An Anticipative Kinematic Limitation Avoidance Algorithm For Collaborative Robots: Two-Dimensional Case

Article *in* Proceedings of the ... IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE/RSJ International Conference on Intelligent Robots and Systems · December 2016

DOI: 10.1109/IROS.2016.7759623

2 authors:

Alexandre Campeau-Lecours
Laval University

Clément Gosselin
Laval University

An Anticipative Kinematic Limitation Avoidance Algorithm For Collaborative Robots: Two-Dimensional Case

This paper is a Post-Print version (ie final draft post-refereeing). For access to Publishers version, please access http://ieeexplore.ieee.org/document/7759623

IEEE/RSJ 2016.

Alexandre Campeau-Lecours and Clément Gosselin[1]

Abstract— **This paper presents an anticipative robot kinematic limitation avoidance algorithm for collaborative robots. The main objective is to improve the performance and the intuitivity of the physical human-robot interaction. One obstacle to achieve this goal is the management of limitations such as joint position limitation, singularities and collisions with the environment. Indeed, in addition to performing a given principal task, human users must pay a close attention to the manipulator configuration in order to handle the kinematic limitations. The proposed anticipative algorithm aims at relieving the human user from having to deal with such limitations. The algorithm is first presented and detailed for each individual limitation of a planar RR serial robot. The framework developed to manage several limitations occurring simultaneously is then presented. Finally, experiments are performed in order to assess the performance of the algorithm.**

I. INTRODUCTION

Collaborative robots working alongside humans are now used in many industrial applications. One of the main challenges of this technological achievement is to ensure human safety. In order to solve this primary concern, many innovations were made in areas such as mechanical design, control, sensors, software and planning [1], [2], [3].

The intuitivity of the interaction to the human user is another very important aspect [4], [5], [6]. To this end, [7], [8] studied the collaboration between humans in order to design robotic trajectories that behave similarly to motions performed by another human operator. In [4], the authors went further and proposed that the robot motions should not only be intuitive and predictable but also legible. From their definition, a legible motion is a functional motion that enables the collaborator to quickly and confidently infer the robot's goal.

One obstacle to the intuitivity of collaborative systems is the management of limitations such as joint position limitation, singularities and collision with the environment. For instance, most commercial collaborative robots will stop or even trigger an emergency fault when such a limitation is reached, which is obviously not intuitive to the operator. Indeed, in addition to their principal task, the operators are required to pay close attention to the manipulator configuration in order to infer the kinematic limitations. This second task is difficult and demanding especially if the operator is not an expert in robot kinematics.

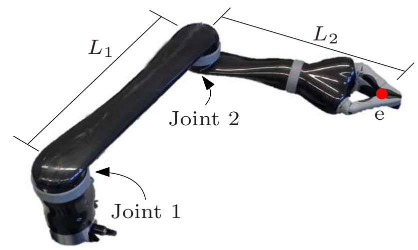

Fig. 1: RR planar robot used in the experiments based on a modified version of the JACO robot arm from Kinova.

The above issues are not a problem in typical industrial applications since the trajectories are planned off-line beforehand. Trajectory planning has been extensively explored in the literature and techniques have been developed to avoid self-collisions and collisions with the environment [9], [10], [11]. However, collaborative robots bring new challenges that may not be effectively handled by these existing techniques. First, collaborative robots are used in unstructured and dynamic environments and must manage many real-time constraints such as colocation with humans [12]. Secondly, existing trajectory planning algorithms are usually designed to find an optimal path between two given points. However, collaborative robots can be controlled directly by a human operator and the final destination is then unknown to the planner. The user is also very likely to move the robot toward configurations that exceed the limitations. The robot kinematic limitations must then be as transparent as possible in order to allow the operator to concentrate solely on his/her task alone.

On the other hand, reactive collision avoidance strategies are more adapted to the situation described above. Algorithms such as potential fields [13], [14], virtual spring-damper systems [15], [12] and virtual-fixtures [16] have been successfully implemented in various applications. These reactive strategies have been extensively explored in the literature. For instance, in [12], an interesting skeleton algorithm was developed and implemented on two 7-DOF serial arms in a torso configuration. This algorithm is able to compute the distance between different robot links and to generate the corresponding repulsive movements using virtual spring-

[1]The authors are with the Department of Mechanical Engineering, Université Laval, Québec, Canada. alexandre.campeau-lecours@gmc.ulaval.ca, gosselin@gmc.ulaval.ca

damper systems in order to avoid self-collisions [12].

