

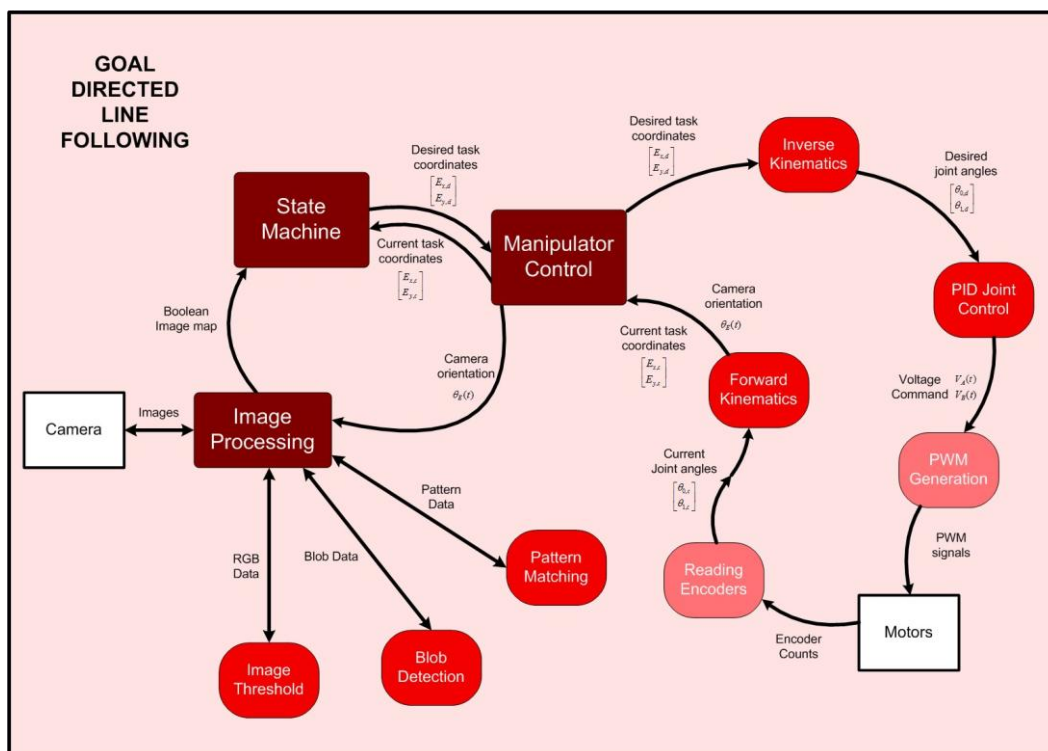
Движение по линии

Рассматриваемые темы

- Движением по линии с помощью обработки изображений, управления манипулятором и конечных автоматов
- Движением по линии в QNET Mechatronic Systems

Предварительные требования

- QNET Mechatronic Systems настроены в соответствии с кратким руководством по началу работы.
- Выполнена лабораторная работа "Управление манипулятором"
- Выполнена лабораторная работа "Обработка изображений"
- Выполнена лабораторная работа "Конечные автоматы"



1 Сведения из теории

Движение по линии объединяет обработку изображений, управление манипулятором и конечные автоматы, позволяя манипулятору QNET Mechatronic Systems автономно перемещался из точки А в точку В, двигаясь только по дороге. Для этого должны выполняться следующие функции:

1. Система должна ожидать, пока пользователь введет конечную точку назначения. Это первая точка на дороге, по которой щелкнет пользователь.
2. При необходимости пользователь может ввести дополнительные точки на маршруте. Эти точки добавляются щелчком по дороге при нажатой кнопке CTRL. Обратите внимание, что система должна следовать по этим точкам в том порядке, в котором они были введены, и, наконец, достичь конечного пункта назначения. Все эти точки добавляются в очередь. По готовности пользователь нажимает кнопку GO!
3. Система переходит в точку в начале очереди, следуя далее только по дороге.
4. Когда манипулятор попадает в окрестность желаемой точки, эта точка удаляется из очереди и повторяет шаг 3.
5. Когда манипулятор достигает конечного пункта назначения, VI завершает работу.

