

# HƯỚNG DẪN SỬ DỤNG MATLAB ONLINE VÀ GOOGLE COLAB CHO MÔN NHẬP MÔN KỸ THUẬT TRUYỀN THÔNG

## I. Mục tiêu

Để hỗ trợ cho quá trình học môn **Nhập Môn Kỹ thuật Truyền thông**, việc làm quen với thực hành lập trình các phương pháp thiết kế hệ thống thu phát liên quan đến điều chế và giải điều chế tín hiệu đóng vai trò rất quan trọng. Việc trực tiếp triển khai và chạy mã nguồn giúp sinh viên:

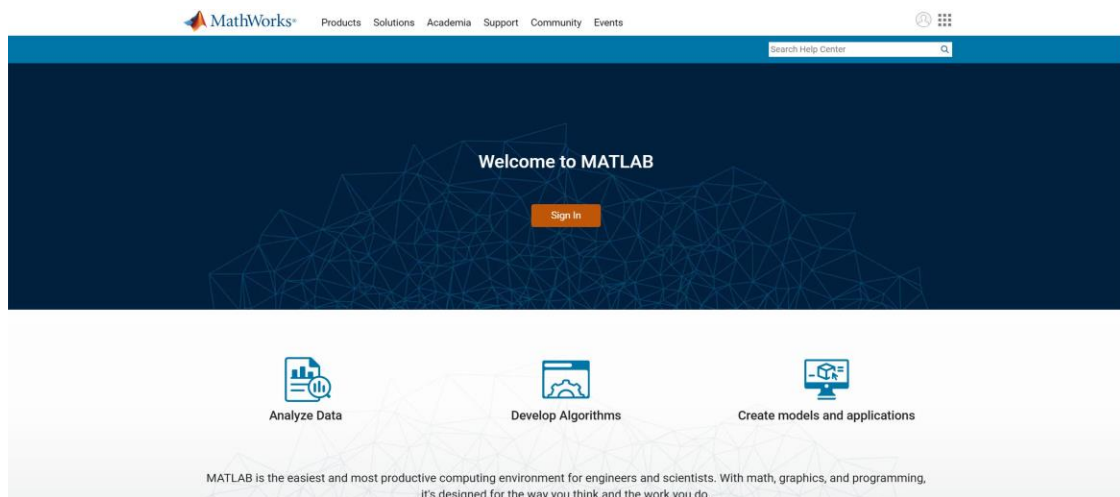
- ✓ Hiểu rõ hơn về bản chất hoạt động của các kỹ thuật truyền thông số.
- ✓ Quan sát trực quan quá trình biến đổi và truyền tải tín hiệu.
- ✓ Củng cố lý thuyết thông qua mô phỏng thực tế.
- ✓ Phát triển tư duy thuật toán và kỹ năng giải quyết vấn đề trong kỹ thuật.

Thực hành lập trình không chỉ là phương tiện hỗ trợ học tập mà còn là cầu nối giữa lý thuyết và ứng dụng thực tiễn trong các lĩnh vực truyền thông.

## II. Chương Trình “Hello World” Trên MATLAB Online

### **Bước 1:** Truy cập MATLAB Online

Vào trang: <https://matlab.mathworks.com> Đăng nhập bằng tài khoản MathWorks cá nhân.



