

Progressive Clamping

Daniel Raunhardt, Ronan Boulic

Abstract—In this paper we propose the progressive clamping method to better model the kinematic anisotropy of joint limits for virtual mannequins or robots. Like recent approaches our method damps only the joints' variation component heading towards the limits. In addition we propose to dynamically express the corrective joint variation as a highest priority constraint that naturally extends the management of inequality constraints. This process is iterative within linear computing cost of the number of independent joints. We present how our approach is exploited for the major classes of rotation joints from one and up to three degrees of freedom. A comparison with other joint limit avoidance methods is given. We demonstrate the validity of our approach on various experiments targeting on the control of virtual mannequins.

I. INTRODUCTION

JOINT limit avoidance is a critical task to achieve to safely control robots. Researchers have regularly proposed control laws combining the correct achievement of desired manipulation tasks together with maintaining the joints as far as possible from the joint limits [11][13]. More recently the slightly relaxed problem of damping only the joint variation towards the limits has been addressed [5][6]. We have adopted this less constraining context as we mostly aim to interactively control 3D characters [8] or virtual mannequins [12]. Indeed, a control law that maximizes the distance of the joints to their limits is not suitable for modeling realistic human-like postures as some joints are often exploited near their limit range (e.g. knee in the standing posture). We now review the prior contributions achieved within the framework of the Gradient Projection Method (GPM) [11] and outline our contribution. The GPM has been widely used to solve the joint limit avoidance problem. It defines a performance criterion as a function of the joint limits. The gradient of this function is projected onto the null space of the main task. Due to this projection, the joint limit avoidance has no effects on the main task. For example Liégeois introduced a simple penalty function that attracts the joints to its midrange center [11]. However, by construction strict joint limit avoidance is not guaranteed. Nelson and Khosla [13] have used a method that minimizes an objective function to find a compromise solution between the main task and the joint limit avoidance. Chang and Dubey [5] have proposed a method using a weighted least-norm solution for a redundant

joint manipulator to strictly guarantee the joint limit avoidance. This method does not try to maximize the distance of the joints to their limits. Instead only the motion in the direction of the joint limit is damped. Chaumette and Marchand presented a more efficient joint limit avoidance method than the classical GPM [2][3]. Their method consists of generating motions by iteratively solving a linear system of equations that is compatible with the main task. They stop any motions that move the joints in the neighborhood of their joint limits. The GPM has been generalized to an arbitrary number of strict priority levels by Siciliano and Slotine [14] and its computation has been made less expensive by Baerlocher [7] under the name of Prioritized Inverse Kinematics (PIK). To avoid the joint limits, this latter has exploited a mechanism called clamping which strictly guarantees joint limit avoidance by adding dynamic constraints with the highest priority level. However, such a strict enforcement may lead to undesired discontinuity in the postural control. In addition we have observed that this technique may also converge to a non-optimal final state in singular contexts despite the use of the damped least squares inverses [10]. For these reasons we propose the progressive clamping technique which smoothly enforces a damping towards the joint limits and searches for an optimal task achievement within the remaining joint variation sub-space so that proposed method helps to avoid the solution to convergence to non-optimal final-states in singular contexts. In this paper we first briefly describe the PIK algorithm including the clamping mechanism for handling joint limits. Then we describe how we smoothly enforce joint limits for revolute, swing and ball-and-socket joints that are frequent for modeling 3D characters and virtual mannequins. We illustrate the interest of the proposed approach on a simple kinematic chain and a virtual mannequin and show that, in some cases, using joint limit avoidance with GPM or clamping does not lead to an optimal solution while our method succeeds.

II. CLAMPING FOR JOINT LIMIT AVOIDANCE

We provide here only a very brief overview of the PIK algorithm that handles an arbitrary number of priority levels and avoids joint limits for the purpose of controlling virtual mannequins or robot manipulators [1][7]. In this approach the controlled articulated structure is organized as a tree of chains, each consisting of an arbitrary number of revolute joints (1 degree of freedom rotation joint).

