Soft-Motion and Visual Control for service robots

Ignacio Herrera Aguilar[†♠] and Daniel Sidobre[†‡]
(iherrera, daniel)@laas.fr

[†] LAAS-CNRS
7, avenue du Colonel Roche
31077 Toulouse, France

[‡] Université Paul Sabatier
118, route de Narbonne
31062 Toulouse, France

[♠] Inst. Tec. de Orizaba
Av. Inst. Tecnológico No. 852
94320 Orizaba, Ver. México

***Abstract*— One important difference between industrial robotic manipulator 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 a trajectory planner that generate the necessary references *on-line* to produce soft motion and a control loop that guarantees the motion characteristics (jerk, acceleration, velocity and position) of the end effector in the Cartesian space, whit and with out visual feedback. Experimental results carried out on a Mitsubishi PA10-6CE arm.**

***Keywords*— Trajectory planing, Soft-Motion, Visual Servoing, Visual Guided Control**

I. Section

Arm manipulator control has been standardized little by little, industrial applications have been developed using different techniques, several restrictions have been satisfied by using robot-like arms with specific application for a limited number of tasks. However, all these applications are confined in structured and safe spaces, that are free of man interaction.

According to the IFR (International Federation of Robotics), a *service robot* is a robot which operates semi or fully autonomously to perform services useful to the well being of humans and equipment.

Robotic manipulator arms are complex mechanical structures for which response to motion moment varies not only with the load but also with the configuration, speeds and accelerations. Most of manipulators use electric servo-motors as actuators. In fact, servo-motors characteristics are one of the factors to define the control law.

In the case of small reduction ratios the use of control laws based on dynamic models give good results, the goal is to maintain the dynamic response of the system inside a certain performance criteria [1]. As strong dependence of model remains, robust or adaptive techniques have been used to solve this disadvantage. In the case of strong reduction ratios, inertia seen by the engines has a low variation and manipulator control can be achieved axis by axis using classical control loops (PID).

The problem of robot control has been divided in three hierarchical levels, a first (lower) level called *control* or *path tracking*, the second level called *trajectory planning* and a third (upper) level of *motion planning*. Using this approach industrial robots can evolve at high speeds satisfying path constraints. Motion Planning Techniques are extensely exposed in [2]. Considering the trajectory planning and the path tracking as an unite, literature presents different works: Geering *et al* [3] propose time-optimal motions using a bang-bang control, Rajan proposes a two steps minimization algorithm [4], temporal/torque constraints are considered in the works of Shin and McKay [5] and Bobrow *et al* [6] and Kyriakopoulos and Saridis propose minimal jerk control [7]. 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 [8] 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 [9] shown that use of jerk provides smoothness, then a third order trajectory planner looks like a good solution.

According to Nelson [10] 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 [11] 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 extense survey can be found in [12] and a complement for manipulation in [13]. We must consider the *visual guided systems*, where the target is obtained from the vision system.

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

ISBN 970-769-070-4, ISRA 2006 Proceedings

control loop is used by assuming that the robot dynamics is negligeable. In this work internal and visual feedback are used in a shared position - vision control, although we consider to extend our works with force feedback. A second experiment is described, where 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 [14], 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 [15], 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 it task space. We have chose the last one.

The first works in the area refers to Paul [16] and Taylor [17]. 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 [18] use a single quintic polynomial for representing the entire trajectory, while Macfarlane [19] extend Andersson's work and uses seven quintic polynomials for industrial robots.

In the case of human interaction Amirabdollahian *et all* [20] use a seventh order polynomial while Seki and Tadakuma [21] propose the use of fifth order polynomial, both of them for the entire trajectory with a minimum jerk model. Herrera and Sidobre [22] 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.

Peters *et al* [24] use visual servoing to guide a robot in a grasping task using 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 [18] 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 Planner

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 [22] the Soft motion can be found by the next planner. We consider firstly the monodimensional and secondly the multidimensional extension.

