

Подготовка модели автомата установки компонентов на печатные платы для имитационного моделирования операции сборки

А.Е. Курносенко, А.Е. Шерстюк (МГТУ им. Н. Э. Баумана)

Рассматривается метод создания цифровой модели автомата установки компонентов на печатные платы для последующего проведения имитационного моделирования операции сборки электронных модулей на печатных платах. Решается задача оптимизации последовательности установки компонентов в постановке задачи о коммивояжере. Представлены обобщенный алгоритм процесса и описание его основных этапов.

Ключевые слова: цифровое производство, печатная плата, установка компонентов, дискретное имитационное моделирование.

Введение

Одной из ключевых тенденций современной промышленности является цифровизация производств. Согласно концепции Industry 4.0, это означает внедрение киберфизических систем в производственные структуры, а именно, разработку комплексных цифровых моделей производств, которые объединяют в единое целое модели выпускаемых изделий, технологических процессов, оборудования, оснащения и персонала [1, 2]. Подобные модели позволяют моделировать различные аспекты производственных систем, включая производительность, надежность, точность, дают возможность проигрывать разнообразные сценарии работы производственных систем вида «что, если», помогают увидеть узкие места в производ-

ственной структуре и наметить пути их устранения [3, 4]. Разрабатываемые в настоящее время решения охватывают многие отрасли промышленности. В данной работе задача построения цифровой модели будет рассматриваться для отрасли электронного приборостроения на примере одной из операций автоматизированной сборки электронных модулей на печатных платах.

Объект исследования

Типовой технологический процесс автоматизированной сборки электронного модуля состоит из следующих основных операций: загрузка печатных плат в буферный накопитель, нанесение паяльной пасты, установка элементов, пайка, отмывка, сушка, функциональный контроль,

