**                                          **

**LABORATORIO DE INGENIERÍA DE CONTROL**

**CURSO 2018/19**

**PRÁCTICA Nº: 4**

**Título: REALIMENTACIÓN DEL ESTADO MEDIANTE TÉCNICAS DE DISEÑO ÓPTIMO (LQR).**

**APELLIDOS, NOMBRE: Martínez Trapiello, Alberto**

**NÚMERO DE MATRICULA: 52731**

**INTRODUCCIÓN**

Una vez comenzada en la anterior práctica la introducción al estudio del espacio de estados, en esta práctica comenzaremos con el diseño de controladores. Para ello compararemos entre diseño mediante el método de realimentación de estados (asignación de polos) y el diseño óptimo, que se basa en encontrar un equilibrio entre los costes y la salida ideal.

Para le diseño mediante asignación de polos se toma la entrada como cero y se realimentan los estados modulados por un vector de k’s. Dado este cambio se le obliga al sistema a tener como polos los deseados (que para este diseño se han determinar todos, no sólo los dominantes).

Para diseñar el control mediante realimentación de estados del sistema en Matlab se cuenta con la función ‘*place’*, la cual dada la matriz de estados, la matriz de entradas y los polos deseados te devuelve el vector de k’s necesario. Este diseño consigue que los polos del sistema se muevan hasta los polos deseados, pero puede ser que no sea posible implementarlo en la práctica o fuera muy costoso.

Para el diseño óptimo, tal y como vimos en el anterior tema, existen unos índices que nos permiten conseguir un control óptimo, tales como el del error cuadrado. Al aplicar estos índices al espacio de estados se consigue llegar a la ecuación de Riccati para poder hallar las k’s óptimas para conseguir el control LQR. Para ello se han de seleccionar unas matrices ‘Q’ (determina la importancia relativa del error) y ‘R’(determina la importancia relativa del coste de energía) adecuadas para nuestro sistema.

Ecuación de Riccati:

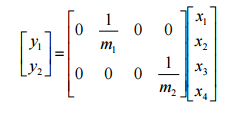
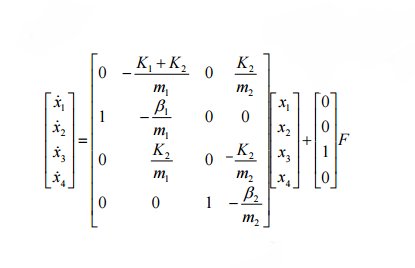
Una vez hallada la matriz P, que es una matriz definida positiva, basta con sustituir en:



**OBTENER EL MODELO DE VARIABLES DE ESTADO**

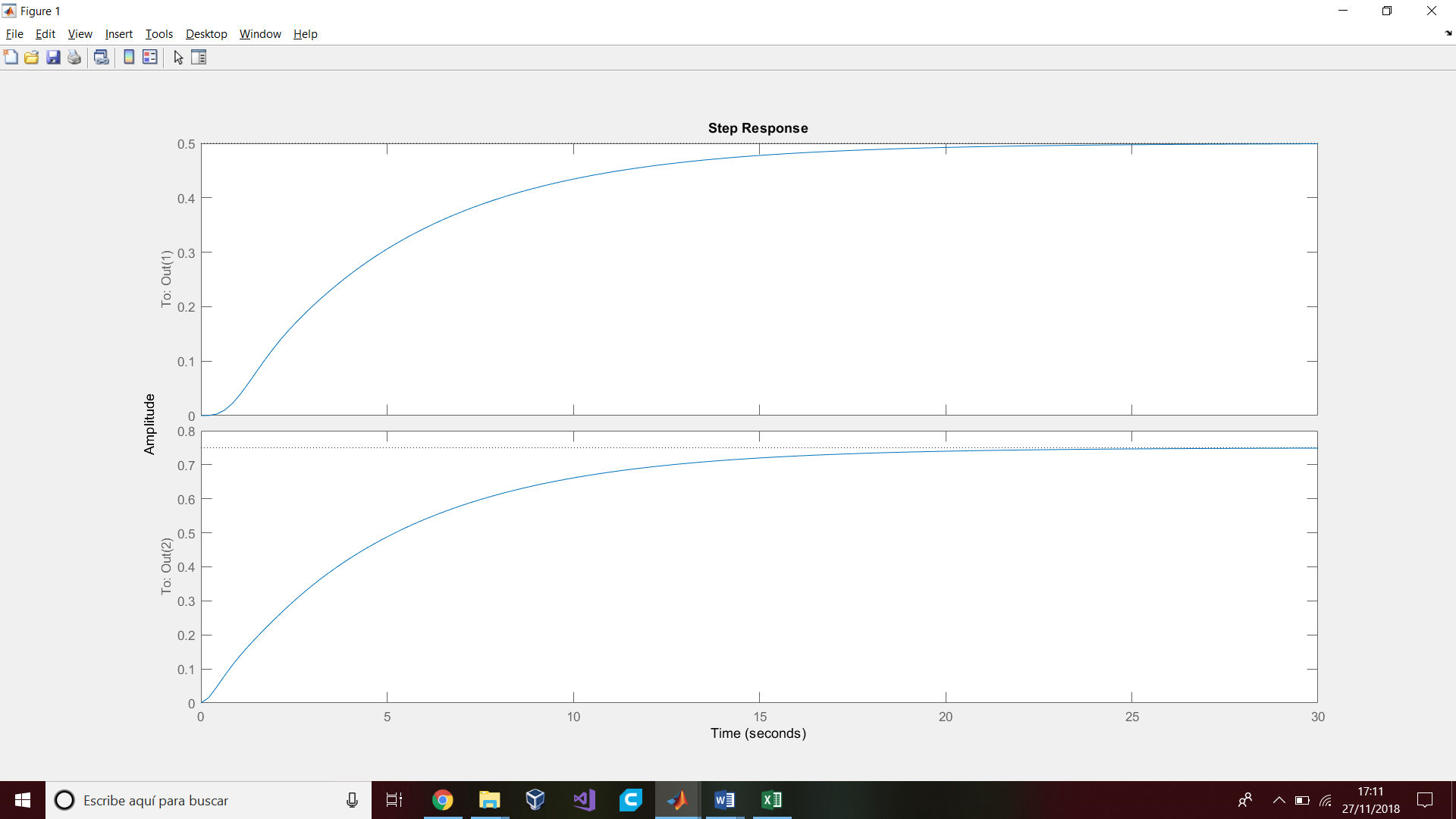
Del estudio del sistema sacamos las ecuaciones diferenciales que rigen su dinámica, y de ellas sacamos las matrices del espacio de estados. Estas ecuaciones se muestran tal y como se vieron durante la introducción al espacio de estados:

Quedando las siguientes matrices de estados:



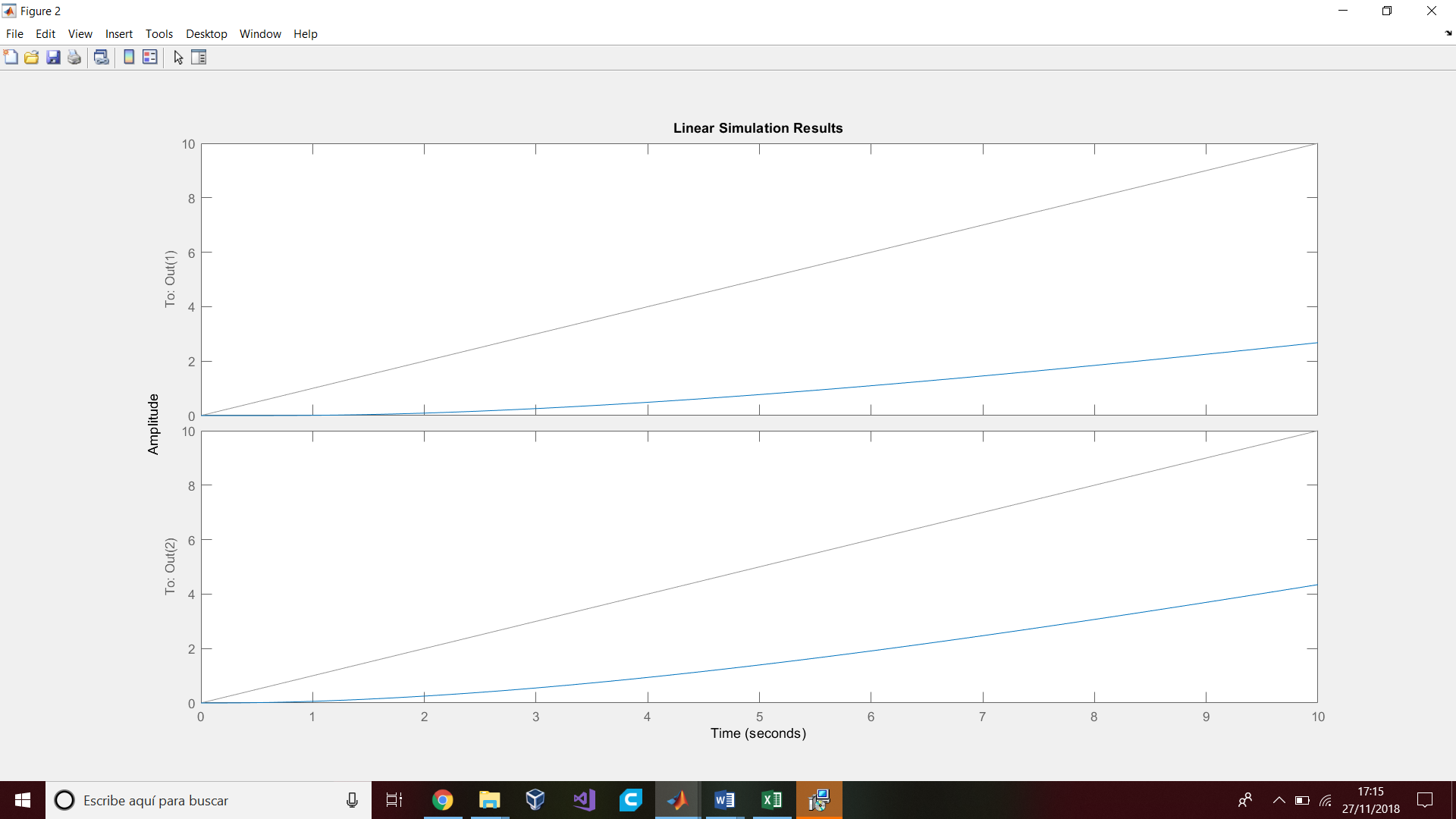
Una vez introducidas las matrices en Matlab se procederá a comprobar la respuesta del sistema ante las distintas entradas mediante las funciones de “step”, y “lsim” con las distintas entradas.

**SALIDA ANTE ESCALÓN:**



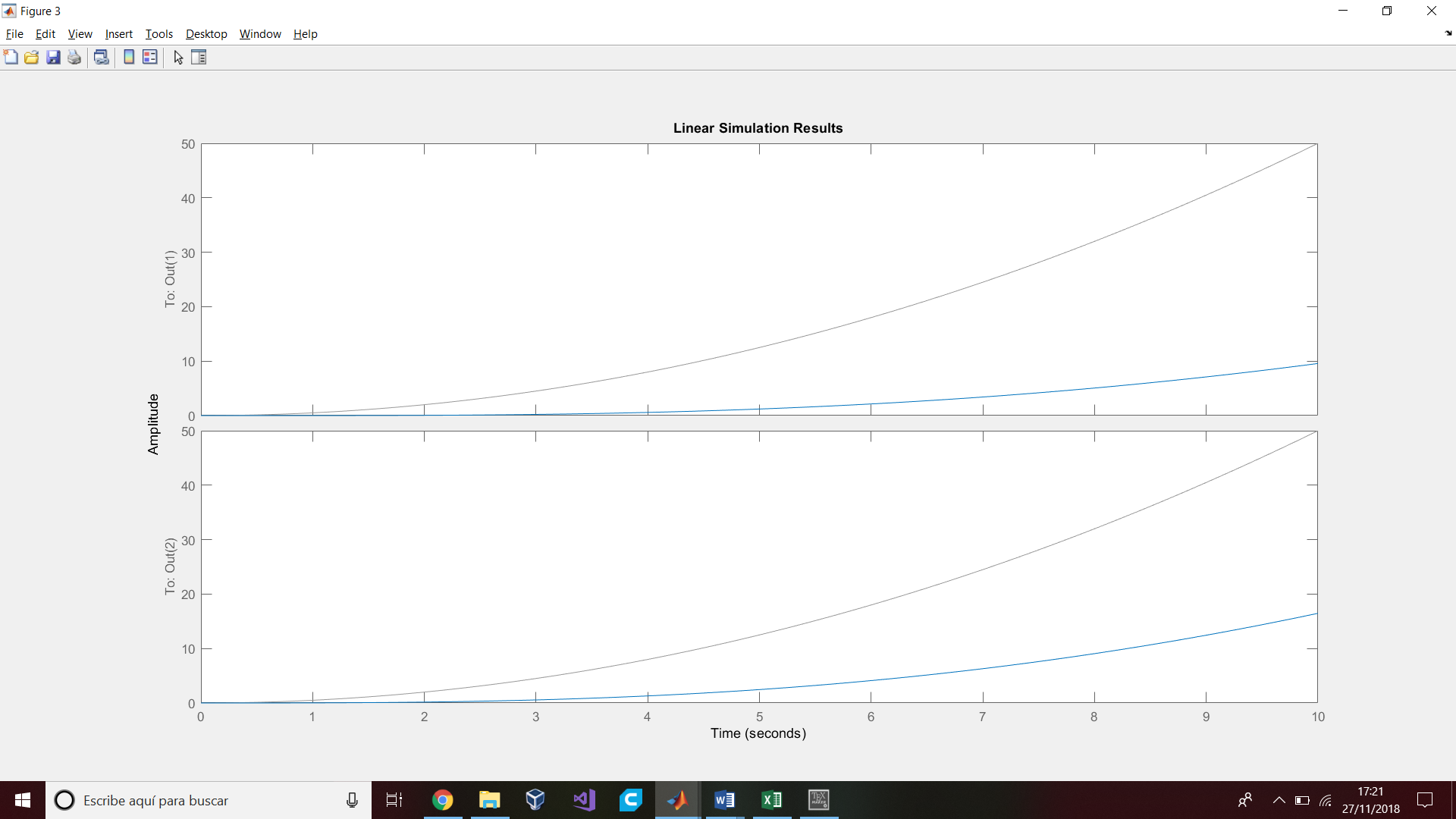
Se puede ver que ambas respuestas son sobreamortiguadas, sobretodo en el caso de la primera que se ve que tiene un punto de inflexión al poco de comenzar. Se nota que es notablemente lento, lo cual puede ser lógico al tratarse de un sistema físico con masas y muelles. Incluso parece que tiene error en régimen permanente notable.

