A penalty method for constrained Multibody kinematics optimisation using a Levenberg-Marquardt algorithm

Claire Livet, Théo Rouvier, Christophe Sauret, Hélène Pillet, Georges Dumont, Charles Pontonnier

A penalty method for constrained Multibody kinematics optimisation using a Levenberg-Marquardt algorithm

Claire Livet[a] , Théo Rouvier[b], Christophe Sauret[b,c], Hélène Pillet[b], Georges Dumont[a] and Charles Pontonnier[a]

[a]Univ Rennes, Inria, CNRS, IRISA - UMR 6074, France; [b] Institut de Biomécanique Humaine Georges Charpak, Arts et Métiers Institute of Technology, France; [c]Centre d'Études et de Recherche sur l'Appareillage des Handicapés, Institution Nationale des Invalides, France;

ARTICLE HISTORY
Compiled June 16, 2022

Abstract
An alternative method for solving constrained Multibody kinematics optimisation using a penalty method on constraints and a Levenberg-Marquardt algorithm is proposed. It is compared to an optimisation resolution with hard kinematic constraints. These methods are applied to two pairs of experiments and models. The penalty method was at least 20 times faster than the optimisation resolution while keeping similar reconstruction errors and constraints violation. The potential of the method is shown to accurately solve the Multibody kinematics optimisation problem in a reasonable amount of time. A computational gain lies in implementing this resolution with a compiled and optimised program code.

KEYWORDS
Biomechanics, Multibody kinematics optimisation, closed loop, constraints

1. Introduction

In musculoskeletal analysis, Multibody kinematics optimisation permits the estimation of joint coordinates from motion capture data (Begon et al., 2018). In the context of optoelectronic motion capture, it consists in searching for the joint coordinates that minimise the distance between experimental markers and markers of a kinematic model(Duprey et al., 2017).

To solve this problem, different methods have been used: Delp et al. (2007) used a general quadratic programming solver, Muller (2017); Van Den Bogert et al. (2013) used a Levenberg-Marquardt algorithm, Pohl et al. (2021); Serrien et al. (2020)employed a Bayesian approach, Bonnet et al. (2017); De Groote et al. (2008) used a Kalman approach, Apkarian et al. (1989) directly identified angles in rotation matrix, Ayusawa and Nakamura (2012) used a Levenberg-Marquardt algorithm, a quasi-Newton algorithm, and a conjugate gradient method and Fohanno et al. (2014)used a feasible sequential quadratic algorithm and a Kalman filtering technique.

According to Hybois et al. (2019); Laitenberger et al. (2014), kinematic constraints applied to an osteoarticular model lead to a better estimation of joint angles than a non-constrained model. Moreover, Pizzolato et al. (2017) underlined the importance of having accurate joint angles estimation for real-time applications. In clinical field, Barrios et al. (2010) reported that real-time feedback

CONTACT Claire Livet. Email: claire.livet@irisa.fr

is already used for gait retraining for instance. It enables the patient to have a feedback about their posture and then correct it if needed.

The challenge is then to take these kinematic constraints into account in real time so as to obtain a correct estimation of joint angles for real-time applications. Few of the proposed methods in the literature allow the kinematic constraints of the human body, such as joint limits or kinematic constraints (shoulder, forearm, ...), to be taken into account in a reasonable computation time (Pizzolato et al., 2017; Sherman et al., 2011). Andersen et al. (2009) solved the Multibody kinematics optimisation problem with hard kinematic constraints without evaluating computation time. Fohanno et al. (2014) proposed a study that considers a closed-loop system (human body with kayak equipment) whose constraints are included in the cost function using a penalty method, with an opening tolerance of 2 mm and compared the performances (e.g. computation time, marker reconstruction error) of two resolution methods. They concluded that the Kalman filter is faster than the feasible sequential quadratic algorithm and well-suited for cycling movements.

The current study aims at comparing two methods taking into account anatomical kinematic constraints during Multibody kinematics optimisation on any type of movement and evaluating their performance in term of accuracy and computation time. We propose to reformulate the hard constraint problem with a penalty method in order to save computation time while keeping accurate joint angle estimation. To quantify the differences in performances (computation time, markers tracking, constraints and joint limits violation) between the solutions of the hard constraint problem and the penalty problem, we tested the methods with different osteoarticular models, for throwing and range of motion trials. These trials and models imposed various kinematic constraints in the Multibody kinematics optimisation problem. Firstly the two methods are described, secondly the models and experimental data are developed, and finally, a comparison on different performance criteria is made.

2. Materials and Methods

2.1. Multibody kinematics optimisation problem

The Multibody kinematics optimisation problem consisted in finding the vector of generalised coordinates $\mathbf{q} \in Q \subset \mathbb{R}^n$ that enabled the markers from the model $\mathbf{x}_{mod}(\mathbf{q}) \in \mathbb{R}^{3m}$ to match as accurately as possible the experimental markers $\mathbf{x}_{exp} \in \mathbb{R}^{3m}$. n and m are the number of generalised coordinates and the number of markers.

We denote $\mathbf{x}^i_{mod}(\mathbf{q}) \in \mathbf{R}^3$, respectively $\mathbf{x}^i_{exp} \in \mathbb{R}^3$, as the 3-D position of the model marker i, respectively the 3-D position of the experimental marker i.

The vector of joint coordinates $\mathbf{q}$ has to lie in the range of motion Q allowed by the model and could be constrained by scleronomic[1] kinematics constraints $\mathbf{c}(\mathbf{q}) \in \mathbb{R}^{n_c}$ (with n_c the number of kinematic constraints). These constraints can be of different natures, such as a closed loop (e.g. shoulder complex) or kinematical dependencies (e.g. scapula sliding).

Optimisation routine

For each motion capture time frame, the problem to solve was:

$$\begin{aligned} &\min_{\mathbf{q}\in Q} \sum_{i=1}^{m} ||\mathbf{x}^i_{exp} - \mathbf{x}^i_{mod}(\mathbf{q})||^2 \\ &\text{such that } \mathbf{c}(\mathbf{q}) = \mathbf{0}. \end{aligned} \tag{1}$$

The value $\mathbf{c(q)}$ can be split with implicit constraints $\mathbf{h}(\mathbf{q}) \in \mathbb{R}^{n_h}$ (with n_h is the number of implicit constraints) and explicit constraints $\mathbf{f}(\tilde{\mathbf{q}}) \in \mathbb{R}^n$, with $\tilde{\mathbf{q}}$

[1]The constraint is only a function of joint positions and not of time.

$\in \tilde{\mathbf{Q}} \subset \mathbb{R}^{\tilde{n}}$ a subset of $\mathbf{q}$:

$$\mathbf{c}(\mathbf{q}) = \begin{cases} \mathbf{h}(\mathbf{q}) = \mathbf{0} \\ \mathbf{q} = \mathbf{f}(\tilde{\mathbf{q}}) \end{cases} \tag{2}$$

For constraints initially written as $\mathbf{h}(\mathbf{q})$, one can try to use for instance a partitioning method (Haug et al., 1999) to obtain explicit equations. Depending on the complexity of $\mathbf{h}(\mathbf{q})$, there is no systematic method to obtain explicit equations and it can be impossible to rewrite implicit equations as explicit equations.