Although reactive collision avoidance strategies can be used to avoid internal or external limitations, they present inherent drawbacks. For instance, because of their reactive nature, they react only when a limitation is infringed. Indeed, a repulsive force is defined in a zone around obstacles and in order to keep this zone as small as possible (to avoid affecting motions that are free from interferences), the repulsive force must be large for a small displacement in the zone close to the obstacles, which yields a large stiffness. However, high stiffness may lead to oscillations and even instability [17]. Several compromises are then usually required to properly tune the reactive algorithm's parameters, which is even more challenging since the resulting behavior depends on many factors such as the robot dynamics (velocity, acceleration). Another challenge, which has not been much explored in the literature, is the management of several limitations occurring simultaneously. The main challenge arises from the fact that an algorithm might be efficient to manage individual limitations but may fail when considering multiple limitations. For instance, when using reactive algorithms, the reaction forces generated by different limitations can interact with each other (for example cancelling each other).

By contrast with the literature, this paper presents an anticipative limitation avoidance strategy aiming at increasing the performance and intuitivity of cooperative systems while alleviating the reactive methods' drawbacks. The proposed interaction kinematics method analytically determines the required actions to slide along the limitations' surface anticipatively rather than generating repulsive motion/forces. The sliding zone around the obstacles can then be very thin with no oscillations nor instability issues. Additionally, because the method is analytical, tuning the parameters is very straightforward. Finally, the proposed method allows the development of a framework to successfully manage several limitations occurring simultaneously.

This paper is structured as follows. The proposed anticipative algorithm is first presented and detailed for each individual limitation of the planar RR serial robot shown in Fig. 1. The framework allowing the management of several limitations occurring simultaneously is then presented. Experiments are then performed in order to assess the performance of the algorithm. Finally, the results are discussed and a conclusion is drawn.

II. Individual limitation avoidance strategy

This section presents the strategies developed to make the robot slide along the kinematic constraints. The robot limitations can be considered as internal (joint limitation, singularities, self-collision) and external (obstacles such as objects, people, protection zones). In the first case, proprioceptive information (such as position sensors or position command) can be used to determine the robot configuration. In the second case, exteroceptive information must be obtained in real-time (with sensors such as 2D cameras, 3D cameras, force/torque sensors), in order to properly react to the environment [12]. For instance, in [18], a 3D kinect camera was used to detect humans in the robot's environment and to generate reactive movements. The limitations are presented in Fig. 2 for a planar RR serial robot. In order to simultaneously manage several limitations, as detailed in section III.C, all the mathematical constraints associated with the different limitations must be expressed using the same set of coordinates (either Cartesian coordinates or joint coordinates). In this paper the Cartesian space is considered.

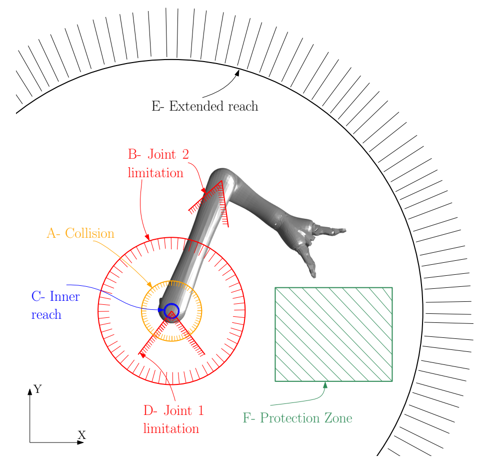

Fig. 2: Workspace limitations for the planar RR robot.

A. Collision with the base

This limitation's objective is to prevent the end-effector from colliding with the base. The limitation is represented by a circle around the base (A in Fig. 2) in which the end-effector should not be able to enter. Instead of stopping the robot when it is instructed to move toward this zone, it is rather desired to slide along the constraint. In order to anticiplately slide on the constraint, the Cartesian command is modified by the sliding algorithm before sending the final Cartesian command to the inverse kinematic algorithm as shown in Fig. 3.