**SALIDA ANTE RAMPA:**



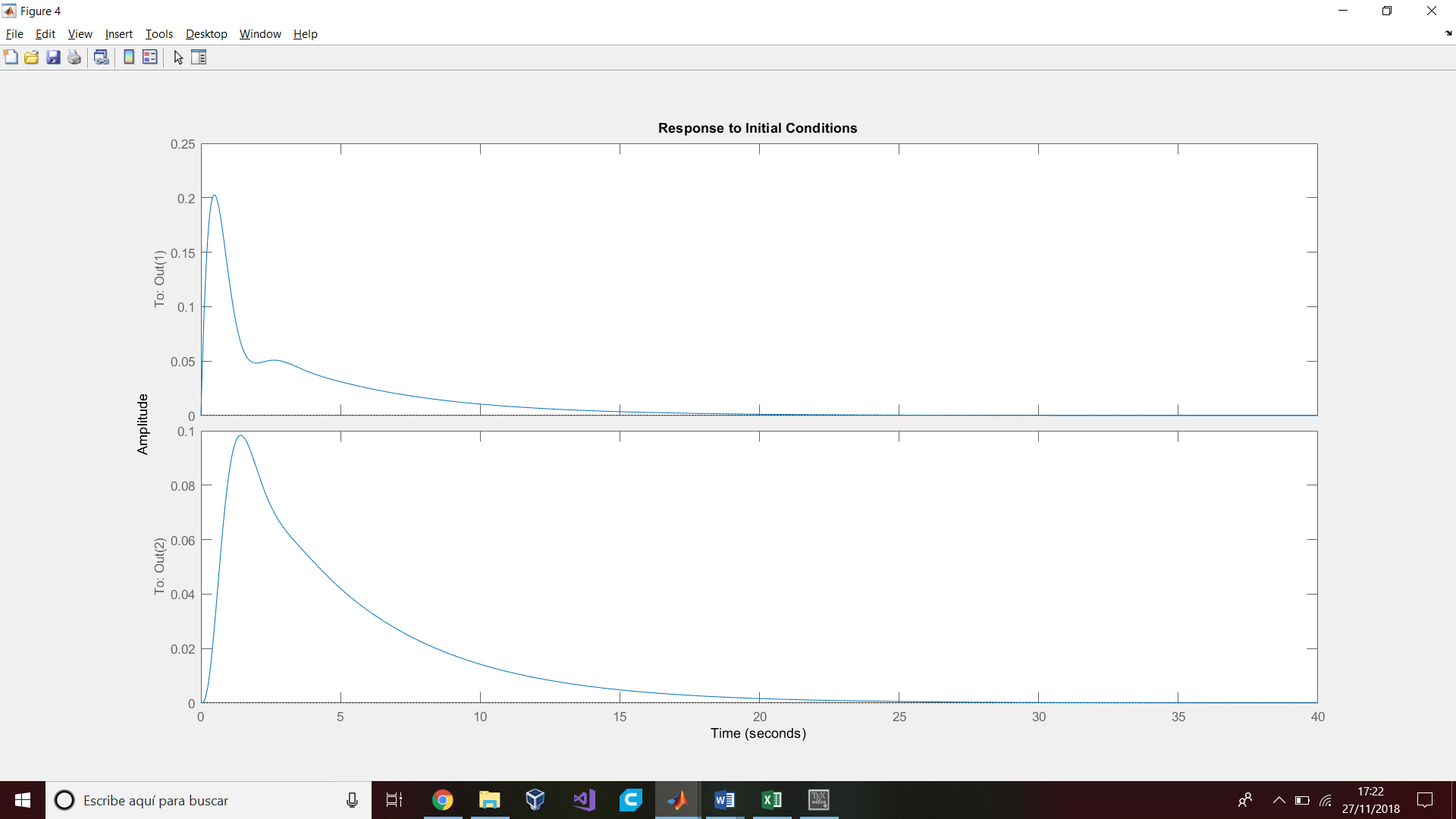
Ante la rampa la respuesta se sigue denotando lenta y no sigue correctamente la salida, lo cual se correlaciona con el error en régimen permanente ante el escalón, ya que si tiene error en posición debería tenerlo a su vez en velocidad.

**SALIDA ANTE PARÁBOLA:**



Por último la respuesta a la parábola sigue arrojando resultados similares a los anteriores de sistema lento, impreciso y no oscilatorio.

**SALIDA ANTE CONDICIONES INICIALES**



Por último con condiciones iniciales y entrada nula se puede ver que claramente que sube rápidamente y tiene un pico importante, pero luego se estabiliza sin oscilar pasado un tiempo.

