

LECTURE 4: INVERSE KINEMATICS

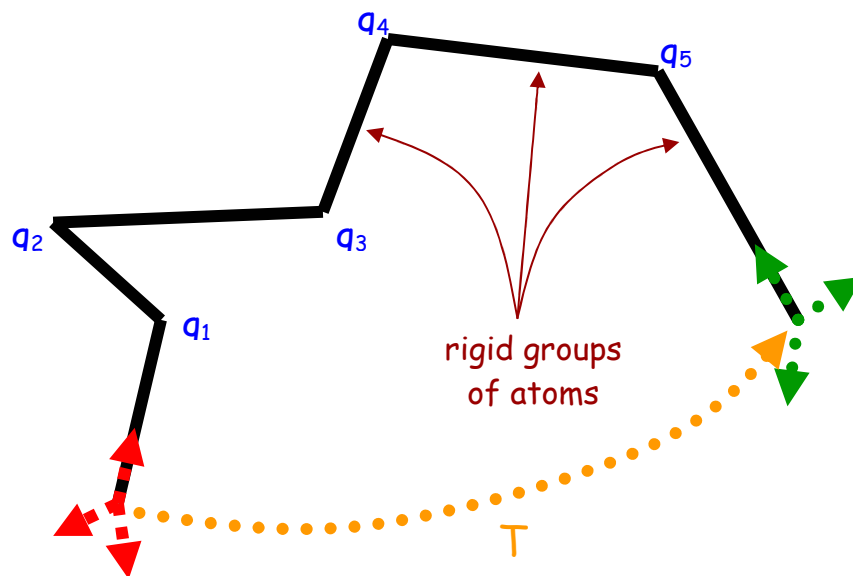
*Lecturer: Jean-Claude Latombe
17 April, 2006*

1 Definition

Given the angles of the joints of a linked structure, Kinematics is the process of finding the position of the end of the structure.

Inverse Kinematics (IK) is the reverse of this: given a kinematic chain of serial linkages and the position of one end relative to the other, we want to find the possible values of the joint parameters.

Within the context of biology, the chain usually refers to a linked rigid group of atoms, where the structure's end position and orientation is known relative to its start position and orientation. We want to be able to calculate the joint parameter values (for example, at atoms $q_1 \dots q_5$ in the diagram below):



2 IK and Proteins

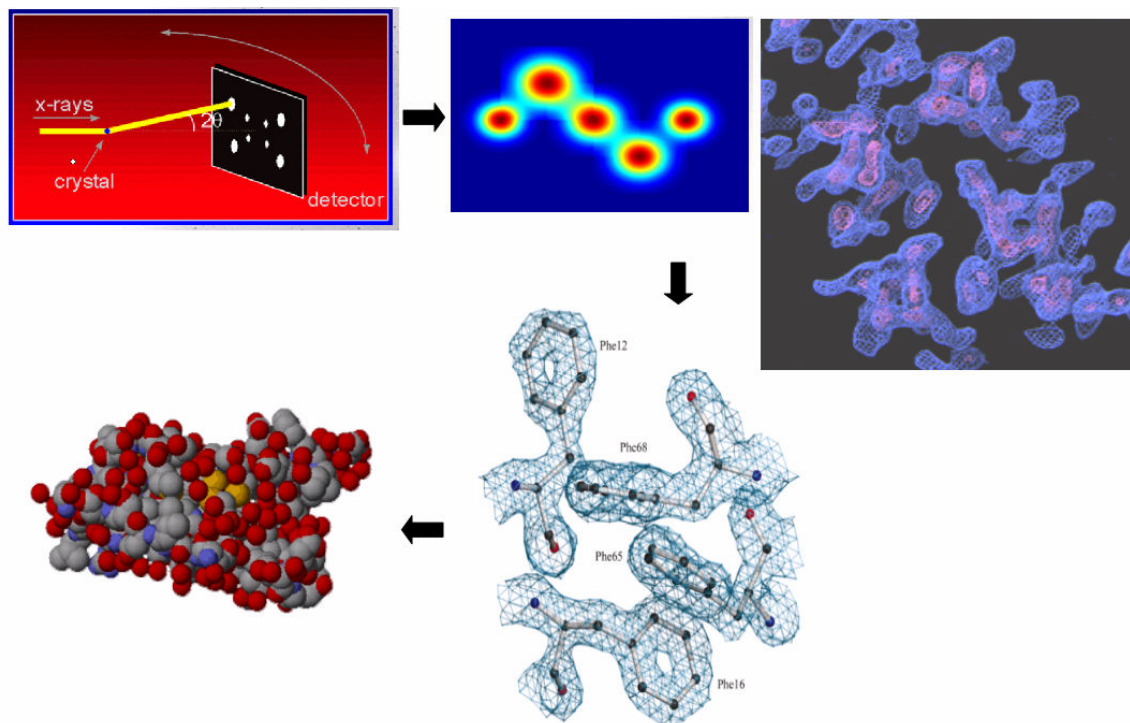
Protein structure determination techniques such as X-ray crystallography enables the 3D coordinates of atoms in a protein to be determined to within a certain resolution.

