Bottom-up Cognitive Analysis of Bionic Inspection Robot for Construction Site

Thomas Bock[a] , Alexey Bulgakov[b], Sergei Emelianov[b] and Daher Sayfeddine[c]

[a] Building Realization and Robotics, Munich Technical University, Germany
[b] Department of Mechatronics in Construction, Southwest State University, Russia
[c] Department of Mechatronics, South Russian State Polytechnic University, Russia
E-mail: thomas.bock@tum.de, a.bulgakow@gmx.de, esg@mail.ru, daher@live.ru

Abstract –

Artificial intelligence aims to make robots more adaptive and versatile depending on the surrounding operating atmosphere. It permits the autopilot of the robot to generate optimal trajectory with reference to the energy efficiency criteria and risk avoidance in order to lead to the end-effector or the working element of the robot to the desired position, thus to perform safely the assigned task.

One of the problems faced during the optimization of algorithms of the central autopilot is the overlearning, similar training sets, permanent operating conditions, which may lead to controversial result: non-adaptability of the robot. This can be clearly seen when training nonlinear model of neural network with exogenous inputs aiming to resolve the extrapolation of the movement function of a mobile agent (nontraditional desired trajectory). Hence the quick change in the operating conditions can lead to undesired outcomes.

In this paper we analyze robot control performance based on situational approach, described using discrete mathematics operators and state-base/ history base, time-base/action-base functions. The knowledge representation will be used to train auto-regressive neural network using situational time series.

Keywords –
Cognitive robot; Artificial intelligence; NARX neural network; Regression

1. Introduction

The successful integration of robotic solutions in the construction field and strictly depend on the site conditions and infrastructure. At construction site, many hindrance are there, which limit the robot maneuverability, operational speed and reaching. Due to the dynamicity in the construction site, the concept of “controlled environment” cannot always be maintained, which develop new challenges to the robot “perception” of the working atmosphere. The sudden changes can be challenging to the robot in terms of processing unit and speed, which may cause injuries to the robot. Changing processing capabilities of the robot only to fit the “if” case is not techno-commercial solution the risk and dynamicity specter is very broad. In view of the above, cognitive concept can be introduced.

By cognitive, we estimate the ability of the robot to learn and develop new movement technics while permanently analyzing the working atmosphere thus to change forecast future actions or to react on sudden changes in the working site.

Cognitive solutions are based on artificial intelligence. For instance, the neural network can be used for extrapolating the movement function and guide the robot to track recently generated trajectory. This is the case when inspecting a site, while construction actions are carried out above the robot. HSE guidelines are optimal for labors, but it cannot be used for robots. This HSE infrastructure is tackled by the cognitive ability of the robot, making him always “alert” to the changes in the working atmosphere.

2. Literature Review

Many literatures focus on the hierarchy, the organization and optimization of the autopilot. For bionic robot, the task is more difficult, since the control and movement tasks are more complex. This is due to the enormous data traffic, analysis and storage required. The adaptability or the morphological changes in the end-effector, the chassis of the robot and the central controller are more difficult to optimize from auto-operational point of view. For instance, tasks such as path planning has become vague and not clear, since path planning is limited to the kinetic aspect of the movement, where offline algorithms can play major roles. The outcome is global step planning before movement. In contrast, trajectory planning is more difficult task. It is related to the dynamic aspect of the movement, the analysis of the surrounding and the decision of action with reference to online algorithms. The result is a local step planning, which requires more computation power to resolve the complexity of the data. As it can be clearly noticed, cognitive is more about dynamic planning.