Важно, чтобы три системы обменивались информацией, как показано на рисунке 1.1.

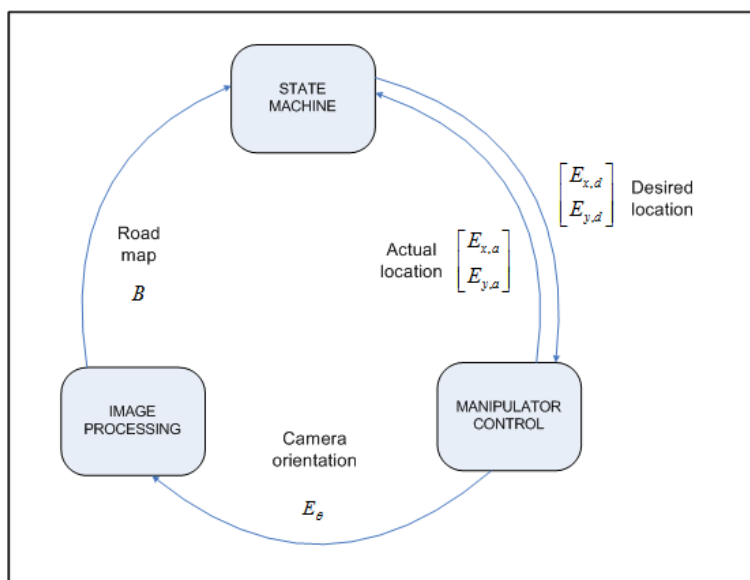


Рисунок 1.1: Обмен данными между тремя системами при движение по линии

State machine – конечный автомат, Manipulator Control – управление манипулятором, Image Processing – обработка изображения, Actual Location – настоящее положение, Camera Orientation – ориентация камеры, Road Map – карта дорог

1.1 Управление манипулятором

Цикл управления манипулятором очень похож на тот, что вы разработали в лабораторной работе "Управление манипулятором". Он получает требуемые координаты рабочего органа из конечного автомата. Он использует обратную кинематику (см. лабораторную работу "Инверсная кинематика"), для преобразования их в необходимые обобщенные координаты, и ПИД-компенсатор (см. лабораторную работу "ПИД-управление положением") для перевода манипулятора в эти координаты. Он преобразует текущие обобщенные координаты в реальные координаты задачи, используя прямую кинематику (см. лабораторную работу "Прямая кинематика") и передает эту информацию в конечный автомат. Прямая кинематика используется также для передачи ориентации камеры в цикл обработки изображения. Это показано на рисунке 1.2.

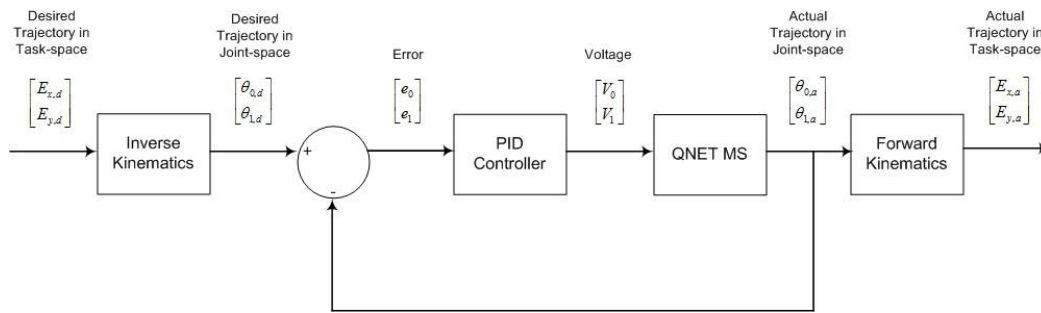


Рисунок 1.2: Цикл управления манипулятором

Inverse Kinematics – инверсная кинематика, PID Controller – ПИД-контроллер, Forward Kinematics – прямая кинематика, Desired Trajectory in Task-space – требуемая траектория в пространстве задачи, Desired Trajectory in Joint-space – требуемая траектория в пространстве обобщенных координат, Error – погрешность, Voltage – напряжение, Actual Trajectory in Task-space – реальная траектория в пространстве задачи, Actual Trajectory in Joint-space – реальная траектория в пространстве обобщенных координат