In X-ray crystallography, x-rays are sent through a crystal that contains a number of copies of a protein, all of which have the same folded structure. The X-rays are diffracted by their interaction with the electrons in the crystal, and the pattern of diffraction is recorded by a detector. Successive images are recorded as a crystal is rotated within the X-ray beam. The final image may be scanned onto a computer and is known as an electron-density map.

This method produces accurate results; however gaps may occur in the electron-density map, where the structure at certain points is not clear.

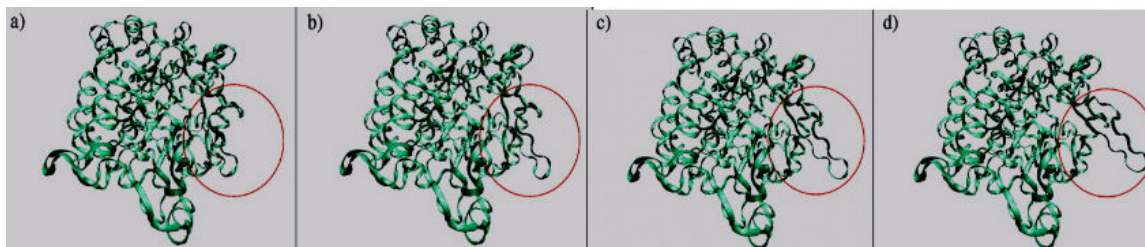
IK provides a useful solution to the problem of filling in the gaps. If the endpoints of the structure are known, as well as the sequence of atoms that belong to the gaps, then we can simply apply IK techniques to calculate the angles between the atoms, and thus complete the unknown areas of the structure.

Automated model building is another method used for structure determination and is used by software systems such as RESOLVE, TEXTAL, ARP/wARP and MAID. However, the completeness of the models produced by these systems varies from between 90% completeness (for distances between 1 and 2.3 Å) to merely 67% completeness (for distances between 2.3 and 3.0 Å). Thus IK methods may be applied here to fill in the gaps in the resulting models.



An alternative to employing automatic IK techniques would be manually completing the models. However, this is typically both labor and time intensive as existing tools are highly interactive. Overall, model completion is a high throughput bottleneck and thus IK is the more preferred technique.

In addition to filling gaps in structure determination, other applications of IK include sampling conformations (form and structure) using homology modeling, chain tweaking for better prediction of a folded state, and the study of the motion space of loops (secondary structure elements that connect α helices and β strands), which often play an important role in enzyme catalysis, ligand binding, and protein-protein interactions. For example, the 17-residue loop motion in the enzyme Amylosucrase is plays an important part in the protein's activity:



3 The Completion Problem

Within the context of structure determination, the completion problem requires that given as input a partial protein structure, two anchor residues, and the amino-acid sequence of the missing fragment (typically 4 to 15 residues long), the output should be a few candidate conformations of the fragment that respect the closure constraint (ensuring the beginning and end of the structure are at the given anchor residues), avoids atomic clashes and satisfies other constraints such as maximizing the match with the electron-density map or minimizing the energy function.

