A study on the use of tactile instructions for developing robot's motions

Fransiska Basoeki+, Fabio DallaLibera♱, Takashi Minato*, and Hiroshi Ishiguro†

+♱†Department of Systems Innovation, Graduate School of Engineering Science,

Osaka University, Engineering Science Building D458,

Machikaneyama 1-3, Toyonaka, Osaka, Japan 560-8531

Tel. & Fax.: +81-6-6850-6360

+Email: fransiska.basoeki@irl.sys.es.osaka-u.ac.jp

♱JSPS Research Fellow at Osaka University

*†ATR Hiroshi Ishiguro Laboratory, Kyoto 619-0288, Japan

Abstract - Developing motions for humanoid robots is time consuming. However, sport and dance instructors can easily adjust their students' postures by simple touches. This suggests the possibility of exploiting touch for motion development, and allows us to propose a methodology based on this concept. To realize such a system, it is required to define how the robot should interpret touches. We propose a supervised learning approach to cope with this issue, and verify its feasibility experimentally. We then study the data collected by the algorithm, and show that the system is practical both for motion development and for studying human-robot tactile communication. In particular, we present considerations on the sparsity that characterize the whole process and suggest how sparsity can be exploited for efficient interpretation of tactile instructions.

Keywords - humanoid robot, human-robot interaction, touch, tactile communication

1. INTRODUCTION

Humanoid robots often have a high number of degrees of freedom, and this makes motion development a challenging task. In fact, it is often impossible to use automatic learning for motion skills. To overcome this issue, various ways of transferring human knowledge into robots were presented in literature. When the task is known in advance, a programmer may transfer his knowledge through the robot's control algorithms, such as by designing a Central Pattern Generator (CPG) structure[1]. When the task is only partially known in advance, the programmer can provide modules or motion primitives, as in the by Mimesis model[2], while the final user specifies the motion task by composing using these elementary modules. Finally, the task may be unknown a priori, and in such case the final user must be able to transfer his knowledge to the robot directly, by using methods such as motion retargeting[3] and kinesthetic demonstration[4].

Our proposed method, Teaching by Touching (TbT), aims to tackle the third case. The idea is to mimic the way sport or dance instructors use touch to correct their students' movements. In conventional approaches based on kinesthetic demonstration[4], the robot is completely passive during the learning process. The user is required to modify the posture of the robot to the desired one by application of the necessary forces. In contrast, we propose to equip the robot with knowledge on touch instructions meanings, and give it an active role in interpreting the given touch. More explicitly, in our setting, when the robot receives a touch, it estimates the user's intention and moves accordingly.

There are several possibilities for defining how touch instructions should be interpreted by the robot. The first possible choice is to force the user to learn and use a fixed protocol. In this setup the touch interpretation algorithms can be very simple. It is sufficient to interpret the instructions according to the specified mapping. However, the user needs to remember which touch instruction leads to which movement, and this may be difficult for inexperienced users. Another approach consists in modeling the way humans communicate through touch. In this case the user can just intuitively provide touch instructions while the robot uses more complex algorithms to interpret the meanings based on all the available information.

Our system takes the second approach, as explained in the following section. Section 3 will present the hardware and the system we developed, which can be used both for motion development and for studying the mapping between touch instructions and desired robot responses. The ability to study the mapping is important, because it allows the extraction of general criteria for interpreting tactile instructions, which can be used for future design of better tactile instruction interpretation algorithms. In this paper, the analysis of sparsity that characterizes the experiment data will be provided as an example of possible studies on the collected data. Section 4 will conclude the paper by summarizing the main results.

2. ALGORITHM

In order to build a model of the mapping between touch instructions and desired responses by supervised learning, it is necessary to collect data of how people expect their touches to be interpreted. Our system collects these data online, during the motion development process. In this way, no initial model is required, and the model can be progressively refined as the user interacts with the robot. Fig.1 presents the data collection methodology employed in our implementation. When the user applies a touch instruction and the robot does not interpret the touch as expected by the user, the user notifies the robot that the interpretation is wrong, and shows the correct interpretation by manually adjusting the robot limbs to the intended posture. The robot records this example of touch instruction and the corresponding desired response into its database, and uses it to improve its subsequent interpretations.

