

Evaluación automática de código para prácticas de modelado y control utilizando Matlab Grader

Yadira Boada^{a,b}, Lissette Boada Acosta^c, Alejandro Vignoni^a

^a Universitat Politècnica de València, Camí de Vera S/N, 46022, Valencia, España.

^b Centro Universitario EDEM, Escuela de Empresarios

La Marina de València, Muelle de la Aduana s/n, 46024, Valencia, España.

^c Centros Médicos Especializados Cruz Roja Ecuatoriana

Papallacta Oe 1 -66 entre Avenida 10 de Agosto y Prensa, Quito, Ecuador

Resumen

La calificación automática de código y prácticas informáticas ofrece varias ventajas prácticas y pedagógicas beneficiosas y en la actualidad es posible mediante una variedad de técnicas. Recientemente, MathWorks ha introducido una solución nativa para el código de MATLAB: MATLAB Grader. Esta interfaz basada en la web facilita el envío de tareas y la calificación automática basada en criterios preprogramados. Las asignaturas de Control Automático (DISA-ETSID-UPV) y Tecnologías Automáticas (CU-EDEM), se han utilizado como prueba en donde la calificación en clase de alguno de los laboratorios informáticos basados en MATLAB se ha sustituido por tareas calificadas automáticamente, principalmente utilizando la plataforma MATLAB Grader.

Palabras clave: calificación automática, prácticas informáticas, matlab grader, innovación docente.

1 INTRODUCCIÓN

La calificación automatizada de las tareas de programación en un entorno de educación superior es ya algo habitual, y se ha detallado ampliamente en la literatura [1, 5, 9] - el primer uso documentado hace unos 60 años[3]. Aunque existen herramientas similares para una amplia gama de aplicaciones (por ejemplo, para tareas de Matemáticas, circuitos eléctricos, informática, etc. [2, 10, 7, 4], las tareas de programación se prestan a la calificación automática inherentemente bien - particularmente en entornos donde se fomenta la modularidad del código y/o el desarrollo impulsado por pruebas. En este documento discutimos nuestras experiencias durante el último año implementando herramientas para la calificación automática de tareas de programación basadas en MATLAB en las asignaturas de Control Automático, del Grado en Ingeniería Aeroespacial, Escuela Técnica Superior del Diseño, en la Universidad Politécnica de Valencia y de Tecnologías de Automatización, del Grado de Ingeniería y Gestión Empresarial, Centro Universitario EDEM.

La calificación automática ofrece una serie de posibles beneficios prácticos y pedagógicos [1, 4] en comparación con el enfoque tradicional *en el laboratorio o enviar y devolver*:

- Un coste operativo reducido, que permite reasignar recursos y presupuesto para seguir apoyando otras actividades docentes.
- Una menor carga cognitiva para el personal, que se ve liberado de la inherentemente onerosa tarea de corregir el código informático.
- Calificación y retroalimentación consistentes y preprogramadas en cohortes independientemente de la cantidad de alumnos.
- Mayor accesibilidad a las evaluaciones, ya que los estudiantes pueden completarlas cuando les resulte conveniente y recibir la información en función de otros compromisos laborales o vitales.
- La posibilidad de recibir información instantánea sobre los envíos.
- Posibilidad de realizar múltiples envíos de mejora continua.
- Incentivación de soluciones de programación modulares o basadas en pruebas: habilidades valoradas por la industria.

Además de estos beneficios potenciales, la crisis de COVID-19 y la reducción de la interacción presencial entre los docentes y los alumnos asociada a ésta ha hecho que la demanda de soluciones de corrección automatizada siga aumentando. Hay algunas áreas potenciales de preocupación con cualquier enfoque de la calificación automatizada:

- El diseño de las preguntas/tareas para que sean compatibles con un esquema de calificación automatizado puede ser demasiado prescriptivo y limitar la creatividad en la solución.

- Los programas de calificación automática son sensibles a los errores u omisiones, se requiere una determinada supervisión humana para garantizar la calidad de las soluciones.
- La gran inversión de tiempo por adelantado necesaria para desarrollar/implementar un esquema de este tipo.
- La retroalimentación automática puede eliminar parte del aprendizaje asociado al enfoque tradicional en el laboratorio, en el que una conversación con el alumno sobre su solución permite obtener una comprensión más profunda.
- Algunas formas de plagio pueden ser más difíciles de detectar.