As $\tilde{\mathbf{q}}$ is a subset of $\mathbf{q}$, by replacing $\tilde{\mathbf{q}}$ by $\mathbf{f}(\tilde{\mathbf{q}})$, the number of unknowns in the Multibody kinematics optimisation can be reduced and so it could reduce computation time during resolution. The problem was then defined as:

$$\begin{aligned} &\min_{\tilde{\mathbf{q}} \in \tilde{Q}} \sum_{i=1}^{m} ||\mathbf{x}^i_{exp} - \mathbf{x}^i_{mod}(\mathbf{f}(\tilde{\mathbf{q}}))||^2 \\ &\text{such that } \mathbf{h}(\mathbf{f}(\tilde{\mathbf{q}})) = \mathbf{0}. \end{aligned} \tag{3}$$

This optimisation problem was constrained by $\mathbf{h}(\mathbf{f}(\tilde{\mathbf{q}}))$ and its solution had to lie in the bounds defined by the range of motion $\tilde{Q}$. It was solved with the interior-point algorithm as described by Byrd et al. (2000, 1999); Waltz et al. (2006), using a tolerance of 1 10^{-6} on the norm of $\mathbf{h}(\mathbf{f}(\tilde{\mathbf{q}}))$, and on the change of the cost function. In this paper, this resolution was called optimisation routine (OR).

Interior-point resolution uses at each iteration one of two methods. First the Hessian associated to the Lagrangian is checked to be positive-definite. If so, a Newton step as described by Bonnans et al. (2006) is used. In the opposite case, the algorithm backups to a conjugate gradient step. After one of these steps is done, the Lagrange multipliers associated to the constraints are updated, using a predictor-corrector algorithm on a barrier function. New iterations (step and Lagrange multipliers update) are done until the change in the function cost is less than the specified tolerance.

Iterative resolution

In order to relax the joint limits constraints, a bound penalisation function was needed. In this study, the scalar function g associated to this bound penalisation was expressed as, for $\tilde{q}^k$ the k^{th} element of $\tilde{\mathbf{q}}$:

$$g(\tilde{q}^k) = \begin{cases} 0 \quad \text{if} \quad \tilde{q}^k \in [\tilde{q}^k_{min}, \tilde{q}^k_{max}] \\ (\tilde{q}^k \tilde{q}^k_{min})^2 \quad \text{if} \quad \tilde{q}^k < \tilde{q}^k_{min} \\ (\tilde{q}^k - \tilde{q}^k_{max})^2 \quad \text{if} \quad \tilde{q}^k > \tilde{q}^k_{max} \end{cases}$$

[Figure 1 about here.]

As it can be seen in FIGURE 1, the $\tilde{q}^k$ were not penalised when they lied in bounds but they were penalised out of bounds. With relaxing the joint limit constraints and the kinematic constraints, the problem to be solved consisted in finding the $\tilde{\mathbf{q}}$ that minimised the following sum:

$$\min_{\tilde{\mathbf{q}}} \left(\sum_{i=1}^{m} ||\mathbf{x}^i_{exp} - \mathbf{x}^i_{mod}(\mathbf{f}(\tilde{\mathbf{q}}))||^2 + \sum_{j=1}^{n_h} ||\gamma h^j(\mathbf{f}(\tilde{\mathbf{q}}))||^2 + \sum_{k=1}^{n} ||\zeta g(\tilde{q}^k)||^2 \right) \tag{4}$$

$\gamma \in \mathbb{R}$ and $\zeta \in \mathbb{R}$ are the penalty factors associated to constraints equations and bound penalisation. They allow to balance the weight for each term of the sum.

Using vector notation as follows:

$$\begin{cases} \mathbf{z} = \begin{pmatrix} \mathbf{x}_{exp} \\ 0 \\ 0 \end{pmatrix} \\ \mathbf{a}(\tilde{\mathbf{q}}) = \begin{pmatrix} \mathbf{x}_{mod}(\mathbf{f}(\tilde{\mathbf{q}})) \\ \gamma \mathbf{h}(\mathbf{f}(\tilde{\mathbf{q}})) \\ \zeta \mathbf{g}(\tilde{\mathbf{q}}) \end{pmatrix} \end{cases} \tag{5}$$

The sum to minimize finally consisted in:

$$\min_{\tilde{\mathbf{q}}} \sum_{i=1}^{m+n_h+n} ||\mathbf{z}^i - \mathbf{a}^i(\tilde{\mathbf{q}})||^2 \tag{6}$$

This optimisation problem was solved using the Levenberg-Marquardt iterative algorithm described by Levenberg (1944); Marquardt (1963); Moré (1978). Respecting the constraints was considered as more important than lying in the joint limits. That is why γ and ζ were respectively set, in the present work, to 150 and 20 after manual tuning. Moreover, these coefficients provided the same constraint tolerance as the interior-point algorithm. Tuning parameters is achieved by first choosing the desired threshold for the norm of kinematic constraints violation and the norm of bound penalisation. Kinematic constraints violation and bound penalisation are then evaluated on a trial using initial values of γ and ζ. Using the results of this trial with initial values, the penalty factors can be tuned higher or lower to adjust the norm of constraints violation and the norm of bound penalisation. With the new values of γ and ζ, kinematic constraints violation and bound penalisation are re-evaluated and compared to their desired accuracy. The cost function variation stopping the algorithm was set to 1 10^{-6}. This resolution is called iterative resolution (IR) in the following paragraphs.

The Levenberg-Marquardt method mixes a Gauss-Newton approach (Bonnans et al. (2006)) with a damping term reducing the numerical issues that appear with ill-conditioned Jacobian matrix.

2.2. Models

In order to compare the methods, two pairs of model and experimental trials were used. One pair implied only joints limits: open loop model and range of motion trials. The other pair implied the mobilisation of closed loop constraints at the forearm and the shoulder: closed loop model and throwing trials.

Open loop model

The open loop model was a full body model of 20 solids and 47 generalised coordinates $\mathbf{q}(=\tilde{\mathbf{q}})$ (Figure 2). It was based on Klein Horsman et al. (2007) for lower limbs and Holzbaur et al. (2005) for upper limbs. For the lower limb, there were feet, tibiae, femurs and pelvis forming 12 generalised coordinates. The upper limb was composed of an abdomen, a thorax, a skull, two clavicles, two scapulae, two humeri, two radii and two hands, forming 35 generalised coordinates.

[Figure 2 about here.]

Closed loop model

The closed loop model was an upper limb model of 17 solids and 67 generalised coordinates $\mathbf{q}$ (Figure 3). The forearm model was taken from Pennestrì et al. (2007) and the shoulder complex model was taken from Seth et al. (2016). The forearm model was composed of a humerus, a radius and ulna while the shoulder complex was composed of a clavicle, a scapula and a thorax.

The shoulder model added other scleronomic constraints than only closed loop ones : each scapula was constrained to slide on a thorax ellipsoid. The scapula was defined with 7 generalised coordinates, with 3 translation coordinates as a function of 4 rotation coordinates. Then $\tilde{\mathbf{q}}$ is composed of only 61 generalised coordinates, excluding the positions coordinates of the scapulae. Three prismatic joints were used to define this sliding of each scapula. One prismatic joint per arm was also used between radius and ulna.

