

SAFETY FOR HUMAN-ROBOT INTERACTION

by

DANICA (DANA) KULIĆ

B.A. Sc., The University of British Columbia, 1998

M.Eng., The University of British Columbia, 1998

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

THE FACULTY OF GRADUATE STUDIES

Mechanical Engineering

THE UNIVERSITY OF BRITISH COLUMBIA

December 2005

© Dana Kulić, 2005

Abstract

This thesis develops human–robot interaction strategies that ensure the safety of the human participant through planning and control. The control and planning strategies are based on explicit measures of danger during interaction. The level of danger is estimated based on factors influencing the impact force during a human-robot collision, such as the effective robot inertia, the relative velocity and the distance between the robot and the human.

A danger criterion is developed for use during path planning based on static and quasi-static danger factors, such as the relative distance and the overall robot inertia. A planner algorithm is proposed that minimizes this criterion. A danger index, developed for the real-time safe control module, tracks dynamic danger parameters such as the relative velocity and the effective inertia at the impact point. The safe control module uses this index to identify and respond to real-time hazards not anticipated in the planning stage. Both the planning and the real-time safe control strategy have been tested in simulation and experiments.

Another key requirement for improving safety is the ability of the robot to perceive its environment, and specifically the human behavior and reaction to robot movements. This thesis also examines the feasibility of using human monitoring information (such as head rotation and physiological monitoring) to further improve the safety of the human robot interaction. A human monitoring module is developed using machine vision and physiological signal monitoring. The vision component tracks the location of the human in the robot’s workspace, as well as the human head orientation. The physiological signal component monitors the human physiological signals such as heart rate, perspiration rate, and muscle contraction, and estimates the human emotional response based on these signals. If anxiety or stress is detected, the robot takes corrective action to respond to the human’s distress.

The planning, control and human monitoring components are integrated in a robotic system and tested with human subjects. A systematic and safe interaction strategy utilizing the methods described above, and applicable to a range of human-robot interaction tasks, is presented.

Table of Contents

Abstract	ii
Table of Contents	iv
List of Tables	vii
List of Figures	viii
Nomenclature	xi
Acknowledgements	xvii
Chapter 1: Introduction	1
1.1 Robot Safety.....	2
1.2 Objectives	3
1.3 System Overview	4
1.4 Contributions and Thesis Outline	6
Chapter 2: Literature Review	9
2.1 Safety for Human–Robot Interaction.....	9
2.1.1 Safety through Reactive Control	10
2.1.1.1 Impact Force Control	10
2.1.1.2 Safeguarding Zones	10
2.1.1.3 Danger Evaluation	12
2.1.2 Planning for Safety	13
2.2 Human monitoring for Human-Robot Interaction	15
2.2.1 Mechanical Systems	16
2.2.2 Visual Monitoring.....	17
2.2.3 Physiological Monitoring	19
Chapter 3: Test-bed Overview	22
3.1 Robot System	23
3.2 Human Monitoring.....	26
3.3 Communication Architecture.....	26
Chapter 4: Path Planning	29
4.1 Approach.....	29
4.1.1 Danger Criterion.....	31
4.1.1.1 Sum-Based Criterion.....	33
4.1.1.2 Product Based Criterion.....	35
4.1.2 Goal and Obstacle Potential Fields.....	36
4.1.3 The Overall Cost Function.....	38
4.2 Implementation	38
4.3 Search Strategy Improvements (Backwards Search)	40
4.4 Simulations	43

4.5	Summary	49
Chapter 5:	On-line Trajectory Planning and Control.....	50
5.1	Overview of the Trajectory Planning Module	50
5.1.1	Velocity Scaling Example	51
5.2	Real Time Safety Module	53
5.2.1	Safety Module Algorithm	53
5.2.1.1	Danger Index Formulation.....	54
5.2.1.2	A one-dimensional example.....	56
5.2.1.3	Stability Analysis	59
5.2.1.4	Real-time Algorithm	65
5.2.1.5	Implementation	68
5.2.1.6	Parameter Selection	69
5.2.2	Simulations	69
5.2.3	Experiments	73
5.2.4	Summary.....	75
Chapter 6:	Human Monitoring	76
6.1	Affective State Estimation	76
6.1.1	Affective State Inference	77
6.1.1.1	Data Processing and Feature Extraction	79
6.1.1.2	Fuzzy Inference Engine	86
6.1.2	Experiments	87
6.1.2.1	Trajectory Generation	88
6.1.2.2	Physiological Sensing	90
6.1.2.3	Experimental Procedure	91
6.1.3	Results	91
6.1.3.1	Subject Reported Response.....	91
6.1.3.2	Estimated Response from Physiological Sensors	95
6.1.4	Affective State Estimation Summary	99
6.2	Machine Vision for Human Monitoring	101
6.2.1	Head Tracking	101
6.2.1.1	Initial Face Detection	101
6.2.1.2	Face Tracking.....	103
6.2.1.3	Head Tracking Experiments.....	107
6.2.2	Head Orientation Estimation	110
6.2.2.1	Head Orientation Estimation Algorithm	110
6.2.2.2	Experimental Results for Head Orientation Estimation.....	115
6.2.3	Body Tracking	116
6.2.4	Summary.....	118
Chapter 7:	System Integration	119
7.1	Integrating Human Monitoring Data for Safe Planning and Control.....	119
7.1.1	Danger Index Modulation.....	120
7.1.2	Trajectory Scaling.....	122
7.2	Case Studies	123
7.3	Summary	141
Chapter 8:	Conclusions and Recommendations	142
8.1	Summary of Contributions.....	142
8.2	Future work.....	145
Bibliography.....	148	

Appendix A. Trajectory Planning	157
A.1.1 Single Joint motion between two points.....	157
A.1.1.1 Stop-Stop End Conditions.....	157
A.1.1.2 Non-zero velocity at end conditions	160
A.1.1.3 Calculating the Cubic Coefficients	162
A.1.2 Multiple Joint Motion.....	165
A.1.3 Waypoints Preprocessing and End Condition Generation	167
A.1.3.1 Generating the Waypoints.....	167
A.1.3.2 Determining the waypoint end conditions	168

List of Tables

Table 4.1. Planar robot simulations weights.....	44
Table 4.2. PUMA560 simulations weights.....	47
Table 5.1. Parameter values for simulations.....	70
Table 5.2. Parameter values for experiments.....	73
Table 6.1. Fuzzy inference engine rulebase.....	87
Table 6.2. Test path naming and descriptions.....	89
Table 6.3. Trajectory Execution Times.....	89
Table 6.4. Subjective results correlation analysis.....	95
Table 6.5. ANOVA for anxiety.....	95
Table 6.6. ANOVA for calm.....	95
Table 6.7. ANOVA for surprise.....	95
Table 6.8. Arousal ANOVA.....	97
Table 6.9. Correlation analysis for estimated arousal.....	97
Table 6.10. Confusion matrix - subject reported vs. estimated arousal.....	98
Table 6.11. Skin segmentation threshold values.....	102
Table A.1. Calculation of the start/end velocity and acceleration.....	161
Table A.2. Segment 2 cubic spline coefficients.....	162
Table A.3. Segment 2 duration.....	163
Table A.4. Segment 3 cubic spline coefficients and duration.....	163
Table A.5. Segment 4 cubic spline coefficients and duration.....	163
Table A.6. Segment 6 cubic spline coefficients.....	164
Table A.7. Segment 6 duration.....	165

List of Figures

Figure 1.1. System overview.....	6
Figure 3.1. System component overview.....	23
Figure 3.2. Experimental setup.....	25
Figure 3.3. Controller hardware.....	25
Figure 3.4. Communications architecture.....	27
Figure 4.1. Planning a safe interaction. Posture (b) has minimized potential hazard to the user.....	30
Figure 4.2. Human, robot representation in a non-interactive task.....	37
Figure 4.3. Human, robot representation in an interactive task.....	38
Figure 4.4. Combined backwards-forwards search algorithm flowchart.....	42
Figure 4.5. Planned path with sum-based danger criterion.....	43
Figure 4.6. Planned path with product-based danger criterion.....	43
Figure 4.7. Comparison between the sum-based and product-based danger criteria.....	44
Figure 4.8. Planned sequence for a PUMA560 robot (product danger criterion).....	46
Figure 4.9. Effect of the danger criterion search on the danger factors (product danger criterion).....	47
Figure 4.10. Planned sequence for a PUMA560 robot with backwards search (product danger criterion).....	48
Figure 5.1. Sample multi-axis path.....	52
Figure 5.2. Sample multi-axis motion with time scaling.....	53
Figure 5.3. Point robot moving between obstacles (one-dimensional case).....	57
Figure 5.4. Comparison between linear impedance and the danger index for a 1 DoF system.....	58
Figure 5.5. Phase portrait of the two obstacle case (1 DoF).....	59
Figure 5.6. Sample time trajectory for the two obstacle case (1 DoF).....	59
Figure 5.7. Plot of non-linear stiffness function $H(x_1)$	62
Figure 5.8. Pseudo code for the multi DoF algorithm.....	67
Figure 5.9. Safety module state diagram.....	68
Figure 5.10. Planar robot simulation (robot clears obstacles).....	71
Figure 5.11. Joint trajectory for planar robot simulation (robot clears obstacles).....	71
Figure 5.12. Planar robot simulation (robot cannot clear obstacles).....	72

Figure 5.13. Joint trajectory for planar robot simulation (robot cannot clear obstacles).....	72
Figure 5.14. CRS robot experiment video frames.....	74
Figure 5.15. Joints 1 - 3 trajectory during CRS robot experiment.	74
Figure 6.1. Sample heart rate signal (Subject 41).....	81
Figure 6.2. Typical SCR response (Subject 48).	83
Figure 6.3. Sample EMG Signal (Subject 17).....	85
Figure 6.4. Robot task positions (a = robot start/end position, b = pick position, c = place/reach position).	88
Figure 6.5. Path PP-PF (pick and place task planned with the potential field planner).	89
Figure 6.6. Path PP-S (pick and place task planned with the safe planner).	90
Figure 6.7. Path RR-PF (reach and retract task planned with the potential field planner).	90
Figure 6.8. Path RR-S (reach and retract task planned with the safe planner).	90
Figure 6.9. Subject reported average anxiety response.	93
Figure 6.10. Subject reported average calm response.	93
Figure 6.11. Subject reported average surprise response.	94
Figure 6.12. Subject reported average attention response.	94
Figure 6.13. Average estimated arousal from physiological sensors.	96
Figure 6.14. EMG activity during question answering.	99
Figure 6.15. Face detection on a sample image.....	103
Figure 6.16. Face detection results.	108
Figure 6.17. Face detection with skin coloured background.	108
Figure 6.18. Tracking results when estimating the face location, scale and orientation.	109
Figure 6.19. Tracking results when estimating the face location and scale only.	109
Figure 6.20. Tracking results under a variety of conditions.	109
Figure 6.21. Average images of the 4 feature vectors (from left to right) for 3 different poses: $\theta_v=0$, $\theta_h=0$ (left 4), $\theta_v=30$, $\theta_h=30$ (centre 4) and $\theta_v=60$, $\theta_h=0$ (right 4).	112
Figure 6.22. A series of cropped frames from the face tracker.	113
Figure 6.23. Three types of deviations of the face tracker.	114
Figure 6.24. Three head-frame series taken from the first experiment and the estimate poses are shown in the parentheses $[\theta_h, \theta_v]$	115
Figure 6.25. Body tracking screen capture.	117
Figure 7.1. Velocity Scaling Test Case.	124
Figure 7.2. Velocity Scaling Test Case danger index.....	125
Figure 7.3. Velocity Scaling Test Case trajectory scaling factor.	125
Figure 7.4. Velocity Scaling Test Case reference trajectory.	126
Figure 7.5. Path Obstruction Test Case 1.	127

Figure 7.6. Obstruction Test Case 1 danger index.	128
Figure 7.7. Obstruction Test Case 1 trajectory scaling.....	128
Figure 7.8. Obstruction Test Case 1 reference trajectory.	129
Figure 7.9. Hand and body tracking results (starting hand position).....	129
Figure 7.10. Hand and body tracking results (ending hand position).....	130
Figure 7.11. Path Obstruction Test Case 2.	131
Figure 7.12. Obstruction Test Case 2 danger index.	132
Figure 7.13. Obstruction Test Case 2 trajectory scaling.....	132
Figure 7.14. Obstruction Test Case 2 reference trajectory.	133
Figure 7.15. Affective State Test Case.....	134
Figure 7.16. Affective State Test Case danger index.	135
Figure 7.17. Affective State Test Case trajectory scaling.	135
Figure 7.18. Affective State Test Case joint trajectory.	136
Figure 7.19. Affective State Test Case estimated arousal.	136
Figure 7.20. Orientation Test Case video frames.	138
Figure 7.21. Orientation Test Case user head orientation.	138
Figure 7.22. Orientation Test Case danger index.	139
Figure 7.23. Orientation Test Case velocity scaling.....	139
Figure 7.24. Orientation Test Case reference trajectory.....	140
Figure 7.25. Head orientation tracking (horizontal angle = 0, vertical angle = 0).	140
Figure 7.26. Head orientation tracking (horizontal angle = -60 degrees, vertical angle = 0).....	141
Figure A.1. Long Motion Profile, joint reaches maximum velocity.	159
Figure A.2. Short Motion Profile. Distance traveled is too short to reach maximum velocity.	160
Figure A.3. Waypoint generation pseudo-code.	168
Figure A.4. Pseudocode for Determining the Section End Conditions.	170
Figure A.5. Pseudocode for Modifying the End Conditions.	171

Nomenclature

A	Jacobian matrix at the origin of the danger index based forcing function
A_m, B_m, C_m	Matrices defining the motion model of a head tracking particle
AR	Aspect Ratio
B	Damping constant
C_{ji}	Cubic or Quintic polynomial coefficients for trajectory segments
D	Bhattacharyya distance between two histograms
D_{CM}	Distance between the robot and person Centres of Mass
DC_{prod}	Product based Danger Criterion
DC_{sum}	Sum-based Danger Criterion
D_G	Distance between the robot end effector and the goal
DI_{TH}	Danger Index Threshold at which the safety module is activated
DI_t	Total danger index, including physical factors and information about the user
D_{max}	Distance between the robot centre of mass and the user at which the danger criterion due to the distance factor becomes zero
D_{min}	Minimum allowable distance between the robot centre of mass and the user
D_O	Distance between the robot and the nearest obstacle
D_{Omin}	Minimum safe distance between the robot and any obstacles for use during planning
D_{total}	Total distance traveled by a joint, across multiple trajectory sections
F	Forcing function based on the danger index
F_d	Damping force
F_{linear}	Forcing function based on linear stiffness and damping
F_{SO}	Virtual force on the robot in the presence of a single obstacle
F_{TO}	Virtual force on the robot in the presence of opposing obstacles
G	Gradient response function for face tracking

H	Non-linear stiffness function based on the danger index
I_b	3x3 Robot inertia tensor about the base
I_{CP}	Robot inertia at the critical point
I_{max}	Maximum safe value of the robot inertia
I_S	Inertia about the robot's sagittal plane
J	Overall planner cost function
K	Danger Criterion Scaling factor in the planner cost function
K_{AS}	Danger Index Scaling factor due to Affective State
K_{OR}	Danger Index Scaling factor due to Head Orientation
M	Number of particles used for head tracking
M_{AS}	Maximum increase in scaling due to the affective state
M_{OR}	Maximum increase in scaling due to head orientation
N_h	Number of bins in a histogram
N_S	Number of pixels in the ellipse perimeter
P	Positive definite matrix derived by Lyapunov's first method
R	Non-linear damping scaling function based on the danger index
S	Head scale in the image plane
S_{AS}	Slope of the affective state sigmoid function
S_{OR}	Slope of the head orientation scaling factor
T_i	Component function of the Lyapunov function V_2
V_i	Lyapunov functions
V_{max}	Velocity at which the contribution of the velocity factor to the danger index becomes one
V_{min}	Velocity at which the contribution of the velocity factor to the danger index becomes zero
W_d, W_i	Weights of the distance and inertia factors in the sum based danger criterion
W_D, W_G, W_O	Weighting factor for the danger, goal and obstacle criteria in the planning cost function
W_h, W_g	Relative weights of the histogram and gradient responses for face tracking
a	Estimated level of arousal
a_c	Midpoint of the arousal scale

a_{exp}	Non-zero joint acceleration at the start or end of a trajectory section
a_{hmax}	Maximum heart rate acceleration
a_{max}	Maximum allowable joint acceleration for trajectory planning
a_n	Normalized heart rate acceleration
a_{raw}	Measured heart rate acceleration
a_{start}, a_{end}	Joint acceleration at the start and end of a trajectory section
c	Desired rate of change of the trajectory parameterized time
c_f	Filtered corrugator EMG
c_n	Normalized corrugator EMG
c_{rest}	Resting corrugator EMG
d	Distance measure
d_{exp}	Distance traveled by a joint prior to the start or following the end of the current trajectory section
d_i	Distance traveled at the end of the trajectory segment i
d_{start}, d_{end}	Specified joint location at the start and end of a trajectory section
dt	Controller time step
f	Feature vector for head orientation estimation
f_{CMprod}	Distance Factor for the product based danger criterion
f_{CMsum}	Distance Factor for the sum based danger criterion
f_D	Distance factor for the danger index
f_G	Goal Seeking Function for the planner cost function
f_I	Inertia factor for the danger index
f_{Iprod}	Inertia Factor for the product based danger criterion
f_{Isum}	Inertia Factor for the sum based danger criterion
f_O	Obstacle Avoidance Function for the planner cost function
f_{pos}	Posture potential field function
f_V	Velocity factor for the danger index
g	Gradient of a Lyapunov function
\mathbf{g}	Array containing the intensity gradient response of all the pixels at the ellipse perimeter
h	Measured heart rate
h_{avg}	Average heart rate
h_i	Template and particle histograms
h_{max}	Maximum heart rate

h_{min}	Minimum heart rate
h_n	Normalized heart rate
J_{max}	Maximum allowable joint jerk for trajectory planning
m	Robot Mass
$p(\cdot)$	Probability function
q	Configuration joint angle
\dot{q}	Measured joint velocity
$quad_{max}, quad_{min}$	Upper and lower quadrant boundaries for the posture potential field function
r	Trajectory scaling parameterized time
s	Distance from the robot critical point to the nearest point on the person
s_f	Filtered skin conductance
s_{max}	Maximum skin conductance
$s_{max}^{resting}$	Maximum resting skin conductance
s_n	Normalized skin conductance
t_{crit}	Time required to complete the motion of the critical joint
t_{exp}	Time expired during joint motion prior to the start or following completion of the current section
t_i	Cumulative time at the end of each trajectory segment
t_i^{short}	Cumulative time at the end of short motion profile trajectory segment
t_{req}	Time required to complete the motion of a non-critical joint
t_{segi}	Time required to complete the motion of each segment i
t_{wait}	Time difference between the time required to complete the motion of the critical joint and a non-critical joint
\bar{v}	Unit vector normal to the robot sagittal plane
v	Relative velocity between the robot critical point and the nearest point on the person
v_{exp}	Velocity at the start/end of a trajectory section
v_i	Velocity at the end of each trajectory segment i
v_{max}	Maximum allowable joint velocity
v_{start}, v_{end}	Starting and ending velocity for a trajectory section
\mathbf{x}	State variable vector for each particle used for head tracking
x, y	Head location in the image plane

x_1	Current position of the robot
x_2	Current velocity of the robot
z	Head orientation feature vectors
$\alpha, \beta, \gamma, \delta$	Proposed component functions of the gradient g
λ_i	Eigenvalues of the Jacobian matrix at the origin of the forcing function
$\mu_{\theta_{xy}}, \sigma_{\theta_{xy}}^2$	Mean and the variance of the pixel at position (x,y) in the feature vector f for pose θ
θ	Head Pose
θ_h, θ_v	Degree of horizontal (pan) and vertical (tilt) head rotation
θ_i	Head orientation in the image plane
θ_c	Head orientation at the midpoint of the head orientation scaling factor
ANOVA	Analysis of Variance
ANSI	American National Standards Institute
bpm	Beats per minute
CoM	Centre of Mass
CP	Critical Point
DC	Danger Criterion
DI	Danger Index
DoF	Degrees of Freedom
dSCR	Rate of change of skin conductance response
ECG	Electrocardiogram
EMG	Electromyogram
HR	Heart Rate
GSR	Galvanic Skin Response
HRI	Human-Robot Interaction
HSV	Hue Saturation Value
IL	Intermediate Location
PC	Personal Computer
PCI	Peripheral Component Interconnect Bus
PF	Potential Field
PID	Proportional Integral Derivative

PP	Pick and Place
PWM	Pulse-Width Modulation
QRS Complex	Voltage signal obtained during a contraction of the left and right ventricles of the heart
RGB	Red Green Blue
RIA	Robotic Industries Association
RR	Reach and Retract
SCR	Skin Conductance Response
SP	Serial Port
Windows RTX	An extension to Windows NT, 2000 or XP from Venturecom that enables the PC to achieve hard real-time capability

Acknowledgements

I would like to thank my supervisor Dr. Elizabeth Croft for her support and guidance during this thesis. I am very grateful for all the great advice, our many brainstorming sessions and stimulating discussions, the stress-relieving runs through the endowment lands, and the merciless red pen. Beyond all the help with this thesis, she has been a patient career counselor, a trusted confidante and an inspiring mentor.

I would like to acknowledge Dave, Justin, Wilson and Vera for all their help setting up the robot controller and the vision system. Thanks also go to the current and former members of the CARIS lab: Bill, Damien, Daniela, Greg and Tao. It was truly a pleasure working and learning with all of you.

I would also like to thank my family for their help, support and encouragement. Boris, Iva, Matea and Palma, you have inspired me with your achievements, and instilled me with the confidence to set the bar high, and go for it.

Finally, I would like to thank Chris for his love, encouragement, help and unflagging support. The many, many hours spent making my slides and figures look awesome, listening to endless practice presentations, and “volunteering” for all the robot experiments are greatly appreciated.

Chapter 1: Introduction

Robots have been successfully employed in industrial settings to improve productivity and perform dangerous or monotonous tasks. Recently, research has focused on the potential for using robots to aid humans outside the strictly “industrial” environment, in medical, office or home settings. One important motivation for using service or personal robots is the aging population in the developed world [1, 2]. Robots that can interact with humans in a safe and friendly manner would allow more seniors to maintain their independence, and could alleviate some of the non-medical workload from health-care professionals. To this end, robots are being designed to perform home-care/daily living tasks¹ [3], such as dish clearing [4], co-operative load carrying [5, 6] and feeding [7, 8], and to provide social interaction [2, 9]. Robots are also increasingly marketed for entertainment purposes [10], and for home maintenance activities [11].

As robots move from isolated work cells to more unstructured and interactive environments, they will need to become better at acquiring and interpreting information about their environment [12]. One of the critical issues hampering the entry of robots into unstructured environments populated by humans is safety [13], and more broadly, dependability [14]. As defined by Lee, dependability incorporates both physical safety and operating robustness [14]. Some robots, which are intended primarily for social interaction [9, 10, 15], avoid safety issues by virtue of their small size and mass and limited manipulability. However, when the tasks of the interaction also include manipulation tasks, such as picking up and carrying items, assisting with dressing, opening and closing doors, etc, larger, more powerful robots will be employed. Such robots (e.g., articulated robots) must be able to interact with humans in a safe and friendly manner while performing their tasks.

¹ The five Activities of Daily Living (ADL) are: (i) transferring to and from bed, (ii) dressing, (iii) feeding, (iv) bathing, (v) toileting.

1.1 Robot Safety

The primary robot safety industrial standard in North America is the ANSI/RIA 15.06 Standard for Industrial Robots and Robot Systems – Safety Requirements [16]. This standard is written specifically for industrial robots, and is not applicable to autonomous or service robots. The standard prescribes that safety is achieved by separating personnel from the robot. For each robot, a restricted space is defined which includes the entire region reachable by any part of the robot, including any tools that may be held by the robot. The restricted space is a subspace of the safeguarded space, which is guarded to prevent hazards to personnel. The safeguarding must be implemented such that access to the hazard is prevented, or the cause of hazard is removed without requiring specific conscious action by the person(s) being protected. The prescribed action to be taken by the robot system upon detecting an intrusion into the safeguarding space is an emergency stop. The emergency stop removes all drive power and all other energy sources.

In Europe, EN-775 contains similar provisions for robotic safety. Similar to ANSI/RIA 15.06, this standard also requires that all persons be absent from the safeguarded space during automatic operation [17]. This implies that each robot must be surrounded by the safeguarding space and that the robot and robot tasks must be designed to allow the maximum number of tasks to be performed with personnel standing outside the safeguarding space.

In these industrial standards, the safety of human–robot interaction is effected by isolating the robot from the human [13, 16, 17]. In effect there is no interaction. As robotic applications transition from isolated, structured, industrial environments to interactive, unstructured, human workspaces, this approach is no longer tenable [13]. Three main approaches can be used to mitigate the risk during human-robot interaction: (i) redesign the system to eliminate the hazard, (ii) control the hazard through electronic or physical safeguards, and, (iii) warn the operator/user, either during operation or by training [16].

While the warn/train option has been used in industry, it had not been deemed effective in that setting [16], and is even less suitable for robot interaction with untrained users.

Examples of mechanical redesign include using a whole-body robot visco-elastic covering, and the use of spherical and compliant joints [18, 19]. Industrial experience indicates that eliminating hazards by design is the most effective risk reduction strategy [16]. However, in unstructured environments, mechanical design alone is not adequate to ensure safe and human friendly interaction. Additional safety measures, utilizing system control and planning, are necessary.

In order to ensure a safe interaction, the robot must be able to assess the level of danger in its current environment, and act to minimize that danger. Safety can further be enhanced if the robot is able to anticipate potential hazards in advance, and plan to avoid those hazards.

In addition, monitoring the human participant during the interaction provides valuable information. This information can enhance the safety of the interaction by providing a feedback signal to robot planning and control system actions [20, 21]. This type of monitoring is important for human–human interaction, where non-verbal cues such as eye-gaze direction, facial expression and gestures are all used as modes of communication [22]. Recently, the use of these interaction modes has received attention in the robotics research community. While the developed systems have been varied, the driving hypothesis has been the same: the safety and intuitiveness of the human-robot interaction can be improved through user monitoring [22, 23].

1.2 Objectives

To ensure the safety and intuitiveness of the interaction, a complete human-robot interaction system must incorporate (i) safe mechanical design, (ii) human friendly interfaces such as natural language interaction and (iii) safe control and planning strategies. This thesis considers the last item, namely, the design of safe control and planning strategies for human–robot interaction with articulated manipulators. Limited physical interaction tasks are considered which require large scale motion of the robot, such as handover tasks. It is envisioned that such a system could also be used

with close-contact interaction tasks, such as feeding, however, an additional module for target tracking and contact force control beyond the scope of this work would be required. The design of safe planning and control strategies is addressed via three components: safe planning, human reaction monitoring, and safe control. In particular, a key goal is to develop planning and control strategies to ensure that unsafe contact does not occur between any point on an articulated robot and a human in the robot's workspace. This work specifically aims to avoid human injury due to impact force during a collision. Other types of injury that could also be sustained during human-robot interaction (e.g., crushing, electrocution, etc.) [24, 25] are not considered.

The human monitoring component concerns the robot perception of the human participant of the interaction, estimating the level of awareness and approval the human exhibits towards the robot's actions, and then using that information in the planning and control algorithms. In this context both safety and perceived safety are considered.

Safe planning and control strategies are essential to ensuring safety during human-robot interaction. Although mechanical re-design of the robot can be used to decrease the force experienced upon impact, planning and control measures can be used to avoid impact altogether. In addition, in dealing with variable and unknown environments, the robot will not be able to rely on pre-existing models of the environment. Therefore, perception of the environment, and in particular, perception of humans in the environment is necessary for effective interaction. This thesis focuses on developing effective strategies for integrating the robot's perceptions into a safe control and planning strategy.

1.3 System Overview

The proposed system architecture assumes a user-directed interaction model. The user must initiate each interaction, but the robot has sufficient autonomy to perform commanded actions without detailed instructions from the user. An overview of the system is presented in Figure 1.1. The system architecture is similar to the hybrid deliberative and reactive paradigm described in [1]. An

approximate geometric path is generated in a slower outer loop, while the detailed trajectory planning and control are performed reactively in real time.

Motion is initiated by a user command (obtained through a command interpreter) that is passed to the planner. The planning functionality is divided into two parts: the global path planner and the local trajectory planner. The global planner module begins planning a geometric path for the robot over large segments of the task, utilizing the safety strategy proposed in Chapter 4. Segment end points are defined by locations where the robot must stop and execute a grip or release maneuver. For example, one path segment is defined from the initial position of the robot to the object to be picked up.

The local planner generates the trajectory along the globally planned path based on real-time information obtained during task execution. The local planner generates the required control signal at each control point. Because the local planner utilizes real-time information, it generates the trajectory on-line in short segments.

During the interaction, the user is monitored to assess the user's reactions to the robot. The local planner uses this information to modify the velocity of the robot along the planned path. The safety control module evaluates the safety of the plan generated by the trajectory planner at each control step. If a change in the environment is detected that threatens the safety of the interaction, the safety control module initiates a deviation from the planned path. This deviation will move the robot to a safer location. The local (trajectory) planner and the safety control module are described in Chapter 5. The human monitoring system is described in Chapter 6.

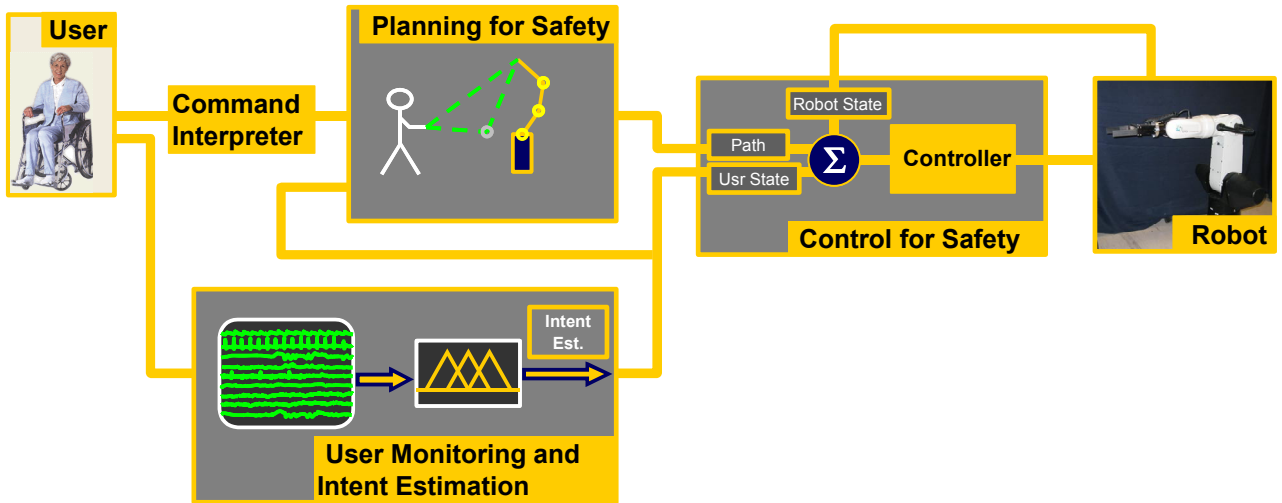


Figure 1.1. System overview.

1.4 Contributions and Thesis Outline

The contributions of this research are as follows:

1. A formulation of a metric for the level of danger in a human-robot interaction for use in planning and control.
2. Development of methodology for safe planning during human-robot interaction, that is applicable to any articulated robot (both non-redundant and redundant), and that can generate safe and valid paths through the entire workspace, including any singularities.
3. Formulation of a real-time reactive robot controller for use during unanticipated safety events. The controller minimizes an explicit measure of danger to reduce or eliminate the hazard. A stability analysis of the controller shows that the non-linear controller is stable throughout the state-space region used.
4. A human monitoring system for estimating the position and orientation of the human participant, and for estimating the human affective state during human-robot interaction. The human monitoring information is then incorporated into the planning and control system.
5. Implementation and testing of an exemplar human-robot interaction system.

The thesis is organized as follows:

- Chapter 2 This chapter reviews the related work in the literature on human-robot interaction with two foci. In the first section strategies for estimating and improving safety at both the planning and control levels are considered. In the second section strategies for improving human-robot interaction through human monitoring are overviewed.
- Chapter 3 In this chapter, the exemplar system used for validating the proposed methodology is described. The robot and sensing hardware are described, as well as the computer and communications architecture used.
- Chapter 4 This chapter presents the formulation of the metric for a danger criterion suitable for use during geometric path planning. A methodology for path planning using the danger criterion is provided, suitable for any open-chain articulated manipulator. Simulations are presented to verify this methodology. The results from this chapter have been published in the article “Safe Planning for Human–Robot Interaction” in the Journal of Robotic Systems [26].
- Chapter 5 In this chapter, a formulation for the danger index, a measure of real-time danger during the interaction, is developed. A real-time trajectory planning methodology for ensuring safety during human-robot interaction is presented. A trajectory scaling procedure is overviewed for use during nominal execution, which is used to modify the velocity along the planned path based on the danger index. When an unanticipated hazard is identified which invalidates the planned path, a reactive real-time trajectory planner is presented, which generates a new trajectory for the robot that minimizes the danger index. Simulations and experiments are presented demonstrating this approach. The real time controller portion of the chapter has been accepted for publication to the Journal of Robots and

Autonomous Systems [27].

- Chapter 6 This chapter details the use of human monitoring for improving the interaction. Two subsystems are used: vision and measurement of physiological signals. The vision section details the algorithms for estimating the human head and body position and the human head orientation from stereo camera images. Experimental results for the vision subsystem are also presented. The physiological monitoring section presents the formulation of an inference engine for estimating human affective state from physiological sensors during direct human-robot interaction. Results from a human-trial validating the inference engine are overviewed.
- Chapter 7 This chapter details the final system integration and testing. The integration of the subcomponents is described, including the fusion of data from multiple human monitoring sensors. User experiments are overviewed which demonstrate the behavior of the system under various conditions.
- Chapter 8 The final chapter overviews the main contributions of the thesis, and provides concluding remarks about the proposed algorithms for human-robot interaction and the developed system. Directions for future work are also outlined.

Chapter 2: Literature Review

This chapter overviews the existing work in human–robot interaction. To begin, Section 2.1 reviews existing strategies for improving safety. These can be broadly divided in path and motion planning for human-robot interaction, and danger estimation and reactive strategies for safety during real-time interaction. Section 2.2 reviews existing strategies for improving the interaction through human monitoring.

2.1 Safety for Human–Robot Interaction

Industrial safety standards [16] focus on ensuring safety by isolating the robot away from humans, and are, therefore, not directly applicable to human-robot interaction applications. However, industrial experience has shown that eliminating hazards through mechanical re-design is often the most effective safety strategy [16]. This approach has also been applied to interactive robots. For example, Yamada et al. [28] develop a whole-body robot viscoelastic covering. The impact force attenuation of the covering is selected based on the human pain tolerance limit. In [29], in addition to a viscoelastic covering, spherical joints are used and mechanical limits are installed on all joints to prevent pinching. Bicchi et al. [30, 31] advocate the use of compliant joints (McKibben actuators) to design an intrinsically safe system that is user back-drivable. Zinn et al. [32, 33] propose using distributed parallel actuation to lower the effective inertia of the robot.

While these and other mechanical re-design approaches have made contributions to reducing the impact force during a collision, they do not prevent the collision from occurring. To ensure safe and human friendly interaction in unstructured environments, additional safety measures, utilizing system control and planning, are necessary as discussed in the following sub-sections.

2.1.1 Safety through Reactive Control

2.1.1.1 Impact Force Control

Impact force controllers aim to ensure the safety of human-robot interaction by minimizing the impact force during human robot contact. Heinzmann and Zelinsky [34, 35] and Matsumoto et al. [20] propose a control scheme based on impact force control for any point on the robot. The robot is controlled such that the impact force with a static object does not exceed a preset value. The impact force controller acts as a saturating filter between the motion control algorithm and the robot.

Lew et al. [36] implemented three controller components to ensure safety when any point of the robot contacts a human, namely: inertia reduction, passivity, and parametric path planning. The inertia reduction controller applies a virtual force reducing the effective inertia of the robot. If a PID controller is used with inertia reduction and the control parameters are chosen to be positive definite, the proposed controller is shown to be passive². Finally, if a person holds the robot in place, parametric path planning is used to ensure that the position error does not accumulate.

As with mechanical re-design, impact force control aims to limit the impact force once collision has already occurred, thus limiting the potential for human injury. However, in these approaches, no control action is taken to attempt to avoid impact. For user safety during human-robot interaction, the controller design must include both reduction of impact force *and* minimization of the impact force through robot motion planning and control prior to impact.

2.1.1.2 Safeguarding Zones

Safeguarding type controllers execute a safety strategy if a person is detected within the work envelope of the robot. In industrial robotics, entry of an operator into a safeguarded zone causes an immediate emergency stop. While the concept of safeguarding zones has been proposed for service robots as well, multiple zones of progressive danger are usually defined to allow the robot to take

² A passive system is defined as a system that cannot deliver more energy than the amount of energy initially stored in than the system.

corrective action if possible. If a human is detected in the safeguarded zone, the default robot control sequence is altered to ensure safety of the human [4, 28, 37, 38].

Bearveldt [4] presents a dish-clearing robot working in an unstructured environment. Three operating zones are defined: when no human is in the work area, when a human is in the work area but at a safe distance, and when a human is an unsafe distance away from the robot. If no human is in the work area, the robot will operate at maximum speed. When a human is detected in the work area, but is still at a safe distance, the robot will operate at reduced speed. Once the human enters the unsafe area, an emergency stop is issued and all robot motion stops.

Yamada et al. [28] combine mechanical safety measures with the safeguarding concept. The robot is covered with a viscoelastic covering that both attenuates the impact force between the robot and the human and signals that the surface has been contacted. The space occupied by the viscoelastic covering itself is considered the safeguarded zone.

Zurada et al. [38] extend the safeguarding concept to handle sensor uncertainty. The system uses independent multiple sensors to detect the presence of intruders in the robot's workspace. A neural network is used to combine the individual sensor data into one overall grid. A fuzzy rule-base is used to generate one of three robot decisions based on the distance between the cell and the robot, and the belief that the cell contains an intruder. These decisions are: "move as intended", "slow down" or "emergency stop".

Karlsson et al. [37] present a similar system of multiple sensors for an industrial robot. The risk level is calculated based on how close the intruder is to the robot, and how fast and in which direction the robot is moving. Three risk level thresholds are defined. When the first threshold is exceeded, a warning is issued. At the second level, the robot motion is slowed down, and at the third threshold, an emergency stop is generated.

2.1.1.3 Danger Evaluation

The methods described above consider a fixed distance around the robot as the safeguarded zone, at which point the (reactive) controller performs a safety action. A more sophisticated approach is to develop a dynamically sized safeguarded zone, based on an explicit evaluation of the current danger. Traver et al. [21] propose two human friendly robotic designs. The “elusive robot” uses the distance between the robot and the human as the danger index. The “ergonomic robot” computes a danger factor based on the robot’s velocity and posture, the human’s direction of motion and eye gaze and the rate of change of the distance between the robot and the human. The “ergonomic robot” is controlled to reduce the calculated danger index. The danger index is used in conjunction with an obstacle avoidance strategy. If the danger index increases above a certain threshold level, the robot will deviate from its planned trajectory to avoid human contact. The potential field approach is used as the obstacle avoidance strategy. In this work, only preliminary simulation results are reported, but the key idea of integrating safety information based on physical factors with information about the user is proposed.

Ikuta et al. [25] developed a danger evaluation method using the potential impact force as an evaluation measure. In their work, the danger index is defined as a product of factors which affect the potential impact force between the robot and the human, such as relative distance, relative velocity, robot inertia and robot stiffness. The danger index can then be used to compare various mechanical design options, or as a control objective. Several design examples are presented, but no control based implementation of the danger index was presented.

Both safeguarding and danger evaluation approaches propose that robot behavior be modified based on the human location and motion during human-robot interaction. The safeguarding approaches define three discrete behaviors, while the danger evaluation methods generate a continuum of behavior. In this work, a continuum approach is taken to ensure a stable and seamless safe-control strategy for the robot behaviour.

2.1.2 Planning for Safety

Motion planning and the a priori identification of potentially hazardous situations as a means of reducing potential robot-safety hazards has received less attention than control-based (reactive) techniques. However, safe planning is important for any interaction that involves motion in a human environment, especially those that may contain additional obstacles. Application examples include service scenarios such as a dish clearing robot [4], services for the disabled, such as approaching the human for a feeding task [39, 40], and pick and place tasks for picking up and delivering common objects [1]. Including safety criteria at the planning stage can place the robot in a better position to respond to unanticipated safety events. Planning is thus used to improve the control outcome, similar to using smooth trajectory design to improve tracking [41, 42].

Several authors consider an a priori evaluation of the workspace to determine motion parameters within the various zones of the workspace [4, 28]. Blanco et al. [43] use distance measures from a laser scanner to generate a Voronoi diagram of the workspace of a mobile manipulator performing cooperative load carrying with a human. Since the Voronoi diagram maximizes distance from obstacles, paths generated along the Voronoi diagram present the safest course.

Khatib [44] developed the potential field approach. In this method, the environment is described by an attractive (goal) potential field, which acts on the end effector, and a repulsive (obstacle) potential field, which acts on the entire robot body. The potential field is specified in the operational space. The potential field is used to generate forces to pull the robot away from any obstacles, and the end effector towards the goal. This approach does not require extensive pre-computation of the global path, can operate on-line, and can be easily adapted to sensor based planning and dynamic obstacles. When a redundant robot is used, this approach can be extended to allow the robot to continue executing the task while avoiding obstacles [45]. Maciejewski and Klein [46] proposed a similar method for redundant manipulators and tasks where a goal trajectory is specified, and not just a goal location. In this approach, the force generated by the obstacle avoidance potential field is

mapped to the null space of the redundant manipulator, so that the robot can continue to execute the goal trajectory while using its redundant degrees of freedom to avoid obstacles. A major issue with these planning methods is that only local search is used, so the robot can reach a local-minimum that is not at the goal location. A second issue is the formulation of the forces applied to the robot in the operational space. This requires the use of the robot Jacobian to translate these forces to joint torques, and introduces position and velocity error near any robot singularities.

Nokata et al. [47] use a danger index based on the impact force between a human and the end effector. The danger index is the ratio of the actual force to the largest “safe” impact force (an impact force that does not cause injury to the human). The danger index is calculated based on factors such as the distance and velocity between the human and the manipulator end effector. Two approaches are proposed for planning the motion of a planar robot end effector: minimizing the greatest danger index along the path, and minimizing the total amount (integral) of the danger index along the path. However, their approach considers only the end effector motion.

Chen and Zalzal [48] use the distance between the robot and any obstacles as a measure of “safeness” in the cost function for path planning for mobile manipulators. A genetic programming approach is used to generate the optimum path given multiple optimization criteria, including actuator torque minimization, torque distribution between joints, obstacle avoidance and manipulability.

Oustaloup et al. [49] describe a method for path planning using potential fields. The method is described for mobile robots, but is extendable to robot manipulators in configuration space. In pre-planning, obstacles are classified according to how much danger they pose. The magnitude of the potential field gradient is varied by fractional integration, with a steeper slope for more dangerous obstacles. The fractional differentiation approach allows for a smooth transition between obstacles, however, this approach is susceptible to local minima.

Brock and Khatib [50] describe the Elastic Strips framework for motion planning for highly articulated robots moving in a human environment. This method assumes a rough plan for accomplishing the task is available, and is fine-tuned on-line based on changes in the environment.

The potential field method in operational space is used to plan the motion, with an attractive field pulling the robot towards the nominal off-line plan, and a repulsive force pushing the robot away from any obstacles. The existence of the pre-planned global path to the goal ensures that the robot does not get stuck in local minima. For redundant manipulators, an additional posture potential field is defined to specify a preferred posture for the robot. The posture field is projected into the null-space of the manipulator, so that it does not interfere with task execution. Although their paper does not deal explicitly with ensuring safety, the posture potential can be used to formulate safety-based constraints.

Most path planning algorithms for human environments focus on maximizing the distance between the robot and any obstacles in the environment. In this work it is proposed that the robot posture can also be optimized during path planning to significantly improve the safety of the manipulator.

In summary, safety for human-robot interaction requires an integrated approach of planning and control that continually evaluates the safety of both the planned and executing tasks to avoid unwanted impacts. At the same time, careful human monitoring is required to ensure that the task can be completed successfully. As discussed in the following section, the awareness and participation of the human in the task is key to allowing the human-robot interaction to proceed safely.

2.2 Human monitoring for Human-Robot Interaction

During human-robot interaction, monitoring of the human can provide valuable information, which can enhance the safety of the interaction and provide a feedback signal for robot actions. Robotic systems that monitor humans can be classified by the type of monitoring they perform. The simplest form of monitoring is the measurement of mechanical forces and displacements during a physical interaction with the robot, such as cooperative load carrying. Another category of monitoring systems is concerned with monitoring communication signals from the human. These types of systems can be further subdivided into visual monitoring or physiological monitoring

systems. Bien et al. [23] provide an overview of the types of communication signals that can be interpreted and used during interaction. These include intent signals, which directly represent the user's purpose, and emotional signals, which indirectly indicate reaction. Bien et al. advocate that soft computing methods are the most suitable methods for interpreting and classifying these types of signals, because these methods can deal with imprecise and incomplete data. .

2.2.1 Mechanical Systems

Mechanical systems measure human intent through the forces and motion imposed by the human during physical contact with the robot or with a common payload. An example of this type of system is the rhythm entrainment system developed by Maeda et al. [51]. The system implements a cooperative rope-turning task between a robot and a human. The goal of the system is to minimize unnecessary mechanical interaction between the two partners by synchronizing the motion of the robot to that of the human. This is accomplished by controlling both the phase and the frequency of the rope turning motion of the robot to match that of the human.

Arai et al. [5] develop a system for co-operative carrying of a long object by a human and a robot in the horizontal plane. The robot measures the forces being exerted by the human operator. Based on these forces, the robot impedance and motion is controlled such that the robot simulates a virtual nonholonomic constraint. A more sophisticated system for cooperative load carrying has been developed by Fernandez et al. [6]. This system allows for horizontal motion on the axis between the robot and the human, height change and rotation about the robot gripper. The human exerted force and torque are measured at the robot gripper, in order to actively assist the user in carrying out the identified task.

Another application where human intent can be read from a mechanical signal is in tasks where the robot can power-assist a human motion. Yamada et al. [52] use early motion of the human operator to estimate the operator's intent using a Hidden Markov Model. One drawback of both Fernandez [6] and Yamada's [52] systems is that only a limited number of operator motions are

anticipated by the system. If the operator initiates a motion not anticipated by the system, the system is likely to misclassify the motion.

2.2.2 Visual Monitoring

Visual monitoring systems utilize camera tracking of the human in the interaction and use this data to guide the interaction. This can include visual tracking of the user's eye gaze and head position, reading of the facial expression or hand gestures.

Song et al. [53] used an end-effector mounted camera to track facial features of the user in a wheelchair mounted robot. The robot is used in a feeding task. The position of the mouth is used as an input to the path planning algorithm, the motion of the robot is executed such that the mouth is centered in the end-effector camera view. In addition, the visual appearance of the mouth is used to assess the user's intention (mouth closed indicates negative intention, mouth opened indicates positive intention).

Traver et al. [21] present a system that handles close interaction between a robot and a human by estimating the danger level of the interaction and modifying the robot trajectory to reduce the danger level. The danger level is calculated based on the robot posture, the robot velocity and the gaze direction of the human. The danger index increases with increased inertia and velocity of the robot, as well as when the person is looking away from the robot.

Heinzmann and Zelinsky [34] and Matsumoto et al. [20] advocate the use of visual interfaces and safe mechanisms as key components of human friendly robots. They present two systems for accurately tracking the eye gaze of the user based on a three level algorithm. The low level tracking consists of template matching for facial features. These features are then used to calculate the head position of a 3D geometrical model of the head. Based on the location of the facial templates and the 3D model, a Kalman filter is used to predict future feature locations. This allows the system to keep tracking when some of the features are no longer visible. The eye gaze can then be extracted from the 3D model eyeball location. In [34], a single camera is used, in [20] a stereo system based on the same

algorithm is described. Although accurate tracking of the eye and head can only be implemented if a close-up of view of the human is available, this system provides a robust gaze tracking method suitable for real-time human-robot interaction.

Stiefelhagen et al. [54] present a method for estimating gaze detection based on head position alone. Because the eye position is not used this method is not useful to determine exact eye gaze location, but can be used when only a general direction area is required. Another advantage of this system is that multiple users can be processed at the same time. The system locates and extracts faces from the camera image using a statistical skin model, and using heuristics to separate hands from faces. Once faces are identified, each face image is preprocessed to normalize the image against different lighting conditions and to pre-detect vertical and horizontal edges. Two neural networks (one for pan and one for tilt) are then used to estimate the head pan/tilt rotation. About 9 degree accuracy is achieved in the worst case. The rotation of the head alone is then used to estimate the focus of attention, by identifying discrete objects in the environment the user could be looking at.

