NASA/TM—1998-208845

ARL–TR–1855

U.S. ARMY
RESEARCH LABORATORY

A Framework for a Supervisory Expert System for Robotic Manipulators with Joint–Position Limits and Joint–Rate Limits

Arthur G.O. Mutambara
Florida A&M University-Florida State University College of Engineering, Tallahassee, Florida

Jonathan Litt
U.S. Army Research Laboratory, Lewis Research Center, Cleveland, Ohio

National Aeronautics and
Space Administration

Lewis Research Center

December 1998

Acknowledgments

The authors would like to thank NASA Lewis Research Center for a Summer Faculty Fellowship and the U.S. Army Strategic and Advanced Computing Center for supporting the research.

Trade names or manufacturers' names are used in this report for identification only. This usage does not constitute an official endorsement, either expressed or implied, by the National Aeronautics and Space Administration.

Available from

NASA Center for Aerospace Information
7121 Standard Drive
Hanover, MD 21076
Price Code: A03

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22100
Price Code: A03

A Framework for a Supervisory Expert System for Robotic Manipulators with Joint-Position Limits and Joint-Rate Limits

Arthur G.O. Mutambara
ME Department, FAMU-FSU College of Engineering
Tallahassee FL 32310-6046

Jonathan Litt
U.S. Army Research Laboratory
Vehicle Technology Center, NASA Lewis Research Center
Cleveland, OH 44135

Abstract

This report addresses the problem of path planning and control of robotic manipulators which have joint-position limits and joint-rate limits. The manipulators move autonomously and carry out variable tasks in a dynamic, unstructured and cluttered environment. The issue considered is whether the robotic manipulator can achieve all its tasks, and if it cannot, the objective is to identify the closest achievable goal. This problem is formalized and systematically solved for generic manipulators by using inverse kinematics and forward kinematics. Inverse kinematics are employed to define the subspace, workspace and constrained workspace, which are then used to identify when a task is not achievable. The closest achievable goal is obtained by determining weights for an optimal control redistribution scheme. These weights are quantified by using forward kinematics.

Conditions leading to joint rate limits are identified, in particular it is established that all generic manipulators have singularities at the boundary of their workspace, while some have loci of singularities inside their workspace. Once the manipulator singularity is identified the command redistribution scheme is used to compute the closest achievable Cartesian velocities. Two examples are used to illustrate the use of the algorithm: A three link planar manipulator and the Unimation Puma 560. Implementation of the derived algorithm is effected by using a supervisory expert system to check whether the desired goal lies in the constrained workspace and if not, to evoke the redistribution scheme which determines the constraint relaxation between end effector position and orientation, and then computes optimal gains.

1 Introduction

In this section the problem of optimal tracking of joint commands while experiencing saturated actuators is explained and discussed. Previous research work in this area is reviewed, in particular the work addressing the accommodation of actuator limits in multivariable systems and the Windup Feedback Scheme [1] is introduced.

1.1 Accommodation of Actuator Saturation

Actuator saturation can cause significant deterioration in control system performance because unmet demand may result in sluggish transients and oscillations in response to set-point changes. Generally, some type of linear control scheme is designed for a system such that the control responsibility is divided up among the actuators. When an actuator saturates, the linear controller may act in a nonlinear manner and abnormal performance can result. This performance may be unacceptable from a safety, cost, or quality standpoint.

An actuator is a physical system so its output, which under normal conditions is a function of its input, is restricted to lie within some boundaries. At the edge of its range of motion, additional input to drive it past the endpoint will have no effect. Thus, even though the control system is demanding more actuation, the effector cannot provide it. Additionally, because of power and wear considerations, the actuator must not be forced against its limit for extended periods. In order to assure this, the command to it must be limited so that it never tries to drive the actuator outside of its unrestricted range. As long as an actuator command stays within the normal bounds, there is no difference between the desired and achievable actuator positions. However, as an actuator command moves beyond the normal range it gets clipped, indicating that a portion of the control demand can not be met.

To help compensate for this problem, a technique has been developed which takes advantage of redundancy in multivariable systems to redistribute the unmet control demand over the remaining useful effectors [1]. This method, the Windup Feedback Scheme, is not a redesign procedure, rather it modifies commands to the effectors with remaining authority to compensate for those which are limited, thereby exploiting the built-in redundancy. The original commands are modified by the increments due to unmet demand, but when a saturated effector comes off its limit, the incremental commands disappear and the original unmodified controller remains intact. This scheme provides a smooth transition between saturated and unsaturated modes as it divides up the unmet requirement over any available actuators. This way, if there is sufficiently redundant control authority, performance can be maintained.

1.2 Accommodation of Manipulator Joint Limits

The work discussed above has been extended to establish a technique for compensating rate and position limits in the joints of a six degree-of-freedom robotic manipulator [2]. The unmet demand as a result of actuator saturation is redistributed among the remaining unsaturated joints. The scheme is used to compensate for inadequate path planning, problems such as joint limiting, joint freezing, or even obstacle avoidance, where a desired position and orientation are not attainable due to an unrealizable joint command. Once a joint encounters a limit, supplemental commands are sent to other joints to best track, according to a selected criterion, the desired trajectory. A standard six degree-of-freedom manipulator has