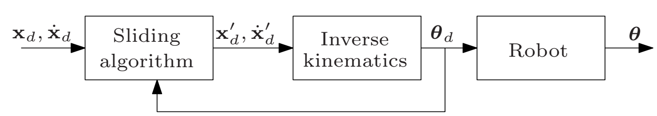

Fig. 3: Proposed anticipative sliding algorihtm.

In the example shown in Fig. 4(a), the end-effector (e) is at the edge of the limitation and the velocity command is represented by vector $\dot{\boldsymbol{x}}_{\boldsymbol{d}}$ (which would result in a collision). In this figure, the unit vector $\boldsymbol{n}$ represents the limitation's normal and the line λ is the limitation's tangent at the end-effector (point e). The modified vector $\dot{\boldsymbol{x}}'_{\boldsymbol{d}}$ is obtained using basic geometry by keeping only the component of vector $\dot{\boldsymbol{x}}_{\boldsymbol{d}}$ perpendicular to $\boldsymbol{n}$, which is equivalent to sliding on constraint λ:

$$\dot{\boldsymbol{x}}'_{\boldsymbol{d}} = \begin{cases} \dot{\boldsymbol{x}}_{\boldsymbol{d}}, & \text{if } \dot{\boldsymbol{x}}_{\boldsymbol{d}} \cdot \boldsymbol{n} \geq 0 \\ \dot{\boldsymbol{x}}_{\boldsymbol{d}} - (\dot{\boldsymbol{x}}_{\boldsymbol{d}} \cdot \boldsymbol{n})\boldsymbol{n}, & \text{otherwise.} \end{cases} \tag{1}$$

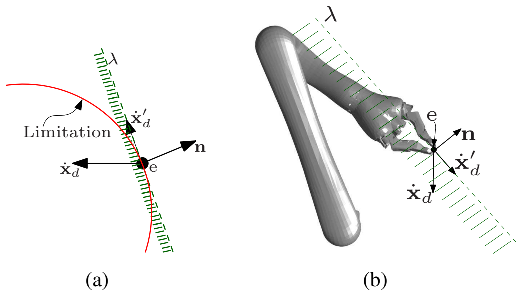

Fig. 4: a) Sliding example on a circle b) Example of limitation tangent λ for case B- Joint 2 limitation

When the projection of the desired Cartesian velocity vector on the vector normal to the limitation is in the outward direction ($\dot{\boldsymbol{x}}_{\boldsymbol{d}} \cdot \boldsymbol{n} \geq 0$), the desired motion is not directed toward the limitation and the desired velocity remains unchanged. Otherwise, the desired velocity is directed towards the limitation and the sliding algorithm from eqn. (1) is applied. The end-effector thus slides anticipatively on the surface of the limitation without having to infringe the limitation to generate a repulsion (as with reactive methods).

B. Limitation on joint 2

This limitation can result from an actuator physical limitation or from a collision between links 1 and 2 (identified by B in Fig. 2). In order to express this limitation in the Cartesian space, the relationship between the joint motions and the end-effector Cartesian motions must be determined. The relation is simply obtained with the Jacobian as:

$$\dot{\mathbf{x}} = \mathbf{J}\dot{\boldsymbol{\theta}} = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} \dot{\boldsymbol{\theta}} \tag{2}$$

where J_{ij} are the Jacobian's components and

$$\dot{\boldsymbol{\theta}} = \mathbf{K}\dot{\mathbf{x}} = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \dot{\mathbf{x}} \tag{3}$$

where $\mathbf{K}$ is the inverse of the Jacobian matrix (J^{-1}) and K_{ij} are the components of $\mathbf{K}$. From these equations, the following relation for joint 2 is obtained:

$$\dot{\theta}_2 = K_{21}\dot{x} + K_{22}\dot{y}. \tag{4}$$

and allows the anticipative determination of joint 2 direction for a given initial Cartesian command $\dot{\boldsymbol{x}}_{\boldsymbol{d}}$.