**Bước 2:** Chọn phiên bản MATLAB Online: Sau khi đăng nhập, chọn “Use MATLAB Online (Basic)” để bắt đầu.

**Bước 3: Tạo file script mới:**Trên giao diện chính, nhấn **New** → **Script** hoặc biểu tượng dấu + ở góc trái.

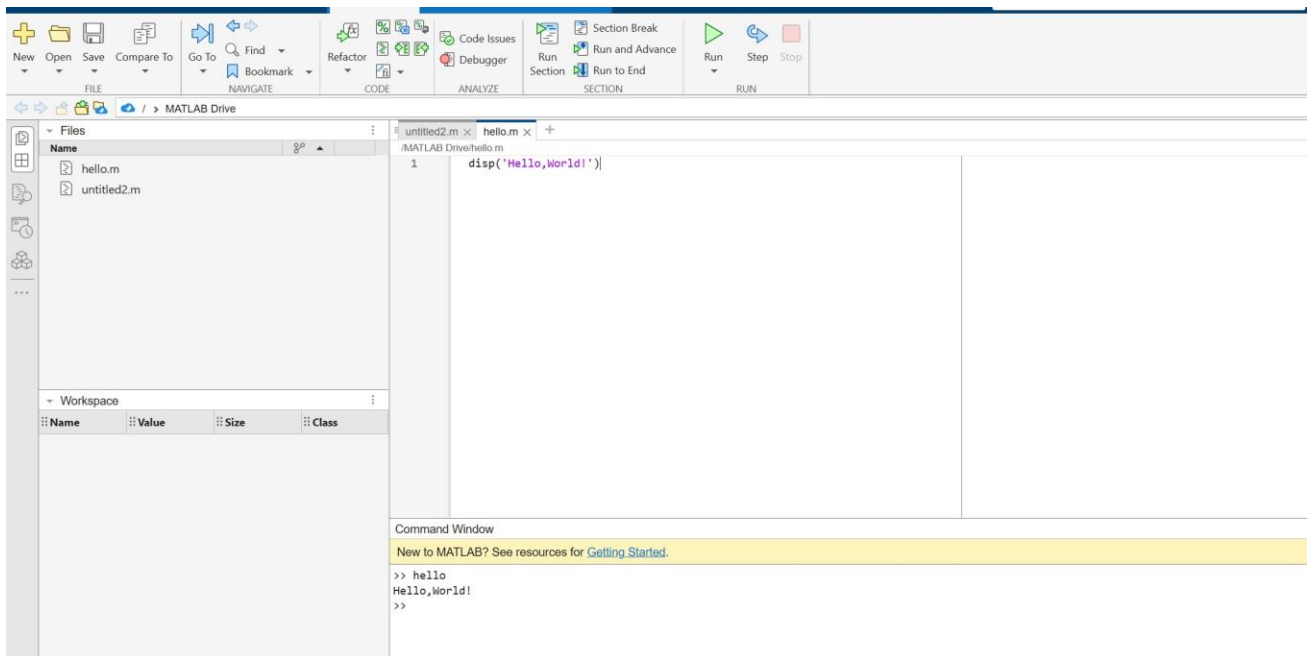
**Bước 4: Nhập nội dung chương trình**

```
% Hello World trong MATLAB Online  
disp('Hello,World!')
```

**Bước 5: Lưu và chạy chương trình:**Nhấn biểu tượng **Save** () để đặt tên file, ví dụ hello.m;  
Nhấn nút **Run** hoặc phím F5 để chạy.

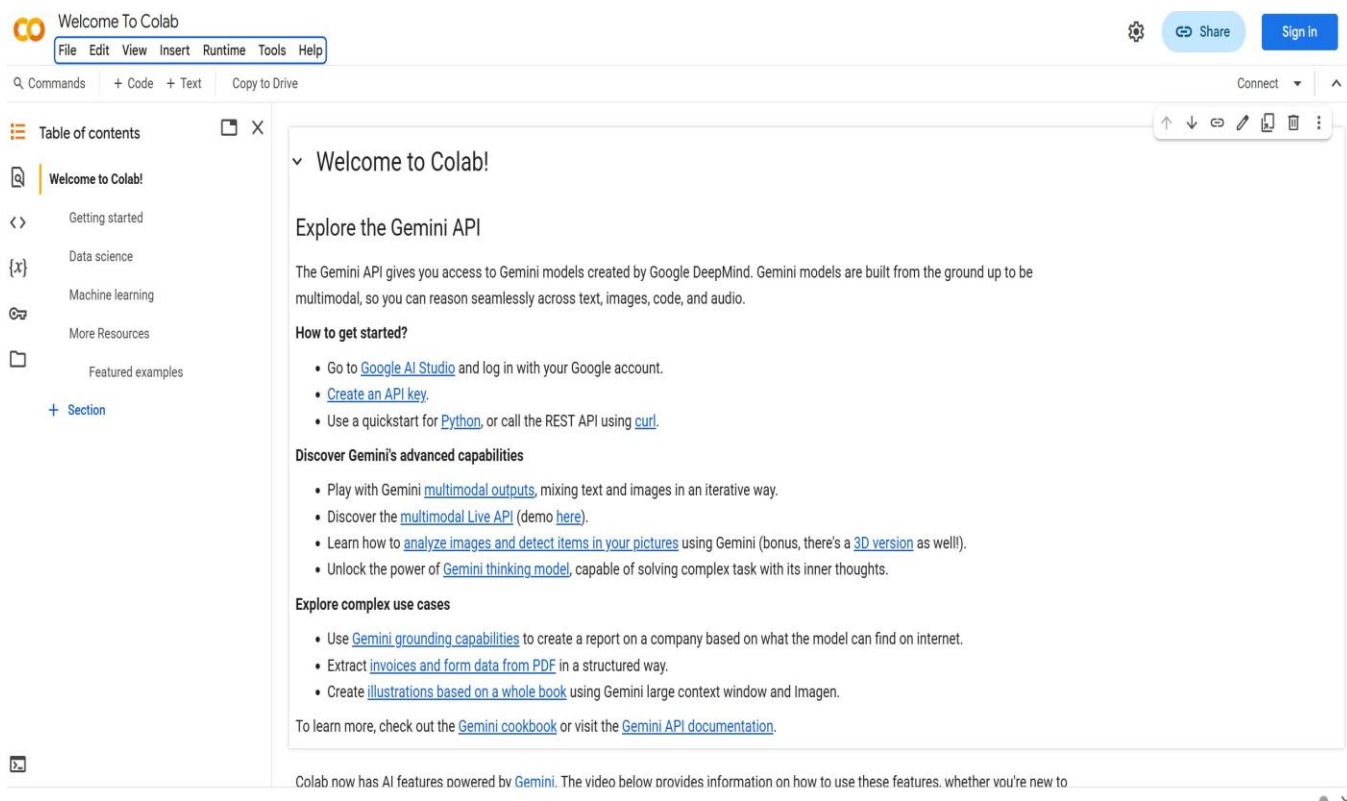
**Kết quả mong đợi trong Command Window:**

```
Hello,World!
```