[Figure 3 about here.]

[Figure 4 about here.]

The kinematic constraints were expressed from a closed loop modeled by a connectivity graph Featherstone (2008), as shown in Figure 4. The loop was open between one solid p and its clone, named shadow s Samin and Fisette (2003). Constraints equations closed the loop by overlaying the frames of the solid and its shadow. This led to six equations (orientation and position). We defined two paths that started from the closest common ancestor and joined the solid and its clone. Both position and orientation of the solid p and of its shadow s were reconstructed following both paths respectively. Then overlaying constraints were expressed as Postiau (2004):

$$\begin{cases} \mathbf{x}_s(\mathbf{q}) - \mathbf{x}_p(\mathbf{q}) = \mathbf{0} \\ \mathbf{R}_s^T(\mathbf{q})\mathbf{R}_p(\mathbf{q}) - I_3 = \mathbf{0} \end{cases} \tag{7}$$

where $\mathbf{R}_j$ is the rotation matrix from closest common ancestor to solid j and $\mathbf{x}_j$ is the position of the solid j in the closest common ancestor frame.

These vector equations gave 12 scalar equations. To reduce the number of equations, as we know $\mathbf{R}^{loop}(q) = \mathbf{R}_s^T(q)\mathbf{R}_p(q)$ is a rotation matrix, we identified the diagonal of $\mathbf{R}^{loop}$ to ones. Indeed, the $\mathbf{R}^{loop}$ is a rotation matrix as a product of rotation matrices,so its columns are orthogonal and their norm are equal to one (rotation matrix properties (Jennings, 2020)). So if the diagonal of $\mathbf{R}^{loop}$ is equal to one, then the rest of $\mathbf{R}^{loop}$ is automatically filled with zeros to keep orthogonality properties. Thanks to this technique, only six equations lasted.

For the closed loop model, closing the loops led to 24 scleronomic equations to respect (6 equations per loop : 2 loops for forearms, 2 loops for shoulders).

Both open loop and closed loop models were adjusted with geometrical parameters were taken from Dumas et al. (2007); Pennestrì et al. (2007); Seth et al. (2016) and then scaled to subjects' size and mass using the CusToM scaling routine Muller et al. (2019); Puchaud et al. (2020) based on motion capture data.

2.3. Experimental data

Range of motion trials

Five female and eleven male subjects (height: 178 ± 7.5 cm, mass: 69.7 ± 10 kg) were asked to mirror the movements of the experimenter. The movements consisted in the sequential actuation of all body joints one by one along their full range of motions (ROM). Their movements were captured using the optoelectronic system Qualisys (200Hz, composed of 22 12 Mpixels cameras). These experiments were approved by a National Ethics Commitee (Comité Opérationnel d'Evaluation des Risques Légaux et Ethiques, 2021-06).

Throwing trials

Overhead throwing trials (THW) raw data were taken from another study Cruz Ruiz et al. (2017) for five other subjects who were asked to throw a ball to a static target at different distances. Eighteen trials per subject were used for this study.

2.4. Methods comparison and performance criteria

Both methods were implemented in CusToM (Muller et al., 2019), a Matlab toolbox for musculoskeletal analysis. Data processing were made on Intel(R) Core(TM) i9-10900 CPU @2.80GHz with a 32.0GB RAM, on a Windows 64bits operating system. Matlab version was R2021a.

The two methods were compared through these metrics:

- Computation frequency (Hz): inverse of the time needed to process the inverse kinematic step of one time frame (unit: Hz)
- Reconstruction error (cm): root mean square error between experimental and model markers
- Constraints violation (position : cm/ rotation : no unit): norm of position constraints and rotation constraints from (7)
- Limits violation (prismatic joint : cm/ revolute joint : °): root of the bound penalisation $p(\tilde{q})$. A limits violation of 0 means that the joint coordinate lied in its joint limits.
- Joint coordinates difference (cm or °): root mean square error between joint coordinates resulting from optimisation routine and those from iterative resolution

In order to determine if the joints coordinates difference was significant or not, we compared it to the noise from the motion capture. Experiments trials used in this study were captured with an accuracy lower than 1 mm, however the motion capture uncertainty was set to 1 mm, as proposed by Chèze (2014) to estimate motion capture noise. As a coarse approximation, we moved one marker from the hand and one marker from the foot of 1mm in every direction. The maximum absolute angle error on the pelvis joint was about 0.1 rad (5.7 °) and the maximum absolute error on the 6 d.o.f's prismatic joint was about 0.1mm. We used these thresholds to determine whether the error between OR and IR was significant or not.

[Figure 5 about here.]

[Figure 6 about here.]

3. Results

Results for computation frequency, reconstruction error, constraints violation and limits violation are gathered in Tables 1 and 2. As there was no constraints nor prismatic joint in the open loop model, there is no value associated to constraints violation and limits violation for prismatic joints in the ROM trials row. This is indicated by N.A. for "non applicable".

Reconstruction error

For every subject and for both models and trials, the mean reconstruction error was under 2 cm. The 0.4 cm difference between throwing trials and ROM trials can be explained by the models difference. Indeed, when the joint coordinates must both follow the markers while respecting the kinematic constraints, it can lead to a poorer tracking of the markers, depending on the ability of the model to properly mimic the motion.

The closed loop model had kinematic constraints that prevented from precisely following the markers of the upper limb.

For the ROM trials, the mean reconstruction error was 1.12 cm (optimisation routine) for the lower limb among the sixteen subjects, whose models were calibrated with motion capture data. For throwing trials, the mean reconstruction error among all subjects was 1.19 cm (optimisation routine).

Although the reconstruction errors from our methods were higher than those from the literature, it should be noted that both methods gave similar reconstruction errors. Indeed, the main goal here was to verify that the results of the two methods were equivalent, which is the case for the reconstruction error.

More precisely, the reconstruction error from the iterative resolution was systematically lower than the one from the optimisation routine. Since the constraints were relaxed in the iterative resolution, the algorithm could find a lower minimum in the reconstruction error. Although the minimum was lower from

one method to another, the reconstruction errors were not substantially different (0.01 cm difference between OR and IR).

Constraints violation

Kinematic constraints were present only in the closed loop model. Optimisation routine results were under $1\ 10^{-7}$ constraints violation, in accordance with the tolerance of $h(f(\tilde{q}))$ in the optimisation routine of $1\ 10^{-6}$. The results from the iterative resolution were under $5\ 10^{-7}$ constraints violation, so they were under the threshold of the optimisation routine. One can then consider that both resolutions equivalently respected kinematic constraints.

Limits violation

As the optimisation routines strictly respected the joint coordinates limits, the limits violation systematically equalled zero. For all the trials, the limits of revolute joints were at their worst exceeded by 2 °for IR algorithm. As it was less than half lower than the threshold of the motion capture noise (5.7 °), this violation can be considered negligible compared to the overall accuracy of the capture.

Prismatic joint limits were never exceeded for all the throwing trials and all the subjects. This means that scapulae stayed in the defined range near the thorax.

Joint coordinates difference

Figure 5 shows joint angles at different joints (one distal joint, one less distal joint and two proximal joints) of the body for a ROM trial while Figure 6 shows joint angles at the upper arm for a throwing trial. The time when the relative difference between the two solutions was higher than the noise (0.1 rad/5.7 ° or 0.1 mm) is underlined by a light straight line. When this difference was due to a limit violation in the iterative resolution, it is underlined by a dark straight line.

For each joint and each trial, we computed the time when the difference between the solutions was higher than motion capture noise. This time was divided by the total time of each trial to obtain a percentage for every trial and every subject. Each point in Figure 7 is associated to a joint and a trial. In Figure 7, the majority of joint results difference were higher than motion capture noise during less than 20% of the time of the trial, as it can be seen with the median in gray vertical line. Other joint results were above motion capture noise during 20% to 100% of the time of the trial, evenly scattered in this range.

[Figure 7 about here.]

For each joint coordinate, each trial and each subject, the root mean square error was computed between the solution from the optimisation routine and the iterative resolution. In Figure 8, each joint coordinate result is categorised :

- if the joint belongs to a loop, it is added in the category "In loop", otherwise it is added in the category "Not in loop"
- if the root mean square error is below motion capture noise, it is added to the category "Below measurement noise", otherwise it is added in the category "Above measurement noise"
- if the joint is a revolute joint (respectively a prismatic joint), it is added to the category "Revolute joints" (respectively "Prismatic joints")

As it can be seen in Figure 8, there was a significant difference between joints in or not in loop for being above measurement noise: there was approximately 4 times more joints in loop than joints not in loop above the measurement noise. It can be noticed that there were more joints in loop than joints not in loop. Still, 24% of the joints in loop were above the measurement noise while 12% of the joints not in loop were above the measurement noise. One can deduce that the majority of the data in Figure 7 above 20% were due to joints in loops.

Figure 8 shows also that the large majority of angles were below the measurement noise and that the prismatic joints solutions were more likely to be different above the measurement noise.

[Figure 8 about here.]

The same analysis can be done for ROM trials: mean results difference was higher than motion capture noise less than 6% of the time of the trial. Maximum 1% of the time was due to joint limit violation, as it can be seen in FIGURE 9.

[Figure 9 about here.]

As it can be seen when comparing FIGURES 7 and 9, the mean percentage of time when the solutions were above the motion capture noise are different : less than 20% for THW trials and less than 4% for ROM trials.

[Table 1 about here.]

[Table 2 about here.]

Computational gain versus reconstruction error precision

The optimisation routine was about 20 times slower than the iterative resolution: 31.6 10^{-2} Hz (OR-THW) vs 6.79 Hz (IR-THW). In both cases, the upper body model model took more time to solve than the full body model, because of a higher number of generalized coordinates and the presence of constraints in the upper body model. Fohanno et al. (2014) reported a frequency of 15 Hz for the use of a Kalman filter on a penalty problem and Ayusawa and Nakamura (2012) reported a frequency of 17 Hz with a quasi-Newton method with a decomposed gradient computation.

The two resolutions were compared with a tolerance of 1 10^{-6} on the variation on the cost function. For the iterative resolution, one could change this tolerance to gain some computational time. This is what is displayed in FIGURE 10: for one subject and each throwing trial the deviation to the optimal reconstruction error -the difference between the reconstruction error at the set tolerance and the reconstruction error at a tolerance of 1 10^{-6} is plotted as a function of the computation frequency. Among all the points, the maximum mean constraint error was under 4 10^{-4} cm. By changing the tolerance to 1 10^{-5}, the computation time was divided by 4 while loosing 0.1 cm of precision in the reconstruction error and keeping acceptable constraint violation (under 1 10^{-3} cm). Changing the tolerance can allow the method to reach the frequencies of the literature.

[Figure 10 about here.]

For the ROM trial, the change function tolerance can be adjusted to 0.5 while having the loss in precision in the reconstruction error under the tolerance of 1 mm (FIGURE 11). The frequency associated to this function tolerance is 57 Hz.

[Figure 11 about here.]

4. Discussion

Reconstruction error

The mean reconstruction errors were under 2 cm, which lied in the literature range reported by Begon et al. (2018) (from 4 mm to 40 mm). More precisely, Puchaud et al. (2020) reported mean reconstruction errors in literature under 1 cm for the lower limb (twenty-six subjects, calibration based on imagery data) while Muller et al. (2015) indicated a maximum reconstruction error for a full-body model of 1.3 cm (one subject, calibration based on motion capture data). Seth et al. (2016) reported a maximum mean reconstruction error for a closed loop shoulder model of 0.19 cm for a flexion shoulder trial (twelve subjects, calibration based on bone-pin motion capture data) while Laitenberger et al. (2014) reported a mean reconstruction error under 0.5 cm for a flexion forearm trial (fifteen subjects, calibration based on motion capture data). Among all the ROM trials, four subjects had a mean marker error reconstruction above 3 cm at the sternum. One of them was a man with a non-standard body mass index and the others were women. The particularities of these subjects were not well represented in the

generic model that we used, based on data representing the 50th percentile male from the U.S. army personnel Gordon et al. (1988); Holzbaur et al. (2005). For the throwing trials, the ellipsoid of the scapulothoracic joint model used Seth et al. (2016) was not calibrated for each subject, which could explain the difference between our reconstruction error and the one from literature.

A poor kinematic reconstruction directly impacts the inverse dynamics step and the muscle recruitment problem. Indeed, Muller (2017) showed that non-zero dynamic residuals were mostly due to kinematic uncertainties. These uncertainties come from both the motion capture and the assumptions of the model, namely idealised pin joints and rigidity of body segments, neglect of soft-tissue artefact interacting with marker positions during trials Faber et al. (2018) is a source of uncertainties.

Towards real-time

Kannape and Blanke (2013) defined the maximum time delay before the feedback to the user loses its relevance at 75 ms. van der Kruk and Reijne (2018) reported that for sports, motion capture varies between 50 and 250 Hz. For a fast movement such as downhill skiing, it is necessary to capture the movement in high frequency to avoid losing information. All captured instants must therefore be processed by the Multibody kinematics optimisation problem. This means that the proposed penalty method cannot process an input stream in real time even at 50 Hz. Furthermore, the computation time has to be increased for taking into account the latency of the motion capture, the labelling process and the data transfer time from the capture system to the computational code, to have a delay under 75 ms. A possible computational gain could be achieved by implementing this resolution with a compiled and optimised program code, which is not the case of this study.

On the reliability of hard constraints

As the iterative resolution was allowed to go beyond the joints limit, it gave a different solution for some joints within the closed loops, like scapula elevation-depression in Figure 6. On the same motion capture, it seems more anatomically correct that the joint articulation went a little bit beyond its joint limit than shifting from 18 ° in less than 0.06 s (scapula elevation-depression, at approximately 0 seconds). This example raises an issue about keeping hard joint limits and kinematic constraints that can lead to wrong joint coordinates. The Figure 7 showed that the majority of joint results difference were higher than motion capture noise during less than 20% of the time of the trial. This may be due to the joints within the loops: as they must stay within joints limits and the loop those two constraints can lead to very different solutions, as for the scapula elevation in Figure 6.