A. Monodimensional Case

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

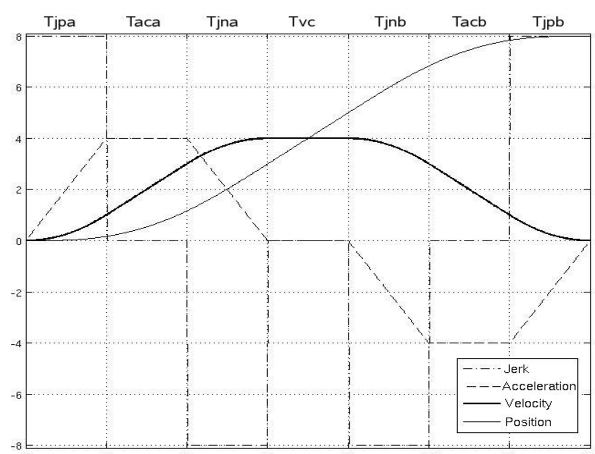

Fig. 1. Jerk, Acceleration, Speed and Position curves

The motion can be separated in seven segments, defined by the time period. 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 (b for differencing from a)
T_{acb}	Acceleration constant time
T_{jpb}	Jerk positive time

Considering one dimension motion and limit conditions, we can find three different canonical cases:

- The motion with a *maximum jerk* (J_{max}):
  $J(t) = J_{max}$
  $A(t) = A_0 + J_{max}t$
  $V(t) = V_0 + A_0 t + \frac{1}{2}J_{max}t^2$
  $X(t) = X_0 + V_0 t + \frac{1}{2}A_0 t^2 + \frac{1}{6}J_{max}t^3$
- The motion with a *maximum acceleration* (A_{max}):
  $J(t) = 0$
  $A(t) = A_{max}$
  $V(t) = V_0 + A_{max}t$
  $X(t) = X_0 + V_0 t + \frac{1}{2}A_{max}t^2$
- Finally, the equations for the motion with a *maximum velocity* (V_{max}):
  $J(t) = 0$
  $A(t) = 0$
  $V(t) = V_{max}$
  $X(t) = X_0 + V_{max}t$

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.

According to figure 1, the motion is realized at limit conditions. To achieve A_{max} from initial condition $A(0) = 0$, we have a jerk time (T_j) that is equal to the time for going from A_{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

$$T_j = T_{jpa} = T_{jna} = T_{jnb} = T_{jpb}$$

$$T_a = T_{aca} = T_{acb} \qquad T_v = T_{vc}$$

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_{max},A_{max} or V_{max}), we obtain a *minimal time motion*. The complexity of the equations system depends on the relation between distance and maximal limits.

1) Point to point motion: 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 ($A(t_f) = 0$ and $V(t_f) = 0$). Using this conditions we can find the necessary times T_{jmax} to achieve A_{max} and T_{amax} to achieve V_{max}.

$$T_{jmax} = \frac{A_{max}}{J_{max}} \qquad T_{amax} = \frac{V_{max}}{A_{max}} - \frac{A_{max}}{J_{max}} \qquad (1)$$

According to this, we can build the Figure 2. Where we can see the velocity profile and the position reached for each profile. In the case 1, we have $T_a = T_{amax}$ and $T_j = T_{jmax}$ that leads to V_{max} while T_v is free, the maximal displacement is function of T_v. In the case 2, we have $T_v = 0$, $T_a = T_{amax}$ and $T_j = T_{jmax}$, V_{max} is reached but there are no motion with this velocity. In the case 3, $T_v = 0$, $T_a = 0$ and $T_j = T_{jmax}$, that leads to A_{max} but there are no motion with this acceleration.

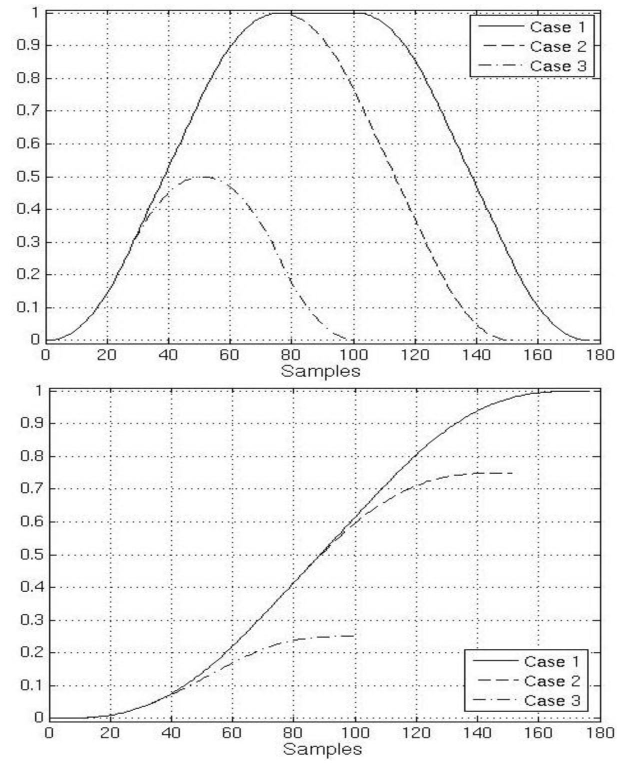

Fig. 2. Velocities and Positions

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

$$D = P_f - P_o \qquad (2)$$

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

  $$T_j = T_{jmax} \qquad T_a = T_{amax} \qquad T_v = 0$$

  we find

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

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

  $$T_j = T_{jmax} \qquad T_a = 0 \qquad T_v = 0$$

  we can find a distance (D_{thr2})

  $$D_{thr2} = 2\frac{A_{max}^3}{J_{max}^2} \qquad (4)$$

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

Since, acceleration and speed curves are symmetrical, the optimal time under these constraints for the trajectory is given by:

$$T_f = 4 * T_j + 2 * T_a + T_v \qquad (5)$$

Calculate distance D (Eq 2)
if $D \geq D_{thr1}$ **then**

$$T_j = T_{jmax} \qquad T_a = T_{amax} \qquad T_v = \frac{D - D_{thr1}}{V_{max}}$$

else
if $D \geq D_{thr2}$ **then**

$$T_v = 0 \qquad T_j = T_{jmax}$$

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

else

$$T_v = 0 \qquad T_a = 0 \qquad T_j = \sqrt[3]{\frac{D}{2J_{max}}}$$

end if
end if

2) 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_c 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 applied 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 applied 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. For 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.

In the case of *off-line trajectory planning*, 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:

$$T_c = T_{jpa} + T_{aca} + T_{jna} + T_{vc} \tag{6}$$

In the case of *on-line trajectory planning*, the current conditions are determined by the instant when the new target is defined.

B. Multidimensional Case

For the multidimensional case, we keeps this 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 motions to this time. Time adjusting is doing 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.

In the case of robot's end effector we use seven motion dimensions. Three dimensions for translation and four for rotation modelized by quaternion. Linear velocities $\mathbf{V}$ obtained can be applied directly as velocity references. On another hand, the evolution of the quaternion $\dot{\mathbf{Q}}$ must be transformed into angular velocities. We use the transformation function proposed in [27].

$$\begin{bmatrix}\mathbf{\Omega}\\0\end{bmatrix} = 2\mathbf{Q_r}^\top\dot{\mathbf{Q}} \qquad \text{where} \qquad \mathbf{Q_r} = \begin{bmatrix} n & k & -j & i\\ -k & n & i & j\\ j & -i & n & k\\ -i & -j & -k & n\end{bmatrix}$$

IV. Control loop

A. Arm manipulator control loop using quaternion feedback

The configuration of six joints arm manipulator is defined by a vector θ of six independent *joint coordinates* which correspond to the angle of the articulations.

$$\theta = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 & q_5 & q_6\end{bmatrix}^T$$

The *Pose* of the manipulator's end effector then is defined by M_b independent coordinates said *Operational Coordinates* which gives the position and the orientation of the final body in the reference frame. The advantages of using quaternions are largely exposed in [28].

We define $\mathbf{P}$ for the position and $\mathbf{Q}$ for the orientation

$$\mathbf{P} = \begin{bmatrix}x\\y\\z\end{bmatrix} \qquad \mathbf{Q} = n + \mathbf{q} \qquad \text{where} \qquad \mathbf{q} = \begin{bmatrix}i\\j\\k\end{bmatrix}$$

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 [29] has shown that the speed of the axis is given by

$$\dot{\theta} = \mathbf{J^{-1}} \begin{bmatrix} \mathbf{V} \\ \mathbf{\Omega} \end{bmatrix} \quad (7)$$

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

In a closed loop control [30], the control law is replaced by

$$\dot{\theta} = \mathbf{J^{-1}} \begin{bmatrix} \mathbf{V} - \mathbf{K_p e_p} \\ \mathbf{\Omega} - \mathbf{K_o e_o} \end{bmatrix} \quad (8)$$

where $\mathbf{K_p}$ and $\mathbf{K_o}$ are diagonal gain matrices, and $\mathbf{e_p}$ and $\mathbf{e_o}$ respectively represent position and orientation error vectors. Yuan [31] uses quaternion feedback in a close loop resolved rate control. The position and orientation tracking error are defined by

$$\mathbf{e_p} = \mathbf{P} - \mathbf{P_d} \qquad \mathbf{e_o} = n_d\mathbf{q} - n\mathbf{q_d} + \mathbf{q_d} \times \mathbf{q} \quad (9)$$

where the index d indicates that they are set points.

Yuan [31] 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 by a proportional integral digital controller of the form:

$$u[k] = u[k-1] + \Delta u[k] \quad (10)$$

with

$$\Delta u[k] = C\left((e[k] - e[k-1]) + \frac{T}{T_i}e[k]\right) \quad (11)$$

To achieve soft motion, we have limited the control law. By limiting $\Delta u[k]$, we limit the acceleration. By limiting $u[k]$, we limit the velocity 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 presence of singularity, 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 to 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.

B. 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.

Our 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 ($\mathbf{P}_i$) loosing one degree of freedom (z_i). The point in camera frame ($\mathbf{P}_c$) can be found by

$$x_c = d\frac{x_i - u_0}{\alpha_u} \qquad y_c = d\frac{y_i - v_0}{\alpha_v} \qquad z_c = d$$

where α_u, α_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 ($\widehat{\mathbf{P}}_w$) from image reconstruction can be found by

$$\widehat{\mathbf{P}}_w = \mathbf{T}_h^w \mathbf{T}_c^h \mathbf{P}_c$$

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

$$\mathbf{e}_w = \mathbf{P}_g - \widehat{\mathbf{P}}_w$$

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

Corke [12] proposes the use of open loop integrators. For each direction, we use a law of control of the form

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

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

$$\mathbf{P}_d^* = \mathbf{P}_d + \mathbf{u}_v$$

The figure 3 shows the control loop.

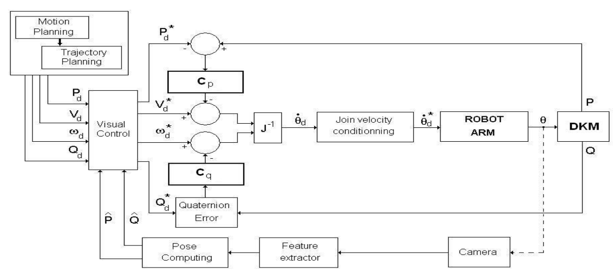

Fig. 3. Control Loop for Visual Servoing Task

C. Visually guided control loop

In this case, the target is defined by the vision system. At the instant that a new target is defined, the algorithm of trajectory planning is applied from the current motion conditions used as Initial Conditions and the target as final position with acceleration and velocity null. The figure 3 shows the control loop.

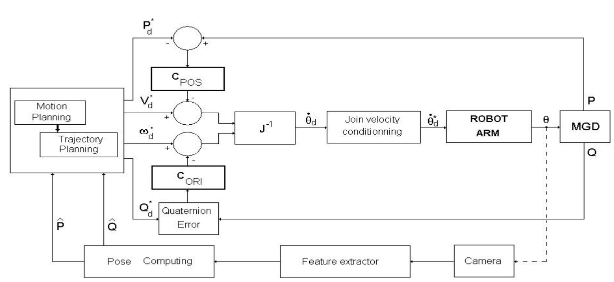

Fig. 4. Control Loop for Visual Guided application

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-Hartemberg parameters: a2 = 0.450 m, r4 = 0.480 m and r6 = 0.30 m (Figure 5). The software control is developed using Open Robots tools. The sampling time is fixed to 10 ms.

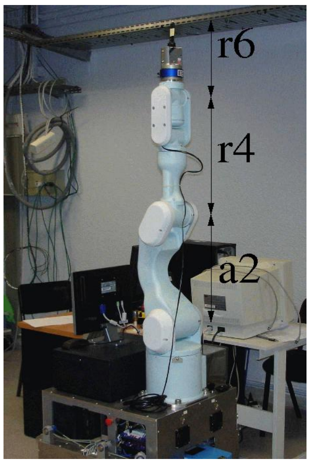

Fig. 5. Robot-Arm

The joint velocities are limited to

Joint	q1	q2	q3	q4	q5	q6
Velocity limit (rad/s	0.5	0.5	0.5	1.5	1.5	1.5

The linear and angular end effector motion are limited to

	Linear Limit	(Angular limit)
J_{max}	$0.900 m/s^3$	$0.600 rad/s^3$
A_{max}	$0.300 m/s^2$	$0.200 rad/s^2$
V_{max}	$0.150 m/s$	$0.100 rad/s$

B. Trajectory planning for fixed trajectory

Consider the Initial configuration for the robot (Figure 5), the trajectory planning of 100 configurations, defined off-line, produces a motion defined in the figure 6.

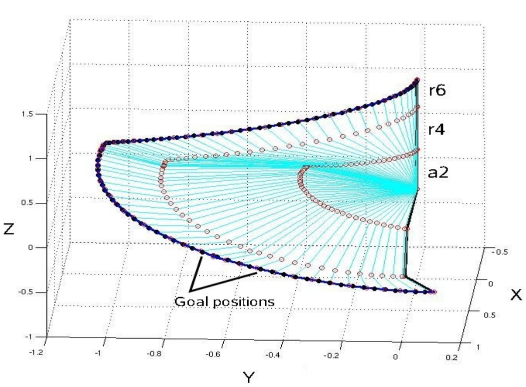

Fig. 6. Manipulator evolution during trajectory tracking

The error along the trajectory is of the order of milimeters. This kind of trajectory planning, has been evalued for grasping tasks. When the model environment is perfect, the grasping is done without problems, even in the case of singular configurations of the arm.

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 7 shows the comparison in Z direction with or without visual servoing. Finally, the figure 8 shows the evolution of the end effector considering the target line.