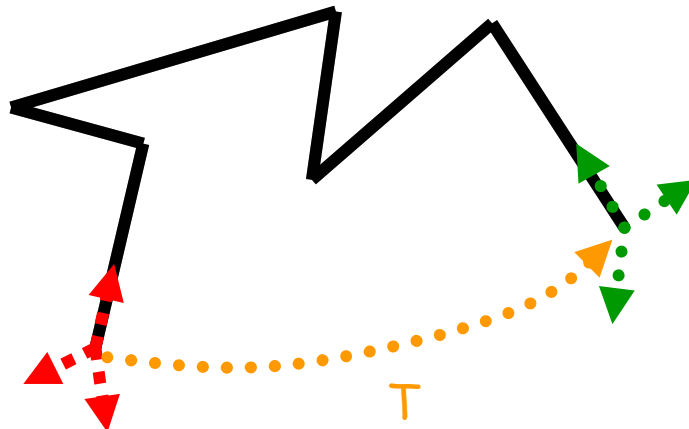
More generically, and to take advantage of the relation this problem has to the Robotics field, we may define the problem as having the following inputs and outputs:

Inputs:

- Closed kinematic chain with n degrees of freedom
- Relative positions/orientations X of end frames
- Target function $T(Q)$ that measures the quality of a conformation. This function may be used to measure the constraints mentioned above such as maximizing the match with the electron-density map.

Outputs:

- Conformation(s) that :
 - Achieve closure (ensure the start and end points stay the same)
 - Optimize the scoring function T (find local optimums of T)

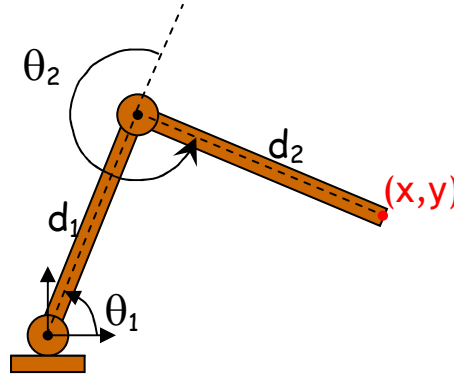


4 Examples

The following examples with an idealized robot illustrate the use of IK to solve the previously mentioned completion problem.

4.1 Example 1: Simple

The simplest case occurs when we have only one intermediate joint, as shown in the diagram below:



It is given that start of the joint is fixed at the origin and oriented at an angle θ_1 to the x-axis and that the end of the joint is at point (x, y) . In addition, the two links are of length d_1 and d_2 . This problem requires us to calculate θ_2 , the orientation of the second link in the framework.

We can calculate the values of x and y by applying forward kinematics:

$$x = d_1 \cos \theta_1 + d_2 \cos (\theta_1 + \theta_2)$$

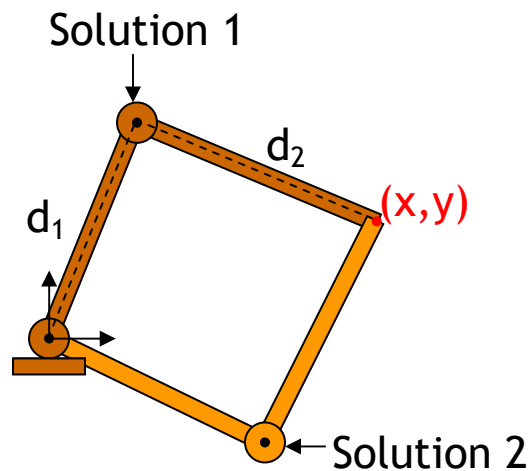
$$y = d_1 \sin \theta_1 + d_2 \sin (\theta_1 + \theta_2)$$

By using inverse kinematics however, we wish to do the reverse (given x and y , find θ_2). We thus need to manipulate the above equations so that the inverse cosine may be calculated:

$$\theta_2 = \cos^{-1} \left[\frac{x^2 + y^2 - d_1^2 - d_2^2}{2d_1d_2} \right]$$

$$\theta_1 = \frac{-x(d_2 \sin \theta_2) + y(d_1 + d_2 \cos \theta_2)}{y(d_2 \sin \theta_2) + x(d_1 + d_2 \cos \theta_2)}$$

