I. Troch, F. Breitenecker, eds. ISBN 978-3-901608-35-3

A PROJECTED NON-LINEAR CONJUGATE GRADIENT METHOD FOR INTERACTIVE INVERSE KINEMATICS.

M. Engell-Nørregård[1], K. Erleben[1]
[1]eScience Center, University of Copenhagen, Denmark.

Corresponding author: M. Engell-Nørregård, eScience Center, University of Copenhagen, Denmark. DK–2100 Copenhagen, Universitetsparken 1, Denmark, mort@diku.dk

Abstract. Inverse kinematics is the problem of posing an articulated figure to obtain a wanted goal, without regarding inertia and forces. Joint limits are modeled as bounds on individual degrees of freedom, leading to a box-constrained optimization problem. We present A projected Non-linear Conjugate Gradient optimization method suitable for box-constrained optimization problems for inverse kinematics. We show application on inverse kinematics positioning of a human figure. Performance is measured and compared to a traditional Jacobian Transpose method. Visual quality of the developed method is evaluated.

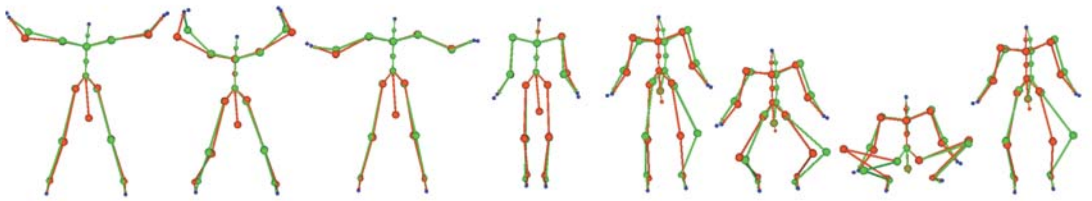


Figure 1: The benchmarking suite used for the testing. The green skeleton is the inverse kinematics driven skeleton, the red skeleton is a motion capture reference. Each pose shows the inverse kinematics skeleton, overlaid with the motion capture reference.

1 Interactive Inverse Kinematics

Inverse kinematics is used in both Graphics and Robotics. Recently attempts at solving the inverse kinematics problem using optimization techniques have been performed [1, 2, 3]. The problem can be formulated as a general problem and arbitrary optimization methods may be used to solve the problem. We want an iterative method which has an acceptable compromise between performance and accuracy. Experience in Graphics indicate that the Non-linear Conjugate Gradient method have worked very well in similar conditions, for example interactive cloth simulation [4]. Therefore Non-linear Conjugate Gradient may be a feasible choice for inverse kinematics.

We present a novel method based on the Non-linear Conjugate Gradient method [5] combined with a projected line search, inspired by the projected gradient method [6]. This box-constrained optimization approach is tested on a human figure using motion captured data as a reference as illustrated in Figure 1. Results show that the method has better accuracy and is faster than comparable methods for Inverse Kinematics. Further the Non-linear Conjugate Gradient method does not suffer from oscillating behavior exhibited by Jacobian methods [7]. Altogether making the Projected Non-linear Conjugate Gradient method a more suitable choice for interactive inverse kinematics.

The main difference between our approach and an unconstrained Non-linear Conjugate Gradient method is the projected line search. Our focus in this paper is to explain the addition of the projected line search to the unconstrained method. The Non-linear Conjugate Gradient method is described sufficiently elsewhere [5]. The projected line search can be explained as a very effective specialization of an active set approach. Instead of adding and removing constraints from an active set one by one, the entire active set is recalculated in each iteration. This is only possible if the constraints are independent of each other and thus the projected line search can only be performed where constraints are bounds on individual variables. Fortunately, this is often the case for real-time character animation and is the standard in several motion capture file formats such as the asf and amc file format from Acclaim [8].

2 An Optimization Approach for Inverse Kinematics

Inverse Kinematics is a widely used technique for posing articulated figures in both simulations, robotics and computer generated graphics [9, 2]. To state the problem more formally: Given a serial mechanism we can set up a coordinate transformation from one joint frame to the next. Thus we can find one transformation that takes a point specified in the frame of the end-effector into the world coordinate frame of the mechanism. We write it in a general way as

$$\vec{y} = \vec{F}(\vec{\theta}) \tag{1}$$

We have the set of joint parameters $\vec{\theta}$. We can change the values of $\vec{\theta}$ and gain explicit control over the position and orientation of the end-effector, $\vec{y}$. This is commonly known as forward kinematics. Given a desired goal position, $\vec{g}$, one seeks the value of $\vec{\theta}$ such that

$$\vec{\theta} = \vec{F}^{-1}(\vec{g}) \tag{2}$$

This is known as inverse kinematics and it is the problem we address in this paper. Motion is synthesized by making a discrete time-sampling $\vec{g}$ and solving each instant independently.

