

Постоянное наблюдение за температурой при помощи беспроводной сети датчиков. Часть 2

Введение

В [первой части](#) было показано как считывать значения напряжений из сети модулей XBee, а также построение и отладка беспроводной сети датчиков температуры. В этом посте будет дано детальное рассмотрение сети датчиков, метода сбора измерений датчиков по квартире за длительный период времени, а также приемы построения графиков.

Сеть датчиков температуры

Квартира покрыта сетью датчиков с 4мя узлами сети. Каждый узел включает в себя от 3 до 4 датчиков, которые расположены внутри комнаты, на улице (телефонный кабель пропущен по оконной раме) или на батарее (рядом с трубами). Датчики на батарее позволяют определить насколько быстро нагревается батарея при поступлении тепла и будут служить опорными точками для определения включено или выключено отопление.



Рисунок 1. Датчик температуры внутри гостиной (прикреплен к проводу от лампы, обычно скрыт фоторамкой)



Рисунок 2. Датчик температуры за окном



Рисунок 3. Датчик температуры на батарее

XBee может получать значения напряжения на выводах 17-20, ниже приведено распределение узлов по комнатам. Как было указано в части 1, в некоторых случаях использовались сдвоенные датчики температуры. Таким образом, периодически будет проверяться точность калибровки датчиков.

Комната	XBee	Адрес XBee	Вывод 17	Вывод 18	Вывод 19	Вывод 20
Гостиная	XBee 1	0013 A200 40B1 8D3E	Не используется	Внутри	Снаружи	Батарея
Кабинет	XBee 2	0013 A200 40B1 8EFB	Не используется	Батарея	Внутри (сдвоенный)	
Кухня	XBee 3	0013 A200 40B1 916C	Внутри (сдвоенный)		Внутри (сдвоенный)	
Спальня	XBee 4	0013 A200 40B2 2000	Батарея	Снаружи	Внутри (сдвоенный)	

Так как данные будут использоваться в различных вычислениях, была создана функция [XBeeSensors](#), возвращающая 1 переменную для комнаты, адресов и расположения отдельных датчиков температуры.

Сбор данных

После того, как датчики были распределены по квартире, требовалось организовать чтение их показаний в файл, который бы анализировался позже.

Сначала, требуется открыть соединение между MATLAB и устройством-координатором сети XBee:

```
xb = xbee('COM3');
```

Затем, с каждого вывода запрашивается значение напряжения и ответ записывается. Так как XBee отвечают в любой последовательности, требуется запрашивать дополнительные данные - адрес устройства, и сравнивать полученный адрес со списком узлов сети XBee используя функцию `ismember()`. По умолчанию, при нахождении отсутствующих значений, MATLAB подставит в выходную матрицу 0, что может привести к ошибке определения корректных данных. Поэтому выходная матрица `volts` инициализируется как матрица из элементов NaN, состоящая из одной строки для каждого XBee, и одного столбца для каждого датчика. Матрица создается при помощи функции `NaN`. Так же выходная матрица всегда будет фиксированного размера, независимо от результата опроса XBee

```
XBeeAddresses = XBeeSensors; % Получение адресов датчиков
numxbees = length(XBeeAddresses); % Получение количества модулей XBee
volts = NaN(numxbees,4); % Инициализация выходной матрицы
ts = now; % Получение текущего времени
for p = 1:4
    [v,src] = readVoltage(xb,p+16); % Чтение значений напряжений
    [tf,loc] = ismember(src,XBeeAddresses); % Сравнение со списком XBee
    loc = loc(tf); % Отбрасываем данные с несуществующих в сети модулей
    volts(loc,p) = v(tf); % Сортировка и запись полученных данных.
end
```

Затем значения напряжений были записаны в файл для хранения и последующего анализа. Так как требовалось записать как численные, так и строковые данные, для записи использовалась функция `fprintf`. Эта функция принимает в качестве аргументов формат-строку и список значений для заполнения формат-строки. В данном случае, формат-строка имеет вид:

```
formatstr = '%0.5f\t%s\t%7.5f\t%7.5f\t%7.5f\n';
```