Anteriormente y en otras asignaturas similares se utilizaban tanto metodologías de calificación de envío y devolución de tareas de MATLAB como así también de evaluación de las practicas en el laboratorio. Impulsados principalmente por la sustitución de los laboratorios presenciales por laboratorios online en el contexto de las clases online, y la necesidad de proporcionar una retroalimentación oportuna y consistente a los estudiantes, en las asignaturas mencionadas se ha implementado la calificación automática de las tareas de laboratorio principalmente utilizando la plataforma MATLAB Grader. MATLAB Grader es la solución nativa de MathWorks para la calificación automática del código MATLAB. En este documento se discuten nuestras experiencias en el desarrollo de tareas utilizando MATLAB Grader, la implementación de este paquete, las percepciones de los estudiantes y el personal docente en torno a esta herramienta y las limitaciones actuales junto con ideas de mejora.

2 MATLAB Grader

MATLAB Grader es una interfaz basada en el navegador para el diseño, distribución distribución, envío, calificación y análisis. Mientras que una biblioteca de problemas de ejemplo basados en la programación está disponible para su despliegue/adaptación, en la asignatura en cuestion se utilizó principalmente el servicio para diseñar evaluaciones personalizadas centradas en la modelización matemática de sistemas físicos utilizando función de transferencia.

Utilizamos la versión “gratuita” de MATLAB Grader 2020. Esta versión emplea una estructura por niveles: Curso, Tarea, Problema y Prueba. Dentro de un Curso, las asignaciones tienen fechas de publicación y vencimiento, ponderaciones, y

cada una puede albergar varios problemas. Un problema en este contexto implica escribir un script de MATLAB o una función de MATLAB en respuesta a un informe proporcionado. Estas instrucciones se especifican a través de un campo de texto enriquecido, en el que imágenes y ecuaciones LaTeX.

Cada problema puede tener varias pruebas (unitarias). Esto permite un enfoque de andamiaje en el diseño del problema y que se otorguen notas parciales si es necesario. Las pruebas unitarias pueden:

- Comparar la igualdad de las variables del alumno elemento por elemento con una solución de referencia.
- Comprobar la igualdad/desigualdad de las variables del alumno con un valor dentro de una banda de tolerancia.
- Comprobar la existencia o ausencia de variables o el uso de funciones intrínsecas específicas.
- Realizar una evaluación más complicada utilizando el código de MATLAB.

MATLAB Grader permite proporcionar una retroalimentación preprogramada para cada prueba de unidad, y el número de intentos permitidos. Cuando se limita el número de intentos, también se pueden implementar pre-pruebas no calificadas para proporcionar algún nivel de orientación/retroalimentación antes de la presentación y la evaluación. MATLAB Grader funciona a través de un navegador web y no requiere que se instale una copia local del software de MATLAB, aunque proporciona un mecanismo para descargar las pruebas previas como un archivo ejecutable de MATLAB fuera de línea que oculta la solución pero sigue proporcionando de la solución, pero que proporciona información a los estudiantes que prefieren trabajar sin conexión. La aplicación de esta herramienta a la enseñanza de la programación de MATLAB y de la ingeniería ya se ha discutido en la literatura.

Por ejemplo, MATLAB Grader (conocido entonces como ‘Cody Coursework’) se discutió ampliamente en [8], donde se utilizó para un subconjunto de las evaluaciones en un curso de curso de modelado y control de sistemas. Nuestras observaciones y experiencias coinciden en gran medida con las presentadas en el trabajo. Algunos de los puntos débiles específicos de Grader mencionados (por ejemplo errores específicos del envío de código) han sido rectificados por MathWorks, mientras que otros (por ejemplo, la detección de plagio) son características aún no implementadas.