The limitation normal unit vector is obtained with $\mathbf{n} = unit([K_{21}, K_{22}]^T)$ as shown in Fig. 4(b). Cartesian commands $\dot{\mathbf{x}}_d$ with a component opposite to $\mathbf{n}$ would lead to a decrease of θ_2 and thus a collision. By using the same sliding algorithm as in section II.A, a modified Cartesian command, $\dot{\mathbf{x}}'_d$, is obtained by sliding on constraint λ (which is perpendicular to $\mathbf{n}$) using eqn. (1).

With this particular robot architecture, the limitation on joint 2 defines a circle around the base as shown in Fig. 2. The limitation could thus also be defined in the Cartesian space as in Section II.A.

C. Inner reach

The inner reach limitation (identified by C in Fig. 2) can lead to two different singularities. If $L_1 = L_2$, the point $x = 0, y = 0$ leads to a singularity. If $L_1 \neq L_2$, the limitation becomes a workspace boundary. In both cases, this limitation can be considered as a circle in the Cartesian space or a limitation on joint 2. This limitation can thus be solved using either the method presented in Section II.A or the method presented in Section II.B.

Figure 2 shows that different limitations such as collision (A), joint 2 (B) and inner reach (C) have the same circle geometry and thus, in practice, only the most restrictive limitation largest circle needs to be considered. With the particular architecture used in the experiments, the limitation on joint 2 is the most restrictive.

D. Limitation on joint 1

This limitation results from a physical limitation of the actuator (identified by D in Fig. 2). From equation 3, the following relation for joint 1 is obtained:

$$\dot{\theta}_1 = K_{11}\dot{x} + K_{12}\dot{y}. \tag{5}$$

Similarly to the limitation on joint 2, the limitation normal unit vector is obtained with $\mathbf{n} = unit([K_{21}, K_{22}]^T)$ and the sliding algorithm from eq. (1) can be used to slide on the constraint (see Fig. 5(a)).

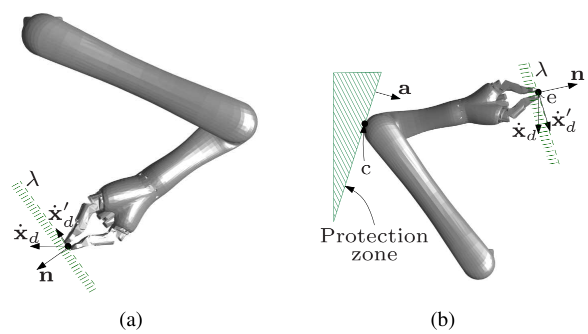

Fig. 5: a) Limitation for case D- Joint 1 limitation b) Limitation for case G- Protection zones (with links)

E. Extended reach

The extended reach limitation (E in Fig. 2), can be obtained in the Cartesian space (a circle) or in the joint space (limitation on θ_2). The unit normal vector $\mathbf{n}$ is simply obtained in the Cartesian space as $unit([-x, -y]^T)$. The sliding algorithm given in eq. (1) can be used to slide on the constraint.

F. Protection zones: end-effector

Protection zones are important to prevent parts of the robot from colliding with objects or humans (F in Fig. 2). Different basic geometric shapes such as circles and rectangles can be used and normal vectors are thus easily obtained in the Cartesian space. Given these unit normal vectors, the sliding algorithm provided in eq. (1) can be used to slide on the protection zone.

G. Protection zones: robot links

Computing the motion allowing the end-effector to slide on the protection zone is relatively simple. However, other parts of the robot may enter in the protection zone as shown in Fig. 5(b) where point c of the robot comes in contact with the zone. Vector **a** represents the limitation normal vector. However, it must be translated into a normal vector at the end-effector since the Cartesian commands are defined at this point.