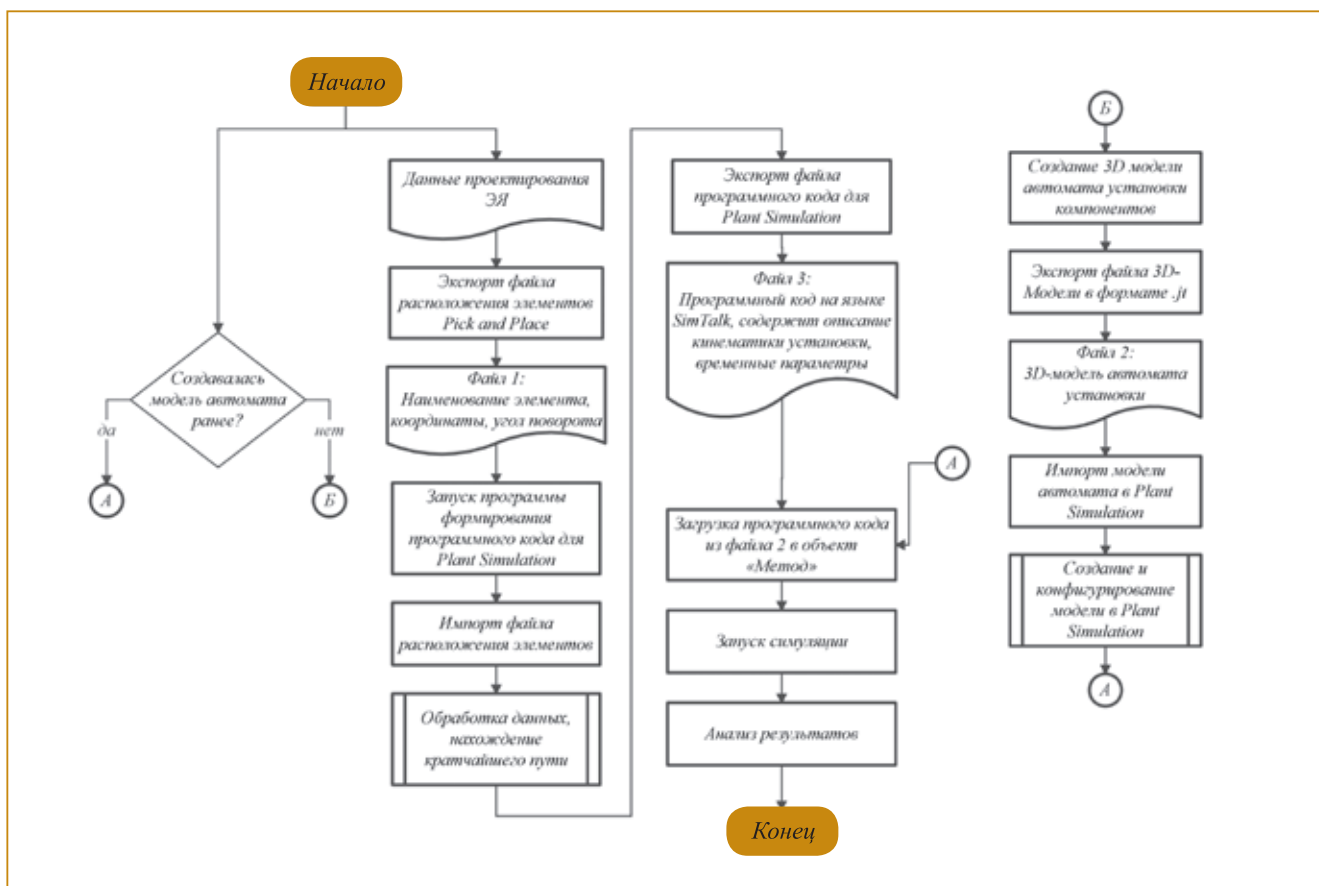


Рис. 1. Блок-схема общего алгоритма операции установки компонентов

выгрузка готовых модулей в буферный накопитель. В условиях мелкосерийного и серийного производства все указанные операции автоматизированы и выполняются на технологической линии в составе нескольких единиц оборудования. Для выполнения операций применяется нанесение паяльной пасты трафаретной печатью, автоматизированная установка компонентов по программе, групповая пайка оплавлением в конвекционной печи, отмывка и сушка в специализированных установках и т.д. Большинство операций являются групповыми: действия над компонентами производятся одновременно, и затрачиваемое время не зависит от их числа. Исключением является операция установки компонентов на печатную плату, так как от числа и координат расположения компонентов напрямую зависит суммарный путь, который проходит сборочная головка автомата при их размещении на плате, а следовательно, и производительность данной операции. Таким образом, операция установки компонентов является наиболее критичной с точки зрения производительности технологической линии и, соответственно, представляет собой потенциальное «узкое место» по производительности сборки. Вследствие этого для цифрового моделирования в данной работе выбрана операция установки компонентов и соответствующее оборудование – сборочный автомат. Работа выполнена на примере автомата Topaz X от компании Assembleon.

Постановка задачи

Среди методов повышения производительности операций установки компонентов выделим следующие: оптимизация наладки питателей, оптимизация группового захвата компонентов из питателей, оптимизация последовательности установки компонентов на плату. Все эти методы можно применять одновременно. В данной работе рассмотрена оптимизация последовательности установки компонентов с целью получения кратчайшего пути обхода сборочной головкой мест установки на печатной плате [5]. Таким образом, необходимо создать цифровую модель автомата, позволяющую определять цикловую производительность сборки с учетом оптимизации последовательности установки компонентов.

Проблему оптимизации последовательности можно описать задачей о коммивояжере в рамках линейного программирования. Коммивояжер, выезжающий из некоторого города, должен посетить каждый из $n-1$ других городов только один раз и вернуться в исходный город. Матрица расстояний между городами известна. Необходимо найти такую последовательность обхода городов, при которой общее расстояние, пройденное коммивояжером, будет минимальным.

Рассмотрим формализованную запись задачи о коммивояжере в терминах сборочной операции. Коммивояжер представляет собой сборочную

головку, последовательно устанавливающую все компоненты на плату, города при этом являются координатами центров установки компонентов.

Математическая модель записывается следующим образом:

$$\sum_{i=1}^n \sum_{j=1}^n d_{ij} x_{ij} \rightarrow \min$$

при

$$\left\{ \begin{array}{l} \sum_{i=1}^n x_{ij}=1; \\ \sum_{j=1}^n x_{ij}=1; \\ j=1,2,\dots,n; \\ x_{ij}=0,1 \forall i,j. \end{array} \right.$$

где n – число компонентов; d_{ij} – расстояние между центрами двух соседних компонентов с номерами i и j .

