




Mejoramiento de un servomotor inteligente de bajo costo Improvement of a low-cost intelligent servomotor

B. Sánchez-García ^{a,*}, E. Bugarin ^a, A. Y. Aguilar-Bustos ^a

^a Departamento de Ingeniería Eléctrica y Electrónica, Tecnológico Nacional de México / IT de Ensenada, 22780, Ensenada, Baja California, México.

Resumen

En este artículo se describe una mejora implementada al servomotor Dynamixel AX-12, un servomotor inteligente de bajo costo muy utilizado en el diseño y construcción de robots avanzados para entretenimiento e investigación. Se detalla la modificación de su software y hardware a fin de cambiar su protocolo de comunicación original por SPI y con esto lograr mayores velocidades de transmisión de datos entre servomotores del mismo tipo y sistemas de cómputo supervisores que operen en tiempo real. Adicionalmente, se le programa un controlador PD con prealimentación para regular tanto la velocidad como la posición angular de su eje de carga con un periodo de muestreo de 1 ms (10 veces mejor respecto a lo alcanzado con el servomotor original). Finalmente, se describe el comportamiento satisfactorio tanto de lo teórico como de lo experimental.

Palabras Clave: Robots avanzados, control de robots, tiempo real, servomotor inteligente de bajo costo.

Abstract

In this article is described an improvement implemented to the Dynamixel AX-12 servo motor, a low-cost intelligent servo motor widely used in the design and construction of advanced robots for entertainment and research. The modification of its software and hardware is detailed in order to change its original communication protocol to SPI and with this to achieve higher data transmission rates between servomotors of the same type and supervisory computer systems that operate in real time. Additionally, a PD controller with feedforward is programmed to regulate both the speed and the angular position of its load axis with a sampling period of 1 ms (10 times better than what was achieved with the original servomotor). Finally, the satisfactory behavior of both the theoretical and the experimental is described.

Keywords: Advanced robots, robot control, real-time, low-cost intelligent servomotor.

1. Introducción

Actualmente, la Robótica como disciplina que trata con el diseño, construcción y uso de robots, tiene un auge importante. Tanto así que hoy en día se pueden ver sus aplicaciones en procesos industriales (como robots soldadores), en hogares (como robots aspiradora) e incluso en la exploración interplanetaria. Siendo una disciplina tan amplia, una clasificación muy aceptada es designar a la Robótica en dos tipos (Sciavicco y Siciliano, 2009): la Robótica Industrial que se enfoca en el desarrollo y aplicación de robots en la industria (se dice que este tipo se encuentra en una edad madura por los desempeños sorprendentes de sus robots estudiados); y la Robótica Avanzada que se centra en el estudio de todos aquellos robots con alto grado de autonomía en ambientes regularmente no estructurados (como los robots de servicio y los robots de campo).

Dentro de la Robótica Avanzada, los robots con patas, o más específicamente, los robots humanoides han sido de gran interés en los últimos años. Estos robots suelen ser muy complejos y bastante costosos; sin embargo, se han desarrollado robots de este tipo a un bajo costo para aplicaciones de entretenimiento o de investigación. Uno de ellos puede ser construido con el kit Bioloid de la empresa coreana Robotis (el robot humanoide que se puede construir con este kit es comúnmente llamado robot Bioloid). En Thai y Paulishen (2011) se le realiza una evaluación a este kit para su uso en cursos de ingeniería en Robótica. Los servomotores usados por Robotis son los denominados Dynamixel y son en la actualidad ampliamente utilizados en robots avanzados para la investigación (Arena *et al.*, 2022; Zyhowski *et al.*, 2022;

*Autor para la correspondencia: al14760608@ite.edu.mx

Correo electrónico: al14760608@ite.edu.mx (Bruno Sánchez-García), ebugarin@ite.edu.mx (Eusebio Bugarin), aaguilar@ite.edu.mx (Ana Yaveni Aguilar-Bustos).

Historial del manuscrito: recibido el 28/03/2023, última versión-revisada recibida el 23/05/2023, aceptado el 02/06/2023, publicado el 11/09/2023. DOI: <https://doi.org/10.29057/icbi.v11iEspecial2.10852>



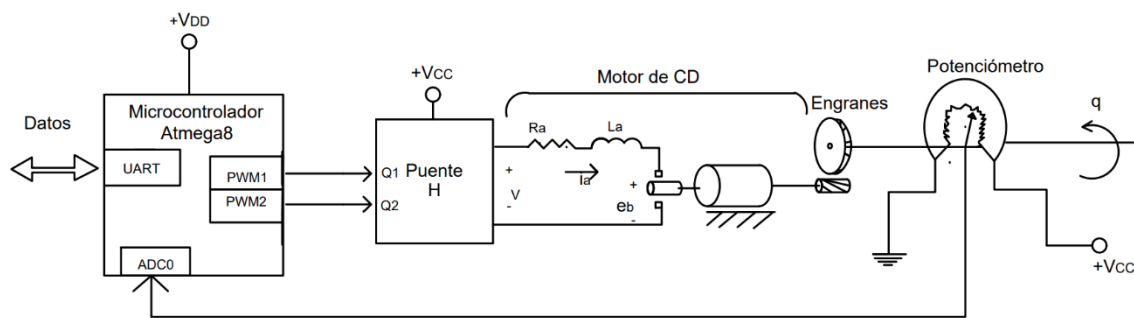


Figura 1: Diagrama a bloques del servomotor Dynamixel AX-12 original.