Head tracking can also be implemented by tracking the eyes or pupils [55, 56]. Even if the exact direction of the eye gaze is not predicted, the detection of pupils can be used as an indicator of whether the person is looking towards the camera (robot) or not. However, pupil detection only provides a binary signal of user awareness, rather than a continuous signal which can be obtained from head or eye-gaze tracking, as described above.

As seen from this brief review, vision based systems are commonly used in human-robot interaction research to locate the presence and location of the human participant, and to determine if the user is aware of and looking at the robot. In this work, the vision system is also used as a tool to provide necessary use location information as well as to provide basic awareness information. However, as discussed in the following sub-section, other monitoring methods can be used to provide a richer view of the users' awareness and intent with respect to the robot.

2.2.3 Physiological Monitoring

Physiological monitoring systems use physiological signals from the user to extract information about the user's reaction to robot motion or actions. Many different physiological signals have been proposed for use in human-computer interfaces, including skin conductance, heart rate, pupil dilation and brain and muscle neural activity. Some of these interfaces have also been proposed for use in human-robot interaction. Although physiological signals have the potential to provide objective measures of the human's emotional response, they are difficult to interpret. One problem comes from the large variability in physiological response from person to person. Another problem is that usually, the same physiological signal is triggered for a range of psychological states; it can be difficult for a controller to determine which emotional state the subject is in, or whether the response was caused by an action of the system, or by an external stimulus. For these reasons, researchers in psychophysiology recommend using more than one signal source, for example, both heart rate and galvanic skin response (GSR), instead of only one indicator. However, human-robot interfaces developed thus far have most frequently used only one physiological mechanism. This stems in part from the difficulty of measuring physiological signals unobtrusively.

Takahashi et al. [57] propose a feeding robot for the disabled. The robot motion is controlled by the user via a head-pointing device. The robot has a camera affixed to its end effector; images from the camera are displayed on a PC monitor. The user uses the head-pointing device to guide robot motion, by pointing and clicking on the camera image displayed on the screen. In order to determine the velocity of the robot, the galvanic skin response of the user is measured. Galvanic skin response is directly related to arousal; an increase in the skin conductivity is linked to both startle and defense responses. The system treats increases in galvanic skin conductivity as signals from the user to decrease robot velocity.

Yamada et al. [58] propose measuring pupil size as an indication of human fear in an interaction with a robot. They conducted tests on 20 subjects, whose eye gaze and pupil size were measured as

the robot's acceleration and jerk were measured. It was assumed that an increase in pupil size was a direct indication of fear. These tests showed that acceptable acceleration and speed are higher when the robot is further away from the subject, and that jerky motion causes fear more easily than smooth acceleration. The authors also tested the possibility of using GSR to measure fear. GSR measurements were rejected because it was found that increases in GSR were also related to body movement, so it would be difficult to differentiate between changes in signal caused by movement and changes in signal caused by a reaction to robot motion.

Sarkar proposes using multiple physiological signals to estimate emotional state, and using this estimate to modify robotic actions to make the user more comfortable [59]. Rani et al. [60] propose analyzing the frequency content of the heart rate signal to distinguish different levels of anxiety during human-robot co-operation. The inter-beat interval (time between consecutive heart beats) is analyzed using the discrete time wavelet transform. The frequency content data are then used to classify the subject's level of anxiety using a fuzzy inference engine. The system was tested on two subjects, using video game playing to elicit the various levels of anxiety and corresponding heart rate signal. In [61], the frequency domain heart rate analysis is combined with skin conductance activity and corrugator and masseter muscle activity to measure human stress. These signals are analyzed with a fuzzy inference engine to estimate stress. The stress information is then used by an autonomous mobile robot to return to the human if the human is in distress. In this case, the robot is not directly interacting with the human; physiological information is used to allow the robot to assess the human's condition in a rescue situation.

Nonaka et al. [62] describe a set of experiments where human response to pick-and-place motions of a virtual humanoid robot is evaluated. In their experiment, a virtual reality display is used to depict the robot. Human response is measured through heart rate measurements and subjective responses. No relationship was found between the heart rate and robot motion, but a correlation was reported between the robot velocity and the subject's rating of "fear" and "surprise".

Research on the use of physiological signals for human-robot interaction is still at the earliest stages. Few results have been reported using physical interaction experiments with multiple sensors. However, physiological sensors present a promising area of research, as they are easier and faster to measure and analyze than vision based data.

Chapter 3: Test-bed Overview³

In order to validate the formulations and methodologies proposed in this work, a test-bed system comprising a robot arm, controller, software and various physiological sensors, as well as an image tracking system were developed for this work. This chapter presents the system software architecture, overviews the robot setup, the developed hardware and software, and the overall system communication architecture.

The system component overview is shown in Figure 3.1. The system architecture is similar to the hybrid deliberative and reactive paradigm described in [1]. An approximate geometric path is generated in a slower outer loop, while the detailed trajectory planning and control are performed reactively in real time. The user issues a command to the robot to initiate the interaction. The command interpreter translates the natural language command (e.g.: pick up the red cup) into a set of target locations and actions (e.g., execute a grip maneuver at coordinates $[x,y,z]$). The planning module is divided into two parts: the global path planner and the local trajectory planner. The global planner module begins planning a geometric path for the robot over large segments of the task. Segment end points are defined by locations where the robot must stop and execute a grip or release maneuver. For example, one path segment is defined from the initial position of the robot to the object to be picked up. The local planner generates the trajectory along the globally planned path, based on real-time information obtained during task execution. The local planner generates the required control signal at each control point. Because the local planner utilizes real-time information, it generates the trajectory in short segments. During the interaction, the user is monitored to assess

³ The development of the test-bed was completed with the help of two undergraduate students working under the candidate's supervision [63, 64]

the user’s level of approval of robot actions. The local planner uses this information to modify the velocity of the robot along the planned path. The safety control module evaluates the safety of the plan generated by the trajectory planner at each control step. If a change in the environment is detected that threatens the safety of the interaction, the safety control module initiates a deviation from the planned path. This deviation will move the robot to a safer location. Meanwhile, the recovery evaluator will initiate a re-assessment of the plan and initiate re-planning if necessary.

The inner control loop operates at 1kHz. The safety measure estimation loop, which depends on the user location estimates from the vision system, operates at 5 – 10 Hz. The user reaction estimation from physiological signals is updated at ~0.5 Hz, while the outer, planning loop does not execute in real-time, and is event based.

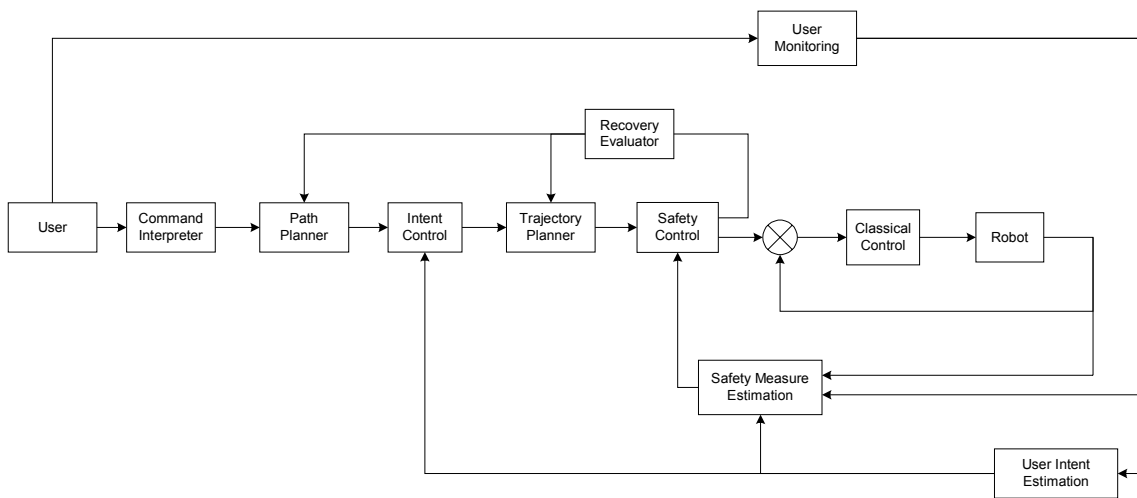


Figure 3.1. System component overview.

3.1 Robot System

The system was tested with the CRS A460 6-DoF manipulator, shown in Figure 3.2. The CRS A460 is a typical laboratory scale robot with a payload of 1kg, which can be used for performing tabletop assistive activities. An in-house, open-architecture controller was developed for operating the robot [63, 64]. The hardware controller includes a power transformation stage, which provides DC power to the joint amplifiers, 3 PWM amplifiers for the lower 3 joints, an integrated linear

amplifier card for the wrist joints, and digital signal processing for reading the joint encoders and homing switches. The controller is shown in Figure 3.3. The hardware controller also contains a digital emergency brake circuit, connected to the controller PC. The emergency brake circuit applies the mechanical brake on joints 2 and 3, and removes power from the robot amplifiers. Power supply is maintained to the encoder circuitry, so that robot position information is maintained by the controller in case of an emergency shutdown.

The hardware is controlled through a Quanser MultiQ PCI card and WinCon software [65], running on a Pentium 4 2.8 GHz computer with a Windows RTX extension. The Quanser software allows Mathworks Simulink diagrams to be automatically converted to real-time executables. All the on-line control software was written in Simulink and as C language S-functions. The low-level controller, safety module, trajectory planner and recovery evaluator execute on the controller computer. At the low level, a proportional-integral-derivative (PID) controller was implemented. The safety module, the trajectory planner and the low level controller execute at a frequency of 1 kHz.

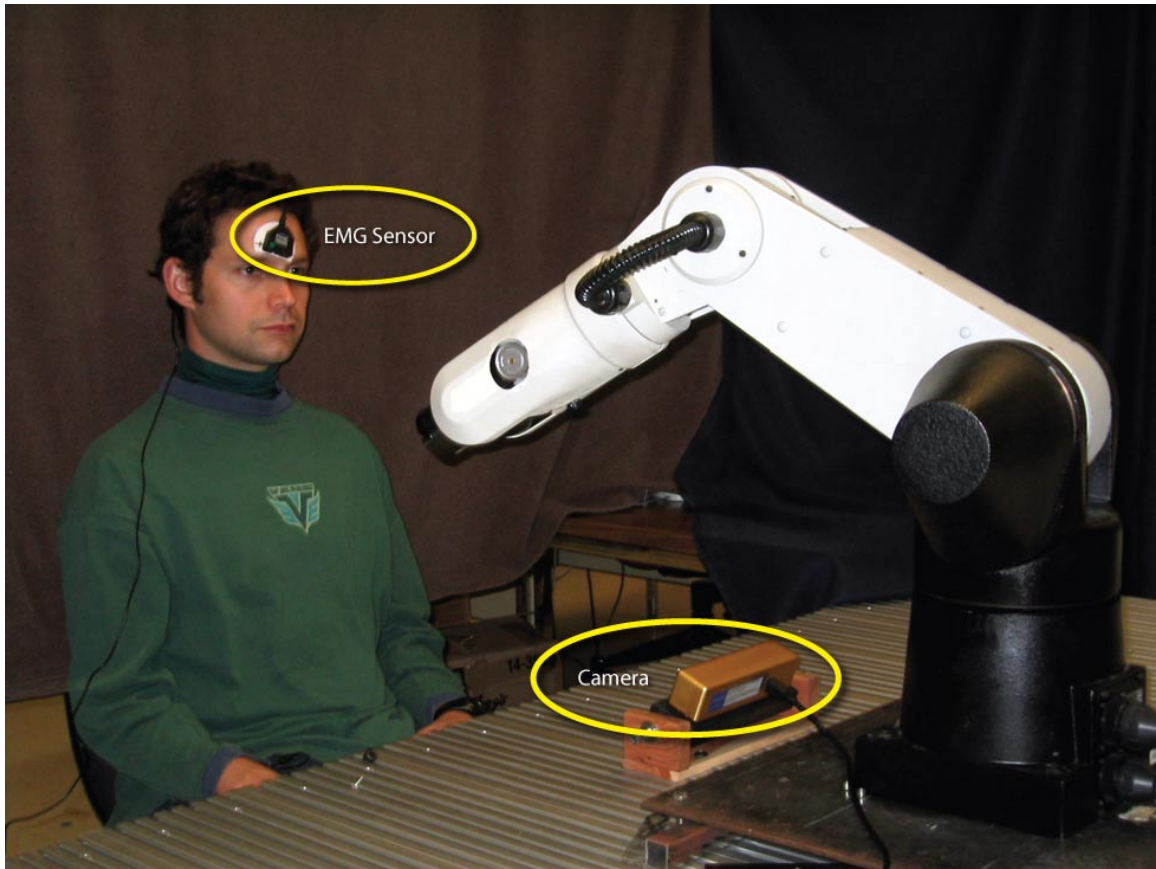


Figure 3.2. Experimental setup.

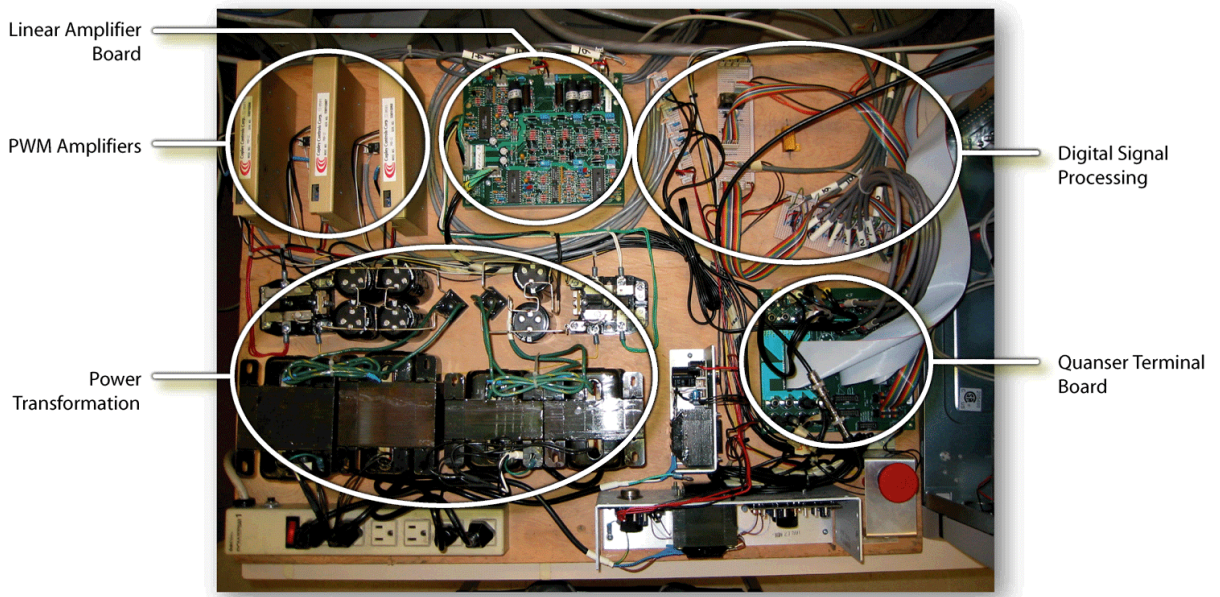


Figure 3.3. Controller hardware.

3.2 Human Monitoring

The video images used for human visual monitoring were obtained from a Point Grey Bumblebee [66] stereo camera mounted in front of the robot base and facing the approximate user location, as shown in Figure 3.2. The camera algorithms ran on a separate Pentium 4 2.8 GHz computer, linked to the controller via a serial connection. For the user's head position estimation, the left image of the stereo camera was used. Both cameras were used to estimate the 3-D location of the human in the environment. The Point Grey stereo routines were used to extract a depth map of the environment. The depth map was then used to generate the estimates of the user location, as described in Chapter 6.

The ProComp Infinity system from Thought Technology [67] was used to gather the user physiological data. The data from the physiological measurement system was collected and processed on a separate Pentium 4 2.8 GHz computer, linked to the robot controller via a serial connection. Heart muscle activity, skin conductance and corrugator muscle activity were measured. In an early feasibility study [68], the respiration rate was also measured, but it was found that data from this sensor is too slow to use in real-time interaction, so respiration rate was not used. The heart muscle activity was measured via electro cardiogram (ECG) measurement using the EKG Flex/Pro sensor. The skin conductance was measured using the SCFlex-Pro sensor. Corrugator muscle activity was measured with the Myoscan Pro electromyography (EMG) sensor. All sensor data was collected at 256 Hz. This rate is sufficient for capturing physiological signal events.

3.3 Communication Architecture

The system processing takes place on 4 separate computers: the planning computer (Planning PC), the camera computer (Camera PC), the physiological sensors computer (Sensors PC) and the robot controller (Robot PC). The communications architecture is shown in Figure 3.4. All the on-line processing (danger index evaluation, trajectory planning, safety module, low level control and recovery evaluator) is executed on the robot controller. It was necessary to separate the processing

onto multiple PCs, to allow the robot controller to execute in real-time at 1kHz, while the other PCs execute at lower rates, or perform non-deterministic processing. Each PC is connected to the robot controller through a separate serial port interface. Serial communications were used because that was the only form of communications allowed by Wincon when executing in real-time mode.

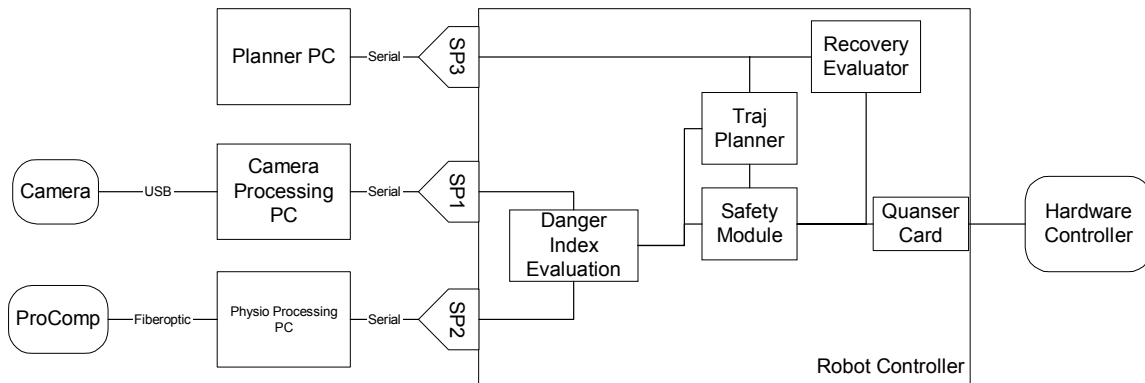


Figure 3.4. Communications architecture.

The communication between the Planner PC and the Robot PC is event based; i.e., new information is transmitted by the planner only in the event of a user command, or at the request from the recovery evaluator. Once the Planner PC generates a new path, it transmits the number of points in the new path, and then all the configurations on the path. Communication between the Camera PC and the Robot PC is periodic but non-deterministic, since the vision processing algorithms do not execute in hard real time. Information flow is one way, from the Camera PC to the Robot PC only. Each time the new user location and head orientation is estimated, it is transmitted to the robot controller.

Communication between the Sensors PC and the Robot PC is one way and runs at approximately 256 Hz, proportional to the physiological signals data collection rate. This rate is also non-deterministic, since the Sensors PC is not running the Windows real-time extension. At each physiological data processing cycle, an estimate of the user affective state is generated and transmitted to the Robot PC.

The experimental use of individual components of the experimental setup is described in Chapters 5 and 6. The integrated system is demonstrated in Chapter 7.

Chapter 4: Path Planning⁴

Path planning for safety is an important component of an overall safety strategy for human–robot interaction; however, it has received less attention than control and impact strategies. Including safety criteria at the planning stage can place the robot in a better position to respond to unanticipated safety events. Planning is used to improve the control outcome, similar to using smooth trajectory design to improve tracking [41, 42]. Herein, a similar approach to [47] is considered. However, in order to address safety in unstructured environments, the whole arm configuration of the manipulator, rather than only the end-effector state, is considered in the planning stage. Within this context, potential danger criteria are formulated and evaluated, using a motion planning framework similar to [50]. Each proposed criterion explicitly considers the manipulator inertia and centre of mass location with respect to the user to evaluate danger. A two stage planning approach is proposed to address issues of potentially conflicting planning criteria. The proposed approach is evaluated in simulation to compare the criteria and to demonstrate their efficacy in an example handoff task.

4.1 Approach

A hazard requiring a change in robot behavior can be defined by a minimum distance between the robot and the person [4, 69], or by using a threshold level of the danger index based on impact force [25, 47]. In this work, an index similar to [25, 47] is proposed, and applied to configuration space planning of the robot motion. By selecting safer configurations at the planning stage, potential hazards can be avoided, and the computational load for hazard response during real time control can

⁴ A version of this chapter has been published. D. Kulic and E. Croft, “Safe Planning for Human-Robot Interaction,” *Journal of Robotic Systems*, vol. 22, no. 7, pp. 383 – 396, 2005.

be reduced, as shown in Figure 4.1. In both panels the robot has the same end-effector location, but in the panel of the right (b), the hazard to the user is minimized by the posture adopted by the robot.

Safe planning is an important component of the safety strategy. For example, if the path to be followed is planned with a general path planning method, the robot may spend the majority of the path in high inertia configurations. If the user suddenly moves closer to the robot, the potential collision impact force will be much higher than if the robot had been in a low inertia configuration, regardless of the real-time controller used to deal with potential collision events.

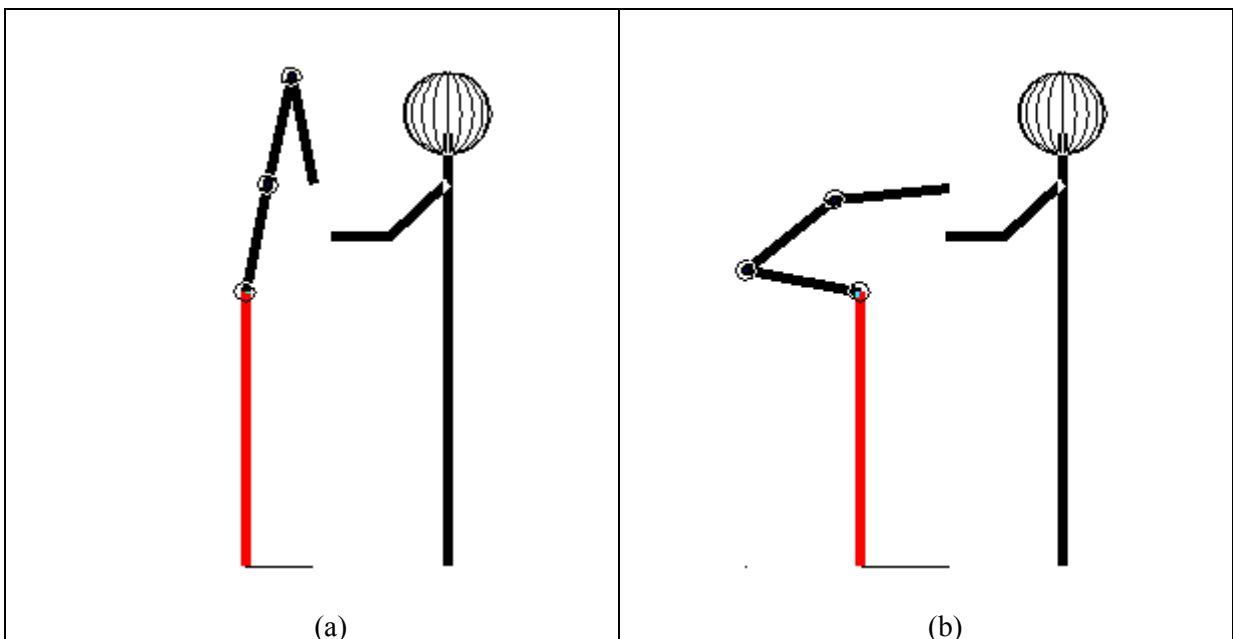


Figure 4.1. Planning a safe interaction. Posture (b) has minimized potential hazard to the user.

When selecting a path planning strategy, there is a tradeoff between fast local methods that may fail to find the goal, and slow global methods [70]. To exploit advantages of both methods, recent path planning algorithms have used a hybrid approach, where global path planning is used to find a coarse region through which the robot should pass, and local methods are used to find the exact path through the region [50].

Similarly, in this approach, the global planner generates a safe contiguous region in space through which the robot can move to complete the given task. This region in space is described by a set of

contiguous configurations, which represent the path. It is then left to the on-line trajectory planner to generate the exact path in the region, and the trajectory along that path. This trajectory is evaluated and, if necessary, corrected at every control step by the safety module to handle the real-time aspects of the interaction.

Since the task planning is done following a user request, the global planner must execute within several seconds at most, to avoid a significant delay between a user request and robot response. To ease the computational load on the global planner, the task is separated into segments. Natural segment separation points occur when the robot is required to pause at a particular location, for example at each grasp or release point. Only the first segment must be planned before the planned path can be passed on to the local planner and the robot can begin executing the task. In this way, global planning of the next segment can continue in parallel with execution of the current segment.

4.1.1 Danger Criterion

The planning module uses the best-first planning approach [70]. In this method, the robot configuration space is discretized into 0.1 rad cells, and a path is found by iteratively building a tree of configurations, with the first configuration at the root. At each iteration step, the neighbours of the leaf configuration with the lowest cost value are added to the tree. The algorithm therefore follows the steepest descent of the cost function, and escapes from local minima by well-filling. The search stops when the target configuration is reached or the entire space has been mapped. The algorithm is resolution complete. In cases when the number of degrees of freedom (DoF) of the robot affecting gross end-effector motion are small (less than 5), the best-first planning approach provides a fast and reliable solution [70]. For highly redundant robots, a different search strategy can be employed, such as randomized planning [71], or probabilistic roadmap planning [72-75]. For example, either random sampling of the configuration space [74], or the Generalized Voronoi Graph [73], can be used to generate a roadmap of the connected free regions in the configuration space. The roadmap represents a subspace of the entire configuration space. The search based on lowering the danger criterion can

then be applied to the roadmap, rather than the entire configuration space, reducing the search time for high DoF manipulators. However, the search criteria presented herein remain identical regardless of the search strategy used. The safest path is found by searching for contiguous regions that: (i) remain free of obstacles, (ii) lead to the goal, and, (iii) minimize a measure of danger (a danger criterion). The planning algorithm seeks a path that minimizes a cost function consisting of a quadratic goal seeking function, a quadratic obstacle avoidance function, and the danger criterion (DC).

The danger criterion is the central contribution of the planner cost function. Since path planning (as opposed to trajectory planning) does not consider robot velocities, a configuration-based (quasi-static) danger criterion is required. To be effective, the danger criterion should be constructed from measures that contribute to reducing the impact force in the case of unexpected human-robot impact, as well as reducing of the likelihood of impact. These can include the relative distance between the robot and the user, the robot stiffness, the robot inertia, the end-effector movement between contiguous configurations, or some combination of these measures, similar to those proposed in [25]. In [47], Nokata et al. use the danger index to find an optimum safe path, however, only the end-effector trajectory with respect to the user is considered. Herein, a safe path for the entire robot structure is planned, explicitly planning the robot posture. However, since some of the factors affecting danger can conflict (e.g., a low stiffness configuration can also be high inertia configuration) it is important to formulate the danger criterion so that conflicting factors do not act to cancel each other out. Herein, the robot inertia and the relative distance between the robot and the user's center of mass are used. The robot stiffness was not included as it can be more effectively lowered through mechanical design [25, 30]. The proposed approach modifies the robot inertia instead, which lowers the effective impedance of the robot. Dynamic factors, such as the relative velocity and acceleration between the robot and the user, are handled by the trajectory planner and the safety module, as will be discussed in Chapter 5.

For optimization purposes, a scalar value representing the effective robot inertia at each configuration must be computed. For a general robot architecture, where the robot's inertia may be distributed in more than one plane, the largest eigenvalue of the 3x3 inertia tensor may be used as the scalar measure. For robots with a single sagittal⁵ plane (e.g. anthropomorphic, SCARA), the scalar inertial value is extracted by calculating the robot inertia about an axis originating at the robot base and normal to the robot's sagittal plane.

$$I_s = \bar{v}^T I_b \bar{v}. \quad (4.1)$$

Here, I_s is the inertia about the v axis, v is the unit vector normal to the robot sagittal plane and I_b is the 3x3 robot inertia tensor about the base.

For each danger criterion factor, a potential field function is formulated as a quadratic function. Quadratic potential functions are most commonly used in general potential field planners. They have good stabilization characteristics, since the gradient converges linearly towards zero as the robot's configuration approaches the minimum [44, 70].

4.1.1.1 Sum-Based Criterion

Two danger criterion formulations are proposed: a sum-based and a product-based criterion. For the sum-based danger criterion, the inertia factor is:

$$f_{I_{sum}}(I_s) = \frac{I_s}{m}, \quad (4.2)$$

where, m is the total mass of the robot, and is used as a normalization factor, such that the units are compatible with the goal attraction function. This function can be interpreted as a quadratic attractive function, attracting each link towards the robot base.

The relative distance factor for the sum-based danger criterion is implemented by a repulsive function between the user and the robot center of mass (CoM). The center of mass distance is used (instead of the closest point distance) to allow the robot end-effector to contact the user during

⁵ The sagittal plane is the vertical plane (plane of symmetry) passing through the center of the outstretched robot arm.

interaction tasks, while maximizing the distance between the user and the bulk of the robot. The potential field is described by equation (4.3) below:

$$f_{CM_{sum}}(D_{CM}) = \begin{cases} D_{CMO} = |D_{CM} - D_{min}| & \\ \frac{1}{2\varepsilon} : & D_{CMO} \leq \varepsilon \\ \frac{1}{2} \left(\frac{1}{D_{CMO}} - \frac{1}{D_{max}} \right)^2 : & \varepsilon < D_{CMO} < D_{max} \\ 0 : & D_{CMO} \geq D_{max} \end{cases} \quad (4.3)$$

In Equation (4.3), D_{CM} is the current distance between the robot center of mass and the user, D_{min} is the minimum allowable distance between the robot centre of mass and the user, and D_{CMO} is computed as the difference between these two distance measures (namely, the separation distance above the minimum limit). D_{max} is the distance at which D_{CMO} , the current distance above the minimum limit, no longer contributes to the cost function (for example, if no human is visible in the environment). ε is a small number used to limit the function for D_{CM} near D_{min} . This potential field is analogous to an obstacle potential field acting between the centers of mass of the user and the robot.

The sum based danger criterion is comprised of the inertia factor (Equation (4.2)) and the centre of mass distance factor described above (Equation (4.3)), as follows:

$$DC_{sum} = W_i \cdot f_{I_{sum}}(I_s) + W_d \cdot f_{CM_{sum}}(D_{CM}) \quad (4.4)$$

Here, W_i and W_d are weights of the inertia and distance term, scaled such that $W_i + W_d = 1$. The weights W_i and W_d are tuned based on the particular robot structure. For low inertia robots, and when the robot is close to the user, the distance factor will dominate the danger criterion, because the distance factor approaches infinity as the robot approaches the person. If inertia reducing behavior is desired for the path in these cases, W_i should be greater than W_d .

4.1.1.2 Product Based Criterion

For the product-based danger criterion, the criteria are scaled such that for each potential function, the level of danger is indicated within the range [0 – 1]. Values greater than one indicate an unsafe configuration.

The product based inertia criterion is defined as:

$$f_{I_{prod}}(I_s) = \frac{I_s}{I_{\max}}, \quad (4.5)$$

where, I_{\max} is the maximum safe value of the robot inertia. In the simulations described in Section 4.4, the maximum robot inertia is used; however, a lower value can be used for high-inertia manipulators. In this case, the maximum safe value can be established based on the largest force magnitude that does not cause pain [76] and the maximum robot acceleration.

For the product based distance criterion, similar to the sum based distance criterion, the center of mass distance between the robot and the user is used. The relative distance criterion for the product-based danger criterion is:

$$f_{CM_{prod}}(D_{CM}) = \begin{cases} k \left(\frac{1}{D_{CM}} - \frac{1}{D_{\max}} \right)^2 & : D_{CM} \leq D_{\max} \\ 0 & : D_{CM} > D_{\max} \end{cases} \quad (4.6)$$

The scaling constant k is used to scale the potential function such that the value of the potential function is zero when the distance between the user and the robot is large enough (larger than D_{\max}), and is one when the distance between the user and the robot is the minimum allowable distance (D_{\min}):

$$k = \left(\frac{D_{\min} \cdot D_{\max}}{D_{\min} - D_{\max}} \right)^2 \quad (4.7)$$

Values of the product-based distance criterion above one indicate an unsafe distance.

The product-based danger criterion is then computed as a product of these contributing factors.

$$DC_{prod} = f_{I_{prod}}(I_s) \cdot f_{CM_{prod}}(D_{CM}). \quad (4.8)$$

4.1.2 Goal and Obstacle Potential Fields

For the goal seeking and obstacle avoidance functions, the customary quadratic potential field functions are used [44, 50, 70]. The goal seeking function f_G is defined as:

$$f_G(D_G) = \frac{1}{2} D_G^2, \quad (4.9)$$

where, D_G is the distance between the end-effector and the goal.

The obstacle avoidance function f_O is defined as:

$$f_O(D_O) = \begin{cases} \frac{1}{2} \left(\frac{1}{D_O} - \frac{1}{D_{Omin}} \right)^2 & : D_O \leq D_{Omin} \\ 0 & : D_O > D_{Omin} \end{cases} \quad (4.10)$$

where, D_O is the distance between the robot and the nearest obstacle, D_{Omin} is the distance from the obstacle at which the obstacle begins to repel the robot (the influence distance). For the obstacle avoidance function, the distance between the robot and the nearest obstacle is taken as the distance between the closest point on the robot and the closest point on the obstacle. The distance between the robot and the nearest obstacle, as well as the distance between the robot and the non-interacting parts of the user are estimated using the hierarchy of spheres representation [77], illustrated in Figures 4.2 and 4.3. In this approach, the robot and the obstacles in the environment are described as a set of enveloping spheres. Initially, a small set of large enveloping spheres is used for each object. If no intersecting spheres are found, the distance between the two closest sphere centers is returned as the distance between the robot and the nearest obstacle or human. If two intersecting spheres are found, the robot and the obstacles are decomposed into a set of smaller enveloping spheres. The process is repeated until a non-intersecting set of spheres is found, or until the maximum level of decomposition is reached, in which case the algorithm reports that a collision has been detected. The level of

decomposition required to find a collision free set of spheres is also used to determine the size of the region within which local trajectory planning may be executed, as in [50].

When defining the enveloping spheres for the user, the current robot task also becomes important. If the goal of the interaction is for the robot to approach and/or contact the user, then it is not appropriate to represent the user simply as an obstacle, as in [21]. Instead, in this work, during pre-planning, each segment of the path is classified as interactive or non-interactive. If the segment is classified as non-interactive, the entire region of space occupied by the user is treated as an obstacle. If the segment is classified as interactive, a smaller set of spheres is used, such that the target area of the user (for example, the hand) is excluded from the obstacle area. By only excluding the contact area, this approach ensures that the robot can reach the intended target, while motion is still restricted to non-target areas of the body. Using this representation also ensures that the robot will slow down as it approaches the target, due to the effect of nearby spheres, as described in Chapter 5. Figure 4.2 shows the robot and the user represented with enveloping spheres in a non-interactive task segment. Figure 4.3 shows the representation during an interactive task segment.

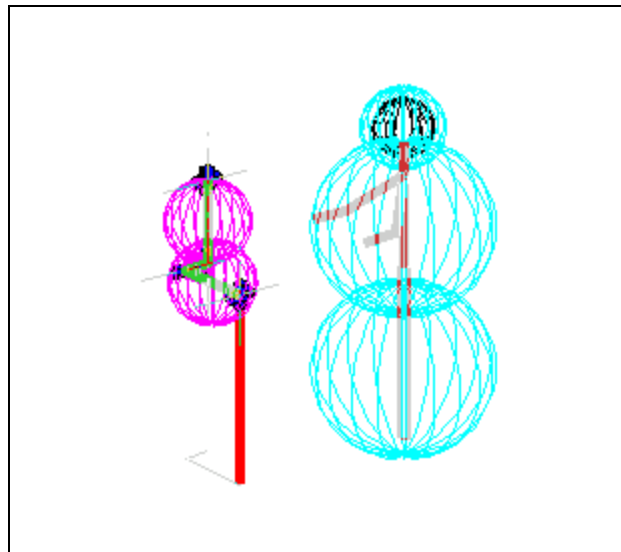


Figure 4.2. Human, robot representation in a non-interactive task.

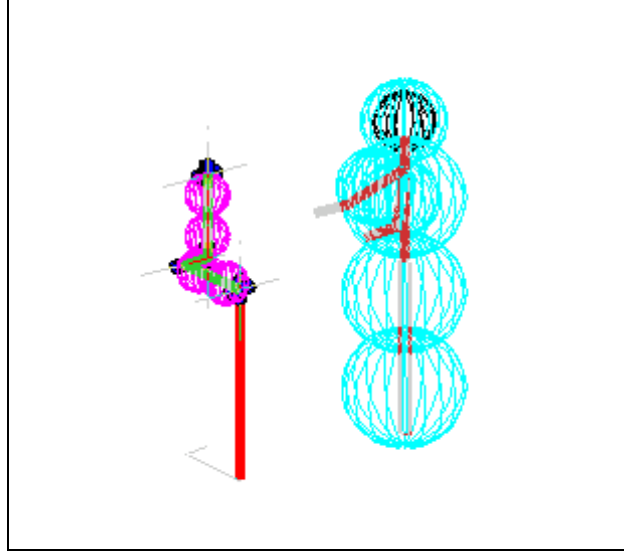


Figure 4.3. Human, robot representation in an interactive task.

4.1.3 The Overall Cost Function

The planning-cost function is generated by combining the goal seeking, obstacle avoidance, and danger criteria. The planned path is generated by searching for a set of configurations that minimize the cost function:

$$J = W_G \cdot f_G(D_G) + W_O \cdot f_O(D_O) + W_D \cdot K \cdot DC. \quad (4.10)$$

Here, W_G is the weighting of the goal seeking criterion, W_O is the weighting of the obstacle avoidance criterion, W_D is the weighting of the danger criterion, and K is a scaling factor. The selection of the weight levels is discussed in the following section.

4.2 Implementation

Using the above cost function, it is likely that the danger criterion will conflict with the goal seeking criteria during the search, leading to local minima and long search times. To avoid this problem, a two-stage search is proposed. In the first stage, maximum priority is placed on minimizing the danger criterion. A threshold is established for determining when an acceptable maximum level of danger is achieved. Once a path is found that places the robot below this

threshold, the second stage of the search is initiated. In this stage, maximum priority is placed on the goal-seeking criterion. In the resulting overall path, the robot will spend most of its time in low danger regions. One can note that, this approach will not result in the shortest distance path. The tradeoff between increased safety and reduced distance can be controlled by modifying the threshold where switching from the first stage to the second stage occurs. The two stages are implemented by modifying the weighting factors. In the first (danger minimization) stage, the danger weighting, W_D is greater than the goal seeking weighting, W_G , while in the second stage, W_G is greater than W_D . As long as the relative weights are set in this manner, the algorithm does not require tuning of the weight levels when using the product based danger criterion. For the sum based danger criterion, if the robot is approaching the person, W_D must be small (0.1 or less) in the second stage to avoid interference with the goal attraction criterion.

Even when the proposed two stage planning approach is used to minimize the conflict between the danger and goal seeking criteria, it is still possible for the goal seeking and the obstacle avoidance to conflict in a cluttered environment, or when joint limits are encountered during the search. The search time is also extended if the robot needs to reverse configurations during the path (for example, from an elbow down starting configuration to an elbow up final configuration). In these cases, a circuitous path is often generated, requiring some post-process smoothing [70]. In particular, if there are several obstacles positioned close to the robot, it may not be possible to complete the stage 1 search within the given threshold. In this case, the user should be notified that a safe path cannot be found in the current environment.

The problem of long search times can also be addressed by taking advantage of particular robot geometry, and searching only through joints that affect the end-effector position. For example, although the PUMA560 is a 6 DoF robot, only the first 3 joints contribute to the gross end-effector movement. After the position path is generated, the remaining 3 joints can be used to maintain a desired end-effector orientation, as required by the payload.

4.3 Search Strategy Improvements (Backwards Search)

The global planning strategy presented above is generally valid for non-redundant as well as redundant robots, as well as robots with either prismatic or articulated joints, or mobile robots. This is because the search is conducted forwards from an initial configuration, the search steps are generated from that initial configuration, and therefore only forward kinematics are required to calculate the workspace potential field functions.

If an inverse kinematics routine is available for the robot, the algorithm search time can be improved by adding a backwards search stage. This addition is useful in those cases when the robot goal is in a crowded area, for example when the robot's goal is the user's hand. In this case, to get to the goal, the robot must go into an area of higher potential field, since the goal is surrounded by obstacles generating a repulsive field. Therefore, the algorithm must perform "well-filling" to find the path, increasing the search time, and, potentially, resulting in a convoluted final path. On the other hand, if the search can be performed backwards, gradient descent can be used to find the lowest potential path to the goal, reducing the search time.

In general, it is always more efficient to search from the cluttered end of the path [78, 79]. The inverse kinematics routine is used to generate the goal configuration, given the goal workspace position and the desired end-effector orientation at the goal. The search is then initiated backwards from the goal configuration towards the start configuration. Once obstacle influence is minimal, the backwards search stops, and the forward search (as described in Section 4.2) is initiated, with the last configuration of the backwards generated path as the goal. This location at which the backwards search stops, and the new goal location for the forwards search is named the intermediate location (IL). If there are multiple solutions to the inverse kinematics problem, the danger criterion at each solution is evaluated, and the solution with the lowest danger criterion is selected.

For continuity, the algorithm must also ensure that the forwards and backwards-generated paths meet at the same point in configuration space. Since the goal and obstacle potential fields are defined

in the workspace, it is possible for an articulated robot to reach the starting point of the backwards path in an incorrect posture (e.g., elbow up vs. elbow down). In this case, the two paths cannot be joined by simply generating a spline between the two postures. This could cause the robot to move into obstacles or move through a more dangerous configuration. Instead, during the initial stage of the forward search, an additional “posture” potential function [50] is added, that favors the starting posture of the backwards path. The posture function is defined as:

$$f_{pos}(q) = \begin{cases} \frac{1}{2}(q - quad_{\min})^2 : & q < quad_{\min} \\ 0 : & quad_{\min} \leq q \leq quad_{\max} \\ \frac{1}{2}(quad_{\max} - q)^2 : & q > quad_{\max} \end{cases} \quad (4.11)$$

Here, q is the search configuration joint angle, and $quad_{\min}$ and $quad_{\max}$ are the quadrant boundaries centered around the each joint angle found in the final configuration of the backwards search (i.e., the joint angles associated with the IL).

The posture function is calculated for each joint; the total posture function is the sum over all the joints. The posture potential is only active while the robot is in the incorrect posture. When the robot reaches the correct posture, the posture potential becomes inactive. To ensure that the correct posture is reached prior to merging with the backwards planned path, an additional condition is added to the switchover from the first to the second stage of the planning. Namely, in addition to the low danger index requirement, the posture potential must also be zero. A flow chart for the complete algorithm, including the backwards search, is shown in Figure 4.4.

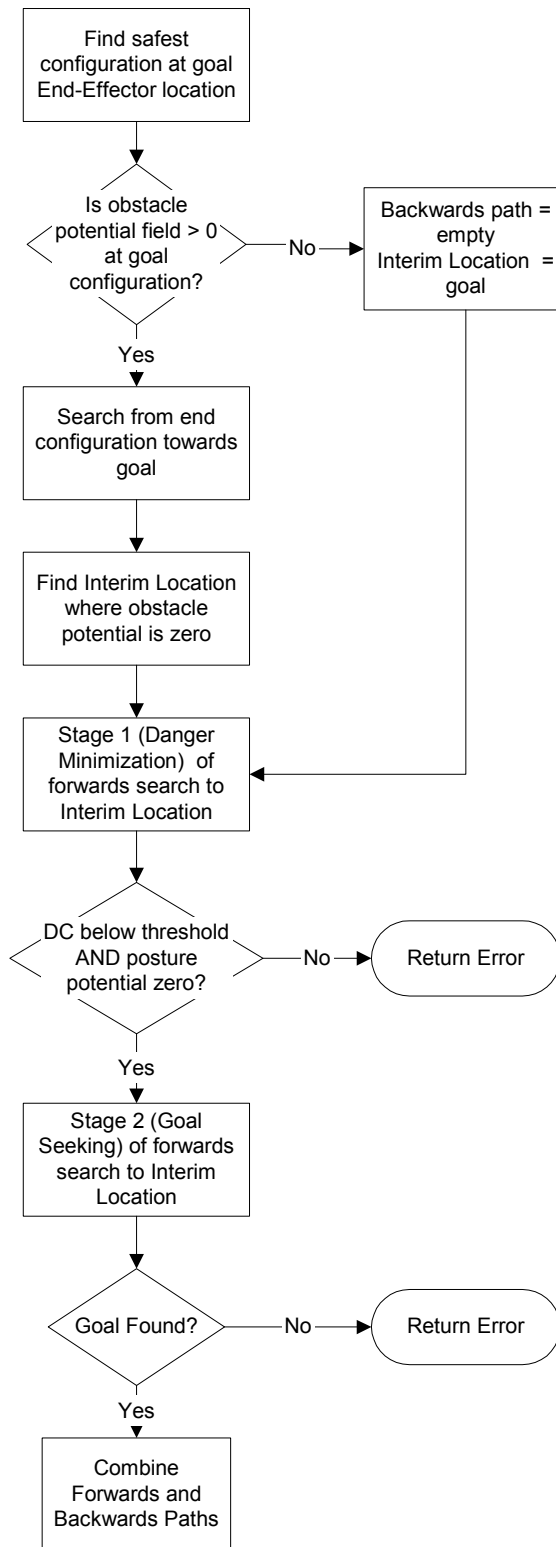


Figure 4.4. Combined backwards-forwards search algorithm flowchart.

4.4 Simulations

A simulation environment was developed to test the planning algorithms with various robot architectures. The robots are modeled using the Robotics Toolbox [24]. Figure 4.5 shows the planned motion of a 3 link planar robot using the basic algorithm (i.e., without the backwards search), with the sum-based danger criterion. The robot's goal is to pick up the object being held by the user. The same task is shown as planned by the product-based danger criterion in Figure 4.6. In both cases, to better illustrate the effect of the danger criterion, only the goal and danger criterion cost functions are included. The cost function weights used for both plans are given in Table 4.1. Figure 4.7 shows a comparison between the user-robot center of mass distance and the robot inertia for the sum-based and the product-based danger criteria.

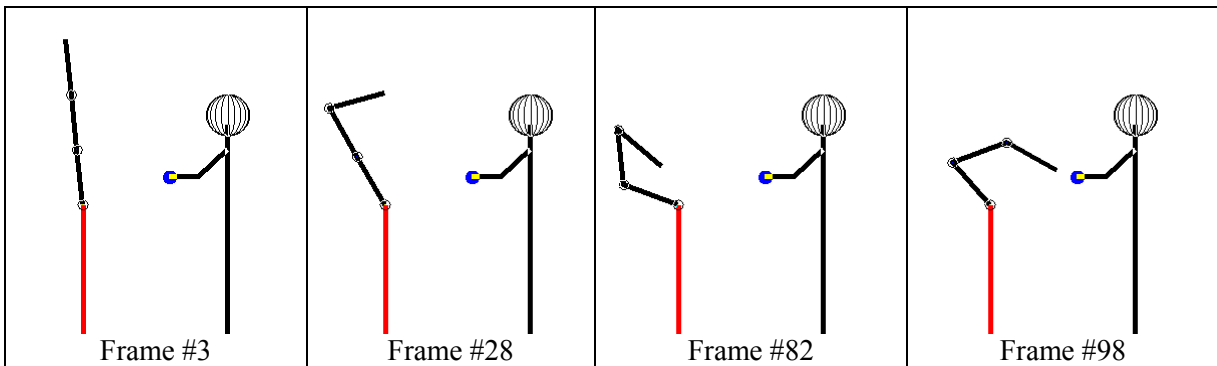


Figure 4.5. Planned path with sum-based danger criterion.

