

Continuous Inverse Kinematics in Singular Position

Patrick Vonwirth and Karsten Berns

Department of Computer Science,
Technische Universität Kaiserslautern, 67663 Kaiserslautern, Germany
{vonwirth,berns}@cs.uni-kl.de
<https://agrosy.informatik.uni-kl.de>

Abstract. The problem of calculating inverse kinematics of a robotic manipulator is known to be non-trivial and not straightforward to solve for centuries. Hence, multiple different approaches have been developed, extended, and further developed that iteratively approximate toward a suitable solution. Unfortunately, all these existing solutions share the problem to get unreliable in singular positions – a standard configuration of human legs, e.g. when standing. Within this work, a simple extension to the iterative *Damped Least Square* algorithm is presented that covers the problematic, singular configuration case. The proposed algorithm is thereby focusing on continuous solving of small, iterative pose changes.

Keywords: Inverse Kinematics, Jacobian, Least Square, Singular Position, Humanoid Robots

1 Introduction

Inverse Kinematics is one of the basic problems in classical robotics. Every robotic arm that e.g. is performing pick and place operations, or performs a trajectory following task, requires to solve the inverse kinematic problem. Many different approaches have been developed to solve this problem over the past years, but none of them is universally valid and suitable for all purposes and requirements.

The difficulty of the inverse kinematics problem arises from a couple of circumstances that usually makes a solution mathematically hard to calculate, non-unique or not existing. However, the main problems arise from the mechanical construction. Currently, the majority of the existing robot manipulators are built with rotational joints. In Cartesian workspace coordinates, these circular motions require a quadratic representation that completely excludes the usage of linear algebra to calculate closed-form solutions. Furthermore, the kinematic function $\mathcal{K}(\cdot)$ is neither injective nor surjective which leads to the result that the inverse kinematic function $\mathcal{K}^{-1}(\cdot)$ mathematically does not exist. Solving the inverse kinematics results in an arbitrary number of possible results, including zero and infinity.

Since linear joints are much simpler to handle than rotational ones, the strategy described in the paper at hand is restricted to rotation-only kinematics. Throughout the paper at hand, the angular configuration of all joints is therefore represented by a vector of angles Θ . The inverse kinematic problem is hence defined by finding an angular configuration Θ_G that satisfies $\mathcal{K}(\Theta_G) = \mathbf{G}$ for an arbitrary desired pose \mathbf{G} , given in the Cartesian workspace. Nevertheless, linear joints can easily be integrated into the proposed methodology.

Due to the basic nature of the inverse kinematics, several different strategies have been developed to solve this problem. Popular in the field of robotics, there are iterative gradient descent algorithms, all implementing a *Newton-Raphson Approximation* variation, of the following general form, iterating over i :

$$\begin{aligned}\mathbf{e}_i &= \mathbf{G} - \mathcal{K}(\Theta_i) && \text{Pose Error against Goal} \\ \boldsymbol{\delta}_i &= \nabla^{-1}(\Theta_i) \cdot \mathbf{e}_i && \text{Linearized Adaptation} \\ \Theta_{i+1} &= \Theta_i + \gamma \boldsymbol{\delta}_i && \text{Damped Joint Update}\end{aligned}\quad (1)$$

with $\nabla(\Theta_i)$ being an arbitrary gradient function at the operating point Θ_i . The parameter $\gamma \in (0, 1]$ specifies the approximation rate of every individual iteration step. Hence, the more inaccurate $\nabla(\cdot)$ represents the reality, the smaller γ has to be chosen in order to keep the iteration process stable at the cost of an increasing number of required iteration steps. An initial guess Θ_0 is required to start the iteration sequence. Usually, Θ_0 is simply set to the actual joint configuration. The iteration process is stopped whenever the workspace squared error drops below a predefined accuracy threshold $\|\mathbf{e}_i\|^2 \leq \epsilon_e$, or no more approximation progress can be observed $0 \leq \|\mathbf{e}_{i-1}\|^2 - \|\mathbf{e}_i\|^2 \leq \epsilon_p$. Furthermore, the iteration process can be limited by a maximum number of iteration steps or computation time.

Simple implementations of the inverse gradient function $\nabla^{-1}(\cdot)$ are the *Jacobian Inverse*, the *Jacobian Transpose*, the *Jacobian Pseudoinverse*, or the so-called *Damped Least Square* (DLS) approach. All of them have been used successfully in early robot control systems, e.g. [2,11,14,16]. While the DLS method especially addresses the problem of $\mathcal{K}(\cdot)$ not being bijective, a variety of extensions and adaptations, e.g. *Selectively DLS* [4] or *User-Defined Weighted DLS* [5], have been developed to further improve the accuracy, stability around singular positions and approximation speed of the basic DLS algorithm [6,8].