The Cartesian velocities at the end-effector and at point c can be obtained respectively with

$$\dot{\mathbf{x}} = \mathbf{J}\dot{\boldsymbol{\theta}} \tag{6}$$

and

$$\dot{\mathbf{x}}_c = \mathbf{J}_c\dot{\boldsymbol{\theta}} \tag{7}$$

where J is the Jacobian at the end-effector and J_c is the Jacobian at point c. Solving (6) for $\dot{\boldsymbol{\theta}}$ and substituting the result in eq. (7), the relation between $\dot{\mathbf{x}}$ and $\dot{\mathbf{x}}_c$ is obtained as

$$\dot{\mathbf{x}}_c = \mathbf{J}_c\mathbf{J}^{-1}\dot{\mathbf{x}}. \tag{8}$$

Since the components of $\mathbf{J}_c\mathbf{J}^{-1}$ give the relation between the Cartesian velocities at the end-effector and at point c, it is possible to predict the velocity at point c for a given command at the end-effector. It is then possible to transfer vector **a** to the normal vector **n** corresponding to the limitation for the end-effector motion instead of the limitation for the given link motion. The normal vector **n** is readily obtained from the components of $\mathbf{J}_c\mathbf{J}^{-1}$.

III. Multiple limitations

The proposed individual sliding method presented in the preceding section offers some advantages such as sliding anticipatively on the constraints rather than reactively. By using this analytical method, parameter tuning is very straightforward and contact oscillations or instability are not an issue. Another advantage of the proposed sliding method is that it allows the management of several limitations occurring simultaneously.

A. Proposed multiple limitation management

The objective of the proposed method is to manage several limitations occurring simultaneously. An important fact to consider is that all the instantaneous constraints presented in Section II are linear which leads to important implications. First, at a given time, a maximum of two limitations are to be considered (in 2D space) since other limitations are necessarily a combination of these two limitations. The second implication is that the resulting output sliding vector is the result of sliding on one and only one constraint. In order to find the output sliding vector ($\dot{\mathbf{x}}'_d$), a modified vector $\dot{\mathbf{x}}'_{di}$ is obtained by sliding on each active constraint i (for instance λ_1 and λ_2) by using eqn. (1). Each resulting vector is then tested to assess if it verifies all the limitation's constraints. As mentioned above, only one final solution is possible, which can be divided in three different cases, namely: (1) only one vector $\dot{\mathbf{x}}'_{di}$ satisfies all the contraints and the final sliding vector $\dot{\mathbf{x}}'_d$ is then equal to $\dot{\mathbf{x}}'_{di}$, (2) two or more vectors $\dot{\mathbf{x}}'_{di}$ satisfy all the constraints and these vectors are necessarily equal. This situation can happen if the initial desired vector does not point towards any limitation or if two limitations are the same. (3) None of the vectors $\dot{\mathbf{x}}'_{di}$ satisfies all the constraints and therefore the ouput vector $\dot{\mathbf{x}}'_d$ is null.

B. Joint vs Cartesian space

In the literature, the limitations are usually considered individually [15], [12]. Additionally, the joint and Cartesian constraints are also usually managed individually. This may lead to important issues when several limitations occur simultaneously. Indeed, the sliding motions to avoid the limitations are performed in sequence. Sliding on a constraint (such as a joint limitation) can thus lead to infringing another constraint (such as a Cartesian constraint). In order to successfully manage simultaneous constraints, all the limitations must be expressed in the same space and must be considered simultaneously. In this paper, all the constraints are expressed in the Cartesian space (see section II) because it is considered more intuitive and because it allows the generation of proper reactions directly at the Cartesian command level before performing the inverse kinematics (see Fig. 3)

C. Other considerations

1) Damped zone: Because the resulting sliding motions are determined analytically, no parameter tuning was required so far. However, in order to generate smoother reactions, a progressive sliding motion is considered. The concept is to slightly augment the limitation's boundary in order to include a damped zone. At the damped zone boundary, the resulting sliding velocity is equal to the initial desired velocity $\dot{\mathbf{x}}_d$ while at the actual limitation boundary it is equal to the modified vector $\dot{\mathbf{x}}'_d$ from eqn. (1). Between these two boundaries, the output sliding velocity proportionally varies between $\dot{\mathbf{x}}'_d$ and $\dot{\mathbf{x}}_d$. The only parameter to tune is the size of the damped zone which can be obtained from the robot's maximum velocity and acceleration and basic dynamic analysis.

