Dynamic visual servoing for surgical robotics
Control II, Thursday, Sept. 10, 14h30-16h

jacques.gangloff@unistra.fr
http://eavr.u-strasbg.fr/
Outline

- Introduction
  - Definitions
  - History & current challenges
  - Classification
  - Examples of tasks
- Measurement estimation
  - Dementhon method
  - Image Jacobian method
Outline

- Control
  - Kinematic control
    - 2D
    - 3D
  - Dynamic control
    - Image acquisition model
    - Manipulator model
    - Typical dynamic visual loop
- Conclusion
1.1 Definitions

- **Exteroceptive sensor**: Sensor that provides measurements on the environment of the system.
  - Examples: camera, ultrasound range sensor, medical imaging system, ...

- **Visual servoing**: Control of a mechanical actuated system using exteroceptive visual informations.
## 1.2 History

<table>
<thead>
<tr>
<th>Date</th>
<th>Authors</th>
<th>Task</th>
<th>Sampling period</th>
</tr>
</thead>
<tbody>
<tr>
<td>1973</td>
<td>Shirai &amp; Inoue (Tokyo, JP)</td>
<td>Grasping</td>
<td>10s</td>
</tr>
<tr>
<td>1979</td>
<td>Prajoux (CNRS, FR)</td>
<td>Tracking</td>
<td>1s</td>
</tr>
<tr>
<td>1984</td>
<td>Weiss (CMU, USA)</td>
<td>Tracking</td>
<td>33ms</td>
</tr>
<tr>
<td>1992</td>
<td>Chaumette (INRIA, FR)</td>
<td>Tracking</td>
<td>40ms</td>
</tr>
<tr>
<td>1996</td>
<td>Corke (CSIRO, AU)</td>
<td>Tracking</td>
<td>16ms</td>
</tr>
<tr>
<td>1998</td>
<td>Gangloff (LSIIT, FR)</td>
<td>Tracking</td>
<td>8ms</td>
</tr>
<tr>
<td>1999</td>
<td>Nakabo (U. Tokyo, JP)</td>
<td>Tracking</td>
<td>1ms</td>
</tr>
</tbody>
</table>
1.2 Some current challenges
(situation in France)

- Robust visual servoing (INRIA Rennes, INRIA Sophia, ISIR Paris, LSIIT Strasbourg)
- Visual servoing of parallel robots (LASMEA, Clermont Ferrand)
- Physiological motion compensation (LSIIT Strasbourg, LIRMM Montpellier)
- Ultrasound imaging visual servoing (INRIA Rennes, LIRMM Montpellier, ISIR Paris)
- CT-scan and MRI visual servoing (LSIIT Strasbourg)
1.3 Classification

- Criteria:
  - Position of the camera(s)
  - Type of control signal
  - Type of measurement
  - Bandwidth
1.3.1 Position of the camera(s)

Eye in hand

Eye to hand
1.3.2 Type of control signal

*Indirect visual servoing*: robot joint positions are controlled by the controller provided by the robot manufacturer. The vision loop controller provides velocity control signals which are converted in position setpoints by the robot controller. This type of visual servoing is used when the sampling period is slow thus corresponding to a slow response time.
1.3.2 Type of control signal

**Direct visual servoing**: robot joints torques or velocities are controlled. Torque or velocity control signals are issued by the controller of the visual loop. They are sent directly to the drives. This architecture is used for high bandwidth visual servoing with a high sampling rate.
1.3.3 Type of measurement

*Position based (3D) visual servoing*: reference and measurement are poses. To reconstruct the measurement it is necessary to have a model of the object of interest.
1.3.3 Type of measurement

*Image based (2D) visual servoing:* reference and measurement are feature coordinates. No 3D reconstruction. Not necessary to have a model of the object of interest.
1.3.3 Type of measurement

