

# Software Defined Radio in MATLAB®

**Mike McLernon**

**MathWorks, Inc.**

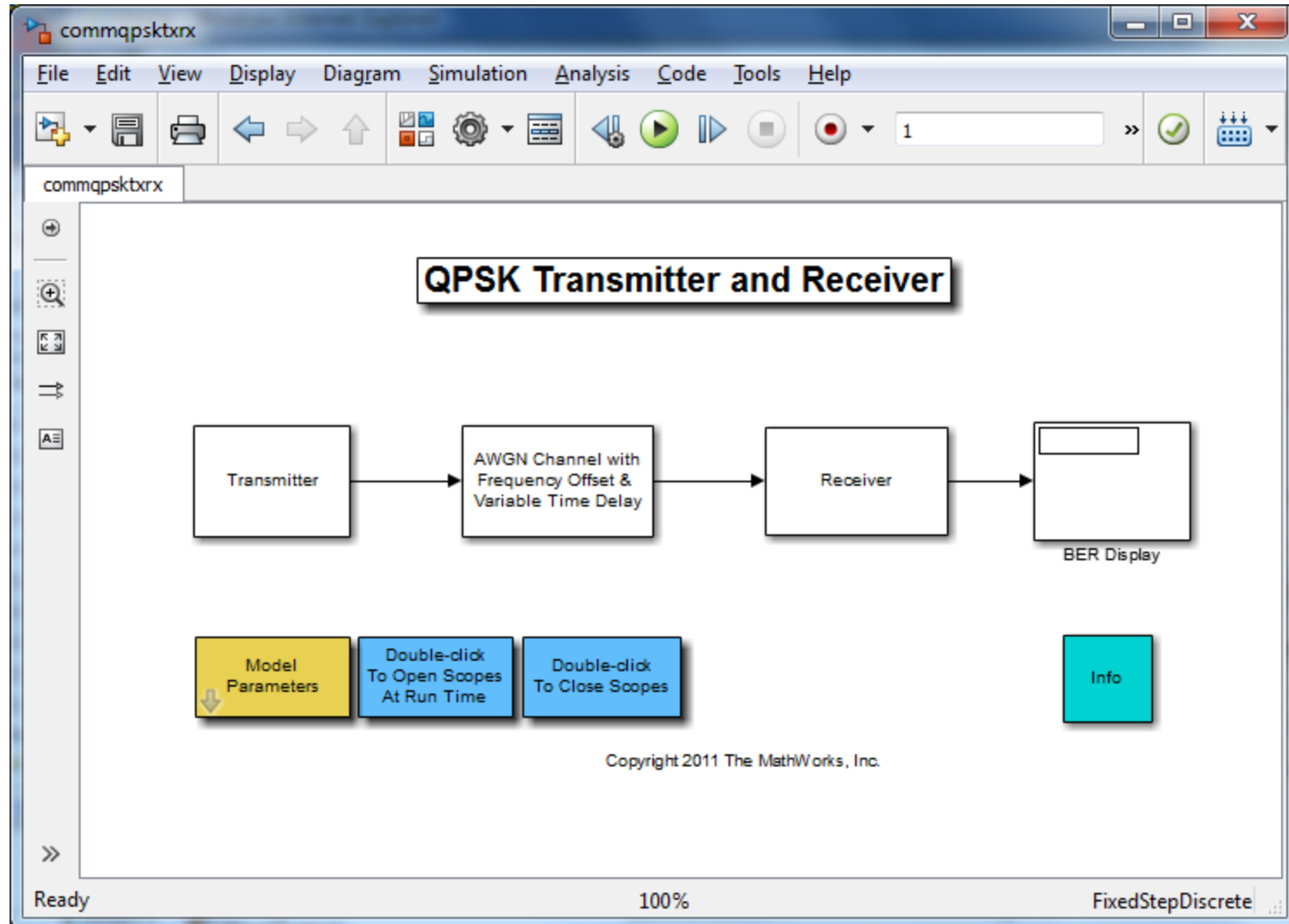
**[Mike.McLernon@mathworks.com](mailto:Mike.McLernon@mathworks.com)**

**5/17/2013**

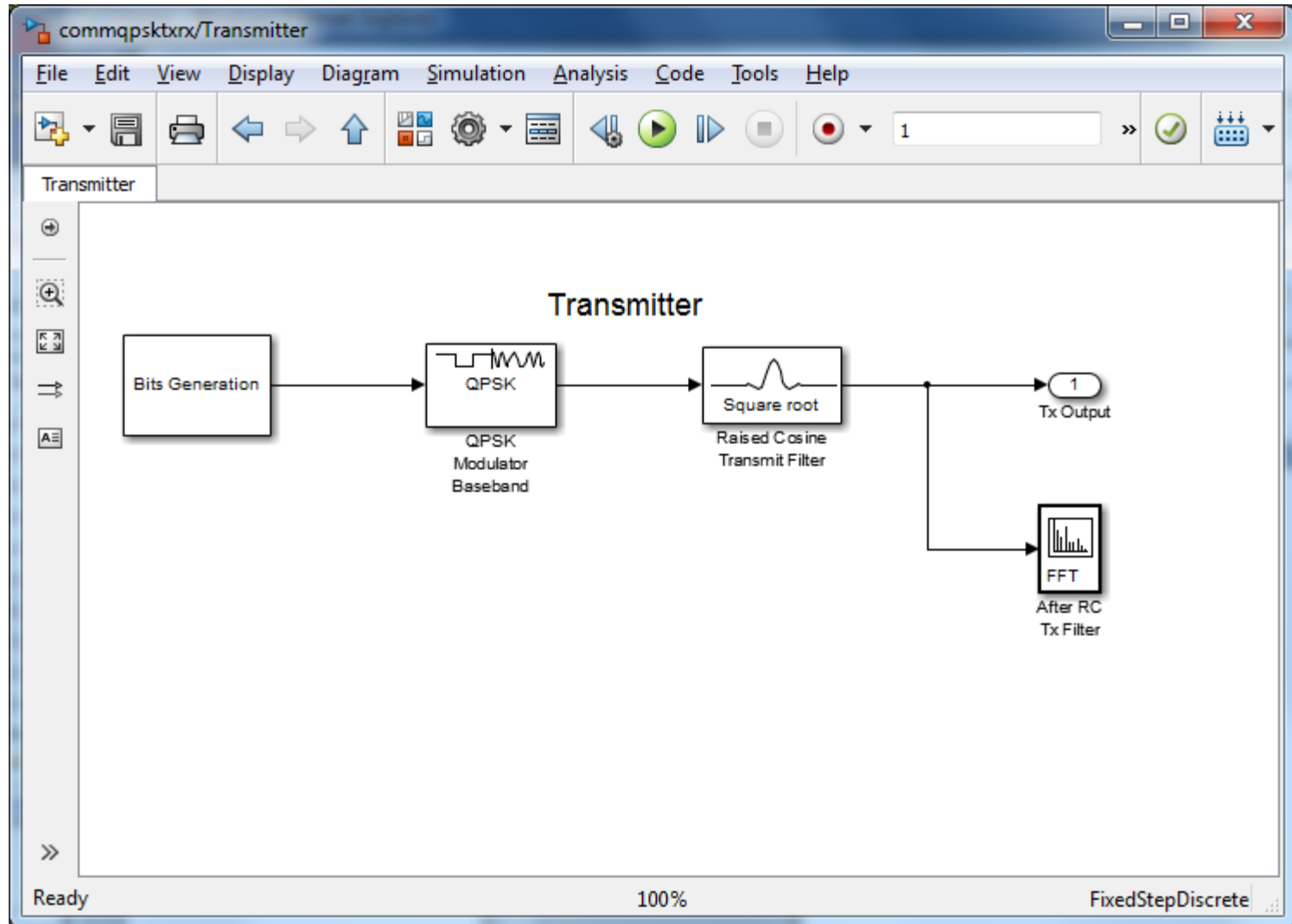
# Agenda

- Simulink® QPSK Transceiver
- MATLAB System Objects
- MATLAB Transceiver

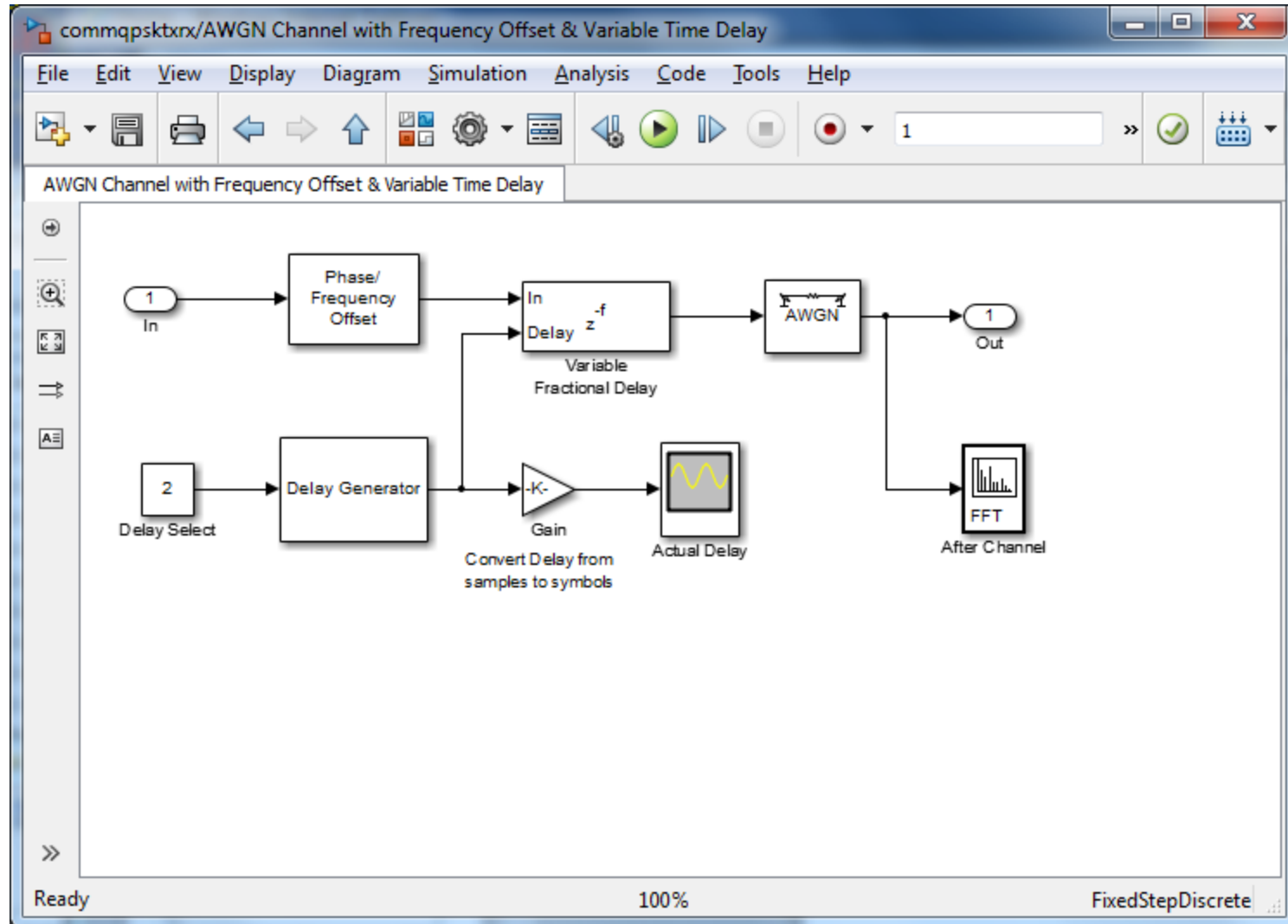
# Simulink QPSK Transceiver



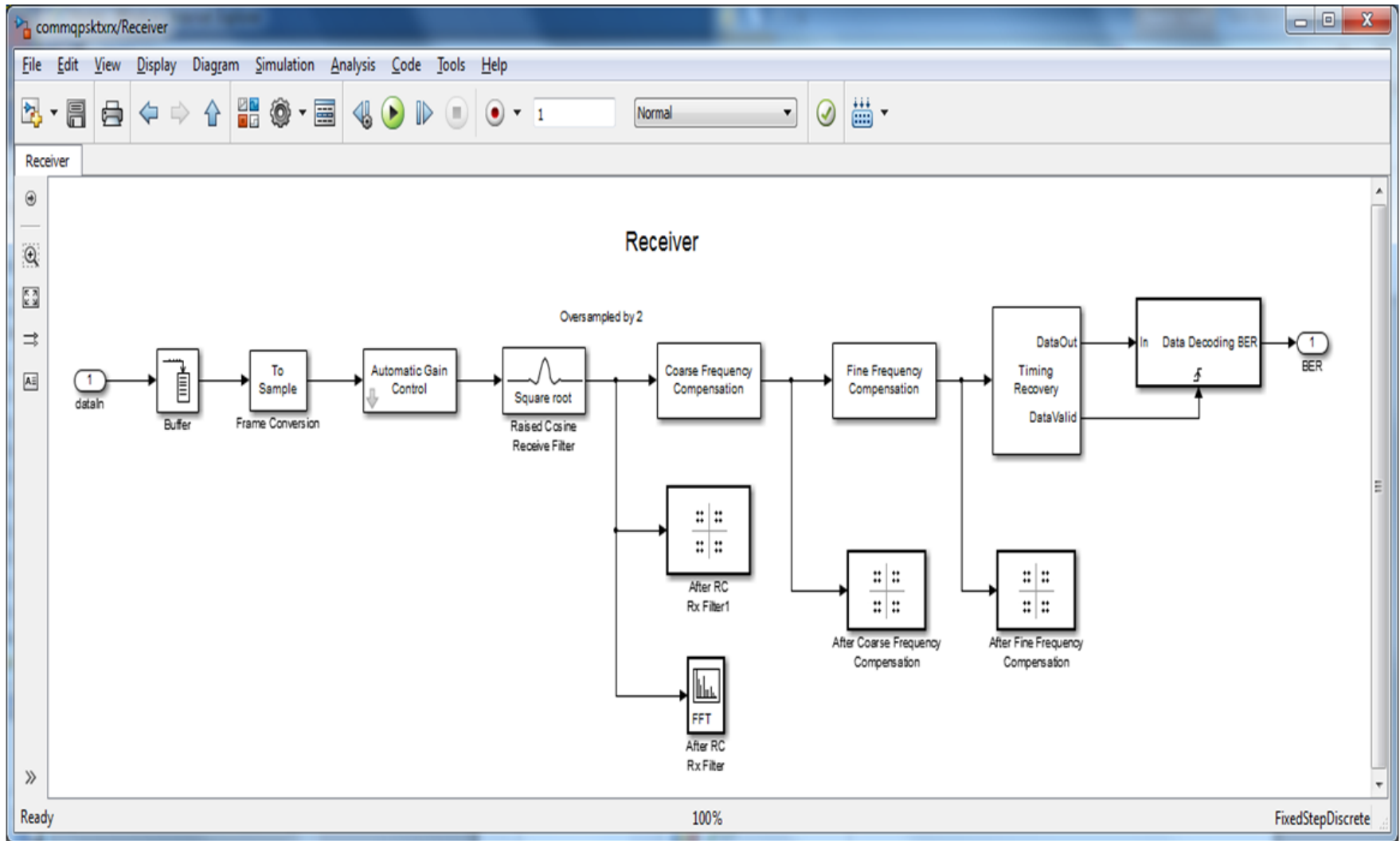
# Transmitter



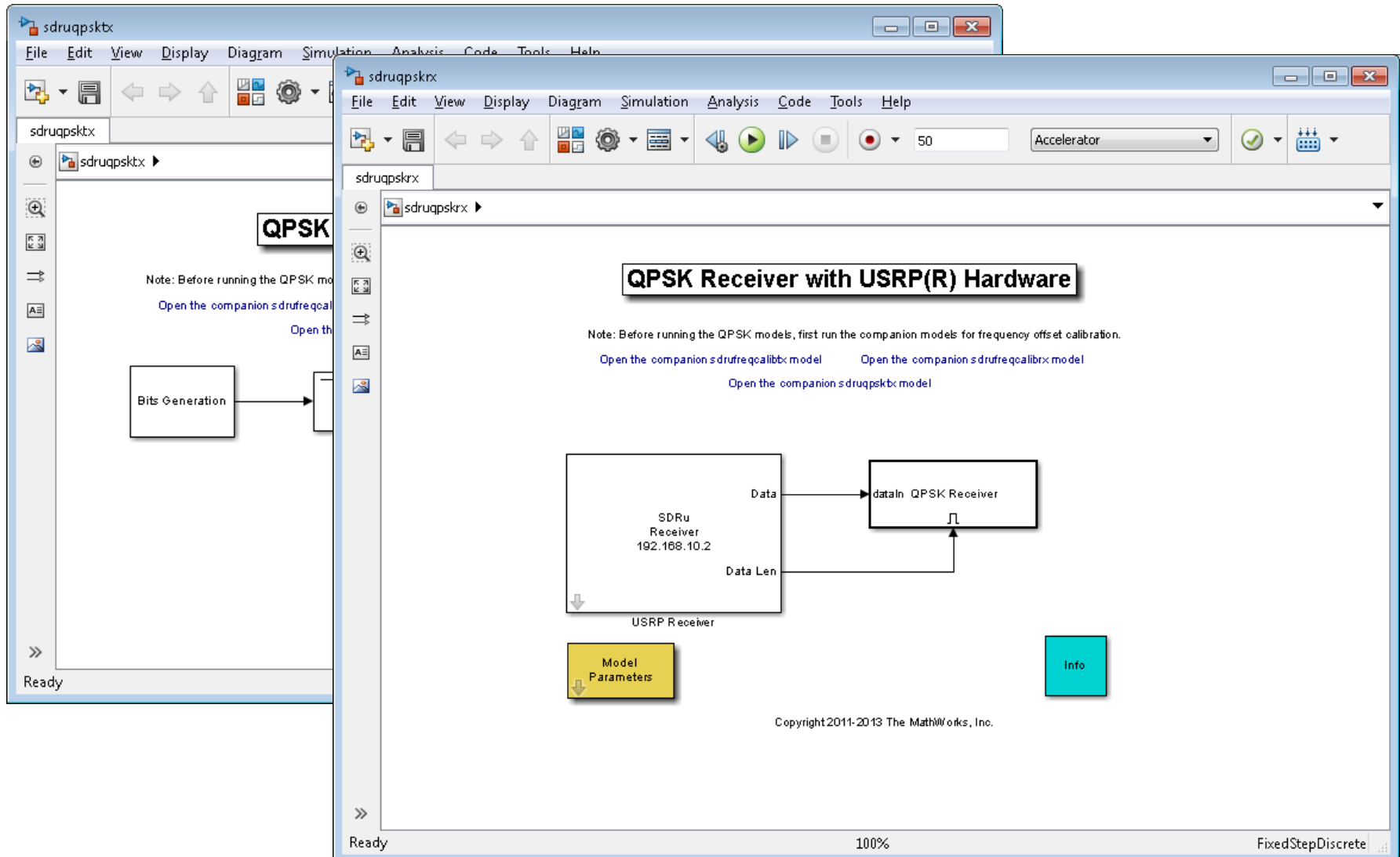
# Channel



# Receiver



# QPSK Transmitter and Receiver with USRP® Radios



The image displays two screenshots of the MATLAB/Simulink software interface, showing the configuration of a QPSK transmitter and receiver using USRP hardware.

**Left Screenshot (sdrupsktx):** Shows the transmitter model. A block labeled "Bits Generation" is connected to the input of the transmitter. A note reads: "Note: Before running the QPSK model, first run the companion sdrufreqcal model." Below the note are two links: "Open the companion sdrufreqcal model" and "Open the companion sdrupsktx model". The status bar at the bottom indicates "Ready".

**Right Screenshot (sdrupskrx):** Shows the receiver model. A central block is labeled "QPSK Receiver with USRP(R) Hardware". A note reads: "Note: Before running the QPSK models, first run the companion models for frequency offset calibration." Below the note are three links: "Open the companion sdrufreqcalibx model", "Open the companion sdrufreqcalibx model", and "Open the companion sdrupsktx model". The main diagram shows an "SDRu Receiver 192.168.10.2" block connected to a "dataIn QPSK Receiver" block. The "SDRu Receiver" block has two outputs: "Data" and "Data Len", both of which are connected to the "dataIn QPSK Receiver" block. Below the "SDRu Receiver" block is a "Model Parameters" block, and to the right is an "Info" block. The status bar at the bottom indicates "Ready", "100%", and "FixedStepDiscrete".

# Agenda

- Simulink QPSK Transceiver
- MATLAB System Objects
- MATLAB Transceiver



# MATLAB System Objects™

- Support for streaming data
- Automatic state handling (for the user, not the author)
- Consistent API for setup and execution
- Initialization phase is separated from the execution phase
- Code generation for desktop acceleration
- Build hierarchical systems in MATLAB

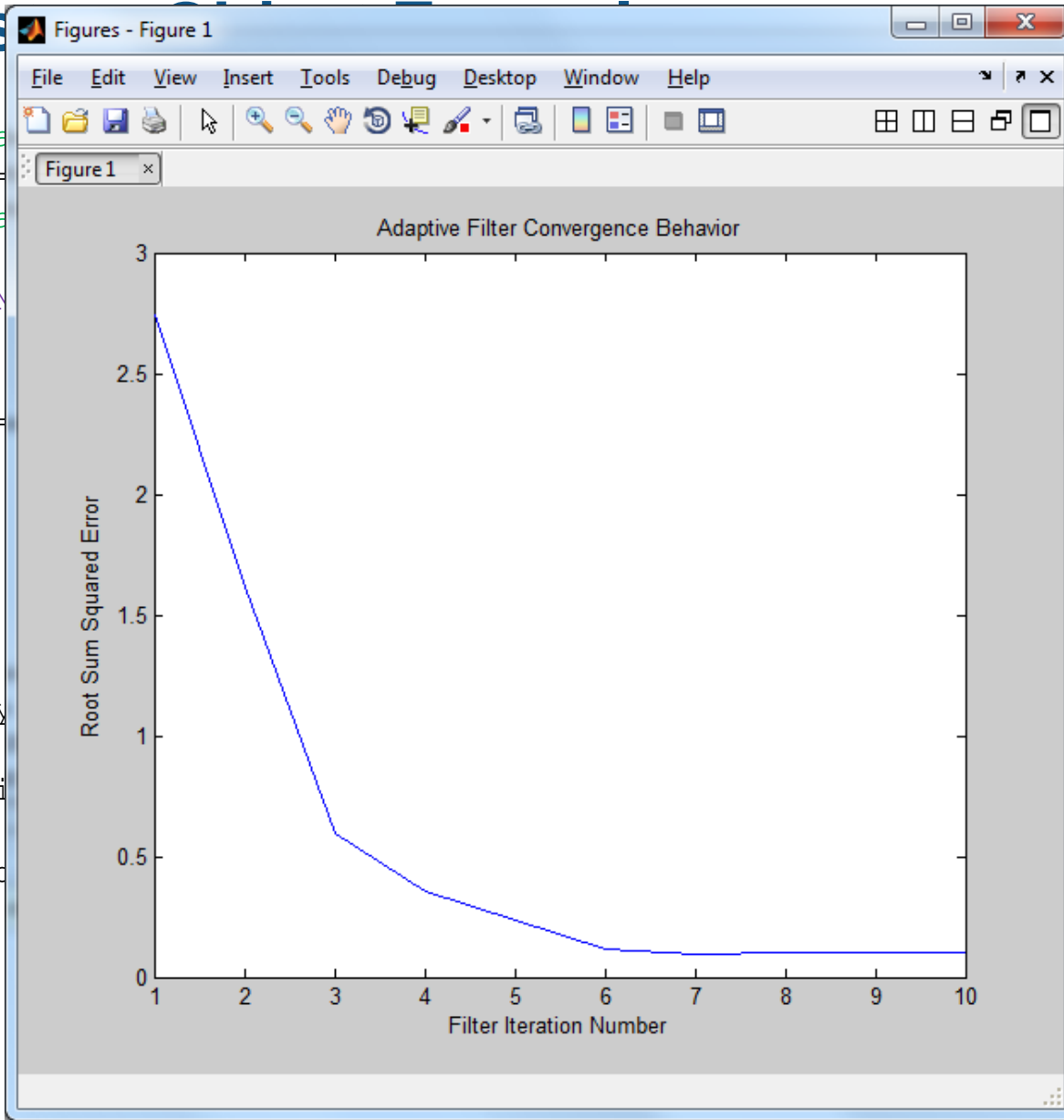
# A Sys

```

% Create
hlms =
% Create
hfilt
'N

% Run
diff =
for i
%
x
%
d
%
[y
%
di
end
plot(c

```



cos)', ...

# Agenda

- Simulink QPSK Transceiver
- MATLAB System Objects
- MATLAB Transceiver

# MATLAB Transceiver

- As Obi-Wan would say, “Use the source, Luke!”

# Demo

# Connecting to the USRP Radio in MATLAB

- Use System objects to transmit or receive data
  - Define IP address
  - Define center frequency, gain, interpolation/decimation, LO offset
  - Generate code for acceleration
  - Burst mode for transmitter and receiver

# USRP System Objects

## comm.SDRuTransmitter

Properties:

```

        IPAddress: '192.168.10.2'
    CenterFrequencySource: 'Property'
    CenterFrequency: 2450000000
    ActualCenterFrequency: 2450000000
    LocalOscillatorOffsetSource: 'Property'
    LocalOscillatorOffset: 0
    ActualLocalOscillatorOffset: 0
    GainSource: 'Property'
    Gain: 8
    ActualGain: 8
    InterpolationFactorSource: 'Property'
    InterpolationFactor: 512
    ActualInterpolationFactor: 512
    UnderrunOutputPort: false
    EnableBurstMode: false
        IPAddress: '192.168.10.2'
    CenterFrequencySource: 'Property'
    CenterFrequency: 2450000000
    ActualCenterFrequency: 2450000000
    LocalOscillatorOffsetSource: 'Property'
    LocalOscillatorOffset: 0
    GainSource: 'Property'
    Gain: 8
    ActualGain: 8
    DecimationFactorSource: 'Property'
    DecimationFactor: 512
    ActualDecimationFactor: 512
    OverrunOutputPort: false
    SampleRate: 1
    OutputDataType: 'int16'
    FrameLength: 362
    
```

## Conclusion

- Connect to USRP radios with either MATLAB or Simulink