

Universidad Autónoma de Madrid. Escuela Politécnica Superior

Proyecto: Rubik Solver

Plataformas para sistemas empotrados

Marcos Asenjo González – Jon Larrea Martínez
27-4-2019

1. Descripción del proyecto a realizar

El proyecto *Rubik Solver* consiste en la construcción de un robot que sea capaz de resolver un cubo de rubik de 3x3x3 gracias a la utilización de 8 servomotores y de una cámara para al inicio detectar el estado del cubo.

El procesador principal encargado de detectar los diferentes colores de cada pieza de las caras del cubo y de realizar el cálculo de los movimientos para resolver el cubo mediante un algoritmo que garantiza menos de 25 movimientos residirá en una Raspberry Pi 2 a la que se le ha acoplado el módulo PiCamera. Por otro lado, es necesario un controlador encargado de gestionar el movimiento de todos los motores para el cual hemos utilizado un Arduino Uno, el cual a través de diversas librerías (creadas específicamente para este proyecto y otras creadas por terceros), nos permite girar los 8 servomotores en el rango [0, 180] grados.

2. Actuadores necesarios

Los actuadores que necesitaremos para este proyecto son 8 servomotores que nos permitirán realizar movimientos sobre las seis caras del cubo de Rubik:

- **4 x DS3218:** Es el servomotor con más fuerza de los dos (20 Kg) y por tanto es el encargado de realizar el movimiento más exigente, girar 4 caras del cubo (cada servo girará una de las caras).
- **4 x S3003:** Este servomotor tendrá la función de acercar y alejar cada uno de los cuatro servomotores DS3218 a las caras del cubo, permitiendo cambiar la cara que van a rotar.

Todos ellos estarán gestionados por el mismo controlador, que a su vez responde a las órdenes del procesador principal. Por tanto, disponemos de un sistema que con cuatro servomotores principales y cuatro secundarios es capaz de controlar la posición de las seis caras del cubo.

3. Sensores necesarios

El único sensor utilizado en esta práctica es el módulo *PiCamera* el cual consiste en un módulo que integra cámara con una lente de ojo de pez y con una resolución de 8MP y que será el responsable de tomar las 6 imágenes correspondientes a cada cara del cubo para posteriormente en la Raspberry Pi procesar dichas imágenes para obtener los colores de las piezas de cada cara. Este módulo ha sido seleccionado debido a su facilidad de uso mediante la librería PiCamera de Python.

4. Control, procesamiento de los datos y comunicaciones

Para la identificación de los colores de las piezas que forman cada cara aprovechamos que la cámara se encuentra en una posición fija respecto al cubo para definir unas secciones cuadradas en las imágenes en cada una de las piezas que forman cada cara de manera que se realiza la media de los tres colores de cada uno de los píxeles de dicha sección para obtener un color representativo de esa cara. Seguidamente se calcula la distancia en el espacio de colores a los 6 colores del cubo y el color con la distancia que sea mínima corresponde con el color de dicha cara. Una vez identificado el vector con los colores de las diferentes caras es necesario aplicar ciertas transformaciones para simular que el cubo tiene la siguiente estructura:

Para actuar sobre los servos y hacer que estos giren las diferentes caras del cubo de acuerdo a la solución aportada por el algoritmo de Kociemba ya calculado utilizaremos un microcontrolador Arduino modelo Uno el cual nos permitirá con la librería servo.h que ya trae por defecto su IDE de programación junto con una librería propia diseñada para controlar los 8 servomotores de una manera sencilla.

Para realizar la conexión entre el Arduino y la Raspberry utilizaremos un cable USB (El que se utiliza en la carga del código en el Arduino) con una conexión serie para el envío de comandos y recepción de ACKs desde la Raspberry Pi 2.

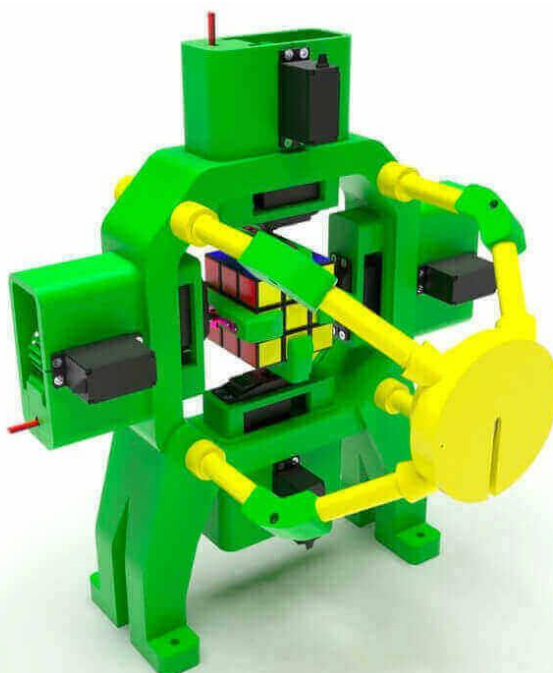
5. Diagrama de montaje mecánico

Debido a que se trata de un proyecto muy ambicioso y que no se disponía de mucho tiempo para realizarlo, hemos decidido utilizar para el montaje los diseños para impresora 3D gratuitos proporcionados por el equipo de OTVINTA. El proyecto que ellos diseñaron se llama RCR3D y es un proyecto orientado a introducir la robótica, la mecánica, la programación y la inteligencia artificial a los niños.

