

Simulación de un Control de Trayectoria para un Vehículo Guiado Autónomamente Tipo Triciclo con Tracción Delantera

Simulation of a trajectory control for an automated guided tricycle type vehicle with front wheel drive

Por: Kevin Miramontes Escobedo orcid.org/0000-0001-6638-448X
 Julio César Solano Vargas orcid.org/0000-0003-4745-0175
 Juan Manuel Barrera Fernández orcid.org/0000-0002-2168-3408
 Centro de Ingeniería y Desarrollo Industrial (CIDESI) sede Querétaro

Dirección electrónica del autor de correspondencia:
kmiramontes@posgrado.cidesi.edu.mx

Recibido: 26 de febrero de 2021
 Aceptado: 25 de mayo de 2021

Cómo citar: Miramontes, K., Solano, J., & Barrera, J. (septiembre, 2021). Simulación de un control de trayectoria para un vehículo guiado autónomamente tipo triciclo con tracción delantera. *Universo de la Tecnológica*, 2(38), 15-20

RESUMEN: Se presenta la primera etapa de una nueva metodología de generación de trayectorias basada en vectores para Vehículos Guiados Autónomamente (AGV's). Con el objetivo de minimizar el uso de diversos sensores y reducir el procesamiento de cómputo es posible recrear rutas de entornos operativos mediante el diseño de vectores. Empleando Matlab se crearon trayectorias con N vectores (de magnitudes y ángulos diversos). Se calculó la función de transferencia del motor que gobierna la dirección del AGV y, se implementó un control Proporcional-Integral-Derivativo (PID) para que el motor ejecutase las trayectorias, cada vector representa una referencia (setpoint) que el control debe lograr. Con el control PID, el AGV siguió efectivamente las trayectorias generadas, dando certeza de que cualquier trayectoria generada con este método puede ser seguida. El control PID abre las puertas para implementar algoritmos de control más sofisticados y evaluar condiciones más complejas.

PALABRAS CLAVE: Generación de trayectorias basada en vectores, Control de trayectorias, Vehículos guiados autónomamente (AGV's), Control Proporcional-Integral-Derivativo (PID), Sistemas de control.

ABSTRACT: The first stage is presented of a new methodology of trajectories generation based on vectors for Automated Guided Vehicles (AGV's). The aim is to minimize the use of various sensors and reduce computational processing. It is possible to recreate operative environments trajectories by vector designs. Trajectories were created with N vectors, with different magnitudes and angles, by using Matlab. The engine transfer function that governs the direction of the AGV was calculated and a Proportional-Integral-Derivative (PID) controller was implemented for the engine to execute the trajectories. Each vector represents a reference (setpoint) that the control must achieve. With the PID control, the AGV effectively followed the generated trajectories giving certainty of any generated trajectory with this method can be followed. The PID control allows to implement more sophisticated control algorithms and evaluate more complex conditions.

KEY WORDS: Trajectories generation based on vectors, trajectory control, automated guided vehicles (AGV's), Proportional-Integral-Derivative (PID) control, control systems.

Introducción

Los vehículos guiados autónomamente (AGV's por sus siglas en inglés), realizan tareas sin la necesidad de un conductor, dentro de su desarrollo el seguimiento y la planeación de trayectorias cubren la necesidad del desplazamiento cinemático. La problemática principal de estos AGV's se encuentra en los métodos de generación y seguimiento de trayectorias que requieren de un procesamiento computacional elevado y la implementación de diversos sensores (Ferguson & Stentz, 2006). En 2017 Mercy et al. (Mercy, Pipeleers, & Parys, 2017) presentaron los algoritmos SPLINE y B-SPLINE que procuran calcular la trayectoria más corta y la aproximación a destino más suavizada posible a partir de puntos específicos en el área de trabajo de un AGV.

En 2017, Qing et al. y en 2019, Zheng et al. (Zheng, Xu, & Zheng, 2019; Qing, Zheng, & Yue, 2017) propusieron los algoritmos A* (A-star) y Dijkstra (D*) que calculan la trayectoria más corta desde un punto inicial a uno final. En 2019, Heinemann y colaboradores (Heinemann, Riedel, & Lechler, 2019) plantearon el algoritmo VFH* (Vector Field Histogram) que evade obstáculos usando sensores infrarrojos o ultrasónicos. En 2020, Wu et al. (Wu, et al., 2020) propusieron el algoritmo SVM (Support Vector Machine) para establecer trayectorias de un punto inicial a uno final en áreas con obstáculos.

