Improving the Tracking Of Objects By A Robot During the Movements

Etienne Burdet

Institute of Robotics

Swiss Federal Institute of Technology (ETH Zürich)

CH-8092 Zürich

SWITZERLAND

Abstract: In this paper, a method is presented for fast learning of visually guided movements. The presented algorithms have been tested with a manipulator tracking manoeuvering target. Three parameters critical for the visuo-motor co-ordination have been identified, and are learned in less than one hour with repeated movements. The conditions for fast and precise vision have been investigated analytically, and the results of this analysis have been used for improving the image processing during the motions. After learning, the robot performs smooth and fast reaching movements and can easily drop small objects into the waggon of a moving model train. Finally, the method is generalised as a methodology for the representation of artificial systems, which provides them with the ability of adapting themselves to many different tasks. It is also explained how biological models can be used in this scheme.

Keywords: Visuo-motor Co-ordination, Learning Robots, Human Motor Control, Vision Processing, Motion Planning

1. Introduction

Because of the extended possibilities enabled by vision systems and the continuous price reduction of hardware, the use of robots guided by a vision system will probably increase drastically in the near future. However, to enable more use of these robots, it is necessary to eliminate some of their drawbacks. One of the most important drawbacks is the lack of flexibility to changes in the environment or the task. This paper presents algorithms enabling a robot to adapt its motion and its vision. The usual automatic calibration of the camera relative to the robot end-effector is extended to the calibration of parameters involved in the co-ordination of the motion with the vision processes. The resulting Adaptable Motion Behaviour (AMB) is tested by tracking manoeuvering targets with a manipulator guided by a vision system.

The tracking of moving objects has been investigated in many papers. Generally, the objects are reached only in a limited area, and the parameters are tuned by the programmer (Allen, 1992; Buttazo, 1994). We require, in contrast, a robot which optimises its parameters autonomously, and is able to reach objects in its whole workspace. In (Wen, 1995; Aboaf, 1988), algorithms for improving the returning of a ball have been proposed. The learning was performed off-line using a free body model of the ball trajectory. Tracking a manoeuvering object is a more difficult task. As well as fast vision processing, efficient tracking algorithms are required, the motion planning has to deal with changes of the target position, and the motor and vision processes have to be closely related. The visuo-motor co-ordination is particularly difficult to model and will therefore be learned.

Before using the robot to track objects, a calibration procedure consisting of several steps must be performed. Efficient reaching movements are obtained by optimising the vision processing with respect to time and precision and by adapting the visuo-motor co-ordination in three steps. Firstly, the spatial transformation of the end -effector with images is identified during specified motions of the manipulator. Secondly, the motor and visual processes are synchronised during repeated movements. Finally, using a human strategy, the velocity is modulated to optimise reaching movements with respect to time and smoothness.

The AMB has been tested in several experiments (Figure 1), and demonstrated by dropping objects into a moving waggon. It is worth noting that almost all figures result from measurements performed during the experiments. Figures resulting from simulation will be specially indicated. In the video (Burdet and Müller, 1996), the robot can be seen while learning reaching movements and loading the waggon.

Two conditions were imposed for the experiments. Firstly, as the algorithms should be usable on generally available systems, only standard low-cost hardware was used. In

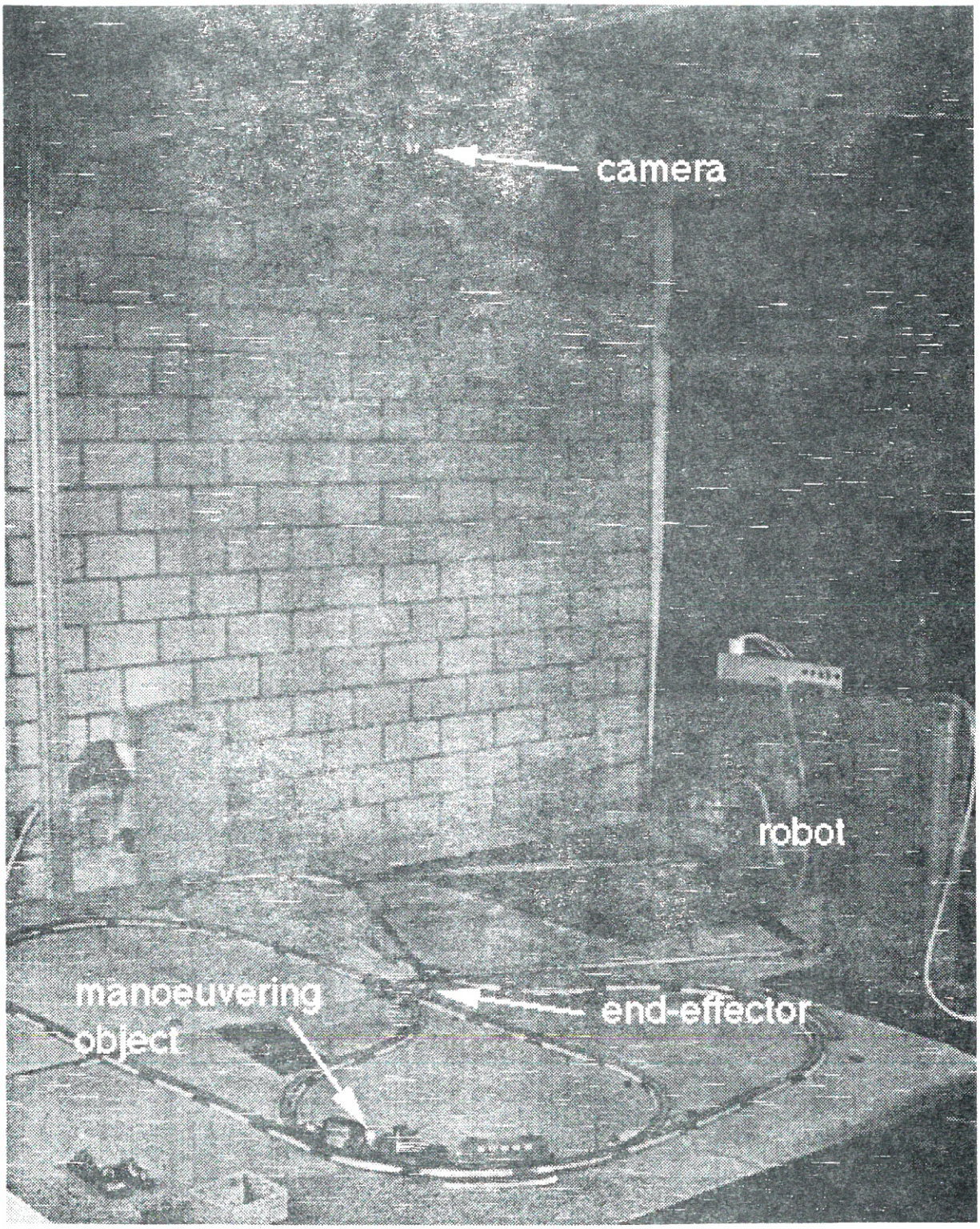

Figure 1. Hardware Used in the Experiments

particular, the vision system consisted of a frame grabber using a 68020 board for vision processing. This simple vision system can only analyse the images slowly. We have therefore investigated analytically which conditions ensure fast and precise vision. Secondly, the learning had to take place automatically and in a short time, so that it could be performed frequently.

From the experience acquired in the design of the AMB, general principles for the representation of robot systems have been deduced which should enable them to perform

many different tasks and adapt their behaviour by repeated tasks.

Adaptable Motion Behaviour (AMB)

The control structure is schematically shown in Figure 2. The vision processing determines the object's position i_0 in pixel co-ordinates. This image position is transformed into a Cartesian position x_{0m} (Section 3.1). The visual tracking corrects for the *reaction time RT*, due to the vision processing and motor reaction, and predicts the actual object's position x_{0p} (Section 2.2). The motion planner determines the trajectory to attain this position and generates velocity and acceleration bounded movements using the *velocity scaling factor* λ (Section 2.3). Finally, the desired trajectory is realised by the controller using feedback from the encoders. q, q_d are the actual and desired joint positions of the manipulator, and e is their difference.

Before the robot can efficiently track moving objects, the vision processing, the hand-eye transformation, the reaction time and the optimal velocity scaling factor are learned during specified motions (Section 3).

2. Description of the Robot

2.1 Hardware

The robot consists of a parallel manipulator moving in a horizontal workspace of 3x2 m^2 and a simple vision system. The manipulator has 3 degrees- of- freedom: the position (x_1, x_2) and the orientation φ. Three VME boards are used to exert the control: A frame grabber stores and thresholds the image taken by the camera, a 68020 board executes the vision algorithm, and a 68040 board plans and controls the arm movements.

The moving object is a light bulb fixed to a model train. The train's rails are arranged in an irregular path, which occupies the whole of the manipulator's workspace, and contains a straight segment which can be repeated for learning. As the robot sees only the object (and not the rails), the object's movement can be considered to be unpredictable.

2.2 Visual Tracking

The unique information the robot has about the object manoeuvre, is that the changes of direction and velocity are performed continuously at a limited rate. It was therefore modelled so that the object acceleration $\ddot{x}$ is a zero mean random variable with autocorrelation function decreasing with time:

$$E\left[\ddot{x}(.)\ddot{x}(.+\tau)\right] = \sigma_m^2 \exp\left(-\beta|\tau|\right). \qquad (1)$$

$E\left[f(.)\right]$ stands for the expected value $\int f(t)dP(t)$, σ_m is the standard deviation of the acceleration, β the *manoeuvering rate,i.e.* the rate of change of the acceleration, and τ the time variable. The state equation of the object motions in discrete time is then (Singer, 1970):

$$\begin{bmatrix} x_{k+1} \\ \dot{x}_{k+1} \\ \ddot{x}_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & T & \dfrac{\beta T - 1 + e^{-\beta T}}{\beta^2} \\ 0 & 1 & \dfrac{1 - e^{-\beta T}}{\beta} \\ 0 & 0 & e^{-\beta T} \end{bmatrix} \begin{bmatrix} x_k \\ \dot{x}_k \\ \ddot{x}_k \end{bmatrix} + V_k \ , \quad (2)$$

where $x_k = \left(x_1(k), x_2(k)\right)$ is the robot position at time k, T the sampling time of the visual analysis and V_k a vector of Gaussian white noise. The observation equation is

$$z_k = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_k \\ \dot{x}_k \\ \ddot{x}_k \end{bmatrix} + W_k \ , \qquad (3)$$

where z_k is the position measurement at time k and W_k a Gaussian white noise corresponding to the measurement error. The trajectory of the manoeuvering object is modelled by Equations (2)and (3).

The object's state is calculated using a Kalman-filter, which guarantees the best estimation in the least square sense (A.E. Bryson, 1975). As a delay occurs, due to image processing and the corresponding motor reaction time, the filter gives an estimate of the object past state. The actual object position is predicted using this estimate and Equation (2) with $V_k \equiv 0$.

