РАЗРАБОТКА БАЛАНСИРУЮЩЕГО РОБОТА

Кучерявый И.А.
Научный руководитель – ст. преподаватель каф. КИТАМ Бронников А.И.
Харьковский национальный университет радиоэлектроники (61166, Харьков, пр. Ленина,14, каф.КИТАМ)
e-mail: captain34690@gmail.com, тел: (095) 511-75-95

The given work is devoted to control theory, holding a variable (in this case, the position of the robot) requires a special controller called the PID (proportional integral derivative). Each of these parameters has a “gain”, commonly called Kp, Ki, and Kd. The PID provides a correction between the desired value (or input) and the actual value (or output). The difference between input and output is called an “error”.

В теории управления, удерживая некоторую переменную (в данном случае позицию робота), требуется специальный контроллер, называемый ПИД (пропорциональная интегральная производная). Каждый из этих параметров имеет «прирост», обычно называемый Kp, Ki и Kd. PID обеспечивает коррекцию между желаемым значением (или входом) и фактическим значением (или выходом). Разница между входом и выходом называется «ошибкой».

ПИД-регулятор уменьшает погрешность до наименьшего возможного значения, постоянно регулируя выход. В нашем самобалансирующем роботе Arduino вход (который является желаемым наклоном в градусах) устанавливается программным обеспечением. MPU6050 считывает текущий наклон робота и подает его на алгоритм PID, который выполняет вычисления для управления двигателем и удерживает робота в вертикальном положении.

PID требует, чтобы значения Kp, Ki и Kd были настроены на оптимальные значения. Инженеры используют программное обеспечение, такое как MATLAB, для автоматического вычисления этих значений. К сожалению, мы не можем использовать MATLAB в нашем случае, потому что это еще больше усложнит проект. Вместо этого мы будем настраивать значения PID. Вот как это сделать:

– сделайте Kp, Ki и Kd равными нулю;

– отрегулируйте Kp – слишком маленький Kp заставит робота упасть, потому что исправления недостаточно, слишком большой Kp заставляет робота идти вперед и назад, оптимальный Kp сделает так, что робот будет совсем немного отклоняться назад и вперед;

– как только Kp установлен, отрегулируйте Kd– оптимальное значение Kd уменьшит колебания, пока робот не станет почти устойчивым и будет удерживать робота, даже если его толкать;

– установите Ki – при включении робот будет колебаться, даже если Kp и Kd установлены, но будет стабилизироваться во времени, правильное значение Ki сократит время, необходимое для стабилизации робота.

Самобалансирующийся робот похож на перевернутый маятник. В отличие от обычного маятника, который продолжает колебаться, после того как его толкают, этот перевернутый маятник не может оставаться сбалансированным сам по себе.

Маятник — это масса m_p прикрепленная на конце невесомого стержня длины l. На другой конец стержня прикреплен двигатель, развивающий максимальный момент M_k и передающий его на колесо массой m_w и радиусом r.

Задача управления – стабилизировать маятник в вертикальном положении и возвращать в начальное положение колесо.

Уравнения движения, описывающие обратный маятник, представимы в следующем виде:

$$r\cos(\theta)lm_p\ddot{\theta} + r^2(m_p + 2m_w)\ddot{\phi} - r\sin(\theta)\theta^2 lm_p = M_k$$
$$\ddot{\phi}\cos(\theta)lm_p r - m_p gl\sin(\theta) + 2m_p l^2\ddot{\theta} = 0$$

Они кажутся довольно неприятными, но сам робот о них ничего не знает, а управление использует линеаризованную модель, то есть такую:

$$\frac{d}{dt}\begin{bmatrix}\phi\\ \dot{\phi}\\ \theta\\ \dot{\theta}\end{bmatrix} = AX + BM_k = \begin{bmatrix}0 & 1 & 0 & 0\\ 0 & 0 & \frac{-m_p g}{r(m_p + 4m_w)} & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & \frac{g(m_p + 2m_w)}{I(m_p + 4m_w)} & 0\end{bmatrix}\begin{bmatrix}\phi\\ \dot{\phi}\\ \theta\\ \dot{\theta}\end{bmatrix} + \begin{bmatrix}0\\ \frac{2}{r^2(m_p + 4m_w)}\\ 0\\ \frac{1}{lr(m_p + 4m_w)}\end{bmatrix}M_k$$

Литература

1. Самобалансирующийся робот на Arduino. [Электронный ресурс]. – URL: https://arduinoplus.ru/delaem-samobalansiruyushhego-robota-na-arduino/#i-4
2. Балансирующий робот на Arduino [Электронный ресурс]. – URL: http://www.robototehnika.ru/content/article/balansiruyushchiy-robot-na-arduino/
3. Робот балансер на Arduino[Электронный ресурс]. – URL: https://habr.com/ru/post/220989/