

6.1 Conformational Space

A conformation of a molecule or a system in general is a complete specification of the spatial placement of the entire system. In the case of molecules it could be the relative positions of all atoms with respect to one another. The conformational space for a system is the set of all possible conformations. It is very similar to the configuration space as in robotics and graphics.

6.1.1 Typical Representations

Typical Representations of the conformation space include:

1. Coordinate Representation:
This representation describes the coordinates of each atom in the molecule and corresponds to the atomistic model of molecules.
2. Torsional Representation:
This corresponds to representing each conformation in terms of the torsional angles of the rotatable bonds and assumes the bond lengths and bond angles staying constant.
3. Intra-Molecular Distance Matrix:
This representation stores the distances between all atoms of the molecule. Despite having a quadratic number of parameters, it does capture some very interesting semantics especially in the case of complex folding systems such as proteins.

As an example in Figure 6.1, we can clearly see the effect of folding in proteins. Apart from the diagonal distances to be small, we also have other distant C_α atoms clearly indicating the presence of the molecule folding onto itself.

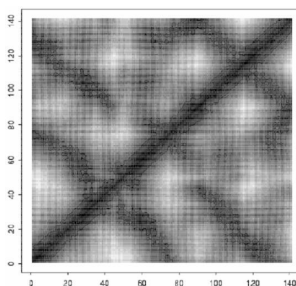


Figure 6.1: Distances between C_α pairs of a protein with 142 residues. Darker squares represent shorter distances.

6.1.2 Conformational Space Metrics and the RMSD Measure

A good metric (with all the mathematical properties of a distance metric) should be able to measure how well the atoms in two conformations can be aligned. In this context the RMSD (mean square distance/deviation) measure seems to have a lot of relevance. A simple metric over conformational space c is a function:

$$d : c, c' \in C \rightarrow d(c, c') \in \mathbb{R}^+ \cup \{0\}$$

such that:

$$\begin{aligned} d(c, c') &= 0 \text{ iff } c = c' \text{ (non-degeneracy) ,} \\ d(c, c') &= d(c', c) \text{ (symmetry) ,} \\ d(c, c') + d(c', c'') &\geq d(c, c'') \text{ (triangular inequality)} \end{aligned}$$

Not all metrics are good. Using only the Euclidean metric may be one of the simplest way, but it does not perform well under most conditions. We shall describe the basic RMSD measure in following paragraphs and evaluate metrics based on RMSD, namely cRMSD and dRMSD.

Given two sets of points in \mathbb{R}^3 : $A = \{a_1, \dots, a_n\}$ and $B = \{b_1, \dots, b_n\}$, the RMSD between A and B is:

$$\text{RMSD}(A, B) = (1/n)[\sum_{i=1}^n \|a_i - b_i\|^2]^{1/2}$$

where $\|a_i - b_i\|$ denotes the Euclidean distance between a_i and b_i in \mathbb{R}^3 . Clearly, $\text{RMSD}(A, B) = 0$ iff $a_i = b_i$ for $\forall i$.

6.1.3 cRMS Distance:

Given a molecule M with n atoms $\{a_1, \dots, a_n\}$ and two conformations c and c' of M where $a_i(c)$ is position of a_i when M is at c , the cRMSD between c and c' is the minimized RMSD between the two sets of atom centers, as follows:

$$\text{cRMSD}(c, c') = \min_T [(1/n) \sum_{i=1}^n \|a_i(c) - T(a_i(c'))\|^2]^{1/2}$$

The minimization is over all possible rigid-body transforms T (can be seen as every atom moving to a new position using the transforms). As shown in Figure 6.2, the transformations by T remove any large distinct distance differences due to their original start position. Usually, cRMSD is restricted to a subset of atoms, e.g. the C_α atoms on a protein backbone. However, this still remains a non-trivial task since it involves finding the aligning transform.

