

# ТЕХНОЛОГИЯ И ТЕХНИКА ПОЛИГРАФИЧЕСКОГО И УПАКОВОЧНОГО ПРОИЗВОДСТВА

## TECHNOLOGY AND EQUIPMENT OF PRINTING AND PACKING MANUFACTURES

---

УДК 621.376.54

**В. С. Юденков, О. В. Бугай**

Белорусский государственный технологический университет

### МОДЕЛИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ СИСТЕМЫ MATLAB + SIMULINK В УЧЕБНОМ ПРОЦЕССЕ

Разработана программная технология пошагового освоения консалтинга, моделирования в UML и Stateflow и их использования в учебном процессе при работе в среде Matlab + Simulink. В работе представлены:

1) понятия математического моделирования: моделирование, физические модели, символические модели и формы их представления, классификация математической модели (одной из этих форм) по фактору, неопределенности, времени, непрерывности моделируемых процессов, а также по типу связи между моделируемыми величинами;

2) понятие консалтинга, предваряющего разработку любого программного обеспечения и информационных систем для выяснения целесообразности их разработки;

3) понятие UML (Unified Modeling Language) – универсального языка моделирования с примерами его использования для объектно-ориентированного моделирования объектов разработки с применением инструментальных средств Rational Rose и графического редактора создания презентаций, а также изложены теория и практикум по событийному моделированию с использованием пакета Simulink Stateflow;

4) технология взаимодействия программы в среде Simulink Stateflow и в микроконтроллере Arduino.

Предложенный подход может быть использован как обучающими, так и обучаемыми с целью более широкого внедрения последними пакета Matlab и его расширений в развитие инновационных технологий.

**Ключевые слова:** консалтинг, модель, математическое моделирование, аналитическое моделирование, имитационное моделирование.

**Для цитирования:** Юденков В. С., Бугай О. В. Моделирование с использованием системы Matlab + Simulink в учебном процессе // Труды БГТУ. Сер. 4, Принт- и медиатехнологии. 2021. № 1 (243). С. 5–11.

**V. S. Yudenzkov, O. V. Bugay**

Belarusian State Technological University

### SIMULATION USING MATLAB + SIMULINK IN THE EDUCATIONAL PROCESS

A software technology has been developed for step-by-step mastering of consulting, modeling in UML and Stateflow and their use in the educational process when working in the Matlab + Simulink environment. The work presents:

1) the concepts of mathematical modeling: modeling, physical models, symbolic models and forms of their representation, classification of a mathematical model (one of these forms) by factor,

uncertainty, time, continuity of the modeled processes, as well as by the type of connection between the modeling values;

2) the concept of consulting, prior to the development of any software and IS to clarify the feasibility of their development;

3) the concept of UML (Unified Modeling Language) is a universal modeling language with examples of its use for object-oriented modeling of development objects using the Rational Rose tools and a graphical editor for creating presentations, as well as the theory and practice of event modeling with using the Simulink Stateflow package;

4) the technology of program interaction in the Simulink Stateflow environment and in the Arduino micro-controller.

The proposed approach can be used by both educators and trainees with the aim of wider introduction by the latter of the Matlab package and its extensions in the development of innovative technologies.

**Key words:** consulting, model, mathematical modeling, analytical modeling, simulation.

**For citation:** Yudenkov V. S., Bugay O. V. Simulation using Matlab + Simulink in the educational process. *Proceedings of BSTU, issue 4, Print- and Mediatechnologies*, 2021, no. 1 (243), pp. 5–11 (In Russian).

**Введение.** Всегда актуально моделирование как новых проектируемых, так и модифицируемых систем и объектов. Такой подход повышает качество создаваемых опытных образцов, снижает время и расходы, связанные с их обработкой. Опытный образец изготовлен. Для определения его свойств наилучшим средством является натуральный эксперимент, который, к сожалению, во многих случаях невозможен, и поэтому приходится прибегать к его моделированию. Такая ситуация имеет место, когда натуральный эксперимент:

- опасен;
- экономически невыгоден;
- долговременен;
- кратковременен;
- протяжен в пространстве;
- невозможен;
- ненагляден.