Without loss of generality, we will choose homogeneous coordinates. Given a chain with n links, we have ${}^{0}T_1, \ldots, {}^{n-1}T_n$ homogeneous coordinate matrices. We will assume that the i^{th} joint depends on the parameters $\vec{\theta}_i$. That is ${}^{i-1}T_i$ a function of $\vec{\theta}_i$. We will specify the the end-effector and the goal placement by the agglomerated vectors

$$[\vec{y}]_n = \begin{bmatrix}\vec{p}^T & \hat{i}^T & \hat{j}^T\end{bmatrix}_n^T, [\vec{g}]_0 = \begin{bmatrix}\vec{p}_{\text{goal}}^T & \hat{i}_{\text{goal}}^T & \hat{j}_{\text{goal}}^T\end{bmatrix}_0^T \in \Re^3 \times S^2 \times S^2, \tag{3}$$

where the bracket notation $[\cdot]_i$ is used to make it explicit that vectors are expressed in the coordinates of the i^{th} joint frame. Using homogeneous coordinates we can write the instantaneous placement of the end-effector as

$$\vec{y} = \begin{bmatrix}\vec{p}\\ \hat{i}\\ \hat{j}\end{bmatrix}_0 = \begin{bmatrix}{}^{0}T_n & 0 & 0\\ 0 & {}^{0}T_n & 0\\ 0 & 0 & {}^{0}T_n\end{bmatrix} \begin{bmatrix}\vec{p}\\ \hat{i}\\ \hat{j}\end{bmatrix}_n = \vec{F}(\vec{\theta}), \tag{4}$$

where ${}^{0}T_n = {}^{0}T_1 \cdots {}^{n-1}T_n$ and $[\vec{p}]_n = \begin{bmatrix}0 & 0 & 0 & 1\end{bmatrix}^T$, $[\hat{i}]_n = \begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}^T$, and $[\hat{j}]_n = \begin{bmatrix}0 & 1 & 0 & 0\end{bmatrix}^T$. Given a branched mechanism containing K kinematic chains, where each chain has exactly one end-effector. We agglomerate the K end-effector functions into one function

$$\vec{y} = \begin{bmatrix}\vec{y}_1 & \ldots & \vec{y}_j & \cdots & \vec{y}_K\end{bmatrix}^T = \begin{bmatrix}\vec{F}_1(\vec{\theta}) & \ldots & \vec{F}_j(\vec{\theta}) & \ldots & \vec{F}_K(\vec{\theta})\end{bmatrix}^T = \vec{F}(\vec{\theta}), \tag{5}$$

where $\vec{y}_j$ is the world coordinate position of the j^{th} end-effector, and $\vec{F}_j(\vec{\theta})$ is the end-effector function corresponding to the j^{th} kinematic chain.

Previous research e.g. [3] has shown that optimization may be a good approach for solving this problem. Further the traditional Jacobian Inverse methods are known to suffer from problems with ill conditioning and unstable behavior. The optimization approach does not suffer from these problems since the Jacobian is not inverted and the ill-posing is countered by the line search. From the end-effector function we create the objective function

$$f(\vec{\theta}) = (\vec{g} - \vec{F}(\vec{\theta}))^T W (\vec{g} - \vec{F}(\vec{\theta})), \tag{6}$$

where W is a symmetric positive definite and possible diagonal matrix and $\vec{g} = \begin{bmatrix}\vec{g}_1^T & \cdots & \vec{g}_K^T\end{bmatrix}^T$ is the agglomerated vector of end-effector goals. The optimization problem is,

$$\vec{\theta}^* = \min_{\vec{\theta}} f(\vec{\theta}) \quad \text{subject to} \quad \vec{l} \leq \vec{\theta} \leq \vec{u} \tag{7}$$

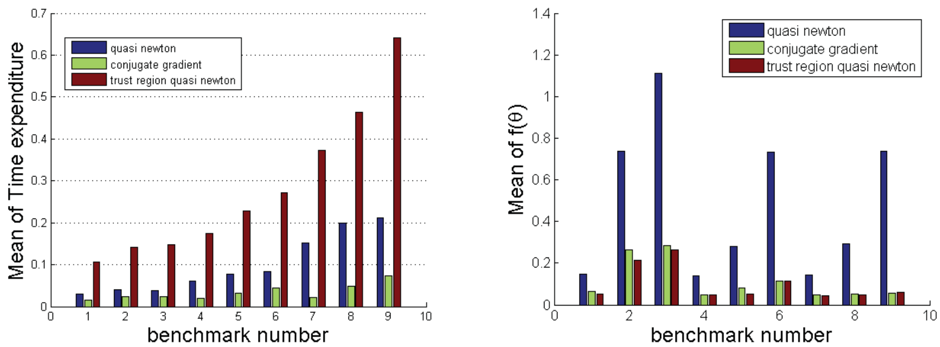
which models the minimum and maximum joint parameter values. Here $\vec{l}$ is a vector containing the minimum joint limits and $\vec{u}$ is a vector of the maximum joints limits. This implies $\vec{l} \leq \vec{u}$ at all times. These limits can be estimated from sampled data as proposed in [10].