Other, Jacobian-free inverse kinematic strategies also have been developed outside the field of robot control. In CGI applications, for example, inverse kinematics are required to map captured (human) motions onto complex, virtual, animated character models. A representative, but not complete, set of these approaches should be mentioned in the following:

Cyclic Coordinate Descent, a heuristic approach, introduced by Wang et al. [15] is especially declared to be suitable for large-distance problems which, unfortunately, contradicts with the motivation of this work. An other heuristic approach, proposed by Courty et al. [7], is based on *Sequential Monte Carlo Simulations* using particle filtering techniques. The authors Grochow et al. propose an AI-based approach that learns suitable inverse kinematic solutions from

previously observed and trained poses [10]. As a natural property of all AI-applications, this method strongly depends on the quality, space coverage, and smoothness of the training data set. With respect to continuous control of a robot manipulator, however, such heuristic methodologies always include some rest risk of unpredictable and undesired behavior. Hence, more deterministic approaches should be preferred over heuristic ones in the concrete application of arbitrary, continuous robot control.

A geometrically inspired approach for the purpose of virtual rope simulation has been presented by Brown et al. [3] and extended to the so-called FABRIK algorithm (*Forward And Backward Reaching Inverse Kinematics*) by Aristidou et al. [1]. The FABRIK algorithm shares the exact same singularity problem as the classical *Newton-Raphson* approximation strategies. However, it comes with the massive drawback that the requirements of rope simulation and robot control are mostly distinct from each other. E.g. introducing joint limits strongly attacks the algorithm's convergence while blowing up its complexity at the same time. Hence, extending a classical, Jacobian-based approximation strategy best meets the requirements of continuous robot control.

Within the work at hand, an extension of the well-known DLS method is presented. The newly introduced extension especially treats the problem of getting stuck in singular positions, i.e. when all gradient vectors are orthogonal to the direction of the desired displacement. In section 2, the basic structure and reasoning of a DLS approximation is introduced. The problem of the DLS iteration getting stuck in singular position is covered in section 3. Section 4 describes the adaptation of the former presented approach for weighted, continuous execution. An experimental validation of the presented methodology is given within section 5 at the example of continuous inverse kinematics calculations of the planar, human-shaped robotic leg CARL. Finally, section 6 concludes the overall results and gives an outlook on the planned operational scenario.

2 Iterative Damped Least Square Approximation

When inverting the kinematics function $\mathcal{K}(\Theta)$ of an arbitrary, robotic manipulator with only rotational joints, there are three main mathematical problems:

1. \mathcal{K} is not linear
2. \mathcal{K} is not surjective
3. \mathcal{K} is not injective

From the non-linearity of \mathcal{K} , it can directly be derived that there is no closed-form solution existing within linear algebra. Although it is possible to manually calculate closed-form solutions for such non-linear problems, such solutions are specialized to one particular kinematic structure. Hence, they lack both, generality and adaptability to special purposes. Iterative, general-purpose approaches that step-wise approximate the result, as introduced in equation (1), are therefore to be preferred over robot-specific solutions. Within every iteration step, these types of algorithms normally linearize \mathcal{K} at one particular operating point,

the so-called *Gradient Field*. Now, simple methods from linear algebra can be used to invert and solve the linearized kinematics. Finally, the operating point is slowly shifted towards the linearized solution. Unfortunately, the usage of an iterative algorithm comes at some additional cost:

4. The number of required iteration steps varies depending on the input
5. The iterative approximation process can get stuck in singular positions

Using an upper limit of iteration steps, the input dependency can be handled. Although, aborting the approximation process leads to a loss of accuracy, in the context of this work, i.e. continuous inverse kinematics calculation, the lost accuracy will be caught up by the execution of the follow-up approximation. An extension to the DLS algorithm that offers the ability to break out of singular positions during the iterative approximation process is presented in section 3

With the introduction of using the pseudoinverse, that is doing nothing else than minimizing the least square error, the problem of \mathcal{K} not being surjective is solved. Whenever a desired goal pose is outside of the robot's reachable workspace, the closest possible solution is returned. The problem of \mathcal{K} not being injective has been treated by adding an additional damping term. The general DLS method extends the calculation formula of the intermediate approximation step $\boldsymbol{\delta}_i$ within equation (1) to

$$\boldsymbol{\delta}_i = \min_{\boldsymbol{\delta}} \| \nabla(\Theta_i)\boldsymbol{\delta} - \mathbf{e}_i \|^2 + \lambda_D \| \boldsymbol{\delta} \|^2 \quad (2)$$

with $\lambda_D > 0$ being an arbitrarily chosen parameter that weights the additional damping term against the error minimization. Analyzing the effect of λ_D , it can be seen that the added damping term is pulling the step-wise change $\boldsymbol{\delta}_i$ towards zero. Especially, it is slowing down the approximation rate of the overall algorithm which is the identical functionality of the parameter γ . Hence, the approximation rate parameter γ can be removed from the equations (1) in case λ_D is present. λ_D has to be chosen large enough in order to keep the approximation process stable, but can automatically be reduced in value while the algorithm is step-wise approaching the desired pose. For some suitable initialization value $\lambda_0 > 0$, a very simple adaptation rule is to exponentially decrease λ_D in every iteration step.

$$\lambda_D = \epsilon + \lambda_i > 0 \quad \text{where } \lambda_i = \alpha \lambda_{i-1} \quad (3)$$

with $0 < \alpha < 1$ being the exponential decrease rate of λ_i and $\epsilon_e > \epsilon > 0$ being an arbitrary, small constant that assures a minimum amount of positive definiteness of the underlying quadratic minimization problem.

