

АНАЛИЗ И МОДЕЛИРОВАНИЕ КОМАНД ДВИЖЕНИЯ ИНТЕЛЛЕКТУАЛЬНОГО МОБИЛЬНОГО РОБОТА

Существует несколько методов построения моделей управления движением интеллектуального мобильного робота (ИМР). Это PID - регуляторы, нечеткая логика, искусственные нейронные сети, нелинейное управление [1]. Конечным результатом этих методов есть управляющие воздействия, которые поступают на исполнительные блоки ИМР: двигатели и различные логические блоки. Однако есть множество действий ИМР, которые нельзя или трудно формализовать математически. В этом случае управляющие воздействия должны поступать на блоки ИМР в виде команд. Без командного управления нельзя обойтись и в иерархических системах управления, где управляющие воздействия могут поступать из различных источников, в качестве которых выступают системы управления верхнего уровня и оператор. Следовательно, задача разработки системы команд управления ИМР, как универсального средства взаимодействия различных систем и подсистем управления с исполнительными блоками ИМР, является актуальной.

Использование командного управления переносит усилия разработчика программного обеспечения системы управления движением с написания программ на языке программирования на более высокий уровень – составление программ функционирования блоков ИМР. Преимуществом программ функционирования блоков является возможность их тщательной проработки и отладки, а также возможность оперативной модификации в условиях испытаний и эксплуатации. Они могут быть получены по результатам работы алгоритмов управления, т.е. по математическим моделям управления движением или по результатам опытных и эксплуатационных испытаний. Это означает, что часто встречающиеся действия ИМР в сходных условиях внешней среды можно запрограммировать некоторым оптимальным образом и затем при появлении подобных условий использовать готовые программы движения.

Кроме этого, одно и то же действие можно выполнить по-разному: быстрее или медленнее, с разной степенью точности, с разными энергетическими затратами и т.д. Таким образом, различные действия ИМР можно представить как набор программ движения. Выбор программы для выполнения можно осуществлять, используя методы искусственного интеллекта. В простейшем случае это может быть выбор по одному или нескольким критериям. Но, прежде чем говорить о выборе, определим свойства и набор команд для реализации управления по командам.

Система команд должна обеспечивать выполнение ИМР и его блоками физических действий (управлять скоростью вращения двигателей, таймерами, датчиками) и логических действий, связанных с определением состояния ИМР и его блоков, переключением этих состояний и взаимодействием с вышестоящими системами.

Рассмотрим некоторые особенности команд управления исполнительными блоками на примере процесса набора скорости ИМР из начального неподвижного положения (начальная скорость равна нулю) до заданного значения (рис.1.).

Чтобы тронуться с места, необходимо развить достаточное усилие на валу силового электродвигателя. Это достигается путем подачи на двигатель команды, устанавли-

вающей скоростью движения V_r , достаточную, чтобы развить это усилие. Затем скорость уменьшается до требуемого значения V .

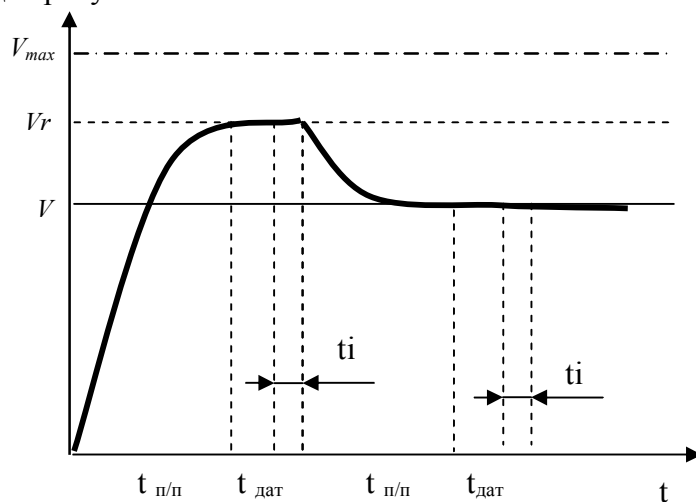


Рис. 1. Динамика набора скорости ИМР.

Алгоритмически этот процесс будет состоять из следующих этапов: включить скорость вращения V_r ; после истечения времени переходного процесса $t_{п/п}$ и времени срабатывания датчика вращения двигателя $t_{дат}$ сравнить полученное значение с исходным; если оно не превышает заданную погрешность, то установить требуемую скорость движения V ; после истечения времени $t_{п/п}$ переходного процесса и времени, необходимого для срабатывания датчика скорости вращения $t_{дат}$, сравнить полученное значение скорости с исходным и выполнить некоторые действия в случае их несовпадения.