**Основная часть.** С учетом изложенного выше при освоении системы Matlab + Simulink следует акцентировать внимание обучающихся на различных аспектах моделирования [1].

Моделирование – это замещение одного изделия-оригинала другим изделием с целью получения информации о свойствах изделия-оригинала.

Что касается классификации, то в общем случае по способу представления модели подразделяются на две группы: физические и символические (языковые). Последние в нашем случае являются предметом рассмотрения.

Символические модели – это абстрактные модели. Они отражают предварительные, приближенные представления о системе-оригинале (объекте моделирования).

Основными формами представления символических моделей являются:

- словесное описание (часто предшествует графическому представлению и математической формализации);
- графическое представление в виде кривых, номограмм, чертежей;

– блок-схемы, матрицы, таблицы (наиболее распространенные способы задания данных о структурной, логической и информационной части изделия);

– математическое описание – это описание в виде формул, уравнений, математических операций над переменными и алгоритмическое описание, используемое для разработки имитационной модели объекта либо системы, не имеющих аналитического описания.

Модели с математическими описаниями называются математическими. Математические модели можно классифицировать по различным признакам:

- 1) по фактору неопределенности:
  - стохастические (случайные, вероятные) – это модели, которые на выходе имеют неоднозначные значения параметров;
  - детерминированные – это модели, в которых для определенной совокупности входных значений параметров на выходе модели может быть получен единственный результат;
- 2) по фактору времени:
  - статические, в этих моделях все параметры и зависимости не меняются во времени в период функционирования модели;
  - динамические, в этих моделях параметры описывают систему в динамике (во времени);
- 3) по фактору непрерывности моделируемых процессов:
  - непрерывные (не содержат дискретных величин и представлены дифференциальными и интегральными уравнениями);
  - дискретные (все переменные заданы дискретными величинами);
- 4) по типу связи между моделируемыми величинами:
  - линейные (формируются в виде одного или системы линейных уравнений);
  - нелинейные (все взаимосвязи выражаются нелинейными функциями).

От типа модели зависит форма представления данных о системе, процессе или ситуации. Удобные формы описания моделируемого объекта во многом облегчают сложный процесс оценки адекватности и (или) верификации моделей.

При решении задач, связанных с количественной оценкой процессов, прибегают к методам математического (аналитического и имитационного) моделирования.

Рассмотрим математическое моделирование в среде Matlab + Simulink [2, 3–6].

Система Matlab – это язык программирования высокого уровня для технических вычислений, расширяемый большим числом пакетов прикладных программ-расширений. Самым известным из них стало расширение Simulink.

Типовой комплекс Matlab + Simulink содержит инструментальные «ящики» Toolboxes с пакетами расширения Matlab и Blocksets для расширения возможностей системы визуально-ориентированного блочного имитационного моделирования динамических систем Simulink.

Так как и без пакетов расширения Matlab представляет собой мощную операционную среду для выполнения огромного числа математических и научно-технических расчетов и создания пользователями своих пакетов расширения, в разработке последних принимают участие многие научные школы мира и ведущие университеты.

В издании [3] убедительно рекомендуется разработку любого программного обеспечения (ПО) и информационных систем (ИС) предварять выполнением тщательного консалтинга для выяснения целесообразности их разработки.

В книге [4] рекомендуют при создании программного обеспечения, информационных систем и не только их прибегать к средствам универсального языка моделирования (Unified Modeling Language-UML). Убедительно предлагается разработку любого проекта начинать с построения UML диаграмм. Метод разработки ПО и ИС без моделирования называют методом «собачьей конуры». В этом случае программисты уподобляются индивиду, который строит конуру, не удосужившись предварительно создать ее чертеж или даже эскиз, вооружившись доской, ножовкой, молотком и гвоздями.

Кстати, к услугам программистов в настоящее время предложена новая версия UML: UML 2.0. Формальная спецификация UML 2.0 опубликована в августе 2005 г., а последняя UML 2.5 в июне 2015 г.

