Recognition of Intentional Violations of Active Constraints in Cooperative Manipulation Tasks

M. Aricò[1], S. A. Bowyer[2], E. De Momi[1], G. Ferrigno[1],
S. Pastorelli[3], F. Rodriguez y Baena[2]

[1]Department of Electronics, Information and Bioengineering, Politecnico di Milano, Italy
[2]Department of Mechanical Engineering, Imperial College London, UK
[3]Department of Mechanical and Aerospace Engineering, Politecnico di Torino, Italy
f.rodriguez@imperial.ac.uk

INTRODUCTION

Active Constraints (ACs) are high-level control algorithms deployed to assist a human operator in man-machine cooperative tasks [1], and define regions within which it is safe for the robot to move and cut [2]. To enhance the performance in cooperative surgical tasks, adaptive constraints have been exploited to optimally adjust the provided level of assistance according to some knowledge of the task, hardware or user. In [3] Hidden Markov Models were used for the run-time detection of the user intention to leave a guidance constraint to circumvent an obstacle. In this work, we present a novel, Neural Network (NN)-based method for the runtime classification of intentional and unintentional violations of ACs, that is trained on either statistical or frequency features from the enforced constraint forces. We investigate which set of parameters yield faster and more reliable classification results, both for guidance and regional constraints.

METHODS

Active Constraints

During cooperative assistance, intentional violations of ACs take place whenever the current action of the user is in disagreement with the purpose of the constraint, typically resulting from sensing limitations of the robotic system. In this case, the constraint is felt as a hindrance, resulting in disturbing interaction forces at the tip. Unintentional violations occur when the user shares the purpose of the constraint and accidental errors in the task execution are made. The classification of the user's intended action during the cooperative task would allow one to optimally adjust the assistance level provided by ACs. ACs can have two purposes [1]:

- *Guidance constraints* are enforced to guide the motion of the tool along a specified trajectory;
- *Regional constraints* are enforced to bound the motion of the tool into certain safe regions.

Both types of constraints were considered in this work, and modeled with a planar geometry according to a conventional viscoelastic constraint model:

$$\boldsymbol{f} = K(\boldsymbol{x} - \boldsymbol{x}_{eq}) - D\dot{\boldsymbol{x}} \qquad (1)$$

where $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$ are the position and velocity of the tool tip; K and D are the stiffness and damping parameters ($K > 0$, attractive fixture), $\mathbf{x}_{eq}$ is the equilibrium point that lies on the constraint where it is closest (Euclidian distance) to the current tool tip position. To enforce the constraints during the cooperative guidance, the commanded torque of the haptic master is computed from the resulting Cartesian force $\boldsymbol{f}$ according to the geometrical Jacobian.

Classification Method

Two NN-based binary classifiers were developed for the runtime identification of "intentional" and "unintentional" violations of ACs. The two approaches exploited different features, extracted from the interaction force signal across the tip-constraint interface, as follows:

1. Statistical (StNN): A feedforward NN was trained and validated on a dataset of 7 statistical features computed on the temporal evolution of the interaction force: mean, variance, energy, maximum value, integral, waveform length, average amplitude change;
2. Spectral (SpNN): A feedforward NN was trained and validated on a dataset of 10 spectral features computed on the time evolution of the force energy distribution using the Wavelet decomposition (9 levels) [4];

The structures of the StNN and SpNN were composed of one hidden layer (15 and 33 neurons respectively, "trial and error" optimization), and one output neuron (hyperbolic tangent activation functions).

EXPERIMENTAL DESIGN

Experimental setup

The Phantom Omni (Sensable Technologies, Inc.) haptic device was used during assisted cooperative tasks. The active constraint controller was implemented using the "PhanTorque"[1] Simulink-compatible library on Matlab/Simulink R2014b platform. Visual feedback of the task execution was provided with a 2-D monitor.

[1] https://sir.upc.edu/wikis/roblab/index.php/Projects/PhanTorqueLibraries

Experimental protocol

To evaluate task-independent properties of the two classifiers, two sets of tasks were considered (Fig. 1):

- *Following task.* The user was asked to accurately move along 2D spline-based paths, assisted by a *guidance constraint*. He/she was asked to circumvent any circular obstacles placed along the path by acting against the constraint;
- *Reaching task*. The user had to accurately place the pointer on several equally spaced targets, which lay within a forbidden region bounded by a *regional constraint* with 50% probability.

