Building Human-Friendly Robot Systems

J. Heinzmann and A. Zelinsky

Research School of Information Sciences and Engineering, The Australian National University,
Canberra, ACT 0200, Australia, http://syseng.anu.edu.au/rsl/

Abstract

To develop human friendly robots we required two key components; smart interfaces and safe mechanisms. Smart interfaces facilitate natural and easy interfaces for human-robot interaction. Facial gestures can be a natural way to control a robot. In this paper, we report on a vision-based interface that in real-time tracks a user's facial features and gaze point. Human friendly robots must also have high integrity safety systems that ensure that people are never harmed. To guarantee human safety we require manipulator mechanisms in which all actuators are force controlled in a manner that prevents dangerous impacts with people and the environment. In this paper we present a control scheme for a whole arm manipulator (WAM) which allows for safe human-robot interaction.

1. Introduction

If robotics technology is to be introduced into the everyday human world, the technology must not only operate efficiently executing complex tasks such house cleaning or putting out the garbage, the technology must be safe and easy for people to use. Executing complex tasks in unstructured and dynamic worlds is an immensely challenging problem. Researchers have realized that to create robotic technology for the human world, the most efficient form is the humanoid robot form, as people have surrounded themselves with artifacts and structures that suit dual-armed bipeds e.g. a staircase. However, merely giving a robot the human shape does not make it friendly to humans. What constitutes a human friendly robot? Firstly, human friendly robots must be easy to use and have natural communication interfaces. People naturally express themselves through language, facial gestures and expressions. Speech recognition in controlled situations using a limited vocabulary with minimal background noise is now possible and could be included in human-robot interfaces. However, vision-based human-computer interfaces are only recently began to attract attention [2][3][8]. With a visual interface a robot could recognise facial gestures such as "yes" or "no", as well being able to determine the user's gaze point i.e. where the person is looking. The ability to estimate a person's gaze point is most important for a human-friendly robot. For example a robot assisting the disabled could pick up items that attract the user's gaze. Our goal is to build "smart" interface for use in human-robot applications. In this paper we describe our work in real-time vision-based interfaces.

The second requirement for a human-friendly robot is that it must possess a high integrity safety system that ensure the robot's actions never result in physical harm to a person. One way to solve the human-safety problem is to build robots that could never harm people, by making mechanisms small, light-weight and slow moving. However, this will result in systems that can't work in the human world i.e. won't be able to lift or carry objects of any significance. The only alternative is to build systems that are strong and fast, but which are human-friendly. Current commercial robot manipulator technology is human-unfriendly. This is because these robots only use point to point control. Point to point control is quite dangerous to use in dynamic environments where situations and conditions can unexpectedly change. If an object unexpectedly blocks a planned point to point path, either the object will be smashed out of the robot's path or the robot will sustain considerable damage. Clearly a compliant force based controller is needed in such situations. However, the technology for force-based control of robots is not yet readily available. The current practice is to add force sensors to existing robot manipulators, usually at the tool plate, which only allows force control of the robot's end point. This means the robot could still easily collide with unexpected objects with other parts of the manipulator. To ensure safety a Whole Arm Manipulation (WAM) approach must be adopted, in which all the robot's joints are force controlled.

In this paper we present a vision-based interface, and a human-friendly motion controller for the MIT-Barrett WAM robot that we have implemented.

2. A Vision-based Interface

Our approach uses a 3 layered system shown in Figure 1. At the lowest level the system performs template correlation in hardware. The measured positions and correlation values are forwarded to the 2D model which takes geometric constraints into account in the image plane and the correlation distortion to generate an estimate of the feature positions. This layer is implemented as a network of Kalman filters [7]. The estimated positions of the features determine the location within the next image frame of the hardware search windows. The 2D image positions of the features are transfered to a 3D model of the feature locations. Using multiple feature triplets the 3D pose of the head is determined and used for other calculations such as gesture recognition or gaze point detection. The 3D model is projected back into the image plane to adapt the constraints in the 2D model. All three layers run at 30Hz.