$X = \|x_{ij}\|_{n \times n}$ – матрица, в которой

$$x_{ij} = \begin{cases} 1, & \text{если в маршрут входит путь} \\ & \text{от компонента } x_i \text{ к компоненту } x_j; \\ 0 & \text{в противном случае.} \end{cases}$$

На решение данной задачи необходимо наложить дополнительные ограничения, связанные с требованием односвязности маршрута:

$$\left\{ \begin{array}{l} \sum_{i=1}^k x_{ij}^{(k)}=1; \\ \sum_{j=1}^k x_{ij}^{(k)}=1. \end{array} \right.$$

где $x^{(k)}$ – любая $(k \times k)$ -мерная подматрица матрицы X , получаемая в результате выбора k диагональных элементов X и выделения соответствующих строк и столбцов. В противном случае будем иметь не один, а два или более замкнутых маршрутов, каждый из которых проходит лишь через часть городов.

Примененные методы и средства

Для решения указанной задачи, исходя из дискретного характера операции установки компонентов, было выбрано имитационное моделирование. Одним из основных достоинств подхода является отсутствие необходимости проведения дорогостоящих и затратных по времени экспериментов в реальных производственных условиях, что обусловило его широкое применение при создании цифровых моделей производств.

Для решения локальной задачи коммивояжера были использованы четыре метода: венгерский

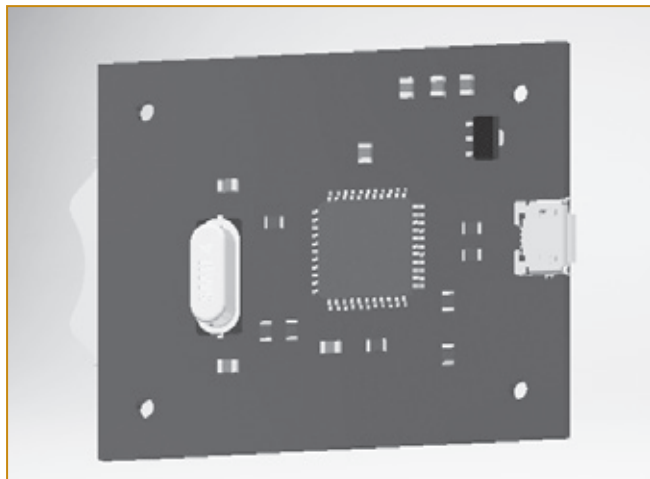


Рис. 2. 3D-модель печатной платы на основе данных проектирования

метод с последующим сшиванием циклов, метод ближнего соседа с локальной оптимизацией и без нее, а также метод многоугольника.

В качестве средства для создания имитационной модели операции установки компонентов был выбран модуль Plant Simulation [6] из состава системы проектирования и моделирования цифровых производств Tecnomatix от компании Siemens Digital Industries.

Описание алгоритма создания цифровой модели и подготовительного этапа

Общий алгоритм создания цифровой модели операций установки компонентов представлен на рис. 1.

Создание цифровой модели начинается с двух процессов, которые можно производить независимо друг от друга: обработки файла данных проектирования электронного модуля (.pcb) и создания 3D-модели самого сборочного автомата. По окончании алгоритма указанная 3D-модель адаптирована в модуле Plant Simulation и содержит необходимую кинематику отдельных узлов автомата, а также временные параметры для конкретной печатной платы, которые будут использованы в дальнейшем моделировании [7].

Методика подразумевает, что у разработчика имитационной модели есть данные проектирования печатной платы, которые содержат, как минимум, размеры печатной платы и посадочные места компонентов с обозначениями каждого компонента, координатами местоположения и углом поворота. Файл данных проектирования (в общем случае он имеет расширение .pcb) можно преобразовать в таблицу расположения элементов Pick and Place с расширением .csv. Совершить эту операцию можно, например, в E-CAD-системе Altium Designer. Данная таблица должна включать следующие столбцы: координату X, координату Y, наименование элемента,

поворот. Данные проектирования предоставляются инженером-конструктором электронного модуля (рис. 2).