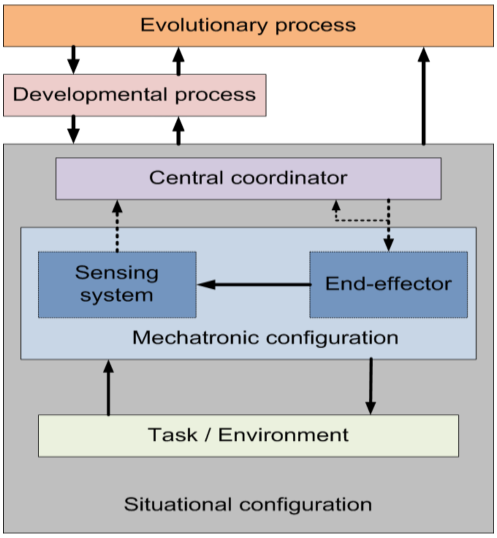

Figure 1. Bottom-up cognitive hierarchy of mechatronic/robotic mechanism

There are different methods to analyze a system. Wireframes, virtual Personas or scenarios, rapid prototyping and several more are not an option to be adopted for real-time planning. Cognitive approach is based on Bottom-Top analysis, where many subsystems are pieced together in such a way, thus to obtain more complex systems or approach. In light of that, the emergent output is controlling the behavior and integrity of the original subsystems. In other words, a significant attention is offered firstly and constantly to subsystems, then to the links between them and lastly to the output. As it may be noticed, the system analysis grows complex bottom-top hence understanding such system requires granulation process. Inspection is a dynamic task in unknown environment where everything is possible. Robots are frequently used in inspection and monitoring tasks such. Task failures are several: operational, power-related and mechanical. The operational reasons can be caused by human-operator in distant-controlled tasks or because of algorithmic challenges. The flow of work between human and robots in different automation strategies is an essential factor to succeed with the integration of the robotic solution. As depicted in fig.2, it is very clear that in "Lights-Out Automation" scheme, the interference of the human-operator in the robotic scope of work does not exist. Geographically, it is not necessary that the robot and operator be in different location. However, the operator is not responsible to monitor the work of the robot. This fact imposes high level of adaptation and perception of the robot.

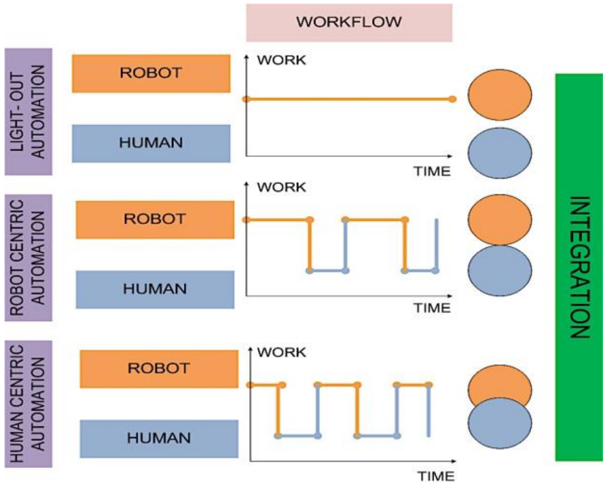

Figure 2. Workflow and integration between Human and Robot

3. Robotic Cognition

One of the main goals of robotics is to replace human being in dangerous tasks; the integration of adaptable intelligent machinery in construction sites encompasses delivering milestones outside the concept of project management in controlled environment. Tuning the robot based on case-to-case approach may cover resolve the planning difficulties. However, misjudging a scenario in the algorithm will lead to undesired results.

In view of that, the situational approach seems to be more adaptable and less risky as the robot knowledge database is epistemic. This terminology explains the ability of robot to do modal logic that formalizes knowledge based on situational/ historical approach.