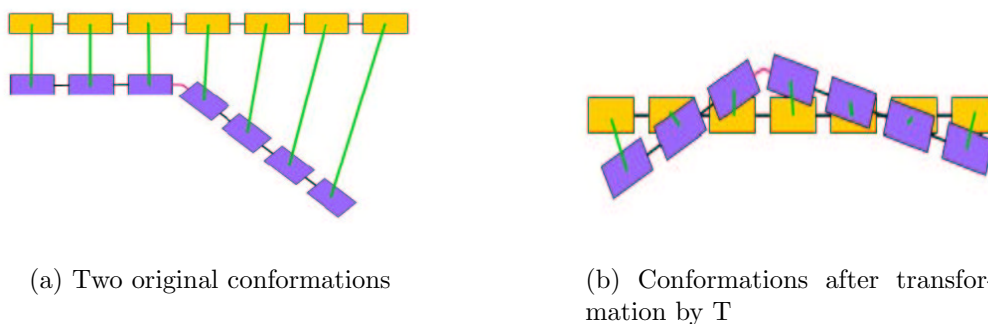


Figure 6.2: Transformation Results

6.1.4 dRMS Distance

Given a molecule M with n atoms $\{a_1, \dots, a_n\}$ and two conformations c and c' of M where $d_{ij}(c)$ is the $n \times n$ symmetrical intra-molecular distance matrix in M at c , the dRMSD between c and c' is:

$$\text{dRMSD}(c, c') = [(1/n(n-1)) \sum_{i=1}^n \sum_{j=i+1}^n [d_{ij}(c) - d_{ij}(c')]^2]^{1/2}$$

Usually d_{ij} is restricted to a subset of atoms in order to reduce the number of terms in the sum. Even though dRMSD does not depend on finding an aligning transform it suffers from having to deal with a quadratic number of parameters. Another problem of dRMSD is that it does not distinguish between two conformations that are the mirror images of one another. An example is given in Figure 6.3.

6.2 Kinematics

The kinematics of a mechanical system specifies the possible motions of this system, without explicit considerations for the forces/torques creating these motions.

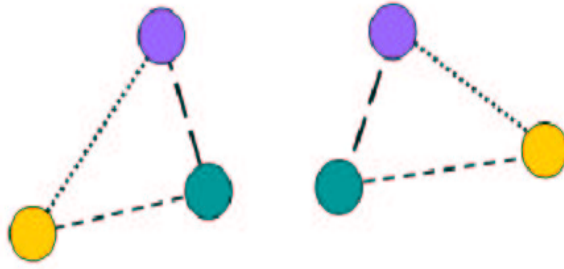


Figure 6.3: *Conformations that are mirror images of each other*

6.2.1 General Concepts and Results

In the context of molecule linkage models we shall only consider kinematic chains with *links* and *joints*. *Links* are rigid bodies which are connected to each other by *joints*. We only consider revolute joints with rotation around a single axis. Figure 6.4 shows an example of such a linkage chain.

6.2.1.1 Links, Joints and Degrees of Freedom

A critical concept for studying the kinematics of a system is the notion of *degree of freedom*. The degrees of freedom of a rigid body is defined as the number of independent movements it has. Figure 6.5 shows a rigid body in 3-D which can be translated along the three axes or rotated independently about each of them. Thus, it has a 6 DOFs in 3-D space.

An important concept that goes hand in hand with the degrees of freedom of a system is that of *velocity space*. The number of DOFs of a system is also known as the dimensional of its velocity space.

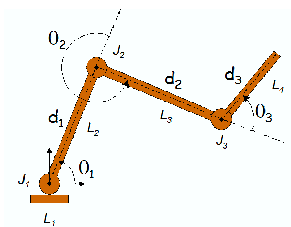


Figure 6.4: *Kinematic Chain: L denotes links and J denotes joints. The base is also a link with its coordinates in the world frame.*

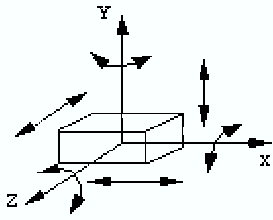


Figure 6.5: *Rigid Body in 3-Dimensional Space. The arrows indicate the 6 degrees of freedom of the body.*

6.2.1.2 General Results for Linkages

Grübler's Formula (1883)

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

(where $k = 3$ for planar linkages and 6 for spatial linkages)

Grübler's Formula is used to calculate the number of DOFs. There are 2 important cases:

- Open chain:

$$\begin{cases} N_{joint} = N_{link} - 1 \\ N_{DOF} = N_{joint} \end{cases}$$

- Closed chain:

$$\begin{cases} N_{joint} = N_{link} - 1 \\ N_{DOF} = N_{joint} - k \end{cases}$$

The chain is closed if the positions of both start and end point are fixed.

There are a few examples of using the formula.

