



Parqueo automático de un carro modelo a escala

Automatic car parking for a scale model car

José Pablo Chaves-Rodríguez¹

Chaves-Rodríguez, J. Parqueo automático de un carro modelo a escala. *Tecnología en Marcha*. Vol. 33, especial Movilidad estudiantil. Pág 64-72.

 <https://doi.org/10.18845/tm.v33i7.5479>



¹ Licenciatura en ingeniería electrónica. Componentes Intel Costa Rica. Correo electrónico: jpablochr@gmail.com

Palabras clave

Arduino; Puente H; Regulador; Sensor de proximidad Inductivo; Sensor ultrasónico; Servomotor.

Resumen

Existe una problemática para muchos usuarios de automóviles el cual es realizar la maniobra de estacionamiento del vehículo en parqueos ya sea en batería o en paralelo, con esto se da que muchas personas les tome mucho tiempo el estacionar, les genera estrés y en algunos casos les es imposible parquear, y los que lo logran pueden realizar esta maniobra de mal manera desperdiciando espacio y/o dejando a otros vehículos con la imposibilidad de salir de donde se encuentran parqueados. Debido a esto el *Department of measurement - Aeronautical Instrumentation Laboratory* de la Universidad Técnica Checa en Praga (*Czech Technical University in Prague*) ha decidido realizar un sistema de parqueo automático aplicado primero a un carro modelo a escala en donde este sistema sea capaz de reconocer donde se encuentra disponible un espacio de parqueo, una vez identificado este, el carro de forma autónoma debe realizar la maniobra de parqueo, sin ningún tipo de asistencia por parte del conductor.

Este sistema debe tomar en cuenta el control de la dirección del vehículo, el desplazamiento en cualquier dirección, al mismo tiempo debe ser capaz de reconocer la distancia que hay entre los objetos alrededor del vehículo, identificar cuando se encuentra al lado de un espacio lo suficiente para estacionar en forma paralela y por último debe ser capaz de evitar colisiones contra objetos u obstáculos que se le presenten al frente y detrás del vehículo y mantener una distancia mínima de seguridad contra estos.

Para lograrlo fue necesario el uso de un microprocesador para la lectura de las entradas y el control de las salidas, sensores ultrasónicos para medir la distancia del vehículo y los objetos alrededor, sensor de proximidad inductivo para medir el desplazamiento lineal del vehículo, un puente H para el control del motor de corriente directa y un servomotor para el control de la dirección del carro.

Keywords

Arduino; Inductive Proximity Sensor; H Bridge; Servo motor; Regulator and Ultrasonic Sensor.

Abstract

There is a problem for many car users which is to maneuver the vehicle is in parking either battery or in parallel, it is given that many people take a long time to park, generates stress and in some cases to them it is impossible to park, and those who fail can perform this maneuver so bad wasting space or leaving other vehicles with the inability to leave where they are parked. Because of this the Department of measurement - Aeronautical Instrumentation Laboratory of the Czech Technical University in Prague (Czech Technical University in Prague) has decided to make an automatic parking applied first to a scale model car where the system is able to recognize where is available parking space, once identified this, the car must perform autonomously parking maneuver without any assistance from the driver.

This system should take into account the directional control of the vehicle, the displacement in any direction, at the same time must be able to recognize the distance between the objects around the vehicle, identify when next to a space enough parking shaped battery and finally be able to avoid collisions with objects or obstacles that are presented in front and behind the vehicle and maintain a minimum safety distance against them.

To achieve this it was necessary to use a microprocessor for reading control inputs and outputs, ultrasonic sensors to measure the distance of the vehicle and objects around inductive proximity sensor for measuring the linear displacement of the vehicle, a bridge H for controlling the DC motor and a servomotor to control the carriage direction.

Introducción

Los conductores de vehículos automotores enfrentan a diario el reto de tener que estacionar sus vehículos entre dos objetos ya sea de forma paralela o batería (como se ilustra en la figura 1), y para esto se requiere de mucha práctica, agilidad y coordinación, y aun así, el resultado del estacionamiento no es siempre el mismo.