3 Using The Gradient Derivative to Break Out of Singular Positions

The problem of singular positions during the iterative approximation of \mathcal{K}^{-1} is visualized in figure 1 at the example of a simple, planar, two-segmented robot

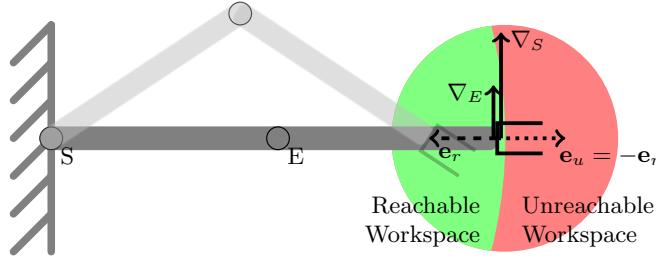


Fig. 1: The problem of getting stuck in singular position: The two vectors ∇_S and ∇_E represent the linearized gradients of the two joints S and E respectively. Normal to these, e_r and its negative e_u represent desired directions of motion. While a motion in the direction of e_r is possible, its negative is not. Due to the motions orthogonality, however, any standard DLS algorithm will stick to the singular position in *both* cases.

arm. It can be seen that both gradient vectors $\nabla_{S/E}$ are parallel to each other. Hence, the combined 2×2 gradient matrix $\nabla(S, E) = [\nabla_S \nabla_E]$ is not of full rank anymore. As a direct consequence, it can be followed that there is one direction vector $\mathbf{u} \neq \mathbf{0}$ existing that always becomes zero when being projected into the image of $\nabla(S, E)$. I.e. the best possible approximation of \mathbf{u} using $\nabla(S, E)$ is the null vector $\mathbf{u} \approx \nabla(S, E)\mathbf{0}$ which encodes the statement of *no further approximation* being possible. However, the example shown in figure 1 proofs the existence of possible motions in such a singular situation.

Corollary: Any iterative approximation algorithm in the general form of equation (1) can get stuck, if and only if the rank of the linear gradient field function $\nabla(\cdot)$ is able to become less than its number of rows (after removing gradients of joints already being in their limit position in the direction of interest).

Obviously, the only solution is to assure that the gradient function $\nabla(\cdot)$ always keeps full rank.

One solution is to guarantee linear independence of the individual (relevant) gradient vectors, i.e. applying small changes in direction. Unfortunately, the circular nature of the reachable workspace (due to the rotational joints) prohibits any change in orientation to the gradient vectors. If a gradient vector in singular position would differ from the tangential plant at the border of the reachable workspace, either a positive or a negative motion into that gradient's direction would virtually allow the system to move into the unreachable space. Furthermore, restricting the motion of such an altered gradient to either positive or negative direction cannot be handled in a simple manner, since both directions are geometrically identical and the right choice depends on the context.

Since the content of $\nabla(\cdot)$ cannot be altered to assure full rank, the gradient matrix has to be extended with additional, orthogonal column vectors. The gradient itself, however, is nothing else than a very rough Taylor approximation

of the real motion function. Hence, one further addend of the Taylor series, i.e. the gradient derivative $\dot{\nabla}(\cdot)$, is a natural extension of $\nabla(\cdot)$. Since the second derivative in the Taylor series is multiplied with a square, it is limited to non-negative multiplication factors. Transferred into the approximation iteration, this means that it is necessary to limit $\dot{\nabla}(\cdot)$ to non-negative multiples. With respect to the example, shown in figure 1, this property is exactly what is required to allow motion in direction of \mathbf{e}_r , but not in its negative direction \mathbf{e}_u . The orthogonality of $\dot{\nabla}(\cdot)$ to $\nabla(\cdot)$ assures full rank of the combined matrix $[\nabla \dot{\nabla}]$. With the extension of the gradient function, also the minimization variables of equation (2) have to be extended respectively, i.e. $\boldsymbol{\delta} \mapsto (\boldsymbol{\delta}, \dot{\boldsymbol{\delta}})^T$. It is easy to see that $\dot{\nabla}(\cdot)$ can be reduced to only relevant, non-zero columns. This way, also the number of additional minimization variables $\dot{\boldsymbol{\delta}}$ and hence the computational effort can be reduced.

When using the standard Jacobian as gradient function, its derivative represents the curvature of the circular motion of a joint. This curvature is geometrically represented by a normal vector of the circular trajectory, pointing towards the joint axis. Figure 2 sketches these vectors at the example of the already introduced planar, two-link robot arm.