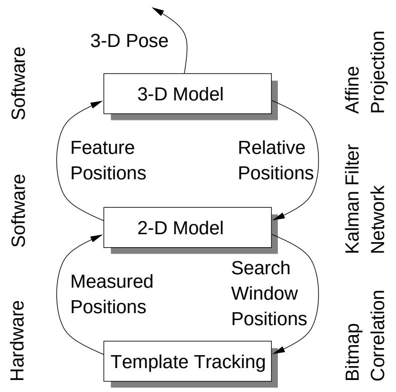

Figure 1: Structure of the Face Tracking system

2.1. Layer 1: Image Correlation

The lowest layer of the system is a Fujitsu color vision template correlator, that uses 8×8 pixel correlation kernels. The correlation function is the Sum of Absolute Differences (SAD). This simple correlation function allows high throughput (~ 200 color templates at 30Hz), however, it also limits the robustness of the system with respect to changes in the illumination. The hardware consists of a single PCI-bus card, running on a Pentium PC.

2.2. Layer 2: Kalman Filter Network

The middle layer's task is to provide robust feature tracking results for the 3D model matching. It is implemented as a network of Kalman filters. The basic idea is that individual search windows track dedicated features of the face and help each other in critical situations. The goal is to find a stable configuration with correct geometric relationships between the search windows and low distortions. To achieve this the tracked features are interconnected by distance vectors defining their relative position. They form a 2D network with features as nodes and vectors as edges. Figure 2 shows a face with the connections for the Kalman filters.

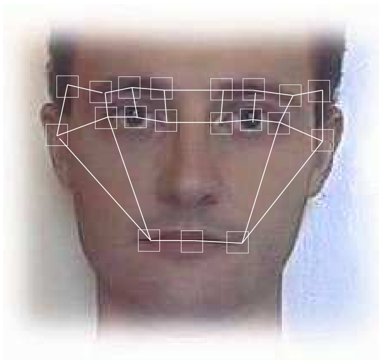

Figure 2: Facial features and the connections between the Kalman filters.

The challenge is to integrate the tracking results of neighbouring features in an effective way. If the tracking results are not consistent due to tracking errors for a subset of the tracked features the results have to be weighted by their reliability. This can be achieved using Kalman filters (e.g. [4]) to estimate the position of the features exploiting the two sources of information, the tracking results and the the geometric constraints of the feature network. A more detailed description of the implemented algorithms can be found in [7].

2.3. Layer 3: 3D Model Matching

The estimation of the pose of the head is a basic functionality of visual face trackers and has been investigated by many researchers [2, 6, 12, 13]. To recover the 3D position of an object from projected features in a monocular image two projection models may be used: perspective and affine projection. The perspective transformation precisely models the actual projection of a 3D scene to the image plane (assuming a pinhole camera). However, the required calculations are complex and time consuming and can deliver up to a fourfold ambiguity in the estimate of the pose. Real-time systems usually use affine transformation because it has simpler calculations and only a twofold ambiguity [6, 12]. Also it is a good approximation of perspective projection provided the depth of the object does not exceed $\frac{1}{10}$ of the distance between camera and object.

The 3D model of the face is matched using an improved variant of the 3 point pose recovery algorithm. For a description of the basic algorithm refer to [1] [10]. The pose estimation algorithm assumes an affine projection of the model points into the image plane. In reality, the features are projected into the image plane under perspective projection causing a systematic error in the pose estimate. We analysed this error and made an improvement to the feature triplet pose recovery algorithm that is able to reduce the systematic angular and translational errors of pose estimation by 75% and 90% respectively.

2.4. Gaze Estimation

The gaze point provides valuable information about what a person is interested in or what a person is referring to. The calculation is performed in 3 stages. In the first step the 3D position of both eye balls is determined using the 3D model of the head and the estimated pose. In the second step a ray from the focal point of the camera through the image plane is calculated. The ray intersects the image plane at the center of the iris of the respective eye. Next the intersection of the ray with the eye ball is calculated. The ray from the center of the eye ball through the intersection point gives the gaze direction of the eye. The gaze directions of both eyes are then merged into a single gaze vector. In the last step the gaze vector is intersected with a world model to derive the gaze point of the person.

Figure 3 shows the geometric constellation of the eye gaze calculations. See [8] for more details of gaze point estimation.

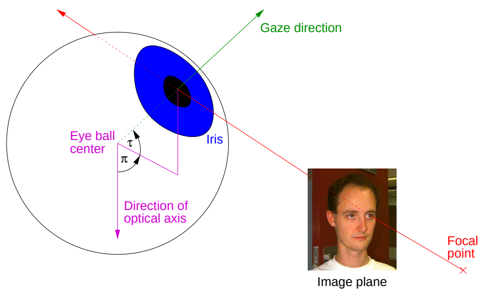

Figure 3: Geometry of the eye gaze calculation.