This analysis is confirmed by the comparison between THW and ROM trials : the differences in OR and IR resolutions were higher in THW trials than in ROM trials. This difference is due to the presence of kinematic constraints in the model used for THW motion. Indeed, in the ROM trials there was only limits violation to handle while in the THW trials, kinematic constraints and joints limits had to be respected at the same time. Respecting only joints limits gave similar results between IR and OR while adding kinematic constraints led to different results for the two strategies.

Perspectives

As highlighted by Cerveri et al. (2005), the non-ideality of human joints can lead to errors in the estimation of kinematical motion. One can then model human joints with clearance joints (Flores and Ambrósio, 2004), as used by Quental et al. (2016) for the gleno-humeral joint. We could consider clearance joints in the iterative resolution, by using a 6-dof joint between two bodies and then by choosing the appropriate penalty factor associated to the constraint of motion in the 6-dof joint. We used a Levenberg-Marquardt method to solve the iterative resolution. Other methods could be explored : extended Kalman filter, which is known to work on different type of motion (e.g. cyclic motions Fohanno et al. (2014) or fencing motion Cerveri et al. (2003)), or classical descent algorithms

(Quasi-Newton method, Conjugate Gradient method) (Ayusawa and Nakamura, 2012). It is also important to assure the kinematic consistency of the velocities and accelerations resulting from the derivatives of the generalised coordinates of a constrained system (Alonso et al., 2010; Moissenet et al., 2012; Silva and Ambrósio, 2002). Future studies should strive to achieve this kinematic consistency using time efficient computation methods.

Conclusion

We compared two Multibody kinematics optimisation methods: a first one using a hard-constraint solving method and a second one using a penalty method. These resolutions were compared through experimental data and two models (full body (open loop) and upper limb with closed loops).

For open loop models, the performances (reconstruction error, limit violation, constraints violation) and the joints coordinates results were equivalent most of the time for both resolutions. For closed loop models, the performances were equivalent but some of the joints coordinates results showed different behaviours. Some sensitivity study on the penalty factors could be made in order to understand their impact on the solutions and give a clear guideline on how to choose them.

Finally, the iterative resolution was 20 times faster than the optimisation routine, which can be interesting in the perspective of real-time Multibody kinematics optimisation applications.

Disclosure statement

The authors declare no conflict of interest.

Funding

This study was conducted as part of the CapaCITIES project, funded by the French National Research Agency (ANR-19-CE19-0007).

References

Alonso, F., Cuadrado, J., Lugrís, U., and Pintado, P. (2010). A compact smoothing-differentiation and projection approach for the kinematic data consistency of biomechanical systems. *Multibody System Dynamics*, (24):67–80.

Andersen, M. S., Damsgaard, M., and Rasmussen, J. (2009). Kinematic analysis of over-determinate biomechanical systems. *Computer Methods in Biomechanics and Biomedical Engineering*, 12(4):371–384.

Apkarian, J., Naumann, S., and Cairns, B. (1989). A three-dimensional kinematic and dynamic model of the lower limb. *Journal of Biomechanics*, 22(2):143–155.

Ayusawa, K. and Nakamura, Y. (2012). Fast inverse kinematics algorithm for large DOF system with decomposed gradient computation based on recursive formulation of equilibrium. *IEEE International Conference on Intelligent Robots and Systems*, pages 3447–3452.

Barrios, J. A., Crossley, K. M., and Davis, I. S. (2010). Gait retraining to reduce the knee adduction moment through real-time visual feedback of dynamic knee alignment. *Journal of Biomechanics*, 43(11):2208–2213.

Begon, M., Andersen, M. S., and Dumas, R. (2018). Multibody Kinematics Optimization for the Estimation of Upper and Lower Limb Human Joint Kinematics: A Systematized Methodological Review. *Journal of Biomechanical Engineering*, 140(3).

Bonnans, J. F., Gilbert, J. C., Lemaréchal, C., and Sagastizábal, C. A. (2006). Numerical optimization: Theoretical and practical aspects. *Numerical Optimization: Theoretical and Practical Aspects*, pages 1–494.

Bonnet, V., Richard, V., Camomilla, V., Venture, G., Cappozzo, A., and Dumas, R. (2017). Joint kinematics estimation using a multi-body kinematics optimisation and an extended Kalman filter, and embedding a soft tissue artefact model. *Journal of Biomechanics*, 62:148–155.

Byrd, R. H., Gilbert, J. C., and Nocedal, J. (2000). A trust region method based on interior point techniques for nonlinear programming. *Mathematical Programming, Series B*, 89(1):149–185.

Byrd, R. H., Hribar, M. E., and Nocedal, J. (1999). An interior point algorithm for large-scale nonlinear programming. *SIAM Journal on Optimization*, 9(4):877–900.

Cerveri, P., Pedotti, A., and Ferrigno, G. (2003). Robust recovery of human motion from video using Kalman filters and virtual humans. *Human Movement Science*, 22(3):377–404.

Cerveri, P., Pedotti, A., and Ferrigno, G. (2005). Kinematical models to reduce the effect of skin artifacts on marker-based human motion estimation. *Journal of Biomechanics*, 38(11):2228–2236.

Chèze, L. (2014). Biomécanique du mouvement et modélisation musculo-squelettique. *Techniques De L'Ingénieur*, 33(0).

Cruz Ruiz, A. L., Pontonnier, C., and Dumont, G. (2017). Low-Dimensional Motor Control Representations in Throwing Motions. *Applied Bionics and Biomechanics*, 2017.

De Groote, F., De Laet, T., Jonkers, I., and De Schutter, J. (2008). Kalman smoothing improves the estimation of joint kinematics and kinetics in marker-based human gait analysis. *Journal of Biomechanics*, 41(16):3390–3398.

Delp, S. L., Anderson, F. C., Arnold, A. S., Loan, P., Habib, A., John, C. T., Guendelman, E., and Thelen, D. G. (2007). OpenSim: Open-source software to create and analyze dynamic simulations of movement. *IEEE Transactions on Biomedical Engineering*, 54(11):1940–1950.

Dumas, R., Chèze, L., and Verriest, J. P. (2007). Adjustments to McConville et al. and Young et al. body segment inertial parameters. *Journal of Biomechanics*, 40(3):543–553.

Duprey, S., Naaim, A., Moissenet, F., Begon, M., and Chèze, L. (2017). Kinematic models of the upper limb joints for multibody kinematics optimisation: An overview. *Journal of Biomechanics*, 62:87–94.

Faber, H., Van Soest, A. J., and Kistemaker, D. A. (2018). Inverse dynamics of mechanical multibody systems: An improved algorithm that ensures consistency between kinematics and external forces. *PLoS ONE*, 13(9):1–16.

Featherstone, R. (2008). *Rigid Body Dynamics Algorithms*. Springer US.

Flores, P. and Ambrósio, J. (2004). Revolute joints with clearance in multibody systems. *Computers and Structures*, 82(17-19):1359–1369.

Fohanno, V., Begon, M., Lacouture, P., and Colloud, F. (2014). Estimating joint kinematics of a whole body chain model with closed-loop constraints. *Multibody System Dynamics*, 31(4):433–449.

Gordon, C. C., Churchill, T., Clauser, C. E., Bradtmiller, B., Mcconville, J. T., Tebbetts, I., and Walker, R. A. (1988). Anthropometric Survey of U.S. Army Personnel: Summary Statistics, Interim Report for 1988. Technical report.