`%0.5f` означает что первое значение – с плавающей точкой, 5 знаков после запятой, `\t` - табуляцию, `%s` - строку, и `%7.5f` означает число с плавающей точкой из 7 знаков, 5 из которых после запятой. Эта формат-строка сгенерирует 6 столбцов в файле с данными: время, адрес, вывод 17, вывод 18, вывод 19, вывод 20

```
fid = fopen('templog.txt','a'); % Добавить данные в файл, если он существует
for s = 1:numxbees
    fprintf(fid,formatstr,...
        ts,XBeeAddresses{s},volts(s,1),volts(s,2),volts(s,3),volts(s,4));
end
fclose(fid);
```

Пример вывода в файл:

```
735719.92500 0013A20040B18D3E 0.60880 0.73666 0.74018 0.74018
735719.92500 0013A20040B1916C 0.74370 0.75425 0.75777 0.74604
735719.92500 0013A20040B22000 0.74370 0.73783 0.74252 0.74135
735719.92500 0013A20040B18EFB 0.60528 0.73783 0.73783 0.74370
```

Периодический опрос датчиков

Для организации опроса с периодом в 2 минуты, команды приведенные выше были организованы в функцию [XBeePoll](#), вызываемую таймером в MATLAB `timer` каждые 2 минуты. В `XBeePoll` был добавлен дополнительный код для проверки на ошибки, чтобы повысить робастность кода путем предотвращения ошибок при прерывании сбора данных и записи значения выходного напряжения батареек.

```
t = timer;
t.Userdata = xb; % Store the XBee object for access by XBeePoll
t.TimerFcn = @XBeePoll; % Set XBeePoll as the timer function
t.Period = 120; % Call the timer function every 120 seconds
t.ExecutionMode = 'FixedRate'; % Run the timer at a fixed rate
```

Затем таймер был запущен:

```
start(t)
```

Команды выше были объединены в скрипт [XBeeStartPoll](#) для повышения переиспользуемости. Когда сбор данных завершается требуется остановить таймер:

```
stop(t)
```

Чтение файлов

Код был запущен в течении часа, чтобы удостоверится в его работоспособности. Данные были сохранены в файл templog.txt (доступен на [File Exchange](#)). Затем, с помощью функции [readtable](#) данные были загружены в [таблицу MATLAB](#). Таблицы – это новый тип данных MATLAB, появившийся в релизе R2013b. Для создания таблицы из файла, необходимо указать, что данные разделены табуляциями ('\t'), и имена переменных не содержатся в файлах.

```
data = readtable('templog.txt','Delimiter','\t','ReadVariableNames',false);  
data(1:8,:)
```

```
ans =
```

```
Var1 Var2 Var3 Var4 Var5 Var6
```

```
7.3572e+05 '0013A20040B18D3E' 0.6088 0.73666 0.74018 0.74018  
7.3572e+05 '0013A20040B1916C' 0.7437 0.75425 0.75777 0.74604  
7.3572e+05 '0013A20040B22000' 0.7437 0.73783 0.74252 0.74135  
7.3572e+05 '0013A20040B18EFB' 0.60528 0.73783 0.73783 0.7437  
7.3572e+05 '0013A20040B18D3E' 0.6088 0.73666 0.74018 0.739  
7.3572e+05 '0013A20040B1916C' 0.7437 0.75425 0.75777 0.74604  
7.3572e+05 '0013A20040B22000' 0.7437 0.73783 0.74252 0.74252  
7.3572e+05 '0013A20040B18EFB' 0.60528 0.73783 0.739 0.7437
```

Этот код создает таблицу MATLAB из 6 переменных (одна переменная – один столбец таблицы), но так как имена переменных не хранятся в файле, требовалось указать их вручную:

```
data.Properties.VariableNames = {'Time','XBee','Pin17','Pin18','Pin19','Pin20'};
```

```
data(1:8,:)
```

```
ans =
```

```
Time XBee Pin17 Pin18 Pin19 Pin20
```

```
7.3572e+05 '0013A20040B18D3E' 0.6088 0.73666 0.74018 0.74018  
7.3572e+05 '0013A20040B1916C' 0.7437 0.75425 0.75777 0.74604  
7.3572e+05 '0013A20040B22000' 0.7437 0.73783 0.74252 0.74135  
7.3572e+05 '0013A20040B18EFB' 0.60528 0.73783 0.73783 0.7437  
7.3572e+05 '0013A20040B18D3E' 0.6088 0.73666 0.74018 0.739  
7.3572e+05 '0013A20040B1916C' 0.7437 0.75425 0.75777 0.74604  
7.3572e+05 '0013A20040B22000' 0.7437 0.73783 0.74252 0.74252  
7.3572e+05 '0013A20040B18EFB' 0.60528 0.73783 0.739 0.7437
```