Официальные диаграммы UML 1.\* , унаследованные UML 2.\* , – это диаграмма деятельности (процедурное и параллельное поведение), классов

(классы, свойства и отношения), взаимодействия (взаимодействие между объектами с акцентом на связях), компонентов (структура и взаимосвязи между компонентами), развертывания (развертывание артефактов в узлы), последовательности (взаимодействие между объектами с акцентом на последовательности), конечных автоматов (как события изменяют объект в течение его жизни), прецедентов (как пользователи взаимодействуют с системой), а неофициальные – объектов (вариант конфигурации экземпляров), пакетов (иерархическая структура времени компиляции).

К новинке в UML 2.\* можно отнести такие диаграммы, как составных структур (декомпозиция класса во время выполнения), обзора взаимодействий (комбинация диаграммы последовательности и диаграммы деятельности), временная (взаимодействие между объектами с акцентом на синхронизации).

К тому же, если версии UML 1.\* в Visio позволяли на диаграммах вариантов использования (прецедентов) показать границы проекта, то в UML 2.\* появилась возможность четко прорисовать на диаграммах последовательности фрейма взаимодействия с использованием операторов alt (из нескольких альтернатив выполняется фрагмент с истинным условием), opt (эквивалент alt с одной веткой), par (все фрагменты выполняются параллельно), loop (фрагмент может выполняться несколько раз, а защита обозначает тело итерации), region (фрагмент может иметь только один поток, выполняющийся за один прием), neg (отрицательный фрагмент; обозначает неверное взаимодействие), ref (ссылается на взаимодействие, определенное на другой диаграмме), sd (используется для очерчивания всей диаграммы последовательности, если это необходимо).

На наш взгляд, в любом случае моделирование ПО и ИС следует начинать с построения диаграмм вариантов использования (Use Case Diagrams). Такой подход позволяет:

- определить общие границы и контекст моделируемого ПО;
- сформулировать общие требования к функциональному поведению создаваемой системы;
- разработать исходную концептуальную модель системы для ее последующей детализации в форме логических и физических моделей;
- подготовить исходную документацию для взаимодействия разработчиков системы с ее заказчиками и пользователями.

Причем в случае создания систем автоматизации организаций в виде диаграмм вариантов использования моделируют бизнес-процессы, а для отдельных подразделений этих организаций – ПО.

Исчерпывающая информация по языку UML дана в издании [5], а в [6] не только по UML, но и по диаграммным техникам структурного системного анализа и проектирования. В книге [6] также представлено описание инструментальных средств BPwin, Erwin, Rational Rose и графического редактора для создания презентаций Visio.

Главное расширение системы Matlab – Simulink служит для имитационного моделирования моделей, состоящих из графических блоков с заданными свойствами (параметрами). Компоненты моделей, в свою очередь, являются графическими блоками и моделями, которые содержатся в ряде библиотек и с помощью мыши могут переноситься в основное окно и соединяться друг с другом необходимыми связями. В состав моделей могут включаться источники сигналов различного вида, виртуальные регистрирующие приборы, графические средства анимации.

С помощью кнопки Open в панели инструментов Matlab можно вызвать обычное для Windows-приложений окно загрузки файлов и загрузить одну из моделей множества демонстрационных примеров на моделирование с помощью Simulink. Например, решение дифференциального уравнения Ван-дер-Поля, реализация метода оптимизации для проектирования систем управления, быстрое преобразование Фурье и другие примеры.

К примеру, расширениями Simulink являются пакеты:

- Nonlinear Control Design Blockset: реализует метод динамической оптимизации для проектирования систем управления;

- Digital Signal Processing (DSP) Blockset: содержит прикладные программы для проектирования устройств, использующих процессоры цифровой обработки сигналов;

- Fixet-Point Blockset: ориентирован на моделирование цифровых систем управления в составе Simulink;

- Stateflow: моделирование событийно-управляемых систем, основанное на теории конечных автоматов;

- SimPower System: моделирование мощных энергетических систем;

- Report Generator: дает информацию о работе системы Matlab и пакета расширения Simulink;

- Real Time Windows Target и WorkShop: мощное средство управления реальными объектами и системами.