- *Simple Linear Linkage*

In Figure 6.4, we have $N_{link} = 4$ and $N_{joint} = 3$

$$\Rightarrow \text{Thus, } N_{DOF} = 3*(4-1) - (3-1)*3 = 3$$

We can specify the corresponding positions and orientation of the linkage. Thus, the endpoint can be positioned anywhere in space with 3 degrees of freedom.

- *Simple Linear Linkage with Fixed End Point*

In Figure 6(a) the coordinates of the terminal point are fixed. One way to model this constraint is to rigidly attach this point to the base of the linkage as depicted in Figure 6(b).

$$\Rightarrow \text{Therefore, } N_{link} = 4 \text{ and } N_{joint} = 4$$

$$\Rightarrow \text{Thus, } N_{DOF} = 3*(4-1) - (3-1)*4 = 1$$

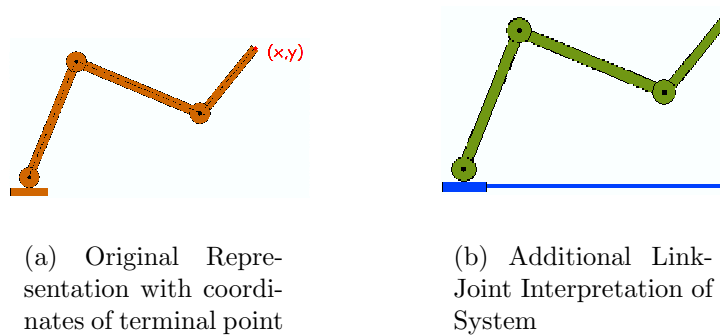


Figure 6.6: Simple Linear Linkage with fixed position of the terminal point

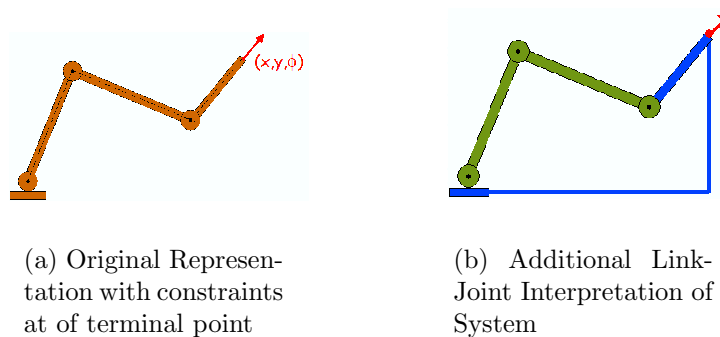


Figure 6.7: Simple Linear Linkage with fixed position/orientation of the terminal point

- *Simple Linear Linkage with Fixed Position and Orientation of End Point*

In Figure 7(a) the position and orientation of the terminal link are fixed. To model this additional constraint we can attach the last link to the linkage base as depicted in Figure 7(b).

⇒ Therefore, $N_{\text{link}} = 3$ and $N_{\text{joint}} = 3$

⇒ Thus, $N_{\text{DOF}} = 3*(3-1) - (3-1)*3 = 0$

6.3 Kinematics of Proteins

6.3.1 Kinematic Models of Molecules Revisited

- Atomistic model

The position of each atom is defined by its coordinates in 3-D space. Constraints on bond lengths/angles are encoded separately.

- Linkage model

The kinematics is defined by internal parameters (bond lengths and angles, and torsional angles). Small local changes may have big global effects since even a small change in one torsional angle rotates the entire remaining molecule making it more prone to errors. Another difficulty with the method is that forces are difficult to express with most force models due to distance based necessitating coordinate and distance computations.

- Simplified Linkage Model

In this model bond lengths and angles are assumed constant and only torsional angles may vary. This leaves less parameters to optimize. However, as a result of assuming all other parameters constant we lose the flexibility that exists in actual molecules with respect to bond distances and angles.

6.3.2 Proteins: Basic Structural Features

Proteins are polymer chains made up of monomeric units called amino acids. The key bonding component of the protein structure is the peptide bond between two amino acids which sets up the protein chain.