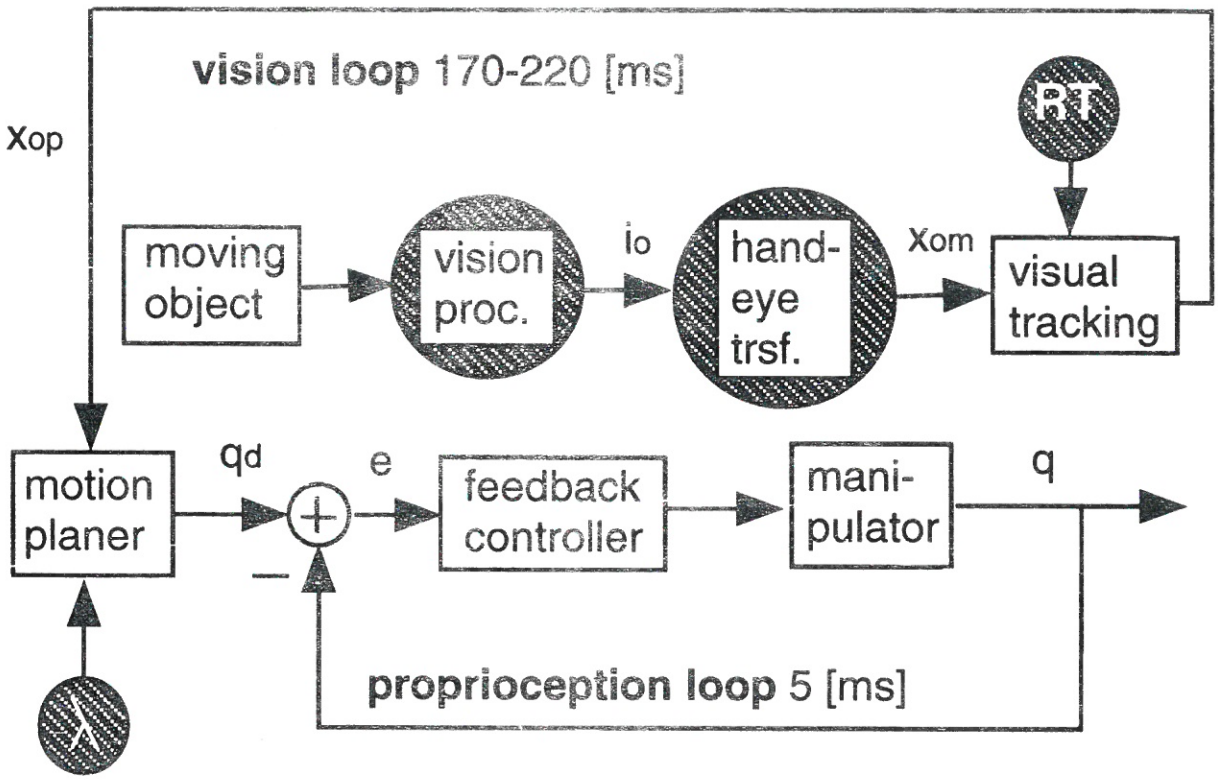

Figure 2. Control Scheme of the Visually Guided Robot (see text of Section 1 for explanations)

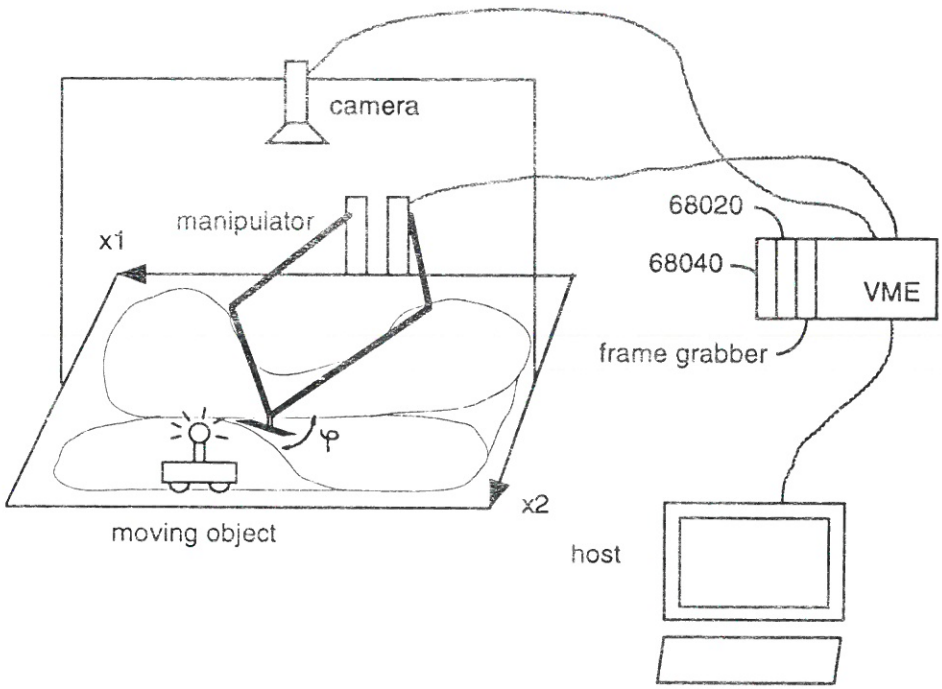

Figure 3. Experimental Apparatus. The horizontal scene is filmed by the camera. The control is executed by the boards contained in the VME-rack (see text of the Section 2.1)

The tracking has been tested by simulating the motions along a complex trajectory. Resulting position functions are shown in Figure 4. The estimated position is superimposed onto the real one. The prediction (corresponding to the actual position) generally follows the real trajectory, shows that the expected positions are approximately Gauss- distributed around the measured positions, which corresponds to the state equation (2).

Simulations also helped in the choice of the

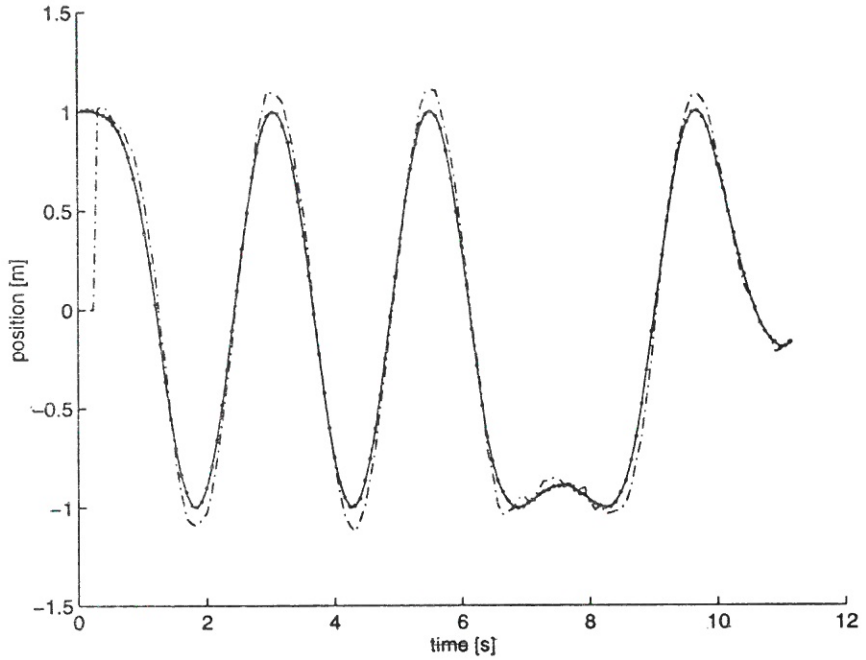

Figure 4. Tracking Along a Complex Trajectory (Simulation). The different curves correspond to real (-), estimated (.) and predicted (-.) positions. The estimated positions are superimposed onto the real ones.

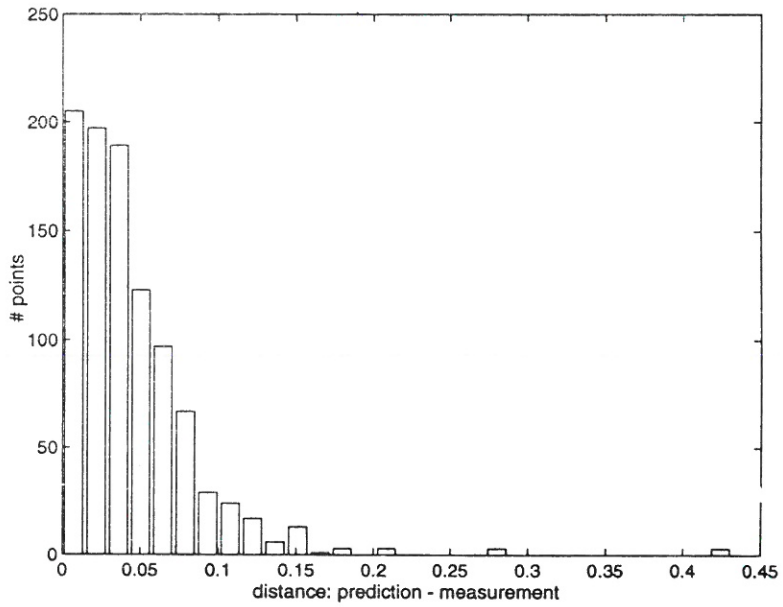

Figure 5. The Measured Object Positions are Nearly Gauss-Distributed Around the Expected Positions

but needs some time to deal with the changes of acceleration. It is not possible to measure the prediction error in the implementation, because the real position is not known, however the results of Section 4 will show that the tracking is also precise in reality. Figure 5

tracking algorithm: Modelling the moving object using a second order model led to estimation errors of about 5 cm, which is more than 7 times larger than with our algorithm. Our model, assuming continuously varying acceleration, also led to better results than a

model assuming abrupt variation of the acceleration (Allen, 1991), and a simple second order filter (Buttazo, 1994).

2.3 Motion Planning

As the robot has to track a moving object, the robot motion planner has to react to changes of the object's position during the movement. In addition, we required that the movements of the manipulator were smooth and fast. Planning using techniques from optimal control theory has been investigated, but required to solve a sixth order polynomial, which must be performed numerically and is impossible to realise in real-time applications with simple hardware (Müller, 1995). An alternative solution had to be found.

A new flexible motion planner has been built which plans smooth motions and is able to handle sensor signals or user inputs at any time (Luthiger, 1996). With this planner it is, for example, possible to reset the destination during the movement. It calculates on-line the next motion increment as a function of the actual state, the destination and upper bounds for the Cartesian velocity and acceleration (Figure 6). This motion planner was used in our experiments. Each incoming expected object position is given as a target to the planner, which automatically tracks it.