The general PIK algorithm relies on an efficient computation of projection operators enforcing the constraints

Manuscript received September 15, 2006. This work was supported in part by the Swiss National Science Foundation 200020-109989.

Daniel Raunhardt is with the Ecole Polytechnique Fédérale de Lausanne, 1015, Switzerland (email: daniel.raunhardt@epfl.ch)

Ronan Boulic is with the Ecole Polytechnique Fédérale de Lausanne, 1015, Switzerland (email: ronan.boulic@epfl.ch)

grouped into an arbitrary number of strict priority levels (Fig. 1). This is only valid within the small region of the current state. For this reason the norm of any constraint variation is limited to a maximal value. The corresponding joint variation solution resulting from the inner priority loop, noted Δq , is checked for the joint limit violation within the clamping loop. In the approach described in [1][7] all joints are one degree of freedom (dof) revolute joints with a range of motion [Min, Max]. These joint limits are handled as inequality constraints. Whenever the next configuration q_{k+1} (defined as the sum of the current configuration and the computed solution $q_k + \Delta q$) violates a joint limit, a new equality constraint is dynamically inserted at the highest priority level into to constraints list to clamp the corresponding joint on its limit. Afterwards the prioritized solution is re-evaluated as long as no additional joint limit is violated. This loop is necessary to guarantee the constraints' error minimization. The cost of the clamping loop is linear to the number of recruited joints (the number of joints taken into account for the prioritized solution computation). In the worst case each joint would be clamped individually at each successive clamping iteration. However, this is seldom the case as more than one joint generally violate their limits simultaneously, or even more frequently no joint violates its limits at all.

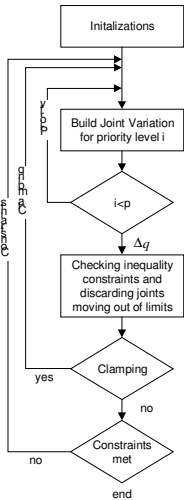


Fig. 1: The Prioritized Inverse Kinematics convergence loop (outer loop) highlighting the construction of the joint variation solution for multiple priority levels (inner loop) and the management of the joint limits through inequality constraints (clamping loop). Once the prioritized joint variation is obtained as the output of the inner loop, the updated configuration is checked for joint limit violation. Any detected subset of violating variations leads to the introduction of temporary equality constraints that clamp the corresponding joints on their respective limit. The prioritized solution is re-evaluated with this updated context as long as additional limit violation is detected (clamping loop).

III. JOINT MODELS

The purpose of a joint model is to represent the translational and rotational dof of a joint. In our context it is necessary to represent precisely enough the non-linearities at human joint limits. Combining two (resp. three) revolute joints to mimic the swing joint (resp. ball-and-socket) leads to a reachability cone with pyramidal basis that is not acceptable for our needs, especially when targeting on the evaluation of human tasks in Virtual Prototyping. Thus, in addition to the simple one dof *revolute* joint, we rely on explicit two dofs (*swing*) and three dofs (*ball-and-socket*) joint types for which we can exploit two types of joint limit models (elliptical cone and spherical polygon). More precisely the *swing* joint model

allows a spherical motion of a limb without axial motion (e.g. the palm movement w.r.t. radius bone, or the lumbar vertebrae mobility [8]). The articulated structure of the virtual mannequin used in the present study can be seen in Figure 2. The humanoid root joint has six dof (three for translation and three for rotation).

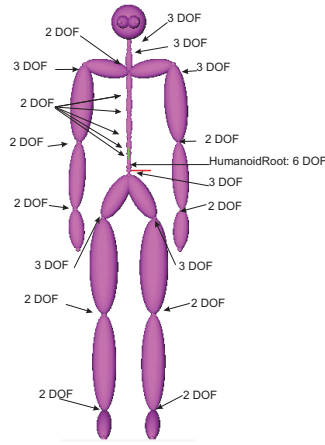


Fig. 2 : Considered degrees of freedom for our virtual mannequin model.