Haug, E. J., Negrut, D., and Engstler, C. (1999). Implicit Runge-Kutta integration of the equations of multibody dynamics in descriptor form. *Mechanics of Structures and Machines*, 27(3):337–364.

Holzbaur, K. R., Murray, W. M., and Delp, S. L. (2005). A model of the upper extremity for simulating musculoskeletal surgery and analyzing neuromuscular control. *Annals of Biomedical Engineering*, 33(6):829–840.

Hybois, S., Puchaud, P., Bourgain, M., Lombart, A., Bascou, J., Lavaste, F., Fodé, P., Pillet, H., and Sauret, C. (2019). Comparison of shoulder kinematic chain models and their influence on kinematics and kinetics in the study of manual wheelchair propulsion. *Medical Engineering and Physics*, 69:153–160.

Jennings, A. (2020). Rigid body mechanics. *Structures*, pages 17–85.

Kannape, O. A. and Blanke, O. (2013). Self in motion: Sensorimotor and cognitive mechanisms in gait agency. *Journal of Neurophysiology*, 110(8):1837–1847.

Klein Horsman, M. D., Koopman, H. F., van der Helm, F. C., Prosé, L. P., and Veeger, H. E. (2007). Morphological muscle and joint parameters for musculoskeletal modelling of the lower extremity. *Clinical Biomechanics*, 22(2):239–247.

Laitenberger, M., Raison, M., Périé, D., and Begon, M. (2014). Refinement of the upper limb joint kinematics and dynamics using a subject-specific closed-loop forearm model. *Multibody System Dynamics*, 33(4):413–438.

Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 2(2):164–168.

Marquardt, D. W. (1963). An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441.

Moissenet, F., Chèze, L., and Dumas, R. (2012). Anatomical kinematic constraints: Consequences on musculo-tendon forces and joint reactions. *Multibody System Dynamics*, 28(1-2):125–141.

Moré, J. J. (1978). The Levenberg-Marquardt algorithm: Implementation and theory. In *Numerical Analysis*, pages 105–116, Berlin, Heidelberg. Springer Berlin Heidelberg.

Muller, A. (2017). *Contributions méthodologiques à l'analyse musculo-squelettique de l'humain dans l'objectif d'un compromis précision performance*. PhD thesis, École normale supérieure de Rennes.

Muller, A., Germain, C., Pontonnier, C., and Dumont, G. (2015). A Simple Method to Calibrate Kinematical Invariants: Application to Overhead Throwing. In *33th International Society of Biomechanics in Sports (ISBS)*, pages 78–81.

Muller, A., Pontonnier, C., Puchaud, P., and Dumont, G. (2019). CusToM: a Matlab toolbox for musculoskeletal simulation. *Journal of Open Source Software*, 4(33):927.

Pennestrì, E., Stefanelli, R., Valentini, P. P., and Vita, L. (2007). Virtual musculo-skeletal model for the biomechanical analysis of the upper limb. *Journal of Biomechanics*, 40(6):1350–1361.

Pizzolato, C., Reggiani, M., Modenese, L., and Lloyd, D. G. (2017). Real-time inverse kinematics and inverse dynamics for lower limb applications using OpenSim. *Computer Methods in Biomechanics and Biomedical Engineering*, 20(4):436–445.

Pohl, A. J., Schofield, M. R., and Ferber, R. (2021). Comparing the performance of Bayesian and least-squares approaches for inverse kinematics problems. *Journal of Biomechanics*, 126(July):110597.

Postiau, T. (2004). *Génération et parallélisation des équations du mouvement de systèmes multicorps par l'approche symbolique*. PhD thesis, UCL - FSA/MECA - Département de mécanique.

Puchaud, P., Sauret, C., Muller, A., Bideau, N., Dumont, G., Pillet, H., and Pontonnier, C. (2020). Accuracy and kinematics consistency of marker-based scaling approaches on a lower limb model: a comparative study with imagery data. *Computer Methods in Biomechanics and Biomedical Engineering*, 23(3):114–125.

Quental, C., Folgado, J., Ambrósio, J., and Monteiro, J. (2016). A new shoulder model with a biologically inspired glenohumeral joint. *Medical Engineering and Physics*, 38(9):969–977.

Samin, J.-C. and Fisette, P. (2003). *Symbolic Modeling of Multibody Systems*. Solid Mechanics and Its Applications. Springer Netherlands, Dordrecht.

Serrien, B., Pataky, T., Baeyens, J. P., and Cattrysse, E. (2020). Bayesian vs. least-squares inverse kinematics: Simulation experiments with models of 3D rigid body motion and 2D models including soft-tissue artefacts. *Journal of Biomechanics*, 109:109902.

Seth, A., Matias, R., Veloso, A. P., and Delp, S. L. (2016). A biomechanical model of the scapulothoracic joint to accurately capture scapular kinematics during shoulder movements. *PLoS ONE*, 11(1).

Sherman, M. A., Seth, A., and Delp, S. L. (2011). Simbody: multibody dynamics for biomedical research. *Procedia IUTAM*, 2:241–261. IUTAM Symposium on Human Body Dynamics.

Silva, M. P. and Ambrósio, J. A. (2002). Kinematic data consistency in the inverse dynamic analysis of biomechanical systems. *Multibody System Dynamics*, 8(2):219–239.

Van Den Bogert, A. J., Geijtenbeek, T., Even-Zohar, O., Steenbrink, F., and Hardin, E. C. (2013). A real-time system for biomechanical analysis of human movement and muscle function. *Medical and Biological Engineering and Computing*, 51(10):1069–1077.

van der Kruk, E. and Reijne, M. M. (2018). Accuracy of human motion capture systems for sport applications; state-of-the-art review.

Waltz, R. A., Morales, J. L., Nocedal, J., and Orban, D. (2006). An interior algorithm for nonlinear optimization that combines line search and trust region steps. *Mathematical Programming*, 107(3):391–408.

Table 1.: Mean and standard deviation (between bracket) for each resolution performance for the different types of models and trials

	Computation frequency (Hz)	**Reconstruction error (cm)**	**Constraints violation**	
			Position (cm)	Rotation (no unit)
OR				
THW	31.6 10^{-2} (2.98 10^{-2})	1.61 (16.9 10^{-2})	3.58 10^{-9} (9.08 10^{-9})	5.66 10^{-9} (1.08 10^{-8})
ROM	77.1 10^{-2} (1.98 10^{-2})	1.21 (1.25 10^{-3})	N.A.	N.A.
IR				
THW	6.78 (2.41)	1.60 (16.6 10^{-2})	7.04 10^{-5}(7.32 10^{-5})	2.12 10^{-8} (2.00 10^{-8})
ROM	24.0 (1.12)	1.20 (1.25 10^{-3})	N.A.	N.A.