Теперь следует преобразовать 64х-битный адрес XBee в более понятное имя:

```
data.XBee = strrep(data.XBee, '0013A20040B18D3E', 'XBee1');
```

```
data.XBee = strrep(data.XBee, '0013A20040B18EFB', 'XBee2');
```

```
data.XBee = strrep(data.XBee, '0013A20040B1916C', 'XBee3');
```

```
data.XBee = strrep(data.XBee, '0013A20040B22000', 'XBee4');
```

```
data(1:8,:)
```

```
ans =
```

```
Time XBee Pin17 Pin18 Pin19 Pin20
```

```
7.3572e+05 'XBee1' 0.6088 0.73666 0.74018 0.74018  
7.3572e+05 'XBee3' 0.7437 0.75425 0.75777 0.74604  
7.3572e+05 'XBee4' 0.7437 0.73783 0.74252 0.74135  
7.3572e+05 'XBee2' 0.60528 0.73783 0.73783 0.7437  
7.3572e+05 'XBee1' 0.6088 0.73666 0.74018 0.739  
7.3572e+05 'XBee3' 0.7437 0.75425 0.75777 0.74604  
7.3572e+05 'XBee4' 0.7437 0.73783 0.74252 0.74252  
7.3572e+05 'XBee2' 0.60528 0.73783 0.739 0.7437
```

To make sure that the read operation worked as I expected, I plotted the raw voltage data from Pin 20 on a single XBee:

```

firstXBee = strcmp(data.XBee,'XBee1');
plot(data.Time(firstXBee),data.Pin20(firstXBee),'-')
datetick
xlabel('Time')
ylabel('Volts')
title('Raw voltage from Pin 20 on one XBee')