Hybrid (2,5D) visual servoing: some DoFs are controlled in the 3D space and others are controlled in the image plane. The goal is to obtain the best decoupling between the 2 loops.
1.3.3 Type of measurement

\[ \dot{\mathbf{F}}^* \]

\( \dot{\mathbf{F}} \)

Controller

(\text{controller +}) \text{ drives} + \text{ robot + camera}

Velocity field in the image

Optical flow

Image

\( \frac{d}{dt} \text{ visual servoing} \) : the reference and the measurement are velocities in the image.
1.3.4 Bandwidth

- Kinematic visual servoing (slow)
  - Continuous time model
  - Robot = pure integrator
  - Controller is a gain
  - Dynamics of vision are neglected
  - Indirect control scheme
1.3.4 Bandwidth

- Dynamic visual servoing (fast)
  - Discrete time model
  - Detailed electromecanical modeling of the robot
  - Detailed modeling of the camera (delays, sampling)
  - Advanced control (predictive, robust, ...)
  - Direct control scheme (velocity or torque)
1.4 Examples

- Tasks:
  - Tracking
  - Grasping
  - Insertion
  - Positioning
1.4.1 Tracking

- 6 DoF fast target tracking (Gangloff et al., Strasbourg University):
  - Eye in hand, 2D, direct
  - 120 Hz
  - Predictive control
  - Velocity control signal
1.4.1 Tracking

- 2 DoF fast target tracking (Nakabo et al., Tokyo University):
  - Eye in hand, 2D, direct
  - 1 kHz
  - Vision chip
  - Torque control signal
1.4.1 Tracking

- 2 DoF tracking of liver (Ott et al., Strasbourg University):
  - Eye in hand, 2D, direct
  - 25 Hz
  - Repetitive control
  - Velocity control signal
  - The robot is an actuated flexible endoscope
1.4.1 Tracking

- 3 DoF tracking of beating heart (Ginhaux et al., Strasbourg University):
  - Eye to hand, 2D, direct
  - Predictive control
  - Velocity control signal
  - 500 Hz
1.4.2 Grasping

- 6 DoF grasping of a cube (Allen et al., Columbia University):
  - Eye in hand, 3D
  - Camera is at the wrist center
1.4.2 Grasping

- 5 DoF satellite grasping (DLR):
  - Eye in hand, 3D
  - Use of a special end-effector to capture the satellite using its nozzles
  - A second robot simulates micro-gravity
  - The insertion phase is guided using LASER rangefinders.
1.4.3 Insertion

- 5 DoF piston insertion (DLR):
  - Eye in hand, 3D
  - The engine block is put on a rotating table
  - The approach of the cylinder is controlled by vision
  - The insertion is force controlled
1.4.3 Insertion

- 4 DoF needle insertion (Nageotte et al., University of Strasbourg):
  - Eye to hand, 2D
  - The vision system is an endoscope
1.4.4 Positioning

- 5 DoF needle positioning (Maurin et al., Strasbourg University):
  - Eye to hand, 3D, indirect (look then move)
  - The vision system is a CT-scan
  - Sequence: acquisition – 3D reconstruction – motion
1.4.4 Positioning

- 1 DoF myocardium stabilization (Bachta et al., Strasbourg University):
  - Eye to hand, 2D, direct
  - 300 Hz
  - Piezoelectric actuation
2. Measurement estimation

- Image acquisition
- Image processing (noise removal, filtering)
- Feature detection (detection of points, lines, ellipses, ...)
- For 3D visual servoing: 3D reconstruction
  - Dementhon method
  - Image Jacobian method
2.1 Dementhon method

- Computes the pose of an object defined by points (not necessarily coplanar)
- Minimum of 4 points
- Iterative method: non-deterministic execution time

References:

- http://www.cfar.umd.edu/~daniel/#pose
2.1.1 Definitions
2.1.1 Definitions
2.1.2 Objectives