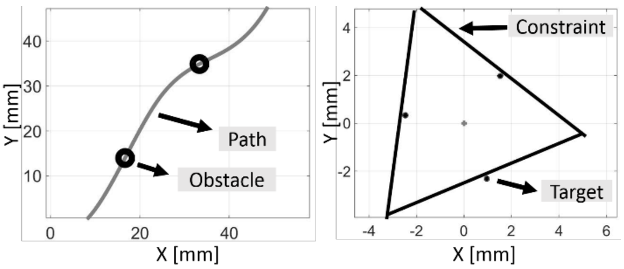

Fig. 1. Following Task (left) and Reaching Task (right)

We asked 12 subjects to perform 10 trials for each of the two tasks. The force signal was filtered, recorded, segmented to extract non-null interactions ($\mathbf{f} > 0$) and labeled as intentional or unintentional according to the known positions of obstacles/region boundaries. Two task-related datasets were built from all users across all trials. Both SpNN and StNN methods were cross-validated on each task dataset, and the performance were computed in terms of sensitivity (Se) and specificity (Sp) over time as:

$$Se(t) = TP(t)/(TP(t)+FN(t)) \qquad (2)$$

$$Sp(t) = TN(t)/(TN(t)+FP(t)) \qquad (3)$$

where TP and TN are the amount of correctly classified intentional (true positive) and unintentional violations (true negative), respectively; FN and FP are the amount of misclassified intentional (false negative) and unintentional violations (false positive), respectively.

RESULTS AND DISCUSSION

The classification threshold applied on the continuous output of the networks was optimized to achieve 90% asymptotic specificity. Consequently, the classifier was evaluated in terms of sensitivity-time profile (Fig. 2), and the minimum time interval to overcome a 90% sensitivity level was obtained. The Se index shows a sigmoidal profile in time for both tasks and methods. As reported in Table I, the 90% sensitivity was achieved for both methods within 1s for the *following* task (mean velocity 4.17 mm/s), and within 3s for the *reaching* task (mean velocity 8.36 mm/s). In the *following* tasks, as no motion limitation was imposed to the user during obstacle avoidance, a greater penetration was recorded with respect to the *reaching* task. Moreover, the StNN method resulted in higher performance with respect to SpNN method, yielding a 60% time reduction for the *following* task, and a 30% reduction for the *reaching* task. The proper classification timing is chosen based on the sensitivity level required by the specific application.

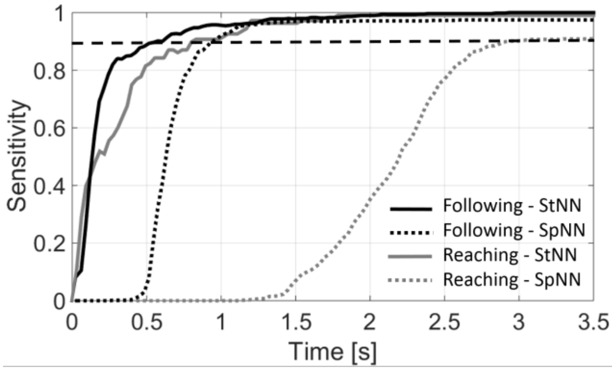

Fig. 2. Sensitivity profiles in time, across methods and tasks.

Table I Classification time and relative constraint penetration among methods and tasks ($\boldsymbol{Se = 90}$%).

Task	**Method**	**Time** [s]	**Penetration** [mm]
Following	StNN	0.559	0.82
	SpNN	0.954	1.21
Reaching	StNN	0.816	0.51
	SpNN	2.911	0.52

CONCLUSIONS

NN-based algorithms were demonstrated to be suitable for the runtime task-independent classification of intentional and unintentional violations of ACs. Better performance was obtained for the *guidance constraints* with respect to regional constraints. Additionally, the use of statistical features yields a faster classification with respect to the use of spectral parameters. Future work could apply multi-objective model selection to find the optimal classifier, introduce data regularization (subsampling) to prevent class unbalance and combine statistical and spectral features. Moreover, some methods to exploit the continuous output of the NN could be investigated to optimally modulate the constraint assistance according to the probability of the user's intention classification in surgical manipulation tasks.

REFERENCES

[1] S. Bowyer, B. L. Davies, and F. Rodriguez y Baena. "Active constraints/virtual fixtures: A survey." In: Robotics, IEEE Transactions on 30.1 (2014): 138-157.

[2] B. Davies, et al. "Active-constraint robotics for surgery." Proceedings of the IEEE 94.9 (2006): 1696-1704.

[3] D. Kragic, et al. "Human-machine collaborative systems for microsurgical applications. In: " The International Journal of Robotics Research 24.9 (2005): 731-741.

[4] S. Pittner, and S. V. Kamarthi. "Feature extraction from wavelet coefficients for pattern recognition tasks." In: Pattern Analysis and Machine Intelligence, IEEE Transactions on 21.1 (1999): 83-88.