Our formulation is a squared weighted norm measuring the distance between the goal positions and the end-effector positions. This formulation is similar to Zhao et al. [3]. The main difference being that we have agglomerated all goals and introduced a general weighting matrix instead of dealing with K square weighted summation terms.

3 Solution by Projected Non-linear Conjugate Gradient Method

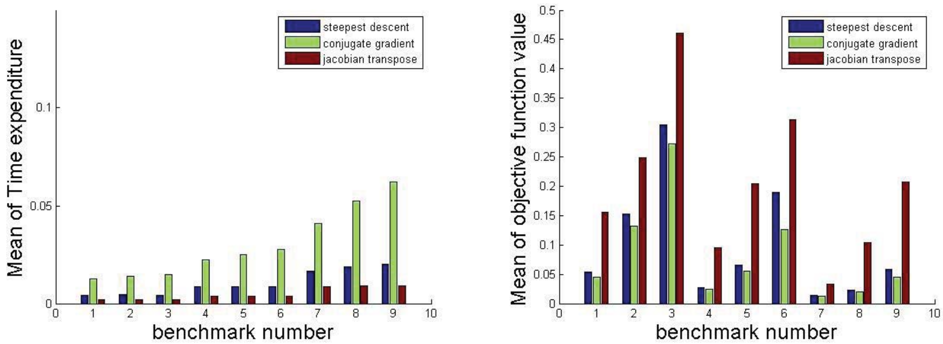
In our context time is important. Therefore we want a method with a low iteration cost. The Non-linear Conjugate Gradient method is a good choice due to its low iteration cost compared to e.g. Quasi-Newton methods, and better accuracy than e.g. Steepest Descent method. Further the method has shown very good results in practical tests [5, 11].

The method proposed in this paper works by calculating a search direction using the Non-linear Conjugated Gradient approach. A line search is then performed along this direction, where each iteration of the line search projects the search direction unto the feasible region. Thus it is ensured that the new iterate is feasible even if the initial iterate was not which makes the method very robust. Any inexact line search technique can be used. Our method was tested with a simple binary line search chosen for its simplicity and performance. To make sure we have a robust method we need an update strategy which can ensure global convergence while guaranteeing an automatics restart procedure every time there is insufficient progress. The Polak Ribiere$^+$ method ensures this. This method



(a) Timing measurements   (b) Accuracy measurements

Figure 2: The projected Non-linear Conjugate Gradient method compared with two Quasi-Newton methods in 9 different benchmark configurations. Notice, that while the projected Non-linear Conjugate Gradient method is much faster, it still compares with the other methods in regards to accuracy.



(a) Timing measurements   (b) Accuracy measurements

Figure 3: The projected Non-linear Conjugate Gradient method compared with Steepest Descent and Jacobian Transpose methods in 9 different benchmark configurations. In these tests the Projected Non-linear Conjugate Gradient method obtain the best results at the cost of a small increase in computation time compared to the Steepest Descent method.

is a simple modification of the Polak Ribiere update method and both are described in detail in [5]. The Polak Ribiere update is given by,

$$\beta_{i+1}^{pr} = \frac{\nabla f_{i+1}(\nabla f_{i+1} - \nabla f_i)}{\|\nabla f_i\|}. \tag{8}$$

Unfortunately it turns out that this is not always a descent direction so a slight modification must be performed to ensure this, which transforms the Polak Ribiere to the Polak Ribiere$^+$ update. The non-descent direction only crops up when β^{pr} is negative so a simple test fixes this, and the Polak Ribiere$^+$ update is thus given by,

$$\beta^{pr+} = max(\beta^{pr}, 0). \tag{9}$$

The upside is that this version has a build in restart procedure since $\beta^{pr+} = 0$ means that the update reverts to $\mathbf{p_{i+1}} = -\nabla f_{i+1}$ which is the steepest descent direction.

