DSL for Automatic Control System Modeling of Technological Process

O. B. Ospanov, A.L Krasavin, D.L. Alontseva, A.V. Russakova
D. Serikbayev East Kazakhstan State Technical University/Department of Instrument Making and Automation of Technological Processes, Ust-Kamenogorsk, Kazakhstan
ospanovolzhas@gmail.com
alexanderkrasavin@mail.ru
dalontseva@mail.ru
arussakova@gmail.com

***Abstract* - This paper concerns the description of experimental system of the processes of automated control modeling. The system supports uniquely created DSL and allows describing signals and typed blocks of automation system.**

I. INTRODUCTION

Currently acute problem of increasing prices and the reduction of natural resources of non-renewable energy sources, together with negative impact of combustion products emission led to the fact that many research works aim at solving problems of optimization and increase of efficiency of energy usage [1-9].

Primarily it should be noted that most coal burning boilers do not have systems of automatic control of air-fuel mixture concentration. As a rule, feeders, regulating coal dust intake, are chosen empirically and are not regulated automatically, while airline unit has only one automatic control loop to maintain constant air quantity flow (when the flow amount is chosen empirically as well) [1-3]. Nozzles adjustment is usually performed manually, and in fact, empirical evidence had shown that worker's experience and skills could significantly influence the efficiency coefficient of a steam system. At the same time, we are perfectly aware that the efficiency of air-fuel mixture combustion with fixed features of applied fuel dust (coal grade, composition and moisture content in coal) highly depends on air–fuel mixture concentration. Besides, the properties of air fuel are changing during a time of a continuous operation of a boiler, thus, on-going adjustments of fuel mixture making system are ineffective to maintain stable efficiency of a boiler [4,5].

The analysis of attempts to apply the methods of automatic control of the fuel mixture making in boiler and in metallurgical industries, leads to the conclusion that these attempts have been made along two main directions: in some cases, the controlled parameter was the concentration of dust. In other cases – it was the concentration of carbon monoxide or oxygen in the combustion products. Accordingly, in both cases the process of automatic regulation provided with constancy in time of a measured parameter, but not power output. Moreover, engineers for particular operating conditions chose optimal (in average) parameter as a rule of thumb. Obviously, as external factors (chemical and fractional composition of coal dust, moisture content in coal, etc.) tend to vary, given methods cannot maintain maximum achievable efficiency of a boiler unit during some period of time. Besides, the stabilized parameter was chosen heuristically, as continuous operation mode of boilers is not suitable for large-scale and long lasting experiments. It should be noted that in recent studies of control systems for boilers, based on tracking of several system parameters (multifactorial system) [6-8], but the measured values also do not have a clear and sustainable link with the generated power. Therefore, previously used attempts to set up automated systems of air-fuel mixture making did not make it possible to maintain modes, similar to optimal for a prolonged period. However, even taking into account this fact, their application displayed significant efficiency increase of boilers, equipped with the systems in comparison with boilers, where parameters of air-fuel mixture were regulated manually [2,4,8]. The latter circumstance allows assuming that the proposed methods of active search for optimal of the fuel-air mixture parameters at a fixed heat output can significantly increase the average efficiency of steam systems.

The difference between our ideas from existing analogues:

-firstly, we propose to minimize fuel consumption, using power output as the main measurable parameter, while implemented to date systems of automatic control of the fuel-air mixture making only stabilized parameters in a certain way correlated with the generated power, but in no way did not determine it;

-secondly, we propose a system, that is able to actively search most preferable parameters for air-fuel mixture at a given (fixed) power output unlike previously proposed systems implemented on the basis of classical feedback theory [8];

-thirdly, to decrease system inertia and to enhance its reliability, in contrast to most previously proposed systems where one measurable parameter was used, we suggest to use several measurable system parameters. Along with indirectly measured power output, we suggest to take into account such factors as carbon dioxide in products of combustion and fuel dust concentration in a mixture being made.