To be able to tune the motions kinematics, a characteristic feature of the movements of the human arm was used. From (Atkeson and Hollerbach, 1985), it is known that if a movement $x(t)$ during a time T_1 has to be

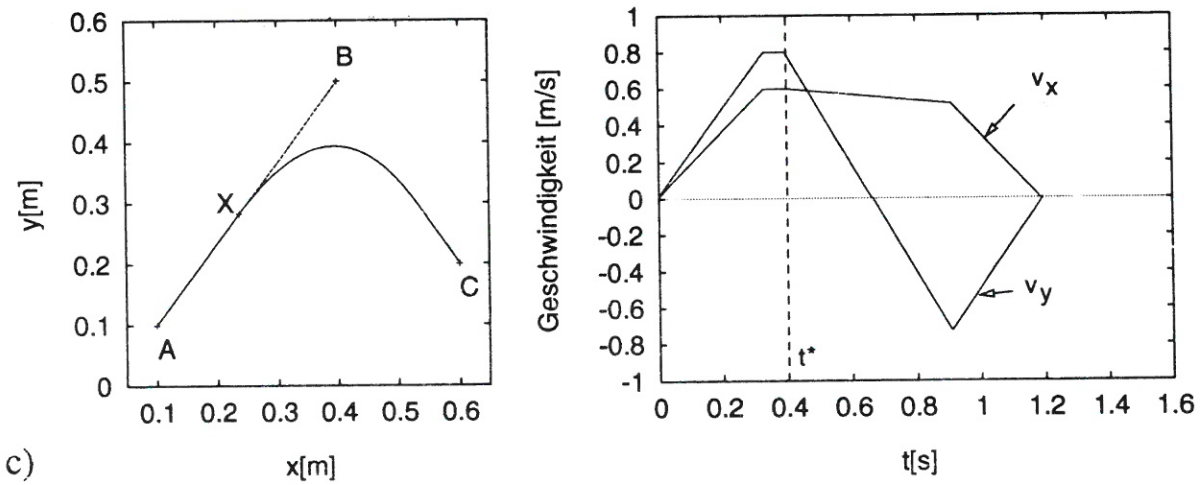

Figure 6. Velocity Profiles Along the x and y Coordinates of a Reaction to a Change of the Target Occurring at Time t^* with the Motion Planner Used in the Experiment

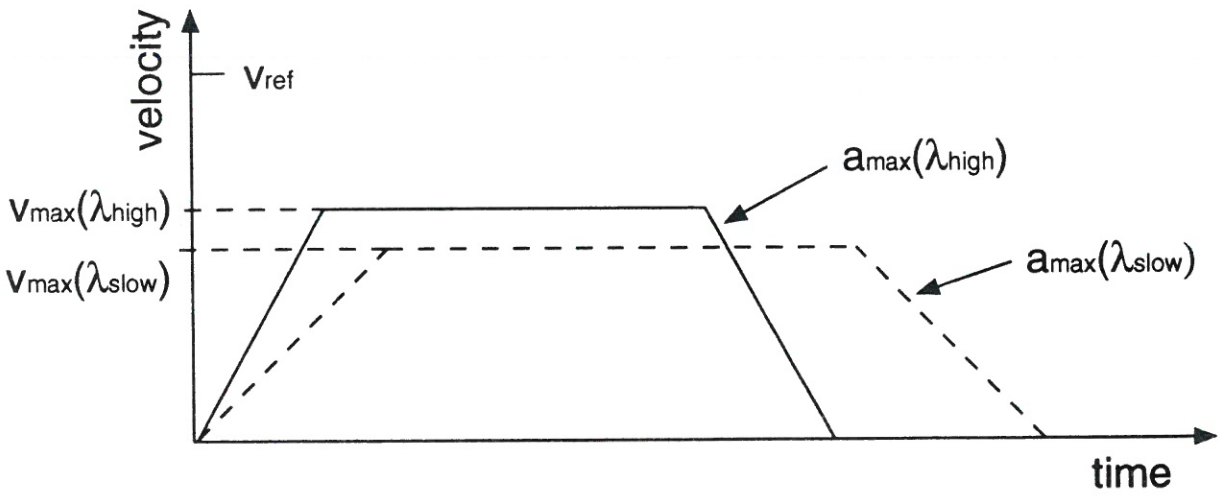

Figure 7. The Velocity Scaling Factor λ Determines the Velocity and the Acceleration of the Movements. It Acts on the Movement Duration and Its Smoothness.

accomplished slower or faster, i.e. in a time T_2 with $T_2 \neq T_1$, the time is simply linearly scaled:

$$x^*(t) = x(\lambda t), \quad \lambda \equiv \frac{T_1}{T_2}. \tag{4}$$

With our motion planner, the movements' kinematics is determined by upper bounds v_{max} and a_{max} for the Cartesian velocity and acceleration. Similar to the "human strategy", the kinematics of robot motions was determined using the following *λ-scaling*:

$$v_{max} \equiv \lambda\, v_{ref}\,, \quad a_{max} \equiv \lambda^2\, a_{ref}\,, \tag{5}$$

where v_{ref} and a_{ref} are constants depending on the manipulator dynamics and the workspace. With this parameterisation, the kinematics can be varied using only the *velocity scaling factor* λ (Figure 7). This parameter controls not only the motion velocity but (as it acts on the acceleration) also its smoothness.

2.4 Fast and Precise Vision Processing

The position of a moving object is determined by calculating the mass center of the white pixels in the binary black and white image. The time required for this operation is approximately proportional to the number of pixels to be examined. Examining the 512x512 pixels of one entire image requires more than 18 s. For real-time applications, it is therefore necessary to restrain the visual analysis to the important pixels.

During the time needed for the analysis of one image, only a limited number of pixels can be examined. We searched for a strategy, thereby the object position could be determined with the maximum possible precision and in minimum time. Accordingly, to this aim, $a \bullet t$, a being the image resolution and t the time needed for vision processing, was used as cost function for the visual analysis. The expected value of this function has obviously to be minimised, i.e.

$$J \equiv E[a \cdot t] \equiv \int a(h)\, t\, dP(h) \tag{6}$$

has to be minimised, with h the two-dimensional position vector in the image and $P(h)$ the distribution of the object positions around the expected positions, i.e. Gauss-distributed (Figure 5).

We used in (Burdet and Luthiger, 1994) a Kohonen neural network for adapting the image processing to the distribution of the measured objects' positions around the expected objects', position. Here, because this distribution is a Gaussian one, a simpler algorithm has been designed. The object is visually tracked along typical trajectories using the algorithm described at Section 2.2, and the error between the expected position and the effective position is calculated. This gives rise to statistics as shown in Figure 4. The distribution of the object positions around the expected positions is approximately Gaussian. The standard deviation of this distribution determines the image processing in the following way: the co-ordinates of the pixels, which will be analysed, are random numbers following a Gaussian distribution with the same standard deviation. These pixel co-ordinates are stored in a Table and added to the expected position calculated by the tracking algorithm to form the window in which the object will be searched.

This process is performed directly after the calibration of the camera relative to the robot workspace described in the next Section. Using this special window instead of analysing the whole image enabled us to reduce the time necessary for analysing one image from 18 s to 80 ms. 80 ms is the minimal time possible with the simple vision system used in our experiment (Müller, 1995).

3. Learning the Visuo-Motor Co-ordination

The visual analysis and motion planning have been treated at Section 2. To reach objects, in addition to a fast, precise visual analysis, and smooth reaction movements, it is necessary that the movements are perfectly co-ordinated with vision processing. How to realise this, will be explained in this Section. Three features are critical for the visuo-motor co-ordination. They are:

- the geometrical transformation between the image and the position of the end-effector (if this is not precise, the robot will move to a wrong location)
- the temporal synchronisation of the motor and vision processes (If this is not perfect,

the robot will move to a past or future position)

- the velocity of the overall movement (if this velocity is too high, the movement is jerky and too fast in comparison to the (slow) visual analysis. This will disturb the manipulation).

These three factors can hardly be determined by a calculation and must therefore be learned. These factors are also not independent. They must therefore be learned in three steps in the order of the above list. Learning procedures and their implementation will be described in the next three Sections. After each step, it was tested if the sub-task had been well learned and if it had in fact contributed to improving the reaching motions.

3.1 Identifying the Hand-Eye Transformation

In order to reach the object accurately, it is necessary that the transformation of the Cartesian position of the end -effector in image coordinates is properly identified. This transformation has also to be very fast to save processing time.

The hand-eye transformation is acquired as follows. A lamp is fixed to the end -effector, which is moving to the nodes of a regular grid in the robot workspace. At each node, the position of the lamp in the Cartesian space and in the image are recorded in a Table. The transformation of any point is then performed using a linear interpolation of its neighbouring nodes. This transformation requires only 14 multiplications and 15 additions independent of the precision (to take more points would simply increase the Table). Using a grid of 16x8=128 points, we obtained a mean error of 2 mm. This corresponds to half the size of a pixel and is two times smaller than the measurement noise.

Three alternative methods were also examined. Using a single linear transformation for the whole workspace was rejected, because the standard deviation of the error was too large (47 mm). To reach the same precision as our method, a feedforward neural network or a Kohonen neural network required more than two times extra operations to perform the hand-eye transformation (Müller, 1995).

3.2 Learning the Reaction Time

If the estimated object position is given to the motion planner as a target, the robot will follow the object, but with a delay. To reach the actual object position, it is necessary to correct the time required for image processing and motor reaction. This *reaction time (RT)* depends principally on the communication between the electronic boards, the timing of the processes on each board and on the motion planning. It cannot be accurately modelled and must therefore be learned. Our aim was to find the RT necessary for reaching the object in most of the cases.

Learning the RT was done as follows. The train carrying the target travels along a closed loop with constant speed (as constant as possible!). After the train passes a light detector, the robot tries to track the moving object during the 30 next images. The sum of the 30 distances between the end -effector and the object is used as a *measure of the synchronicity*. The same movement is performed 10 times and the mean synchronicity measure is minimised relative to the RT (It is necessary to use mean values, because reaching includes many different processes whose timing are only partially reproducible (Müller, 1995)). A stochastic optimisation was used to find the optimal RT. This means that the actual RT is perturbed stochastically, the corresponding values of the cost function using the old and the new RT are compared, and the RT corresponding to the smaller value of the cost function is used in the next movements.

The RT depends on the kinematics of the robot movements. Therefore, it was necessary to find the optimal RT for each velocity scaling factor λ. The result of the optimisation is shown in Figure 8. This function is stored in a Table and used to perform further movements.

It was first experimentally confirmed that the learned RT did not depend on the velocity of the moving object and was valid for other trajectories (Müller, 1995). Secondly, it must be checked that reaching is improved by optimisation. With this aim, the *grasping time* T_{grasp} was measured as a function of the RT, for different λ. T_{grasp} is defined as the first time at which the distance between the end -effector and the target is less than 1 cm in 4 consecutive images. Figure 9 shows the result for λ=1. As expected, the grasping time is small for the optimal RT and the associated standard

deviation is even smaller there than in any other point. For a lower RT, most of the trials fail. This means that the chosen cost function really improves the reaching movements.