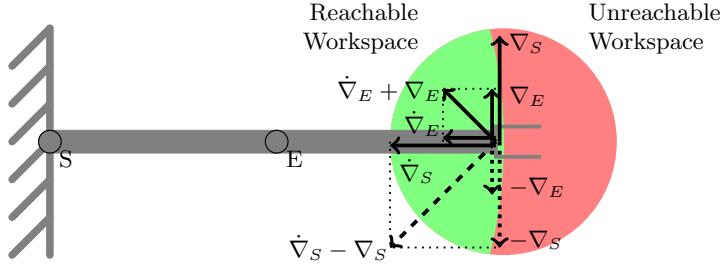


Fig. 2: Extended gradient vectors, visualized at the example of a planar, two-link manipulator in scalar position. Visualized are the gradient vectors $\nabla_{S/E}$, the gradient derivatives $\dot{\nabla}_{S/E}$ as well as two possible gradient-extension-vectors $(\dot{\nabla} \pm \nabla)_{S/E}$ respectively to both joints S and E.

The quadratic coupling of the first and second derivative terms in the Taylor series cannot be represented with linear equations. It is therefore not possible to couple the minimization variables of both, $\nabla(\cdot)$ and $\dot{\nabla}(\cdot)$ adequately. However, the second-order terms are only required in singular positions. A linear coupling via the gradient function itself is therefore a suitable solution as long as $\dot{\boldsymbol{\delta}}$ is strongly kept close to zero in non-singular positions. Furthermore, the complete approximation solution of \mathcal{K}^{-1} is kept within the non-extended vector $\boldsymbol{\delta}$. The

DLS equation (2) hence extends to

$$\begin{pmatrix} \boldsymbol{\delta}_i \\ \cdot \end{pmatrix} = \min_{\boldsymbol{\delta}, \dot{\boldsymbol{\delta}} \geq 0} \| \nabla(\boldsymbol{\Theta}_i) \boldsymbol{\delta} + (\dot{\nabla} \pm \nabla)(\boldsymbol{\Theta}_i) \dot{\boldsymbol{\delta}} - \mathbf{e}_i \|^2 + \lambda_D \| \boldsymbol{\delta} \|^2 + \lambda_S \| \text{diag}(\nabla^T(\boldsymbol{\Theta}) \mathbf{e}_i) \dot{\boldsymbol{\delta}} \|^2 + \epsilon \| \dot{\boldsymbol{\delta}} \|^2 \quad (4)$$

with $\lambda_S \gg 0$ being an arbitrary positive parameter much bigger than zero. Equal to equation (3), ϵ is an arbitrary, small constant to assure positive definiteness. $\text{diag}(\cdot)$ is a function returning a diagonal matrix with the argument vector as diagonal elements. Its argument $\nabla^T \mathbf{e}$ encodes the scalar product of each gradient vector with the desired motion. Hence, its entries become zero in their singular positions. Since \mathbf{e} is continuously decreasing throughout the approximation process, it is necessary to choose λ_S large enough to compensate for the decreasing length of \mathbf{e} . One suitable definition of λ_S is hence

$$\lambda_S = \frac{\lambda_{\inf}}{\| \mathbf{e}_i \|^2} \gg 0 \quad (5)$$

with λ_{\inf} being some arbitrary, big constant. The term $(\dot{\nabla} \pm \nabla)$ encodes column-wise addition or subtraction. Which columns to add and which to subtract can be chosen arbitrarily or even differently in every iteration step. The chosen operations only should encode a zigzag-like motion of the manipulator. In most robotic applications, it will be sufficient to fix that to a single, constant configuration. Note, that the addition or subtraction has to be chosen complementary to the desired zigzag motion. E.g.: A gradient subtraction from the derivation extension will be compensated by an addition of the original gradient.

4 Continuous, Weighted Inverse Kinematics

As introduced in equations (1) and (4), every iteration step of the overall approximation consists of two internal steps. At first, the linearized approximation $\boldsymbol{\delta}_i$ is calculated. Second, the operating point, i.e. the approximation result is updated with the former calculated linear approximation. These two sequential steps can be combined into a single equation that directly optimizes the linearized result vector. From equation (1), it can be derived that

$$\boldsymbol{\Theta}_{i+1} = \boldsymbol{\Theta}_i + \gamma \boldsymbol{\delta}_i \quad \mapsto \quad \boldsymbol{\delta}_i = \boldsymbol{\Theta}_{i+1} - \boldsymbol{\Theta}_i \quad (6)$$

with $\gamma = 1$ being removed as described in section 2. Inserting that into the extended DLS equation (4) leads to

$$\begin{pmatrix} \boldsymbol{\Theta}_{i+1} \\ \cdot \end{pmatrix} = \min_{\boldsymbol{\Theta}, \dot{\boldsymbol{\delta}} \geq 0} \| \nabla(\boldsymbol{\Theta}_i) (\boldsymbol{\Theta} - \boldsymbol{\Theta}_i) + (\dot{\nabla} \pm \nabla)(\boldsymbol{\Theta}_i) \dot{\boldsymbol{\delta}} - \mathbf{e}_i \|^2 + \lambda_D \| (\boldsymbol{\Theta} - \boldsymbol{\Theta}_i) \|^2 + \lambda_S \| \text{diag}(\nabla^T(\boldsymbol{\Theta}_i) \mathbf{e}_i) \dot{\boldsymbol{\delta}} \|^2 + \epsilon \| \dot{\boldsymbol{\delta}} \|^2 \quad (7)$$

Physical joint limits can now be introduced by simply adding component-wise lower and upper bounds to the minimization variables Θ since they now encode the physical result joint configuration: $\Theta_{min} \leq \Theta \leq \Theta_{max}$.

Since λ_D and λ_S are positive, user-defined constants, they both can be moved inside the three appropriate squared length expressions. Let \mathbf{W} be a symmetric, positive semidefinite matrix in the same dimension of the desired goal pose \mathbf{G} . With that matrix, the n-dimensional relevance of the desired goal pose can be encoded in the space of the desired pose \mathbf{G} . By a simple left-multiplication of the first minimization term, the goal accuracy gets weighted with respect to \mathbf{W} . Equivalently, the two error-based abortion criteria become weighted: $\|\mathbf{W}\mathbf{e}_i\|^2 \leq \epsilon_e$ and $\|\mathbf{W}\mathbf{e}_{i-1}\|^2 - \|\mathbf{W}\mathbf{e}_i\|^2 \leq \epsilon_p$. More precise, the quadratic error towards the desired pose gets weighted by \mathbf{W}^2 .

Defining $\nabla_i := \nabla(\Theta_i)$ and $\Theta^+ := (\Theta, \dot{\delta})^T$ and $\nabla_i^+ := [\nabla_i \ (\dot{\nabla}_i \pm \nabla_i)]$, equation (7) can now be rewritten to

$$\begin{aligned} \left(\begin{array}{c} \Theta_{i+1} \\ \cdot \end{array} \right) &= \min_{\Theta^+} \|\mathbf{W}\nabla_i^+ \Theta^+ - \mathbf{W}(\mathbf{e}_i + \nabla_i \Theta_i)\|^2 \\ &\quad + \left\| \begin{bmatrix} \sqrt{\lambda_D} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \Theta^+ - \begin{pmatrix} \sqrt{\lambda_D} \Theta_i \\ \mathbf{0} \end{pmatrix} \right\|^2 + \left\| \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \epsilon \mathbf{I} + \sqrt{\lambda_S} \operatorname{diag}(\nabla_i^T \mathbf{e}_i) \end{bmatrix} \Theta^+ \right\|^2 \end{aligned} \quad (8)$$

In order to apply general-purpose solver (e.g. *qpOASES* [9]) on the presented minimization problem, it has to be brought into a quadratic normal form:

$$\left(\begin{array}{c} \Theta_{i+1} \\ \cdot \end{array} \right) = \min_{\Theta^+} \frac{1}{2} \Theta^{+T} \mathbf{H} \Theta^+ + \Theta^{+T} \mathbf{g} \quad (9)$$

$$\text{s.t. } \left(\begin{array}{c} \Theta_{min} \\ \mathbf{0} \end{array} \right) \leq \Theta^+ \leq \left(\begin{array}{c} \Theta_{max} \\ \pi/2 \end{array} \right) \quad (10)$$

with:

$$\mathbf{H} := \nabla_i^{+T} \mathbf{W}^2 \nabla_i^+ + \epsilon \mathbf{I} + \begin{bmatrix} \lambda_i \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \lambda_S \operatorname{diag}(\nabla_i^T \mathbf{e}_i)^2 \end{bmatrix} \quad (11)$$

$$\mathbf{g} := -\nabla_i^{+T} \mathbf{W}^2 (\mathbf{e}_i + \nabla_i \Theta_i) - \begin{pmatrix} \lambda_i \Theta_i \\ \mathbf{0} \end{pmatrix} \quad (12)$$

Note, that λ_D has been replaced by $\epsilon + \lambda_i$ as introduced in equation (3).

5 Continuous Inverse Kinematics on the Planar Robotic Leg Carl

The introduced inverse kinematics approach is especially designed to be executed in singular positions. In classical robotics, especially industry robots, this system state usually is mechanically avoided in order to overcome the associated mathematical problems. On the other hand, in nature, singular leg configurations are one of the key features to lower power demands on muscles. In singular position, the main load is shifted away from the muscles and onto the skeletal