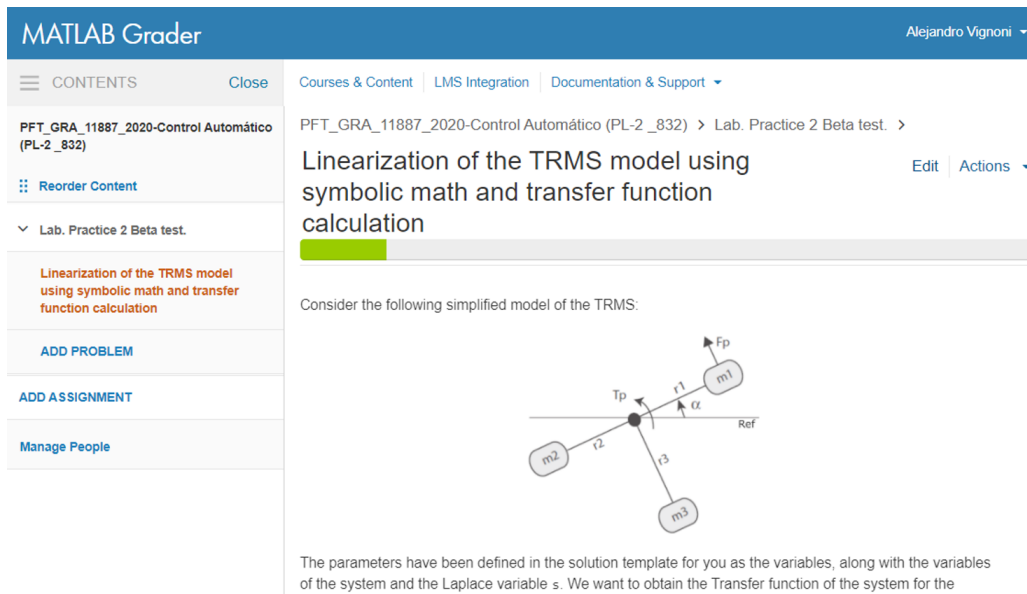


Figura 1: Presentación del problema en MATLAB Grader

En [11] utilizaron principalmente MATLAB Grader como una herramienta de revisión/estudio para los ejercicios formativos en clase. Los ítems de evaluación todavía se marcaban "a mano", y una combinación de Grader y scripts de "comprobación" personalizados estaban disponibles como recursos adicionales. En particular, descubrieron que la mayoría de los estudiantes utilizaban estas herramientas opcionales para validar sus respuestas a pesar de tener que modificar su código para cumplir con las restricciones necesarias para la "calificación" automática.

MATLAB Grader también se utilizó para los elementos evaluables en [6] para calificar las tareas basadas en MATLAB relativas a las vibraciones. Esta herramienta fue seleccionada debido al limitado tiempo disponible para proporcionar retroalimentación y un nivel variable de habilidad de programación en la clase. En las observaciones finales, el autor señaló que la plataforma era fácil de usar y una herramienta de enseñanza eficaz, aunque subrayó que debe prestarse una atención especial a la elaboración del enunciado del problema y de las pruebas de la unidad, una evaluación que también se puede realizar en el caso de los estudiantes. de los problemas y las pruebas de la unidad, una apreciación que también coincide con nuestra experiencia.

3 CASO DE ESTUDIO

En las asignaturas Control Automático y Tecnología de Automatización se utilizó MATLAB Grader para una de las evaluaciones de prácticas de laboratorio en 2020. El principal obstáculo

para una transición completa a MATLAB Grader fue y sigue siendo la falta de integración interactiva de Simulink. En esta sección, describimos nuestra experiencia con las asignaciones basadas en Grader a través de un estudio de caso de la asignatura Control Automático. Esta tarea de laboratorio implicaba el modelado matemático y la simulación de sistemas mecánicos simples. Un componente de pre-laboratorio enseñaba las habilidades necesarias junto con la metodología para plantear el modelado de sistemas mecánicos y consistía en un script de MATLAB en vivo (archivo .mlx) de demostración interactiva y un vídeo de 30 minutos de duración. Después de trabajar en el pre-laboratorio (no calificado), los estudiantes completaron una evaluación relacionada usando MATLAB Grader.

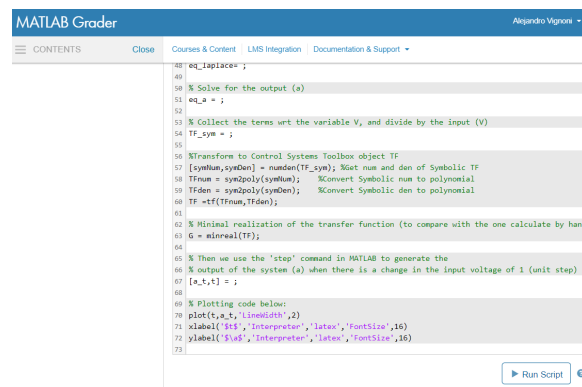


Figura 2: Vista del Alumno. Aquí el alumno debe ingresar su código y puede ejecutar el script para ver el resultado.