IV. PROGRESSIVE CLAMPING FOR JOINT LIMIT AVOIDANCE

A. Motivation

We define for each joint type a progressive clamping function that behaves like the standard clamping at the limit (cf. section II) and has a continuously decreasing influence until it vanishes at a certain distance to the limit. In addition, only a variation *towards* the limit is processed, nothing is performed otherwise similarly to [2][5]. Our function is generalized to two and three dofs joints as it only damps the joint variation component that brings the joint towards the limits, leaving orthogonal components unaltered. One of our critical requirements is to prevent the joint limit violation and to reduce the joint variation discontinuities near the limits. Therefore we advocate the extension of the strict clamping presented in section II to a smooth progressive clamping where each joint variation component towards the limits is increasingly damped as it happens closer to the limit. The resulting alteration of the joint variation is enforced with highest priority equality constraints leading to the construction of a null space on which the tasks constraints are achieved. This is the opposite of prior approaches that first build the task solution and then evaluate a compatible joint variation contribution to handle the behavior on or before the joint limit [2][5][11].

B. Joint models

1) Revolute joint

For simplicity, we use the same notation as Chaumette and Mansard [2]. Let us denote \bar{q}_{\min} and \bar{q}_{\max} the lower and upper limits that are not to be crossed. We define two damping activation thresholds \tilde{q}_{\min} and \tilde{q}_{\max} (Fig. 3) respectively for the lower and upper limit. Now we want to find a function h , defined over the damping intervals and

null outside, that has a minimal value of 0 at \tilde{q}_{\min} (resp. at \tilde{q}_{\max}) and a maximum value of 1 at \bar{q}_{\min} (resp. \bar{q}_{\max}). Such a function is best expressed through the *normalized damping activation distance* d defined as:

$$d = |\tilde{q} - q_k| / |\tilde{q} - \bar{q}| \quad (1)$$

where q_k is the current joint value.

A quadratic function could be used but our experience has shown that a cubic step function is more advantageous as it grows faster as a function of d , hence making it possible to keep the damping intervals smaller for a similarly smooth clamping behavior. Indeed it is preferred to minimize the size of the progressive clamping regions as our algorithm causes a re-evaluation of the PIK solution each time some joint variations are clamped (cf. section II and IV C).

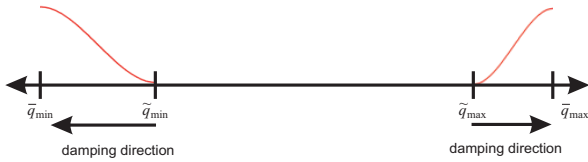


Fig. 3: Joint limits \bar{q} , damping activation thresholds \tilde{q} , damping functions varying from 0 to 100% (red) and damping directions.

Let Δq be the joint variation proposed by the PIK algorithm. The progressive clamping algorithm first evaluates whether the candidate new joint state q_{k+1} , defined as $q_k + \Delta q$, is to be altered or not. For this we evaluate the damping function defined as follow:

$$h = 1 \quad \text{if } q_{k+1} < \bar{q}_{\min} \text{ or } \bar{q}_{\max} < q_{k+1} \quad (2)$$

$$h = -2d^3 + 3d^2 \quad \text{if } q_{k+1} < q_k < \tilde{q}_{\min} \text{ or } \tilde{q}_{\max} < q_k < q_{k+1} \quad (3)$$

$$h = 0 \quad \text{otherwise} \quad (4)$$

A null value of h leaves the joint variation unaltered while an exact unit value requires to adjust the joint variation so that the joint is exactly clamped on the limit (as already described in section II). Otherwise, the new joint value q_{k+1}' is given by:

$$q_{k+1}' = q_k + (1-h)\Delta q \quad (5)$$

We now examine the additional differences for other joint types prior to provide an overview in subsection C.