3.3 Optimising the Velocity

Learning the correct RT enables the manipulator to reach the moving object in most of the cases. To improve the reaching movements, it

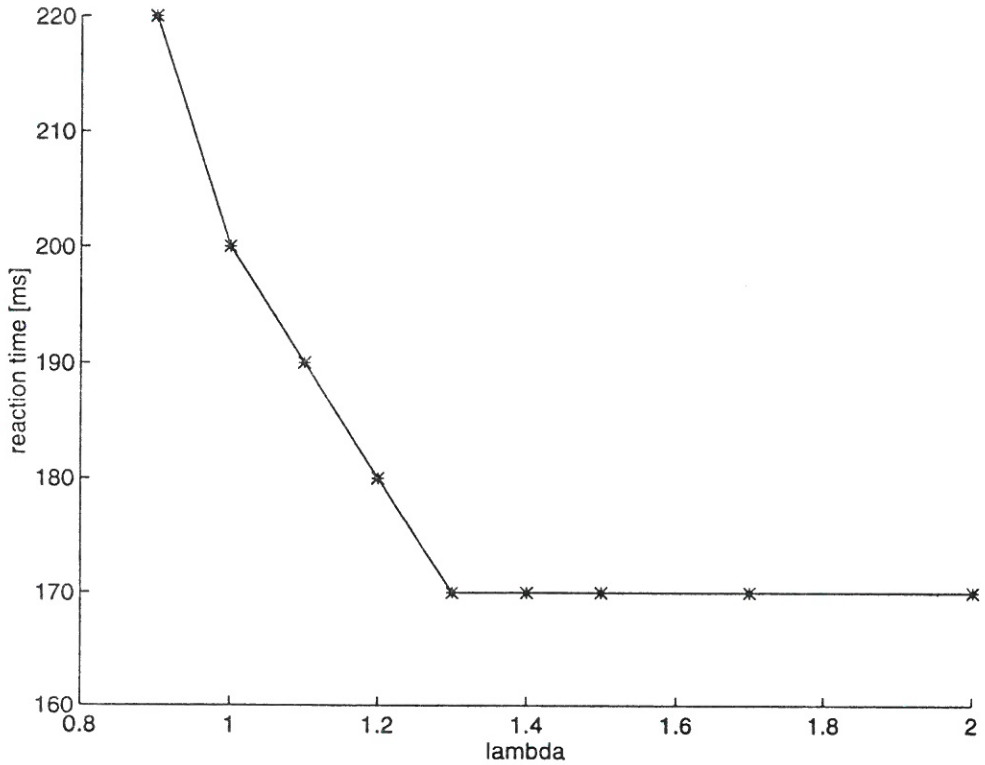

Figure 8. Learned Reaction Time as a Function of the Velocity Scaling Factor λ

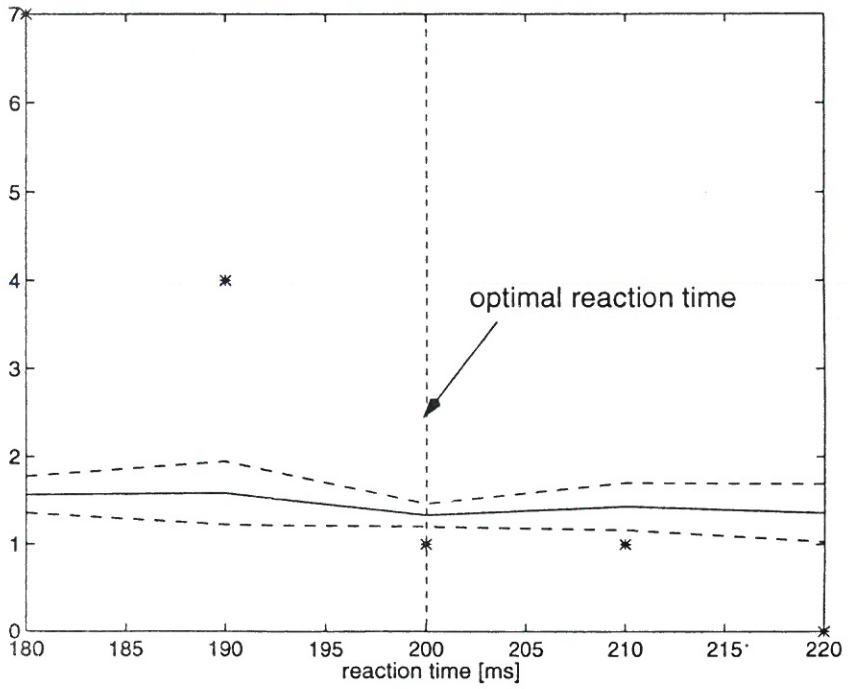

Figure 9. The optimal RT (=200 ms for λ=1) allows a better grasp. -: mean grasping time, - -: mean grasping time ± standard deviation, *: number of times missed

is important to use a velocity profile which maximises the accuracy and minimises the reaching time. In human arm motions with high accuracy, it has been observed that the velocity is adapted to the required accuracy and synchronised with the visual analysis (Milner, 1992). It was further shown that this was probably achieved by optimising the movements with respect to smoothness and speed (Hoff, 1992; Burdet, 1996). Following the human example, we chose to improve the motions by optimising them with respect to speed and smoothness.

$$J = \alpha \overline{T}_{grasp} + \lambda, \tag{7}$$

where $\overline{T}_{grasp}$ is the mean grasping time over 10 trials, λ the velocity scaling factor, and α an arbitrary constant. The minimisation of $\overline{T}_{grasp}$ allows the robot to reach the object quickly. λ determines the upper bounds of the Cartesian velocity and acceleration (see Equation (5)). Thus a small λ leads to smooth motion (Figure 10). Finally, α is controlling the relative importance of time to smoothness optimisation.

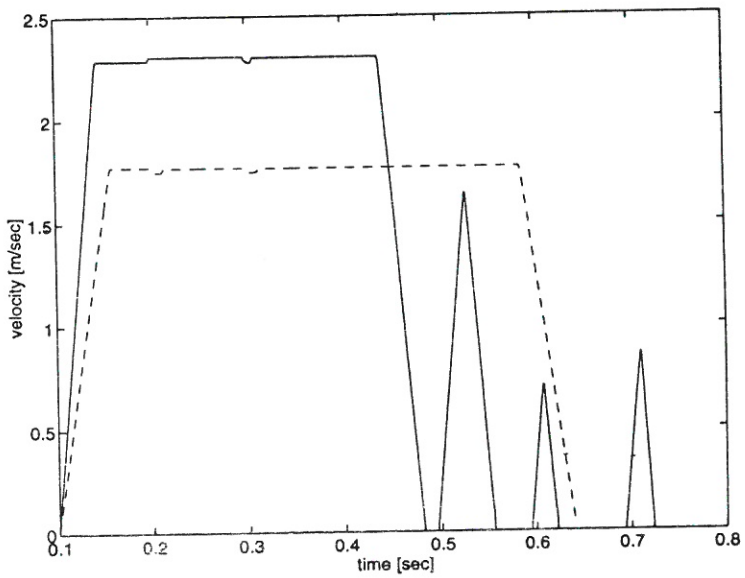

Figure 10. Trajectory with too motion velocity too high (continuous lines) and with optimal velocity (dashed line) resulting from a simulation. Trajectories with the velocity set too high are jerky because the robot sometimes travels too far and has to come back, or it has to stop, because the precision attained in the visual analysis is not yet sufficient.

There exist several algorithms for calculating the optimal trajectory between two points (A.E. Bryson, 1975; Brobow, 1988). Recent works have shown that it is even possible to consider in the optimisation the uncertainty inherent to real systems (Allen, 1993; Hoff, 1992). However, visually guided movements can significantly be modified at each time. Therefore, using these algorithms, it would be necessary to calculate the optimal trajectory at each time, which is impossible with simple hardware, because the optimisation requires a relatively long time. Thus, these methods are not usable for sensor guided movements. We searched for an alternative method, thereby the optimisation is valid not only for one movement, but for a large class of movements.

Let us consider the cost function

Thus the minimisation of this function should lead to smooth and fast reaching movements.

The improvement resulting from the optimisation is particularly important when the objects to be grasped require complex image processing. In this case, the movement can be started towards an approximate location, of which determination requires little image processing, and the target position is corrected with refined visual analysis during the motion. When the velocity bound is too high, the end-effector moves too far and has to go back, resulting in unnecessary oscillations, see Figure 12.

The task we have implemented consists of grasping a long object. Long objects must be grasped widthwise, thus this task requires the knowledge of the object's center of mass and

orientation. A fast algorithm for determining the orientation of long objects requiring only 1 multiplication/ pixel and 4 additions/ pixel, was devised (Müller, 1995). The position was determined as before. The object may be grasped only after the position and orientation have been accurately determined, i.e after about 1 s.

corresponding $\overline{T}_{grasp}$ and number of times missed have been measured (Figure 11). We can see that $\overline{T}_{grasp}$ is decreasing until λ=1.3, which corresponds to the learned value, and remains constant up to this point. The number of missed times is also minimal up to this point. Thus the learned value is really optimal.