The problem of complex automated control systems modeling is relevant due to technical equipment evolution and increasing complexity of controlled objects. Software modeling as a rule requires high proficiency in the field of software development, that is not always possessed by Engineers usually are not skilled in the sphere of software, though on the other hand, IT specialists lack essential knowledge in the field of system automation and control.

Nowadays the world's practice recognizes two main approaches to this problem solving:

1) Development of visual systems programming, which allows the specialist to create the model of a system using graphical editor by means of combining the blocks – primitives. Simulators of electric schemes (MicroCap, Proteus etc.) can serve as examples of this approach.

2) Modeling system development supports problem-oriented language development.

The main examples of such an approach are the systems of synthesis and verification of logic circuits, that support software programming languages: HDL (Hardware Design Language), Verilog and VHDL (VHSIC - Very high speed integrated circuits Hardware Description Language). In recent years, attempts have been made to create problem-oriented languages for modeling the processes of some specialized areas of industrial automation [9], as well as DSL (Domain Specific Language) for model – driven development of robotics [10].

This work concerns the description of experimental system of modeling of automated control processes. The system supports uniquely created DSL and allows describing signals and typed blocks of automation system. As well as HDL, the suggested DSL allows to create new models by means of primitives' aggregation with description of blocks-chain program as well as by means of module behavior description.

II. Methods

Simulation is an extremely important tool for all and every control engineer who develops the system of automation and control of technological processes in industry.

For non-linear plants, there is often no alternative for control engineer but only trial-and-error approach, using computer simulation. Thus, hardly any control engineer would not use simulation at least occasionally. Market offers many highly effective special-purpose simulation software tools, e.g. for the simulation of electronic circuits, or for the simulation of dynamics of multicore systems, and there is (or at least used to be) a good reason for that. However, there is no market to offer special-purpose control system simulators, in spite of the fact that control is such an important application of simulation.

The modeling of control systems can be considered as a particular case of modeling dynamic systems. Indeed, in most cases, the model should represent the interaction between environment (plant to be controlled) and the control system. Plants are mostly represented by continuous time systems whose behavior is often described by partial or ordinary differential equations. Therefore, the modeling system should be a continuous time modeling system. On the other hand, digital controllers can be represented mathematically as discrete event systems. These circumstances demand methods that can deal with heterogeneous components that exhibit a variety of different behaviors.

In many practical cases, the most natural and adequate mathematical model of a plant is a system of differential equations. However, with respect to modeling the control system, we come to a peculiar paradox: mathematically, in such a model, the controlling action is a deterministic function of time, whereas in reality we cannot predict control function of action before the modeling process being carried out. For example, let's consider classical problem of control of an inverted pendulum (Fig. 1).

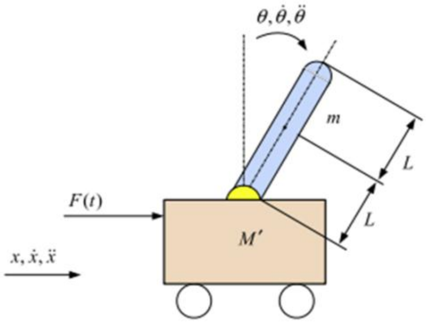

Figure 1: Inverted pendulum

With the notation x – cart position, θ – pendulum angle and F – applied force, the system can be described with the differential equations (1):

$$\begin{cases} \ddot{x} = \frac{1}{m+M} \cdot \left(F(t) + m \cdot l \cdot \left(\ddot{\theta}^2 \cdot l \cdot \cos(\theta) - \dot{\theta} \cdot \sin(\theta) \right) \right) \\ \ddot{\theta} = \frac{m \cdot l}{I} \cdot \left(g \cdot \sin(\theta) - \ddot{x} \cdot \cos(\theta) \right) \end{cases} \quad (1)$$

As it is known, introducing variables $x_1 = x,\ x_2 = \dot{x}, x_3 = \theta, x_4 = \dot{\theta}$, we can reduce the problem to the standard form of a system of ordinary differential equations (2).

$$x_i = f_i(x_1, x_2, \dots, x_n, t) \quad (2)$$

Obviously, to integrate the system, the dependence of the force F on time must be determined. So we need the simulation results to start the simulation process. Of course, this seeming paradox can be resolved by introducing mathematical description of the controller in the model. However, in most cases, simulation is used just in cases where the mathematical description of the complete control system is difficult to performer, so this theoretical approach is useless. It is important to note that the methods of numerical integration of systems of ordinary differential equations are highly developed, and at present a great progress has been made in the development of software that implements these methods.