2) Swing joint

The joint of the two dofs are coupled. The variation of the first dof may limit the possible variation of the second dof. So instead of acting independently on the two rotation dimensions, we act in a two stage process:

- First, we damp only the component of the joint variation that brings the joint towards its nearest limits.
- Second, an additional clamping check is made to ensure that the damped state is within the valid joint domain.

So first, any joint variation orthogonal to and heading

towards the limit has to be damped while any variation parallel to the limit is kept unchanged. In our implementation, the 2D swing joint is parameterized by an exponential map vector belonging to a plane. An elliptical joint limit domain centered on the origin in that plane builds an elliptical cone limit in 3D space. We have chosen to perform the damping over an uniform thickness \hat{q} along the border of the ellipse in the exponential map (Fig. 4).

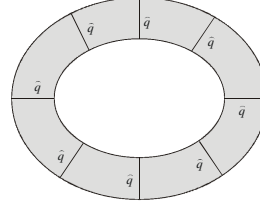


Fig. 4: Uniform activation thickness \hat{q} inside an elliptic joint limit domain for the progressive clamping of a 2D swing joint.

The damping function is acting only along the unidimensional axis passing through the point defining the current state q_k and orthogonal to the closest point on the ellipse cp_k . With such conventions, the normalized damping activation distance d_k is given by:

$$d_k = (\hat{q} - |cp_k - q_k|) / \hat{q} \quad (6)$$

The same cubic step function of d_k is then used to damp the orthogonal component Δq^\perp of the 2D joint variation vector Δq . The parallel component Δq^\parallel is left unchanged. This decomposition is valid as long as the joint variation norm is small compared to the damping region thickness \hat{q} .

We have:

$$h = 1 \quad \text{if } q_{k+1} \text{ is outside the limit} \quad (7)$$

$$h = -2d_k^3 + 3d_k^2 \quad \text{if } d_k > d_{k+1} > 0 \quad (8)$$

$$h = 0 \quad \text{otherwise} \quad (9)$$

where d_{k+1} is defined with equation (6) with the candidate state q_{k+1} and the corresponding closest point on the ellipse cp_{k+1} (Fig. 5).

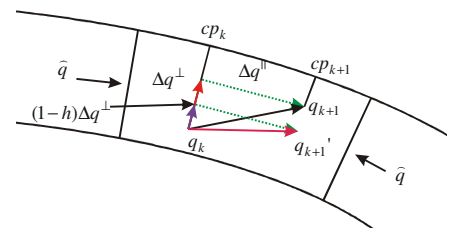


Fig. 5: Damping the joint variation for a 2D elliptical limit. Only the variation component orthogonal to the limit is damped. Please note that the joint variation should be small compared to the damping region thickness; the correct proportion is not respected here for the sake of clarity.

A null value leaves the joint variation unaltered like in the revolute joint case. On the other hand we process differently the case of the unit value which is treated in the second stage as an additional clamping check. For intermediate values of h the damped joint state q_{k+1}' is computed as:

$$q_{k+1}' = q_k + (1-h)\Delta q^\perp + \Delta q^\parallel \quad (10)$$

The second stage consists simply in an additional clamping check that process a possible violation detection of the first stage ($h=1$) and a (less frequent) limit violation of the resulting damping state due to the non-linear joint limit. In case a violation is detected the joint state is clamped to its nearest valid position as seen in Fig. 6. This mechanism allows the joint state to slide as much as possible on the limit instead of getting stuck or slowed down by a more restrictive approach.

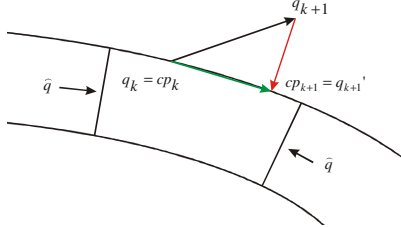


Fig. 6: Computation of the next joint state if the current state is exactly on the limit and the joint variation is heading outside the validity domain.