Table 2.: Mean and standard deviation (between bracket) for each resolution performance for the different types of models and trials (continued)

	Limits violation		**Joint coordinate difference**	
	Prismatic joints (cm)	Revolute joints (°)	Prismatic joints (cm)	Revolute joints (°)
OR				
THW	0	0	N.A.	N.A.
ROM	0	0	N.A.	N.A.
IR				
THW	0	$6.33\ 10^{-2}$ ($15.8\ 10^{-2}$)	$35.7\ 10^{-2}$ ($45.1\ 10^{-2}$)	4.31 (6.77)
ROM	N.A.	$5.36\ 10^{-2}$ ($4.07\ 10^{-3}$)	N.A.	2.20 (6.58)

List of Figures

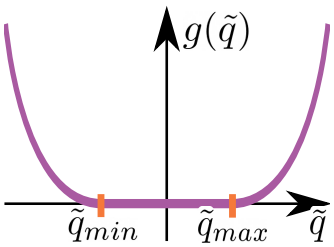

Figure 1.

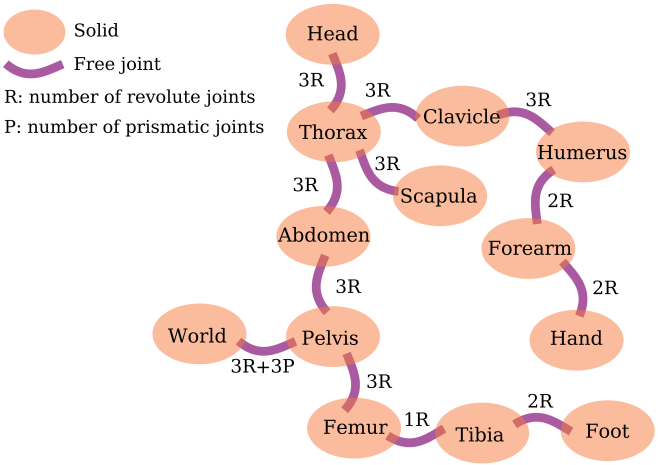
Solid
Free joint
R: number of revolute joints
P: number of prismatic joints
Head
Thorax
Clavicle
Humerus
Scapula
Forearm
Abdomen
Hand
World
Pelvis
Femur
Tibia
Foot
3R
3R
3R
3R
3R
3R
2R
2R
3R
3R+3P
3R
1R
2R

Figure 2.

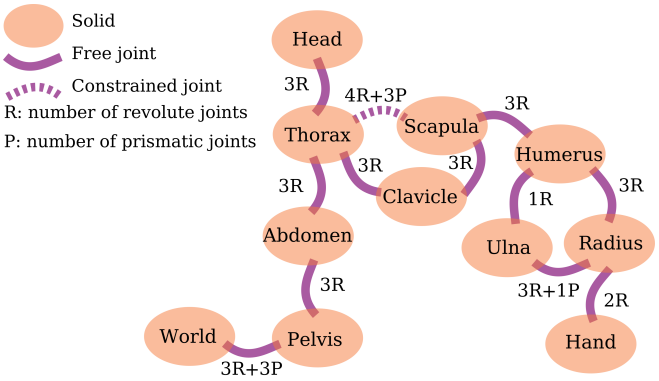
Solid
Free joint
Constrained joint
R: number of revolute joints
P: number of prismatic joints
Head
Thorax
Scapula
Humerus
Clavicle
Ulna
Radius
Hand
Abdomen
Pelvis
World
3R
4R+3P
3R
3R
3R
3R
3R
1R
3R
3R
3R+1P
2R
3R+3P

Figure 3.

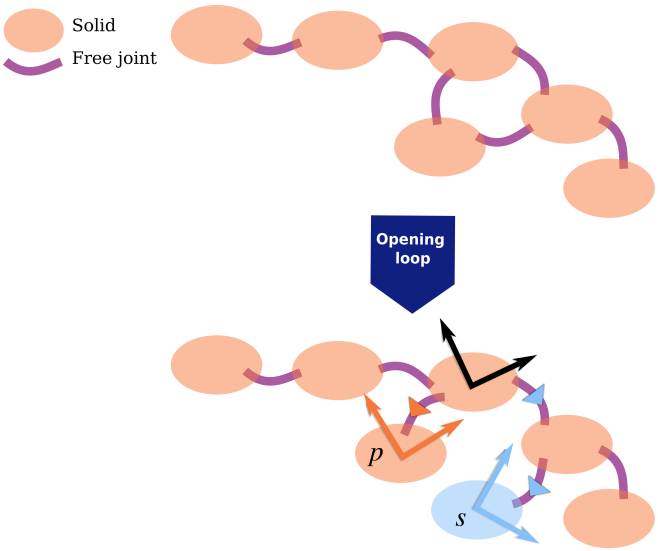

Figure 4.

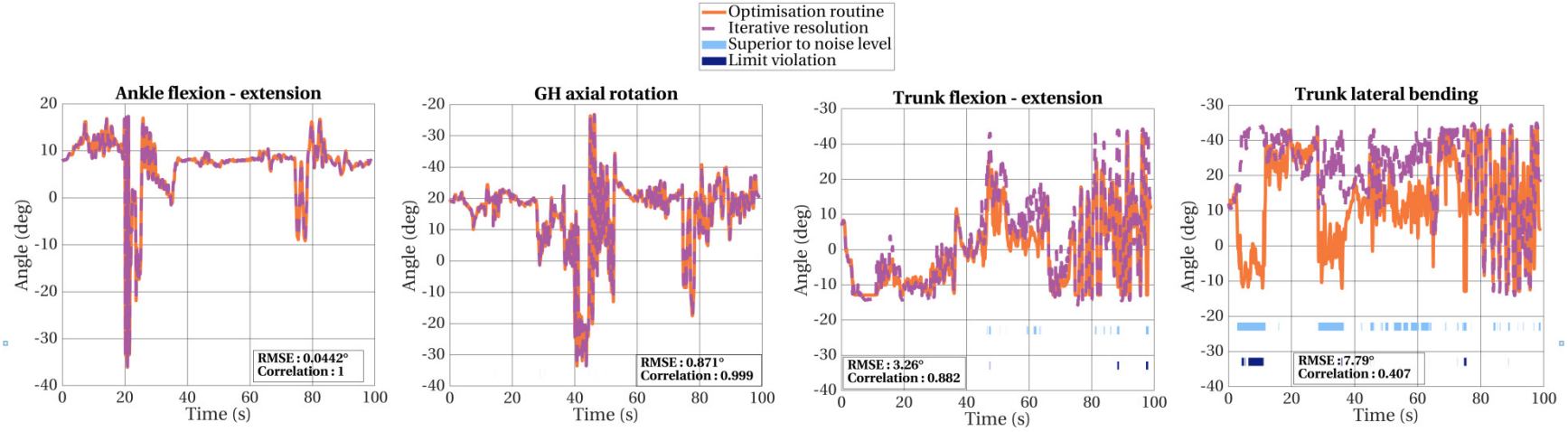
Optimisation routine
Iterative resolution
Superior to noise level
Limit violation
Ankle flexion - extension
RMSE : 0.0442°
Correlation : 1
GH axial rotation
RMSE : 0.871°
Correlation : 0.999
Trunk flexion - extension
RMSE : 3.26°
Correlation : 0.882
Trunk lateral bending
RMSE : 7.79°
Correlation : 0.407
Angle (deg)
Time (s)

Figure 5.