El montaje consta de una estructura principal que rodea el cubo la cual proporciona acceso a 4 brazos servocontrolados que disponen de una pinza rotativa en un extremo y que se acoplarán a 4 de las caras del cubo. Además, dispone de dos soportes que sostienen la estructura principal.

De la estructura principal se extiende hacia la parte trasera un soporte donde va colocada la cámara.

La estructura principal cuenta con 4 espacios para colocar los brazos que se acercan y alejan del cubo con el espacio correspondiente para el servomotor que los extiende y retrae. Por otro lado, estos brazos cuentan cada uno con un espacio donde se coloca el servo que va conectado a la pinza que permite girar cada cara.



El montaje eléctrico cuenta con 2 fuentes de alimentación bien diferenciadas. La primera de ellas (Vcc1) alimenta a la Raspberry Pi y esta a su vez al Arduino y es de 5V. La segunda a su vez hemos decidido dividirla en dos fuentes ambas de 5 voltios, pero una de 1 amperio y otra de 2 amperios. Esto es así porque los motores encargados de mover las caras del cubo (principales) necesitan una mayor potencia que los encargados de acercar y alejar las pinzas de las caras del cubo (secundarios). La cámara no necesita ninguna alimentación externa ya que es proporcionada por la Raspberry Pi.

7. Estructura del programa principal

El programa está dividido en dos partes y dos fases.

La primera parte es la encargada de tomar las fotografías de las diferentes caras, analizarlas para detectar el color de cada cara, generar una representación del estado inicial del cubo y realizar el cálculo de los movimientos necesarios para resolver ese cubo mediante el algoritmo de Kociemba. Esta primera parte está escrita en Python y será ejecutada por la Raspberry.

La segunda parte se encargará de, dependiendo del comando recibido por la conexión serie mover los servos de forma que no haya bloqueos entre ellos para realizar el movimiento indicado. Esta parte estará programada en C++ y será ejecutada por el Arduino.

Definiremos ambas fases como la fase de identificación (fase 1) y la fase de resolución (fase 2).

La fase de identificación consiste en que la Raspberry envía comandos al Arduino para que este coloque cada una de las 6 caras del cubo delante de la cámara para tomar 6 fotografías. Una vez tomadas las 6 fotografías, se procede a identificar qué color corresponde con cada cara, indicado por la pieza central de la misma y finalmente a obtener la lista de movimientos que solucionan esa configuración del cubo en concreto mediante el algoritmo de Kociemba.

Una vez se ha obtenido la lista de movimientos, la Raspberry los va enviando uno a uno al Arduino, que mediante comunicación serie va realizando las combinaciones de movimientos de los motores que se corresponden con dicho movimiento del cubo. Una vez ha terminado de ejecutar la combinación de movimientos, le envía una señal ACK a la Raspberry para indicar que está a la espera del siguiente movimiento.

8. Estructura del programa principal

Las pruebas que hemos realizado sobre el sistema son, en primer lugar, modulares y luego hemos realizado pruebas de integración.

Pruebas Modulares

Pruebas del módulo de la cámara

Se colocará el cubo en el soporte en sus distintas posiciones y se anotarán las decisiones de la cámara sobre el cubo. Se comparará manualmente si la decisión del color tomada por la cámara coincide con el cubo en esa posición. En caso de no coincidir, se ajustarán los patrones de la decisión del color hasta que tome la decisión correcta repetidas veces sobre todos los colores.

Pruebas del procesador principal

Se colocará el cubo con una configuración de las piezas preestablecida y se resolverá manualmente el algoritmo de Kociemba para obtener la lista de movimientos

necesarios para dicha configuración. Se imprimirá por pantalla la lista obtenida tras aplicar el algoritmo en la Raspberry y se comparará si coinciden. Se realizará este procedimiento para varios patrones de configuración del cubo.

Pruebas del controlador y de los servomotores

Se definirá en un fichero una serie de movimientos de actuadores predefinidos, de forma que sea fácil realizar un seguimiento de si se están cumpliendo o no. Se simulará con este fichero el envío de órdenes por parte de la Raspberry al microcontrolador de forma que se pueda realizar una inspección visual de que las órdenes recibidas en el Arduino conducen a los movimientos correctos de los servomotores.

Pruebas de Integración

Primero probaremos la integración de la Raspberry con la cámara colocando el cubo en una configuración predeterminada y viendo si los pasos resultantes para esa configuración (una vez hemos leído el cubo con la cámara) son los adecuados.

Las pruebas de integración de la Raspberry con el Arduino y los servomotores son simplemente probar la resolución del cubo en distintas configuraciones. La consecución con éxito del sistema es muy visual, si se ha resuelto correctamente el cubo en sus distintas configuraciones la integración habrá resultado un éxito.

9. Problemas encontrados

Los principales problemas para llevar a cabo el proyecto han sido de dos tipos: de motores y de algoritmos.