Fig.1

The examples in the database are taken as input by a Kernel regression algorithm, which is used by the robot for interpreting touch instructions. Formally, the input $I \in \mathbb{R}^{n+m+o}$ consists of the touch sensor information $\bar{I} \in \mathbb{R}^{n}$, where *n* is the number of sensors. The context of the robot $\check{I} \in \mathbb{R}^{m+o}$ is included because it could influence the correct interpretation of the given touch instructions. In our implementation, the context is given by the current positions of the *m* motors and accelerometer values ($o = 2$) which represent the robot's orientation with respect to the gravity. The output is a robot response $M_* \in \mathbb{R}^{m}$, in the form of rotation angles of the motors.

The output of the algorithm is defined as $M_* = \sum_{i=1}^{E} \omega(I_*, I_i) M_i$, where *E* is the number of examples in the database, I_* is the current input from the sensors (touch and context), I_i is the *i*-th example input, M_i is the *i*-th example posture change, and $\omega(I_*, I_i)$ is a kernel function, defined as

$$\omega(I_*, I_i) = \begin{cases} 0 & if\, \exists s: s \in \Psi_i \wedge s \notin \Psi_* \\ \dfrac{\prod_{s \in \Psi_i} \bar{I}_*^{(s)} / \bar{I}_i^{(s)}}{1 + \sqrt{\|\check{I}_* - \check{I}_i\|_2^2 + \sum_{s: \notin \Psi_i} (\bar{I}_*^{(s)})^2}} & otherwise \end{cases} \qquad (1)$$

with $\bar{I}_i^{(s)}$ denoting the input from the *s*-th tactile sensor and Ψ_i the set of sensors pushed in the *i*-th example, i.e. $\Psi_i = \{s: \bar{I}_i^{(s)} > 0\}$ (and similarly $\Psi_* = \{s: \bar{I}_*^{(s)} > 0\}$). This kernel assures that when the sensors are pushed further the joint rotation angles are larger. Its denominator has the role of reducing the influence of examples which contexts $\check{I}_i$ differ strongly from the current context $\check{I}_*$. The condition imposes to ignore all examples that contains touch sensors which do not belong to the set of the currently pushed sensors Ψ_*.

The output of the algorithm M_* is used to modify the motion. In particular, the motion of the robot is defined as a set of keyframes F_k, which specify the positions of the motors for a certain time t_k of the motion (positions for times $t_k < t < t_{k+1}$ are obtained by linear interpolation of F_k and F_{k+1}). After selecting a time instant *t* on a GUI, the user can touch the robot, and modify its posture at time t from F_t to $F_t + M_*$ by simple touches.

3. EXPERIMENT

To verify the feasibility of the approach, we requested four people who never used the TbT system to develop a motion consisting of the first half of *Algorithm Taisō*, a dance from a Japanese TV show. The users were provided with a video of the dance motion, which they used as a reference of the dance movements and for deciding whether or not the robot motion they each created was satisfactory.

3.1 HARDWARE

The experiments were performed using a robot capable of recognizing touch, M3-Neony[5], a small sized humanoid robot equipped with a high number of tactile sensors over its whole body surface. M3-Neony features 22 servomotors, 90 tactile sensors (shown in Fig. 2), 3 accelerometers, 2 gyroscopes, 2 cameras and 2 microphones. The tactile sensors are based on photointerruptors which translate the changes of the force applied to the robot into changes in the light received by the phototransistors, as shown in Fig. 3. The experimental setup can be seen in Fig. 4. In particular, a pedal is used to notify the robot that its tactile interpretation is wrong, and allows the user to start providing the correct interpretation, which is stored by the robot as a new (I_i, M_i) example of the database.

Fig.2

Fig.3

Fig.4

3.2 RESULTS

All of the users who participated in the experiment performed the task with the proposed system without difficulties, or need for assistance, showing that the system is intuitive for first time users. The development times for the four users were respectively 221, 222, 176 and 476 minutes. Direct inspection of video recordings of the interaction shows that the users spent a lot of time, beyond our expectations, on stabilizing the robot. This infers the importance of applying self-stabilizing techniques[6], even for the development of quasi-static motions. Future works will incorporate automatic stabilization, in order to ease the motion development for inexperienced users.