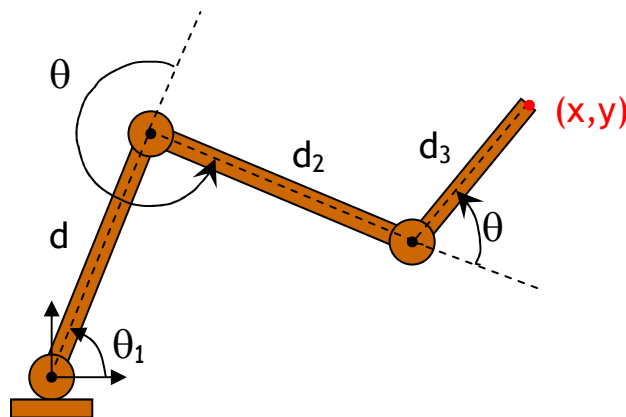
Generally, this will give us two solutions:



However, if the joint lies along the straight line joining the start and end anchors, then we can only have one solution.

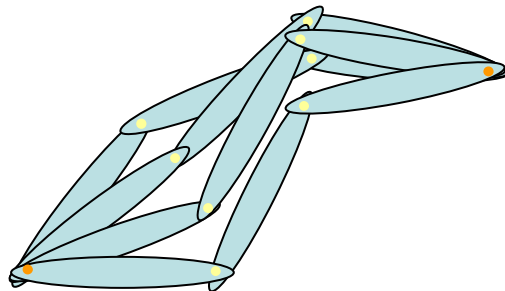
4.2 Example 2: More Complicated

A more complicated example would have more than one linkage such as the following:

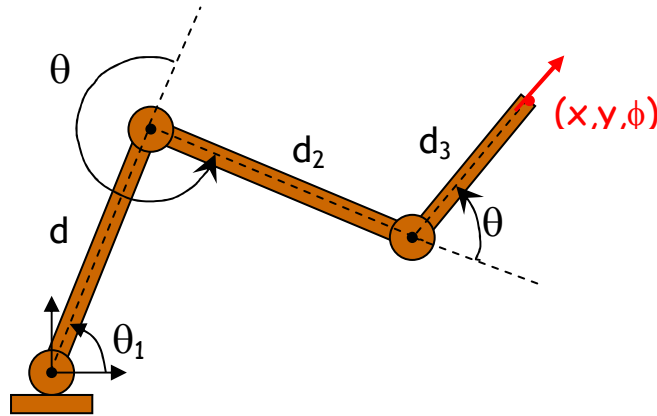


In this example, there are an infinite number of solutions. This can be seen by assuming one of the angles was fixed (say θ_1). It would then be possible to move the robot internally by modifying θ_2 and θ_3 .

As can be seen in the following diagram, this is called a redundant linkage, as it would be possible to get an infinite number of solutions by simply moving the structure internally without disrupting the positions of the end points. In effect, we have a 1-D self motion space that describes all the possible shapes of the robot.



Continuing with this same example, if we assume that the orientation of the last link (defined by ϕ) was to be fixed, then we would not have the previous redundancy and the self-motion space would be a point (0-D). This means we would arrive at a finite number of solutions:



5 General Results from Kinematics

In order to generalize the results, we need to calculate the number of degrees of freedom (DOFs) or a linkage or the dimensionality of velocity space. The equation to calculate this is as follows (Grübler 1883):

$$N_{\text{DOF}} = k (N_{\text{link}} - 1) - (k - 1) N_{\text{joint}}$$

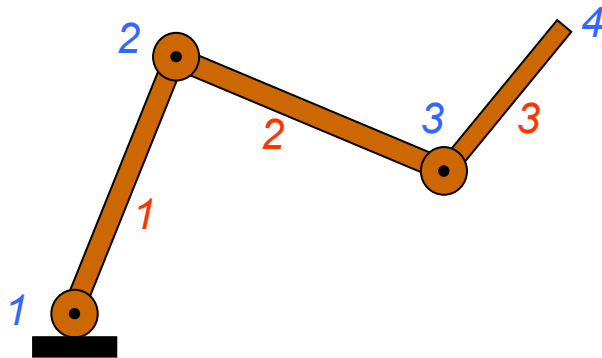
Where $k = 3$ if the linkage is planar i.e. joints like in 2-D space or $k=6$ if joints are in 3-D space.

Using this formula is not completely trivial as finding the correct values of N_{link} and N_{joint} is not completely straight-forward.

In general, we have two cases that need to be considered:

5.1 Open Chain

In this case, the starting anchor is considered a link as it can move arbitrarily. Hence $N_{\text{joint}} = N_{\text{link}} - 1$ and so $N_{\text{DOF}} = N_{\text{joint}}$.



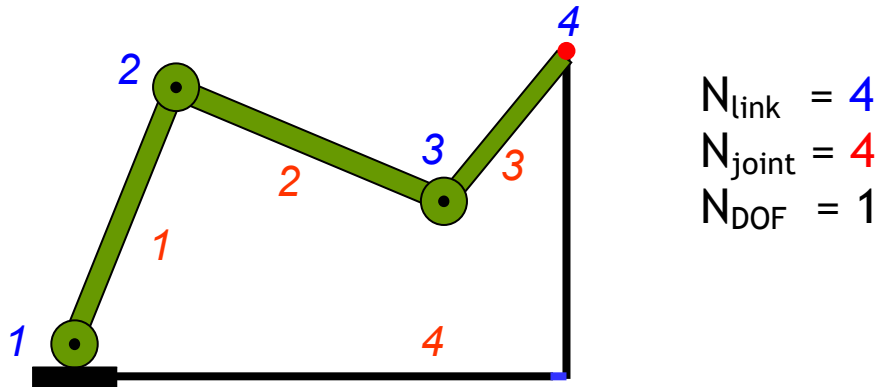
$$N_{\text{link}} = 4$$

$$N_{\text{joint}} = 3$$

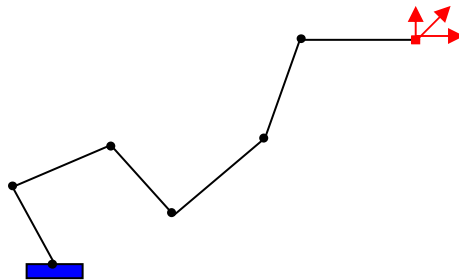
$$N_{\text{DOF}} = 3(4-1) - (3-1)3 = 3$$

5.2 Closed Chain

In this case, the end anchor is fixed in space relative to the start anchor, and so it is considered a joint (a passive joint). Hence $N_{\text{joint}} = N_{\text{link}}$ and so $N_{\text{DOF}} = N_{\text{joint}} - k$:



In the case of a 6 joint chain in 3-D space, a closed chain (fixing the start and end anchors) gives us 0 degrees of freedom and at most 16 solutions for joint angles. Usually for proteins we get around 6 to 10 distinct solutions, but it is possible to get 16.



6 Analytical (Exact) Techniques

Analytical techniques may be applied to the case where we have 6 joints as follows (the principle of the technique is always the same):

- Write forward kinematics in the form of polynomial equations, where $t = \tan(\theta/2)$
- Simplify, e.g. using the fact that two consecutive torsional angles φ and ψ have intersecting axes (this is not immediately obvious, it is only in this case that the fact applies)
- Solve using the specification of the structure

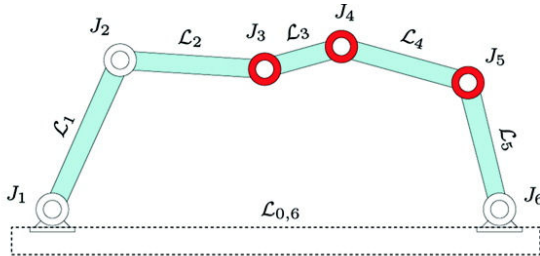
(Note: See Coutsias, Seok, Jacobson, Dill, 2004 for more information)

There are no techniques currently available to let us analytically solve for greater than 6 degrees of freedom (since the number of solutions is infinite) however, we can sample solutions by using a decomposition method to randomly sample conformations of closed chains.