3.1. Action Representation

The essential idea about cognitive robotics in the way of presenting the knowledge based and embedded it into the control system by giving more flexibility-authority trade-offs. While, the concept of probabilistic models and theories will remain in adoption, the major concentration of the cognitive science in robotics is about monitoring the changing word [1]. The function representing the changes can be based on calculus, or by comparing modal to non-modal, state-base/ history base, time-base/action-base functions and knowledge representations [2,3,4,5]. Hence, a single action of the robot has two obviously simultaneous derivatives: START- END combined with additional time argument and new fluent with the following successor state axiom [6]:

$$\boldsymbol{C_T(x,t,do(a,s))} \equiv \exists \boldsymbol{t'}(\boldsymbol{a} = \boldsymbol{S}(\boldsymbol{x},\boldsymbol{t'}) \wedge \boldsymbol{t'} \leq \boldsymbol{t}) \vee \boldsymbol{C_T}(\boldsymbol{x},\boldsymbol{t},\boldsymbol{s}) \wedge \neg\exists \boldsymbol{t'}(\boldsymbol{a} = \boldsymbol{E}(\boldsymbol{x},\boldsymbol{t'}) \wedge \boldsymbol{t'} \leq \boldsymbol{t} \quad (1)$$

Where $\boldsymbol{C_T} -$ is the time argument; $\boldsymbol{S} -$ start action; $\boldsymbol{E} -$ end action; $\boldsymbol{t'} -$ is the additional time argument; $\boldsymbol{x} -$ control task/ handled object; $\boldsymbol{a} -$ action and $\boldsymbol{s} -$ is the situation of sensed fluent value. It is worth to note that equation (1) include the additional time frame argument which does not encompass the failure time necessary for correction or repetitive actions.

3.2. Sensing

Sensing is the source of change database required in order to build idea about the surrounding word. While the situation calculus propose introducing a special fluent and axioms describing the truth-value correlated to the aspect of situation, other state-base/history base comparing algorithms seem to be more efficient in term of consumption of computation power and time. The combination of the two approaches lead to optimized hybrid algorithm, which we will adopt in later stage in this paper.

If the control task consists of tracking color based on histogram and geometric approach, then we can use the following a predicate defining the robot learning by executing recognition actions in different situation and generating binary results. The following can serve as demonstrative case:

$$\begin{aligned}
&\boldsymbol{R_s} = (\boldsymbol{senseC}(\boldsymbol{x}), \boldsymbol{s} \equiv \boldsymbol{C}(\boldsymbol{x}, \boldsymbol{\varrho}, \boldsymbol{s}); \\
&\boldsymbol{R} = \left(\vec{\boldsymbol{a}}.\boldsymbol{A}, \vec{\boldsymbol{b}}.\boldsymbol{1}, \boldsymbol{s}\right) \stackrel{\text{def}}{=} \boldsymbol{R_s}\left(\boldsymbol{A}, \boldsymbol{do}(\vec{\boldsymbol{a}}, \boldsymbol{s})\right) \wedge \boldsymbol{R}(\vec{\boldsymbol{a}}, \vec{\boldsymbol{r}}, \boldsymbol{s}); \\
&\boldsymbol{R} = \left(\vec{\boldsymbol{a}}.\boldsymbol{A}, \vec{\boldsymbol{b}}.\boldsymbol{0}, \boldsymbol{s}\right) \stackrel{\text{def}}{=} \neg\boldsymbol{R_s}\left(\boldsymbol{A}, \boldsymbol{do}(\vec{\boldsymbol{a}}, \boldsymbol{s})\right) \wedge \boldsymbol{R}(\vec{\boldsymbol{a}}, \vec{\boldsymbol{r}}, \boldsymbol{s}).
\end{aligned} \tag{2}$$

Where, $\boldsymbol{R_s} -$ the sense fluent value; $\boldsymbol{C} -$ is the colour function; $\boldsymbol{x} -$ is the tracked object; $\boldsymbol{\varrho} -$ is the RGBHV value of the colour; $\vec{\boldsymbol{r}} -$ is the vector or binary results or cognitive reasoning of the robot.