An advantage of our system is, as previously stated, the possibility of studying the features of the way users employ touch in the context of motion teaching. One of the main features that can be noticed by direct data inspection is the sparsity in the touch instruction meanings taught by the users. In practice, users mainly touched few sensors and associated them with the movement of few motors. Gini index was used to evaluate this sparsity quantitatively. The Gini index is a measure of statistical dispersion, ranging from 0 (equal distribution) to 1 (distribution concentrated at a single point), usually employed as a measure of inequality or sparsity[7]. Formally, given a vector $f = [f_1 \cdots f_n]$, its Gini index can be computed by taking the following two steps:

1. Reorder the values of f in increasing order, i.e. define $g = [f_{i1}, f_{i2}, f_{in}]$ with $i_j, 1 \leq j \leq n$ permutation of $1 \cdots n$ such that $|f_{i_j}| < |f_{i_k}|\ \forall 1 \leq i < j \leq k$

2. Compute the index as $1 - \frac{2\sum_{l=1}^{n}|g_l|(n-l+1/2)}{n\sum_{l=1}^{n}|g_l|}$

Table 1 reports the mean of the Gini index for the sensor vectors $\bar{I}_i$ and the motor vectors M_i, averaged over the whole set of examples $1 \le i \le E$.

Table1

This high Gini index values suggest that the mapping between sensor inputs $\bar{I}_i \in \mathbb{R}^n$ and motor outputs $M_i \in \mathbb{R}^m$ could be sparse as well. To verify this hypothesis, we approximate the mapping with a linear function, and study the effect of enforcing sparsity on the mapping. More specifically, for a given example e, let us consider all of the preceding $M_i, 1 \le i \le e \le E$. Let us train a linear predictor $B_e \in \mathbb{R}^{m \times n+1}$ such that $M_i \approx B_e[1 \quad \bar{I}_i]^T \forall i \le e$. Let us set the *k*-th row of $B_e^{(k)}$ as $\left[b_e^{(k)} \quad \beta_e^{(k)}\right]$, with $b_e^{(k)} \in \mathbb{R}$ and $\beta_e^{(k)} \in \mathbb{R}^n$ as minimizers for the cost function

$$\frac{1}{2e}\sum_{i=1}^{e}\left(M_i^{(k)} - b_e^{(k)} - \beta_e^{(k)}\bar{I}_i\right)^2 + \lambda\left[(1-\alpha)\frac{1}{2}\left\|\beta_e^{(k)}\right\|_2^2 + \alpha\left\|\beta_e^{(k)}\right\|_1\right] \quad (2)$$

where $M_i^{(k)}$ is the rotation for the *k*-th motor in M_i, $\|.\|_2$ denotes the Euclidean norm, $\|.\|_1$ the ℓ_1 norm, and $\alpha, \lambda \in \mathbb{R}$ are constants.

When $\alpha \to 0$, this minimization resembles a classic linear regression and when $\alpha \to 1$ the minimization favors sparsity[8]. We note, however, that increasing α does not necessarily bring a sparsity enforcement measured by any sparsity measurement criteria. This appears very clearly by observing Fig. 5, which reports the Gini index, for different values of α, of the rows of $B_e^{(k)}$. In particular, Fig. 5 shows the Gini index of the rows $B_e^{(k)}$, averaged over the rows k (corresponding to the motors) of each predictor B_e, averaged over all of the predictors $B_e, 1 \le e \le E-1$. We notice that an increase of α initially causes a decrease of the Gini index. The index starts to slowly increase only for $\alpha > 0.43$, and has an abrupt increase for $\alpha > 0.98$.

Fig.5

Similarly, the Gini index of the predicted motor responses $p_{e+1} = B_e \begin{bmatrix} 1 \\ \bar{I}_{e+1} \end{bmatrix}$, averaged over the examples $1 \leq e \leq E - 1$, alternately increases and decreases until α reaches 0.98, which is when the index starts increasing very rapidly, as visible in Fig. 6. The average prediction error, plotted in Fig. 7, is defined for each predictor B_e as

$$\epsilon_{e+1} = \left\| M_{e+1} - B_e \begin{bmatrix} 1 \\ \bar{I}_{e+1} \end{bmatrix} \right\|_2 \tag{3}$$

It can be seen that this error starts to decrease at α>0.98. The fact that where the increase of the actual sparsity measured by Gini index occurs, which is at α>0.98, corresponds to where the strong decrease in the error starts, provides support to our hypothesis that increasing the sparsity of the predictor can lead to increases in the predictor performances.