```

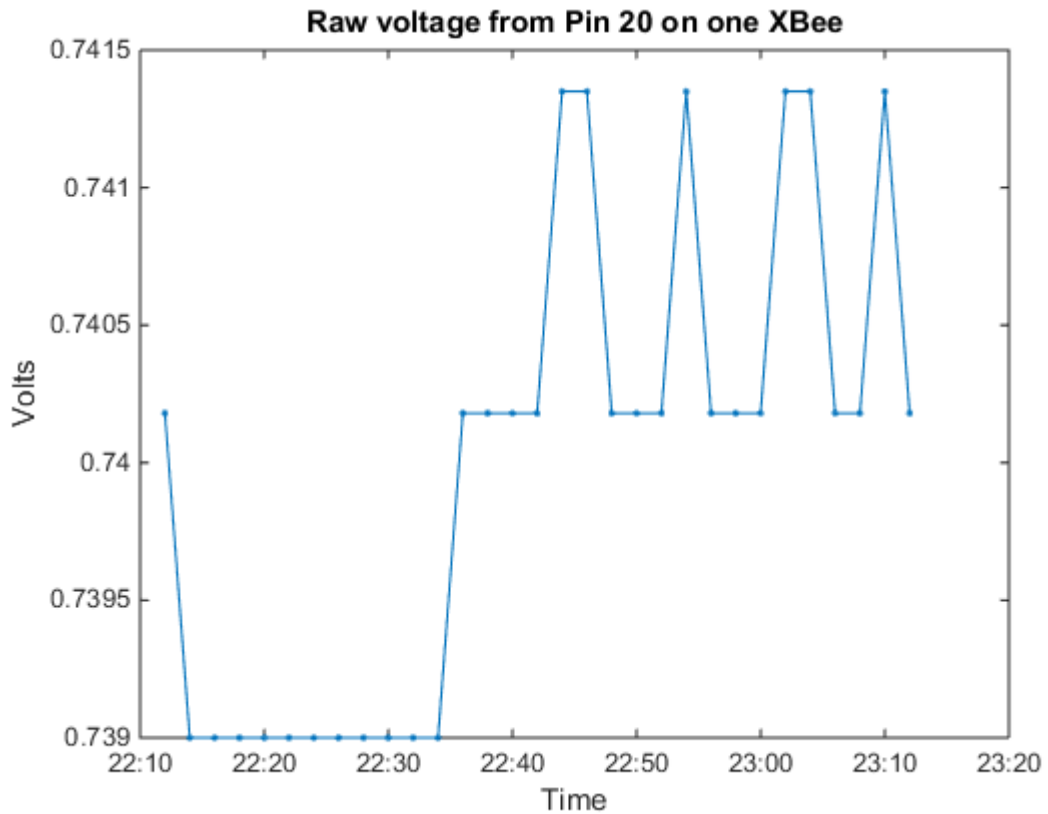


Рисунок 4. Необработанные данные за 1 час с одного XBee

Очистка и реорганизация данных

И хотя старт был успешен, требовалось упростить работу с данными. Для этого данные были очищены и реорганизованы. Исходные данные из файла содержали несколько записей для каждой временной отметки (одна для каждого XBee), и данные с 4х датчиков температуры, для примера, на выводе 17 перемешаны в одном столбце. С данными было бы работать намного проще, если бы они были только один ряд для каждой уникальной временной отметке, а данные каждого датчика были бы в уникальном столбце.

Например, до реорганизации данных таблица выглядела так:

Время	Xbee	Вывод 17	Вывод 18	Вывод 19	Вывод 20
10:12	Xbee 10	0.6088	0.7367	0.7402	0.7402
10:12	Xbee 30	0.7437	0.7543	0.7578	0.7460
10:12	Xbee 40	0.7437	0.7378	0.7425	0.7414
10:12	Xbee 20	0.6053	0.7378	0.7378	0.7437

10:14	Xbee 1	0.6088	0.7367	0.7402	0.7390
10:14	Xbee 3	0.7437	0.7543	0.7578	0.7460
10:14	Xbee 4	0.7437	0.7378	0.7425	0.7425
10:14	Xbee 2	0.6053	0.7378	0.7390	0.7437

После реорганизации, данные должны выглядеть так:

Время	Вывод 17				Вывод 18				Вывод 19				Вывод 20			
	XBee 1	XBee 2	XBee 3	XBee 4	XBee 1	XBee 2	XBee 3	XBee 4	XBee 1	XBee 2	XBee 3	XBee 4	XBee 1	XBee 2	XBee 3	XBee 4
10:12:8	0.6083	0.6057	0.7437	0.7437	0.7367	0.7378	0.7542	0.7378	0.7402	0.7378	0.7578	0.7425	0.7402	0.7437	0.7460	0.7413
10:14:8	0.6083	0.6057	0.7437	0.7437	0.7367	0.7378	0.7542	0.7378	0.7402	0.7390	0.7578	0.7425	0.7390	0.7437	0.7460	0.7413

Для этого сначала требуется преобразовать временные метки. Частота выборки – 2 минуты, поэтому требуется округлить значения меток:

```
precision = 60; % секунды
precision = precision/(60*60*24); % конвертация из секунд в дни
data.Time = round(data.Time/precision)*precision;
```

Затем требуется записать все показания датчиков за одну временную метку в одну строку таблицы. Так как используется тип данных `table`, можно воспользоваться встроенной функцией `unstack`, которая появилась как и таблицы в MATLAB R2013b и предназначена именно для этой задачи. При вызове `unstack`, указывается столбец-индикатор (в этом случае - `XBee`) для обозначения источника каждого измерения, и какие переменные будут разобраны (в данном случае, `Pin17`, `Pin18`, `Pin19`, и `Pin20`). Оставшаяся переменная (`Time`) используется для обозначения уникальных измерений. В получившейся таблице будет по строке на каждую временную метку, и один столбец для каждой комбинации вывода и уникального `XBee`. Используется функция `mean`, чтобы в ситуации, когда существует больше одного измерения напряжения на выводе `XBee`, эти измерения усреднились, а не суммировались (поведение по умолчанию для `unstack`)

```
data = unstack(data, {'Pin17','Pin18','Pin19','Pin20'}, 'XBee','AggregationFunction',@mean);
data(1:2,:)
ans =
Time Pin17_XBee1 Pin17_XBee2 Pin17_XBee3 Pin17_XBee4 Pin18_XBee1 Pin18_XBee2 ...
```

```
7.3572e+05 0.6088 0.60528 0.7437 0.7437 0.73666 0.73783 ...
7.3572e+05 0.6088 0.60528 0.7437 0.7437 0.73666 0.73783 ...
```

После разбора данных можно извлекать показания напряжения и время. В таблицах MATLAB доступ к данным осуществляется 3мя способами.

