Soft Motion Trajectory Planning and Control for Service Manipulator Robot

Ignacio Herrera-Aguilar * † Daniel Sidobre* †

* LAAS - CNRS
† Universite Paul Sabatier
7, avenue du Colonel Roche
31077 Toulouse, France

Abstract— One important difference between industrial robotic manipulators and service robot applications is the human interaction, which introduce safety and comfort constraints. In this paper, we define soft motions conditions to facilitate this cohabitation. We propose an on-line trajectory planner that generates the necessary references to produce soft motion and a control loop that guarantees the end effector’s motion characteristics (jerk, acceleration, velocity and position) in the Cartesian space, by using quaternion feedback. We propose two visual feedback control loops: a visual servoing control loop in a shared position and the upper level called control. Rajan proposes a two steps minimization algorithm [5], temporal/torque constraints are considered in the works of Shin and McKay [6] and Bobrow et al [7] and Kyriakopoulos and Saridis propose minimal jerk control [8]. The objectives of the trajectory planner are improving tracking accuracy and reducing manipulator wear by providing smooth references to the servo-motors control, by doing this the end-effector’s motion is smooth too (Smooth motion). An important remark is that the smoothness is obtained by the limits on velocity, acceleration and jerk of each joint that provides a good performance in industrial applications.

We define Soft motion in opposition to Smooth motion as a continuous movement with limited condition in jerk, acceleration and velocity of robot’s end effector in the Cartesian space. So the movement has soft start, soft stop and soft evolution even under rotations.

Lambrechts [9] proposes the utilization of a fourth order trajectory planner for single axis point to point motion control. Here, the influence of the input (reference to servo system) is considered to achieve desired performance while using a classical control (PD). Hogan [10] shown that use of jerk provides smoothness, then a third order trajectory planner looks like a good solution.

According to Nelson [11] force and vision feedback complement one another. Vision allows accurate part alignment within imprecisely calibrated and dynamically varying environments, without requiring object contact, in other words, vision provides global 3D information on the environment. Force sensors provide localized but accurate contact 3D information. Nelson proposed three levels to integrate force and vision: traded, hybrid and shared control. In these cases, a control loop position is realized for each link. Baeten [12] using the Task Frame Formalism presents an alternative for the shared control. In both cases, end effector’s pose information from the internal sensors and motion characteristics are not considered.

Visual feedback is commonly termed visual servoing, hence vision is a part of a control system where it provides feedback about the state of the environment. In the last three decades visual servoing systems have been studied, an extensive survey can be found in [13] and a complement for manipulation in [14].

In this paper, we consider the use of an arm manipulator actuated by servo-motors with strong reduction ratios for
applications in service robots where low operation speeds are needed to ensure safety, we chose to control the end effectors pose (position and orientation) in Cartesian space. A kinematic control loop is used by assuming that the robot dynamics is negligible. In this work internal and visual feedback are used, we propose two approaches. In the first approach, a shared position - vision control loop is used, here we consider to extend our works with force feedback. While in the second approach, a sensor driven (visual guided) schema is presented, here the on-line capabilities of the trajectory planner are tested.

Why a shared position - vision control? Considering the trajectory generated by Lopez-Damian and Sidobre in [15], the success of the grasping task depends on the quality of the model. Considering a non-perfect model, we consider that the trajectory tracking can be compensate by visual information. An experimental and simple example, shows how a visual servoing loop reduce the errors in the model.

This paper presents in the next section the related work. Section III describe the soft motion trajectory planner. In section IV, the control loop. Finally, experimental results and conclusions are presented respectively in sections V and VI.

II. RELATED WORK

A. Trajectory Planning

According to Brady [16], Trajectory Planning converts a description of a desired motion to a trajectory defining the time sequence of intermediate configurations of the arm between the origin and the final destination. Literature shows two different approaches. The first one considers working in joint space and the second one in task space. We have chose the last one.

The first works in the area refers to Paul [17] and Taylor [18]. Paul use homogeneous coordinates, presents a matrix equation that relates the representation of a configuration as a sequence of frames, local to arm joints, to a representation that is external to the arm and determined by the application. Paul considers constant acceleration. Taylor presents a technique for achieving straight lines, by choosing midpoints between two desired configurations. Taylor propose the use of quaternions for rotation.

To realize smooth motion and tracking, several approaches has been presented, such as trapezoidal or bell-shaped velocity profiles using cubic, quartic or quintic polynomials. Andersson [19] use a single quintic polynomial for representing the entire trajectory, while Macfarlane [20] extend Andersson’s work and uses seven quintic polynomials for industrial robots.

In the case of human interaction Amirabdollahian et all [21] use a seventh order polynomial while Seki and Tadakuma [22] propose the use of fifth order polynomial, both of them for the entire trajectory with a minimum jerk model. Herrera and Sidobre [1] propose seven cubic equations to obtain soft motion in robot service application.

B. Shared vision-position control

Castano and Hutchinson [23] introduce visual compliance that is realized by a hybrid vision/position control structure. There the two degrees of freedom parallel to the image plane are controlled using visual feedback and the remaining degrees are controlled by position feedback in a monocular eye-to-hand configuration.


Hager [25] using also stereo eye-to-hand configuration, defines a set of primitive skills to enforce specific task-space kinematic constraint between a robot end-effector and a 3D target feature.

C. Visually guided control

The ping pong player presented by Andersson in [19] is the reference. Here, by using visual stereo information the system dinamically change the trajectory. The control is realized in joint space with continuous acceleration, velocity and position.

Lloyd and Hayward [26] presents a technique for blending path segments for sensor-driven tasks.

III. SOFT MOTION TRAJECTORY PLANNING

We consider the planning of a trajectory defined by a set points generated by motion planning techniques. The motion planner calculate the trajectory which the end effector must follow in space. However, the temporal characteristics of this movement are independent. According to [1] the Soft motion can be found by the next planner. We consider firstly the monodimensional and secondly the multidimensional extension.

A. Unidimensional Case

Firstly, we consider the canonical case of the figure 1 without lost of generality.

![Fig. 1. Jerk, Acceleration, Speed and Position curves](image)

The motion can be separated in seven segments, defined by the time period. Considering two sections, we use the subindex
(a and b) for differencing. We have:
- \( T_{jpa} \): Jerk positive time
- \( T_{aca} \): Acceleration constant time
- \( T_{jna} \): Jerk negative time
- \( T_{vc} \): Velocity constant time
- \( T_{jnb} \): Jerk negative time
- \( T_{acb} \): Acceleration constant time
- \( T_{jpb} \): Jerk positive time

Considering one dimension motion and limit conditions, we can find three different type sections by integration:

- The motion with a maximum jerk (\( J_{\text{max}} \)):
  \[
  J(t) = J_{\text{max}} \\
  A(t) = A_0 + J_{\text{max}} t \\
  V(t) = V_0 + A_0 t + \frac{1}{2} J_{\text{max}} t^2 \\
  X(t) = X_0 + V_0 t + \frac{1}{2} A_0 t^2 + \frac{1}{6} J_{\text{max}} t^3
  \]

- The motion with a maximum acceleration (\( A_{\text{max}} \)):
  \[
  J(t) = 0 \\
  A(t) = A_{\text{max}} \\
  V(t) = V_0 + A_{\text{max}} t \\
  X(t) = X_0 + V_0 t + \frac{1}{2} A_{\text{max}} t^2
  \]

- Finally, the equations for the motion with a maximum velocity (\( V_{\text{max}} \)):
  \[
  J(t) = 0 \\
  A(t) = 0 \\
  V(t) = V_{\text{max}} \\
  X(t) = X_0 + V_{\text{max}} t
  \]

where \( J(t), A(t), V(t), X(t) \) represents jerk, acceleration, velocity and position functions respectively. \( A_0, V_0 \) and \( X_0 \) are the initial conditions.

In the object to guarantee soft motion, we define the intervals:
\[
J(t) \in [-J_{\text{max}}, J_{\text{max}}] \\
A(t) \in [-A_{\text{max}}, A_{\text{max}}] \\
V(t) \in [-V_{\text{max}}, V_{\text{max}}]
\]

**B. Point to point motion**

According to figure 1, the motion is realized at limit conditions. To achieve \( A_{\text{max}} \) from initial condition \( A(0) = 0 \), we have a jerk time \( (T_j) \) that is equal to the time for going from \( A_{\text{max}} \) to 0. During \( T_j \), the acceleration increase or decrease linearly according to the jerk. At this point, it is important to observe a symmetry in acceleration and an anti-symmetry in jerk. Now, we consider velocity, the symmetry effect is present too, but this time according to acceleration. During the constant acceleration time \( (T_a) \), the velocity increase or decrease linearly according to the acceleration. Finally, \( T_v \) is defined as the constant velocity time. We have then
\[
T_j = T_{jpa} = T_{jna} = T_{jnb} = T_{jpb} \\
T_a = T_{aca} = T_{acb} \quad T_v = T_{vc}
\]

Our system calculates times \( T_j, T_a \) and \( T_v \), whose make possible to obtain the desired displacement between an origin position and a final position. As the end effector moves under maximum motion conditions \( (J_{\text{max}}, A_{\text{max}} \) or \( V_{\text{max}} \)), we obtain a minimal time motion. The complexity of the equations system depends on the distance between the positions and the maximal limits.

The point to point motion requires to reach the destination. Physical limitations are not considered, and in order to guarantee the emergency soft stop on desired path, null final conditions in acceleration and velocity are fixed \( (A(f) = 0 \) and \( V(f) = 0) \). Using this conditions we can find the necessary times \( T_{jmax} \) to achieve \( A_{\text{max}} \) and \( T_{amax} \) to achieve \( V_{\text{max}} \).

\[
T_{jmax} = \frac{A_{\text{max}}}{J_{\text{max}}} \quad T_{amax} = \frac{V_{\text{max}}}{A_{\text{max}}} - \frac{A_{\text{max}}}{J_{\text{max}}}
\]

According to this, we can build the Figure 2. Where we can see the maximal possible displacement in the case 2 when \( T_v = 0, T_a = T_{amax} \) and \( T_j = T_{jmax} \), and in case 3 when \( T_v = 0 \) and \( T_a = 0 \) while \( T_j = T_{jmax} \).

![Fig. 2. Velocities and Positions](image)

We define the distance \( (D) \) as the difference between the
origin ($P_o$) and destination ($P_f$) positions.

$$D = P_f - P_o$$  \hspace{1cm} (2)$$

We have two limit conditions:

- Condition 1: Case 2, where $V_{max}$ is reached. It means, $A_{max}$ is reached too. Then we have to find the traversed distance ($D_{thr1}$). Using the limit times

$$T_j = T_{jmax} \hspace{1cm} T_a = T_{amax} \hspace{1cm} T_v = 0$$

we find

$$D_{thr1} = \frac{A_{max} V_{max}}{J_{max}} + \frac{V_{max}^2}{A_{max}}$$  \hspace{1cm} (3)$$

- Condition 2: Case 3, where only $A_{max}$ is reached. Using

$$T_j = T_{jmax} \hspace{1cm} T_a = 0 \hspace{1cm} T_v = 0$$

we can find a distance ($D_{thr2}$)

$$D_{thr2} = \frac{2 A_{max}^3}{V_{max}^2 J_{max}}$$  \hspace{1cm} (4)$$

Considering the conditions (Eqs. 3 and 4) we can formulate the algorithm 1

**Algorithm 1 Maximum Jerk Algorithm**

Calculate distance $D$ (Eq 2)