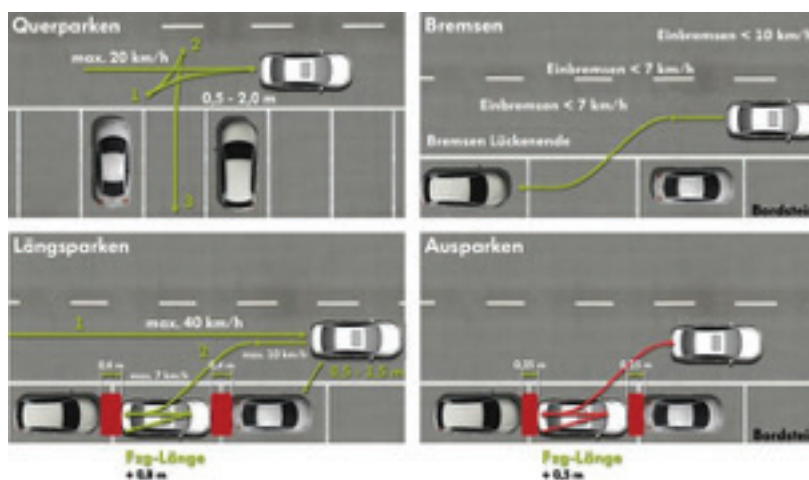


Figura 1. Ilustración de tipos de estacionamiento y maniobra para realizarlos, a la izquierda arriba estacionamiento en forma de batería y derecha arriba en forma paralela.

Es por esto que se cree en la necesidad de realizar un sistema automatizado que asista al conductor y le permita estacionar el vehículo sin la necesidad del usuario de intervenir. Aun no hace muchos años que llegaron los primeros sistemas a algunas marcas y aunque son cada vez más, no esta tan generalizado.

Este artículo aborda los detalles en el diseño de un sistema de parqueo automático de un carro modelo a escala en posición paralela, donde se describirá todos los elementos necesarios, así como los algoritmos de control, donde lo que se busca es realizar un algoritmo de control automático retroalimentado con sensores que le permitan reconocer un espacio libre de parqueo batería, y estacionar el vehículo de forma autónoma evitando la colisión con objetos durante la maniobra.

Diseño de solución

Para el desarrollo del diseño se hizo una segmentación según los objetivos planteados, en el cual se dividió en 5 partes, Control de desplazamiento, control de dirección, control para evasión de colisiones, reconocimiento de espacio libre de estacionamiento y Algoritmo de parqueo.

Control de desplazamiento

Para el control de desplazamiento se utilizó un sistema de retroalimentación el cual es un sensor inductivo de proximidad, se detectan los impulsos enviados por el sensor al girar las ruedas y su conteo sirve como medida de la distancia recorrida y su frecuencia como medida de la velocidad.

Para escoger el controlador del motor se debe asegurar que este sea capaz de entregar la tensión y la corriente demandada por el motor para romper la inercia, así como la capacidad de invertir el sentido de giro, ya que este es el momento que el motor exige mayor potencia, a la vez debe tener un sistema de entrada que nos permita controlar la entrega de tensión al motor y así poder definir la potencia que se entrega al motor, para efectos de este artículo se hará referencia de un controlador que tiene como entrada una señal de PWM para regular la entrega de tensión, así como señales que nos permiten definir si el motor gira en sentido de las agujas del reloj o en contra.

Una vez esto, se seleccionó el contador de pulsos que el sensor inductivo emite. Se debe implementar un integrado que nos permita reconocer cuando estos pulsos se dan y los almacene en un valor digital binario, para lo cual se recomienda un contador asíncrono ascendente, este nos permite contar los pulsos recibidos del sensor sin necesidad de un reloj y también cuenta con una señal de control de reinicio.

