

# SATÉLITES DE TELECOMUNICAÇÕES

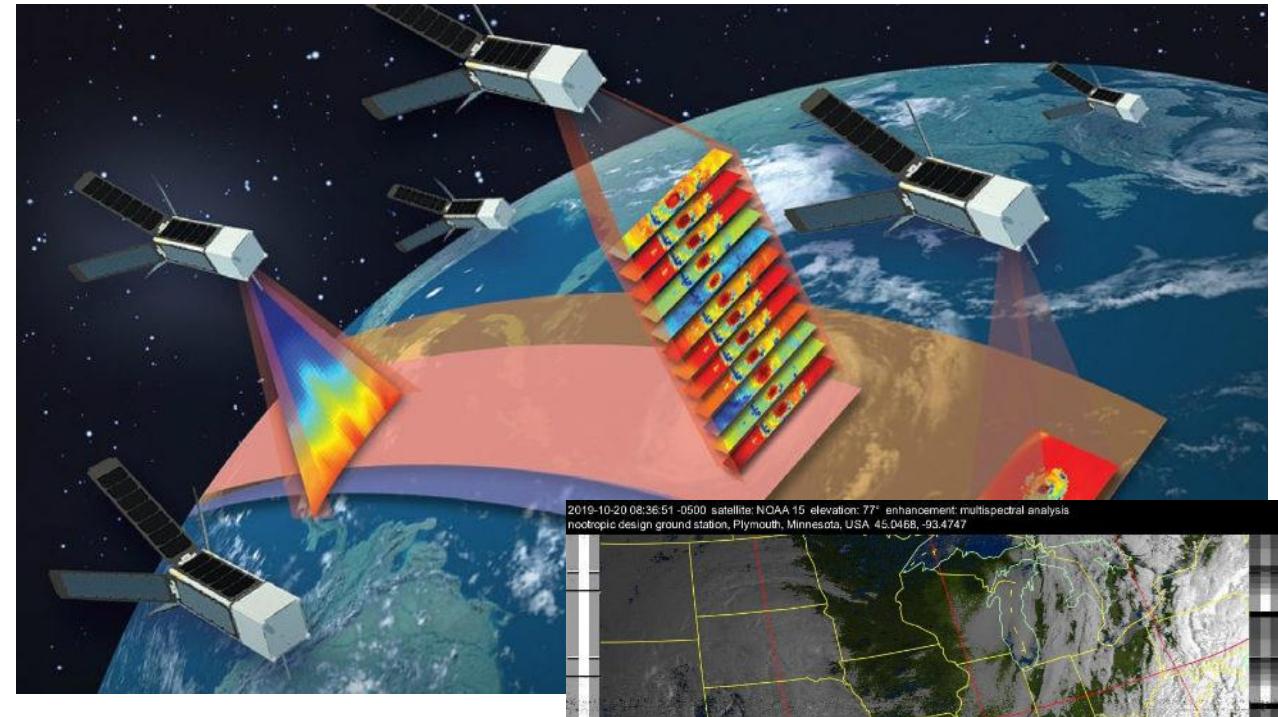
Desenvolvimento de  
Software Defined  
Radio (SDR) para  
estações terrestres de  
comunicação



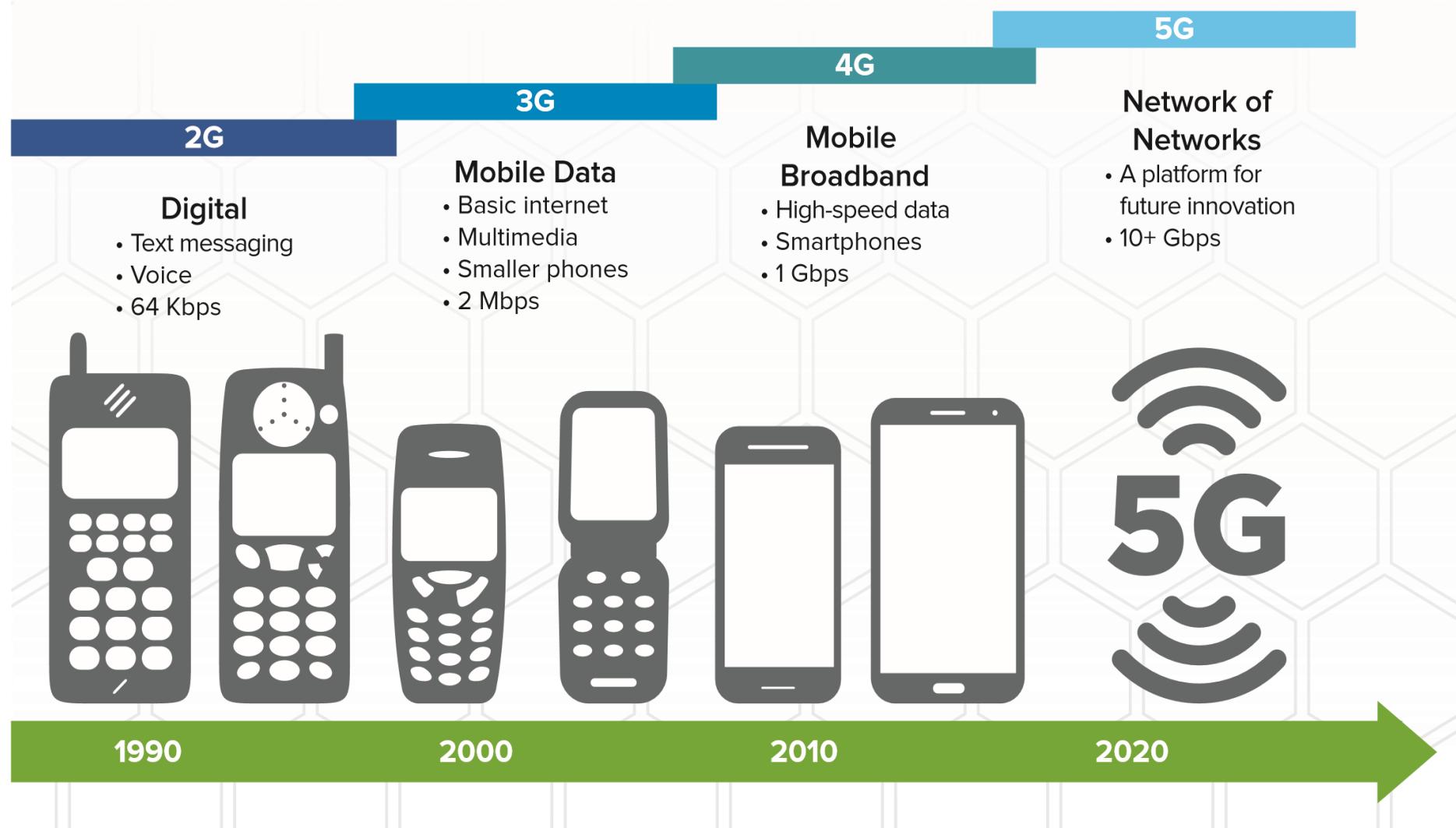
**Instalar Matlab**

Datas: 20 e 27 de Outubro, 03 de Novembro

1<sup>a</sup> aula



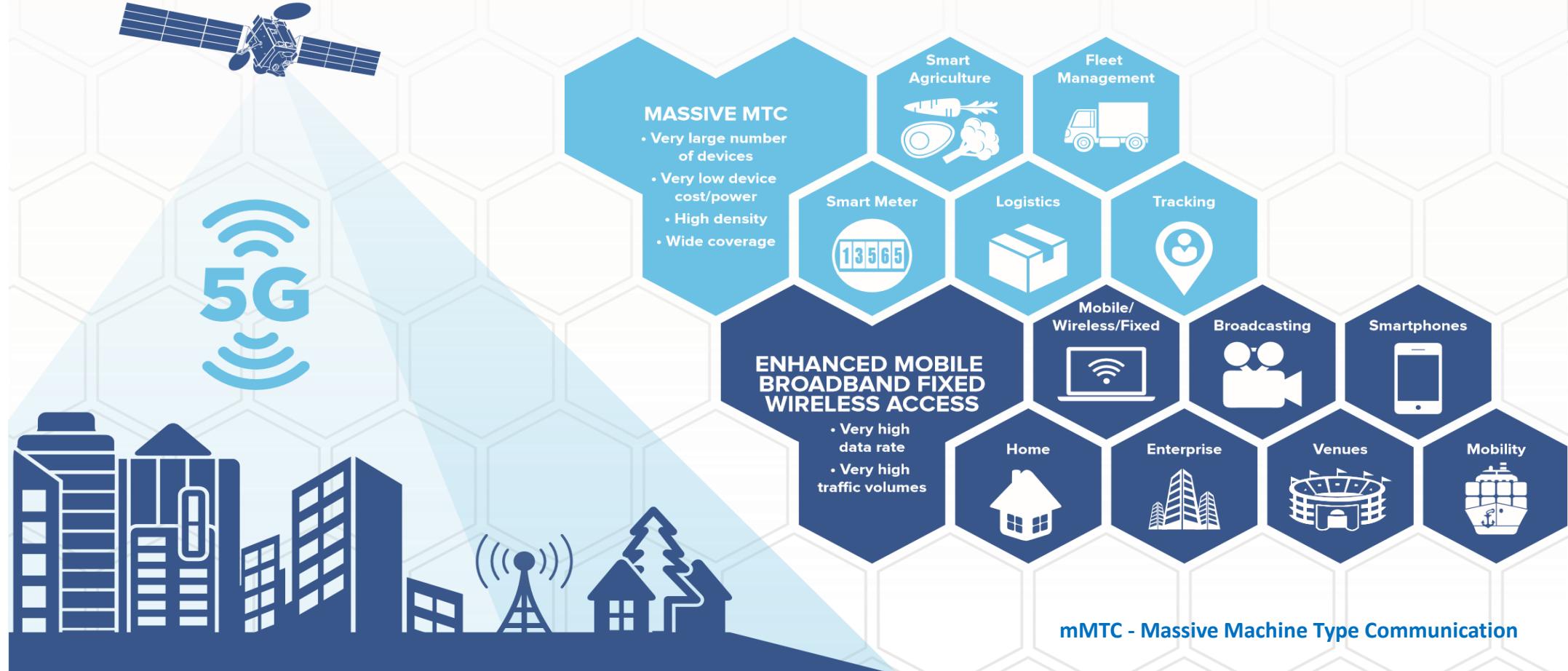
# INTRODUCTION TO THE SATELLITE TELECOMMUNICATIONS



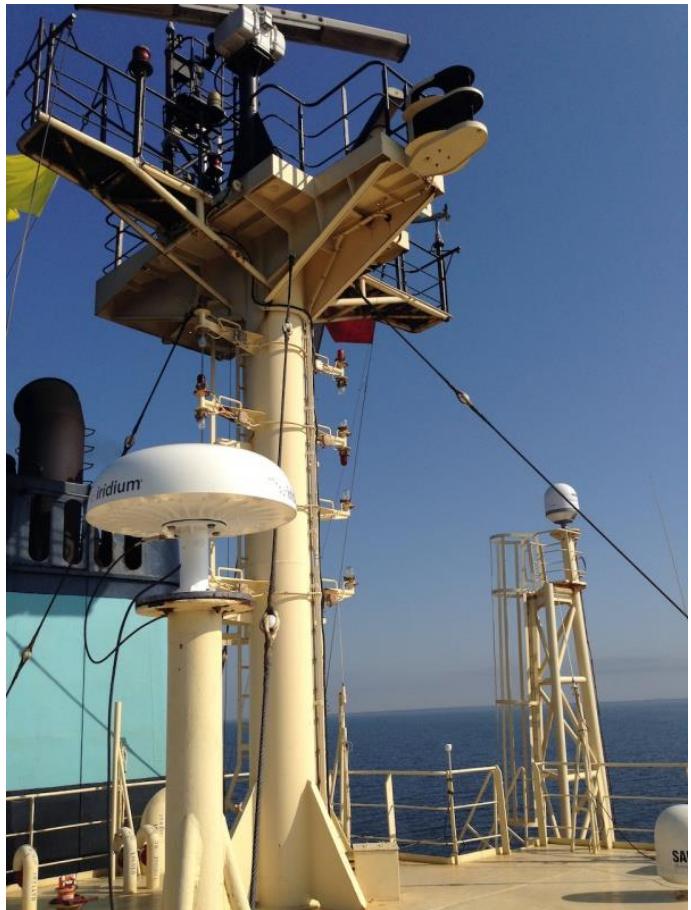
Ref: The Role of Space-based Communications in the 5G Era, Intelsat whitepaper

# INTRODUCTION TO THE SATELLITE TELECOMMUNICATIONS

## The Role of Space in 5G Use Cases



# INTRODUCTION TO THE SATELLITE TELECOMMUNICATIONS



# INTRODUCTION TO THE SATELLITE TELECOMMUNICATIONS

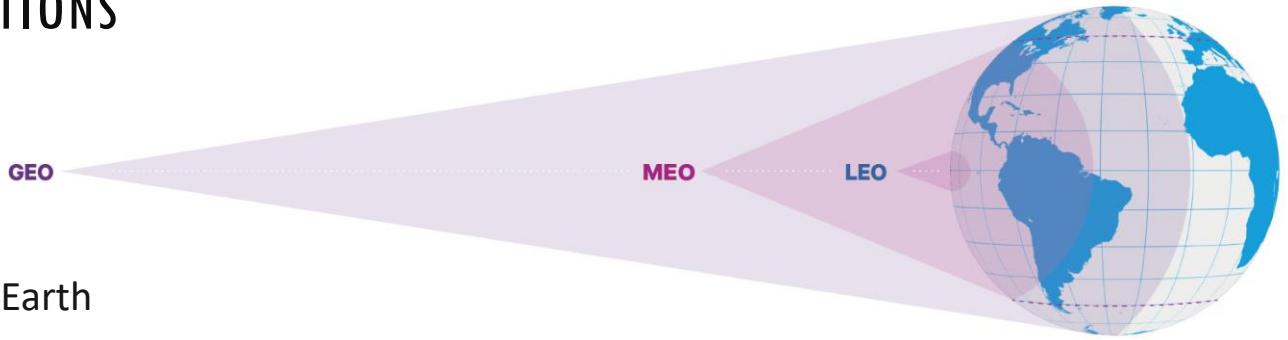
## Communications satellite

### GEO – Geosynchronous Orbit (Altitude 35,786km)

Match the rotation of the Earth → stationary to the Earth

Hundreds of GEO satellites are in orbit today

Delivering services such as weather data, broadcast TV, and some low-speed data communication (high latency)



### NGSO – Non-Geostationary Orbit

#### MEO – Medium Earth Orbit (Altitude 2000 to 35,786km)

It has been used for GPS and other navigation applications.

High Throughput Satellites (HTS), based on MEO constellations have been deployed to deliver low-latency, high-bandwidth data connectivity to service providers, government agencies, and commercial enterprises.

MEO satellites bring fibre-like performance to remote areas where laying fibre is not viable, such as cruise, commercial maritime, aero, offshore platforms, network backhaul in difficult terrain, and humanitarian relief operations.

#### LEO – Low Earth Orbit (Altitude 180 to 2000km)

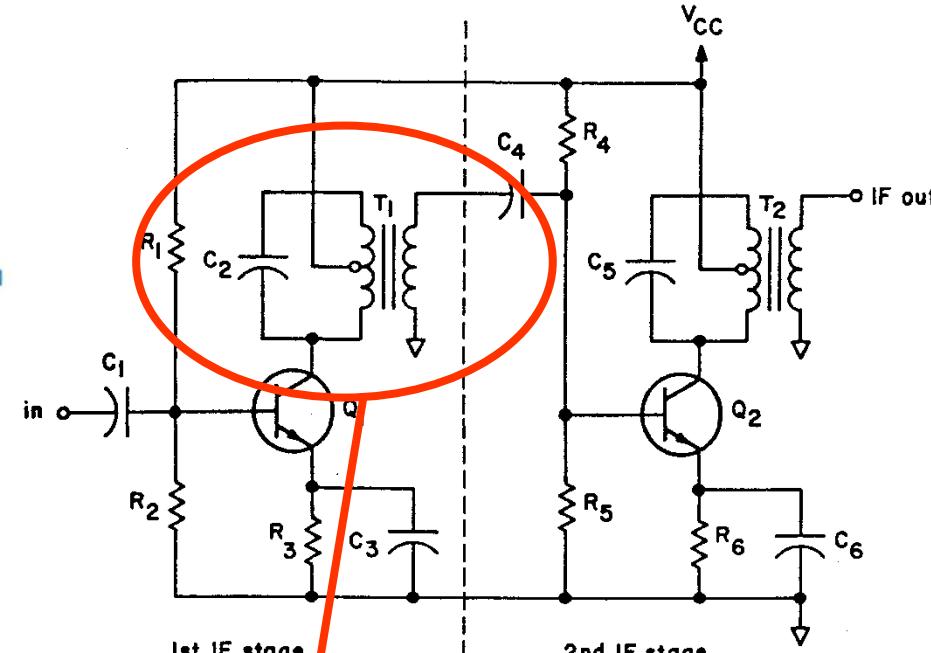
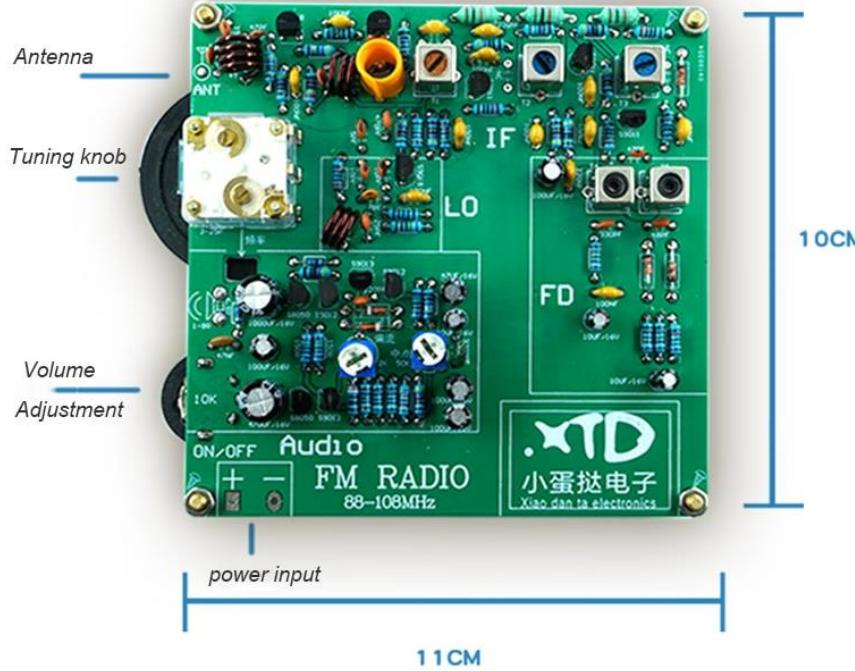
LEO is densely populated with thousands of satellites in operation today, primarily addressing science, imaging, and low-bandwidth telecommunications needs. The next generation of HTS LEO satellites intends to serve communication markets such as mass-consumer and enterprise broadband internet.

