Supplementary Materials for

Estimating human movements using memory of errors

Daqi Dong, Stan Franklin, and Pulin Agrawal
The University of Memphis, TN, U.S.A.
ddong@memphis.edu, franklin@memphis.edu, pagrawal@memphis.edu

In this document, we provide supplementary information for the paper, "Estimating human movements using memory of errors", submitted to BICA 2015. We first introduce LIDA[*], a systems-level cognitive model, and its Sensory Motor System (SMS). We then introduce a design for embedding an estimation process into the SMS, which process is implemented using our modified Kalman filter (see Section 3.2 in the paper). Finally we provide the details of a LIDA-based agent; by using that agent, we are able to test the estimation process of our new model (See section 4 in the paper).

1 LIDA and the SMS

The LIDA model is a systems-level cognitive model (Franklin, Madl et al. 2014). It implements and fleshes out a number of psychological and neuropsychological theories, but is primarily based on Global Workspace Theory (Baars 1988, Baars 2002). The model is grounded in the LIDA cognitive cycle (see Figure 1). The simulated human mind can be viewed as functioning via a continual, overlapping, sequence of these cycles. Each cognitive cycle consists of three phases: 1) The LIDA agent first senses the environment, recognizes objects, and builds its understanding of the current situation. 2) By a competitive process, as specified by Global Workspace Theory (Baars 1988), it then decides what portion of the represented situation should be attended to and broadcast to the rest of the system. 3) Finally, the broadcast portion of the situation supplies information allowing the agent to choose an appropriate action to execute, and modulates learning.

Two aspects of action are represented in LIDA as the distinct processes of action selection and action execution. Action selection has been described in previous work (Franklin, Madl et al. 2014), while recently we have simulated action execution using a Sensory Motor System (SMS) (Dong and Franklin 2015). The SMS responds by transforming a high-level desired action into an executable low-level action sequence of motor commands, and then executes them. The SMS is implemented by two modules in LIDA: Sensory Motor Memory and Motor Plan Execution depicted in the bottom left corner of Figure 1.

Two other LIDA modules, Action Selection and Sensory Memory, provide relevant information—a selected behavior and the sensory data through a dorsal stream channel[†] respectively—as separate inputs to the SMS. The selected behavior is a data structure resulting from the preceding action selection in the LIDA cognitive cycle. It comprises three components: a context, an action[‡], and a result. With some reliability, the result is expected to occur when the action is taken in its context. The SMS sends out motor commands as its output to the environment. Note that in Figure 1, the Sensory

[*] For historical reasons LIDA stands for Learning Intelligent Distribution Agent.
[†] In LIDA, a dorsal stream channel directly passes sensory data from the sensory memory to the action execution process.
[‡] In this context, this term refers to a component of a behavior. This differs from the general usage, such as in the phrase "action execution". In this paper, we use "action" in the general sense, while "the action of a behavior" refers to a particular component of that behavior.

Motor Learning channel, which is issued by the Global Workspace and sent to the Sensory Motor Memory, has not yet been modeled in the SMS (Franklin, Madl et al. 2014, Dong and Franklin 2015) because the current work does not address the learning process. Future work is expected to do so.