3.3. Knowledge and Decision

Similarly to the action representation and sensing, knowledge is also presented using special fluent K depending on situational approach. In this regard, if a parametric function F representing the object is known at a certain situation, then F is known in all other situations. Hence we can write:

$$\boldsymbol{K}(\boldsymbol{F}, \boldsymbol{s}) \stackrel{\text{def}}{=} \forall \boldsymbol{s'}.\boldsymbol{K}(\boldsymbol{s'}, \boldsymbol{s}) \supset \boldsymbol{F}[\boldsymbol{s'}] \tag{3}$$

The successor state axiom based on action update the knowledge base using the following model:

$$\begin{aligned}
\boldsymbol{K}\left(\boldsymbol{s''}, \boldsymbol{do}(\boldsymbol{a}, \boldsymbol{s})\right) \equiv \exists \boldsymbol{s'}.\boldsymbol{s''} &= \boldsymbol{do}(\boldsymbol{a}.\boldsymbol{s'}) \wedge \boldsymbol{K}(\boldsymbol{s'}, \boldsymbol{s}) \\
&\wedge [\boldsymbol{R_s}(\boldsymbol{a}, \boldsymbol{s'}) \equiv \boldsymbol{R_s}(\boldsymbol{a}, \boldsymbol{s})].
\end{aligned} \tag{4}$$

As it can be noticed, it is sufficient to know the action is a certain previous situation so the robot can start building on it. In other term, an action is accessible in following situation $\boldsymbol{s'}$ or $\boldsymbol{s''}$ only if the action satisfies the condition in equation (2).

Using equations (1) to (4), we can generalize that by giving sequence of actions performed by the robot, uniform parametric function, the cognition task become a regression approach [8] since it is based on previous situational incident $\boldsymbol{s_0}$.

3.4. Auto-Regression Task

As a matter of fact, neural network are widely used in regression task. While it can suffer for overlearning, long analogic and action comparison, neural network remains one of the most influential cognitive and reasoning approaches in robotics.

The aim of the prediction using NARX is to be passive mostly during successful recognition of the agent [9]. During this time NARX can be trained by the updated positions of the mobile agent represented by data pairs (input- Output). This vector is described in (5)

$$\begin{aligned}
&u(n) = (u_1(n), \dots u_K(n))', \\
&d(n) = (d_1(n), \dots d_L(n))' \quad n = 1..T
\end{aligned} \tag{5}$$

Where $u(n) -$ is the input set, $d(n) -$ is the output set, n represents the time, K is the number of perceptron in the input layer, L is the number of perceptron is the output layer.

The training of the NARX is achieved using BPTT because the output of the network is not fed into the tracking algorithm. As discussed earlier, the training is done passively following three steps.

The first step consists of calculating and discovering the status of the activation functions $x(n)$ of each perceptron starting from $u(n)$, $x(n-1)$ and $y(n-1)$ or the activation of the output layer if it is fed into a certain perceptron;

The second step includes the calculation of the back-propagation error of each perceptron starting from $n = T..1$, $x(n)$ and $y(n)$ for each instance of time n. this is achieved using the following system of equations (6)

$$\delta_j(T) = (d_j(T) - y_j(T))(\partial f(u))/\partial u|_{u=z_j(T)}$$

$$\delta_i(T) = [\sum_{j=1}^{L} \delta_j(T)\, w_{ji}^{out}](\partial f(u))/\partial u|_{u=z_i(n)} \tag{6}$$

$$\delta_j(n) = [\left(d_j(n) - y_j(n)\right)\sum_{i=1}^{N} \delta_i(n+1) w_{ji}^{back}]\,(\partial f(u))/\partial u|_{u=z_j(n)}$$

$$\delta_i(n) = [\sum_{i=1}^{N} \delta_i(n+1) w_{ji} + \sum_{j=1}^{L} \delta_j(n) w_{ji}^{out}](\partial f(u))/\partial u|_{u=z_i(n)}$$