Выше мы акцентировали внимание на расширениях Simulink, а ниже лишь отметим, что и в Matlab входит множество пакетов расширения. Это пакеты:

- математических вычислений, существенно усиливающих математические возможности системы, повышающих скорость, эффективность и точность математических вычислений (8 пакетов);

- анализа и синтеза систем управления (7 пакетов);

- пакет идентификации систем;
- для обработки сигналов и изображений (3 пакета);

- прочие пакеты прикладных программ (7 пакетов);

- пакеты расширения Matlab 6.5 (5 пакетов);

- новейшие пакеты расширения Matlab 8 + Simulink 6 (2 пакета типа Toolbox и 1 пакет типа Blockset).

Проведем анализ событийного моделирования с использованием пакета расширения Simulink Stateflow [7].

Событийное моделирование – моделирование, при котором структура модели может меняться в зависимости от событий, происходящих в процессе моделирования. Это кардинально отличает его от имитационного моделирования, реализованного в Simulink, когда получают имитацию процесса по заданным исходным данным. Пакет расширения Simulink Stateflow в симбиозе с Simulink обеспечивает проведение событийного моделирования.

Пакет событийного моделирования Stateflow основан на теории конечных автоматов. Он позволяет представить функционирование системы на основе цепочки правил, которые задают соответствие событий и действий, выполняемых в ответ на эти события. Пакет Stateflow предназначен прежде всего для анализа, моделирования таких систем, как:

- детерминированные системы управления;
- диспетчерская служба различных транспортных средств;

- периферийные устройства и контроллеры для компьютеров;

- средства графического интерфейса пользователя (GUI);

- элементы человеко-машинного интерфейса (Men Machine Interface – MEI);

- гибридные системы на основе средств Simulink и ряда пакетов расширения (Control System, Digital Signal Processor и др.);

- наглядные интерактивные уроки по моделированию и проектированию систем.

Графический интерфейс пользователя пакета Stateflow имеет следующие компоненты:

- графический редактор SF-диаграмм;

- обозреватель для анализа SF-диаграмм (Stateflow Explorer);

- навигатор (Stateflow Finder) для поиска в SF-программах нужных объектов;
- отладчик SF-моделей;
- генератор кодов для работы совместно с расширением для работы в реальном времени Real Time Workshop.

Эти компоненты обеспечивают интуитивное понятие и простые приемы работы с пакетом Stateflow, позволяющим создавать SF-модели в виде графических SF-диаграмм динамического типа.

Подробную информацию по перечисленным выше компонентам пакета Stateflow с примерами их использования можно найти в книге [7].

Как правило, демонстрационный пример начинается с построения SF-диаграмм в графическом редакторе. Затем дано пошаговое представление использования четырех перечисленных выше компонентов, начиная с обозревателя Stateflow Explorer. Все начинается с таких простых SF-диаграмм, как:

- источник переменного напряжения, подключенный к осциллографу;
- контроль температуры бойлера с помощью SF-контроллера;
- вычисление цикла с управляющей целочисленной переменной  $i$ , меняющейся от 1 до 10 с шагом 1;
- SF-диаграмма модели скользящего с трением бруска;
- другие.

Затем следует рассмотрение более сложных примеров:

- моделирование системы автоматической трансмиссии автомобиля;
- моделирование отказоустойчивой системы контроля топлива;
- другие.

Следует отметить, что каждый из представленных примеров снабжен обстоятельным пошаговым описанием и иллюстрацией процесса создания рассматриваемой модели.

В качестве примера приведем подключение и управление микроконтроллером Arduino в среде Matlab и Simulink.

В обычной практике MATLAB используется в двух вариантах:

- 1) программирование на интерпретированном языке программирования MATLAB – пишутся программы, выполняются, получается результат;
- 2) использование графического пакета моделирования MATLAB SIMULINK – составляются блок-схемы, моделируются, получается результат.

И все это выполняется исключительно на компьютере, без привлечения внешних модулей.

При этом MATLAB поддерживает как возможность связи со внешними устройствами через коммуникационные порты, так и трансляцию в высокоуровневые языки программирования (C и C++) с их последующей компиляцией под любую возможную архитектуру, будь то ARM, AVR, процессоры цифровой обработки сигналов и прочее, лишь бы в системе был установлен требуемый компилятор/кросс-компилятор.