Nowadays, freely available libraries of so-called solvers are available to developers, so it is highly desirable to use these components as part of modeling systems. Another significant problem is the way in which continuous signals are represented in implementation of modeling system. Most software systems and special-purpose languages designed for modeling dynamic systems use the discrete time model.

The process of simulation is to consistently calculate the state of the system at each time interval. Of course, this model of computation is not free from contradictions. If system is an aggregation of subsystems, subsystems may be connected in ways that yield the degree of ambiguity in computation. For example, assume that subsystem A has two outputs, one goes to subsystem B and another to subsystem C. Subsystem B has an output that feeds C. In this case, we may calculate the output of C whenever we have computed one of its inputs. Assuming that A has been processed, than we have the choice to calculate the outputs of B or of C. Depending on the choice of processing B or C, the outputs of C may have different values. Simultaneous events may in fact yield a nondeterministic behavior.

To reduce the inertia of our developed intelligent system for optimal energy-efficient control of the processes of the fuel-air mixture making in steam-driven boilers and to enhance its reliability, we suggest using several measured parameters of the system. Together with indirectly measured output power, it is assumed to take into account the concentration of carbon dioxide in the products of combustion as well as concentration of fuel dust in the mixture being made. The main idea of a proposed method is an active search of optimal composition at a given power. It is similar to gradient approach of searching the extremum of function of several variables used in numerical methods. Of course, hill-climbing technique, being applied in numerical analysis, cannot be directly applied to the problem considered due to a number of factors, the most important of which are the following two: first, the rate of convergence should be comparable to the rate of change of external factors (the given system operates in real time). Secondly, the value of deviating figure of generated power from the set value is strictly limited. To meet all the requirements it is assumed to use the vector of inputs of automated control system and to elaborate a mathematical model of external parameters influence on power output and measured parameters.

Basic research methods: mathematic modelling and live experiment: algorithm testing on model and experimental works on intelligent control system implementation, testing the model at actual data. An experimental stand will be built consisting of a small-sized boiler, burning nozzle, fuel intake system and necessary automation tools to construct a mathematical model, describing dependence of outputs on regulation characteristics (fuel consumption and concentration of air-fuel mixture) and external factors (coal grade, moisture content in the mixture, ambient temperature, etc.).

III. Results And Discussions

We introduce experimental simulation software platform around the ecosystem Racket integrated with a domain-specific language (DSL) tool-chain for modeling of automatic control systems. Racket is a LISP-family general purpose programming language, as well as a platform for language creation and implementation. Racket ecosystem consists of implementation of Racket language itself (including run-time system, libraries, JIT – compiler) along with development environment called Dr. Racket.

The key point of a project is usage of metaprogramming and code generation techniques for implementation of described above. Powerful Racket macro system allows not only to expand the syntax of the core language, but also to perform some validation of code.The key features of the DSL are the separation of the description of the controller and the description of the plant, and use of the FRP (Functional Reactive Programming) technique to describe the controller.

We believe that the FRF approach is especially suitable for modeling automatic control systems for a number of reasons: firstly, a high level of abstraction allows developer describing the system ignoring the internal implementation of continuous and discrete signals and synchronization problems. Secondly, a high level of abstraction allows choosing different methods of signal representation for program implementation. In recent years, there has been a growing interest in the use of functional programming methods for modeling dynamic systems, and considerable progress has been made in this area [10,11]. The DSL expansion has much in common with the Haskell DSL YAMPA, in particular the use of so-called signal combinators.