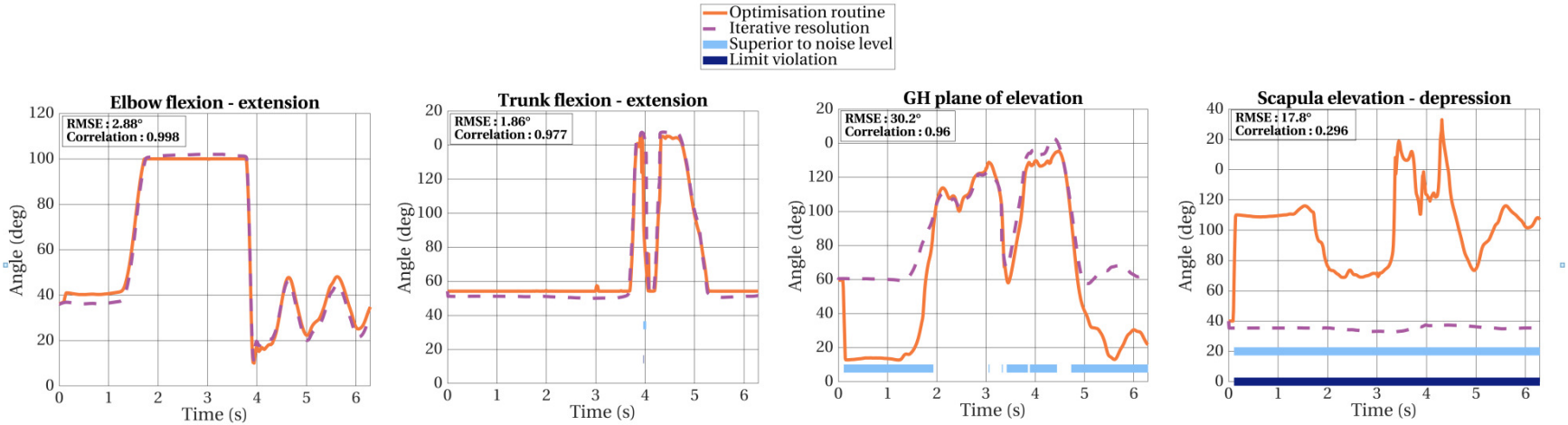
Optimisation routine
Iterative resolution
Superior to noise level
Limit violation
Elbow flexion - extension
RMSE : 2.88°
Correlation : 0.998
Trunk flexion - extension
RMSE : 1.86°
Correlation : 0.977
GH plane of elevation
RMSE : 30.2°
Correlation : 0.96
Scapula elevation - depression
RMSE : 17.8°
Correlation : 0.296
Angle (deg)
Time (s)

Figure 6.

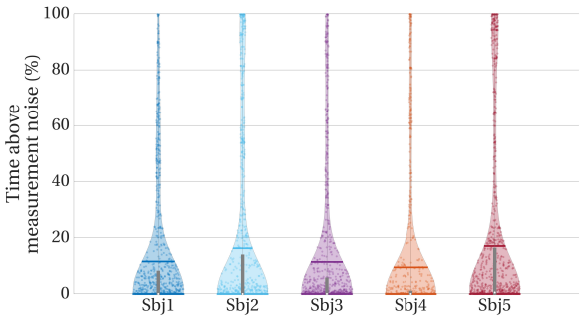
Time above
measurement noise (%)
100
80
60
40
20
0
Sbj1
Sbj2
Sbj3
Sbj4
Sbj5

Figure 7.

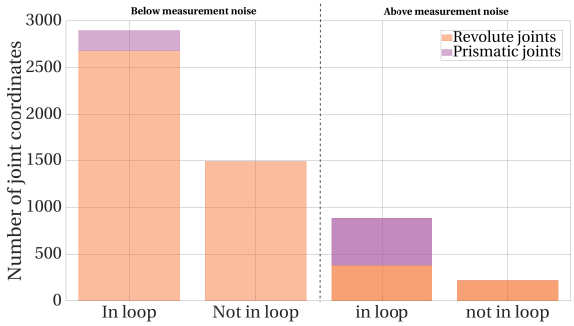
Below measurement noise
Above measurement noise
Revolute joints
Prismatic joints
Number of joint coordinates
3000
2500
2000
1500
1000
500
0
In loop
Not in loop
in loop
not in loop

Figure 8.

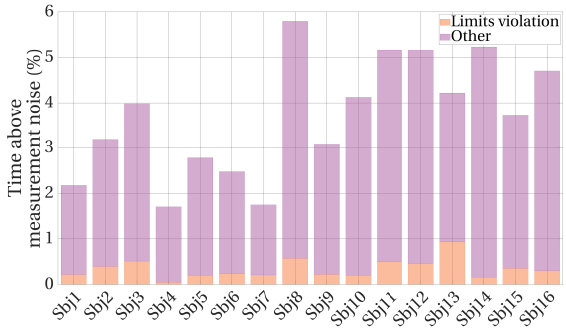
Limits violation
Other
Time above measurement noise (%)
0
1
2
3
4
5
6
Sbj1
Sbj2
Sbj3
Sbj4
Sbj5
Sbj6
Sbj7
Sbj8
Sbj9
Sbj10
Sbj11
Sbj12
Sbj13
Sbj14
Sbj15
Sbj16

Figure 9.

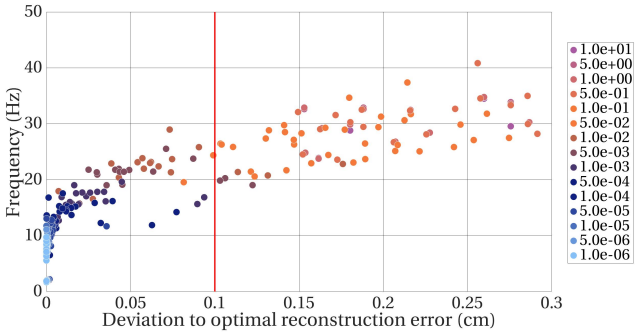
Frequency (Hz)
Deviation to optimal reconstruction error (cm)
0
10
20
30
40
50
0
0.05
0.1
0.15
0.2
0.25
0.3
1.0e+01
5.0e+00
1.0e+00
5.0e-01
1.0e-01
5.0e-02
1.0e-02
5.0e-03
1.0e-03
5.0e-04
1.0e-04
5.0e-05
1.0e-05
5.0e-06
1.0e-06

Figure 10.

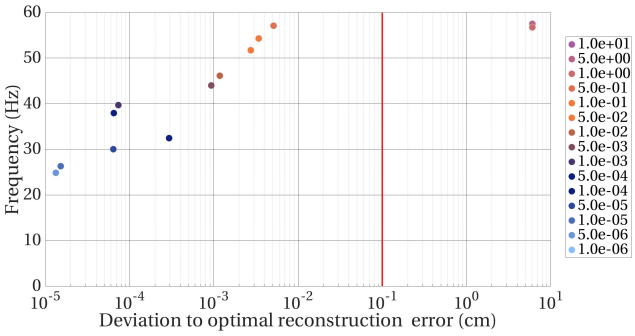
Frequency (Hz)
Deviation to optimal reconstruction error (cm)
1.0e+01
5.0e+00
1.0e+00
5.0e-01
1.0e-01
5.0e-02
1.0e-02
5.0e-03
1.0e-03
5.0e-04
1.0e-04
5.0e-05
1.0e-05
5.0e-06
1.0e-06

Figure 11.