1.2 Обработка изображения

Цикл обработки изображений получает полутоновое изображение из камеры QNET Mechatronic Systems и поворачивает его, используя информацию об ориентации камеры из цикла управления манипулятором. Он использует пороговое значение для преобразования изображения в двоичный формат (см. лабораторную работу «Пороговая обработка изображения») и выполняет ресэмплирование изображения для создания двоичной карты. Двоичная карта - небольшое двоичное изображение, которое вместо числа для каждого местоположения пикселя содержит булевскую переменную со значением TRUE (дорога) или FALSE (не дорога). Он передает эту двоичную карту в конечный автомат. Это показано на рисунке 1.3.

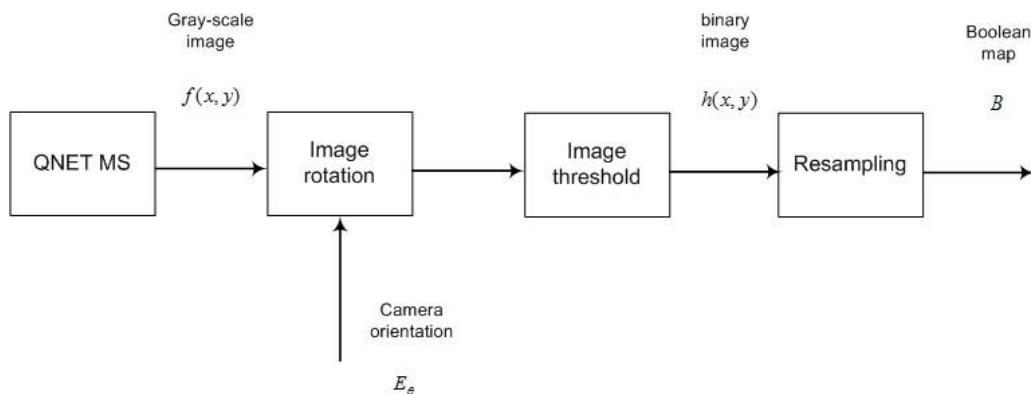


Рисунок 1.3: Цикл обработки изображения

Image rotation – вращение изображения, Image Threshold – пороговая обработка изображения, Resampling – ресэмплирование, Gray-scale image – полутоновое изображение, binary image – двоичное изображение, Boolean map – булевская карта, Camera orientation – ориентация камеры

1.3 Конечный автомат

Конечный автомат аналогичен автомату, представленному в лабораторной работе "Конечный автомат". Он содержит пять состояний, как показано на рисунке 1.4.

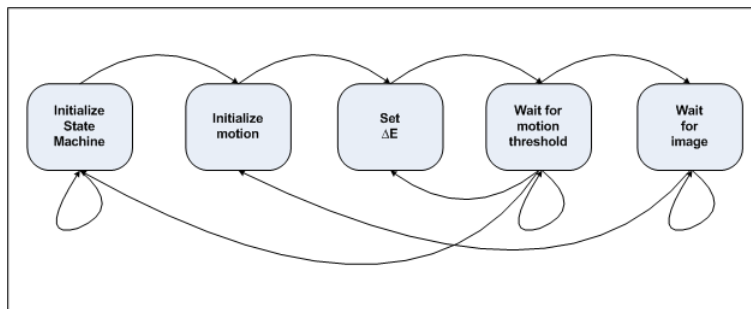


Рисунок 1.4: Конечный автомат

В состоянии Initialize State Machine (Инициализация конечного автомата) система ожидает, чтобы пользователь ввел точки, через которые должна пройти система. Первая точка вводится щелчком по изображению дороги и считается точкой назначения. Манипулятор достигает этой точки последней, потому что она всегда хранится в конце очереди.