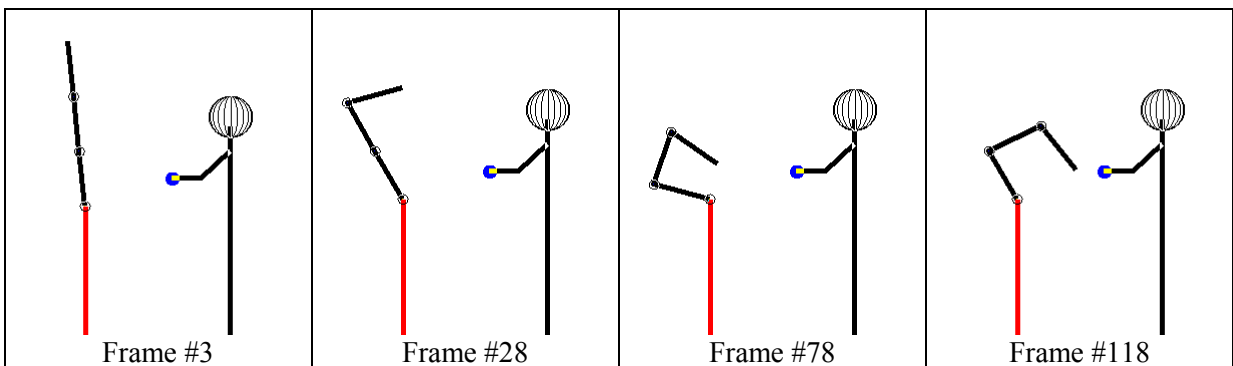


Figure 4.6. Planned path with product-based danger criterion.

Table 4.1. Planar robot simulations weights.

	W_G	W_O	W_D
Stage 1	0.2	0	0.8
Stage 2	0.9	0	0.1

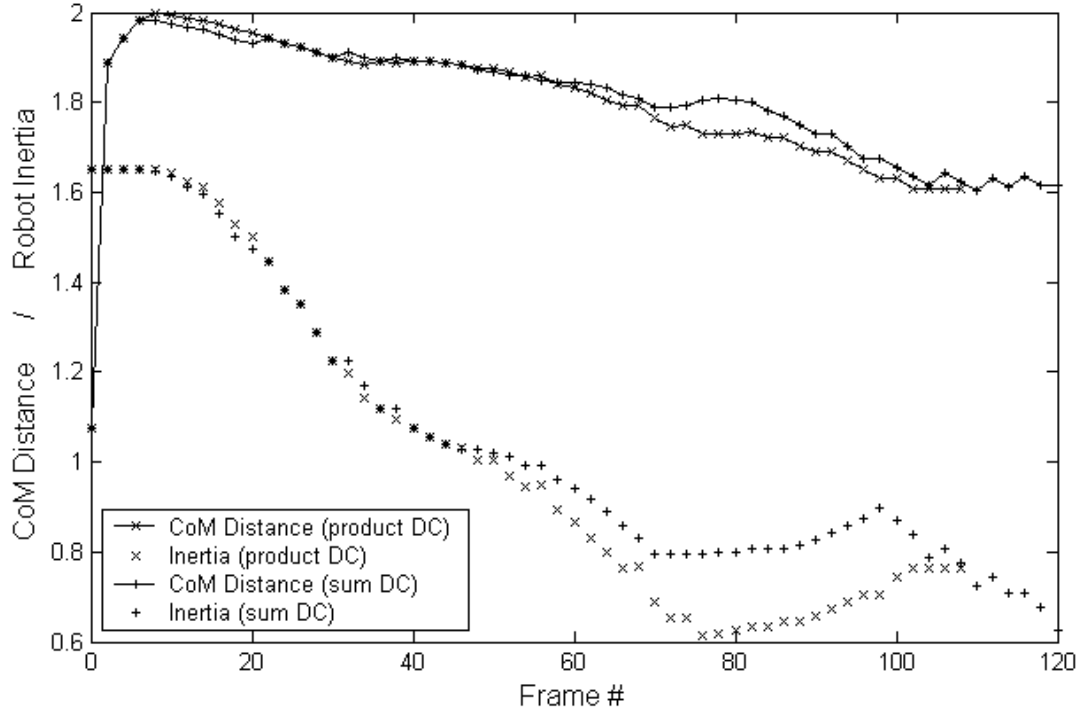


Figure 4.7. Comparison between the sum-based and product-based danger criteria.

Figure 4.5 and Figure 4.6 illustrate the differences between the two danger criterion formulations. The sum-based danger criterion implies that the factors affecting the danger can be considered separately when assessing the level of danger. One advantage of the sum-based criterion is that the formulation is similar to other quadratic cost functions normally used in the potential field approach, and is distance based. Therefore, the sum-based criterion does not need to be scaled when combined with the other criteria (i.e., $K = 1$). The center of mass distance factor is a repulsive potential field, and can, therefore, become infinite in magnitude when the center of mass distance between the robot and the user (D_{CM}) is close to the minimum safe distance (D_{min}). Thus, when the robot and the user are close together, the distance factor will dominate over the inertia factor. This effect is illustrated in the last frame of the Figure 4.5 sequence. As a result, near the point of interaction, the sum based

criterion results in a higher inertia, as can be seen in Figure 4.7. In general, the disadvantage of such a sum based formulation is that one of the factors always tends to dominate the others. Furthermore, for the sum-based criterion it is difficult to define the threshold at which one should switch from the danger minimization stage to the goal seeking stage, since the danger criterion is a combination of the robot link distances from the robot base and the distance from the robot to the person.

The product-based danger criterion implies that the factors affecting the danger criterion need to be considered collectively when assessing the level of danger. For example, if the distance between the robot and the person is large, the other contributing factors will not be minimized either. In Figure 4.6, since the distance between the robot and the person is small, both the distance factor and the inertia factor are minimized. In addition, when both factors have significant magnitude, the danger criterion gradient will be steepest, ensuring that the danger criterion is prioritized over the other criteria. Because the two factors scale each other, both are minimized to achieve the required safety level. Another advantage of the product-based criterion is that the criterion represents a clear indication of the level of danger, ranging from 0 to 1 (values greater than 1 are possible when the distance between the robot and the user (D_{CM}) is smaller than the minimum safe distance (D_{CMmin})). Therefore, it is easier to specify the switch threshold as the desired level of danger. However, for the product-based criterion, a scaling factor (K) must be chosen so that the danger criterion is on the same scale as the goal and obstacle criteria.

In the majority of cases, the product-based danger criterion is more suitable. The product-based criterion is more suitable for redundant robots, where both the inertia and Centre of Mass (CoM) distance factors can be minimized, regardless of the CoM distance. When the robot is close to the person, the product-based danger criterion will decrease inertia and increase CoM distance. On the other hand, close to the person, the sum-based danger criterion becomes dominated by the distance factor, so inertia is not reduced as significantly. The sum-based danger criterion may be more suitable with large, under-articulated robots. In this case, the difference between the maximum and

minimum robot inertia may not be very significant, whereas the strong CoM distance action will ensure that the robot does not get too close to the user.

Figure 4.8 shows a planned motion sequence with a PUMA 560 robot, using the basic algorithm with the goal, obstacle and danger criteria. The product based danger criterion is used. Table 4.2 gives the weights used in the search. For comparison, a path was generated using the best-first planner without any danger criterion. As illustrated in Figure 4.9, the danger criterion pushes the CoM of the robot away from the person along the majority of the path, as well as significantly reducing the robot's inertia.

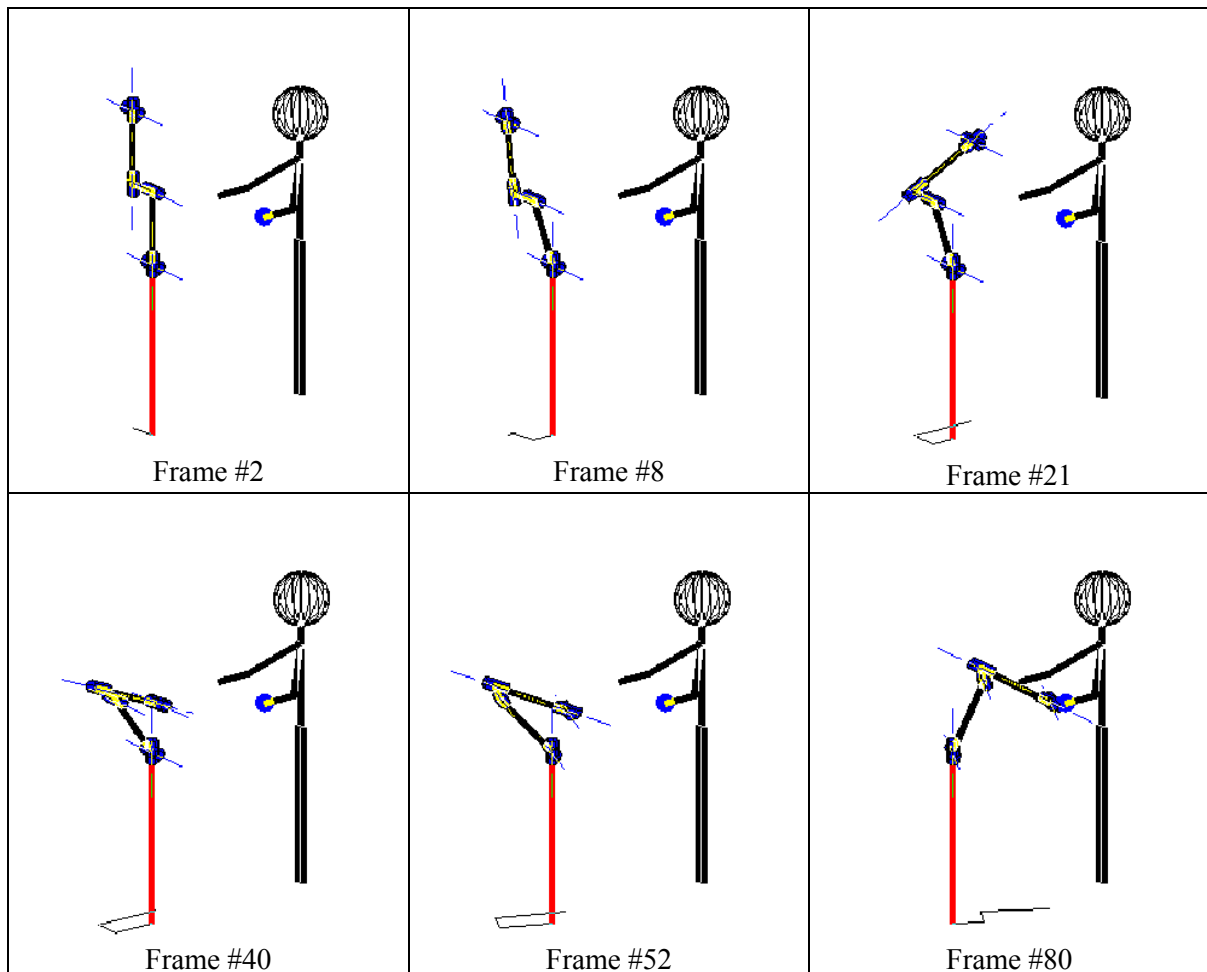


Figure 4.8. Planned sequence for a PUMA560 robot (product danger criterion).

Table 4.2. PUMA560 simulations weights.

	W_G	W_O	W_D
Stage 1	0.1	0.2	0.7
Stage 2	0.7	0.2	0.1

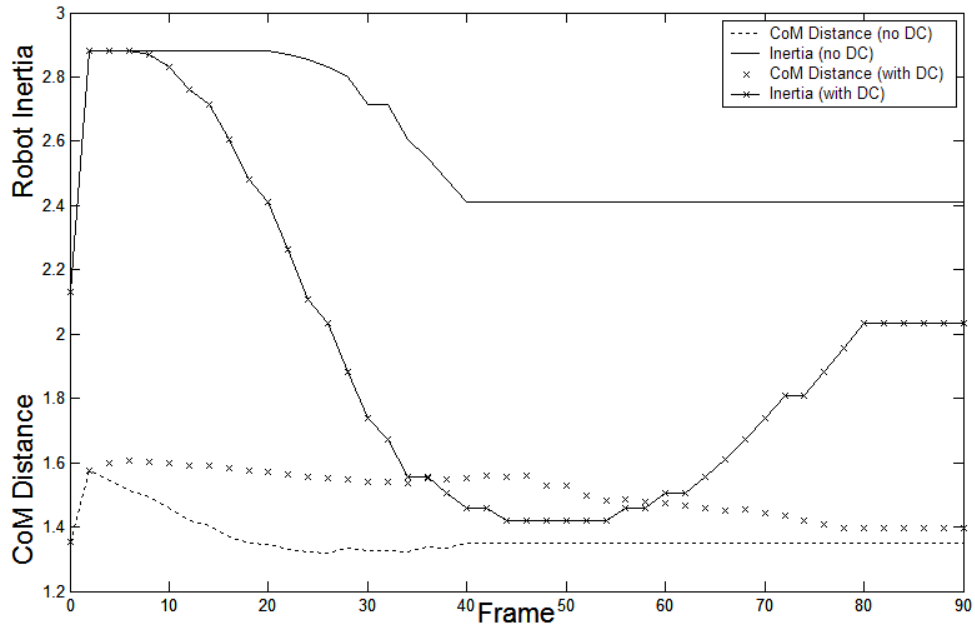


Figure 4.9. Effect of the danger criterion search on the danger factors (product danger criterion).

Figure 4.10 shows a planned sequence using the modified algorithm, with the backwards search added. In this case, the initial robot pose is the reverse of the required final pose generated by the backwards plan. The same weights were used as for the basic algorithm, as specified in Table 4.2.

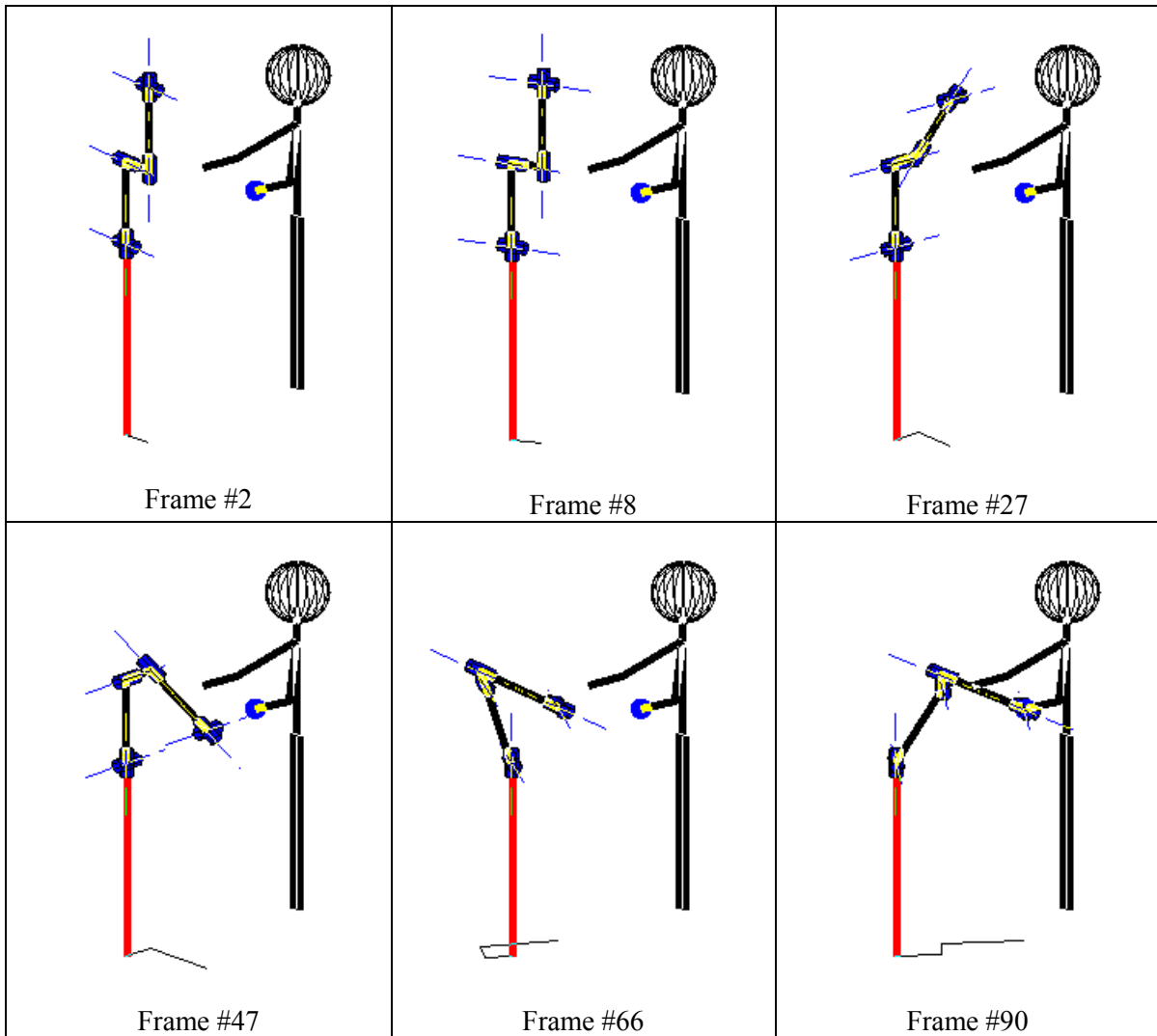


Figure 4.10. Planned sequence for a PUMA560 robot with backwards search (product danger criterion).

Initially, while the danger is low, the posture function dominates the potential field, and the robot moves first to move to the correct posture. As the robot comes closer to the person, the danger criterion begins to dominate the potential field, and the robot inertia is reduced. Once the danger index has been reduced, the robot moves towards the goal. Posture correction is performed during low danger sections of the path. As discussed in Section 4.3 and shown in the flowchart in Figure 4.4, the backwards search is only performed when the goal location is within the influence distance of obstacles. In this case, the basic planner must find the entrance into an obstacle region. Using the

backwards search, finding a path from the obstacle enclosed goal location to a free region is much easier [79]. Once a configuration free from obstacle influence is found through the backwards search, the forwards search, incorporating the danger criterion, is initiated to this configuration. The posture potential must then be added to the forwards search cost function to ensure that the forwards and backwards paths join at the same robot configuration. This allows the planner goal and obstacle fields to be defined in the workspace, while still ensuring a contiguous path in the joint space.

4.5 Summary

The proposed safe planner reduces the factors affecting danger along the path. Using the two-stage planning approach reduces the depth of local minima in the cost function, allowing the planner to execute quickly. Minimizing the danger criterion during the planning stage ensures that the robot is in a low inertia configuration in the case of an unanticipated collision, as well as reducing the chance of a collision by distancing the robot centre of mass from the user. This advance-planning approach puts the robot in a better position to deal with real-time safety hazards.

When an inverse kinematics routine is available for an articulated robot, the performance of the planner can be further improved by adding a backwards search. That is, the path is generated backwards from the goal when the goal location is in an area crowded by obstacles. To ensure that the forwards and backwards generated paths meet, a posture potential is added to the total cost function. By including the posture potential directly into the cost function, rather than splining the two paths after they are generated, the algorithm ensures that posture correction occurs during low-danger sections of the path.

Chapter 5: On-line Trajectory Planning and Control

Once the path for the robot motion is generated, as discussed in Chapter 4, the velocity profile along the trajectory is generated on-line, in response to real-time conditions. In addition, if events unanticipated during the planning stage occur during the execution of the plan, reactive real-time planning is needed to move the robot to avoid an unanticipated collision. These two functions are performed by the trajectory planning module and the real-time safety module, respectively. The time scaleable trajectory module is overviewed in the first section. The real-time safety module, which modulates the trajectory planner, and more specifically, the real-time danger index and its stability, are discussed in the second section. Simulations and experimental work using the online trajectory planning and control system with the experimental setup described in Chapter 3 are presented to demonstrate the behavior of the system.

5.1 Overview of the Trajectory Planning Module

The trajectory planning module generates the velocity and acceleration profile to be followed along the path generated by the path planner. The problem of trajectory planning is that of matching the end conditions for a set of coordinates over a series of path segments, while respecting the kinematic limits for each coordinate (i.e., robot joint axis). The details of how these limits are met for each coordinate simultaneously is described in detail in Appendix A. In this case, as is typical in robotics, the trajectories are generated to maximize the robot velocity along the path without violating kinematic limits. The trajectory is described in terms of parameterized time t .

Once the trajectory is generated for the maximum velocity trajectory, time scaling is applied during path execution to adjust the trajectory to meet interaction constraints, such as changes to the environment. Time scaling is performed by controlling the rate at which r increases (i.e., the rate at which the robot advances along the path). The rate of change of r can be modified during run-time at each execution step, according to (5.1).

$$r_i = r_{i-1} + c \cdot dt, \quad (5.1)$$

where c is the desired rate of change, with $c = 1$ indicates full speed, and $c = 0$ indicates full stop. dt is the magnitude of the time step.

The trajectory is described by a series of cubic and quintic polynomials, as described in Appendix A. The trajectory planner is implemented such that the time scale along the trajectory can be modified at any time, according to (5.1). At the same time, the trajectory planner must ensure that during changes to c , the kinematic constraints (max/min velocity and acceleration) are respected, and that the acceleration stays continuous through the velocity changes. For this reason, changes to the time scale variable c are low-pass filtered with a second order filter. The cut-off frequency of the low-pass filter is selected such that the maximum acceleration (after a step change to c) is equal to the maximum robot acceleration.

5.1.1 Velocity Scaling Example

A sample joint trajectory is used to demonstrate the behavior of the velocity scaling. Figure 5.1 shows the position, velocity and acceleration of a sample multi-axis trajectory. In this case, no time scaling is applied, and the path is executed at maximum velocity and acceleration.

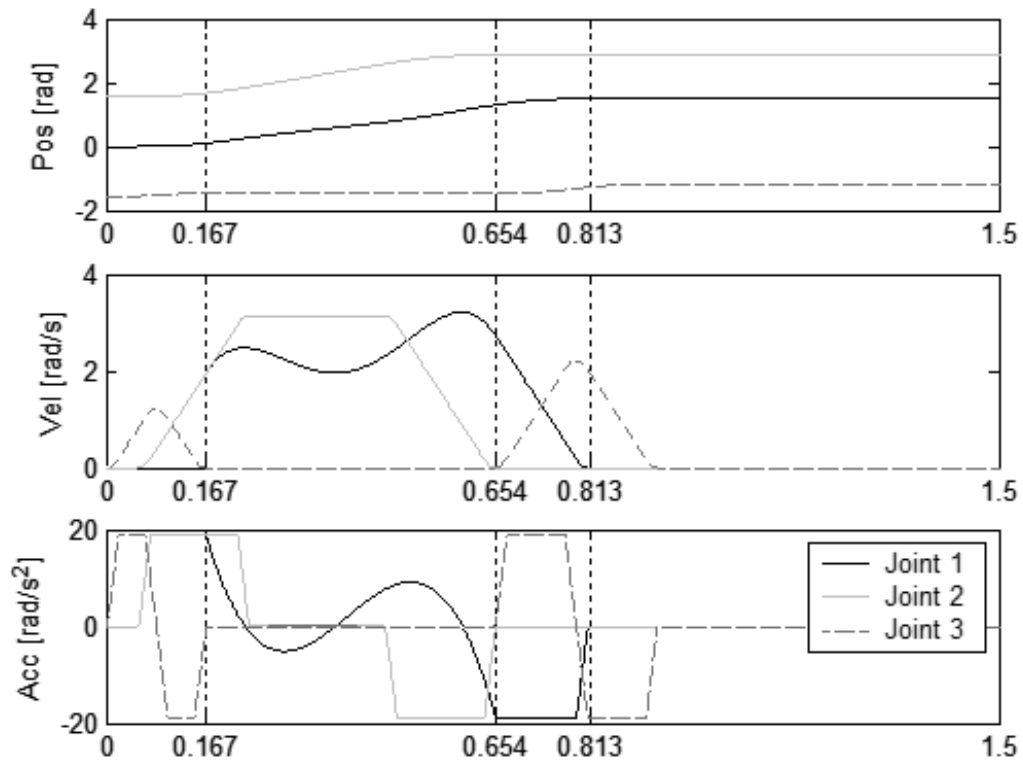


Figure 5.1. Sample multi-axis path.

Figure 5.2 shows the same motion as Figure 5.1, with time scaling applied. The time scale factor c is generated by a square wave changing from 0 to 1 (i.e., stop motion to full speed), which represents the worst-case scenario. The resulting scaled time r and the filtered value of r is shown in the bottom axis of Figure 5.2. As can be seen in the figure, the velocity and acceleration for the motion remain continuous and below the kinematic limits of the robot.

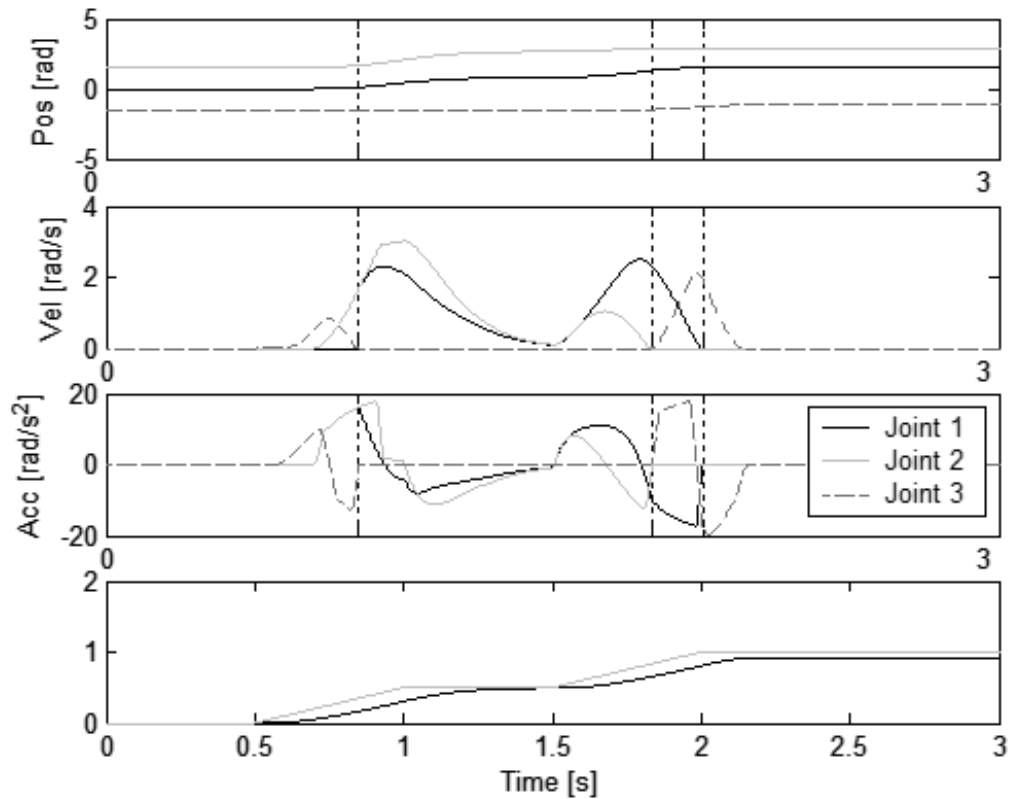


Figure 5.2. Sample multi-axis motion with time scaling.

5.2 Real Time Safety Module⁶

The safety control module evaluates the safety of the plan generated by the trajectory planner at each control step. If a change in the environment is detected that threatens the safety of the interaction, the safety control module initiates a deviation from the planned path. This deviation will move the robot to a safer location.

5.2.1 Safety Module Algorithm

The safety module is responsible for reacting to sudden changes in the environment, not anticipated during the planning stage. The inputs into the safety module are: the proposed next

⁶ A version of this section has been accepted for publication. D. Kulić and E. Croft, "Safety Based Control Strategy for Human-Robot Interaction," *Journal of Robotics and Autonomous Systems*, 2005.

configuration of the robot as provided by the trajectory planner, including the velocity and acceleration information; the current user configuration; and estimates of the user's reactions (as discussed in Chapters 6 and 7). Based on this information, the safety module evaluates the current level of danger of the interaction. If the level of danger is low, the proposed plan can proceed, otherwise, a corrective decision is made and an alternate trajectory is generated and passed to the low-level controller.

The alternate trajectory generated by the safety module lowers the estimated danger present in the interaction. This approach is most similar to the impedance type strategies presented in [44, 80]. A virtual force, calculated by the safety module, pushes the robot away from the person or obstacle. The safety module performs as a one-step-ahead local planner in real-time, moving the robot to a location of lowest danger. The virtual force generated is based on a set of factors affecting the potential impact force during a collision, such as the relative distance and velocity between the robot and the obstacle [80] and the effective impedance at the impact point [44]. However, unlike [44, 80], the algorithm executes in joint space, and is therefore able to perform safely and accurately at all points in the workspace, including any robot singularities.

5.2.1.1 Danger Index Formulation

The key element of the safety module is the estimation of the level of danger, i.e., the *danger index*. Unlike the “static” danger criterion utilized in Chapter 4, the danger index is a dynamic quantity. Thus, the design and evaluation of the danger index must consider the effect of the index on the evolving robot trajectory, in particular, its effect on the stability of the motion.

The danger index is constructed from measures that have an effect on the potential impact force during a collision. As suggested by Ikuta et al. [25], the danger index should include the distance between the robot and the human, the relative velocity between them, as well as the inertia and stiffness of the robot. Since the robot is an articulated linkage of bodies, for the real-time control application in this work it is not sufficient to consider only the end-effector as done in [25]; instead

the entire robot body must be considered as a potential source of impact. For each link, the point closest to the person is considered; this point is called a *critical point*. The danger index is estimated for each critical point. The factors included are the distance between the robot and the person at the critical point being considered, their relative velocity, and the effective inertia at the critical point. As discussed in Chapter 4, the stiffness can be more effectively lowered through mechanical design [25, 30].

The distance factor f_D is given in equation (5.2),

$$f_D(s) = \begin{cases} k_D \left(\frac{1}{s} - \frac{1}{D_{\max}} \right)^2 & : s \leq D_{\max} \\ 0 & : s > D_{\max} \end{cases}, \quad (5.2)$$

where, s is the distance from the critical point to the nearest point on the person. The scaling constant k_D is used to scale the distance factor function such that the value of the function is zero when the distance between the human and the robot is large enough (larger than D_{\max}), and is one when the distance between the human and the robot is the minimum allowable distance (D_{\min}).

$$k_D = \left(\frac{D_{\min} D_{\max}}{D_{\min} - D_{\max}} \right)^2 \quad (5.3)$$

Values of the distance factor above one indicate an unsafe distance.

The velocity factor f_V is based on the magnitude of the velocity component, v , between the critical point and the nearest point on the person along the line joining these two points (the approach velocity). The approach velocity, v , is defined to be positive when the robot and the human are moving towards each other.

$$f_V(v) = \begin{cases} k_V (v - V_{\min})^2 & : v \geq V_{\min} \\ 0 & : v < V_{\min} \end{cases}. \quad (5.4)$$

The scaling constant, k_V , is used to scale the velocity factor function such that the value of the function is zero when the velocity is lower than V_{\min} and one when the velocity is V_{\max} . V_{\min} is set to a negative value (i.e., f_V is zero when the robot is moving away from the person).

$$k_V = \left(\frac{1}{V_{\max} - V_{\min}} \right)^2. \quad (5.5)$$

Values of the velocity factor above one indicate an unsafe approach velocity.

The inertia factor is defined as:

$$f_I(I_{CP}) = \frac{I_{CP}}{I_{\max}}, \quad (5.6)$$

where, I_{CP} is the effective inertia at the critical point, and I_{\max} is the maximum safe value of the robot inertia.

The danger index is then the product of the distance, velocity and inertia factors.

$$DI = f_D \cdot f_V \cdot f_I. \quad (5.7)$$

The use of a product of factors rather than a sum formulates the danger index as a combination of dependent factors [81]. As a result, the danger index is zero through most of the workspace, and is only non-zero when all the conditions for a potential impact are present: namely, small distance, positive relative velocity with respect to the person and high robot inertia. This formulation avoids false positives that would be present if a sum of factors was used. Since the danger index is used to activate evasive action, it is important to avoid false positives, which would induce unnecessary evasive action.

Once the danger index is calculated, it is used to generate the virtual force to push the robot away from the human as in [44, 80].

5.2.1.2 A one-dimensional example

To visualize the action of the danger index, it is helpful to first consider the one-dimensional case. Consider a point robot acting moving along a line. Three scenarios are possible: (i) there are no (human) obstacles on either side the robot, (ii) there is an obstacle on one side of the robot, or (iii) there is an obstacle on both sides of the robot, as shown in Figure 5.3.

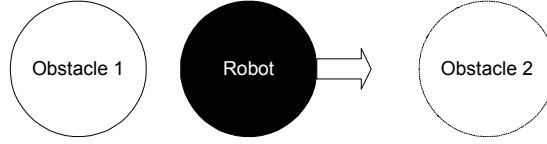


Figure 5.3. Point robot moving between obstacles (one-dimensional case).

Since the safety module always acts to decrease the danger at the highest critical point, only the nearest obstacles are considered; additional obstacles further away from the robot do not affect module behavior. If there are no obstacles on either side of the robot, the robot proceeds as planned, as the danger index is zero. If the obstacle is on one side of the robot only, and the danger index is non-zero, a virtual force (F_{SO}) pushing the robot in the direction away from the obstacle is generated, proportional to the danger index.

$$F_{SO} = Km \cdot DI(s, v). \quad (5.8)$$

This is analogous to a virtual impedance between the robot and the obstacle, similar to [80]. However, unlike [80], the impedance is non-linear, resulting in a stronger evasive action as the danger increases. Figure 5.4 shows a comparison between the linear impedance (LI) and the proposed danger index (DI) method for the single degree of freedom system. The robot initial configuration is at 0.2m away from the obstacle, with a velocity of 1m/s towards the obstacle. Close to the obstacle, when the danger index is high, the non-linear impedance results in a faster reaction to move away from the obstacle.

If obstacles are present on both sides of the robot, two virtual impedances are present, one between each obstacle and the robot. In this case, the resultant force (F_{TO}) is the difference between the two impedances, based on the danger index calculated with respect to each obstacle.

$$F_{TO} = Km \cdot [DI(s_1, v_1) - DI(s_2, v_2)]. \quad (5.9)$$

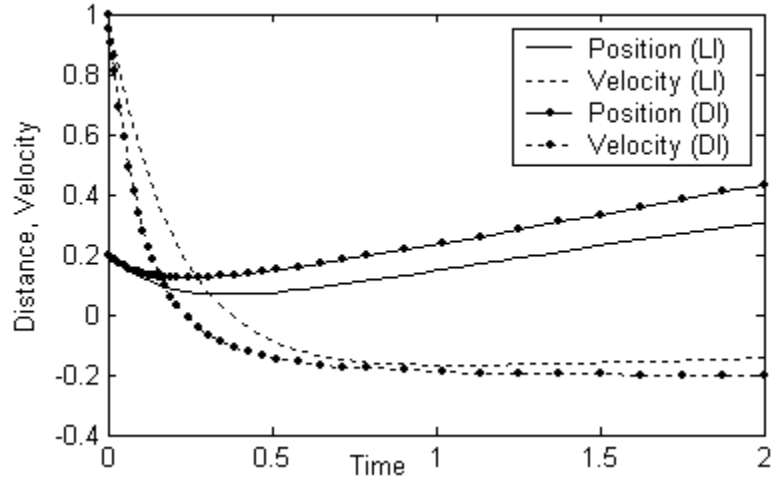


Figure 5.4. Comparison between linear impedance and the danger index for a 1 DoF system.

To illustrate the behavior of the two-obstacle case, consider the simple case when both obstacles are stationary and the robot is moving between them, as shown in Figure 5.3. In this case, $v_1 = -v_2$.

The system is then characterized by a second order differential equation:

$$\ddot{s} = \frac{1}{m} \cdot F_{TO}, \quad (5.10)$$

where \ddot{s} is the acceleration and m is the mass of the point robot.

Figure 5.5 shows the phase portrait of the system, and Figure 5.6 shows a sample time trajectory. The system is stable about the equilibrium point $s = 0.5$, $v = 0$, at the midpoint between the two obstacles. However, care must be chosen when selecting the danger index parameters, to avoid oscillatory behavior about the equilibrium point. Parameter selection is discussed in more detail in the following section and in Section 5.2.1.6.

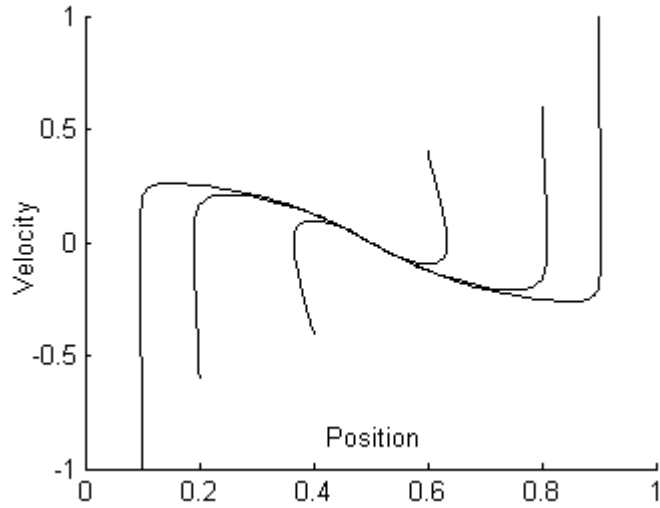


Figure 5.5. Phase portrait of the two obstacle case (1 DoF).

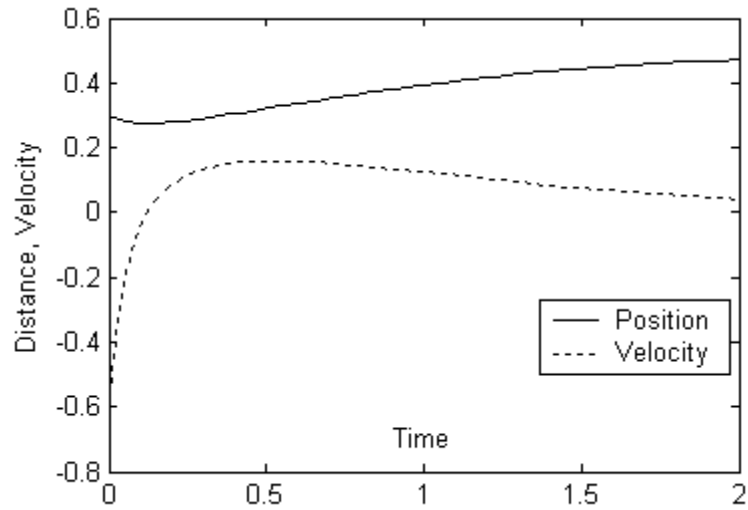


Figure 5.6. Sample time trajectory for the two obstacle case (1 DoF).

5.2.1.3 Stability Analysis

For the case where two obstacles are present on opposite sides of the robot, one can consider the case where the two obstacles are stationary. Let the distance between the two obstacles be equal to $2d$. Let x_1 be the distance away from the midpoint between the two obstacles, and x_2 the velocity.

Then, assuming that $D_{max} > 2d$, the system will be described by the following set of differential equations:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 = F = \frac{1}{m} &\begin{cases} k_D \cdot \left(\frac{1}{d+x_1} - \frac{1}{D_{max}} \right)^2 \cdot k_V \cdot (x_2 + V_{min})^2 & : x_2 \leq V_{min} \\ k_D \cdot \left(\frac{1}{d+x_1} - \frac{1}{D_{max}} \right)^2 \cdot k_V \cdot (x_2 + V_{min})^2 & : V_{min} < x_2 < -V_{min} \\ -k_D \cdot \left(\frac{1}{d-x_1} - \frac{1}{D_{max}} \right)^2 \cdot k_V \cdot (x_2 - V_{min})^2 & \\ -k_D \cdot \left(\frac{1}{d-x_1} - \frac{1}{D_{max}} \right)^2 \cdot k_V \cdot (x_2 - V_{min})^2 & x_2 \geq -V_{min} \end{cases} \end{aligned} \quad (5.11)$$

This system has an equilibrium point at $x_1 = x_2 = 0$. Distance measures, i.e., d , D_{max} and D_{min} are always positive. The coefficients k_D and k_V are always positive. The mass m is always positive.

Using Lyapunov's first method, one can show that the system is stable at the origin. The Jacobian matrix at the origin of the differential Equation (5.10) is given by:

$$A = \begin{bmatrix} 0 & 1 \\ -\frac{4}{m} \cdot k_D \cdot k_V \cdot \frac{V_{min}^2}{d^2} \cdot \left(\frac{1}{d} - \frac{1}{D_{max}} \right) & \frac{4}{m} \cdot k_D \cdot k_V \cdot V_{min} \cdot \left(\frac{1}{d} - \frac{1}{D_{max}} \right)^2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \quad (5.12)$$

The eigenvalues of A are:

$$\lambda_{1,2} = \frac{1}{2} a_{22} \pm \frac{1}{2} \sqrt{4a_{21} + a_{22}^2} . \quad (5.13)$$

The origin will be stable if the real part of both eigenvalues is negative; i.e., if a_{22} and a_{21} are both negative. Since k_D , k_V and m are always positive, and $d > D_{max}$, a_{21} is, by inspection, negative, and for a_{22} negative, V_{min} must be less than zero.

To examine the extent of the stability region defined by this method, the Lyapunov function corresponding to the linear estimation is given by:

$$V_1 = x^T P x, \quad (5.14)$$

where the matrix P is positive definite and is the solution to

$$PA + A^T P = -Q, \quad (5.15)$$

where Q is any positive definite matrix. Taking Q as the identity matrix, and substituting Equation (5.12) into Equation (5.15), the matrix P is given by:

$$P = \begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix} = \begin{bmatrix} \frac{a_{22} + a_{21} - 1}{2a_{21}} + \frac{1}{2a_{22}} & -\frac{1}{2a_{21}} \\ -\frac{1}{2a_{21}} & \frac{1 - a_{21}}{2a_{21}a_{22}} \end{bmatrix}. \quad (5.16)$$

Since a_{21} and a_{22} are both negative, p_{11} , p_{12} and p_{22} are all positive.

The derivative of this Lyapunov function along the trajectories of Equation (5.11) is given by

$$\dot{V}_1 = 2p_{11}x_1x_2 + 2p_{12}x_2^2 + 2(p_{12}x_1 + p_{22}x_2)F. \quad (5.17)$$

\dot{V}_1 will be negative in the neighborhood of the origin, however, the second term is always positive and proportional to x_2^2 , making \dot{V}_1 positive when the robot velocity is large. In addition, the last term becomes large and positive when x_2 is small and x_1 is close to an obstacle, i.e., $x_1 \rightarrow -d$, and $p_{12}x_1 < p_{22}x_2$. A different Lyapunov function is needed to show stability throughout the state space.

To find a function covering the region where repulsive forces from both obstacles affect the robot (i.e., $-V_{min} < x_2 < V_{min}$), a non-quadratic Lyapunov function is required. To find the structure required, the structure of the forcing function F is analyzed. For the case where $|x_2| \leq |V_{min}|$, the force, F , defined in Equation (5.11) can be re-written as follows:

$$F = -(V_{min}^2 + x_2^2)H(x_1) + 2V_{min}x_2R(x_1), \quad (5.18)$$

where,

$$H(x_1) = -\frac{k_D k_V}{m} \left(\frac{1}{(d+x_1)^2} - \frac{2}{D_{max}(d+x_1)} - \frac{1}{(d-x_1)^2} + \frac{2}{D_{max}(d-x_1)} \right), \quad (5.19)$$

and

$$R(x_1) = \frac{k_D k_V}{m} \left\{ \left(\frac{1}{d + x_1} - \frac{1}{D_{\max}} \right)^2 + \left(\frac{1}{d - x_1} - \frac{1}{D_{\max}} \right)^2 \right\}. \quad (5.20)$$

The forcing function, F , can be compared to a linear impedance controller, where the forcing function is of the form:

$$F_{linear} = \frac{1}{m} (-kx_1 - bx_2). \quad (5.21)$$

A plot of $H(x_1)$ is shown in Figure 5.7. $H(x_1)$ is comparable to a non-linear spring element, which becomes stiffer the more the spring is compressed. The forcing function, F , is analogous to a spring-damper system, where the spring becomes stiffer (i.e., provides greater resistance) closer to the obstacle. The spring is also scaled by the velocity term, which means that it becomes stiffer at higher approach velocities. This means that a slow approach trajectory can move closer to the human “obstacle” than a fast trajectory, as is desirable for human-robot interaction.

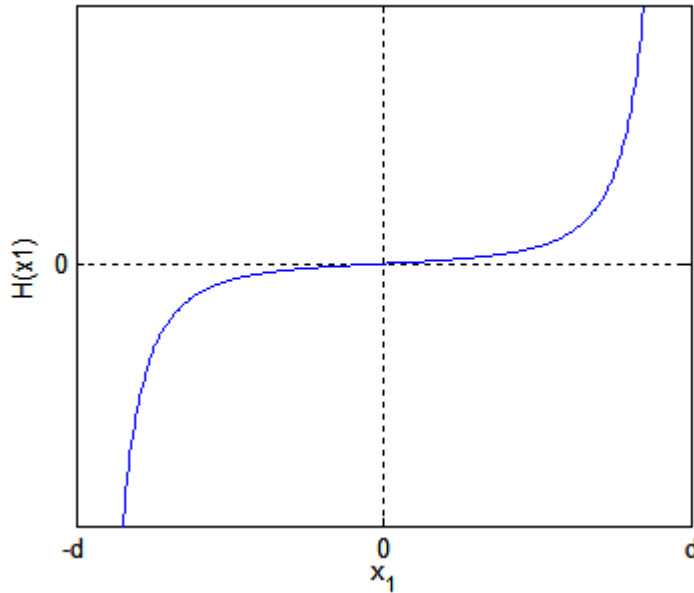


Figure 5.7. Plot of non-linear stiffness function $H(x_1)$.

The damping term $2V_{min}x_2R(x_1)$ is also analogous to the linear damping (recall that $V_{min} < 0$). The damping is scaled by the function $R(x_1)$. $R(x_1)$ is always positive and increases quadratically as the

robot position approaches the obstacle. Thus, the closer the robot approaches to the obstacle, the more damping is applied, as compared to a linear system where the applied damping is constant.

Using the variable gradient method [82], stability can be shown for a dynamic system with the forcing function, F , as defined in Equation (5.11). A Lyapunov function candidate is sought, where the derivative of the function has the form:

$$\dot{V}_2 = g_1(x_1, x_2) \cdot \dot{x}_1 + g_2(x_1, x_2) \cdot \dot{x}_2, \quad (5.22)$$

where, $g(x_1, x_2)$ is the gradient of a positive definite function V_2 .

