New Real-Time Robot Motion Algorithms Using Parallel VLSI Architectures

Arati S. Deo*
Joseph R. Cavallaro*
Ian D. Walker*

Abstract. Real time robot control presents a major numerical challenge. The inverse kinematics problem of determining joint motion for a specified end effector trajectory is formulated as a damped least-squares problem which balances accuracy against feasibility. This approach leads to the Singularity Robust Inverse which yields feasible joint motions even at or in the neighborhood of undesirable singular configurations. We present a new technique that optimally approximates the desired end-effector trajectory with physically realizable joint velocities at all manipulator configurations. This technique uses the Levenberg-Marquardt algorithm to compute an optimal damping factor. The Singular Value Decomposition (SVD) of the manipulator Jacobian plays a key role in this algorithm which is computationally intensive and currently limited to off-line planning. This new algorithm will be run on a parallel VLSI architecture under development at Rice. This system includes a custom CORDIC VLSI array for computing the SVD of a matrix. The array is connected to a linear array of Texas Instruments TMS320C30 DSP processors. The DSP array computes many of the matrix operations in parallel and uses the CORDIC SVD array as a co-processor.

1. Introduction. A manipulator arm consists of several rigid links connected usually in series by joints. An end-effector (hand) is attached to the end of the arm. The positions of the joints determine the configuration of the arm. An important parameter of a manipulator is the number of degrees of freedom that it possesses. Robotic arms which have more degrees of freedom than are required by the specified task are termed redundant. These arms have an infinite choice of joint configurations corresponding to any end-effector position. A comprehensive summary of the issues involved in the control of redundant manipulators can be found in [9].

A critical problem, that of calculating necessary positions, velocities and accelerations of the joints from given kinematic parameters of the end-effector is called Inverse Kinematics. This involves the determination of the arm configuration given the position and orientation of the hand. For most manipulators, a closed-form inverse kinematic function does not exist at the position level. As a result, inverse kinemat-

* Department of Electrical and Computer Engineering, Rice University, Houston, TX 77251. This work was supported in part by the National Science Foundation under grant MIP-8909498 and under grant MSS-9004391.
ics is usually carried out at the velocity or acceleration level. Inverse kinematics of redundant manipulators is difficult ([9], [13]) due to nonuniqueness of solution.

The damped least-squares technique has been used to damp the norm of the joint velocities and joint accelerations of the manipulator using a suitable damping factor. In the existing literature on this technique, approximate methods of estimating the damping factor have been used. In this paper we show how the Levenberg-Marquardt algorithm can be used to compute an optimal damping factor.

2. Inverse Kinematics at Velocity Level. The direct kinematic relation of a robotic manipulator at the velocity level [13] is given by

\[ \dot{\mathbf{x}} = J(\mathbf{q}) \dot{\mathbf{q}} \]

Here \( \dot{\mathbf{x}} \) is the \( m \times 1 \) end-effector velocity vector while \( \dot{\mathbf{q}} \) is an \( n \times 1 \) vector representing the joint velocities. \( J(\mathbf{q}) \) is the manipulator Jacobian matrix with dimensions \( m \times n \).

For non-redundant manipulators, \( m = n \) while for redundant manipulators \( m < n \). The conventional method of computing the inverse kinematics of a manipulator at the velocity level from the above relation ([9], [13]) is

Non-redundant manipulator : \[ \dot{\mathbf{q}} = J^{-1} \dot{\mathbf{x}} \]

Redundant manipulator : \[ \dot{\mathbf{q}} = J^+ \dot{\mathbf{x}} = J^T (JJ^T)^{-1} \dot{\mathbf{x}} \]

where it is assumed that \( \text{rank}(J) = m \). Here \( J^{-1} \) is the inverse while \( J^+ \) is the pseudoinverse of the Jacobian. The solution \( J^{-1} \dot{\mathbf{x}} \) is unique up to finite variations for non-redundant arms. For redundant arms, \( J^+ \dot{\mathbf{x}} \) has minimum norm among the infinite number of joint velocity vectors that satisfy (1). In the following, all discussion related to the pseudoinverse also applies to the inverse in case of non-redundant arms since the pseudoinverse and inverse are equivalent for a square Jacobian (full rank).

All manipulators have configurations in their workspace, known as singular configurations, at which their freedom of movement is restricted due to peculiar kinematic alignment of the links. At such configurations, extremely high joint velocities are required even for small end-effector motions in the restricted or singular directions. The Jacobian gets ill-conditioned as the arm approaches a singular configuration.