La evaluación del Laboratorio consistía en un

problema (Figura 1) que evaluaba los resultados de aprendizaje fundamentales: Modelado de un sistema mecánico utilizando el Symbolic Toolbox, cálculo del punto de equilibrio, linealización y obtención de la función de transferencia. Para problema, se permitió solo un intento y, aproximadamente el 50% de los estudiantes hicieron la entrega y obtuvieron la máxima puntuación. Las pruebas dentro del problema incluían cuatro pruebas no evaluadas (pretest) y tres evaluadas. Los estudiantes podían trabajar en una parte del problema, ejecutar el código correspondiente (Figura 2) asegurarse de que su código superaba la(s) prueba(s) no evaluadas necesaria(s) antes de continuar (Figura 3).

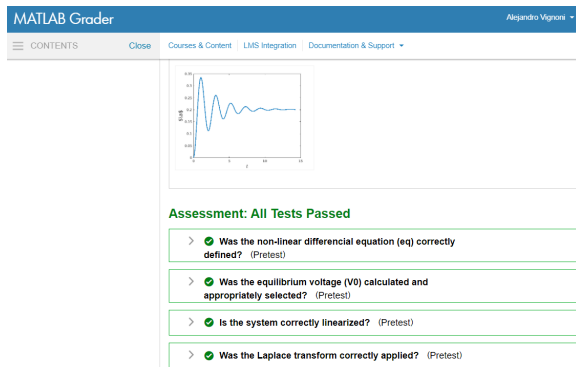


Figura 3: Pruebas no evaluadas. El alumno puede saber si su código cumple con las pruebas no evaluadas propuestas por el profesor antes de hacer la entrega definitiva.

En cada prueba se codificaron comentarios sobre los errores más comunes y se mostraron cuando la prueba fallaba. Esta retroalimentación era adicional a la salida por defecto de MATLAB y mensajes de retroalimentación/error de MATLAB Grader. En la Figura 3 se muestra una vista de las pruebas aprobadas/no aprobadas, mientras que en la Figura 4 se ve la ventana del docente, donde hay que ingresar la retroalimentación correspondiente en caso de que el alumno responda incorrectamente.

MATLAB Grader ofrece un conjunto de herramientas para el análisis básico de las soluciones de los alumnos. Encontramos la vista de 'Resumen de estado' de cuántos estudiantes habían empezado y completado la tarea como un valioso recordatorio durante las clases programadas, y el mapa de soluciones (Figura 5) donde se pueden ver todas las soluciones entregadas por los alumnos en un mapa de tamaño y orden de entrega. Además se puede ver también las soluciones de manera individual como se ve en la Figura 6. También se puede ver toda la estadística de un curso con muchas tareas, con los desgloses específicos del

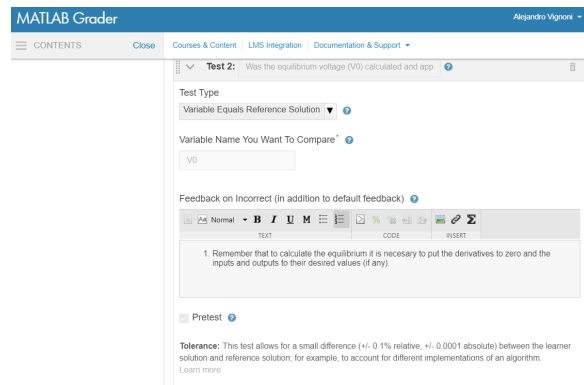


Figura 4: Vista del docente, evaluación (retroalimentación) al alumno cuando su respuesta es incorrecta).

problema y de las pruebas son útiles para identificar los "puntos más confusos" y las pruebas potencialmente mal planteadas, o pruebas insuficientemente granulares.

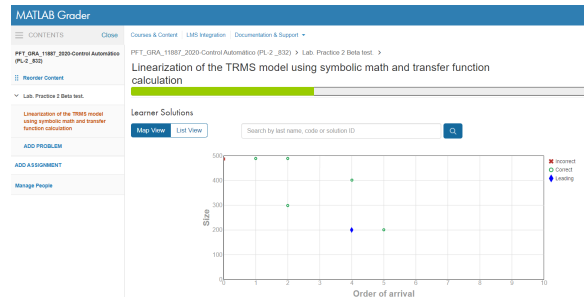


Figura 5: Vista de Mapa de Soluciones. Cada solución presentada por un alumno se representa por un punto en el mapa.