6.3.2.1 Amino Acids

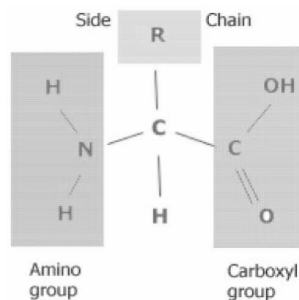


Figure 6.8: *Generic Structure of Amino Acids*

Amino acids are the basic structural units of proteins. An alpha-amino acid consists of an amino group, a carbonyl group, a hydrogen atom, and a distinctive *R* group bonded to a carbon atom, which is called the alpha-carbon. An *R* group is referred to as a side chain. Figure 6.8 shows the above defined structure.

The 20 amino acids that are found within proteins determine the biological activity of the protein as they contain the necessary information to determine

how that protein will fold into a three dimensional structure, and the stability of the resulting structure.

6.3.2.2 Peptide Bonds

Even though amino acids form the actual chain, peptide bonds are an important aspect of protein structure as well since they are the force behind the linking in the protein chain. The peptide bond is slightly shorter than a standard single bond due to partial de-localization of π electrons from the carbonyl group into orbitals shared with the lone pair electrons of the amide nitrogen which inhibits rotation around the peptide bond. Thus, the four atoms bound to the carbonyl carbon and amide nitrogen to form a plane (Figure 6.9).

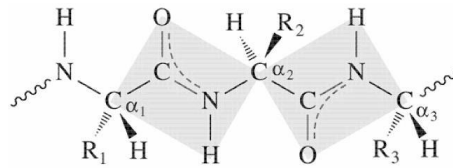


Figure 6.9: *Induced planarity in the peptide bond*

6.3.3 Protein Linkage Model

The salient features of the linkage models of proteins are (given in Figure 6.10):

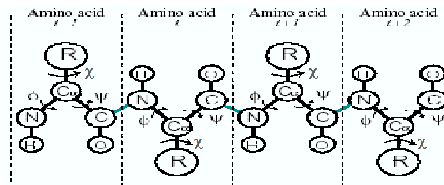


Figure 6.10: *Linkage Model for Proteins*

- The sequence of N-C $_{\alpha}$ -C atoms forming the backbone. This is analogous to the links in the simple linear linkage.
- Rotatable bonds (torsion angles) along the backbone defining the $\phi - \psi$ torsional degrees of freedom. These bonds perform the role of joints with one degree of freedom.
- Small side-chains corresponding to the amino acids which have a χ degree of freedom about the C $_{\alpha}$ -C $_{\beta}$ bond.

6.3.3.1 Example of Kinematic Chain in Proteins

In Figure 6.10 we have 5 amino acids (assuming no role of the χ torsion), 10 links and 10 joints. Applying Grübler's Formula in 3-D we have:

$$N_{\text{DOF}} = 6*(n-1) - (6-1)*n = n - 6 = 4 \text{ (for } n = 10\text{)}$$

6.4 Forward Kinematics

Forward Kinematics is the problem of determining the positions of the individual links in the world coordinate frame given the rotation values of each joint in the linkage. We assume there is a fixed base in the given frame. As in Figure 6.11, given the data about the joint angles and link lengths, we can simply compute the world frame coordinates of (x, y) using these formulas:

$$\begin{cases} X = d_1 * \cos \theta_1 + d_2 * \cos (\theta_1 + \theta_2) \\ Y = d_1 * \sin \theta_1 + d_2 * \sin (\theta_1 + \theta_2) \end{cases}$$

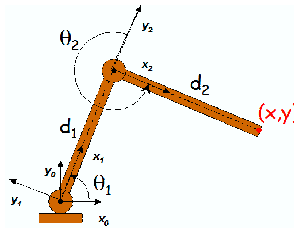


Figure 6.11: *Simple Linear Chain with 3 links and 2 joints*

6.5 Inverse Kinematics (IK)

Different from Forward Kinematics, Inverse Kinematics of a serial linkage is the problem of determining the joint angles given the position and/or orientation of the last link. We can also view it as the problem of finding the values of the degree-of-freedom parameters when given a kinematic chain and its position and/or orientation of one end relative to the other (closed chain). Inverse kinematics is especially useful for proteins because of the following reasons:

- Filling gaps under structure determination by X-ray crystallography
- Sampling conformations using homology modeling
- Studying the motion space of “loops” (secondary structure elements connecting α helices and β strands), which often play a key role for enzyme catalysis, ligand binding (induced fit) and protein to protein interactions.

