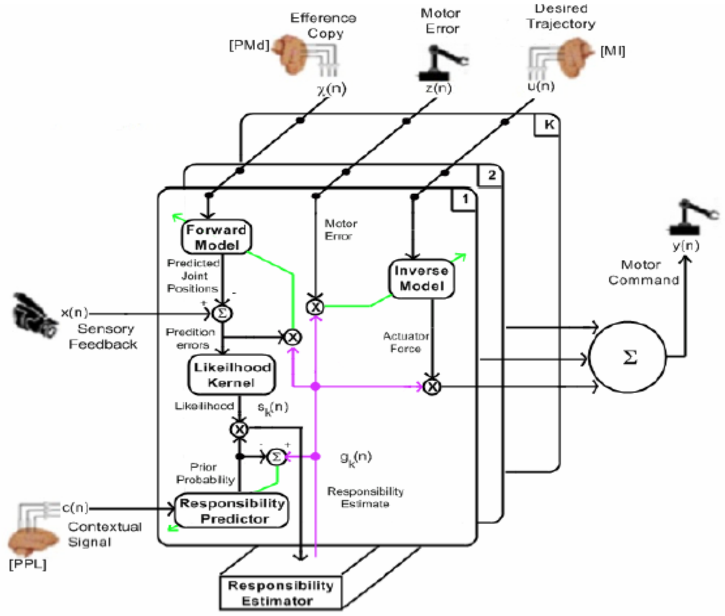
A report on "A New Architecture for Deriving Dynamic Brain-Machine Interfaces - José Fortes, Renato Figueiredo, Linda Hermer-Vazquez, José Príncipeand Justin C. Sanchez "

Soham Chakraborty

Summary

The paper proposes a new architecture for the development of an advanced Brain Machine Interface (BMI) that will help people with motor disabilities regain independent motor control. The software in existing BMIs (which allow the control of robotic devices) typically use computational models (eg linear adaptive filters and neural networks) which have to be trained using the motion of the users' limbs. This is not feasible in cases where the user is not able to control his/her limbs (such as people who are paralyzed). The DDDBMI architecture envisages the use of computational models that mimic the functions of the brain cortices, involved in motor control ,to overcome some the problems facing current BMIs.

The new architecture is based on a concept called MPFIM (Multiple Pair Forward Inverse Models), proposed by Wolpert and Kawato. *The MPFIM consists of multiple pairs of models, each comprising a forward model (for movement planning in the premotor cortex) and an inverse model (for movement execution in the primary motor cortex)* [1]. Based on real time feedback from sensors these model pairs are used to generate motor control inputs. A diagrammatic representation of this is given below.



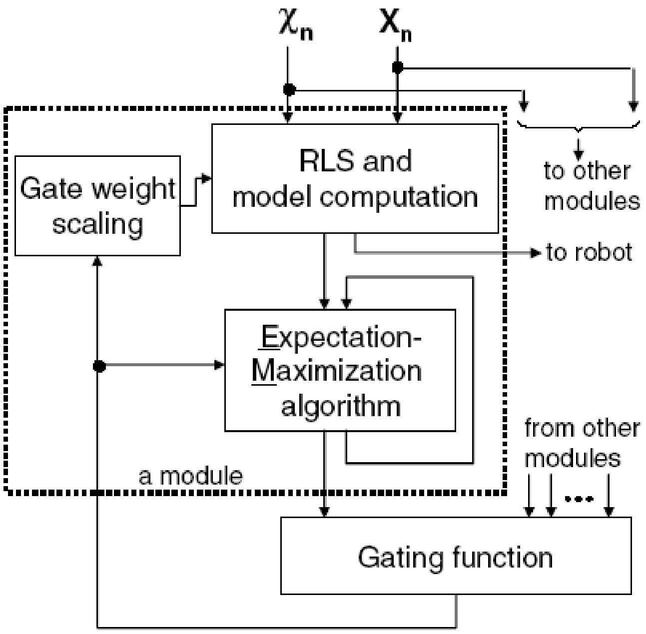
---

[1] Page 3 of LNCS 3993 - A new architecture for deriving dynamic brain machine interfaces - forte

Each of the K planes represent a model pair. The left half of a plane depicts the Forward model (along with the Likelihood model and the Responsibility Predictor which provide a scheme to select the Forward model), while the right half is a representation of the Inverse model. The scheme for selecting a Forward model involves computing the error in each model (using Likelihood model) and weighting them across all the models (using Responsibility Predictor). The Inverse model receives its input (the desired movement trajectory) from the premotor cortex. The sensory feedback mechanism (depicted by the video camera) consists of a camera and object recognition software. The outputs from all or a subset of the model pairs are combined to generate the control signals for the robotic arm.

To estimate the number of parameters that need to be updated in real time, we consider a system with 100 model pairs. *To predict the next position, one needs to use 500 milliseconds of neuronal firing data (which are binned every 50 ms); therefore, the number of parameters is given by ten times the product of the number of neurons by the number of outputs. For 100 neurons this results in M=3,000 parameters per linear model, leading to a total of 300,000 parameters total for 100 models* [1]. Current generation workstations are inadequate to the meet the computational demands of the proposed DDDBMI system. Therefore, such a system has to be implemented in a grid-computing environment.

To gain a better understanding of the computational processes involved consider the simplified schema of the DDDBMI architecture given below.



Let K denote the number of model pairs (represented by a module in the diagram) and N represent the number of input samples in a given time window, obtained from the sensory feedback x(n). The vector X_r represented the array of

[1] Page 5 of LNCS 3993 - A new architecture for deriving dynamic brain machine interfaces - forte

N samples obtained in the r th time window. Similarly the vector $\varkappa_n$ represents N samples of the efference copy signal obtained in the r th time window. Computation starts when $\varkappa_n$ and X_r are supplied to all the K modules as input. Each of the modules produce 2 outputs. One of them is combined with similar outputs of the other models to produce 3 co-ordinates for the robot device. The output is supplied to the gating function which uses this information to determine which of the modules will be used to process the future inputs. With the exception of the gating function all the calculations are local to a module.

Analysis

The Dynamic Data Driven aspect of the BMI is implemented by the concept of MPFIM which uses the input from a variety of sources (brain cortices, motion sensors) to dynamically adapt the parameters of the models and decide which models to use(which are essentially neural networks or linear filters).

The DDDBMI architecture is designed to dynamically select the model pairs whose combined outputs produce the required co-ordinates which serve as the input for the robot control device. As mentioned earlier all the inputs to the system are broadcast to each of the modules (regardless of whether their outputs will be used in computing the input to the robot control device). Note, a module refers to a pair of Inverse and Forward models. Each module in turn produces two outputs - one for calculating the robot control device input and other is fed into the gating function. The robot control signals are produced by the Inverse model while the output of the Forward model is fed to the gating function.

The Forward model portion contains of two sub-components - the likelihood kernel which locally computes the error in the model and the responsibility predictor which assigns weights to this error. This weighted error is then fed to the gating function which then decides which of the model pairs would be used in the next cycle. Thus the computation is dynamically data driven because in each cycle the gating function selects on the fly (based on the data it receives from each module) the models that would be used in the next cycle and assigns weights to their outputs.

The above architecture is primarily designed to be a test bed to help address the design challenges facing the development of BMIs. Since computational demands of the DDDBMI system can, at present, only be fulfilled by grid computers, a practical realization of this architecture would have to wait until a more portable hardware/software platform for its implementation becomes available.