Очевидно, что для реализации подобного управления необходима команда, задающая скорость вращения, и команда ожидания истечения заданного времени. В целом, кроме отсчета времени, в системе управления движением необходимо ожидание и других событий: например, реакции датчиков ИМР (пройденного расстояния, угла поворота, скорости движения и др.).

Команды, выполняющие подобные действия, отнесем к самому низкому уровню управления. Этот уровень будем называть автономным. С командами данного типа вышестоящим системам оперировать неудобно, так как необходимо учитывать внутреннее состояние ИМР и множество других факторов, в результате чего для систем верхнего уровня увеличиваются затраты на непроизводительные рутинные действия, увеличивается нагрузка на канал связи и усложняется реализация управления в реальном масштабе времени.

Конструктивно и логически систему управления движением ИМР можно представить как набор параллельно функционирующих блоков [2]. Такая структура упрощает построение динамической модели системы управления и повышает ее живучесть, так как в случае возникновения исключительной ситуации в одном из блоков, остальные продолжают функционировать.

Детальный анализ процессов, происходящих в системе управления [2], привел к следующим выводам:

1. В системе команд должны быть команды двух уровней: высокого и низкого. Высокий уровень для управления ИМР как единым целым и низкий - для непосредственного управления блоками ИМР.
2. Блоки, которые управляются командами, должны иметь собственные наборы команд и программный модуль (назовем его командный процессор) выполняющий эти команды. Это позволит реализовать принцип параллельного и независимого функционирования блоков.
3. Для реализации управления от разных независимых источников необходимы команды, реализующие одинаковые действия, но выполняющиеся в разных режимах: системном и автономном. Это необходимо для исключения конфликтов между выполняющимися в данный момент командами и поступающими извне.
4. Порядок следования команд разных блоков в потоке команд может быть произвольным.
5. Программные блоки реагируют только на «свои» команды.
6. Физическое действие или маневр ИМР представляется как последовательность элементарных действий, задаваемых с помощью команд данной системы. Так как действия ИМР имеют протяженность во времени, а команды программы маневра могут поступить все сразу, то каждый блок должен иметь специальный буфер для временного накопления команд.

Следуя данным выводам, все многообразие команд разбито на две группы: команды, которые управляют ИМР как единым целым (табл. 1), и команды, управляющие блоками ИМР (табл. 2). Команды, управляющие ИМР как единым целым, выполняются только в системном режиме. Однако системам верхнего уровня доступны и команды (табл. 2.). Это позволяет при необходимости системам верхнего уровня все управление взять на себя, и, в свою очередь, системе управления движением ИМР самостоятельно управлять ИМР по заложенным в нее алгоритмам в случае отсутствия систем верхнего уровня или потери с ними связи.

С помощью команд системы верхнего уровня могут получить информацию о текущем состоянии ИМР; изменить состояние блоков ИМР; выполнить любой маневр из множества маневров, хранящихся в памяти ИМР; имитировать наступление событий.

Табл. 1.

Команды управления ИМР

| Мнемоника | <i>Действие</i> |
|------------------|---|
| SETL_V | Установить линейную скорость V. |
| SETA_G | Установить угол поворота передней оси G. |
| DIST_D | Пройти заданное расстояние D |
| DATA_I_D | Передать ИМР информацию вида I с численным значением D (координаты ИМР, траектории, препятствий и т.п.) |
| SETP_N | Выполнить автономную программу с номером N |
| STOPPR | Остановить (сбросить) автономную программу ИМР |
| SET_E_M | Установить условие M наступления события с номером E. |

Табл. 2

Команды управления блоками ИМР

| Мнемоника | Действие |
|-----------|--|
| INIT_N | Выполнить начальный сброс блока с номером N |
| SETS_N_S | Установить слово состояния S блока с номером N |
| GETS_N | Получить состояние блока с номером N |
| ROT_NW | Устанавливает направление и скорость вращения W двигателя или колеса с номером N. |
| WAIT_NE_M | Устанавливает условие M наступления события с номером E для блока с номером N и ожидает его. |
| SET_NE_M | Аналогична команде WAIT_NE_M, но наступления события не ожидает. |

Автономный и системный режимы выполнения команд позволяют избежать конфликтов при выполнении автономной программы и поступлении системной команды. В ИМР команды движения высокого уровня представляются последовательностью автономных команд или программ для соответствующих блоков и передаются этим блокам для выполнения. Это позволяет комбинировать системный и автономный режимы, т.е. системная команда может запустить автономную программу ИМР. При этом автономные команды в потоке команд от внешних систем игнорируются.