The following steps are carried out iteratively to ultimately reach an approximate solution:

1. Decompose closed chain into:

- 6 “passive” joints
- n-6 “active” joints



2. Sample the active joint parameters

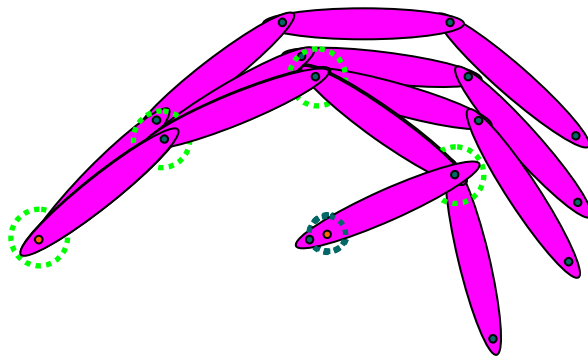
3. Compute the passive joint parameters using the exact analytical solver technique

7 CCD (Cyclic Coordinate Descent) Method

This method is relatively easy to use and generates a random conformation with one end of chain at required position and orientation. This is repeated multiple times until the other end is at required position and orientation or algorithm converges to a local minimum. The steps that are repeated are as follows:

1. Pick one DOF
2. Change to minimize closure distance

As can be seen in the diagram below, one end is fixed and the structure is rotated around the 1st, 2nd, 3rd and 4th joints.



When we apply the CCD method to proteins, we calculate the closure distance (S) as follows:

$$S = \|\bar{N} - N\|^2 + \|\bar{C}_\alpha - C_\alpha\|^2 + \|\bar{C} - C\|^2$$

In order to minimize the closure distance we then need to compute:

$$q_i \text{ s.t. } \frac{\partial S}{\partial q_i} = 0$$

where q_i is the position of one of an atom.

The advantages of the CCD method are that it is fairly simple to compute, with the main difficulty being calculating the partial differentiation and that it does not suffer from the singularity problem (which we shall see the Jacobian method encounters). In addition when using CCD, it is possible to constrain each joint independently of all other joints, so some joints may be allowed to move more than others.

One disadvantage is that the CCD method may converge to a local minimal that may not be a valid solution. A way to deal with this problem is to use Ramachandran Maps, which assign probabilities to ϕ - ψ pairs. At each iteration, we change a pair (ϕ_i, ψ_i) by:

- Computing change to ϕ_i
- Computing change to ψ_i based on change to ϕ_i
- Accept the new pair with probability $\min(1, P_{\text{new}}/P_{\text{old}})$

This provides us with a way to move away from the local minima.

8 Jacobian Matrix

The general problem can be expressed and solved in terms of matrix calculations as follows:

- Specify Q , an n -vector of internal coordinates where $n \geq 6$
- Specify X , a 6-vector defining the position and orientation of endpoints
- Having fixed Q , use forward kinematics to specify the endpoint position as $X = F(Q)$ i.e. $x_i = f_i(q_1, q_2, \dots, q_n)$
- Differentiating this equation yields:

$$dx_i = [\partial f_i(Q)/\partial q_1] dq_1 + \dots + [\partial f_i(Q)/\partial q_n] dq_n$$

This may be re-written as:

$$dX = J dQ$$

where J is the Jacobian matrix:

$$\begin{pmatrix} \partial f_1(Q)/\partial q_1 & \partial f_1(Q)/\partial q_2 & \dots & \partial f_1(Q)/\partial q_n \\ \partial f_2(Q)/\partial q_1 & \partial f_2(Q)/\partial q_2 & \dots & \partial f_2(Q)/\partial q_n \\ \dots & \dots & \dots & \dots \\ \partial f_6(Q)/\partial q_1 & \partial f_6(Q)/\partial q_2 & \dots & \partial f_6(Q)/\partial q_n \end{pmatrix}$$

The calculation of the two cases we need to consider is described in the next subsections.

8.1 Case where $n = 6$

In this case, J is a square 6×6 matrix. The problem definition is that given X , we need to find Q such that $X = F(Q)$. The method for solving the problem is as follows:

- Start at any $X_0 = F(Q_0)$
- Interpolate linearly between X_0 and X , producing the sequence $X_1, X_2, \dots, X_p = X$
- For $i = 1, \dots, p$ do:
 - a) $Q_i = Q_{i-1} + J^{-1}(Q_{i-1})(X_i - X_{i-1})$
 - b) Reset X_i to $F(Q_i)$ - this is required since we are doing a linear approximation, so we don't exactly reach X_i initially.