2.5. Face Tracking Experiments

The visual feature tracking and pose estimation was evaluated using our MARITA system (Mannequin Robot for Investigation of Tracking Accuracy). MARITA consists of a mannequin head mounted onto a cable driven high performance pan-tilt-device. The two degrees of freedom can rotate the head around it's vertical axis as well as tilt the head for- and backward. Rotations about the z-axis (facial normal) are not possible, however, the error in estimating this fixed angle can still be measured. Also the transformation from the X-Y Euler angles of the mechanism to the Z-Y-X Euler angles used for the internal pose representation causes small rotations about the z-axis. MARITA allows us to position the mannequin head to an accuracy better than 0.1° and test the accuracy of the pose estimate of the face tracker under realistic conditions.

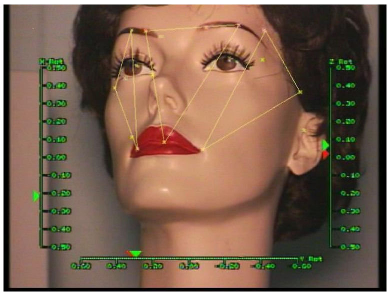

Figure 4: MARITA articulated mannequin head provides the ground truth for the head pose accuracy.

To track the pose of the mannequin head a set of 19 features were manually selected, including 6 features around each eye, 3 features in the mouth area and 2 features near each ear. The small white crosses in Figure 4 indicate the position of the features. Not all 19 features are used for pose estimation, 5 features are sufficient. The other features provide cues for the two lower layers in order to make tracking stable and robust to noise and partial occlusions. The lines indicate the feature triplets, one parallel to the facial plane made up from the mouth and the eyebrows and one triplet on each side including the corner of the mouth, eyebrows and ears. Only the features in the corners of the triangles are actually used for the pose estimation. With this selection of features and triangles the spatial orientation of the head can be measured by at least two triplets for a wide range of configurations.

To measure the accuracy parameters of our face tracker we connected it directly to the controller of MARITA. A supervisory program moved the head into different position and then a sample of the pose estimators performance is taken over a period of 500 frames.

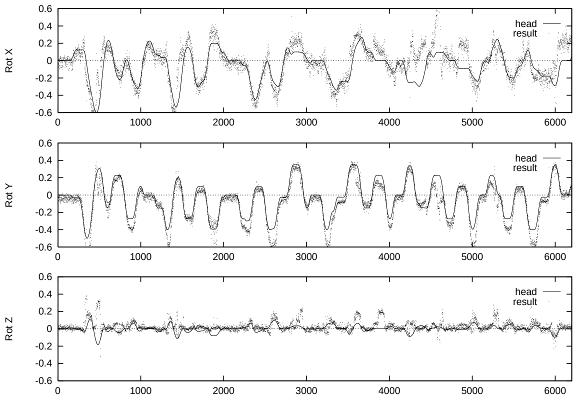

Figure 5: Angular traces for X, Y and Z axis over 6000 frames

As parameters for the performance the bias (mean error) between the correct pose and the measured pose as well as the standard deviation of this pose were measured. The points for evaluation were spaced 0.1rad apart, ranging from -0.5 to 0.5rad in y-rotation and -0.3 to 0.3rad in x-rotation. However, for some of the 77 positions the illumination changed sufficiently to cause failure of the template tracking, and thus errors in the pose estimation.

The estimates of the rotation around the x-axis contain the highest error of the three axis because it's estimate is highly dependent on correct tracking of the features around the ear which proved to be difficult to track due to the lack of unique texture. The mean absolute bias in the area where the features were tracked successfully (68%) is 0.10479rad (6.0°) for the x-axis.

Both the estimates for the y- and z-rotation contain fewer errors. The bias is generally close to 0, particularly on the sides of the plot where the head was rotated around the y-axis about 30°, the tracking failed and abrupt changes in the magnitude of the error occur. The mean absolute biases were 0.06699rad (3.8°) on 75% of the area for the y-rotation and 0.04075rad (2.3°)on 74% for the rotation around the z-axis.

The x- and y-axis translations can be measured fairly precise, the mean absolute bias is 6.97mm and 14.08mm in x- and y-direction. As expected from a monocular system, the error for the depth measurement is a magnitude higher. On average the z-translation contains an error of 60.04mm in the well tracking area, 10% of the overall distance of 600mm from the camera.