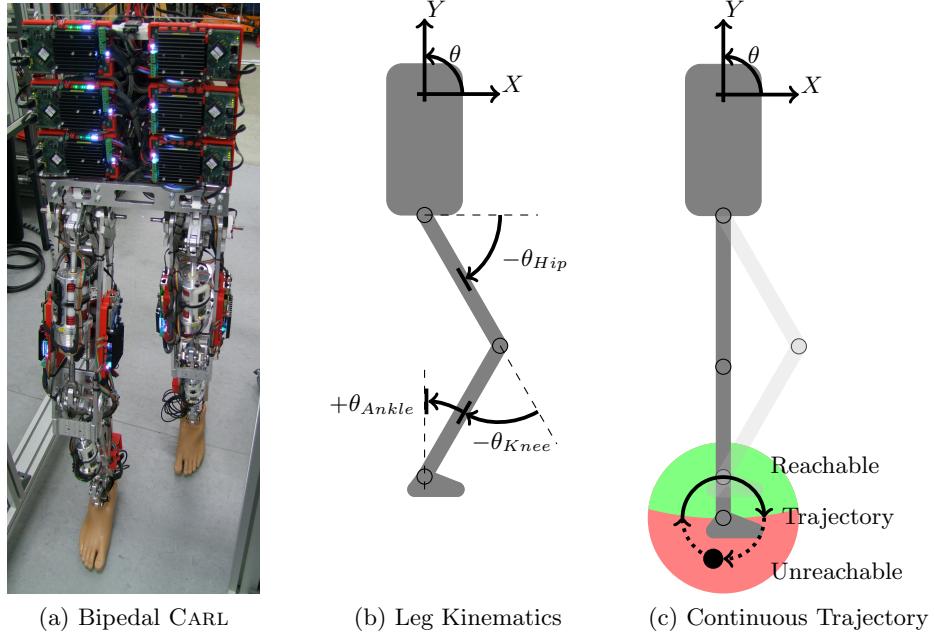


Fig. 3: Inverse Kinematics applied to the planar robotic leg CARL [12]. On the left side, figure (a) pictures the bipedal version of CARL. The kinematic definitions of one leg are described in figure (b). To the right, figure (c) sketches the desired circular trajectory that has been used to validate the presented inverse kinematics approach. The dashed part indicates an impossible desired trajectory outside the reachable workspace.

structure, hence supporting endurance and energy efficiency. The human-shaped robotic leg CARL, first introduced by Schütz et al. in [12], therefore offers perfect kinematics to evaluate the proposed inverse kinematics approach.

CARL is a compliant, planar robotic leg with three degrees of freedom in hip, knee and ankle joint. A two-legged, bipedal version of CARL is shown in figure 3a. The kinematic structure of CARL, as well as its actuation concept, is strongly related to human nature. Thigh and shank are constructed with an equal length of 0.42 m, the foot is a standard medical prosthesis [12]. Figure 3b sketches the angle definitions of the kinematic structure of CARL. Further details on the concrete kinematic equations can be found in [13]. Equivalent to nature, the knee is restricted to backward flexion only. Hence, the zigzag-motion to leave singularity ($\nabla \pm \nabla$), as introduced in equation (4), can be fixed to a knee flexion combined with hip and ankle rotations in opposite directions. As gradient function $\nabla(\cdot)$ the Jacobian matrix $\mathbf{J}(\Theta)$ is used.

In order to evaluate the validity and performance of the proposed inverse kinematics approach, the algorithm is executed on two different continuous tra-

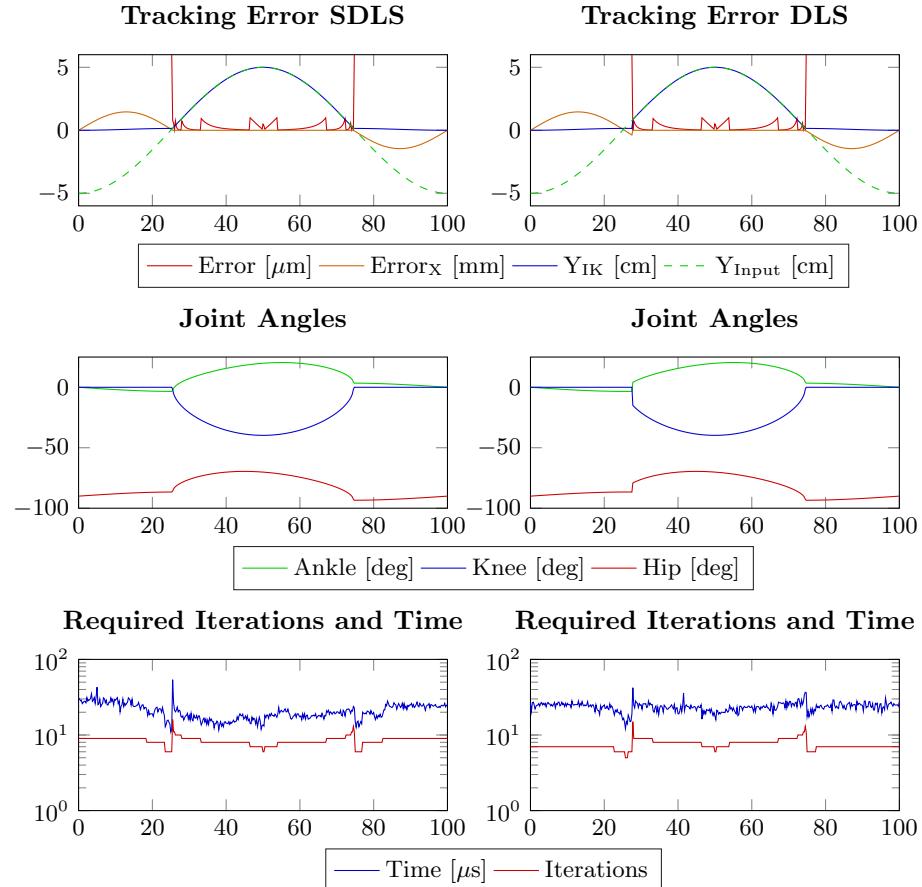


Fig. 4: Performance evaluation of the newly introduced inverse kinematics approach (SDLS) on the left side against the pure DLS methodology on the right side at the example of a continuous, circular trajectory as sketched in figure 3c.

jectories and in both times in comparison to a standard DLS approach. Both algorithms are implemented identically and executed under equal conditions. Standard DLS execution is simply achieved by switching off the newly proposed extension while keeping all other parameters identical.