Каждый блок имеет полный набор команд (табл. 2.). «Свои» команды блоки определяют по номеру N, который является частью кода операции команды. При разработке формата команд учитывалась возможная дальнейшая модернизация команд и введение новых. Поэтому для команд выбран формат 4 байта, позволяющий закодировать большое число комбинаций. К тому же, в настоящее время это самый распространенный формат микропроцессоров. Формат команды представлен на рис.2. Команда имеет поле «Код операции» и два «поля параметров». Параметры 1 и 2 определяются в зависимости от назначения команды. Параметром 1 может быть угловая скорость вращения двигателя или колеса, номер события, координаты положения ИМР, координаты траектории и др. Параметр 2 задает численное значение параметра 1.

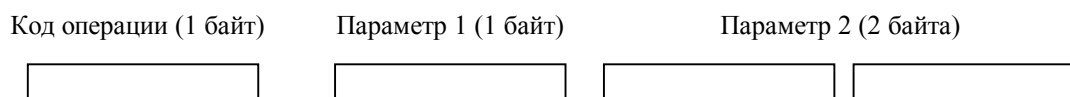


Рис.2. Формат команды ИМР.

В системе принято, что команды, связанные с приемом и передачей информации, изменением состояний выполняются мгновенно после поступления, т.е. не имеют динамики. Алгоритмы команд, связанных с управлением электромеханическими блоками и событиями, развертываются во времени, т.е. имеют динамику. Программная реали-

зация таких команд основана на объектно-ориентированном представлении динамических процессов с помощью диаграмм состояний и событий [3]. По данному методу также построена динамическая модель командного процессора рис.3. программных блоков ИМП. Диаграмма учитывает все возможные варианты состояний командного процессора при выполнении команд. Все эти состояния строго определены. С каждым из них связаны определенные действия, которые выполняются, когда процессор переходит в это состояние. Переход в новое состояние вызывают определенные события, генерируемые во времени.

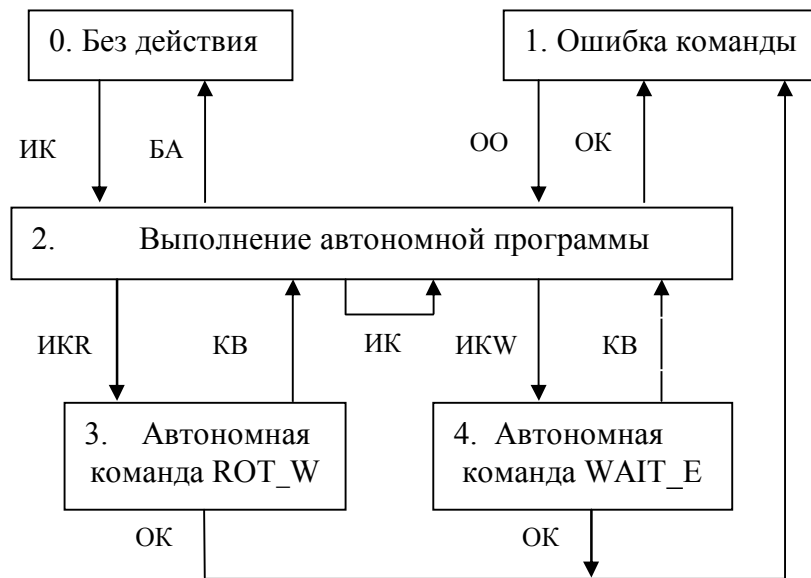


Рис. 3. Диаграмма состояний командного процессора блока двигателей.

На данной диаграмме действуют события: ИК – идентифицирована команда; БА – буфер автономных команд пуст; ОО – ошибка команды обработана; ОК – ошибка команды; ИКР – идентифицирована команда ROT_W; КВ – команда выполнена; ИК – идентифицирована команда, не требующая развертывания во времени; ИКW – идентифицирована команда WAIT_E.