Figure 5 shows the trace plot for a different experiment where the head was moved to random positions over 6000 frames. The plots show both the actual position marked as a continuous line and the estimates in each frame as dots of the X, Y and Z axis rotations. The X rotation is tracked well, however, with a visually higher standard deviation than the Y rotation caused by unreliable localisation of the features in the ear area. The estimate of the y-rotation contains the least error and the standard deviation is comparably small. Only negative rotations seem to be estimated exaggerated. As mentioned before, the transformation from the mechanical system to the internal representation of the spatial orientation cause only small rotations around the z-axis.

3. Human-Safe Robots

If a robot is to be used in the direct vicinity of people, control methods such as point-to-point position control are not suited. A popular way to achieve compliant behaviour of a robot is to add a force-torque sensor to the wrist of the robot. However, this allows force control of the hand only while other parts of the manipulator could collide with a person without the controller noticing. Instead, a whole arm manipulation (WAM) approach must be adopted in which all joints of the robot are force controlled.

The hardware platform for this project is a Barrett WAM robot, the commercial version of the MIT arm built by Townsend and Salisbury [15]. The mechanical design of the robot is well suited for a human-friendly robot. The robot has 7DOF and all joints are driven by cable transmissions. Due to the low friction, the robot can easily mechanically backdriven by a person. Also, since the transmission has no backlash, the joint torques can be directly derived from the motor torques and the dynamic model of the robot.

The basic mode of the robot is to compensate for gravity (Zero-G behaviour). In this mode the robot is completely passive and appears to be weightless. This behaviour is achieved by using feed-forward control only. We consider this mode to be human safe, since the robot is completely passive, all of the robot's actions are directly initiated by the operator. The behaviour of the robot can be easily predicted even by non-experts. We believe the robot should be viewed as a tool rather than a responsible entity. See [9] for more detail on safety for human friendly robots and implementation details of Zero-G.

By specifying additional torques other than those torques required for gravity compensation the robot can achieve autonomous motion. This is required for all tasks the robot is to perform without the physical interaction of a person. The danger of autonomous motion in the workspace is the possible collision of the robot with people. With today's sensing technology it is not possible to detect the presence and position of humans in the robots workspace with 100Collisions of the robot with human operators can not be ruled out, and therefore, the threat posed by such a collision has to be restricted to a level the operator believes is safe and comfortable.

3.1. Impact Force

Impact force provides a good measure of how dangerous a potential collision with a person could be. The collision between the mechanical system and an obstacle can occur with any point p on the surface of the mechanical system. We are using the impact model defined in Walker et al. [16] which assumes an impulsive contact. The impact force $\hat{\mathbf{F}}$ of a particular point p on the surface of a serial manipulator with an stationary obstacle is given by

$$\hat{\mathbf{F}} = \frac{-(1+e)\mathbf{v}^t\mathbf{n}}{\mathbf{n}^t\mathbf{J}_p(\theta)\mathbf{I}^{-1}(\theta)\mathbf{J}_p^t(\theta)\mathbf{n}} \tag{1}$$

Here, $\mathbf{v} \in \mathbb{R}^3$ is the velocity of the mechanical system at the point p of impact, $\theta \in \mathbb{R}^n$ are the joint angles, $\mathbf{J}_p(\theta)$ is the manipulator Jacobian for the point p in the mechanical system where the contact occurs, $\mathbf{I}(\theta)$ is the n×n inertia matrix, and $\mathbf{n} \in \mathbb{R}^3$ is the unit contact normal vector. In the worst case $\mathbf{v}$ is aligned with $\mathbf{n}$ but pointing in the opposite direction $(\mathbf{v} = -s_v\mathbf{n})$, $s_v = |\mathbf{v}|$ and the impact is elastic $(e = 1)$. In this worst case scenario 1 simplifies to:

$$\begin{aligned}
\hat{\mathbf{F}}_{max} &= \frac{-2\mathbf{v}^t(-\frac{1}{s_v}\mathbf{v})}{(-\frac{1}{s_v}\mathbf{v}^t)\mathbf{J}_p(\theta)\mathbf{I}^{-1}(\theta)\mathbf{J}_p^t(\theta)(-\frac{1}{s_v}\mathbf{v})} && (2)\\
&= \frac{2s_v\mathbf{v}^t\mathbf{v}}{\mathbf{v}^t\mathbf{J}_p(\theta)\mathbf{I}^{-1}(\theta)\mathbf{J}_p^t(\theta)\mathbf{v}} && (3)\\
&= \frac{2s_v^3}{\mathbf{v}^t\mathbf{J}_p(\theta)\mathbf{I}^{-1}(\theta)\mathbf{J}_p^t(\theta)\mathbf{v}} && (4)
\end{aligned}$$

