

УДК 681.5

ПРОГРАММНО-АППАРАТНОЕ МОДЕЛИРОВАНИЕ ФИЗИЧЕСКИХ ПРОЦЕССОВ В WOLFRAM MATHEMATICA С ИСПОЛЬЗОВАНИЕМ МИКРОКОНТРОЛЛЕРНЫХ УСТРОЙСТВ

Миргород Ю.С.

Научный руководитель – ст. преподаватель Буров А.Л.

Проблемы разработки современных систем управления теплофизическими и другими процессами непосредственно связаны с задачами моделирования реальных физических объектов, с проектированием систем управления и их тестированием.

Программно-аппаратное моделирование (ПАМ) – это моделирование физических систем в реальном времени на основе их математических моделей. ПАМ активно применяется в тех случаях, когда физическая система не доступна по причинам её дороговизны или небезопасности. Благодаря тому, что моделирование происходит в реальном времени, человек или управляющая система регулирования может напрямую взаимодействовать с физической моделью.

В настоящее время большинство систем автоматического управления имеют возможность изменять параметры управления при работе непосредственно на установке (например, регулятор уровня воды в барабане парогенератора в зависимости от мощности котла). Исследование возможностей применения ПАМ при проектировании систем управления позволяет предусмотреть некоторые параметры работы установки. Таким образом, не имея доступа к реальной проектируемой или модернизируемой установке, ПАМ позволяет обеспечить моделирование работы некоторых подсистем установки в условиях, приближенных к реальным, что дает возможность эффективнее производить настройку параметров системы управления на этапе проектирования.

Для демонстрации возможностей программно-аппаратного моделирования с использованием Wolfram Mathematica / SystemModeler в данной работе произведена симуляция нелинейной модели системы из двух связанных трубопроводами баков на микроконтроллере Arduino Uno. Изменяемый параметр в системе – расход жидкости в первый бак, который влияет на уровень в обоих резервуарах. Данные уровни являются выходными величинами. Для задания расхода в первый бак используется потенциометр, выходные значения уровней жидкости передаются через последовательное соединение и считывается фреймворком микроконтроллера.

Математическая модель

В системе из двух баков уровни жидкости в обоих резервуарах контролируются расходом в верхний бак (рисунок 1).

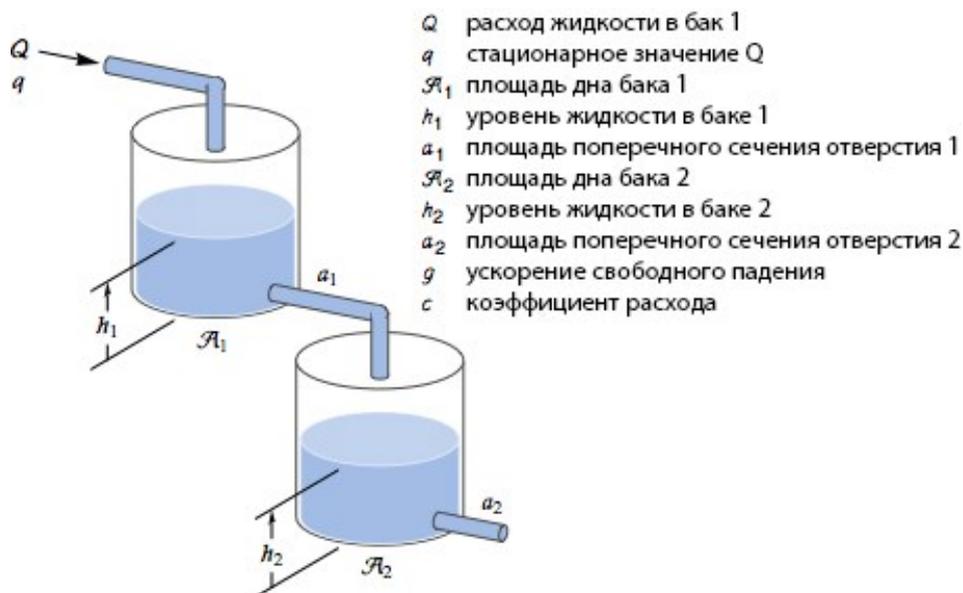


Рисунок 1. Схема моделируемой системы

Определим физические параметры системы:

$$\text{In}[\bullet]:= \text{pars} = \{c \rightarrow 0.7, g \rightarrow 9.8, A_1 \rightarrow \pi 4^2, a_1 \rightarrow \pi 0.25^2, A_2 \rightarrow \pi 3^2, a_2 \rightarrow \pi 0.25^2, q \rightarrow 1\};$$

Для вывода дифференциальных уравнений используем закон Бернулли и уравнения материального баланса:

$$\text{In}[\bullet]:= \text{eqs} = \{A_1 h_1' [t] == Q[t] - c a_1 \sqrt{2 g h_1 [t]}, A_2 h_2' [t] == c a_1 \sqrt{2 g h_1 [t]} - c a_2 \sqrt{2 g h_2 [t]}\};$$

В начальный момент времени для расхода $Q = q$ определим уровни жидкости в обоих баках, обеспечивающие стационарный режим системы, т.е. режим с $h_1'(t) = 0$ и $h_2'(t) = 0$:

$$\text{In}[\bullet]:= \text{Solve}[\{\theta == Q[t] - c a_1 \sqrt{2 g h_1 [t]}, \theta == c a_1 \sqrt{2 g h_1 [t]} - c a_2 \sqrt{2 g h_2 [t]}\}, \{h_1 [t], h_2 [t]\}] /.$$

$$\text{Out}[\bullet]:= \left\{ \left\{ h_1 [t] \rightarrow \frac{q^2}{2 c^2 g a_1^2}, h_2 [t] \rightarrow \frac{q^2}{2 c^2 g a_2^2} \right\} \right\}$$

Определим соответствующую нелинейную систему с входом Q и выходами h_1 и h_2 :

$$\text{In}[\bullet]:= \text{dtank} = \text{NonlinearStateSpaceModel} [$$

$$\text{eqs}, \left\{ \left\{ h_1 [t], \frac{q^2}{2 c^2 g a_1^2} \right\}, \left\{ h_2 [t], \frac{q^2}{2 c^2 g a_2^2} \right\} \right\}, \{\{Q[t], q\}\}, \{h_1 [t], h_2 [t]\}, t] /.$$

pars

$$\text{Out}[\bullet]= \left(\begin{array}{c|c} \{x_1 [t], 2.70077\} & \frac{Q[t] - 0.608494 \sqrt{x_1 [t]}}{16 \pi} \\ \hline \{x_2 [t], 2.70077\} & \frac{0.608494 \sqrt{x_1 [t]} - 0.608494 \sqrt{x_2 [t]}}{9 \pi} \\ \hline & x_1 [t] \\ \hline & x_2 [t] \end{array} \right) \mathcal{N}$$

Внедрение модели

Преобразуем полученную нелинейную систему в модель с дискретным временем:

`In[•]:= τ = 0.25;`

`ddtank = ToDiscreteTimeModel[dtank, τ]`

$$\text{Out[•]} = \begin{pmatrix} \{x_1[k], 2.70077\} & 0.00497359 (Q[k] - 0.608494 \sqrt{x_1[k]}) + x_1[k] \\ \{x_2[k], 2.70077\} & 0.00884194 (0.608494 \sqrt{x_1[k]} - 0.608494 \sqrt{x_2[k]}) + x_2[k] \\ & x_1[k] \\ & x_2[k] \end{pmatrix} \begin{matrix} \mathcal{N} \\ \\ \\ 0.25 \end{matrix}$$

Для контроля расхода в первый бак произведем масштабирование показания потенциометра в пределах от 1 до 2:

`In[•]:= pscaler = (1 + $\frac{u}{5}$) \mathcal{N}` _{0.25};

Настроим соединение элементов модели для вывода значения расхода в первый бак вместе с уровнями жидкости в резервуарах (рисунок 2):

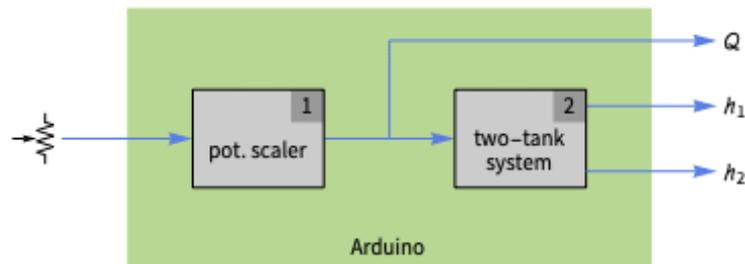


Рисунок 2. Схема соединения модели

`In[•]:= sys = SystemsConnectionsModel[{pscaler, ddtank}, {{1, 1} → {2, 1}}, {{1, 1}}, {{1, 1}, {2, 1}, {2, 2}}]`

`Out[•]= SystemsConnectionsModel` [Inputs: 1, Outputs: 3]

Загрузка пакета для работы с микроконтроллерами:

`In[•]:= Needs["MicrocontrollerKit`"]`

Внедрение модели:

`In[•]:= M = MicrocontrollerEmbedCode[sys, <| "Target" → "ArduinoUno", "Inputs" → "A0" → "Analog", "Outputs" → Table["Serial", 3] |>, "COM3"]`

`Out[•]= MicrocontrollerCodeData` [Status: ●●, Target: ArduinoUno]

Считывание смоделированных данных

Открываем последовательное соединение с устройством:

```
In[*]:= y = {};
dv = DeviceOpen["Serial", "COM3"]

Out[*]:= DeviceObject[
  + [
    [
      Class: Serial ID: 1
      Status: ● Connected (COM3)
    ]
  ]
]
```

Определяем начальный, разделительный и конечный байты в последовательном соединении:

```
In[*]:= {sb, db, eb} = Lookup[M["Serial"], {"StartByte", "DelimiterByte", "EndByte"}]
Out[*]:= {19, 44, 17}
```

Определяем функцию для разбора данных, приходящих через последовательное соединение:

```
In[*]:= parseData[{sb, flow__, db, h1__, db, h2__}] :=
  ToExpression@FromCharacterCode@# & /@ {{flow}, {h1}, {h2}}
parseData[___] := Sequence[]
```

Создаём задание на считывание значений каждую секунду (одновременно подаём скачок задания на потенциометре):

```
In[*]:= task = SessionSubmit@ScheduledTask[
  If[DeviceExecute[dv, "SerialReadyQ"],
    AppendTo[y, DeviceReadBuffer[dv, "ReadTerminator" → eb]];
    DeviceExecute[dv, "ReadFlush"];
  ],
  1,
  1;
];
```

По истечении 30 минут удаляем задние и закрываем соединение с устройством:

```
In[*]:= TaskRemove[task];
DeviceClose[dv];
```

Отообразим изменение уровня жидкости в двух баках в результате подачи скачка задания потенциометром (рисунок 3):

```
In[*]:= p1 = ListLinePlot[Transpose[parseData /@ y][[2 ;; 3]], PlotRange → All]
```

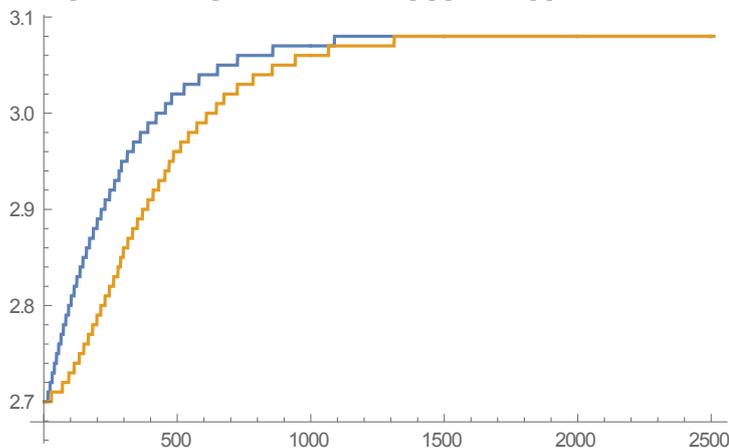


Рисунок 3. Изменение уровней жидкости в баках

Определим значение скачка расхода, поданного в систему (рисунок 4):

```
In[*]:= allu = (parseData /@ y) [[All, 1]];
u = UnitStep[t] + UnitStep[t - FirstPosition[allu, _? (# != 1. &)][[1]]]
      (allu[[-1]] - allu[[1]]);
Plot[Evaluate@u, {t, 0, Length@allu}]
```