The pseudoinverse fails to prevent high joint velocities (in the neighborhood of singular configurations) because it chooses the minimum-norm joint velocity vector among all the accurate ones. In the neighborhood of singular configurations, it is extremely likely that all solutions satisfying (1) will be infeasible in which case, the pseudoinverse could yield infeasible joint velocities. This equality constraint imposed by (1) needs to be relaxed to obtain feasible solutions near singularities.

2.1. Damped Least-Squares Technique. The Damped Least-Squares formulation compromises between accuracy and feasibility of the solution.

\[
\text{DLP-1} \quad \min \{ \| \dot{\mathbf{x}} - J \dot{\mathbf{q}} \|^2 + \lambda \| \dot{\mathbf{q}} \|^2 \}
\]

where \( \lambda \) is known as the damping factor. This formulation has been found to be effective in tackling the problem of an ill-conditioned Jacobian resulting in high joint velocities near singularities. It involves a trade-off between the accuracy with which the desired end-effector trajectory is followed and the feasibility of the joint velocities. The trade-off parameter is the damping factor \( \lambda \).

The DLP-1 problem does not give any insight into how the damping factor should be determined. Hence we formulate a different problem which gives the same solution as DLP-1 but also indicates the method to compute an optimal damping factor. Since
we wish to keep joint velocities feasible at all configurations while tracking the end-effector trajectory as accurately as possible, we solve the following problem ([3], [6]).

\textbf{DLP-2} \quad \begin{align*}
\min & \quad \|\dot{x} - J\ddot{\theta}\|^2 \\
\text{subject to} & \quad \|\ddot{\theta}\|^2 \leq \Delta^2
\end{align*}

This problem defines a joint velocity vector to be feasible if its norm is less than or equal to a maximum permissible value \(\Delta\). The solution to DLP-1 and DLP-2 is

\begin{equation}
\dot{\theta}^*_v = (J^TJ + \lambda I)^{-1}J^T\ddot{x}.
\end{equation}

In addition, from DLP-2 we obtain the condition \(\lambda \geq 0\) and \(\lambda(\|\dot{\theta}^*_v\|^2 - \Delta) = 0\). The matrix \(J^* = (J^TJ + \lambda I)^{-1}J^T\) is called the Singularity Robust Inverse (SRI). Note that in DLP-2, the damping factor \(\lambda\) is actually the Lagrange multiplier.

Unfortunately, the matrix \((J^TJ + \lambda I)\) is never invertible for \(\lambda = 0\) for redundant manipulators since for such arms (1) forms an underdetermined system of equations. As a result, the solution (2) is not applicable to redundant manipulators in the case of zero damping i.e. when the pseudoinverse yields a feasible solution. The more appropriate form of the SRI for redundant manipulators [3] is

\begin{equation}
\dot{\theta}^*_v = J^T(JJ^T + \lambda I)^{-1}\ddot{x}
\end{equation}

since it is valid for the case \(\lambda = 0\) as well, provided \(J\) has full row rank. The equivalence of (2) and (3) can be shown using the singular value decomposition (SVD) of \(J\). The SRI as given by (3) solves an optimization problem which is essentially the same as DLP-2 but differs slightly [3] in the mathematical formulation.

\textbf{DLP-3} \quad \begin{align*}
\min & \quad \|u\|^2 \\
\text{subject to} & \quad \dot{x} = J\ddot{\theta} + u \\
& \quad \|\ddot{\theta}\|^2 \leq \Delta^2
\end{align*}

The solution to DLP-3 is obtained as

\begin{equation}
\dot{\theta}^*_v = J^T(JJ^T + \lambda I)^{-1}\ddot{x}
\end{equation}

with \(\lambda \geq 0\) and \(\lambda(\|\dot{\theta}^*_v\|^2 - \Delta^2) = 0\).

Note that (4) is valid for the case \(\lambda = 0\) for both non-redundant and redundant manipulators since the matrix \((JJ^T)\) is invertible whenever \(J\) has full row rank. The following statements summarize the solution to problem DLP-3:

1. If \(\lambda = 0\) and \(\|\dot{\theta}^*_v\| \leq \Delta\), then \(\dot{\theta}^*_v = \dot{\theta}^+\) \(= J^+\ddot{x}\) is the solution.
2. If \(\lambda > 0\) and \(\|\dot{\theta}^*_v\| = \Delta\), then \(\dot{\theta}^*_v\) (given by (4)) is the solution.

Thus, whenever the pseudoinverse solution yields a joint velocity norm less than or equal to the allowable norm \(\Delta\), the SRI is equal to the pseudoinverse and the optimal damping factor is \(\lambda = 0\). Intuitively, this makes sense since the pseudoinverse gives the minimum-norm solution among all the exact solutions. Therefore, if \(\|J^+\ddot{x}\| < \Delta\), then there is a feasible accurate solution and hence no damping is required. But, if \(\|J^+\ddot{x}\| > \Delta\), then statement (2) says that the optimal value of the damping factor is one that will yield a joint velocity norm equal to \(\Delta\) and this will be the minimal amount of damping required to obtain a feasible solution.

