

Sinais e Sistemas

Indicações para a instalação do sistema Python

25 de Fevereiro de 2015

1 Introdução

Os trabalhos de laboratório de Sinais e Sistemas são realizados com base na linguagem de programação Python. Este documento fornece indicações para os alunos que queiram instalar o Python nos seus próprios computadores, de modo a poderem treinar antecipadamente a realização das questões de carácter laboratorial. Faz-se notar que os trabalhos são dimensionados de forma a que os alunos tenham tempo para realizar toda a parte laboratorial numa aula de laboratório, desde que se tenham preparado adequadamente. Não é, portanto, indispensável que treinem antecipadamente a realização dos trabalhos, embora se compreenda que possam querer fazê-lo.

Neste documento dão-se dois conjuntos de indicações distintos. Se tiver o sistema operativo Windows e quiser criar uma configuração muito semelhante à existente no laboratório, siga as indicações dadas na Secção 2 (esta é a forma de instalação mais simples). Se não tiver o sistema operativo Windows ou se, tendo-o, pretender usar uma configuração diferente da usada no laboratório, siga as indicações dadas na Secção 3.

Se tiver sugestões para a sua melhoria deste documento, agradece-se que as envie, por *e-mail*, para o responsável da disciplina.

2 Instalação semelhante à do laboratório

Os trabalhos de laboratório são realizados usando a distribuição WinPython, versão 3.3.3.3. É disponibilizada aos alunos uma versão do WinPython já configurada da forma adequada à realização dos trabalhos.

O WinPython é portátil, pelo que a instalação se resume quase só a descarregar um ficheiro comprimido e a descomprimi-lo num local adequado. O WinPython ocupa cerca de 1,2 GB de espaço em disco.

Comece por descarregar da página da disciplina o ficheiro `LabSS-XXbit.7z` (em que `XX` é 32 ou 64), adequado à sua versão do Windows (de 32 ou de 64 bits; se tiver dúvidas sobre qual ficheiro deverá usar, use o de 32 bits). Em seguida descomprima o ficheiro obtido, por meio do utilitário 7-Zip, disponível em <http://www.7-zip.org/>. Faça a descompressão para um local que considere apropriado (por exemplo para o seu Desktop). Nesse local ficará uma pasta denominada `LabSS`.

No interior da pasta `LabSS` existem duas pastas: `WinPython-XXbit-3.3.3.3` e `Lab`. A primeira contém o WinPython, já configurado da forma adequada à realização dos trabalhos de laboratório. A segunda está vazia, e destina-se a colocar os ficheiros que são fornecidos para a realização de cada trabalho.

Abra a pasta `WinPython-XXbit-3.3.3.3`. Dos vários ficheiros que estão no seu interior, irá usar o `Spyder` ou `Spyder.exe`, que é um ambiente de desenvolvimento para Python. Dê duplo clique nesse ficheiro (não confunda com o `Spyder (light)` ou `Spyder (light).exe`). Ao fim de algum tempo abrir-se-á uma janela com o título “Spyder (Python 3.3)”. Ao fim de mais algum tempo aparecerão no interior dessa janela várias linhas de texto, terminando com uma linha que contém “`In [1]:`”. Esse texto é produzido pela consola

IPython, que é uma consola interactiva de Python que está a ser executada dentro do Spyder. É na consola IPython que irá realizar o seu trabalho.

Dê, na consola IPython, o comando `pwd`, que informa de qual a pasta em que está a trabalhar. Verifique que está na pasta `Lab`, referida acima. Note que, na informação que é fornecida pelo comando `pwd`, as barras (“\”) aparecem repetidas.

Se conseguiu realizar com sucesso os passos indicados acima, o sistema está pronto para realizar os trabalhos de laboratório. Para realizar cada trabalho, coloque na pasta `Lab` os ficheiros que são fornecidos para esse trabalho, e em seguida realize o trabalho como indicado no respectivo enunciado (por exemplo, para o primeiro trabalho, comece por dar o comando `run -i lab1`).

Se não conseguiu realizar com sucesso os passos indicados acima, o WinPython não estará, provavelmente, configurado da forma apropriada para realizar os trabalhos de laboratório. Poderá tentar configurá-lo manualmente, seguindo as indicações dadas na Secção 3.