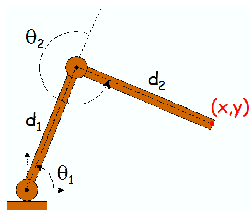


Figure 6.12: *Inverse Kinematics problem: given (x,y) we need to find θ_1 and θ_2*

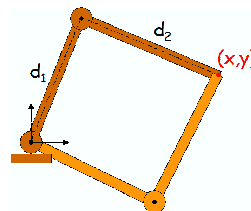


Figure 6.13: *Two solutions for θ_1 and θ_2 in the configuration given in Figure 6.12.*

Figure 6.12 shows a simple example presented for ease of understanding. In this case a closed form solution exists. In Figure 6.12 we can easily compute the values of θ_1 and θ_2 using the principles of geometry. It can be proven that:

$$\begin{cases} \theta_2 = \cos^{-1} \left(\frac{(x^2+y^2-d_1^2-d_2^2)}{2*d_1*d_2} \right) \\ \theta_1 = \frac{-x*(d_2*\sin \theta_2)+y*(d_1+d_2*\cos \theta_2)}{y*(d_2*\sin \theta_2)+x*(d_1+d_2*\cos \theta_2)} \end{cases}$$

However, a major point to be noted is that two solutions to the above equations exist due to the fact that $\cos^{-1}(x)$ has two values. These solutions are as shown in Figure 6.13. Another example in Figure 6.14 shows that there is a linkage with 3 joints and the coordinates for the end point is fixed at (x,y) . In such a situation, we have an infinite number of solutions, redundant linkage and what we know as self-motion space.

6.5.1 Generic Problem Definition

Given any input protein structure with missing fragments (each typically 4 to 15 residues long), and an amino-acid sequence of each missing fragment, we are to output conformations of fragments or distribution of conformations that satisfy the closure constraint for IK (anchoring at two ends), and any other constraints, for example maximizing the match with electron density map, minimizing the energy function. This IK problem is yet again very similar to robotics as in calculating joint angle to lift a glass of water.

6.5.2 Number of Expected Solutions

In general, if the degrees of freedom is less than the number of constraints then no solutions can exist. If the two are equal, then there exists a finite number of solutions. In the final case of having more degrees of freedom than constraints, we can have a possibly infinite number of solutions. For example, given a serial linkage with 6 joints and a fixed end point, we can have up to 16 solutions.

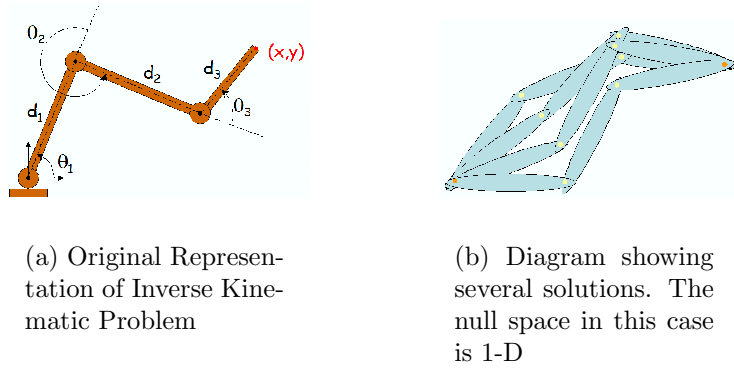


Figure 6.14: Visualization of the under-constrained system leading to infinite solutions

6.5.3 Analytical IK Techniques

Analytical techniques exist only for systems with up to 6 degrees of freedom in 3-D space. Otherwise it may become too complicated. The methodology used is as follows:

- Reduce the forward kinematic equations in polynomial form by expressing all trigonometric quantities in terms of $t = \tan \theta/2$, replacing sin and cos.
- Use domain specific properties in order to simplify the types of configuration possible for the given problem. Coutsias et al. (2004) [C04] apply such form of search space pruning in the context of protein structures. Their idea is to use the fact that consecutive torsional angles ϕ and ψ have intersecting axes.
- Solve the equations analytically, using calculus.

6.6 Incremental IK Techniques

Presence of trigonometric terms make IK problems non-linear. Therefore, it is difficult to find an analytical solution. The more efficient solution is finding a good approximation and estimating its rate of convergence.