To provide detailed training to the fundamentals of FRP, the language of YAMPA and its practical application to the modeling of dynamic systems, we recommend referring to the article [12-14], and below we give only a brief description of the main concepts of the language. The basic concept of functional reactive programming is that of a signal. Signal can represent any continuous, time-varying value. One can think of a signal as having polymorphic type (3):

$$Signal\ \ a = Time \rightarrow a \tag{3}$$

That is a value of type Signal A is a function that maps suitable values of time (double in most of cases) to a value of type A. It should be noted that type A is arbitrary, in particular, is a functional type. Thus, the signal is a high-level abstraction that does not always conform the intuitive notion of a signal as a time-varying value of a physical value. Nevertheless, the direct conforming between the physical equations (i.e. the specification) and the FRP code (i.e. the implementation) is exact.

A key feature of the YAMPA language that is avoidance of the signals as first-class objects usage contrasts YAMPA with other programming languages. In other words, programmer cannot access the value of a signal at given moment of time or create a signal. Instead, the programmer has an access only to signal convertors, or what we prefer to call signal functions. A signal function is just a function that maps signals to signals (4):

$$\text{SF a b} \rightarrow \text{Signal a} \rightarrow \text{Signal b} \tag{4}$$

However, the actual representation of the type SF in Yampa is hidden (i.e. SF is abstract), so one cannot directly plot signal functions or apply them to signals. Instead of allowing the user to define arbitrary signal functions from scratch (which makes it all too easy to introduce time- and space-leaks), we provide a set of primitive signal functions and a set of special composition operators (or “combinators”) that enables to define more complex signal functions. These primitive values and combinators together provide a disciplined way to define signal functions that, fortuitously, avoids time- and space-leaks. Yampa program expresses the composition of a possibly large number of signal functions into a composite signal function that is than "run" at the top level by a suitable interpreter.

One of the significant differences between the DSL that we developed and YAMPA is the introduction of a signal source. As the source of the signal, interpolators are used, which "generate" a continuous signal by discrete values stored in a text file. Plant is modeled by a system of ordinary differential equations of the first order. The syntax of the controller description is as follows:

$$(\mathbf{plant}\ <\mathit{name\ of\ module}>((\mathit{input\ signals}))\ ((\mathit{output\ signals}))\ \ \mathit{func}-\mathit{list})$$

$$\left(((\mathit{internal\ signal\ description}))\ (\mathit{map\ func}-\mathit{list\ signal}-\mathit{list})\right)$$

Semantically plant definition is a closure, returning signal function, which arguments are input signals, and the result of conversion are output signals. Parameter funk-list is a list of functions (f_1, f_2,…, fn) representing the right part of ODE system in form (2)

IV. Conclusion

Automated control systems modeling is an area where the methods of declarative languages enable the technic to advance. The choice of the method of representing continuous signals is extremely important in the design of modeling systems as a whole. When modeling hybrid systems, in particular automatic control systems, it is desirable to be able to mix different representations of the signal during the simulation process.

The use of the FRP technique, in which the signals are first-class objects, allows great flexibility in the choice of the representation of continuous signals. Equally important are the advantages of FRP in the modeling of hybrid systems [15].

Although we have not completed an implementation of experimental software platform, this paper demonstrates our basic design approach and maps out the design landscape. We expect that further research of the links between functional reactive programming and automated control systems modeling will produce significant advances in this field.

The practical implementation of our developed software platform to design the intelligent system for optimal energy-efficient control of the air-fuel mixture making in steam – driven boilers can increase the efficiency of intake fuel control in steam boiler units will allow increasing the efficiency of steam boiler units, reducing the amount of chemical and mechanical fuel underburning, and, consequently, reducing the amount of combustion products, polluting the atmosphere.

Acknowledgment

The study was conducted with the financial support of the Science Committee of MES of RK in the framework of the program target financing for the 2017 biennium by the program 0006/PTF-17

References

[1] Ya.L. Packer, V.E. Egorov, "Investigation and analysis of dust incineration, taking into account the unevenness of processes over time," *Teploenergetika,* no. 6, pp. 49-52, 1971. (In Russian).