A gradient is sought of the form [82]:

$$g(x) = \begin{bmatrix} \alpha(x)x_1 + \beta(x)x_2 \\ \gamma(x)x_1 + \delta(x)x_2 \end{bmatrix}. \quad (5.23)$$

Substituting Equations (5.23) and (5.18) into Equation (5.22), the Lyapunov function takes the following form:

$$\begin{aligned} \dot{V}_2 = & \alpha(x)x_1x_2 + \beta(x)x_2^2 - \gamma(x)(V_{\min}^2 + x_2^2)H(x_1)x_1 + 2\gamma(x)V_{\min}R(x_1)x_1x_2 \\ & - \delta(x)(V_{\min}^2 + x_2^2)H(x_1)x_2 + 2\delta(x)V_{\min}R(x)x_2^2 \end{aligned} \quad (5.24)$$

One can note that $H(x_1)x_1$ is always positive, so the third term in Equation (5.24) is always negative. Choosing $\alpha(x)$, $\beta(x)$, $\gamma(x)$ and $\delta(x)$ such that the coupled non-negative terms cancel out, i.e.,:

$$(\alpha(x)x_1 + 2\gamma(x)V_{\min}R(x_1)x_1 - \delta(x)(V_{\min}^2 + x_2^2)H(x_1))x_2 = 0, \quad (5.25)$$

results in:

$$\dot{V}_2 = (\beta(x) + 2\delta(x)V_{\min}R(x))x_2^2 - \gamma(x)(V_{\min}^2 + x_2^2)H(x_1)x_1. \quad (5.26)$$

Selecting

$$\delta(x) = \delta(x_2) = \frac{\delta_c}{V_{\min}^2 + x_2^2}, \quad (5.27)$$

where δ_c is a positive constant, and choosing $\beta(x)$ and $\gamma(x)$ as positive constants β_c and γ_c respectively, and substituting into Equation (5.25) yields:

$$\alpha(x) = \alpha(x_1) = \frac{\delta_c H(x_1) - 2\gamma_c V_{\min} R(x) x_1}{x_1}. \quad (5.28)$$

To ensure $g(x)$ is a gradient, requires:

$$\beta_c + \frac{\partial \alpha(x_1)}{\partial x_2} x_1 + \frac{\partial \beta_c}{\partial x_2} = \gamma_c + \frac{\partial \gamma_c}{\partial x_1} x_1 + \frac{\partial \delta(x_2)}{\partial x_1} x_2. \quad (5.29)$$

$$\beta_c = \gamma_c$$

The resulting gradient is:

$$g(x) = \begin{bmatrix} \delta_c H(x_1) - 2\gamma_c V_{\min} R(x_1) x_1 + \gamma_c x_2 \\ \gamma_c x_1 + \frac{\delta_c}{V_{\min}^2 + x_2^2} \end{bmatrix}. \quad (5.30)$$

The Lyapunov function with the desired gradient g is found by integrating along the axes:

$$\begin{aligned} V_2 &= \int_0^{x_1} g_1(y_1, 0) dy_1 + \int_0^{x_2} g_2(x_1, y_2) dy_2 \\ V_2 &= \int_0^{x_1} (\delta_c H(y_1) - 2\gamma_c V_{\min} R(y_1) y_1) dy_1 + \int_0^{x_2} \left(\gamma_c x_1 + \frac{\delta_c x_2}{V_{\min}^2 + x_2^2} \right) dy_2 \end{aligned} \quad (5.31)$$

$$V_2 = T_1(x_1) + T_2(x_1) + T_3(x_1, x_2) + T_4(x_2)$$

where,

$$\begin{aligned} T_1(x_1) &= \frac{k_D k_V \delta_c}{m} \left(\frac{1}{d+x_1} + \frac{1}{d-x_1} - \frac{2}{d} + \frac{2}{D_{\max}} \ln \left(\frac{d^2 - x_1^2}{d^2} \right) \right) \\ T_2(x_1) &= -\frac{2k_D k_V \gamma_c V_{\min}}{m} \left(\frac{x_1^2}{D_{\max}^2} + \frac{d}{d+x_1} + \frac{d}{d-x_1} - 2 + \left(1 + \frac{2d}{D_{\max}} \right) \ln \left(\frac{d^2 - x_1^2}{d^2} \right) \right). \quad (5.32) \\ T_3(x_1, x_2) &= \gamma_c x_1 x_2 \\ T_4(x_1) &= \frac{\delta_c}{2} \ln \left(\frac{V_{\min}^2 + x_2^2}{V_{\min}^2} \right) \end{aligned}$$

T_1 , T_2 and T_4 are always positive, while T_3 can be negative. The constants γ_c and δ_c can be chosen such that V_2 is always positive and \dot{V}_2 is always negative in the region $|x_1| \leq |d|$, $|x_2| \leq |V_{\min}|$. To ensure that \dot{V}_2 is always negative,

$$\gamma_c < -2\delta_c V_{\min} R(0), \quad (5.33)$$

since $R(x_1)$ has a minimum at 0,

$$R(0) = \frac{2k_D k_V}{m} \left(\frac{1}{d} - \frac{1}{D_{\max}} \right)^2. \quad (5.34)$$

V_2 will be positive in the region where both obstacles are exerting a force if

$$\gamma_c < \frac{\delta_c}{d} \ln(2). \quad (5.35)$$

In the region of the state space where $|x_2| > |V_{\min}|$, only the force pushing the robot away from a single (closer) obstacle is applied (Equation (5.11)). In this case, the applied force always opposes the direction of motion, slowing down the robot and forcing it into the region $|x_2| \leq |V_{\min}|$, where forces resulting from both obstacles are applied.

5.2.1.4 Real-time Algorithm

When the algorithm is extended to an articulated robot operating in three-dimensional task-space, each robot joint is evaluated sequentially as a series of one degree of freedom systems. This results in a local sequential planner, where backtracking motion is not considered [83]. Backtracking motion requires global planning, which cannot be executed in real time. Instead, the goal of the safety module is to generate a plan to move the robot to the safest possible location in real time. The safety module then issues a request to the planner module to generate a global plan – either for retraction or to continue the initiated task.

The one-dimensional algorithm must be modified when applied to a multi-link manipulator, because the position and velocity of each link is affected by motion of any proximal links. Proximal

joints must consider not only critical points at the corresponding link, but all critical points at links distal to the joint. Figure 5.8 shows pseudo-code for the full algorithm. The algorithm proceeds in two steps. First, the danger index at each potential collision location is calculated (procedure CalculateDangerIndex, shown on line 1 of Figure 5.8). In this procedure, all non-zero locations are stored as critical points. If the danger index at any critical point is above a defined threshold, DI_TH , then the planned trajectory is discarded and a new trajectory is generated by the safety module. The desired safe motion for each joint is calculated, starting with the base joint (procedure GetNextCommand, shown on line 21 of Figure 5.8). For each joint, all critical points distal to the joint are considered (i.e., not just critical points at that joint) (line 24). The directions of motion are evaluated in the joint (configuration) space. Analogous to the one dimensional case, three scenarios are possible: (i) there are no critical points distal to this joint, (ii) all critical points generate virtual forces in a single direction, or (iii) critical points generate forces in opposing directions.

If no critical points are present, a virtual damping force (F_d) is applied to stop any motion of that joint (that is, the prior motion assigned by the planned trajectory):

$$F_d = -B\dot{q}, \quad (5.36)$$

where B is a damping constant and \dot{q} is the measured joint velocity. If all critical points generate virtual forces in a single direction, F_{SO} is applied as specified in Equation (5.8) above (line 30). If opposing critical points are present, F_{TO} is applied as specified in Equation (5.9) (line 32). For each joint, a new trajectory is then generated based on the desired acceleration and the current position and velocity.

The stability of the multi-link algorithm is analogous to the 1 DoF case, provided the low-level controller acts to decouple the manipulator joints. A PID controller implicitly considers the robot as a set of independent DoFs, as torque due to off-diagonal inertia terms and the Coriolis and centrifugal terms is treated as a disturbance. This approach is appropriate for geared robots, and robots moving with low velocity and acceleration. For direct-drive robots, a low-level PID controller should not be

used; instead, the joints should be explicitly decoupled by using inverse dynamics in the feedback loop to cancel out the coupled terms.

```
1 procedure: CalculateDangerIndex
2
3 for each robot joint
4     if linkLength(joint) = 0
5         continue
6     end if
7
8     for each obstacleSphere
9         Find the point on the link closest to the obstacleSphere (the critical point)
10        Find the distance vector from the CP to the obstacleSphere
11        Find the velocity along the distance vector
12        Calculate the DI for this CP
13        if DI > 0
14            Store this CP
15        end if
16    end for
17 end for
18
19 return maximum DI given all stored CPs
20
21 Procedure: GetNextCommand
22
23 for each robot joint
24     for each CP distal to this joint
25         Find direction to move joint to decrease DI for this CP (no motion if the
26         robot is in a singularity)
27     end for
28
29     if all CP directions are the same
30         Calc. virtual force using Equation (5.8) with highest DI critical point
31     else
32         Calc. virtual force using Equation (5.9), with highest DI critical point
33         from each direction
34     end if
35 end for
```

Figure 5.8. Pseudo code for the multi DoF algorithm.

5.2.1.5 Implementation

The safety module is implemented as a state machine. The state machine diagram is shown in Figure 5.9. The danger index is calculated in each state. If the danger index is low and the module is not “Engaged”, the planned trajectory is passed on to the low-level robot-controller. The module switches from any state to engaged when the highest danger index is above a threshold. Once evasive action has been taken, the module goes to “Slowdown” to damp any remaining motion at which point a request is generated to the recovery evaluator module to generate a new plan, and then transitions to the “Wait for Plan” state, until a new plan is generated.

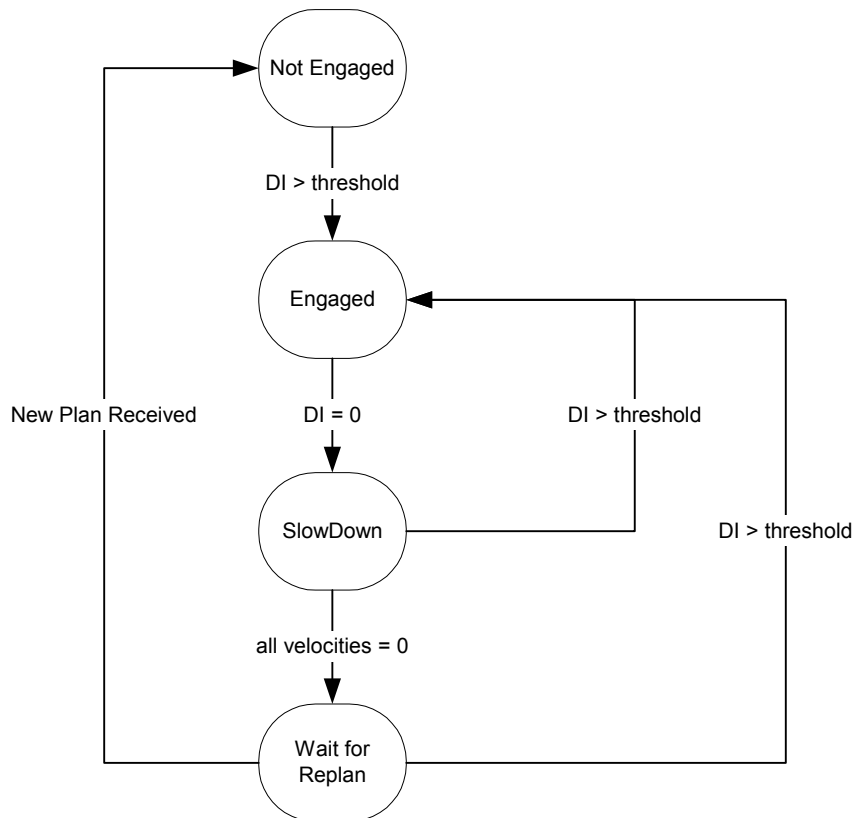


Figure 5.9. Safety module state diagram.

For the subsystem testing described in this chapter, the inertia factor is set to 1. To calculate the distances and velocities between each robot link and the human (or other obstacles) the vision system described in Chapter 6 was used.

5.2.1.6 Parameter Selection

Selecting the parameters for the distance and velocity factors will determine the onset and magnitude of the control action in the event that a hazard is detected. In particular, the minimum safe distance (D_{min}) and the maximum safe velocity (V_{max}) determine when the danger index climbs above 1 and control action strongly increases. These parameters should be based on the physical characteristics and capabilities of the robot. D_{min} can be estimated based on the maximum robot deceleration and velocity, while V_{max} can be estimated based on indices of injury severity, such as the Gadd Severity Index or the Head Injury Criterion, as proposed in Bicchi et al. [30, 31]. The largest distance (D_{max}) and lowest velocity (V_{min}) at which the safety module begins to consider a potential hazard are then selected. The distance D_{max} should be based on the physical size of the robot, and the geometry of the workspace. V_{min} should be smaller than zero to ensure stability. It is desirable to set the ranges between D_{min} and D_{max} , and V_{min} and V_{max} to be large, so that there is time to react before the danger is imminent. However, setting these ranges to be excessively large prevents the robot from successfully operating in a cluttered environment. Too large a range between V_{min} and V_{max} also reduces the effective damping, which can result in oscillations in the 2nd order system at each joint. False positive reactions of the safety module can be significantly reduced by setting the reaction threshold, DI_TH , above zero. In the simulations below, the reaction threshold was set to 0.3.

5.2.2 Simulations

To illustrate the behavior of the safety module, simulations on a 3-DoF planar robot were generated. Each of the robot links is 0.4m long. The obstacles have a radius of 0.2m. Table 5.1 shows the settings for all the algorithm parameters. In each case, the initial robot trajectory was from $[0,0,0]$ (horizontally stretched out) to $[\pi/2;0;0]$ (upright). The robot is animated using the Robotics Toolbox for Matlab [24].

Table 5.1. Parameter values for simulations.

Parameter	Value
Dmin	0.4
Dmax	0.8
Vmin	-0.2
Vmax	1
DI_TH	0.3

Figure 5.10 shows the behavior of the safety module in a sample simulation. The higher obstacle is stationary, while the lower obstacle moves 0.5m vertically up starting from rest at the start of the simulation and stopping halfway through. In this case, the safety module is able to generate a trajectory to clear the robot from the obstacles. As can be seen in the figure, the robot stays further away from the lower obstacle, since the lower obstacle is moving towards the robot, therefore the velocity factor at the critical point between that obstacle and the robot will be higher than the velocity factor at the upper obstacle. Figure 5.11 shows the joint positions along the generated trajectory. The initially commanded (planned) robot trajectory is shown in gray, while the actual trajectory generated by the safety module is shown in black.

Figure 5.12 shows frames of a sample trajectory in a case when the safety module is unable to generate an escape trajectory. In this case, the upper obstacle blocks the possible escape of joint 1, while the lower obstacle moves upwards. In this case, the robot remains between the obstacles, equidistant between the two obstacles in terms of the danger index. Figure 5.13 shows the joint trajectories. While the lower obstacle is moving towards the robot, the robot is closer to the upper obstacle. Once the lower obstacle stops, the robot moves to the middle, between the two obstacles.

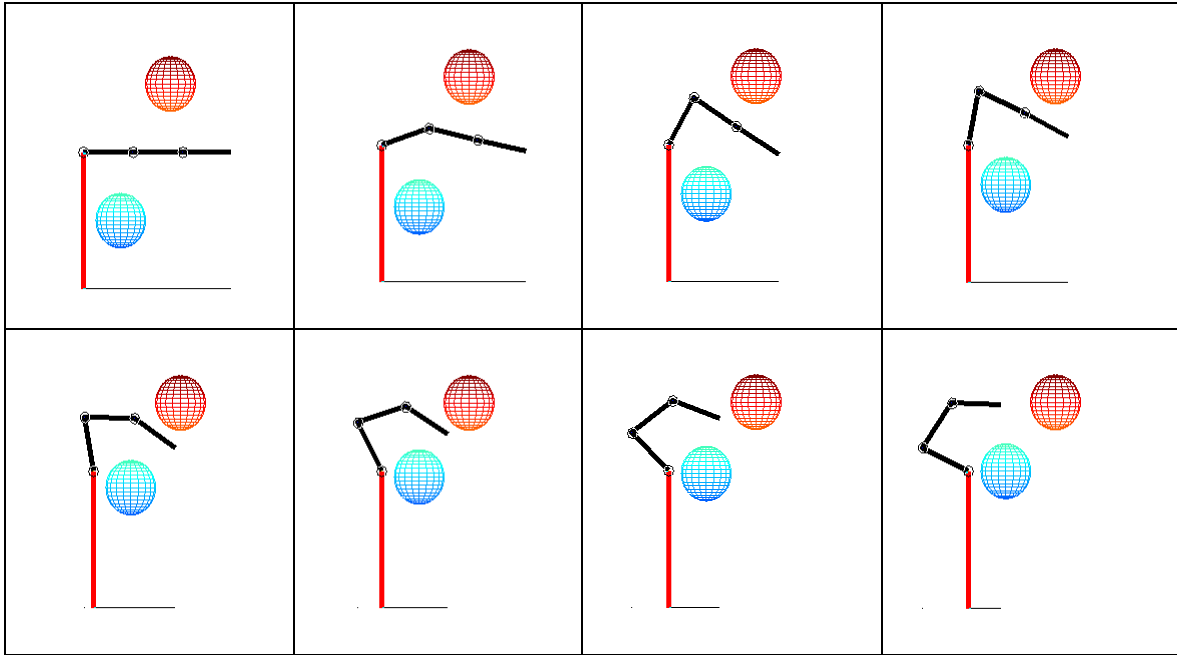


Figure 5.10. Planar robot simulation (robot clears obstacles).

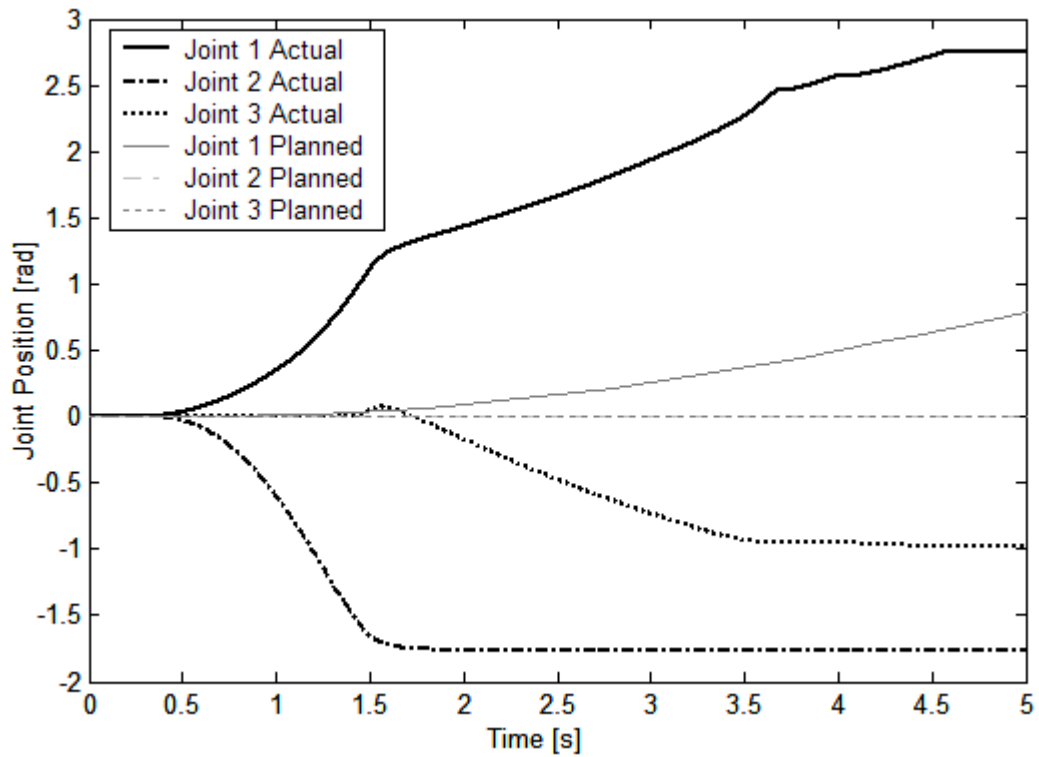


Figure 5.11. Joint trajectory for planar robot simulation (robot clears obstacles).

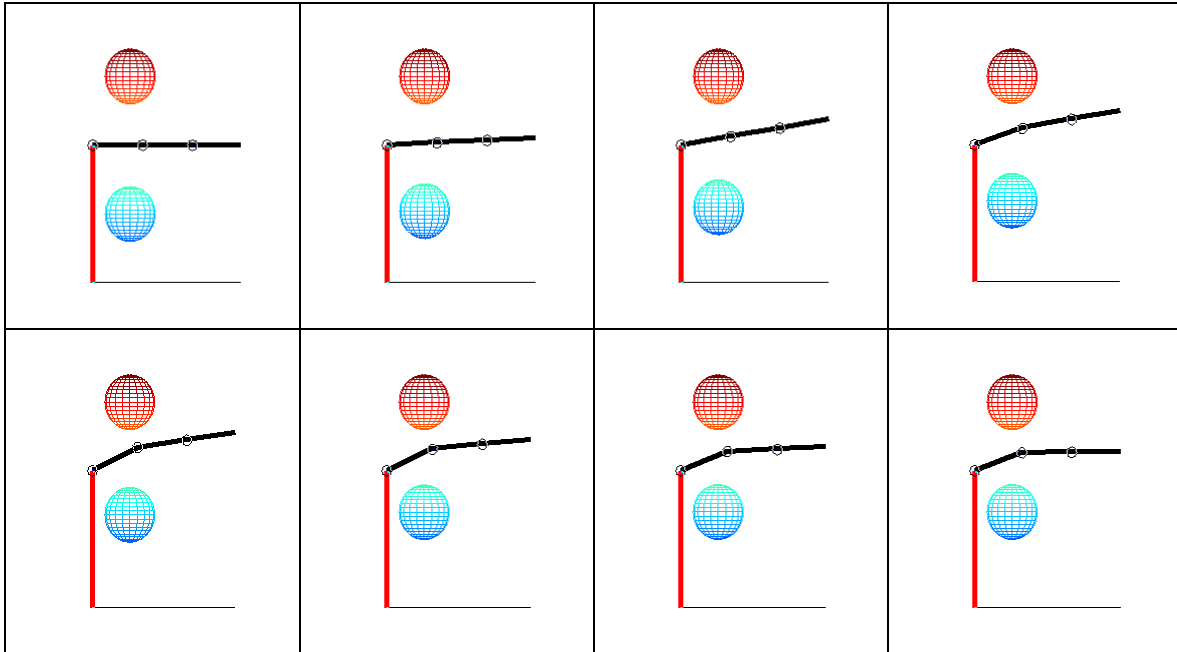


Figure 5.12. Planar robot simulation (robot cannot clear obstacles).

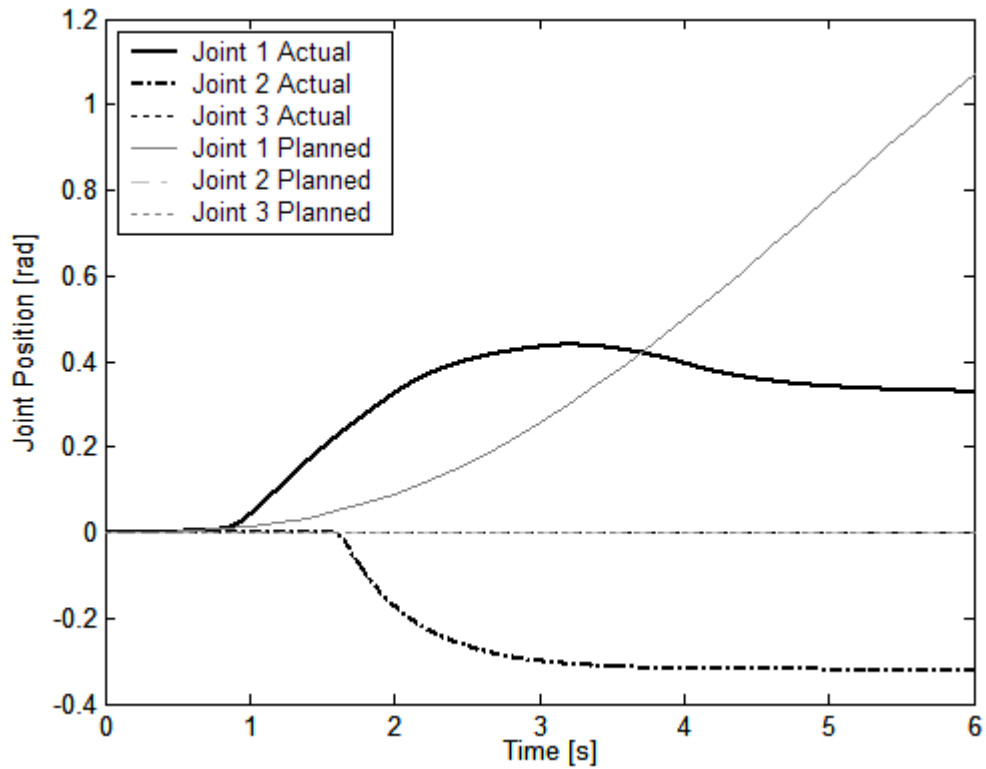


Figure 5.13. Joint trajectory for planar robot simulation (robot cannot clear obstacles).

5.2.3 Experiments

The safety module was tested in standalone mode with the experimental setup described in Chapter 3. The safety module was tested without the planner or recovery evaluator. A default trajectory was issued to simulate the planned trajectory. The person would then move to block the robot's path, and the safety module would engage to move the robot to a safe location.

Table 5.2 shows the values of the parameters used during the experiments.

Table 5.2. Parameter values for experiments.

Parameter	Value
Dmin	0.6
Dmax	1.0
Vmin	-0.5
Vmax	0.5
DI_TH	0.3

Figure 5.14 shows video frames from a sample experiment. In this case, the initial trajectory moves the robot from the upright position towards the table in front of the human. As the human raises his hands towards the robot, the safety module activates and the robot moves upwards and away from the human. Figure 5.15 shows the joint position trajectories for the lower three joints during the sample experiment. The gray lines show the initial (planned) trajectory, while the black lines show the trajectory generated by the safety module. Other sample experiments are shown in Chapter 7, where the safety module is used with the integrated system.



Figure 5.14. CRS robot experiment video frames.

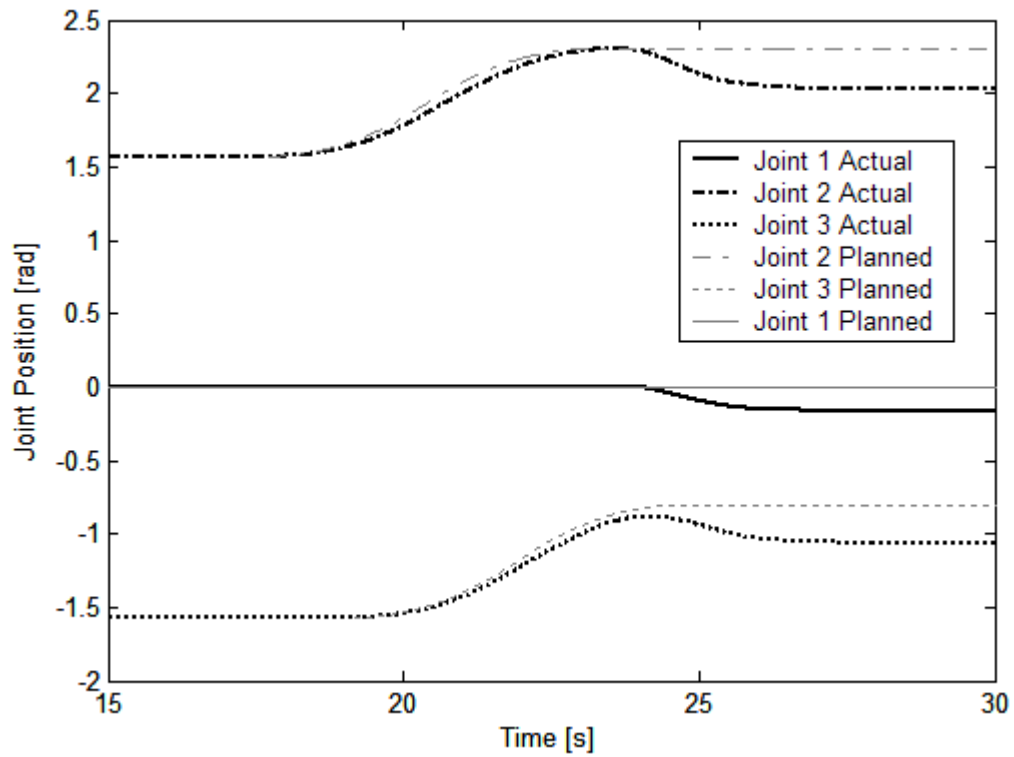


Figure 5.15. Joints 1 - 3 trajectory during CRS robot experiment.

5.2.4 Summary

The danger-index-based safety module described in this chapter provides a methodology to ensure human safety during a human-robot interaction in real time. The level of danger in the interaction due to a potential collision is explicitly defined as the danger index. A sequential one-step ahead trajectory planner (the safety module) generates robot motion by minimizing the danger index. The danger index acts as a non-linear impedance that reacts faster at smaller distances and higher velocities than a comparable linear impedance and is provably stable. The full algorithm can be used for redundant or non-redundant manipulators, and operates correctly at all robot configurations, including singularities. The integration of this safety module with the overall system is discussed and demonstrated in Chapter 7.

Chapter 6: Human Monitoring

During human-human interaction, non-verbal communication signals are frequently exchanged in order to assess each participant's emotional state, focus of attention and intent. Many of these signals are indirect; that is, they occur outside of conscious control. By monitoring and interpreting indirect signals during an interaction, significant cues about the emotional state of each participant can be recognized [22]. Recently, research has focused on using non-verbal communication, such as eye-gaze [20, 55, 84, 85], facial orientation [54, 86, 87], facial expressions [88-92] and physiological signals [22, 59, 61, 68, 93] for human-robot and human-computer interaction. Robot vision is also important for detecting the location of the human in the environment, as well as the presence of any obstacles. In this thesis, the focus was on two human monitoring technologies: physiological signal monitoring and machine vision; these topics as utilized in this work comprise the two main sections of this chapter.

6.1 Affective State Estimation

Although not used during interpersonal interaction, physiological signals are particularly well suited for human-robot interaction, as they are relatively easy to measure and interpret using on-line signal processing methods [59, 61, 93]. By using non-verbal information such as physiological signals, the robot can estimate user approval of its performance without requiring the user to continuously issue explicit feedback [20, 21]. In addition, changes in some non-verbal signals precede a verbal signal from the user. Observation of physiological information can allow the robot control system to anticipate command changes, creating a more responsive and intuitive human-robot interface.

Physiological monitoring systems have previously been used to extract information about the user's reaction, both for human-computer and human-robot interaction. Signals proposed for use in human-computer interfaces include skin conductance, heart rate, pupil dilation and brain and muscle neural activity. Bien et al. [23] advocate that soft computing methods are the most suitable methods for interpreting and classifying these types of signals, because these methods can deal with imprecise and incomplete data.

Sarkar proposes using multiple physiological signals to estimate emotional state, and using this estimate to modify robotic actions to make the user more comfortable [59]. Rani et al. [60, 61] use heart-rate analysis and multiple physiological signals to estimate human stress levels. In [61], the stress information is used by an autonomous mobile robot to return to the human if the human is in distress. In this case, the robot is not directly interacting with the human; physiological information is used to allow the robot to assess the human's condition in a rescue situation.

Nonaka et al. [62] describe a set of experiments where human response to pick-and-place motions of a virtual humanoid robot is evaluated. In their experiment, a virtual reality display is used to depict the robot. Human response is measured through heart rate measurements and subjective responses. No relationship was found between the heart rate and robot motion, but a correlation was reported between the robot velocity and the subject's rating of "fear" and "surprise". All the above studies use virtual environments such as a video game [60, 61], or a virtual robot [62] to simulate an interaction situation. To the authors' knowledge, no studies have been performed to date testing methods suitable for real-time affective state estimation during a physical human-robot interaction.

6.1.1 Affective State Inference

The affective state is estimated based on measured physiological signals such as heart rate, skin conductance and facial muscle contraction. An important question when estimating human emotional response is how to represent the emotional state. Two different representations are commonly used in emotion and emotion detection research: one using discrete emotion categories (anger, happiness,

fear, etc.), and the other using a two-dimensional representation of valence and arousal [94]. Valence measures the degree to which the emotion is positive or negative, and arousal measures the strength of the emotion. The valence/arousal representation adopted herein appears adequate for the purposes of robotic control, and is easier to convert to a measure of user approval. This representation system has also been favored for use with physiological signals and in psychophysiological research [22, 94-96].

Three physiological signals were selected for measurement: skin conductance response (SCR), heart rate and corrugator muscle activity. These three signals have been shown to be the most reliable indicators of affective state in psychophysiological research [22, 95, 96]. Respiration rate was also considered in an early study [68], but was rejected as unsuitable for on-line interaction applications due to the slow physiological response of the signal.

Skin conductance response (SCR) is a strong indicator of affective arousal. Several studies [95-97] have shown that skin conductance is positively correlated with arousal. Bradley and Lang [95] report that 74% of subjects exhibit this correlation.

Corrugator muscle activity measured via electromyogram (EMG) is negatively correlated with valence. The corrugator muscle, located just above each eyebrow close to the bridge of the nose, is responsible for the lowering and contraction of the brows, i.e., frowning, which is intuitively associated with negative valence. Bradley and Lang [95] reported corrugator muscle activity levels that were well above the baseline level for negative valence stimuli, slightly above baseline level for neutral valence stimuli, and slightly below baseline level for positive stimuli. In their study, more than 80% of subjects showed this correlation.

Unlike the SCR and corrugator EMG response, heart activity is governed by many variables, including physical fitness, posture, and activity level as well as emotional state. It is, therefore, more difficult to obtain significant correlation between heart activity and emotional state. In addition, heart rate activity is also dependent on context. In tests using external stimuli to generate the emotional response (such as picture viewing), heart rate response is initially decelerative, while tests using

internal stimulus (recalling emotional imagery) showed an accelerative response [95]. Since the target application is based on external stimuli (i.e., observing the robot's actions), the external stimuli results were used. Using these results, heart rate deceleration is associated with the orienting response (i.e., increased arousal). Heart rate at the baseline, with no heart rate acceleration or deceleration, is associated with low arousal, while high heart rate and heart rate acceleration are associated with high arousal.

Another key finding from psychophysiological research is that physiological responses can be highly variable between individuals, as well as vary for the same individual depending on the context of the response [95, 98, 99]. Pre-processing of the data is necessary prior to inference, in order to extract the relevant features of the signals and to normalize the signal features so that a single inference engine can be used across individuals. The preprocessing of the selected signals is discussed below. Then, the fuzzy rule-base developed on these signal features is discussed in Section 6.1.1.2.

6.1.1.1 Data Processing and Feature Extraction

Heart Rate

Heart activity is measured by measuring the electrical signal of the heart muscle through an electrocardiogram (ECG). The first 3 seconds of Figure 6.1 show a typical ECG signal with the repeated QRS⁷ complex. The ECG signal is then analyzed to extract the heart rate. The algorithm and open source code from [100] are used to extract the heart rate. The algorithm first performs band-pass filtering on the raw ECG signal, and the signal is then differentiated and smoothed by a 80ms moving average window. Peaks are then detected in the resulting signal, and detection heuristics are applied to avoid detecting multiple peaks for a single heart beat. These rules include enforcing a minimum interval of 200ms between peaks, and checking for QRS wave characteristics (i.e., both

⁷ The QRS complex corresponds to the electrical current that causes contraction of the left and right ventricles of the heart, and in a typical ECG signal are the most clearly identifiable features.

positive and negative slopes in the raw signal) to ensure that baseline drift is not misclassified as a peak. The algorithm also automatically determines the threshold at which a peak should be considered a beat. The algorithm detects a heartbeat with an average delay of 0.36 seconds. Although this algorithm is quite robust, noise caused by excessive subject movement can cause the beat detection to fail. Some subjects will move their torso suddenly (i.e., flinch) when presented with a rapid robot motion. The muscle contractions in the shoulder and abdominal muscles during the torso motion introduce noise to the ECG signal causing additional beats to be detected. Figure 6.1 shows a typical signal during a sudden motion by the subject. The start of robot motion is indicated with the square wave signal (low = stopped, high = moving). To eliminate the spurious effects of sudden movements, a check is performed after the heart rate is calculated. If the generated heart rate is more than 30 bpm higher in magnitude from the previously averaged heart rate, that measurement is discarded. Since the average change for heart rate in this type of experiment is expected to range from 2 – 15 bpm (change from baseline) [94], this threshold is well above the rate of change that could be seen in a genuine heart acceleration.

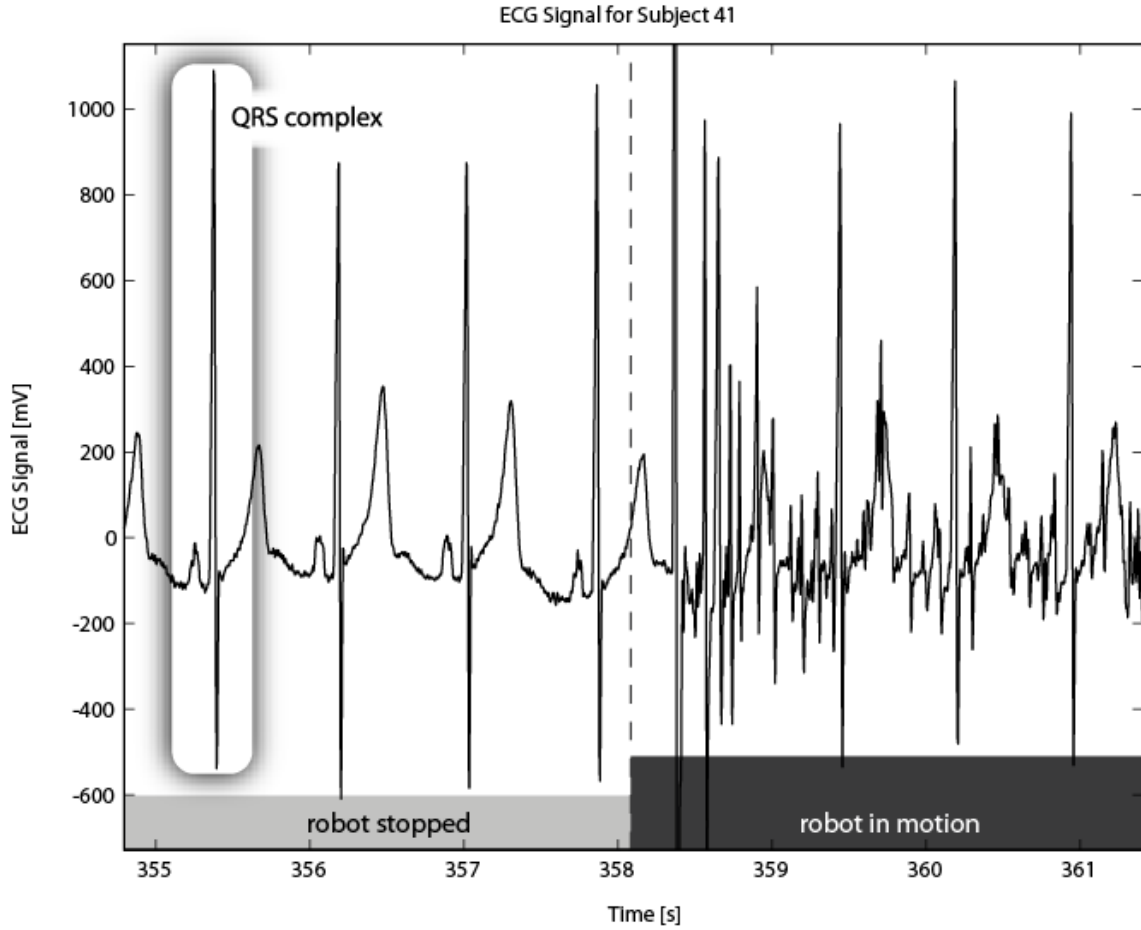


Figure 6.1. Sample heart rate signal (Subject 41).

Once a beat is detected, the beat-to-beat time is used to calculate the heart rate. The heart rate is then smoothed using a 3 sample averaging filter. The average heart rate is also updated. The signal is then normalized to the $[-1,1]$ range based on the average heart rate:

$$h_n = \frac{h - h_{avg}}{h_{max} - h_{min}}, \quad (6.1)$$

where h_n is the normalized heart rate, h is the measured heart rate, h_{avg} is the average heart rate, and h_{min} and h_{max} are the minimum and maximum heart rate, respectively. To generalize the normalization across subjects, $h_{min} = 0.7h_{avg}$ and $h_{max} = 1.5h_{avg}$ are used.

The inference engine also uses the heart rate acceleration to detect accelerative or decelerative periods at the start of an affective response. The heart rate acceleration is calculated by

differentiating the smoothed heart rate signal. The heart rate acceleration is normalized to range between $[-1,1]$, based on the normal heart rate acceleration range [98].

$$a_n = \frac{a_{raw}}{|a_{hmax}|}, \quad (6.2)$$

where a_n is the normalized heart rate acceleration, a_{raw} is the raw heart rate acceleration and a_{hmax} is the maximum observable heart rate acceleration [98].

Recent research [60, 61] has reported the use of frequency domain heart rate analysis for use in emotional state estimation. However, since heart rate data is very slow (less than 1 sample per second), using frequency windowing methods such as windowed Fourier analysis or wavelets results in multi second delays, rendering the data unsuitable for real-time interaction.

Skin Conductance Response

Skin conductance response (SCR) is measured by passing a small current between two electrodes placed on two fingers (of the same hand), and measuring the conductance. Increased perspiration tends to increase the measured conductance. A typical SCR response to robot motion is shown in Figure 6.2. Both the baseline level of SCR and the magnitude of response are highly variable between individuals. Note that, in addition to the specific SCR response (i.e., response to a stimulus), the SCR signal frequently exhibits non-specific responses (for example, the smaller peaks following the specific response peak).

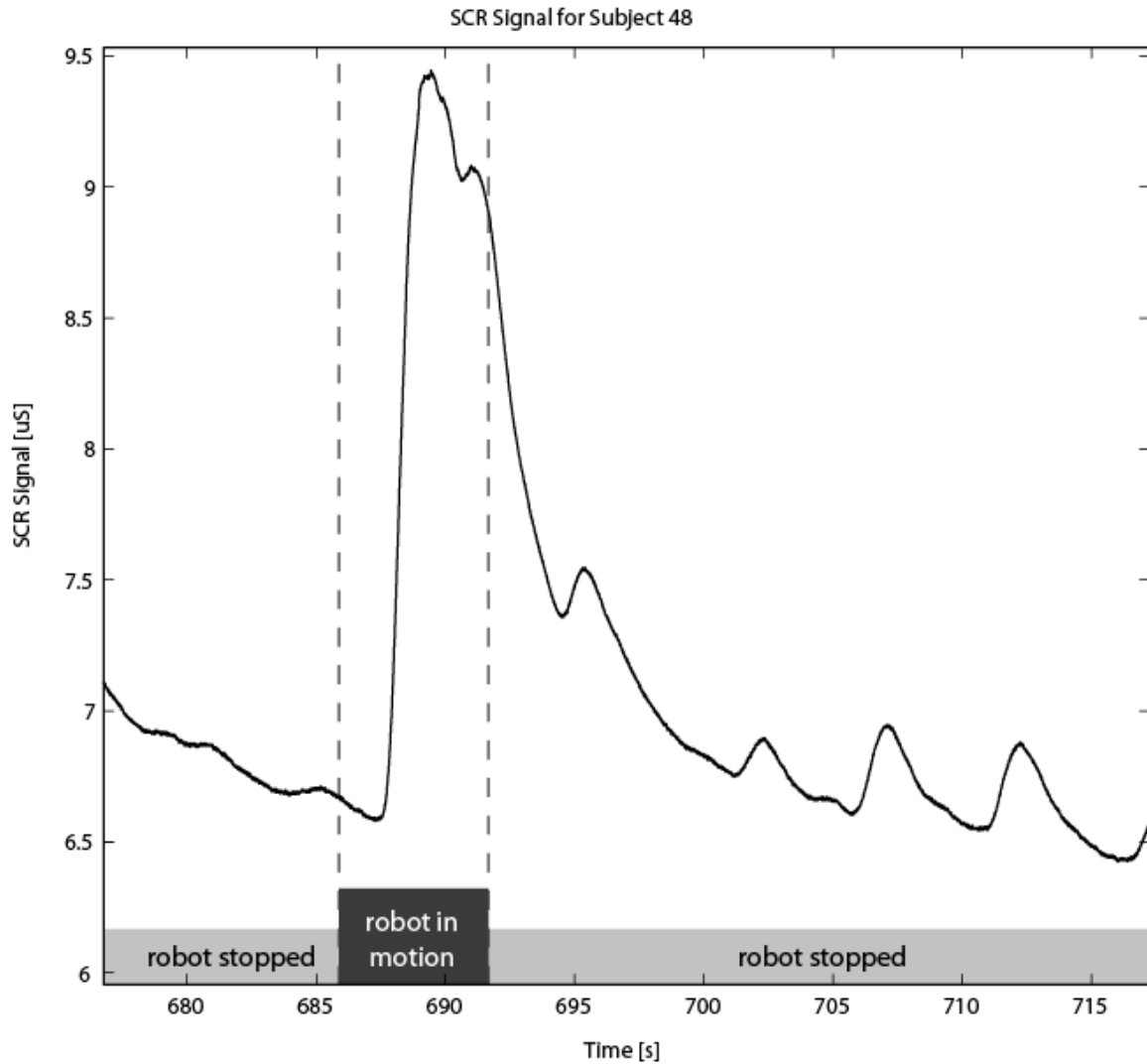


Figure 6.2. Typical SCR response (Subject 48).

Two features were extracted from the skin conductance response: the level of skin conductance response (SCR) and the rate of change of the skin conductance response (dSCR). Normalization of the SCR signal is problematic, because the baseline level of the signal tends to drift, and SCR response is habituating. In previous studies [68, 101], the data was normalized to range between [0,1] using baseline data [68], or, using the minimum and maximum values in the preceding 30 seconds [101]. When using baseline data, the normalization fails to account for signal drift. Normalization using only data in the preceding 30 seconds produces long periods of saturation in the normalized signal. For example, following a long period of low amplitude response, a large response will tend to

saturate the normalized signal for several seconds, thus not giving an accurate normalized signal. To avoid saturation, a band-pass filter was used instead to remove both the low frequency drift and the high frequency measurement noise. A [0.5Hz, 5Hz] 3rd order Butterworth filter was used to perform the filtering. The normalized signal was generated as shown in Equation 6.3.

$$s_n = \frac{s_f}{s_{max}}, \quad (6.3)$$

where s_n is the normalized skin conductance, s_f is the band-pass filtered skin conductance and s_{max} is the maximum skin conductance for the subject. The best results are obtained when s_{max} is known for a subject a priori (through previous tests with the robot), however when the system is being used with an unknown subject, a good estimate for s_{max} based on experimental data is:

$$s_{max} = 0.2 + 2.3s_{max}^{resting}, \quad (6.4)$$

where $s_{max}^{resting}$ is the maximum value of the SCR signal during the initial (resting) phase of the trial. The relationship between the maximum resting value of the SCR and the maximum SCR value during high arousal is fairly linear, with a correlation coefficient of 0.89. The value of s_{max} can then be adjusted during robot operation as more data is acquired for the subject.

Prior to calculating the rate of change of skin conductance response (dSCR), the raw SCR data is low-pass filtered with a 5th order 5Hz Butterworth filter. The dSCR response is then calculated by differentiating the filtered SCR data and normalizing so that the data ranges from [-1,1], based on the normal range of rise times for SCR [99]. The inference engine is robust to changes in the normalization procedure for the slope of the skin conductance response (dSCR), as only the direction of the slope is input to the inference engine (i.e., is the signal increasing or decreasing) and not the magnitude.

Corrugator Muscle Activity

Corrugator Muscle activity was measured using an electromyogram (EMG), which measures the electrical activity in the muscle during contraction. The EMG response is shown in Figure 6.3.

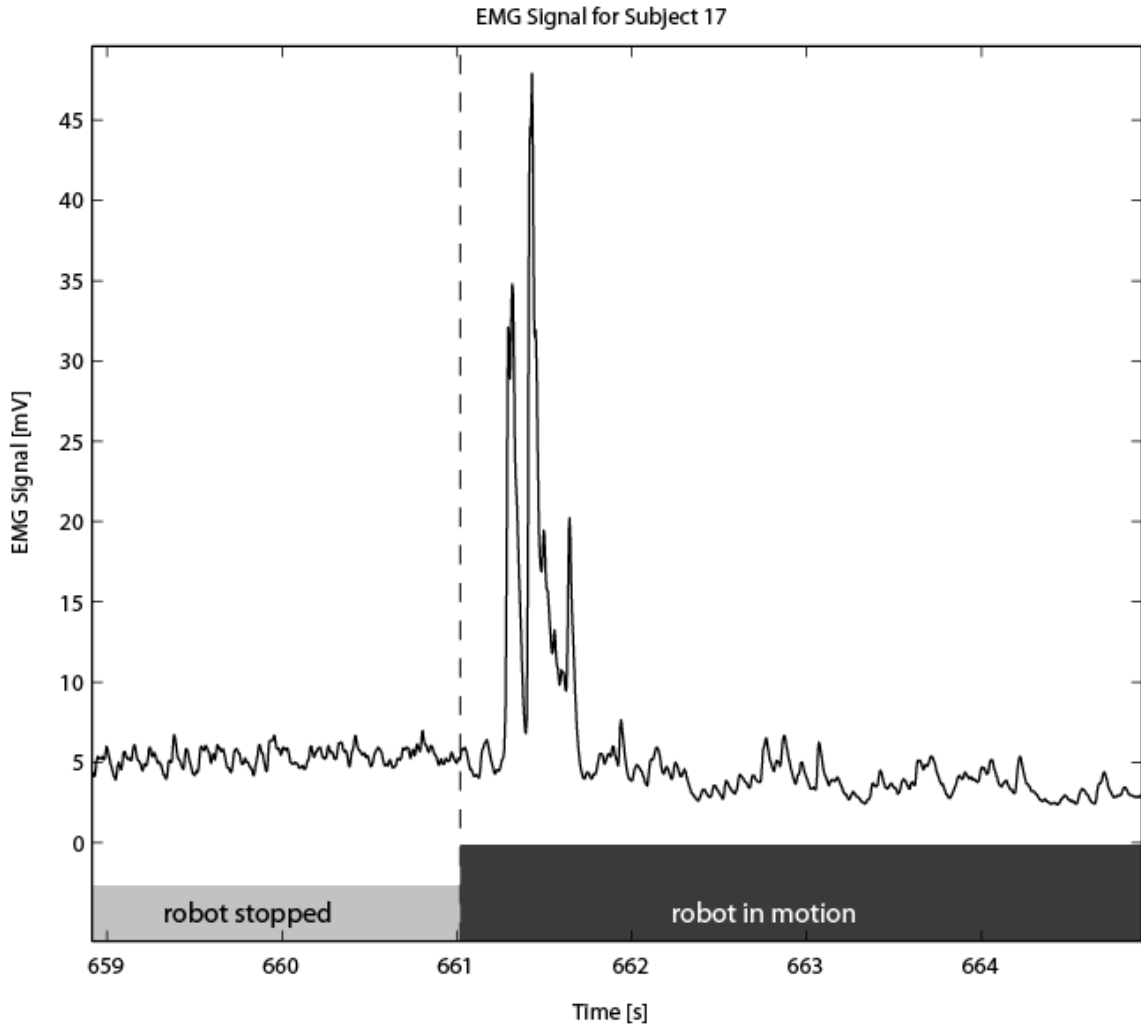


Figure 6.3. Sample EMG Signal (Subject 17).