3 Instalação genérica

Os trabalhos de laboratório são realizados utilizando o Python de forma interactiva, através da consola IPython. O objectivo que se pretende atingir com as indicações dadas a seguir é dispor de uma consola IPython adequadamente configurada. Mais concretamente, pretende-se dispor duma consola IPython com o módulo *pylab* correctamente importado, e com o sistema de gráficos configurado de modo a que os gráficos sejam traçados em janelas separadas e a que seja possível fazer zoom deles e deslocá-los.

Existem diferentes sistemas operativos para computadores, e para cada um dos sistemas operativos mais comuns existem diversas distribuições de Python. Por isso as indicações que se dão a seguir são algo genéricas, devendo os alunos adaptá-las a cada caso concreto.

3.1 Escolha da distribuição de Python

Escolha uma distribuição de Python adequada ao sistema operativo do seu computador. Existem actualmente duas versões da linguagem Python em utilização generalizada: a 2 e a 3. Estas duas versões não são compatíveis entre si. Deverá ser escolhida uma distribuição da versão 3. Estas distribuições têm, geralmente, um número de versão da forma 3.xxx (por exemplo 3.1.2.4).

A distribuição de Python a escolher deverá incluir a consola IPython e os módulos *numpy*, *scipy*, *matplotlib* e *pylab*. No sistema operativo Windows, a distribuição deverá ainda incluir o módulo *winsound*, que permite usar o comando `play`, que é usado nalguns trabalhos de laboratório. Nos outros sistemas operativos este módulo não está disponível, e terá de verificar se o comando `play` funciona.

Instale a distribuição que escolheu.

Note que, instalando uma versão do Python diferente da que é utilizada no laboratório, é provável que não consiga correr alguns ficheiros pré-compilados, como o `systema2` do 1º trabalho e o `systema1` do 2º trabalho. Isto tem a ver com o facto de os ficheiros pré-compilados serem específicos da versão e sub-versão do Python usada para os compilar, e poderem não correr noutras versões. O número completo da versão de Python que foi usada para compilar esses ficheiros é 3.3.3, que é a versão que vem incluída no WinPython 3.3.3.3.

3.2 Arranque da consola IPython

Uma vez instalada uma distribuição de Python, será necessário arrancar a consola IPython. Na maioria das distribuições, a consola IPython pode ser arrancada dando, numa linha de comandos (ou numa *shell*) o comando `ipython3` ou `ipython` (pode ser necessário, primeiro, procurar qual a pasta em que está o ficheiro executável do `ipython3` ou do `ipython`). Quando arrancada correctamente, a consola escreve algum texto

introdutório, terminando com uma linha que contém “In [1]:”.

Várias das distribuições de Python incluem um ambiente de desenvolvimento. Nesse caso poderá preferir usá-lo, e arrancar a consola IPython através dos menus desse sistema, em vez de o fazer através da linha de comandos.

A consola IPython é interactiva, isto é, executa imediatamente os comandos que lhe são dados. Uma vez arrancada essa consola, verifique se está a funcionar correctamente. Para isso dê, na linha que contém “In [1]:”, o comando `2+3`. Deverá obter a resposta “Out[1]: 5”.

Para verificar se está a utilizar a versão 3 do Python, comece por dar o comando `print 2+3`. Deverá obter uma mensagem de erro, porque este comando está no formato do Python 2. Em seguida dê o comando `print(2+3)`, que está no formato do Python 3. Deverá obter o resultado “5”.

Note que, além da consola IPython, existem outras consolas interactivas de Python, e nomeadamente existe uma chamada simplesmente Python Console. Deve assegurar-se de que a consola que está a utilizar é a IPython, e não outra. Não deverá também utilizar um “IPython Notebook” (que abre no *browser*) nem uma “IPython QT Console”. Estas consolas não permitem obter os gráficos em janelas separadas, o que é necessário para a realização dos trabalhos.

3.3 Importação do módulo *pylab*

Para utilizar a consola IPython na realização dos trabalhos de laboratório, será necessário importar, da forma correcta, o módulo *pylab*. Ao importar este módulo são automaticamente importados os módulos *numpy*, destinado ao cálculo numérico, e *matplotlib*, destinado ao traçado de gráficos.