Los problemas relacionados con los motores han sido varios:

- **Problemas con el voltaje:** Tras observar que el voltaje que proporcionábamos a los motores era insuficiente decidimos aumentarlo. Eso nos permitió observar que los motores principales funcionaban correctamente si se les suministraba cierta potencia. Tras observar que la fuente de alimentación presente en el laboratorio era insuficiente, y al no contar con una fuente de alimentación suficiente para los ocho servomotores, decidimos alimentar los dos grupos de motores con fuentes independientes (una de mayor potencia para los servomotores DS3218 y una auxiliar para los otros cuatro). De esta manera ya recibían la potencia necesaria.
- **Problemas con PWM:** Los servomotores principales no funcionaban correctamente con la librería Servo.h incluida en el IDE de Arduino (no llegaban a girar los 180 grados necesarios para nuestro proyecto), por lo que tuvimos que generar nuestras propias señales de PWM poniendo el valor del pin a 1 y a 0 en función del ciclo de trabajo deseado. Esta implementación consigue que los servomotores giren los 180 grados necesarios pero a costa de que todos giren de manera simultánea en determinadas ocasiones no coloca los servomotores en la posición ideal.
- **Impresión 3D:** Debido al modelo 3D seleccionado, algunos motores tenían problemas al retraer y recoger las pinzas debido a que las piezas 3D estaban muy ajustadas y provocaban una alta fricción impidiendo el correcto funcionamiento. Esto se

solucionó lijando las piezas para reducir su volumen y que no provocasen gran oposición a los motores. Además, también tuvimos que colocar unas gomas en las pinzas para que estas se adhiriesen mejor al cubo y que éste no se resbalase.

Los problemas relacionados con los algoritmos han sido los siguientes:

- **Algoritmo de reconocimiento de colores:** Debido a que este algoritmo se basa en una calibración realizada bajo unas condiciones lumínicas determinadas, en el momento que se quiere realizar la detección de las caras en un entorno diferente éste falla en algunos casos como por ejemplo confundiendo el rojo con el naranja o el amarillo con el blanco. Esto se ha solucionado permitiendo la revisión manual de los colores una vez han sido detectados para su corrección en el caso de que el algoritmo se equivoque.
- **Algoritmo de resolución del cubo:** Este algoritmo lanza una excepción en caso de que la representación del cubo que le proporcionemos no sea válida porque no se pueda resolver ya que alguna pieza esté identificada mal. Esto se solucionó cuando averiguamos la forma de representar el cubo correctamente y de cómo transformar las piezas detectadas en la fase de identificación a la representación del cubo esperada por el algoritmo.

10. Limitaciones

Las limitaciones que encontramos para este proyecto son las siguientes:

- **Limitación temporal:** En una secuencia de movimientos no puede comenzar un servomotor a rotar hasta que no haya terminado el movimiento del motor anterior, ya que podría haber choques o rotaciones sobre el cubo no esperadas. Por tanto, es necesario introducir un tiempo de espera entre movimientos que sea suficiente para que no haya movimientos de motores simultáneos (en nuestro caso es un segundo). Además, en los movimientos simultáneos de motores, para evitar la desincronización se reduce la velocidad con la que giran, controlando ángulo a ángulo su movimiento. Todo esto introduce un retardo que hace que la resolución del cubo dure varios minutos, luego este proyecto tiene una severa limitación temporal.
- **Siempre para las mismas dimensiones del cubo:** Las pinzas de los servomotores están diseñadas de acuerdo a un determinado cubo con unas determinadas dimensiones, y aunque la mayoría de cubos tienen estas dimensiones, nuestro robot no es capaz de resolver cubos que tengan más o menos dimensión.
- **Siempre para nueve piezas por cara:** Además de tener que tener una dimensión determinada, los cubos a resolver siempre tienen que tener nueve piezas por cara. Esto es así por dos razones, la primera es que la cámara coge nueve secciones de la imagen para determinar el color de las piezas y la segunda es que los movimientos de los servomotores sólo son posibles si hay tres filas de piezas tanto en horizontal como en vertical.
- **Siempre los mismos colores:** Al haber calibrado la cámara para discriminar entre nueve colores determinados, si se cambian los colores del cubo el sistema sería incapaz de hacerlos corresponder con los nueve colores que tiene previamente calibrados, por lo que nos encontraríamos con una gran cantidad de errores en la determinación de los colores de las piezas.
- **Poca capacidad de escalado:** Al tener un diseño fijo tomado de un proyecto externo, no contamos con la capacidad de escalar este proyecto en algunos aspectos. Por ejemplo, los soportes de los motores están específicamente pensados para los

servomotores DS3218 y S3003 por lo que nuevos motores con otras dimensiones no podrían ser empleados.

11. Enlaces a demostraciones

En los siguientes enlaces se puede ver una demostración de la detección de colores por parte de la cámara y de la resolución del cubo.

Detección de colores:

<https://www.youtube.com/watch?v=uxbxk4a846E>

Resolución del cubo:

<https://www.youtube.com/watch?v=ATUM0LDtaI8>