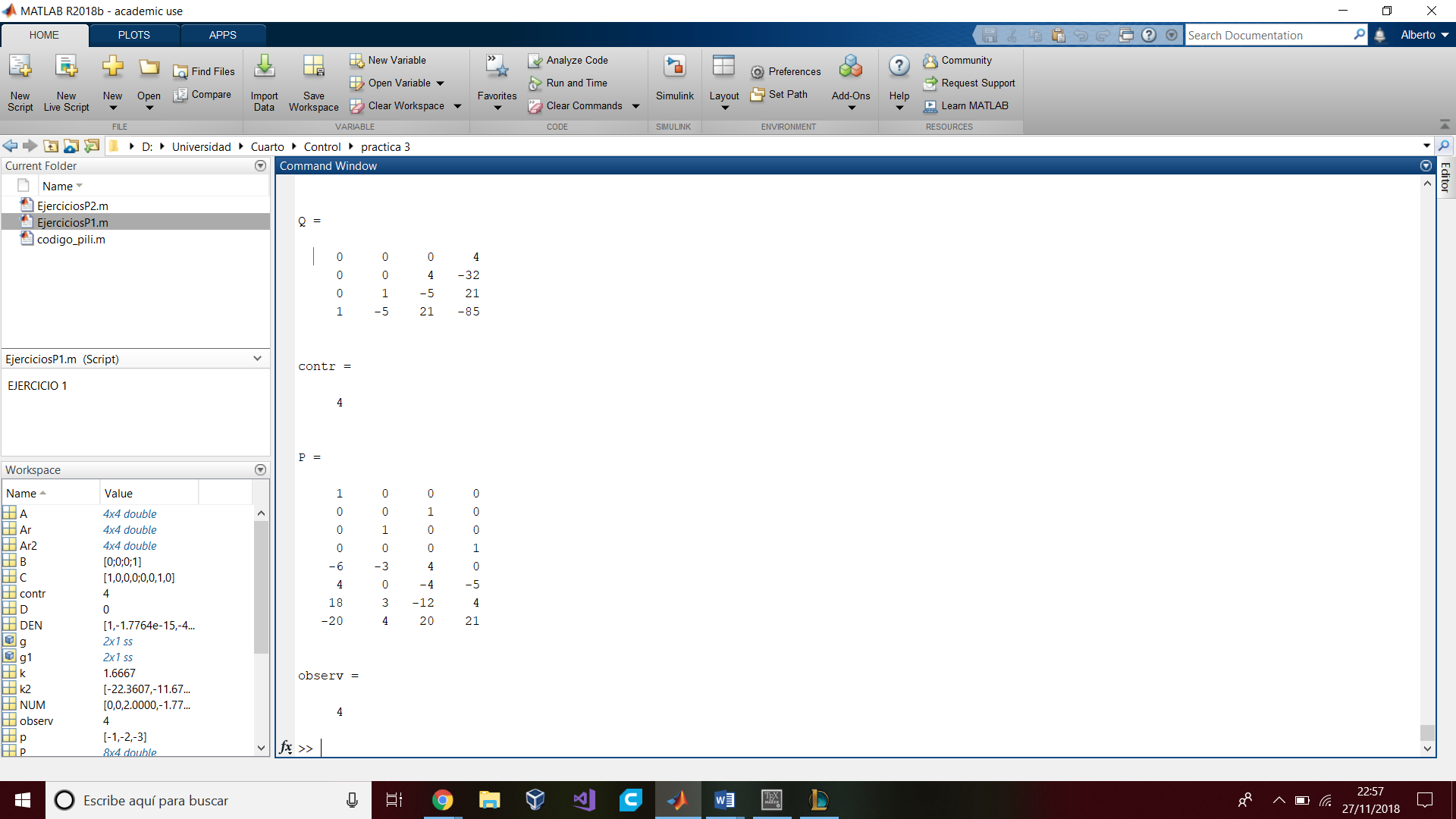
Se procede a realizar el segundo ejercicio, pero en este caso en lugar de contar con el sistema físico y luego hallar las ecuaciones y el espacio de estados, contamos con los polos, ceros y ganancia del sistema. Para poder trabajar en Matlab con estos datos hacemos uso de “zp2ss”.

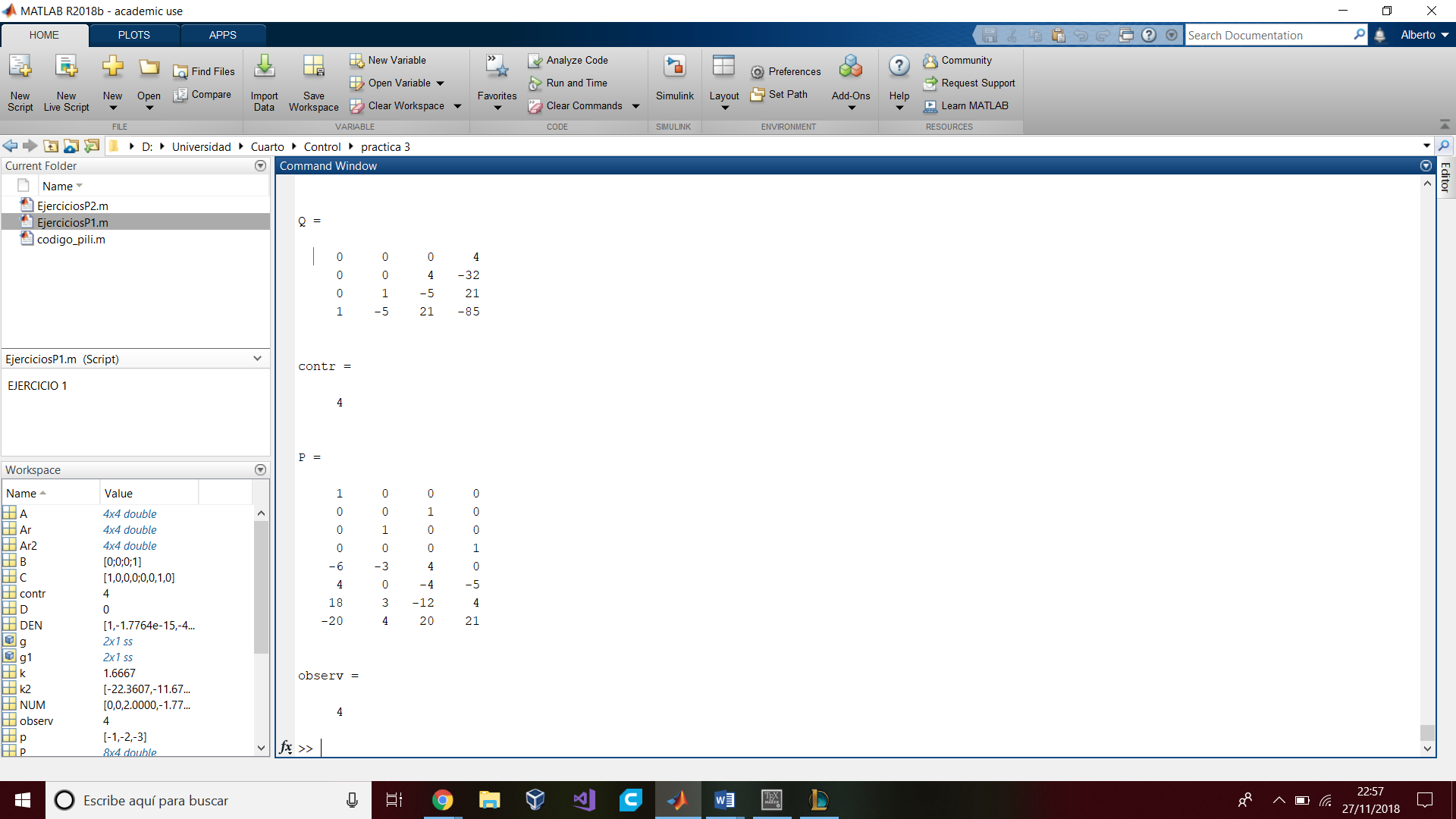
**CONTROLABILIDAD Y OBSERVABILIDAD**

Como condición necesaria para el futuro diseño de controladores se han de evaluar estas condiciones. La controlabilidad es imprescindible para poder diseñar un controlador, ya que permite transferir los estados del sistema desde un estado inicial a cualquier otro mediante un vector de control sin restricciones, en un intervalo finito.

Por otra parte, la observabilidad no es requisito imprescindible para el control, pero asegura que se puede obtener un estado mediante la observación de la salida durante un intervalo de tiempo.

Para calcular las matrices de controlabilidad y observabilidad contamos con dos funciones de Matlab “ctrb” y “obsv” que no realizan el cálculo. Por lo que con tan solo calcular su rango podremos ver si el sistema cuenta con estas dos propiedades.