Verifique se a consola IPython tem já o módulo *pylab* correctamente importado (isso acontece nalguns sistemas). Para isso dê, nessa consola, os comandos

```
ion()
x = arange(0,20*pi,0.1)
plot(x,sin(x))
```

Se tiver obtido o gráfico duma sinusóide (o gráfico poderá ter surgido numa janela que esteja “atrás” de outras janelas que estejam abertas), o módulo *pylab* está, provavelmente, correctamente importado. Nesse caso pode passar à Secção 3.4. Se não obteve o gráfico indicado (provavelmente terá recebido algumas mensagens de erro), o módulo *pylab* não está correctamente importado. Nesse caso realize os passos indicados a seguir, para importar esse módulo.

Se arrancou a consola IPython a partir da linha de comandos, experimente fechá-la e arrancá-la agora com o comando `ipython3 --pylab` (ou `ipython --pylab`). Verifique, da forma indicada acima, se o módulo *pylab* está correctamente importado. Se sim, no futuro passe a arrancar a consola IPython da forma indicada.

Se não conseguiu ter o módulo *pylab* correctamente importado de nenhuma das formas indicadas acima, dê, dentro da consola IPython, o comando `%pylab`. Em seguida verifique, como indicado acima, se o módulo ficou correctamente importado. Se assim for, no futuro deverá dar este comando sempre que arrancar a consola IPython.

3.4 Configuração do sistema gráfico

Comece por fechar a consola IPython que tinha aberta, e arranque-a novamente, da forma que determinou ser necessária na Secção 3.3.

O módulo gráfico, *matplotlib*, pode ser utilizado em dois modos: interactivo, ou não interactivo. Pretende-se usá-lo no modo interactivo. Verifique em que modo está, dando o comando `isinteractive()`. Se obtiver a resposta “True”, passe ao parágrafo seguinte. Se obtiver a resposta “False”, active o modo interactivo dando o comando `ion()`. Deverá utilizar sempre o modo interactivo na realização dos trabalhos de laboratório. Caso contrário, ao dar comandos de traçado de gráficos, os gráficos não ficarão imediatamente visíveis.

Pretende-se que o sistema trace cada gráfico numa nova janela, e que seja possível fazer zoom do gráfico e deslocá-lo. Para verificar se isso sucede, dê novamente os comandos

```
x = arange(0,20*pi,0.1)
plot(x,sin(x))
```

Deverá ter obtido novamente o gráfico duma sinusóide. Pretende-se que o gráfico tenha aparecido numa nova janela (note que essa janela poderá ter aparecido “atrás” de outras janelas que já estivessem abertas). Se o gráfico apareceu na própria janela da consola IPython (a seguir ao comando `plot` que introduziu), tente mudar o *backend* que é utilizado para desenhar gráficos. Para isso, proceda como indicado na Secção 3.5. Se o gráfico apareceu numa nova janela, realize os passos indicados a seguir, para verificar se pode fazer zoom do gráfico e deslocá-lo.

Na janela em que o gráfico apareceu, clique na cruz com setas nas quatro pontas. O cursor do rato deverá mudar para uma cruz com setas nas quatro pontas. Em seguida:

- Para fazer zoom, coloque o cursor sobre um ponto do gráfico, pressione o botão direito do rato, e desloque o rato horizontalmente e/ou verticalmente, mantendo o botão direito pressionado. Note que nalguns sistemas o zoom só começa a funcionar à segunda tentativa.
- Para deslocar o gráfico, coloque o cursor sobre um ponto do gráfico, pressione o botão esquerdo do rato, e desloque o rato mantendo o botão pressionado.
- Para que o gráfico volte à forma original, clique no símbolo com a forma duma casa.

Se não conseguiu fazer zoom do gráfico e/ou deslocá-lo, tente mudar o *backend* que é utilizado para desenhar gráficos. Para isso proceda como se indica na secção 3.5.

Se conseguiu fazer zoom do gráfico e deslocá-lo, a configuração da consola IPython está concluída. Quando tornar a utilizá-la deverá usar a mesma configuração. Prossiga para a Secção 3.6.