if $D \geq D_{thr1}$ then

$$T_j = T_{jmax} \hspace{1cm} T_a = T_{amax}$$

$$T_v = \frac{D - D_{thr1}}{V_{max}}$$

else

if $D \geq D_{thr2}$ then

$$T_v = 0 \hspace{1cm} T_j = T_{jmax}$$

$$T_a = \sqrt{\frac{A_{max}^2}{4J_{max}} + \frac{D}{A_{max}} - \frac{3A_{max}}{2J_{max}}}$$

else

$$T_v = 0 \hspace{1cm} T_a = 0$$

$$T_j = \sqrt[3]{\frac{D}{2J_{max}}}$$

end if

end if

Since, the acceleration and speed curves are symmetrical. The optimal time for the trajectory in considering the constraints is given by:

$$T_f = 4 * T_j + 2 * T_a + T_v$$  \hspace{1cm} (5)$$

C. Multipoint Trajectory Planner

The strategy presented in previous section is extended for the multipoint case to go from $P_0$ to $P_n$. We define the current position $P_i$ as a position in the interval $P_i$ and $P_{i+1}$ where $i = 0...n-1$. The trajectory is computed by successive application of seven cubic equations. For each segment, we consider initial conditions defined by previous segment at ($P_c$), and zero final conditions at ($P_{i+1}$) for acceleration and velocity.

Considering the trajectory generation knowing only the destination position ($P_{i+1}$), we compute the stop position ($P_s$) from the current motion conditions. Considering the stop position and the destination, we can find four possibilities.

- Start Motion
  The "easy" case, we apply previous algorithm. Because the current conditions are nulls.
- Same Direction Motion ($P_s > P_{i+1}$)
  The motion is in the same direction, the new destination is after the stop position. We apply the set of equations presented on section 3.1 with the current motion conditions.
- Halt Motion ($P_s = P_{i+1}$)
  The stop position and the new destination are equal. We consider to stop from current motion conditions.
- Change Direction Motion ($P_s < P_{i+1}$)
  This case is found when the final position $P_{i+1}$ is before the stop position $P_s$. If we consider a natural evolution of the system of equations, some conditions have multiple solutions. To guarantee real-time applications, we have separate the Change Direction Motion in two, losing the optimal time. Firstly, a halt motion. Secondly, a start motion in the other direction.

We consider switch position $P_c$ at the moment when the motion begin to slow down. This position defines the current position $P_c$ as initial position for the next segment at time $T_c$ defined by:

$$T_c = T_{jpa} + T_{aca} + T_{jna} + T_{ve}$$  \hspace{1cm} (6)$$

The figure 3 shows the evolution for two destinations from the origin position ($P_0 = X(0) = 0$) to $P_1 = 0.5$ and $P_2 = 1.5$. Limits parameters are $J_{max} = 8$, $A_{max} = 2$, $V_{max} = 1$ and sampling time is $T_s = 0.01$. According to figure, the motion going through $P_1$ in a non null velocity without losing the soft motion condition.

D. Multidimensional Case

For the multidimensional case, we keeps the same strategy. Each dimension is independent each other. To guarantee trajectory tracking, we consider the motion between two points as a straight-line motion in $n$ dimensional space. The only way for assuring straight-line tracking is assuring that each dimension motion has the same duration. Then, we compute the final time for each dimension. Considering the largest motion time, we readjust the other dimension motion to this time. Time adjusting is done by decreasing limit conditions. In other words, the motion is minimum time for one direction.
In the other directions, the motions are conditioned by the minimum one. Figure 3 shows a two-dimensional example with limit parameters $J_{\text{max}} = 8$, $A_{\text{max}} = 2$ and $V_{\text{max}} = 1$ for $X$ direction and $J_{\text{max}} = 4$, $A_{\text{max}} = 1$ and $V_{\text{max}} = 0.5$ for $Y$ direction. The origin is defined by the pair $(0,0)$, the first destination point by $(1,0.5)$ and the final destination point is $(1.2,1.5)$.