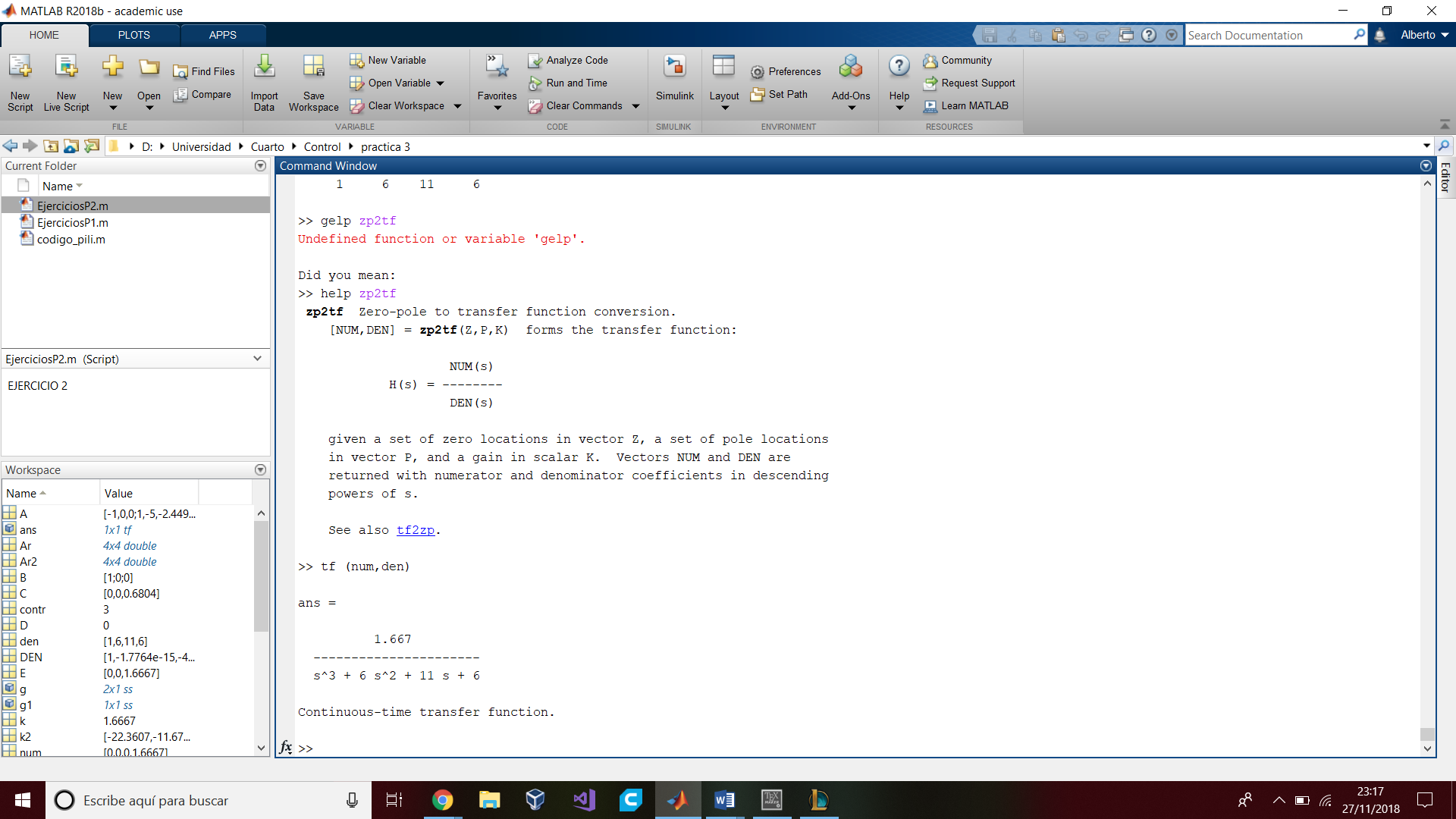




Al comprobar que el rango de ambas matrices coincide con la dimensión, se puede concluir que se trata de un sistema controlable y observable.

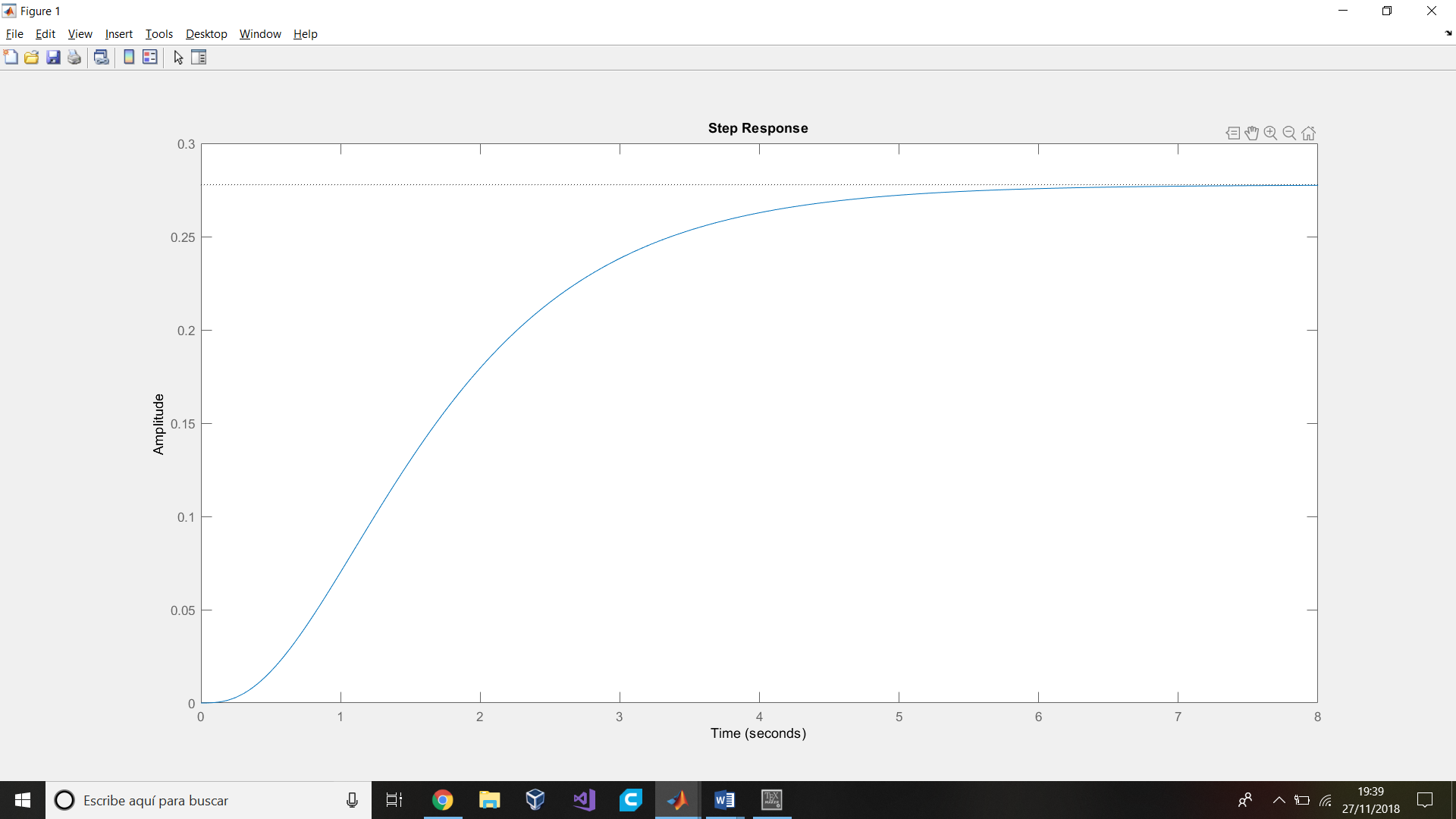
**EJERCICIO 2**

Primero a través del uso de las funciones “zp2tf” podemos pasar de los polos, ceros y ganancia del sistema (que son los datos del enunciado) a la función de transferencia y posteriormente podemos pasar al espacio de estados (también se puede usar “zp2ss” para pasar directamente).