Let $M_{co}$, the homogeneous transformation between the camera frame $R_c (O, i, j, k)$ and the frame linked to the object $R_o (M_o , u, v, w)$:

$$M_{co} = \begin{bmatrix}
  i_u & i_v & i_w & \text{Tx} \\
  j_u & j_v & j_w & \text{Ty} \\
  k_u & k_v & k_w & \text{Tz} \\
  0 & 0 & 0 & 1
\end{bmatrix}$$

Coordinates of $i$ in $R_o$ denoted $^o i$

Coordinates of $j$ in $R_o$ denoted $^o j$

$^o i \times ^o j$

$T_x = x_0 Z_0 / f$, $T_y = y_0 Z_0 / f$ et $T_z = Z_0$

To estimate $M_{co}$, it is sufficient to estimate the coordinates of $i$ and $j$ in $R_o$ and the coordinates of $M_0$ in $R_c$. The coordinates of $M_0$ in $R_c$, is obtained from the depth $Z_0$ : if $x_0$ and $y_0$ are the coordinates of $m_0$ in $R_c$ then $T_x = x_0 Z_0 / f$ and $T_y = y_0 Z_0 / f$.
2.1.3 Orthographic projection

Let \( m_i (x_i, y_i, f)_{R_c} \) the perspective projection of \( M_i (X_i, Y_i, Z_i)_{R_c} \) in the image plane and \( p_i (x_i', y_i', f)_{R_c} \) its orthographic projection.

Let:

\[
x'_i = f \frac{X_i}{Z_0} \quad \text{and} \quad y'_i = f \frac{Y_i}{Z_0}
\]

And:

\[
x_i = f \frac{X_i}{Z_i} \quad \text{and} \quad y_i = f \frac{Y_i}{Z_i}
\]

Let \( s = f/Z_0 \) be the magnification factor of the orthographic projection.

Let:

\[
x'_i = f \frac{X_0}{Z_0} + f \frac{(X_i - X_0)}{Z_0} = x_0 + s (X_i - X_0)
\]

\[
y'_i = y_0 + s (Y_i - Y_0)
\]
2.1.4 Fundamental relationship

Equations:

\[ M_0 M_i \cdot \frac{f}{Z_0} i = x_i (1 + \varepsilon_i) - x_0 \]  \hspace{1cm} (1)

\[ M_0 M_i \cdot \frac{f}{Z_0} j = y_i (1 + \varepsilon_i) - y_0 \]  \hspace{1cm} (2)

with: \[ \varepsilon_i = \frac{1}{Z_0} M_0 M_i \cdot k \]

Demonstration: see paper.
2.1.4 Fundamental relationship

Equations (1) et (2) can be put in the compact form:

\[
\begin{align*}
M_0 M_i \cdot I &= \xi_i \\
M_0 M_i \cdot J &= \eta_i
\end{align*}
\]

With:

\[
\begin{align*}
I &= \frac{f}{Z_0} i \\
J &= \frac{f}{Z_0} j \\
\xi_i &= x_i (1 + \varepsilon_i) - x_0 \\
\eta_i &= y_i (1 + \varepsilon_i) - y_0
\end{align*}
\]

Let \([U_i V_i W_i], \quad ^0I = [I_u I_v I_w] \) and \(^0J = [J_u J_v J_w] \) respectively the coordinates of \( M_0 M_i, I \) and \( J \) in frame \( R_o \).