El presente trabajo es la primera fase para el desarrollo de una nueva metodología de generación de trayectorias basadas en vectores que contribuya a disminuir el procesamiento computacional elevado y el uso (relativamente) excesivo de sensores; con el objetivo de realizar un seguimiento de trayectorias aplicando teoría de vectores y la implementación de algoritmos de control, y con esto hacer posible la creación de trayectorias para cualquier área de trabajo.

Materiales y Métodos

Generación de trayectorias: A partir de la evaluación de un recorrido deseado, con una trayectoria pretendida, se puede determinar la cantidad de vectores requeridos, así como sus correspondientes distancias y ángulos. Esto

permite trazar una ruta con características particulares, diseñar una trayectoria basada en vectores que se ordena en una tabla de vectores generalizada, misma que el AGV ejecuta secuencialmente para cumplir la trayectoria deseada. La Tabla 1 muestra el caso específico para la generación con vectores de una trayectoria de $0^\circ - 90^\circ$, en donde DV es la dimensión del vector, θ es el ángulo de dirección y k un valor constante de magnitud.

Tabla 1
Vectores para trayectoria de $0^\circ - 90^\circ$

$0^\circ - 90^\circ$	
DV	θ
k	0
k	4.5
k	9
⋮	⋮
k	85.5
k	90

La cantidad de vectores a generar para la experimentación depende de lo suavizada que se quiera la trayectoria, se observó que se requerían 20 vectores para obtener una buena apreciación de la ruta y fueron suficientes para demostrar el objetivo del experimento. Por otro lado, el ángulo de desviación de cada vector es equivalente a la cantidad de vectores generados para poder alcanzar los grados totales.

Simulación de trayectorias propuestas: Posterior a la creación de las tablas de trayectorias propuestas, estas se procesan en Matlab, lo que permite obtener una clara e intuitiva visualización de la trayectoria que se pretende seguirá el AGV. La Figura 1 muestra dicho ejercicio para el caso específico de la trayectoria $0^\circ - 90^\circ$ (ver Tabla 1).

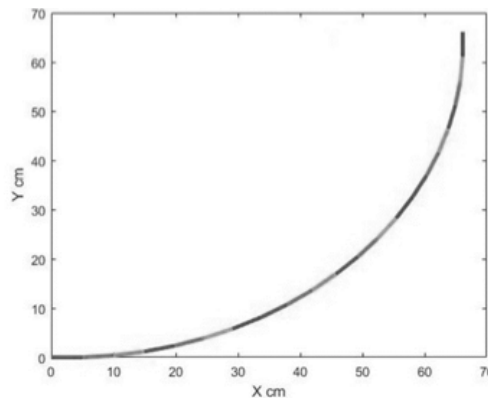


Figura 1. Visualización de trayectoria de $0^\circ - 90^\circ$

Haciendo uso de la herramienta Simulink de Matlab, se puede evaluar el comportamiento del control para las trayectorias. La secuencia de vectores que conforman alguna trayectoria es ingresada a Simulink para graficarse y en consecuencia obtener una referencia para considerar que el control (seguimiento) de trayectoria se logra óptimamente. La gráfica de la Figura 2 muestra el caso de la trayectoria $0^\circ - 90^\circ$ a modo de escalones (steps).

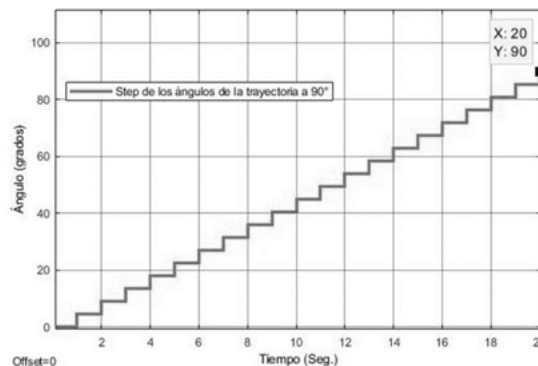


Figura 2. Escalones correspondientes a trayectoria de $0^\circ - 90^\circ$

Función de transferencia de la dirección del AGV: Se realizó el análisis matemático de un servomotor electrónicamente conmutado, para el análisis de control de posición, este es el que da la pauta para el control de la trayectoria. Por lo tanto, la función de transferencia para el control de posición está dada por la ecuación (1) (Fernández, 2019).