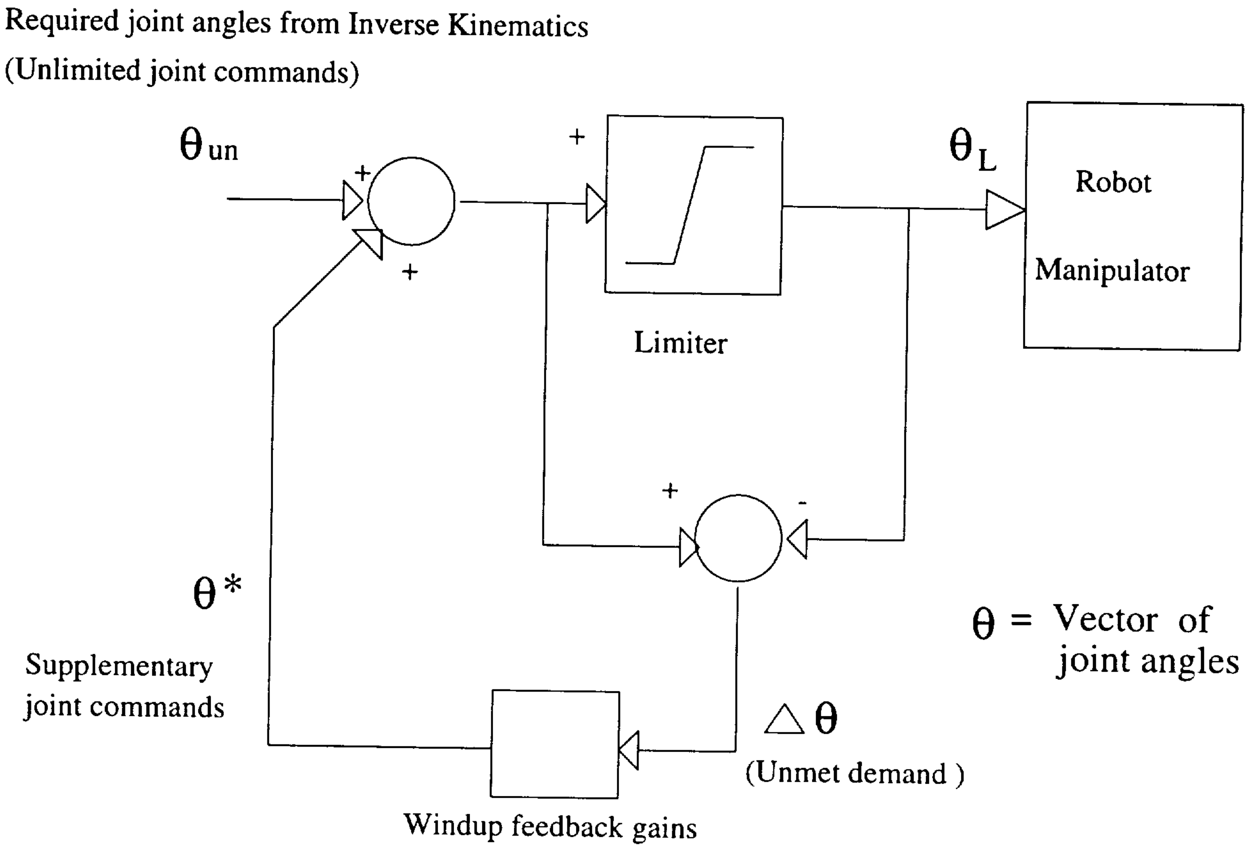

Figure 1: Windup Feedback Scheme

six independently controlled joints. The position and orientation of the end effector, each of which is described in three dimensions, are fully determined by the angles of the joints. As long as the appropriate joint angles are achievable, the desired position and orientation can be obtained. However, when the specified joint trajectories cannot be followed due to a command beyond the range of the actuator, positions and orientations downstream from the limited joint will all be affected, causing in some cases extreme deviations from the expected values. The Windup Feedback Scheme, shown in Figure 1, was designed to compensate for actuator saturation in a multivariable system by supplementing the commands to the remaining actuators to produce the desired effect on the output. In this case the output consists of the gripper position and orientation. For each joint which saturates, a degree of freedom is lost, but the remaining joints can be used to track the desired path within the physical limits of the manipulator.

2 Problems of Joint-Position Limits and Joint-Rate Limits

The objective of the work presented in this report is to systematically address the problem of path planning of robotic manipulators with *joint-position limits* and *joint-rate limits.* This entails a manipulator that moves autonomously in a dynamic environment and carries out variable tasks, including picking up randomly scattered items. The environment is unstructured and cluttered. The question considered is whether the robotic manipulator can achieve all of its desired goals, all the time. If it does not, what is the closest achievable goal and how can this be attained? This problem arises when the desired position and orientation of the manipulator end-effector requires joint-positions which are not attainable.

Beyond the path planning problem, in many applications it is important to reach a goal position and orientation with a particular (static) velocity. This raises the question of converting from Cartesian velocities (goal frame velocities) to the required joint-rates (velocities). The inverse of the Jacobian matrix is used to carry out this transformation. If the Jacobian matrix is not invertible it means the required Cartesian velocities are not achievable, implying joint-rate limits.

3 Manipulator Spaces and Kinematics

In this section the notions of actuator space, joint space, Cartesian space and manipulator kinematics are introduced. The identification of joint limits is then accomplished by using inverse kinematics to define the subspace, workspace and constrained workspace for robotic manipulators.

3.1 Actuator Space, Joint Space and Cartesian Space

The position of all the links of a manipulator of n degrees of freedom can be specified with a set of n joint variables. This set of variables is often referred to as the $n \times 1$ *joint vector.* The space of all such joint vectors is referred to as *joint space.* The Cartesian space is the space that contains all the end-effector positions and orientations where the position is measured along orthogonal axes. This space is also referred to as task-oriented space or operational space. Each kinematic joint is moved directly or indirectly by some sort of actuator. In some cases two actuators work together as a differential pair to move joints. The notion of actuator values leads to the definition of the *actuator space* as that space which contains all the actuator vectors, where an actuator vector is a set of actuator values.

3.2 Manipulator Kinematics

Kinematics is defined as the "geometry of motion", the branch of dynamics which treats motion without regard to forces which cause it. It studies position, velocity, acceleration and all higher order derivatives of position variables. Forward kinematics (F.K.) is the static geometrical problem of computing the position and orientation of the end-effector of the manipulator. Specifically, given a set of joint angles, the forward kinematic problem is to compute the position and orientation of the the tool frame relative to the base frame [3]. Put differently, forward kinematics involve changing the representation of the manipulator

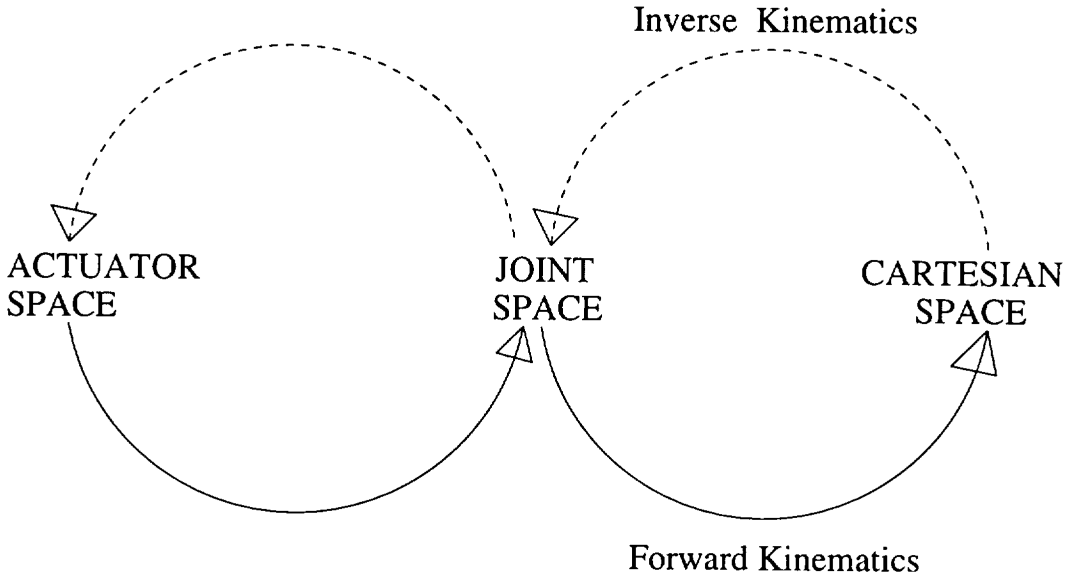

Figure 2: Manipulator Spaces and Kinematics

position from the joint space description into a Cartesian space description as illustrated in Figure 2. The inverse kinematics (I.K.) problem is posed as follows: Given the position and orientation of the end-effector of the manipulator, calculate all the possible sets of joint angles which could be used to attain this given position and orientation. This is a fundamental problem in the practical use of manipulators. This can also be understood as changing the representation of the manipulator position from the Cartesian space description into a joint space description as illustrated in Figure 2 .

3.3 Identification of Joint Limits by Inverse Kinematics

Unlike forward kinematics, the inverse kinematic problem is not simple. The inverse kinematics equations are nonlinear, their solution is not always easy or even possible in a closed form. Also the questions of existence of a solution, and of multiple solutions, arise. The issue of the existence of a solution or lack of it answers the question of whether goal (or task) is attainable.

Inverse kinematics are used to define *subspace, workspace* and *constrained workspace* for the robotic manipulator. The workspace is defined as that volume of space which the end-effector of the manipulator can reach with fixed joint lengths and *no joint limits* ($0 \leq \theta_j \leq 360\,\text{deg}$). Subspace is then defined as the workspace of a generic robot with *infinitely variable* joint lengths and no joint limits ($0 \leq \theta_j \leq 360\,\text{deg}$). Constrained workspace is defined as the workspace where the robotic manipulator has fixed joint lengths and joint limits ($\theta_{min} \leq \theta_j \leq \theta_{max}$) . These spaces are used to solve the problem of identifying when a task of the manipulator is not achievable. The relationship between these spaces is summarized as follows,

$$\text{SPACE} \supseteq \text{SUBSPACE} \supseteq \text{WORKSPACE} \supseteq \text{CONSTRAINED WORKSPACE}.$$

For example, a 6 DOFs manipulator's subspace is the entire 3-D space while its workspace is a portion of 3-D space and its constrained workspace is an even smaller portion of 3-D space. Similarly, a three link planar manipulator's subspace is the entire generic plane and its workspace and constrained workspace are decreasing portions of the plane.

The issue of the closest achievable goal is then addressed by using systematically derived weights in the Windup Feedback Scheme. The weights determine the feedback gains shown in Figure 1. A derivation of their relationship with the optimal gains appears in previous work [2]. These weights are obtained and quantified by using forward kinematics, which are used to show which joint angles affect position and orientation. Thus defining, for example, which joint angle requirements may be compromised or ignored in order to obtain the goal position while being flexible about the orientation. In a path planning problem the position is more critical than the orientation. Orientation becomes important when the problem of obstacle avoidance is also considered.

4 Resolution of Joint Rate Limits

When the Jacobian matrix is not invertible it means the required Cartesian velocities are not achievable, implying joint rate limits. The conditions under which these limits (singularities of manipulator mechanisms) occur, are systematically identified. All generic manipulators have singularities at the boundary of their workspace, while some have loci of singularities inside their workspace [3]. As indicated before, the manipulator workspace is derived by using inverse kinematics. A workspace boundary singularity occurs when the manipulator is fully stretched out or folded back. This occurs when the end effector is near or at the boundary of its workspace. A workspace interior singularity occurs away from the workspace boundary, and is caused by two or more joint axes lining up. A manipulator in a singular configuration has lost one or more degrees of freedom. Once the manipulator singularity is identified the Windup Feedback Scheme is used to compute the closest achievable Cartesian velocities (goal frame velocities). The Jacobian is a function of joint angles which transforms joint velocities into Cartesian velocities. The inverse of the Jacobian function facilitates the reverse operation, i.e., transforms the Cartesian velocities into joint velocities (joint-rates).

$$\mathbf{V}(k) = \mathbf{J}\{\Theta(k)\}\dot{\Theta}(k) \tag{1}$$

$$\Longrightarrow \dot{\Theta}(k) = \mathbf{J}^{-1}\{\Theta(k)\}\mathbf{V}(k) \tag{2}$$

Consequently, if the Jacobian is not invertible it means that there are joint-rate limits (singularities of the mechanism).

5 Expert Systems Concepts

An *expert system* is a computer program using expert knowledge to attain high levels of performance in a narrow problem area. The process of building expert systems is called *Knowledge Engineering.* This is an integral part of the field of *Artificial Intelligence*; a part of Computer Science involving the development of intelligent computer programs . Backward chaining is an inference method where the system starts with what it wants to prove, e.g. Z, and tries to establish the facts it needs to prove Z. Forward chaining is an inference method where rules are matched against facts to establish new facts. Expert systems exhibit intelligent behavior by skillful application of heuristics.

