

Опыт применения трёхзначной логики в управлении технической системой

Григорий Беднов (РТУ МИРЭА)

В статье рассмотрена адаптация трёхзначной логики Брусенцова в управлении некомпьютерной техникой, произведена оригинальная попытка структурировать логику работы устройств на примере разработки распределительно-дозировочного автомата, а также сочетать базовый анализ ситуации с управлением на полевом уровне.

Введение

В процессе цифровизации техники одним из важнейших видов данных является логический. Это подтверждается, например, языками программирования МЭК 61131-3 (LD, FBD, ...), применяемыми в программируемых логических контроллерах (ПЛК) [1].

Однако какая логика лежит в основании данного типа данных – вопрос, решаемый по-разному. Рассматривая лишь дискретные конечнозначные логики, стоит обратить особое внимание на двухзначную и трёхзначную логики.

«Противоборство» двух этих систем – давнее [2], но особой вехой стало построение троичных компьютеров «Сетунь», которые в первый и на данный момент единственный раз произведены серийно; увы, их новая, значительно переработанная, версия «Сетунь-70» не успела попасть в серийное производство.

Группа под руководством Николая Петровича Брусенцова (главный конструктор «Сетуний») после запрета на разработку новых компьютеров работникам вузов занялась в том числе подробным теоретическим анализом накопленного опыта, согласованием с классической двухзначной булевой алгеброй и отображением логики естественного языка [2, 3].

Основная часть

Целью работы является разработка конечного распределительно-дозировочного автомата для управления транспортно-технической системой с тремя резервуарами с жидкостью и шестью бункерами (включая сливной) с использованием трёхзначной логики Брусенцова.

Объектом управления является типичная простая транспортно-тех-

ническая система: на конвейерной линии установлен контейнер на поворотнике, ограниченном в движении лишь углами верхней полуокружности (от 0° до 180°). Имеются три поворотных датчика (нормальное положение 90° и два крайних (0° и 180°)). Над конвейерной линией расположены резервуары (подача жидкости), с краёв установлены бункеры (приём жидкости). В местах, где они расположены, установлены датчики положения. Также на краях конвейерной линии установлены аварийные «сторожевые» датчики положения.

Разрабатываемый управляющий автомат должен быть:

- 1) безопасным и надежным, для чего:
 - a) исправно работать, если все необходимые для этого устройства работают исправно;
 - b) предупреждать оператора о поломках, несовместимых с продолжением циклической работы;
- 2) универсальным и расширяемым, для чего:
 - a) согласовываться с однородными ему автоматами, в идеале – перепрограммироваться;
 - b) начинать свою циклическую работу в любой изначально заданной конфигурации (возможно, оставшейся после аварийного отключения);
 - c) схема управляющего автомата (алгоритм его действий) должна быть спроектирована прежде всего «сверху вниз» с использованием понятных человеку высокоуровневым и абстрагированным от оборудования процедур;
 - d) легко расширяться дополнительными анализирующими модулями, которые позволят автомату встроиться в сложные управляющие системы более высокого уровня (например, MES);

3) эффективным – автомат не должен совершать лишних действий, не относящихся к реализации п. 1, 2.

Для начала работы контейнер К должен быть опустошённым и находиться в нормальном положении под резервуаром, с которого начнётся программа 1.

Программа 1 должна быть начата, когда контейнер К находится в нормальном положении под резервуаром R3 – это необходимое условие. Необходимо налить жидкость из резервуара R3, смешать с жидкостью из резервуара R1, вылить всё в бункер B5. По окончании программы должны быть созданы условия для программы 2 и т.д.

Окажется, что даже в таком нехитром наборе действий нештатных ситуаций много: может быть сломан любой из задействованных датчиков и исполнительных механизмов. При этом для определённости будем считать, что, поскольку датчики положения являются нормально разомкнутыми контактами, они могут сломаться, только давая ложноотрицательный сигнал (будем считать, что вероятность ложноположительного сигнала несоизмеримо меньше). Также будем считать, что датчики достаточно надёжны, чтобы не ломаться одновременно.

Однако как же мы, даже при таких допущениях, можем упорядочить управление и анализ происходящего? На помощь приходит теория трёхзначной логики.

Введём особую функцию, открытую Н.П. Брусенцовым, которая является обобщением дизъюнкции на множество – трёхзначная дизъюнкция по множеству, «логический интеграл» ниже:

$$\bigvee_{x \in X} \begin{cases} 1, \text{ если } x \text{ найден (или доказано, что будет найден)} \\ 0, \text{ если } x \text{ не будет найден} \\ \sigma \text{ (неопределённость), если пока неизвестно} \end{cases}.$$

Данный подход подобен тому, как мы, доставая из сумки разные вещи, пытаемся найти нужную, и только найдя нужную либо перебрав их все (или вовсе обнаружив незнакомые

нам посторонние предметы и поняв, что произошла путаница, и эта сумма вообще не наша), мы можем закончить поиски.

Рассмотрим применение трёхзначной логики на примере поворотного устройства, состоящего из трёх датчиков поворота и исполнительного механизма.