Ref: Stewart Sanders, GEO, MEO, AND LEO, "How orbital altitude impacts network performance in satellite data services" , SES whitpaper

# SOFTWARE-DEFINED RADIO IN SATELLITE COMMUNICATIONS

# Introduction to Software-Defined Radio

## Hardware (defined) Receivers



They are designed for a specific application and can hardly be adapted for other functionalities

They are tuned in the factory and remain unchanged

# Software Defined Radio – Original Concept

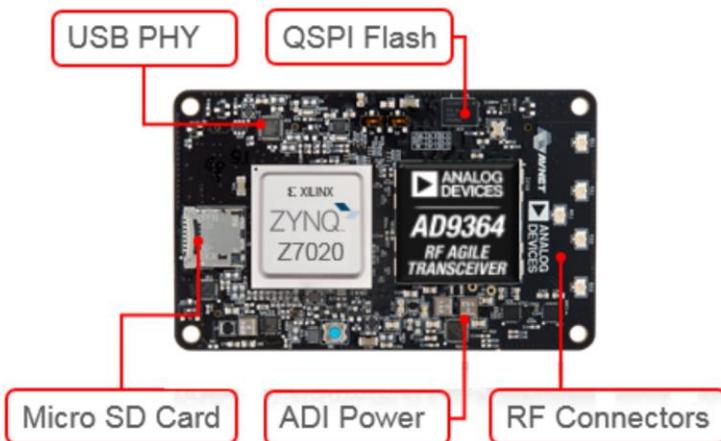
## What's the Software Defined Radios (SDRs) ?

- A radio in which some or all of the physical layer functions are software defined.
- This is contrasted with older radios where all of their functionality is determined by hardware.
- SDR serves as an enabling technology for intelligent radios

## SDR, Why Use It?

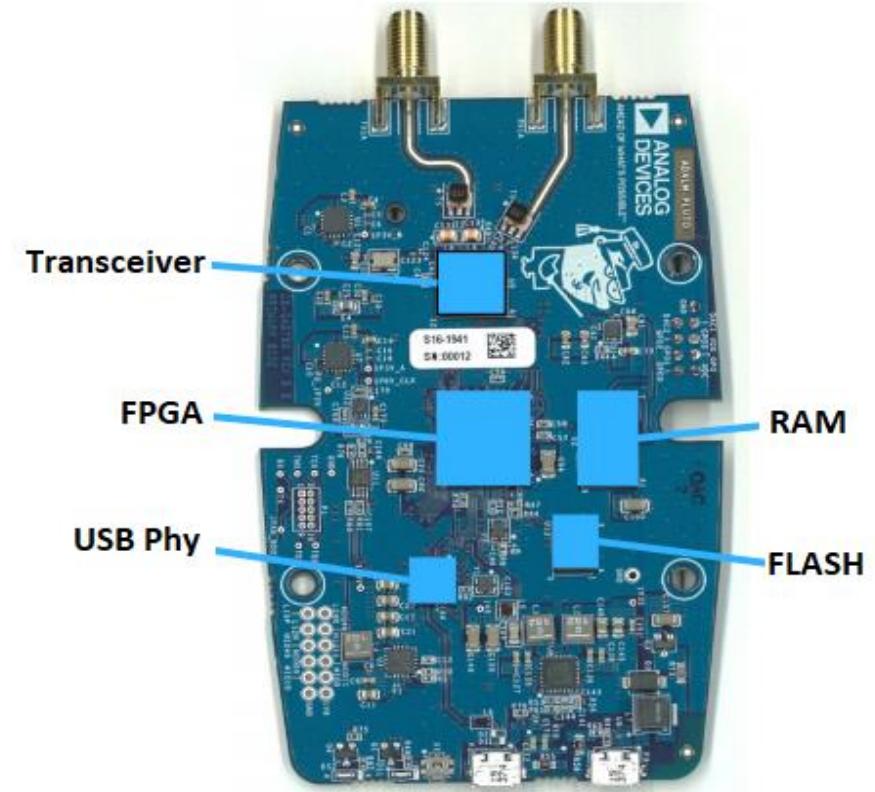
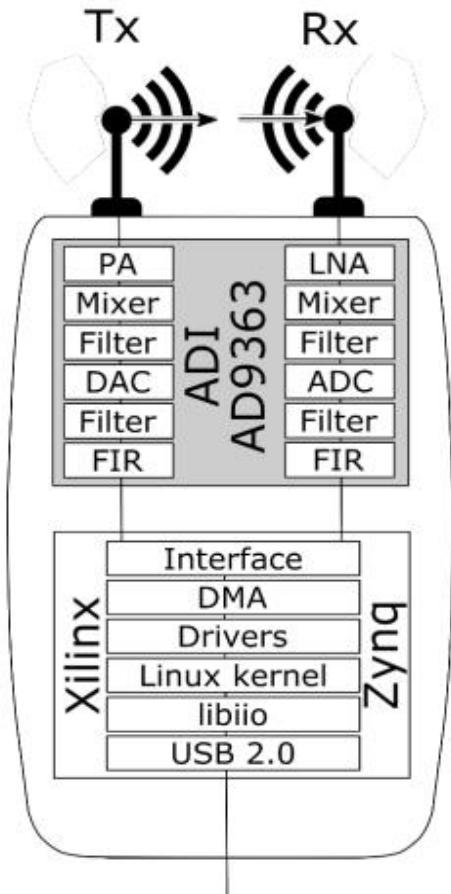
- Flexibility
- RAT (radio access technology) agnostic
- Can be updated on the fly
- Interoperability
- Easier to upgrade by a non-technician or remotely
- Minimal infrastructure requirements for use in the field
- Brings Analog and Digital World Together

# Commercial and Educational SDRS



# Educational SDR –Adalm Pluto

## Adalm-Pluto

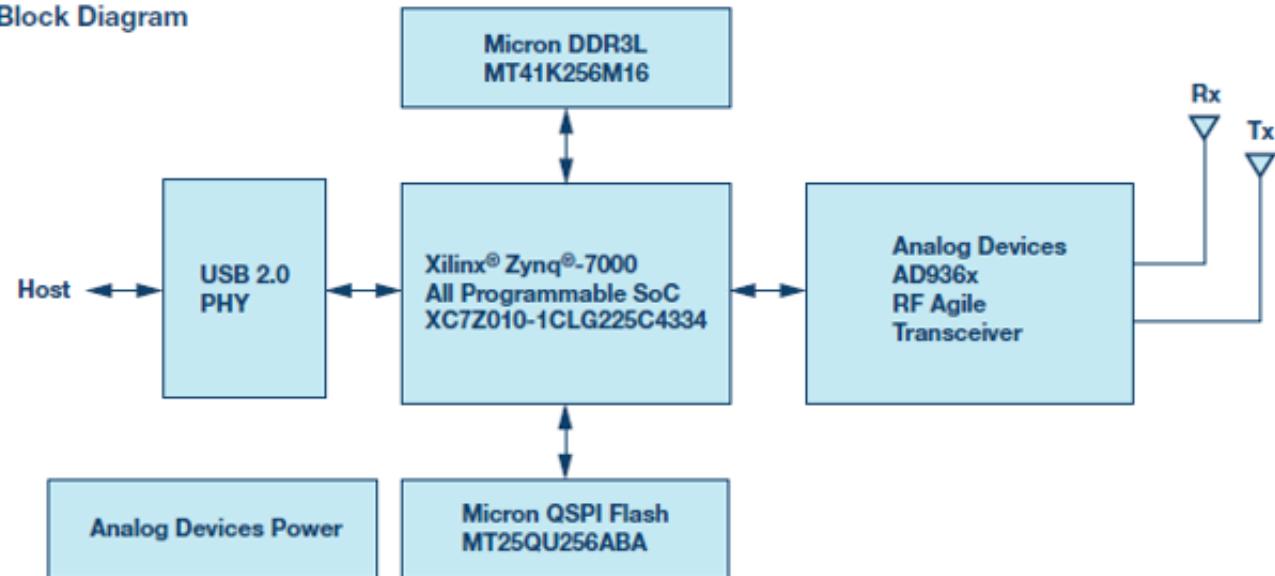


# Educational SDR –Adalm Pluto

## Adalm-Pluto

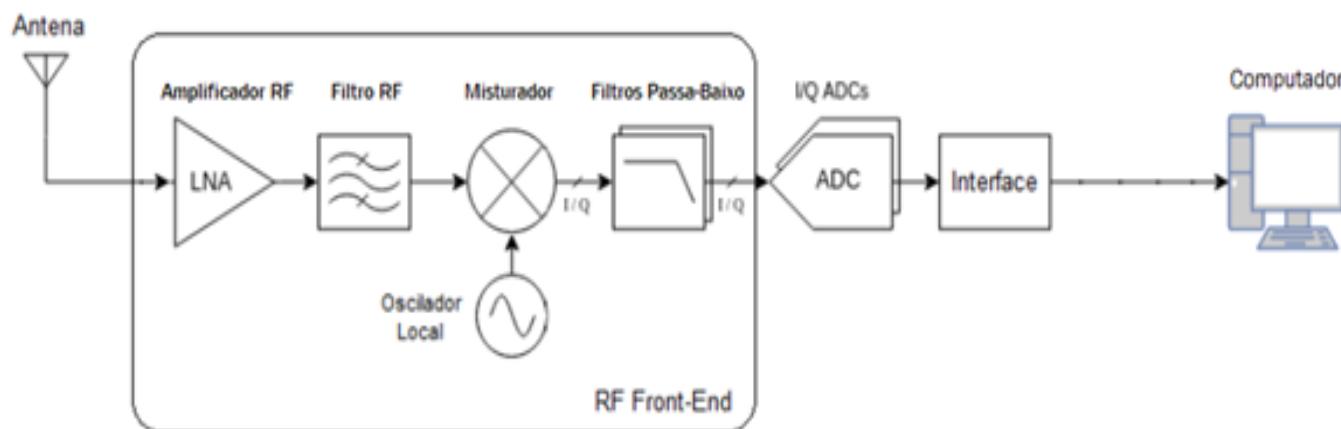
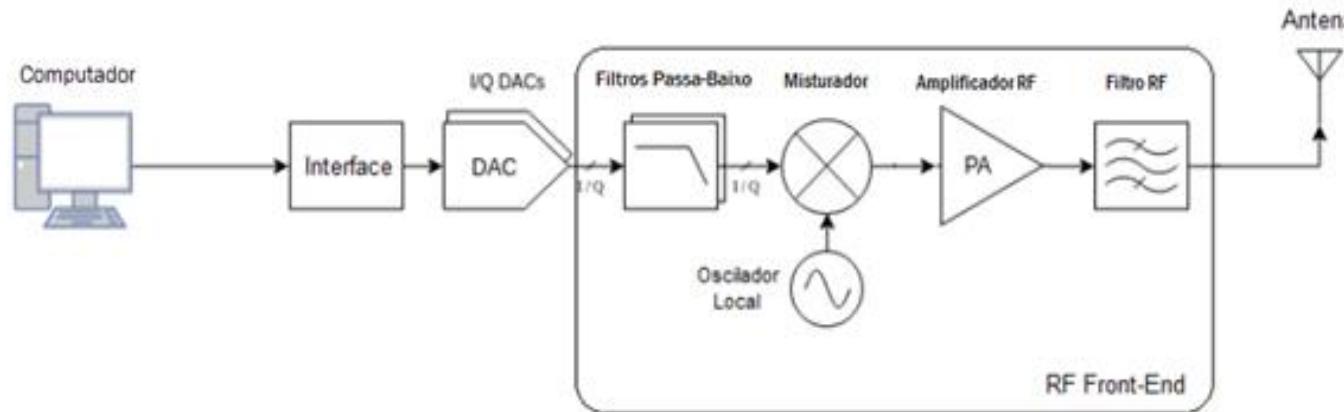
Specifications	Typical
<i>Power</i>	
DC Input (USB)	4.5 V to 5.5 V
<i>Conversion Performance and Clocks</i>	
ADC and DAC Sample Rate	65.2 kSPS to 61.44 MSPS
ADC and DAC Resolution	12 bits
Frequency Accuracy	±25 ppm
<i>RF Performance</i>	
Tuning Range	325 MHz to 3800 MHz
Tx Power Output	7 dBm
Rx Noise Figure	<3.5 dB
Rx and Tx Modulation Accuracy (EVM)	-34 dB (2%)
RF Shielding	None
<i>Digital</i>	
USB	2.0 On-the-Go
Core	Single ARM Cortex®-A9 @ 667 MHz
FPGA Logic Cells	28k
DSP Slices	80
DDR3L	4 Gb (512 MB)
QSPI Flash	256 Mb (32 MB)

Simplified Block Diagram

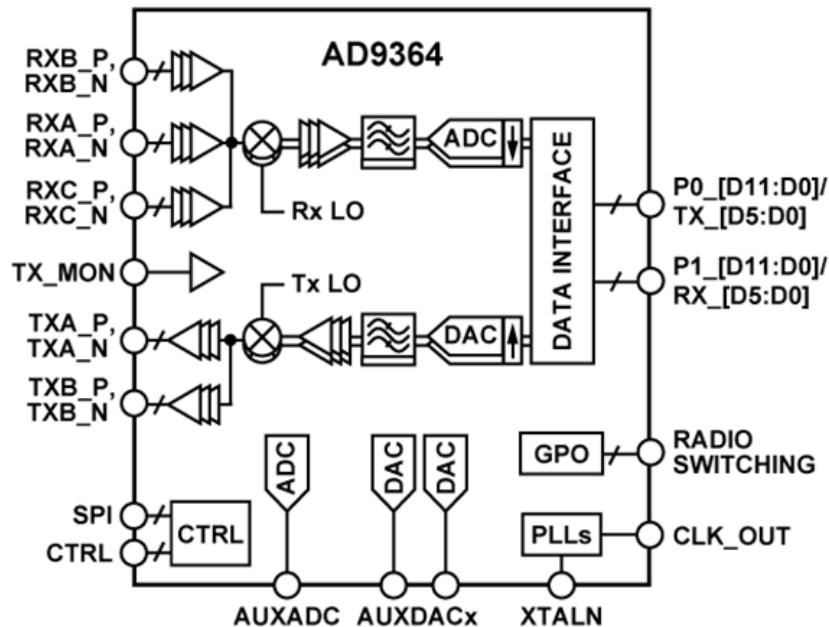


RF Transceiver	LO Tuning Range	Bandwidth
AD9363 (factory default)	325 - 3800 MHz	20 MHz
AD9364	70 - 6000 MHz	56 MHz

# SDR –Software-defined radio



# SDR –Software-defined radio



- RF 1 × 1 transceiver with integrated 12-bit DACs and ADCs
- Band: 70 MHz to 6.0 GHz
- Supports TDD and FDD operation
- Tunable channel bandwidth: <200 kHz to 56 MHz
- Dual receivers: 6 differential or 12 single-ended inputs
- Superior receiver sensitivity with a noise figure < 2.5 dB
- RX gain control
- Real-time monitor and control signals for manual gain
- Independent automatic gain control

The receiver consists of 2 high dynamic range ADCs digitize the received I and Q signals and pass them through configurable decimation filters and 128-tap FIR filters to produce a 12-bit output signal at the appropriate sample rate.