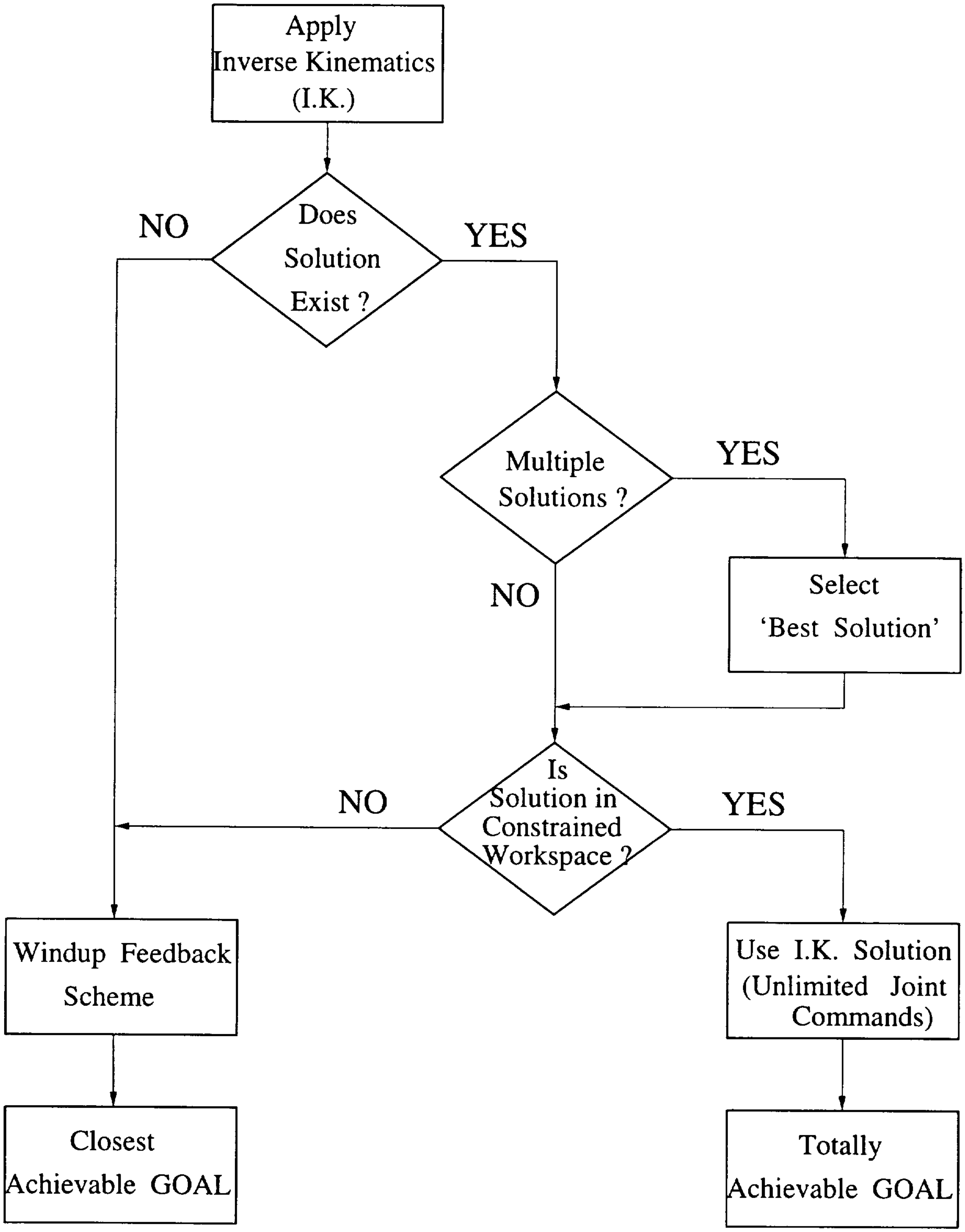

Figure 3: The Role of The Expert System

Expert Systems apply expert knowledge to real world problems. Such applications include diagnosis, interpretation, prediction, design, planning, control and supervision. Expert systems are advantageous over human experts because their knowledge is permanent, easy to transfer, easy to document, consistent, predictable and affordable. However they also have disadvantages with respect to human experts; lack of creativity, inability to adapt and the absence of both common sense knowledge and sensory experience.

5.1 A Supervisory Expert System

Implementation of the derived algorithms was done by way of a *supervisory expert system.* In the application of interest, the expert system is used to check whether the *goal* lies in the *constrained workspace*, if not the expert system evokes the Windup Scheme, decides on the constraint relaxation between end effector position and orientation (depending on task) and then systematically computes and assigns Windup Feedback gains. The expert system design and implementation was carried out in the language CLIPS 5.1 and LISP.

5.2 An Expert System for the Windup Feedback Scheme

The problem of path planning when there are joint-position limits is resolved by the supervisory expert system. The flowchart in Figure 3 illustrates and summarizes the role of the expert system. The criteria used in choosing the 'best solution' include the following factors: closest solution, minimum energy, fastest option and the movement of smaller joints first. If the goal does not lie in the constrained workspace this means that the goal is not totally attainable, i.e, $\Theta_l(k) \neq \Theta_{un}(k)$. In this case the Windup Feedback Scheme is then evoked. The expert system determines constraint relaxation between end effector position and orientation, depending on the specific task. In this way, it systematically computes the Windup Feedback gains.

6 Examples

Two detailed examples are used to illustrate and demonstrate the supervisory expert system algorithm; A three link planar manipulator and the *Unimation PUMA 560.* These robotic manipulators were chosen because they have simple kinematics which adequately manifest the problems of joint-position limits and joint-rate limits.

6.1 A Three Link Planar Manipulator (3R Mechanism)

The three link planar manipulator moves in a plane and consists of three links, three revolute joints and three parallel axes of rotation as illustrated in Figure 4. Its subspace is the generic plane and its workspace a circular plane defined by the links and joints. L_1, L_2 and L_3 are the link lengths, x and y represent the position of the base of the end-effector, and ϕ is the orientation of the end-effector. The joint angles (revolute joints) are represented by θ_1, θ_2 and θ_3.

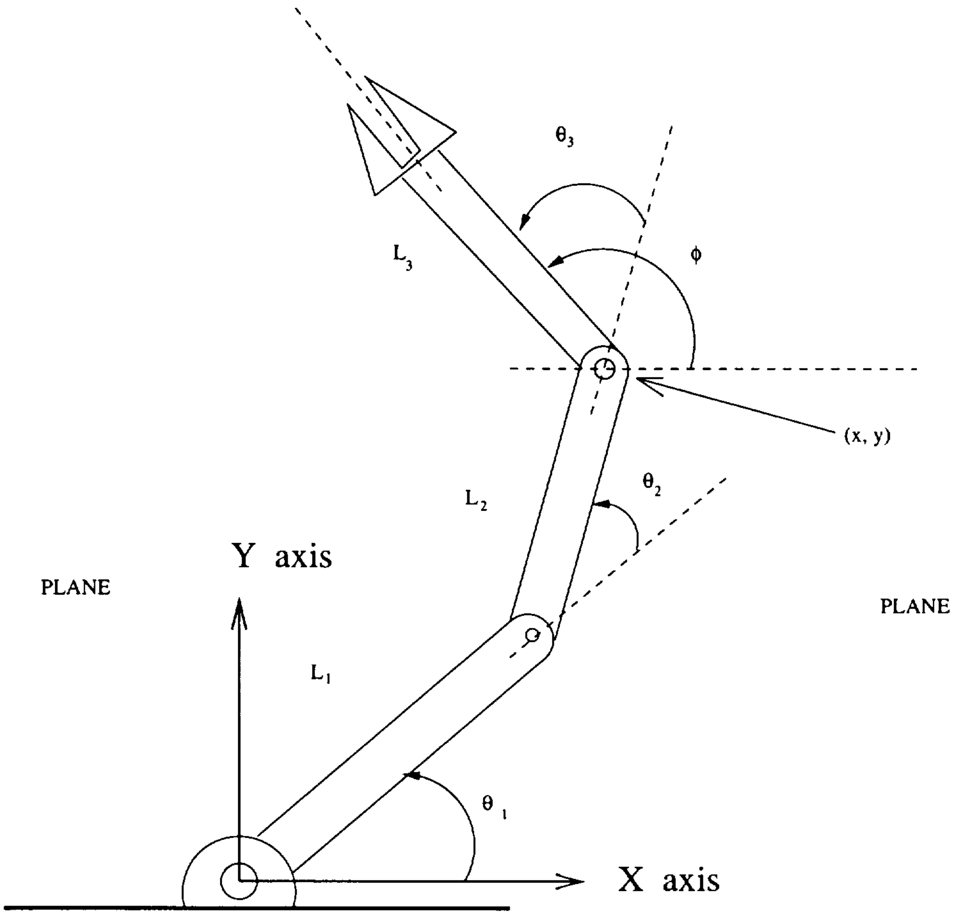

Figure 4: Three Link Planar Manipulator

Subspace

The subspace of the three link planar manipulator is the generic plane which is represented by the following structure