The limits of a swing joint can also be modeled as spherical polygons (Fig. 7). This is especially useful for complex human joints such as the shoulder or the hip. The same damping approach can be exploited for this type of validity domain too. Similarly the thickness \hat{q} is defined as the orthogonal distance to the limit as seen in Fig. 7 and only the joint variation orthogonal to the limit is damped. In this context however the points delimitating the spherical polygon lie on a sphere, so arcs and 3 dimensional points have to be used for the computations of the damping.

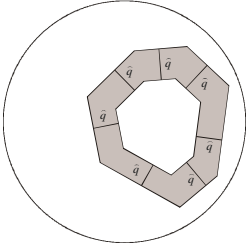


Fig. 7: Joint limit modeled as a spherical polygon with highlighting the damping region with constant thickness.

3) Ball-and-socket joint

The *ball-and-socket* joint is modeled as a combination of a swing (2 dofs) and an additional axial rotation (one dof revolute joint). Thus, for the damping we just combine the damping of a swing joint and the one of a revolute joint.

C. Putting all together

The progressive clamping has been integrated in the Prioritized Inverse Kinematics architecture as a generalization of the existing clamping loop (cf. II). It takes place just after the computation of the solution Δq . Whenever triggering the damping conditions the solution is altered into a damped variation $\Delta q'$. The final stage is the same as in the strict clamping architecture, i.e. enforcing the inequality constraints such as joint limits.

The detailed steps of the complete algorithm can be summarized as follows (Fig. 8):

1. Compute the solution Δq .
2. Check if there are any joints to be damped (cf. IV A and B). Output the potentially altered variation $\Delta q'$.
3. Check whether $q + \Delta q'$ violates any joint limit or not. If it is the case, determine for each violating joint, the variation that clamp it *on* its limit.
4. If any joint variation has been damped or clamped, add a new temporary equality constraint per detected joint. These will be achieved at the highest priority level to strictly enforce the correspondingly altered joint variations. Goto (1) to re-evaluate the solution with the updated equality constraint set.
5. Otherwise, if no joint is damped or clamped, update the articulated structure state with the joint variation and remove the temporary equality constraints.
6. If the task constraints are not met, goto (1) to compute a new convergence step.

The double stage of damping and clamping allows to correctly handle complex joint limit shapes as often used in human articulated structures. The cost of our approach is the same as for the clamping loop. In the worst case each joint would be individually submitted to the progressive clamping at each successive iteration of the clamping/progressive clamping loop. That is why we have chosen a cubic damping function to minimize the size of the progressive clamping regions hence reducing the number of re-evaluations of the PIK solution.

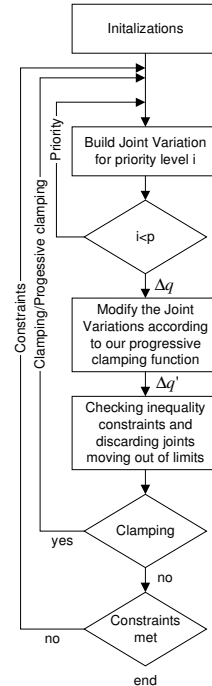


Fig. 8: The Prioritized Inverse Kinematics convergence loop together with the progressive clamping. Once the prioritized joint variation is obtained as an output of the inner loop, the updated configuration is updated accordingly to our progressive joint clamping. Afterwards the updated configuration is checked for joint limit violation. Any detected subset of progressive clamping or violating variations leads to the introduction of temporary equality constraints. The prioritized solution is re-evaluated with this updated context as long as additional progressive clamping or limit violation is detected (clamping/progressive clamping loop).

V. EXPERIMENTAL RESULTS

Three types of end effector controls have been exploited in the following experiments: position and/or orientation control, and the position control of the center of mass (given

the mass distribution of the articulated structure). The latter is especially useful for ensuring the balance of virtual mannequins.