$$G_L(s) = \frac{\Omega(s)}{T_L(s)} = \frac{r_a + L_a s}{L_a j s^2 + (r_a j + L_a B_v) s + (r_a B_v + k_e k_t)} \tag{1}$$

Donde $G_L(s)$ es la Función de transferencia, $\Omega(s)$ es la Velocidad de rotación angular (salida), $T_L(s)$ es el Par de carga (entrada), r_a es la Resistencia de devanado, L_a es la Inductancia del devanado, j es el Momento de inercia del motor, B_v es el coeficiente de fricción viscosa, k_e es el coeficiente de la fuerza electromotriz y k_t es el coeficiente de carga.

Diseño de control PID: Se utilizó la herramienta *autotune* de Matlab, cuya función se basa en analizar con base a una entrada unitaria los tiempos de respuesta y de asentamiento del sistema en el dominio de la frecuencia y establece las ganancias mediante métodos empíricos de sintonización, en este caso el método de Ziegler-Nichols (Fernández, 2019). Utilizando la herramienta *Simulink* de Matlab, se genera el diagrama de bloques para la simulación del control PID. En la Figura 3 se muestra el diagrama de bloques, donde el bloque T contiene los datos de la trayectoria a controlar.

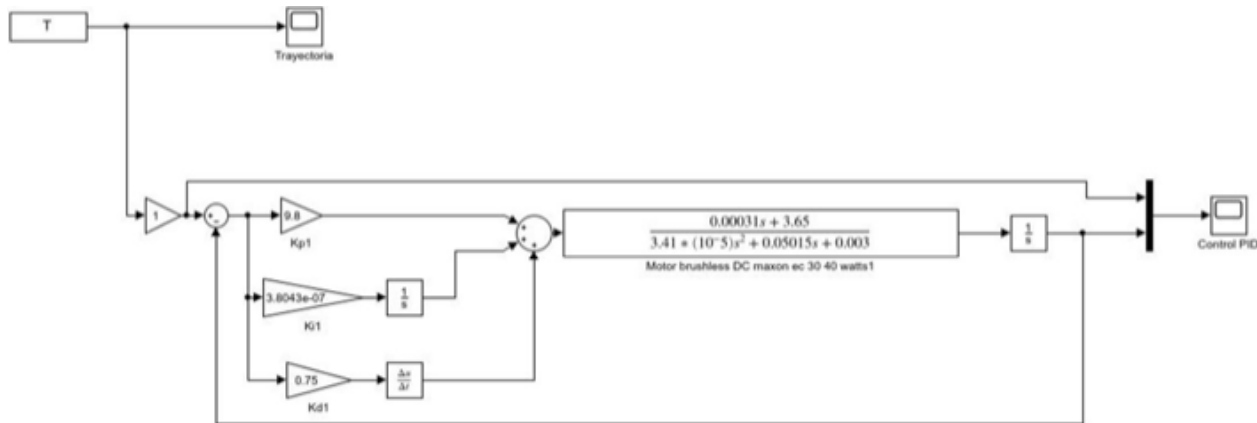


Figura 3. Control PID para el seguimiento de trayectorias

Diseño para la implementación de trayectorias generadas: La Figura 4 muestra el diseño para la implementación (en el AGV) de trayectorias generadas. Partiendo de una trayectoria generada, esta se envía a los controles de (motores) dirección y tracción y, los sensores retroalimentan la ejecución de las trayectorias, estableciendo un control óptimo de lazo cerrado para compensar el error.