Hence when the pseudoinverse solution results in an infeasible joint velocity vector, the SRI will provide a feasible solution with minimum deviation from the specified trajectory if the damping factor \(\lambda\) is such that \(\|\dot{\theta}^*_v\| = \Delta\), i.e. \(\|J^T(JJ^T + \lambda I)^{-1}\ddot{x}\| = \Delta\).
As a result, the solution of DLP-3 is closely related to solving the following non-linear equation in $\lambda$ in order to compute the optimal damping factor:

$$
\phi(\lambda) \equiv \| \hat{\lambda}^* \| - \Delta = \| J^T (J J^T + \lambda I)^{-1} \hat{\xi} \| - \Delta = 0
$$

except when $\phi(0) \leq 0$, in which case the solution is given by the pseudoinverse. Therefore, the problem of computing the optimal damping factor reduces to devising a method of solving (5) whenever $\| J^T \hat{\xi} \| > \Delta$.

**2.2. Computing the optimal damping factor.** Equation (5) can be solved using an efficient iterative method which has been developed in the context of solving an unconstrained minimization problem by trust region methods ([7], [8], [10]). In these methods a local quadratic model of the objective function is minimized within a trust region around the current iterate. This problem involves a quadratic function with a quadratic constraint and hence is similar to the DLP-2 and DLP-3 problems. Using the results obtained in solving the trust region problems, an iteration which is more efficient than Newton's method and which uses a local rational approximation $\phi(\lambda)$, can be employed [7] to solve (5). This iteration is

$$
\lambda_{k+1} = \lambda_k - \left[ \frac{\phi(\lambda_k) + \Delta}{\Delta} \right] \frac{\phi'(\lambda_k)}{\phi(\lambda_k)}
$$

Another way of viewing (6) is [10] that it is Newton's method is applied to

$$
\frac{1}{\Delta} - \frac{1}{\phi(\lambda)} = 0
$$

This iteration with suitable safeguarding [7] results in an efficient algorithm for computing the optimal damping factor $\lambda^*$.

If the singular value decomposition of $J$ [12] is available, then

$$
\phi(\lambda) \equiv \sqrt{\sum_{i=1}^{m} \frac{\sigma_i^2 \gamma_i^2}{(\sigma_i^2 + \lambda)^2}} - \Delta = 0
$$

(where $J = UDV^T$, $U^T \hat{\xi} = [\gamma_1 \gamma_2 \ldots \gamma_m]^T$ and $\sigma_i$ is the $i$th singular value of $J$) can be solved for the optimal damping factor $\lambda^*$ directly in the form given above. Since the $\sigma_i$'s and the $\gamma_i$'s become available through the SVD, the values of $\phi(\lambda_k)$ and $\phi'(\lambda_k)$ can be computed directly. In section 4, we present a parallel VLSI/DSP architecture which computes the SVD and the inverse kinematic algorithm described above.

3. **Simulations.** This section presents simulations performed on a three degree-of-freedom planar redundant arm using the inverse kinematic techniques described in the previous sections. The planar arm has one degree of redundancy since its task space is two-dimensional as it is required to follow a specified trajectory in the plane.

The specified trajectory is a square of dimensions 0.89 x 0.89 units. The starting configuration is $\theta_0 = [5^\circ \quad 175^\circ \quad 175^\circ]^T$ which is nearly singular. The end-effector is initially at the lower left corner of the square and is made to move in a counterclockwise direction along the sides of the square.

When the inverse kinematics for this trajectory is carried out using the pseudoinverse (Figure 1), the plot of the joint velocity vector norm against time has two very
high peaks. The first one corresponds to moving from the initial configuration, when the end-effector is made to track a singular direction. The second case is when the end-effector is near the top right corner of the square and the manipulator is in an outstretched configuration (an external singularity). High joint velocities are needed to make the manipulator move horizontally from this configuration.

Compare the above simulation with the simulation for the same trajectory in Figure 1 which uses the SRI to compute the joint velocities of the manipulator. Here, the manipulator has managed to avoid the two peaks that were seen in the pseudoinverse simulation and the joint velocity norm has stayed feasible throughout the motion. This has been achieved by compromising, to a minimal extent, the accuracy with which the end-effector has followed the specified trajectory. Computation of the optimal damping factor took two iterations on an average for this simulation.