8.2 Case where $n > 6$

In this case, J is a $6 \times n$ matrix and thus is not longer square. Hence we cannot invert J and use it in the same equation as for the first case. What we can do instead is calculate J^+ (the pseudo inverse) such that $JJ^+ = I$. Thus we use the following equation:

$$dQ = J^+ dX$$

Furthermore, if we assume that $\text{rank}(J) = 6$, then the null space $\{dQ_0 \mid J dQ_0 = 0\}$ has dimension $= n - 6$, and so:

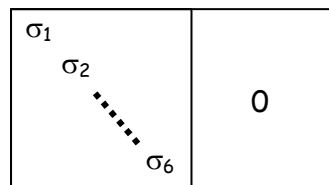
$$dQ = J^+ dX + dQ_0$$

where dQ_0 is arbitrarily chosen in the null space.

8.2.1 Computation of J^+

The pseudo inverse may be calculated using the SVD (Singular Value Decomposition) method. The steps in this method are as follows:

1. J is expressed as a multiplication of three matrices $U \Sigma V^T$ where:
 - U in a 6×6 square orthonormal matrix
 - V is an $n \times 6$ square orthonormal matrix
 - Σ is of the form $\text{diag}[\sigma_i]$:



2. $J^+ = V \Sigma^+ U^T$ where $\Sigma^+ = \text{diag}[1/\sigma_i]$

In order to get the null space mentioned earlier, we add $(n-6)$ 0-vector columns to matrix Σ , hence it becomes a $6 \times n$ sized matrix. Gram-Schmidt orthogonalization is then used to create an $(n-6)$ basis of null space. The result is diagrammatically as follows:

$$dX = \begin{matrix} & \underbrace{\hspace{10em}}_{J} & \\ & \begin{matrix} U_{6 \times 6} & \Sigma_{6 \times 6} & V_{6 \times n}^T \end{matrix} & \\ \begin{matrix} dX \\ \vdots \end{matrix} & = & \begin{bmatrix} \text{yellow } 6 \times 6 & \text{yellow } 6 \times 6 \text{ with diagonal dots} & \text{white } 6 \times (n-6) \\ & & \text{white } (n-6) \times (n-6) \end{bmatrix} & \begin{matrix} dQ \\ \vdots \end{matrix} \end{matrix}$$

$$dX = \begin{matrix} & \underbrace{\hspace{10em}}_{J} & \\ & \begin{matrix} U_{6 \times 6} & \Sigma_{6 \times n} & V_{n \times n}^T \end{matrix} & \\ \begin{matrix} dX \\ \vdots \end{matrix} & = & \begin{bmatrix} \text{yellow } 6 \times 6 & \text{yellow } 6 \times 6 \text{ with diagonal dots} & \text{brown } 6 \times (n-6) \text{ with } 0 \\ & & \text{yellow } (n-6) \times (n-6) \\ & & \text{brown } (n-6) \times (n-6) \end{bmatrix} & \begin{matrix} dQ \\ \vdots \end{matrix} \end{matrix}$$

Gram-Schmidt orthogonalization $\rightarrow$ (arrow pointing to the bottom brown block in V^T)

$$dX = \begin{matrix} & \underbrace{\hspace{10em}}_{J} & \\ & \begin{matrix} U_{6 \times 6} & \Sigma_{6 \times n} & V_{n \times n}^T \end{matrix} & \\ \begin{matrix} dX \\ \vdots \end{matrix} & = & \begin{bmatrix} \text{yellow } 6 \times 6 & \text{yellow } 6 \times 6 \text{ with diagonal dots} & \text{brown } 6 \times (n-6) \text{ with } 0 \\ & & \text{yellow } (n-6) \times (n-6) \\ & & \text{brown } (n-6) \times (n-6) \end{bmatrix} & \begin{matrix} dQ \\ \vdots \end{matrix} \end{matrix}$$

(n-6) basis N of null $\rightarrow$ (arrow pointing to the bottom brown block in V^T)