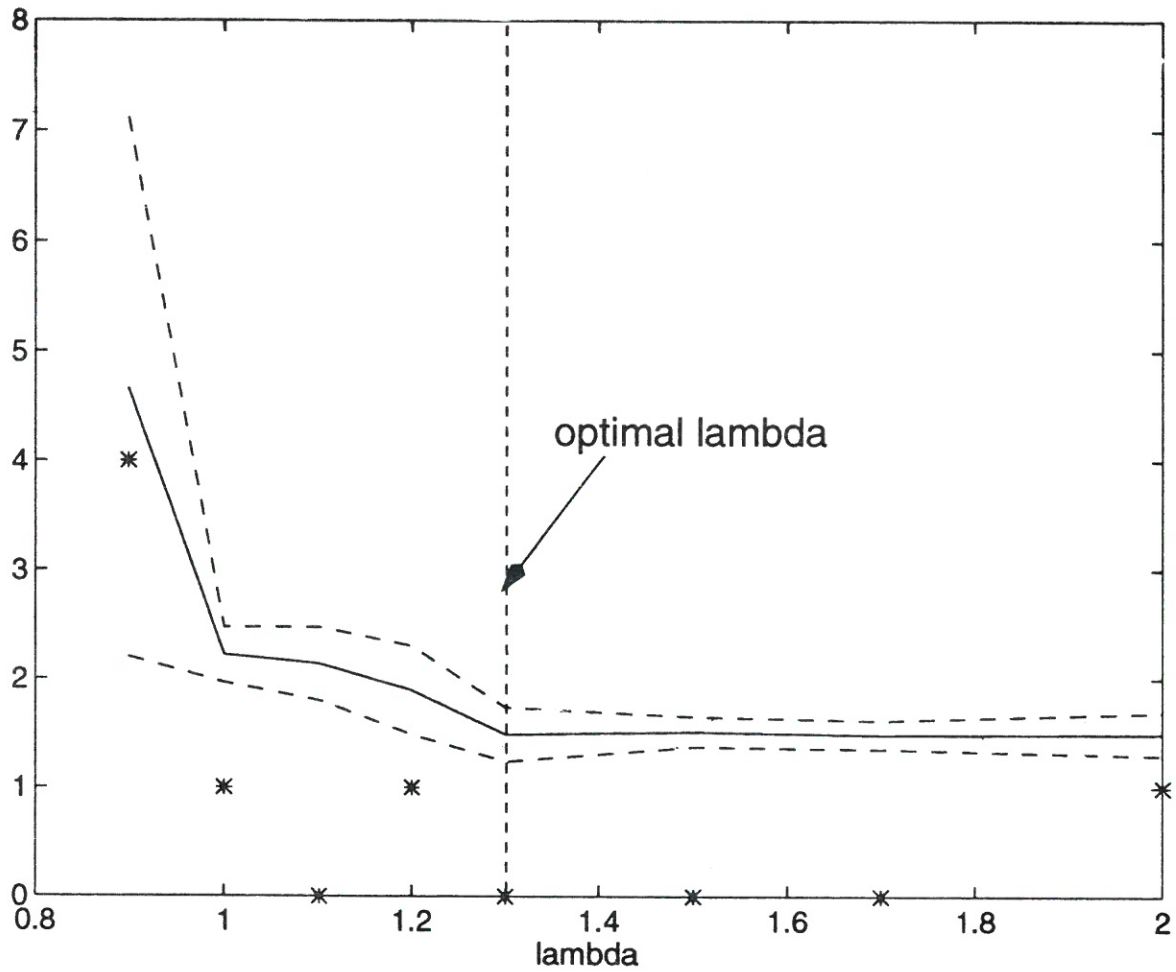

Figure 11. Test for the optimisation of the velocity. -: mean grasping time [s], - - : mean grasping time ± standard deviation [s], *: number of times missed

T_{grasp} was defined as the first time at which the distance and orientation between the end-effector and the object are less than a specified constant in four consecutive images. For each different value of λ, the robot performed 10 reaching movements starting in locations randomly distributed over a 20x20 cm^2 area. A stochastic optimisation with α=1 was used to determine λ.

To investigate the efficiency of the learning, λ has systematically been varied and the

We have experienced that with the optimal λ, reaching motions are smoother, faster, and the vibrations are reduced. Figure 12 shows the difference between trajectories performed with different λ. For λ too low, too much time is needed to reach the target or the target is not reached at all. With λ too high, the motions are jerky. Motions performed with an optimal value for λ are smooth and fast.

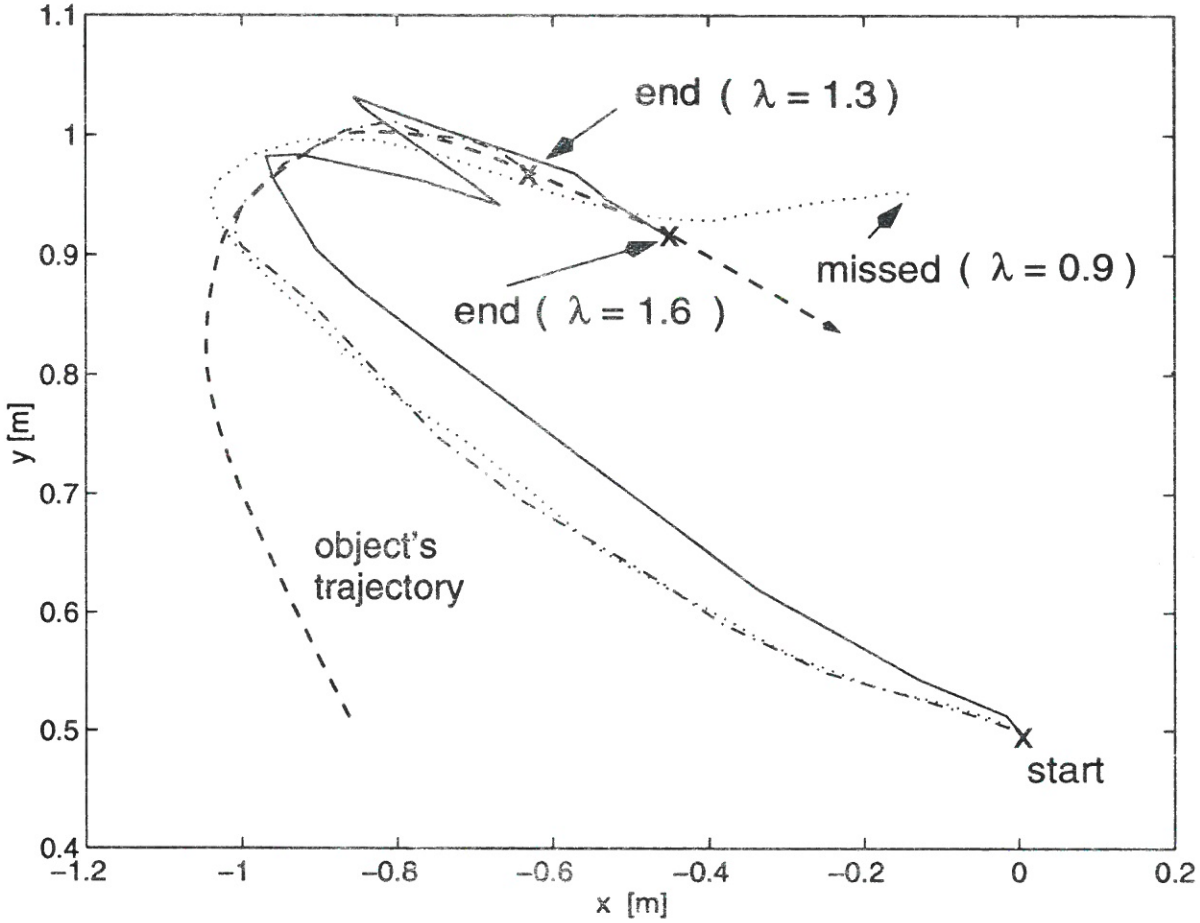

Figure 12. Trajectories with λ too high (λ=1.6), too low (λ=0.9) and optimally set (λ=1.3). The startpoint is (0,0.5) and the trajectory ends when the object is "grasped".

4. Learning... and Loading A Moving Waggon

Let us briefly review the learning process described at Section 3. First, an accurate calibration of the camera relative to the robot end -effector was performed in about a quarter of an hour. Then, a fast and precise vision processing was learned in less than 1 minute. Thirdly, the temporal synchronisation of the motion and the vision processes were improved by performing specified movements for about half-an- hour. Finally, the velocity was adapted to the desired accuracy and to the vision processing in about 1/4 hour. Thus the learning was performed in about 1 hour. We note that the generalisation to movements in three-dimensional space would only increase the time needed for the static calibration of cameras relative to the robot's end -effector.

To test the whole system after learning, the following task has been defined. An electro magnet is fixed at the end -effector of the robot, which is moving in a plane 12 cm over the table. The robot has to drop small metallic objects into an 3x5 cm^2 moving waggon. The robot releases the object when the robot and waggon positions are matched in four consecutive images and the normal and tangential object acceleration is small. This guarantees a correct prediction of the target's position. The parameters learned with the procedures of preceding Sections are used. In particular, the optimal velocity scaling factor is determined while starting the robot movements at random locations in the workspace.

Similar tasks were performed in (Allen, 1992) and (Buttazo, 1994). In (Allen, 1992) a robot was able to track and grasp a train moving with a velocity of up to 0.1 m/s in a circle of radius

0.25 m (which gives a normal acceleration of 0.04 m/s^2). In (Buttazo, 1994), a mouse-catching robot was developed, which moved in a straight line catching objects moving perpendicularly to this line, with velocity up to 0.5 m/s and acceleration up to 1.5 m/s^2. The sampling rate of the images was 20 ms and, in our experiment, it was 80 ms, due to the limitations of the frame grabber used for vision processing.

After learning, our robot is able to load the waggon when it is moving up to a velocity of 0.6 m/s and has a normal acceleration up to 1.4 m/s^2 every time. The robot follows the waggon until the releasing criteria are fulfilled, and then drops the object into it. At higher velocities, the robot follows the waggon, but, as the acceleration is too high, it does not release the object. Thus, using cheaper hardware than (Allen, 1992) or (Buttazo, 1994), we obtained similar or better results with respect to the kinematics of the catched object.

5. Analysis and Generalisation of the AMB

A relatively complex task, the tracking of manoeuvering objects, has been learned with a robot. In this Section we will investigate which principles can be deduced from the preceding developments and be applied to the learning of other complex tasks with a robot.

5.1 The Underlying Structure of the AMB

The task the AMB had to accomplish was to adapt the vision processing and the visuo-motor co-ordination in order to perform fast, smooth and efficient reaching movements. In order to reduce the complexity, this task was divided into the following sub-tasks: the spatial co-ordination of the manipulator with the vision, the visual tracking, the image processing, the temporal synchronisation of the motor with the visual processes, the motion planning, and the movement execution. These six sub-tasks and the corresponding features are listed in the two Tables of Figure 14. Each sub-task depends on some components