El sistema de procesamiento de datos debe ser capaz de controlar e interpretar los sensores, de manera que pueda interpretar los valores obtenidos de los sensores, pasarlos a unidades métricas conocidas para así ser ingresados a algún algoritmo de control que pueda generar la salida al sistema que evite una colisión para el caso del sensor ultrasónico o de un desplazamiento determinado para el sensor inductivo de proximidad.

Para la implementación de la solución se va a utilizar un sistema embebido, (puede ser el Arduino ADK) dado que este nos permita realizar todas las operaciones deseadas, se ocupa un control de muchas entradas/salidas digitales, así como el uso de bibliotecas que nos facilitan el control de servomotores y la bibliotecas PID que nos permite realizar el control automático tanto del desplazamiento como de reducción de velocidad ante obstáculos [1].

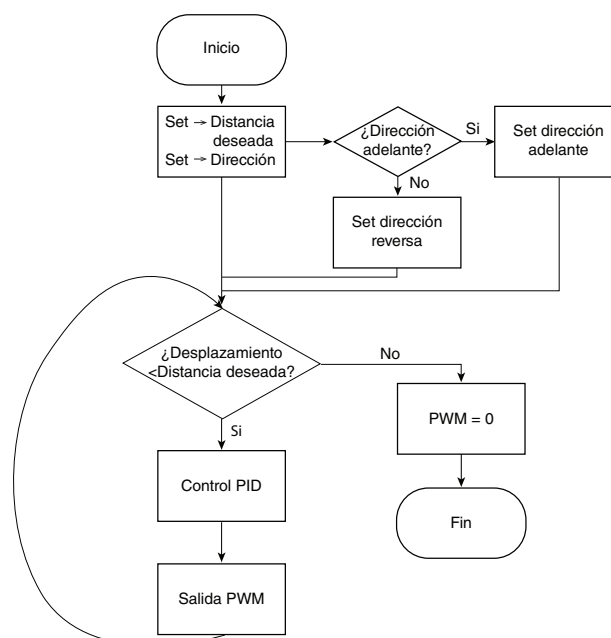


Figura 2. Algoritmo para desplazamiento del carro modelo a escala [2].

Como se puede observar en el algoritmo (figura 2) al inicio se le define una dirección de desplazamiento y una distancia deseada por desplazarse en esa dirección, una vez esto, el algoritmo se encarga de reconocer a que dirección debe moverse, y se colocan los valores lógicos en las entradas del puente H para que defina esta dirección, colocando en alto o bajo (según sea la dirección deseada) en las señales de control. Dado esto se lleva la aplicación del control automático que nos permite conocer cuánto debe ser la tensión entregada al controlador del motor para lograr este desplazamiento mediante el ajuste del valor de PWM [3].

Para el control del desplazamiento lo primero que debemos hacer es un modelo matemático de la planta, esto consiste en obtener la velocidad del motor CD con respecto a la tensión de entrada, que luego esta tensión de entrada se convierte en una relación de PWM por lo que al final se convierte en un función de transferencia de m/PWM, para esto lo primero que se hace es conocer el modelo teórico que debemos esperar de la planta. Una vez esto se debe de obtener el modelo experimental que represente está función en nuestro sistema.

La lectura del valor de entrada (PWM), el tiempo (en segundos) y la salida (en metros) son leídos por el sistema embebido (Arduino ADK por ejemplo). Estos valores deben ser llevados a una herramienta matemática como MATLAB, que nos permita con estos datos generar el modelo matemático de la planta, y una vez obtenido esto buscar el control automático que sea necesario para poder controlar el desplazamiento del vehículo [2].

Para el control automático de la planta se debe buscar un modelo que nos permita controlar dicha planta mediante tipos de control clásicos (PD, PI o PID), escogiendo el que mejor se ajuste y nos permita tener el menor tiempo de estabilización y un valor en estado estacionario de 1, obteniendo algo como lo ilustrado en la figura 3 [4].