В этом состоянии можно также добавлять дополнительные точки на маршруте. Они добавляются в начало очереди в порядке, определяемом последовательностью щелчков (при нажатой клавише CTRL). Так, вторая точка, введенная щелчком, останется на позиции 1, третья - на позиции 2, и т.д. Когда пользователь нажмет GO!, система переходит в следующее состояние.

Состояние Initialize Motion (Инициализация движения) задает список приоритетов направлений, в которых двоичная карта B должна быть проверена на наличие значений TRUE, исходя из текущего положения манипулятора (в цикле управления манипулятором) и необходимой точки из начала очереди.

В состоянии Set Delta E (Задание расстояния) выбирается точка из двоичной карты B (в цикле обработки изображений) со значением TRUE, в порядке приоритета направлений, расстояние между соответствующими пикселями преобразуется в расстояние, в сантиметрах, которое должен преодолеть манипулятор (см. лабораторную работу "Сопоставление с шаблоном", чтобы вспомнить векторную кинематику). Это расстояние передается в цикл управления манипулятором.

В состоянии Wait for Motion Threshold (Ожидание достижения порога передвижения) выполняется проверка, является ли текущее положение манипулятора (в цикле управления манипулятором) достаточно близко к желаемой точке, и если да, переходит в состояние Set Delta E для следующей точки. Когда пройдены все точки, система ожидает в состоянии Wait for Image (Ожидание изображения), пока не станет доступно новое изображение. Это гарантирует, что последнее изображение соответствует полученному в текущем положении. При получении нового изображения система возвращается в состояние Initialize Motion и повторяет цикл, пока не достигнет точки в очереди.

При достижении точки из начала очереди, она удаляется из очереди и процесс повторяется для следующей точки. Когда удалены все точки, система останавливается и завершает выполнение VI. Пример потока показан на рисунке 1.5.