Fig.6

Fig.7

The response to a tactile instruction, M_e, i.e. the variation of the motor angles due to touch, clearly lies in $\mathbb{R}^m$, where m is the number of motors, or more precisely in a smaller space $C \subset \mathbb{R}^m$ that is defined by the robot's physical constraints. In a previous work[9] it was shown experimentally that the responses can actually be constrained to a linear subspace of C. This is similar to the fact, often observed in literature, that many tasks can be accomplished by movements that lie in a small linear subspace of all the possible movements[10].

More interestingly, DallaLibera, et al[9] verified that if we know some part of the task, concretely some keyframes $F_1 \cdots F_k$ assumed during the task, and we compute the linear subspace $S = span < F_1 \cdots F_k >$, then with good probability the responses to tactile instructions lie in this linear subspace. This fact can be used in the following way. Given a tactile input $\bar{I}_i$, the estimated desired movement is computed (this phase corresponds, in our simple linear regression study, in computing $p_{e+1} = B_e \begin{bmatrix} 1 \\ \bar{I}_{e+1} \end{bmatrix}$).

Then, since we know that there is a good probability that the desired movement lies in S, we project p_{e+1} on S. In particular, let us define the matrix $\Gamma_e = [F_1 \cdots F_{k'}]$ of the frames available before the couple $(\bar{I}_{e+1}, M_{e+1})$ is taught, and a set of multipliers ρ_{e+1} such that $p_{e+1} \approx \Gamma_e \rho_{e+1}$. Let us then choose ρ_{e+1} as the minimizer of

$$||p_{e+1} - \Gamma_e \rho_{e+1}||_2^2 + \gamma ||\rho_{e+1}||_1 \quad (4)$$

where γ is used to enforce the sparsity[11] in the coefficients that multiply the frame values.

As was done before, we can use the Gini index to study the effect of γ on the actual enforced sparsity. In the first instance, we can analyze its effect on ρ_{e+1}. Fig. 8 reports the Gini index, averaged over $1 \le e \le E - 1$, of the coefficient vector ρ_{e+1}. We notice that by increasing γ, the Gini index of ρ_{e+1} monotonically increases up to γ=5.8, and then it starts to decrease very slowly. The Gini index, averaged over $1 \le e \le E - 1$, of the projected prediction $q_{e+1} = \Gamma_e \rho_{e+1}$ is plotted in Fig. 9. Also in this case, the sparsity monotonically increases until γ=5.8, when it then begins to decrease very slowly. The average error for the projected prediction, computed as the average over $1 \le e \le E - 1$ of

$$\delta_e = ||M_{e+1} - \Gamma_{e+1} \rho_{e+1}||_2 \quad (5)$$

is reported in Fig. 10. We notice that the best predictions are obtained for strictly positive γ (the minimum error is obtained for γ=3.52), i.e. we see that enforcing sparsity (by non-zero γ) to some extent can improve our estimation of the desired motor response.

Fig.8

Fig.9

Fig.10

4. CONCLUSIONS AND FUTURE WORK

In this paper, we presented a system which allows teaching whole body motions to humanoid robots by tactile interaction. Conceptual aspects of the exploitation of touch for motion development were briefly discussed, and a practical system implementation, which employs a small-sized robot equipped with 90 tactile sensors over the whole body, was briefly introduced. This system has a two-fold role. On the one side, it allows inexperienced users to develop robot motions. On the other side, it allows studying the way users employ touch to intuitively communicate with robots.

As an example of the possible analysis that can be done on the meaning associated to touch instructions, this paper reported a study on the importance of sparsity in the mapping between pressed sensors and desired robot movements. First, quantitative measures of the sparsity of the input signals (pressed touch sensors) and of the output signals (motors that should be moved) were reported. Successively, the possibility of improving the mapping from tactile instructions to motor movements by imposing sparsity was investigated. Finally, the idea of using the frames of the motion itself to improve the prediction on the robot movement was studied. In particular, it was shown that if we project our prediction on the linear subspace defined by few frames (again, by imposing sparsity in a coefficient vector), then the prediction of the robot movement desired by the user can be improved.

This analysis provides us with important criteria for the design of new algorithms for tactile interpretation. In particular, it tells us that by ensuring sparsity, performance can be greatly improved. For instance, if a neural network is employed for mapping tactile instructions to robot responses, then using rectifying neurons[12], which assure sparsity, appears to be a good choice. Future works will deal with the extension of the analysis reported here, and the design of a better performing algorithm for tactile interpretation based on this analysis.

REFERENCES

1. Righetti L, Ijspeert A (2006) Design methodologies for central pattern generators: an application to crawling humanoids. In: Robotic Science and Systems 2006, pp 191–198.
2. Inamura T, Nakamura Y, Ezaki H, et al. (2001) Imitation and primitive symbol acquisition of humanoids by the integrated mimesis loop. In: Proceedings of the 2001 IEEE International Conference on Robotics and Automation, pp 4208–4213.
3. Nakaoka S, Nakazawa A, Yokoi K, et al. (2003) Generating whole body motions for a biped humanoid robot from captured human dances. In: Proceedings of the 2003 IEEE International Conference on Robotics and Automation, pp 3905–3910.
4. Hersch M, Guenter F, Calinon S, et al. (2008) Dynamical system modulation for robot learning via kinesthetic demonstrations. IEEE Transactions on Robotics, 24(6):1463–1467.
5. Minato T, DallaLibera F, Yokokawa S, et al. (2009) A baby robot platform for cognitive developmental robotics. In: Workshop on ”Synergistic Intelligence” at Proceedings of the 2009 IEEE International Conference on Intelligent Robots and Systems.
6. Kuroki Y, Blank B, Mikami T, et al. (2003) Motion creating system for a small biped entertainment robot. In: Proceedings of the 2003 IEEE International Conference on Robotics and Automation, pp 1394–1399.
7. Zonoobi D, Kassim AA, Venkatesh YV (2011) Gini Index as Sparsity Measure for Signal Reconstruction from Compressive Samples. IEEE Journal of Selected Topics in Signal Processing, 5(5):927-932.

8. Friedman JH, Hastie T, Tibshirani R (2010) Regularization paths for generalized linear models via coordinate descent. Journal of Statistical Software, 33(1):1–22.

9. DallaLibera F, Basoeki F, Minato T, et al. (2011) Teaching by touching: Interpretation of tactile instructions for motion development. In: Proceedings of the 2011 IEEE International Conference on Intelligent Robots and Systems, pp 3480–3487.

10. Lim B, Ra S, Park FC (2005) Movement primitives, principal component analysis, and the efficient generation of natural motions. In: Proceedings of the 2005 IEEE International Conference on Robotics and Automation, pp 4630–4635.

11. Lee H, Battle A, Raina R, et al. (2007) Efficient sparse coding algorithms. In: Advances in Neural Information Processing Systems, pp 801–808.

12. Glorot X, Bordes A, Bengio Y (2010) Deep sparse rectifier neural networks. In: Proceedings of the 14th International Conference on Artificial Intelligence and Statistics. vol. 15.

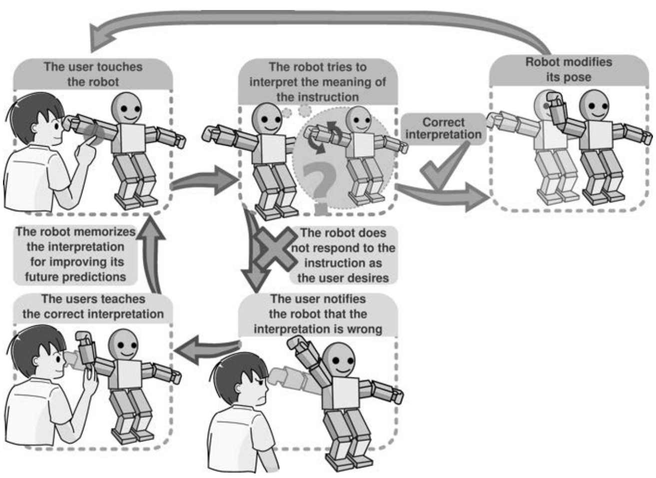

Fig.1. Teaching by Touching motion development process.

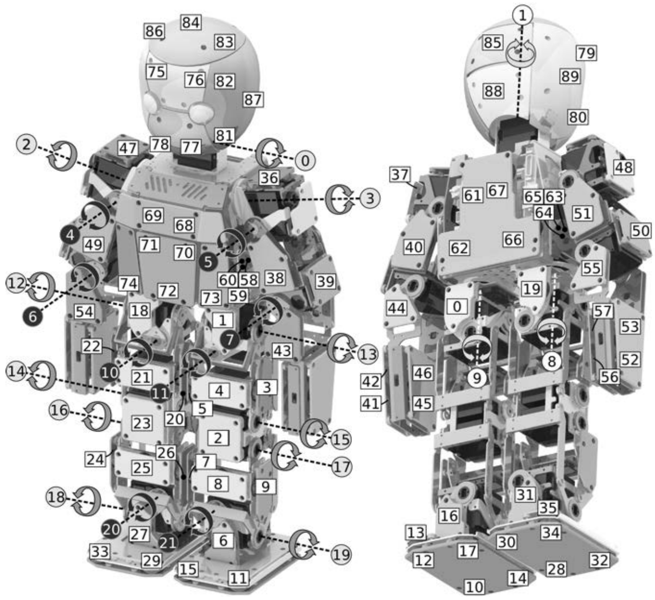

Fig.2. Diagram of M3 Neony's motors and sensors.

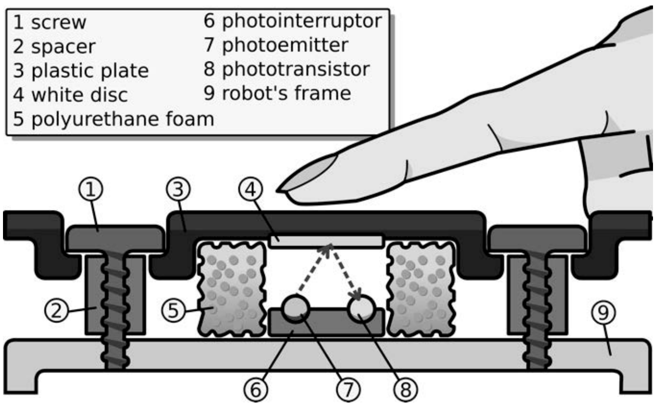

Fig.3. Schema of the tactile sensors of M3-Neony.

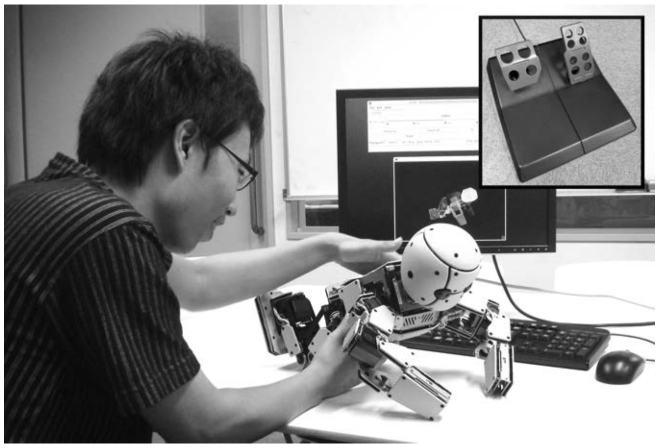

Fig.4. Experiment environment, pedal is shown in inset.

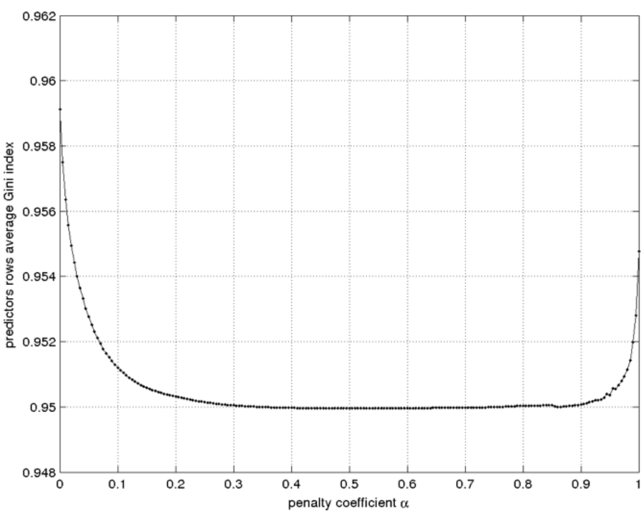

Fig.5. Average Gini index of the rows $B_e^{(k)}$ for different values of α.

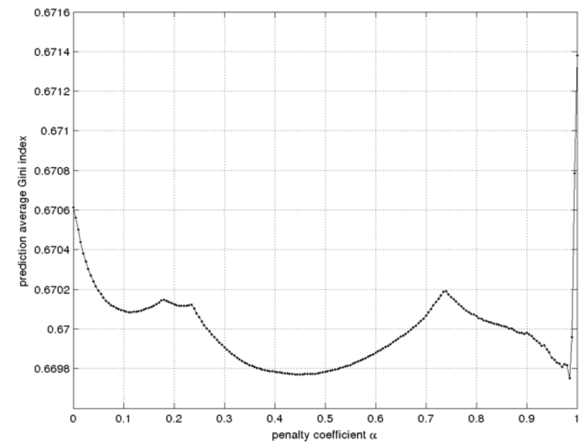

Fig.6. Average Gini index of the predicted motor response p_{e+1} for different values of α.

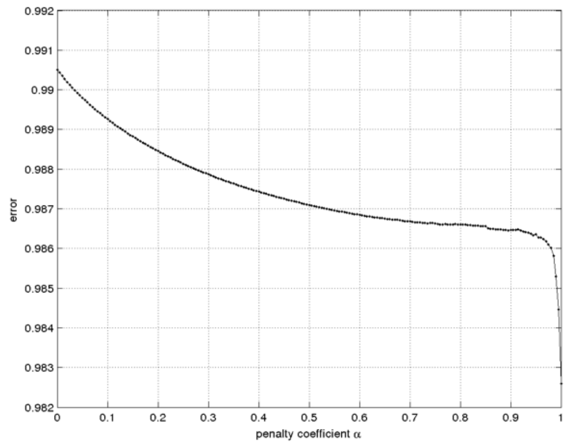

Fig.7. Average error ε_e for different values of α.

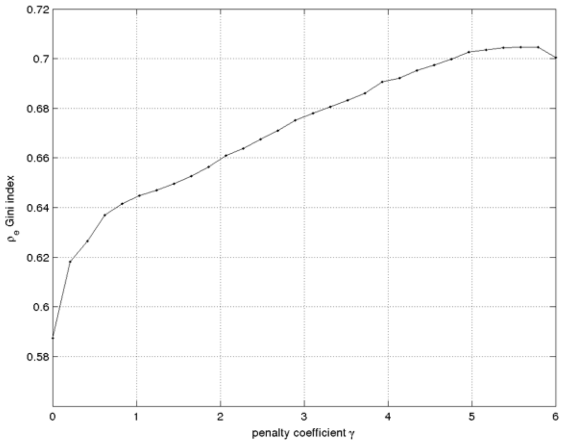

Fig.8. Average Gini index of the coefficients ρ_{e+1} for different values of γ.

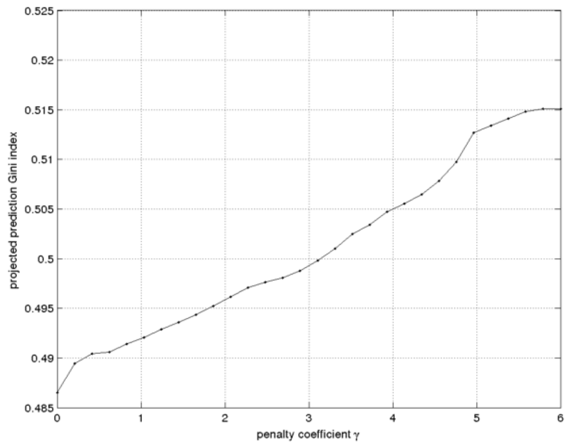

Fig.9. Average Gini index of the motor responses q_{e+1} for different values of γ.

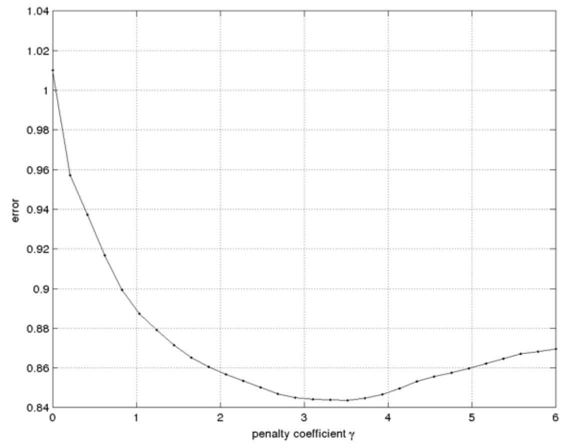

Fig.10. Average error δ_e for different values of γ.

Table 1. Gini Index

	A	B	C	D
$avg_{1\leq i\leq E} < G(\bar{I}_i) >$	0.96	0.90	0.98	0.92
$avg_{1\leq i\leq E} < G(M_i) >$	0.82	0.78	0.87	0.70