A. Kinematic chain

This experiment highlights the behavior of our proposed approach with multiple conflicting tasks. Each joint can rotate clockwise or counterclockwise (± 1 rd). We have defined three positional tasks to be achieved by three effectors distributed on the chain (Fig. 9 left). Each effector positioning task has a distinct priority. The behavior of the kinematic chain during the convergence with progressive clamping is illustrated in Fig. 9 middle and right.

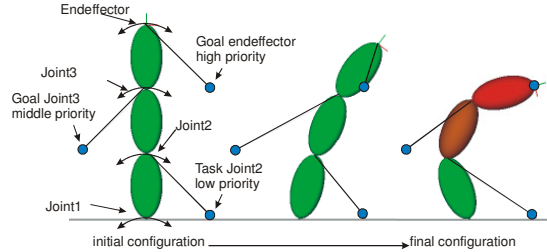


Fig. 9: Goal definitions with different priority levels for a simple kinematic chain and the behavior over time of our progressive clamping method. The strength of the red color indicates the value of our damping cost function (green = no damping, red = full damping).

For a comparison with other approaches we have performed this experiment also for the clamping approach and the GPM. For the GPM the joint limit avoidance has the lowest priority among all other tasks. Figure 10 depicts the behavior of the joints and the decrease of the error during convergence for these three methods.

The joint2_GPM and the joint3_GPM violate their limit as the null space dimension is insufficient to achieve this additional optimization. The progressively clamped joints produce the smoothest curves values (e.g. compare joint3_progressive_clamped with joint3_clamped). The “over swing” observed for the other approaches is not present for progressive clamping which is an advantage especially for the animation of virtual mannequins. The error convergence of these three methods is almost the same as seen in Fig. 10 (the final error is not null due to the conflicting tasks).

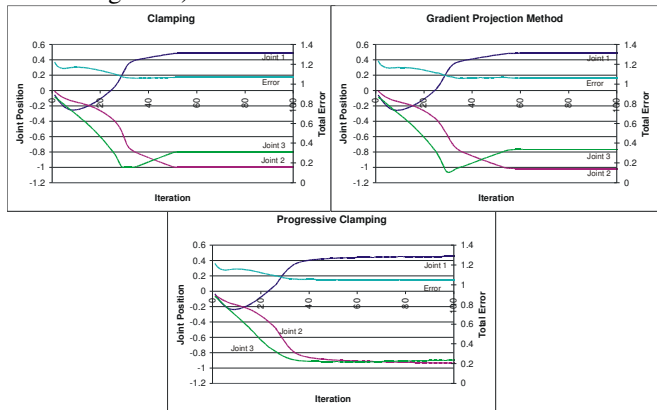


Fig. 10: Behavior of the different one dof revolute joints with conflicting tasks. Joint limits are $[-1, 1]$, the activation damping thresholds are for each

joints set to -0.5 resp. 0.5 .

B. Virtual mannequin

We consider now the virtual mannequin presented in Fig. 2. The virtual mannequin has to achieve a balanced posture requiring a high suppleness. The toes of the right foot and the hand of the right arm have to reach the same position in its back as if to remove a thorn from the foot. This is achieved with two middle priority positional tasks. We use an orientation low priority task to turn the head so that the virtual mannequin can see his hand and foot behind him. During the whole motion the left foot has to stay on the ground which is reflected by the two highest priority tasks (one for the toes and one for the heel). During the whole sequence the virtual mannequin has to keep its balance; this is done by projecting the center of mass over the left foot with a higher priority than the other reaching tasks.

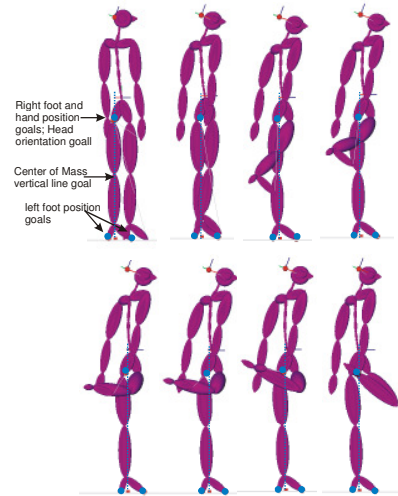


Fig. 11: The first posture of the virtual mannequin on the upper left states the initial configuration including the different task definitions. The evolution over time of the progressive clamping is seen from the left to the right. As seen in this sequence the virtual mannequin achieves the different goals.