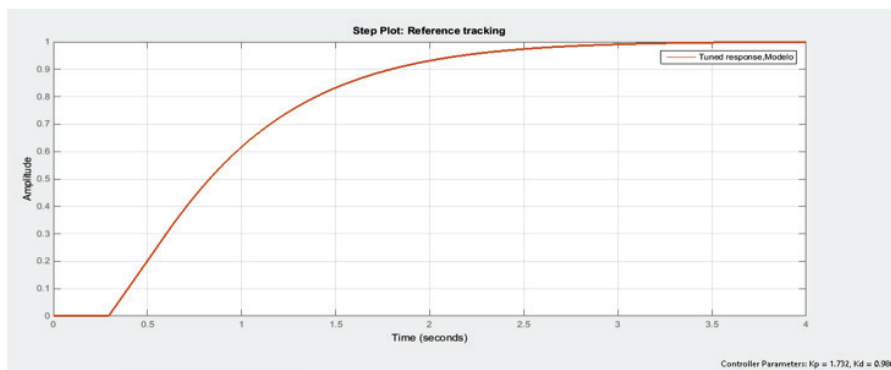


Figura 3. Respuesta al escalón del sistema con control PD [2].

Conociendo los valores de k_p , k_i y k_d de nuestro controlador se incorpora este regulador junto con la planta. Se debe llevar simulaciones con alguna herramienta que nos permita conocer cual sera el comportamiento de la planta ante este control implementado. Para esto se utiliza una vez más la herramienta de MATLAB simulink.

Control de Dirección

Para el control de la dirección se implementa dentro del sistema embebido un algoritmo que controla el servomotor de modelismo donde se indica un ángulo deseado del motor y gira el eje del motor hasta este valor, dicho algoritmo se ilustra en la figura 4. Para el ajuste del ángulo de dirección se utilizó como entrada un valor definido por el usuario para corregir la dirección del vehículo, una vez hecho esto se le entrega el valor al procesador de datos.

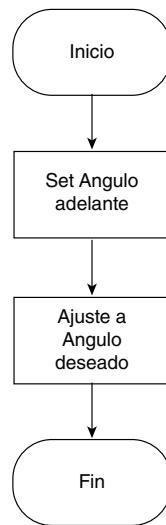


Figura 4. Algoritmo de control de dirección [2].

Control para evasión de colisiones

Cuando se está realizando la maniobra de parqueo debemos procurar no colisionar contra objetos o vehículos alrededor durante las maniobras de parqueo, por lo tanto para esto debemos realizar una búsqueda de la distancia de los objetos aledaños a nuestro vehículo y cuando este se encuentre cerca de chocar contra alguno de estos que detenga el avance y se aleje de este. Para lograr esto con el uso del sensor ultrasónico se definió el algoritmo mostrado en la figura 5, que nos permite hacer que el carro ante la posible colisión contra un objeto este se aleje de este y al mismo tiempo que no choque contra otro que se encuentre al frente de este, como el caso cuando se encuentra entre dos carros.