The first desired trajectory is a continuous, circular motion as sketched in figure 3c, starting in the lowest point outside the reachable workspace. Second, the projection of that circular motion onto the vertical Y-axis is used as linear trajectory. Both trajectories are especially chosen to be partially outside the reachable workspace of CARL and to pass singular positions with mathematically problematic directions of motion. The reachable part of both trajectories approximately correlate with a positive Y_{Input} . All four algorithm executions are configured identically with a desired approximation accuracy of $1 \mu\text{m}$, i.e.

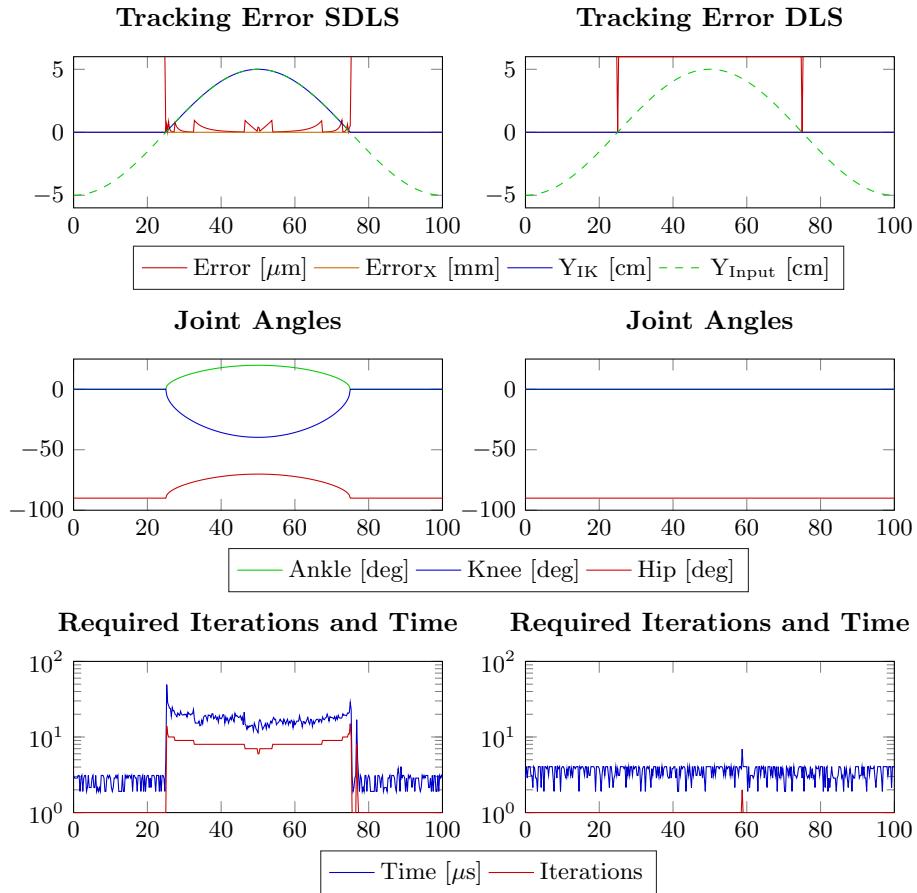


Fig. 5: Performance evaluation of the newly introduced inverse kinematics approach (SDLS) on the left side against the pure DLS methodology on the right side at the example of a continuous, linear trajectory. The linear trajectory is generated as a projection of the circular one as sketched in figure 3c onto the pure vertical Y-axis.

$\epsilon_e = 1 \times 10^{-12}$, $\lambda_0 = 1.5$, $\alpha = 0.3$, $\epsilon = \epsilon_p = 0.1 \cdot \epsilon_e$ and $\lambda_{\text{inf}} = 1 \times 10^9$. The results of the circular and linear experiments are plotted in the figures 4 and 5 respectively.

It can be seen that the approach, introduced within the work at hand, is capable of tracking both trajectories within the physical limits of the robot. The standard DLS approach, however, produces a sudden motion peak when leaving the singular position at 25% on the circular trajectory. On tracking the linear motion, the pure DLS approach fails in total. This exactly represents the singularity problem of standard gradient-descent methodologies as explained in

figure 1 within section 3. The standard DLS approximation is not able to leave the singular position, orthogonal to the direction of all gradients.

In both trajectories, the robot passes between reachable and unreachable workspace in singular position. These edge cases are clearly visible at around 25% and 75% by two peaks of the required iterations. The first peak represents the transition into the reachable workspace area and the ability to leave the singular position. The second peak indicates the opposite motion. From the executions of the circular trajectory, it can be seen that the proposed DLS extension produces a much smoother transition than the pure DLS approach when it comes to leave the singular position. Within the rest of the trajectory, no relevant effect can be observed. Hence, it can be summarized that the proposed singularity extension to the classical DLS approximation strategy is a valid solution to overcome the problematic edge case of leaving singular position at no relevant additional computational cost. Furthermore, the introduced DLS extension reduces the algorithm's overall sensitivity to its parametrization.