Затем экспортированная таблица Pick and Place поступает на вход специально разработанного программного обеспечения (ПО). Цель данного ПО заключается в трансляции таблицы с компонентами и их координатами в текстовый формат языка SimTalk, используемого в модуле Plant Simulation, для описания перемещения блока сборочных головок, фиксирующих устройств, а также создания временных параметров для режимов работы автомата.

Краткий алгоритм работы программы. На вход поступает таблица формата .csv, содержащая четыре столбца. Каждая строка этой таблицы – отдельный компонент электронного модуля. Используя методы объектно-ориентированного программирования, каждый компонент создает экземпляр класса elems. Тот в свою очередь основан на классе типа point (геометрическая точка, содержит координаты и ссылки на ребра между точками). Для каждого ребра высчитывается собственная длина, и присваиваются ссылки на две точки, которое оно соединяет. Число всех ребер между точками рассчитывается по формуле $\frac{n-1}{2} \cdot n$, где n – число электронных компонентов. На этом подготовительный этап работы ПО завершен.

Решение задачи коммивояжера

Далее производится решение задачи коммивояжера для заданных условий. А именно, накладываются следующие ограничения: сборочный блок содержит только восемь установочных головок, следовательно, за один проход возможно захватить и установить восемь компонентов. Таким образом, задача коммивояжера решается раз $\frac{n}{8}$, где n – число компонентов. Группа точек для каждого цикла установки выбирается исходя из длины связей: выбираются восемь наикратчайших из оставшихся. С учетом названных ограничений задача решается только для восьми соседних компонентов.

Кратко рассмотрим методы решения задачи. Венгерский метод основан на ряде эквивалентных преобразований над матрицей размером $n \times n$, где n – число элементов. В каждой ячейке i, j содержится расстояние между городами i и j . Путем последовательной редукции по строкам и столбцам находят нули матрицы таким образом, чтобы выполнялось условие эквивалентности после каждой операции. Но, как правило, венгерский метод образует замкнутые циклы, что для поставленной задачи недопустимо. В этом случае необходимо прибегнуть к их сшиванию, а именно, в каждом и последующем цикле заменяется каждая пара звеньев. Затем высчитывается длина получившегося цикла. После

перебора всех пар звеньев выбирается последовательность с минимальной длиной.

Метод ближнего соседа заключается в присоединении к маршруту города с минимальным расстоянием из еще не выбранных. В качестве усовершенствования метода также применяется алгоритм локальной оптимизации. Путь является локальным оптимумом Z в окрестности $G(Z^0)$, если длина маршрута $C(Z) \leq C(Z^0) \forall Z \in G(Z^0)$. Оптимизация реализуется путем перебора полного перебора всех элементов окрестности.

Идея метода многоугольника заключается в последовательном наращивании числа звеньев замкнутого маршрута, начиная с треугольника, до образования замкнутого многоугольника, вершинами которого являются все города.

Разработанное ПО вычисляет путь для каждого метода и выбирает оптимальный (минимальный по длине). Затем, на основе полученных последовательностей прохождения блока, создается скрипт на языке SimTalk. Часть кода будет повторяться, отличаться будут только массивы координат, число циклов перемещений, а также формула расчета времени.

Описание создания и адаптации 3D-модели

Перед тем как получить имитационную модель, необходимо создать цифровую 3D-сборку самого автомата установки компонентов [8].

На основе чертежей от производителя или измерения реального объекта в программах трехмерного твердотельного проектирования (SolidWorks, Inventor, Solid Edge и др.) строится сборка, состоящая из отдельных узлов автомата. Эти узлы в дальнейшем будут анимированы, поэтому их будет четыре: сборочный блок, подвижная направляющая, блок питателей и поддерживающие плату снизу упоры (рис. 3), однако методика не запрещает делить сборку на большее число узлов в целях увеличения геометрического соответствия цифрового объекта реальному. После создания модели 3D-сборки автомата разработчик сохраняет ее в файле формата JT.

Полученный файл необходимо импортировать в Plant Simulation и адаптировать для дальнейшего использования. Для этого разработчик создает в модуле Plant Simulation новый объект типа «Фрейм». Он представляет собой пространство, где 3D-объекты наделяются служебными атрибутами, дающими возможность проводить дальнейшее моделирование. По умолчанию область окажется пустой. В нее необходимо импортировать ранее сохраненный файл JT, и расположить объект в произвольном месте. В дальнейшем центр объекта устанавливается посередине фрейма с помощью задания координат через свойства объекта.

