

Tarea 3

Javier Contreras

◁ Entrega: 18 de Abril, 2017 ▷

1. Simulación de un robot en Python

1.1. Modelo dinámico

Para mayor fidelidad, se buscó un modelo que incorporase variables como la matriz de inercia del móvil. El modelo usado en 2D, proveniente de [2, pg. 5] y usando las ecuaciones de Lagrange, corresponde a:

$$(m + \frac{2I_w}{R^2})\dot{v} + m_c d \omega^2 = \frac{1}{R}(\tau_r + \tau_l)$$

$$(I + \frac{2L^2}{R^2}I_w)\dot{\omega} + m_c d \omega v = \frac{L}{R}(\tau_r - \tau_l)$$

Donde m es la masa del cuerpo, m_c la masa sin las ruedas, I_w el momento de inercia de cada rueda, R el radio de cada rueda, I el momento de inercia total respecto al eje yaw con respecto al centro de gravedad, d es la distancia desde el centro de masa al eje de rotación, τ_r y τ_l los torques correspondientes a las ruedas derecha e izquierda respectivamente, y ω y v las velocidades angular y lineal. Por simplicidad en la modelación gráfica del robot, se decidió descartar el valor d , asumiendo entonces que el centro de masa del móvil siempre está por sobre el eje de rotación.

Se incorporó además un término $\dot{v} = -c, \dot{\omega} = -c$, donde c es una constante equivalente al roce total. Esto fue implementado con una simple multiplicación de v por un valor < 1 en cada paso de la simulación.

1.2. Control Manual

Las flechas *arriba* y *abajo* ambas corresponden un torque equitativo entre los motores igual a la mitad del torque máximo definido para el robot. Cuando se pulsa una flecha lateral, sin haber movimiento hacia adelante o atrás, los torques son opuestos aditivos de manera que el robot gire en el mismo lugar. Si hay movimiento tanto hacia los frentes como hacia los lados, se le entrega a un motor el torque máximo y al otro exactamente la mitad.

Todo esto queda codificado en la siguiente expresión del código:

```
tl = tq*((F - B)*(1 + R*(not L)) + (R - L)*(not (F or B)))  
tr = tq*((F - B)*(1 + L*(not R)) + (L - R)*(not (F or B)))
```

Donde tr y tl son los torques derecho e izquierdo respectivamente, y tq es un valor fijo de pulso de torque equivalente a la mitad del torque máximo permitido al robot.