$$\begin{bmatrix} cos\phi & -sin\phi & 0 & x \\ sin\phi & cos\phi & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

where the variables x, y, and ϕ take arbitrary values.

Workspace

The corresponding workspace for the three link planar manipulator, when there are no joint limits, is of the form

$$\begin{bmatrix} cos(\theta_1+\theta_2+\theta_3) & -sin(\theta_1+\theta_2+\theta_3) & 0 & L_1cos\theta_1 + L_2cos(\theta_1+\theta_2) \\ sin(\theta_1+\theta_2+\theta_3) & cos(\theta_1+\theta_2+\theta_3) & 0 & L_1sin\theta_1 + L_2sin(\theta_1+\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

$$\begin{array}{c} \text{No joint} \\ \text{limits} \end{array} \Longrightarrow \begin{array}{l} 0 \leq \theta_1 \leq 360^0 \\ 0 \leq \theta_2 \leq 360^0 \\ 0 \leq \theta_3 \leq 360^0 \end{array} \; where \, L_1, \; L_2, \; L_3 \; \equiv \text{constants}$$

Constrained Workspace

When there are joint limits the workspace is reduced, resulting in the constrained workspace which is represented as follows

$$\begin{bmatrix} cos(\theta_1+\theta_2+\theta_3) & -sin(\theta_1+\theta_2+\theta_3) & 0 & L_1cos\theta_1 + L_2cos(\theta_1+\theta_2) \\ sin(\theta_1+\theta_2+\theta_3) & cos(\theta_1+\theta_2+\theta_3) & 0 & L_1sin\theta_1 + L_2sin(\theta_1+\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{array}{c} \text{Joint} \\ \text{limits} \end{array} \Longrightarrow \begin{array}{l} \theta_{1_{min}} \leq \theta_1 \leq \theta_{1_{max}} \\ \theta_{2_{min}} \leq \theta_2 \leq \theta_{2_{max}} \\ \theta_{3_{min}} \leq \theta_3 \leq \theta_{3_{max}} \end{array} \; where \, L_1, \; L_2, \; L_3 \; \equiv \text{constants}$$

Inverse Kinematic Solution

The generic end-effector frame or general goal is given by the generic homogeneous transformation matrix,

$${}^{B}_{W}T = \left[\begin{array}{ccc:c} & & & \vdots \\ & {}^{A}_{B}R & & \vdots \quad {}^{A}P_{B_{ORG}} \\ & & & \vdots \\ \hdashline 0 & 0 & 0 & \vdots \quad 1 \end{array}\right]$$

$${}^{B}_{W}T = \left[\begin{array}{ccc:c} r_{11} & r_{12} & r_{13} & x \\ r_{21} & r_{22} & r_{23} & y \\ r_{31} & r_{32} & r_{33} & z \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right] \quad \begin{array}{c} \text{True for any} \\ \text{end-effector} \\ \text{in 3-D Space} \end{array}$$

The following are the conditions to be satisfied for a solution to exist:

(1) ${}^{B}_{W}T$ = SUBSPACE (must have the structure of the SUBSPACE)

(2) ${}^{B}_{W}T$ = WORKSPACE (must be solvable for joint angles $\theta_1, \; \theta_2 \cdots \theta_n$)

(3) ${}^{B}_{W}T$ = CONSTRAINED WORKSPACE (must be solvable for acceptable joint angles $\theta_1, \; \theta_2 \cdots \theta_n$)

Condition (3) is the tightest test. It is the *necessary and sufficient* condition.

For the 3-link planar manipulator

$$ {}^{B}_{W}T = \begin{bmatrix} cos\phi & -sin\phi & 0 & x \\ sin\phi & cos\phi & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \equiv \text{SUBSPACE} $$

Constrained Workspace Test

To test whether required joint angles lie within the constrained workspace the subspace is equated to the constrained workspace. The resulting equations are then solved for the joint angles θ_1, θ_2 and θ_3.

$$ \begin{bmatrix} c\phi & -s\phi & 0 & x \\ s\phi & c\phi & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_{123} & -s_{123} & 0 & L_1c_1 + L_2c_{12} \\ s_{123} & c_{123} & 0 & L_1s_1 + L_2s_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

The operator Atan2(y, x) computes the inverse tangent function, $\tan^{-1}(y/x)$, but uses the signs of both x and y to determine the quadrant in which the resulting angle lies. Hence, the inverse kinematic solution gives the following results:

$\theta_2 = \text{Atan2}(s_2,\ c_2)$

$\theta_1 = \text{Atan2}(y,\ x)$ - $\text{Atan2}(k_1,\ k_2)$

$\theta_3 = \phi - \theta_1 - \theta_2$

$c_2 = \frac{x^2+y^2-L_1^2-L_2^2}{2L_1L_2}$

$s_2 = \pm\sqrt{1-c_2^2}$

$k_1 = L_1 + L_2c_2$

$k_2 = L_2s_2$

The following are the conditions required to be met for these solution equations to hold (exist).

(a) $L_1 > 0,\ L_2 > 0$

(b) $-1 \leq \frac{x^2+y^2-L_1^2-L_2^2}{2L_1L_2} \leq 1$

$\Rightarrow \theta_2 = \text{Atan } 2(s_2, c_2)$

(c) $\boxed{\theta_{2_{min}} \leq \theta_2 \leq \theta_{2_{max}}}$ joint limits test.

(d) if x = y = 0 $\Rightarrow \theta_1$ is arbitrary $0 \leq \theta_1 \leq 360^0$
else θ_1 =Atan2(y, x) - Atan2 (k_2, k_1)

(e) $\boxed{\theta_{1_{min}} \leq \theta_1 \leq \theta_{1_{max}}}$

$\Rightarrow \theta_3 = \phi - \theta_2 - \theta_1$

(f)$\boxed{\theta_{3_{min}} \leq \theta_3 \leq \theta_{3_{max}}}$

Determination of Weights for Windup Scheme by Expert System

The *goal* is expressed as

$$\begin{bmatrix} x \\ y \\ \phi \end{bmatrix} = \begin{bmatrix} L_1 c_1 + L_2 c_{12} \\ L_1 s_1 + L_2 s_{12} \\ \theta_1 + \theta_2 + \theta_3 \end{bmatrix},$$

where x, y define the position of the base of the end-effector and ϕ defines the orientation of the end-effector. Only θ_1 and θ_2 are important for the position. The joint angles θ_1, θ_2 and θ_3 are all important for the orientation. Consequently, weights can be used to penalize errors in θ_1 and θ_2 for a path planning problem. Since θ_3 does not affect the position, errors in it are less important. Forward kinematics are used to establish how position and orientation depend on joint angles, such that joint angles that do not affect position are identified. The expert system uses these facts in determining the weights of the Windup Feedback Scheme.

6.2 The Unimation PUMA 560 (6 DOFs Manipulator)

The Puma 560 is a rotary joint manipulator with six revolute joints and six degrees of freedom. Its joint axes 4, 5 and 6 all intersect at a common point. Furthermore, these joint axes 4, 5 and 6 are all mutually orthogonal establishing the Puma wrist mechanism. The frame assignments for a general PUMA 560 are shown in Figure 5, where frame 0 and frame 1 are set as coincident. The subspace of the PUMA is the generic 3-D space, but its workspace is limited by its link lengths and joint limits to a portion of 3-D space.

Subspace

The subspace of the PUMA is the generic 3-D space which is represented as follows

$${}^B_W T = {}^0_6 T = \left[\begin{array}{ccc:c} r_{11} & r_{12} & r_{13} & x \\ r_{21} & r_{22} & r_{23} & y \\ r_{31} & r_{32} & r_{33} & z \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right].$$

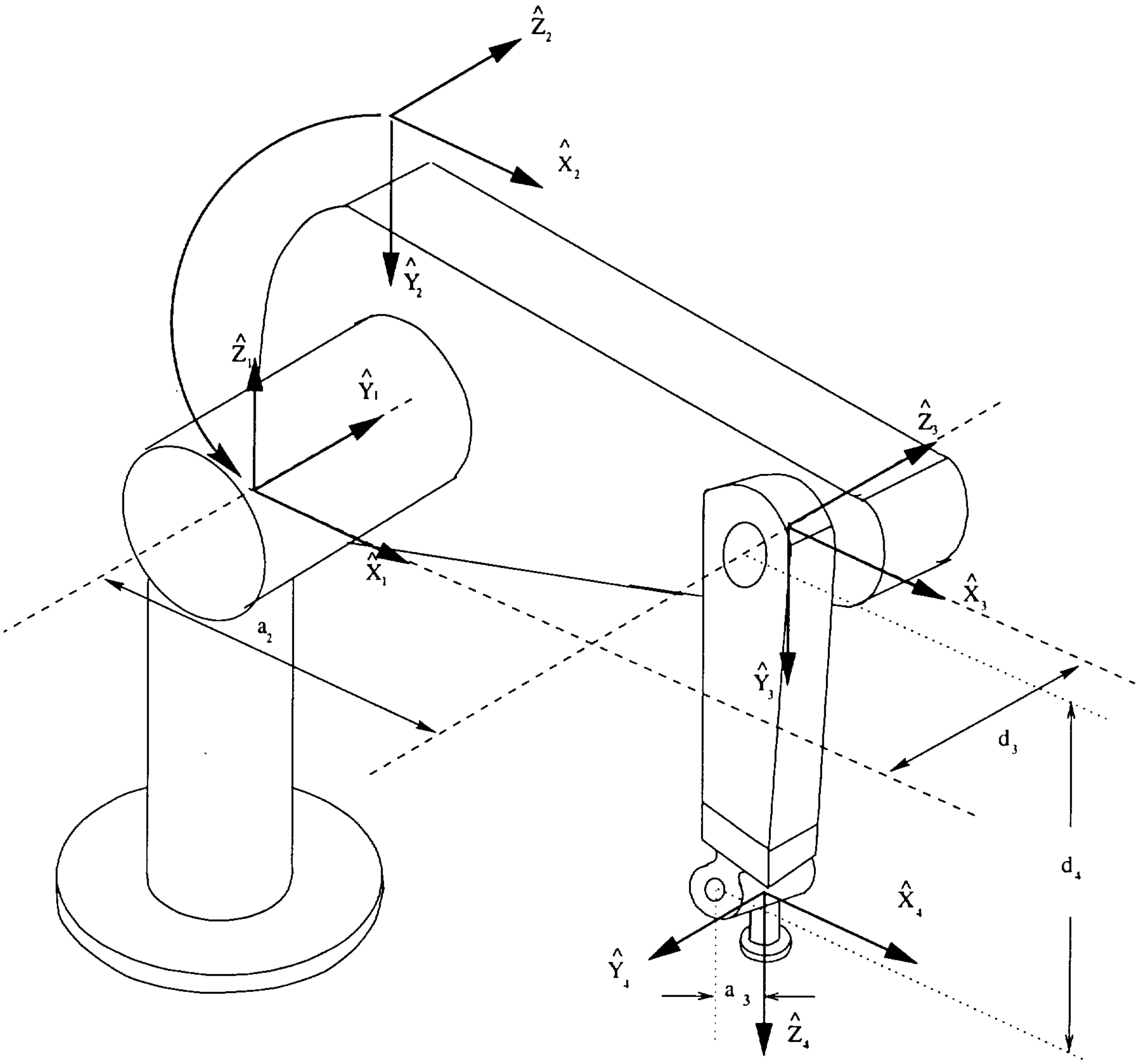

Figure 5: Frame assignments for the PUMA 560 manipulator

This is the general goal frame of the end-effector. The workspace is of the same form but it is limited to a portion of 3-D space by the finite link lengths of the PUMA.