Необходимо задать соответствующий класс объекта, чтобы автомат получил необходимые свойства сборочного оборудования (время выполнения операции, число циклов до переналадки и пр.).

Для этого разработчик выбирает команду «Сделать объект симуляционным». Здесь задается название автомата (Topaz) и выбирается класс объекта – «Последовательные процессы» – «Сборочная станция». На данном этапе автомат получает свойства по умолчанию.

Объекты симуляции, взаимодействующие с мобильными элементами, должны иметь как минимум вход и/или выход. Для задания местоположения входов/выходов существует отдельный тип объектов «Соединитель». Слева от автомата, то есть в месте, куда будут поступать печатные платы с предшествующей единицы оборудования, устанавливается «Соединитель» с названием «Вход». Справа аналогично создается объект «Выход».

Следующим действием в программе задаются объекты, которые в дальнейшем будут анимированы. Для этого выбирается объект «Сборочный блок» и применяется команда «Сделать объект анимируемым» с названием MountingBlock. Данные названия будут использоваться также и в прикладном ПО, поэтому для корректной работы программ они должны быть одинаковыми. Аналогично данную операцию нужно совершить для подвижной направляющей, блока питателей и поддерживающих упоров с присвоением им названий HFixator, Feeders и VFixator соответственно.

На данном этапе узлы становятся самостоятельными анимируемыми объектами, для которых в дальнейшем появится возможность назначать кинематику и временные параметры процесса.

В модели создается объект типа «Метод». Он необходим, чтобы создавать, изменять и удалять другие объекты и атрибуты при помощи языка SimTalk. На текущем этапе файл остается пустым, однако ему необходимо создать связь с входом автомата. Это делается для того, чтобы скрипт в файле активировался тогда, когда мобильные объекты (печатные платы) будут поступать на вход модели автомата.

Для реализации данного действия следует на вкладке «Событие» выбрать созданный объект «Метод» у поля «Вход».

Модель автомата установки компонентов (рис. 4) готова. Для активации моделирования объекта внутри модуля Plant Simulation необходим следующий минимальный набор объектов: на вход автомата устанавливается объект типа «Источник»,

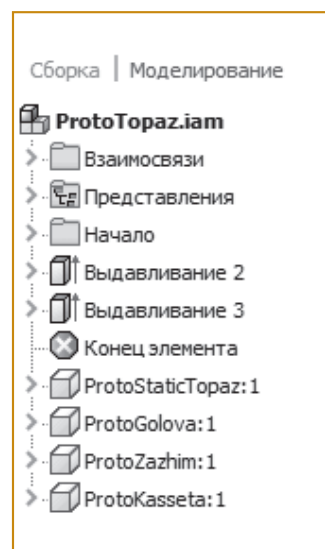


Рис. 3. Дерево объектов сборки автомата

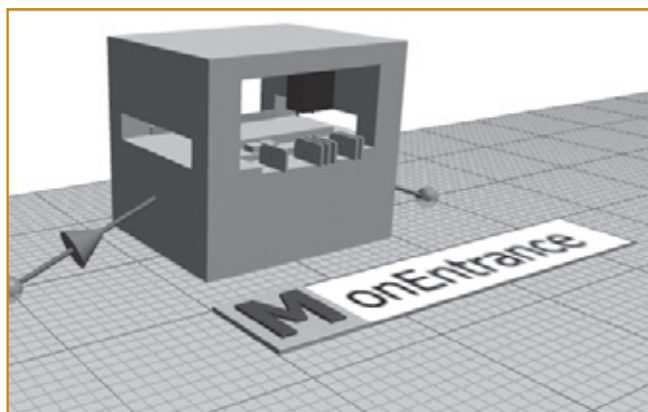


Рис. 4. Внешний вид имитационной модели автомата установки компонентов

который генерирует объекты типа «Мобильный компонент», на выход устанавливается объект типа «Сток». После нажатия на кнопку активации Plant Simulation запустит анимацию: создаст модель печатной платы, она поступит на вход автомата, передвинется к рабочей области, подвижная направляющая сдвинется на расстояние, необходимое для обеспечения фиксации платы. Затем установочный блок из исходной точки начнет передвижение от питателей через камеры системы технического зрения к печатной плате. После установки всех компонентов фиксирующие устройства освобождают печатную плату, и конвейер перемещает ее на выход объекта-автомата.