При отсутствии в буфере команд процессор находится в состоянии 0 до появления события ИК, которое переводит его в состояние 2, где производится анализ очередной команды из буфера. Если очередная команда не имеет алгоритма с ожиданием событий, то она сразу же выполняется и процессор опять возвращается в состояние 2. Если очередная команда связана с электромеханическими блоками, т.е. произошло событие ИКР (команда ROT_W) или с управлением событиями (команда WAIT_E), то процессор переходит соответственно в состояние 3 и 4, из которых возвращается в состояние 2 по событию КВ в случае отсутствия ошибок или переходит в состояние 1 в случае обнаружения ошибки во время выполнения команды (событие ОК). Событие ОК также поступает в автономную систему управления ИМП, где анализируется специ-

альным блоком обработки ошибок. Из состояния 1 процессор может выйти, если поступит событие ОО, которое генерируется блоком обработки ошибок, учитывающего характер ошибки, состояние блоков ИМР и принимающего решение о возможности продолжения работы процессора команд.

Состояния 3 и 4 имеют собственные диаграммы состояний рис.4 и рис.5. Эти диаграммы используются как для автономных, так и для системных команд. Это упрощает программную реализацию команд, т.к. диаграмма состояний команды в данном случае не зависит от режима выполнения команды. Диаграммы состояний команд позволяют в любой момент времени определить на каком этапе выполнения находится команда и правильно выйти из нее при сбросе или останове программы движения, а также фиксировать ситуации не наступления событий, например, в случае отказа датчиков.



Рис. 4. Диаграмма состояний команды WAIT_E

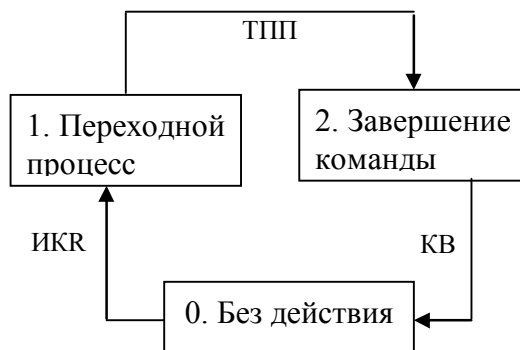


Рис. 5. Диаграмма состояний команды ROT_NW

На диаграмме команды WAIT_E действуют следующие события: ИКВ – идентифицирована команда WAIT_E; ИКТ – установить таймер задержки; ИКД – установить признак датчика; ТЗ – таймер задержки истек; СД – состояние датчика достигнуто. В этой команде после наступления ожидаемого события формируется соответствующее событие, которое поступает на диаграмму процессора команд и в систему управления, где может использоваться для синхронизации процессов, происходящих в системе управления.

На диаграмме команды ROT_NW действуют события:

ТПП – таймер переходного процесса истек, ИКР – идентифицирована команда ROT_NW, КВ – команда выполнена. Эта команда в основном используется для управления электродвигателями. В команде при переходе в состояние 1 запоминается текущее состояние датчика пройденного расстояния колесом, связанным с данным двигателем или датчика вращения вала двигателя, если двигатель используется для других целей. В состоянии 2 этот датчик анализируется и генерируется событие КВ (команда

выполнена), если показания датчика в пределах нормы или ОК (ошибка команды), если нет. Эти события, как и в случае с командой WAIT_E, поступают на диаграмму состояний процессора команд и в систему управления.

Обсуждаемая система команд была опробована и используется в системе управления мобильным роботом “Intelligence-10”, разработанном в Донецком государственном институте искусственного интеллекта. Система команд позволила разработать программы движения (разгон, торможение, разворот на заданный угол и др.), в которых параллельно функционируют три силовых двигателя. Достоинством этих программ является простота их модификации: изменение и добавление новых команд осуществляется просто, т.к. они записаны в отдельном текстовом файле и не требуют перекомпиляции программного обеспечения системы управления. В дальнейшем предполагается использовать эту систему команд для манипулятора, устанавливаемого на тележку ИМР. Данная система команд будет совершенствоваться в плане расширения команд высокого уровня и приближения их к естественному человеческому языку. Например, введением команд «Двигаться осторожно», «Двигаться в оптимальном режиме» и т.п.

ЛИТЕРАТУРА

1. Бурдаков С.Ф., Первозванский А.А., Смольников Б.А., Стельмаков Р.Э., Штайнле С.В., Юдин В.И.// Проблемы создания мобильных роботов с элементами искусственного интеллекта // Материалы VIII международной научно-технической конференции «Экстремальная робототехника». СПб., 1997. С. 5÷13.
2. Звенигородский А.С., Вороной С.М. Принципы построения систем управления движением интеллектуальных роботов в условиях неопределенности внешней среды // Искусственный интеллект, 1999 №1, С.348-353.
3. Шлеер С., Меллор С./ Объектно-ориентированный анализ: моделирование мира в состояниях: Пер. с англ. – Киев: Диалектика, 1993. - 240 .:ил.