Constrained Workspace

When there are joint limits the PUMA workspace is reduced to a constrained a workspace of the form

$$^0_1T(\theta_1)\ ^1_2T(\theta_2)\ ^2_3T(\theta_3)\ ^3_4T(\theta_4)\ ^4_5T(\theta_5)\ ^5_6T(\theta_6),$$

where

$$ {}^{0}_{1}T = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{1}_{2}T = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s\theta_2 & -c\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{2}_{3}T = \begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & a_2 \\ s\theta_3 & c\theta_3 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{3}_{4}T = \begin{bmatrix} c\theta_4 & -s\theta_4 & 0 & a_3 \\ 0 & 0 & 1 & d_4 \\ -s\theta_4 & -c\theta_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{4}_{5}T = \begin{bmatrix} c\theta_5 & -s\theta_5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s\theta_5 & c\theta_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{5}_{6}T = \begin{bmatrix} c\theta_6 & -s\theta_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s\theta_6 & -c\theta_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} . $$

The joints are limited such that $\Theta_{min} \leq \Theta \leq \Theta_{max}$, where the vector of joint angles is given by

$$ \Theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \end{bmatrix} . $$

Constrained Workspace Test

To test whether required joint angles lie within the constrained workspace the subspace is equated to the constrained workspace as follows,

$$\left[\begin{array}{ccc:c} r_{11} & r_{12} & r_{13} & x \\ r_{21} & r_{22} & r_{23} & y \\ r_{31} & r_{32} & r_{33} & z \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right] = {}^{0}_{1}T(\theta_1)\ {}^{1}_{2}T(\theta_2)\ {}^{2}_{3}T(\theta_3)\ {}^{3}_{4}T(\theta_4)\ {}^{4}_{5}T(\theta_5)\ {}^{5}_{6}T(\theta_6).$$

The resulting equations are solved for the joint angles θ_1, θ_2, θ_3, θ_4, θ_5 and θ_6, such that the following condition is satisfied,

$$\Theta_{min} \leq \Theta \leq \Theta_{max}. \tag{3}$$

As a consequence the following results are obtained,

$$\theta_1 = \text{Atan2}(y,\ x) - \text{Atan2}(d_3,\ \pm\sqrt{x^2 + y^2 - d_3^2})$$

$$\theta_2 = \theta_{23} - \theta_3$$

$$\theta_{23} = \text{Atan2}[(-a_3 - a_2c_3)z - (c_1x + s_1y)(d_4 - a_2s_3),\ (a_2s_3 - d_4)z - (a_3 + a_2c_3)(c_1x + s_1y)]$$

and

$$\theta_3 = \text{Atan2}(a_3,\ d_4) - \text{Atan2}(K,\ \pm\sqrt{a_3^2 + d_4^2 - K^2})$$

$$K = \frac{x^2 + y^2 + z^2 - a_2^2 - a_3^2 - d_3^2 - d_4^2}{2a_2}$$

$$\theta_4 = \text{Atan2}(-r_{13}s_1 + r_{23}c_1,\ -r_{13}c_1c_{23} - r_{23}s_1c_{23} + r_{33}s_{23})$$

$$\theta_5 = \text{Atan2}(s_5,\ c_5)$$

$$\theta_6 = \text{Atan2}(s_6,\ c_6)$$

$$s_6 = -r_{11}[c_1c_{23}s_4 - s_1c_4] - r_{21}[s_1c_{23}s_4 + c_1c_4] + r_{31}[s_{23}s_4]$$

$$\begin{aligned} c_6 = {} & r_{11}[(c_1c_{23}c_4 + s_1s_4)c_5 - c_1s_{23}s_5] + r_{21}[(s_1c_{23}c_4 - c_1s_4)c_5 - s_1s_{23}s_5] \\ & - r_{31}[s_{23}c_4c_5 + c_{23}s_5] \end{aligned}$$

The plus-or-negative signs appearing in the expressions of θ_1 and θ_3 lead to four solutions. Additionally there are four more solutions obtained by flipping the wrist of the manipulator. For each of the four solutions computed above, the flipped solution is given by

$$\begin{aligned} \theta_4' &= \theta_4 + 180^0 \\ \theta_5' &= -\theta_5 \\ \theta_6' &= \theta_6 + 180^0. \end{aligned}$$

For a given position and orientation of the PUMA end-effector there are eight I.K solutions, i.e, eight sets of joint angles. Four of these are shown in Figure 6. Some or all of the

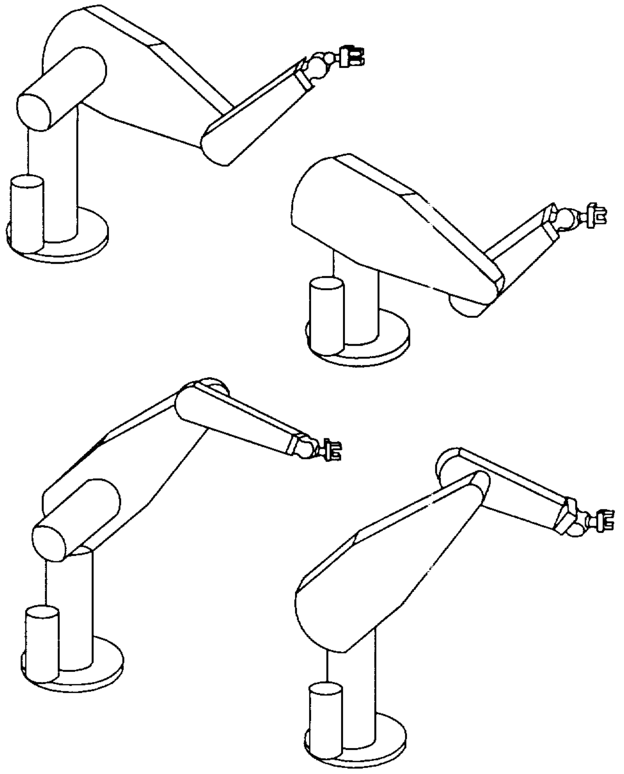

Figure 6: Four I.K. Solutions for the PUMA 560

solutions may be discarded because of joint limit violations. If all the solutions are discarded then the Windup Feedback Scheme is used to obtain tl.e closest achievable solution. If there are multiple solutions the closest solution is picked whe :e the criteria of choice might include: least energy use, obstacle avoidance, movement of smaller links and least movement of links.

