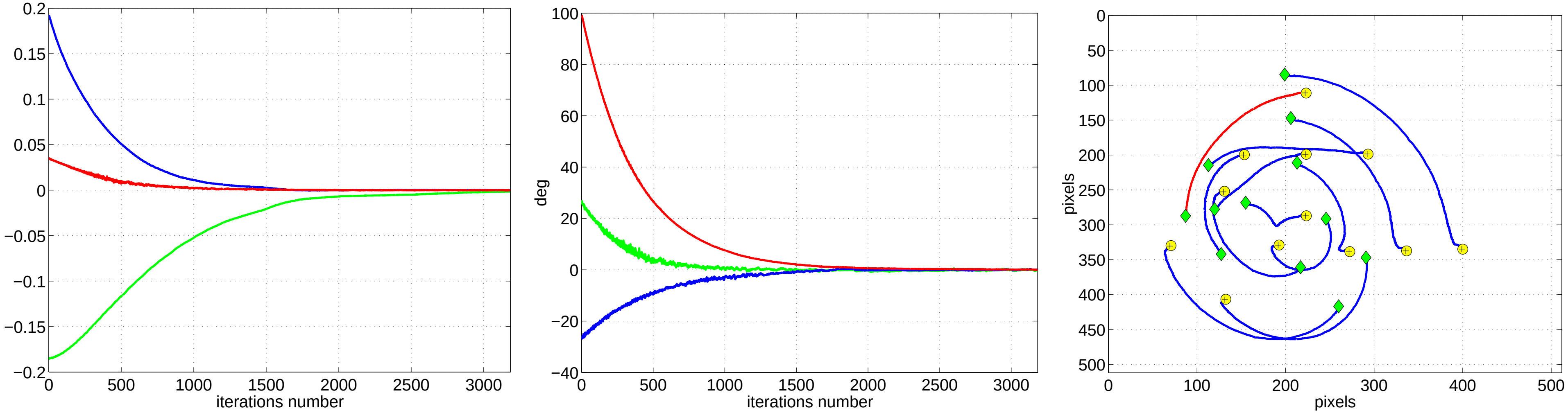




Coarse calibration



Bad calibration



Other 2 1/2 D VS scheme

$$\mathbf{s} = \begin{pmatrix} {}^{c^*}\mathbf{P}_{cx} \\ {}^{c^*}\mathbf{P}_{cy} \\ {}^{c^*}\mathbf{P}_{cz} \\ x \\ y \\ \theta \end{pmatrix} \begin{array}{l} \left.\begin{array}{c} \text{translation} \\ \text{to} \\ \text{realize} \end{array}\right\} \\ \left.\begin{array}{c} \text{image point} \\ \text{coordinates} \end{array}\right\} \\ \to \text{orientation} \end{array} \Rightarrow \mathbf{L}_{\mathbf{s}}^T = \begin{pmatrix} {}^{c^*}\mathbf{A}_c & \mathbf{0}_3 \\ \frac{1}{Z}\mathbf{L}_{\omega v} & \mathbf{L}_{\omega} \end{pmatrix}$$

Advantages:

- Camera trajectory : straight line in 3D space
- Trajectory in the image of the selected point : straight line

Drawback:

J only block-triangular

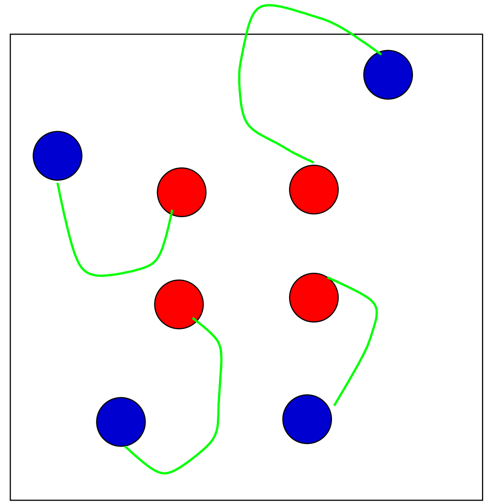
$\Rightarrow$ analytical conditions for global stability difficult to obtain

Control when $k > n, n = 6$ using 2D visual features

- Control law $\mathcal{T}_q = -\lambda \, \widehat{\mathbf{L}_{\mathbf{s}}^T}^{+}_{|\mathbf{s}=\mathbf{s}^*} \, (\mathbf{s} - \mathbf{s}^*)$

  stable if $\widehat{\mathbf{L}_{\mathbf{s}}^T}^{+}_{|\mathbf{s}=\mathbf{s}^*} \, \mathbf{L}_{\mathbf{s}}^T > 0$ (only around $\mathbf{s}^*$)