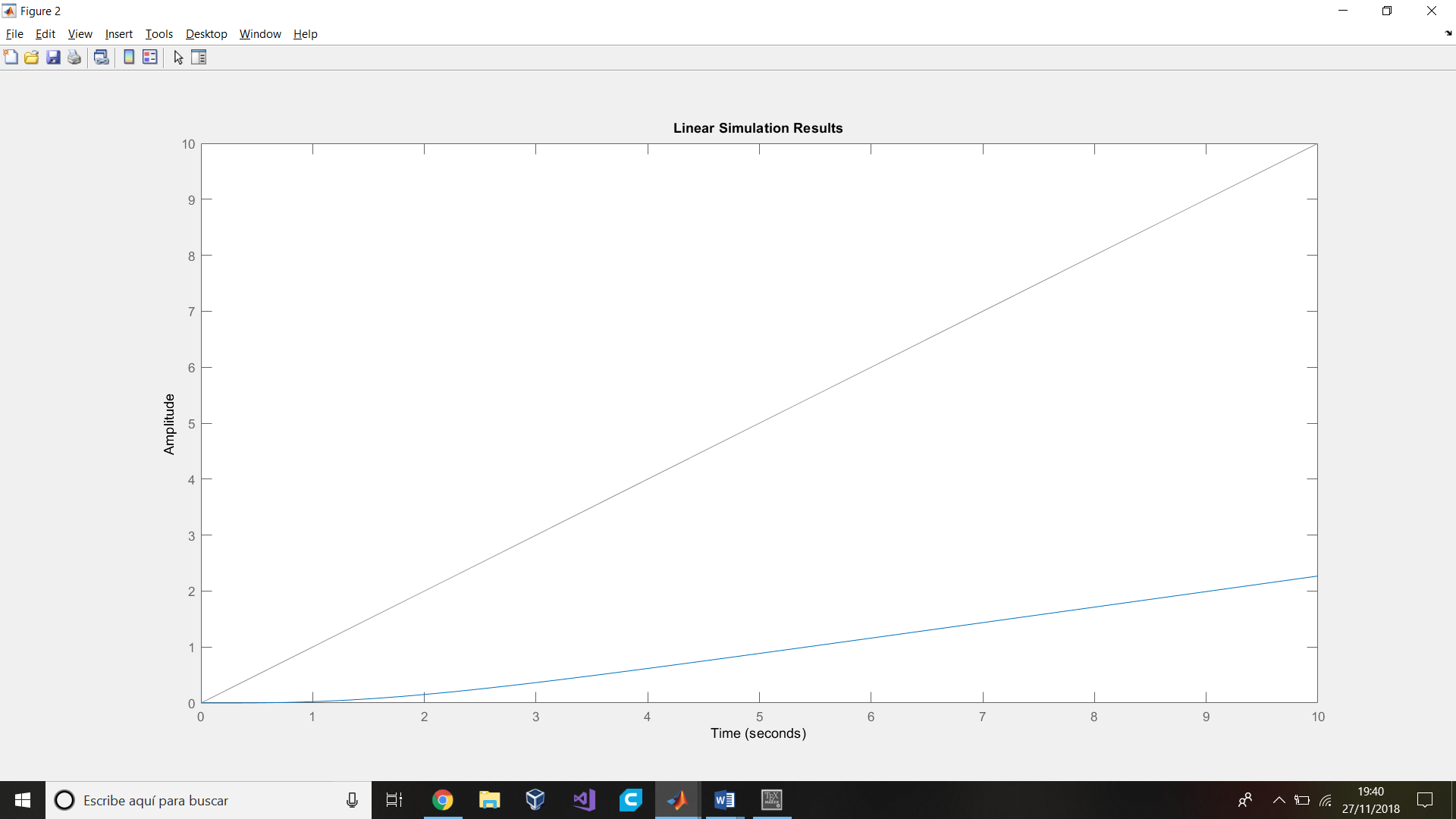
Una vez halladas las matrices del espacio de estados se procede a realizar el resto de apartados como anteriormente.