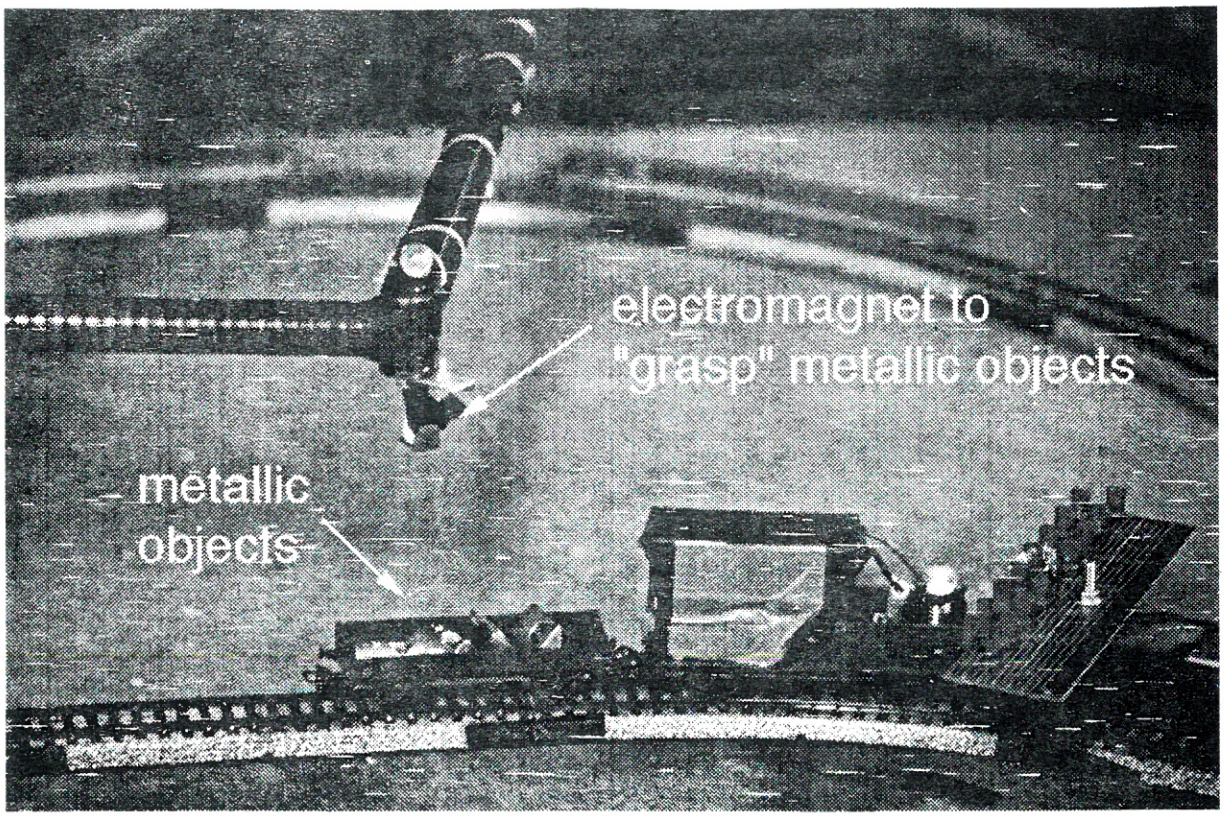

Figure 13. Robot Dropping a Small Object into a Moving Waggon

described by a few parameters. For example, the motion planning has to consider the mechanical structure of the manipulator and the motors limitations, and the movements are compactly parameterised by the velocity scaling factor λ. To each sub-task there corresponds some additional requirements, which are fulfilled when a cost function is minimised. The motion planning has, for example, to be such that the reaching movements are fast and smooth. This is realised by minimising the function $J = \alpha \overline{T}_{grasp} + \lambda$.

movement execution) and has then to be optimised after all these sub-tasks.

The optimisation of each sub-task was performed in a pragmatic way. The spatial co-ordination and the movement execution are highly reproducible. Therefore, the first could be stored in a Table, and the second was improved by means of analytical methods. The system's uncertainty was too high for other sub-tasks, so that they had to be improved by

	SPATIAL COORDINATION	VISUAL TRACKING	IMAGE PROCESSING
physical components	- vision system	- model of obj. mvt. - vision system	- vision system
parameters	table-look-up	$\beta, \sigma_m, \sigma_W$	σ
constraints	- pixel size -time for the transf. - encoders precision - manip. elasticity	- meas. error - obj. manoeuvr.	- time
additional requirements	precision	best expectation of the state	fast, precise movements
objective function	$\sum_{points}$, spatial error	$E[a\ t]$	$E[(x_k - \hat{x}_{k\|k})^2]$
rel. to biol	not known	"retinal" struct.	not known

	TEMPORAL SYNCHRONISATION	MOTION PLANNING	MOVEMENT EXECUTION
physical components	- vision system - VME + boards	- mechan. struct.	- manip. dyn.
parameters	reaction time	λ	ω
constraints	time: - vision system - comm. betw. boards - motion planner	- motors limit.	- motors limit.
additional requirements	best reaching	fast reaching, smooth mvts	precise movements
objective function	$\sum_{images}$ d (obj., man.)	$\alpha \overline{T}_{grasp} + \lambda$	$(Pe + D\dot{e})^2$, $e \equiv q_d - q$
rel. to biol	not known	similar to model	similar to model

Figure 14. Representation of the AMB. The task of performing fast, smooth and efficient reaching movements of a moving object has been divided into six simpler sub-tasks. More explanations are given in the text of Section 5.1

The learning of the task corresponds to learning all of the sub-tasks, i.e. to minimise the cost function from each of the sub-tasks. It was therefore necessary to study the relationships between the sub-tasks in order to know which order they might be learned (Figure 15). The Tables of Figure 14 were very useful for this examination. For example, the motion planning is dependent on all other tasks (except for the

either using a statistical analysis, or stochastic optimisation. Accordingly, mean values over several movements were used as cost functions.

5.2 A Representation for Learning Systems

The lack of flexibility of present robots is one of the most important reasons for their restricted

use in industry. To extend the application area of robots, it is necessary to grant them *the ability of executing many different tasks and of adapting to new situations*. In some dictionaries, this property is given as the definition of intelligence (*Le Petit Larousse Illustré, article intelligence*). Intelligence is the property used to differentiate human from (other) animals or animals from machines (Richard, 1991). In fact, until recently, it has been thought that the ability of adapting oneself efficiently to new tasks is only carried out by humans. Researchers from the artificial intelligence and cybernetics communities suggested however that, by the combination of simple rules, it is possible to obtain behaviours which are so complex that any human would judge them to be intelligent (Penrose, 1989; Braitenberg, 1986). It is beyond the scope of this paper to evaluate whether it is possible or even desirable to realise such "intelligent" robots. However, we believe that it is possible to construct robots which possess a certain autonomy and adaptability.

Zimmermann, 1991). However, efficient learning algorithms do not guarantee efficient learning. Relative to the efforts invested in the realisation of learning systems, the results obtained are limited. We believe that the reasons for this come not only from the difficulty of the task, but also depend on the approach made. It has often been thought that it is possible, considering the system as a black box, to optimise the whole system in one step. This concept does not operate, because of the high complexity of most systems . For example, we have shown that it is practically impossible to identify the dynamics of a robot operating in its entire workspace with a look-up-table or an artificial neural network (Burdet and Luthiger, 1995). By contrast, we proved in an implementation that using a suitable model, the manipulator's dynamics can easily be identified (Burdet, 1996). A good model of the arm dynamics is a prerequisite for learning to perform movement optimally (C.H. An and Hollerbach, 1988; Burdet and Luthiger, 1995).

Bernstein suggested that the human motor

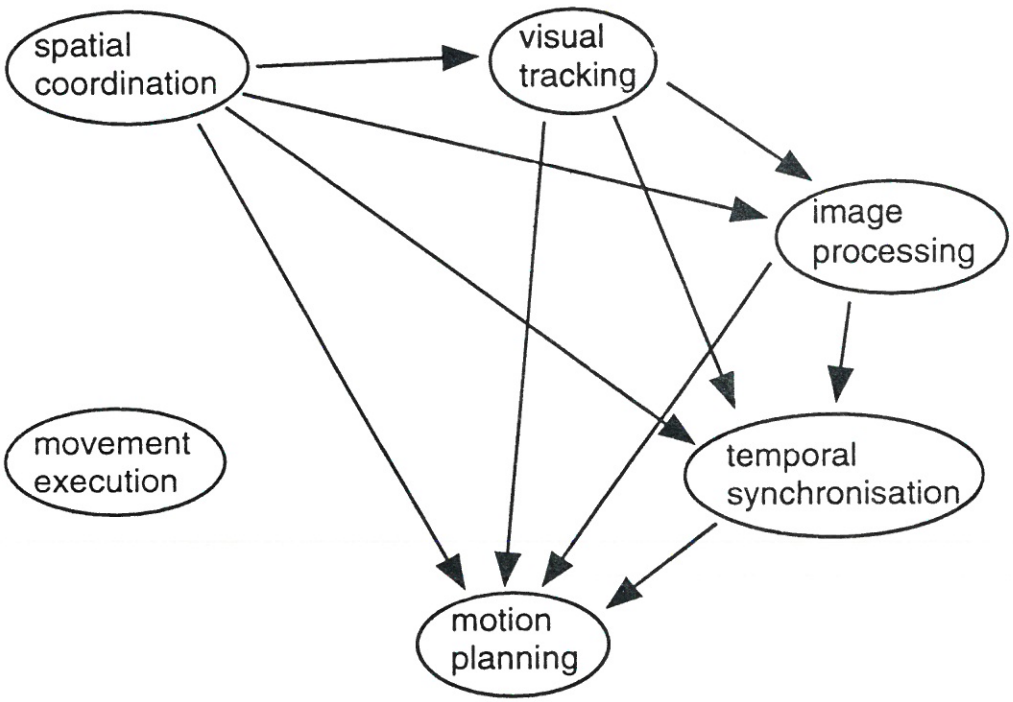

Figure 15. The sub-tasks of the AMB are partially ordered according to their relationships with each other. A→B means "B depends on A". The subtask "movement execution" is independent of the other sub-tasks, because the position of the manipulator is measured only by the encoders and not by using vision

In the last fifteen years, the adaptation properties of robots have been an important topic of research (Kaiser, 1995), and efficient learning algorithms have been developed (A.E. Bryson, 1975; Lan, 1990; D. Nauck, 1994;

system reduces the complexity of motor processes by using some *motor invariants* (Bernstein, 1967). We believe that, similarly, the emergence of robots able to attain a certain autonomy and adaptability, requires the definition of invariants of the robots behaviour.

The main problem to design intelligent systems in the above sense may consist in finding a suitable representation. The system has to be designed in order to allow a variety of different tasks to be realised, but with a minimal set of parameters, so that the behaviour should easily be adapted.

In the preceding Sections, we have described how a relatively complex task, the efficient tracking of manoeuvering objects, can be learned by a robot. This has been realised with techniques inspired by biological models, and through a pragmatic approach. In this Section, it has been suggested that an important factor for learning complex tasks with a robot is a suitable representation of the system. We have searched for those features in the representation of the AMB which can be generalised. In the following, we propose guidelines for the representation of robots based on the representation of the AMB, which should enable them to adapt themselves to many different tasks. In Section 5.2.2, it will be explained how biological models can be used in such a representation.

5.2.1 Guidelines for the Representation of Learning Systems

A robot system is composed of physical components, it is submitted to several external and internal constraints, has to perform some tasks and to meet additional requirements.

The following guidelines for the system's representation should favour its adaptability to many different tasks (Figure 16):

1. The system's components are represented using a minimal set of parameters allowing the largest number of possible configurations.
2. A minimal set of sub-tasks is defined, so that every task should be described by a combination of these sub-tasks.
3. Each sub-task has a cost function which corresponds to constraints and additional requirements: these are fulfilled when the function is minimised with respect to the parameters.
4. To allow actual optimisation, the cost functions have to depend on a minimum number of parameters, ideally only one.
5. To deal with the uncertainty inherent to real processes, the cost function must be the expected value of the stochastic functions corresponding to the processes, and the improvement must be performed by a statistical analysis or a stochastic optimisation.
6. The optimisation at the task-level corresponds to the optimisations of each of the sub-tasks. Therefore, it must be known which sub-tasks depend on each other , and that two sub-tasks cannot be mutually dependent. Mathematically, this means that the sub-tasks dependencies must form a partial ordering (Lang, 1993).