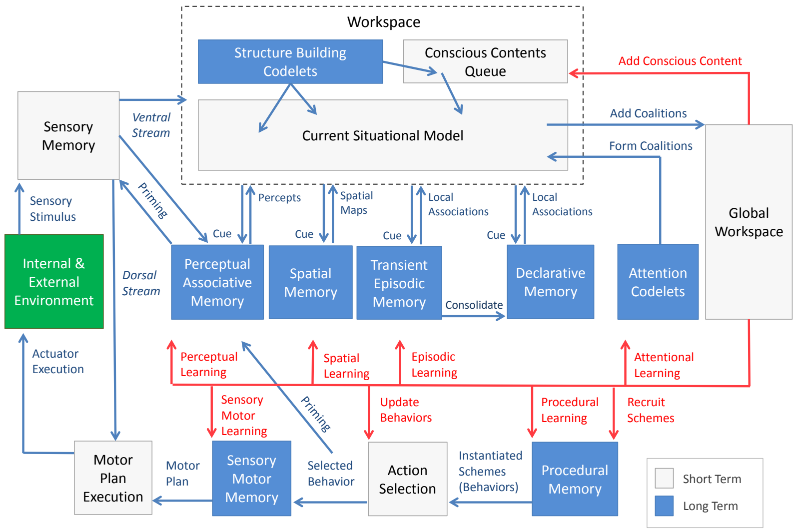


Figure 1: LIDA Cognitive Cycle Diagram

2 Adding an Estimation Process into the SMS

As shown in Figure 2, the original SMS generates low-level motor commands to actuators within the environment. It is driven by 1) a high-level goal-directed action provided by LIDA's Action Selection module, and 2) the sensory data perceived from LIDA's Sensory Memory. We added an estimation process into the SMS of LIDA to implement our modified Kalman filter (See Section 3.2 in the paper). The inputs to this estimation process comprise a copy of motor commands (the efference copy) together with real sensory data. The newly added estimation process in the SMS provides estimated sensory data to the SMS's original "motor command generation" component. Inside this new process, we implement two sub-modules, an internal model and a correction process, which accomplish the prediction and correction steps of the modified Kalman filter respectively. See Eqs. (1)~(3), (5), and (8) introduced in the paper for detailed computational expressions of the prediction and correction steps.

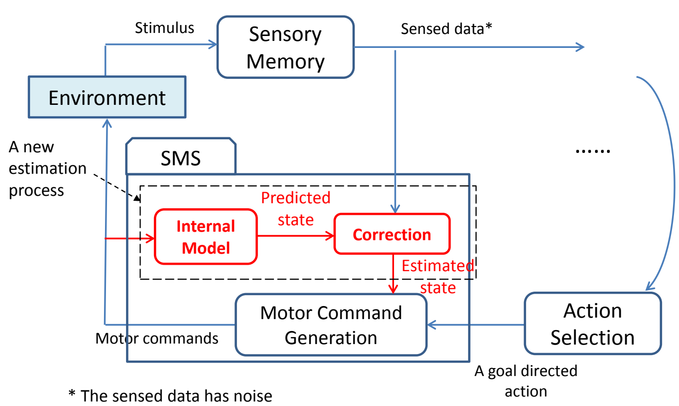


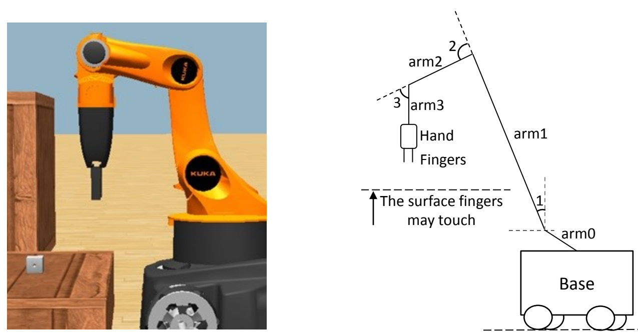
Figure 2: An estimation process in the Sensory Motor System (SMS) of LIDA

We have observed that the motor commands sent out to the actuators need time to be executed, which means that at a given time, the motor commands and the sensory data input into the estimation process may not be consistent. To deal with this, we have created a FIFO (First In First Out) queue for storage of the input motor commands in the internal model and set the queue's length to one. Thus, there is a one-step delay between the motor commands and the current sensory data used for the estimation.

3 Experimental Setup

We use a software robot simulation, a robot controller, and a virtual experimental environment to execute actions for testing the added estimation process. Specifically, these are the youBot, the LIDA Framework (Snaider, McCall et al. 2011), and Webots (www.cyberbotics.com) respectively. We consider this robotic simulation to be a LIDA-based software agent.