  No real control of the image trajectories

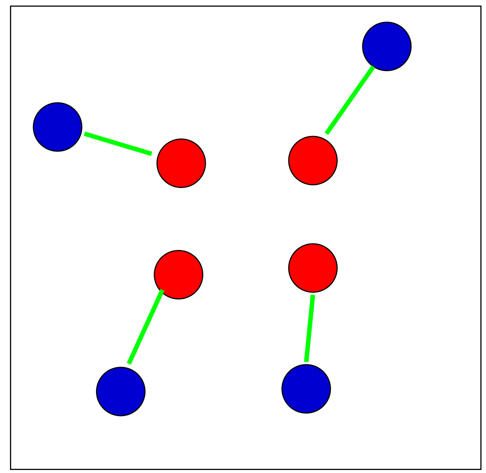


- Control law $\mathcal{T}_q = -\lambda \, \widehat{\mathbf{L}_{\mathbf{s}}^T}^{+} \, (\mathbf{s} - \mathbf{s}^*)$

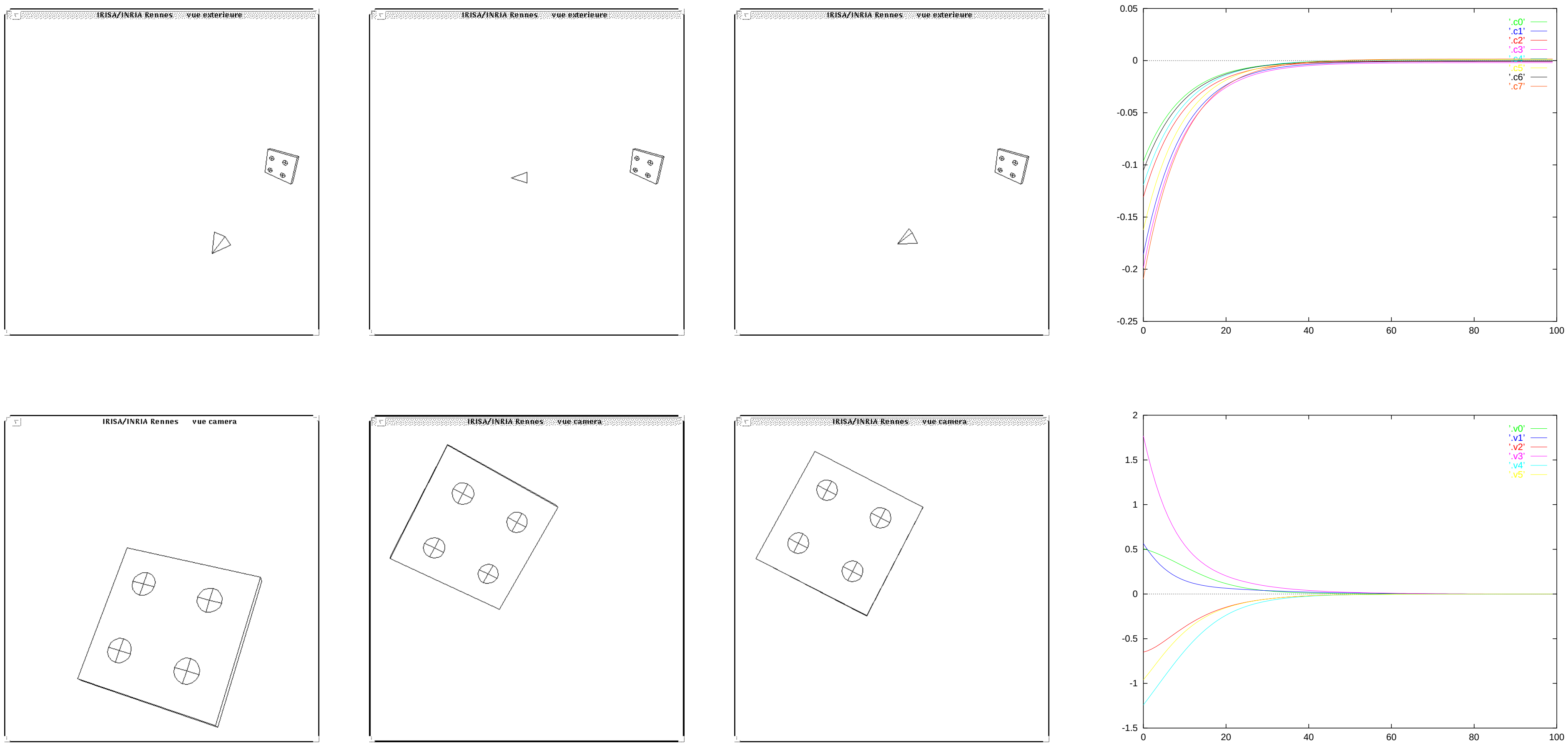
  tries to ensure $\dot{\mathbf{s}} = -\lambda \, (\mathbf{s} - \mathbf{s}^*)$