6 Conclusion and Outlook

The problem of inverse kinematics is a well-known, but still non-trivial problem. One main solution strategy in robotics is to iteratively approximate toward a valid solution. Although able to solve the inverse kinematics problem in general, such a strategy becomes unreliable in and close to singular positions. Unfortunately, being in singular position is a standard configuration of human legs. For human-shaped robots like CARL [12] that are especially designed to offer research on natural, human-like locomotion this single edge case of being in singular position becomes the default and hence a major problem.

Within the work at hand, a new extension to the iterative *Damped Least Square* (DLS) approximation [5] is introduced. The described extension especially treats the mathematically difficult edge case of leaving a singular position while it does not affect the overall rest functionality. Furthermore, a mathematical formulation is derived that specializes the individual iteration steps towards continuous calculations. This way, the proposed algorithm especially supports fast solutions to continuous motions.

The functionality of the newly introduced concept is validated at the example of two different, continuous trajectories. Its performance is estimated in comparison to the equivalent algorithm without the newly developed extension. The results show that within the problematic singular position, the reliability and performance of the classical DLS algorithm is clearly increased by the work at hand. However, detailed and in-depth performance optimization is not covered within the scope of this work. Improvements in convergence and calculation performance are hence expected to be possible by combining the work at hand with other DLS variations that e.g. focus on parameter optimization like the so-called *Selectively DLS* by Buss et al. in [4].

References

1. Aristidou, A., Lasenby, J.: FABRIK: A fast, iterative solver for the Inverse Kinematics problem. *Graphical Models* 73(5), 243–260 (Sep 2011)
2. Balestrino, A., De Maria, G., Sciavicco, L.: Robust Control of Robotic Manipulators. *IFAC Proceedings Volumes* 17(2), 2435–2440 (Jul 1984)
3. Brown, J., Latombe, J.C., Montgomery, K.: Real-time knot-tying simulation. *The Visual Computer* 20(2), 165–179 (2004)
4. Buss, S.R., Kim, J.S.: Selectively Damped Least Squares for Inverse Kinematics. *Journal of Graphics Tools* 10(3), 37–49 (Jan 2005)
5. Chiaverini, S., Egeland, O., Kanestrom, R.: Achieving user-defined accuracy with damped least-squares inverse kinematics. In: Fifth International Conference on Advanced Robotics 'Robots in Unstructured Environments. pp. 672–677 vol.1. IEEE, Pisa, Italy (1991)
6. Chiaverini, S., Siciliano, B., Egeland, O.: Review of the damped least-squares inverse kinematics with experiments on an industrial robot manipulator. *IEEE Transactions on Control Systems Technology* 2(2), 123–134 (Jun 1994)
7. Courty, N., Arnaud, E.: Inverse Kinematics Using Sequential Monte Carlo Methods. In: Articulated Motion and Deformable Objects. pp. 1–10. Springer Berlin Heidelberg, Berlin, Heidelberg (2008)
8. Duleba, I., Opalka, M.: A comparison of Jacobian-based methods of inverse kinematics for serial robot manipulators. *International Journal of Applied Mathematics and Computer Science* 23(2), 373–382 (2013)
9. Ferreau, H.J., Kirches, C., Potschka, A., Bock, H.G., Diehl, M.: qpOASES: A parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation* 6(4), 327–363 (2014)
10. Grochow, K., Martin, S.L., Hertzmann, A., Popovic, Z.: Style-Based Inverse Kinematics. In: ACM SIGGRAPH 2004 Papers. SIGGRAPH '04, vol. 10, pp. 522–531. Association for Computing Machinery, New York, NY, USA (2004)
11. Nakamura, Y., Hanafusa, H.: Inverse Kinematic Solutions With Singularity Robustness for Robot Manipulator Control. *Journal of Dynamic Systems, Measurement, and Control* 108(3), 163–171 (1986)
12. Schütz, S., Nejadfar, A., Mianowski, K., Vonwirth, P., Berns, K.: CARL – A Compliant Robotic Leg Featuring Mono- and Biarticular Actuation. In: IEEE-RAS International Conference on Humanoid Robots (2017)
13. Vonwirth, P., Nejadfar, A., Mianowski, K., Berns, K.: SLIP-Based Concept of Combined Limb and Body Control of Force-Driven Robots. In: Advances in Service and Industrial Robotics. vol. 84, pp. 547–556. Springer International Publishing, Cham (2020)
14. Wampler, C.: Manipulator Inverse Kinematic Solutions Based on Vector Formulations and Damped Least-Squares Methods. *IEEE Transactions on Systems, Man, and Cybernetics* 16(1), 93–101 (Jan 1986)
15. Wang, L.C., Chen, C.: A combined optimization method for solving the inverse kinematics problems of mechanical manipulators. *IEEE Transactions on Robotics and Automation* 7(4), 489–499 (Aug/1991)
16. Wolovich, W., Elliott, H.: A computational technique for inverse kinematics. In: The 23rd IEEE Conference on Decision and Control. pp. 1359–1363. IEEE, Las Vegas, Nevada, USA (Dec 1984)