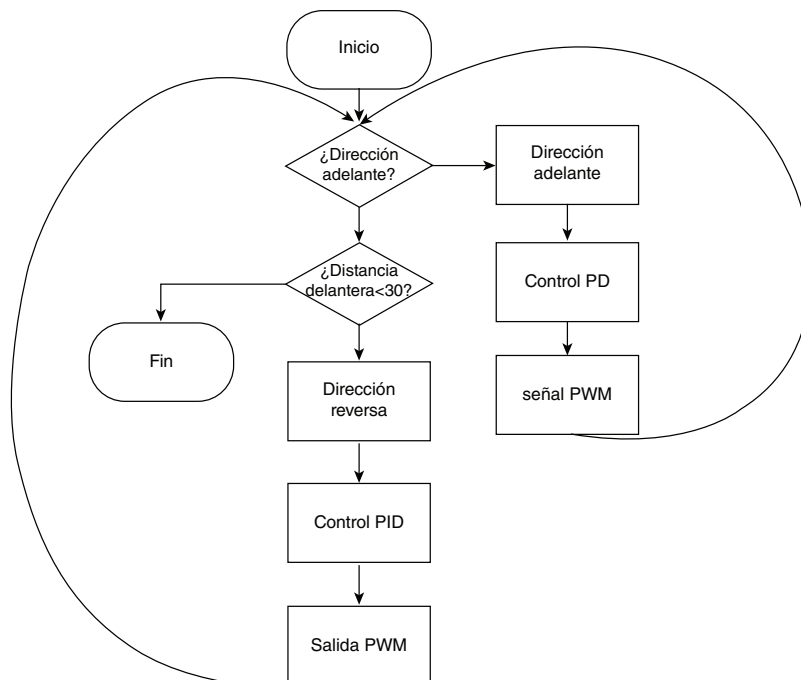


Figura 5. Algoritmo para evitar colisiones durante maniobras de parqueo [2].

Primero se desarrolla un modelo analítico basado en un análisis teórico del comportamiento de la planta, el mismo que ya se explicó en la sección de control de desplazamiento.

Para el control de evasión de colisiones se emplea un sistema de realimentación a lazo cerrado, se crea un control capaz de que si hay un cambio en la distancia, se detecte como una perturbación la cual debe ser corregida por el sistema de control diseñado, lo cual deja un esquema de la siguiente manera (figura 6) [2]:

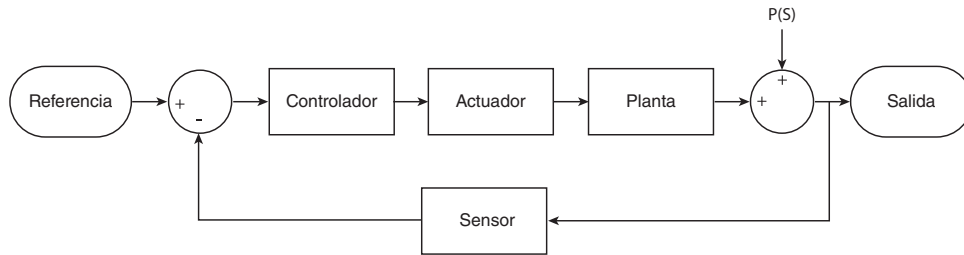


Figura 6. Diagrama de bloques del sistema de control de colisión [2].

Reconocimiento de espacio libre de estacionamiento

Para la implementación de este algoritmo se usan sensores ultrasónicos colocados a los extremos del carro modelo a escala como entrada al sistema, y como salida se tiene el desplazamiento mediante el uso del motor CD controlado por el puente H, donde la lógica utilizada para implementar esto se observa en la figura 7.

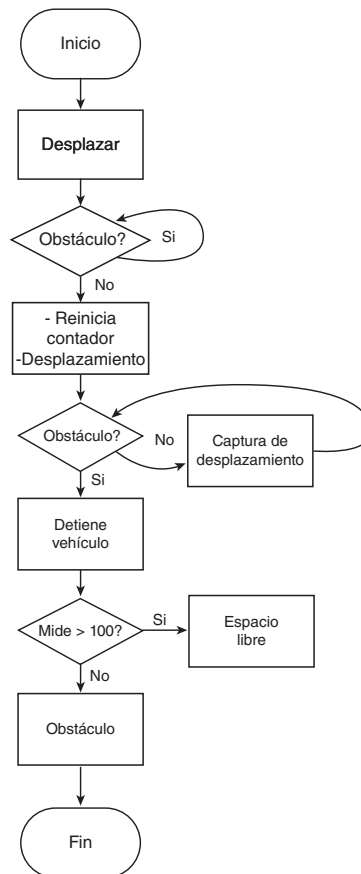


Figura 7. Algoritmo para el reconocimiento de espacios libres de parqueo [2].