One feature was extracted from the corrugator muscle EMG data: the level of response $C_{corrEMG}$. The EMG data was low-pass filtered and smoothed using a 5th order Butterworth filter with a cutoff frequency of 5 Hz. The data was normalized to range between [0,1], based on the resting EMG level measured during the initialization phase, as shown in Equation 6.5.

$$c_n = \frac{c_f - c_{rest}}{5c_{rest}} \quad (6.5)$$

where c_n is the normalized corrugator EMG, c_f is the current (filtered) measured value of the corrugator EMG and c_{rest} is the resting corrugator EMG level measured during the initialization phase.

6.1.1.2 Fuzzy Inference Engine

Physiological data is highly variable, both between individuals and interaction contexts. In addition, quantitative descriptions for the relationship between physiological measures and emotional categories are not available. However, psychophysiological research [94-96, 98, 99, 102], and recent work on emotion recognition [22, 93] can provide qualitative relationships. A fuzzy inference engine is well suited for encoding these types of relationships [23, 68].

The five features extracted from the physiological signals, as described in Section 6.1.1.1 (heart rate, heart rate acceleration, skin conductance response, rate of change of skin conductance, corrugator muscle response) were input into a fuzzy inference engine to estimate the emotional response of the subject. The fuzzy rule base used to estimate the emotional state is similar to the rule base reported in [68, 101]. The five extracted features (HeartRate, HRAccel, SCR, dSCR and CorrugEMG) were fuzzified using simple trapezoidal input membership functions. The outputs of the fuzzy engine were the estimated valence and the estimated arousal.

Table 6.1 shows the rule base for the system. This rule base was derived using data from psychophysiological research [95-99]. Physiological responses can be highly variable between individuals, as well as vary for the same individual depending on the context of the response. In addition, not all subjects present with the same physiological response. For example, 74% of subjects exhibit a correlation between skin conductance response and arousal [95]. Therefore, the rule base was structured such that reliable outputs would be obtained even if a subject did not exhibit all of the responses characterized by existing research. For this reason, each input was handled with separate rules (e.g., If SCR = HIGH then AR = HIGH), rather than combining indices (e.g. If SCR = HIGH and HR = HIGH then AR = HIGH).

In Table 6.1, rules 1 - 8 encapsulate the relationship between the skin conductance response and arousal. If the skin conductance is high or increasing, arousal is high. Rules 9 – 13 describe the relationship between corrugator muscle EMG and valence. High corrugator muscle activity corresponds to negative valence, while very low corrugator muscle activity (below the resting level) indicates positive valence. Rules 14 – 17 relate heart activity to affective state. Constant heart rate at the baseline corresponds to low arousal, while high heart rate and heart rate acceleration are associated with high arousal. Due to the additional variables affecting heart rate response, heart rate rules were under weighted relative to the SCR and EMG rules.

Table 6.1. Fuzzy inference engine rulebase.

1. If (SCR is ZERO) and (dSCR is ZERO) then (Arousal is LOW)
2. If (SCR is LOW) and (dSCR is NEG) then (Arousal is LOW)
3. If (SCR is LOW) and (dSCR is ZERO) then (Arousal is LOW)
4. If (SCR is LOW) and (dSCR is POS) then (Arousal is MED)
5. If (SCR is MED) then (Arousal is MED)
6. If (SCR is HIGH) then (Arousal is HIGH)
7. If (SCR is ZERO) and (dSCR is NEG) then (Arousal is LOW)
8. If (SCR is ZERO) and (dSCR is POS) then (Arousal is MED)
9. If (CorrugEMG is NEG) then (Valence is POS)
10. If (CorrugEMG is ZERO) then (Valence is ZERO)
11. If (CorrugEMG is LOW) then (Valence is ZERO)
12. If (CorrugEMG is MED) then (Valence is NEG)
13. If (CorrugEMG is HIGH) then (Valence is VNEG)
14. If (HeartRate is VNEG) then (Arousal is HIGH)
15. If (HRAccel is VNEG) then (Arousal is MED)
16. If (HeartRate is VPOS) then (Arousal is HIGH)(Valence is NEG)
17. If (HRAccel is VPOS) then (Arousal is HIGH)(Valence is NEG)

6.1.2 Experiments

The fuzzy inference engine was tested in a human-robot interaction trial. The trial protocol was reviewed and approved by the UBC Behavioral Research Ethics Board. The experiment was designed to generate various robot motions and to evaluate both the human subjective response and

physiological response to the motions. The affective state was estimated on-line during the experiment, using the inference engine described in Section 6.1.1.

The experiment was performed using the CRS A460 6 DoF manipulator, using the setup described in Chapter 3. A group of 36 human subjects were tested; 16 were female and 20 were male. The age of the subjects ranged from 19 to 56, with an average age of 29.2.

6.1.2.1 Trajectory Generation

Two different tasks were used for the experiment: a pick-and-place motion (PP), similar to the trajectory displayed to subjects in [62] and a reach and retract motion (RR). These tasks were chosen to represent typical motions a robot could be asked to perform during human-robot interaction. For the pick-and-place motion, the pick location was specified to the right and away from the subject, and the place location was directly in front and close to the subject. For the reach and retract motion, the reach location was the same as the place location. For both tasks, the robot started and ended in the “home” upright position. Each of the selected positions is shown (from the subject’s point of view) in Figure 6.4.

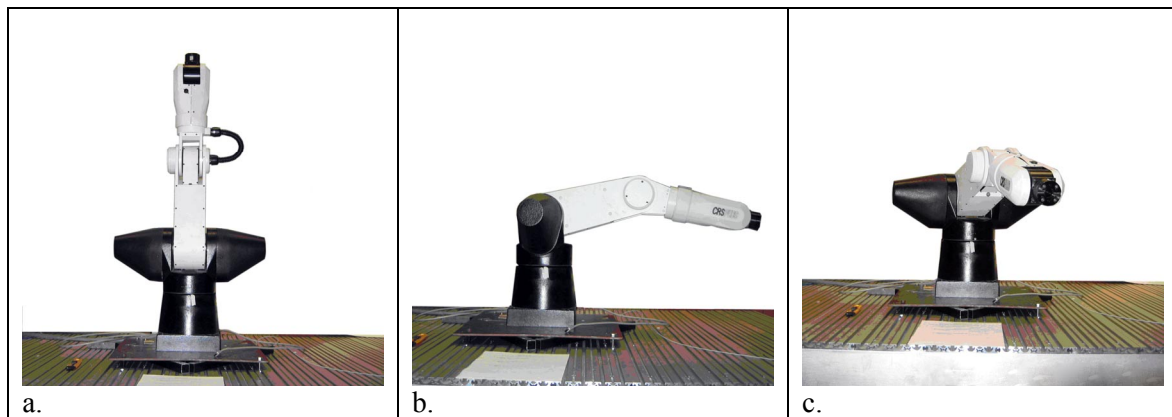


Figure 6.4. Robot task positions (a = robot start/end position, b = pick position, c = place/reach position).

Two planning strategies were used to plan the path of the robot for each task: a conventional potential field (PF) method with obstacle avoidance and goal attraction [44], and a safe path method

(S) described in Chapter 4. Point to point planning was not used, as this type of planning would not be suitable for an interactive, human environment. The four motions tested are detailed in Table 6.2. Figure 6.5, Figure 6.6, Figure 6.7 and Figure 6.8 show frames of video data depicting each motion type.

Table 6.2. Test path naming and descriptions.

Path	Description
PP-PF	Pick and Place task planned with potential field planner
PP-S	Pick and place task planned with the safe planner
RR-PF	Reach and retract task planned with the potential field planner
RR-S	Reach and retract task planned with the safe planner

Given the path points generated for each task by the two planners, a motion trajectory was generated using the trajectory planner described in Chapter 5. For each path, trajectories at three different speeds were planned (slow, medium, fast), resulting in 12 trajectories. To generate the slow, medium and fast trajectories, trajectory scaling values of 0.1, 0.5 and 1.0 were used, respectively. Table 6.3 specifies the time to execute each trajectory.

Table 6.3. Trajectory Execution Times.

Path	Speed [seconds]		
	Slow	Medium	Fast
PP-PF	27.68	8.74	6.37
PP-S	65.11	16.22	10.11
RR-PF	19.78	5.56	3.78
RR-S	46.69	10.94	6.47



Figure 6.5. Path PP-PF (pick and place task planned with the potential field planner).

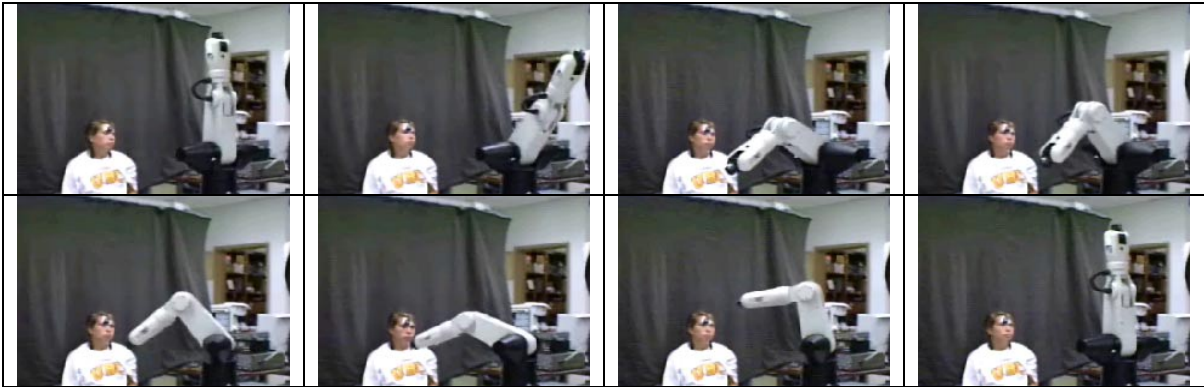


Figure 6.6. Path PP-S (pick and place task planned with the safe planner).



Figure 6.7. Path RR-PF (reach and retract task planned with the potential field planner).



Figure 6.8. Path RR-S (reach and retract task planned with the safe planner).

6.1.2.2 Physiological Sensing

The ProComp Infinity system from Thought Technology [67] was used to gather the physiological data, as described in Chapter 3. As discussed in Section 6.1.1.1, heart muscle activity, skin conductance and corrugator muscle activity were measured. The heart muscle activity was measured via electro cardiogram (ECG) measurement using EKG Flex/Pro sensor. The skin conductance was measured using the SCFlex-Pro sensor. Corrugator muscle activity was measured with the Myoscan Pro electromyography (EMG) sensor.

6.1.2.3 Experimental Procedure

For each experiment, the human subject was connected to the physiological sensors and seated facing the robot. The robot was initially held motionless for a minimum of 30 seconds to collect baseline physiological data for each subject. The robot then executed the 12 trajectories described above. The trajectories were presented to each subject in randomized order. After each trajectory had executed, the subject was asked to rate their response to the motion in the following emotional response categories: anxiety, calm and surprise. The Likert scale (from 1 to 5) was used to characterize the response, with 5 representing “extremely” or “completely” and 1 representing “not at all”. The subject was also asked to rate whether the robot attracted and/or held their attention during the motion, on the same Likert scale, with 5 representing “full attention”, and 1 representing “not attentive at all”.

For each trajectory, the average arousal and valence over the duration of the trajectory were calculated from the physiological sensors data as processed by the inference engine described in Section 6.1.1.2.

6.1.3 Results

The data generated through the user study was analyzed in two stages. The subject reported responses were analyzed to determine how the various robot motions affected the subject’s perceived anxiety, calm and surprise, and to determine if the safe planned motions were perceived to be less threatening. The estimated responses were analyzed to assess the effectiveness of the inference engine and the relationship between the physiological responses and perceived affective state.

6.1.3.1 Subject Reported Response

Figures 6.9 – 6.12 show the average subjective response and a comparison of the average responses between the potential field and the safe planned paths for the subject rated anxiety, calm, surprise and attention, respectively. Table 6.4 shows the correlation analysis between the subjective

responses and the trajectory speed for each trajectory type. For each set of variables, the probability value (the p-value) was computed from a two-sided t-test. The p-value indicates the probability that the correlation was observed by chance. Due to the large sample size, the p-value for all correlations was less than 0.0001.

As expected, for each trajectory, there is a strong positive correlation between anxiety and speed, and surprise and speed, and a negative correlation between calm and speed. There is also strong positive correlation between anxiety and surprise, and strong negative correlation between anxiety and calm and surprise and calm. Correlation among the subjective emotional responses is shown to validate the use of the valence-arousal emotional model. There is also a weak correlation between the level of attention reported and the emotional responses.

A comparison of the graphs in Figures 6.9 – 6.12 indicates that for each motion type (pick and place or reach and retract), on average the subjects reported lower levels of anxiety and surprise, and higher levels of calm, for the safe planned paths. This observation is confirmed by a three factor analysis of variance (ANOVA) performed on each of the subjective responses. The three factors are Plan (potential field (PF) vs. safe plan (S)), Task (reach and retract (RR) vs. pick and place (PP)), and Speed. The significant factors at $p < 0.05$ for anxiety, calm and surprise are shown in Table 6.5, Table 6.6 and Table 6.7, respectively. The ANOVA tables indicate the sum of squares (i.e., squared residuals from the average), the degrees of freedom, the mean square (sum of squares divided by the degrees of freedom) and the test statistics for each factor. For the test statistics, the value of the test statistic F and the probability that the factor variance is due to chance are reported. For these responses, the plan, speed and plan*speed interaction were found to be significant factors. For the attention subjective response, only speed was a significant factor ($p < 0.0001$). For the emotion ratings (anxiety, calm and surprise), the results show a statistically significant reduction in anxiety and surprise (and increase in calm) when the safe planner is used when compared with the generic potential field planner. The plan*speed interaction indicates differing slopes (for example, slower increasing anxiety) between the conventional and safe planners.

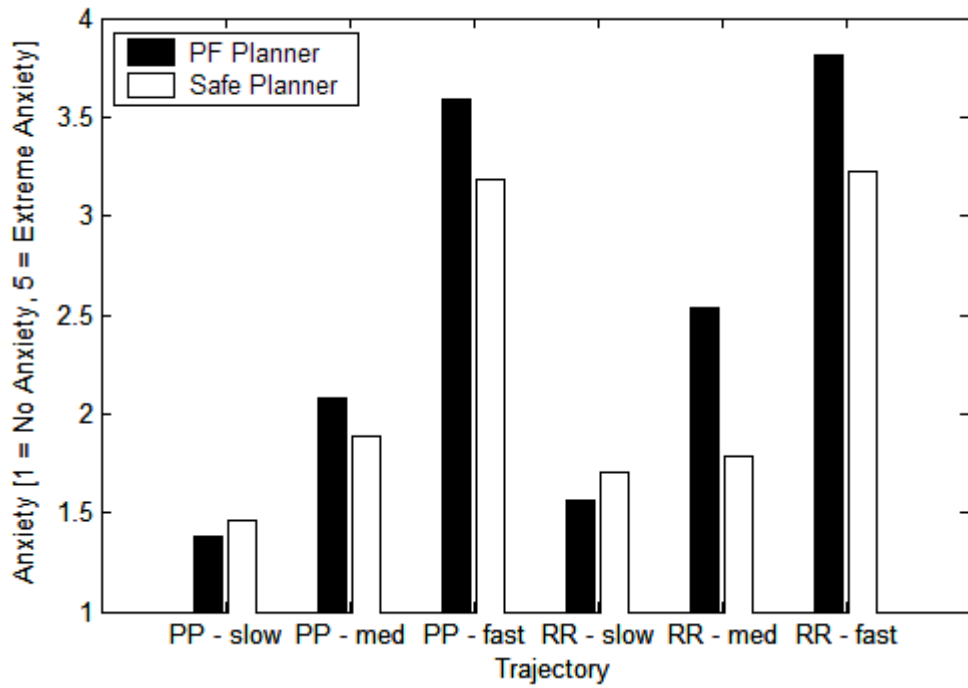


Figure 6.9. Subject reported average anxiety response.

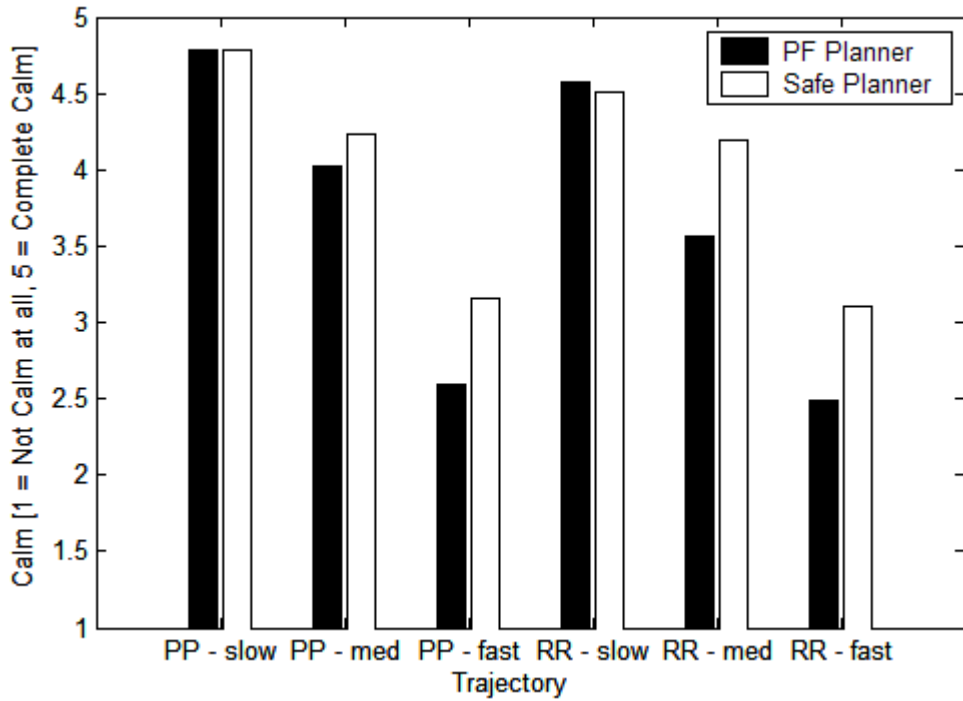


Figure 6.10. Subject reported average calm response.

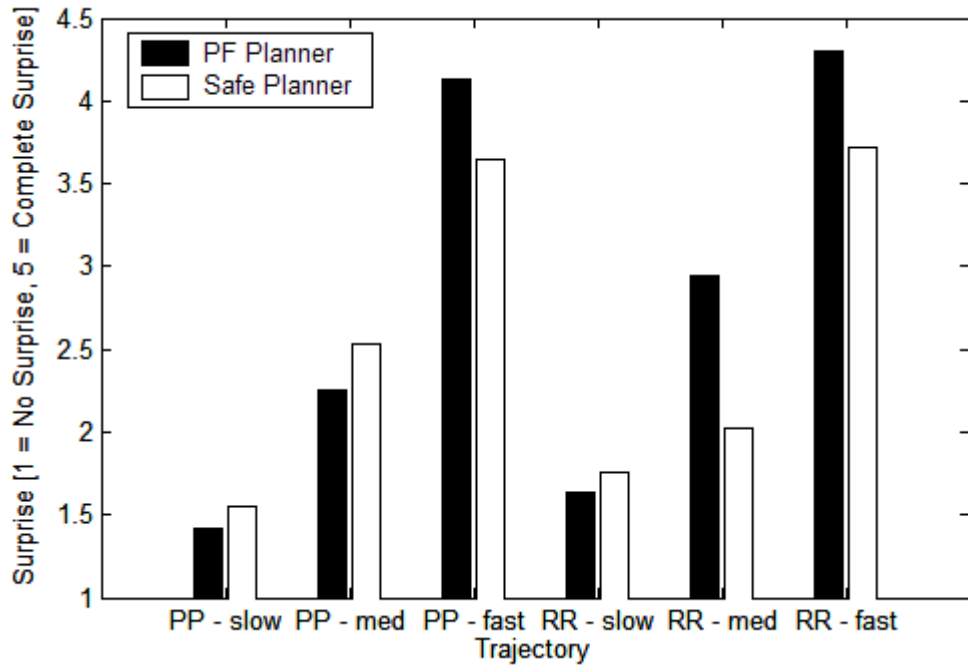


Figure 6.11. Subject reported average surprise response.

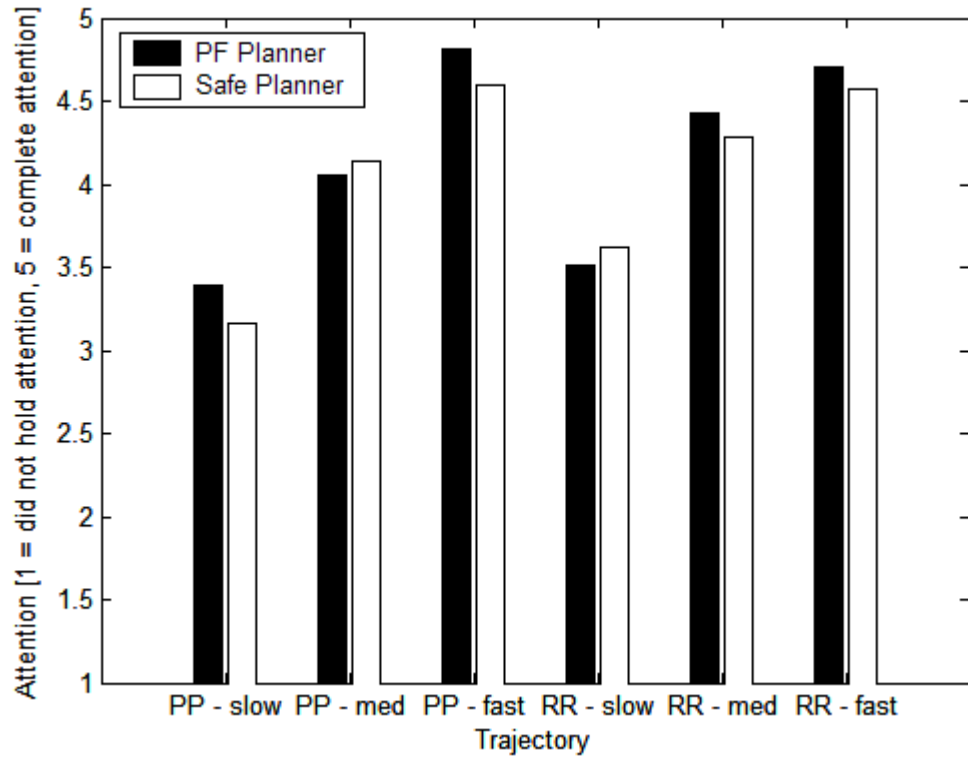


Figure 6.12. Subject reported average attention response.

Table 6.4. Subjective results correlation analysis.

	Anxiety	Calm	Surprise	Attention	Speed
Anxiety	1.0000	-0.8174	0.7832	0.3760	0.6436
Calm		1.0000	-0.7119	-0.3258	-0.6413
Surprise			1.0000	0.3968	0.6843
Attention				1.0000	0.4476
Speed					1.0000

Table 6.5. ANOVA for anxiety.

Significant Factor	Sum of Squares	Degrees of freedom	Mean Squares	F	Probability the factor variability is due to chance
Plan	9.0422	1	9.0422	9.8287	0.001837
Speed	301.39	2	150.695	163.8018	0
Plan*Speed	7.5428	2	3.7714	4.0994	0.017241
Error	391.9132	426	0.91998		
Total	709.8883	431			

Table 6.6. ANOVA for calm.

Significant Factor	Sum of Squares	Degrees of freedom	Mean Squares	F	Probability the factor variability is due to chance
Plan	11.5052	1	11.5052	14.1383	0.00019357
Speed	259.2928	2	129.6464	159.3171	0
Plan*Speed	7.8576	2	3.9288	4.828	0.008446
Error	346.6632	426	0.81376		
Total	625.3189	431			

Table 6.7. ANOVA for surprise.

Significant Factor	Sum of Squares	Degrees of freedom	Mean Squares	F	Probability the factor variability is due to chance
Plan	6.5023	1	6.5023	5.9369	0.015236
Speed	430.5104	2	215.2552	196.5382	0
Plan*Speed	8.4178	2	4.2089	3.8429	0.022177
Error	466.5694	426	1.0952		
Total	912	431			

6.1.3.2 Estimated Response from Physiological Sensors

Arousal

Figure 6.13 shows the average estimated arousal for each trajectory tested. Table 6.8 shows the ANOVA for arousal, showing all significant factors at $p < 0.05$. As can be seen from the results, the estimated arousal behaves similarly to the subject reported data. Estimated arousal tends to increase with speed, and this is the most significant factor in the ANOVA analysis. The plan type is the

second most significant factor, showing that arousal is significantly lower when safe planned paths are used. Similarly to the subjective response data, the plan*speed interaction is also statistically significant. Unlike the subjective response data, for the estimated arousal, the task type is also a significant factor, i.e., there is a significant difference in estimated arousal between the pick and place and reach and retract tasks. Table 6.9 shows the correlation analysis between the estimated arousal, the subject-reported responses and the trajectory speed for each path. All of the correlations were found to be significant at $p < 0.001$. The estimated arousal is positively correlated with the reported anxiety and surprise, and negatively correlated with calm. The estimated arousal is also strongly correlated with the trajectory speed. However, unlike the subject reported responses, the estimated arousal also has a small negative correlation ($r = -0.1334$, $p = 0.0055$) with the presentation order; i.e., trajectories presented toward the end of the trial tend to elicit less response than early trajectories, regardless of the trajectory type. This is likely due to the habituation properties of most of the physiological responses, especially the SCR [99].

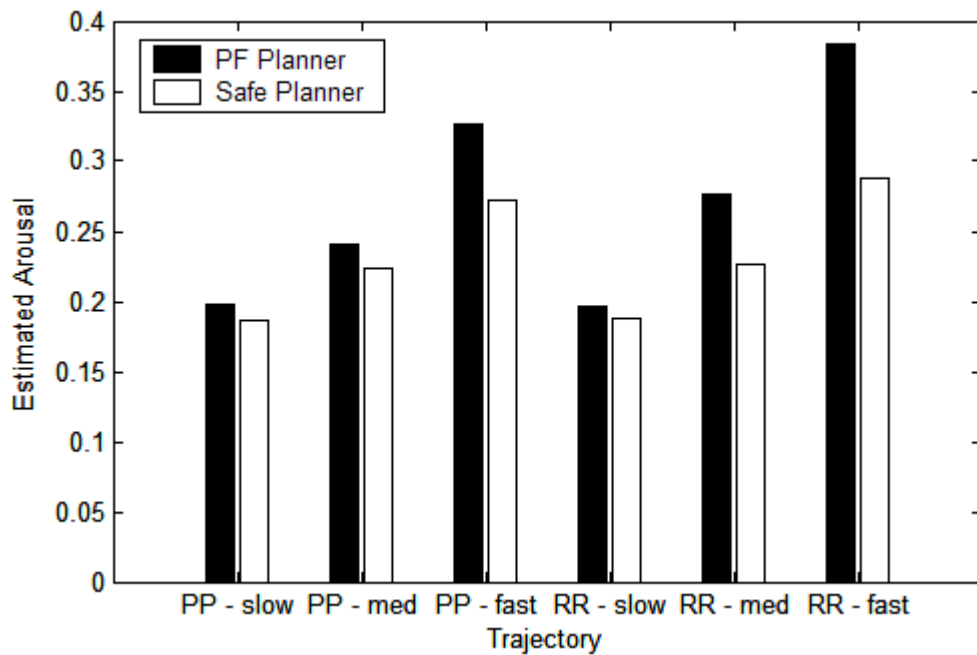


Figure 6.13. Average estimated arousal from physiological sensors.

Table 6.8. Arousal ANOVA.

Significant Factor	Sum of Squares	Degrees of freedom	Mean Squares	F	Probability the factor variability is due to chance
Task	0.03803	1	0.03803	13.57	0.0003
Plan	0.16803	1	0.16803	59.94	0
Speed	1.15162	2	0.57581	205.4	0
Plan*Speed	0.07712	2	0.03856	13.75	0
Error	1.19144	425	0.0028		
Total	2.6265	431			

Table 6.9. Correlation analysis for estimated arousal.

	Estimated Arousal
Reported Anxiety	0.5369
Reported Calm	-0.5694
Reported Surprise	0.5646
Reported Attention	0.3766
Speed	0.6611

As described in Section 6.1.1.2, two signals were used for arousal estimation: SCR and heart rate. SCR is the more reliable indicator. SCR response to robot motions was observed for all subjects. Heart rate response was not as reliable, as only some of the subjects exhibited detectable heart rate response to robot motion stimuli. Of the 36 subjects, 10 exhibited heart rate deceleration following motion onset, consistent with the orienting response [103]. However, neither the magnitude of the deceleration nor the magnitude of the heart rate decrease varied with the type of stimulus. Therefore, heart rate deceleration is more useful in detecting the presence of stimulus, rather than the severity of the stimulus. Only 3 of the 36 subjects exhibited heart rate acceleration in response to robot motions. For these subjects, the magnitude of the heart rate increase was correlated to the speed of the robot motion.

The fuzzy inference engine design addresses both HR responsive and non-responsive subjects. Only very negative (VNEG) and very positive (VPOS) heart rate responses activate the rule-base; these types of responses are not observed for subjects who do not have a heart rate response to robot motion. Using only the extremum of the response also allows the inference engine to avoid false positives due to changes in heart rate correlated with breathing or slight posture changes [98].

Table 6.10 shows the confusion matrix for comparing subjective responses to estimated arousal. The subjective arousal was estimated as the maximum of the anxiety and surprise response and the inverse of the calm response. For the subjective arousal, a response of 1 on the Likert scale was classified as low, a response of 2 or 3 was classified as medium, and a response of 4 or 5 was classified as high. As can be seen from this table, the inference engine performs better when reported emotional response is high. When subjects rated their anxiety or surprise as medium or high, estimated arousal was correctly classified as high 69% of the time. All of the misclassification of high subjective arousal was to the medium estimated arousal. That is, in no case was elevated arousal misclassified as low arousal. The inference engine is not effective in the low reported arousal region, where most of the responses are classified in the medium arousal range. The inference engine is far more likely to generate a false positive (high arousal when low surprise/anxiety are reported), rather than a false negative. This is due to the fact that both heart rate and SCR response occurs during the orienting response [103], especially for novel stimuli. When subjects are presented with robot motion, some SCR response occurs even when they rate their anxiety and surprise as low.

Table 6.10. Confusion matrix - subject reported vs. estimated arousal.

		Subject Reported Arousal		
		Low	Med	High
Estimated Arousal	Low	4	3	0
	Med	80	125	45
	High	20	54	101

Valence

The emotional state inference engine also attempted to estimate valence, based primarily on the corrugator muscle EMG signal. Corrugator muscle activity has been reported to have strong correlation with negative valence in existing studies [60, 61], and had shown promise during earlier studies with the inference engine using images as stimuli [68]. However, EMG activity was not reliably detected for the majority of subjects in response to the robot motion stimulus, confirming our earlier results [101]. Most subjects showed little variation of the EMG signal during robot motion.

EMG activity was more likely to be observed following robot motion, while the subject was thinking about and announcing their subjective rating of the motion. Figure 6.14 shows a common pattern of EMG response. During the robot motion, a very small change in EMG is observed, however, following the end of robot motion, while the subject is thinking about and articulating their subjective responses, significantly more EMG activity is observed. This result indicates that corrugator EMG activity observed in [61, 68] is associated with cognitive processing, rather than instinctive response such as the startle reflex or fight-or-flight response.

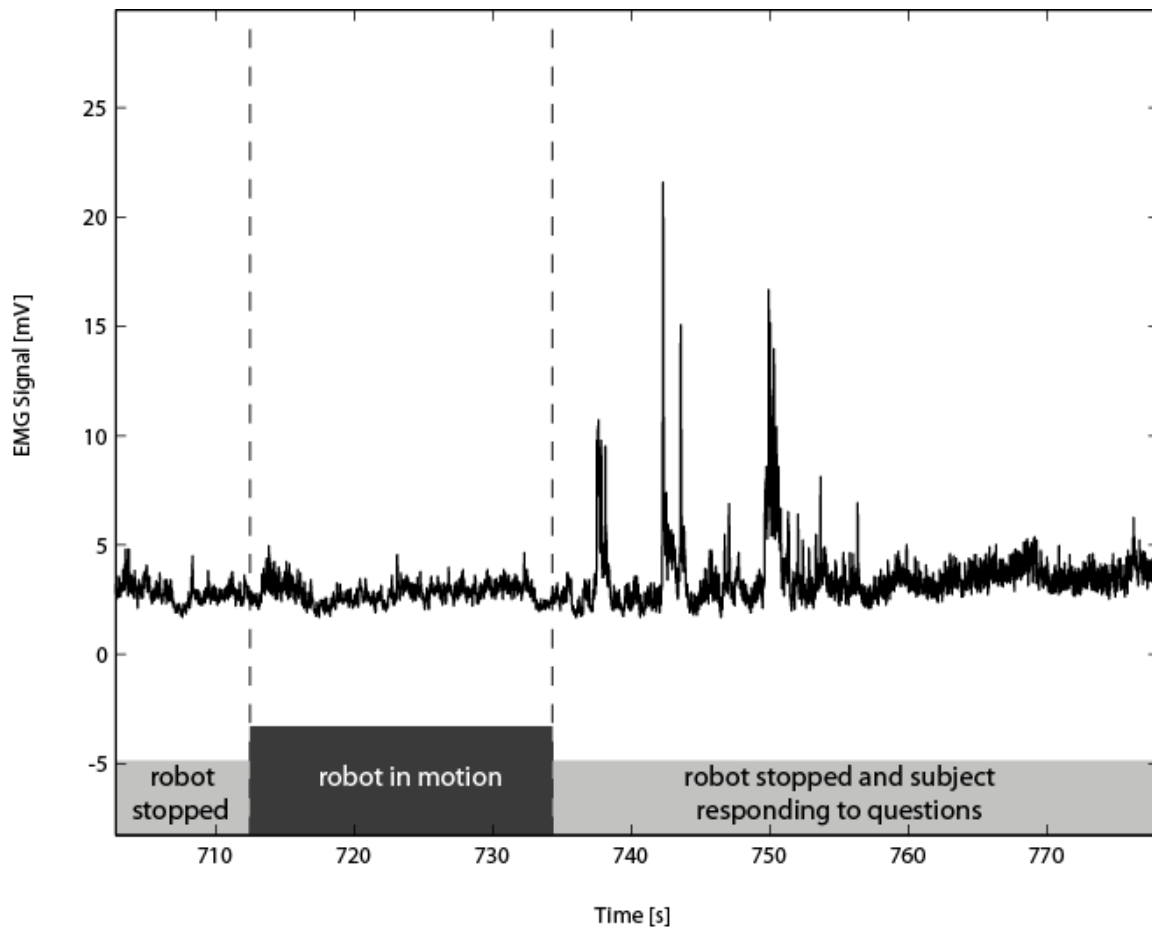


Figure 6.14. EMG activity during question answering.

6.1.4 Affective State Estimation Summary

Two types of robot motions were presented to human subjects during the study: motions planned with a conventional potential field planner, and motions planned with the safe planner. Subjects

reported significantly less anxiety and surprise, and reported feeling more calm when safe planned motions were presented, as compared to the conventional potential field planner.

The results from this study indicate that physiological signals provide a potentially useful additional signal for use during human-robot interaction. Fast robot motions tend to reliably elicit a strong arousal response. For slow motions, a key question is whether the measured arousal indicates a measure of an involuntary reaction the subject may not be aware of (such as the startle reflex orienting response), or a measure of a consciously experienced emotional state such as anxiety or surprise.

To the author's knowledge, this is the first study to examine the response of the corrugator muscle to robot motion as the stimulus. An important result from the user trial is that corrugator muscle activity does not reliably occur for this type of stimulus. Even when high levels of anxiety (a negative valence emotion) were reported, corrugator muscle activity was not reliably present. More research is needed to understand the relationship between corrugator muscle activity and affective state for this type of stimulus.

Affective state estimated using physiological signals presents a valuable additional channel of communication for use in robot control, safety monitoring, and for improving the perceived safety of a human-robot interaction. This additional communication modality can be useful for improving the robot responsiveness to the human participant, especially in cases where other channels of communication may be unavailable, for example in cases of disability. The use of affective state during direct human-robot interaction is described in Chapter 7. The following section describes the second robot perception modality developed in this work, namely, the estimation of the user's position and orientation through the use of machine vision.

6.2 Machine Vision for Human Monitoring⁸

Machine vision is a key technology for human robot interaction, and has been used extensively in many different applications, including human body tracking [104], head orientation [86, 87] and eye gaze tracking [84], and affective state estimation from facial expression analysis [88, 92]. In this work, a head tracking algorithm is used to accurately track the head location. Once the head location is known, the head orientation is estimated to determine if the human is able to see the robot. Head orientation rather than eye gaze tracking is used to estimate the user's focus of attention, because a close-up view of the user's face is generally required to estimate eye gaze direction, which may not be available in a human-robot interaction scenario. In addition, with head orientation estimation, it is possible to estimate a wide range of orientations [87], which is not possible with eye gaze tracking. The head location, together with 3 dimensional stereo camera information is also used to determine the 3D location of the human body. The head and body location information is passed to the robot controller and used for planning and control, as discussed in Chapters 4, 5 and 7.

6.2.1 Head Tracking

In order to improve the robot's ability to perceive the user, a head tracking system is required which will be capable of automatically detecting and tracking human faces in the robot's environment. The vision system should be able to recognize human faces oriented frontally, as well as faces that are partially turned, or in profile at different scales in the image.

Automated face tracking is divided into two components: initial face detection, and the subsequent face tracking.

6.2.1.1 Initial Face Detection

A combination of two features is used to detect faces in an image: skin colour and head shape. In order to extract skin coloured regions from the image, the colour RGB image is first converted into

Some of the work from this section of the thesis was implemented by undergraduate students under the supervision of the candidate [110 - 112].

Hue-Saturation-Value (HSV) space. Several researchers [105-107] have shown that skin colour occupies a very narrow region in the H-S plane, and that differences in value (intensity) account for differences in skin colour between light and dark pigmented skin. However, segmentation of skin regions using HSV space alone is not adequate, since many surfaces commonly found in the human environment will have colour similar to skin in the HSV space, such as wood, and also white walls in certain lighting conditions.

To reduce the number of falsely detected skin pixels, skin pixel segmentation is performed in two stages. In the first stage, a strong threshold is used to classify skin pixels. Then, a second pass is made through the image, using a weaker threshold to classify pixels that are adjacent to identified skin pixels. The two sets of thresholds are specified in Table 6.11. These values are derived from the single threshold applied in [106].

Table 6.11. Skin segmentation threshold values.

	Hue	Saturation
Strong Threshold	$0 < h < 30^\circ$	$0.23 < s < 0.68$
Weak Threshold	$-15^\circ < h < 45^\circ$	$0.23 < s < 0.68$

From all the identified skin coloured regions, those that do not satisfy size and aspect ratio requirements for a potential face are removed. The size requirement discards regions that cover less than 20 pixels of area. A maximum aspect ratio of 2.5 was allowed. The region bounding each of the remaining face candidates is searched to find an elliptical contour within the region with a high intensity gradient response. Figure 6.15 shows the sequence of steps performed during face detection on a sample image. The top left image shows the results from the pass of skin extraction, using the strong threshold. The top right image shows the skin regions after the weak threshold is applied. The bottom left image shows the intensity gradient of the image, and the bottom right image shows the extracted faces.

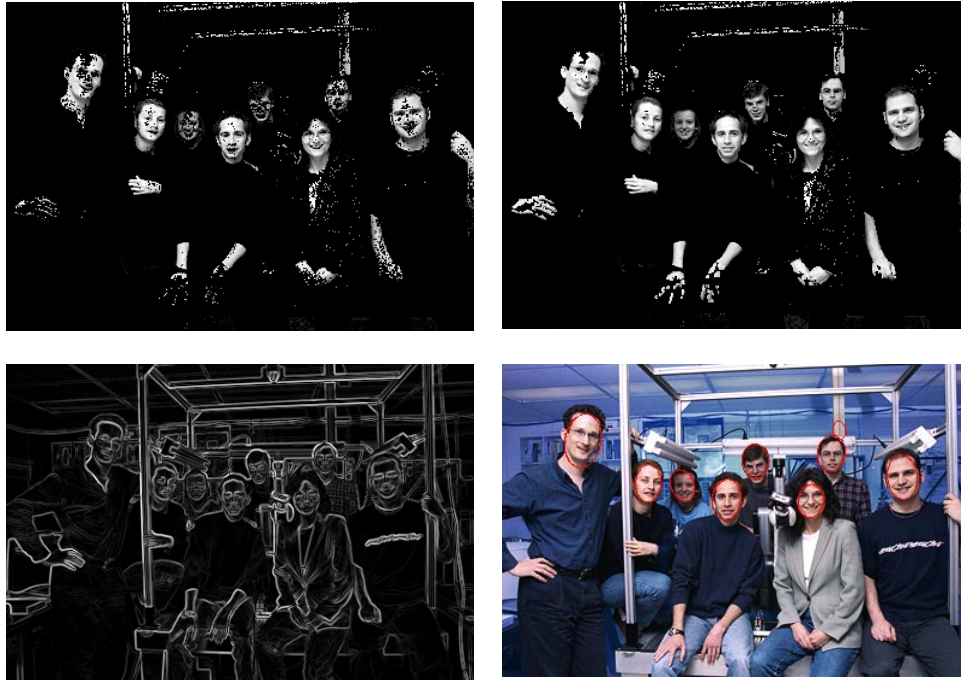


Figure 6.15. Face detection on a sample image.

Once the best fitting skin-coloured ellipse is found, a colour histogram is calculated from the region. The colour histogram is also based in HSV space, and uses 8x8 bins to collect colour distribution data in a two dimensional hue-saturation histogram structure. A separate 1-D histogram of the intensity is also generated with coarser granularity, using only 4 bins.

6.2.1.2 Face Tracking

Once faces have been detected in the initial frame, they are subsequently tracked using a probabilistic tracking algorithm based on a particle filter, similar to [108]. A deterministic tracking approach was initially implemented, using the algorithm found in [109], however, it was found that this tracker was easily distracted by skin coloured objects in the environment. The probabilistic approach allows the system to recover from a distracting object, by temporarily tracking both the target region as well as the distraction region. This system also allows tracking through temporary occlusions of the target.

A particle filter is a discrete implementation of the Monte-Carlo sequential tracking algorithm. The initial location of the tracking target is known within a certain probability distribution, and a model exists for predicting the motion of the target. The initial distribution is approximated by M samples from the continuous distribution (the particles). For each particle independently, a prediction is made for the next frame, based on the motion model. The new location of each particle is then evaluated based on the image data in the next frame. Weights are assigned to each particle, based on how well the new particle location corresponds to the image data. Based on these weights, the discrete distribution is re-sampled to provide the M set of particles to be used in the next frame. In the re-sampling step, new particles are assigned so that the number of new particles at each old particle location is proportional to the weight of that (old) particle. The expected value of the distribution is used as the estimated location of the target in each frame.

Particle Initialization

In order to utilize the particle filtering algorithm, an initial particle distribution must be provided. Although it is possible to initialize the algorithm with a random distribution, this approach may require many frames to lock in on the target, and may not be successful if there are many distracting objects in the image. To improve tracking in the beginning frames, the initial distribution is generated from the face location found in the first frame. 100 particles are generated, by creating small perturbations around the initial location.

Motion Model

Similar to [108, 109], a velocity prediction model is used to predict the motion of the particles. Each particle represents an ellipse. For each particle, the motion is characterized by the state equation (6.6).

$$\mathbf{x}_{t+1} = A_m \cdot \mathbf{x}_t + B_m \cdot \mathbf{x}_{t-1} + C_m \cdot rand(-1,1) \quad (6.6)$$

where \mathbf{x} is the state variable vector for each particle, the matrices A_m and B_m define the state dynamics, and C_m specifies the allowed random deviation. $rand(-1,1)$ is a function that generates a random value between -1 and 1 . Two different descriptions of the state variable vector were tested: the full description, including position, scale, aspect ratio and orientation ($\mathbf{x} = [x \ y \ S \ AR \ \theta_i]^T$), and a reduced description including only position and scale ($\mathbf{x} = [x \ y \ S]^T$). The vector $[x,y]$ denotes the position of the center of the ellipse, S represents the scale, encoded as the minor axis length of the ellipse, AR denotes the aspect ratio, and θ_i represents the orientation of the ellipse. When the reduced description was used, the aspect ratio was fixed at $AR = 1.3$, and the orientation was fixed such that the ellipse was vertical. For the x,y position, a constant velocity assumption is used; the scale, aspect ratio and orientation are assumed constant with some standard deviation. All variables are assumed to vary independently of each other. The values for A_m , B_m , and C_m are specified below. For the reduced state vector description, the same matrix values were used, with the matrices truncated.

$$A_m = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad B_m = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad C_m = [2 \ 2 \ 2 \ 0 \ 0.01]^T.$$

Goodness of Fit Evaluation

Once a new location for each particle is generated, the new location must be evaluated against the image data. The evaluation is performed using a combination of histogram and intensity gradient response. Using a combination of colour and shape allows the tracker to perform well in environments where there is skin coloured background, and also to be more robust to out of plane rotation of the head (i.e., rotation normal to the image plane), when only a small skin coloured region may be visible. The evaluation function is a weighted sum of the colour histogram response and the ellipse intensity gradient response. To compute the colour histogram response, the Bhattacharyya distance is calculated between the histogram enclosed by the ellipse of each particle and the model

histogram generated from the face detection algorithm in the first frame. The Bhattacharyya distance is defined as:

$$D(h_1, h_2) = [1 - \sum_{n=1}^{N_h} \sqrt{h_1(n)h_2(n)}]^2. \quad (6.7)$$

Here, h_1 and h_2 are the model and particle histograms, and N_h is the number of bins in each histogram.

The ellipse intensity gradient response is obtained by calculating the normalized sum of the gradient magnitude around the perimeter of the ellipse.

$$G = \frac{1}{N_s} \sum_{i=1}^{N_s} |\mathbf{g}_i|. \quad (6.8)$$

Here, \mathbf{g} is an array containing the intensity gradient response of all the pixels at the perimeter and N_s is the number of pixels at the perimeter.

The total response is then given by:

$$R_t = W_h D + W_g G. \quad (6.9)$$

In all the experiments reported herein, $W_h = W_g = 0.5$ were used.

Weight Assignment

Given the evaluation of each particle's location, weights are assigned to each particle. The weights are assigned by assuming that the total response (R) will decrease exponentially as \mathbf{x} moves away from the target location, such that

$$w_{t+1}^m = K \cdot \exp(\lambda \cdot R), \quad (6.10)$$

where w^m is the weight assigned to particle m , K is a constant such that $\sum w^m = 1$ and λ is the constant of proportionality. A similar assumption was made in [108] based on observations from successful tracking runs, and using only the histogram difference D , instead of R . In [108], the value of $\lambda = 20$ was used, based on experiments with labeled sequences. The same value of λ was used here also.

6.2.1.3 Head Tracking Experiments

The face detection and tracking system was initially implemented in Matlab, for ease of programming and debugging. Once the program was operational, the code was optimized and implemented in C to achieve real-time performance (~ 16 frames / sec) [110]. This section shows the initial results obtained with the Matlab code, results using the C code (integrated with the other vision components) are shown in Chapter 7.

Face Detection

The face detection module was tested separately from the tracking module on individual images before integrating the two modules. The face detection module exhibited excellent detection when the image background was not similar to skin colour (See Figure 6.15, Figure 6.16). Using the HSV colour space segmentation described in Section 6.2.1.1, cardboard, light wood, and white walls under weak illumination are sometimes classified as skin colour (See Figure 6.17). When the face was in front of a skin coloured background, detection can still be achieved if a good gradient exists at the edge of the head, for example, if the subject has dark coloured hair. However, because of the large skin coloured segment in which the ellipse edge must be detected, detection in this case can take much longer. The detection module fails if the background is skin coloured and a strong edge does not exist at the edge of the head. In this case, the intensity gradient search is attracted to the edges of the background object. The face detection module also fails if two faces are next to each other, such that they are segmented as one skin coloured region. In this case, only the face with the bigger intensity gradient at its edge is detected. This problem could be addressed by fitting a distribution over the gradient intensity search results, and testing if the distribution is multi-modal. If the distribution is multi-modal, a face candidate can be assigned to each mode.