3.1 Comentarios/resultados/experiencias

La respuesta de los estudiantes a MATLAB Grader ha sido generalmente positiva, aunque todavía no hemos realizado encuestas a gran escala. Los estudiantes comunicaron su apreciación por la naturaleza de las tareas para llevar a casa y la retroalimentación instantánea/consistente. También apreciaron que la interfaz estuviera basada en la web y no requiriera una instalación local de MATLAB o un ordenador con una potencia de procesamiento significativa. Sin embargo, los estudiantes señalaron varios problemas con la plataforma/implementación, por ejemplo:

- El tiempo de respuesta: en algunos casos (especialmente para problemas complicados) el tiempo para evaluar un problema podía acercarse o superar el minuto. Esto era especialmente complicado para problemas en los que

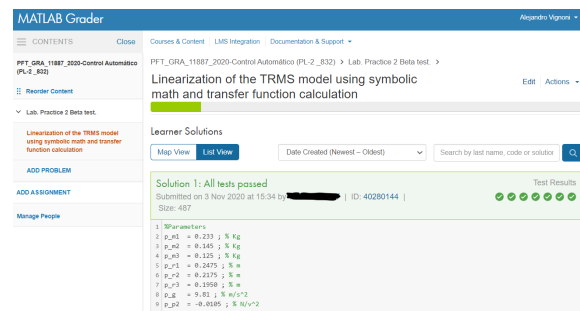


Figura 6: Vista de Solución individual. Ejemplo de una solución presentada por un alumno, su código y su respectiva evaluación.

la iteración/ajuste/prueba y error era fundamental para el proceso proceso.

- Errores del sistema: el sistema MATLAB Grader parece en general robusto, pero no infalible. Por ejemplo, en algunos casos el problema consistía en que se "bloqueaba" el cuadro de texto de entrada y se requería un cierre de sesión para poder resolver el problema. Anecdóticamente, estos problemas parecen haber sido más comunes en nuestra primera implementación de asignaciones basadas en MATLAB Grader (cuando el producto era razonablemente nuevo) ya que la estabilidad/robustez parecen haber mejorado con el tiempo.
- Un fallo por nuestra parte a la hora de explicar que prácticas comunes de codificación de MATLAB (por ejemplo borrar la ventana de comandos usando el `clc` o la memoria usando `clear` en la parte superior de un de un script) causaban errores en el código de evaluación en MATLAB Grader y darían lugar a mensajes de error poco útiles o poco claros.
- Errores con nuestra implementación - como con cualquier corrección y evaluación automatizada, los errores en nuestras pruebas unitarias o nuestra incapacidad de imaginar otros mecanismos potenciales para resolver el problema, condujeron a la frustración de los alumnos. Afortunadamente, la interfaz de MATLAB Grader permite modificar las pruebas y la recalificación de la cohorte "sobre la marcha" sean razonablemente rápidas y sencillas.

Durante esta prueba se utilizó la versión "gratuita" de Grader, requiriendo que los estudiantes se registraran para obtener una cuenta de MathWorks e "inscribirse" en MATLAB Grader en la primera semana del semestre.

La evaluación se realizó a través de la interfaz web y las calificaciones se exportaron a una hoja de cálculo y se cargaron manualmente en PoliformaT-UPV periódicamente a lo largo del semestre. Estas limitaciones no se consideraron una carga sustancial, además los alumnos manifestaron que no les importaba registrarse en una cuenta o que las calificaciones no fueran visibles inmediatamente en Poliformat, ya que poder ver su nota en la plataforma MATLAB Grader era suficiente. Además, el personal desarrolló un código para reformatear la salida de Excel de MATLAB Grader a un formato que pudiera subirse a PoliformaT (la plataforma basada en Sakai utilizada en la UPV) redujo el trabajo que suponía transferir manualmente las notas. EN el futuro se prevee utilizar la versión "premium" de MATLAB Grader que se incluye ahora con la MathWorks Campus-Wide, que viene con la integración del Sistema de Gestión de Aprendizaje (LMS)(por ejemplo, PoliformaT, Moodle, Blackboard, Canvas, etc.). El acceso a través del LMS no requeriría una cuenta de MathWorks y se ejecutaría completamente a través de PoliformaT. Las fechas de vencimiento y la ponderación de las tareas también se establecen en PoliformaT y las calificaciones se actualizan automáticamente en el libro de notas.

Tanto en la versión "gratuita" como en la "premium", la principal limitación que encontramos en MATLAB Grader era una integración incompleta/insuficiente de Simulink. Simulink es el producto de MATLAB para la simulación de sistemas físicos mediante diagramas de bloques. Actualmente, MATLAB Grader sólo permite un archivo Simulink proporcionado por el instructor para ser llamado de forma no interactiva a través de la línea de comandos.