Let \( A \) the matrix of the coordinates \([U_i V_i W_i]\) of points \( M_i \), \( x' \) is the vector of the \( \xi_i \) and \( y' \) is the vector of the \( \eta_i \). It yields the following equations:

\[
A^0 I = x' \quad A^0 J = y'
\]
2.1.5 Resolution

With at least 3 points $M_i$, other than $M_0$, the rank of matrix $A$ is 3. So $^oI$ and $^oJ$ are obtained from the inverse or the pseudo-inverse $B$ of $A$:

\[
^oI = B x' \quad ^oJ = B y' \quad \text{avec } B=[A^T A]^{-1} A^T
\]

This yields:

\[
^o i = \frac{^o I}{\| I \|} \quad ^o j = \frac{^o J}{\| J \|} \quad ^o k = ^o i \times ^o j
\]

The depth $Z_0$ is derived from the norm of $I$ or $J$. It yields:

\[
T_x = x_0 \frac{Z_0}{f} \quad T_y = y_0 \frac{Z_0}{f} \quad T_z = Z_0
\]
2.1.6 Algorithm

Equations (1) and (2) put into matrix form (3) can be solved by fixing a value for $\varepsilon_i$. The initial value of $\varepsilon_i$ in the algorithm is 0, then, at each new iteration $\varepsilon_i$ is reevaluated thanks to the following equation:

$$
\varepsilon_i = \frac{1}{Z_0} M_0 M_i \cdot k
$$

The iterations stop when the difference between two successive values of $\varepsilon_i$ falls below a defined threshold. Usually, this falls needs less than 10 iterations to converge.

The assumption made by arbitrary fixing $\varepsilon_i = 0$ is equivalent to the assumption of a planar object parallel to the image plane. Of course, in the general case, this is false but not false enough to make the algorithm diverge.
2.1.6 Algorithm

Iteration 1:

\[ Z_0 \]

Image plane

\[ f \]

\[ k \]

\[ j \]

\[ i \]
2.1.6 Algorithm

Iteration 2:

Image plane

$Z_0$

$M_0$

$M_1$

$M_2$

$H$

$M_i$

$O$

$k$

$i$

$j$

$f$
2.1.6 Algorithm

Iteration 3:

Plan image

\[ H, \quad M_0, \quad M_1, \quad M_2 \]

\[ Z_0, \quad f, \quad \text{Plan image} \]

\[ i, \quad j, \quad k \]
2.1.6 Algorithm

Iteration n:

[Diagram showing a 3D coordinate system with points M₀, M₁, and M₂, and labels Z₀, Image plane, i, j, k, C, H, f.]
2.1.6 Algorithm

Initialization: \( n=0, N = \text{number of points}. \) Initialization of matrix \( A \) \((N\mathbb{I})x3\)

Compute \( B \) \(3x(N\mathbb{I})\), pseudo-inverse of \( A \). \( \varepsilon_{i(0)} = 0, i = 1...N \)

Compute \( i, j, Z_0 \):

Compute \( x' \) and \( y' \) of dimension \( N\mathbb{I} \) which coordinates are:

\[
x_i(1+\varepsilon_{i(n-1)}) - x_0 \quad \text{et} \quad y_i(1+\varepsilon_{i(n-1)}) - y_0
\]

Compute \( I \) and \( J \):

\[
I = Bx' \quad J = By'
\]

Compute \( o\), \( j \) and \( Z_0 \):

\[
s_1 = (I \cdot I)^{1/2} \quad s_2 = (J \cdot J)^{1/2} \quad o = \frac{I}{s_1} \quad j = \frac{J}{s_2} \quad Z_0 = \frac{2f}{s_1+s_2}
\]

Compute \( \varepsilon_{i(n)} \):

\[
\varepsilon_{i(n)} = \frac{1}{Z_0} M_0 M_i \cdot k \quad i = 1...N-1
\]

\( n=n+1 \) \quad \text{yes} \quad \text{no} \quad \text{Construction of} \ M_{co} \text{from the last values of} \ i, j \text{and} \ Z_0 \}

\[ |\varepsilon_{i(n)} - \varepsilon_{i(n-1)}| > \text{thr}\]
2.2 Image Jacobian method

- $p^*$: desired pose
- $p$: current pose
- Find $e = p - p^*$
- Task: target tracking
- Maintain $e$ as small as possible
2.2.1 Formulation