Tracking

As described in Section 6.2.1.2, two types of tracking implementation were tested: one attempting to track the face location, scale and orientation, and one tracking only location and scale. The results

show that colour and contour information alone do not provide adequate information to track both scale and orientation. This can be seen in the two sequences in Figure 6.18 and Figure 6.19. In Figure 6.18, the full implementation is used, attempting to track orientation. As can be seen in this sequence, the intensity gradient is greatest at the edge of the top of the head, while the gradient at the chin tends to be weak. Therefore, as the head tilts sideways, an ellipse with a smaller scale, but with more of its edge on the top of the head will elicit a greater response than the correctly oriented ellipse. Figure 6.19 shows the tracking results from the same sequence, tracking only the location and scale of the ellipse, while keeping the orientation vertical.

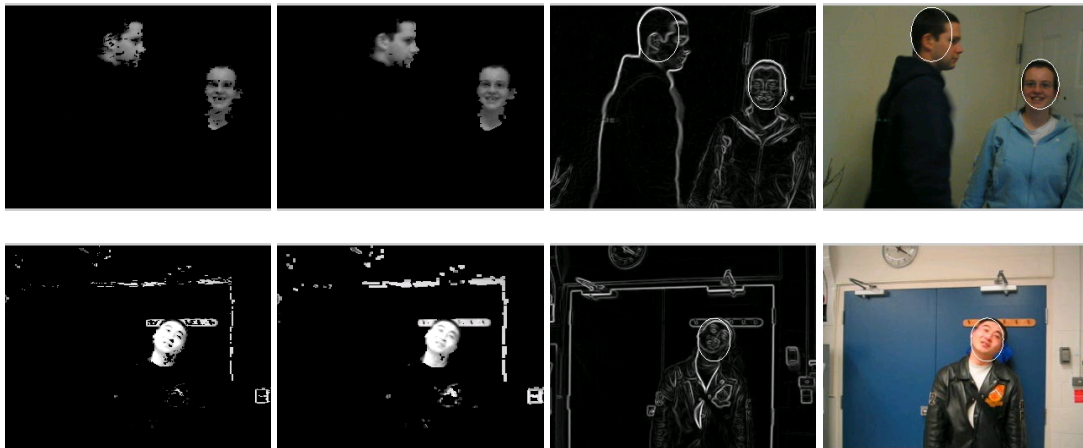


Figure 6.16. Face detection results.



Figure 6.17. Face detection with skin coloured background.

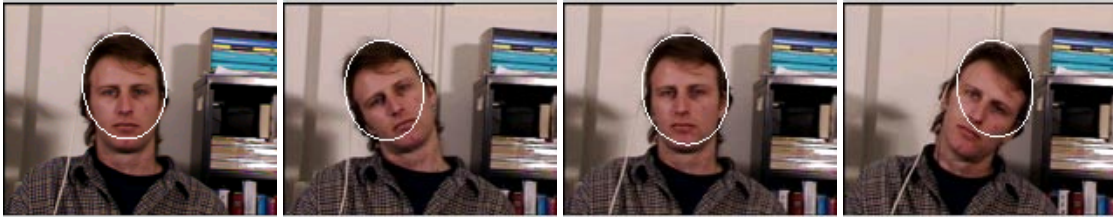


Figure 6.18. Tracking results when estimating the face location, scale and orientation.*

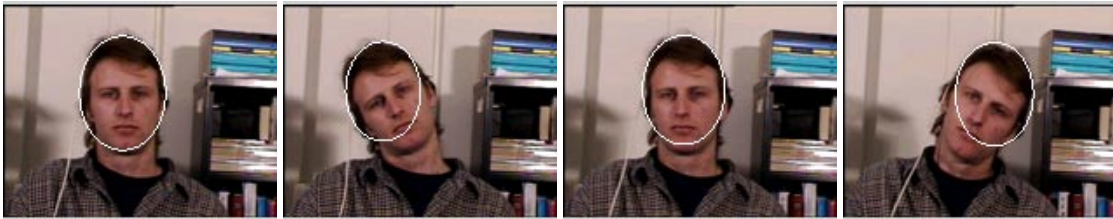


Figure 6.19. Tracking results when estimating the face location and scale only.



Figure 6.20. Tracking results under a variety of conditions.

Figure 6.20 shows the tracker performance under various conditions. As can be seen from these sequences, the tracker works well with arbitrary head rotation, through temporary occlusion as well as

* This image sequence was obtained from <ftp://csr.bu.edu/headtracking>

360° rotation. However, because the initial histogram model of the face is obtained automatically, the tracker performance is sensitive to the initial positioning of the head. The first two sequences show results when the initial positioning was with the subject facing towards the camera. In general, this type of positioning leads to the best results. In the third sequence, the female subject is facing towards the front of the camera when the initial histogram is generated, while the male subject is viewed in profile. The tracker can track the male subject well while he is in profile, but performance degrades when the subject faces the camera in sequence 4. This happens because the initial histogram model of the face was obtained when the subject was facing away from the camera (See Figure 6.16, Case 1). In this case, the histogram contains more hair than skin colour, so a portion of the face is a better match than the full face. This problem is addressed in the final system by having multiple histogram models of the person, since training data for a variety of orientations is obtained by the head orientation estimation module, as described in the following section.

6.2.2 Head Orientation Estimation

Once the head position has been obtained from a real-time video stream, the face orientation is estimated. This information can then be used to determine if the person is looking at the robot. The face orientation detection algorithm extracts the part of the image containing the face for each frame, handles the inaccuracy in the real-time tracking of face position and estimates the direction the head facing [111].

6.2.2.1 Head Orientation Estimation Algorithm

The algorithm is modified from the head-orientation estimation algorithm developed by Wu and Toyama [87]. The algorithm of Wu and Toyama is insensitive to illumination and able to estimate a wide range of head orientations from a low-resolution image. The use of low-resolution images allows variable distance between the robot camera and the person, and increases the speed of the algorithm, making it suitable for real-time estimation. The original algorithm described by Wu and

Toyama uses a 3-D ellipsoidal head model to represent the head pose, however, to simplify the calculations and reduce processing time, a 2D model is employed herein, as suggested by Brown and Tian [86].

Probabilistic Head Orientation Estimation

The algorithm of Wu and Toyama assumes that a cropped squared image of the face is given. The face image first undergoes a preprocessing phrase that scales the image, eliminates the background and enhances contrast. The image is first converted to gray-scale and rescaled to a 32 x 32 image using bilinear interpolation, as recommended by Brown and Tian [86]. Then, an elliptical mask is applied to the image to exclude any background and non-head portions of the image. The diameters of the scaled elliptical mask are 0.85 x 32 pixels on the vertical axis and $0.85 \times 32 / 1.2$ pixels on the horizontal axis. The masked image is histogram equalized to enhance contrast.

After the preprocessing phrase, 4 image-based-feature vectors are extracted from the masked image: (1) convolution with a Gaussian filter at scale 1, (2) convolution with rotationally-invariant Gabor templates with period 3.4 at scale 1, (3) period 6 at scale 2 and (4) period 10 at scale 4. To obtain rotationally invariant Gabor templates, the sum of 4 orientations is used, at 0, 45, 90 and 135 degrees. Because the convolution outputs are highly correlated with image brightness, each of the 4 feature vectors is normalized. The normalization improves performance under different illuminations. The 4 feature vectors (each one is a 32 x 32 vector) capture different characteristics of the face. As shown in Figure 6.21, the first feature (convolution with a Gaussian) provides an outline of a face and the second to the fourth features (convolution with Gabor templates) sharpen the profile of the face at different orientations.

The head-orientation space is discretized into 35 poses: every 30 degrees of vertical rotation (from -60 to 60 degrees inclusively) and horizontal rotation (from -90 to 90 degrees inclusively), where the frontal pose is 0 degrees both vertically and horizontally.

The algorithm requires training data set that contains a number of cropped squared face images for each head pose. The system is trained as follows: The face images are first pre-processed as described above and the 4 feature vectors are obtained for each masked image. For each pose, the mean and the variance of each pixel/element in each feature vector is calculated. Let $\mu_{\theta_{xy}}$ and $\sigma_{\theta_{xy}}^2$ be the mean and the variance of the pixel at position (x,y) in the feature vector f for pose θ . Then, the head-pose model calculated is a set $\{\mu_{\theta_{xy}}, \sigma_{\theta_{xy}} : \forall \theta, f, x, y\}$. Figure 6.21 shows the average images ($\mu_{\theta_{xy}}$) of the 4 feature vectors for 3 different poses. θ_v and θ_h indicate the degree of vertical and horizontal rotation, respectively.



Figure 6.21. Average images of the 4 feature vectors (from left to right) for 3 different poses: $\theta_v=0, \theta_h=0$ (left 4), $\theta_v=30, \theta_h=30$ (centre 4) and $\theta_v=60, \theta_h=0$ (right 4).

To estimate the head orientation from a face image, the image is pre-processed and the 4 feature vectors, z , are extracted. The estimated pose is determined by the maximum a posteriori pose given z using Bayes rule:

$$\theta^* = \arg \max_{\theta} p(\theta | z) = \arg \max_{\theta} \frac{p(z | \theta)p(\theta)}{p(z)}. \quad (6.11)$$

Assuming $p(z)$ and $p(\theta)$ are constant, then (6.11) becomes:

$$\theta^* = \arg \max_{\theta} p(z | \theta) \quad (6.12)$$

Next, the pixels and the features are assumed to be independent. Equation (6.12) then becomes:

$$\theta^* = \arg \max_{\theta} \prod_f \prod_x \prod_y p(z_{,fxy} | \theta) \quad (6.13)$$

Assuming each pixel is normally distributed, a logarithm of (6.13) results in:

$$\begin{aligned}
\theta^* &= \arg \max_{\theta} \sum_f \sum_x \sum_y - \frac{(z_{fxy} - \mu_{\theta fxy})^2}{\sigma_{\theta fxy}^2} \\
&= \arg \min_{\theta} \sum_f \sum_x \sum_y \frac{(z_{fxy} - \mu_{\theta fxy})^2}{\sigma_{\theta fxy}^2}
\end{aligned} \tag{6.14}$$

Partial Template Matching

The head orientation estimation is highly dependent on the face tracking algorithm. If the face tracker consistently gives the same cropped face image for each face orientation, then the algorithm described above will perform as reported in Wu and Toyama [87]. However, the face tracking algorithm does not always return an accurate location of the face. Therefore, the pose estimation algorithm can be improved by taking into consideration the behavior of the face tracker.

Since the face tracker is based on a probabilistic particle filter, the tracked face location fluctuates around the real face location. Figure 6.22 shows a series of cropped frames from the tracked face. As can be seen from these images, the real face is not always at the centre of the cropped image. Generally, the deviations of the face tracker can be categorized into 3 types as shown in Figure 6.23. Because of these deviations, the assumption that $p(z)$ is constant becomes invalid.



Figure 6.22. A series of cropped frames from the face tracker.

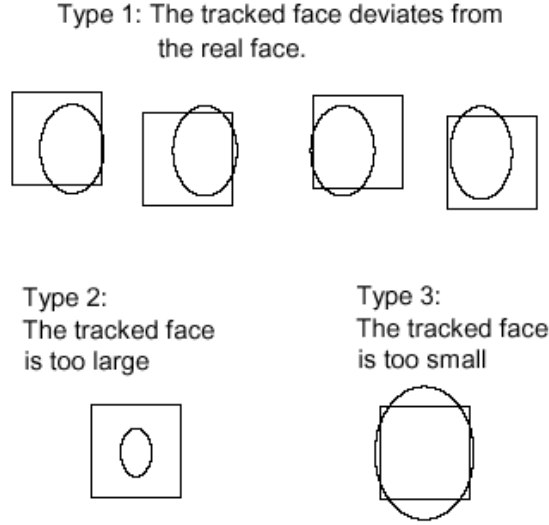


Figure 6.23. Three types of deviations of the face tracker.

To deal with the deviations from face tracker, a partial template matching method is used in the estimation. The estimation in Section 6.2.2.1 is applied not only to the cropped face image given by the face tracker, but also to other cropped face images deviating from the tracked face location. This estimation is applied on 11 cropped face images: (1) the tracked face location, (2)-(9) 3 pixels away from the tracked face location (one image each 45 degrees), (10) the tracked face area scaled down by 1.1 and (11) the tracked face area scaled up by 1.1. The cropped images (2)-(9) address the type-1 deviation, and the cropped images (10) and (11) address the type-2 and type-3 deviations respectively. Let z_{fxy}^i be the 4 feature vectors extracted from the cropped image i . The head pose estimation becomes:

$$(\theta^*, i^*) = \arg \min_{\theta, i} \sum_f \sum_x \sum_y \frac{(z_{fxy}^i - \mu_{\theta fxy})^2}{\sigma_{\theta fxy}^2}, \quad (6.15)$$

where θ^* is the estimated pose and i^* is the cropped face image that contains the face.

6.2.2.2 Experimental Results for Head Orientation Estimation

For each head pose, 100 cropped face images from the face tracker were collected for model training using a real-time video stream. All face images come from a single person. To minimize the background noise in the model training, a dark screen was placed behind the test subject to cover all the background. This improves the performance of the face tracker. The actual head orientation is determined by the test subject, with the aid of marks on the floor and wall to indicate the rotated positions. The average images of the 4 feature vectors for 3 different poses shown in Figure 6.21 are taken from the trained model.

The head orientation estimation algorithm can correctly estimate the head orientation for all head poses most of the time. For instance, among the 209 frames that were saved during the experiment, only in 20 frames the estimated head pose wrong. Figure 6.24 shows 3 head-frame series taken from these 209 frames. Note that the real head orientation is determined manually from the saved face image, which may cause a small error. For example, in the frame series 1 in Figure 6.24, the estimated pose for the third image is determined to be wrong, while the estimations for other images in the same series are determined to be correct.

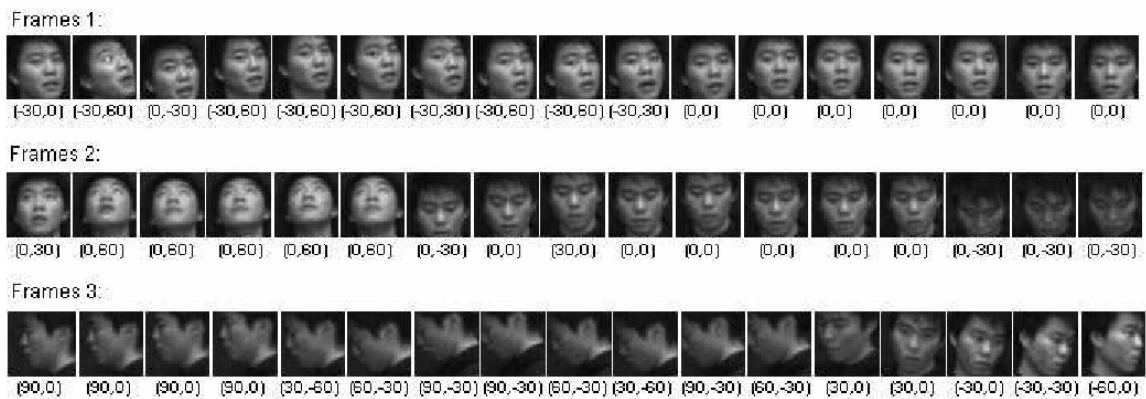


Figure 6.24. Three head-frame series taken from the first experiment and the estimate poses are shown in the parentheses $[\theta_h, \theta_v]$.

The algorithm can estimate a wide range of head orientation, as shown in the frame series 3 in Figure 6.24, but error increases in side view. By comparing series 2 with series 3 in Figure 6.24, the estimation error is observed to be higher for the side view than the frontal view. In general, the error

rate for near side view ($\theta_h = +/- 60/90$) is around 20% and for near frontal view ($\theta_h = +/-30/0$) is roughly 5%.

The last few images in series 3 also show that the algorithm works well even for a quick change of head orientation as long as the face is tracked, because a constant $p(\theta)$ is used.

The combined head tracking and head orientation tracking executes at ~8 frames/second.

6.2.3 Body Tracking

Once the head location is known in the image frame, a 3D location of the head and body is computed using the stereo camera. The body is represented by a set of spheres representation [77]: one sphere to represent the head, one sphere to represent the torso, and additional spheres to represent the hands, if the hands are visible in the scene.

The Bumblebee stereo camera [66] provides firmware for calculating the disparity between the two camera images. The camera outputs a point cloud listing all the pixels and their corresponding x,y,z locations in the camera frame for those pixels where a valid correlation and disparity have been found. Given the head location pixels, the 3D location for the head is calculated by averaging the x,y,z co-ordinates of all the head location pixels that have a valid disparity generated from the stereo routines. The radius of the head is generated from the major axis of the head ellipse, provided by the head tracking module.

Once the head location is known, the body location is calculated by assuming that the person is seated or standing, such that their body is vertically below the head. It is assumed that the body occupies the entire region vertically below the head, reaching to the bottom of the frame. The radius of the sphere enclosing the torso is therefore calculated as half of the distance from the bottom of the head sphere to the edge of the frame.

Hand locations are determined using colour segmentation. Hand tracking was initially implemented using skin colour segmentation [112], however, it was found that this method was prone to error if the hand location crossed the head location in the image plane. The blob tracking method

implemented [104] tended to join the two skin coloured blobs, so that the hand could not be tracked accurately for arbitrary hand movements. To ensure that the hand location was always accurately tracked, subjects were asked to wear blue gloves during testing, and the environment was controlled to eliminate other blue objects. The location of the blue blob was segmented using colour segmentation in the normalized RGB space [113]. Pixels corresponding to the blue blob were then located in the stereo point cloud, and the hand sphere(s) were generated by calculating the average x, y, z locations of the pixels corresponding to the blue blob. Figure 6.25 shows a sample screen capture of the body tracking output. The raw camera output with a projection of the enclosing spheres is shown in the left image, while the 3D spheres representing the head, body and hand are shown in the right image, overlaid onto the stereo camera generated point-cloud.

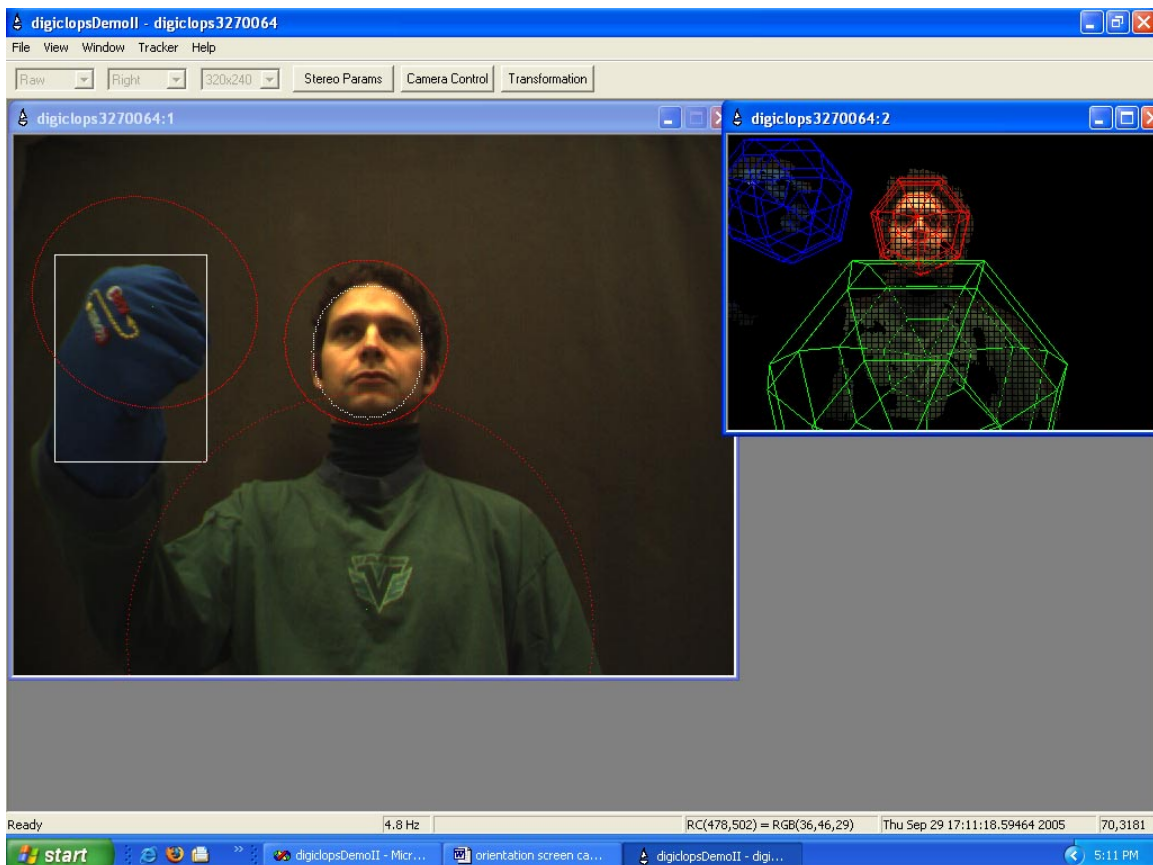


Figure 6.25. Body tracking screen capture.

6.2.4 Summary

The vision system generates an estimate of the user location and head orientation. The head location is tracked using a particle filter based on an elliptical shape and colour model. Given the head location, the head orientation is estimated using partial template matching. The 3D location of the human body in the workspace is estimated using the stereo information generated by the camera firmware and the known head location. Colour based segmentation is used to detect and track the location of the hand(s). The human location and head orientation estimates are used during planning and control, as described in the following Chapter.

Chapter 7: System Integration

This chapter describes the integration and overall system testing of the human-robot interaction strategies described in the preceding chapters. The methodology for the use of human monitoring information to improve safety at both the planning and control stages is described. Next, the integrated system is tested through case studies, to demonstrate the operation of the system under various conditions.

7.1 Integrating Human Monitoring Data for Safe Planning and Control

The objective of the integrated system is to improve the safety of human robot interaction by reducing both the potential for a collision to occur, and reducing the impact force in the event of a collision. The strategy is to design and modify the motion prior to the occurrence of a collision. This objective is accomplished through three concurrent planning processes, each considering a different time frame horizon. The safe path planner, described in Chapter 4, considers the long-term safety strategy; namely, the robot path is planned prior to the start of the interaction to minimize the potential impact force along the path. This strategy places the robot in a better position to respond, should an unexpected hazard occur during real-time execution. Once task execution begins, velocity scaling along the planned path, as described in Section 5.1, is used to address potential safety hazards that appear while the robot motion is in progress, but that are far enough away to give the robot time to decelerate to a safe approach speed along the planned path. This velocity scaling constitutes the medium-term strategy. Once the robot is close to the human, there will be very little time to react, should a sudden hazard occur. In this short time frame, the reactive safety controller, described in

Section 5.2, is used for those cases where an imminent hazard is present, and the robot must deviate from the planned path to ensure the safety of the user.

Human monitoring information is used to enhance the robot's knowledge of the user's response (or lack of awareness) to the interaction. Since this information is not available prior to the start of motion, human monitoring information can only be incorporated into the medium and short term planning, i.e., by using the human monitoring information during trajectory scaling and safety controller response. The user's behavior during the interaction has important implications for the safety of the interaction; therefore, this information is used to modulate the estimate of the danger index.

7.1.1 Danger Index Modulation

The danger index describes the current level of danger in the interaction, and is used to decide when the robot needs to take corrective action and modify its course, as described in Chapter 5. During subcomponent testing described in that chapter, the danger index was calculated based only on the physical location and velocities of the robot and user. However, if user monitoring information is available, the danger index can be improved by including this information. For example, analysis of industrial robot accidents has found that the majority of accidents in industry occur when the human operator is not aware of the robot's motion [13]. Therefore, the danger index should be increased if the human is facing away from the robot, and is therefore less likely to be observing and aware of the robot motion.

In this implementation, two indices are used to modulate the danger index: the user head orientation and affective state. Similarly to the kinematic factors comprising the danger index, the user status components are formulated as scaling factors. The user status components only affect robot behavior if the potential for a hazard already exists based on the kinematic danger factors.

The head orientation scaling factor is formulated as a sigmoid function based on the horizontal angle ("pan angle") of the head. The orientation of 0 degrees indicates that the user is oriented facing

the robot. The sigmoid function is used to ensure a smoothly differentiable scaling factor, and to adjust for differences in visibility at different head orientations. For example, in the -15 to $+15$ degree range, changes to head orientation should not significantly impact the danger index, as robot visibility is not affected. Similarly, in the $60 - 90$ degree range, the scaling factor should be close to the maximum, as the robot will not be fully visible from this orientation. The factor changes most rapidly in the $15 - 60$ degree range, where changes to robot visibility occur. The scaling factor is given by:

$$K_{OR} = 1 + \frac{M_{OR}}{1 + e^{-S_{OR}(\theta_h - \theta_c)}} \quad (7.1)$$

K_{OR} is the scaling factor due to head orientation, M_{OR} is the maximum increase in scaling due to the orientation, S_{OR} controls the slope of the sigmoid function, θ_h is the horizontal head orientation, and θ_c is the head orientation at the midpoint of the sigmoid. $M_{OR} = 2$, $S_{OR} = 0.2$, and $\theta_c = 30$ degrees were used in all the test cases described in Section 7.2.

Similarly, for the user affective state, a sigmoid function was used to generate the scaling factor. The arousal measure was used as the estimate of the affective state, as the arousal was shown to be strongly correlated to the user reported anxiety and surprise, as described in Chapter 6. Since the affective state estimation is less accurate at low arousal levels, the sigmoid function is used to filter out spurious arousal responses of low magnitude (arousal level less than 0.3). The scaling factor due to affective state is given by:

$$K_{AS} = 1 + \frac{M_{AS}}{1 + e^{-S_{AS}(a - a_c)}} \quad (7.2)$$

K_{AS} is the scaling factor due to affective state, M_{AS} is the maximum increase in scaling due to the affective state, S_{AS} controls the slope of the affective state sigmoid function, a is the current level of arousal, and a_c is the midpoint of the arousal scale. $M_{AS} = 2$, $S_{AS} = 20$, and $a_c = 0.5$ were used during all of the experiments.

Once the scaling factors were computed, the danger index was formulated as follows:

$$DI_t = K_{OR}K_{AS}DI, \quad (7.3)$$

i.e., the danger index is a product of physical interaction factors affecting the potential impact force, degree of user awareness of the robot, as estimated by the head orientation, and level of user anxiety. The product of factors formulation ensures that corrective action is only taken by the robot controller when all the conditions for a hazard are present. This means that the robot can continue to operate at maximum velocity far away from the user, regardless of the user awareness of the robot or the user's affective state. Once the robot gets close to the user, lack of awareness, or increased affective arousal will modify the robot behavior to ensure user safety. The robot behavior can be modified by activating the safety controller in the case of an immediate hazard, or, if more time is available, through trajectory scaling.

7.1.2 Trajectory Scaling

The velocity along the planned path can be scaled dynamically during task execution, as described in Section 5.1. Trajectory scaling is used to slow the robot down as it is approaching a potential hazard, for example, when it approaches a person. By reducing the velocity down to zero, the robot can also handle temporary obstructions of the planned path. The scaling constant r (see the Equation (5.1)) is calculated in real time, and is based on the integrated danger index,

$$r = V_{max} - K_D DI_t, \quad (7.4)$$

where V_{max} is the normalized maximum desired velocity (where $V_{max} = 1$ is the physical robot maximum), K_D is a scaling constant, and DI_t is the modulated danger index as described in the previous sub-section. The value of r is limited to the range [0,1].

The use of the danger index for both trajectory scaling and trajectory modification allows for a smooth transition between the planned motion and a reactive trajectory in case of a hazard. When a hazard is first identified, the danger index will increase, thus lowering the velocity along the path. If the reduced velocity does not lower the danger index, then the hazard is likely caused by changes in the external environment, and a change in path is required.

The user head orientation and affective state are therefore used to modify both the short and medium term safety strategies. Lack of awareness by the user or agitated affective state will result in the robot slowing down sooner, and reacting quicker to a potential hazard, thus improving user safety.

7.2 Case Studies⁹

Once the system was implemented on the test-bed, as described in Chapter 3, case studies were executed to test and demonstrate the behavior of the system components during direct human-robot interaction. The experimental protocol for the case studies was reviewed and approved by the UBC Behavioral Research Ethics Board.

Velocity Scaling Test Case

The Velocity Scaling Test Case shows the effect of localized velocity scaling. In this case, the robot is executing a pick and place task with maximum velocity ($V_{max} = 1$). The pick location is to the right of the user. Figure 7.1 shows sample frames taken from the video sequence for this test case. Figure 7.2 shows the danger index during task execution. Figure 7.3 shows the trajectory scaling factor during the motion. Figure 7.4 shows the resulting trajectory of the first three robot joints. Note that the motion begins 7.4s following the start of data acquisition. The motion is planned using the safe planner. The safe planner maximizes the robot distance away from the user by reaching backward towards the pick location, as shown in Figure 7.1(b,c). At locations in the path where the robot nears the user, the planner reduces the robot inertia by bringing the body towards the robot base, as shown in Figure 7.1(c). The robot reaches the pick location in Figure 7.1(d). The first half of the motion is executed at full velocity, as can be seen in Figure 7.3. As the robot approaches the user to access the place location (directly in front of the user), the inertia along the planned path is again lowered, as seen in Figure 7.1(e). As the robot approaches the user, the danger index increases, as shown in Figure 7.2, resulting in a decreased velocity along the trajectory, due to the trajectory

⁹ Videos of the experiments can be viewed at www.mech.ubc.ca/~dana/thesis.html

scaling. As the robot approaches the person, the velocity initially decreases rapidly, and then levels out to zero as the robot approaches the target position, as shown in Figure 7.1(e-h). The danger index and the trajectory scaling factor are quite noisy, due to the noise from the human position tracking. The danger index is not filtered; it is used to identify a hazardous situation, and a filter would introduce a delay. However, once the velocity scaling factor is passed to the trajectory planning module, the scaling factor is filtered to ensure that changes in time scaling meet the maximum velocity, acceleration/deceleration and jerk constraints of the robot. This results in a smooth command trajectory, as shown in Figure 7.4.

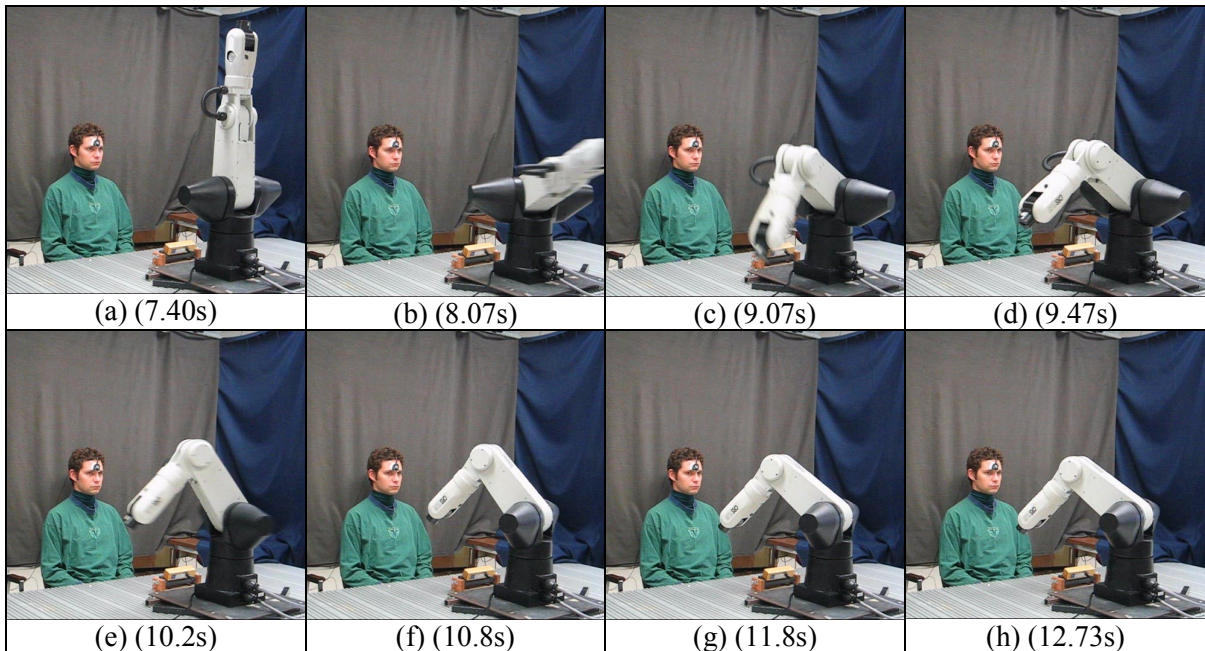


Figure 7.1. Velocity Scaling Test Case.

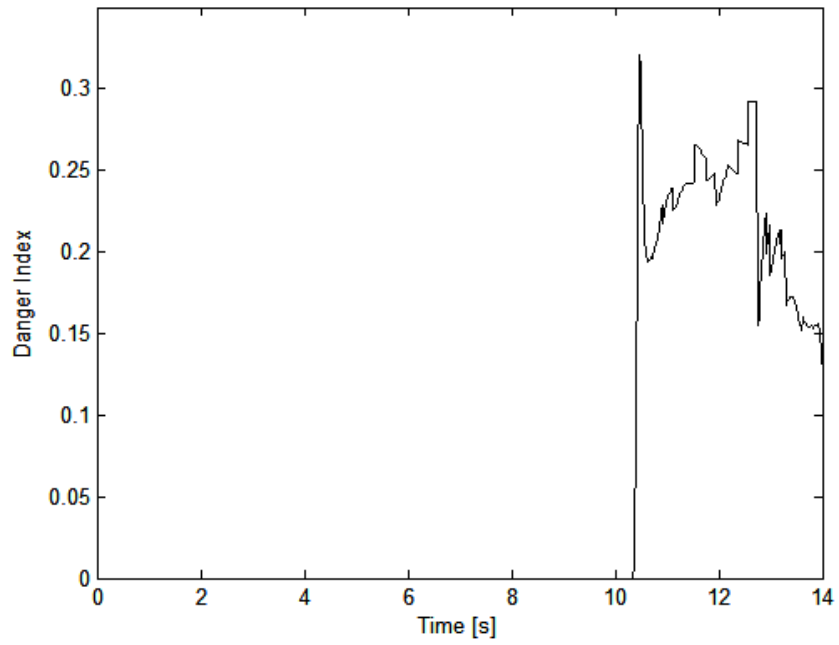


Figure 7.2. Velocity Scaling Test Case danger index.

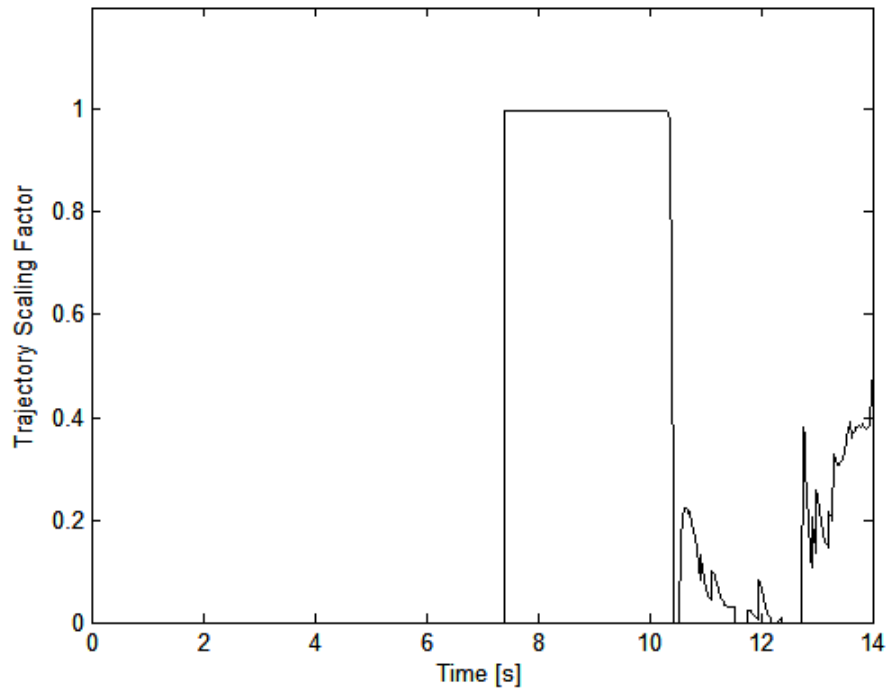


Figure 7.3. Velocity Scaling Test Case trajectory scaling factor.

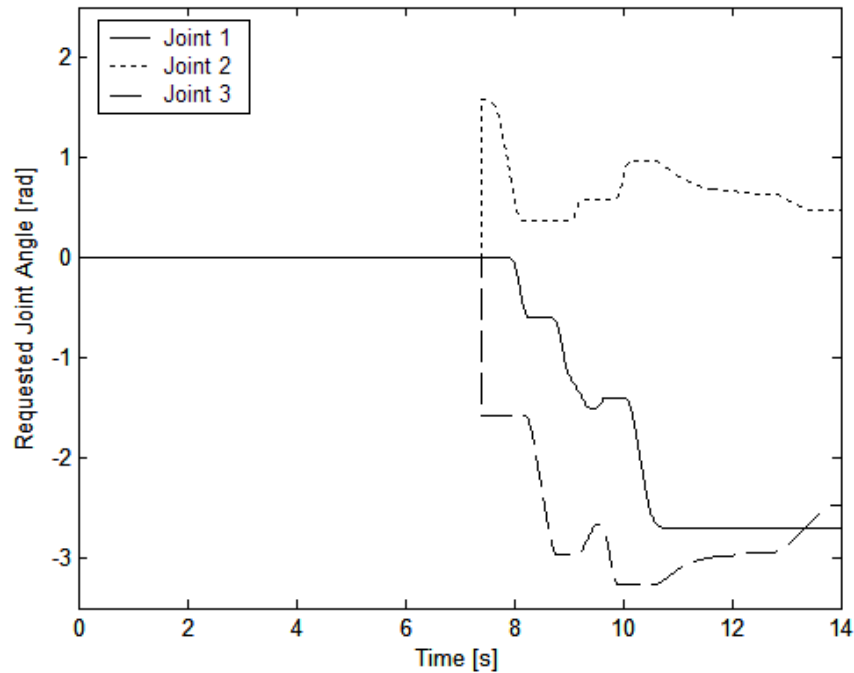


Figure 7.4. Velocity Scaling Test Case reference trajectory.

Obstruction Test Cases

The next two case studies demonstrate the vision based user position tracking, and illustrate the interaction between the trajectory scaling and the reactive safety module. In both Obstruction Test Case 1 and 2, the user moves his hand to obstruct the robot’s planned path. In the first case, the user moves while the robot is still far away from the potential collision point, so that the robot has enough time to decelerate to a stop a safe distance away from the user while staying on the planned path. In this case, once the user removes the obstruction, the robot can continue with the planned motion. Because the user is asked to move into the robot’s path, the maximum normalized velocity was lowered to 0.35 for these test cases, to ensure safety of the user.

Figure 7.5 shows selected frames from the video sequence for the Obstruction Test Case 1. Note that the start of robot motion occurs 7.15s after the start of data acquisition. The robot’s task is to approach a position above the table directly in front of the user, simulating a pick-up task. The

robot's path is planned using the safe planner. Once the safe path is generated, the robot begins executing the task, initially at the maximum normalized velocity, in this case 0.35. The robot first lowers its inertia while staying away from the user, and then approaches the user once the geometric danger criterion has been lowered, as described in Chapter 4. Figure 7.6 shows the danger index during task execution. Figure 7.7 shows the trajectory scaling factor, and Figure 7.8 shows the resulting requested trajectory for the first three joints. As the robot starts approaching the user, the user moves his hand such that it obstructs the robot's planned path (see Figure 7.5(b,c)). Figure 7.9 and Figure 7.10 show the hand and body tracking results from the vision system at the start and at the end of hand motion, respectively. The trajectory scaling factor decreases to zero, and the robot comes to a stop a safe distance away from the user (Figure 7.5(d)). Once the user removes his hand from the path, the robot continues to finish execution of the path. Note that the robot's velocity is reduced as it approaches the user.

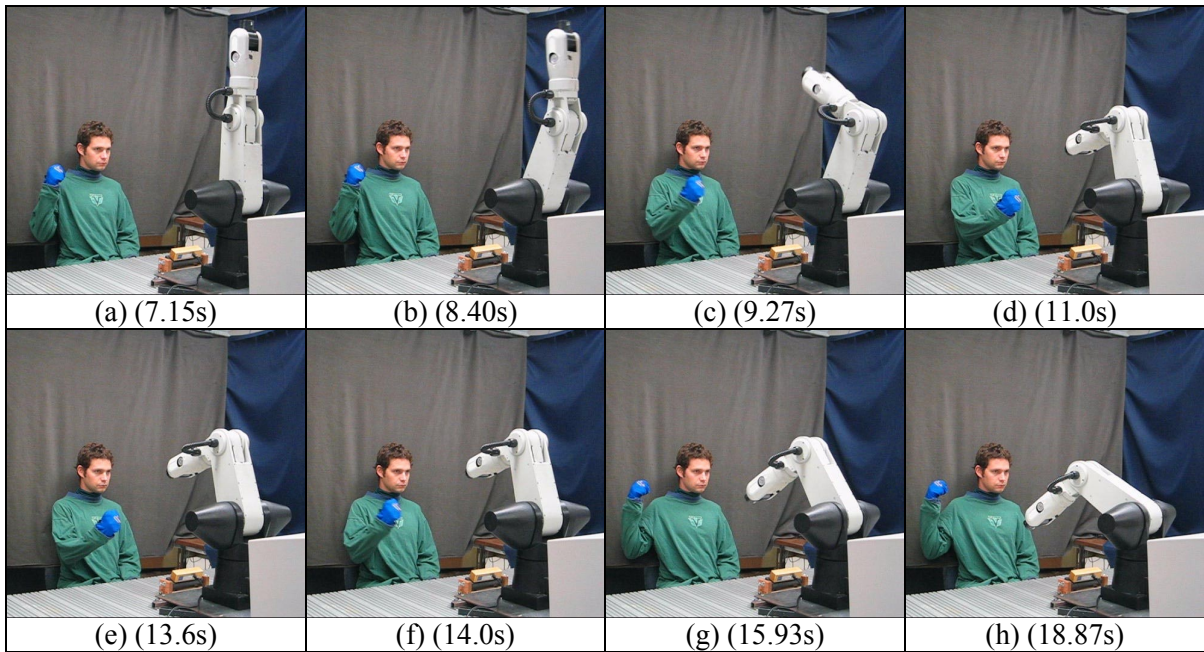


Figure 7.5. Path Obstruction Test Case 1.

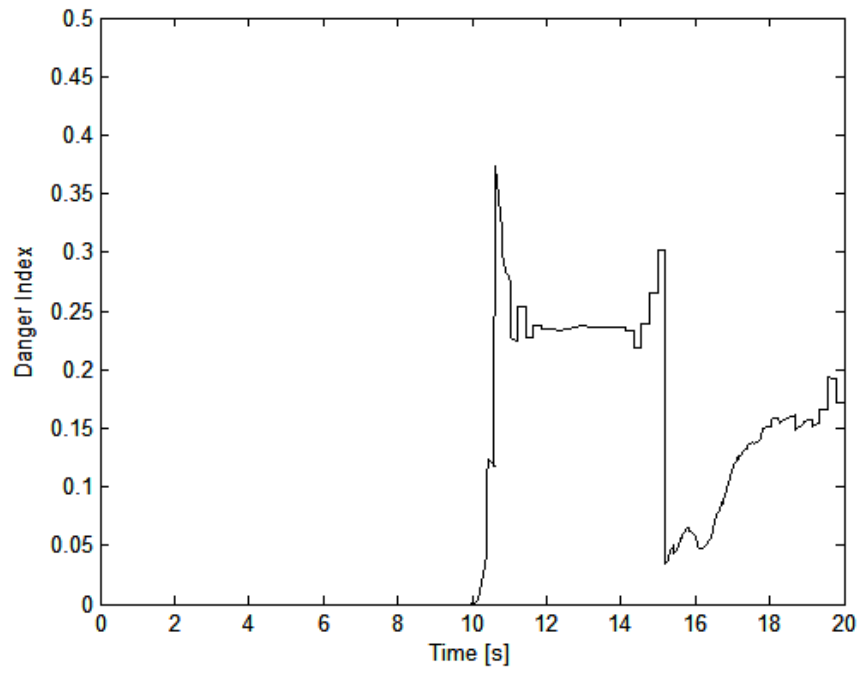


Figure 7.6. Obstruction Test Case 1 danger index.

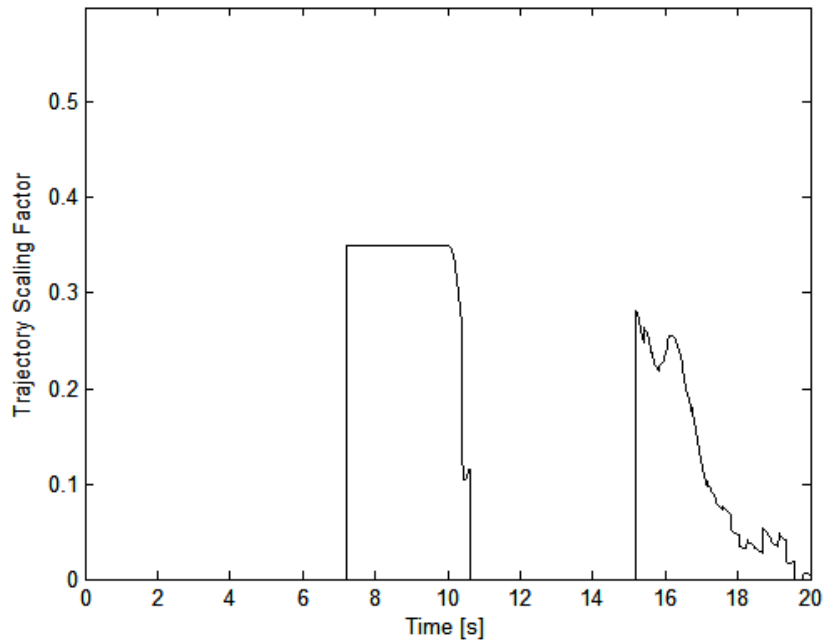


Figure 7.7. Obstruction Test Case 1 trajectory scaling.

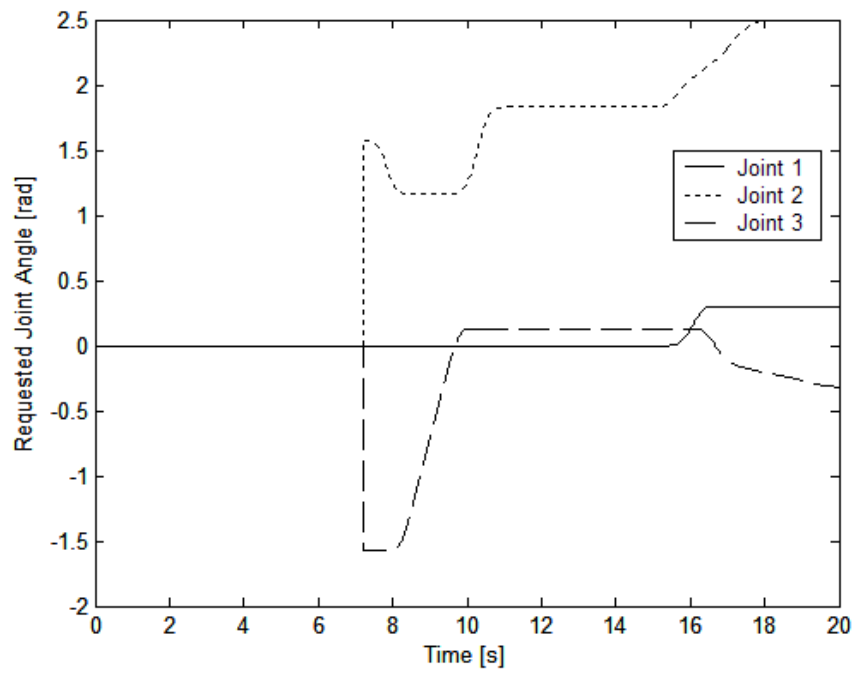


Figure 7.8. Obstruction Test Case 1 reference trajectory.

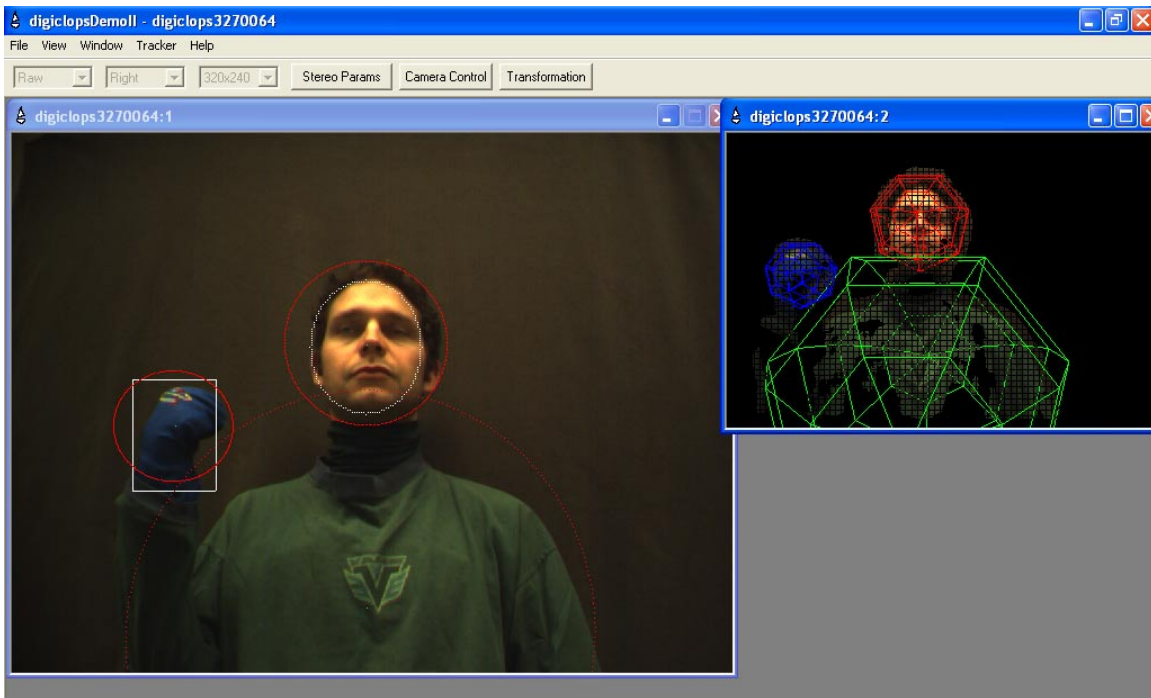


Figure 7.9. Hand and body tracking results (starting hand position).

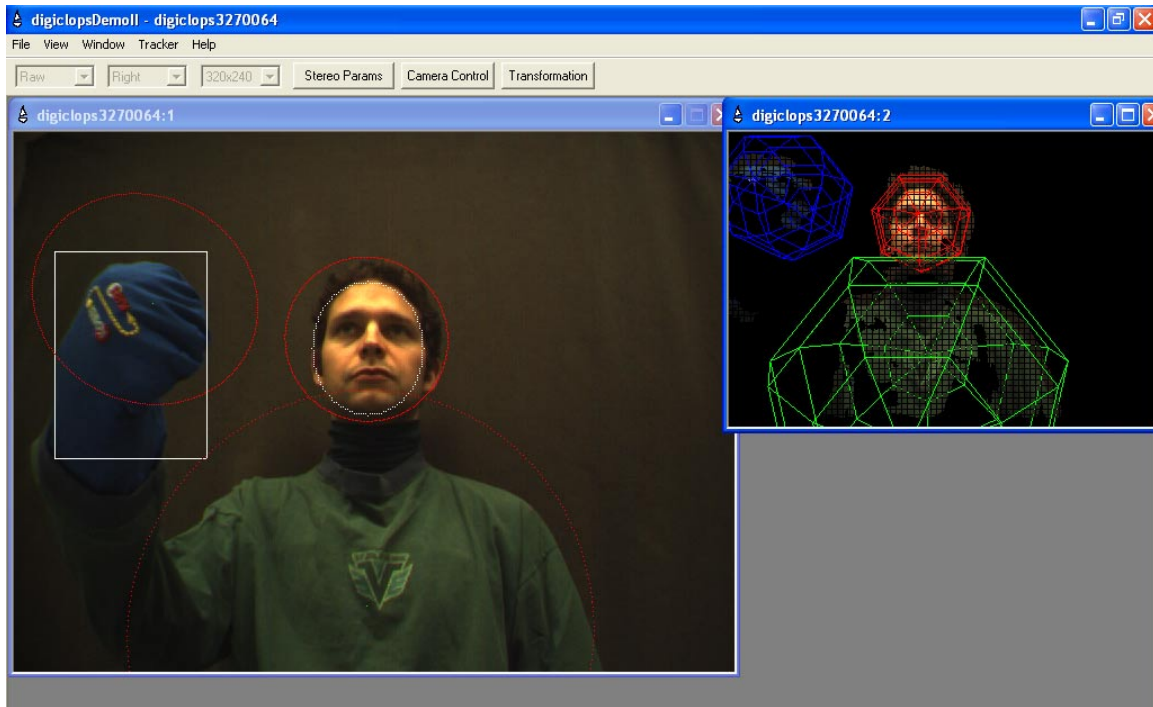


Figure 7.10. Hand and body tracking results (ending hand position).

In Obstruction Test Case 2, the user moves to obstruct the robot's path when the robot is already close to the user. In this case, there is not enough time for the robot to decelerate along the planned path and still maintain a safe distance from the user. The safety module is activated and generates a new path pushing the robot to a safe location away from the user.

Figure 7.11 shows selected frames for the video sequence for this test case. Figure 7.12 shows the danger index during the test case. Figure 7.13 shows the trajectory scaling. Figure 7.14 shows the commanded trajectory for the first three joints. The robot motion begins 6.07s after the start of data acquisition. The planned path is the same as in the Obstruction Test Case 1. Similarly to the previous case, the user moves his hand to obstruct the robot's path, as shown in Figure 7.11(b,c). However, the user moves his hand later in the path, so that it is not possible for the robot to decelerate to a stop a safe distance away from the user. This can be seen from Figure 7.12 and Figure 7.13. Even though the velocity is scaled to zero, the danger index continues to climb, as the robot cannot slow down fast enough along the planned path to maintain a safe distance. Once the danger index climbs above the

safe threshold, the safety module generates an alternate trajectory seeking to minimize the danger index, as described in Chapter 5. The safety module is activated at 10.28s into the experiment. Once the safety module is activated, it acts to minimize the danger index, as shown in Figure 7.12. The trajectory generated by the safety module moves all three lower joints, as shown in Figure 7.14, and in Figure 7.11(f-h).

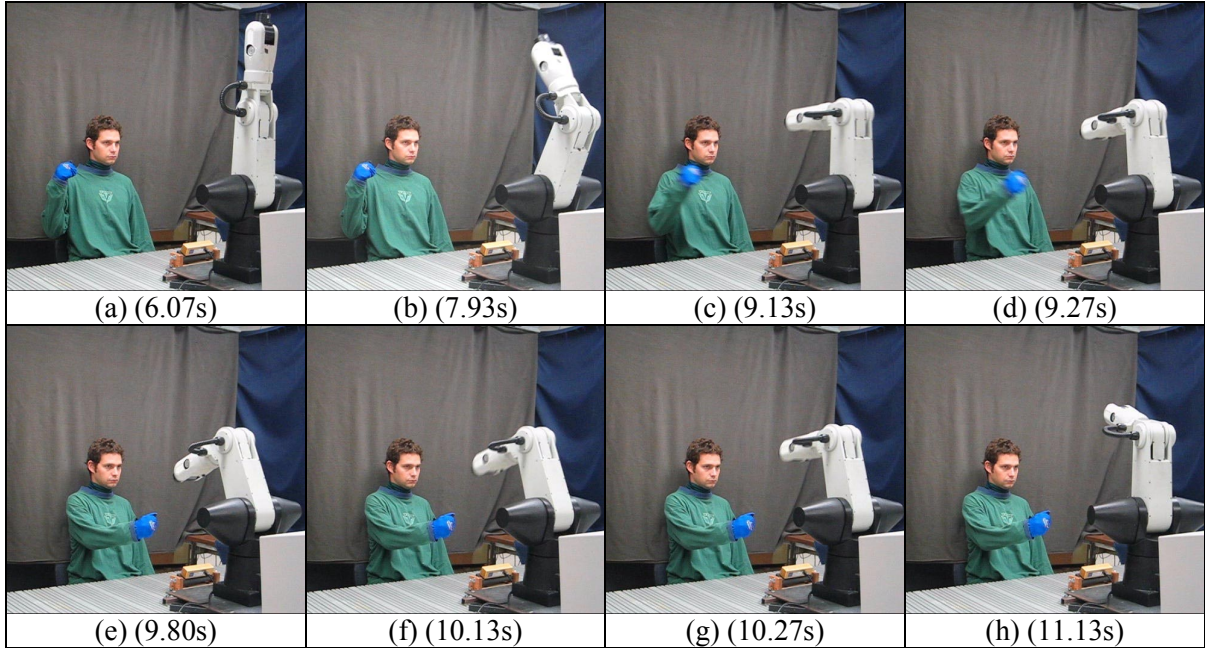


Figure 7.11. Path Obstruction Test Case 2.

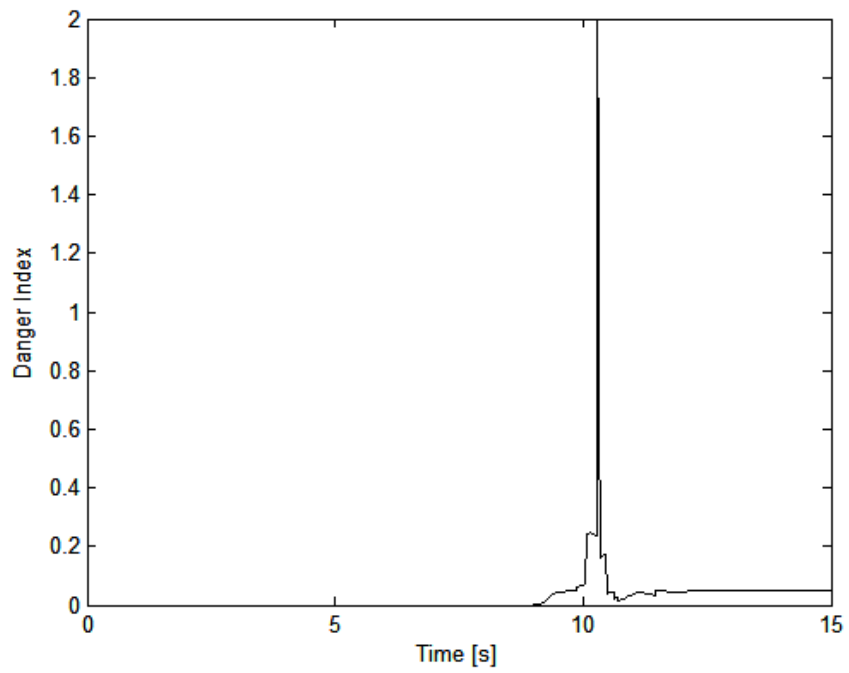


Figure 7.12. Obstruction Test Case 2 danger index.

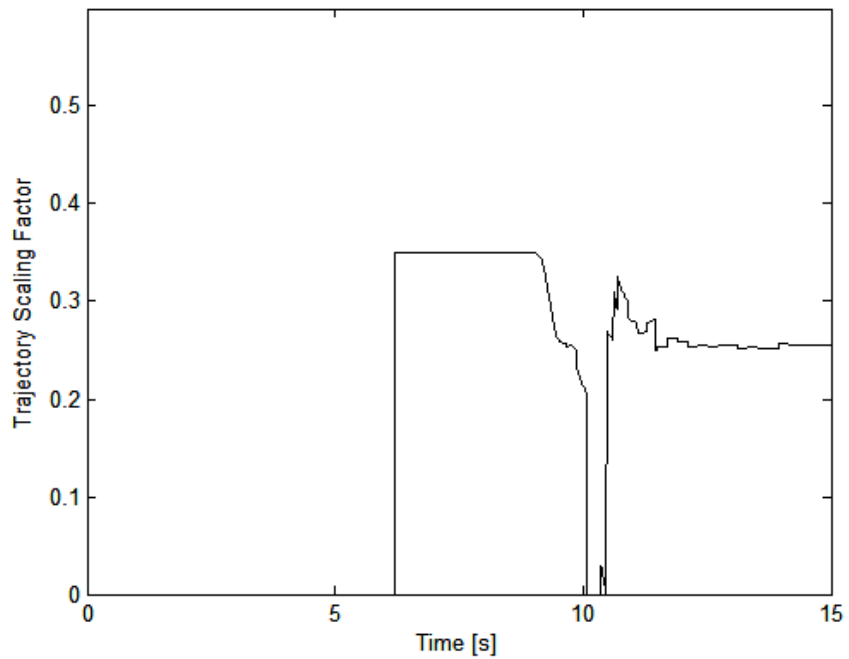


Figure 7.13. Obstruction Test Case 2 trajectory scaling.

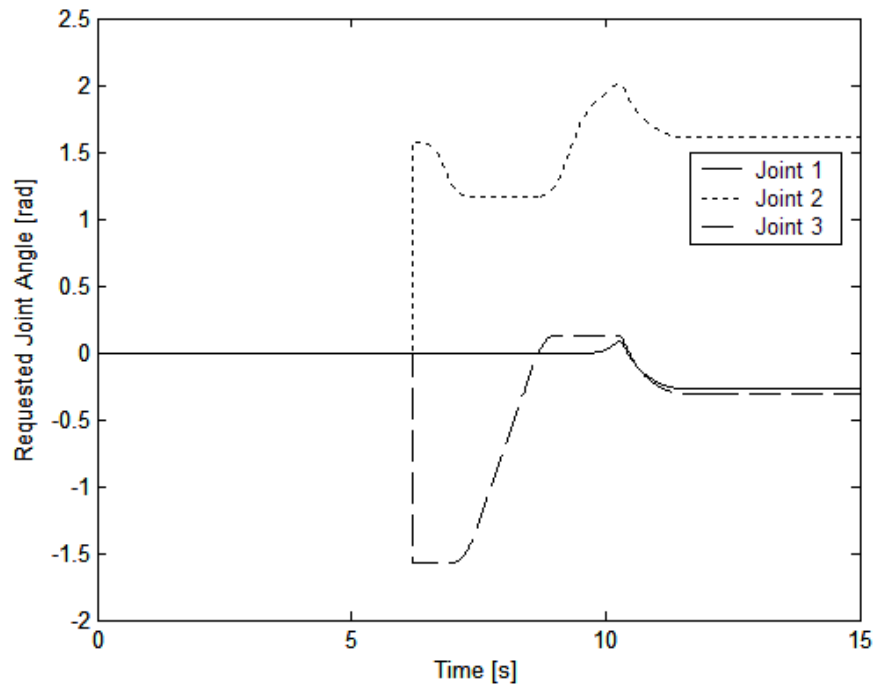


Figure 7.14. Obstruction Test Case 2 reference trajectory.

Affective State Test Case

The Affective State Test Case demonstrates the impact of the user affective state. The affective state was estimated using the fuzzy inference engine described in Chapter 6. Prior to the start of the experiment, baseline data was collected for the subject. This baseline data was then used to normalize the physiological signals prior to their input into the fuzzy inference engine, as described in Section 6.1.1. The same pick-up task as above is used, however, the normalized maximum robot velocity is set to 0.85, in order to elicit a strong response from the user. Figure 7.15 shows sample frames from the video sequence taken during the test. Figure 7.16 shows the integrated danger index during the test case. Figure 7.17 shows the velocity scaling, and Figure 7.18 shows the resulting joint trajectory. Figure 7.19 shows the level of arousal estimated during the test case. The robot is initially moving at the maximum specified velocity, as can be seen in Figure 7.15(a-d). Following the user affective reaction, as shown in Figure 7.19, the robot is slowed down and then stopped, as shown in

Figure 7.15(e-g). Note that there is approximately a 2s delay between the start of the robot motion and user affective response.

Once the reaction of the user subsides, the robot completes its mission, at a lowered velocity, as shown in Figure 7.15(h).

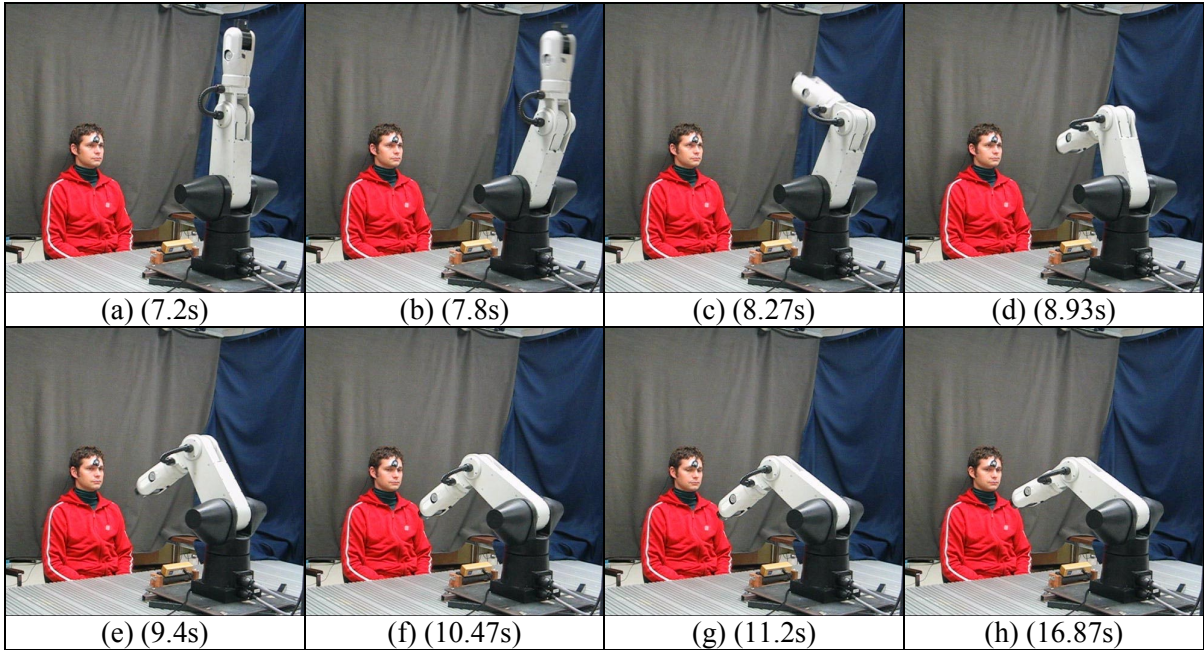


Figure 7.15. Affective State Test Case.

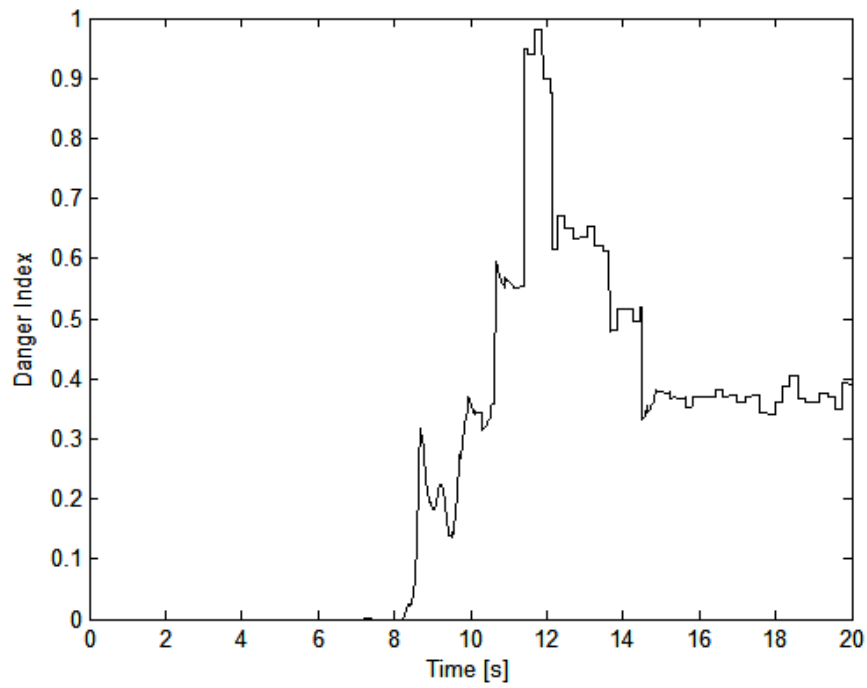


Figure 7.16. Affective State Test Case danger index.

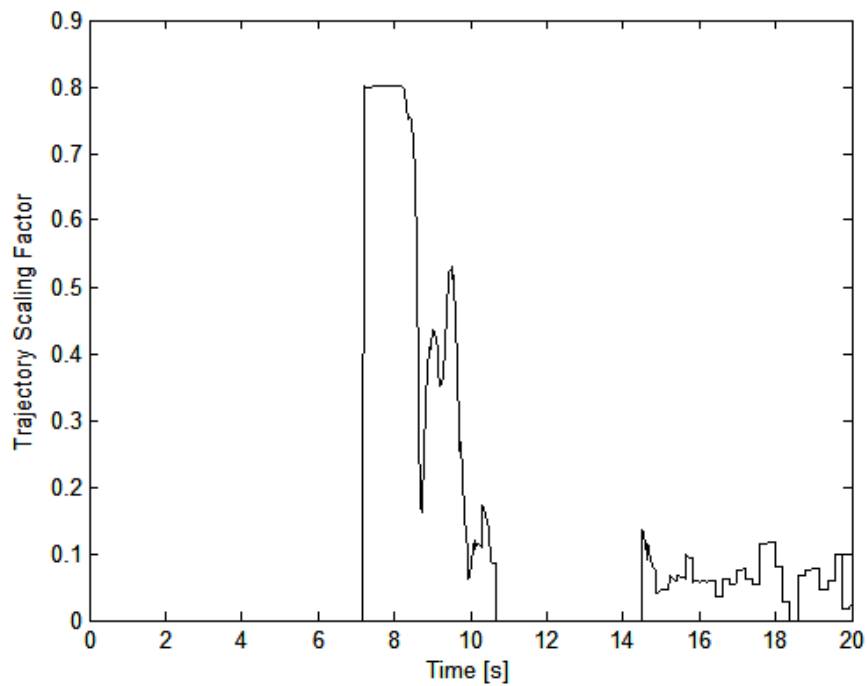


Figure 7.17. Affective State Test Case trajectory scaling.

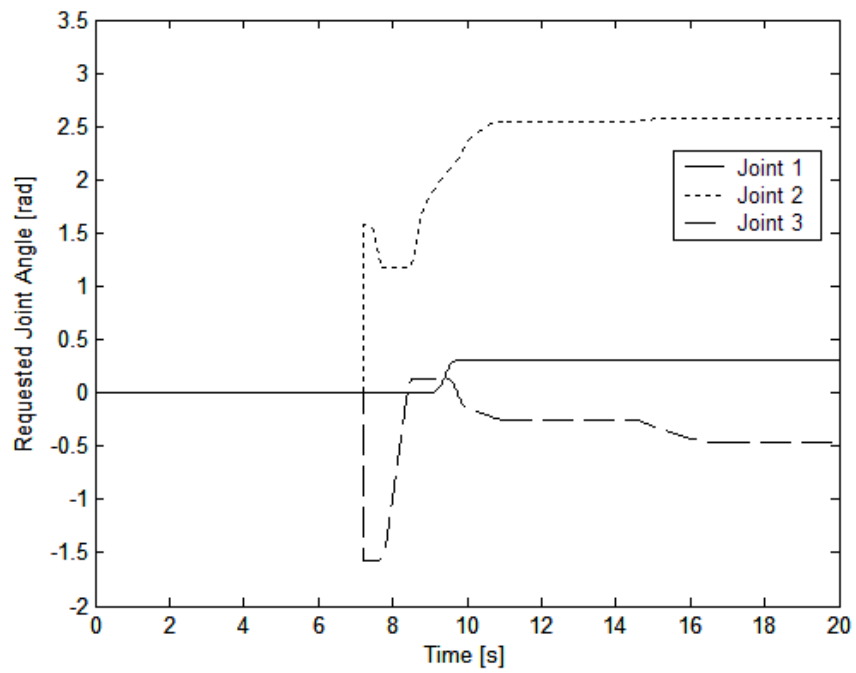


Figure 7.18. Affective State Test Case joint trajectory.

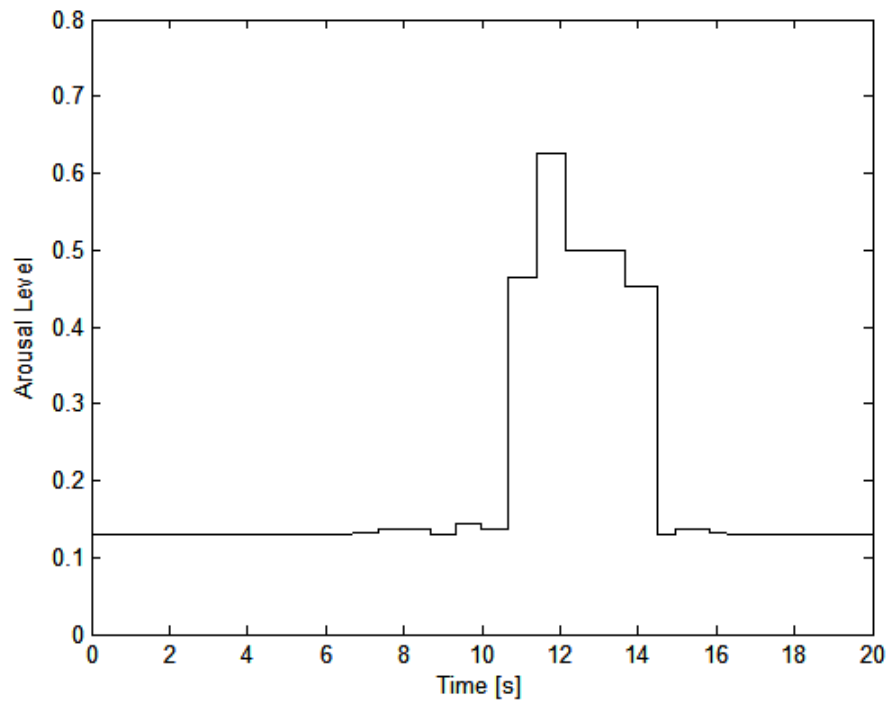


Figure 7.19. Affective State Test Case estimated arousal.

Orientation Test Case

The Orientation Test Case demonstrates system behavior during user head orientation changes. Prior to the start of the experiment, the head orientation module was trained by capturing images of the user at each orientation category. Only the horizontal head angle (pan) was used during the experiment. Figure 7.20 shows sample frames from a video sequence of the experiment. Figure 7.21 shows the user horizontal head orientation in radians, as reported by the head pose estimation component of the user monitoring module. Figure 7.22 shows the total danger index during the experiment. Figure 7.23 shows the velocity scaling, and Figure 7.24 shows the resulting trajectory commands for the first three of the robot joints. For this test case, the maximum normalized velocity of the robot (V_{\max}) was set to 0.35. The robot's task is to approach the area of the table directly in front of the user, simulating a pick-up task. The robot starts out from the upright position, as shown in the first Frame. Initially, the user is oriented towards the robot, with the horizontal head orientation angle of 0 degrees. The robot begins its motion towards the user, as shown in Figure 7.20(b). The motion begins 5.5s after the start of data acquisition. After the robot motion has already started, the user turns away from the robot, as shown in Figure 7.20(c), and in Figure 7.21. Figure 7.25 and Figure 7.26 show the results of the orientation tracking displayed by the vision system. The 3D position display has been turned off for clarity. Since the robot is still far away from the user, the motion proceeds, however at a decreased velocity, as can be seen from the decreased velocity scaling in Figure 7.23, and the resulting joint trajectories in Figure 7.24. As the robot approaches the person, the velocity scaling decreases to zero, stopping the robot at a safe distance from the user, as seen in Figure 7.20(e). At 13.93s (Figure 7.20(f)), the user turns back towards the robot. At this point, the danger index is lowered, and the velocity scaling is increased correspondingly. The robot can now proceed closer to the user, and complete the planned task. Note that the velocity of the robot slows again as the robot approaches the user, since the danger index increases due to the decreased distance between the robot and the user.

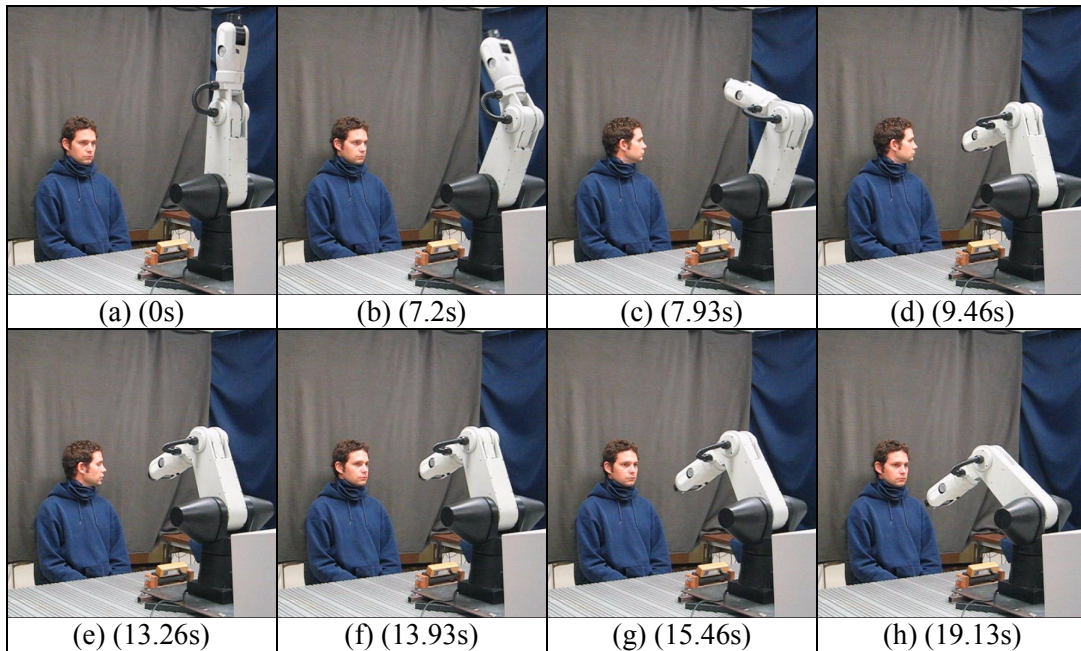


Figure 7.20. Orientation Test Case video frames.

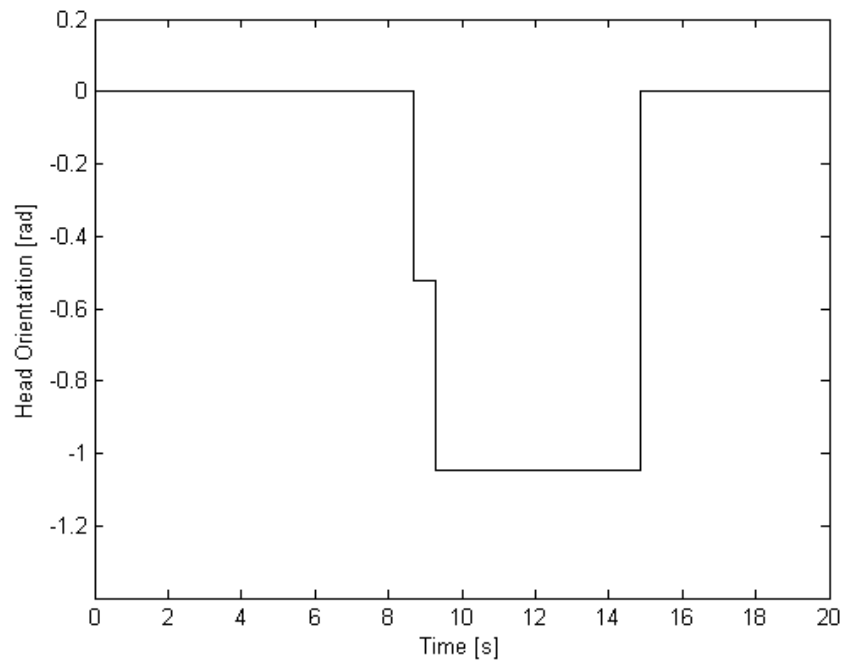


Figure 7.21. Orientation Test Case user head orientation.

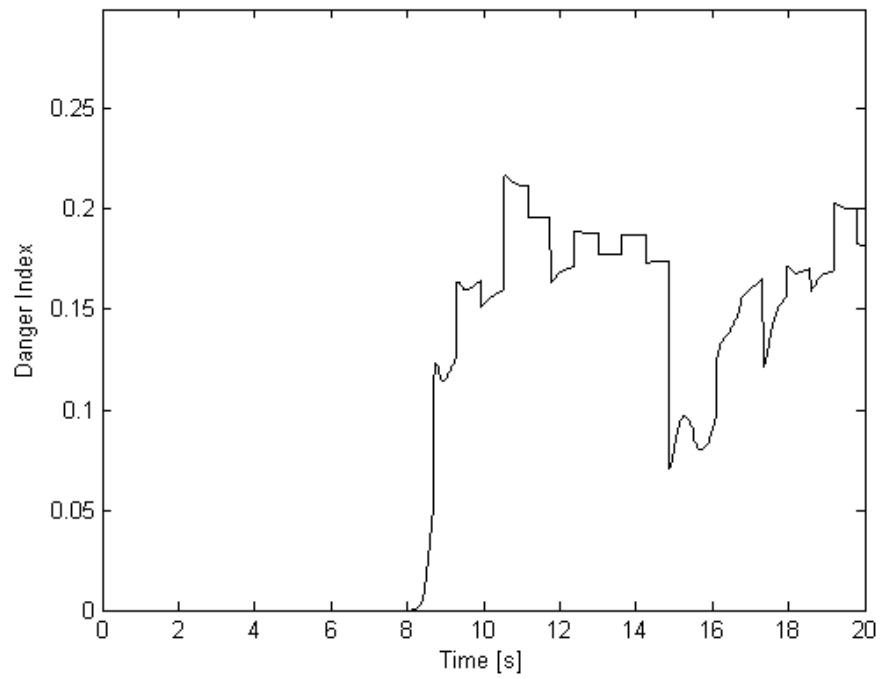


Figure 7.22. Orientation Test Case danger index.

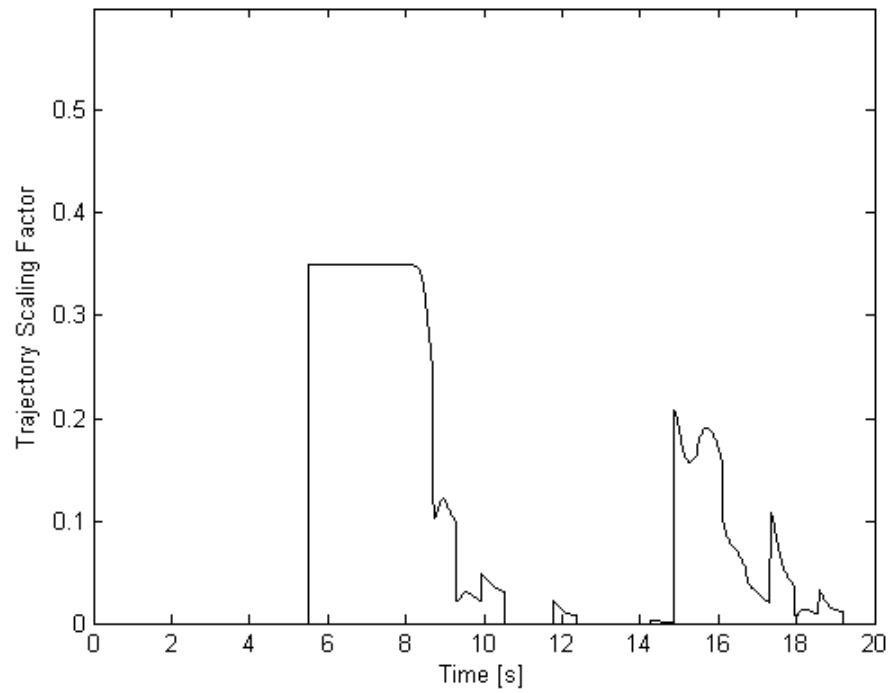


Figure 7.23. Orientation Test Case velocity scaling.

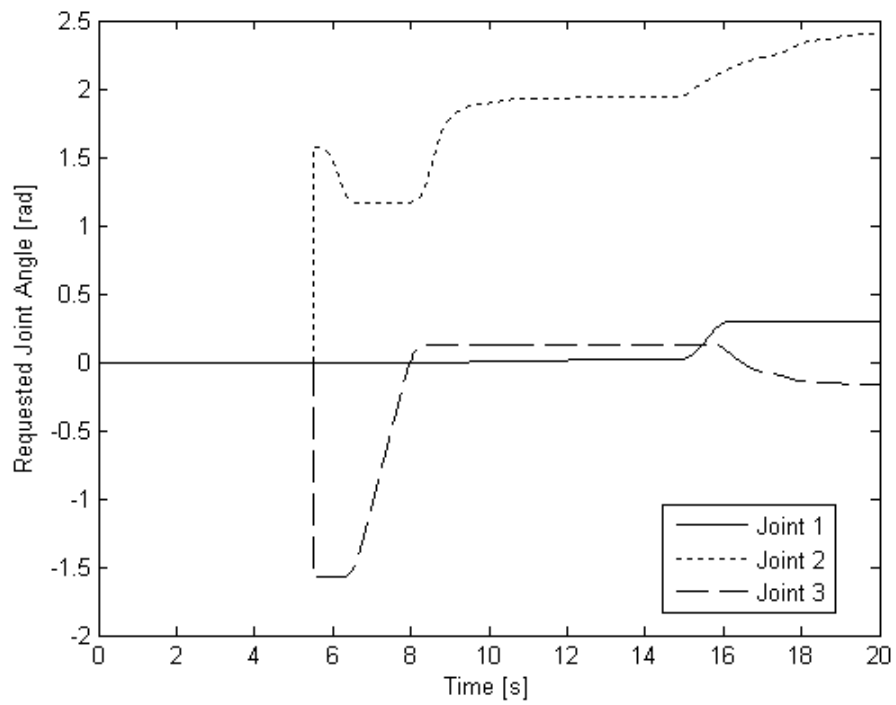


Figure 7.24. Orientation Test Case reference trajectory.

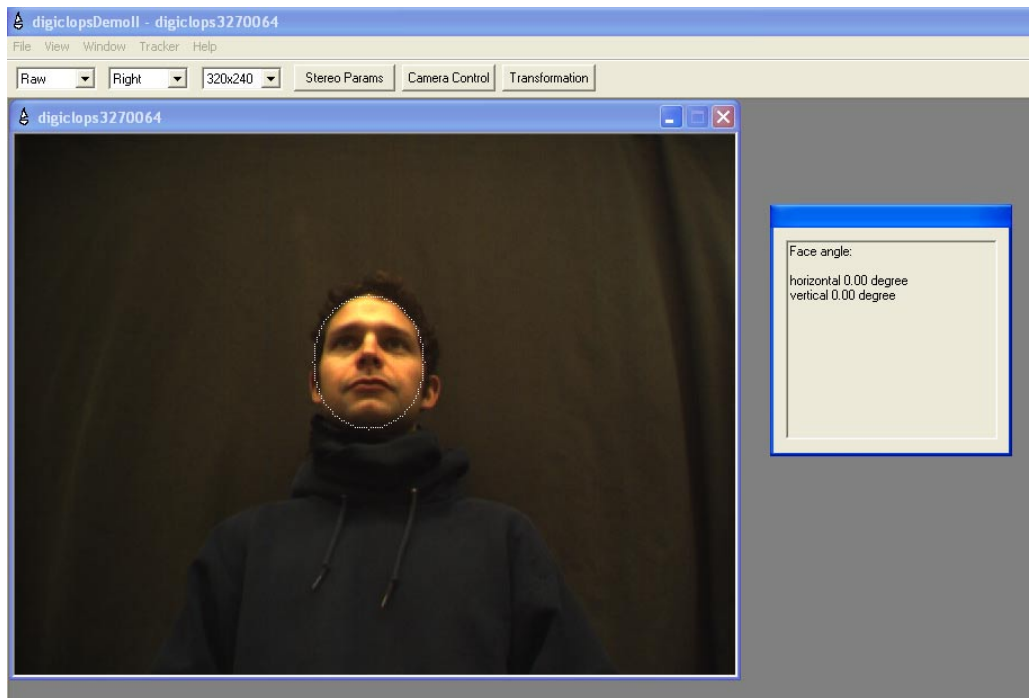


Figure 7.25. Head orientation tracking (horizontal angle = 0, vertical angle = 0).

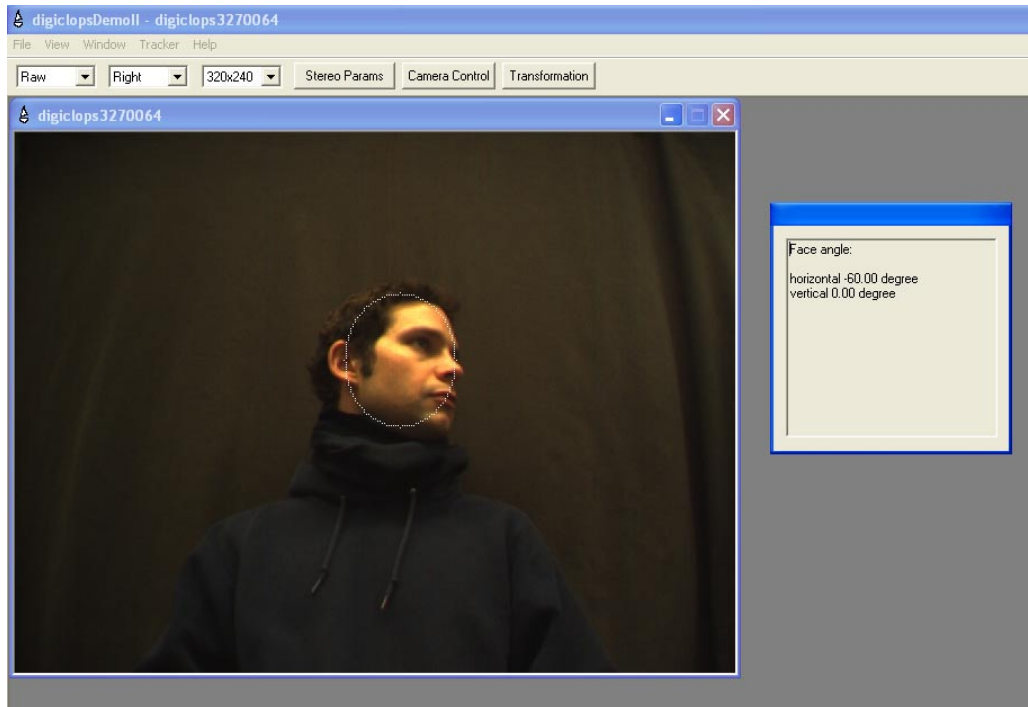


Figure 7.26. Head orientation tracking (horizontal angle = -60 degrees, vertical angle = 0).

7.3 Summary

The test cases discussed above demonstrate the functionality of the overall system, and the integrated operation of the system components. The robot ensures human safety by planning and modifying its trajectory at three different time horizons: long-term path planning, medium term trajectory planning and short-term reactive control. At each stage, a quantitative level of danger is used to guide the decision making process. The robot also has available information about the user location and head orientation, and an estimate of the user affective state in terms of the level of arousal. This information is used to modulate the estimated level of danger, to further improve the safety and intuitiveness of the interaction.

Chapter 8: Conclusions and Recommendations

In this thesis, a novel methodology for ensuring safety during human-robot interaction during planning and control has been presented, based on an explicit quantification of the level of danger in the interaction. Specifically, a methodology for assessing the level of danger at both the planning and control stages has been developed. Planning and control algorithms have been proposed for minimizing the estimated danger during the interaction. Further, a human monitoring system has been proposed for enhancing the safety of the interaction through the use of visual and human physiological information. To this end, a novel methodology for analyzing and processing physiological data to estimate user affective state during real-time human-robot interaction has been developed and tested. All of the above methods have been developed into physical system components that have been integrated and validated on a robot platform. The details of the specific contributions arising from this work are summarized in the following section.

8.1 Summary of Contributions

Danger Measure Formulation

A novel formulation for quantifying the level of danger in a human-robot interaction has been proposed in this work. The danger evaluation is divided into two components, namely: (i) the static and quasi-static measures that require long term planning for optimization, and, (ii) the dynamic measures, which can be optimized on-line. The danger evaluation is based on factors that affect the impact force during a human-robot collision at any point of the robot body. The *danger criterion* encapsulates the static measures such as the robot inertia and the centre of mass distance between the

robot and the user, and is used during path planning. Two different formulations were proposed and evaluated and the product of factors formulation was shown to give superior performance.

The *danger index* describes the dynamic hazards such as the distance and velocity between the potential contact points on the robot and the user, and the effective robot inertia at the contact point. The danger index is evaluated and optimized in real-time. A product of factors formulation is also used for the danger index, so that false positives when estimating the level of danger are minimized. An approach for incorporating human monitoring information into the danger index is also proposed, so that the reactions of the user affecting safety, such as lack of awareness of the robot, or strong affective reaction, are also incorporated into robot control.

Safe Path Planning

A novel path planning method was proposed which minimizes the danger criterion along the path, while achieving the desired task. The path planning method is applicable to any articulated robot structure, including redundant manipulators. This work is the first in the literature to develop a path planning algorithm based on an explicit danger criterion that considers the entire structure of the articulated robot. Safe path planning is a vital component of a safety strategy for human-robot interaction, because it can significantly improve the ability of the robot to respond to an unanticipated hazard. The planning methodology handles cluttered environments by implementing a two-stage approach and incorporating backwards planning if an inverse kinematics model of the robot is available.

The developed planning methodology was implemented on a 6 DoF industrial robot and tested in user trials. The safe planned motions were compared to motions planned using a nominal potential field planner [44]. The user trials show that subjects perceive safe planned motions to cause less anxiety and surprise as compared to the potential field planner.

Safety Based Control Strategy

A reactive motion planning and control strategy for articulated robots during real-time human-robot interaction was developed in this thesis. The proposed controller acts to minimize the danger,

once a hazard is identified requiring a change from the initial plan. The algorithm was shown to behave stably and safely at all points in the robots workspace, and is applicable to a variety of robot architectures.

The controller was implemented and tested on an industrial robot. The controller performance was validated during tests with human subjects, and safe and stable operation of the controller was demonstrated. The developed controller can be used either as part of the overall safety strategy, or as a standalone unit, making it potentially applicable to industrial robots.

Human Monitoring System

The ability of the robot to detect and estimate the user's affective state is important for ensuring safe, intuitive and human-like interaction. A methodology was developed for estimating user affective state in real-time during human-robot interaction from physiological signals such as heart rate and skin conductance. The two dimensional valence-arousal model [95] was used to represent the affective state. The inference engine was developed using results from psycho-physiological research [86, 94-96, 98, 99, 102]. The method was experimentally validated during user trials. To the author's knowledge, this is the first time affective state estimation has been implemented and tested during real-time interaction with a physical robotic device. Experimental results show that physiological signals show promise for use in real-time interaction, particularly in predicting user arousal, which is shown to be correlated to anxiety and surprise. However, unlike studies using picture viewing [94, 95], or video game playing [60, 61] to elicit the physiological response, experimental results using robot motion as the stimulus show that corrugator muscle activity does not appear to be a good indicator of valence. Therefore, valence could not be reliably predicted using the methodology developed. More research is required to further improve arousal estimation, and to develop a reliable method of estimating valence.

Implementation and Testing of an Exemplar System

A human-robot interaction test-bed was developed and implemented, incorporating the safe planner, safe controller and human monitoring functions. A methodology for smooth integration of

safety strategies for differing planning horizons was proposed. The integrated system was implemented and tested in a series of exemplar real-time human-robot interaction test cases. The proposed safe planning, control and human monitoring algorithms were verified under various conditions, and their correct behavior demonstrated.

The developed test-bed provides a working initial prototype to demonstrate the proposed strategies. However, further research is needed to develop a robust, safe and intuitive human-robot interaction system, as discussed in the following section.

8.2 Future work

In general, further work is needed to improve the robot's perception of the environment and the user, and to improve the robots ability to perform long-range planning during the interaction. Although human-robot interaction is a extremely broad and fertile area for research, four specific areas have been identified from this work as high potential areas for moving this research forward; namely, (i) affective state estimation, (ii) vision based human monitoring, (iii) on-line re-planning, and (iv) user trials.

Improving Affective State Estimation

More research is needed to improve the robot's ability to perceive the user's affective state. When physiological signals are used, the fuzzy inference approach developed here, as well as other proposed methods [22, 23, 60, 61, 93, 97] implement estimation instantaneously, i.e., they consider a single time slice of the data. However, most physiological signatures are time varying signals. A process model that includes modeling of signals as a function of time is hypothesized to provide a better characterization of the signals and improve estimation capabilities.