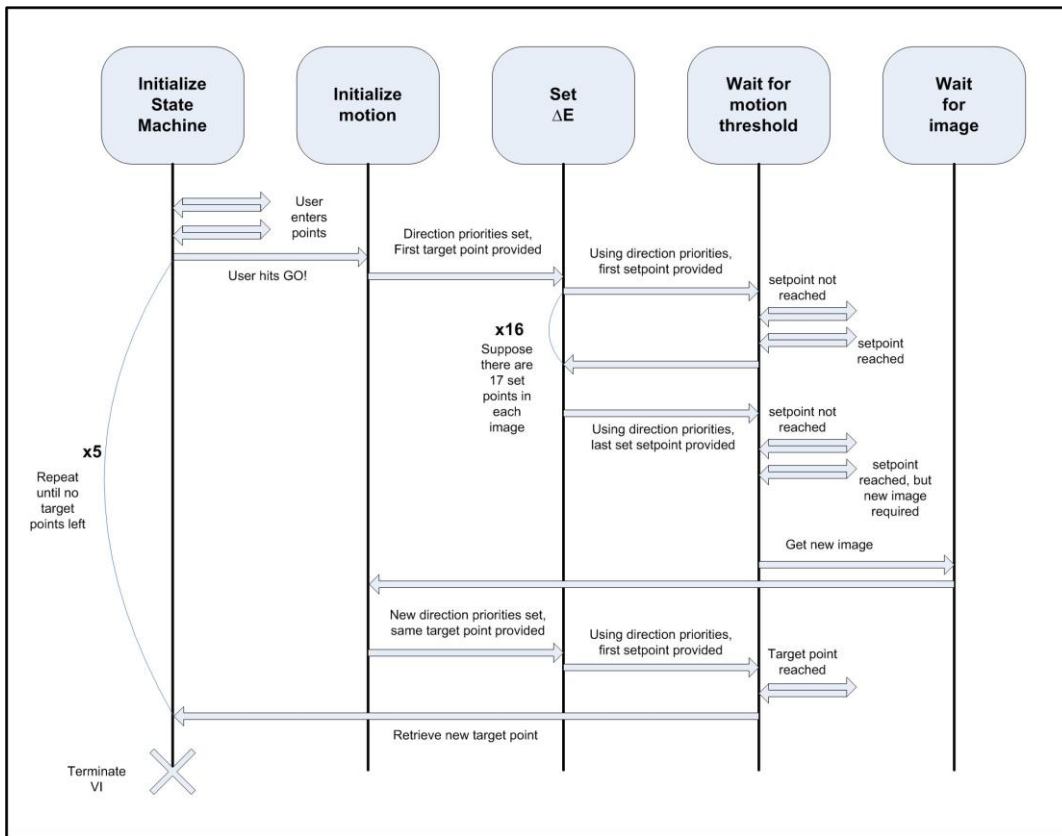


Рисунок 1.5: Пример потока конечного автомата

2 Упражнения в лаборатории

2.1 Соединения в системных циклах

1. Откройте проект `Mechatronic Systems.lvproj`, и из раздела `Quanser ELVIS RIO | Subsystems`, откройте `Goal directed line follower.vi`. Откройте блок-диаграмму и найдите цикл управления манипулятором (Manipulator Control). Подключите узел `Desired task coordinates` ко входу `Task coordinates (cm) Inverse Kinematics subVI`. Подключите выход `Task coordinates (cm) Forward Kinematics subVI` к узлу `Current task coordinates`. Подключите также выход `Camera orientation (deg) Forward Kinematics subVI` к узлу `Camera orientation`.
2. Перейдите к циклу обработки изображений (Image Processing). Подключите узел `Camera orientation (deg) Image Processing subVI`. Подключите также выход `Real-time boolean path map Image Processing subVI` к узлу `Boolean map`.
3. Перейдите к циклу State Machine и найдите состояние `Initialize State Machine`. Подключите узел `Current task coordinates` к узлу `Current location` конечного автомата. Соедините узел `Boolean map` к узлу `Boolean map` конечного автомата. Наконец, подключите узел `Desired location` конечного автомата к узлу `Desired task coordinates`.

2.2 Движение по линии

1. Запустите VI. Когда линейка Calibration заполнится, щелкните по изображению дороги в точке, показанной на рисунке 2.1. Это точка назначения. Щелкните по кнопке GO! Достигла ли система конечной точки назначения? Остановите VI и дайте ответ на основании состояния системы (System State).