With $\mathbf{v} = \mathbf{J}(\theta)\dot{\theta}$ and $s_v = |\mathbf{v}| = |\mathbf{J}(\theta)\dot{\theta}|$, $\hat{\mathbf{F}}_{max}$ can be expressed as a function of the state $(\theta, \dot{\theta})$ of the mechanical system.

$$\hat{\mathbf{F}}_{max} = \frac{2|\mathbf{J}_p(\theta)\dot{\theta}|^3}{\dot{\theta}^t\mathbf{J}_p^t(\theta)\mathbf{J}_p(\theta)\mathbf{I}^{-1}(\theta)\mathbf{J}_p^t(\theta)\mathbf{J}_p(\theta)\dot{\theta}} \tag{5}$$

$\hat{\mathbf{F}}_{max}$ is undefined for $\mathbf{v} = \mathbf{J}(\theta)\dot{\theta} = 0$. However, for a stationary point the impact energy with a stationary obstacle equals 0 and $\hat{\mathbf{F}}_{max}$ is defined accordingly.

3.2. Maximum Impact Force

To describe the ability of a robot to cause impact we introduce the term *Maximum Impact Force (MIF)*. We define MIF to be the maximum impact force that a moving mechanical system can cause with a static environment. In this case we have to consider all points $p \in P$ on the surface of the robot. This makes $\hat{\mathbf{F}}_{max} : P \times \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}$ a function of the point p and the dynamic state $(\theta, \dot{\theta})$ of the system. We can now define the MIF Π of the system as

$$\Pi = \sup_{p \in P} \Pi_p, \quad \text{where}\Pi_p = \hat{\mathbf{F}}_{max}(p, \theta, \dot{\theta}) \tag{6}$$

The MIF is a well-defined scalar value that provides a upper limit for any impact of the system with a static object. Since it assumes elastic impact and aligned surface normals, the limit is rather conservative.

We can now define a limit Π_{max} for the impact force which can be considered safe in the event of a collision of the robot with a person. The maximum impact force we allow for a robot defines the *safe region* in the state space where the following inequality holds

$$\Pi_p \leq \Pi_{max} \quad \forall p \in P \tag{7}$$

3.3. Controlling the Impact Force

Many researchers have investigated the behaviour of robot controllers for impact situations [11, 14, 17]. The transition between free space motion and constraint force controlled motion after contact requires special consideration since the robot may fall into an oscillatory mode bouncing on the surface or become even unstable. Therefore, the main focus of this research was to ensure stable control during the transition. A secondary goal was to reduce the impact force qualitatively. None of the proposed schemes are able to bound the impact force quantitatively.

The goal of our novel *Impact Force Control* scheme is to guarantee that a collision of the robot with a static obstacle will not result in an impact force larger than a preset value. This is achieved by always keeping the robot within the safe region in state space. However, external forces may still force the robot outside the safe region, e.g. when pushed by the operator.

The *Impact Force Controller* acts as a saturating filter between the motion control algorithm and the robot. It provides a safety envelope which encapsulates the robot. The impact controller is passing the torque vector generated by the motion control algorithm unchanged to the robot as long as it complies with the safety constraints set in the impact controller. If the desired torque vector does not satisfy the safety constraints, a clipping function has to be applied to the torque vector. The resulting safe torque vector guarantees that the robot can not leave the safe region.

A suitable state-dependent control constraint is defined in the following inequality.

$$\frac{d}{dt}\Pi_p \leq \frac{1}{t_c}(\Pi_{max} - \Pi_p) \;\; \forall p \in P \tag{8}$$

The time constant t_c defines how fast the robot may gain MIF. It can be set in a way that the motion of the robot is predictable for the operator. The clipping function has to project the desired torque vector from the motion control algorithm into the safe region in the torque vector space. One solution is to find the safe torque vector closest to the desired torque vector. Note that it is not always sufficient to scale down the desired torque vector since the 0-torque vector is not always in the safe region. A moving robot in Zero-G can increase its MIF without any further torques applied to the system. In this case the Impact Controller will slow the robot down to enforce the safe region constraints. If the robot is pushed out of the safe region by external forces, the right side of Equation 8 becomes negative and the Impact-Controller enforces a reduction of MIF according to the time constant t_c.