  Possible local minima

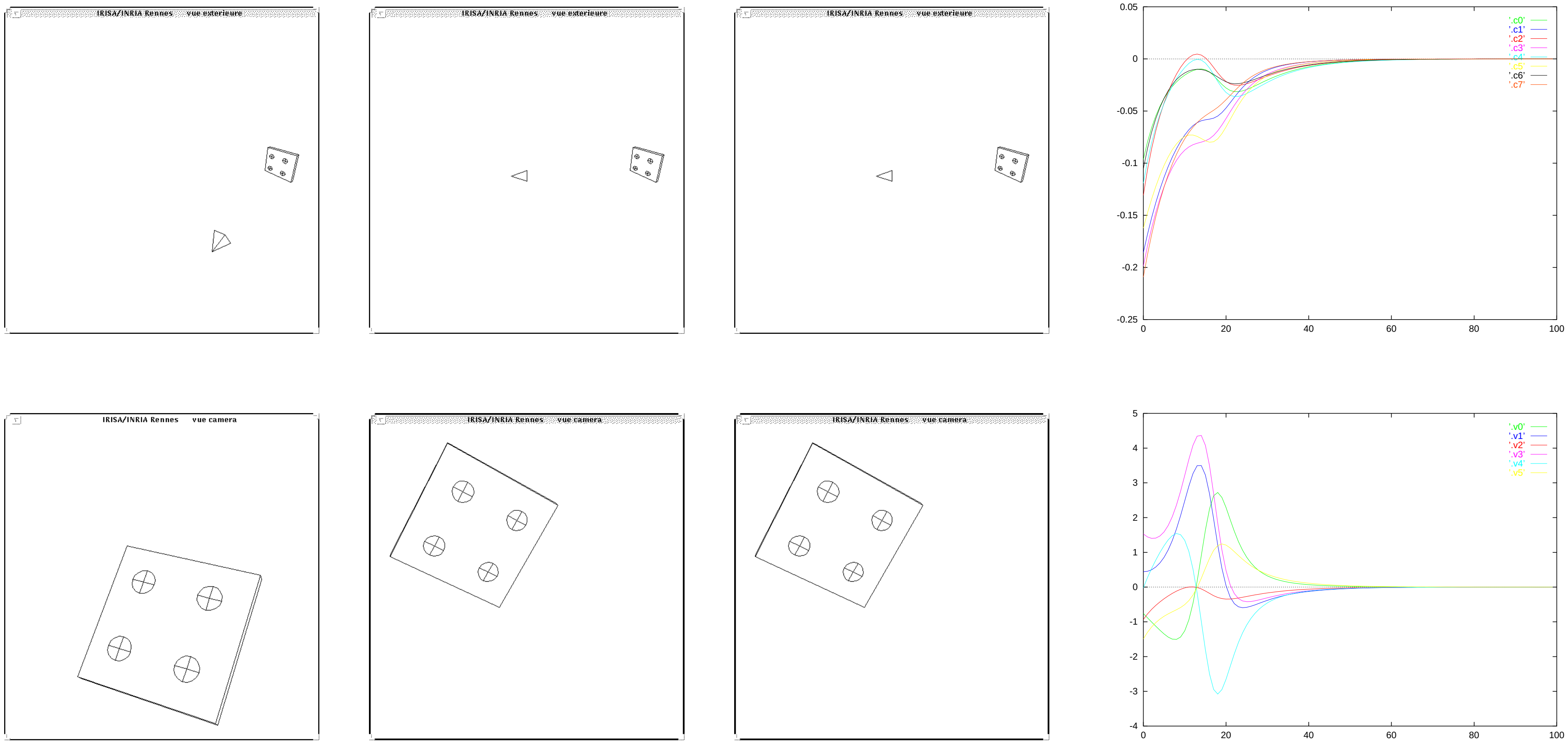
  because $\mathbf{C}$ $(= \widehat{\mathbf{L}_{\mathbf{s}}^T}^{+})$ is not constant