4 CONCLUSIONES

MATLAB Grader ha demostrado ser una herramienta útil para lograr los resultados señalados al principio de este trabajo. Se ha conseguido una reducción sustancial de los costes de corrección. En general, el personal se siente más feliz enseñando que corrigiendo, y aunque se requirió una importante inversión inicial para desarrollar las tareas de las asignaciones, se ha recuperado en gran medida en una iteración del curso. Hasta este punto, el desarrollo de las tareas basadas en MATLAB Grader fue en general más fácil y más rápido. La interfaz para escribir problemas es sencilla, basada en el navegador/GUI y no requiere un conocimiento profundo del marco de pruebas unitarias o del backend de MATLAB un usuario inexperto podría "escribir" evaluaciones para comparar las soluciones de los alumnos con un resul-

tado de referencia en un par de *clicks*.

La funcionalidad de evaluación se limita a las calificaciones de igualdad/tolerancia y la integración interactiva de Simulink aún no se ha implementado. También se ha logrado una retroalimentación y calificación consistente e instantánea entre los estudiantes de las asignaturas en cuestión. También se ha conseguido que las tareas de laboratorio basadas en MATLAB Grader reciban comentarios y calificaciones para cada uno de los estudiantes. Los alumnos y alumnas apreciaron la retroalimentación inmediata, y la autonomía proporcionada por la plataforma MATLAB Grader, aunque entendían que sólo estaba disponible la retroalimentación posterior a la evaluación utilizando la de las pruebas unitarias o de las pruebas no evaluadas. Otro beneficio importante de la plataforma Grader fue la posibilidad de que los estudiantes enviaran múltiples soluciones, continuamente mejoradas, sin intervención del/los profesores - un paso hacia la independencia en el aprendizaje.

MATLAB Grader ha fomentado e incentivado, al menos, una comprensión/apreciación rudimentaria de la modularidad del código y el desarrollo basado en pruebas, aunque estos no son los principales resultados de aprendizaje de nuestro curso. Si bien los problemas iniciales en torno a la fiabilidad/robustez/velocidad del MATLAB Grader de MATLAB creemos que la utilización de de estas herramientas ha sido un esfuerzo que ha merecido la pena.

Agradecimientos

Este trabajo fue parcialmente financiado por MINECO/AEI, EU DPI2017-82896-C2-1-R y MICINN/AEI, EU PID2020-117271RB-C21. Además los autores agradecen la ayuda de José Luís Navarro por su ayuda en lo inicios de la utilización de MATLAB Grader.

Referencias

- [1] Christopher Douce, David Livingstone, and James Orwell. Automatic test-based assessment of programming: A review. *Journal on Educational Resources in Computing (JERIC)*, 5(3):4–es, 2005.
- [2] Ronald H Heck. *Studying educational and social policy: Theoretical concepts and research methods*. Routledge, 2004.
- [3] Jack Hollingsworth. Automatic graders for programming classes. *Communications of the ACM*, 3(10):528–529, 1960.

- [4] Stephan Krusche and Andreas Seitz. Artemis: An automatic assessment management system for interactive learning. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, pages 284–289, 2018.
- [5] Xiao Liu, Shuai Wang, Pei Wang, and Dinghao Wu. Automatic grading of programming assignments: an approach based on formal semantics. In *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*, pages 126–137. IEEE, 2019.
- [6] PE Natasha Smith. Integration of instructional technology tools including matlab grader to enhance learning in a hybrid vibrations course.
- [7] Sagar Parihar, Ziyaan Dadachanji, Praveen Kumar Singh, Rajdeep Das, Amey Karkare, and Arnab Bhattacharya. Automatic grading and feedback using program repair for introductory programming courses. In *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education*, pages 92–97, 2017.
- [8] Rebecca M Reck. Student and faculty reflections after using mathworks' Cody coursework in a control systems course. In *2018 IEEE Frontiers in Education Conference (FIE)*, pages 1–5. IEEE, 2018.
- [9] C Sánchez, D Muñoz de la Peña, and F Gómez-Estern. Generación automática de problemas de diseño de controladores para sistemas lineales autoevaluables con doctus. *Revista Iberoamericana de Automática e Informática Industrial*, 17(1):1–9, 2020.
- [10] CJ Sangwin. New opportunities for encouraging higher level mathematical learning by creative use of emerging computer aided assessment. *International Journal of Mathematical Education in Science and Technology*, 34(6):813–829, 2003.
- [11] James E Toney and Adithya Jayakumar. Utilizing the full range of matlab capabilities in the classroom. 2019.



© 2021 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>).