The real time aspect dictates a fast approach for constraining the result. Since the problem is posed as a box-constrained problem using a projected line search is the obvious choice. The conjugated gradient step can be easily adapted to the projection approach since it shares many features with the steepest descent method. They are both line search methods, the Conjugate Gradient method is derived by using the steepest descent as its starting point and the subsequent search directions are based on linear combinations of the steepest descent directions in this and previous iterations. Projection is performed by a simple mathematical procedure in which each element in the search vector is set to

$$\vec{P} = \min(\vec{u}, \max(\vec{l}, \vec{\theta} + \vec{p})) - \vec{\theta} \tag{10}$$

where $\vec{\theta}$ is the previous iterate, $\vec{p}$ is the search direction. and $\vec{P}$ is the projected search direction.

4 Improved Performance and Quality

The method was tested against 4 other methods in a benchmark suite consisting of 9 different configurations where spatial-coherence and maximum allowed number of iterations was varied. A motion capture animation was used and the system tried to mimic the motion of this animation using only information on the end-effector positions and joint limits. Figure 2(a) and 2(b) show comparisons with two Quasi-Newton methods while Figure 3(a) and 3(b) show comparisons with the traditional Jacobian Transpose and the Steepest Descent methods. Figure 1 show the visual results of the Projected Non-linear Conjugate Gradient method.

Our primary interest is to be able to guarantee that the Conjugate Gradient method can run approximately 20 - 25 fps (frames per second). This corresponds to a maximum frame time of approximately 0.05 seconds. As shown in Figure 2(a) the Conjugate Gradient method has a time expenditure below 0.05 seconds in all benchmark tests except number 9 which is below 0.1 second. The Trust region method takes more than 0.1 seconds so this method is not a viable choice for this type of architecture. The line search Quasi–Newton method obtain acceptable results for the 3 first benchmarks. The Conjugate Gradient method is several times faster than the other methods used in the benchmark test. The chosen computer architecture has a tremendous impact on the time spend for these tests. We used a laptop with an Intel 1.66 Ghz. CPU, which is sufficiently modest to make sure our results are general.

The quality of the solutions is quantified by measuring the closeness of the solution to the desired result. In our case the desired result is an objective value below 0.05 units which roughly corresponds to 1 *cm* in real world terms. As shown in Figure 2(b) and 2(b) the Projected Non-linear Conjugate Gradient method provides a nice compromise between accuracy and performance, since it compares to the Steepest Descent method in performance and the Trust-region Quasi-Newton method in accuracy.

4.1 Discussion and Conclusion

The projected Non-linear Conjugate Gradient method has been implemented in the OpenTissue Library where it is possible to test Inverse Kinematics against motion captured data references. This makes it possible to test performance with regards to accuracy as well as time expenditure. Our results imply that the projected Non-linear Conjugate Gradient method out-performs traditional methods as well as comparable optimization alternatives e.g the Quasi-Newton methods, either in performance, accuracy or both. With regards to the steepest descent method it is a matter of whether time-expenditure or accuracy is valued the most. The steepest descent is faster but the Projected Non-linear Conjugate gradient obtains better results.

5 Acknowledgments

This research was done as part of the HUMIM project under the eScience Center, Faculty of Science, University of Copenhagen. The data used in this project was obtained from mocap.cs.cmu.edu. The database was created with funding from NSF EIA-0196217.

6 References

[1] Ho, E. S. L., Komura, T., and Lau, R. W. H. (2005) In VRST ’05: Proceedings of the ACM symposium on Virtual reality software and technology New York, NY, USA: ACM. pp. 163–166.

[2] Fêdor, M. (2003) In SCCG ’03: Proceedings of the 19th spring conference on Computer graphics New York, NY, USA: ACM Press. pp. 203–212.

[3] Zhao, J. and Badler, N. I. (1994) *ACM Trans. Graph.* **13(4)**, 313–336.

[4] Baraff, D. and Witkin, A. (1998) In SIGGRAPH ’98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques New York, NY, USA: ACM. pp. 43–54.

[5] Nocedal, J. and Wright, S. J. (1999) Numerical optimization, Springer Series in Operations ResearchSpringer-Verlag, New York.

[6] Rosen, J. B. (1961) *Journal of the Society for Industrial and Applied Mathematics* **9(4)**, 514–532.

[7] Wellman, C. Inverse kinematics and geometric constraints for articulated figure manipulation Master’s thesis Simon Fraser University (1993).

[8] Acclaim entertainment : Went bankrupt in 2004, motion capture format placed in public domain.

[9] Buss, S. R. Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. Unpublished survey april 2004.

[10] Engell-Nørregård, M. and Erleben, K. (2009) In WSCG 2009: 17-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision : .

[11] Hager, W. W. and Zhang, H. (2006) *ACM Trans. Math. Softw.* **32(1)**, 113–137.