Determination of Windup Feedback Weights Using Expert System

Equating the following two matrices and solving the equations produced allows the determination of the Windup Feedback Scheme weights.

$$\left[\begin{array}{ccc:c} r_{11} & r_{12} & r_{13} & x \\ r_{21} & r_{22} & r_{23} & y \\ r_{31} & r_{32} & r_{33} & z \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right] = \left[\begin{array}{ccc:c} a_{11} & a_{12} & a_{13} & P_x \\ a_{21} & a_{22} & a_{23} & P_y \\ a_{31} & a_{32} & a_{33} & P_z \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right]$$

$$a_{11} = c_1[c_{23}(c_4c_5c_6 - s_4s_6) - s_{23}s_5c_6] + s_1(s_4c_5c_6 + c_4s_6)$$

$$a_{12} = c_1[c_{23}(-c_4c_5s_6 - s_4c_6) + s_{23}s_5s_6] + s_1(c_4c_6 - s_4c_5s_6)$$

$$a_{13} = -c_1(c_{23}c_4s_5 + s_{23}c_5) - s_1s_4s_5$$

$$a_{21} = s_1[c_{23}(c_4c_5c_6 - s_4s_6) - s_{23}s_5c_6] - c_1(s_4c_5c_6 + c_4s_6)$$

$$a_{22} = s_1[c_{23}(-c_4c_5s_6 - s_4c_6) + s_{23}s_5s_6] - c_1(c_4c_6 - s_4c_5s_6)$$

$$a_{23} = -s_1(c_{23}c_4s_5 + s_{23}c_5) + c_1s_4s_5$$

$$a_{31} = -s_{23}(c_4c_5c_6 - s_4s_6) - c_{23}s_5c_6$$

$$a_{32} = -s_{23}(-c_4c_5s_6 - s_4c_6) + c_{23}s_5s_6$$

$$a_{33} = s_{23}c_4s_5 - c_{23}c_5$$

$$P_x = x = c_1[a_2c_2 + a_3c_{23} - d_4s_{23}] - d_3s_1$$

$$P_y = y = s_1[a_2c_2 + a_3c_{23} - d_4s_{23}] + d_3c_1$$

$$P_z = z = -a_3s_{23} - a_2s_2 - d_4c_{23}$$

Position of tool base depends on θ_1, θ_2 and θ_3 only.

It is important to note that the coordinates $[x\ y\ z]^T$ specify the position while the angles $[\gamma\ \beta\ \alpha]^T$ (roll, pitch and yaw) specify the orientation. This is the general representation of a generic manipulator in 3-D space. For the PUMA 560 these angles are computed as follows,

$$\beta = \text{Atan2}(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2})$$

$$\alpha = \text{Atan2}(r_{21}/c\beta, r_{11}/c\beta)$$

$$\gamma = \text{Atan2}(r_{32}/c\beta, r_{33}/c\beta).$$

From forward kinematics, the position coordinates are functions of joint angles θ_1, θ_2 and θ_3 only, while the orientation is a function of all joint angles. Thus, the feedback weights are computed while taking cognizance of these facts. For example, in a path planning problem (where the orientation is not critical) errors in θ_1, θ_2 and θ_3 are more heavily penalized than errors in θ_4, θ_5 and θ_6.

7 Conclusions

The problems of joint-position limits and joint-rate limits in robotic manipulators have been addressed by using inverse kinematics and forward kinematics in conjunction with a supervisory expert system. Inverse kinematics were employed to define the subspace, the workspace and the constrained workspace, which were then used to identify whether or not a manipulator task is achievable. The closest achievable goal is obtained by using weights in the conventional Windup Feedback Scheme where these weights are quantified by using forward kinematics. It has been shown that robotic manipulators have singularities at the boundaries of their workspace, while some have loci of singularities inside the workspace. At the manipulator singularity, the Windup Feedback Scheme is used to compute the closest achievable Cartesian velocities. A three link planar robotic manipulator and the Unimation Puma 560 were effectively used to illustrate the theory developed. Future work might include considering robot manipulator dynamics and forces that cause motion which are neglected in kinematics, i.e., go beyond static (joint and Cartesian) positions, static forces and static velocities.

Acknowledgment

The authors would like to thank the NASA Lewis Research Center for a Summer Faculty Fellowship and the U.S. Army Strategic and Advanced Computing Center for supporting the research.

References

[1] J. Litt and G. Roulette, "A method for exploiting redundancy to accommodate actautor limits in multivariable systems," in *NASA Technical Memorandum 106859, ARL Technical Report 906*, 1994.

[2] J. Litt, A. Hickman, and T. Guo, "A new technique for compensating joint limits in a robot manipulator," in *NASA Technical Memorandum 107330, ARL Technical Report 1101*, 1996.

[3] J. J. Craig, *Introduction to Robotics: Mechanics and Control*, Addison-Wesley, 1989.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE December 1998	3. REPORT TYPE AND DATES COVERED Technical Memorandum

4. TITLE AND SUBTITLE

A Framework for a Supervisory Expert System for Robotic Manipulators with Joint-Position Limits and Joint-Rate Limits

5. FUNDING NUMBERS

WU-505-23-0Z-00
1L161102AH45

6. AUTHOR(S)

Arthur G.O. Mutambara and Jonathan Litt

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

NASA Lewis Research Center
Cleveland, Ohio 44135-3191
and
U.S. Army Research Laboratory
Cleveland, Ohio 44135-3191

8. PERFORMING ORGANIZATION REPORT NUMBER

E-11458

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, DC 20546-0001
and
U.S. Army Research Laboratory
Adelphi, Maryland 20783-1145

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA TM—1998-208845
ARL-TR-1855

11. SUPPLEMENTARY NOTES

Arthur G.O. Mutambara, Florida A&M University-Florida State University College of Engineering, Tallahassee, Florida 32310-6046; and Jonathan Litt, U.S. Army Research Laboratory, NASA Lewis Research Center. Responsible person, Jonathan Litt, organization code 0300, (216) 433-3748.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Category: 63 Distribution: Nonstandard

This publication is available from the NASA Center for AeroSpace Information, (301) 621-0390.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 words)***

This report addresses the problem of path planning and control of robotic manipulators which have joint-position limits and joint-rate limits. The manipulators move autonomously and carry out variable tasks in a dynamic, unstructured and cluttered environment. The issue considered is whether the robotic manipulator can achieve all its tasks, and if it cannot, the objective is to identify the closest achievable goal. This problem is formalized and systematically solved for generic manipulators by using inverse kinematics and forward kinematics. Inverse kinematics are employed to define the subspace, workspace and constrained workspace, which are then used to identify when a task is not achievable. The closest achievable goal is obtained by determining weights for an optimal control redistribution scheme. These weights are quantified by using forward kinematics. Conditions leading to joint rate limits are identified, in particular it is established that all generic manipulators have singularities at the boundary of their workspace, while some have loci of singularities inside their workspace. Once the manipulator singularity is identified the command redistribution scheme is used to compute the closest achievable Cartesian velocities. Two examples are used to illustrate the use of the algorithm: A three link planar manipulator and the Unimation Puma 560. Implementation of the derived algorithm is effected by using a supervisory expert system to check whether the desired goal lies in the constrained workspace and if not, to evoke the redistribution scheme which determines the constraint relaxation between end effector position and orientation, and then computes optimal gains.

14. SUBJECT TERMS Robotics; Expert systems; Artificial intelligence	15. NUMBER OF PAGES 24 16. PRICE CODE A03

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT