

Генерация математических моделей линейных динамических систем, заданных структурными схемами

М. Л. Симонов, К. С. Зайцев, Н. П. Попова

Аннотация Целью данной работы является разработка метода вычисления математической модели структурной схемы в виде уравнений в пространстве состояний. Вычисление модели строится в виде итерационного процесса. Начальным состоянием математической модели принимается модель, состоящая из несоединенных элементов структурной схемы. Иерархия следования элементов, в общем случае не устанавливается, и является произвольной, но неизменной в процессе вычисления. Итерационный процесс состоит из шагов, на каждом из которых обрабатывается одно соединение. Соединение связывает выходной и входной порты элементов(а) на структурной схеме. Полезной является реализация алгоритма расчета текущего соединения без исключения и без обнуления соответствующих этому соединению строк и столбцов матриц уравнений состояния. Это соответствует получению эквивалентной, модели структурной схемы с доступными для подачи сигнала входными портами и открытыми для наблюдения выходными портами всех элементов, составляющих структурную схему. Показано, что для линейной динамической системы в случае наличия в ней разрешимой системы алгебраических контуров удается получить математическую модель динамической системы в виде уравнений состояния. Выводится условие, при выполнении которого система алгебраических контуров может быть разрешимой.

Ключевые слова – математическая модель, структурная схема, алгебраический контур, объектно-ориентированная разработка

Статья получена 15 августа 2022.

Симонов Михаил Леонидович, Национальный Исследовательский Авиационный Университет МАИ, доцент, sim_ml149@mail.ru

Зайцев Константин Сергеевич, Национальный Исследовательский Ядерный Университет МИФИ, профессор, KSZajtsev@mephi.ru

Попова Наталья Петровна, Общество с ограниченной ответственностью, Специальный Инженерный Центр, начальник отдела, Kostovsky@yandex.ru

1. ВВЕДЕНИЕ

Прикладное автоматическое управление имеет во многом завершенные формы и является предметом учебных дисциплин университетов. Вместе с тем, автоматизированные системы проектирования систем управления (Computer-Aided Control System Design - CACSD) являются базовым инструментом анализа и синтеза в различных предметных областях [1, 2]. Традиционным способом графического представления математических моделей является структурная схема, определяющая основные функциональные элементы системы и их взаимные связи [3].

Среди первых разработок в данной области были инструменты, основанные на языках моделирования и предназначенные для решения систем обыкновенных дифференциальных уравнений (ODE). С 1974 года были созданы более сорока языков моделирования непрерывных систем, которые развивались из попыток в цифровой форме представить поведение аналоговых вычислительных машин, широко используемых до 1960-х годов. Известная разработка в этого периода – язык Continuous System Simulation Language (CSSL), который определял систему как соединение блоков, эмулировал основные аналоговые блоки моделирования и находил численное решение ODE. Advanced Continuous Language (ACSL к ФОРТРАНУ) был одной из успешных реализаций CSSL [4]. Далее появились две важные разработки: декларативное, основанное на уравнении (equation-based language) моделирование и объектно-ориентированное моделирование. Декларативные языки не налагают фиксированную причинную (fixed

causality) связь на модель [5]. В этих языках, модель определяется рядом уравнений, которые устанавливает отношения между состояниями, их производными и временем. Определенная процедура в процессе моделирования выполняет преобразование этих уравнений в программное обеспечение, которое выполняется компьютером. Преимущество декларативных языков в том, что пользователи не должны определить математическую причинную связь уравнений, так, что модель может использоваться для любой причинной связи, наложенной другой системной компонентой [6, 7].

Предметом данной статьи является разработка метода (алгоритма) построения математической модели структурной схемы для линейных непрерывных по времени систем.

2. СОПУТСТВУЮЩИЕ РАБОТЫ

Разработка графических пользовательских интерфейсов, обеспечивающих моделирование, во многих случаях основана на структурных схемах. Значимыми разработками являлись - SystemBuild [8] и SIMULAB (позже переименованный в Simulink) [9]. Оба продукта использовали настраиваемый набор библиотек блоков, которые были интегрированы в две мощные интерактивные системы вычислений: MATRIX_x и MATLAB.

В настоящее время, система инженерных и научных расчетов MATLAB и система моделирования Simulink, получившие широкое признание, интегрирует много новых методологических разработок из области управления в связанные с управлением панели инструментов [10, 11]. Система MATLAB, также развивается как язык объектно-ориентированного программирования. Существующие инструменты SACSD постоянно расширяются, и включают новые классы моделей и новые вычислительные алгоритмы.

Структурные схемы применяются, чтобы представить и визуализировать зависимости в моделях сложных динамических систем. Ряд технологически ориентированных форм структурных схем стандартизован. Стандарт - ИЕС 61131-3:2003 описывает языки программирования для программируемых логических контроллеров [12]. В данном случае графический язык представляет собой программное средство для описания диаграммы множества соединенных функциональных блоков. Стандарт ISO

10628:2014 [13] - диаграммы для химической и нефтехимической промышленности, определяет представление блок-схем, и служит для обозначения общего потока заводских процессов, (process flow diagram) (PFG).

Вычислительный подход к моделированию, в ряде случаев, имеет много преимуществ перед физическим моделированием [14], поскольку разработка вычислительной модели легче и является менее затратной [15]. Динамические модели обычно представляются системами обыкновенных дифференциальных уравнений [16, 17], дифференциально-алгебраическими уравнениями [18] или дифференциальными уравнениями в частных производных. В математическом моделировании, основанном на численных методах, выполняется представление модели в виде вычислительного алгоритма [19-21]. В случае использования структурных схем для представления модели необходимым шагом является реализация того или иного метода формирования модели совокупности составляющих схему взаимосвязанных частей.

3. МЕТОДИКА

3.1. Постановка задачи

Множество состояний математической модели системы или процесса связано с метрическим пространством состояний системы или процесса. При взаимодействии системы с другими системами вводится множество управлений с элементами $u \subseteq U$ и множество воздействий $w \subseteq W$. Каждая система может быть разделена на взаимосвязанные подсистемы. При этом пространство состояний X представляется в виде суммы подпространств состояний, являющиеся сечением пространства X . Уравнения состояния для детерминированной линейной непрерывной управляемой системы во времени при детерминированных воздействиях можно записать в виде

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t); x(t_0) = x_0; u(t_0) = u_0. \\ y(t) &= Cx(t) + Du(t), \end{aligned} \quad (1)$$

где x – вектор состояния размерности n , u – вектор входа размерности k , y – вектор выхода размерности m , а матрица состояния A , матрица входа B , матрица выхода C , матрица прямой

сквозной связи (матрица обхода) D имеют согласующиеся размерности.

Структурная схема задана набором элементов схемы и линиями связи, соединяющими эти элементы. Каждый элемент имеет определенное число входов и выходов, которые на схеме однозначно представлены входными и выходными портами. Линия связи однозначно определяет связываемую пару: выходной порт и входной порт.

Каждый элемент в структурной схеме задан своим математическим описанием в виде уравнений состояния в пространстве состояний вида (1).

Ставится задача для заданной структурной схемы получить математическую модель в виде уравнений состояния вида (1).

3.2. Вычислительный метод построения математической модели

Пусть в структурной схеме присутствует N линейных элементов, математически описываемых в переменных состояния (1).

Вычисления математической модели строятся в виде итерационного процесса. Начальным состоянием математической модели (нулевой шаг итерации, как и все последующие, указывается верхним индексом в скобках) принимается модель, состоящая из несоединенных элементов структурной схемы.

$$\begin{aligned} \dot{x}(t) &= A^{(0)}x(t) + B^{(0)}u^{(0)}(t); \\ y^{(0)}(t) &= C^{(0)}x(t) + D^{(0)}u^{(0)}(t), \end{aligned} \quad (2)$$

где:

$$\begin{aligned} A^{(0)} &= \text{diag}(A_1, A_2, \dots, A_N); \\ B^{(0)} &= \text{diag}(B_1, B_2, \dots, B_N); \\ C^{(0)} &= \text{diag}(C_1, C_2, \dots, C_N); \\ D^{(0)} &= \text{diag}(D_1, D_2, \dots, D_N). \end{aligned} \quad (3)$$

В этих уравнениях

N – число структурных элементов размерности которых $n_1, n_2, \dots, n_N, k_1, k_2, \dots, k_N, m_1, m_2, \dots, m_N$; вектор состояния x размерности $n=n_1+n_2+\dots+n_N$, вектор входа $u^{(0)}$ размерности $k=k_1+k_2+\dots+k_N$, вектор выхода $y^{(0)}$ размерности $m=m_1+m_2+\dots+m_N$.

Порядок следования элементов произволен, но неизменяем в процессе вычислений. Это означает,

что все входные и выходные порты структурной схемы, имея свой порядковый номер в пределах своего структурного элемента, однозначно приобретают уникальный порядковый номер в пределах описания всей структурной схемы ((2)-(3).

Итерационный процесс вычисления математической модели состоит из шагов, на каждом из которых обрабатывается одно соединение, представленное на структурной схеме одной линией связи, однозначно определяющей один выходной порт схемы и один входной порт схемы.

Принимая за начальное состояние математической модели схемы соотношения ((2)-(3), выбираем на первом шаге итерационного процесса соединение (линию связи), тем самым однозначно определяем выходной порт, пусть $i^{(0)}$, и входной порт, пусть $j^{(0)}$ в модели ((2)-(3). Тогда условием выполнения данного соединения в модели схемы ((2) будет равенство:

$$u_j^{(0)} = y_i^{(0)}, \quad (4)$$

(в этом выражении верхний индекс, обозначающий номер итерации для индексов i и j опускается, так как номер итерации указывается в самом выражении). Использование (4) совместно с ((2)-(3) дает модель схемы на первом шаге итерационного процесса:

$$\begin{aligned} x &= A^{(1)}x + B^{(1)}u^{(1)}; \\ y^{(1)} &= C^{(1)}x + D^{(1)}u^{(1)}. \end{aligned} \quad (5)$$

Здесь вектор свободных выходов на первой итерации $y^{(1)}$ получается исключением из вектора выхода нулевой итерации компоненты $y_i^{(0)}$. Вектор свободных входов на первой итерации $u^{(1)}$ получается исключением из вектора входов нулевой итерации компоненты $u_j^{(0)}$.

Если $1 - d_{i,j}^{(0)} \neq 0$, то матрицы модели схемы на первом шаге вычисляются через матрицы модели нулевого шага по формулам (6-7):

$$A^{(1)} = A^{(0)} + \frac{1}{1-d_{ij}^{(0)}} b_j^{(0)} c_i^{(0)};$$

$$B^{(1)} = B_j^{(0)} + \frac{1}{1-d_{ij}^{(0)}} b_j^{(0)} d_{i/j}^{(0)}.$$

$$C^{(1)} = C_i^{(0)} + \frac{1}{1-d_{i,j}^{(0)}} d_{j/i}^{(0)} c_i^{(0)};$$

$$D^{(1)} = D_{i,j}^{(0)} + \frac{1}{1-d_{i,j}^{(0)}} d_{j/i}^{(0)} d_{i/j}^{(0)}.$$

где матрица $B_j^{(0)}$ формируется из матрицы $B^{(0)}$ исключением из нее $j^{(0)}$ -ого столбца $b_j^{(0)}$; матрица $C_i^{(0)}$ формируется из матрицы $C^{(0)}$ исключением из нее $i^{(0)}$ -ой строки $C_i^{(0)}$; матрица $D_{i,j}^{(0)}$ формируется из матрицы $D^{(0)}$ исключением из нее $i^{(0)}$ -ой строки $d_i^{(0)}$ и $j^{(0)}$ -ого столбца $d_{j/i}^{(0)}$; $d_{j/i}^{(0)}$ – $j^{(0)}$ -ый столбец матрицы $D^{(0)}$, из которого исключена $i^{(0)}$ -тая компонента; $d_{j/i}^{(0)}$ – $i^{(0)}$ -ая строка матрицы $D^{(0)}$ без $j^{(0)}$ -ой компоненты.

Продолжаемый далее итерационный процесс вычисления математической модели системы управления описывается приведенными ниже рекуррентными соотношениями.

Пусть на l -ом шаге итерационного процесса (в модель уже введены 1-е, 2-е, ..., l -ое соединения) текущая математическая модель схемы управления рассчитана в виде:

$$\dot{x} = A^{(l)}x + B^{(l)}u^{(l)};$$

$$y^{(l)} = C^{(l)}x + D^{(l)}u^{(l)}.$$

На $(l+1)$ -ом шаге в схеме рассматривается $(l+1)$ -я линия связи с однозначно определенными входным и выходным портами структурной схемы, тем самым однозначно (но, быть может, нетривиально) определяются выход, пусть $i^{(l)}$, и вход, пусть $j^{(l)}$, в модели (8), построенной на шаге l . Тогда условием

выполнения данного соединения в модели (8) будет равенство:

$$u_j^{(l)} = y_i^{(l)},$$

использование которого дает модель на $(l+1)$ шаге итерационного процесса:

$$\dot{x} = A^{(l+1)}x + B^{(l+1)}u^{(l+1)};$$

$$y^{(l+1)} = C^{(l+1)}x + D^{(l+1)}u^{(l+1)},$$

где матрицы рекуррентно определяются как:

$$A^{(l+1)} = A^{(l)} + \frac{1}{1-d_{i,j}^{(l)}} b_j^{(l)} c_i^{(l)};$$

$$B^{(l+1)} = B_j^{(l)} + \frac{1}{1-d_{i,j}^{(l)}} b_j^{(l)} d_{i/j}^{(l)};$$

$$C^{(l+1)} = C_i^{(l)} + \frac{1}{1-d_{i,j}^{(l)}} d_{j/i}^{(l)} c_i^{(l)};$$

$$D^{(l+1)} = D_{i,j}^{(l)} + \frac{1}{1-d_{i,j}^{(l)}} d_{j/i}^{(l)} d_{i/j}^{(l)}.$$

В соотношениях (11) предполагается, что $1 - d_{i,j}^{(l)} \neq 0$, и используются обозначения:

$b_j^{(l)}$ – $j^{(l)}$ -ый столбец матрицы $B^{(l)}$; $c_i^{(l)}$ – $i^{(l)}$ -ая строка матрицы $C^{(l)}$; $d_{j/i}^{(l)}$ – $j^{(l)}$ -ый столбец матрицы $D^{(l)}$ без $i^{(l)}$ -го элемента; $d_{i/j}^{(l)}$ – $i^{(l)}$ -ая строка матрицы $D^{(l)}$ без $j^{(l)}$ -го элемента; $B_j^{(l)}$ – матрица, полученная из $B^{(l)}$ удалением $j^{(l)}$ -го столбца; $C_i^{(l)}$ – матрица, полученная из $C^{(l)}$ удалением $i^{(l)}$ -ой строки; $D_{i,j}^{(l)}$ – матрица, полученной из $D^{(l)}$ удалением $i^{(l)}$ -ой строки и $j^{(l)}$ -го столбца.

3.3. Определение номеров входных и выходных портов схемы

Индексы $i^{(l)}$, $j^{(l)}$ на каждом шаге надо пересчитывать с учетом удаленных на прошлом шаге строк и столбцов матриц входа, выхода и обхода модели схемы. Возможна реализация вычислений без удаления, а с обнулением соответствующих строк и столбцов. Тогда получается эквивалентная, но избыточная замкнутая схема с полноразмерными векторами входа и выхода, соответствующими входным и выходным портам всех элементов, составляющих структурную схему. В этом случае определение индексов входных/выходных портов на всех шагах алгоритма остается тем же, что и на нулевом шаге. Для строгого получения результирующей модели схемы следует удалить аннулированные в результате реализации структурной связи строки и столбцы матриц входа, выхода и обхода такой избыточной модели.

3.4. Случай обнаружения в структурной схеме алгебраического контура

Наличие алгебраических контуров вызывает проблемы при использовании численных методов [22, 23, 24]. В данном разделе будет показано, что для линейной динамической системы в случае наличия в ней разрешимой системы алгебраических контуров удастся получить математическую модель динамической системы в виде уравнений состояния. При этом выводится условие, при выполнении которого система алгебраических контуров может быть разрешимой, и тогда математическую модель структурной схемы удастся получить. При невыполнении этого условия структурную схему следует перестроить.

Вернувшись к предлагаемому методу, отметим, что условием разрешения алгебраического контура на первом шаге вычислений является соотношение $1 - d_{i,j}^{(0)} \neq 0$. Аналогично на l -том шаге алгоритма условием разрешения алгебраического контура является: $1 - d_{i,j}^{(l-1)} \neq 0$. Если на промежуточном шаге алгоритма обнаруживается, что $1 - d_{i,j}^{(l)} = 0$, то нужно рассмотреть соответствующего соединения отнести на последний шаг алгоритма, а на данном шаге перейти к следующему соединению из набора соединений.

Пусть на l -том шаге алгоритма обнаруживается, что все оставшиеся не обработанные соединения образуют систему алгебраических контуров, каждый из которых в отдельности не может быть разрешен относительно скалярного выхода описанным выше алгоритмом. Пусть таких соединений m . Для каждого k -того из них известны номер входного порта схемы j_k и номер выходного порта схемы i_k . образуем из них векторы $J = [j_1 \cdots j_k \cdots j_m]$ и $I = [i_1 \cdots i_k \cdots i_m]$. Тогда условием соединения будет векторное равенство:

$$u_J^{(l)} = y_I^{(l)}, \quad (12)$$

где вектор $u_J^{(l)}$ получен выбором из вектора $u^{(l)}$ элементов с номерами, задаваемыми вектором J , а вектор $y_I^{(l)}$ получен выбором из вектора $y^{(l)}$ элементов с номерами, задаваемыми вектором I , а уравнения, записанные в виде:

$$\begin{aligned} x(t) &= A^{(l)}x(t) + B^{(l)}u^{(l)}(t); \\ y^{(l)}(t) &= C^{(l)}x(t) + D^{(l)}u^{(l)}(t); \\ u_J^{(l)} &= y_I^{(l)}, \end{aligned} \quad (13)$$

определяют исходные данные для построения модели схемы, если система алгебраических контуров окажется разрешимой.

Обозначим через $C_I^{(l)}$ матрицу, полученную выбором из матрицы $C^{(l)}$ строк с номерами, задаваемыми вектором I , и через $D_I^{(l)}$ матрицу, полученную выбором из матрицы $D^{(l)}$ строк с номерами, задаваемыми вектором I . В результате имеем:

$$y_I^{(l)} = C_I^{(l)}x + D_I^{(l)}u^{(l)}. \quad (14)$$

Обозначим через $D_{I,J}^{(l)}$ матрицу, полученную выбором из матрицы $D_I^{(l)}$ столбцов с номерами, задаваемыми вектором J , и через $D_{I,\bar{J}}^{(l)}$ матрицу,

составленную из остальных столбцов. Тогда справедливо уравнение:

$$y_I^{(l)} = C_I^{(l)} x + D_{I,J}^{(l)} u_J^{(l)} + D_{I,J}^{(l)} u. \quad (15)$$

Используя условие соединения (12), получим систему уравнений, описывающих систему алгебраических контуров:

$$(I_m - D_{I,J}^{(l)}) u_J^{(l)} = C_I^{(l)} x + D_{I,J}^{(l)} u \quad (16)$$

условием разрешимости которой является:

$$\det(I_m - D_{I,J}^{(l)}) \neq 0, \quad (17)$$

где I_m – единичная матрица размерности $m \times m$.

Если условие (17) выполняется, то система алгебраических контуров (16) имеет решение:

$$u_J^{(l)} = (I - D_{I,J}^{(l)})^{-1} C_I^{(l)} x + (I - D_{I,J}^{(l)})^{-1} D_{I,J}^{(l)} u. \quad (18)$$

Обозначим через $B_J^{(l)}$ матрицу, полученную выбором из матрицы $B^{(l)}$ столбцов с номерами, задаваемыми вектором J , и через $D_{I,J}^{(l)}$ матрицу, составленную из остальных столбцов. Обозначим также через $D_J^{(l)}$ матрицу, полученную выбором из матрицы $D^{(l)}$ столбцов с номерами, задаваемыми вектором J , и через $D_{I,J}^{(l)}$ матрицу, составленную из остальных столбцов. Из вектора выхода $y^{(l)}(t)$ в системе уравнений (13) сформируем вектор открытых портов системы $y(t)$. Для этого образуем матрицу $C_I^{(l)}$ вычеркиванием из матрицы $C^{(l)}$ строк, задаваемых вектором I , матрицу $D_{I,J}^{(l)}$ вычеркиванием из матрицы $D_J^{(l)}$ строк, задаваемых вектором I , и матрицу $D_{I,J}^{(l)}$ вычеркиванием из матрицы $D_J^{(l)}$ строк,

задаваемых вектором I . Тогда из системы уравнений (13) можно найти, что:

$$\begin{aligned} \dot{x}(t) &= A^{(l)} x(t) + B_J^{(l)} u_J^{(l)}(t) + B_J^{(l)} u(t); \\ y(t) &= C_I^{(l)} x(t) + D_{I,J}^{(l)} u_J^{(l)}(t) + D_{I,J}^{(l)} u(t), \end{aligned} \quad (19)$$

откуда с использованием выражения (18) системы алгебраических контуров имеем:

$$\begin{aligned} \dot{x}(t) &= [A^{(l)} + B_J^{(l)} (I - D_{I,J}^{(l)})^{-1} C_I^{(l)}] x(t) + [B_J^{(l)} + B_J^{(l)} (I - D_{I,J}^{(l)})^{-1} D_{I,J}^{(l)}] u(t); \\ y(t) &= [C_I^{(l)} + D_{I,J}^{(l)} (I - D_{I,J}^{(l)})^{-1} C_I^{(l)}] x(t) + [D_{I,J}^{(l)} + D_{I,J}^{(l)} (I - D_{I,J}^{(l)})^{-1} D_{I,J}^{(l)}] u(t). \end{aligned} \quad (20)$$

Полученная система уравнений является моделью схемы в переменных состояния для рассмотренного случая наличия в схеме разрешимой системы алгебраических контуров (16).

Если условие (17) не выполняется, то это означает, что в исследуемой схеме присутствует неразрешимая система алгебраических контуров. Такую структурную схему следует переформировать.

4. ПРИМЕР РАБОТЫ МЕТОДА

Работу вычислительного метода покажем на примере, структурная схема которого представлена на рисунке 1.

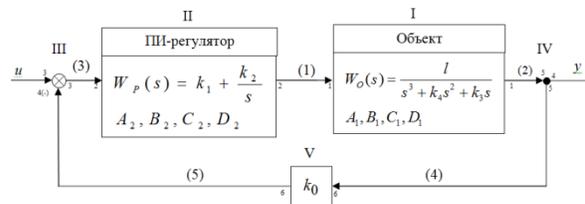


Рис. 1. Пример структурной схемы линейной системы.

Для примера взяты следующие числовые данные: $k_0 = 1.0$, $k_1 = 0.445$; $k_2 = 0.0083$; $k_3 = 2.0$; $k_4 = 3.0$; $l = 4.0$.

Каждому элементу схемы поставлен в соответствие свой порядковый номер, обозначаемый на схеме римскими цифрами.

Элемент, именуемый Объект, пронумерован как первый. Передаточная функция Объекта известна как $W_o(s) = \frac{4}{s^3 + 3s^2 + 2s}$. По условиям использования предлагаемого метода математическая модель каждого элемента схемы задается уравнениями состояния. Данному Объекту соответствуют матрицы математической модели в пространстве состояний:

$$A_1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -2 & -3 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0 \\ 0 \\ 4 \end{bmatrix}, \quad C_1 = [1 \ 0 \ 0], \quad D_1 = [0]$$

Элемент схемы, именуемый ПИ-регулятор, пронумерован как второй. Его передаточная функция известна как $W_p(s) = k_1 + k_2/s$. Ему соответствует математическая модель в пространстве состояний, определяемая матрицами: $A_2=[0]$, $B_2=[0.0083]$, $C_2=[1]$, $D_2=[0.445]$.

Сумматор пронумерован как третий элемент схемы, разветвление – как четвертый элемент схемы, коэффициент усиления k_0 в цепи обратной связи – как пятый элемент. Эти элементы в пространстве состояний задаются матрицами обхода, соответственно:

$$D_3 = [1 \ -1], \quad D_4 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad D_5 = [1].$$

Порты каждого элемента схемы пронумерованы согласно его математической модели.

Все данные об элементах схемы, необходимые для построения математической модели схемы, представлены в таблице 1.

Таблица 1 – Набор элементов схемы.

Номер на рис.	Размерности	Матрица состояния	Матрица входа	Матрица выхода	Матрица обхода
I	$n=3, k=1, m=1$	$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -2 & -3 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 4 \end{bmatrix}$	$[1 \ 0 \ 0]$	$[0]$
II	$n=1, k=1, m=1$	$[0]$	$[0.0083]$	$[1]$	$[0.445]$
III	$n=0, k=2, m=1$	-	-	-	$[1 \ -1]$
IV	$n=0, k=1, m=2$	-	-	-	$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$
V	$n=0, k=1, m=1$	-	-	-	$[1]$

	$m=1$				
--	-------	--	--	--	--

Все объявленные на рисунке соединения в схеме представлены в таблице 2.

Таблица 2 – Набор соединений схемы.

Номер на рис.	Выходной порт			Входной порт		
	Элемент (блок)	Номер порта элемента (блока)	Номер порта схемы (i)	Элемент (блок)	Номер порта элемента (блока)	Номер порта схемы (j)
(1)	II	1	2	I	1	1
(2)	I	1	1	IV	1	5
(3)	III	1	3	II	1	2
(4)	IV	2	5	V	1	6
(5)	V	1	6	III	2	4

Каждое соединение определяется указанием для него соединяемых выходного и входного портов. Порт в схеме однозначно определяется номером элемента из таблицы 1 и порядковым номером порта данного элемента. Введем сквозную (по всей схеме) нумерацию как входных, так и выходных портов по порядку нумерации элементов в схеме и портов в каждом из них. Заданный таким образом номер назовем номером порта схемы. Участвующий в соединении порт представлен в таблице 2 как номером порта элемента, так и номером порта схемы.

На рисунке 1 все соединения пронумерованы согласно порядку их следования в таблице 2 арабскими цифрами в скобках. Порты пронумерованы как порты схемы маленькими арабскими цифрами.

Построим математическую модель структурной схемы в виде уравнений в пространстве состояний. Для определения соответствующих матриц используем предложенный метод.

Из таблицы 1 определим начальные данные: число структурных элементов схемы $N=5$, вектор состояния схемы x размерности $n=4$, вектор входа $u^{(0)}$ размерности $k=6$, вектор выхода $y^{(0)}$ размерности $m=6$.

$$A^{(0)} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -2 & -3 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}; \quad B^{(0)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 4.0000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.0083 & 0 & 0 & 0 & 0 \end{bmatrix};$$

$$C^{(0)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}; \quad D^{(0)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.445 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0000 & -1.0000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

Столбцы и строки матрицы обхода и, соответственно, столбцы матрицы входа и строки матрицы выхода пронумерованы индексами (маленькими цифрами), являющимися номерами

входных и выходных портов схемы, которые указаны в таблице 2.

Работаем последовательно по шагам.

Шаг 1-й.

Согласно таблице 2 обрабатывается соединение (1): $i=2, j=1$. Это индексы строк и столбцов, указанные маленькими цифрами справа и снизу матрицы обхода на предыдущем шаге. Так как $1-d_{2,1}^{(0)}=1 \neq 0$, то с использованием выражения (11) матрицы модели на этом шаге итерационного процесса будут найдены как:

$$A^{(1)} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -2 & -3 & 4 \\ 0 & 0 & 0 & 0 \end{bmatrix}; \quad B^{(1)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1.7800 & 0 & 0 & 0 & 0 \\ 0.0083 & 0 & 0 & 0 & 0 \end{bmatrix};$$

$$C^{(1)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}; \quad D^{(1)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{matrix} 1 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix}$$

Шаг 2-й.

Согласно таблице 2 обрабатывается соединение (2): $i=1, j=5$. Так как $1-d_{1,5}^{(1)} \neq 0$, то с использованием выражения (11) матрицы модели на этом шаге итерационного процесса будут найдены как:

$$A^{(2)} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -2 & -3 & 4 \\ 0 & 0 & 0 & 0 \end{bmatrix}; \quad B^{(2)} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1.7800 & 0 & 0 & 0 \\ 0.0083 & 0 & 0 & 0 \end{bmatrix};$$

$$C^{(2)} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}; \quad D^{(2)} = \begin{bmatrix} 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{matrix} 3 \\ 4 \\ 5 \\ 6 \end{matrix}$$

Двигаясь таким же образом дальше, обрабатываем соединения (3) для $i=3, j=2$ и (4) для $i=5, j=6$.

На последнем 5-м шаге согласно таблице 2 обрабатывается соединение (5) $i=6, j=4$. Так как $1-d_{6,4}^{(4)} \neq 0$, то с использованием выражения (11) матрицы модели на этом шаге итерационного процесса будут найдены как:

$$A^{(5)} = \begin{bmatrix} 0 & 1.0000 & 0 & 0 \\ 0 & 0 & 1.0000 & 0 \\ -1.7800 & -2.0000 & -3.0000 & 4.0000 \\ -0.0083 & 0 & 0 & 0 \end{bmatrix}; \quad B^{(5)} = \begin{bmatrix} 0 \\ 0 \\ 1.7800 \\ 0.0083 \end{bmatrix};$$

$$C^{(5)} = [1 \ 0 \ 0 \ 0]; \quad D^{(5)} = [0]$$

На этом заканчивается набор соединений таблицы 2 и завершается построение математической модели в виде уравнений состояния для структурной схемы, представленной на рисунке 1. Математическая модель схемы в полном соответствии с ее изображением на рисунке 1 имеет один свободный вход, пронумерованный как входной порт схемы 3, и один свободный выход, пронумерованный как выходной порт схемы 4, и представлена полученными матрицами уравнений состояния.

5. ПОДСИСТЕМЫ В СТРУКТУРНЫХ СХЕМАХ

Подсистемой структурной схемы называют группу элементов в схеме со всеми взаимосвязями. Подсистема в структурной схеме системы используется в виде одного элемента. Понятие подсистем является одним из понятий метамодели систем MATLAB- Simulink, при этом поддерживается понятие иерархии подсистем. С использованием подсистем создается иерархическая модель. Подсистема находится на одном уровне, а элементы, которые составляют подсистему, находятся на другом.

Создание и использование подсистем предполагает организацию библиотеки подсистем с управлением пользовательским доступом к подсистемам, с установкой параметров полномочий.

В подсистему объединяются функционально связанные элементы, которые имеет тенденцию сохранять структуру своих взаимосвязей и параметры в процессе проектирования. Количество выведенных на экран элементов моделируемой схемы сокращается. Все это делает схему более доступной для понимания и удобной для использования в работе.

Предлагается рассматривать подсистему в виде элемента с математической моделью в виде матриц уравнений состояния, рассчитываемых предлагаемым вычислительным методом.

Пусть при проектировании системы сразу предусматривается и определяется её подсистема. Тогда с использованием данного метода можно

вычислить математическую модель такой подсистемы в пространстве состояний, которую можно использовать как математическую модель нового элемента (подсистема-элемент), готового для применения в данной родительской структурной схеме и не только данной.

В случае, когда выделение и создание подсистемы становится насущным в уже набранной сложной структурной схеме, средствами графического интерфейса выделяется часть родительской схемы, и переносится в окно создания новой структурной схемы с полной информацией о составляющих эту подсистему элементах и их соединениях. Принципиально информации об элементах и соединениях *подсистемы, хранящейся в описании* родительской схемы, достаточно, чтобы автоматически рассчитать математическую модель подсистемы с использованием предлагаемого метода.

Далее такую подсистему можно оформить, как подсистему-элемент и использовать такой элемент, как в исходной родительской схеме, так и в других схемах.

Подсистемы позволяют создавать иерархическую модель, включающую различные уровни. Перемещение по иерархии подсистем предполагает наличие таких возможностей, как открытие подсистемы для просмотра, открытие подсистемы для редактирования, возврат из открытой подсистемы в родительскую систему, развертывание подсистемы, состоящее в перемещении содержания подсистемы в родительскую схему со сглаживанием иерархии на один уровень.

Постановка задачи реализации многоуровневой иерархии подсистем при расчете математической модели как родительской схемы, так и всех ее подсистем, принципиально ничего не меняет в сути предлагаемого вычислительного метода. Решение такой задачи определяется условиями и особенностями применения метода и лежит в области использования средств проектирования графического интерфейса.

6. ОБСУЖДЕНИЕ

В теории систем соединения могут быть представлены некоторыми типами графов: линейный граф (linear graph), граф связей (bond graph), граф потока сигналов (signal flow graph), гибридный граф связей – temporal causal graph (TCG) или структурными схемами [25]. В работе

[26] TCG служит для выявления алгебраических контуров и эквивалентных преобразований для их удаления.

Алгебраический контур рассматривается в системе моделирования Simulink [10, 24, 27]. Отмечается, что в общем случае его наличие в динамической системе переводит исследуемую систему из разряда обыкновенных дифференциальных уравнений (ODE) в систему дифференциально-алгебраических уравнений (DAE). Предлагаются возможности выявления, диагностики и отработки алгебраических контуров. Пользователю предлагаются различные стратегии реагирования. В общем случае, при выявлении алгебраического контура, Simulink для его разрешения использует методы решения нелинейных алгебраических уравнений на каждом шаге моделирования. Отмечается ограниченность возможностей средств решения алгебраического контура.

В статье представляется решение в виде математического описания, используемого для реализации этапа построения модели линейной динамической системы, заданной структурной схемой. На каждом этапе обработки последовательности соединений формируются соответствующие математические модели, и выявляется наличие алгебраических контуров в модели. Необработанные соединения образуют m алгебраических контуров. Получена система уравнений, описывающих систему алгебраических контуров и условие существования и единственности ее решения. На основании полученного решения системы уравнений выполнено построение математической модели структурной схемы, которая на этапе построения, включает разрешимые алгебраические контуры. В противном случае структурную схему следует переформировать. Процесс строится, как процедура проверки математического условия, и связан с обработкой конкретного соединения. Пользователь может задавать последовательность обработки соединений и иметь информацию о входных и выходных портах, линии связи между которыми приводят к возникновению алгебраических контуров. Данная возможность является желательной при формировании систем со многими входами и выходами (MIMO) высокого порядка.

7. ЗАКЛЮЧЕНИЕ

Вычисление математической модели представляет собой итерационный процесс, состоящий из шагов, на каждом из которых обрабатывается одно соединение, представленное на структурной схеме одной линией связи.

Метод удобен для разработки программно-ориентированного алгоритма и использования его при разработке программного обеспечения исследования сложных динамических систем высокого порядка классическими и современными методами.

Метод дает возможность обнаружить в структурной схеме алгебраический контур, а в общем случае – систему алгебраических контуров. При этом выводится условие, при выполнении которого система алгебраических контуров разрешима, и удастся получить математическую модель структурной схемы. При невыполнении этого условия структурную схему следует перестроить.

Рассмотрена возможность использования данного метода для реализации многоуровневой иерархии подсистем в структурной схеме. Математическая модель как родительской схемы, так и всех ее подсистем, может быть рассчитана с использованием предлагаемого вычислительного метода. Практическое решение такой задачи определяется условиями и особенностями применения метода и лежит в области использования средств проектирования графического интерфейса.

БИБЛИОГРАФИЯ

- Golnaraghi, F., & Kuo, B. C. (2017). *Automatic control systems*. McGraw-Hill Education.
- Varga, A. (2014). *Computer-aided control systems design: Introduction and historical overview*. Encyclopedia of Systems and Control.
- Karayanakis, N. M. (1995). *Advanced system modelling and simulation with block diagram languages*. CRC Press Inc. 350 pp.
- Mitchell, E. E. & Gauthier, J. S. (1976). *Advanced continuous simulation language (ACSL)*. *Simulation*, 26(3), p. 72-78.
- Baumann, D., Solowjow, F., Johansson, K. H., & Trimpe, S. (2020). *Identifying causal structure in dynamical systems*. arXiv preprint arXiv: 2006.03906.
- Sinha, R., Paredis, C. J., Liang, V. C. & Khosla, P. K. (2001). *Modeling and simulation methods for design of engineering systems*. *J. Comput. Inf. Sci. Eng.*, 1(1), p. 84-91.
- Zhu, A., Pauwels, P., & De Vries, B. (2021). *Smart component-oriented method of construction robot coordination for prefabricated housing*. *Automation in Construction*, 129, 103778.
- Shah, S. C., Floyd, M. A., & Lehman, L. L. (1990). *MATRIX_x: Control design and model building CAE capability*. *Computer-Aided Control Systems Engineering (M. Jamshidi and CJ Herget, eds.)*(Elsevier, Amsterdam, 1985).
- Grace, A. C. (1991). *SIMULAB, An integrated environment for simulation and control*. In 1991 American Control Conference (pp. 1015-1020). IEEE.
- The MathWorks (2019). *Simulink user's guide*. https://www.mathworks.com/help/releases/R2019b/pdf/doc/simulink/sl_using.pdf. Version R2019b [Online; accessed Jul. 2022].
- The MathWorks (2020d). *MATLAB*. [Online; accessed Jul. 2022] <https://www.mathworks.com/products/matlab.html>.
- Gsellmann, P., Melik-Merkumians, M., Zoitl, A., & Schitter, G. (2020). *A novel approach for integrating IEC 61131-3 engineering and execution into IEC 61499*. *IEEE Transactions on Industrial Informatics*, 17(8), 5411-5418.
- Steindl, G., & Kastner, W. (2021, May). *Transforming OPC UA information models into domain-specific ontologies*. In 2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS) (pp. 191-196). IEEE.
- Willard, J., Jia, X., Xu, S., Steinbach, M., & Kumar, V. (2020). *Integrating physics-based modeling with machine learning: A survey*. arXiv preprint arXiv:2003.04919, 1(1), p. 1-34.
- Самарский А. А., Михайлов А. П. *Математическое моделирование: Идеи. Методы. Примеры*. 2-е изд., ред. - Москва: ФИЗМАТЛИТ, 2005. 320 с.
- Shampine, L. F. (2018). *Numerical solution of ordinary differential equations*. Published by Routledge. 632 p.
- Noonburg, V. W. (2020). *Differential equations: from calculus to dynamical systems (Vol. 43)*. American Mathematical Soc..
- Campbell, S., Ilchmann, A., Mehrmann, V., & Reis, T. (Eds.). (2019). *Applications of differential-algebraic equations: examples and benchmarks*. Cham: Springer International Publishing.
- Самарский А. А., Гулин А. В. *Численные методы: Учебное пособие для вузов*. - Москва: Наука, 1989. -432 с.
- Yang, W. Y., Cao, W., Kim & oth. (2020). *Applied numerical methods using MATLAB*. John Wiley & Sons.
- DeRusso, P. M., Desrochers, A. A., Roy, R. J., & Close, C. M. (1997). *State variables for engineers*. John Wiley & Sons, Inc..
- Adegbege, A. A., & Levenson, R. M. (2021). *Dynamic multivariable algebraic loop solver for input-constrained control*. *IEEE Transactions on Automatic Control*, 67(6), 3051-3058.
- Jaskolka, M., Pantelic, V., Wassyn, A., & Lawford, M. (2020). *Supporting modularity in Simulink models*. arXiv preprint arXiv:2007.10120.
- Mosterman, P. J., & Ciolfi, J. E. (2004, December). *Using interleaved execution to resolve cyclic dependencies in time-based block diagrams*. In 2004 43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601) (Vol. 4, pp. 4057-4062). IEEE.
- Knauer, U. & Knauer, K. (2019). *Algebraic graph theory*. In *Algebraic Graph Theory*. de Gruyter. <https://doi.org/10.1515/9783110617368>
- Mosterman, P. J., Manders, E. J., & Biswas, G. (2000, May). *Qualitative dynamic behavior of physical system models with algebraic loops*. In Eleventh International Workshop on Principles of Diagnosis (pp. 155-162).
- Jaskolka, M. (2020). *Making Simulink Models Robust with Respect to Change (Doctor of Philosophy dissertation)*, McMaster University Hamilton, ON, Canada. 206 p. https://macsphere.mcmaster.ca/bitstream/11375/26126/2/Jaskolk_a_Monika_2020Dec_PhD.pdf [Online; accessed Aug 2022].

Generation of mathematical models of linear dynamic systems given by block diagrams

M. L. Simonov, K. S. Zaytsev, N. P. Popova

Abstract — The purpose of this work is to develop a method for calculating the mathematical model of a block diagram in the form of equations in the state space. The calculation of the model is built as an iterative process. The initial state of the mathematical model is a model consisting of unconnected elements of the block diagram. The hierarchy of the elements, in the general case, is not established, and is arbitrary, but unchanged during the calculation. The iterative process consists of steps, each of which processes one connection. A connection links the output and input ports of the element(s) in the block diagram. It is useful to implement the algorithm for calculating the current connection without exception and without zeroing the rows and columns of the matrices of state equations corresponding to this connection. This corresponds to obtaining an equivalent, block diagram model with input ports available for signaling and output ports open for observation of all elements that make up the block diagram. It is shown that for a linear dynamical system in the case of the presence of a solvable system of algebraic contours in it, it is possible to obtain a mathematical model of the dynamical system in the form of equations of state. A condition is derived under which the system of algebraic contours can be solvable.

Keywords – mathematical model, block diagram, algebraic contour, object-oriented development.

REFERENCES

1. Golnaraghi, F., & Kuo, B. C. (2017). Automatic control systems. McGraw-Hill Education.
2. Varga, A. (2014). Computer-aided control systems design: Introduction and historical overview. Encyclopedia of Systems and Control.
3. Karayanakis, N. M. (1995). Advanced system modelling and simulation with block diagram languages. CRC Press Inc. 350 pp..
4. Mitchell, E. E. & Gauthier, J. S. (1976). Advanced continuous simulation language (ACSL). Simulation, 26(3), p. 72-78.
5. Baumann, D., Solowjow, F., Johansson, K. H., & Trimpe, S. (2020). Identifying causal structure in dynamical systems. arXiv preprint arXiv: 2006.03906.
6. Sinha, R., Paredis, C. J., Liang, V. C. & Khosla, P. K. (2001). Modeling and simulation methods for design of engineering systems. J. Comput. Inf. Sci. Eng., 1(1), p. 84-91.
7. Zhu, A., Pauwels, P., & De Vries, B. (2021). Smart component-oriented method of construction robot coordination for prefabricated housing. Automation in Construction, 129, 103778.
8. Shah, S. C., Floyd, M. A., & Lehman, L. L. (1990). MATRIX: Control design and model building CAE capability. Computer-Aided Control Systems Engineering (M. Jamshidi and CJ Herget, eds.)(Elsevier, Amsterdam, 1985).
9. Grace, A. C. (1991). SIMULAB, An integrated environment for simulation and control. In 1991 American Control Conference (pp. 1015-1020). IEEE.
10. The MathWorks (2019). Simulink user's guide. https://www.mathworks.com/help/releases/R2019b/pdf/doc/simulink/sl_using.pdf. Version R2019b [Online; accessed Jul. 2022].
11. The MathWorks (2020d). MATLAB. <https://www.mathworks.com/products/matlab.html>. [Online; accessed Jul. 2022].
12. Gsellmann, P., Melik-Merkumians, M., Zoitl, A., & Schitter, G. (2020). A novel approach for integrating IEC 61131-3 engineering and execution into IEC 61499. IEEE Transactions on Industrial Informatics, 17(8), 5411-5418.
13. Steindl, G., & Kastner, W. (2021, May). Transforming OPC UA information models into domain-specific ontologies. In 2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS) (pp. 191-196). IEEE.
14. Willard, J., Jia, X., Xu, S., Steinbach, M., & Kumar, V. (2020). Integrating physics-based modeling with machine learning: A survey. arXiv preprint arXiv:2003.04919, 1(1), p. 1-34.
15. Samarsky A. A., Mikhailov A. P. Mathematical modeling: Ideas. Methods. Examples. 2nd ed., rev. - Moscow: FIZMATLIT, 2005. 320 p.
16. Shampine, L. F. (2018). Numerical solution of ordinary differential equations. Published by Routledge. 632 p.
17. Noonburg, V. W. (2020). Differential equations: from calculus to dynamical systems (Vol. 43). American Mathematical Soc..
18. Campbell, S., Ilchmann, A., Mehrmann, V., & Reis, T. (Eds.). (2019). Applications of differential-algebraic equations: examples and benchmarks. Cham: Springer International Publishing.
19. Samarsky A. A., Gulin A. V. Numerical methods: Textbook for universities. - Moscow: Nauka, 1989. 432 p.
20. Yang, W. Y., Cao, W., Kim & oth. (2020). Applied numerical methods using MATLAB. John Wiley & Sons.
21. DeRusso, P. M., Desrochers, A. A., Roy, R. J., & Close, C. M. (1997). State variables for engineers. John Wiley & Sons, Inc..
22. Adegebe, A. A., & Levenson, R. M. (2021). Dynamic multivariable algebraic loop solver for input-constrained control. IEEE Transactions on Automatic Control, 67(6), 3051-3058.
23. Jaskolka, M., Pantelic, V., Wassyng, A., & Lawford, M. (2020). Supporting modularity in Simulink models. arXiv preprint arXiv:2007.10120.
24. Mosterman, P. J., & Cioffi, J. E. (2004, December). Using interleaved execution to resolve cyclic dependencies in time-based block diagrams. In 2004 43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601) (Vol. 4, pp. 4057-4062). IEEE.
25. Knauer, U. & Knauer, K. (2019). Algebraic graph theory. In Algebraic Graph Theory. de Gruyter. <https://doi.org/10.1515/9783110617368>

26. Mosterman, P. J., Manders, E. J., & Biswas, G. (2000, May). Qualitative dynamic behavior of physical system models with algebraic loops. In Eleventh International Workshop on Principles of Diagnosis (pp. 155-162).
27. Jaskolka, M. (2020). Making Simulink Models Robust with Respect to Change (Doctor of Philosophy dissertation), McMaster University Hamilton, ON, Canada. 206 p. https://macsphere.mcmaster.ca/bitstream/11375/26126/2/Jaskolka_Monika_2020Dec_PhD.pdf [Online; accessed Aug 2022]