Let: \( \dot{F} = L_F T_c \) with:
\( \dot{F} \) the velocities of the points in the image,
\( L_F \) the image Jacobian,
\( T_c \) the velocity skrew of the camera frame.
\( \dot{e} = L_e T_c \) with:
\[
L_e = \begin{bmatrix}
R_{c^*c} & 0 \\
0 & L_{\vartheta u}
\end{bmatrix}
\]
\( e = [t_{c^*c} \ \vartheta u]^T \)

\( t_{c^*c} \) and \( R_{c^*c} \) : resp. translation vector and rotation matrix between \( R_c^* \) and \( R_c \)
u is the unit vector defining the rotation axis and \( \vartheta \) the rotation angle

\[
L_{\vartheta u} = I_3 - \frac{\vartheta}{2} [u]_x + \left( 1 - \frac{\text{sinc} \vartheta}{\text{sinc}^2 \frac{\vartheta}{2}} \right) [u]^2_\times
\]

\( [u]_\times = \begin{bmatrix}
0 & -u_z & u_y \\
u_z & 0 & -u_x \\
-u_y & u_x & 0
\end{bmatrix} \)
2.2.2 First order approximation

Since $\dot{F} = L_F T_c$ and $\dot{e} = L_e T_c$

it yields: $\dot{e} = L_e L_F^+ \dot{F}$ with $L_F^+ = (L_F^T L_F)^{-1} L_F^T$

First order approximation: $e \approx L_e L_F^+ (F - F^*)$ for $F - F^*$ and $e$ small

with: $F$ and $F^*$ resp. the current and desired feature vector

Considering that $L_e = I$ it yields:

\[ e \approx L_F^+ (F - F^*) \]

- Suited for tracking since $e$ is supposed to be small
- $L_F$ can be computed beforehand
- Very efficient algorithm
2.2.3 Case of point features

The scene is defined by \( n \) points :
\[
F = [x_1 y_1 x_2 y_2 \ldots x_n y_n]^T
\]
with \((x_i y_i)\) the coordinates in the image plane of point \( i \)

In this case :

\[
L_F = \begin{bmatrix}
-\frac{G_x}{Z_1} & 0 & \frac{x_1}{Z_1} & \frac{x_1 y_1}{G_y} & -\frac{G_x^2 + x_1^2}{G_x} & \frac{G_x}{y_1} \\
0 & -\frac{G_y}{Z_1} & \frac{y_1}{Z_1} & \frac{G_y^2 + y_1^2}{G_x} & -\frac{x_1 y_1}{G_y} & -\frac{G_y}{x_1} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
-\frac{G_x}{Z_n} & 0 & \frac{x_n}{Z_n} & \frac{x_n y_n}{G_y} & -\frac{G_x^2 + x_n^2}{G_x} & \frac{G_x}{y_n} \\
0 & -\frac{G_y}{Z_n} & \frac{y_n}{Z_n} & \frac{G_y^2 + y_n^2}{G_x} & -\frac{x_n y_n}{G_y} & -\frac{G_y}{x_n}
\end{bmatrix}
\]

with \( G_x \) and \( G_y \) the lens magnification along \( x \) and \( y \) (pixel unit)

\( Z_i \) is the depth of point \( i \) in 3D space
2.2.4 Example

\[ M_0 \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \quad M_1 \begin{pmatrix} 0.1 & 0 & 0 \end{pmatrix} \quad M_2 \begin{pmatrix} 0 & 0.1 & 0 \end{pmatrix} \quad M_3 \begin{pmatrix} 0 & 0 & 0.1 \end{pmatrix} \]

\[ M_{c*o} = \begin{bmatrix} 1 & 0 & 0 & 0.15 \\ 0 & 0 & -1 & 0.1 \\ 0 & 1 & 0 & 0.3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \]

\[ F_d = \begin{bmatrix} 500 & 333.33 & 333.33 & 333.33 & 375 & 250 & 500 & 0 \end{bmatrix}^T \]