Price: \$130.00

# Space qualified SDRS

The screenshot shows the GOMSpace website's product page for the Space qualified Software Defined Radio (SDR) Platform. At the top, there is a navigation bar with links for SOLUTIONS, PLATFORMS, PRODUCTS, CONSTELLATION MANAGEMENT, CAREERS, INVESTOR RELATIONS, and CONTACT. There are also search, star, and user icons. Below the navigation is a banner featuring a satellite view of Earth with green and blue data overlays. The main content area has a breadcrumb navigation: Products > Mission Payloads > Software Defined Radio. On the left, there is a large image of the SDR hardware, which is a black rectangular module with various connectors and markings like "TRIM" and "Z180". Below this image is a button labeled "ADD TO QUOTE REQUEST" with a star icon. The main text on the right describes the SDR platform as being space qualified for sensing and communication applications using advanced FPGA technology. It details its use for high-speed S-band ground links and inter-satellite links, mentioning its application in GOMX-3 for signal sensing and spectrum characterization. It also notes its compact design and advanced processing capabilities like image processing. The page concludes with a "Highlighted features" section listing flight-proven software, a standard motherboard concept, and support for FPGA and transceiver modules.

**Space qualified Software Defined Radio (SDR) Platform for Sensing and Communication Applications using powerful and advanced FPGA Technology**

The SDR platform is utilized for a high-speed S-band ground link and for inter-satellite links utilizing S- or K band in a highly miniaturized radio system for long distance communication.

On GOMX-3 the SDR platform is used for signal sensing with an L-band antenna for spectrum characterization. Significant calculation capacity for advanced signal processing and detection techniques is available in a very compact design.

The FPGA module can be available for other advanced processing requirements such as image processing etc. utilizing the high capacity of the system. Software is available for in-orbit programming or applying scripting tools limiting size of program uploads.

To help our customers get started with developing applications on our SDR platform we offer a [standard training program](#). To learn more about this download our flyer in the menu to the left or contact our sales department.

**Highlighted features:**

- Flight proven Software Defined Radio platform software for payload application development
- Standard GomSpace Motherboard – daughter board concept, up to 4 modules
- FPGA module and Transceiver modules

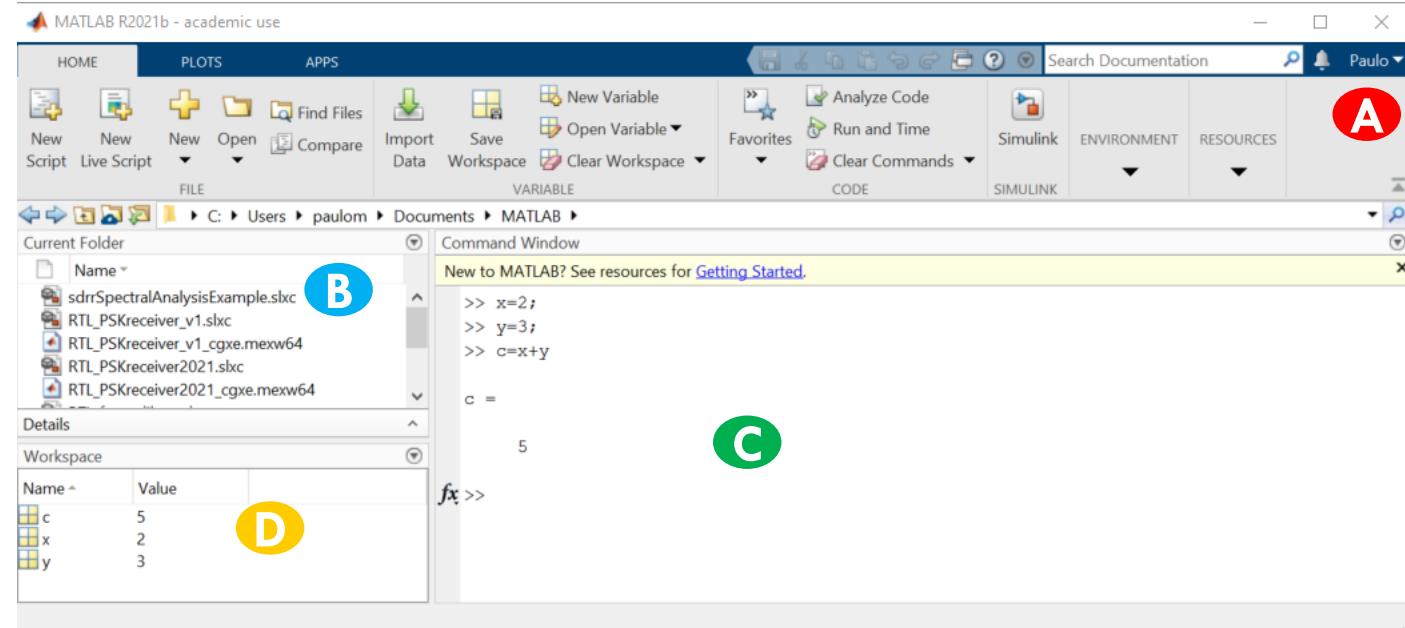
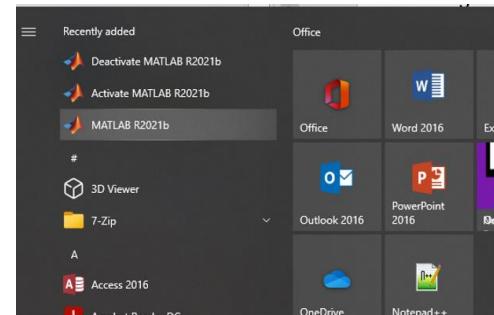
# EXERCISE 1

# GETTING STARTED WITH MATLAB AND SIMULINK

# EXERCISE 1) GETTING STARTED WITH MATLAB AND SIMULINK

- 1) Open Matlab
- 2) Explore the MATLAB interface
- 3) Execute Code Via the Command Window

Example:



The **Menu Ribbon**, shown here with the main 'Home' tab open. This allows you to work with files and variables, open Simulink, setup the environment, and access help and support.

The **Current Folder** pane shows the files within the present folder. This facility acts like Windows Explorer — you can open files, as well as move, rename and delete them.

The **Command Window** is the place for directly entering commands, including single statements or short sets of statements, running scripts and calling functions.

The **Workspace** shows all of the variables currently held in memory, of various types (scalars, arrays, matrices, strings, structures, etc.). These can be inspected / edited here.

# EXERCISE 1) HELP

## Command Window

New to MATLAB? See resources for [Getting Started](#).

```
>> cos(pi)  
  
ans =  
  
-1  
  
>> help cos  
cos Cosine of argument in radians.  
cos(X) is the cosine of the elements of X.  
  
See also acos, cosd, cospi.  
  
Documentation for cos  
Other functions named cos  
  
>> doc cos
```

Help

cos

## Documentation

CONTENTS

Documentation Home

MATLAB

Mathematics

Elementary Math

Trigonometry

**cos**

ON THIS PAGE

Syntax

Description

Examples

Input Arguments

Output Arguments

More About

Extended Capabilities

See Also

Other uses of cos

All Examples Functions

**COS**

Cosine of argument in radians

**Syntax**

`Y = cos(X)`

**Description**

`Y = cos(X)` returns the cosine for each element of `X`. The `cos` function operates element-wise on arrays. The function accepts both real and complex inputs.

- For real values of `X`, `cos(X)` returns real values in the interval [-1, 1].
- For complex values of `X`, `cos(X)` returns complex values.

**Examples**

**Plot Cosine Function**

Plot the cosine function over the domain  $-\pi \leq x \leq \pi$ .

```
x = -pi:0.01:pi;
plot(x,cos(x))
grid on
```

# EXERCISE 1) GETTING STARTED WITH MATLAB AND SIMULINK

Documentation Examples Functions More ▾

## MATLAB

The Language of Technical Computing

Millions of engineers and scientists worldwide use MATLAB® to analyze and design the systems and products transforming our world. The matrix-based MATLAB language is the world's most natural way to express computational mathematics. Built-in graphics make it easy to visualize and gain insights from data. The desktop environment invites experimentation, exploration, and discovery. These MATLAB tools and capabilities are all rigorously tested and designed to work together.

MATLAB helps you take your ideas beyond the desktop. You can run your analyses on larger data sets, and scale up to clusters and clouds. MATLAB code can be integrated with other languages, enabling you to deploy algorithms and applications within web, enterprise, and production systems.

**Get Started**  
Learn the basics of MATLAB

**Language Fundamentals**  
Syntax, array indexing and manipulation, data types, operators

**Data Import and Analysis**  
Import and export data, including large files; preprocess data, visualize and explore

**Mathematics**

R2021b

[Release Notes](#)  
[PDF Documentation](#)

Documentation Search R2021b Documentation

CONTENTS Documentation Home MATLAB

## PDF Documentation for MATLAB

**MATLAB Primer**

MATLAB Desktop Tools and Development Environment  
MATLAB Data Import and Export  
MATLAB Mathematics  
MATLAB Data Analysis  
MATLAB Programming Fundamentals  
MATLAB Object-Oriented Programming  
MATLAB Graphics  
MATLAB 3-D Visualization  
MATLAB App Building  
MATLAB External Interfaces  
MATLAB C/C++, Fortran, Java, and Python API Reference  
MATLAB MAT-File Format  
MATLAB Function Reference  
MATLAB Release Notes

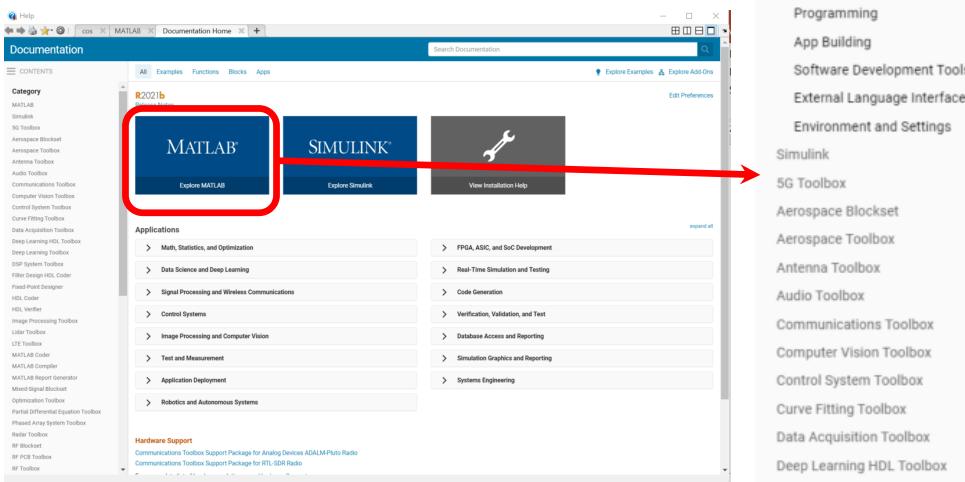
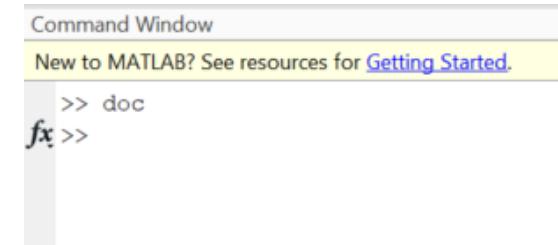
**Related Papers**  
MATLAB ODE Suite  
Implicitly Restarted Arnoldi/Lanczos Methods For Large Scale Eigenvalue Calculations

# EXERCISE 1) GETTING STARTED WITH MATLAB AND SIMULINK

```
Command Window
New to MATLAB? See resources for Getting Started.
>> doc
fx >>
```

The screenshot shows the MATLAB Documentation Home page for R2021b. The top navigation bar includes links for Help, MATLAB, Documentation Home, and a search bar labeled "Search Documentation". Below the navigation, there are tabs for All, Examples, Functions, Blocks, and Apps. A sidebar on the left lists various MATLAB toolboxes under the "Category" heading. The main content area features three large buttons: "MATLAB" (Explore MATLAB), "SIMULINK" (Explore Simulink), and "View Installation Help" (with a wrench icon). Below these are sections for "Applications" (Math, Statistics, and Optimization; Data Science and Deep Learning; Signal Processing and Wireless Communications; Control Systems; Image Processing and Computer Vision; Test and Measurement; Application Deployment; Robotics and Autonomous Systems) and "Hardware Support" (Communications Toolbox Support Package for Analog Devices ADALM-Pluto Radio; Communications Toolbox Support Package for RTL-SDR Radio).

# EXERCISE 1) GETTING STARTED WITH MATLAB AND SIMULINK



The image shows the MATLAB Documentation page for the "MATLAB" category. The page title is "MATLAB" and the subtitle is "The Language of Technical Computing". The content includes a "Get Started" section with a red box around it, followed by sections on "Language Fundamentals", "Data Import and Analysis", "Mathematics", "Graphics", "Programming", "App Building", "Software Development Tools", "External Language Interfaces", and "Environment and Settings". On the right side, there are links for "Release Notes" and "PDF Documentation", with the "PDF Documentation" link also highlighted with a red box. A feedback section at the bottom asks "How useful was this information?" with a scale from 1 to 5.

# Introducing Simulink

Simulink is a block diagram environment for multidomain simulation and Model-Based Design

It can be created block diagrams, where blocks represent parts of a system.

A block can represent a physical component, a small system, or a function.

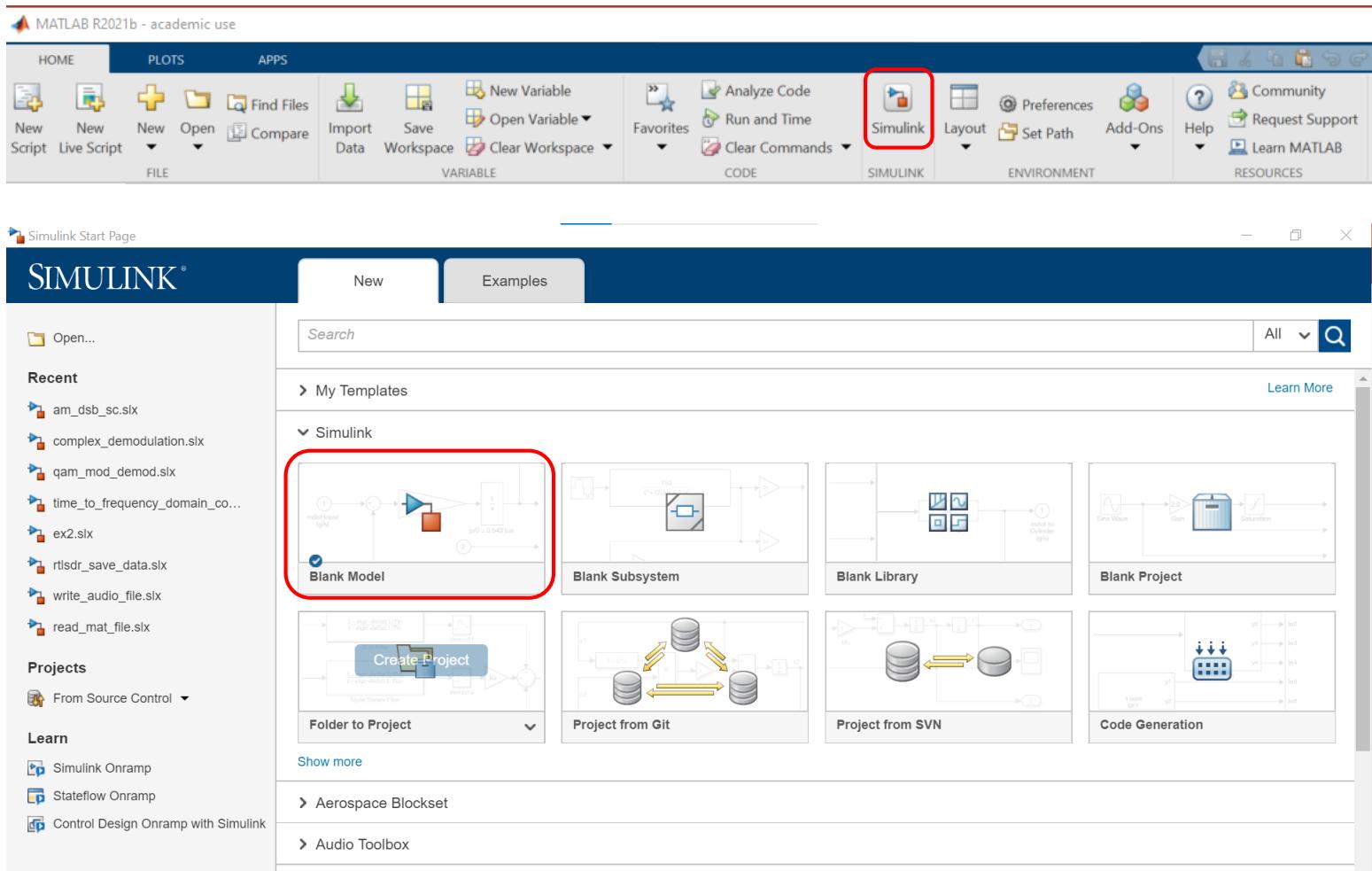
Simulink provides block libraries that are collections of blocks grouped by functionality

## How to start?

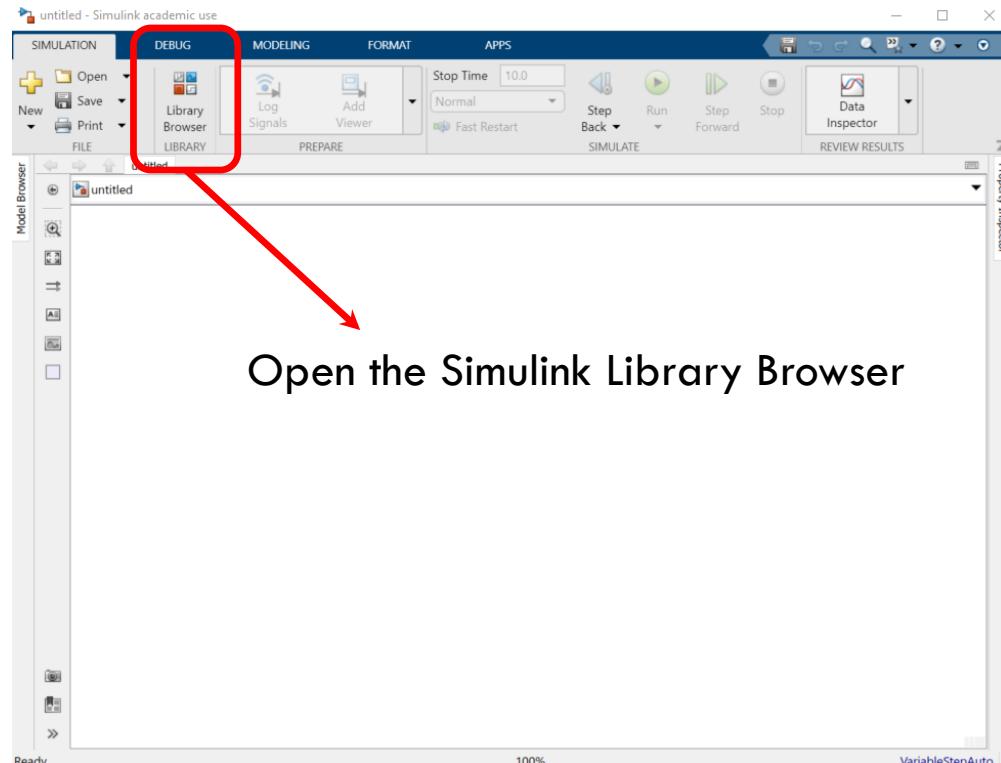
- To start Simulink, from the MATLAB Home tab click in Simulink



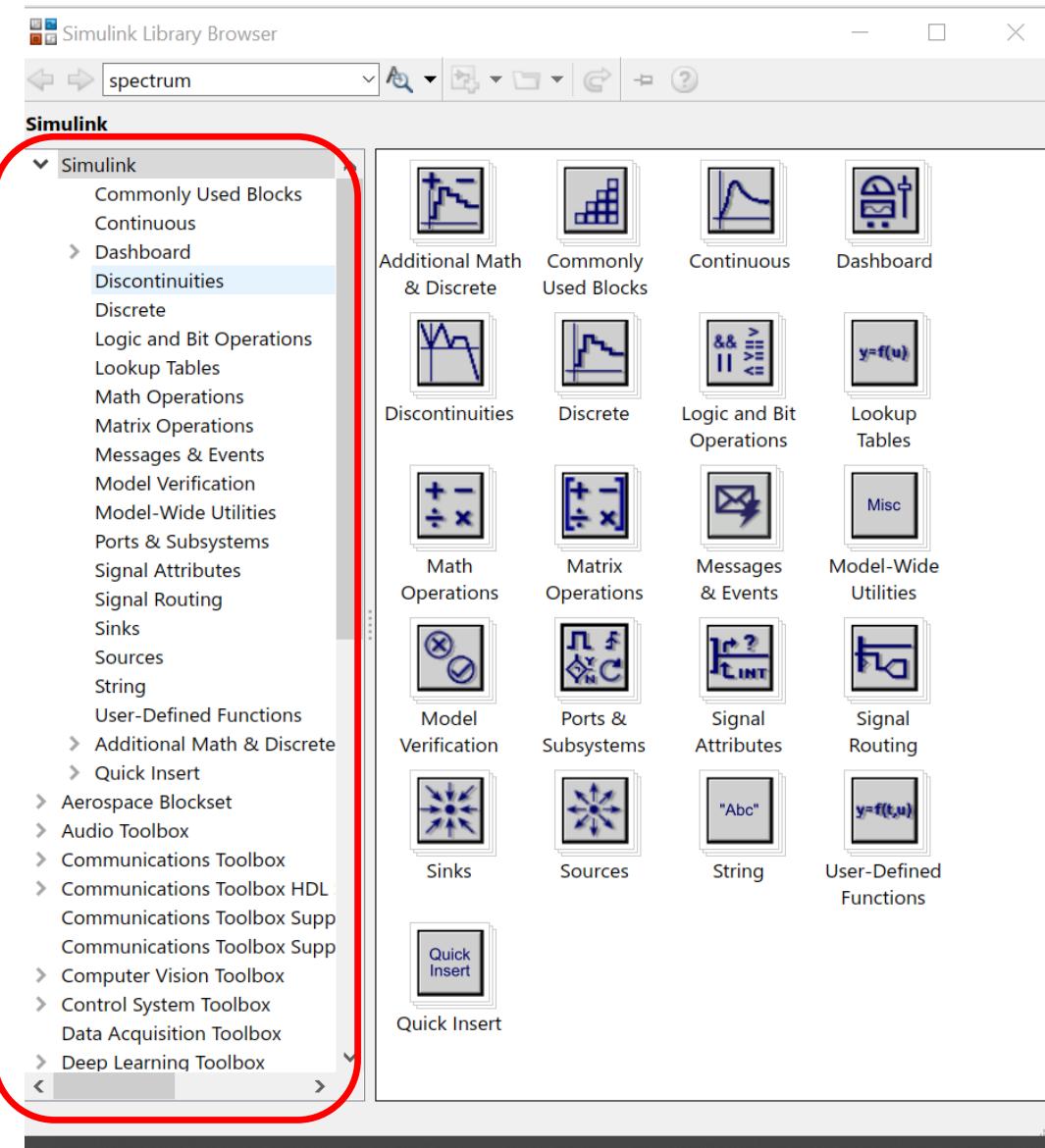
- At the new window, click on “Blank Model”



# Introducing Simulink



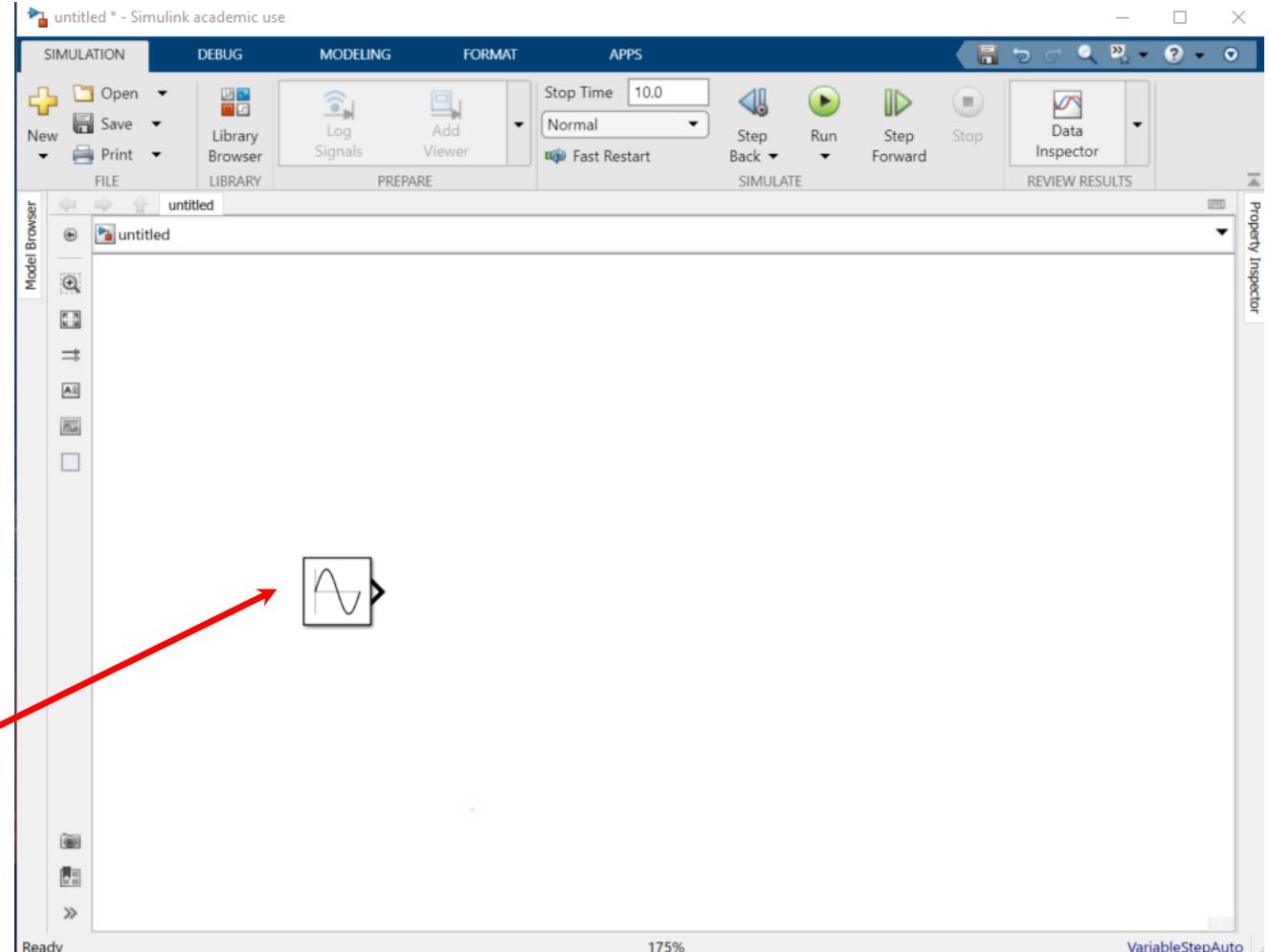
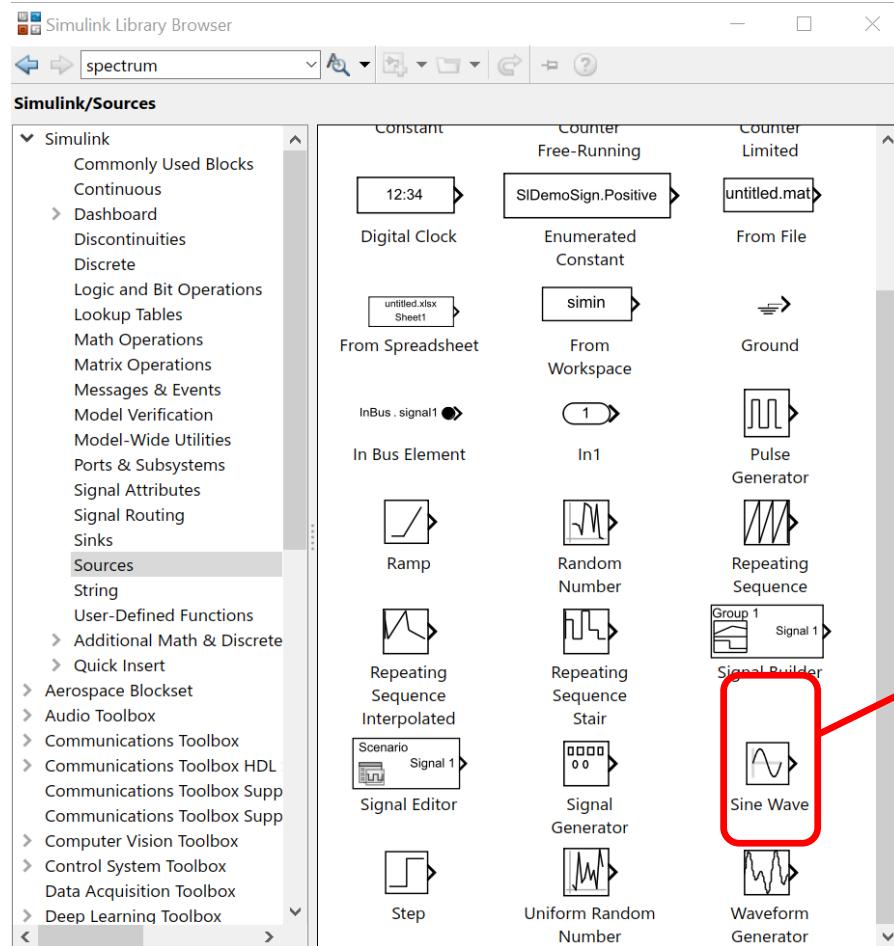
Open the Simulink Library Browser



List of available libraries

# Introducing Simulink

## Building and Simulated a First System



To introduce it into the design, on the Step block, and drag it into the left hand side of the 'first\_system' Simulink model, then release the mouse.

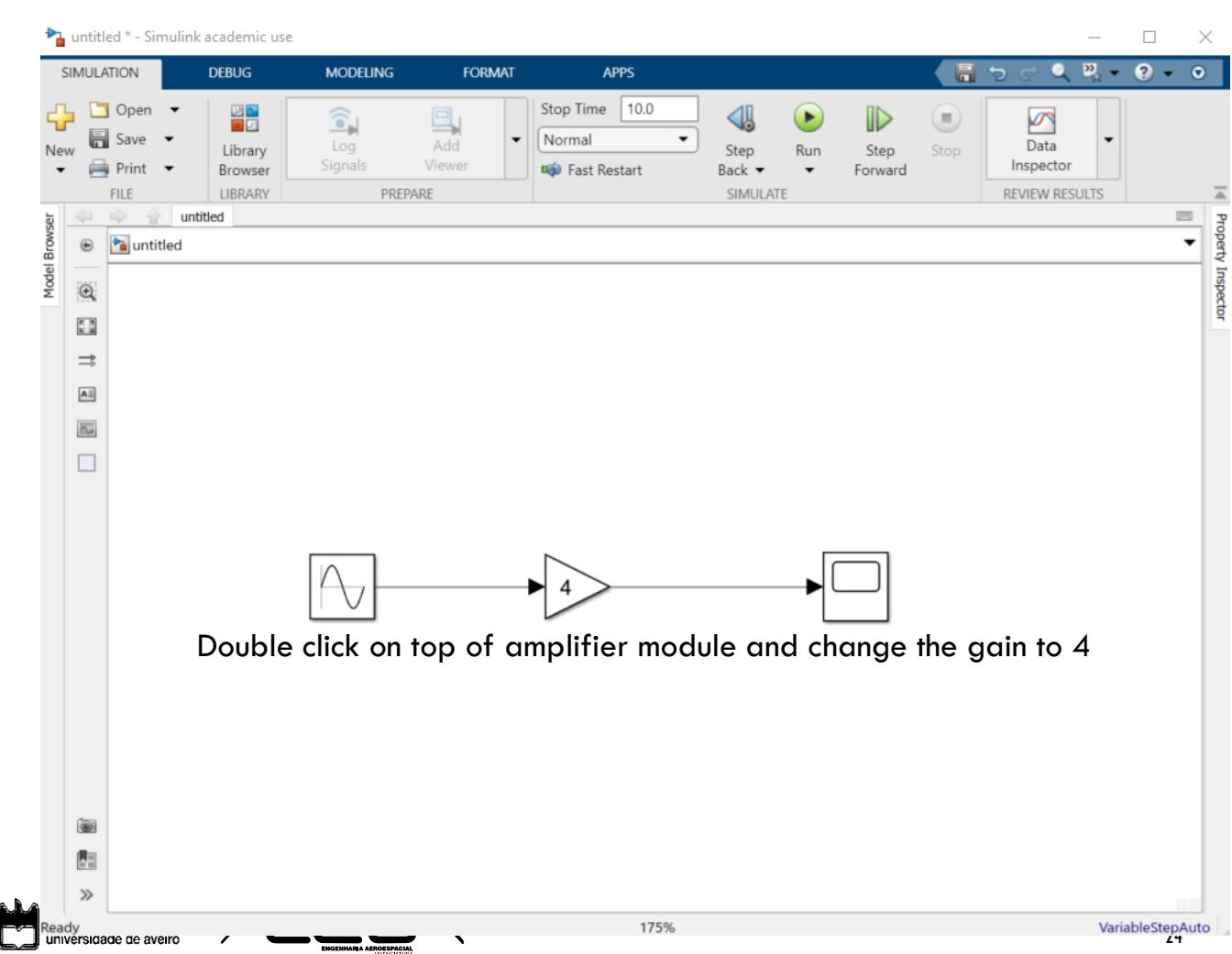
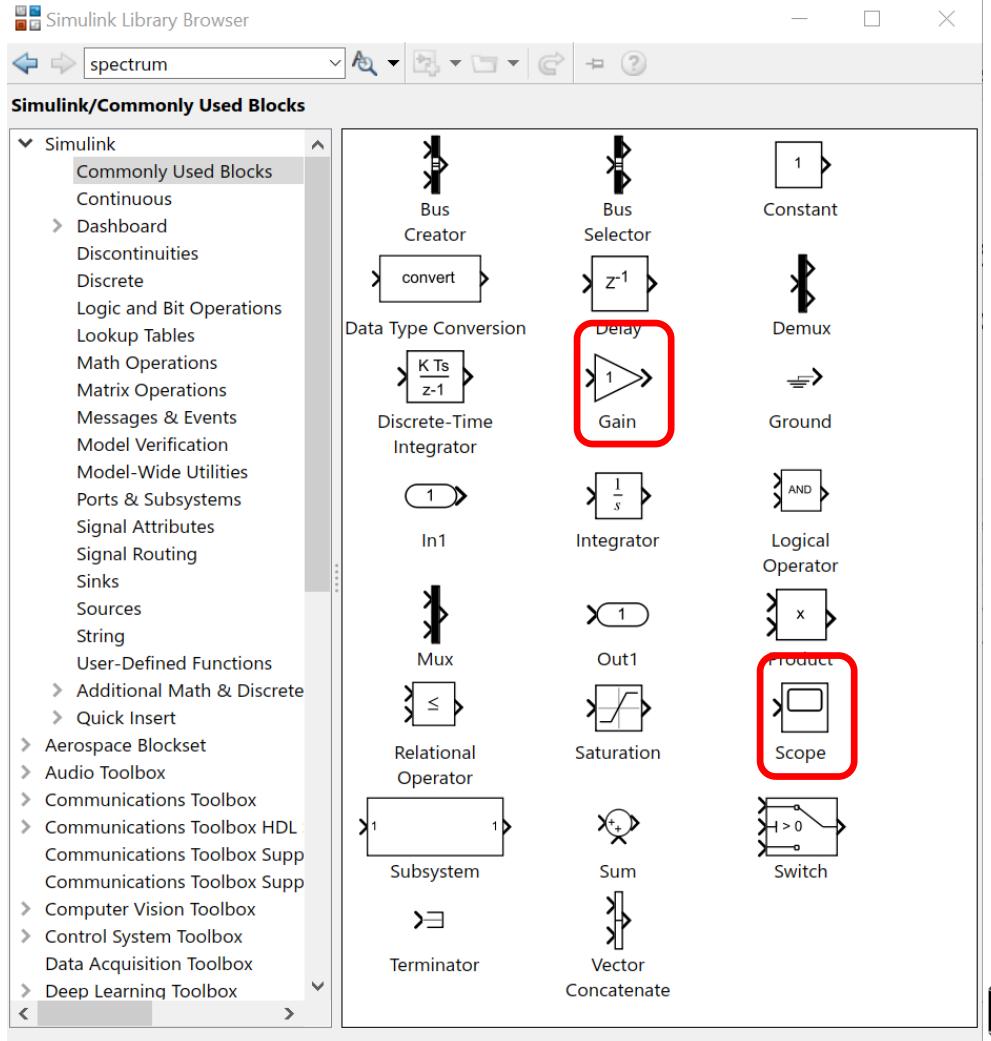
# Introducing Simulink

## Building and Simulated a First System

Insert Gain and Scope and connect the blocks together with wires.

Firstly, hover the mouse over the output port of the source, until you see a cross-hair appear.

Then  and drag the mouse across to the input port of the Amplifier

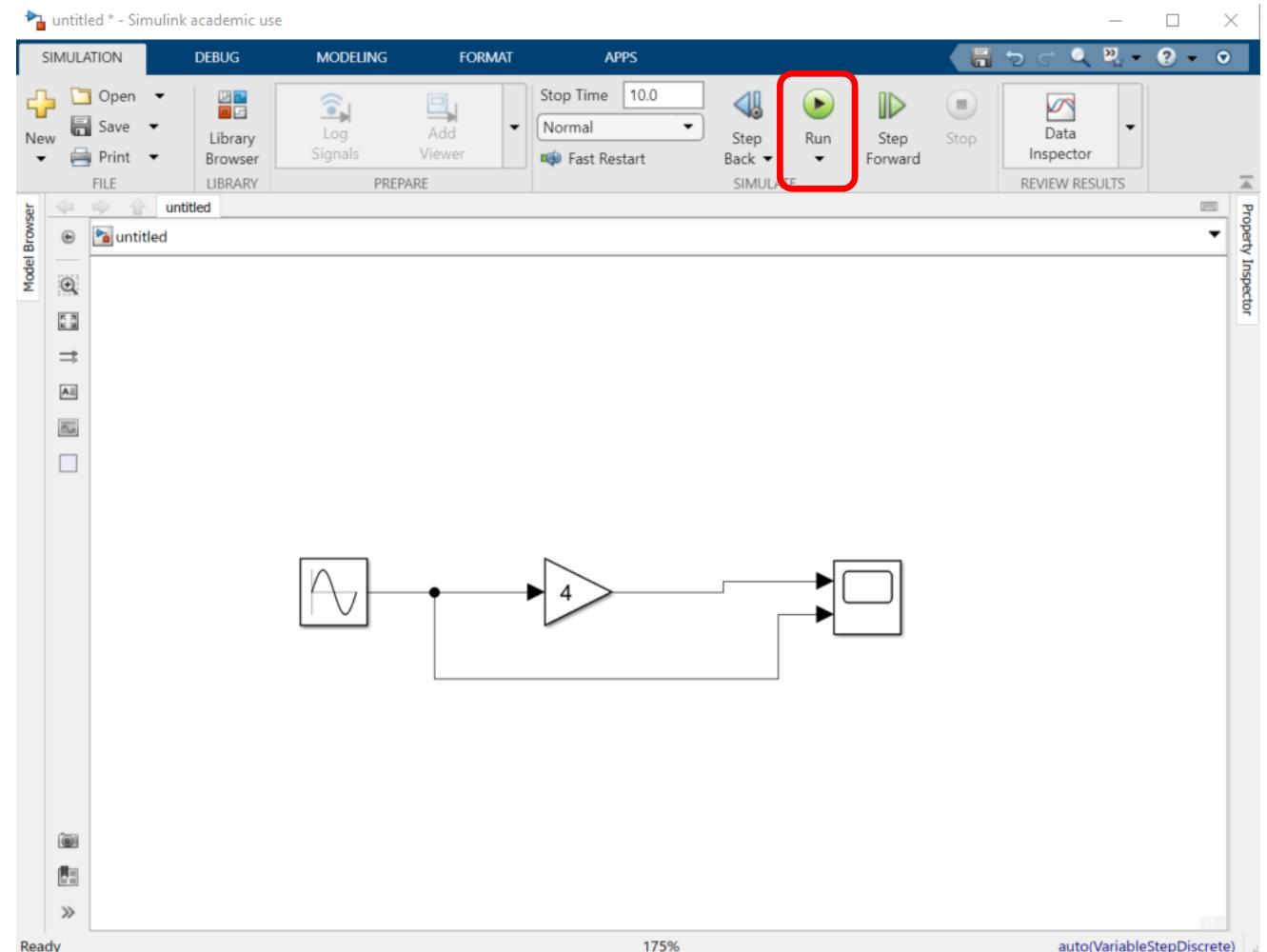
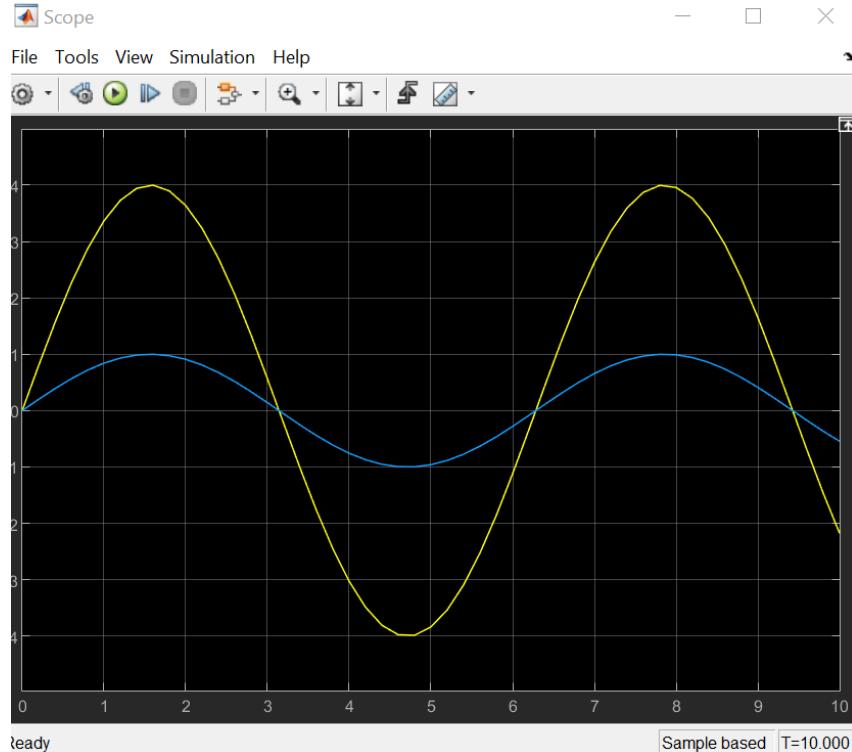


# Introducing Simulink

## Building and Simulated a First System

Try to include a new wire from source to scope ☺

Click on Run and click on Scope

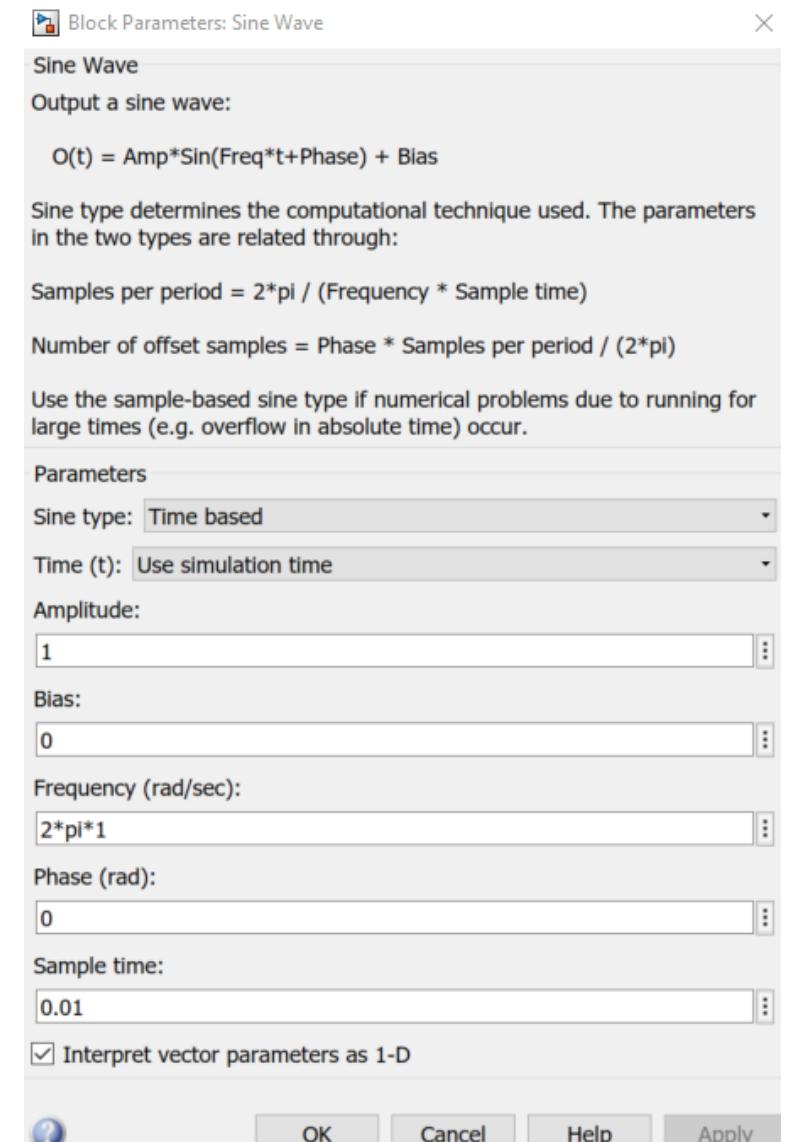


- Is this what you were expecting?
- Determine the mathematical expression of the signal before and after the amplifier
- Try to change the gain and see what happens

# Introducing Simulink

## Building and Simulated a First System

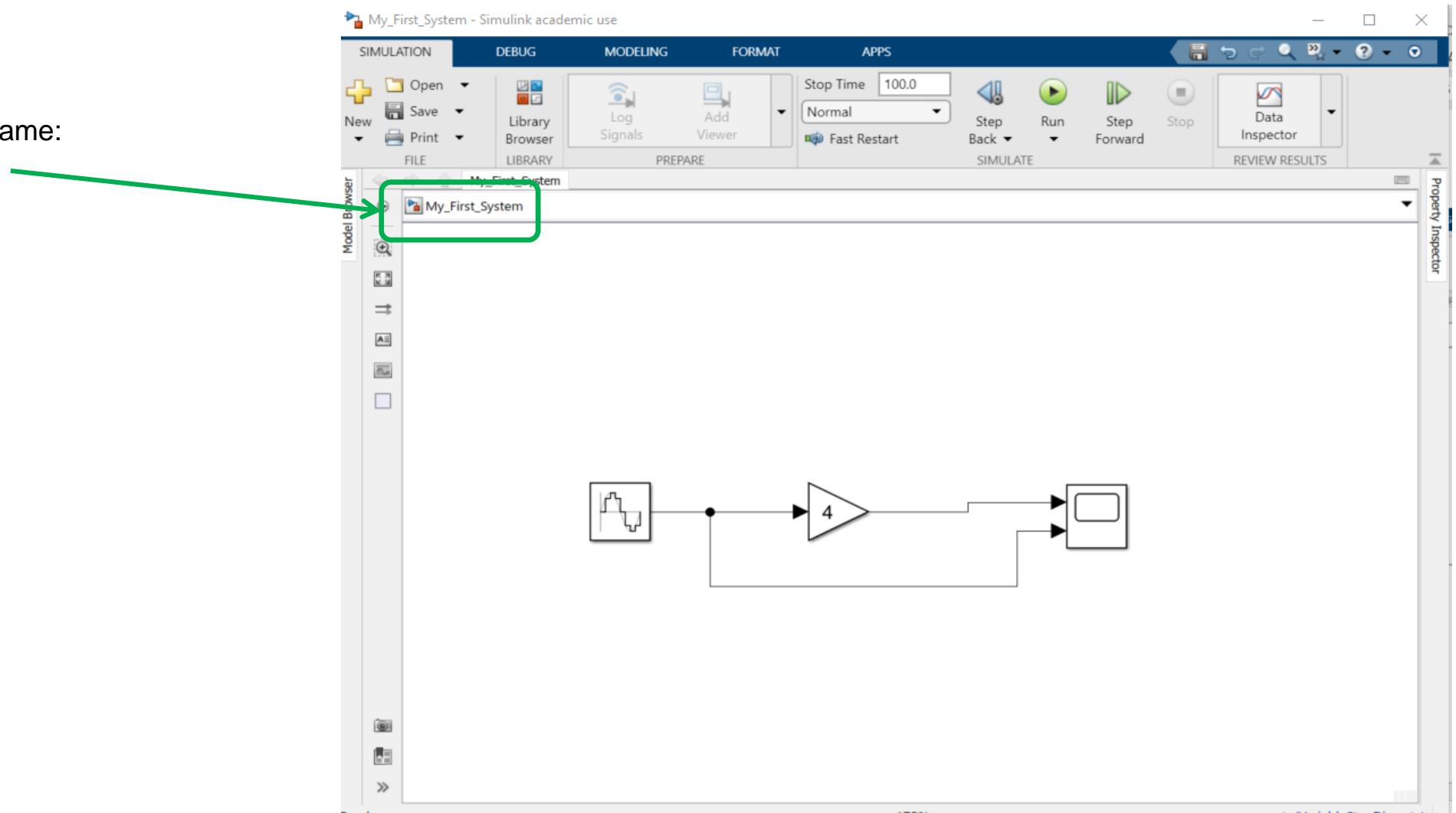
- Open the Block Parameters of Sine Wave by double clicking
- Change de Frequency (rad/s) to  $2\pi$  and Sample time to 0.01
- Analyze the waveforms at the scope and determine the new mathematical expressions



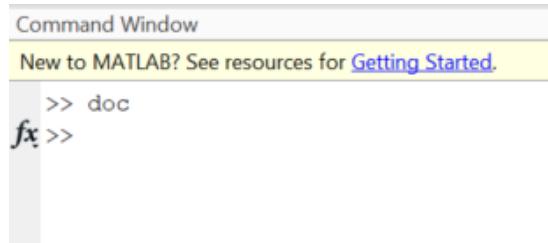
# Introducing Simulink

## Building and Simulated a First System

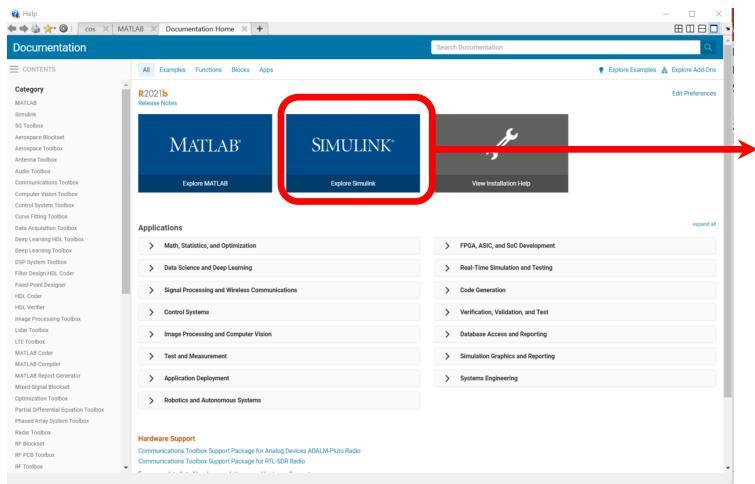
Save in your directory with the name:  
“My\_First\_System”



# EXERCISE 1) GETTING STARTED WITH MATLAB AND SIMULINK



>> doc  
fx >>



Documentation Search R2021b Documentation

All Examples Functions Blocks

## Simulink

Simulation and Model-Based Design

R2021b

**Release Notes** **PDF Documentation**

**Get Started with Simulink**

Learn the basics of Simulink

**Applications**

Example models illustrating specific functionality and applications

**Simulink Environment Fundamentals**

Simulation and Model-Based Design

Simulink® is a block diagram environment for multidomain simulation and Model-Based Design. It supports system-level design, simulation, automatic code generation, and continuous test and verification of embedded systems. Simulink provides a graphical editor, customizable block libraries, and solvers for modeling and simulating dynamic systems. It is integrated with MATLAB®, enabling you to incorporate MATLAB algorithms into models and export simulation results to MATLAB for further analysis.

CONTENTS

- Simulink
- Get Started with Simulink
- Applications
- Simulink Environment Fundamentals
- Modeling
- Simulation
- Project Management
- Block Authoring and Simulation Integration
- Simulink Supported Hardware
- 5G Toolbox
- Aerospace Blockset
- Aerospace Toolbox
- Antenna Toolbox
- Audio Toolbox
- Communications Toolbox

# EXERCISE 1) GETTING STARTED WITH MATLAB AND SIMULINK

Documentation

CONTENTS

All Examples Functions Blocks

## Simulink

R2021b

Simulation and Model-Based Design

Simulink® is a block diagram environment for multidomain simulation and Model-Based Design. It supports system-level design, simulation, automatic code generation, and continuous test and verification of embedded systems. Simulink provides a graphical editor, customizable block libraries, and solvers for modeling and simulating dynamic systems. It is integrated with MATLAB®, enabling you to incorporate MATLAB algorithms into models and export simulation results to MATLAB for further analysis.

[Release Notes](#)

[PDF Documentation](#)



**Get Started**  
Learn the basics of Simulink

**Applications**  
Example models illustrating specific functionality and applications

**Simulink Environment Fundamentals**

Documentation

CONTENTS

Documentation Examples Functions Blocks

« Documentation Home  
« Simulink

## PDF Documentation for Simulink

[Simulink Getting Started Guide](#)

[Simulink User's Guide](#)

[Simulink Graphical User Interface](#)

[Simulink Reference](#)

[Developing S-Functions](#)

[Simulink Release Notes](#)

**Modeling Guidelines**

[Modeling Guidelines for High-Integrity Systems](#)

[Modeling Guidelines for Code Generation](#)

[MathWorks Automotive Advisory Board Control Algorithm Modeling Guidelines Using MATLAB, Simulink, and Stateflow](#)

# EXERCISE 1) GETTING STARTED WITH MATLAB AND SIMULINK

Self-Paced Online Courses

Home | My Courses

Watch an introduction (1 min) » View courses you've started



Browse self-paced online courses

Getting Started (12)

MATLAB (4)

Simulink (5)

AI, Machine Learning, and Deep Learning (5)

Math and Optimization (6)

Image and Signal Processing (3)

Explore over 50 virtual and in-person classroom courses

## MATLAB

**MATLAB Onramp**  
15 modules | 2 hours | Languages  
Get started quickly with the basics of MATLAB.

**MATLAB Fundamentals**  
18 modules | 25 hours | Languages  
Learn core MATLAB functionality for data analysis, modeling, and programming.

**MATLAB for Data Processing and Visualization**  
11 modules | 9 hours | Languages  
Create custom visualizations and automate your data analysis tasks.

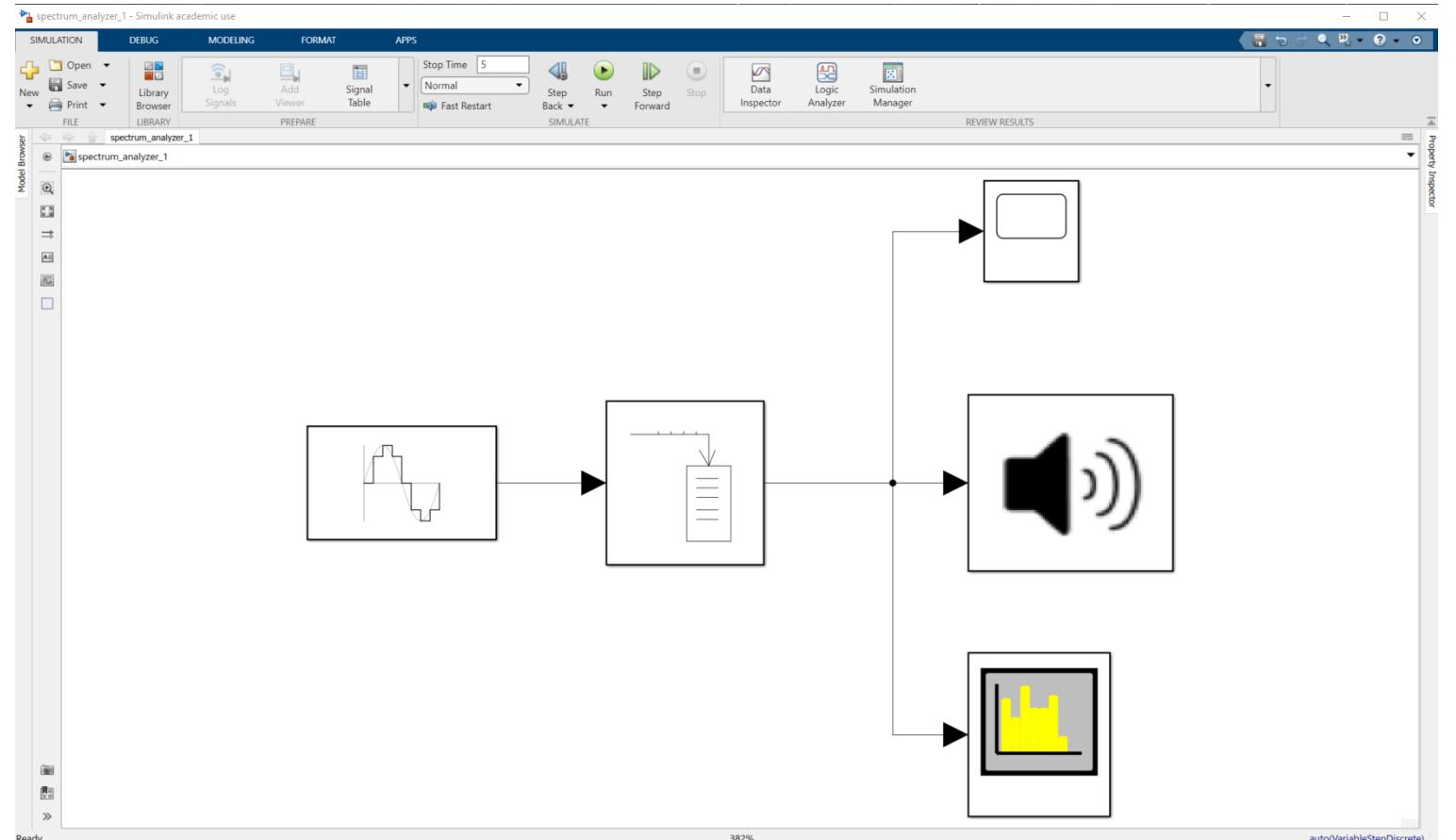
**MATLAB Programming Techniques**  
10 modules | 16 hours | Languages

# Frequency Domain Plots

- Open the model

Spectrum\_analyser\_1.slx

- Analyze the time domain waveform and the spectrum and listen the sound
- Double the frequency and analyze again
- Repeat until you cannot ear the sound
- Take the conclusions

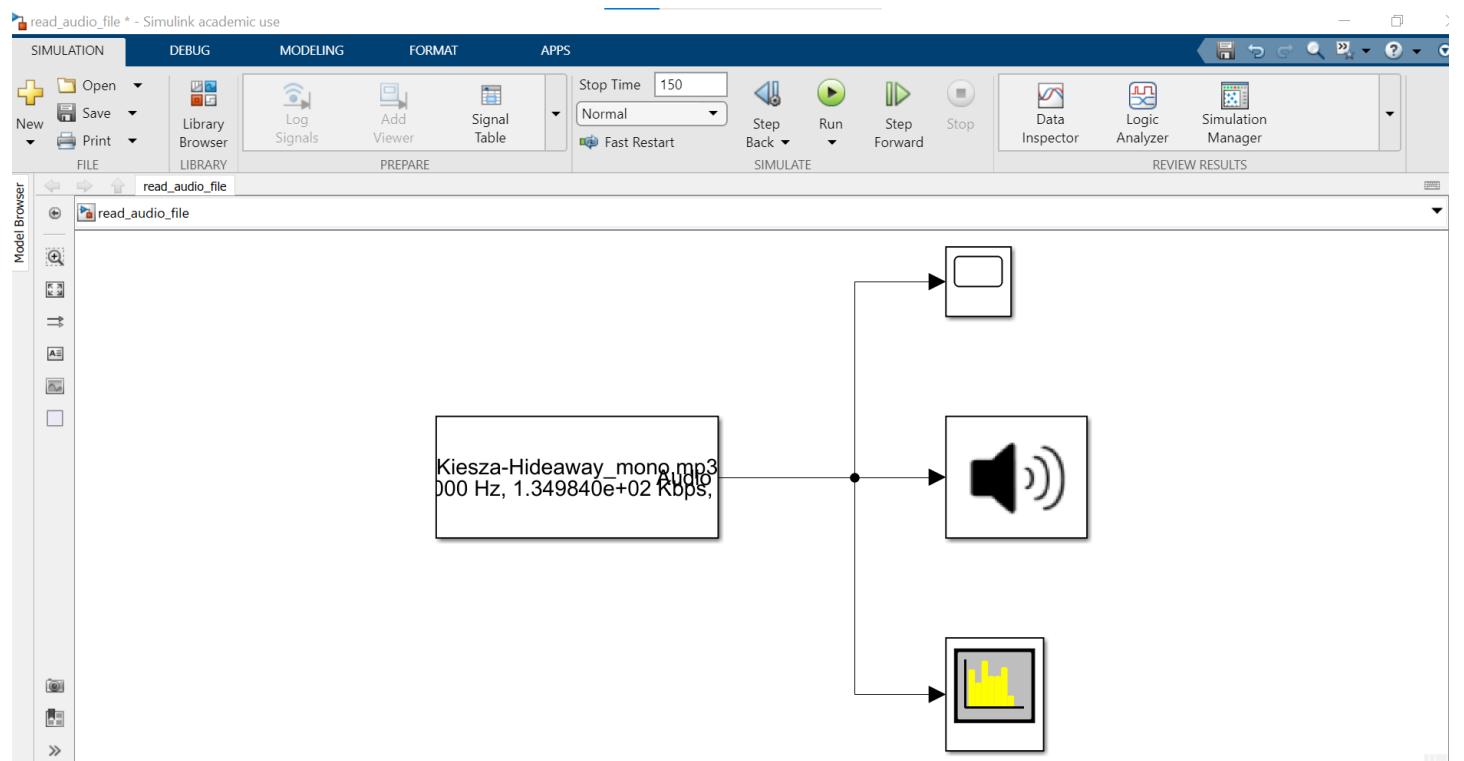


# Frequency Domain Plots

- Open the model:

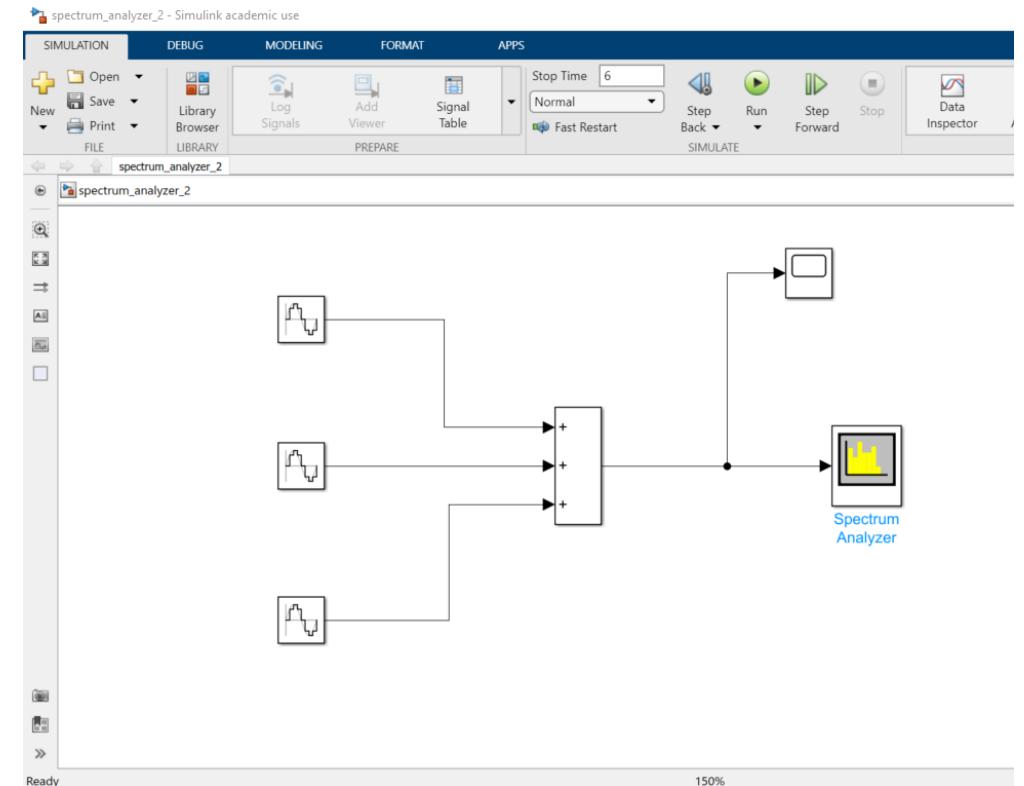
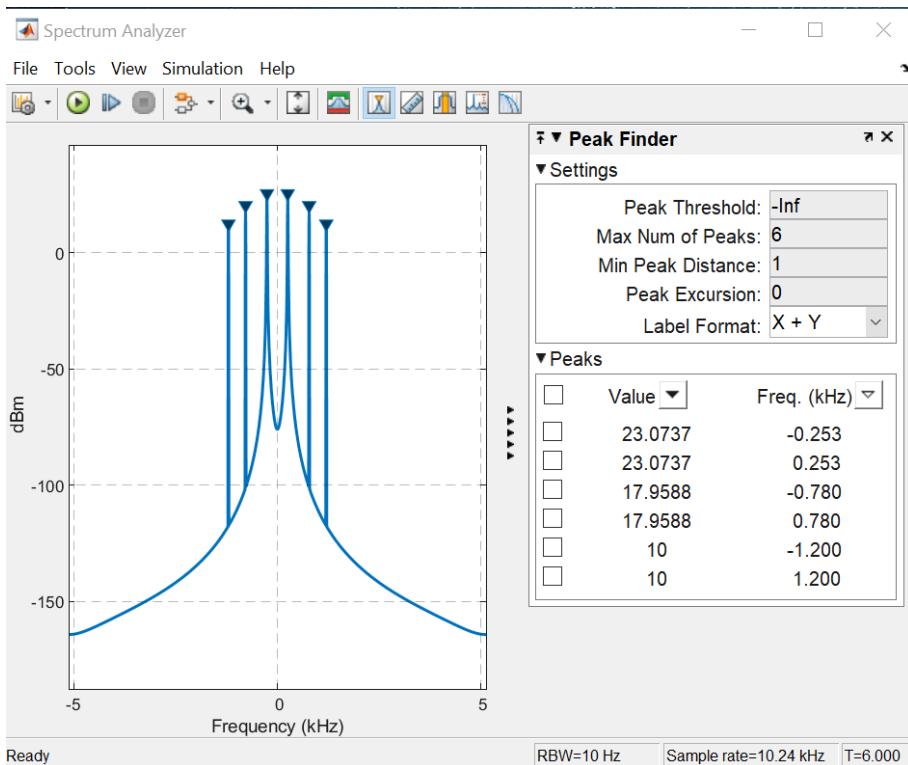
[read\\_audio\\_file.slx](#)

- Analyze the time domain waveform and the spectrum and listen the sound.
- Comment the waveforms

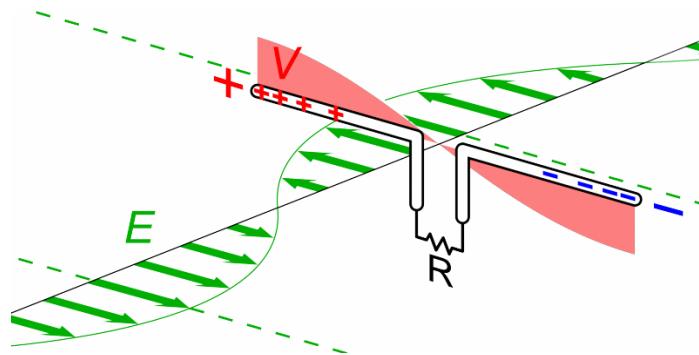


# Frequency Domain Plots

- Open the model: [Spectrum\\_analyser\\_2.slx](#)
- Analyze the signal in time and frequency domain
- In frequency domain, determine the frequency and power of each spectrum line.
- Compare with the theoretical values

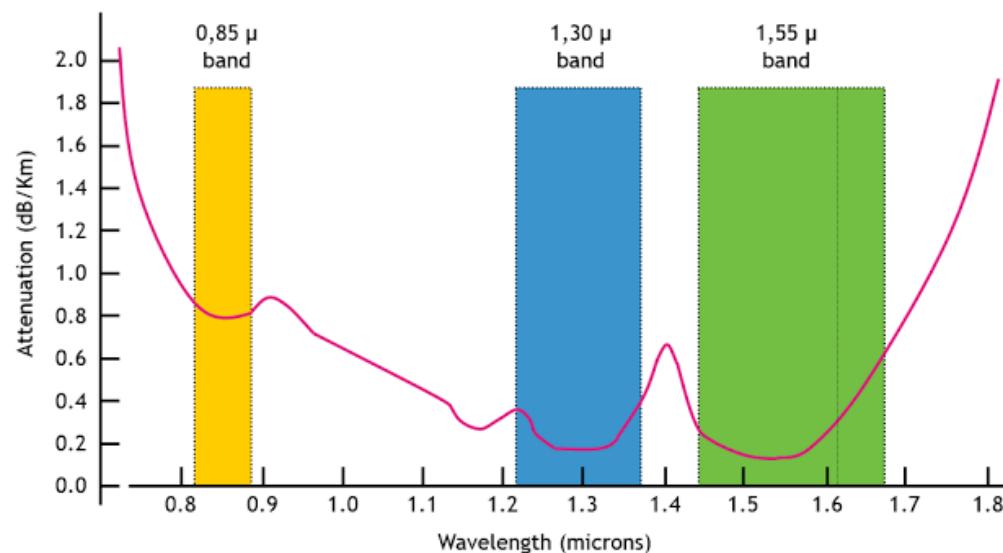
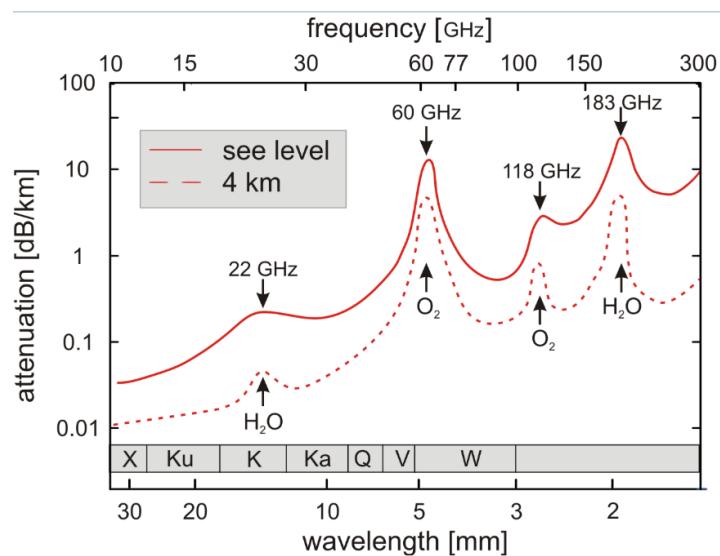


# Modulation

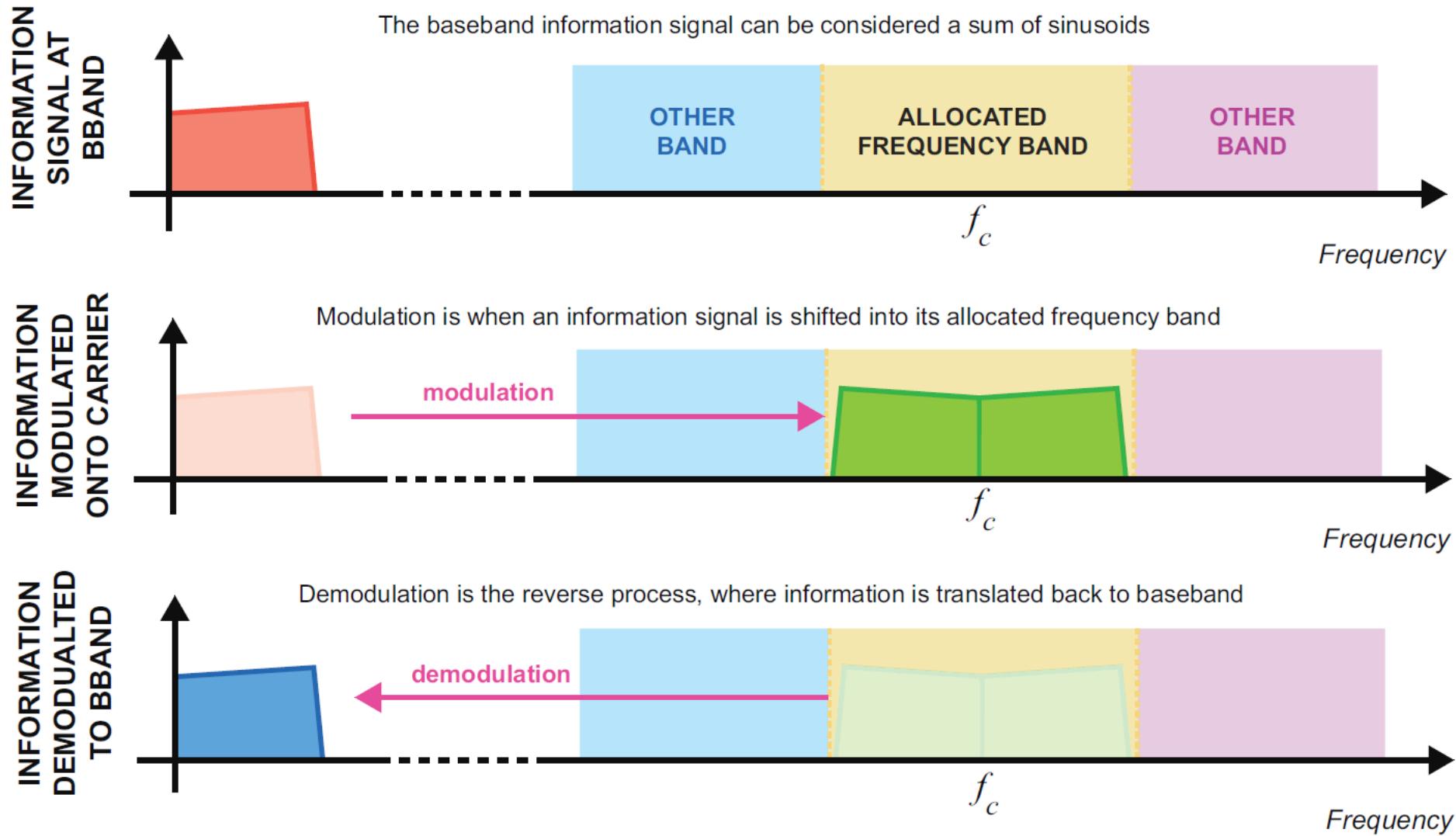


<b>F (Hz)</b>	$\lambda = \frac{c}{f}$
1KHz	300 km!
1MHz	300m
300MHz	1m
3GHz	10cm
30GHz	1cm

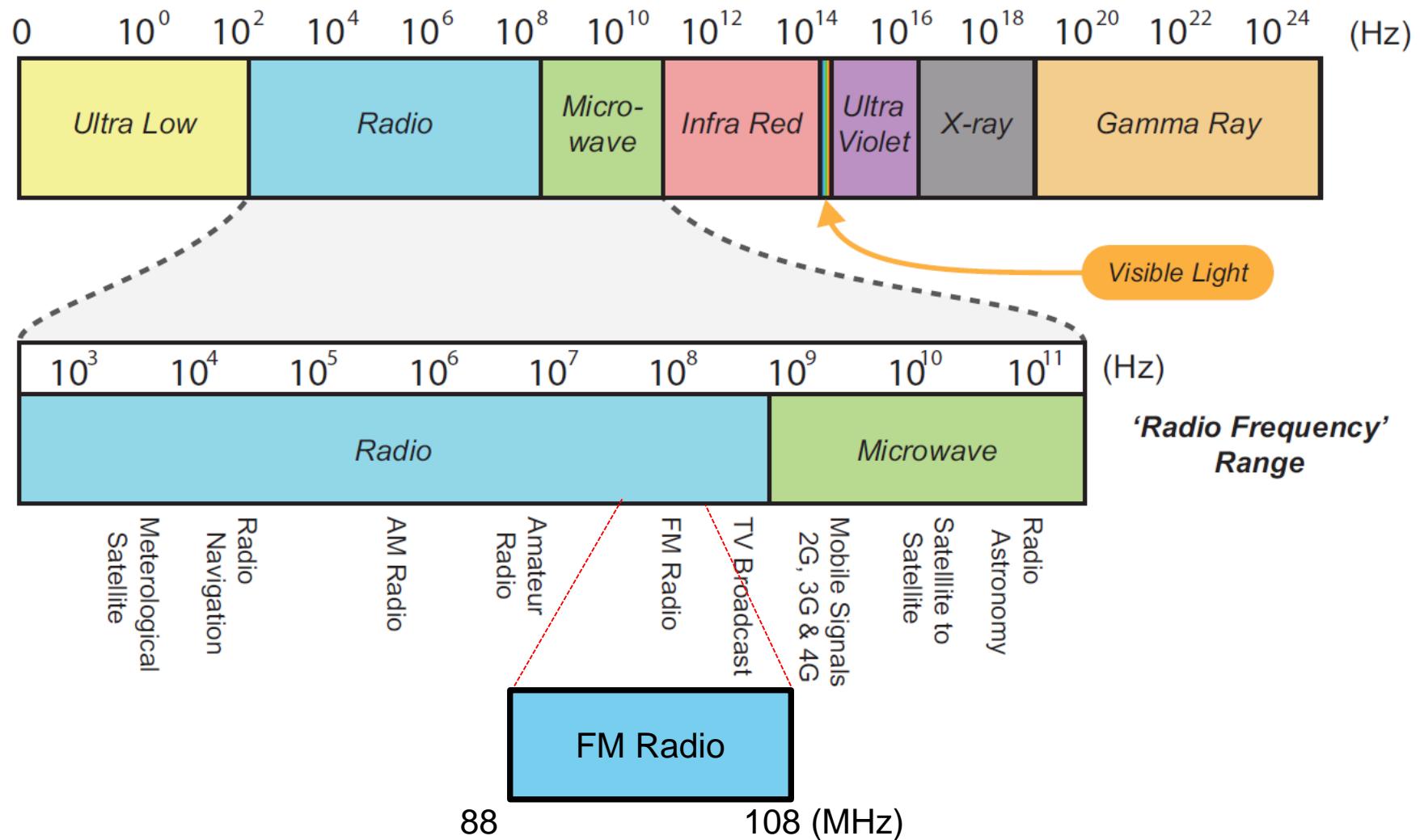
<https://www.electronicsforu.com/resources/learn-electronics/transmission-basics-designing-dipole-antenna>