We note the following:

- In the case of independent sub-tasks, the cost function of the entire task is the sum of the functions of the sub-tasks. The sub-tasks can then be learned in an arbitrary order and even simultaneously. In this case, the cost functions define a mathematical *measure* on the task space (Lang, 1993). Such a case occurs when the task consists of a series of subtasks to be performed, as for example when an object has to be caught, and further to be placed in a precise location.
- The above guidelines can seem abstract or useless. However, they can help structure the design, and therefore can save time and avoid errors. For example, using the dependencies order relation, one can check in a simple way if two sub-tasks are mutually dependent. If they are, then these sub-tasks are in fact similar, and have to be learned in the same step.
- The proposed optimisation scheme leads to a suboptimal solution at the task level. The principal advantage of such an optimisation is feasibility. The above described stepwise optimisation is much more likely to be achieved than a global optimisation of the task in one step.

5.2.2 How Biological Models Can Enable the Emergence of Learning Robot Systems

There exist several reasons why biological strategies should not be blindly copied. Firstly, artificial systems have goals different from biological systems'. Secondly, the artificial "hardware" and biological "hardware" are quite different, so that in some cases the biological behaviour cannot be reproduced using an artificial system. Thirdly, there are sometimes simpler ways to realise the biological behaviour than using the biological model.

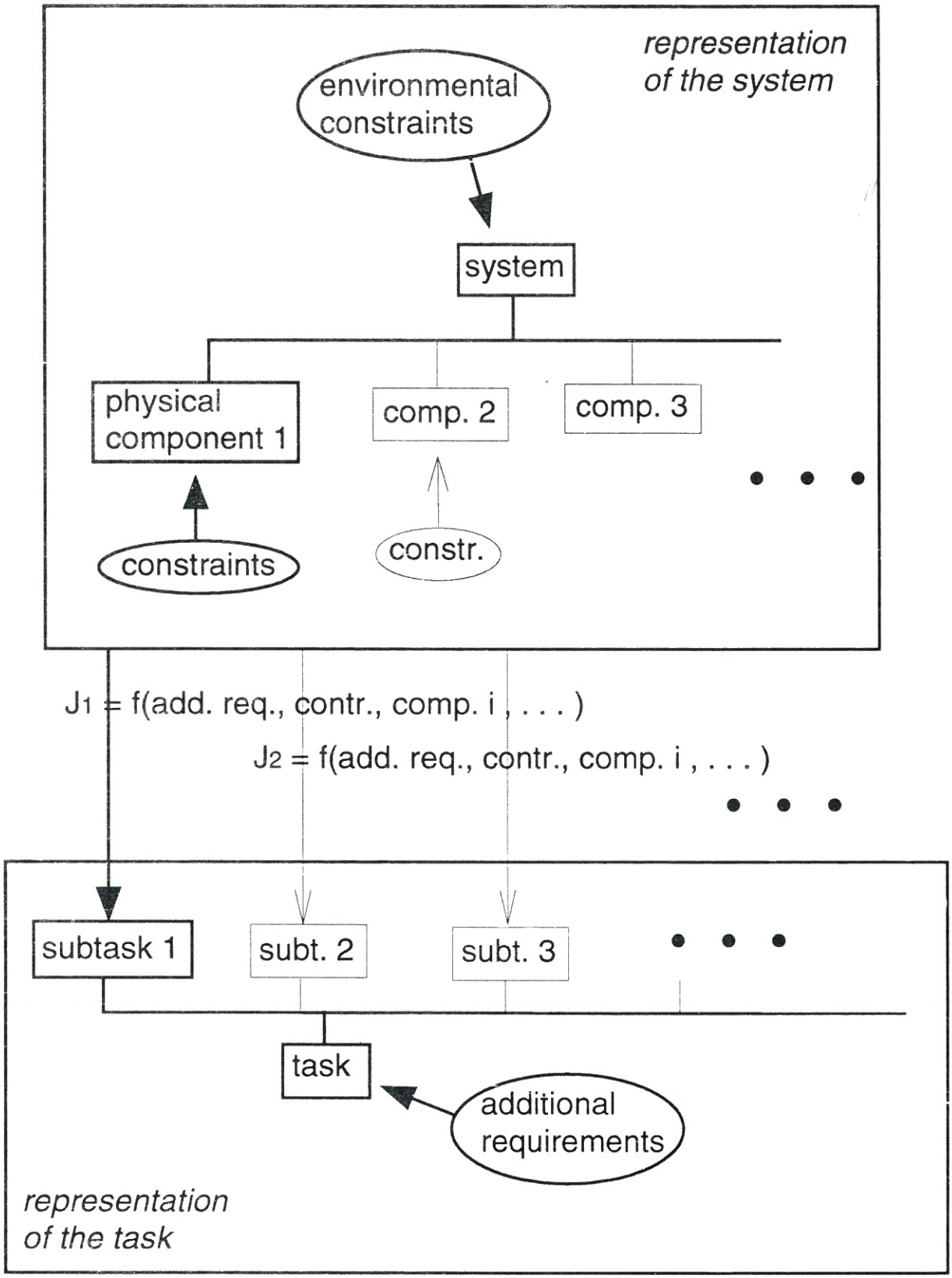

Figure 16. A Representation of a Robot System Enabling Adaptability to Many Different Tasks

However, biological examples are often the unique proof of existence available. One application of biological models in robotics is to guide the intuition. They can give ideas of useful properties to add to the behaviour of robotic systems. For example, in our case, the human improving its arm movements during practice (Schneider, 1989; Piaget, 1936; van Hofsten, 1990) gave us the idea of the AMB.

We have proposed how to subdivide complex tasks into simpler sub-tasks. In this scheme, biological models can be used directly for some sub-tasks. That is the second role of biological models in robotics. In the AMB, for example, the motion planning strategy and the optimisation of the velocity came directly from the human model. Another example is how the observation of insects gait has been used to control walking machines (Cruse, 1990;1994).

6. Conclusions

In this work, a method for learning efficient reaching movements has been proposed and demonstrated in an implementation. Using the AMB, the robot autonomously learned efficient visually guided movements. The usual (static) calibration of the camera relative to the robot workspace has been extended to the calibration of parameters involved in dynamic processes. The procedures composing the AMB are efficient and fast, and can be used for automatic calibration of robots controlled by a vision system.

The importance of the system's representation for the learning of complex tasks has been pointed out. It has been noticed how the principles according to which the AMB has been designed, can be generalised to a methodology for the representation of complex artificial systems, and how biological models can be used in this scheme. The performances of the AMB have shown that this methodology can lead to robots able to adapt themselves to changing environmental conditions. This should extend the flexibility of present robots and therefore decrease their cost.

From a control theoretical standpoint, the control of our robot and of the human arm are similar. Recent work has produced evidence that the motion is planned in extrinsic Cartesian co-ordinates (Shadmehr and Mussa-Ivaldi, 1993), that the visual control is exerted at discrete times (Milner, 1992) and the spinal control and muscle elasticity roughly correspond to feedback control (McIntyre, 1990). The control scheme of Figure 2 is therefore a coarse model for the control of the human arm, and our implementation has proved that this model works. Our artificial robot system, learning visually guided movements, can be compared with a young child learning reaching motions (van Hofsten, 1990).

Acknowledgment

Many thanks to P. Bühler, L. Buggiantella, E. Kaderli, J. Luthiger, M. Meyer, J. Müller and A. Pfenninger for their contributions to and their interest in this work.

Appendix

Conditions for Fast and Precise Vision Processing

During the time Δt, needed for the analysis of one image, only a limited number of pixels M can be examined. We search for a strategy, thereby the object position is determined with the maximum possible precision and in a minimum time. Accordingly, to this aim, $a \bullet t$, a being the image resolution and t the time needed for vision processing, is used as cost function for the visual analysis. This function naturally has to be minimised not for one image, but for the mean of several images:

$$J \equiv E[a \cdot t] \equiv \int a(h)\, t\, dP(h)\,. \qquad (8)$$

h is the two-dimensional position vector in the image, and $P(h)$ the distribution of the object positions around the expected positions. For convenience, we take

$$\int dP(h) \equiv 1\,, \qquad (9)$$

and further hypothesize that the time needed for examining M pixels is proportional to M. To simplify the calculation, we will also consider a continuous pixel density.

Accuracy

To simplify the calculations, a relationship between the accuracy and the pixel density must first be established. Let pixels be arranged in an

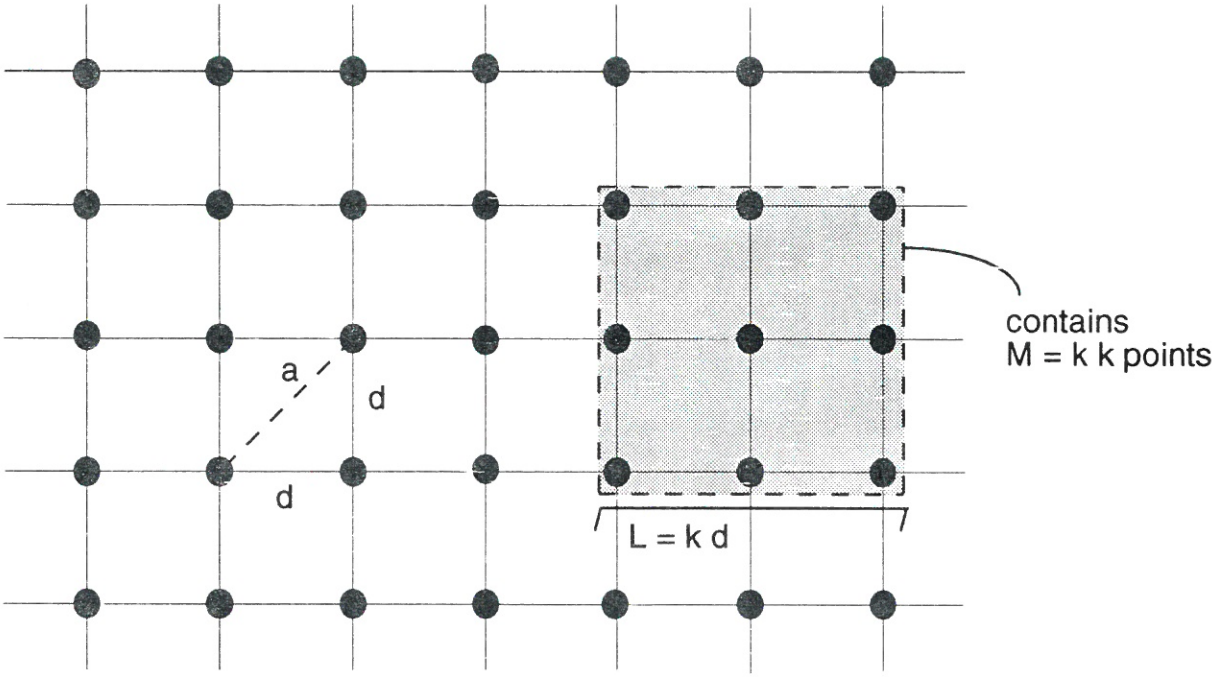

Figure 17. The Geometrical Relationship Between the Resolution a and the Grid Step d

infinitesimal regular quadratic grid with step d. The resolution of this grid is (see Figure 17).

$$a = \sqrt{2}d \,. \tag{10}$$

The area of a large square with side $L = kd$ is

$$V = L^2 = (kd)^2 \,. \tag{11}$$

The mass, i.e. the number of pixels in the square, is given by

$$M = k^2 \,, \tag{12}$$

and the pixels density ρ is

$$\rho = \frac{M}{V} \,. \tag{13}$$

From Equations (10) through (13) we get

$$a(h) = \frac{\sqrt{2}}{\sqrt{\rho(h)}} \,. \tag{14}$$

Optimal Windowing

Restricting the visual analysis to a window smaller than the entire image is a common technique for reducing the time for image processing (Buttazo, 1994; Wen, 1995). Generally the window size is determined by the actual velocity and acceleration of the moving object (Buttazo, 1994). The advantage of this strategy is that when the velocity is small, the window will also be small, and correspondingly, the precision will be high. Nevertheless, errors in the velocity and acceleration predictions leads sometimes to a failure in reaching the target, so that in the next image the window has to be large enough to find the object once again, and the processing time will be $2\Delta t$ and not Δt.