\[ t_{c*o} = \begin{bmatrix} 0.1 & 0 & 0 \end{bmatrix}^T \quad \theta = 0.1 \text{ rad} \quad u = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T \]

\[ F = \begin{bmatrix} 173.3 & 448.7 & 519.9 & 448.7 & 128.9 & 359.3 & 167.5 & 100.3 \end{bmatrix}^T \]

\[ L_F(F - F_d) = \begin{bmatrix} 0.0990 \\ 0.0021 \\ 0.0016 \end{bmatrix} \]

\[ \begin{bmatrix} 0.9861 \\ 0.0594 \\ -0.1549 \end{bmatrix} \]

\[ G_x = G_y = 1000 \]

Matlab script: jacod3D.m
3. Control

- Kinematic control
  - 2D (IBVS)
  - 3D (PBVS)
- Dynamic control
3.1.1 2D kinematic vis. servoing

\[ F^* \rightarrow e \rightarrow -k \hat{L}_F^+ \rightarrow 1 \rightarrow T_c \rightarrow L_F \rightarrow \frac{1}{s} \rightarrow \dot{F} = e \]

\( F^* \): desired image \quad \( F \): measured current image

Control law: \( T_c^* = -k \hat{L}_F^+ e \) with \( \hat{L}_F^+ \) estimated pseudo-inverse of image Jacobian

\( T_c = T_c^* \): the robot is supposed to be perfect

\[ \dot{e} = \frac{d}{dt} (F - F^*) = \dot{F} \] with \( F^* \) supposed to be constant

According to the definition of the image Jacobian: \( \dot{F} = \dot{e} = L_F T_c \)

Since \( T_c = T_c^* = -k \hat{L}_F^+ e \) it yields: \( \dot{e} = -kL_F \hat{L}_F^+ e \)

For 3 points and if we suppose \( \hat{L}_F = L_F \) the solution is:

\[ e(t) = e(0) \exp(-kt) \] where \( e(0) \) is the error at \( t=0 \)

See simulink simulation 2Dservo.mdl
3.1.2 3D kinematic vis. servoing

\[ p^* : \text{desired pose} \quad p : \text{reconstructed current pose} \]

Control law: \( T_c^* = -k \hat{L}_p^{-1} e \) with \( \hat{L}_p^{-1} \) estimated inverse of \( L_p \)

\( T_c = T_c^* : \text{the robot is supposed to be perfect} \)

\( \dot{e} = \frac{d}{dt} (p - p^*) = \dot{p} \) with \( p^* \) supposed to be constant

According to the definition of \( L_p : \dot{p} = \dot{e} = L_p T_c \)

Since \( T_c = T_c^* = -k \hat{L}_p^{-1} e \) it yields: \( \dot{e} = -k L_p \hat{L}_F^{-1} e \)

If we suppose \( \hat{L}_p = L_p \) the solution is:

\[ e(t) = e(0) \exp(-kt) \text{ where } e(0) \text{ is the error at } t=0 \]
3.1.3 Limitations

- Simplified continuous-time kinematic 2D VS:

- Simplified discrete-time kinematic 2D VS:
3.1.3 Limitations

- Response of the discrete-time 2D VS with $k = 0.34$:
3.2 Dynamic control

- Image acquisition model
- Manipulator model
- Typical dynamic visual loop
3.2.1 Image acquisition model

- Acquisition sequence:
  - Integration of light-generated charges in pixels
  - Readout of the charges:
    - Global shutter: all charges are read together
    - Rolling shutter: each charge is read sequentially
    - Progressive scan: the whole image is scanned at once
    - Interlaced scan: even and odd lines are scanned alternatively

- Image transfer to the acquisition board:
  - Analog transfer: coaxial cable (50 Mpix/s max)
  - Digital transfer: USB2 (480 Mbits/s), Firewire (800 Mbits/s), LDVS (4 Gbits/s), Gigethernet (1 Gbits/s), Camlink (7.14 Gbits/s)
3.2.1 Image acquisition model