3.4. Implementing Impact Force Control

To implement the proposed scheme which guarantees limited MIF during autonomous motions of the robot the constraints have to be transfered into motor torque space. Considering that $\Pi_p(t) = \Pi_p(\theta(t), \dot{\theta}(t))$, Equation 8 can be expanded to the following representation.

$$\frac{\partial \Pi_p}{\partial \dot{\theta}}\ddot{\theta} + \frac{\partial \Pi_p}{\partial \theta}\dot{\theta} \leq \frac{1}{t_c}(\Pi_{max} - \Pi_p(\theta, \dot{\theta})) \tag{9}$$

$$\frac{\partial \Pi_p}{\partial \dot{\theta}}\mathbf{M}^{-1}\tau \leq \frac{1}{t_c}(\Pi_{max} - \Pi_p) - \frac{\partial \Pi_p}{\partial \theta}\dot{\theta} + \frac{\partial \Pi_p}{\partial \dot{\theta}}\mathbf{M}^{-1}\mathbf{N} \tag{10}$$

Defining the torque constraint vector $\mathbf{c}_p \in \mathbb{R}^n$ and scalar limitation value l_P to

$$\mathbf{c}_p = \frac{\partial \Pi_p}{\partial \dot{\theta}}\mathbf{M}^{-1} \tag{11}$$

$$l_p = \frac{1}{t_c}(\Pi_{max} - \Pi_p) - \frac{\partial \Pi_p}{\partial \theta}\dot{\theta} + \frac{\partial \Pi_p}{\partial \dot{\theta}}\mathbf{M}^{-1}\mathbf{N} \tag{12}$$

yields the simplified form of the torque constraints.

$$\mathbf{c}_p\tau \leq l_p \;\; \forall p \in P \tag{13}$$

The Inequalities 13 define closed halfspaces in the torque space. The limiting hyperplanes are defined by $\mathbf{c}_p\tau = l_p$. The safe region in torque space is the intersection of the halfspaces. Note that the intersection of halfspaces is always convex.

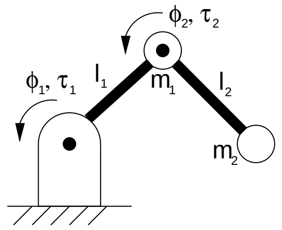

Figure 6: Two link robot with two point masses at the end of the links.

3.5. Clipping Torques

If one or more of the constraints in Equation 13 are not satisfied for the command torque τ_c the command from the controller is bringing the robot in a potentially dangerous state. Since the motor torques need to be updated to ensure safe operation, a safe torque needs to be derived from the desired command torque τ_c. The logical choice for such a safe torque vector τ_s is the closest torque vector to τ_c which is safe.

$$\tau_s = \tau \text{ such that } ||\tau - \tau_c|| \text{ is minimal} \tag{14}$$

This clipping of the torque vector to the closest safe torque vector is achieved by calculating the orthogonal projection of τ_c onto the hyperplanes defined by Equation 13 and the affine subspaces defined by their intersections and the intersections of the lower dimensional subspaces and so on. The safe torque vector τ_s is well defined since the the halfspaces defined by the torque constraints are closed and their number is $< \infty$.

3.6. 2D Example

The impact control scheme has been extensively tested on a simulated 2DOF planar robot as shown in Figure 3.6. with two point masses m_1 and m_2 (see [5]). It is assumed in the simulation that no gravitational effects are acting on the robot. The mass matrix equals

$$\mathbf{M}(\theta) = \begin{bmatrix} l_2^2 m_2 + 2q + l_1^2(m_1 + m_2) & l_2^2 m_2 + q \\ l_2^2 m_2 + q & l_2^2 m_2 \end{bmatrix} \tag{15}$$

where $q = l_1 l_2 m_2 \cos\theta_2$ and the velocity term equals

$$\mathbf{N}(\theta, \dot{\theta}) = \begin{bmatrix} -m_2 l_1 l_2 \dot{\theta}_2^2 \sin\theta_2 - 2m_2 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \sin\theta_2 \\ m_2 l_1 l_2 \dot{\theta}_1^2 \sin\theta_2 \end{bmatrix} \tag{16}$$