We executed this experiment twice, once with our progressive clamping approach and once with the classical clamping. The visual behaviors and the final stable states of these two experiments are similar. In Fig. 11 successive postures adopted during the progressive clamping are displayed.

Although both methods are visually similar there are some important differences to note. The progressive clamping approach is particularly well suited to redistribute the PIK solution among joints by damping the joints that are close to their limit and moving toward it, and by requesting a greater contribution from the other joints that are moving in their “free” undamped region. We can see this phenomenon on the knee of the right leg. Our knee joint model for the virtual mannequin consists of two independent dof *revolute* joints allowing an independent *flexion* and *twist* rotation of the knee. Thus, for each dof the progressive clamping is evaluated independently. Fig. 12 illustrates the evolution of the total error between progressive clamping and clamping. Also

the values for our progressive clamping function (continuous within $[0,1]$) and classical clamping (either 0 or 1) are given in Fig. 12 for the two dofs. The error decrease of our progressive clamping approach is higher. This means that we reach the final stable configuration within less iterations of the PIK algorithm. The progressive clamping reaches a total error of 0.1 within 393 iterations while the classical clamping method requires 455 iterations. The reason for this better error convergence is the progressive damping of the *twist* dof leading to an early transfer of the solution to the *flexion* dof. Thus, the right knee flexes faster with this approach which leads to a faster convergence in that case. In general, such a behavior represents better the kinematic anisotropy of joint movements as the joint mobility is more and more hindered by resistive soft tissues or bone to bone impingement as its state gets closer to its limit value. Fig. 12 also highlights the progressive clamping of the right knee flexion joint starting around the end of the convergence.

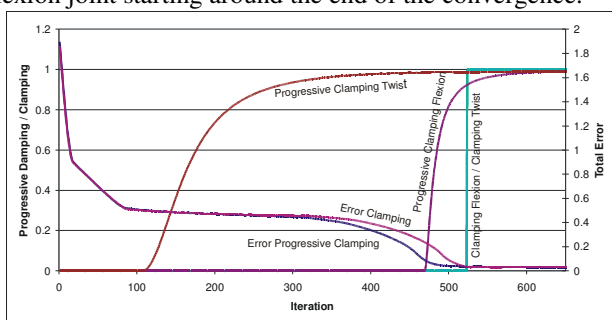


Fig. 12: The evolution of the errors for progressive clamping and clamping over the number of iterations. The values of our progressive clamping function for the two independent dof have two lie between $[0,1]$, where 0 means no damping and 1 full damping. The values of the two dof of the knee joint for clamping can either have the discrete values of 0 resp. 1 to indicate “no clamping” resp “clamping”.

The increasing “viscosity” introduced by the progressive clamping is also rather useful to prevent the kinematic singularity that occur close to the joint limits (e.g. knee or elbow full extension) which is a general problem of the clamping technique described in [10]. By reducing variations towards and allowing variations away from such singular postures the progressive clamping is an additional factor of convergence stability, in addition to the damping factor λ described in [10]. It is difficult to find a general compromise between the definition of the progressive clamping regions and the damping factor λ . On the one hand, if the progressive clamping regions are too big, the process may take too long as progressive clamping causes re-evaluations of the PIK solution. On the other hand, if the damping factor λ is large the convergence time is too prohibitive. We have often set the activation thresholds between 0.1 rd and 0.3 rd which has worked well for the majorities of the examples we have tested. A compromise solution would be to use progressive clamping only for the joints that may lead to kinematic singularities while using simple clamping for the remaining joints.