**SALIDA ANTE ESCALÓN:**

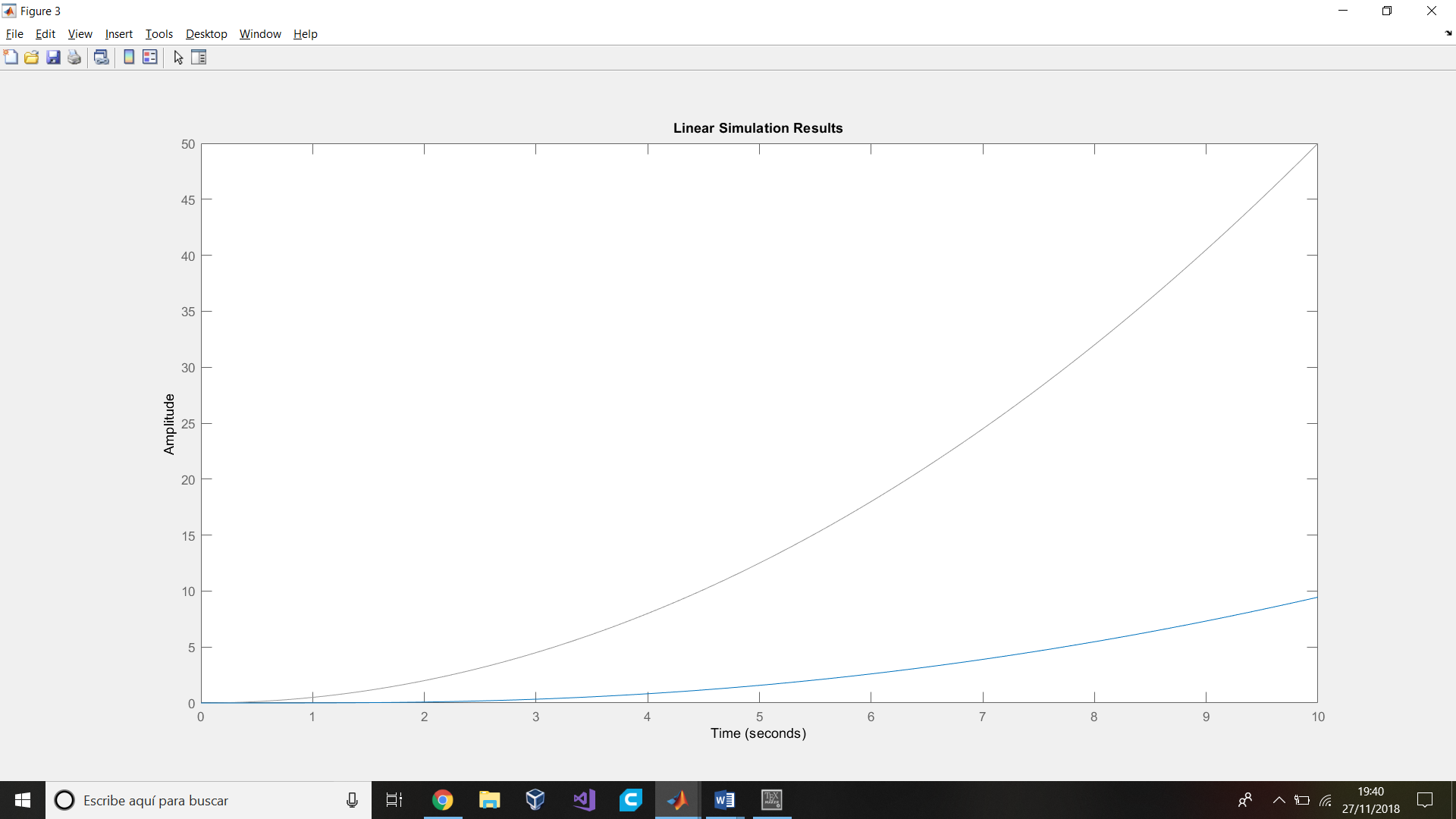


Para este sistema se puede ver que es también sobreamortiguado y tiene error en régimen permanente.