Figure 4 shows a two-dimensional example with limit parameters $J_{\text{max}} = 8$, $A_{\text{max}} = 2$ and $V_{\text{max}} = 1$ for $X$ direction and $J_{\text{max}} = 4$, $A_{\text{max}} = 1$ and $V_{\text{max}} = 0.5$ for $Y$ direction. The origin is defined by the pair $(0,0)$, the first destination point by $(1,0.5)$ and the final destination point is $(1.2,1.5)$. 
E. Manipulators case

In the case of robot’s end effector we use seven dimensions motion. Three dimensions for translation \( \mathbf{P} \) and four for rotation \( \mathbf{Q} \) (quaternion).

\[
\mathbf{P} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad \mathbf{Q} = n + \mathbf{q} \quad \text{where} \quad \mathbf{q} = \begin{bmatrix} i \\ j \\ k \end{bmatrix}
\]

Linear velocities \( \mathbf{V} \) obtained can be applied directly as velocity references. On another hand, the evolution of the quaternion \( \mathbf{Q} \) must be transformed into angular velocities. We use the transformation function proposed in [27].

\[
\begin{bmatrix} \Omega \\ 0 \end{bmatrix} = 2 \mathbf{Q}_r^T \dot{\mathbf{Q}} \quad \text{where} \quad \mathbf{Q}_r = \begin{bmatrix} n & k & -j & i \\ -k & n & i & j \\ j & -i & n & k \\ -i & -j & -k & n \end{bmatrix}
\]

F. Mobile Manipulators case

In this case, the seven dimensions motion are increased by the mobile platform’s mobilities. For mobile platforms, we consider the motion over the plane \( \left( X_p, Y_p \right) \) and the direction \( \left( \theta_p \right) \). Then, for a mobile manipulators we have a ten dimensional trajectory planner. Non holonomic constraints must be satisfied at motion planning level, but the solution is not trivial. To guarantee the end effector’s Soft Motion conditions, the time needed to track the platform path must be at least equal to the time needed to track the end effector’s path. If these condition is guarantee, the motion along the end effector’s path can be time conditioned.

IV. ROBOT MANIPULATOR’S CONTROL LOOP USING QUATERNION FEEDBACK

In this section, we focus our attention on the arm manipulator’s control loop, because the user’s comfort depends on the arm motion. The control loop needs to consider the Soft Motion constraints.

The configuration of six joints arm manipulator is defined by a vector \( \theta \) of six independent joint coordinates which correspond to the angle of the articulations.

\[
\theta = [q_1 \ q_2 \ q_3 \ q_4 \ q_5 \ q_6]^T
\]

The Pose of the manipulator’s end effector then is defined by seven coordinates, 3 for \( \mathbf{P} \) and 4 for \( \mathbf{Q} \), said Operational Coordinates which gives the position and the orientation of the final body in the reference frame.

Resolved motion rate control means that the movements generated by the servo-motors of the articulations of the manipulator combine to produce a uniform displacement. In other words, the servo-motors evolve at different speeds with an aim of obtaining the desired total movement. Whitney [28] has shown that the speed of the axis is given by

\[
\dot{\theta} = \mathbf{J}^{-1} \begin{bmatrix} \mathbf{V} \\ \Omega \end{bmatrix}
\]

where \( \mathbf{V} \) and \( \Omega \) represents the linear and angular velocities of the robot’s end effector. And \( \mathbf{J} \) is the Jacobian matrix.

In a closed loop control [29], the control law is replaced by

\[
\dot{\theta} = \mathbf{J}^{-1} \begin{bmatrix} \mathbf{V} - \mathbf{K}_p \mathbf{e}_p \\ \Omega - \mathbf{K}_o \mathbf{e}_o \end{bmatrix}
\]

where \( \mathbf{K}_p \) and \( \mathbf{K}_o \) are diagonal gain matrices, and \( \mathbf{e}_p \) and \( \mathbf{e}_o \) respectively represent the position and orientation error vectors. Yuan [30] uses quaternion feedback in a closed-loop resolved rate control. The position and orientation tracking error are defined by

\[
\mathbf{e}_p = \mathbf{P} - \mathbf{P}_d \quad \mathbf{e}_o = n_dq - n_dq_d - q_d \times q
\]

where the index \( d \) indicates that they are set points.