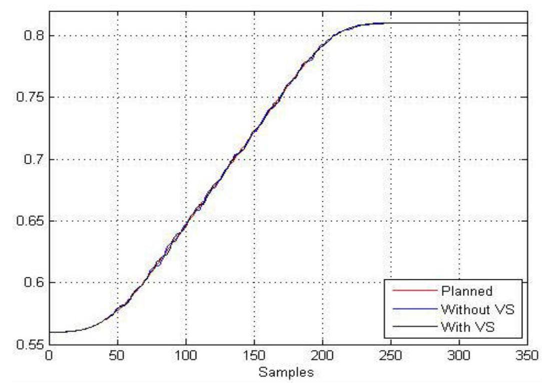

Fig. 7. Z motion comparation

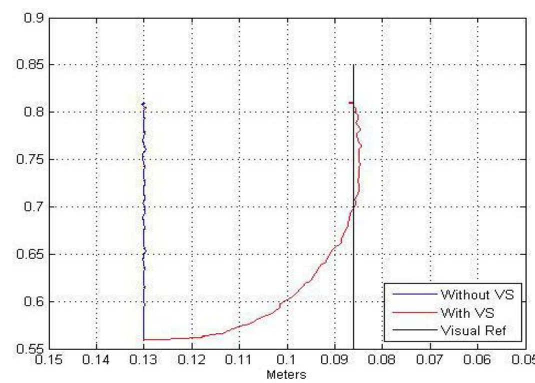

Fig. 8. Z vs Y, Correction of trajectory

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 first example is doing when the orientation is fixed. To avoiding the problems inherent of the measure of the target's position, the limits conditions on jerk, acceleration and velocity has been reduced. We assume that the target's velocity is reachable by the robot, to guarantee the tracking. The trajectory planner has shown, in this case his *on-line* capabilities. The computing of the trajectory between the current position and the target position is done in less than 10 mS.

VI. CONCLUSIONS

We have presented a general approach to manipulation control for service robot applications. The grasp planner produces a trajectory, the trajectory planner takes this information to produce the reference to the different servo-controllers. The control loop presented has a an excellent performance during the path tracking. This is advantageous when the model is perfect. For imperfect models, the use of visual feedback offers an alternative. In this paper, we shows two ways to introduce visual information in the control loop, visual servoing or visual guided.

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 accounts real-time modifications of curves.

Experimental example which represent a very early result, shows the validity of the approach. 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

[1] M. J. Er, "Recent developments and futuristic trends in robot manipulator control," in *Asia-Pacific Workshop on Advances in Motion Control Proceedings*, 1993, pp. 106–111.

[2] J. C. Latombe, *Robot Motion Planning*. Kluwer Academic Press, 1991.

[3] S. A. H. Hans P. Geering, Lino Guzella and C. H. Onder, "Time-optimal motions of robots in assembly tasks," *IEEE Transactions on Automatic Control*, vol. AC-31, pp. 512–518, June 1986.

[4] V. T. Rajan, "Minimum time trajectory planning," in *IEEE Proc. Int. Conf. on Robotics and Automation*, vol. 2, Mars 1985, pp. 759–764.

[5] K. G. Shin and N. D. McKay, "Minimum-time control of robotic manipulatos with geometric path constraints," *IEEE Transactions on Automatic Control*, vol. AC-30, pp. 531–541, June 1985.

[6] S. D. J. E. Bobrow and J. S. Gibson, "Time optimal control of robotic manipulators along specified paths," *Int. J. Robotic Research*, vol. 4, pp. 3–17, 1985.

[7] K. J. Kyriakopoulos and G. N. Saridis, "Minimum jerk path generation," in *Proc. IEEE International Conference on Robotics and Automation*, 1988, pp. 364–369.

[8] M. B. Paul Lambrechts and M. Steinbuch, "Trajectory planning and feedforward design dor high performance motion systems," in *American Control Conference*, Boston, June 2004.

[9] N. Hogan, "An organizing principle for a class of voluntary movements," *The Journal of Neuroscience*, vol. 4, pp. 2745–2754, November 1984.

[10] B. J. Nelson, J. Morrow, and P. H. Kohsla, "Robotic manipulation using high bandwidth force and vision feedback," *Mathl. Comput. Modelling*, vol. 24, pp. 11–29, 1996.

[11] H. B. Johan Baeten and J. D. Schutter, "Shared control in hybrid vision/force robotic servoing using the task frame," in *Proc. IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, Lausanne Switzerland, October 2002, pp. 2128–2133.