En este algoritmo se inicia reconociendo donde termina el primer objeto al lado del vehículo, una vez esto, se inicia la cuenta del desplazamiento hasta encontrar el siguiente objeto delante de este, si la distancia entre estos dos objetos es mayor o igual a una distancia determinada significa que es posible realizar la maniobra de parqueo.

La distancia mínima requerida para parquear se conoce mediante el desplazamiento del vehículo de manera manual hasta reconocer cuanto es el mínimo espacio que requería este carro para poder colocarse entre dos objetos, después de varias iteraciones nos lleva al valor referente.

Algoritmo de parqueo

Cuando se tienen los diferentes módulos necesarios para realizar la maniobra de parqueo solo nos queda generar un último algoritmo que llame a estos de forma correcta cuando sea necesario para alcanzar el objetivo de estacionar el vehículo sin colisiones, para esto se sigue el flujo que se muestra en la figura 8, donde el rombo “Espacio libre” llama al algoritmo que nos permite identificar si hay espacio suficiente para realizar la maniobra de parqueo, una vez que se tiene identificado el espacio, se debe empezar la maniobra de retroceso, (se debe aclarar que dado que el control de dirección es a lazo abierto no hay una retroalimentación por lo tanto este algoritmo que realiza el retroceso sigue una serie de cambios en el ángulos de la llantas, el cual se llega a este mediante la iteración), conjuntamente con las maniobras de retroceso se debe correr el algoritmo que nos indica si el vehículo se encuentra contra un objeto a colisionar, en el que como se mencionó en la sección “Control para evasión de colisiones” se ve que una vez alcanzada la distancia en reversa contra el objeto que se encuentra en la parte atrás del espacio de parqueo, este se desplaza de forma lineal hacia adelante y se acomoda de forma que quede a una distancia definida contra el objeto delante del espacio de parqueo [2]. (Chaves Rodriguez, 2015)

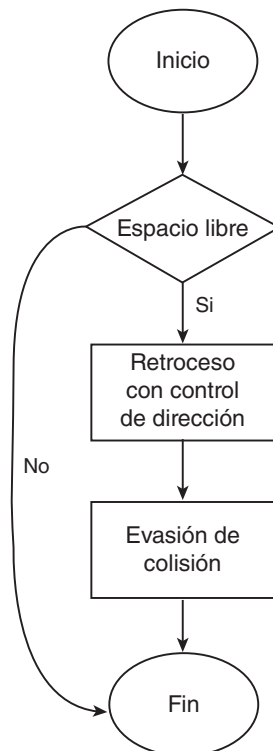


Figura 8. Diagrama de parqueo automático de carro a control remoto [2].

Conclusiones

El diseño descrito anteriormente es una propuesta a un sistema de control para el parqueo automático de un carro modelo a escala donde se muestra que con el uso del control clásico podemos lograr un parqueo automático donde la necesidad de sensores son solo dos. El procesamiento de datos es llevado por un sistema embebido con poca demanda de potencia pero que nos permite aplicar la manipulación de los datos recibidos por los sensores y las señales de salida. El control de la dirección sin una retroalimentación no nos permite saber exactamente hacia donde se mueve el vehículo con exactitud, por lo tanto es recomendable usar un sistema de retroalimentación de la dirección ya que si las condiciones son fuera de las esperadas en el espacio de parqueo el algoritmo de parqueo puede fallar.

Referencias

- [1] Arduino. (10 de Junio de 2015). *Arduino*. Obtenido de <http://www.arduino.cc/en/Main/ArduinoBoardMegaADK?from=Main.ArduinoBoardADK>
- [2] Chaves Rodriguez, J. P. (2015). *Parqueo automático de un carro a control remoto*. Cartago: Instituto Tecnológico de Costa Rica.
- [3] Chen, H.-C. (2013). *An H-bridge driver using gate bias for DC motor control*. Taiwan: IEEE.
- [4] Ogata, K. (2010). *Ingeniería de Control Moderno*. Madrid: PEARSON EDUCATION, S.A.