Заключение и выводы

Анимация объекта обеспечивает не только визуальную наглядность процесса. Время, затрачиваемое на перемещение подвижных узлов, варьируется в зависимости от числа компонентов и конструкций печатных плат. Оно учитывается в служебных атрибутах модуля Plant Simulation, блокируя объект для внешних событий до тех пор, пока не завершит операцию установки компонентов. Таким образом модель точно отражает поведение реального физического объекта.

Среди достоинств предлагаемой методики отметим, что после создания модели автомата в среде Plant Simulation пользователю для каждого нового электронного модуля достаточно сгенерировать новую таблицу Pick and Place, загрузить в ПО, получить код SimTalk и заменить его в объекте «Метод». Plant Simulation автоматически пересоздаст анимацию для новой платы и пересчитает время, на которое будет занят автомат.

Предложенная методика создания цифровой модели автомата установки компонентов позволяет оценить и одновременно оптимизировать его производительность,

Курносенко Алексей Евгеньевич – доцент, **Шерстюк Артем Евгеньевич** – магистрант кафедры ИУ-4

«Проектирование и технология производства электронной аппаратуры»
Московский государственный технический университет им. Н. Э. Баумана
(национальный исследовательский университет).

Контактный телефон 8(499)263-65-53.

E-Mail: akurn@bmstu.ru sherstyukae@student.bmstu.ru

составив оптимальную последовательность установки одним из четырех предлагаемых методов. Методика универсальна в части моделирования технологических операций и может быть применена, в том числе и для групповых процессов. Объединяя построенные на ее основе цифровые модели по входам/выходам, можно получить комплексную модель технологической сборочной линии.

Список литературы

1. Shakhnov, V.A., Kurnosenko, A.E., Demin, A.A., Vlasov, A.I. Industry 4.0 Visual Tools for Digital Twin System Design. *Advances in Intelligent Systems and Computing*, 2020, 1295, 864–875.
2. Курносенко А.Е., Шахнов В.А. Цифровая трансформация при подготовке производства изделий электроники. Автоматизация. Современные технологии, 2021. Т. 75. № 2. С. 51–56.
3. Шахнов В.А., Курносенко А.Е. Моделирование цифрового производства электронной аппаратуры в рамках концепции "Индустрия 4.0" // *Цифровая трансформация промышленности: тенденции, управление, стратегии*. Тр. I международной научно-практической конф. 2019. С. 585–594.
4. Alexey E. Kurnosenko, Ivan V. Levin, Valeria O. Semenyakina, Evgeny R. Zakharov and Artem E. Sherstyuk. Simulation Modeling Methods and Tools in the Study of Electronics Preproduction. *ITM Web of Conferences* 35, 04015 (2020).
5. Kurnosenko A.E. and Arabov D.I. Optimization of electronic components mounting sequence for 3d mid assembly process Proc. Breakthrough directions of Scientific Research in NRNU MEPhI: Development Perspectives in the Framework of the Strategic. *KnE Engineering* 3(6), 2018, 311–321. DOI: 10.18502/keg.v3i6.3009.
6. Курносенко А.Е., Левин И.В., Семенякина В.О., Захаров Е.Р., Шерстюк А.Е. Методы и средства имитационного моделирования при изучении технологической подготовки производства электронной техники // В сб.: *Цифровые технологии в инженерном образовании: новые тренды и опыт внедрения*: Сб. трудов международного форума. 2020. с. 310 – 314.
7. Левин И.В., Курносенко А.Е., Машина Н.А. Решения для проектирования, моделирования и оптимизации производств электронной аппаратуры // *Информационные технологии в проектировании и производстве*. 2018. № 1. с. 26 – 32.
8. Курносенко А.Е., Никольский Т.В., Ломаченко А.С. Подготовка модели оборудования к выполнению имитационного моделирования В САПР // *Информационные технологии в проектировании и производстве*. 2020. № 2. с. 24 – 30.