Shirai *et al.*, 2022; Kusnerova *et al.*, 2020, Iqbal *et al.*, 2020). Es importante mencionar que el robot Bioloid se construye con 18 servomotores Dynamixel AX-12.

El objeto de estudio del presente artículo es precisamente el servomotor Dynamixel AX-12, un servomotor inteligente de bajo costo; el cual, como modelo Dynamixel particular, también es ampliamente utilizado en la actualidad. Por ejemplo, en el área de la rehabilitación para accionar un prototipo de una silla de ruedas que utiliza un sensor novedoso que admite movimientos de manos y dedos como entrada para dirigir sus movimientos (Fereidouni *et al.*, 2022). En Singh *et al.* (2020) se utilizó para agregar un grado de libertad a un escáner 2D y poder realizar nubes de puntos en 3D; esto utilizando la información de la posición que ofrece el servomotor mismo. También se ha utilizado para mover las ruedas de un robot balancín diseñado para terrenos difíciles (Tsai *et al.*, 2022) y en robots del tipo araña con aplicaciones interesantes como la detección de minas en zonas desmilitarizadas (Oh y Kim, 2021).

De manera que es natural que estos servomotores Dynamixel AX-12 hayan sido ampliamente estudiados. En Bugarin *et al.* (2014) se les realizó un análisis tanto teórico como experimental obteniéndose su modelo y el valor de sus parámetros. En Zilong *et al.* (2015) se les utiliza como objeto de estudio para analizar actuadores con control de posición interno no modificable.

2. Problemática

Generalmente los robots avanzados requieren un número elevado de servomotores, de esta manera surge una problemática importante que tiene que ver con la velocidad de comunicación entre estos servomotores y probablemente con un sistema de cómputo supervisor. El Dynamixel AX-12 se comunica a través del protocolo de comunicación serial UART (Universal Asynchronous Receiver Transmitter) en modo semi-dúplex; es decir, envía y recibe información por un solo canal (y no al mismo tiempo, lo cual no es favorable para una comunicación de datos rápida; ver Apéndice A). Este protocolo está integrado al microcontrolador interno Atmega8 que posee el propio servomotor (de ahí lo de “inteligente”). De hecho, este protocolo se ha estudiado detalladamente a fin de obtener respuestas más rápidas en Bestmann *et al.* (2019). En ese trabajo se realizaron evaluaciones de varios métodos para mejorar la conexión entre servomotores y sensores, obteniendo como el mejor resultado una frecuencia de más de 1 KHz al

utilizar el bus RS-485. Además, también se menciona que se podrían lograr mejoras adicionales en las tasas de ciclo reduciendo el tiempo de respuesta de los servomotores o extendiendo el protocolo Dynamixel para reducir la cantidad de instrucciones por ciclo.

Por otro lado, el microcontrolador interno Atmega8 del servomotor AX-12 también cuenta con un protocolo de comunicación SPI (Serial Peripheral Interface); el cual puede configurarse en modo dúplex (lo que significa que puede enviar y recibir datos al mismo tiempo). No obstante, este protocolo no es utilizado en el servomotor; de manera que esto es lo que se aprovecha para realizar la mejora propuesta en este artículo. El protocolo SPI puede incrementar la velocidad máxima de transmisión de datos; la cual puede llegar hasta los 100 Mbps (a diferencia de la UART que sólo alcanza como máximo 20 Mbps).

Un estudio que se centró en el análisis de los retardos del Dynamixel AX-12 dio como resultado un consumo de 17 ms desde que el servomotor recibe el comando de posición deseada hasta que inicia el movimiento del eje del motor (Andrew-Smith y Jivraj, 2010).

Consecuentemente, este artículo detalla el mejoramiento del servomotor Dynamixel AX-12 en cuanto a su protocolo de comunicación, lo cual redundará en la obtención de mejores desempeños al utilizarse en robots avanzados con sistemas embebidos de tiempo real.