Where $\delta_j(T) -$ is the backpropagation error of the output perceptron, $\delta_i(T) -$ is the back propagation error of the perceptron located in the hidden layer with activation $x_i(T)$, $\delta_j(n)$ and $\delta_i(n)$ are consequently the backpropagation error of the output perceptron and the one located in the hidden layer in an earlier time T layer and $z_i(n) -$ is the potential of each perceptron.
After finding the back propagation error, the weights connecting different perceptron are calculated using the following system (7)

$$w_{ij} = w_{ij} + \gamma \sum_{n=1}^{T} \delta_i(n)\, x_j(n-1)$$

$$w_{ij}^{in} = w_{ij}^{in} + \gamma \sum_{n=1}^{T} \delta_i(n)\, u_j(n)$$

$$w_{ij}^{out} = w_{ij}^{out}\gamma \begin{cases} \sum_{n=1}^{T} \delta_i(n)\, u_j(n) \; if\; j - is\; an\; output\; p \\ \sum_{n=1}^{T} \delta_i(n)\, x_j(n) \; if\; j\; is\; a\; hiden\; per \end{cases} \tag{7}$$

$$w_{ij}^{back} = w_{ij}^{back} + +\gamma \sum_{n=1}^{T} \delta_i(n)\, y_j(n-1)$$

Where $\boldsymbol{w_{ij}} -$ is the weight connecting the hidden perceptron, $\boldsymbol{w_{ij}^{in}}$, $\boldsymbol{w_{ij}^{out}}$ and $\boldsymbol{w_{ij}^{back}}$ are the input, output and feedback weights consequently, $\boldsymbol{\gamma} -$ is an incremental small value that is used during the minimization of the squared error.

3.5. Results

The NARX structure consists of a single time-series data pairs as was described earlier, which contains all the historical data about the movement of the agent. The hidden layer includes 10 perceptron and a single output is generated thereafter predicting the possible location of the agent. The structure model of the used NARX is shown is figure.3.

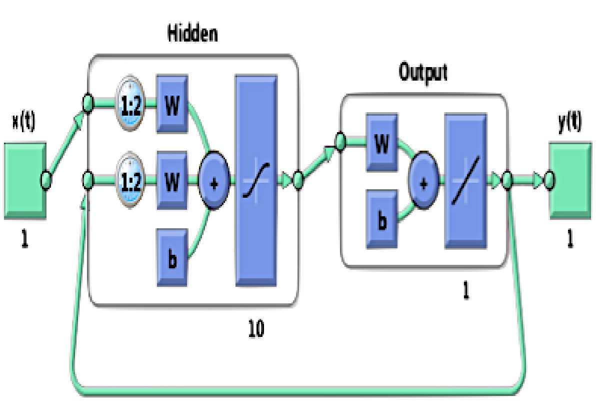

Figure 3. NARX structure model used to predict position

The training is achieved based on 1000 epochs and was concluded in one minute and eight seconds. The result is obtained based on minimal gradient 1^{-10}.

It is important to mention that the NARX acts like recurrent neural network with embedded memory. It allows the NARX to “remember” the output of the perceptron located in any layer by unfolding the dependencies of the forecasted series far longer than a conventional recurrent neural network. This shows an impact when forecasting nonlinear, aperiodic and unknown data sets.

In light of this, the prediction can be tested on different nonlinear functions conserving the periodicity, the damping factor and chaotic movement of the tracked agent.

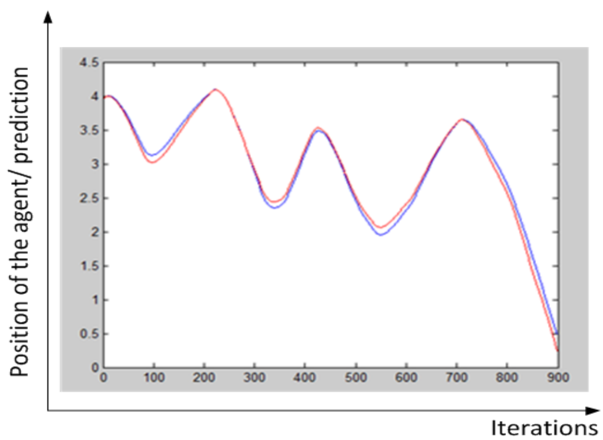

Figure 4. Simulation results

Simulation results of the predicting the next position of the agent are shown in figure.4. It is clear from the curves that the estimation (red) was very close to the real values presented in the time series (blue).