2) Jacobian conditioning: In section II, some limitation's constraints were defined using the inverse Jacobian matrix, which is not defined in singular configurations. This potential drawback is alleviated by the fact that the robot does not reach such a limitation since proper motions are generated to slide around the singularity. However, in a configuration close to a singularity, the inverse Jacobian matrix is not well defined which introduces numerical errors in the definition of the constraint. This issue also arises in reactive methods [12]. In order to alleviate this issue, the robot configuration can be kept at a reasonable distance from the singularity (determined experimentally). As a safeguard and in a worst case scenario, the motion can be stopped if the singularity limitation is infringed upon.

IV. Experimentation

The main objective of the proposed interaction kinematics algorithm is to increase the performance and intuitiveness for humans physically interacting with a robot. In order to assess the performance of the proposed sliding algorithm, experiments were performed with and without the sliding algorithm (in the latter case a standard implementation in which the velocity of the end-effector is set to zero when a limitation is infringed upon is used). Subjects were not told which algorithm was used and the order was varied between subjects. The experiments were performed on a modified version of the JACO arm from Kinova as shown in Fig. 1. The controller is implemented on a real-time QNX computer with a sampling period of 2ms and the algorithms are programmed using Simulink/RT-LAB. Ten (10) participants aged between 22 and 41 participated in the experiments which were approved by the ethics committee of Université Laval certificate no. 2016-011/12-02-2016. A video shows excerpts of the experiments.

A. First experiment - Full attention

The first experiment consisted in performing the trajectory shown in Fig. 6 where the completion time was recorded. The participants were instructed to reach the circle points while avoiding the triangles. The results are shown in Fig. 7. The average completion time using a standard approach (without sliding algorithm) is $40.0s$ with a standard deviation of $6.3s$ The average completion time using the proposed sliding algorithm is $31.9s$ with a standard deviation of $3.8s$. The completion time was reduced by 20% by using the sliding algorithm which is considered significant by a Mann-Whitney-Wilcoxon non parametric test one-tailed (p = $0.00064 < 0.05$). The participants generally declared that controlling the robot without using the sliding algorithm was more difficult and less intuitive since the robot stopped each time it encountered a limitation instead of automatically sliding on it.

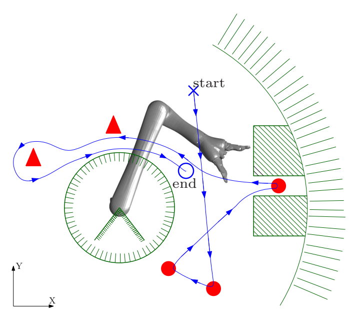

Fig. 6: Trajectory performed by the participants in the experiment.

B. Second experiment - Divided attention

The second experiment is similar to the first one but the human subject had to simultaneously accomplish a second task. This task consisted in naming a colour appearing on a computer screen at regular intervals of $2s$. In order to add difficulty to the decision process, the colour appeared on a "colour" word. For instance, for the word "blue" with a red texture, the answer was "red". The completion

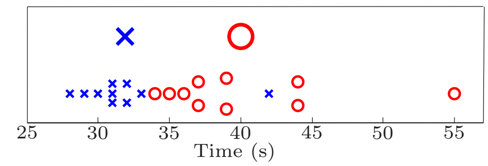

Fig. 7: Experimental results for the first experiment (full attention). The 'X' data points correspond to the proposed anticipative sliding method while the 'O' data points correspond to the standard kinematic implementation. Larger marks represent the average of the corresponding group of results.

time and the errors were recorded. The errors were defined as: (1) not naming a colour (-2 points), (2) naming an incorrect colour (-1 point) and (3) colliding with an object (represented as a triangle in the schematic of Fig. 6) (-3 points). This kind of experiment is often used in order to assess the attentional load of a cognitive process in psychology. In this experiment, the task is said to be in divided attention since the subjects also have to perform a secondary task (which is closer to the reality of an industrial application). In our specific case, the hypothesis is that when the sliding algorithm is not active, the users have to pay a closer attention to the robot configuration to avoid the kinematic limitations, which is cognitively demanding. Using the proposed sliding algorithm should then further increase the users' performance. The results of the second experiment are shown in Fig. 8. The average completion time using a standard approach (without the sliding algorithm) is $63.4s$ with a standard deviation of $9.3s$ and with an average error score of 4.7 points. The average completion time using the proposed sliding algorithm is $43.2s$ with a standard deviation of $3.93s$ and with an average error score of 0.9 point. The completion time was therefore reduced by 32% by using the sliding algorithm which is considered significant by a Mann-Whitney-Wilcoxon non parametric test one-tailed (p = $0.00009 < 0.05$). The average error score was reduced by a factor of 5.2.