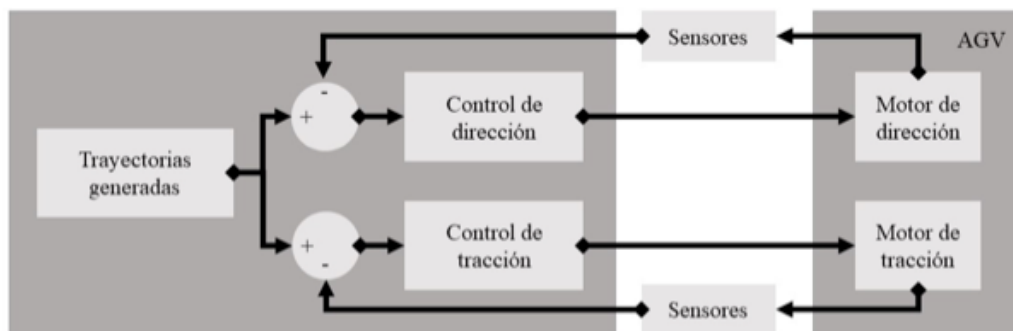


Figura 4. Diseño de control de trayectorias

Resultados

Se contemplaron cuatro trayectorias básicas con base a la generación de vectores, estas trayectorias son de 0° - 90°, de 0° - 180°, de 0° - 360° y en forma de S (véase la Tabla 2), a partir de la generación de las primeras tres trayectorias es posible generar otras que sean más complejas.

Tabla 2

Trayectorias propuestas a generar

0° - 90°		0° - 180°		0° - 360°		S	
DV	θ	DV	θ	DV	θ	DV	θ
<i>k</i>	0	<i>k</i>	0	<i>k</i>	0	<i>k</i>	0
<i>k</i>	4.5	<i>k</i>	9	<i>k</i>	18	<i>k</i>	9
<i>k</i>	9	<i>k</i>	18	<i>k</i>	36	<i>k</i>	18
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
<i>k</i>	85.5	<i>k</i>	171	<i>k</i>	342	<i>k</i>	9
<i>k</i>	90	<i>k</i>	180	<i>k</i>	360	<i>k</i>	0

Una vez que los datos generados en la Tabla 2 son ingresados en Matlab se grafican las trayectorias propuestas, con esto dando validez de que los datos de la tabla son correspondientes a cada una de las trayectorias deseadas. Se observó que, ingresando una cantidad de veinte vectores, las curvas en la trayectoria se generaban de manera más suave en comparación a generar una cantidad más baja. Por lo tanto, es necesario generar una cantidad mayor de vectores cuando se requiera generar una trayectoria con curvas. En la Figura 5 se muestran las trayectorias generadas en Matlab.