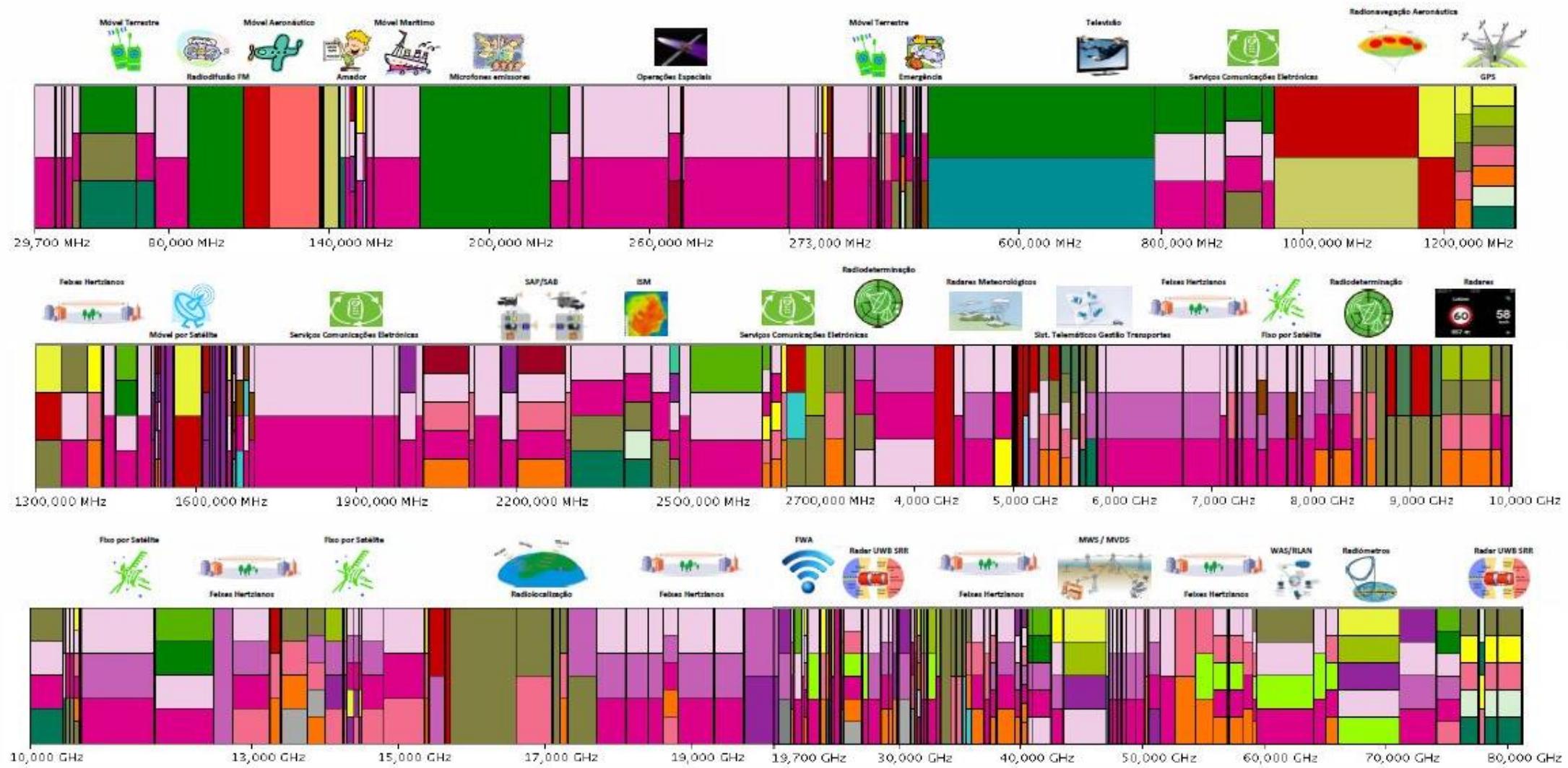
# Modulation



# Electromagnetic Spectrum

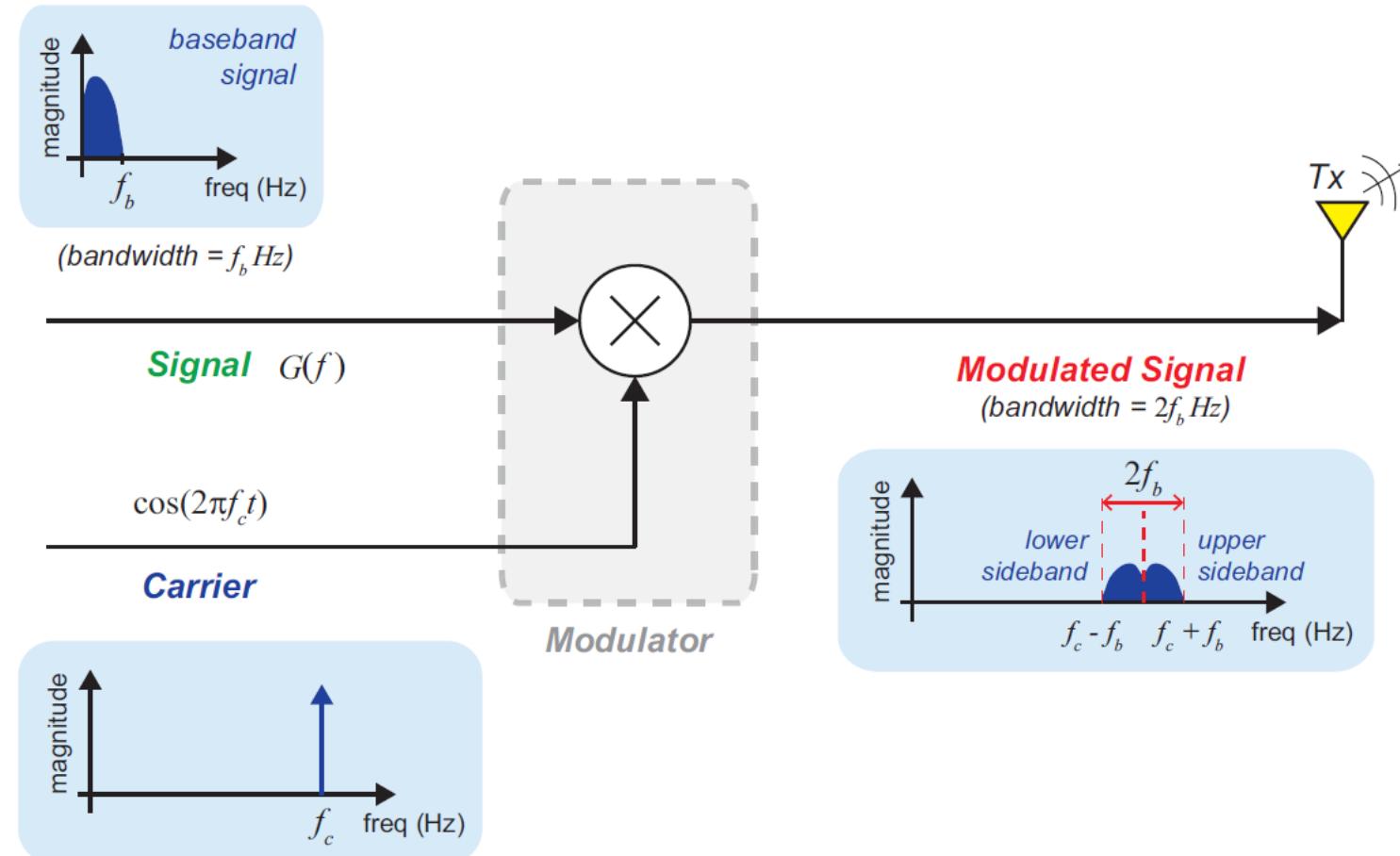


# Electromagnetic Spectrum



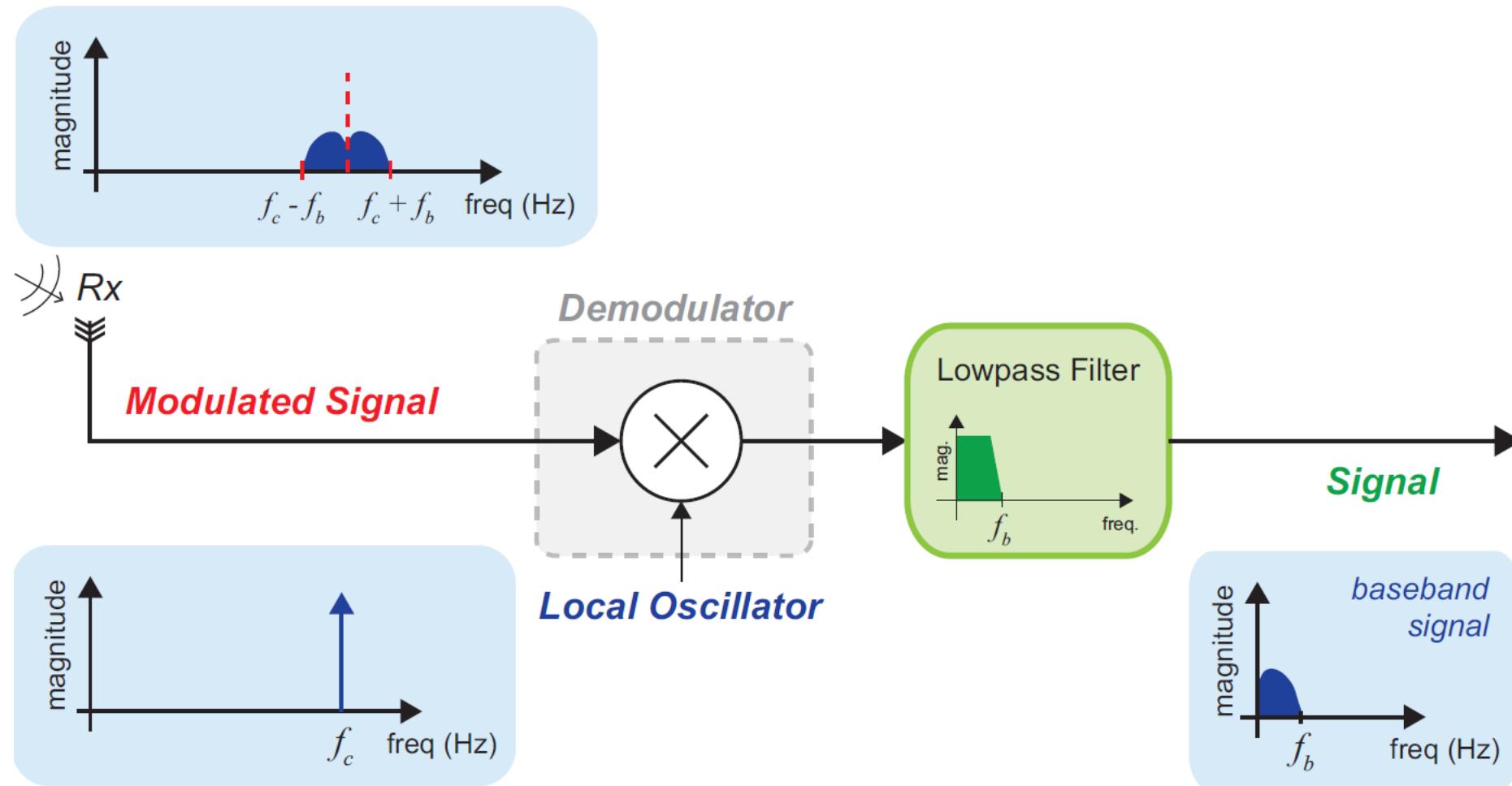
# Amplitude Modulation

Modulating a baseband signal to produce an upper and lower sideband



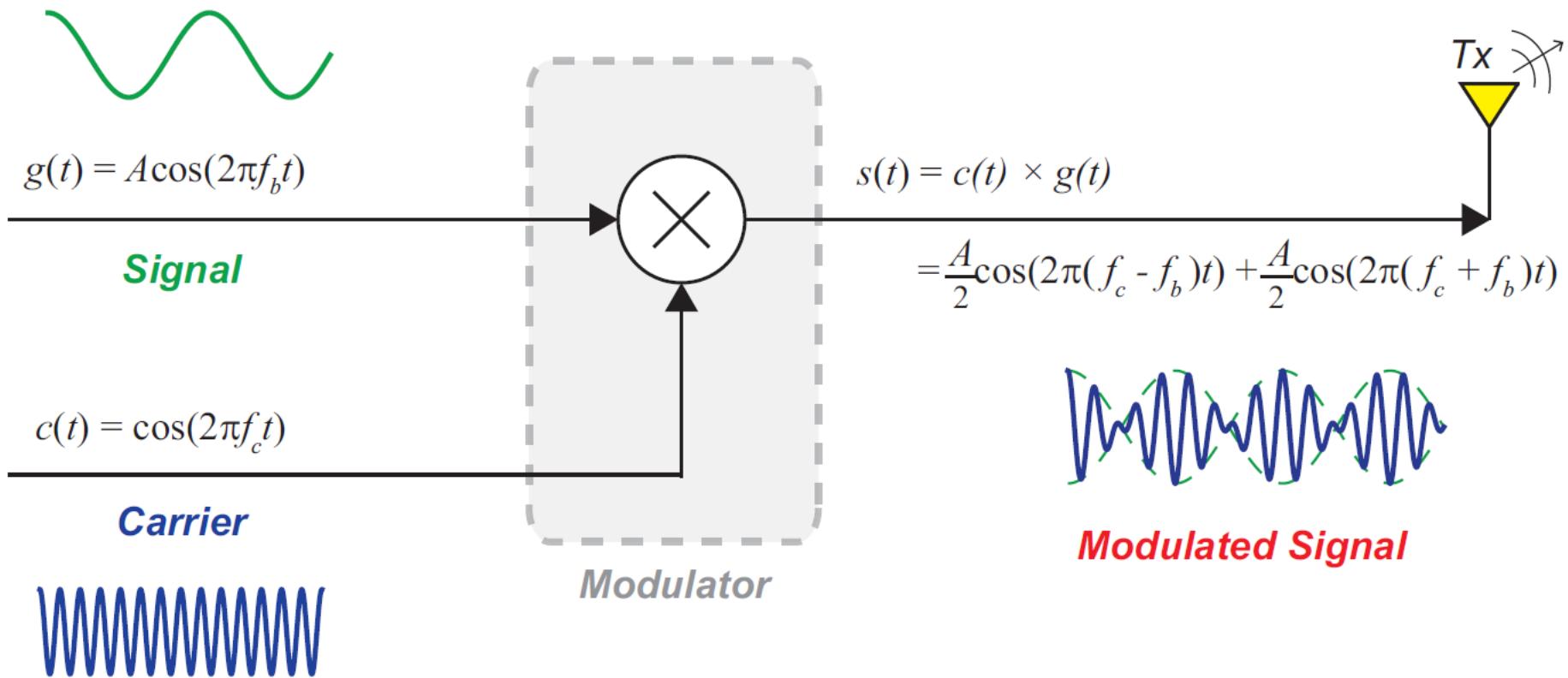
# Amplitude Demodulation

Amplitude demodulation of AM transmitted signal



# Amplitude Modulation

Example: Modulating a baseband signal based on a simple line spectra (cosine wave baseband)

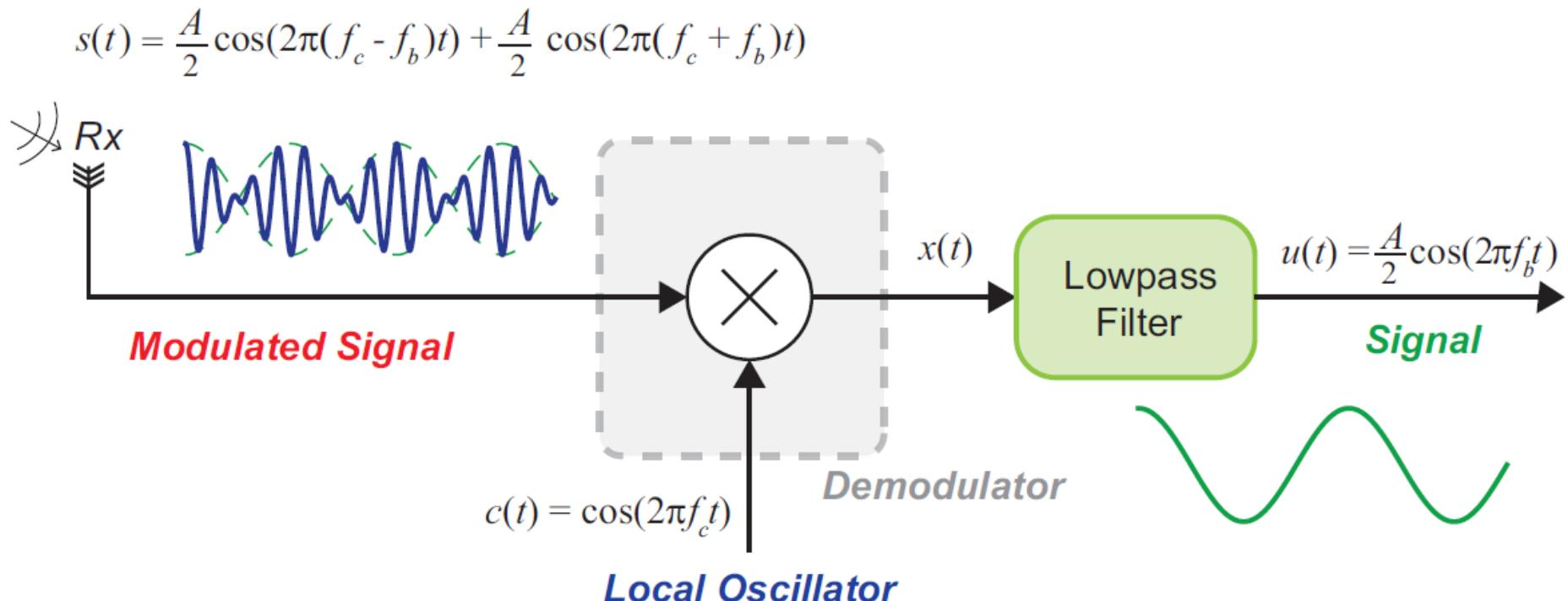


## Exercise

Get the mathematic expression  $s(t)$  from  $g(t)$  and  $c(t)$

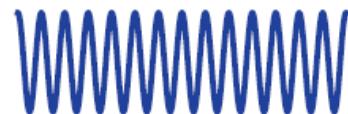
# Amplitude Demodulation

Amplitude demodulation of AM transmitted signal where a baseband signal is a simple line spectra (cosine wave baseband)



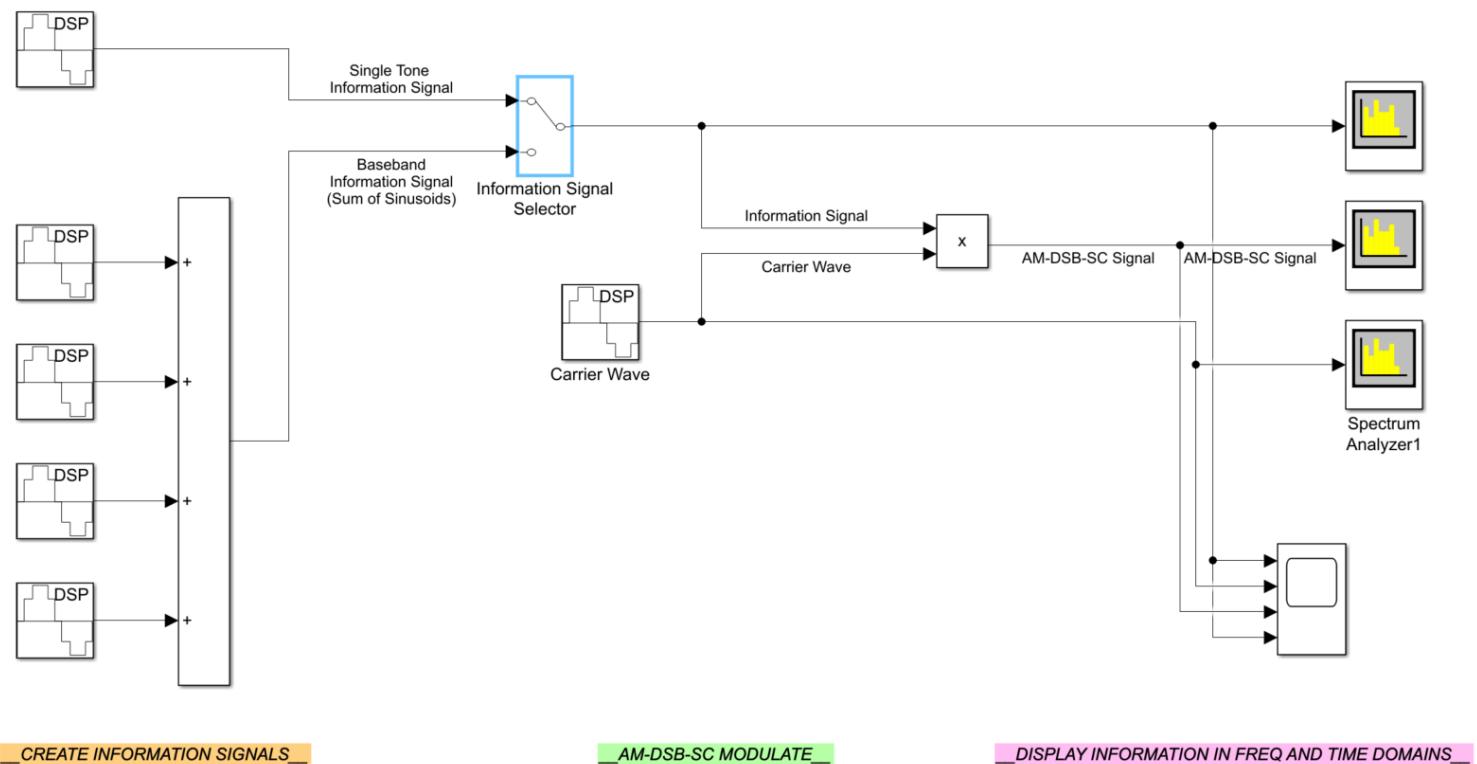
## Exercise

Get the mathematic expression  $u(t)$  from  $s(t)$  and  $x(t)$



# Frequency Domain Plots

- Open the model: [am\\_dsb\\_sc.slx](#)
- With the switch set to “Single Tone analyze the signals at the Time Scopes and Spectrum Analyzers. Comment the results
- Set the Information Signal selector switch to “Baseband Information Signal and re-run the simulation. Compare the results



# References

Robert W. Stewart, Kenneth W. Barlee, Dale S. W. Atkinson, Louise H. Crockett, "Software Defined Radio using MATLAB & Simulink and RTL-SDR", 1st edition, September 2015.

The Role of Space-based Communications in the 5G Era, Intelsat whitepaper

MATLAB® Primer, 2021

Simulink® Getting Started Guide, 2021

Communications Toolbox Support Package for Analog Devices® ADALM-Pluto Radio User's Guide, 2019.

Communications Toolbox Support Package for Analog Devices® ADALM-Pluto Radio Reference, 2019.