One of the important practical approaches to for modeling a linkage is based on Numerical Methods. Most of them use linear approximations of the same type. The basis for incremental approximation is the Jacobian matrix. Here is a derivation of the Jacobian matrix:

- Let $\theta = n$ -vector of internal coordinates (usually the values θ_i for all the joints).

- Let $X = 6$ -vector for the end-point orientation for defining $x, y, z, \alpha, \beta, \gamma$.
- The relationships can then be written as:

$$\begin{cases} X &= F(\theta). \\ dX &= J * d\theta. \\ dX_i &= \sum_{j=1}^n \frac{\partial f_i(\theta)}{\partial \theta_j} * d\theta_j \end{cases}$$

where F is the forward kinematics transform and J is the Jacobian

- We may present this sum in the form of the Jacobian matrix:

$$J = \begin{bmatrix} \frac{\partial f_1(\theta)}{\partial \theta_1} & \frac{\partial f_1(\theta)}{\partial \theta_2} & \cdots & \frac{\partial f_1(\theta)}{\partial \theta_n} \\ \frac{\partial f_2(\theta)}{\partial \theta_1} & \frac{\partial f_2(\theta)}{\partial \theta_2} & \cdots & \frac{\partial f_2(\theta)}{\partial \theta_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m(\theta)}{\partial \theta_1} & \frac{\partial f_m(\theta)}{\partial \theta_2} & \cdots & \frac{\partial f_m(\theta)}{\partial \theta_n} \end{bmatrix}$$

We are assuming that $6 \leq n$ as otherwise we would not have any solutions. The methods adopted totally depend on the relation between m and n . The following subsections show the common methods used.

6.6.1 Optimization Based Methods: Cyclic Coordinate Descent (CCD)

The main idea of optimization based methods is to consider the primary equation $\theta = f^{-1}(X)$ as a minimization problem. Thus, the equation could be transformed into:

$$E(\theta) = (P - X(\theta))^2$$

The CCD [W91] is based on minimization applied to each joint separately. The steps in one pass are ordered from the most distant segment to the base segment. A number of passes are made over the manipulator to find the global minimum of the above equation. Since only one joint variable is changing at any time, an analytic solution can be used to significantly speed up the minimization problem.

CCD consists of the following steps:

- 1) Generation of random conformation. One end of closed chain is fixed and the other end is moving.

- 2) Repeat the generation step 1) if the moving end has not reached the required position or stopped at local minimum.
 - Pick one DOF
 - Change its position in order to minimize the closure distance

6.6.1.1 Analysis of CCD

CCD has several advantages:

- Simplicity
- No singularities
- Each DOF can be constrained independently from all others

The disadvantage of CCD is:

- There is no method to differentiate local minimum from global. Therefore, it may stop on the local min.

6.6.1.2 CCD with Ramachandran Maps

The approach of Ramachandran Maps is to assign the probabilities to $\phi - \psi$ pairs (figure).

- 1) Estimate the probabilities to $\phi - \psi$ pairs
- 2) Change a pair (ϕ_i, ψ_i) at each iteration:
 - Compute change to ϕ_i
 - Compute change to ψ_i based on change to ϕ_i
- 3) Accept the probability $\min(1, \frac{P_{new}}{P_{old}})$

6.6.2 Jacobian Based Method

The Jacobian based method attempts to approximate a solution. The problem can be formulated as follows: Given X, find θ such that $X = F(\theta)$. We look for solution in the form $X = J*dQ$ and start from arbitrary initial guess $X_0 = F(\theta_0)$.

Jacobian J is an $m \times n$ matrix. Parameter m has dimension of X and therefore $m = 6$. In order to approach the problem for every n, Jacobian based method decomposes it into subproblems where $n = 6$. After solving the smaller problems, it recombines them using summation.

6.6.2.1 Case $n = 6$

The minimal number of required parameters is 6 according to the Grübler's formula for spatial linkages. 3 parameters are required for translation and 3 for orientation.