In the simulation both link lengths and both masses are set to 1. The time constant t_c is set to 1.The finite set P used for the simulation contains two points, p_1 and p_2 which are coincident with the location of the two point masses m_1 and m_2. The elements of P should be chosen carefully since they can only approximate the actual set of points on the surface. The impact energy constraints can only be guaranteed for points in P, unconsidered points may exceed the limits on the impact energy. Therefore, in an implementation of the impact control scheme P should be comprehensive enough to provide a good approximation. Because alignment of the impact and normal vectors is assumed, the orientation of the surface does not influence the result. However, the relation of positions to the center of mass and the joint axis have a strong influence. More elaborate sets P that consider proper physical dimensions of the robot give more precise results, but the applicability and correctness of the impact control theory remains as long as P is finite. Various tests of the robot, initially inside and outside the safe region have been conducted successfully. Figure 7 presents a simulation result. We have implemented a 4DOF Impact Controller for the WAM, we are now extending the controller to 7DOF.

3.7. Impact Control Architecture

Since the Impact Controller is not part of the actual motion control any motion control algorithm can be used with the Impact Controller. Because the Impact Controller lowers the saturation level of the actuator, the only motion algorithms that can be used must be appropriate for saturating actuators. An easy way to reduce saturation effects on the motion control algorithm is by feeding back the MIF of the robot and the distance of the current torque vector from the constraint boundaries to the motion control algorithm so the desired speed of the trajectory can be adjusted when the robot reaches saturation. The Impact Controller can also be used during the development of motion control algorithms when the possibility of software errors poses a serious risk to human operators.

4. Human-Robot Co-Existence

We see the main application area of human-friendly robots as helping hands in complex tasks that require more than person, and for the elderly or disabled. For such applications areas, people and robots must share the workspace. As a first step in this direction we are currently implementing a system which combines the perceptual abilities of the gaze tracking application with the dexterity of the WAM arm. Our face tracking system is utilized as an interface for human-robot interaction. Presently, we are considering two approaches to indicate a person's intention to a robot: Gestures and Gaze point.

Gesture recognition such as "yes" or "no" can be detected in real-time. Our group has already developed such a system [18]. Human-Robot interaction using gaze point could be realized in such a way that allows the selection of an object of interest by fixing one's eyes on the object. The robot could then pick up the object and pass it to the user. This is an intuitive way for people to interact with robots.

5. Conclusions

The vision-based interface that we have developed can determine the user's gaze point in real-time and in an unobtrusive manner. The mechanical design of the WAM robot is fundamental to the realisation of a human-friendly robot. Additional hardware and software safety measures provide a high level of safety. The encapsulation of the Zero-G module by the safety envelope guarantees minimum restrictions to achieve efficient robot motions. This provides the user with the abstraction of a light and gentle arm that only meekly resists when manipulated by a person. Human-friendly robots are now within technological reach.

Acknowledgements

Thanks to Jon Kieffer for his valuable help in this work.

References

[1] Tao Daniel Alter. 3d pose from three corresponding points under weak-perspective projection. Technical Report AIM-1378, MIT, AI Lab, July 92.

[2] A. Azarbayejani, T. Starner, B. Horowitz, and A. Pentland. Visually controlled graphics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):602–605, 1993.

[3] M.J. Black and Y. Yacoob. Tracking and recognizing rigid and non-rigid facial motions using parametric models of image motion. In *Proceedings of the International Conference on Computer Vision ICCV*, pages 374–381, Boston, MA, June 1995.

[4] S.M. Bozic. *Digital and Kalman Filtering*. London: Edward Anold, 1979.

[5] John J. Craig. *Introduction to Robotics*. Addison Wesley, 2 edition, 1986.

[6] A.H. Gee and R. Cipolla. Fast visual tracking by temporal consensus. *Image and Vision Computing*, 14:105–114, 1996.