[12] S. H. Gregory D. Hager and P. Corke, "Tutorial tt3: A tutorial on visual servo control," in *IEEE Robotics and Automation Conference*, Minneapolis, April 1996.

[13] D. Kragic and H. I. Christensen, "Survey on visual servoing for manipulation," Computational Vision and Active Perception Laboratory, Fiskartorpsv 15 A 10044 Stockholm, Sweden, Tech. Rep. ISRN KTHA/NA/P-02/01-SE, 2002.

[14] E. Lopez-Damian, D. Sidobre, and R. Alami, "A grasp planner based on inertial properties," in *Proc. IEEE International Conference on Robotics and Automation*, Barcelona, Spain, April 2005, pp. 766–771.

[15] T. J. M. Brady, J. Hollerbach and T. Lozano-Perez, *Robot Motion, Planning and Control*. Cambridge, Massachusetts: The MIT Press, 1982.

[16] R. P. C. Paul, "Manipulator cartesian path control," *IEEE Trans. Syst., Man Cybern.*, vol. 9, pp. 702–711, May 1979.

[17] R. H. Taylor, "Planning and execution of straight line manipulator trajectories," *IBM Joournal of Research and Development*, vol. 23, pp. 424–436, 1979.

[18] R. L. Andersson, "Aggressive trajectory generator for a robot ping-pong player," *IEEE Control Systems Magazine*, vol. 9, pp. 15–20, February 1989.

[19] S. Macfarlane and E. Croft, "Jerk-bounded manipulator trajectory planning: Design for real-time applications," *IEEE Transactions on Robotics and Automation*, vol. 19, pp. 42–52, February 2003.

[20] R. L. Farshid Amirabdollahian and W. Harwin, "Minimum jerk trajectory control for rehabilitation and haptic applications," in *IEEE International Conference on Robotics and Automation*, May, 2002, pp. 3380–3385.

[21] K. Seki and S. Tadakuma, "Minimum jerk control of power assiting robot based on uman arm behavior characteristic," in *International Conference on Systems, Man and Cybernetics*, 2004, pp. 722–721.

[22] I. Herrera and D. Sidobre, "On-line trajectory planning of robot manipulator's end effector in cartesian space using quaternions," in *15th Int. Symposium on Measurement and Control in Robotics*, 2005.

[23] A. Castano and S. Hutchinson, "Visual compliance: Task directed visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 10, pp. 334–342, June 1994.

[24] M. E. C. R. A. Peters, M. Bishay and K. Negishi, "Visual servoing for a service robot," *Robotics and Autonomous Systems*, vol. 18, pp. 213–224, 1996.

[25] G. D. Hager, "A modular system for robust positioning using feedback from stereo vision," *IEEE Transactions on Robotics and Automation*, vol. 13, pp. 582–595, August 1997.

[26] J. Lloyd and V. Hayward, "Trajectory generation for sensor-driven and time-varying tasks," *International Journal Robotics Research*, vol. 12, pp. 380–393, August 1993.

[27] H. Bruyninckx and J. Shutter, "Introduction to intelligent robotics," Katholieke Universiteit de Leuven, Tech. Rep., 2001.

[28] J. Funda, R. H. Taylor, and R. P. Paul, "On homogeneous transforms, quaternions and computational efficiency," *IEEE Trans. on Robotics and Automation*, vol. 6, pp. 382–388, Juin 1991.

[29] D. Whitney, "Resolved motion rate control of manipulators and human prostheses," *IEEE Trans. Man-Machine Syst.*, vol. 10, pp. 47–53, Juin 1969.

[30] C. H. Wu and R. P. Paul, "Resolved motion force control of robot manipulator," *IEEE Trans. Syst., Man Cybern.*, pp. 266–275, May 1982.

[31] J. S. C. Yuan, "Closed-loop manipulator control using quaternion feedback," *IEEE J. Robotics and Automotion*, vol. 4, pp. 434–440, August 1988.