[2] M Kuang, Z. Li, C. Liu, and Q Zhu, "Experimental study on combustion and NOx emissions for a down-fired supercritical boiler with multiple-injection multiple-staging technology without overfire air," *Applied Energy,* vol. 106, pp. 254-261, 2013.

[3] M.Ya. Hesin *Indicators of combustion process for automation of combustion*. Moscow: Energy, 1969, pp.62. (In Russian)

[4] A. Mallik, "State feedback based control of air-fuel-ratio using two wide-band oxygen sensors," *ASCC, Malaysia,* pp. 1-6 [Proceedings of 10th Asian Control Conference Malaysia, p. 1-6, 2015].

[5] G. Conte, M. Cesaretti and D. Scaradozzi, "Combustion control in domestic boilers using an oxygen sensor," *IEEE Control and Automation (MED `06), Italy* [14th Mediterranean Conf., p. 1-4, 2006].

[6] P. Neuman, B. Sule, P. Zitek. and T. Dlouhy, "Non-linear engineering simulator of a coal fired steam boiler applied to fault detection of optimum combustion control," *IFAC Conference on Fault detection, supervision and safety for technical processes, Hungary,* p. 905-910, 2000.

[7] W. Tan, H.J. Marquez and T. Chen, "Multivariable Robust Controller Design for a Boiler System, *"IEEE Trans. on Control Systems Technology,* vol.10, no5, pp. 735-742, Sept. 2002.

[8] S. Simani, S. Beghelli, "PID controller design application based on boiler process model identification," *46th IEEE Conference, USA,* pp. 1064-1069, December 2007.

[9] J. Pieter Mosterman, "An overview of hybrid simulation phenomena and their support by simulation packages," in *Lecture Notes in Computer Science,* number 1569, Fritz W. Vaadrager and Jan H. van Schuppen Eds. Hybrid Systems: Computation and Control '99, 1999, pp. 165-177.

[10] M. Moser, M Pfeiffer., J. Pichler, "Domain-specific modeling in industrial automation: Challenges and experiences," *Proceedings of the first International Workshop on Modern Software Engineering Methods for Industrial Automation*, pp. 42-51, 2014.

[11] V. Djukic, A. Popovic, J.P. Tolvanen, "Domain-Specific Modeling for Robotics: From Language Construction to Ready-made Controllers and End-user Applications," *Proceedings of the third Workshop on Model-Driven Robot Software Engineering*, pp. 47-54, 2016.

[12] Z. Wan and P. Hudak, "Functional reactive programming from first principles," *in Proceedings of the ACM SIGPLAN, USA,* pp. 242-252, 2000 [Conference on Programming language design and implementation, PLDI '00, USA, pp. 242–252, 2000] doi: http://doi.acm.org/10.1145/349299.349331.

[13] H. Nilsson, A. Courtney, and J. Peterson, "Functional reactive programming, continued," *in Proceedings of the 2002 ACM SIGPLAN workshop on Haskell, USA,* pp. 51–64, 2002] doi: http://doi.acm.org/10.1145/581690.581695.

[14] P. Hudak, A. Courtney, H. Nilsson, and J. Peterson, "Arrows, robots, and functional reactive programming," in *Advanced Functional Programming, Vol. 2638 of Lecture Notes in Computer Science,* J. Jeuring and S. Jones, Eds. Springer: Berlin / Heidelberg, 2003, pp. 1949-1955. URL http://dx.doi.org/10.1007/978-3-540-44833-4 6. 10.1007/978-3-540-44833-4 6.

[15] H. Nilsson, J. Peterson, P. Hudak, "Functional hybrid modeling," in *Proceedings of PADL'03, USA,* vol. 2562, pp. 376-390, January2003 [5th International Workshop on Practical Aspects of Declarative Languages, USA, p. 376-390, 2003].