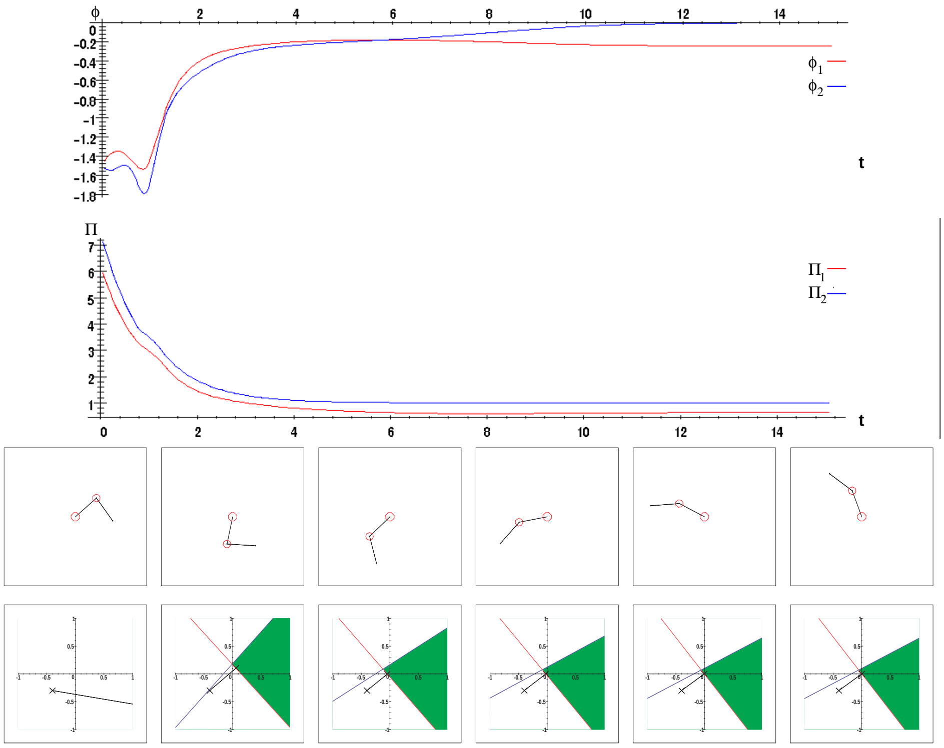

Figure 7: Simulation result of the robot initially in an unsafe state. According to the safety constraint defined in the Impact Controller, the MIF is reduced over time and approaches 1 asymptotically. The top graph shows the joint velocities (θ_1 continuous and θ_2 dotted), the second graph shows the impact energy of the two points considered (P_1 continuous and P_2 dotted). The two bottom rows show the robot configurations and the constraints in joint space.

[7] J. Heinzmann and A. Zelinsky. Robust real-time face tracking and gesture recognition. In *Proc. of the Int. Joint Conf. on Artificial Intelligence, IJCAI'97*, volume 2, pages 1525–1530, August 1997.

[8] J. Heinzmann and A. Zelinsky. 3-d facial pose and gaze point estimation using a robust real-time tracking paradigm. In *Proc. of the IEEE Int. Conf. on Automatic Face and Gesture Recognition*, pages 142–147, April 1998.

[9] J. Heinzmann and A. Zelinsky. A safe-control paradigm for human-robot interaction. *Journal on Intelligent Robotic Systems*, to appear, 1999.

[10] D. P. Huttenlocher and S. Ullman. Recognizing solid objects by alignment with an image. *International Journal of Computer Vision*, 5(2):195–212, November 1990.

[11] J. Hyde and M. Cutkosky. Controlling contact transitions. *IEEE Control Systems*, 14(1):25–30, 1994.

[12] T. Maurer and C. von der Malsburg. Tracking and learning graphs and pose on image sequences of faces. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition '96*, pages 176–181, 1996.

[13] A. Saulnier, M.-L. Viaud, and D. Geldreich. Real-time facial analysis and synthesis chain. In *Proceedings of the International Workshop on Automatic Face- and Gesture Recognition 95*, pages 86–91, Zürich, 1995.

[14] T.J. Tarn, Y. Wu, N. Xi, and A. Isidori. Force regulation and contact transition control. *IEEE Control Systems*, 16(1):32–40, 1996.

[15] W. T. Townsend and J. K. Salisbury. Mechanical design for whole-arm manipulation. *Robots and Biological Systems: Toward a New Bionics?*, pages 153–164, 1993.

[16] Ian D. Walker. Impact configurations and measures for kinematically redundant and multiple armed robot systems. *IEEE Trans. on Robotics and Automation*, 10(5):670–683, Oct 1994.

[17] Y. Xu, J.M. Hollerback, and D. Ma. A nonlinear pd controller for force and contact transient control. *IEEE Control Systems*, 15(1):15–21, 1995.

[18] A. Zelinsky and J. Heinzmann. Real-time visual recognition of facial gestures for human computer interaction. In *Proc. of the Int. Conf. on Automatic Face and Gesture Recognition*, pages 351–356. IEEE Computer Society Press, October 1996.