В продаже имеются специальные платы, которые позволяют подключить в MATLAB какое-то реально устройство и управлять/следить за ним. Платы эти весьма дороги, так почему бы это не отдать под контроль плате Arduino? Она проста, к ней просто подключить всякие устройства, да и список возможных устройств внушителен.

Arduino – это небольшая плата с собственным процессором и памятью. На плате также есть пара десятков контактов, к которым можно подключать всевозможные компоненты: лампы, датчики, моторы, чайники, роутеры, магнитные дверные замки и вообще всё, что работает от электричества. В процессор Arduino можно загрузить программу, которая будет управлять всеми этими устройствами по заданному алгоритму. Таким образом, можно создать бесконечное количество уникальных устройств, сделанных своими руками и по собственной задумке. Для того чтобы понять идею, рассмотрим инсталляцию среды.

1. Распаковываем пакет в `c:\arduinoIO`. Запускаем Matlab.

В командной строке Matlab переходим в папку `>>cd c:\arduinoIO`.

Выполняем команды: `>>install_arduino` и `>>saverpath`. Первая произведет установку путей, вторая их сохранит.

Далее необходимо подготовить саму плату – нужно залить в нее прошивку-ретранслятор. Ее задача – принимать команды из порта и выполнять их в железе. Вариантов такого сервера несколько. Один вариант поддерживает только функции работы с аналоговыми и цифровыми портами, другой – еще и шаговые двигатели. Для последнего требуется специальная библиотека, которую можно взять здесь: <https://github.com/adafruit/Adafruit-Motor-Shield-library/zipball/master>.

Ее требуется распаковать в папку `Arduino-XXX/libraries/`.

Открываем файл `adiosrv.pde` или `srv.pde` из среды Arduino и прошиваем. Перед прошивкой необходимо подключить устройство к порту USB и убедиться, что оно определилось операционной системой и при необходимости установить драйвер устройства.

Возвращаемся в консоль MATLAB и создаем новый объект:

```
>> a=arduino('COM4').
```

Вводим команды

```
>> a.pinMode(13,'OUTPUT')
```

```
>> a.digitalWrite(13,1)
```

и получаем уровень 1 на 13 пине Ардуино.

И на плате загорается светодиод.

Разумеется, что одновременно могут работать несколько плат, ведь для каждой из них создается свой уникальный объект. Для нашего случая этот объект называется 'a'.

2. Подключение Arduino к SIMULINK.

Для подключения потребуются компоненты: MATLAB (R2010a or later); Simulink; Simulink Coder; Embedded Coder.

Скачиваем пакет Simulink Support Package for Arduino (<http://www.mathworks.com/matlab-central/fileexchange/30277>) и распаковываем его в c:\arduino\_simulink; скачиваем среду Arduino IDE и распаковываем ее в c:\ArduinoTarget.

Выполняем следующие команды для добавления путей:

```
>> cdc:\arduino_simulink
```

```
>> addpath(fullfile(pwd,'arduino'),fullfile(pwd,'blocks'),fullfile(pwd,'demos'))
```

```
>> savepath.
```

Далее обновляем кастомизации:

```
>> sl_refresh_customizations.
```

Подключаем плату Arduino к компьютеру. Прошивать ее чем-либо не требуется.

Указываем путь к среде arduino (путь надо смотреть в папке)

```
>> arduino.Prefs.setArduinoPath('c:\Arduino-Target')
```

или

```
>> arduino_ec.Prefs.setArduinoPath('c:\ArduinoTarget\Arduino')
```

и определяем текущую платформу следующей командой:

```
>> arduino.Prefs.setBoard('mega').
```

Список всех доступных платформ выводится командой

```
>> arduino.Prefs.setBoardw/ATmega328)
```

'pro5v' (Arduino Pro or Pro Mini (5V, 16 MHz) w/).  
Проверяем доступные порты:

```
>> comPorts=arduino.Prefs.searchForComPort  
comPorts = 'COM4'
```

и подключаемся к нему:

```
>> arduino.Prefs.setComPort('COM4').
```

Теперь можно запустить имеющиеся примеры.

Открываем Simulink.

Создаем новую модель.