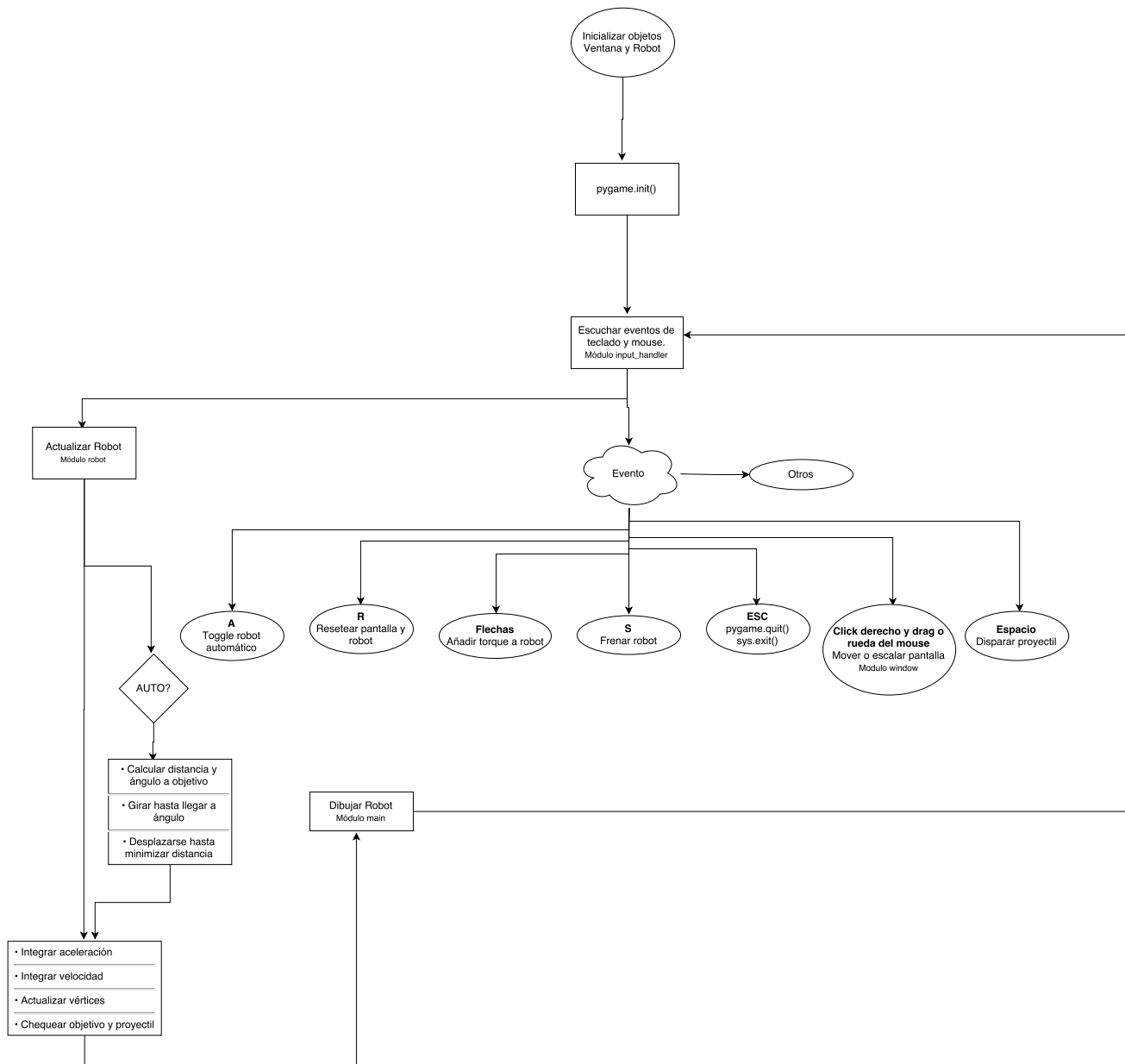
1.3. Control Automático

Tal como se describe en [3, pg. 188], para cada punto objetivo se calculó una trayectoria cúbica con velocidades iniciales y finales iguales a cero, tanto para la velocidad lineal como la angular. El valor \dot{v} o $\dot{\omega}$ de esta trayectoria, una vez restada la constante de roce c , luego se reemplazó en las ecuaciones de movimiento previa, y resolviendo para los torques se llegó a lo siguiente:

$$\tau_R \rightarrow -\frac{ct_f^3 (R^2(I + Lm) + 2I_w L(L + 1)) + 6\Delta\theta(2t - t_f) (IR^2 + 2I_w L^2) + 6dL(2t - t_f) (2I_w + mR^2)}{2LRt_f^3}$$
$$\tau_L \rightarrow \frac{ct_f^3 (R^2(I - Lm) + 2I_w (L - 1)L) + 6\Delta\theta(2t - t_f) (IR^2 + 2I_w L^2) - 6dL(2t - t_f) (2I_w + mR^2)}{2LRt_f^3}$$

Como el valor de t se debe reiniciar cada vez que el punto objetivo cambia, el programa se asegura de que al terminar cada etapa del movimiento (i.e. rotar y desplazarse) o al cambiar el objetivo con el click del mouse, una variable interna de la clase robot esté rastreando esto.

Figura 1: Diagrama de flujo simplificado



En este caso se vió innecesaria la implementación de un controlador PD, pues las funciones de control descritas caracterizan completamente el sistema dinámico implementado, y la única variable que efectivamente afecta la exactitud de la trayectoria es el margen de error aceptable para decir que una maniobra ha finalizado.

Sin embargo, en casos de control reales es imperativo poseer un controlador propiamente tal, pues ninguna cantidad de matemática (por lo menos para sistemas incrustados como Arduino) puede caracterizar todas las posibles variaciones en roce con el suelo, arrastre atmosférico, deslices, variaciones en los motores, etc.

Bibliografía

- [1] Shankar Sastry S. M. Murray R. Li Z. *A Mathematical Introduction to Robotic Manipulation*. Ed. por Inc. CRC Press. 1994.
- [2] Abu Hatab A. R. Dhaouadi. "Dynamic Modelling of Differential-Drive Mobile Robots using Lagrange and Newton-Euler Methodologies: A Unified Framework". En: *Advances in Robotics Automation* (2013).
- [3] Vidyasagar M. W. Spong M. Hutchinson S. *Robot Modeling and Control*. Ed. por Inc. John Wiley Sons. 2006.