Yuan [30] use a shows global asymptotic convergence for \( K_p > 0 \) and \( K_o > 0 \). The control law 8 can be interpreted as a position proportional controller plus velocity feedforward for each direction. In our application we change the proportional controller at each direction by a proportional integral digital controller of the form [31]:

\[
u[k] = u[k-1] + \Delta u[k]
\]

with

\[
\Delta u[k] = C \left( (e[k] - e[k-1]) + \frac{T}{T_i} e[k] \right)
\]

To achieve soft motion, we have limited the control law. By limiting \( \Delta u[k] \), we limit the acceleration at each dimension. By limiting \( u[k] \), we limit the velocity at each dimension and we avoid the integral saturation problem. Considering this controller and the robot as integrator, we have two integrator in the control loop, whose provide a velocity tracking.

To guarantee the tracking in case of singularity, Buss [32] propose different solutions, we have selected the damped least squares method for inverse kinematics.

\[
\mathbf{J}^{-1} \simeq \mathbf{J}^T (\mathbf{J}^T \mathbf{J} + \lambda \mathbf{I})^{-1}
\]

It is known that in the proximity of a singularity, the joint velocity references exceed the limits (\( \dot{\theta} \rightarrow \infty \)). To avoid this problem, we propose to limit the velocity reference by weighting the velocities in function of the largest exceeding.

A. Shared Position - Vision Control Loop

Typically, robotic tasks are specified with respect to one or more coordinate frames. Using the homogeneous representation \( \mathbf{T}_a^b \) that represents the transformation of frame \( b \) with respect to frame \( a \). Let \( w \) denote the world frame, \( h \) the hand frame, \( c \) the camera frame, \( i \) the image frame, \( g \) the gripper frame and \( o \) the object frame.

These approach can be considered as a dynamic look-and-move system. One point is defined by its frame position. In this case, we formulate the problem in terms of homogeneous coordinate transformations. One point in the world (\( \mathbf{P}_w \)) is projected in the image frame (\( \mathbf{P}_i \)) losing one degree of freedom (\( z_i \)). The point in camera frame (\( \mathbf{P}_c \)) can be found by

\[
x_c = d \frac{x_i - u_0}{\alpha_u} \quad y_c = d \frac{y_i - v_0}{\alpha_v} \quad z_c = d
\]
where $\alpha_u$, $\alpha_v$, $u_0$ and $v_0$ are the pin-hole camera parameters and $d$ is the depth in the image. The point in world frame ($\hat{P}_w$) from image reconstruction can be found by

$$\hat{P}_w = T_h c P_c$$

The visual error for each direction in position can be found by

$$e_w = P_g - \hat{P}_w$$

Orientation errors can be found applying geometrical relations between different measured points. Angular and linear velocity feedback imply object velocities in the image plane through an image Jacobian, here we only consider position error for showing the control loop advantages.

Corke [13] proposes the use of open loop integrators. For each direction, we use a law of control of the form

$$u_v = (K_{vp} + K_{vi} z - \frac{1}{z}) e_w$$  (12)

where $K_{vp}$ and $K_{vi}$ are chosen to respect jerk constraints.

$$P_d^* = P_d + u_v$$

The figure 5 shows the control loop.

**B. Visually guided control loop**

In this case, where the target is defined by the vision system. Considering a tracking problem for each 3D target position, the system must be capable to compute the motion from the current conditions to the new target position. Here, we are not considering the target motion characteristics, but the end effector’s motion must satisfy the soft motion conditions. Considering the conditions presented in Multipoint Trajectory planning (section III.c), the algorithm of trajectory planning is applied from the current motion conditions used as Initial Conditions and the target position as final position with acceleration and velocity nulls.