Открываем библиотеку компонентов и выбираем указанную библиотеку. Запускаем моделирование в среде Simulink.

**Заключение.** Эта публикация акцентирует внимание на общих понятиях консалтинга, моделирования в UML, Stateflow и их использовании в учебном процессе при работе в Matlab + Simulink.

На наш взгляд, такая «дорожная карта» приобретения знаний о консалтинге, UML, системе Matlab и ее расширениях будет способствовать более полному ее внедрению в развитие инновационных технологий.

### Список литературы

1. Липатова Н. Г. Имитационное моделирование процессов таможенного контроля: монография. М.: Изд-во Российской таможенной академии, 2015. 146 с.
2. Дьяконов В. П. Matlab. Полный самоучитель. М.: ДМК-Пресс, 2012. 768 с.
3. Калянов Г. Н. CASE-технологии. Консалтинг в автоматизации бизнес-процессов. М.: Горячая линия – Телеком, 2002. 320 с.
4. Буч Г., Рамбо Д., Джекобсон А. Язык UML: Руководство пользователя. М.: ДМК, 2000. 432 с.
5. Леоненков А. В. Самоучитель UML. СПб.: БХВ-Петербург, 2001. 304 с.
6. Бугай О. В., Юденков В. С. САПР программного обеспечения издательско-полиграфического комплекса: учеб. пособие. Минск: БГТУ, 2008. 174 с.
7. Дьяконов В. П. Simulink 5/6/7: Самоучитель. М.: ДМК-Пресс, 2008. 784 с.

### References

1. Lipatova N. G. *Imitatsionnoye modelirovaniye protsessov tamozhennogo kontrolya: monografiya* [Simulation of customs control processes: monograph]. Moscow, Izdatel'stvo Rossiyskoy tamozhennoy akademii Publ., 2015. 146 p.
2. Dyakonov V. P. *Matlab. Polnyy samouchitel'* [Matlab. Complete tutorial]. Moscow, DMK-Press Publ., 2012. 768 p.
3. Kalyanov G. N. *CASE-tekhnologii. Konsalting v avtomatizatsii biznes-protsessov* [CASE technologies. Consulting in the automation of business processes]. Moscow, Goryachaya liniya – Telekom Publ., 2002. 320 p.
4. Booch G., Rambeau D., Jacobson A. *Yazyk UML: Rukovodstvo pol'zovatelya* [UML: User's Guide]. Moscow, DMK Publ., 2000. 432 p.

5. Leonenkov A. V. *Samouchitel' UML* [Self-study guide UML]. St. Petersburg, BKhV-Peterburg Publ., 2001. 304 p.

6. Bugay O. V., Yudenkov V. S. *SAPR programmного obespecheniya izdatel'sko-poligraficheskogo kompleksa: ucheb. posobiye* [CAD software for publishing and printing complex: textbook]. Minsk, BGTU Publ., 2008. 174 p.

7. Dyakonov V. P. *Simulink 5/6/7: Samouchitel'* [Simulink 5/6/7: A tutorial]. Moscow, DMK-Press Publ., 2008. 784 p.

### Информация об авторах

**Юденков Виктор Степанович** – кандидат технических наук, доцент кафедры полиграфического оборудования и систем обработки информации. Белорусский государственный технологический университет (220006, г. Минск, ул. Свердлова, 13а, Республика Беларусь). E-mail: yudenkov@belstu.by

**Бугай Осип Викентьевич** – кандидат технических наук, доцент кафедры полиграфического оборудования и систем обработки информации. Белорусский государственный технологический университет (220006, г. Минск, ул. Свердлова, 13а, Республика Беларусь).

### Information about the authors

**Yudenkov Viktor Stepanovich** – PhD (Engineering), Assistant Professor, the Department of Printing Equipment and Information Processing Systems. Belarusian State Technological University (13a, Sverdlova str., 220006, Minsk, Republic of Belarus). E-mail: yudenkov @ belstu.by

**Bugay Osip Vikent'yevich** – PhD (Engineering), Assistant Professor, the Department of Printing Equipment and Information Processing Systems. Belarusian State Technological University (13a, Sverdlova str., 220006, Minsk, Republic of Belarus).

*Поступила 02.02.2021*