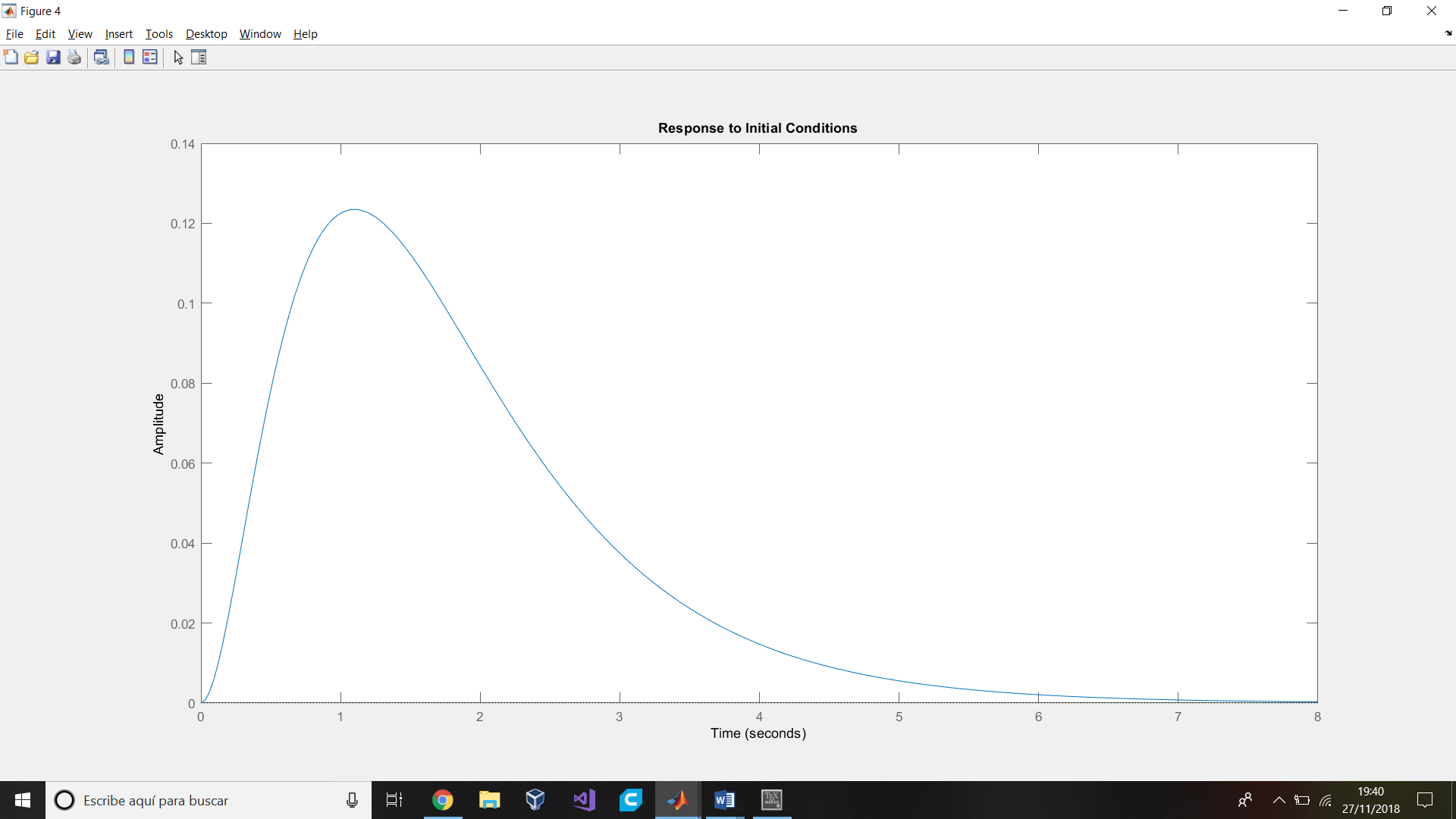
**SALIDA ANTE RAMPA:**



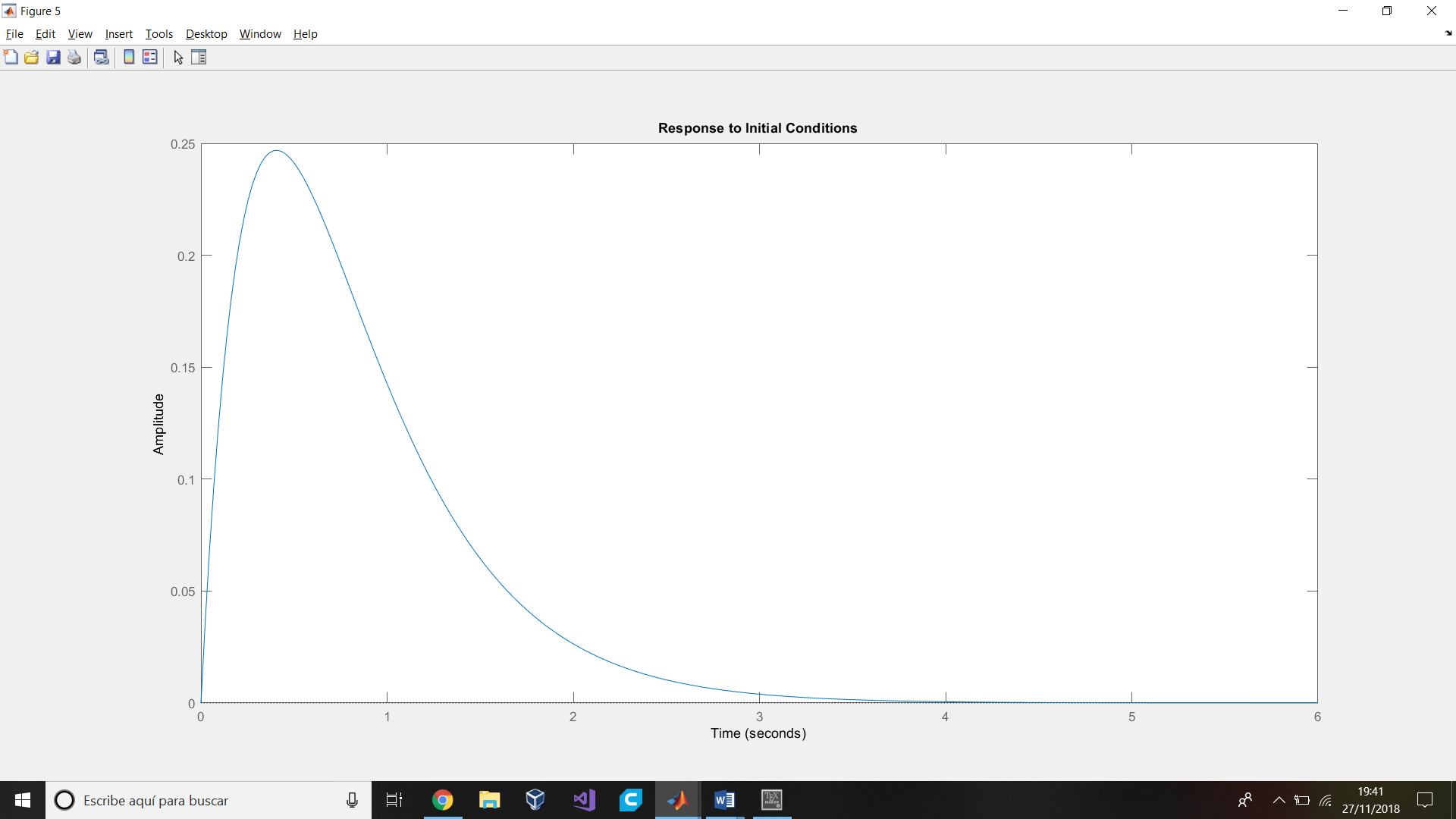
**SALIDA ANTE PARÁBOLA:**



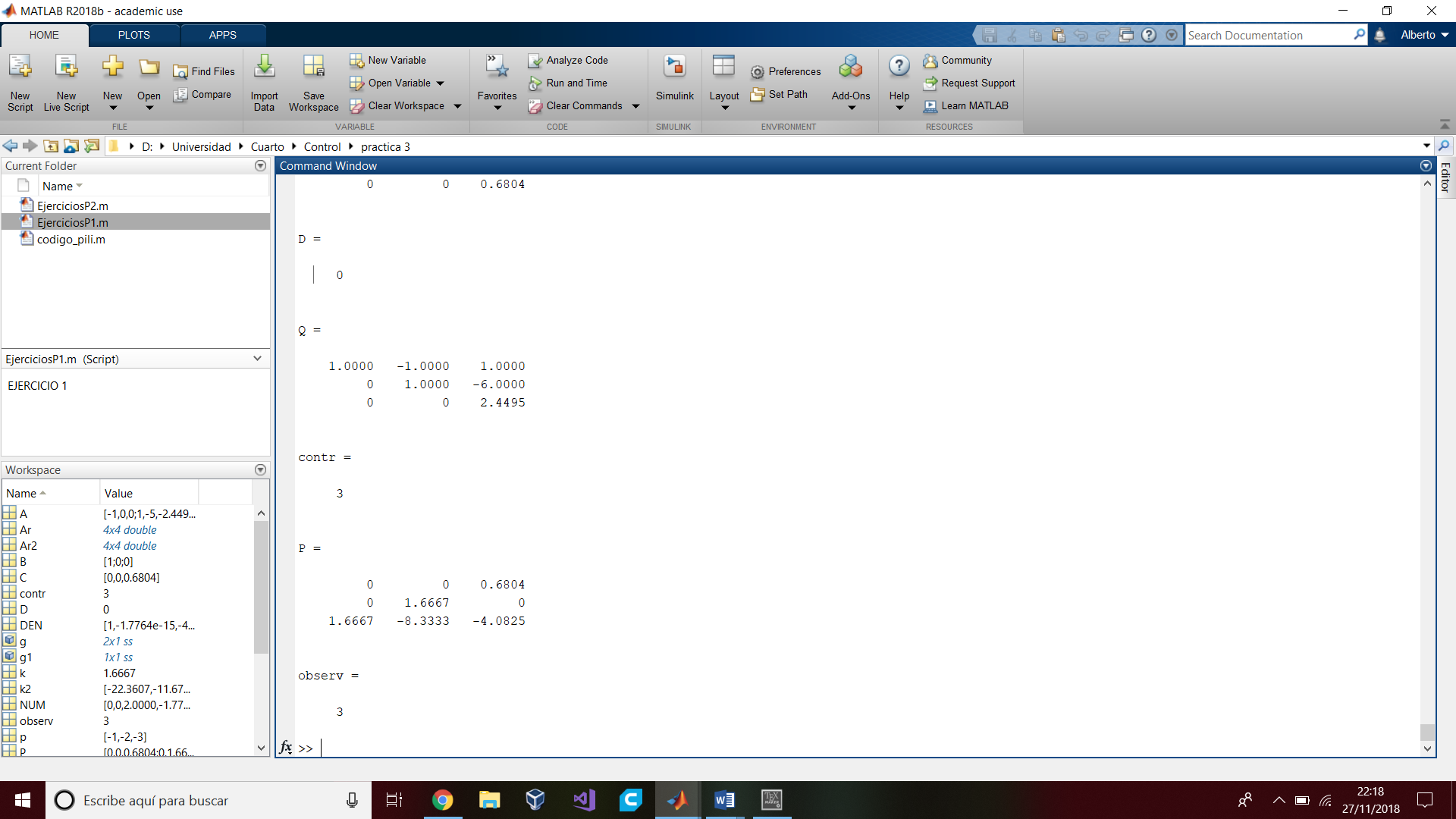
**SALIDA ANTE CONDICIONES INICIALES X0 = [1, 0 ,0]**



**SALIDA ANTE CONDICIONES INICIALES X0 = [0, 1 ,0]**



**CONTROLABILIDAD Y OBSERVABILIDAD**



Al comprobar que el rango de ambas matrices coincide con la dimensión, se puede concluir que se trata de un sistema controlable y observable.

**CONCLUSIONES**

Para concluir, en esta práctica nos hemos podido introducir en el espacio de estados y empezar a comprender las virtudes que este presenta frente al estudio mediante funciones de transferencia. Ya que en este caso se consigue acceso a más variables gracias a la observabilidad, además de trabajar con matrices que son más agradecidas para trabajar en ordenador.

Cabe destacar que, aunque sea mejor trabajar en el espacio de estados, las herramientas matemáticas y conceptos aprendidos (estabilidad, comportamiento dinámico y estático de sistemas lineales, asignación de polos, etc.) hasta ahora, siguen presentes, aunque se trabaje sobre ellos de distinta manera. Por ello gracias a los dos ejemplos evaluados en esta práctica hemos podido comprobar las ventajas de esta nueva metodología, así como su relación con lo anteriormente aprendido.