По умолчанию трёхзначные дизъюнктивные состояния всех датчиков считаются неизвестными, не встреченными, так как мы не знаем, срабатывают ли они или они уже сломаны. В частности, датчик нормального положения $VDN = 0$. Очевидно, когда встречены датчики поворота налево и направо $VDKL \cdot VDKR = 1$, но не встречен центральный по положению датчик нормального поворота, он уже не сработает (ведь поворотник проехал его, и ничего не произошло). Кроме того, после срабатывания $DN = 1$, само собой, $VDN = 1$.

Аналогично с датчиками положения на конвейерной линии, с той лишь разницей, что там все датчики (кроме крайних, которые не используются при работе и являются аварийными) контролируются своими соседями указанным выше образом.

Кроме того, заметим: помимо того, что надо «запоминать» посещённые и пропущенные датчики, нужно также научиться их «забывать». В нашем случае кажется разумным решение, что система будет запоминать факты отсутствия датчиков до своей полной перезагрузки по сигналу STOP (ведь ситуация с нерабочим датчиком, в общем-то, уже является аварийной), а из посещённых датчиков будет «помнить» и отмечать как достоверно существующий только последний встреченный датчик. Таким образом, система удаляет единицу с интегральной трёхзначной дизъюнкции датчика, если соседнее с ней значение также оказалось равно единице. Если в процессе работы соседний датчик не обнаружился, то значение сохраняется. Пока датчик сломан, но система по каким-то причинам продолжает работать (например, неполадка обнаружена, но датчик не задействуется), логика сломанного датчика должна переключаться в режим трансляции, связывая два соседних с ней датчика, как будто они теперь стали соседними.

Что касается состояний $VDKL$ и $VDKR$ для задач поворота контейнера налево и направо, то, к сожалению, они могут быть сломаны, либо может не работать поворотный механизм. Различить две эти ошибки невозможно, так как датчики поворота не были оснащены аварийными контрольными датчиками.

Заключение

Полученная система даёт самые базовые функции SCADA-системы, не являясь ею, перенося вычисления на уровень соответствующего ПЛК в соответствии с концепцией вычислений «на краю» либо, в теории, вовсе исключая контроллер из управления.

Однако, прежде всего, полученная система – управляющая, причём выстроенная сравнительно однотипно и, при необходимости, децентрализованно.

Чтобы понять, в какую сторону следует направлять исполнительный механизм, нужно:

- выбрать датчик положения (или поворота, зависит от подсистемы), возле которого нужно остановиться;
- определить, с какой стороны находится последний встреченный датчик этой подсистемы относительно выбранного (функция знака трёхзначна);
- начать движение в соответствующем направлении до тех пор, пока дизъюнктивное состояние выбранного датчика не станет определённым (утвердительным при успехе и логически отрицательным при неудаче).

Данный подход позволяет сформировать свод базовых команд. Мы видим, что усложнение логики работы устройств в самом основании заметно упростило построение вышестоящей системы, что оправдывает использование трёхзначной логики в технике.

Если мы допускаем, что наши датчики «умные» и способны сами «общаться» между собой (хотя бы каждый со своими соседями), то в самом рабочем цикле центральный контроллер не нужен вовсе либо нужен лишь как задатчик правил и шагов межмашинного взаимодействия (получение производственного задания от MES-системы), что вполне близко современным идеям Интернета вещей. Закономерное развитие технологий в этом

направлении может подвести нас к использованию языка программирования Forth (совместимого с трёхзначной логикой – его аналог использовался в «Сетуни-70» [2]) и дальнейшей глубокой интеграции с технологическими информационными системами по путям, достаточно близким к описанным в [4].

Хочется также заметить, что для передачи требуемого от алгоритма направленного сигнала между двумя умными датчиками вполне достаточно как раз одного трёхзначного канала, работающего следующим образом:

- узел А посылает в качестве сигнала «-1» и принимает значение «1»;
- узел Б посылает в качестве сигнала «1» и принимает значение «-1»;
- любой узел, приняв сигнал от противоположного, выставляет значение канала «0», освобождая его.

Построение цифровых схем управления (реальных или виртуальных), связывание их в сети – путь, целиком соответствующий предоставляемым трёхзначной логикой возможностям, и разработка распределительно-дозировочного автомата явно показала: применение трёхзначной логики в управлении технической (некомпьютерной) системой возможно, из теории трёхзначной логики адаптировано в практическую плоскость понятие дизъюнкции по множеству V_x , разрабатываются оригинальные методы.

Литература

1. ГОСТ Р МЭК 61131-3-2016. Контроллеры программируемые. Ч. 3. Языки программирования. М.: Стандартинформ, 2016.
2. Владимирова Ю.С. О логических исследованиях Н.П. Брусенцова // Сборник трудов SoRuCom-2017. Четвёртая международная конференция «Развитие вычислительной техники в России и странах бывшего СССР: история и перспективы». Зеленоград, 2017 г. М.: ФГБОУ ВО РЭУ им Г.В. Плеханова, 2017. С. 26–29.
3. Брусенцов Н.П. Начала информатики. М.: Фонд «Новое тысячелетие», 1994. 176 с.
4. Romanov M.P., Kasbirskaia E.N., Romanov A.M., Kholopov V.A., Kharichenko A.I. A novel architecture for the executive-level control systems for a discrete machinery production // Mechatronics, automation, control. М.: Издательство «Новые технологии», 2017. С. 64–72.