- Typical image acquisition time diagram:

\[ t_i = \text{integration time} \]
\[ t_v = \text{transfer time} \]
\[ t_p = \text{processing time} \]
\[ t_c = \text{control signal computation time} \]

Equivalent image sampling time

Model = Delay \( D \) \( \Leftrightarrow z^{\text{int}(\frac{D}{T_v} + 0.5)} \)
3.2.1 Image acquisition model

- Motion blur:

\[
\text{Model} = \text{averaging filter} = \frac{1 + z^{-1}}{2}
\]
3.2.2 Manipulator model

- Computed torque:

\[
\tau^* = D(q) + \hat{F}_d(q) + F_v
\]

\[
\dot{q} = \frac{1}{s} \text{ Equivalent model: } \frac{\dot{q}(s)}{\tau^*(s)} = \frac{1}{s}
\]
3.2.2 Manipulator model

- Virtual Cartesian motion device:

\[
\begin{align*}
T(z) &= \frac{p(z)}{\dot{p}^*(z)} = (1 - z^{-1}) Z \left\{ L_p L_c H(s) \hat{L}_c^{-1} \hat{L}_p^{-1} \right\} \\
&\approx (1 - z^{-1}) Z \left\{ \frac{H(s)}{s^2} \right\}
\end{align*}
\]
3.2.2 Manipulator model

- Notations:

\[ \dot{p}^* : \text{reference velocity} \]
\[ ZOH : \text{Zero Order Hold (model of the DACs)} \]
\[ \hat{L}_p : \text{estimated Jacobian of the representation s.t.} \quad \dot{p} = \hat{L}_p T_c \]
\[ \hat{L}_c : \text{estimated Jacobian of the robot s.t.} \quad T_c = \hat{L}_c \dot{q} \]
\[ \dot{q}^* : \text{joint velocity reference} \]
\[ C_v : \text{velocity controller of the decoupled robot} \]
\[ \tau^* : \text{torque reference for the computed torque algorithm} \]
\[ \dot{q} : \text{joint velocity} \]
\[ H(s) : \text{model of the velocity controlled decoupled robot} \]
\[ T_c : \text{velocity screw of the camera} \]
\[ p : \text{output (can be a pose (2D) or a feature vector(3D))} \]
\[ T_s : \text{sampling period} \]
3.2.3 Typical dynamic visual loop loop

\[ p^* \rightarrow C(z) \rightarrow z^{-1} \dot{p}^* \rightarrow T(z) \rightarrow p \]

\[ \hat{p} \]

\[ z^{-d} \frac{1+z^{-1}}{2} \]

\( p^* \): reference
\( \hat{p} \): visual measurement
\( C(z) \): visual loop controller

- **Notes:**
  - Controllability: the dimension of \( p \) should be less or equal to the number of degrees of freedom of the manipulator in order for the system to be controllable. So in case of IBVS, the number of features coordinates should be restricted to 6. When using more feature coordinates, PBVS should be preferred.
  - The additional delay in the direct transfer is due to the time needed to compute the control signal (applied at the next iteration).
4. Conclusion

- **Kinematic visual servoing:**
  - Slow, simple to implement
  - Suited when robustness and accuracy are more important than speed
  - Applications: positioning of surgical instruments with respect of static structures (brain surgery, orthopedic surgery)

- **Dynamic visual servoing:**
  - Fast, can be tricky to tune
  - Suited when advanced control techniques are relevant (predictive control, repetitive control)
  - Applications: physiological motion compensation
4. Conclusion

- Perspectives:
  - High bandwidth visual servoing with medical imaging feedback (X-ray, CT, MRI, Ultrasound)
  - Issue: high bandwidth and low latency access to the low level data

Thank you!