Let us now find which radius the (circular) window must have to minimise the cost function (8), which takes into account both the case where the object is inside and the case when it is outside of this window.

$$J(r) \doteq P[d(oc,wc) \le r]\frac{\Delta t}{\sqrt{\rho(r)}} + P[d(oc,wc) > r]\frac{2\Delta t}{\sqrt{\rho(2R)}} \quad (15)$$

where $P[d(oc,wc) \le r]$ denotes the probability that the distance $d(oc, wc)$ between the mass center of the object oc and the center of the window wc is less than r. $\rho(r)$ is the pixel density, if the pixel mass $M(\Delta t)$ analysed in the time Δt is uniformly distributed onto the window. Finally, $\doteq$ means "is proportional to".

Further let $R(\Delta t)$ be the radius of the smallest circle, in which the moving object can surely be found. For example, when the velocity of the object is bounded by v_{max}, then $R = v_{max}\Delta t$. Equation (15) then becomes

$$J(r) = \lambda\sqrt{\frac{V(r)}{M}\Delta t} + (1-\lambda)\sqrt{\frac{V(2R)}{M}2\Delta t} = \Delta t\frac{k_1}{\sqrt{M}}(4R + \lambda(r-4R)) \quad (16)$$

with C constant and $\lambda(r) \equiv P[d(oc,wc) \le r]$.

When the object positions are Gauss-distributed around the expected positions, then

$$\lambda = 1 - e^{-\frac{1}{2}\left(\frac{r}{\sigma}\right)^2} \quad (17)$$

and it follows that

$$J(r) = \Delta t\frac{C}{\sqrt{M}}\left(4R + 1 - e^{-\frac{1}{2}\left(\frac{r}{\sigma}\right)^2}\right)(r-4R). \quad (18)$$

It can easily be proved that $dJ(r)/dr$ is negative in $[0,R]$, therefore $J(r)$ is minimal for $r = R$.

We conclude that the window must be chosen to be large enough so that the target should never be missed. A windowing-method depending on the velocity as in (Buttazo, 1994), is not optimal in the sense of Equation (8).

7.3 Nonhomogeneous Pixel Density

We will show here that the analysis with a uniform pixel density can be further improved in the sense of Equation (8). In the middle of the human retina there is an area, called the fovea, with higher cell density and higher sensitivity. Would such a nonhomogeneous pixel density also improve the properties of the artificial retina?

Let the window be dimensioned so that the object is found with probability 1. In this case, it follows, using Equation (14), that:

$$J = \Delta t\int dP(h)a(h) \doteq \Delta t\int \frac{dP(h)}{\sqrt{\rho(r)}}, \quad (19)$$

with Δt the time needed for the analysis of one image. When the density depends only on the radius r, then $dP = r d\varphi\, dr\, F(r)$, where $F(r)$ is the radial density function. It then follows that:

$$J = 2\pi\int_0^R \frac{F(r)}{\sqrt{\rho(r)}} r dr. \quad (20)$$

Uniform Pixel Distribution

If the M pixels are uniformly distributed within a circle of radius R, then

$$\rho = \frac{M}{\pi R^2}, \quad (21)$$

and it follows that:

$$J = R\sqrt{\frac{\pi}{M}} \cong 1.77\frac{R}{\sqrt{M}} \quad (22)$$

Pixel Distribution Proportional to the Distribution of the Object's Positions

If the pixels are distributed proportionally to the density of the object position around the expected position, then $\rho(r) = MF(r)$, and it follows that:

$$J = \frac{2\pi}{\sqrt{M}}\int_0^R \sqrt{F(r)}\, r dr. \quad (23)$$

For Gauss-distributed objects' positions, this gives:

$$J = \frac{2\pi}{\sqrt{M}}\int_0^R \frac{1}{2\pi\sigma} e^{-\frac{1}{2}\left(\frac{r}{\sigma}\right)^2} r dr = \frac{2\sigma}{\sqrt{M}}\left(1 - e^{-\frac{1}{4}\left(\frac{R}{\sigma}\right)^2}\right) \quad (24)$$

If $R = 2\sigma$ is used, it finally follows that:

$$J = \frac{R}{\sqrt{M}}(1 - e^{-1}) \cong 0.63\frac{R}{\sqrt{M}}. \qquad (25)$$

The comparison of Equations (22) and (25) clearly shows that it is better to use a pixel density proportional to the object distribution than a uniform distribution.

Summary

The results of the above analysis indicate that for fast and precise visual analysis, only some pixels in the vicinity of the expected position should be analysed. The density of the examined pixels should further be proportional to the density of the objects positions around the expected positions.

REFERENCES

ABOAF, E. W., **Task-Level Robot Learning,** Technical Report, Artificial Intelligence Laboratory, MIT, MA, 1988.

A. E. BRYSON and HO, Y.C., **Applied Optimal Control**, Halsted., 1975.

ALLEN, P. K., **Real-time Visual Servoing**, IEEE International Conference on Robotics and Automation, 1991.

ALLEN, P. K., **Trajectory Filtering and Prediction for Automated Tracking and Grasping of A Moving Object**, IEEE International Conference on Robotics and Automation, 1992.

ALLEN, P. K., **Modeling Dynamic Uncertainty in Robot Motions**, IEEE International Conference on Robotics and Automation, 1993.

ATKESON, C. G. and HOLLERBACH, J.M., **Kinematic Features of Unrestrained Vertical Arm Movements**, JOURNAL OF NEUROSCIENCE, 1985.

BERNSTEIN, N., **The Coordination and Regulation of Movements**, PERGAMON PRESS, 1967.

BRAITENBERG, V., **Vehikel Experimente mit Kybernetischen Wesen**, VIEWEG-VERLAG, 1986.

BROBOW J. E., **Optimal Robot Path Planning Using the Minimum-time Criterion**, IEEE JOURNAL OF ROBOTICS AND AUTOMATION, 1988.

BURDET, E., **Algorithms of Human Motor Control and their Implementation in Robotics**, PH.D Thesis, ETH-Zürich, 1996.

BURDET, E. and LUTHIGER, J., **Adaptable Vision/Motion Coordination for A Robot**, : Perac94: From Perception To Action, 1994.

BURDET, E. and LUTHIGER, J., **Three Learning Architectures To Improve Robot Control: A Comparison,** European Workshop on Learning Robots, 8th European Conference on Machine Learning, 1995.

BURDET, E. and MÜLLER, **J., A Robot Learning Reaching Motions** (video), IEEE International Conference on Robotics and Automation, 1996.

BUTTAZO, G. C., **Mousebuster: A Robot for Real-time Catching,** IEEE ROBOTICS AND AUTOMATION MAGAZINE, 1994.

C.H. AN, ATKESON, C. G. and HOLLERBACH, J.M., **Model-based Control of A Robot Manipulator**, MIT-PRESS, 1988.

CRUSE, H., **What Mechanisms Coordinate Leg Movement in Walking Arthropods?,** TRENDS IN NEUROSCIENCES, 1990.

CRUSE, H., **A Neural Net Controller for A Six-Legged Walking System**, Perac94: From Perception To Action, 1994.

D. NAUCK, KLAWONN, F. AND KRUSE, R., **Neuronale Netze und Fuzzy Systeme**, VIEWEG-VERLAG, 1994

HOFF, B. R., **A Computational Description of the Organization of Human Reaching and Prehension**, PH.D Thesis, University of Southern California, 1992.

M.Kaiser (Ed.) **Proceedings of the Fourth European Workshop on Learning Robots**, University of Karlsruhe, 1995.

LAN, C., **Neural Networks, Theoretical Foundations and Analysis**, 1990.

LANG, J., **Algebra**, ADDISON-WESLEY, 1993.

Le Petit Larousse Illustré, Article intelligence (n.d.).

LUTHIGER, J., **Sensororientierte und Sensorgeführte Bahnplanung**, PH.D Thesis, ETH-Zürich, 1996.

MCINTYRE, J., **Utilizing Elastic Properties for the Control of Posture and Movement**, PH.D Thesis, MIT, MA, 1990.

MILNER, T.E., **A Model for the Generation of Movements Requiring Endpoint Precision**, NEUROSCIENCE, 1992.

MÜLLER, J., **Optimierung der visuo-motorischen Koordination**, Technical Report, Institut für Robotik, ETH-Zürich, 1995.

PENROSE, R., **The Emperor's New Mind**, OXFORD UNIVERSITY PRESS, 1989.

PIAGET, J., **La naissance de l'intelligence chez l'enfant**, Delachaux et Nestlé, 1936.

RICHARD, J. F., **Intelligence**, Encyclopedia Universalis, 1991.

SCHNEIDER, K., **Changes in Limb Dynamics During the Practice of Rapid Arm Movements**, JOURNAL OF BIOMECHANICS, 1989.

SHADMEHR, R. and MUSSA-IVALDI, F.A., **Geometric Structure of the Adaptive Controller of the Human Arm**, Technical Report, Artificial Intelligence Laboratory, MIT, MA, 1993.

SINGER, R.A., **Estimating Optimal Tracking Filter Performance for Manned Manoeuvering Targets**, IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS, 1970.

VAN HOFSTEN, C., **Early Development of Grasping An Object in Space-time**, in M. A. Goodale (Ed.) VISION AND ACTION. THE CONTROL OF GRASPING, ABLEX, 1990.

WEN, J., **Hybrid Approach of Neural Networks with Knowledge Based Explicit Models**, PH.D THESIS, ETH-Zürich, 1995.

ZIMMERMANN, H.-J., **Fuzzy Set Theory and Its Applications**, KLUWER ACADEMIC PRESS, 1991.