Reaching a local minimum using $\widehat{\mathbf{L}_{\mathbf{s}}^{T}}^{+}$

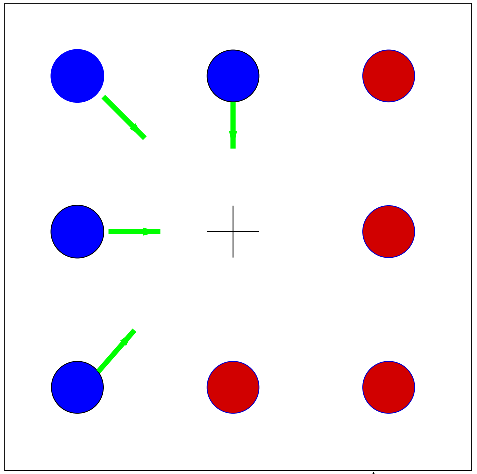


Reaching the global minimum using $\widehat{\mathbf{L}_{\mathbf{s}}^{T}}^{+}_{|\mathbf{s}=\mathbf{s}^*}$

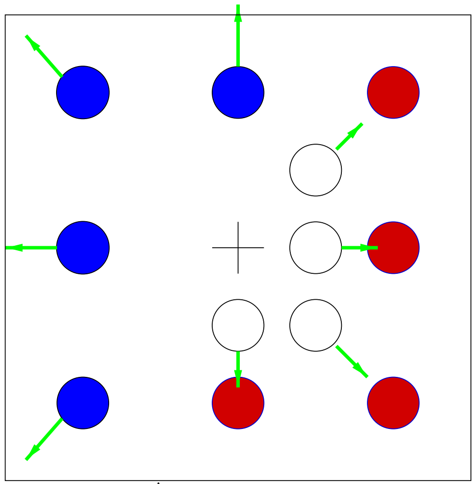


Reaching a singularity of $\mathbf{L}_{\mathbf{s}}^T$

Example : rotation of 180° around optical axis
$\mathbf{s}$ composed of image points coordinates



Using $\widehat{\mathbf{L}_{\mathbf{s}}^T}^+$



Using $\widehat{\mathbf{L}_{\mathbf{s}}^T}^+_{|\mathbf{s}=\mathbf{s}^*}$

At singularity, rank $\mathbf{L}_{\mathbf{s}}^T = 2$.

No problem if $\mathbf{s}$ is composed of 2D straight lines parameters

Target tracking

PI Controller

Integral term classical in Automatic Control to reduce tracking errors

If $\mathbf{I}_k$ is the estimation of $\frac{\partial \mathbf{e}_1}{\partial t}$ at iteration k, we have :

$$\begin{aligned} \mathbf{I}_{k+1} &= I_k + \mu\, \mathbf{e}_{1k} \;\; \text{with } \mathbf{I}_0 = 0 \\ &= \mu \sum_{j=0}^{k} \mathbf{e}_{1j} \end{aligned}$$

Efficient to track a target at constant velocity :

$$\mathbf{I}_{k+1} = \mathbf{I}_k \text{ if } \mathbf{e}_{1k} = 0$$

Target tracking by estimating the target velocity

If it is possible to measure the camera velocity, we get:

$$\widehat{\frac{\partial \mathbf{e}_1}{\partial t}} = \widehat{\dot{\mathbf{e}}_1} - \left(\widehat{\frac{\partial \mathbf{e}_1}{\partial \mathbf{r}}} \right) \mathcal{T}_c$$

with

$$\begin{cases} \widehat{\frac{\partial e_1}{\partial r}} = C \widehat{L_s^T} \\ \\ \widehat{\dot{e}_{1k}} = \frac{e_k - e_{k-1}}{\Delta t} \end{cases}$$

A Kalman filter may then be used.

1) Modeling issues

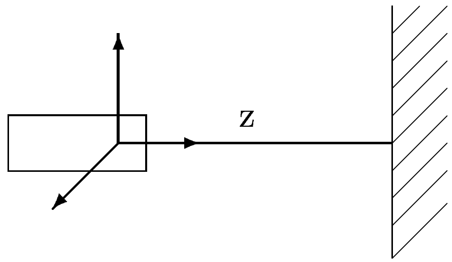
2) Classification of the vision-based tasks

3) Control issues

$\Rightarrow$ 4) Use of other exteroceptive sensors

Proximetric sensor

A narrow beam proximetric sensor provides the range z from the sensor to the nearest object in the sensor axis.



$$\mathbf{L}_Z^T = \begin{pmatrix} 0 & 0 & -1 & 0 & 0 & 0 \end{pmatrix}$$

if the object surface is perpendicular to the sensor axis.

Applications :

- Obstacles avoidance
- Virtual link

Plane-to-plane contact

$$\mathbf{L}^T_{Z_i|R_O} = \begin{pmatrix} 0 & 0 & -1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbb{I}_3 & -\tilde{\mathbf{S}}_i \\ 0 & \mathbb{I}_3 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & 0 & -1 & -Y_i & X_i & 0 \end{pmatrix}$$

where $\mathbf{S}_i = (X_i, Y_i, 0)_{|R_O}$.

$$\Rightarrow \mathcal{S}^* = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$