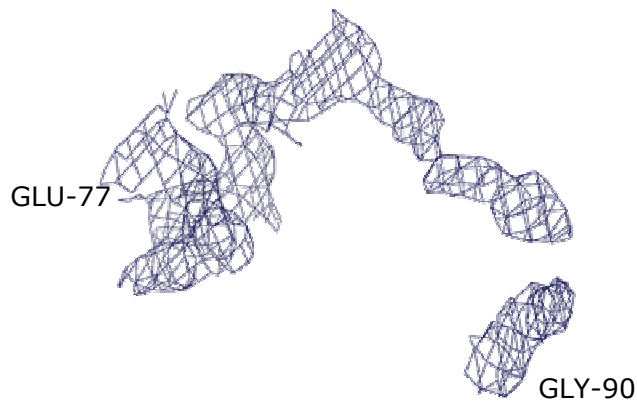
8.2.2 Minimization of Target Function T with Closure ($n > 6$)

Finally, in order to minimize the target function, when we have as input a chain with both ends at goal positions and orientations, we repeat the following:

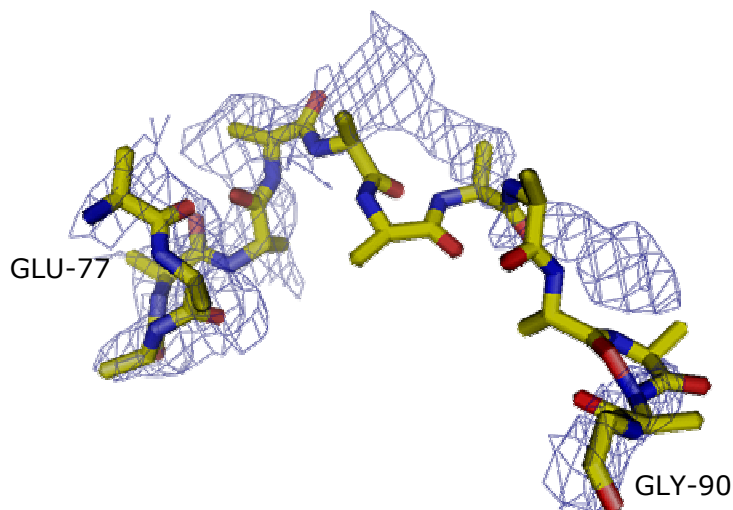
1. Compute Jacobian matrix J at current q
2. Compute null-space basis N using SVD of J
3. Compute gradient $\nabla T(q)$
4. Move along projection $NN^T y$ of $y = -\nabla T(q)$ onto N until minimum is reached or closure is broken. This yields a new q .

9 Visual Application Example

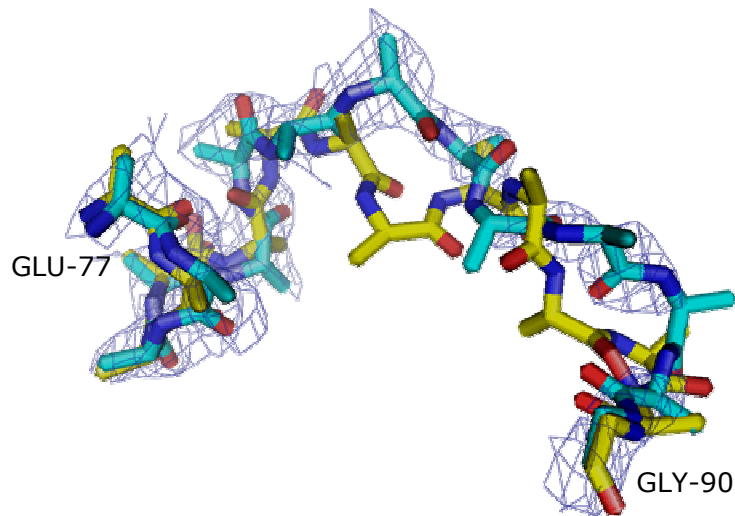
The following diagram shows the poor quality regions of an electron density map:



After generating an open chain at random, CCD is used just to close the kinematic chain. The resultant conformation obtained by CCD is shown in yellow. What we now have is a closed structure that does not as yet match very well to the original electron density map.



Finally, we fix the two ends of the chain and tweak the degrees of freedom using the concept of the pseudo-inverse. By moving the chain in the null space, we are able to align it closer to the map. The cyan chain depicts the conformation after this optimization in the null space has been carried out.



References:

Diagrams and some descriptions were obtained from CS273 Lecture 4 slides: inverse-kinematics.ppt