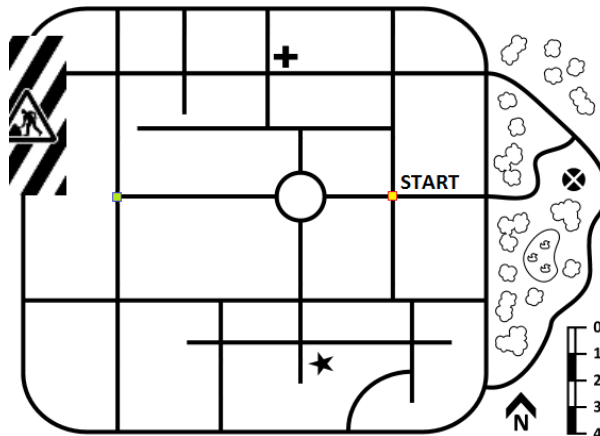


Рисунок 2.1: Точка назначения на изображении дороги на лицевой панели

2. На блок-диаграмме перейдите к состоянию `Waiting for Motion Threshold` конечного автомата. Посмотрите на цикл, который проверяет, близко ли к заданной точке вы находитесь (флаг 2). Обратите внимание, что вход `Threshold subVI` установлен равным 0,01 см. Поэтому конечный автомат ждет, когда манипулятор окажется на расстоянии 0,01 см от заданной позиции. Почему это затруднительно?
3. Квадратная двоичная карта получается ресемплированием каждого второго пикселя из прямоугольного исходного изображения размером 160 на 128 пикселей, а коэффициент преобразования сантиметров в пиксели составляет 0,0275 см/пиксель. Каково минимальное расстояние между текущими и заданными координатами точки? Измените значения входа `Threshold subVI`, заданного на предыдущем шаге, с 0,01 см на 0,2 см и прокомментируйте этот выбор на основании того, что ширина дороги составляет 0,1 см. Запустите VI с той же конечной точкой, что на предыдущем шаге. Улучшился результат?

2.3 Использование дополнительных точек маршрута для навигации

1. Запустите VI. Выберите конечную точку, как показано на рисунке 2.2. Сколькими путями манипулятор может достичь этой точки? Щелкните по кнопке GO! Достиг ли манипулятор пункта назначения? Остановите VI и прокомментируйте результат.

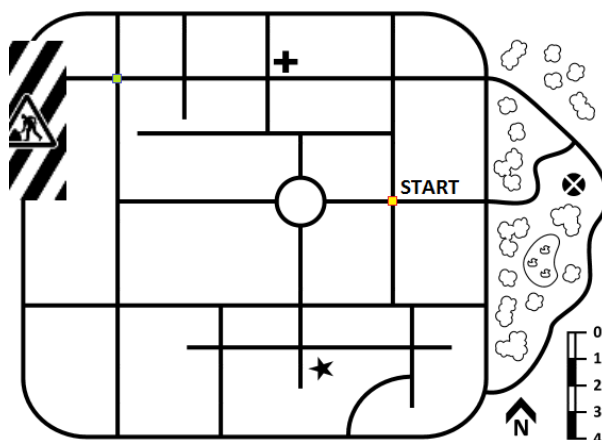


Рисунок 2.2: Точка назначения на изображении дороги на лицевой панели

2. Запустите VI. Выберите конечную точку, точку 1, как показано на рисунке 2.3. Добавьте две маршрутные точки, щелкнув по точке 2, а затем по точке 3, при нажатой клавише CTRL. Обратите внимание, что сначала манипулятор достигнет точки 2, затем точки 3, и наконец точки 1, потому что система движется к конечной точке назначения, следуя по точкам маршрута в том порядке, в каком вы задали. Достиг ли манипулятор конечной точки? Остановите VI и прокомментируйте результат.

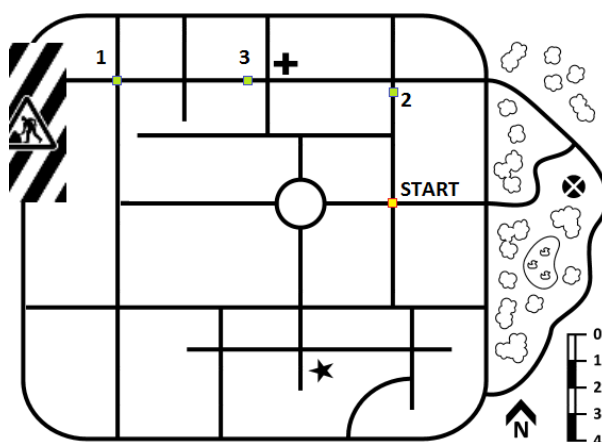


Рисунок 2.3: Точка назначения и точки маршрута на изображении дороги на лицевой панели

Дополнительные маршрутные точки улучшают характеристику системы и всегда должны использоваться во избежание неопределенности пути, по которому необходимо следовать. По умолчанию конечный автомат рассчитывает приоритет направлений на основе расстояния между текущим положением и точкой назначения. Поэтому, если точка назначения дальше на восток, чем на север, манипулятор всегда будет двигаться на восток, даже если путь ведет в тупик, поскольку система не продумывает вперед, а определяет маршрут по необходимости.