3. Mejoramiento del servomotor AX-12

3.1. Modificación del hardware del servomotor

El servomotor Dynamixel AX-12 básicamente se compone de un motor eléctrico de corriente directa (CD), una caja de engranes con relación reductora $1:r$ (donde $r = 254$) y un microcontrolador Atmega8; como lo muestra la Figura 1 (consultar Bugarin *et al.*, (2014) para el detalle de cada uno de los parámetros y variables de este servomotor).

Su etapa de potencia corresponde a un puente “H” que es accionado por dos salidas PWM del microcontrolador (una salida PWM para cada sentido de giro). Para medir la posición angular del eje de carga, un potenciómetro se encuentra conectado a la entrada ADC0 del microcontrolador. Y el envío de datos hacia y desde el servomotor se logra a través de un protocolo de comunicación serial UART TTL semi-dúplex a 1Mbps.

Para implementarle el protocolo de comunicación SPI y para poder programar el microcontrolador Atmega8 se conectaron 4 cables (RESET, MOSI, MISO y SCK; faltando SS) directamente a la placa electrónica del Dynamixel AX-12; como lo muestra la Figura 2.

Ahora bien, originalmente el Dynamixel AX-12 usa una salida PWM compartida con el pin SS (que se utiliza para la comunicación por SPI). Debido entonces a que el pin SS del Atmega8 se encuentra ligado (se comparte) a la salida PWM2 es necesario modificar dicha salida PWM; por lo que se sustituye la salida PWM2 por un pin digital (PD1) que es tomado directamente del microcontrolador Atmega8 (ver Figura 2).

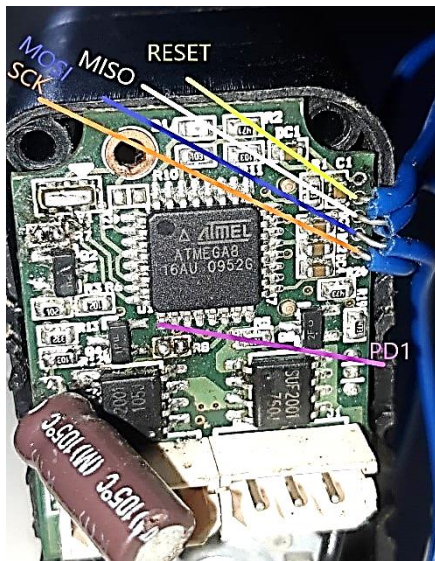


Figura 2: Cables agregados en la placa electrónica del servomotor Dynamixel AX-12.

En la Figura 3 se muestra el diagrama de bloques del Dynamixel AX-12 modificado. De acuerdo a este diagrama de bloques ahora se cuenta sólo con una salida PWM (PWM1), pero se diseña y construye un circuito combinacional (con dos compuertas AND y una NOT) para lograr el cambio de sentido de giro del eje del motor (utilizando el pin digital PD1). Este circuito combinacional tiene la tabla de verdad mostrada en la Tabla 1; donde puede observarse que de estar activa la señal PWM1, la salida digital PD1 indica si se activa Q1 (sentido antihorario) o si se activa Q2 (sentido horario).

PWM1	PD1	Q1	Q2
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	0

Tabla 1: Tabla de verdad utilizada para el diseño del circuito combinacional.

3.2. Modificación del software del servomotor

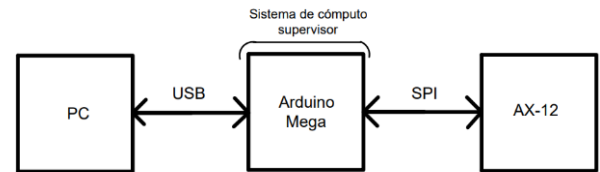


Figura 4: Arreglo para la experimentación.

Para programar el microcontrolador interno Atmega8 se utiliza el software Microchip Studio y el software AVRDUDESS como herramienta de grabación; con un Arduino UNO como programador.

Es importante mencionar que debido a que no estaban disponibles las fuentes del programa original se realizó un nuevo programa que toma en cuenta estas modificaciones de hardware y que usa un sistema de cómputo supervisor (un Arduino Mega) en comunicación con una PC (ver Figura 4) para el análisis de datos fuera de línea. De manera que se utilizó 1 reloj (timer) para la ejecución periódica (estrictamente cada 1 ms; es decir, se garantiza tiempo real) del controlador de movimiento a ejecutarse, el ADC0 para la lectura de la posición del eje de carga del servomotor, 1 señal de salida PWM para la etapa de potencia y comunicación ahora de datos flotantes por SPI entre el servomotor y el supervisor. El sistema de cómputo supervisor también se programó para operar en tiempo real (con relojes o “timers”) para el guardado de variables y su adecuado análisis posterior en la PC (con el envío de datos, después de experimentación, por USB).

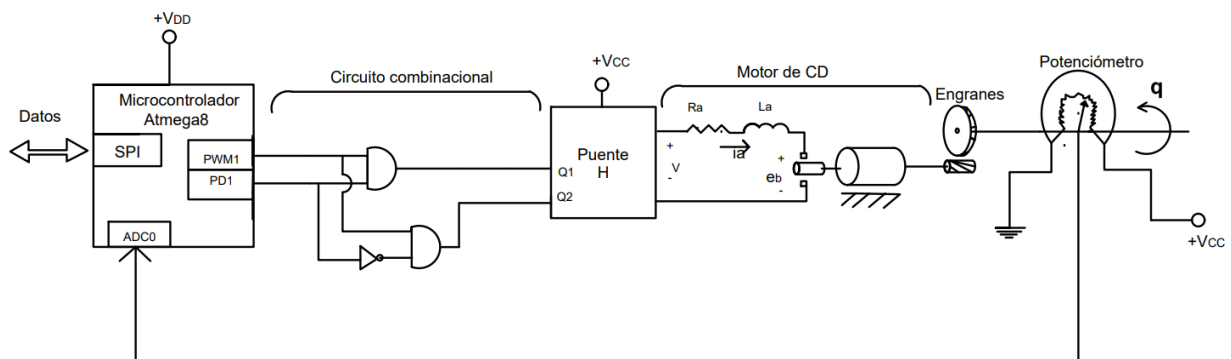


Figura 3: Diagrama a bloques del servomotor Dynamixel AX-12 modificado.

4. Implementación de un control PD con prealimentación y comparativas experimentales

4.1. Control PD con prealimentación

Para verificar lo sustancial del mejoramiento realizado al servomotor se detallan experimentos (tanto en el servomotor original como en el mejorado) utilizando el control PD con prealimentación propuesto por Bugarin *et al.*, (2014). En este sentido, se tiene que el modelo del conjunto motor y caja de engranes es¹

$$A\ddot{q} + B\dot{q} = v \quad (1)$$

donde $A = 0.07$, $B = 1.7505$, \dot{q} es la velocidad angular, \ddot{q} es la aceleración angular y v es el voltaje de armadura. El servomotor original tiene programado internamente (sin acceso a él) un controlador PD; esto es

$$v = k_p(q_d - q) + k_d \frac{d}{dt}(q_d - q) \quad (2)$$

donde $k_p = 80$, $k_d = 0.3$, q es la posición angular y q_d es la posición angular deseada. De manera que la ecuación de lazo cerrado del servomotor es

$$A\ddot{q} + B\dot{q} = k_p(q_d - q) + k_d \frac{d}{dt}(q_d - q) \quad (3)$$

con q_d la entrada al sistema y q la salida. Realizándole el cambio de variables " $q_d = u$ ", el controlador es

$$k_d \dot{\xi} + k_p \xi = A\ddot{q}_r + B\dot{q}_r \quad (4)$$

$$u = q_r + \xi \quad (5)$$

donde u es la nueva entrada del sistema, q_r la posición angular de referencia a seguir (con \dot{q}_r la velocidad de referencia y \ddot{q}_r

la aceleración) y ξ una variable de estado agregada por el controlador. Entonces este controlador (2), (4) y (5) es dinámico, corresponde con el controlador PD con prealimentación y está demostrado formalmente que cumple con el objetivo de llevar asintóticamente $q(t)$ a $q_r(t)$.

En el mismo artículo de Bugarin *et al.*, (2014) se menciona que $k_p = 80$ es demasiado grande, lo que en general demanda voltajes de armadura también grandes (saturándose frecuentemente este voltaje). De manera que, a modo de confirmar lo propuesto, un primer experimento es implementado en el servomotor mejorado con $K_p = 10$ y $K_d = 0.1$ (ganancias más pequeñas para no sobrepasar los límites del voltaje permitido, alrededor de 12 V); tanto para el controlador (2), PD sin prealimentación, como para el (2), (4) y (5), PD con prealimentación.

En la Figura 5 se puede observar la respuesta de la posición angular con respecto al tiempo usando estos dos controladores para una señal senoidal de referencia de 45 grados de amplitud a 1 Hz de frecuencia. Note que la respuesta obtenida utilizando el controlador con prealimentación sí alcanza la posición de referencia, a diferencia de la respuesta lograda con el controlador sin prealimentación que refleja un error de aproximadamente 15 grados en amplitud. En la Figura 6 se muestran los voltajes entregados al motor. Puede apreciarse que en ningún caso se llega a la saturación de dicha variable. Con este experimento se da evidencia de que la mejora es funcional, pues se logran respuestas bastante aceptables con un periodo de muestreo de 1 ms (tanto para el servomotor AX-12 como para el sistema supervisor).

4.2. Comparativas experimentales

Para verificar lo sustancial de la mejora, se realizó un experimento comparativo entre el servomotor mejorado y el original. El controlador utilizado será el PD con

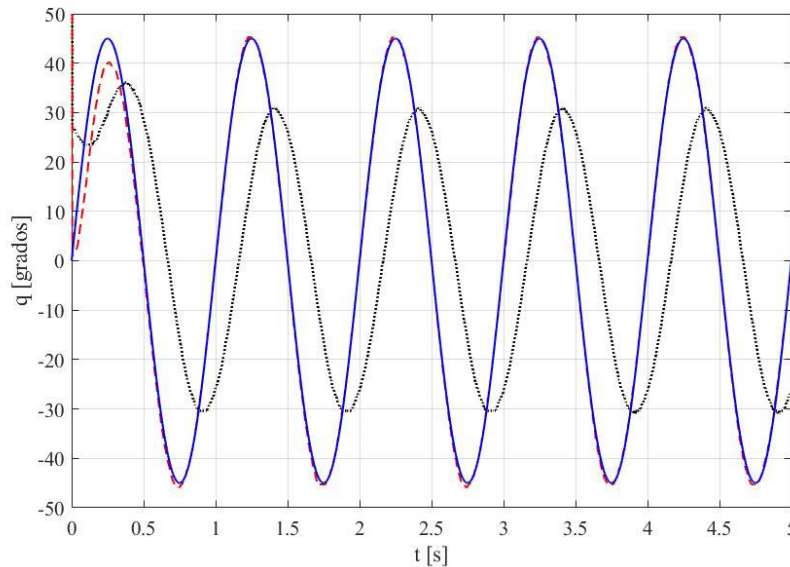


Figura 5: Gráfica de la posición angular contra el tiempo para el primer experimento (la línea continua azul muestra la posición angular de referencia, la línea punteada negra la posición angular con el controlador (2) y la línea discontinua roja la posición angular con el controlador (2), (4) y (5)).

¹ Es importante hacer notar que el modelo y el valor de sus parámetros también han sido tomados de Bugarin *et al.*, (2014).

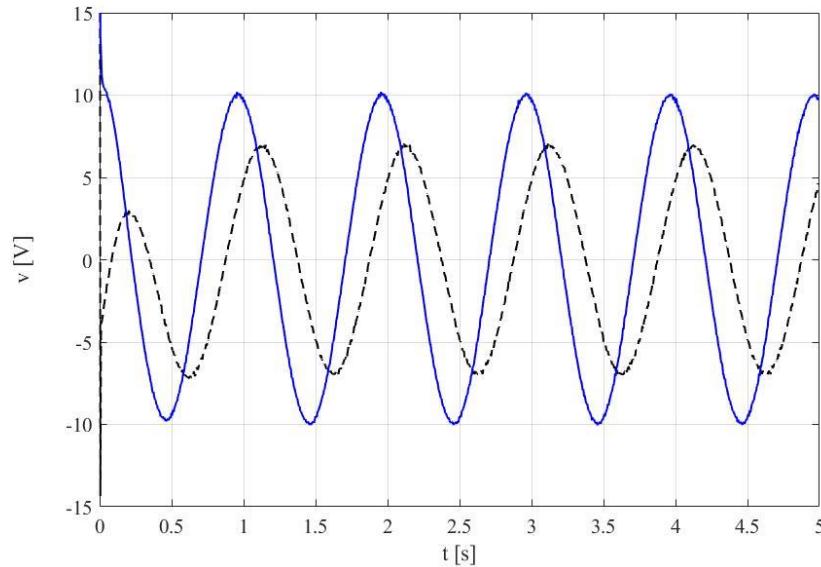


Figura 6: Gráfica del voltaje contra el tiempo entregado al motor en el primer experimento (la línea continua azul muestra el voltaje entregado con el controlador con prealimentación y la línea discontinua negra con el controlador sin prealimentación).

prealimentación con una señal de referencia más demandante: una señal senoidal de 5 grados de amplitud a 5 Hz de frecuencia. Para el caso del servomotor mejorado se respeta 1 ms para los periodos de muestreo tanto en el servomotor AX-12 como en el sistema supervisor. En tanto que, para el servomotor original, según experimentos preliminares (ver Apéndice A), se recomienda utilizar un periodo de muestreo de 10 ms en el sistema supervisor (más pequeño ese periodo de muestreo implica pérdida de datos en la comunicación bidireccional). Sin embargo, es importante destacar que el experimento con el servomotor original se realizó con un periodo de muestreo de 7.5 ms en el sistema supervisor (tratando de obtener la mejor respuesta con el servomotor original). Esto se puede experimentar así debido a que el controlador (2) está implementado dentro del servomotor y las lecturas de la posición actual del servomotor, al ser enviadas

por el puerto UART, utilizan un “buffer” que va acumulando las lecturas (lecturas que en la parte (4)-(5) del controlador no son necesarias).

En la Figura 7 se muestran las posiciones angulares respecto al tiempo obtenidas en este experimento comparativo. Como se puede observar, se tuvo una mejor respuesta por parte del Dynamixel AX-12 mejorado.

5. Conclusiones

Se ha detallado la mejora realizada al servomotor Dynamixel AX-12, un servomotor inteligente de bajo costo, para su utilización en el diseño y construcción de robots avanzados (robots no industriales, generalmente con aplicaciones de campo y de servicio, que requieren una autonomía más eficaz). Se le realizaron modificaciones tanto

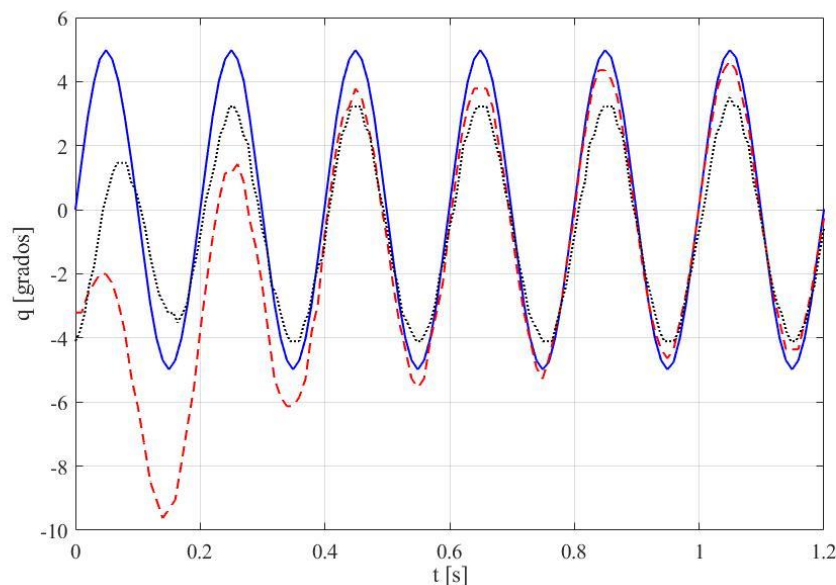


Figura 7: Gráfica de posición contra el tiempo para el experimento comparativo (la línea continua azul muestra la posición de referencia, la línea discontinua roja la posición del servomotor mejorado y la línea punteada negra la posición del servomotor original).

de hardware como de software para construir robots con sistemas embebidos de tiempo real, los cuales necesitan garantía de tiempos tanto en la transmisión de datos como en la ejecución de sus leyes de control de movimiento. Se llevaron a cabo pruebas experimentales que permiten valorar que la mejora ha sido sustancial, pues se le diseñó y se le validó satisfactoriamente un controlador de movimiento con una comunicación de datos flotantes de tiempo real hacia un sistema de computó supervisor que también operaba en tiempo real. Además, el servomotor Dynamixel AX-12 modificado fue reprogramado por completo, lo que implica que pueden implementársele diferentes leyes de control de movimiento o de fuerza (en modo voltaje, par, velocidad o posición) e incluso agregarle aditamentos como sensores necesarios para la adecuada ejecución de tareas de los robots avanzados actuales.

Agradecimientos

Los autores agradecen el apoyo financiero brindado por el Tecnológico Nacional de México y el Instituto Tecnológico de Ensenada.

Referencias

- Arena, P., Patane, L., Spinoso, A. G. (2022). A new embodied motor-neuron architecture. *IEEE Transactions on Control Systems Technology*, 30:5, pp. 2212-2219.
- Bestmann, M., Gildenstein, J., Zhang, J. (2019). High-Frequency Multi Bus Servo and Sensor Communication Using the Dynamixel Protocol. In: Chalup, S., Niemueller, T., Suthakorn, J., Williams, MA. (eds) *RoboCup 2019: Robot World Cup XXIII. RoboCup 2019. Lecture Notes in Computer Science*, vol 11531. Springer, Cham. DOI: https://doi.org/10.1007/978-3-030-35699-6_2
- Bugarin, E., Castañeda-García, L. J., y Aguilar-Bustos, A. Y. (2014). Experimental analysis of the dynamixel ax 12 servomotor and its wireless communication. In *Advances in computing science, control and communications*, pp. 37–46. Springer-Verlag Berlin Heidelberg.
- Fereidouni, S., Hassani, M. S., Talebi, A., Rezaie, A. H. (2022) A novel design and implementation of wheelchair navigation system using Leap Motion sensor, *Disability and Rehabilitation: Assistive Technology*, 17:4, 442-448, DOI: 10.1080/17483107.2020.1786734
- Iqbal, J., Xu, R., Halloran, H., Li, C. (2020). Development of a Multi-Purpose Autonomous Differential Drive Mobile Robot for Plant Phenotyping and Soil Sensing. *Electronics* 2020, 9, 1550; doi:10.3390/electronics9091550
- Kusnerova, M., Repka, M., Harnicarova, M., Valíček, J., Danel, R., Kmec, J., Palkova, Z. (2020). A new way of measuring the belt friction coefficient using a digital servomotor. *Measurement*. <https://doi.org/10.1016/j.measurement.2019.107100>
- Oh Y., Kim J. H. (2021). System Design and Implementation of Multi-legged Spider Robots for Landmine Detection in the Demilitarized Zone. 2021 18th International Conference on Ubiquitous Robots (UR). Gangneung, Korea (South), pp. 228-234. DOI: 10.1109/UR52253.2021.9494703.
- Sciavicco, L., Siciliano, B. (2009). *Robotics: Modelling, Planning and Control*. Springer-Verlag, London, 632 pp.
- Shirai, K., Shimamura, K., Koubara, A., Shigaki, S., Fujisawa, R., (2022). Development of a behavioral trajectory measurement system (Bucket-ANTAM) for organisms moving in a two-dimensional plane. *Artificial Life and Robotics* (2022) 27:698–705. <https://doi.org/10.1007/s10015-022-00811-5>
- Singh, R., Khurana, A. and Kumar, S. (2020). Optimized 3D laser point cloud reconstruction by gradient descent technique, *Industrial Robot*, Vol. 47 No. 3, pp. 409-421. DOI: <https://doi.org/10.1108/IR-12-2019-0244>
- Smith, J. A. y Jivraj, J. (2010). Analysis of robotis dynamixel ax-12+ actuator latencies.
- Thai, C. N. y Paulishen, M. (2011). Using robotis bioloid systems for instructional robotics. In 2011 Proceedings of IEEE Southeastcon, pp. 300–306. IEEE.
- Tsai C. C., Hsu W. T., Tai F. C., Chen S. C. (2022). Adaptive Motion Control of a Terrain-Adaptive Self-Balancing Leg-Wheeled Mobile Robot over Rough Terrain. 2022 International Automatic Control Conference (CACs), Kaohsiung, Taiwan, pp. 1-6, DOI: 10.1109/CACS55319.2022.9969857.
- Zilong, S., Gang, Z., Denis, E. (2015). Modelling and control of actuators with built-in position controller. *IFAC Papers Online* 48(11), pp. 837-842.
- Zyhowski, W.P., Zill, S.N., Szczecinski, N.S. (2022). Load Feedback from a Dynamically Scaled Robotic Model of *Carausius Morosus* Middle Leg. *Biomimetic and Biohybrid Systems. Living Machines 2022. Lecture Notes in Computer Science*, Springer.

Apéndice A. Experimentos preliminares

En este apéndice se describen algunos experimentos en lazo cerrado realizados con el servomotor Dynamixel AX-12 original. Como se ha mencionado este servomotor cuenta internamente con un controlador de posición PD. Ahora bien, para aplicaciones en robótica avanzada es deseable utilizar varios servomotores con lazos externos de control. De esta manera, para verificar su uso en este contexto, aquí se presentan experimentos (con sólo un servomotor) utilizando un controlador proporcional como lazo externo de control. Es decir, se desea controlar la posición y velocidad del eje de carga del servomotor utilizando su protocolo original de comunicación para retroalimentar la posición angular medida del eje de carga e inyectarle a la entrada del lazo interno de control la señal de un controlador proporcional como lazo externo de control. Lo anterior con auxilio de un Arduino Mega que se programa para operar en tiempo real y con esto obtener garantía de los tiempos de ejecución en los experimentos. Debe hacerse notar que esta estrategia no logrará el objetivo de control; sin embargo, si nos dará evidencia de su posible uso en este contexto.

En este sentido, la señal de entrada en los experimentos corresponde con una señal senoidal de 5 grados de amplitud y 5 Hz de frecuencia. La ganancia del controlador proporcional externo fue $k_p = 5$. Se realizan experimentos utilizando varios periodos de muestreo. Los resultados obtenidos se muestran en las figuras A.1, A.2, A.3 y A.4 para 50 ms, 20 ms, 10 ms y 5 ms; respectivamente. Las gráficas en azul corresponden con la señal senoidal deseada y en rojo con la posición medida. Tomando en consideración sólo la posible pérdida de datos (atribuida al protocolo de comunicación), se observa que esto no ocurre para el experimento de 50 ms. Sin embargo, resulta evidente que sí existe pérdida de datos para los experimentos de 20 ms y 10 ms. El caso de 5 ms debe ser observado con mayor detenimiento. Ciertamente también existe pérdida de datos en este último caso; sin embargo, el comportamiento que se observa es errático. Lo anterior es debido a que el protocolo UART de comunicación en el servomotor utiliza un “buffer” que va acumulando los datos del experimento (perdiendo datos en el caso de que sean demasiados; esto es, cuando existe desbordamiento), lo que puede provocar que no se procesen precisamente los datos correspondientes con cada periodo de muestreo (esto no permite operar en tiempo real a ciertos periodos de muestreo que pudieran considerarse rápidos para este protocolo de comunicación).

Finalmente, debido a que los periodos de muestreo de 10 ms y 20 ms presentaron respuestas un tanto aceptables, se realizaron 10 corridas para cada uno de ellos obteniéndose un promedio de 1.8 muestras perdidas por cada 1000 muestras en

el caso de 10 ms y 2.0 muestras perdidas por cada 1000 muestras para el caso de 20 ms (valores muy similares). De esta manera es que se recomienda, en el caso de necesitar procesamiento de tiempo real (no estricto, blando o suave) con retroalimentación, un periodo de muestreo de 10 ms.

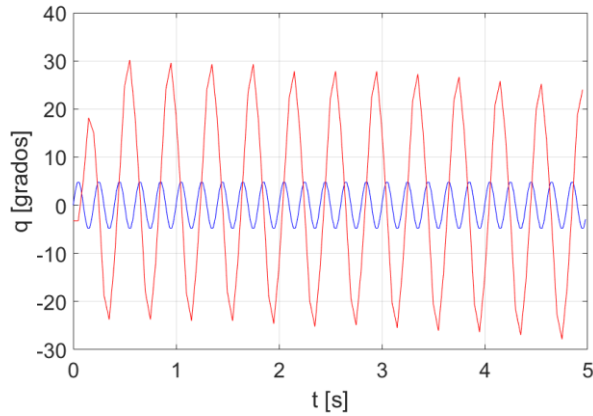


Figura A.1: Gráfica de posición contra el tiempo utilizando un periodo de muestreo de 50 ms.

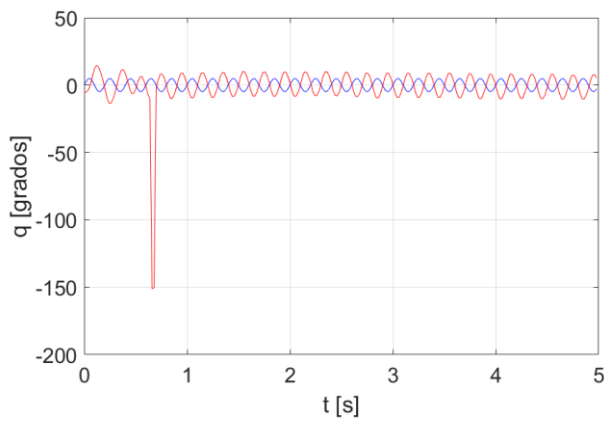


Figura A.2: Gráfica de posición contra el tiempo utilizando un periodo de muestreo de 20 ms.

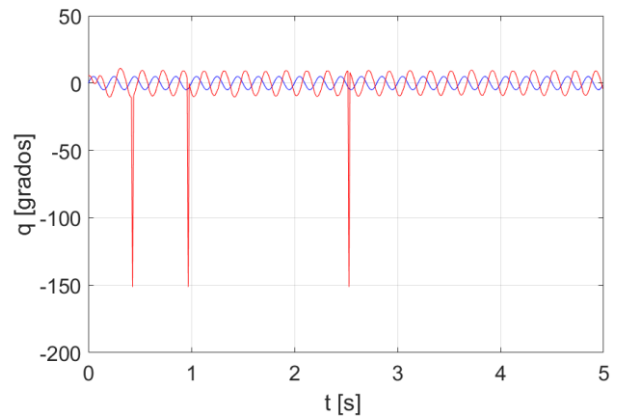


Figura A.3: Gráfica de posición contra el tiempo utilizando un periodo de muestreo de 10 ms.

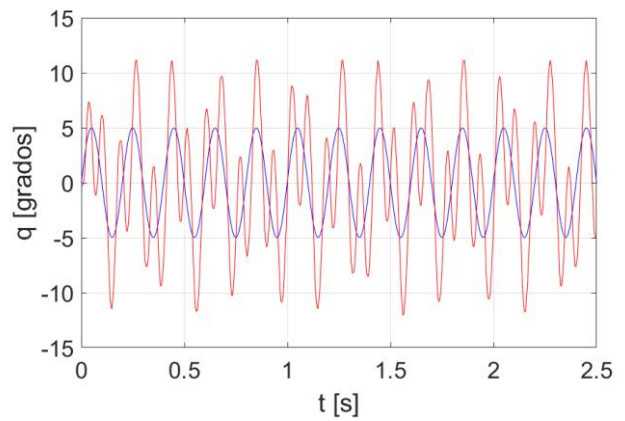


Figura A.4: Gráfica de posición contra el tiempo utilizando un periodo de muestreo de 5 ms.