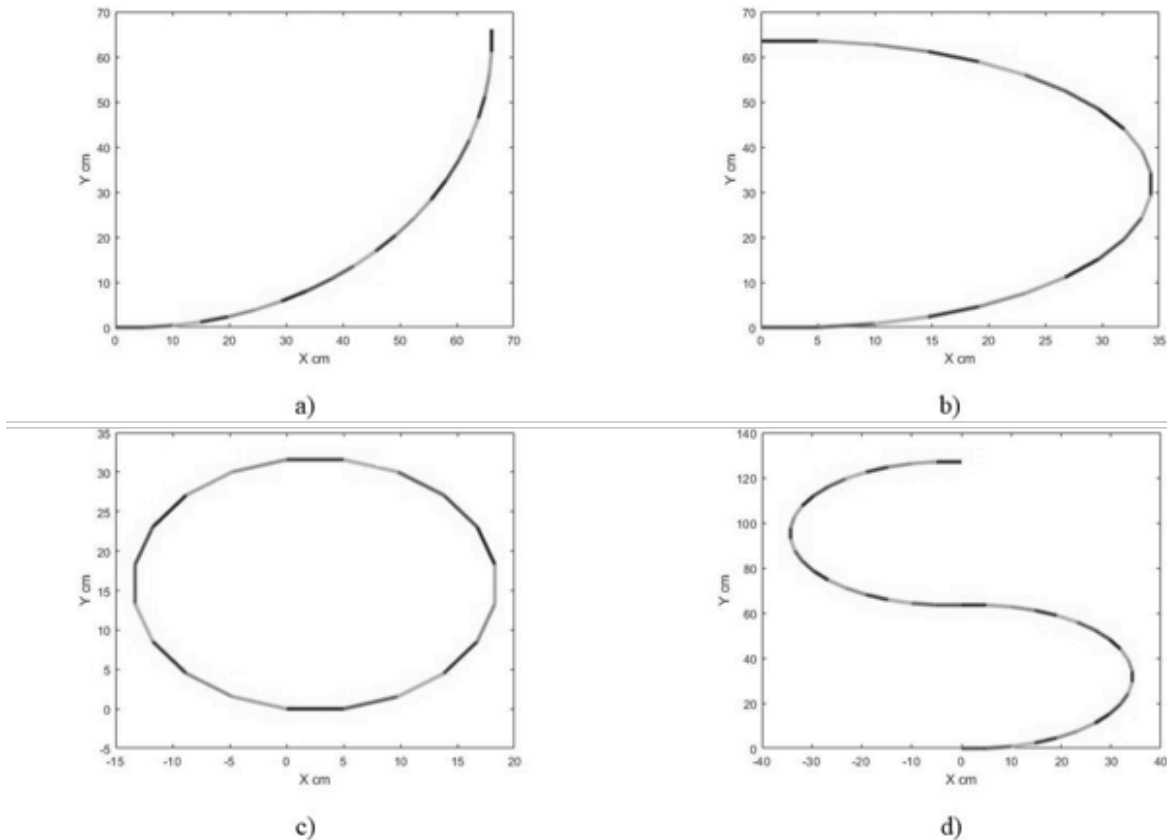


Figura 5. Trayectorias generadas: a) 0° - 90°, b) 0° - 180°, c) 0° - 360° y d) en S

Los valores correspondientes de las ganancias para el control PID obtenidos con la herramienta *autotune* de Matlab son, $k_p=9.8$, $k_i=3.8043^{-7}$, $k_d=0.75$. El resultado de la aplicación del control PID con estas ganancias aplicadas a cada trayectoria se aprecia en la Figura 6.