2.4 Обход зоны дорожных работ

Вспомните из лабораторной работы "Упреждающая кинематика", что QNET Mechatronic Systems не может достичь желаемых точек в области, показанной на рисунке 2.4. Если целевая точка заставляет манипулятор пройти по этому маршруту, он должен запросить обходной путь к конечной точке назначения. То есть конечный автомат должен стереть все оставшиеся точки маршрута, сохранить конечную точку и затребовать новые точки маршрута у пользователя в состоянии Initialize State Machine. Цель этого упражнения - модифицировать конечный автомат для выполнения этой функциональности.

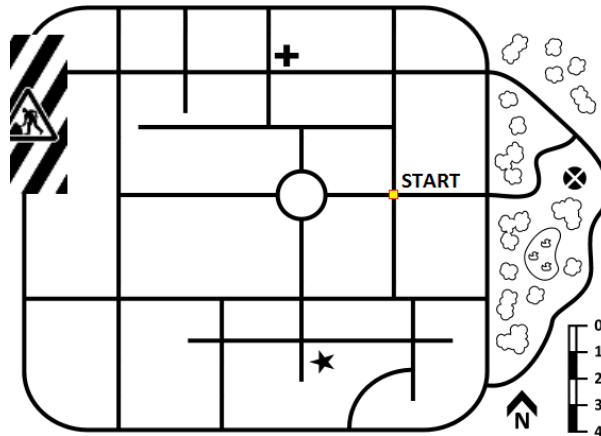


Рисунок 2.4: Зона дорожных работ на изображении дороги QNET Mechatronic Systems

1. На первом шаге необходимо определить, что манипулятор находится в зоне дорожных работ. На блок-диаграмме перейдите к состоянию Initialize Motion конечного автомата. Откройте subVI с символом дорожных работ. Обновите код в соответствии с показанным на рисунке 2.5. Это включает определение, находится ли манипулятор в области размером 2 x 4 см с центром (0,15 см, 5 см) рядом со светофором на северо-западе. Если манипулятор в этой области, то следующим состоянием должно быть New route request (Запрос нового маршрута) вместо Set Delta E, что обрабатывается другим subVI с меткой следующего состояния Quanser. sub-VI с символом дорожных работ также расположен в состоянии Wait for motion threshold конечного автомата, но обновляется автоматически.

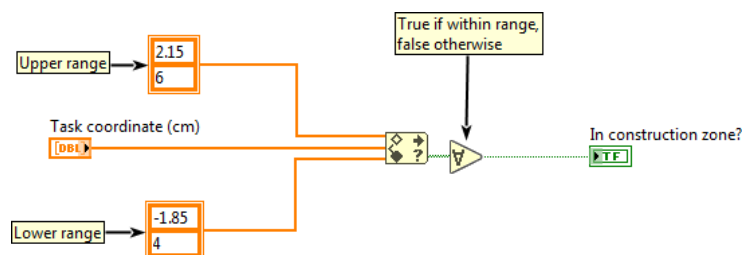


Рисунок 2.5: Определение зоны дорожных работ в состоянии Initialize Motion

2. На следующих шагах выполняется очистка очереди точек, за исключением конечной точки назначения, очистка соответствующих точек с изображения дороги, команда манипулятору на уход из зоны дорожных работ и переход в состояние Initialize State Machine для сброса операций. Эти шаги уже выполнены для вас. Запустите VI и назначьте системе точку назначения и маршрутные точки, как показано на рисунке 2.6. Когда система, после выполнения подпрограммы обработки зоны дорожных работ, запросит новые точки, назначьте точки, показанные на рисунке 2.7. Обратите внимание, что точка 1 уже существует, поскольку является конечной точкой. Соответствует ли передвижение по линии требуемой функциональности?