The youBot is a software robot. Its actuators are a mobile base, an arm, and two grippers. As shown in Figure 3 (a) and (b), the youBot arm comprises multiple segments linearly connected by joints; the end segment plays the role of a hand, and the two grippers attached to it serve as fingers. The LIDA Framework is an underlying computational software framework, designed according to the LIDA Model, and implementing most of that model's proposed modules. This Framework is used to construct the LIDA-based controller for the youBot. Webots is a mobile robot simulation software package. It offers an environment, a set of ready-made sensors and actuators, and programmable controllers controlling robots living in its virtual world. We use Webots as an experimental environment in which to manipulate the youBot in order to test a computational SMS newly embedded with an estimation process.



(a) (b)

Figure 3: (a) A snapshot of the youBot, and (b) its schematic diagram

In the LIDA Framework's Environment module (see Figure 1), we have added noise to joints 1 ~ 3 (see Figure 3 (b)) by randomly setting their angles with a normal distribution: the mean is the actual measure of the angle, and the STD is 0.1 degrees. In this way, uncertain sensory data is sent to the estimation process. We use the added process to estimate the youBot's finger positions during executing an action in the above uncertain situation.

Regarding the action the agent executes in the experiment, we simulated a lifting movement using the youBot. In this context, lifting refers to an action in which the agent grips an object and moves it upwards. Furthermore, we created a program to play the role of a simulated "experimenter", which is able to send the agent a command to stop its lifting. Because the agent always starts the lifting when the simulation begins, a starting command is not prepared by the "experimenter". In detail, this program is designed to send a stop command after a certain number of virtual time units[§] have passed, to the Environment module (see Figure 1) of the agent's controller. On the agent's side, first it senses the stop command from the Environment module as an input to its Sensory Memory; and then this command is sent to the Current Situational Model (CSM) as part of the agent's current understanding of the environment. The command is represented as a stop node in the CSM. A special Attention Codelet is implemented to attend to this stop node, and form it into a special data structure, a coalition (Baars 2002, Franklin, Madl et al. 2014), sending the coalition then to the Global Workspace (GW). In the GW, the coalition containing the stop node might win a competition among different coalitions, and thus be broadcast to the rest of the system as the conscious content. There are multiple schemes stored in Procedural Memory (PM); a scheme is a template of a behavior (see Section 1 for the details of a behavior in LIDA), and is supposed to be instantiated to a behavior. We prepared a special scheme that (1) will be recruited by the arrival of the stop node in the conscious content, and (2) contains an action component for executing a stop command. Then when the stop node comes through the conscious content to PM, this scheme is chosen and instantiated into a behavior that has an action component for stop. Finally when this behavior arrives to the Sensory Motor System (SMS), the currently running lifting movement is stopped.

In the above discussion, we describe the different cognitive processes processing through multiple LIDA cognitive modules in a linear way in order to make the process more easily understood. But actually, each LIDA module acts independently and asynchronously with other modules in the Framework.

References

Baars, B. J. (1988). A cognitive theory of consciousness. New York, Cambridge University Press.

Baars, B. J. (2002). "The conscious access hypothesis: origins and recent evidence." Trends in cognitive sciences **6**(1): 47-52.

Dong, D. and S. Franklin (2015). "A New Action Execution Module for the Learning Intelligent Distribution Agent (LIDA): The Sensory Motor System." Cognitive Computation.

Dong, D. and S. Franklin (2015). "A New Action Execution Module for the Learning Intelligent Distribution Agent (LIDA): The Sensory Motor System." Cognitive Computation: 1-17.

Franklin, S., et al. (2014). "LIDA: A Systems-level Architecture for Cognition, Emotion, and Learning." IEEE Transactions on Autonomous Mental Development **6**(1): 19-41.

[§] The agent executes at unit intervals in Webots virtual time.