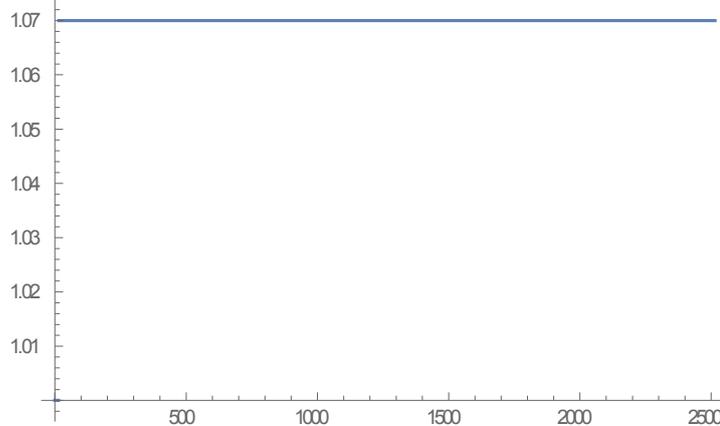


Рисунок 4. Скачок задания, поданный в систему

Сравниваем результаты с аналитическим решением в Wolfram Mathematica (рисунок 5):

```
In[*]:= OutputResponse[dtank, u, {t, 0, Length@allu}];
Show[p1, Plot[%, {t, 0, Length@allu}, PlotStyle -> Dashed]]
```

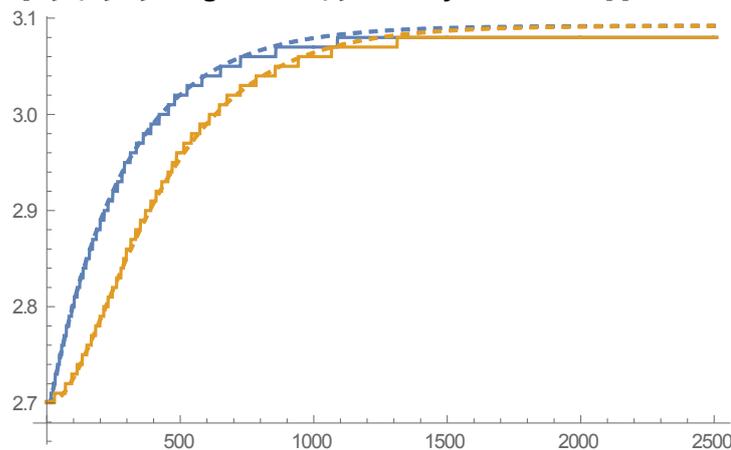


Рисунок 5. Сравнение результатов моделирования

При изменении расхода жидкости с $1.00 \text{ м}^3/\text{с}$ до $1.07 \text{ м}^3/\text{с}$ уровень в резервуарах вырос с 2.70 м до 3.08 м .

При отладке модели с использованием ПАМ и при обычном моделировании физического процесса в Wolfram Mathematica были получены одинаковые графики изменения уровней жидкости в баках. Можно сделать вывод, что использование программно-аппаратного моделирования не влияет на выходные результаты работы модели.

Рассмотренный пример использования ПАМ демонстрирует потенциал применения подобных систем на этапе принятия решений. Использование ПАМ при проектировании сложных систем управления увеличивает эффективность и точность процесса, а также обеспечивает полное соответствие физических моделей их математическому описанию.

Литература

1. Вынгра А.В. Использование программно-аппаратного моделирования при проектировании системы управления судовой холодильной установкой / А.В. Вынгра, Б.А. Авдеев // Вестник Государственного университета морского и речного флота имени адмирала С.О. Макарова. – 2017. – Т. 9. – № 4. – С. 806–813. DOI: 10.21821/2309-5180-2017-9-4-806-813.
2. Wolfram Language & System Documentation Center: Microcontroller Kit [Электронный ресурс] / Wolfram Language & System Documentation Center. – Режим доступа: <https://reference.wolfram.com/language/MicrocontrollerKit/tutorial/Overview.html>. – Дата доступа: 21.04.2019
3. J. Du, Y. Wang, C. Yang, and H. Wang, "Hardware-in-the-loop simulation approach to testing controller of sequential turbocharging system", Proceedings of the IEEE International Conference on Automation and Logistics, 2007.
4. A. Palladino, G. Fiengo, F. Giovagnini, and D. Lanzo, "A Micro Hardware-In-the-Loop Test System", IEEE European Control Conference, 2009.