1. Доступ ко всему столбцу, используя имя столбца
2. Доступ к части таблицы, используя круглые скобки.
3. Извлечение данных, используя фигурные скобки.

Последние 2 способа точно такие же как и для массивов ячеек в MATLAB. При использовании круглых скобок для доступа к данным в типе `table`, результат будет иметь тип `table` и содержать извлечённые данные.

Однако, так как требуется матрица из значений типа `double`, необходимо использовать фигурные скобки, чтобы извлечь показания напряжений из типа данных `table` и возврата матрицы из типа `double`.

```
ts = data.Time;
volts = data(:,2:end);
```

Теперь можно преобразовать напряжение в градусы Фаренгейта

```
tempC = (volts*100-50);
tempF = tempC*9/5+32;
```

и удалить данные для выводов, не подключенных к датчикам.

```
tempF = tempF(:,3:end);
```

Для воспроизводимости, была создана функция [XBeeReadLog](#) а так же добавлены дополнительные проверки для повышения надежности кода а также сортировки вывода в алфавитном порядке на основе положения каждого датчика

Посторенние графиков

После получение матрицы температур (tempF) и временных меток (ts), можно легко построить график:

```
plot(ts,tempF)
xlabel('Time')
ylabel('Temperature (\textcircled{F})')
title('One Hour of Test Temperature Data')
```

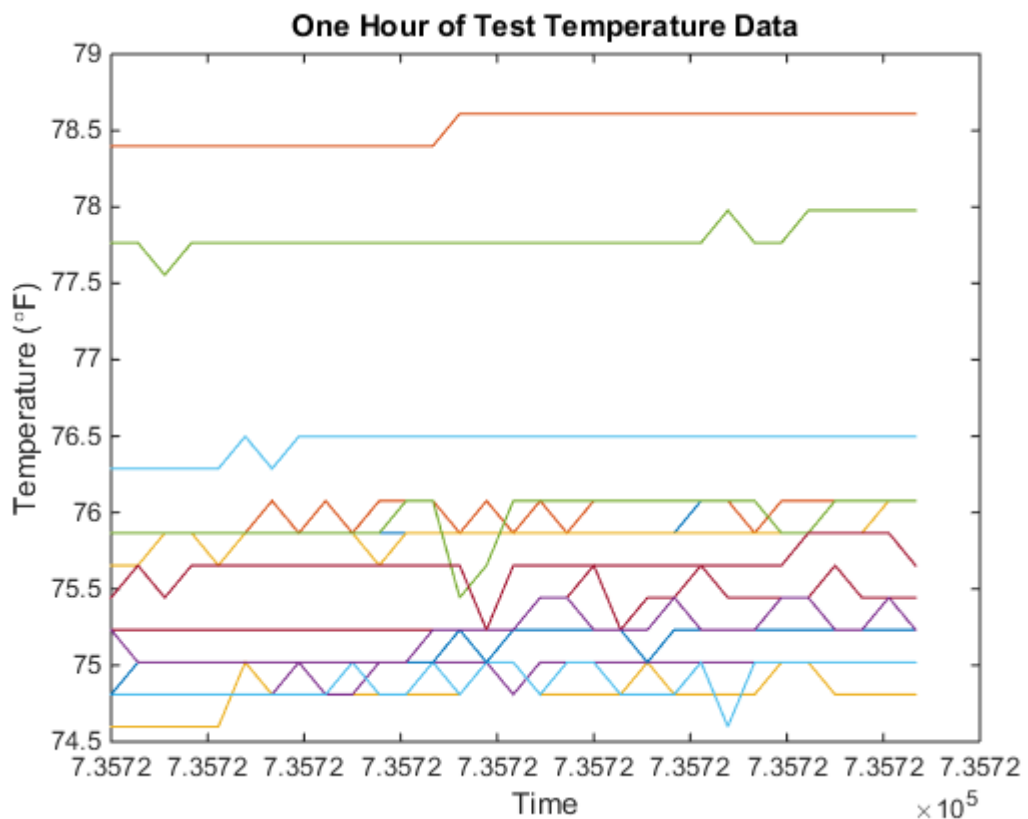


Рисунок 5. Тестовые данные за 1 час

После добавления дополнительного функционала, была создана функция [plotTemps](#). Скомбинировав XBeeReadLog и plotTemps, теперь можно легко читать и визуализировать данные.

```
[tempF, ts, location, lineSpecs] = XBeeReadLog('templog.txt',60);
plotTemps(ts,tempF,location,lineSpecs)
```