3.5 Mudança de *backend*

Para a realização dos trabalhos de laboratório, é necessário que os gráficos sejam desenhados em janelas separadas, e que seja possível fazer zoom deles e deslocá-los. Se isso não acontece no seu sistema, tente mudar o *backend* que o módulo *matplotlib* usa para traçar os gráficos. Dois *backends* que serão, provavelmente, adequados são o “TkAgg” e o “Qt4Agg”. Nalguns sistemas, o primeiro é significativamente mais rápido que o segundo.

Para mudar o *backend* que é utilizado, proceda da seguinte forma:

- Se está a usar a consola IPython dentro dum ambiente de desenvolvimento, verifique se acedendo, através dos menus desse sistema, às “Options”, “Preferences”, ou semelhante, pode mudar o *backend* que a consola IPython utiliza. Se puder, mude para outro *backend*, reinicie a consola IPython e verifique se os gráficos já são traçados da forma desejada. Note que, nalguns sistemas, os *backends* “TkAgg” e “Qt4Agg” são designados por nomes diferentes, como por exemplo “Tkinter” e “Qt”.
- Se está a usar a consola IPython dentro dum ambiente de desenvolvimento e não consegue mudar o *backend* através dos menus, experimente não usar o sistema de desenvolvimento, e em vez disso arranque a consola IPython directamente a partir da linha de comandos, como indicado na Secção 3.2. Quando se arranca a consola desta forma, normalmente os gráficos ficam a aparecer em janelas separadas. Se passou agora a arrancar a consola IPython através da linha de comandos, repita os passos indicados nas Secções 3.3 e 3.4.
- Se, inicialmente, já tinha arrancado a consola IPython através da linha de comandos, ou se está a usar um sistema de desenvolvimento mas não consegue mudar o *backend* através dos menus, o procedimento a utilizar é mais complexo. Tenha em conta as seguintes indicações, que são algo “técnicas”:

- A forma correcta de fazer a mudança do *backend* dentro da consola IPython consiste em dar o comando `matplotlib.use('backend')`, em que, onde está “`backend`”, deverá figurar o nome do *backend* que se pretende utilizar – por exemplo `matplotlib.use('TkAgg')`. Este comando só pode ser dado antes de ser usado qualquer comando de traçado de gráficos, o que torna a sua utilização pouco prática para experimentar os vários *backends*, porque implica reiniciar a consola IPython a cada nova tentativa.
- Uma alternativa consiste em usar o comando `switch_backend('backend')`, o qual pode ser dado depois de comandos de traçado de gráficos. Tenha em conta, no entanto, que esta forma de utilizar este comando é algo “ilegal”, pelo que pode dar origem a erros.
- Depois de determinar, em definitivo, qual o *backend* que pretende utilizar, deverá passar a utilizar o comando `matplotlib.use('backend')` antes do primeiro comando de traçado de gráficos.
- Para obter uma lista dos *backends* que estão disponíveis, será necessário proceder duma forma um pouco peculiar. Dê o comando `switch_backend('a')`. Como não existe nenhum *backend* chamado “a”, receberá uma mensagem de erro. No final dessa mensagem deverá figurar uma lista dos *backends* que estão disponíveis. Note que, normalmente, muitos desses *backends* não são apropriados para traçar gráficos no ecrã, e/ou exigem a instalação de módulos adicionais.

3.6 Realização dos trabalhos de laboratório

Esta secção dá indicações de como iniciar a realização dos trabalhos de laboratório, depois de ter instalado o Python e de ter configurado a consola IPython.

Comece por criar, num local que considere apropriado, uma pasta para conter os ficheiros relativos ao trabalho que quer realizar. Neste texto designaremos essa pasta por `Lab`. Em seguida coloque nessa pasta os ficheiros fornecidos para a realização desse trabalho.

Arranque a consola IPython e configure-a da forma que determinou, anteriormente, ser adequada. Dê o comando `pwd` para verificar em que pasta está a trabalhar. Note que, na informação que é fornecida pelo comando `pwd`, as barras (“\”) aparecem repetidas. Dê o comando `cd` apropriado para mudar para a pasta `Lab` (por exemplo `cd C:\Users\MyName\Desktop\Lab` em Windows, ou `cd ~/Lab` em Linux).

A partir deste ponto deverá poder realizar o trabalho tal como indicado no respectivo enunciado (por exemplo, para o primeiro trabalho, comece por dar o comando `run -i lab1`).