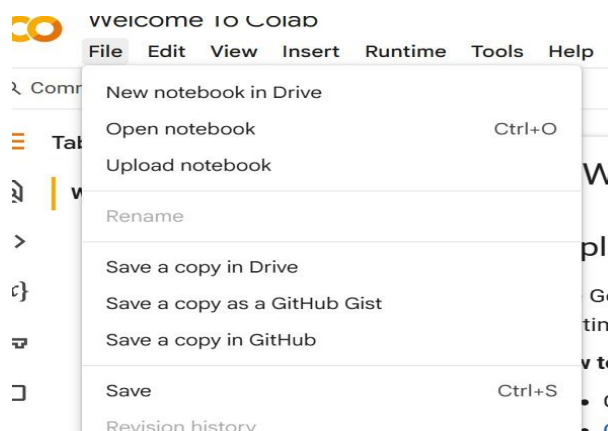


### III. Chương Trình “Hello World” Trên Google Colab (Python)

**Bước 1:** Truy cập Google Colab: <https://colab.research.google.com>



**Bước 2: Tạo notebook mới: Trên thanh menu, chọn File → New notebook**



**Bước 3: Nhập nội dung chương trình vào ô code**

```
# Hello World trong Python
print("Hello,World!")
```

**Bước 4: Chạy chương trình:** Nhấn nút ở bên trái ô hoặc dùng phím tắt **Shift + Enter**.  
Kết quả mong đợi



## IV. Ứng dụng Google Colab vào phân tích và xử lý tín hiệu – Cần những gì?

### Các thư viện hữu ích khi xử lý tín hiệu trên Google Colab

1. **NumPy (import numpy as np):** Lý do cần: Hỗ trợ các phép toán đại số tuyến tính, xử lý mảng, số ngẫu nhiên – tương tự như MATLAB; Dùng để: Tạo tín hiệu, thao tác tín hiệu (dịch, cộng, nhân, FFT, conv...), tạo vector thời gian.
2. **Matplotlib (import matplotlib.pyplot as plt):** Lý do cần: Thư viện trực quan hóa dữ liệu – giúp bạn vẽ biểu đồ giống plot() trong MATLAB; Dùng để: Vẽ tín hiệu theo thời gian, vẽ phổ tín hiệu, biểu đồ histogram, eye diagram, v.v.
3. **SciPy (from scipy import signal):** Lý do cần: Cung cấp các công cụ mô phỏng hệ thống rời rạc, lọc, biến đổi tín hiệu; Dùng để: Thiết kế và áp dụng các bộ lọc FIR/IIR, tính tích chập (convolution), xử lý tần số, hàm tương quan...

### Chú ý khi lập trình trên Google Colab

Google Colab dựa trên Jupyter Notebook, nên các đoạn mã được chia thành các cell (ô lệnh). Có 2 loại chính:

1. **Code Cell:** Mục đích:Viết và chạy mã Python. Mỗi ô độc lập.  
Có thể chạy lại từng cell mà không cần chạy lại toàn notebook.
2. **Text (Markdown) Cell:** Mục đích: Ghi chú, tiêu đề, công thức .

### Ví dụ đơn giản: Tín hiệu hình sin cơ bản

#### Cell 1: Import thư viện cần thiết

```
import numpy as np
import matplotlib.pyplot as plt
```

### Cell 2: Tạo trục thời gian và tín hiệu sin

```
fs = 1000 # Tần số lấy mẫu (Hz) t = np.arange(0, 1, 1/fs) # Trục thời gian từ 0 đến 1 giây với bước lấy mẫu 1/fs  
f = 5 # Tần số tín hiệu sin (Hz) x = np.sin(2 * np.pi * f * t) # Tạo tín hiệu sin
```

### Cell 3: Hiển thị tín hiệu bằng biểu đồ

```
plt.figure(figsize=(8, 3))  
plt.plot(t, x)  
plt.title('Tín hiệu hình sin cơ bản (f = 5 Hz)')  
plt.xlabel('Thời gian (s)')  
plt.ylabel('Biên độ')  
plt.grid(True)  
plt.tight_layout()  
plt.show()
```

Đầu ra:

