

INFLUENCIA DE LA LATENCIA EN EL CONTROL DE AGVS A TRAVÉS DE REDES 5G

J. Enrique Sierra

Universidad de Burgos, ASTI Mobile Robotics, jesierra@ubu.es

Matilde Santos

Universidad Complutense de Madrid, msantos@ucm.es

Resumen

En este trabajo se analiza la influencia de la latencia en el control de un AGV (Vehículo de Guiado Automático) a través de una red 5G. Se describe la cinemática del vehículo y las variables de control del PID utilizado. Se utiliza como trayectoria de referencia una lemniscata. Se obtienen las ecuaciones del polinomio para calcular el punto de cruce de la trayectoria con el sensor de guiado. Finalmente se estima, mediante simulación, la máxima latencia permitida antes de que el AGV se salga de su trayectoria y se desestabilice para esa trayectoria bajo ciertas condiciones.

Palabras clave: Control en red, AGVs, 5G, simulación, latencia.

1 INTRODUCCIÓN

El acrónimo AGV proviene del término Vehículo de Guiado Automático. Estos son vehículos de transporte sin conductor empleados en la industria para automatizar los flujos intralógicos y las cadenas de producción, dotando a estos procesos de una gran flexibilidad espacial y temporal. Por ello desempeñan un papel clave en el nuevo paradigma establecido por la llegada de la industria 4.0.

Estas nuevas aplicaciones productivas demandan tecnologías de comunicación avanzadas. Por este motivo la industria 4.0 es uno de los verticales escogidos para experimentación dentro del proyecto europeo 5G-EVE (5G European Validation platform for Extensive trials) [1]. Este proyecto desarrollará e interconectará cuatro centros europeos de telecomunicaciones (localizados en Grecia, España, Francia e Italia) para crear una infraestructura 5G común de extremo a extremo. La infraestructura será empleada para experimentar y validar todas las nuevas capacidades ofrecidas por las comunicaciones 5G: ultra baja latencia, alta capacidad, alta disponibilidad, “slicing”, etc.

Uno de los casos de uso dentro del proyecto 5G-EVE es la virtualización de los algoritmos de control de AGVs. A día de hoy los AGVs tienen un PLC encargado de gobernar el lazo de control interno. Para ello recoge la información de los sensores de guiado, toma las decisiones de control apropiadas y genera las señales de consigna para controlar la velocidad de los motores. Actualmente no es posible desplazar el lazo de control interno fuera del AGV considerando las tecnologías de comunicación actuales. Gracias a las nuevas capacidades ofrecidas por el 5G, estos algoritmos de control internos podrán moverse fuera del AGV, pudiéndose reubicar en una máquina virtual. De esta forma, los beneficios de la virtualización del software también podrán ser explotados en este campo de aplicación.

La virtualización de sistemas ofrece, entre otras, varias ventajas como son:

- Redundancia: Ayuda a garantizar una alta disponibilidad.
- Ahorro: Se reduce el coste derivado del número de máquinas físicas, reduciendo también los costes de inversión en equipos, mantenimiento, consumo, etc.
- Escalabilidad: Una sola máquina física puede dar lugar a varias máquinas virtuales. Así se facilita el despliegue de las actualizaciones y el control de versiones, puesto que hacer un cambio en una máquina física permite hacer cambios en todas las máquinas virtuales que corran en dicha máquina física.
- Independencia del hardware: El funcionamiento del software puede ser independiente del hardware físico (PLC) en el que se esté ejecutando. Con esto también se logra independencia del proveedor de hardware.

Para que esto sea posible, las redes de comunicaciones deben garantizar una latencia baja y estable, además de una alta disponibilidad. La modificación de la red para ensayar con diferentes latencias conlleva una inversión de recursos importante, por lo tanto, antes de ejecutar los experimentos en una red real conviene validar la propuesta en un modelo de simulación. Para

ello en este trabajo se presenta un análisis del impacto de la latencia de la red en el AGV, y se obtiene la máxima latencia admisible teórica antes de desestabilizar el control del AGV para esa trayectoria concreta. Los resultados del estudio ayudarán a definir los límites de los experimentos a llevar a cabo en el proyecto 5G-EVE.

Otros trabajos anteriores estudian el efecto de la latencia en sistemas controlados en red. Por ejemplo, en [5] se encuentran las características de latencia y pérdidas de paquetes de red que garantizan la estabilidad de una planta lineal. Donkers propone un entorno de modelado basado en sistemas lineales conmutados [4]. En [6, 7 y 14] se presentan el estado del arte sobre metodologías de control de sistemas en red, tanto centralizados como distribuidos. En nuestro caso, a diferencia de [4, 5], el sistema a controlar es no lineal.

El principal objetivo de este trabajo es, pues, extraer el programa de un PLC real que está funcionando actualmente en un AGV y ejecutarlo en tiempo real en una máquina virtual que recibe y envía todas las señales al AGV físico. De esta forma todas las ventajas de la virtualización de sistemas anteriormente comentadas podrían trasladarse al ámbito de los AGVs. Actualmente en el mercado de los servicios cloud se habla de IaaS (Infraestructura como servicio), PaaS (plataforma como servicio), SaaS (software como servicio); este nuevo planteamiento favorecería la aparición del paradigma AaaS (AGVs como servicio). Para ello es fundamental estudiar la influencia de la latencia en el funcionamiento del AGV.

Para ello, en este trabajo se describe el entorno de simulación desarrollado para poder realizar experimentos teóricos con los que estudiar el efecto de la latencia en el control de AGVs. Se ha replicado la trayectoria que seguirán los AGVs en los experimentos reales que se llevarán a cabo en los laboratorios de 5TONIC [2] en el marco del mencionado proyecto 5G-EVE [1].

La estructura de esta contribución es la siguiente. En la sección 2 se describe el modelo de simulación empleado. Los resultados de las diferentes simulaciones son analizados en la sección 3. El documento finaliza con las conclusiones y líneas de trabajo futuras.

2 DESCRIPCIÓN DEL MODELO

2.1 AGV

Para la experimentación y demostración del funcionamiento de la propuesta se emplearán dos

AGVs modelo Easybot de la empresa ASTI Mobile Robotics [3] (Figura 1).



Figura 1: Modelo de AGV empleado

La cinemática del AGV Easybot es una mezcla entre un vehículo diferencial y un triciclo. La cabeza tractora es un robot diferencial, y el cuerpo está unido con la unidad de tracción a través de un eje sobre el que pivota, y éste se comporta como un triciclo. En este trabajo nos centraremos en estudiar el efecto de la latencia en el control de la cabeza tractora que es donde va ubicado el sensor de guiado.

Las ecuaciones del movimiento de la cabeza tractora vienen dadas por el modelo cinemático de un robot diferencial [10]:

$$\dot{x}_h = \frac{V_l + V_r}{2} \cos(\Phi_h) \quad (1)$$

$$\dot{y}_h = \frac{V_l + V_r}{2} \sin(\Phi_h) \quad (2)$$

$$\dot{\Phi}_h = \frac{V_r - V_l}{L_h} \quad (3)$$

Donde V_l es la velocidad longitudinal de la rueda izquierda, V_r es la velocidad longitudinal de la rueda derecha [m/s], x_h , y_h y Φ_h son las coordenadas X e Y del centro de la cabeza tractora y la orientación en un sistema de coordenadas inercial [m] y [rad], y L_h es la distancia entre las ruedas [m].

Para controlar el movimiento de la cabeza tractora se emplean las señales de control V_c y w_c . V_c es la velocidad de traslación de consigna [m/s] y w_c es la velocidad angular de consigna [rad/s]. A partir de estas señales de control se calculan las velocidades de consigna de las ruedas V_{l_c} y V_{r_c} [m/s]:

$$V_{l_c} = \frac{2V_c - w_c \cdot L_h}{2} \quad (4)$$

$$V_{r_c} = \frac{2V_c + w_c \cdot L_h}{2} \quad (5)$$

El sistema de guiado en el Easybot informa de la desviación entre el AGV y un camino preestablecido en el entorno de trabajo. Pueden emplearse diferentes tipos de sensores: ópticos (para seguir una línea pintada en el suelo), magnéticos (para seguir una banda magnética), inductivos (para seguir un cable

enterrado en el suelo), etc. En este caso el Easybot viene equipado con un sensor magnético (Figura 2), que proporciona la señal de error de guiado, err_{gui} .

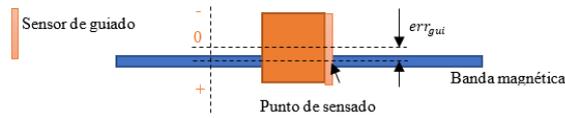


Figura 2: Sensor de guiado en el AGV

La velocidad de traslación de referencia, V_c , viene dada por la aplicación mientras que la velocidad angular de consigna, w_c , se ajusta para corregir la desviación respecto de la banda:

$$w_c = K_P \cdot err_{gui} + K_D \cdot \dot{err}_{gui} + K_I \cdot \int err_{gui} dt \quad (6)$$

El AGV viene equipado con un PLC esclavo a bordo. Este PLC recoge cada 10ms la información de los sensores y las entradas físicas, y lo envía a un PLC maestro ubicado en una máquina virtual. El PLC maestro, tras una latencia τ_{ME} , procesa toda esta información, toma las decisiones de control y genera las señales de control de los motores. Estas señales son enviadas de nuevo a través de comunicaciones 5G al PLC esclavo, el cual las procesa τ_{EM} más tarde traduciéndolas a señales físicas reales para los motores (Figura 3). Se supone despreciable el tiempo de procesamiento en los PLCs respecto de las latencias.

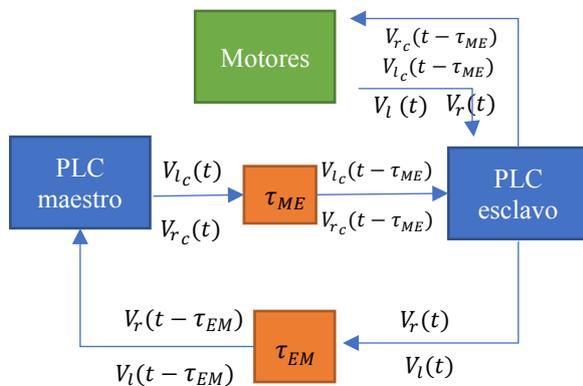


Figura 3: Arquitectura de PLCs

2.2 ENTORNO

Además de los AGVs, en el entorno se dispone de una estación de carga que permite al AGV estar en funcionamiento de forma ininterrumpida. Mientras se están cargando los AGV no pueden moverse pero sí realizar otras acciones como enviar y recibir comunicaciones, esperar condiciones, etc. Además hay una caja de control de tráfico que evita que los AGVs se bloqueen. Estos elementos pueden observarse en el diagrama esquemático del circuito de la figura 4.

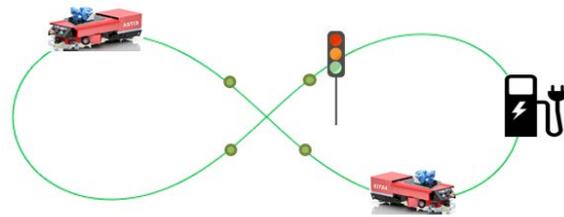


Figura 4: Trazado del circuito con sus elementos

La trayectoria descrita por el AGV es una lemniscata de Bernoulli [13]. Se define como el lugar geométrico de los puntos P tales que el producto de su distancia a dos focos, F1 y F2, situados a una distancia de $2a$ entre sí, es constante y vale a^2 . La ecuación (7) describe esta curva en coordenadas cartesianas excluidas la traslación y rotación:

$$(x^2 + y^2)^2 - 2a^2(x^2 - y^2) \quad (7)$$

Para calcular el error de guiado se calculan los puntos de corte de la recta descrita por el sensor de guiado y la lemniscata. Para ello se sustituye la ecuación de una recta ($y = mx + b$) dada por una pendiente m y un punto de corte en el eje de coordenadas, b , en la ecuación (7) y se obtiene el polinomio de orden cuatro dado por (8):

$$k_4x^4 + k_3x^3 + k_2x^2 + k_1x + k_0 = 0 \quad (8)$$

Donde las constantes k_1 a k_4 vienen dadas por las expresiones:

$$k_4 = m^4 + 2m^2 + 1 \quad (9)$$

$$k_3 = 4m^3b + 4mb \quad (10)$$

$$k_2 = b^2(2 + 6m^2) - 2a^2(1 + m^2) \quad (11)$$

$$k_1 = 4m(b^3 + a^2b) \quad (12)$$

$$k_0 = b^4 + 2a^2b^2 \quad (13)$$

Existen varios métodos para calcular las raíces de un polinomio. En este caso se realiza a través del cálculo de los valores propios de la matriz compañera [8].

3 RESULTADOS

En esta sección se describen los resultados obtenidos mediante la ejecución de simulaciones. Para las simulaciones se ha empleado Python 3.7 [12] con las librerías *numpy* [9] y *pylab* [11]. La tabla 1 recoge los parámetros empleados en las simulaciones.

Tabla 1: Parámetros empleados en la simulación

	Descripción	Valor
a	Distancia de lemniscata	2.5
L_h	Distancia entre ruedas	0.4 m

La figura 5 muestra el seguimiento de la trayectoria realizado por el robot cuando se fija V_c a 0.7m/s y no hay latencia en la red de comunicación, con $K_p=10$, $K_d=2$ y $K_i=0$. El punto verde representa el origen del robot, la línea en negro la trayectoria del robot y en rojo la curva lemniscata. Se puede observar que el AGV sigue bien la trayectoria de referencia, especialmente en la intersección de las curvas.

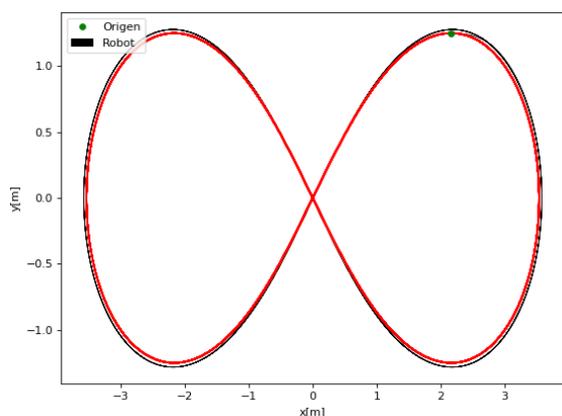


Figura 5: Seguimiento del AGV de la trayectoria

La figura 6 representa el error de guiado observado por el sensor. Se puede apreciar como el error es cíclico, pasa por cero cuando el robot pasa por el centro de la lemniscata y se hace máximo en sus extremos.

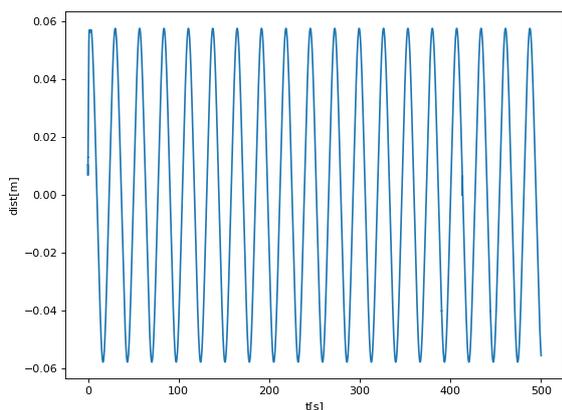


Figura 6: Error registrado en el sensor de guiado

Si se emplea un PID más agresivo el robot sigue con menor error la trayectoria deseada. En las figuras 7 y 8 se muestran los resultados con $K_p=100$, $K_d=20$ y $K_i=0$. El problema, que se verá más adelante, es que

aumentar las constantes del PID hace el sistema más sensible a la latencia en la red.

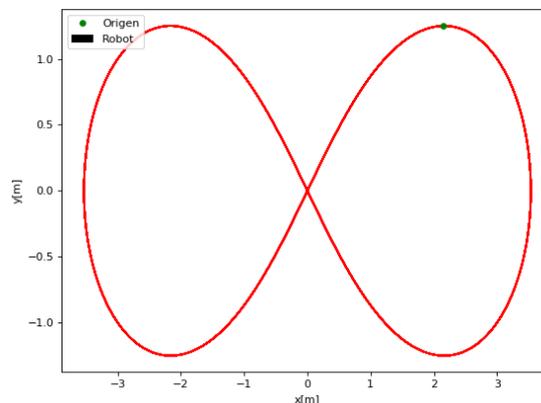


Figura 7: Seguimiento del AGV de la trayectoria con un regulador PID más agresivo

En este caso (figura 8), el error se ha reducido en magnitud.

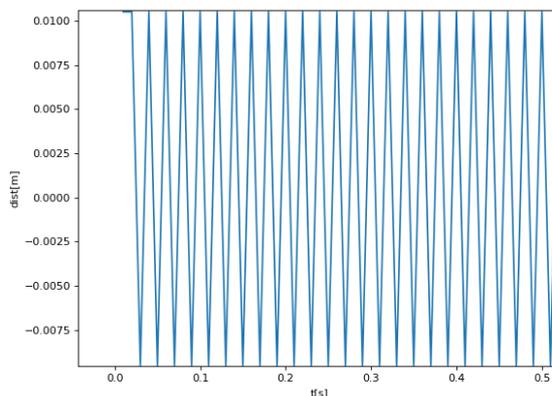


Figura 8: Error registrado en el sensor de guiado con regulador PID más agresivo

Conforme la latencia va aumentando el sistema oscila más y más hasta que llega un punto en el que se desestabiliza, el sensor no ve la banda y el vehículo se detiene. Las figuras 9 y 10 muestran un ejemplo en el cual el AGV se sale de la banda y se detiene. En este caso $V_c=0.6$ m/s y la latencia en el envío de información desde el PLC esclavo al PLC maestro es 160 ms. En estas gráficas puede observarse como la oscilación va en aumento conforme avanza el robot, señal de que el sistema se desestabiliza.

El valor de latencia para el cual el AGV se desestabiliza en esa trayectoria varía con la velocidad de consigna y las ganancias del control PID, K_p , K_d y K_i . La figura 11 representa el mínimo valor de latencia en la comunicación de PLC esclavo a PLC maestro que hace que el AGV pierda banda para diferentes velocidades V_c y diferentes parámetros del regulador PID. De forma general se puede apreciar cómo conforme aumenta V_c la latencia necesaria para que el

AGV se salga disminuye. También se observa cómo el empleo de un controlador PID más agresivo hace que estos valores disminuyan.

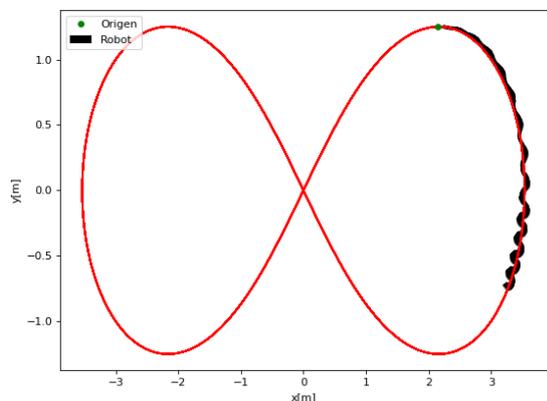


Figura 9: Seguimiento del AGV de la trayectoria con 160ms de latencia

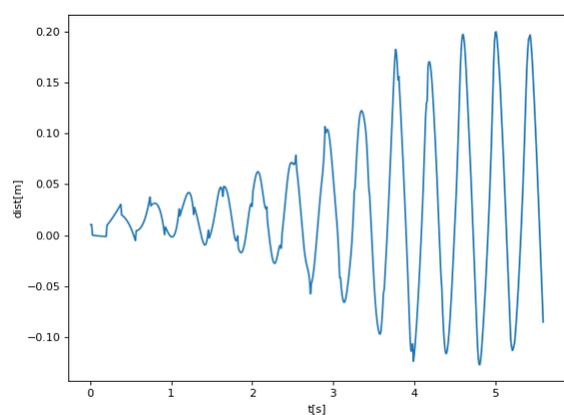


Figura 10: Error registrado en el sensor de guiado con una latencia mayor

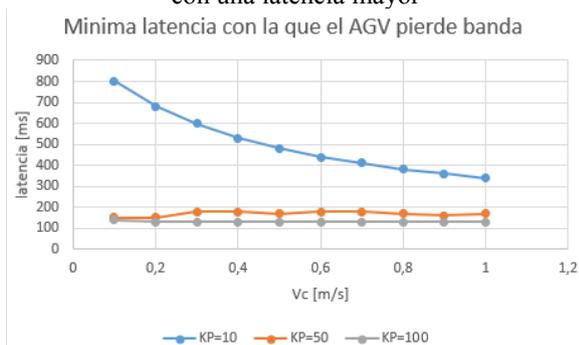


Figura 11: Mínima latencia esclavo-maestro para pérdida de banda

La figura 12 muestra la mínima latencia en el sentido PLC maestro a PLC esclavo que hace que el AGV se desestabilice. En general las conclusiones son similares a las que se observan en la figura 11, pero se obtienen valores numéricos más bajos.

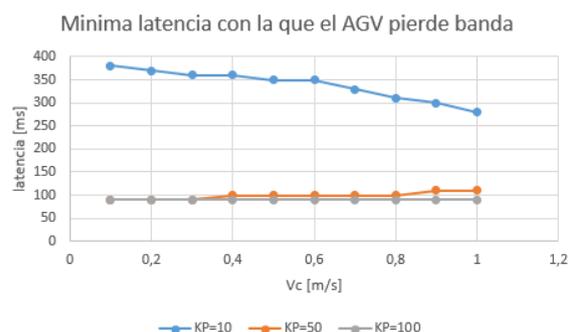


Figura 12: Mínima latencia maestro-esclavo para pérdida de banda

4 CONCLUSIONES Y TRABAJOS FUTUROS

En este trabajo se analiza la influencia de la latencia en el control de un AGV, controlado por un PID a través de una red 5G, para una trayectoria lemniscata. Se han desarrollado las ecuaciones del polinomio para calcular el punto de cruce de la trayectoria con el sensor de guiado y se obtiene mediante simulación la máxima latencia permitida antes de que el AGV pierda la banda y por lo tanto la estabilidad para el seguimiento de esa trayectoria concreta.

Como trabajos futuros cabe destacar, por un lado, la obtención de las condiciones de estabilidad generales del sistema a través del análisis de la estabilidad de Lyapunov. En este trabajo se ha estimado de forma experimental una latencia para la cual, dada la trayectoria y ley de control predefinidas, el AGV se desestabiliza en simulación, pero cabría estudiar la estabilidad en otras situaciones para obtener garantías.

Por otro lado, el trabajo se completaría con la comparación de los resultados obtenidos con los experimentos reales que se llevarán a cabo con los AGVs en el marco del proyecto 5G-EVE.

Agradecimientos

Este trabajo ha sido financiado en parte por el proyecto 5G-EVE [1] (European Commission's Horizon 2020), bajo el grant agreement 815074.

English summary

IMPACT OF LATENCY ON AGV CONTROL BY 5G NETWORKS

Abstract

In this work, a model to study the impact of the latency on an AGV controlled by 5G network is proposed. The

kinematic of the vehicle and the controlled variables are described. The trajectory drawn by the AGV is a lemniscate. The equations of the polynomial to compute the crossing point between the trajectory and the guiding sensor are obtained. Finally, the maximum permitted delay to avoid the AGV loses its circuit is found.

Keywords: Networked control, AGVs, 5G, simulation, latency.

Referencias

- [1] 5G-EVE 2019. <https://www.5g-eve.eu/concept/>.
- [2] 5TONIC 2019. <https://www.5tonic.org/>
- [3] ASTI Mobile Robotics (2019). <https://asti.es/es/agvs#easybots>.
- [4] Donkers, M. C. F., Heemels, W. P. M. H., Van de Wouw, N., & Hetel, L. (2011). Stability analysis of networked control systems using a switched linear systems approach. *IEEE Transactions on Automatic control*, 56(9), 2101-2115.
- [5] García-Rivera, M., & Barreiro, A. (2007). Analysis of networked control systems with drops and variable delays. *Automatica*, 43(12), 2054-2059.
- [6] Ge, X., Yang, F., & Han, Q. L. (2017). Distributed networked control systems: A brief overview. *Information Sciences*, 380, 117-131.
- [7] Guinaldo, M., Sanchez, J., Dormido, S. (2017). Event-based control for networked systems: From centralized to distributed approaches. *Revista Iberoamericana de Automática e Informática Industrial*, 14(1), 16-30.
- [8] Horn, R. A., & Johnson, C. R. (2012). *Matrix analysis*. Cambridge university press.
- [9] Numpy (2019). <https://www.numpy.org/>
- [10] Oriolo, G. (2019) Control of nonholonomic systems. https://www.dis.uniroma1.it/~oriolo/cns/cns_slides.pdf.
- [11] pylab (2019). <https://scipy.github.io/old-wiki/pages/PyLab>
- [12] Python 3.7 (2019). <https://www.python.org/downloads/release/python-on-370/>
- [13] Schappacher, N. (1997). Some milestones of lemniscatotomy. *Lecture Notes in Pure and Applied Mathematics*, 257-290.
- [13] Zhang, X. M., Han, Q. L., & Yu, X. (2015). Survey on recent advances in networked control systems. *IEEE Transactions on Industrial Informatics*, 12(5), 1740-1752.



© 2019 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution CC BY-NC-SA 4.0 license (<https://creativecommons.org/licenses/by-nc-sa/4.0/deed.es>).