1. Problem: Given X , find θ such that $X = F(\theta)$.
2. Produce an initial guess (X_0, θ_0) such that $X_0 = F(\theta_0)$.
3. Iteration:
 - (a) Interpolate linearly between X_0 and X ($X_1 \dots X_p$).
 - (b) for $j = 1$ to p do
 - i. $\theta_i = \theta_{i-1} + \alpha * J^{-1}(\theta_{i-1})(X_i - X_{i-1})$
 - ii. $X_i = F(\theta_i)$

6.6.2.2 Case $n > 6$

In this case matrix J is not square because $m = 6$. Therefore, we can only find a pseudo inverse J^+ following the relationship $J * J^+ = I$. After finding a pseudo inverse of J , we obtain the next approximation:

$$\theta = J^+ dX + d\theta_0$$

The computation of J^+ is done by performing singular value decomposition (SVD) of J . Thus we get $J = U \Sigma V^T$ as shown in Figure 6.16. U is $m \times m$

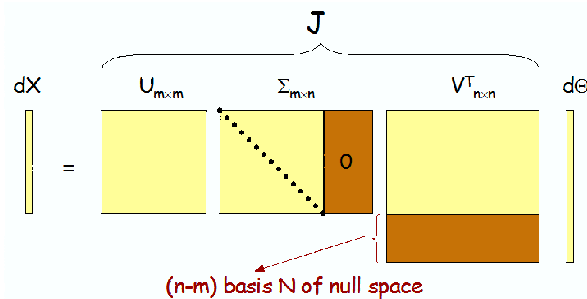


Figure 6.15: Result of SVD of J

square orthonormal matrix, V is $n \times n$ square orthonormal matrix. The picture reveals that the null space $\{dQ_0 \mid J dQ_0 = 0\}$ has $\dim = n - 6$.

The pseudo inverse J^+ is computed based on singular values of J :

- 1) $\sigma_i =$ Singular Values for J
- 2) $\Sigma^+ = \text{diag}(\frac{1}{\sigma_i})$
- 3) $J^+ = V \Sigma^+ U^T$

6.6.3 Minimization of Target Function T with Closure when $n > 6$

In this section we consider the situation when both position and orientations for the starting chain are fixed. Lotan et. al (2004) [LBDC04] proposed the algorithm for calculation of approximation.

Repeat

1. Compute Jacobian matrix J at current q
2. Compute null-space of J using singular value decomposition
3. Compute gradient $\nabla T(q)$
4. Move along projection of $-\nabla T(q)$ onto N until minimum is reached or closure is broken and obtain new q.

6.6.4 Decomposition Method for Randomly Sampling Conformations of Closed Chains

When n is becoming very large, it is impossible to use exact methods of modeling the closed chain. Therefore, additional effort is needed to decompose the modeling task. Here we consider the sampling-based decomposition algorithm [CSL02].

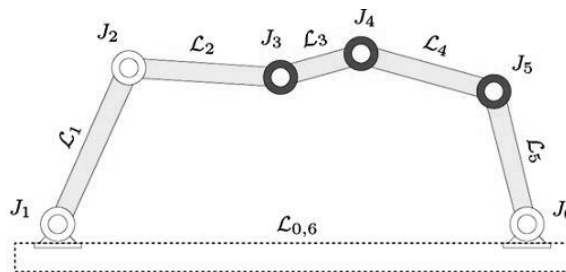


Figure 6.16: Example of decomposition into active subchain(s)

1. Find a passive (non-moving) sub-chain with 6 dof's. Thus, we subdivide the computation of the whole chain into 2 active sub-chains. Totally, they have $n-6$ dof's.
2. Sample the dof's parameters of the active sub-chains
3. Compute dof's of passive sub-chain using exact IK solver

References

- [CSJD04] Coutsias E., Seok, C., Jacobson, M. and Dill, K. (2004) *A Kinematic View of Loop Closure*. Journal of Computational Chemistry, 2004.
- [WC91] Wang, L. and Chen, C. (1991) *A Combined Optimization Method for Solving the Inverse Kinematics Problem of Mechanical Manipulators*. IEEE Transactions On Robotics and Automation, 1991, **v.7**, 489-498
- [GL96] Golub, G. and Van Loan, C. (1996) *Matrix computations*. Johns Hopkins University Press, 1996, **ed. 3**
- [LBDL04] Lotan, I., van den Bedem, H., Deacon, A. and Latombe, J.-C (2004) *Computing Protein Structures from Electron Density Maps: The Missing Loop Problem*. In Proceedings of 6th Workshop on Algorithmic Foundations of Robotics, (WAFR'04).
- [CSL02] Cortes, J., Simeon, T. and Laumond, J. *A random loop generator for planning the motions of closed kinematic chains using PRM methods* IEEE International Conference on Robotics Automation, 2002, 2141-2146.