The figure 5 shows the proposed control loop.

**V. EXPERIMENTAL RESULTS**

**A. Experimental Platform**

We have tested the control loop in a PA10-6CE Mitsubishi manipulator, called Jido. Jido is controlled by a PCI Motion Control CPU Board in a Pentium IV Personal Computer. Three links define the manipulator, using Denavit-Hartenberg parameters: $a_2 = 0.450$ m, $r_4 = 0.480$ m and $r_6 = 0.30$ m (Figure 7). The software control is developed using Open Robots tools. The sampling time is fixed to 10 ms.

The linear and angular end effector motion are limited to

<table>
<thead>
<tr>
<th>Linear Limit</th>
<th>Angular Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$J_{max}$</td>
<td>0.900 m/s$^3$</td>
</tr>
<tr>
<td>$A_{max}$</td>
<td>0.300 m/s$^2$</td>
</tr>
<tr>
<td>$V_{max}$</td>
<td>0.150 m/s</td>
</tr>
</tbody>
</table>

**B. Trajectory planning for a grasping trajectory**

Consider a set of configurations needed to grasp a box [15], as showed in the first image of the figure 8. The photo sequence in the same figure, shows the motion performed by the arm manipulator.

When the model environment is perfect, the grasping is done without problems, even in presence of singularities in the arm configuration. The third photo of the sequence shows this case.

The performance of the trajectory planner can be see in the figure 9. The first plot shows the end effector’s acceleration and velocity along the X dimension, the second plot shows the
position along the same dimension, while the third plot shows the error between the position computed by the trajectory planner and the position of the robot’s end effector. We can see that the error is of the order of less than one millimeter.

C. Visual servoing for straight line motion

In this experiment, we consider a displacement from the origin position \( P_o \) to the destination position \( P_f \) of 0.25 m in the Z axis, and visual servoing in Y direction in order to center the gripper on the line.

The figure 10 shows the experimental context.

The figure 11 shows the comparison in Z direction with or without visual servoing. In both cases, the reference in velocity and position along the Z dimension are the same. Both trajectories are identical. There are not effect along the Z dimension when there are a correction on the Y dimension.

The figure 12 shows the evolution of the end effector considering the target line. Here we can see the correction in the position in Y, while the motion is done along the Z dimension.

Finally, the figure 13 shows the visual error during the visual servoing. By chosing \( K_{vp} \) and \( K_{vi} \) from 12. There are a trade-off in the selection of gains for the control loops between the tracking and the soft motion conditions.

The complexity of this experiment is not the tracking of the line. We must consider that the trajectory has been designed for one different position along the Y dimension. The robot’s end effector orientation is fixed. The correction of the position is done without losing the orientation and the others trajectory planned dimensions (X and Z).

D. Visually guided

Here, we tested the trajectory planner when the tracking of a moving target is done. For simplicity, the orientation is fixed in the first example. To avoid the problems inherent with the measurement of the target’s position, the limit conditions on jerk, acceleration and velocity have been reduced. We assume that the target’s velocity is reachable by the robot,
Fig. 10. Camera configuration and straight line target

Fig. 11. Z motion comparison

Experimental examples which represent a very early result, show the validity of the approach.

The trajectory planner is simpler than previous solutions. It uses seven cubics curves for each segment in one direction. The time to compute the trajectory is compatible with on-line planning to take into account real-time modifications of curves.

Considering the use of a visual servoing control loop during a grasp task, the approach presented is valid, but there is an open question, which object features are available?

The control loop presented deals with the possibility of using external velocities, we are going to introduce a force loop in order to define a complete manipulation controller for service robot.

ACKNOWLEDGEMENT

The work described in this paper was partially conducted within the EU Integrated Project COGNIRON ("The Cognitive Companion") and funded by the European Commission Division FP6-IST Future and Emerging Technologies under Contract FP6-002020. The authors acknowledge the support of the DGEST-SFERE program.

REFERENCES