VI. CONCLUSION

In this paper we have proposed a method to better model the kinematic anisotropy of joint limits that damps only the part of the motion that bring the joints towards their limit based on a progressive damping function. Our integration into the PIK algorithm still guarantees the arbitrary number of priority levels and the strict joint limit avoidance. Experiments were made on simple kinematic chain and a virtual mannequin. The experimental results are convincing. We also verified that our method is helpful in kinematic singularity avoidance which generally occur with fully extended articulations.

REFERENCES

- [1] P. Baerlocher, R. Boulic: “An Inverse Kinematic Architecture Enforcing an Arbitrary Number of Strict Priority Levels”, in *The Visual Computer, Springer Verlag*, 20(6), pp 402-417, 2004.
- [2] F. Chaumette, E. Marchand: “A new redundancy-based iterative scheme for avoiding joint limits: Application to a visual servoing”, in *IEEE Int. Conf. On Robotics and Automation*, vol. 2, San Francisco, CA, Apr. 2000, pp. 1720-1725
- [3] N. Courty, E. Marchand, B. Araldi: “Through-the-eyes control of a virtual humanoid”, *IEEE Computer Animation*, Seoul, Korea, pp. 74-83, November 2001.
- [4] R. Boulic, R. Mas, D. Thalmann: “Interactive Identification of the Center of Mass Reachable Space for an Articulated Manipulator”, *Proc. of International Conference of Advanced Robotics ICAR 97*, Monterey, pp. 589-594, July 1997.
- [5] T. F. Chang and R.-V. Dubey: “A weighted least-norm solution based scheme for avoiding joint limits for redundant manipulators”, in *IEEE Trans. On Robotics and Automation*, 11(2):286-292, April 1995.
- [6] E. Marchand, F. Chaumette, A. Rizzo: “Using the task function approach to avoid robot joint limits and kinematic singularities in visual servoing”, in *Proc. Of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'96*, 12(3), pp. 1081-1090, November 1996.
- [7] P. Baerlocher, R. Boulic: “Task-priority formulations for the kinematic control of highly redundant articulated structures”, in *Proc. Of IROS'98*, Victoria, Canada, October 1998.
- [8] N. Magnenat-Thalmann, D. Thalmann: “Handbook of Virtual Humans”, Wiley, ISBN 0-470-02361-3, 2004.
- [9] C. Welman: “Inverse Kinematics and Geometric Constraints for Articulated Figure Manipulation”, Master Thesis, Simon Fraser University, 1993.
- [10] A. A. Maciejewski, C. A. Klein, “Numerical Filtering for the Operation of Robotic Manipulators through Kinematically Singular Configurations”, *Journal of Robotic Systems*, Vol. 5 N°6, pp. 527-552, 1988.
- [11] A. Liegeois: “Automatic supervisory control of the configuration and behaviour of multibody mechanism”, in *IEEE Trans. On Systems, Man Cyber. SMC-7*(12), pp. 868-871, 1997.
- [12] C. Phillips, J. Zhao, N. I. Badler: “Interactive real-time articulated figure manipulation using multiple kinematic constraints”, in *ACM Symposium on Interactive 3D Graphics*, pp. 245-250, March 1990.
- [13] B. Nelson, P. K. Khosla: “Strategies for increasing the tracking region of an eye-in-hand system by singularity and joint limit avoidance”, *Int. J. Robot. Res.*, vol. 14, no. 3, pp. 255-269, June 1995.
- [14] B. Siciliano, J-J. Slotine: “A General Framework for Managing Multiple Tasks in Highly Redundant Robotic Systems”, *Proc. of International Conference of Advanced Robotics ICAR 91*, Vol 2, pp 1211-1215, ISBN 0-7803-0078-5, 1991.
- [15] K. Yamane, Y. Nakamura: “Synergetic CG Choreography through Constraining and Deconstraining at Will”, in *IEEE Trans. On Robotics and Automation*, 855-862, May 2002.