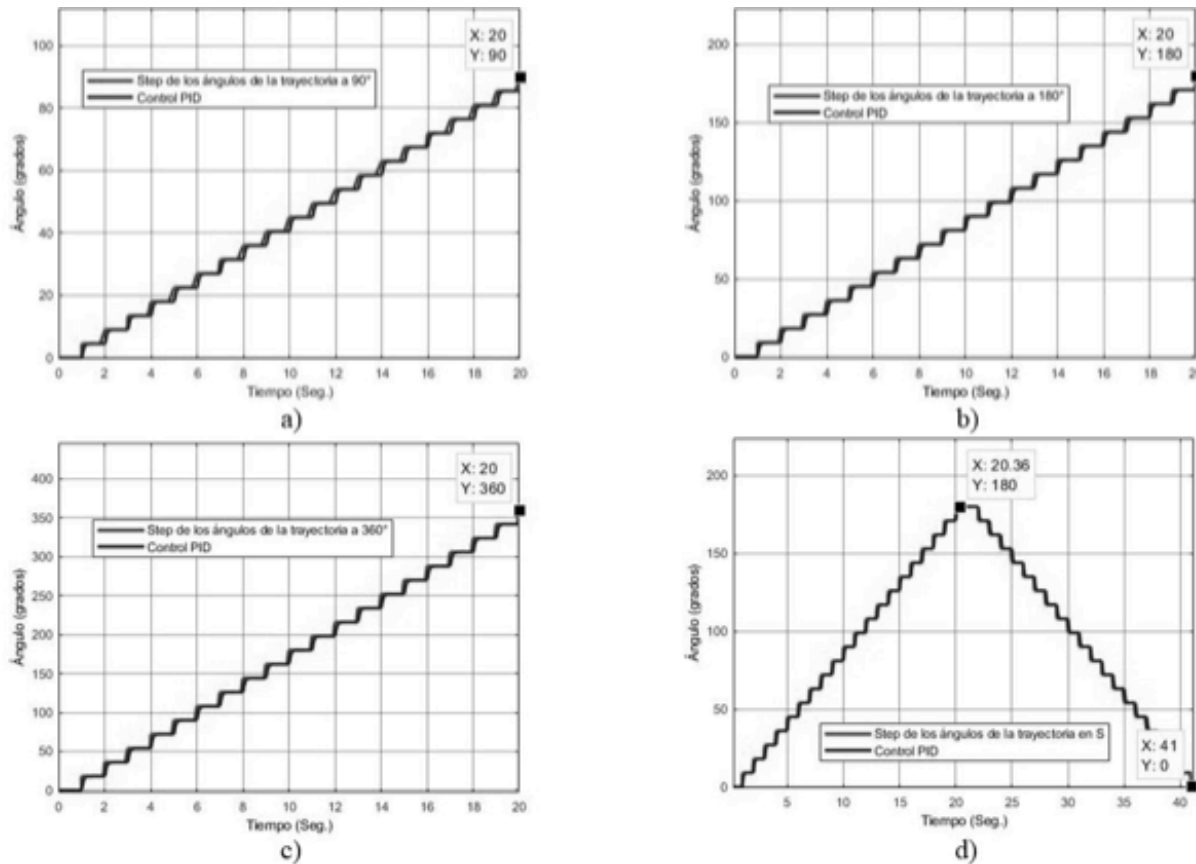


Figura 6. Control PID para trayectorias generadas: a) 0° - 90°, b) 0° - 180°, c) 0° - 360° y d) en S

Realizando el análisis del sistema a la señal de referencia con base a los parámetros descritos por Nise (2020), el control automático implementado otorga un tiempo de levantamiento (T_r) de 142 milisegundos, un tiempo de asentamiento (T_s) de 2.272 segundos y con un error en estado estable cercano al 0%. Esto indica que el AGV requerirá cerca de 2 segundos para alcanzar el cambio de dirección deseado (ver Figura 7).

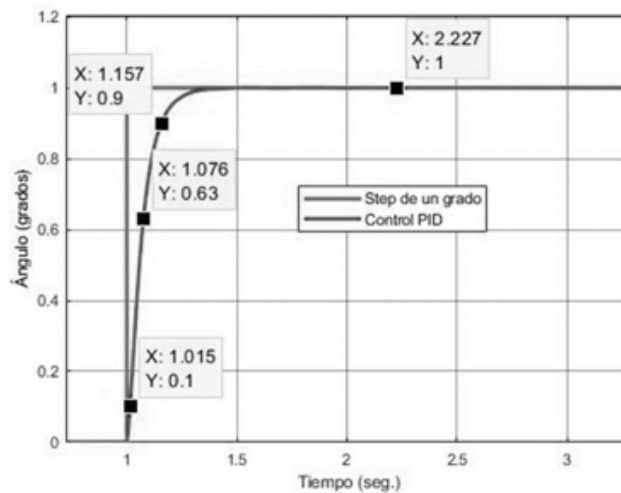


Figura 7. Respuesta transitoria del control PID propuesto

Conclusiones

La generación de trayectorias mediante vectores hace posible trazar cualquier ruta que se desee asignar al AGV, con ello, se da flexibilidad para la planificación y aplicación de estas trayectorias en diferentes condiciones de trabajo. Adicionalmente se comprobó que con la implementación de un control clásico como el PID se da un seguimiento fiel a la trayectoria propuesta, teniendo en cuenta que el tiempo mínimo de reacción del controlador es de 2 segundos, abriendo así también la posibilidad de aplicar algoritmos de control más sofisticados para satisfacer necesidades más demandantes.

Referencias

- Ferguson, D., & Stentz, A. (2006). Using interpolation to improve path planning: The Field D* algorithm. *Journal of Field Robotics*, 79-101.
- Fernández, J. M. (2019). Control de servomotores de baja tensión mediante la implementación de control adaptativo. Santiago de Querétaro: CIDESI.
- Heinemann, T., Riedel, O., & Lechler, A. (2019). Generating Smooth Trajectories in Local Path Planning for Automated Guided Vehicles in Production. *Procedia Manufacturing*, 98-105.
- Mercy, T., Pipeleers, G., & Parys, R. (2017). Spline-based motion planning for autonomous guided vehicles in a dynamic environment. *IEEE Transactions on Control Systems Technology*, 2182-2189.
- Nise, N. (2020). Control systems engineering. John Wiley & Sons.
- Qing, G., Zheng, Z., & Yue, X. (2017). Path-planning of Automated Guided Vehicle based on Improved Dijkstra. 2017 29th Chinese control and decision conference (CCDC), 7138-7143.
- Wu, X., Sun, C., Zou, T., Li, L., Wang, L., & Liu, H. (2020). SVM-based image partitioning for vision recognition of AGV guide paths under complex illumination conditions. *Robotics and Computer-Integrated Manufacturing*, 101856.
- Zheng, T., Xu, Y., & Zheng, D. (2019). AGV Path Planning based on Improved A-star. 2019 IEEE 3rd Advanced Information Management, Communicates, Electronic and Automation Control Conference (IM-CEC), 1534-1538.