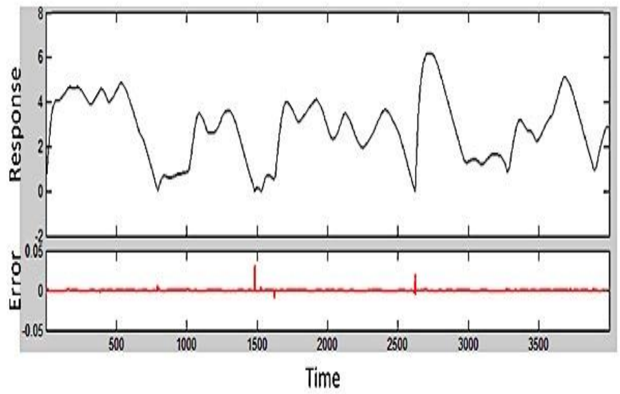

Figure 5. Response of the NARX with reference to the time series (black) and the back propagation error (red) over time

We can notice that the NARX was successful in estimating the next position of the dynamic agent, movement, which was described randomly using predesigned time series in based on magnitude and phase form (input output coupled pairs).

Although the regression approach could resolve the situational challenge theoretically, it is very challenging to adopt the same concept in very dynamic circumstances. This is explained as follows: a long operational life cycle will create massive arrays to be analyzed. This will cause delays in decision-making and render the cognitive central coordinator not optimal to use.

On the other hand the progression analysis aims to concentrate on the current scenario only. In progressing a database forward, the historical information will not of use. The other drawback of the progression is that it is not always possible [10].

The main question remains as if to categorize cognitive approach as:

- A projection task determining whether or not some condition will hold after a sequence of actions has been performed starting in some initial state (forecasting condition based on current action).
- A legality task: determining whether a sequence of actions can be performed starting in some initial state (forecasting action based on situation).

References

[1] M. Fichtner, A. Großmann, and M. Thielscher. Intelligent execution monitoring in dynamic environments. Fundamenta Informaticae, 57:371–392, 2003.

[2] F. Bacchus and F. Kabanza. Using temporal logics to express search control knowledge for planning. Artificial Intelligence, 116(1–2): 123–191, 2000.

[3] C. Boutilier, R. Reiter, M. Soutchanski, and S. Thrun. Decision-theoretic, highlevel agent programming in the situation calculus. In Proc. Of the AAAI-00 Conference, pages 355–362, 2000.

[4] H. Levesque and G. Lakemeyer. Handbook of Knowledge Representation Edited by F. van Harmelen, V. Lifschitz and B. Porter. pp. 869-886. 2008

[5] Baker, A. A Simple Solution to the Yale Shooting Problem. In H. L. R. Brachman and R. Reiter (Eds.), Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning, pp. 11–20. Morgan Kaufmann Publishers, Inc. 1989

[6] J. Pinto. Integrating discrete and continuous change in a logical framework. Computational Intelligence, 14(1), 1997

[7] F. Pirri and M. Romano. A situation-Bayes view of object recognition based on symgenons. In Proc. Of the Third International Cognitive Robotics Workshop, Edmonton, 2002.

[8] R. Reiter. The frame problem in the situation calculus: A simple solution and a completeness result for goal regression. In Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy, pages 359–380. Academic Press, New York, 1991

[9] S. Shapiro and M. Pagnucco. Iterated belief change and exogenous actions in the situation calculus. In Proc. Of the ECAI-04 Conference, pages 878–882, 2004.

[10] F. Lin and R. Reiter. How to progress a database. Artificial Intelligence, 92(1–2):131–167, 1997.