The participants also generally declared that controlling the robot with the sliding algorithm was much easier and more intuitive. Additionally, they mentioned that the sliding algorithm represented a biggest advantage when they could not devote their full attention to the task. Indeed, in addition to the secondary task, the participants had the burden to pay a close attention to the robot kinematic configuration. This is correlated by the fact that the sliding algorithm leads to a more significant improvement when the task is performed in divided attention (improvement of 20% in full attention and improvement of 32% in divided attention). This result is considered significant by a Mann-Whitney-Wilcoxon non parametric test one-tailed (p = $0.005 < 0.05$) performed on the difference between the completion time of the experiments using the algorithms (divided attention time minus full attention time) compared to the difference of the experiments without the algorithms. These results seem to indicate that the proposed anticipative sliding algorithm not only increases the performance, but it also leads to a reduction of the required attentional load.

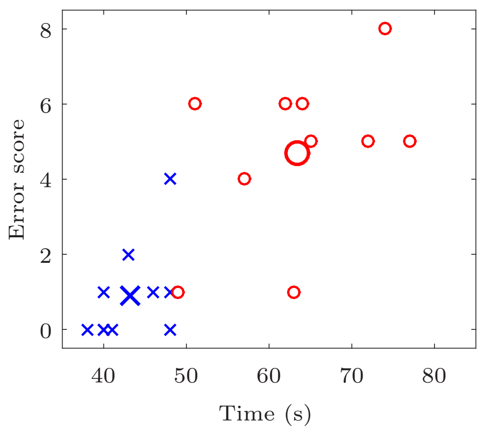

Fig. 8: Experimental results for the second experiment (divided attention). The 'X' data points correspond to the proposed anticipative sliding method while the 'O' data points correspond to the standard kinematic implementation. Larger marks represent the average of the corresponding group of results.

C. Third experiment - Multiple limitations

In the third experiment, the robot started from an initial position (-0.2, 0.03) and was instructed to reach a target position autonomously while avoiding protection zones (see Fig. 9). The experiments were performed with the proposed sliding algorithm and with a reactive method based on virtual spring-damper system. For the first test (a), the protection zones were connected and the robot stopped at the limitation zone with both methods. For the second test (b), with the reactive algorithm, the robot stopped at the edge of the triangle instead of passing through since the repulsion was too strong. The tuning of the reactive method is challenging since the reaction depends on many parameters such as the repulsion zone's stiffness and damping, the end-effector velocity and acceleration and the shape of the protection zone.

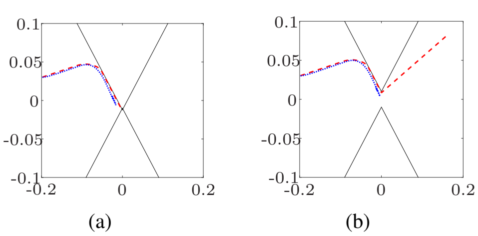

Fig. 9: Example of sliding on multiple constraints. The solid blue line corresponds to the reactive method while the dashed red line corresponds to the anticipative method.

V. Conclusion

This paper presented an anticipative robot kinematic limitation avoidance algorithm for collaborative robots. The main objective of this work is to improve the performance and intuitivity of the physical human-robot interaction while alleviating the drawbacks of reactive methods. A first experiment, with full attention on the task, revealed that the anticipative sliding algorithm improved the human performance by 20% compared to a standard implementation (for a particular given task). A second experiment performed in divided attention revealed that the performance were further improved (32%). These results indicate that the proposed anticipative sliding algorithm not only increases the performance, but also leads to a reduction of the required attentional load. Future work will focus on extending the algorithm to the three-dimensional case along with implementing real-time obstacle avoidance with a 3-D camera sensor.