A further issue when estimating affective state is the differentiation between involuntary response (such as the startle reflex or the flight-or-flight response) and cognitive responses. More research is required to determine which physiological channels can be used to estimate which type of response.

A better differentiation between these two types of responses could improve affective state estimation and provide additional information about the user's intent during the interaction.

Finally, other modalities can be used to perform affective state estimation, such as facial expressions [88-92] or prosody in voice [114-116]. Such modalities could also be incorporated into a human-robot interaction system. The use of multiple modalities has the potential of further improving affective state estimation.

Improved vision system

Better human detection and tracking are required to ensure that the robot is aware of the exact human location during interaction. The simple system developed in this work is not adequate to ensure fail-safe operation during arbitrary human-robot interaction scenarios. One key issue that remains to be addressed is the problem of occlusion. Using a single camera significantly limits the available workspace where the user is visible. Even within this workspace, the camera can become occluded either by the robot or by the user. While the camera is occluded, the robot does not have information on the user's position, which is unacceptable from a safety standpoint. A user tracking system needs to be developed which provides the user location and orientation at all times during the interaction and through arbitrary occlusions. This system may be implemented through the use of multiple cameras, and through the use of a more detailed model to describe the user (for example, an inverse kinematics model), which can be used to predict the user location during temporary occlusions. Alternately, other technologies can also be used to implement human tracking, such as magnetic trackers.

Prediction and on-line re-planning

The system presented herein does not have the capability to perform path re-planning in real-time. In the current system, if a hazard is identified, the robot generates a reactive plan to evade the immediate hazard, abandoning the task goal. Developing fast algorithms for high dimensional configuration planning or path modification could improve the robot productivity.

If a more accurate model of the human motion is available, the robot could also plan its motion based on the predicted behavior of the user, which could potentially improve the safety of the interaction by anticipating user behavior and moving proactively to avoid the hazard.

User Trials

Further user trials are necessary to evaluate the proposed interaction strategy, and to determine user perceptions of the proposed system. User trials are required to determine users' evaluation of the system behavior, particularly the robot's response to affective state information. A key question is whether users will perceive a robot that responds to changes in their affective state as more intuitive and safer.

Long-term trials are also required to learn more about the effects of habituation. The current affective state estimation is designed to handle short-term habituation, but it is not known whether this approach is adequate for handling habituation over extended periods of operation. An adaptive affective state estimation algorithm may be required to address long-term habituation.

This work proposes strategies for improving the safety of human-robot interaction in the planning and control stages, by minimizing the potential for a collision to occur, and by minimizing the impact force in the case of a collision. Safety is improved through planning to minimize danger at different time horizons, and by including information about the user's behavior through the use of human monitoring. An initial working prototype was developed to demonstrate the developed strategies. However, more work is needed to improve the robot's perception of the user, as well as the robot's planning and decision-making capabilities, to ensure fail-safe operation with untrained users in arbitrary environments.

Bibliography

- [1] R. Bischoff and V. Graefe, "HERMES - A Versatile Personal Robotic Assistant," *Proceedings of the IEEE*, vol. 92, no. 11, pp. 1759 - 1779, 2004.
- [2] K. Wada, T. Shibata, T. Saito, and K. Tanie, "Effects of Robot-Assisted Activity for Elderly People and Nurses at a Day Service Center," *Proceedings of the IEEE*, vol. 92, no. 11, pp. 1780 - 1788, 2004.
- [3] J. M. Weiner, R. J. Hanley, R. Clark, and J. F. Van Nostrand, "Measuring the Activities of Daily Living: Comparisons Across National Surveys," *Journal of Gerontology: Social Sciences*, vol. 45, no. 6, pp. 229 - 237, 1990.
- [4] A. J. Bearveldt, "Cooperation between Man and Robot: Interface and Safety," presented at IEEE International Workshop on Robot Human Communication, pp. 183-187, 1993.
- [5] H. Arai, T. Takubo, Y. Hayashibara, and K. Tanie, "Human-Robot Cooperative Manipulation Using a Virtual Nonholonomic Constraint," presented at IEEE International Conference on Robotics and Automation, pp. 4063 - 4069, 2000.
- [6] V. Fernandez, C. Balaguer, D. Blanco, and M. A. Salichs, "Active Human - Mobile Manipulator Cooperation Through Intention Recognition," presented at IEEE International Conference on Robotics and Automation, pp. 2668 - 2673, 2001.
- [7] E. Guglielmelli, P. Dario, C. Laschi, and R. Fontanelli, "Humans and technologies at home: from friendly appliances to robotic interfaces," presented at IEEE International Workshop on Robot and Human Communication, pp. 71 - 79, 1996.
- [8] K. Kawamura, S. Bagchi, M. Iskarous, and M. Bishay, "Intelligent Robotic Systems in Service of the Disabled," *IEEE Transactions on Rehabilitation Engineering*, vol. 3, no. 1, pp. 14 - 21, 1995.
- [9] C. Breazeal, "Socially intelligent robots: research, development, and applications," presented at IEEE International Conference on Systems, Man and Cybernetics, Tucson, AZ USA, pp. 2121-2126, 2001.
- [10] Aibo Robotic Dog, Online: <http://www.us.aibo.com/>.
- [11] Roomba Robotic Vacuum Cleaner, Online: <http://www.irobot.com>
- [12] A. Pentland, "Perceptual Intelligence," *Communications of the ACM*, vol. 43, no. 3, pp. 35-44, 2000.

- [13] P. I. Corke, "Safety of advanced robots in human environments," Discussion Paper for IARP, Online 1999.
- [14] C. W. Lee, Z. Bien, G. Giralt, P. I. Corke, and M. Kim, "Report on the First IART/IEEE-RAS Joint Workshop: Technical Challenge for Dependable Robots in Human Environments," IART/IEEE-RAS 2001.
- [15] Sony Qrio, Online:<http://www.sony.net/SonyInfo/QRIO/technology/index5.html>.
- [16] "RIA/ANSI R15.06 - 1999 American National Standard for Industrial Robots and Robot Systems - Safety Requirements." New York: American National Standards Institute, 1999.
- [17] S. P. Gaskill and S. R. G. Went, "Safety Issues in Modern Applications of Robots," *Reliability Engineering and System Safety*, vol. 52, pp. 301-307, 1996.
- [18] Y. Yamada, T. Yamamoto, T. Morizono, and Y. Umetani, "FTA-Based Issues on Securing Human Safety in a Human/Robot Coexistence System," presented at IEEE Systems, Man and Cybernetics SMC'99, pp. 1068-1063, 1999.
- [19] Y. Yamada, Y. Hirawawa, S. Huang, Y. Umetani, and K. Suita, "Human - Robot Contact in the Safeguarding Space," *IEEE/ASME Transactions on Mechatronics*, vol. 2, no. 4, pp. 230-236, 1997.
- [20] Y. Matsumoto, J. Heinzmann, and A. Zelinsky, "The Essential Components of Human - Friendly Robot Systems," presented at International Conference on Field and Service Robotics, pp. 43-51, 1999.
- [21] V. J. Traver, A. P. del Pobil, and M. Perez-Francisco, "Making Service Robots Human-Safe," presented at IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000), pp. 696-701, 2000.
- [22] R. Picard, *Affective Computing*. Cambridge, Massachusetts: MIT Press, 1997.
- [23] Z. Bien, "Soft Computing Based Emotion / Intention Reading for Service Robot," *Lecture Notes in Computer Science*, vol. 2275, pp. 121 - 128, 2002.
- [24] P. I. Corke, "A Robotics Toolbox for Matlab," *IEEE Robotics and Automation Magazine*, vol. 3, no. 1, pp. 24-32, 1996.
- [25] K. Ikuta and M. Nokata, "Safety Evaluation Method of Design and Control for Human-Care Robots," *The International Journal of Robotics Research*, vol. 22, no. 5, pp. 281-297, 2003.
- [26] D. Kulic and E. Croft, "Safe Planning for Human-Robot Interaction," *Journal of Robotic Systems*, vol. 22, no. 7, pp. 383 - 396, 2005.
- [27] D. Kulic and E. Croft, "Safety Based Control Strategy for Human-Robot Interaction," *Journal of Robotics and Autonomous Systems*, In Press, 2005.

- [28] Y. Yamada, Y. Hirawawa, S. Huang, Y. Umetani, and K. Suita, "Human - Robot Contact in the Safeguarding Space," *IEEE/ASME Transactions on Mechatronics*, vol. 2, no. 4, pp. 230-236, 1997.
- [29] Y. Yamada, T. Yamamoto, T. Morizono, and Y. Umetani, "FTA-Based Issues on Securing Human Safety in a Human/Robot Coexistence System," presented at IEEE Systems, Man and Cybernetics SMC'99, pp. 1068-1063, 1999.
- [30] A. Bicchi, S. L. Rizzini, and G. Tonietti, "Compliant design for intrinsic safety: General Issues and Preliminary Design," presented at IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1864-1869, 2001.
- [31] A. Bicchi and G. Tonietti, "Fast and "Soft-Arm" Tactics," *IEEE Robotics and Automation Magazine*, vol. 11, no. 2, pp. 22-33, 2004.
- [32] M. Zinn, O. Khatib, and B. Roth, "A new actuation approach for human friendly robot design," presented at IEEE International Conference on Robotics and Automation, New Orleans, LA, USA, pp. 249-254, 2004.
- [33] M. Zinn, O. Khatib, B. Roth, and J. K. Salisbury, "Towards a Human-Centered Intrinsically Safe Robotic Manipulator," presented at IARP-IEEE/RAS Joint Workshop on Technical Challenges for Dependable Robots in Human Environments, Toulouse, France, 2002.
- [34] J. Heinzmann and A. Zelinsky, "Building Human - Friendly Robot Systems," presented at International Symposium of Robotics Research, pp. 305-312, 1999.
- [35] J. Heinzmann and A. Zelinsky, "Quantitative Safety Guarantees for Physical Human-Robot Interaction," *The International Journal of Robotics Research*, vol. 22, no. 7-8, pp. 479 - 504, 2003.
- [36] J. Y. Lew, Y. T. Jou, and H. Pasic, "Interactive Control of Human/Robot Sharing Same Workspace," presented at IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 535-539, 2000.
- [37] B. Karlsson, N. Karlsson, and P. Wide, "A dynamic safety system based on sensor fusion," *Journal of Intelligent Manufacturing*, vol. 11, pp. 475 - 483, 2000.
- [38] J. Zurada, A. L. Wright, and J. H. Graham, "A Neuro-Fuzzy Approach for Robot System Safety," *IEEE Transactions on Systems, Man and Cybernetics - Part C: Applications and Reviews*, vol. 31, no. 1, pp. 49-64, 2001.
- [39] E. Guglielmelli, P. Dario, C. Laschi, R. Fontanelli, M. Susani, P. Verbeeck, and J. C. Gabus, "Humans and technologies at home: from friendly appliances to robotic interfaces," presented at IEEE International Workshop on Robot and Human Communication, pp. 71 - 79, 1996.
- [40] K. Kawamura, S. Bagchi, M. Iskarous, and M. Bishay, "Intelligent Robotic Systems in the Service of the Disabled," *IEEE Transactions on Rehabilitation Engineering*, vol. 3, no. 1, pp. 14 - 21, 1995.

- [41] S. Macfarlane and E. Croft, "Jerk-Bounded Robot Trajectory Planning - Design for Real-Time Applications," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 1, pp. 42-52, 2003.
- [42] K. Erkorkmaz and Y. Altintas, "High Speed CNC System Design: Part I - Jerk Limited Trajectory Generation and Quintic Spline Interpolation," *International Journal of Machine Tools and Manufacture*, vol. 41, no. 9, pp. 1323-1345, 2001.
- [43] D. Blanco, C. Balaguer, and L. Moreno, "Safe Local Path Planning for Human - Mobile Manipulator Cooperation," presented at IARP/IEEE-RAS Joint Workshop on Technical Challenge for Dependable Robots in Human Environments, 2002.
- [44] O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," *The International Journal of Robotics Research*, vol. 5, no. 1, pp. 90-98, 1986.
- [45] O. Khatib, "Inertial Properties in Robotic Manipulation: An Object-Level Framework," *The International Journal of Robotics Research*, vol. 14, no. 1, pp. 19 - 36, 1995.
- [46] A. A. Maciejewski and C. A. Klein, "Obstacle Avoidance for Kinematically Redundant Manipulators in Dynamically Varying Environments," *The International Journal of Robotics Research*, vol. 4, no. 3, pp. 109 - 117, 1985.
- [47] M. Nokata, K. Ikuta, and H. Ishii, "Safety-optimizing Method of Human-care Robot Design and Control," presented at Proceedings of the 2002 IEEE International Conference on Robotics and Automation, Washington, DC, pp. 1991-1996, 2002.
- [48] M. Chen and A. M. S. Zalzal, "A Genetic Approach to Motion Planning of Redundant Mobile Manipulator Systems Considering Safety and Configuration," *Journal of Robotic Systems*, vol. 14, no. 7, pp. 529-544, 1997.
- [49] A. Oustaloup, B. Orsoni, P. Melchior, and H. Linares, "Path Planning by fractional differentiation," *Robotica*, vol. 21, pp. 59 - 69, 2003.
- [50] O. Brock and O. Khatib, "Elastic Strips: A Framework for Motion Generation in Human Environments," *The International Journal of Robotics Research*, vol. 21, no. 12, pp. 1031-1053, 2002.
- [51] Y. Maeda, A. Takahashi, T. Hara, and T. Arai, "Human-Robot Cooperation with Mechanical Interaction based on Rhythm Entrainment," presented at IEEE International Conference on Robotics and Automation, pp. 3477 - 3482, 2001.
- [52] Y. Yamada, Y. Umetani, H. Daitoh, and T. Sakai, "Construction of a Human/Robot Coexistence System Based on A Model of Human Will - Intention and Desire," presented at IEEE International Conference on Robotics and Automation, pp. 2861 - 2867, 1999.
- [53] W. K. Song, D. J. Kim, J. S. Kim, and Z. Bien, "Visual Servoing for a User's Mouth with Effective Attention Reading in a Wheelchair-based Robotic Arm," presented at IEEE International Conference on Robotics and Automation, pp. 3662 - 3667, 2001.

- [54] R. Stiefelhagen, J. Yang, and A. Waibel, "Tracking Focus of Attention for Human-Robot Communication," presented at IEEE-RAS International Conference on Humanoid Robots, 2001.
- [55] A. Haro, M. Flickner, and I. Essa, "Detecting and Tracking Eyes By Using Their Physiological Properties, Dynamics, and Appearance," presented at International Conference on Computer Vision and Pattern Recognition, pp. 163 - 168, 2000.
- [56] C. H. Morimoto and M. Flickner, "Real-Time Multiple Face Detection Using Active Illumination," presented at IEEE International Conference on Automatic Face and Gesture Recognition, pp. 8 - 13, 2000.
- [57] Y. Takahashi, N. Hasegawa, K. Takahashi, and T. Hatakeyama, "Human Interface Using PC Display With Head Pointing Device for Eating Assist Robot and Emotional Evaluation by GSR Sensor," presented at IEEE International Conference on Robotics and Automation, pp. 3674 - 3679, 2001.
- [58] Y. Yamada, Y. Umetani, and Y. Hirawawa, "Proposal of a Psychophysiological Experiment System Applying the Reaction of Human Pupillary Dilation to Frightening Robot Motions," presented at IEEE International Conference on Systems, Man and Cybernetics, pp. 1052 - 1057, 1999.
- [59] N. Sarkar, "Psychophysiological Control Architecture for Human-Robot Coordination - Concepts and Initial Experiments," presented at IEEE International Conference on Robotics and Automation, Washington, DC, USA, pp. 3719-3724, 2002.
- [60] P. Rani, J. Sims, R. Brackin, and N. Sarkar, "Online stress detection using psychophysiological signals for implicit human-robot cooperation," *Robotica*, vol. 20, pp. 673-685, 2002.
- [61] P. Rani, N. Sarkar, C. A. Smith, and L. D. Kirby, "Anxiety detecting robotic system - towards implicit human-robot collaboration," *Robotica*, vol. 22, pp. 85-95, 2004.
- [62] S. Nonaka, K. Inoue, T. Arai, and Y. Mae, "Evaluation of Human Sense of Security for Coexisting Robots using Virtual Reality," presented at IEEE International Conference on Robotics and Automation, New Orleans, LA, USA, pp. 2770-2775, 2004.
- [63] J. Ho, "Open Architecture Controller for a CRS 465A Robotic Arm," Industrial Automation Laboratory, University of British Columbia 2003.
- [64] D. Meger, "CRS A465 Open Architecture Controller: Software User's Guide and Software Design Document," Industrial Automation Laboratory, University of British Columbia 2003.
- [65] Quanser Q8, Online:http://www.quanser.com/english/html/solutions/fs_Q8.html.
- [66] Pt. Grey Bumblebee, Online:<http://www.ptgrey.com/products/bumblebee/index.html>.
- [67] Thought Technology Ltd., Online:www.thoughttechnology.com.

- [68] D. Kulic and E. Croft, "Estimating Intent for Human-Robot Interaction," presented at IEEE International Conference on Advanced Robotics, Coimbra, Portugal, pp. 810-815, 2003.
- [69] V. J. Traver, A. P. del Pobil, and M. Perez-Francisco, "Making Service Robots Human-Safe," presented at IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000), pp. 696-701, 2000.
- [70] J.-C. Latombe, *Robot Motion Planning*. Boston, MA: Kluwer Academic Publishers, 1991.
- [71] J. Barraquand and J.-C. Latombe, "Robot motion planning: A distributed representation approach," *The International Journal of Robotics Research*, vol. 10, no. 6, pp. 628 - 649, 1991.
- [72] J. M. Ahuactzin and K. K. Gupta, "The Kinematic Roadmap: A Motion Planning Based Global Approach for Inverse Kinematics of Redundant Robots," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 4, pp. 653 - 669, 1999.
- [73] H. Choset and J. Burdick, "Sensor-Based Exploration: The Hierarchical Generalized Voronoi Graph," *The International Journal of Robotics Research*, vol. 19, no. 2, pp. 96 - 125, 2000.
- [74] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566 - 580, 1996.
- [75] Y. Yu and K. K. Gupta, "Sensor-based Probabilistic Roadmaps: Experiments with an Eye-in-Hand System," *Advanced Robotics*, vol. 14, no. 6, pp. 515 - 536, 2000.
- [76] Y. Yamada, K. Suita, K. Imai, H. Ikeda, and N. Sugimoto, "A failure-to-safety robot system for human-robot coexistence," *Journal of Robotics and Autonomous Systems*, vol. 18, pp. 283 - 291, 1996.
- [77] B. Martinez-Salvador, A. P. del Pobil, and M. Perez-Francisco, "A Hierarchy of Detail for Fast Collision Detection," presented at IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000), pp. 745-750, 2000.
- [78] Y. K. Hwang and N. Ahuja, "Gross Motion Planning - A Survey," *ACM Computing Surveys*, vol. 24, no. 3, pp. 219-291, 1992.
- [79] K. Kondo, "Motion Planning with six degrees of freedom by multistrategic, bidirectional heuristic free space enumeration," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 267-277, 1991.
- [80] T. Tsuji and M. Kaneko, "Noncontact Impedance Control for Redundant Manipulators," *IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans*, vol. 29, no. 2, pp. 184-193, 1999.

- [81] D. Kulic and E. Croft, "Safe Planning for Human-Robot Interaction," presented at IEEE International Conference on Robotics and Automation, New Orleans, USA, pp. 1882-1887, 2004.
- [82] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, New Jersey: Prentice Hall, 2002.
- [83] K. K. Gupta and Z. Guo, "Motion Planning for Many Degrees of Freedom: Sequential Search with Backtracking," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 6, pp. 897-906, 1995.
- [84] Y. Matsumoto, T. Ogasawara, and A. Zelinsky, "Behavior Recognition Based on Head Pose and Gaze Direction Measurement," presented at IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2127 - 2132, 2000.
- [85] J. G. Wang and E. Sung, "Study on Eye Gaze Estimation," *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics*, vol. 32, no. 3, pp. 332 - 350, 2002.
- [86] L. M. Brown and Y. L. Tian, "Comparative Study of Coarse Head Pose Estimation," presented at Workshop on Motion and Video Computing, 2002.
- [87] Y. Wu and K. Toyama, "Wide-Range, Person and Illumination - Insensitive Head Orientation Estimation.," presented at International Conference on Image Processing, 2002.
- [88] G. Donato, "Classifying Facial Actions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 10, pp. 974 - 989, 1999.
- [89] P. Ekman, W. V. Friesen, and P. Ellsworth, *Emotion in the Human Face*. New York: Pergamon Press, 1972.
- [90] I. Essa and A. Pentland, "Coding, Analysis, Interpretation, and Recognition of Facial Expressions.," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 757 - 763, 1997.
- [91] M. Pantic and L. J. M. Rothkrantz, "Automatic Analysis of Facial Expressions: The State of the Art," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1424 - 1445, 2000.
- [92] Y. L. Tian, T. Kanade, and J. F. Cohn, "Recognizing Action Units for Facial Expression Analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 97 - 115, 2001.
- [93] R. Picard, "Toward Machine Emotional Intelligence: Analysis of Affective Physiological State," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 10, pp. 1175 - 1191, 2001.
- [94] M. M. Bradley, "Emotion and Motivation," in *Handbook of Psychophysiology*, J. T. Cacioppo, L. G. Tassinary, and G. G. Berntson, Eds., 2 ed. Cambridge: Cambridge University Press, 2000, pp. 602 - 642.

- [95] M. M. Bradley and P. J. Lang, "Measuring Emotion: Behavior, Feeling and Physiology," in *Cognitive Neuroscience of Emotion*, R. D. Lane and L. Nadel, Eds. New York: Oxford University Press, 2000.
- [96] P. J. Lang, "The Emotion Probe: Studies of Motivation and Attention," *American Psychologist*, vol. 50, no. 5, pp. 372 - 385, 1995.
- [97] P. Ekman, R. W. Levenson, and W. V. Friesen, "Autonomic Nervous System Activity Distinguishes Among Emotions," *Science*, vol. 221, pp. 1208-1210, 1983.
- [98] K. A. Brownley, "Cardiovascular Psychophysiology," in *Handbook of Psychophysiology*, J. T. Cacioppo, L. G. Tassinary, and G. G. Berntson, Eds. Cambridge: Cambridge University Press, 2000.
- [99] M. E. Dawson, "The Electrodermal System," in *Handbook of Psychophysiology*, J. T. Cacioppo, L. G. Tassinary, and G. G. Berntson, Eds. Cambridge: Cambridge University Press, 2000.
- [100] P. S. Hamilton and W. J. Tompkins, "Quantitative investigation of QRS detection rules using the MIT/BIH arrhythmia database," *IEEE Transactions on Biomedical Engineering*, vol. BME-33, pp. 1157 - 1165, 1986.
- [101] D. Kulic and E. Croft, "Anxiety Detection during Human-Robot Interaction," presented at IEEE International Conference on Intelligent Robots and Systems, Edmonton, Canada, 2005.
- [102] J. T. Cacioppo and L. G. Tassinary, "Inferring Psychological Significance From Physiological Signals," *American Psychologist*, vol. 45, no. 1, pp. 16 - 28, 1990.
- [103] A. Ohman, A. Hamm, and K. Hugdahl, "Cognition and the Autonomic Nervous System: Orienting, Anticipation and Conditioning," in *Handbook of Psychophysiology*, J. T. Cacioppo, L. G. Tassinary, and G. G. Berntson, Eds., 2 ed. Cambridge: Cambridge University Press, 2000, pp. 533 - 575.
- [104] C. Wren, A. Azarbajejani, T. Darrell, and A. Pentland, "Pfinder: Real-Time Tracking of the Human Body," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 80 - 785, 1997.
- [105] R. Kjellden and J. Kender, "Finding Skin in Color Images," presented at International Conference on Automatic Face and Gesture Recognition, pp. 312 - 317, 1996.
- [106] K. Sobottka and I. Pitas, "Segmentation and Tracking of Faces in Color Images," presented at IEEE International Conference on Image Processing, pp. 483 - 486, 1996.
- [107] M. H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting Faces in Images: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 34 - 58, 2002.
- [108] P. Perez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," presented at European Conference on Computer Vision, 2002.

- [109] S. Birchfield, "Elliptical Head Tracking Using Intensity Gradients and Color Histograms," presented at IEEE Conference on Computer Vision and Pattern Recognition, 1998.
- [110] D. Meger, "Work Term Report: Summer 2004," Industrial Automation Laboratory, UBC 2004.
- [111] C. W. W. Poon, "Real-time Head Orientation Estimation," *Department of Computer Science*, B. Sc. Vancouver: University of British Columbia, 2005.
- [112] V. Bereg, "Work Term Report: Summer 2005," Industrial Automation Laboratory, UBC 2005.
- [113] L. G. Shapiro and G. C. Stockman, *Computer Vision*. Upper Saddle River, New Jersey: Prentice Hall, 2001.
- [114] T. S. Polzin, "Verbal and Non-Verbal Cues in the Communication of Emotions," presented at IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 2429 - 2432, 2000.
- [115] M. Slaney and G. McRoberts, "Baby Ears - A Recognition System for Affective Vocalizations," presented at IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 985 - 988, 1998.
- [116] C. W. Wightman and M. Ostendorf, "Automatic Labeling of Prosodic Patterns," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 4, pp. 469 - 481, 1994.
- [117] M. W. Spong and M. Vidyasagar, *Robot Dynamics and Control*: Wiley, 1989.

Appendix A. Trajectory Planning

The trajectory planner takes as input a sequence of points through which the joint must pass, these are called the waypoints. A section is defined as the path between two consecutive path waypoints. A pre-processing step determines the start and end conditions for each section. The start and end waypoint locations, and the start and end conditions at each waypoint are then passed to the trajectory planner. The trajectory planner plans the motion profile for each section, using a sequence of cubic splines. There is generally more than one cubic spline per section. Each cubic spline is called a segment.

A.1.1 Single Joint motion between two points

Assuming that only one joint is moving in a given section, the minimum time trajectory between two points which satisfies all the dynamic constraints is achieved with the trapezoidal acceleration profile [117]. The trapezoidal acceleration profile consists of up to 7 cubic trajectory segments. Several variants of the trapezoidal acceleration profile are possible, depending on the distance and end conditions between the two points and the magnitude of the dynamic constraints.

A.1.1.1 *Stop-Stop End Conditions*

If the joint is at rest at the start and stop of the motion, two variants are possible. If the distance between the two points is large enough for the joint to achieve maximum velocity, the ‘Long Motion Profile’ followed will be as shown in Figure A.1. The minimum time required for each segment, and the start/end conditions for each segment are calculated based on the dynamic constraints as shown in Equations (A.1).

$$\begin{aligned}
t_1 &= \frac{a_{\max}}{j_{\max}} \\
v_1 &= \frac{1}{2} j_{\max} t_1^2 \\
d_1 &= d_{start} + \frac{1}{6} j_{\max} t_1^3 \\
v_2 &= v_{\max} - a_{\max} t_1 + j_{\max} t_1^2 \\
\Delta t_2 &= \frac{v_2 - v_1}{a_{\max}} \\
d_2 &= d_1 + v_1 \Delta t_2 + \frac{1}{2} a_{\max} \Delta t_2^2 \\
t_2 &= t_1 + \Delta t_2 \\
t_3 &= t_2 + t_1 \\
v_3 &= v_{\max} \\
d_3 &= d_2 + v_2 t_1 + \frac{1}{2} a_{\max} t_1^2 - \frac{1}{6} j_{\max} t_1^3 \\
t_5 &= t_3 + \frac{(d_{end} - d_{start}) - 2d_3}{v_{\max}}
\end{aligned} \tag{A.1}$$

The last 3 segments are symmetric reflections in the velocity and acceleration domain of the first three segments; therefore the minimum time required for segments 5, 6 and 7 will be the same as for segments 3, 2 and 1. In (A.1), d_{start} and d_{end} indicate the starting and ending joint positions, respectively, while j_{\max} , a_{\max} and v_{\max} indicate the maximum jerk, acceleration and velocity for the joint. It is assumed that the kinematic limits are identical in either direction of motion. Variables t_i , v_i and d_i indicate the time, velocity and position at the end of the i -th segment. The equations (A.1) assume that position 2 is larger than position 1 (i.e., a forward motion), for backwards motions the appropriate sign reversal is required.

One can note that in order to reach maximum velocity, the trajectory section from stop point to stop point must be $2d_3$ in length. That is, d_3 is the minimum distance required by a joint to ramp up to, or down from, maximum velocity.

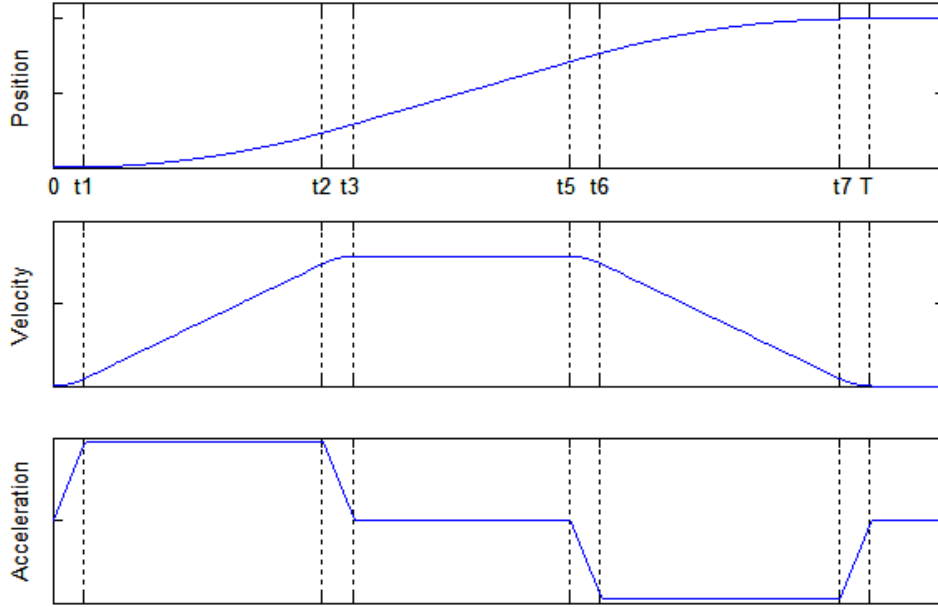


Figure A.1. Long Motion Profile, joint reaches maximum velocity.

If the distance between the two endpoints is too small for maximum velocity to be reached, the ‘Short Motion Profile’, shown in Figure A.2, is used. The first and last segments (starting/ending from rest and reaching maximum acceleration) are identical to the segments in (A.1) above. For the 2nd segment, the minimum time required for each segment, and the start/end conditions for each segment are calculated based on the dynamic constraints as shown in Equations (A.2).

$$\begin{aligned}
 t_2^{short} &= \frac{-a_{\max}^2 + \sqrt{a_{\max}^4 + 4(d_{\text{end}} - d_{\text{start}})a_{\max}j_{\max}^2}}{2a_{\max}j_{\max}} \\
 d_2^{short} &= \frac{1}{6}j_{\max}t_1^3 + \frac{1}{2}j_{\max}t_1t_2^{short2} - \frac{1}{2}j_{\max}t_1^2t_2^{short} \\
 v_2^{short} &= -\frac{1}{2}j_{\max}t_1^2 + j_{\max}t_1t_2^{short}
 \end{aligned} \tag{A.2}$$

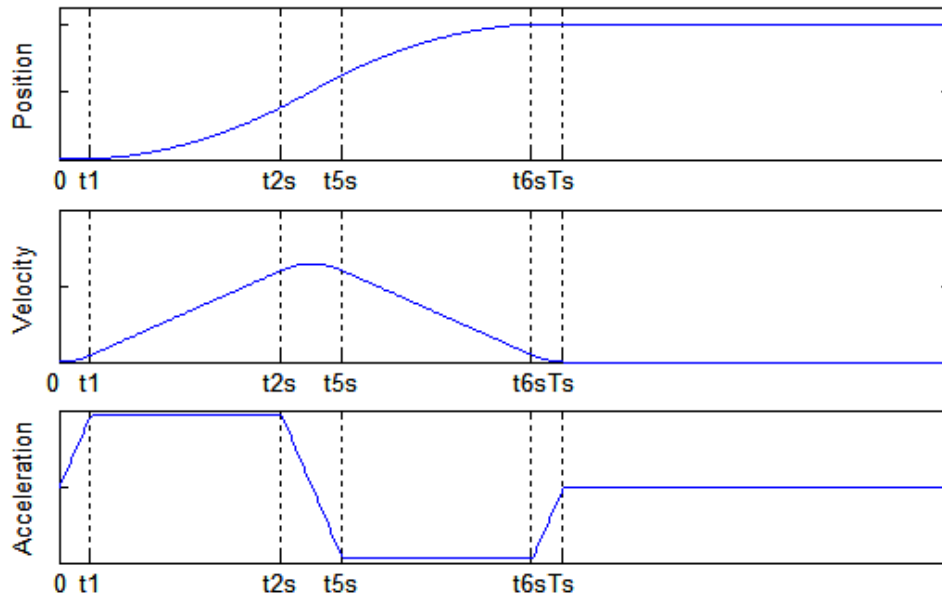


Figure A.2. Short Motion Profile. Distance traveled is too short to reach maximum velocity.

In general, it is also possible for the commanded distance to be too short for the full acceleration limit to be reached. This will occur if $(d_{end} - d_{start}) < d_l$. The trajectory planner does not handle this case, since the path planner (described in Chapter 4) and the preprocessor (Section A.1.3) ensure that the desired motion distance is always greater than d_l .

A.1.1.2 Non-zero velocity at end conditions

In order to process/scale the trajectory, each segment of the trajectory is formatted as a cubic polynomial. Thus the trajectory preprocessor must compute the end conditions for each trajectory segment on the fly. In the case of single joint motion, the joint is always moving at a limit of either jerk, acceleration or velocity, i.e., it is always following either the Long or the Short Motion Profile, depending on the total distance to be traveled by the joint. The start/end conditions for each section are specified by the pre-processor as the distance away from the stop point, d_{exp} , as shown in Table A.1.

Table A.1. Calculation of the start/end velocity and acceleration.

Start/End Distance Specified	Condition	Start/End Conditions
$d_{exp} = 0$		Joint is starting/ending at rest $v_{exp} = 0$ $a_{exp} = 0$
$0 < d_{exp} < d_3$		Joint has already started moving in the previous segment, or will finish the motion in the subsequent segment, however full velocity has not been reached v to be determined according to (A.3) $a_{exp} = a_{max}$
$d_{exp} \geq d_3$		Joint has already started moving in the previous segment, or will finish the motion in the subsequent segment, full velocity has been reached $v_{exp} = v_{max}$ $a_{exp} = 0$

Based on the distance specified, the trajectory planner determines which segment the joint is currently executing, and the starting velocity and acceleration for the motion. If the distance away from the start/stop point is specified as d_3 or greater (from equation A.1), the joint is moving at full velocity. Otherwise, the joint is accelerating or decelerating. To simplify the algorithm, waypoints are limited to occur during either the constant velocity or the constant acceleration/deceleration segments. This simplification is not overly constraining, since the constant jerk sections are generally much shorter than the constant acceleration sections. The time elapsed in the adjoining waypoints, and the starting/ending velocity and acceleration are calculated as shown in (A.3).

$$t_{exp} = \frac{-v_1 + a_{max}t_1 + \sqrt{(-a_{max}t_1 + v_1)^2 - 2a_{max}(d_1 - d_{exp} - v_1t_1 + \frac{1}{2}a_{max}t_1^2)}}{a_{max}}$$

$$v_{exp} = \frac{1}{2}j_{max}t_1^2 + a_{max}(t_{exp} - t_1) \quad (A.3)$$

$$a_{exp} = a_{max}$$

A.1.1.3 Calculating the Cubic Coefficients

Once the starting and ending conditions for each segment are known, the cubic coefficients are generated. Each quintic has the following form:

$$q_i(r) = C_{0i} + C_{1i}r + C_{2i}r^2 + C_{3i}r^3, \quad (\text{A.4})$$

where C_{ji} are the cubic coefficients, and r is the parameterized time. The cubic coefficients and the time required to traverse each segment are calculated as follows:

Segment 1

If the joint is starting from rest (starting $d_{exp} = 0$), segment one will be the increase from zero to maximum acceleration.

$$\begin{aligned} C_{01} &= d_{start} \\ C_{11} &= 0 \\ C_{21} &= 0 \\ C_{31} &= \frac{1}{6} j_{max} \\ t_{seg1} &= t_1 \end{aligned} \quad (\text{A.5})$$

Segment 2

The next segment is the constant acceleration pulse. The coefficients will depend on the starting conditions, as shown in Table A.2. The time required for the segment will depend on the starting conditions and the section length.

Table A.2. Segment 2 cubic spline coefficients.

Starting $d_{exp} = 0$	$0 < \text{starting } d_{exp} < d_3$
$C_{02} = d_{start} + d_1$	$C_{02} = d_{start}$
$C_{12} = v_1$	$C_{12} = v_{exp}$
$C_{22} = \frac{1}{2} a_{max}$	
$C_{32} = 0$	

Table A.3. Segment 2 duration.

Starting $d_{exp} = 0$		$0 < \text{starting } d_{exp} < d_3$	
$D_{total} > 2d_3$	$D_{total} \leq 2d_3$	$D_{total} > 2d_3$	$D_{total} \leq 2d_3$
$t_{seg2} = t_2$	$t_{seg2} = t_2^{shrt}$	$t_{seg2} = t_2 - t_{exp}^{starting}$	$t_{seg2} = t_2^{short} - t_{exp}^{starting}$

Segment 3

The next segment is a decrease from maximum acceleration. If the section is short ($D_{total} \leq 2d_3$), this segment will take the acceleration from a_{max} to $-a_{max}$. If the section is long ($D_{total} > 2d_3$), the segment will take the acceleration to zero.

Table A.4. Segment 3 cubic spline coefficients and duration.

$D_{total} > 2d_3$	$D_{total} \leq 2d_3$
$C_{03} = d_{start} + d_2 - d_{exp}^{starting}$	$C_{03} = d_{start} + d_2^{short} - d_{exp}^{starting}$
$C_{13} = v_2$	$C_{13} = v_2^{shoer}$
$C_{23} = \frac{1}{2} a_{max}$	
$C_{33} = -\frac{1}{6} j_{max}$	
$t_{seg3} = t_{seg2} + t_1$	$t_{seg3} = t_{seg2} + 2t_1$

Segment 4

The fourth segment is only present if $D_{total} > 2d_3$, i.e., if maximum velocity is reached.

Table A.5. Segment 4 cubic spline coefficients and duration.

Starting $d_{exp} < d_3$	starting $d_{exp} = d_3$
$C_{04} = d_{start} + d_3 - d_{exp}^{starting}$	$C_{04} = d_{start}$
$C_{14} = v_{max}$	
$C_{24} = 0$	
$C_{34} = 0$	
$t_{seg4} = t_{seg3} + \frac{d_{start} + d_{exp}^{starting} - d_{end} - d_{exp}^{ending} - 2d_3}{v_{max}}$	

Segment 5

If the joint reaches maximum velocity, the fifth, sixth and seventh segments will be reflections of the first three segments. If maximum velocity is not reached, the sixth and seventh segments will mirror the first two segments. The fifth segment is only present if maximum velocity has been reached, and the joint will start decelerating to a stop during this segment (i.e., $D_{total} > 2d_3$ and the ending $d_{exp} < d_3$).

$$\begin{aligned}
 C_{05} &= d_{end} - d_3 + d_{exp}^{ending} \\
 C_{15} &= v_{max} \\
 C_{25} &= 0 \\
 C_{35} &= -\frac{1}{6} j_{max} \\
 t_{seg5} &= t_{seg4} + t_1
 \end{aligned} \tag{A.6}$$

Segment 6

The sixth segment is present if the joint begins decelerating in this segment, i.e., the ending $d_{exp} < d_3$.

Table A.6. Segment 6 cubic spline coefficients.

$D_{total} > 2d_3$	$D_{total} \leq 2d_3$
$C_{06} = d_{end} - d_2 + d_{exp}^{ending}$	$C_{06} = d_{end} - d_2^{short} + d_{exp}^{ending}$
$C_{16} = v_2$	$C_{16} = v_2^{shoer}$
$C_{26} = -\frac{1}{2} a_{max}$	
$C_{36} = 0$	

Table A.7. Segment 6 duration.

Ending $d_{exp} = 0$	
$D_{total} > 2d_3$	$D_{total} \leq 2d_3$
$t_{seg6} = t_{seg5} + t_2 - t_1$	$t_{seg6} = t_{seg5} + t_2^{short} - t_1$
$0 < \text{ending } d_{exp} < d_3$	
$D_{total} > 2d_3$	$D_{total} \leq 2d_3$
$t_{seg6} = t_{seg5} + t_2 - t_{exp}^{ending}$	$t_{seg6} = t_{seg5} + t_2^{short} - t_{exp}^{ending}$

Segment 7

The final segment is present only if the ending condition is specified as a stop (i.e., ending $d_{exp} = 0$).

$$C_{07} = d_{end} - d_1$$

$$C_{17} = v_1$$

$$C_{27} = -\frac{1}{2} a_{max} \tag{A.7}$$

$$C_{37} = \frac{1}{6} j_{max}$$

$$t_{seg7} = t_{seg6} + t_1$$

A.1.2 Multiple Joint Motion

When multiple joints are moving in a section, the time required for the motion of each joint is first calculated, using the single joint motion algorithm described in Section A.1.1. The joint with the longest required time is considered the *critical joint*, and the required time of this joint is the *critical time*, t_{crit} . For this joint, the single joint motion profile is used. For those joints that have a required time that is shorter than the critical time, an alternate motion profile must be used so that all motions are coordinated over the same time period (the critical time), and the start and end point locations and conditions are respected.

Two strategies are used to generate the non-critical motion profiles, depending on the type of motion of the non-critical joints. If the non-critical joint has a stop condition (i.e., $d_{exp} = 0$) on

either the starting or the ending waypoint, a wait segment is inserted at the end with the stop condition.

$$t_{wait} = t_{crit} - t_{req} , \quad (A.8)$$

where t_{req} is the time required for the joint to complete its motion, t_{crit} is the critical time, and t_{wait} is the time to wait at the waypoint with the stop condition.

If the joint does not have a stop condition on either the starting or ending waypoint (i.e., $d_{exp} > 0$), a single quintic trajectory is used. The quintic trajectory has the form:

$$q_i(r) = C_{0i} + C_{1i}r + C_{2i}r^2 + C_{3i}r^3 + C_{4i}r^4 + C_{5i}r^5 . \quad (A.9)$$

To calculate the quintic coefficients C_{ji} , the starting and ending position, velocity and acceleration for the joint are calculated according to Table A.1. The desired time for the trajectory is t_{crit} . The conditions result in the following system of equations,

$$\begin{aligned} q(0) &= d_{start} & q(t_{crit}) &= d_{end} \\ \dot{q}(0) &= v_{start} & \dot{q}(t_{crit}) &= v_{end} , \\ \ddot{q}(0) &= a_{start} & \ddot{q}(t_{crit}) &= a_{end} \end{aligned} \quad (A.10)$$

which are solved to obtain the quintic coefficients.

$$\begin{aligned} C_{00} &= d_{start} \\ C_{10} &= v_{start} \\ C_{20} &= \frac{1}{2} a_{start} \\ C_{30} &= -\frac{3a_{start}t_{crit}^2 + 12v_{start}t_{crit} + 20d_{start} - a_{end}t_{crit}^2 + 8v_{end}t_{crit} - 20d_{end}}{2t_{crit}^3} \\ C_{40} &= \frac{3a_{start}t_{crit}^2 + 16v_{start}t_{crit} + 30d_{start} - 2a_{end}t_{crit}^2 + 14v_{end}t_{crit} - 30d_{end}}{2t_{crit}^4} \\ C_{50} &= -\frac{a_{start}t_{crit}^2 + 6v_{start}t_{crit} + 12d_{start} - a_{end}t_{crit}^2 + 6v_{end}t_{crit} - 12d_{end}}{2t_{crit}^5} \end{aligned} \quad (A.11)$$

$$t_{seg0} = t_{crit}$$

Once all the spline coefficients have been determined, the trajectory planner outputs a command value at each control step, based on the input parameterized time r according to (A.4)

and (A.9). The input r is calculated by the top level process based on the desired speed of motion and the size of the time step, according to Equation (5.1).

A.1.3 Waypoints Preprocessing and End Condition Generation

The preprocessing step performs two functions: first, the points generated by the planner (Chapter 4) are parsed to determine the minimum number of required waypoints. Then, the waypoints are analyzed to determine the velocity and acceleration conditions at each waypoint.

A.1.3.1 Generating the Waypoints

The planner generates points based on the quantization of its search space, which is generally 0.1 radians. Therefore, a path description will contain many more points than are necessary to generate the trajectory. The first preprocessing step parses the points generated by the planner to find the minimum number of waypoints required to represent the path. A waypoint is defined as any path point where at least one joint is either: starting from rest, coming to a stop, or changing the direction of motion.

The pseudocode shown in Figure A.3 describes the parsing algorithm.


```

Step 1: Store the first path point as a waypoint

Step 2:
For each path point i, starting at path point 2
For each joint j

Calculate
Dprev = distance between path point i and path point (i-1)
Dcurr = distance between path point (i+1) and path point i

b) If  $|D_{prev}| > 0$  and  $D_{curr} = 0$ 
Joint is stopping at this path point, store point i as a
Waypoint

If  $D_{prev} = 0$  and  $|D_{curr}| > 0$ 
Joint is starting at this path point, store point i as a
Waypoint

d) If  $\text{sign}(D_{prev}) \neq \text{sign}(D_{curr})$ 
Joint is changing the direction of motion at this path
point, store point i as a waypoint

Step 3: Store the last path point as a waypoint

```

Figure A.3. Waypoint generation pseudo-code.

A.1.3.2 Determining the waypoint end conditions

Once the waypoints have been generated, they are parsed to determine the required end conditions at each waypoint. The preprocessor also ensures that the specified conditions for each section can be achieved while meeting all the dynamic constraints, especially during multiple joint motion. For example, if one joint is moving at maximum velocity through a distance of 0.2 radians, while the other joint must start and end at rest through this same distance, the second joint will be the critical joint. However, since the distance being traveled by both joints is so short, it will not be possible for the first joint decelerate from maximum velocity, and then accelerate back to maximum velocity in order to meet the end point conditions, without breaking the maximum acceleration or jerk constraints. In this case, the preprocessor inserts a stop point at the preceding waypoint for the fast joint, to ensure that all the dynamic constraints are met.

First, the required start and end conditions are calculated for each section (between each set of consecutive waypoints) assuming independent single joint motion at each joint, as shown in the algorithm in Figure A.4.

```

Procedure: FindEndConditions
For each joint
    If i = 0 (this is the first waypoint)
        StartCondition = 0;
    Else
        StartCondition = EndCondition of the previous section.
    End If

    If I = N (this is the last waypoint)
        EndCondition = 0;
    Else
        Calculate Dcurr = dist between wpoint (i+1) and wpoint i

        If abs(Dcurr) > 0
            Find the waypoint at which the joint stops, Ws
            DtoStop = Ws - Dcurr;
        Else
            DtoStop = 0;
        End If

        If (DtoStop = 0) or (Dcurr = 0)
            Return EndCondition = 0;
        End If

        Calculate the total distance of the motion Dtotal
        Dtotal = StartCondition + Dcurr + DtoStop;

        If (StartCondition + abs(Dcurr)) > Dtotal/2
            joint goes more than half way through the
            total distance by the end of this section

            If abs(Ws) < d3
                Joint must start accelerating in this
                Section. Remaining distance to stop is Ws
                Return EndCondition = abs(Ws);
            End If
        Else
            Joint covers less than half the distance
            By the end of this section

            If (StartCondition + abs(Dcurr)) < d3
                This is a short section, joint is
                Still accelerating
                Return -(StartCondition + abs(Dcurr))
            End If
        End If

        Otherwise, joint will be at full velocity at the end
        Of this section
        EndCondition = d3
    End If
End For

```

Figure A.4. Pseudocode for Determining the Section End Conditions.

Once the initial end conditions are determined, they are parsed again to determine if any of the motions will result in dynamic constraints being exceeded during multiple joint motion. If necessary, the end conditions are modified before being passed to the trajectory planner. The pseudocode for the algorithm is shown in Figure A.5.

```
Procedure: ModifyEndConditions

Given start, end conditions at the current section,
Find start, end conditions of the next section for each joint using
Procedure FindEndConditions

For each joint
    Find out if the next section will be a short section (i.e.
    starting and/or stopping through a short distance)

If a short section is present for any joint in the next section
    For each joint
        If (nextStartCondition > 0) AND (nextEndCondition > 0)
            Set currentEndCondition = 0
        End If
    End for
End If
```

Figure A.5. Pseudocode for Modifying the End Conditions