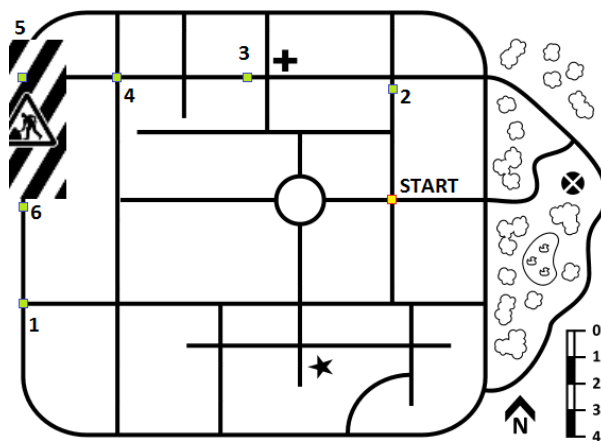


Рисунок 2.6: Маршрутные точки в зоне дорожных работ

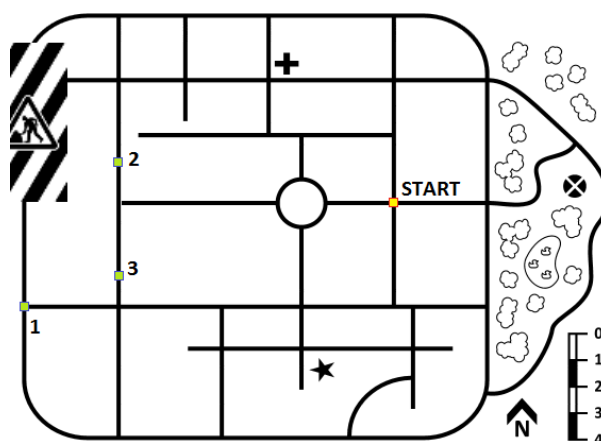


Рисунок 2.7: Маршрутные точки вокруг зоны дорожных работ

2.5 Дальнейшие размышления

1. В этой лаборатории зона дорожных работ определялась вручную с помощью кинематики. Можно ли использовать для этого сопоставление с шаблоном? Есть ли у этого метода свои недостатки?
2. Перечислите несколько способов улучшения алгоритма следования по линии.
3. Назовите два сходства между этим экспериментом и реальной ситуацией, например, управлением беспилотным автомобилем? Перечислите несколько допущений в этом эксперименте, которые не обязательно будут выполняться для беспилотного автомобиля

© 2016 Quanser Inc., Все права защищены.

Quanser Inc.
119 Spy Court
Markham, Ontario
L3R 5H6
Canada
info@quanser.com
Телефон: 1-905-940-3575
Факс: 1-905-940-3576

Отпечатано в Маркхем, Онтарио.

Для получения дополнительной информации о продукции, предлагаемой Quanser Inc., посетите, пожалуйста, веб-сайт:
<http://www.quanser.com>

Этот документ и программное обеспечение, описанное в нем, предоставляются в соответствии с лицензионным соглашением. Ни программное обеспечение, ни этот документ не могут использоваться или копироваться способом, отличным от указанных в соответствии с условиями этого лицензионного соглашения. Quanser Inc. предоставляет следующие права: а) право воспроизводить работу, включать работу в один или несколько наборов и воспроизводить работу, включенную в наборы, б) создавать и воспроизводить усовершенствования при условии принятия разумных мер четко определить изменения, внесенные в оригинальную работу, с) распространять и публиковать работу, в том числе включенную в наборы и д) распространять и открыто выполнять усовершенствования. Вышеупомянутые права могут быть реализованы на всех носителях и в форматах, которые теперь известны или будут разработаны в будущем. Эти права предоставляются и ограничены следующим : а) вы не можете использовать какие-либо права, предоставленные вам в вышеуказанном виде, любым способом, который в первую очередь предназначен или ориентирован для коммерческой выгоды или частной денежной компенсации и б) вы должны сохранять в целостности все уведомления об авторских правах для Работы и ссылаться на Quanser Inc. Эти ограничения не могут быть изменены без предварительного письменного разрешения Quanser Inc.

Плата QNET MECHATRONIC SYSTEMS -