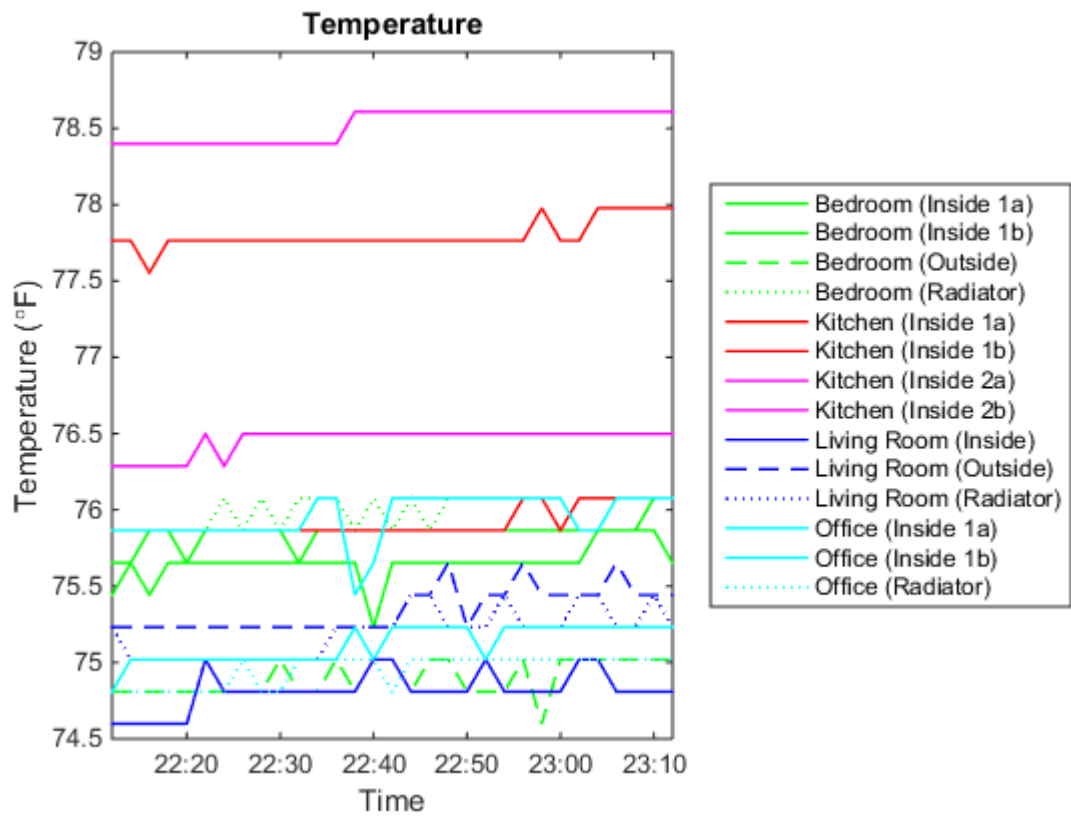



Рисунок 6. Результат работы функции plottemps

Следующие шаги

Начало положено. Однако, в большинстве случаев, два спаренных датчика показывали различающиеся значения температуры, поэтому требовалось провести калибровку. Она будет рассмотрена в следующем посте

[Скачать код](#)

Публикации по данной теме:

[Постоянное наблюдение за температурой при помощи беспроводной сети датчиков. Часть 1](#)

[Постоянное наблюдение за температурой при помощи беспроводной сети датчиков. Часть 3](#)

[Постоянное наблюдение за температурой при помощи беспроводной сети датчиков. Часть 4](#)



Контакты

exponenta.ru

E-mail: **info@exponenta.ru**

Тел.: +7 (495) 009 65 85

Адрес: **115088 г. Москва,**

2-й Южнопортовый проезд, д. 31, стр. 4

mathworks.com



© 2012 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.