Acknowledgment

This work was supported by The Natural Sciences and Engineering Research Council of Canada (NSERC).

References

[1] J. Kruger, T. Lien, and A. Verl, "Cooperation of human and machines in assembly lines," {*CIRP*} *Annals - Manufacturing Technology*, vol. 58, no. 2, pp. 628 – 646, 2009.

[2] A. De Santis, B. Siciliano, A. De Luca, and A. Bicchi, "An atlas of physical human–robot interaction," *Mechanism and Machine Theory*, vol. 43, no. 3, pp. 253–270, 2008.

[3] C. Gosselin, T. Laliberté, B. Mayer-St-Onge, S. Foucault, A. Lecours, V. Duchaine, N. Paradis, D. Gao, and R. Menassa, "A friendly beast of burden: A human-assistive robot for handling large payloads," *Robotics & Automation Magazine, IEEE*, vol. 20, no. 4, 2013.

[4] A. D. Dragan, S. Bauman, J. Forlizzi, and S. S. Srinivasa, "Effects of robot motion on human-robot collaboration," in *ACM/IEEE Int. Conference on Human-Robot Interaction*, 2015, pp. 51–58.

[5] A. Lecours, B. Mayer-St-Onge, and C. Gosselin, "Variable admittance control of a four-degree-of-freedom intelligent assist device," in *Proceedings of the 2012 IEEE International Conference on Robotics and Automation (ICRA)*, May 2012, pp. 3903–3908.

[6] A. Lecours and C. Gosselin, "Computed-torque control of a four-degree-of-freedom admittance controlled intelligent assist device," in *Experimental Robotics*. Springer, 2013, pp. 635–649.

[7] E. Gribovskaya, A. Kheddar, and A. Billard, "Motion learning and adaptive impedance for robot control during physical interaction with humans," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 4326–4332.

[8] E. Burdet, G. Ganesh, C. Yang, and A. Albu-Schäffer, "Interaction force, impedance and trajectory adaptation: by humans, for robots," in *Experimental Robotics*. Springer, 2014, pp. 331–345.

[9] L. E. Kavraki, P. Švestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *Robotics and Automation, IEEE Transactions on*, vol. 12, no. 4, pp. 566–580, 1996.

[10] K. Glass, R. Colbaugh, D. Lim, and H. Seraji, "Real-time collision avoidance for redundant manipulators," *Robotics and Automation, IEEE Transactions on*, vol. 11, no. 3, pp. 448–457, 1995.

[11] J. Kuffner, K. Nishiwaki, S. Kagami, Y. Kuniyoshi, M. Inaba, and H. Inoue, "Self-collision detection and prevention for humanoid robots," in *IEEE Int. Conference on Robotics and Automation*, vol. 3. IEEE, 2002, pp. 2265–2270.

[12] A. De Santis, A. Albu-Schäffer, C. Ott, B. Siciliano, and G. Hirzinger, "The skeleton algorithm for self-collision avoidance of a humanoid manipulator," in *Advanced intelligent mechatronics, 2007 IEEE/ASME international conference on*, 2007, pp. 1–6.

[13] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The Int. Journal of robotics research*, vol. 5, no. 1, 1986.

[14] S. Haddadin, R. Belder, and A. Albu-Schäffer, "Dynamic motion planning for robots in partially unknown environments," in *IFAC World Congress (IFAC2011), Milano, Italy*, vol. 18, 2011.

[15] F. Seto, K. Kosuge, and Y. Hirata, "Self-collision avoidance motion control for human robot cooperation system using robe," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005.

[16] S. A. Bowyer, B. L. Davies, and F. R. y Baena, "Active constraints/virtual fixtures: A survey," *IEEE Transactions on Robotics*, vol. 30, no. 1, pp. 138–157, Feb 2014.
[17] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," in *IEEE Int. Conference on Robotics and Automation*, 1991, pp. 1398–1404.
[18] F. Flacco, T. Kröger, A. De Luca, and O. Khatib, "A depth space approach to human-robot collision avoidance," in *IEEE International Conference on Robotics and Automation*, 2012, pp. 338–345.