Singular Value Decomposition using CORDIC. The singular value decomposition [4] of an $p \times p$ matrix $M$ is $M = U\Sigma V^T$, where $U$ and $V$ are orthogonal matrices and $\Sigma$ is a diagonal matrix of singular values. The SVD is a computationally complex algorithm which benefits from a VLSI parallel array such as the Brent, Luk, and Van Loan systolic array [1]. Each processor element contains a $2 \times 2$ submatrix. The CORDIC algorithms have been applied to the Jacobi method for the SVD to produce a processor array [2] which is currently being implemented at Rice University [5]. The CORDIC algorithms [11] provide a fast hardware method to calculate vector rotations and inverse tangents which are essential operations for the SVD.

The CORDIC iteration equations which are to be implemented in hardware are:

$$x_{i+1} = x_i + \delta_i y_i 2^{-i}, \quad y_{i+1} = y_i - \delta_i x_i 2^{-i}, \quad z_{i+1} = z_i + \delta_i \theta_i.$$ 

The variable, $x_i$, contains the total rotation angle applied, $\theta_i$ is the current rotation angle increment, and $\delta_i = \pm 1$. Through the appropriate selection of each $\delta_i$, either the initial $z_0$ value can be reduced to zero (vector rotation) or the initial $y_0$ value
can be reduced to zero (inverse tangent). These equations can be implemented with simple structures: registers, shifters, adders, and a small ROM.

4. Novel VLSI CORDIC / DSP Architecture. In this section, we map high level robotics algorithms onto an advanced special purpose VLSI architecture containing both CORDIC and Digital Signal Processing (DSP) arrays. DSP chips, such as the Texas Instruments TMS320C30 chip, are used since they possess a number of important features for real-time computation. DSP chips are highly integrated and contain on-board RAM and a high-speed floating-point multiply-and-accumulate unit. The TMS320C40 will also contain additional hardware communication ports to enhance the speed of a DSP array. Both parts can be programmed in a high-level language, such as 'C', and have real-time operating system software.

The entire inverse kinematics solution can be contained within the architecture, and is separate from the host machine, which frees up the host for more intensive tasks such as planning, fault tolerance, etc.. The Inverse Kinematics algorithm may be structured as follows: (the quantity \( \theta_{t+1} \) will be available via sensory information and \( \hat{\theta}_t \) is available from off-line end effector planning).

**Algorithm VLSI CORDIC/DSP Inverse Kinematics()**

begin;
compute Jacobian \( [J(\hat{\theta}_{t-1})] \);
compute SVD of \( [J(\hat{\theta}_{t-1})] \) and hence \( [J^+(\hat{\theta}_{t-1})] \);
if \( ||J^+\hat{\theta}|| > \Delta \) find optimal \( \lambda^* \);
compute SRI \( [J^*(\hat{\theta}_{t-1})] \) and then \( \hat{\theta}_t = [J^*(\hat{\theta}_{t-1})]\hat{\theta}_t \);
else \( \hat{\theta}_t = [J^+(\hat{\theta}_{t-1})]\hat{\theta}_t \);
output \( \hat{\theta}_t \);
end;

In our architecture, the Jacobian is to be calculated using the CORDIC array and DSP chips. Further details on the computation of the Jacobian may be found in [12]. The CORDIC array is next used to compute the SVD of the Jacobian. The optimal damping factor calculations will be performed in the DSP array that acts as a host for the CORDIC SVD array. Finally, \( J^* \) can be formed through the DSP array.

Having obtained the SRI, the joint velocities are obtained in parallel via the interconnected array of DSP processors. For our application (that of an eight degree of freedom manipulator) we use four Texas Instruments DSP chips, together with our CORDIC SVD array. In Figure 2, the SVD array performs an \( 8 \times 8 \) SVD, and is connected to 4 DSP processors (connections to SVD array marked in figure). Each processor is responsible for the parallel computations associated with two joints, and produces two of the joint velocities.

Each DSP chip receives two rows of each matrix. The vectors to be multiplied are broadcast to all DSP chips. The results are then combined to produce the new \( \hat{\theta}_t \)'s through a final series of pipelined operations in the DSP array.

5. Summary. The Damped Least-Squares technique obtains feasible joint velocities of a robot manipulator at all configurations by using a damping factor to damp the joint velocities. This paper shows how the Levenberg-Marquardt algorithm used in trust region methods can be used to efficiently compute an optimal damping factor, which results in minimum end effector deviation while ensuring feasibility at all configurations. We additionally describe an efficient special-purpose VLSI architecture to compute the joint velocities.
Fig. 2. Linear Array of DSP Processors connected to a 4 × 4 array of CORDIC SVD Processors.

REFERENCES


