

Universidad Nacional de Educación a
Distancia
Escuela Técnica Superior de Ingeniería Informática
Departamento de Informática y Automática

TESIS DOCTORAL

CONTRIBUCIONES AL MODELADO,
SIMULACIÓN Y CONTROL EN INGENIERÍA
AGRÍCOLA

CONTRIBUTIONS TO MODELING,
SIMULATION AND CONTROL IN
AGRICULTURAL ENGINEERING

Agustín Pérez Castro
Madrid, 2018

Tesis doctoral

Contribuciones al modelado, simulación y
control en ingeniería agrícola

Contributions to modeling, simulation and
control in agricultural engineering



Universidad Nacional de Educación a Distancia
Departamento de Informática y Automática

Doctorando: Agustín Pérez Castro
Directores: José Sánchez Moreno
María del Mar Castilla Nieto

Madrid, 2018

Tesis doctoral

Contribuciones al modelado, simulación y control en ingeniería agrícola

Contributions to modeling, simulation and control in agricultural engineering



Universidad Nacional de Educación a Distancia
Departamento de Informática y Automática

Doctorando

Agustín
Pérez Castro

Director

José
Sánchez Moreno

Codirector

María del Mar
Castilla Nieto

A María del Mar.

A mi familia.

Agradecimientos

Me gustaría agradecer a todas aquellas personas que, con su apoyo y colaboración, han contribuido a la realización de esta tesis, mencionando especialmente:

- A los profesores José Sánchez Moreno y María del Mar Castilla Nieto como directores de la tesis, por haberme guiado y solucionado siempre mis dudas.
- Al resto del personal del Departamento de Informática y Automática de la UNED. Sin olvidar especialmente a mis compañeros de despacho.
- A todos los miembros del grupo de investigación en Automática, Robótica y Mecatrónica, de la Universidad de Almería, por acompañarme en mis primeros pasos en el mundo universitario.
- A los amigos que me traje de Brasil, especialmente a Julio Normey y Helton Scherer.
- A mis amigos: Ana, Alejandro, Amalia, Miguel, María del Mar, Antonio, Paco y Juanillo.
- Una mención especial a mi mujer María del Mar por su ayuda y apoyo.
- Y por último, a toda mi familia, que han estado conmigo en todas las decisiones que he tomado. Mis padres, Agustín y Amparo, mi hermano Álvaro y mi hermana Ampí.

A todos ellos, gracias.

Resumen

Con una población mundial que se espera alcance los 9 mil millones en 2050, unos recursos naturales cada vez más escasos y un cambio climático cada vez más presente y menos discutido; se hace más necesario que nunca la búsqueda de soluciones que permitan mejorar el rendimiento de los cultivos y optimizar el uso de recursos. Uno de los vectores que pueden resultar de utilidad para atacar el problema está en el desarrollo del modelado, simulación y control en Ingeniería Agrícola. Sin embargo, no se debe olvidar la importancia que tiene también la transferencia de dicho conocimiento y la creación de herramientas para el uso práctico del mismo.

Con este contexto en mente se han concebido las herramientas que se presentan en la presente tesis doctoral con tres objetivos diferenciados: el desarrollo de una herramienta abierta para experimentación de estrategias de modelado y control en Ingeniería Agrícola, el desarrollo de una aplicación para la gestión óptima del fertirriego; y, el desarrollo de una librería para el modelado y simulación de fotobiorreactores.

De esta forma, el resultado final de la presente tesis doctoral, además de las publicaciones y técnicas concebidas durante el desarrollo de la misma, son tres herramientas de software bien diferenciadas. Dichas herramientas están orientadas a la transferencia de conocimiento y la mejora en educación en Ingeniería Agrícola. Además de esto, según el uso que se dé a dichas herramientas, podrán suponer un instrumento para el sector productivo, dada su facilidad de uso, o incluso un buen material para la experimentación y prueba de desarrollos en investigación.

Abstract

With a world's population that is expected to reach 9 billion by 2050, natural resources becoming more and more exhausted and climate change increasingly present and less discussed, it is more necessary than ever to look for solutions which allow both to improve crop yields, and to optimize the use of resources. One of the strategies that can be useful to solve this problem is the development of modeling, simulation and control techniques in Agricultural Engineering. Nevertheless, it is also very important to guarantee the transference of this knowledge and the creation of tools for its practical use.

Within this framework, the tools presented in this PhD. dissertation have been conceived with three differentiated objectives: the development of an open tool for experimentation of modeling and control strategies in Agricultural Engineering, the development of an application for optimal fertigation management; and, the development of a library for the modeling and simulation of photobioreactors.

In this way, the final result of this PhD. dissertation, apart from the publications and techniques conceived along its development, are three well differentiated software tools. These tools are oriented to the transference of knowledge and the improvement in education in Agricultural Engineering. Furthermore, as a function of the use of these tools, they might be an instrument for the productive sector, since they are easy to use, or even a good material for experimentation and testing of new research developments.

Índice general

Agradecimientos	VI
Resumen	VIII
Abstract	X
1. Introducción, motivación y objetivos	1
1.1. Introducción	1
1.2. Desarrollo de una herramienta abierta para experimentación de estrategias de modelado y control en Ingeniería Agrícola	2
1.3. Desarrollo de una aplicación para la gestión óptima del fertirriego ..	4
1.4. Desarrollo de una librería para el modelado y simulación de fotobiorreactores	6
Referencias	9
2. Sistemas de experimentación y herramientas	13
2.1. Introducción	13
2.2. Desarrollo de una herramienta abierta para experimentación de estrategias de modelado y control en Ingeniería Agrícola	13
2.3. Desarrollo de una aplicación para la gestión óptima del fertirriego ..	18
2.4. Desarrollo de una librería para el modelado y simulación de fotobiorreactores	19
Referencias	20
3. Estrategias, técnicas y decisiones de diseño	23
3.1. Introducción	23
3.2. Desarrollo de una herramienta abierta para experimentación de estrategias de modelado y control en Ingeniería Agrícola	23
3.2.1. Desarrollo del lazo interno de control	23
3.2.2. Interfaz EasyJava	26
3.3. Desarrollo de una aplicación para la gestión óptima del fertirriego ..	30

3.4.	Desarrollo de una librería para el modelado y simulación de fotobiorreactores	32
3.4.1.	Dominios	33
3.4.2.	Esquema de composición	33
3.4.3.	Arquitectura	35
	Referencias	36
4.	Conclusiones y futuros trabajos	39
4.1.	Introducción	39
4.2.	Desarrollo de una herramienta abierta para experimentación de estrategias de modelado y control en Ingeniería Agrícola	39
4.3.	Desarrollo de una aplicación para la gestión óptima del fertirriego ..	40
4.4.	Desarrollo de una librería para el modelado y simulación de fotobiorreactores	41
4.5.	Conclusiones generales	43
	Referencias	45
5.	Publicaciones	47
5.1.	Introducción	47
5.2.	Publicaciones principales	47
5.3.	Resto de publicaciones	49
5.4.	Preprints de las publicaciones principales	53

Capítulo 1

Introducción, motivación y objetivos

1.1. Introducción

Según estudios recientes, está previsto que la población mundial supere los 9 mil millones en el año 2050 y, por lo tanto, es necesario incrementar la producción agrícola en un 60% con respecto a los niveles existentes en 2005 para asegurar la creciente demanda y la seguridad alimentaria [14]. Sin embargo, existen diversos factores, como el cambio climático o el agotamiento de los recursos naturales, que pueden influir negativamente en ese incremento de la producción agrícola y que hacen de esta tarea un gran desafío. Asimismo, estos factores ocasionan que la toma de decisiones sea cada vez más compleja y para optimizar dichas decisiones y automatizar la aplicación de las mismas es necesario disponer de los mecanismos de modelado, simulación y control apropiados.

Por otro lado, el crecimiento y el desarrollo socio-económico que se ha producido en el mundo de la ingeniería agrícola ha estado relacionado con la evolución de la tecnología. Actualmente, casi el 40% de la población mundial tiene acceso a Internet, y dentro del 5% más pobre, 7 de cada 10 casas tienen teléfono móvil [14, 40]. La gran aceptación e integración de las nuevas tecnologías han permitido reducir los costes de información y transacción, han creado nuevos trabajos y permiten realizar una gestión sostenible de los recursos naturales. Esto hace más fácil y más necesario que nunca la transferencia del conocimiento y la creación de herramientas que faciliten la aplicación de los mismos. De esta forma, esta integración de las nuevas tecnologías en el área de ingeniería agrícola puede suponer una contribución bastante significativa para alcanzar las necesidades futuras de producción agrícola.

Además, el uso de las nuevas tecnologías y la creación de herramientas permite también aumentar el conocimiento, lo cual facilita la integración de la tecnología entre agricultores, pero también puede revolucionar los sistemas de prevención temprana mediante una mejor calidad de los datos y de análisis. Por lo tanto, este tipo de he-

rramientas pueden utilizarse tanto en el ámbito de la educación y la investigación, así como, por los propios agricultores e ingenieros.

Actualmente, se pueden encontrar aplicaciones relacionadas con la agricultura y su adaptación al cambio climático que abarcan desde la monitorización climática mediante satélites y redes de sensores, hasta proyectos que incrementan la conciencia de los individuos ayudándolos a identificar y gestionar sus propias necesidades de adaptación. El principal objetivo de esta tesis es la propuesta, diseño y desarrollo de una serie de herramientas enfocadas al mundo de la ingeniería agrícola. Más concretamente, en las siguientes secciones de este capítulo se puede encontrar una descripción de la motivación y los principales objetivos que se pretendían alcanzar con el desarrollo de cada una de las herramientas.

En lo que refiere a la estructura de la tesis en este capítulo se explica la motivación de los problemas que se han decidido abordar. En el Capítulo 2 se realiza una revisión de los sistemas de experimentación y aplicaciones/lenguajes que se han utilizado en el desarrollo de las herramientas. A continuación, en el Capítulo 3 se habla sobre las estrategias, técnicas y decisiones de diseño más importantes que se han tomado o desarrollado a lo largo de los tres trabajos principales de esta tesis. Posteriormente, en el Capítulo 4, se exponen las principales conclusiones y líneas de continuación que se extraen de cada una de las publicaciones, así como, una discusión sobre las conclusiones generales de la tesis. Finalmente, en el Capítulo 5, se explican de forma resumida todas las publicaciones que se han realizado durante el desarrollo de la tesis y se incluyen los *preprints* de las publicaciones principales.

1.2. Desarrollo de una herramienta abierta para experimentación de estrategias de modelado y control en Ingeniería Agrícola

Gran parte de los procesos que se llevan a cabo en ingeniería, sobretudo en áreas como la ingeniería química o la ingeniería agrícola, deben satisfacer estrictas condiciones de temperatura, humedad, etc. Por lo tanto, es necesaria la implementación de distintas estrategias de control para garantizar que dichas condiciones son satisfechas y, de esta manera, que los procesos se desarrollen con normalidad. Es por esta razón, entre otras, por lo que la enseñanza de ingeniería de control es un pilar básico en la formación de los futuros ingenieros agrícolas, químicos y, en general, cualquier ingeniería. Además, no hay que olvidar que una parte importante en el desarrollo de las estrategias de control es el modelado previo de los procesos que se desean controlar.

Pese a que el aprendizaje teórico es necesario, tanto para estudiantes como para investigadores en distintos campos de la ingeniería, también es cierto que la experimentación real proporciona una información que no es menos importante. Esta

información sustenta la confirmación de las posibles teorías desarrolladas así como la absorción y clarificación de los contenidos teóricos. De esta forma, se proporciona a los estudiantes el entrenamiento requerido para poder aplicar sus conocimientos en cualquier campo de la ingeniería [16, 36].

Sin embargo, la experimentación real se tiene que enfrentar con bastante frecuencia a dos grandes problemas: el primero de ellos son los altos costes que representan tener disponibles instalaciones para la experimentación real [15, 31], el segundo de estos problemas es la distancia que los estudiantes pueden percibir entre los contenidos teóricos y su implementación práctica. Estos dos problemas han sido extensivamente analizados y es posible encontrar en la literatura diferentes artículos relacionados con el tema. Una de las soluciones más extendidas para resolver estos problemas es el desarrollo de laboratorios virtuales remotos. Más concretamente, en [4] se hace una revisión de los principales problemas a los que hay que prestar atención en lo que respecta a los laboratorios virtuales remotos accesibles a través de Internet. Por otro lado, en [13], se hace una exposición sobre la experiencia obtenida a través del desarrollo de laboratorios virtuales remotos a lo largo de los últimos años. Además, en ese trabajo se incluyen algunos consejos para aquellos que quieren desarrollar nuevos laboratorios virtuales remotos. Por último, en [38], se hace un análisis de las principales variables que influyen en la aceptación de este tipo de laboratorios por los estudiantes.

Por tanto, el objetivo perseguido durante el desarrollo de esta herramienta era el de resolver, de la mejor forma posible, los problemas mencionados previamente. Concretamente, el problema relativo a los costes de la instalación se han resuelto reduciendo los costes relativos de la misma. Dicha reducción de costes se ha abordado desde dos perspectivas, por una lado se ha hecho la instalación accesible a una mayor cantidad de público haciéndola accesible a través de Internet y, por otro lado, se ha proporcionado una plataforma que permita que dicho sistema de experimentación tenga una mayor flexibilidad. En lo que respecta al problema de la distancia percibida por los estudiantes entre los conocimientos teóricos y su uso práctico, se ha abordado mediante el uso de un laboratorio virtual que este directamente enlazado con su representación real. Esto último se ha conseguido permitiendo que el código implementado en el laboratorio virtual pueda ser directamente utilizado en el sistema real.

La Figura 1.1 resume perfectamente cual se pretende que sea la visión que el estudiante o el investigador tienen sobre el sistema que se ha desarrollado. Más específicamente, el usuario puede acceder a la herramienta de forma local, ya sea a través de una aplicación de escritorio o a través del acceso a una página web. Tras esto, el usuario de la misma puede implementar cualquier código para el sistema; como puede ser un controlador PID, un proceso de identificación, un controlador multivariable, un control óptimo, etc. Posteriormente, tras probar que su código se ejecuta de forma correcta en el sistema simulado, y tras verificar que la ejecución del mismo le proporciona los resultados esperados, el usuario puede, sin realizar ningún

cambio en su código, ejecutar el mismo en el sistema real y obtener los resultados de dicha ejecución.

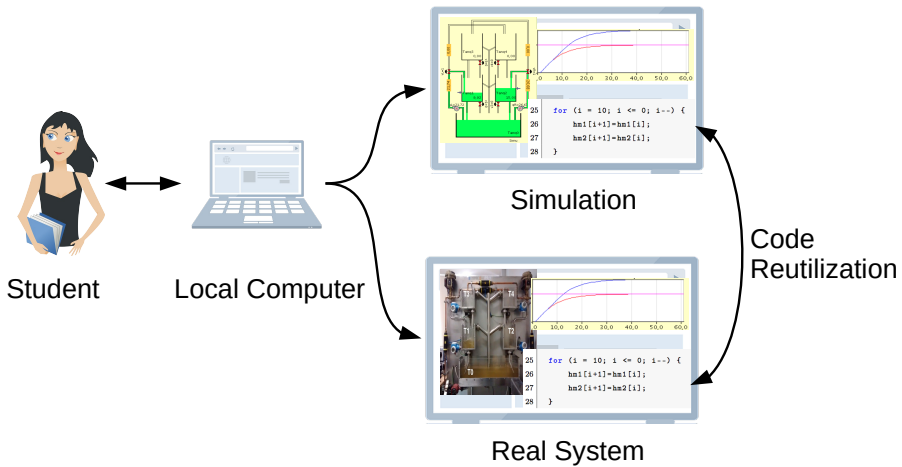


Figura 1.1 Vista esquemática del estudiante/investigador

1.3. Desarrollo de una aplicación para la gestión óptima del fertirriego

El cultivo de hortalizas en invernadero es uno de los mayores sectores económicos de la provincia de Almería (España), donde existe la mayor concentración de invernaderos del mundo [34], con más de 37000 hectáreas de estos invernaderos están dedicados al cultivo de hortalizas [26]. Lo anterior da como resultado una producción superior a los 3 millones de toneladas, donde el tomate es el cultivo más importante. Además, las exportaciones suponen casi el 70% de la producción en un negocio de 2000 millones de euros anuales [32]. Sin embargo, esta gran producción supone también una gran demanda de recursos de tierra y agua. Hay que destacar que el 80% del agua de riego utilizada procede de fuentes subterráneas, lo que está dando lugar a la sobreexplotación de acuíferos [18]. Dicha explotación de los recursos subterráneos se ha visto agravada por la reciente y rápida expansión de la superficie ocupada por los invernaderos en la cuenca mediterránea [26]. Como consecuencia de lo anterior, el agua puede ser considerada un factor importante en la sostenibilidad del sistema de cultivo de hortalizas en invernaderos [30]. Por esta razón, varias instituciones están trabajando en el desarrollo de diversas técnicas que permitan optimizar el uso de agua y fertilizantes en el riego, con el objetivo de redu-

cir los problemas ambientales asociados al proceso de fertirriego, para mitigar el grave déficit estructural de agua y el uso excesivo de fertilizantes [19], que agotan los acuíferos y reducen la calidad del agua [8, 33].

Como resultado, Almería, al igual que otras zonas áridas y semiáridas del mundo, ha promovido el uso de la fertirriego por goteo, la captación de agua de fuentes secundarias, la reutilización del agua de drenaje y el desarrollo de nuevas tecnologías relacionadas con el uso eficiente del agua y los nutrientes, como pueden ser los controladores avanzados de fertirriego. El objetivo de estos sistemas es suministrar con precisión tanto agua como fertilizantes. A su vez, esto puede resultar en asfixia radicular, inundación del sustrato o contaminación del agua subterránea [19]. Por el contrario, un déficit nutricional o hidrológico puede ser provocado si el proceso de riego no proporciona suficiente agua, lo que puede conducir a una disminución de la producción e incluso puede ser peligroso para el crecimiento de los cultivos [12]. Por lo tanto, el uso de sistemas de control automático y herramientas de gestión son fundamentales para suministrar agua y nutrientes en la cantidad y frecuencia requeridas para cualquier cultivo [23]. Por esta razón, la eficiencia en la cantidad de agua y nutrientes suministrados es un problema práctico de considerable interés, que tiene una importancia económica significativa [23]. Los requisitos del sistema de control de fertirriego deben determinarse en función del rendimiento deseado del cultivo, los análisis del agua de riego y drenaje, el contenido de nutrientes del suelo/sustrato y el tipo de sistema de riego [20]. Por lo tanto, una herramienta de gestión adecuada puede ayudar a este fin, ya que la gestión eficiente de estos sistemas es bastante complicada y, por lo tanto, requiere un conocimiento profundo del cultivo, la distribución del agua y los nutrientes y otros fenómenos químicos como la precipitación, la absorción de nutrientes, etc.

Actualmente, en el mercado existen varios ejemplos de sistemas de apoyo a la decisión, que pueden ayudar a facilitar los cálculos de fertirriego; sistemas como *Ecofert* [6], *Smart-fertilizer* [2], *SSCA* [3], *Optifer* [27] y varias herramientas de Excel como *Homo-Agricola* [1].

Por todo lo anterior, el objetivo que persigue este trabajo es el desarrollo de una herramienta de fertirriego que proporcione a estudiantes, ingenieros agrónomos y agricultores un mayor grado de configuración y una capacidad de análisis igual o superior a las herramientas existentes. Por ejemplo, permite la configuración de diferentes equipos de fertirriego, desde los más avanzados controladores de fertirriego (con hasta 12 tanques) hasta los sistemas más básicos; es capaz de realizar predicciones del agua requerida para los cultivos incluyendo el análisis de las aguas de drenaje; y proporciona resultados automáticos estimados en función de un algoritmo de optimización. Además, se ha realizado un diseño para que dicha aplicación pueda ser utilizada en móviles y tabletas y, de esta forma, les permita ayudar al proceso de decisiones accediendo desde cualquier lugar a los datos.

El resultado final ha sido una aplicación que permite a los usuarios realizar una

gestión adecuada de los fertilizantes para garantizar unas condiciones óptimas para el crecimiento de los cultivos al mismo tiempo que se gestiona el uso del agua y los fertilizantes. Además, permite el control simultáneo de varios sistemas de cultivo mediante el uso de bases de datos persistentes. De esta manera, el uso de la aplicación proporcionará a ingenieros agrícolas, agricultores y estudiantes una poderosa herramienta de apoyo para los sistemas de toma de decisiones en la gestión de fertirrigación. De esta forma la aplicación que se ha desarrollado es capaz de determinar los fertilizantes que deben aplicarse por litro de agua en los tanques de fertirrigación. Este cálculo se inicia con diferentes análisis (suelo, agua de riego y, en el caso del cultivo sin suelo, drenaje) y se obtiene una solución ideal para cada cultivo y estado de crecimiento. Esta solución óptima se incluye en el perfil de invernadero/agricultor. Para ello, se requiere información adicional relacionada con el tipo de sistema de fertirrigación, el tamaño y número de tanques de fertirrigación, la concentración de fertilizantes y las características del ácido nítrico y fosfórico, etc. Además, la aplicación es capaz de evaluar los diferentes análisis de drenaje y de suelo para modificar la solución óptima en función de los resultados obtenidos. Así, la aplicación se espera que funcione tanto como un sistema de aprendizaje como una herramienta de ayuda en la toma de decisiones para los sistemas de fertirrigación, lo que aumenta la accesibilidad a la metodología utilizada para determinar los parámetros óptimos de fertirrigación de cualquier sistema de cultivo.

1.4. Desarrollo de una librería para el modelado y simulación de fotobiorreactores

En los últimos años, hay una tendencia en la investigación del cultivo de microalgas para la producción de energía como alternativa a los combustibles fósiles y al biodiesel [5, 7]. Además, también está adquiriendo relevancia para la producción de componentes bioquímicos orientados a la alimentación y la medicina [24]. Sin embargo, el cultivo de microalgas requiere unas instalaciones adecuadas para su crecimiento [29, 39], las cuales pueden ser clasificadas en dos grupos bien diferenciados, fotobiorreactores abiertos y cerrados.

La principal característica de los fotobiorreactores abiertos es que el agua del cultivo está en contacto directo con el aire ambiente. Estos sistemas abiertos pueden tener forma de estanque, lago o algún gran recipiente. La principal ventaja de este tipo de fotobiorreactores es que son más fáciles y baratos de construir y mantener que los cerrados. Por otro lado, sus principales desventajas están relacionadas con su exposición al ambiente, siendo estas: la evaporación del agua, la contaminación de los cultivos, la necesidad de grandes superficies de terreno y los problemas de distribución de la luz en cultivos relativamente profundos.

En los fotobiorreactores cerrados, la luz, el aire, y los nutrientes pueden ser controlados para, de esta forma, poder intentar conseguir un cultivo óptimo. Sus principales

desventajas están asociadas a los costes de construcción y mantenimiento. En lo que respecta a sus puntos a favor; no están tan expuestos a problemas de contaminación de los cultivos, tienen mayores tasas de producción que los abiertos, tienen una menor evaporación del medio de cultivo, y, el estar en un entorno cerrado y controlado, hace más sencillo el control y la manipulación del cultivo. Dentro de los fotobiorreactores cerrados se pueden encontrar distintas geometrías que buscan una distribución más eficiente de la luz. Las geometrías más comunes son la plana, la anular y la tubular [9]

El diseño y desarrollo de fotobiorreactores, con el objetivo de maximizar cualitativa y cuantitativamente la producción, involucra diferentes aspectos que van desde como se recibe y distribuye la radiación, la transferencia de materia, la escalabilidad, la nutrición, la temperatura, el pH del cultivo y hasta las estrategias de control necesarias para mantener algunas de las condiciones anteriormente mencionadas. Por todo lo anterior, es necesario disponer de herramientas para modelar y simular los diferentes aspectos que puedan afectar al desempeño y rendimiento de los fotobiorreactores. El desarrollo de estos modelos permitirá a los investigadores la evaluación de los diseños y de distintas estrategias con el fin de mejorar la productividad. Además, previo a la implementación de sistemas de experimentación reales, el disponer de modelos permitirá el desarrollo de una plataforma de simulación, prueba y aprendizaje previa a la construcción de dichos sistemas.

Aunque el interés por los fotobiorreactores tubulares no es algo reciente, no existen métodos o software específico para su modelado [28]. La complejidad del modelado y la identificación de parámetros, las dificultados en la medición de sistemas biológicos y la diversidad existente en la configuración de los fotobiorreactores son, en cierto modo, las principales dificultades que encontramos en el modelado y simulación de estos sistemas. Todo lo anterior nos conduce a los objetivos de uno de los trabajos incluidos en la presente tesis. Tal y como se indica en el título de la presente sección, el objetivo perseguido es el desarrollo de una librería para el modelado y simulación de fotobiorreactores. Aunque el objetivo final de dicha librería es permitir el modelado y simulación de cualquier tipo de fotobiorreactor, para la primera versión se ha decidido implementar un fotobiorreactor tubular cerrado. Sin embargo, se pretende que todo lo implementado, o la mayor parte de lo mismo, pueda ser reutilizado para el modelado de otro tipo de fotobiorreactores y esto se tiene en cuenta a la hora de seleccionar el tipo de modelado, las ecuaciones seleccionadas y la arquitectura de la librería.

La librería se ha implementado en *Modelica*, más concretamente con la herramienta *Dymola*. En la Sección 2.4 se muestran las características más importantes de dichos lenguaje y herramienta. Para entender un poco mejor la motivación para crear este tipo de librería se expone en los siguientes párrafos una pequeña visión del estado del arte en el modelado y simulación de fotobiorreactores en el momento previo a la realización de la librería.

En el momento en que se decidió el desarrollo de la librería, solo era posible encontrar aplicaciones desarrolladas desde cero, sin el apoyo de librerías específicas; o el uso de programas de modelado y simulación más o menos genéricos para el modelado y simulación de algunos aspectos parciales, como pueden ser, la dinámica de fluidos o los procesos de transferencia de calor. A continuación, se enumeran algunas de las aplicaciones utilizadas en la literatura, explicando brevemente el ámbito en el que se han utilizado y su utilización específica:

- *Aspen Plus* es usado en [11] para hacer un estudio de la viabilidad económica de un fotobiorreactor abierto y otro cerrado. En la publicación citada *Aspen Plus* es utilizado para el modelado de los balances de masa y energía.
- En [35], donde se presenta el diseño de un fotobiorreactor vertical para la producción de oxígeno a pequeña escala, se hace uso de *COMSOL Multiphysics* para estudiar la transferencia de calor.
- En [25] se utiliza *Labview* para evaluar el potencial en la generación de biomasa y en la captura de CO_2 teniendo en cuenta diferentes tipos de especies de algas, tipos de fotobiorreactores y distintas localizaciones. Como resultado de este trabajo se desarrolla la herramienta *EvAlgae*
- En [41], la herramienta *ANSYS Fluent* es utilizada para realizar una simulación numérica del rendimiento en la agitación del medio de cultivo, la intensidad luminosa recibida por las algas y los patrones en el fluido en diferentes tipos de fotobiorreactores.
- Utilizando *Matlab* y *Simulink* como herramientas genéricas de programación, modelado y simulación; en [17] se implementa, simula, calibra y valida un modelo dinámico de cultivo de microalgas. Este modelo tiene en cuenta los procesos de transferencia de masa y energía y los fenómenos biológicos. Las ecuaciones y datos utilizados en este trabajo serán el inicio de la librería que se pretende desarrollar.
- *MeteoNorm* se utiliza en [21] para generar los datos meteorológicos necesarios para simular y desarrollar un modelo de temperatura de un fotobiorreactor cerrado para investigar los parámetros que influyen en su comportamiento térmico.
- En [22], se utiliza *SuperPro Designer* para modelar el proceso de captura de CO_2 , para esto se considera tanto el crecimiento de la biomasa como el proceso de separación de oligoelementos. El objetivo final que se persigue es el de desarrollar una tecnología sostenible, a escala piloto, para la captura de los gases de combustión de una central eléctrica para la producción de biodiesel mediante el uso de microalgas.
- El software *Tracepro* es utilizado en [37] para diseñar un fotobiorreactor que absorba de forma óptima la radiación. Este trabajo se basa en una patente de un colector solar y utiliza cables de fibra óptica para la distribución de la luz.
- También se pueden encontrar trabajos que utilizan combinaciones de las herramientas anteriores. Por ejemplo, en [10], se utilizan *COMSOL Multiphysics* y *Aspen Plus* para estudiar la dinámica de fluidos y la absorción de CO_2 en un fotobiorreactor helicoidal.

Las herramientas que se han enumerado previamente solo describen aspectos parciales del proceso completo y, además, algunas de ellas no permiten el cálculo de transitorios, mostrando únicamente las relaciones estáticas. Esto hace interesante el desarrollo de la librería en *Modelica*, ya que, además de admitir un modelado más completo y dinámico, nos permite la interconexión con controladores para probar su rendimiento o realizar simulaciones más completas mediante el uso de otros modelos/librerías que pueden ser interconectados. La conexión con otras librerías permitirá, por ejemplo, la inclusión de modelos de radiación solar y condiciones ambientales en cualquier zona y simular la colocación del fotobiorreactor en entornos específicos con sus propias dinámicas como podría ser el caso de un invernadero.

Referencias

- [1] Web oficial de homo-agricola. Último acceso 23/09/2018. URL <https://sites.google.com/site/revistahomoagricola/home>.
- [2] Web oficial de smart-fertilizer. Último acceso 23/09/2018. URL <https://web.smart-fertilizer.com>.
- [3] Web oficial de ssca. Último acceso 23/09/2018. URL <http://www.ssca.ca/28-last-added-projects/71-ssca-fertilizer-blend-calculator>.
- [4] B. Balamuralithara and P. C. Woods. Virtual laboratories in engineering education: The simulation lab and remote lab. *Computer Applications in Engineering Education*, 17(1):108–118, 2009.
- [5] O. Bernard. Hurdles and challenges for modelling and control of microalgae for co2 mitigation and biofuel production. *Journal of Process Control*, 21(10):1378–1389, 2011.
- [6] M. V. Bueno-Delgado, J. M. Molina-Martínez, R. Correoso-Campillo, and P. Pavón-Mariño. Ecofert: An android application for the optimization of fertilizer cost in fertigation. *Computers and Electronics in Agriculture*, 121:32–42, 2016.
- [7] P. K. Campbell, T. Beer, and D. Batten. Life cycle assessment of biodiesel production from microalgae in ponds. *Bioresource technology*, 102(1):50–56, 2011.
- [8] F. Carvajal, F. Agüera, and J. Sánchez-Hermosilla. Water balance in artificial on-farm agricultural water reservoirs for the irrigation of intensive greenhouse crops. *Agricultural water management*, 131:146–155, 2014.
- [9] A. P. Carvalho, L. A. Meireles, and F. X. Malcata. Microalgal reactors: a review of enclosed system designs and performances. *Biotechnology progress*, 22(6):1490–1506, 2006.
- [10] K. Cheenkachorn, N. Choosri, A. Chutrapukdeekul, and T. Kangsadan. Computational modeling of microalgae culture using a helical photobioreactor. In *Proceedings of the 9th IASME/WSEAS International Conference on Fluid Mechanics & Aerodynamics (FMA'11)*. August, pages 23–25, 2011.

- [11] R. Davis, A. Aden, and P. T. Pienkos. Techno-economic analysis of autotrophic microalgae for fuel production. *Applied Energy*, 88(10):3524–3531, 2011.
- [12] I. Domeño, N. Irigoyen, and J. Muro. Evolution of organic matter and drainages in wood fibre and coconut fibre substrates. *Scientia horticultrae*, 122(2): 269–274, 2009.
- [13] F. Esquembre. Facilitating the creation of virtual and remote laboratories for science and engineering education. *IFAC-PapersOnLine*, 48(29):49–58, 2015.
- [14] FAO. Information and Communication Technology (ICT) in Agriculture. A Report to the G20 Agricultural Deputies.
- [15] J. Fayolle, C. Gravier, N. Yankelovich, and E. Kim. Remote lab in virtual world for remote control of industrial processes. In *Multimedia and Expo (ICME), 2011 IEEE International Conference on*, pages 1–4. IEEE, 2011.
- [16] L. D. Feisel and A. J. Rosa. The role of the laboratory in undergraduate engineering education. *Journal of Engineering Education*, 94(1):121–130, 2005.
- [17] I. Fernández, F. Ación, J. Fernández, J. Guzmán, J. Magán, and M. Berenguel. Dynamic model of microalgal production in tubular photobioreactors. *Bioresource technology*, 126:172–181, 2012.
- [18] M. Fernández, A. González, J. Carreño, C. Perez, and S. Bonachela. Analysis of on-farm irrigation performance in mediterranean greenhouses. *Agricultural water management*, 89(3):251–260, 2007.
- [19] M. Gallardo, R. Thompson, J. Rodríguez, F. Rodríguez, M. Fernández, J. Sánchez, and J. Magán. Simulation of transpiration, drainage, n uptake, nitrate leaching, and n uptake concentration in tomato grown in open substrate. *Agricultural water management*, 96(12):1773–1784, 2009.
- [20] T. H. Gieling. *Control of water supply and specific nutrient application in closed growing systems*. sn], 2001.
- [21] V. Goetz, F. Le Borgne, J. Pruvost, G. Plantard, and J. Legrand. A generic temperature model for solar photobioreactors. *Chemical engineering journal*, 175:443–449, 2011.
- [22] P. Iancu, V. Pleşu, and S. Velea. Flue gas co₂ capture by microalgae in photobioreactor: a sustainable technology. *Chemical Engineering*, 29, 2012.
- [23] J. Magán, N. Moreno, D. Meca, and F. Cánovas. Response to salinity of a tomato crop in mediterranean climate conditions. In *International Symposium on Growing Media and Hydroponics 644*, pages 479–484, 2001.
- [24] J. J. Milledge. Commercial application of microalgae other than as biofuels: a brief review. *Reviews in Environmental Science and Bio/Technology*, 10(1): 31–41, 2011.
- [25] J. Miñon Martínez. *Modelo de biomasa algal para la captura de dióxido de carbono y su desarrollo en un software de evaluación*. PhD thesis, Universidad de Valladolid, 2012.
- [26] J. A. S. Molina. *Contribuciones al modelado y simulación de sistemas de cultivo intensivos. Contributions to the modelling and simulation of intensive growing systems*. PhD thesis, Universidad de Almería, 2015.

- [27] F. Pagán, M. Ferrández-Villena, D. Fernández-Pacheco, J. Rosillo, and J. Molina-Martínez. Optifer: An application to optimize fertiliser costs in fertigation. *Agricultural Water Management*, 151:19–29, 2015.
- [28] Š. Papáček, V. Štumbauer, D. Štys, K. Petera, and C. Matonoha. Growth impact of hydrodynamic dispersion in a couette–taylor bioreactor. *Mathematical and Computer Modelling*, 54(7-8):1791–1795, 2011.
- [29] A. Richmond. *Handbook of microalgal culture: biotechnology and applied phycology*. John Wiley & Sons, 2008.
- [30] F. Rodríguez, M. Berenguel, J. L. Guzmán, and A. Ramírez-Arias. Climate and irrigation control. In *Modeling and control of greenhouse crop growth*, pages 99–196. Springer, 2015.
- [31] J. Rossiter and Y. Shokouhi. Developing virtual laboratories for introductory control. In *Control (CONTROL), 2012 UKACC International Conference on*, pages 1025–1030. IEEE, 2012.
- [32] J. A. Sánchez, F. Rodríguez, J. L. Guzmán, and M. R. Arahál. Virtual sensors for designing irrigation controllers in greenhouses. *Sensors*, 12(11):15244–15266, 2012.
- [33] F. Sánchez-Martos, A. Pulido-Bosch, L. Molina-Sánchez, and A. Vallejos-Izquierdo. Identification of the origin of salinization in groundwater using minor ions (lower andarax, southeast Spain). *Science of the Total Environment*, 297(1-3):43–58, 2002.
- [34] J. Sánchez-Molina, J. Reinoso, F. Ación, F. Rodríguez, and J. López. Development of a biomass-based system for nocturnal temperature and diurnal CO₂ concentration control in greenhouses. *Biomass and Bioenergy*, 67:60–71, 2014.
- [35] K. J. Sink. *Design and Construction of a Lighting System to Illuminate a Photobioreactor*. PhD thesis, Ohio University, 2011.
- [36] M. Stefanovic, D. Tadic, S. Nestic, and A. Djordjevic. An assessment of distance learning laboratory objectives for control engineering education. *Computer Applications in Engineering Education*, 23(2):191–202, 2015.
- [37] B. Tamburic, F. W. Zemichael, P. Crudge, G. C. Maitland, and K. Hellgardt. Design of a novel flat-plate photobioreactor system for green algal hydrogen production. *International journal of hydrogen energy*, 36(11):6578–6591, 2011.
- [38] R. Tirado, R. S. Herrera, M. A. Márquez, A. Mejías, and J. M. Andújar. Comparing remote laboratories from the student perspective. *IFAC-PapersOnLine*, 48(29):176–181, 2015.
- [39] C. Ugwu, H. Aoyagi, and H. Uchiyama. Photobioreactors for mass cultivation of algae. *Bioresource technology*, 99(10):4021–4028, 2008.
- [40] World Bank Group. Digital Dividends.
- [41] L. Wu, Z. Li, and Y. Song. Hydrodynamic conditions in designed spiral photobioreactors. *Bioresource technology*, 101(1):298–303, 2010.

Capítulo 2

Sistemas de experimentación y herramientas

2.1. Introducción

El principal objetivo de este capítulo es introducir los sistemas con los que se ha trabajado para el desarrollo de cada una de las herramientas presentadas en esta tesis doctoral, así como, el software y librerías específicas utilizadas para el desarrollo de las mismas.

2.2. Desarrollo de una herramienta abierta para experimentación de estrategias de modelado y control en Ingeniería Agrícola

Sistema de cuatro tanques

El motivo de que se haya elegido este sistema para implementar la herramienta de experimentación es que se trata de uno de los sistemas más relevantes utilizados en la didáctica del control multivariable. El esquema básico fue desarrollado por Eric Johansson del *Lund Institute of Technology* (Suecia) y publicado entre otros en [17, 18, 16]. Dicho sistema, cuyo esquema básico se muestra en la Figura 2.1, está compuesto por cuatro tanques, dos bombas y dos válvulas de tres vías. En lo que respecta a su funcionamiento se puede desglosar en los siguientes pasos: las bombas extraen el líquido del depósito de desagüe inferior, y este caudal se distribuye entre los cuatro depósitos a través de las válvulas de tres vías. Es importante mencionar que las tuberías sobre los tanques superiores están cruzadas y esto hace que las dos bombas tengan influencia sobre los dos tanques inferiores. Lo anterior tiene como efecto que las válvulas de tres vías modifiquen la dinámica del sistema. En [16] se puede encontrar una explicación completa del sistema y una discusión sobre su dinámica y aplicaciones del control multivariable.

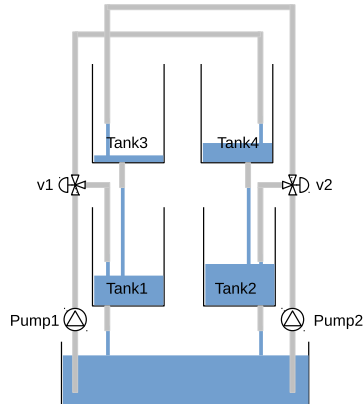


Figura 2.1 Esquema del sistema de cuatro tanques

Lo anterior es referente al modelo teórico que dio origen al sistema de cuatro tanques, el esquema del sistema real tiene algunos cambios. Más concretamente, las principales diferencias son la adición de cuatro válvulas de cierre (ON/OFF) en la base de los tanques, para la generación de perturbaciones, y cuatro caudalímetros para medir el caudal en cada una de las ramas del sistema, véase Figura 2.2. Todos los sensores y actuadores son monitorizados y controlados por un PLC (*Programmable Logic Controller*), el cual es manejado por una herramienta SCADA (*Supervisory Control And Data Acquisition*) que ha sido desarrollada para el trabajo. El sistema real se encuentra en el Departamento de Informática y Automática (UNED), véase Figura 2.3. La Tabla 2.1 muestra una lista de los componentes del sistema.

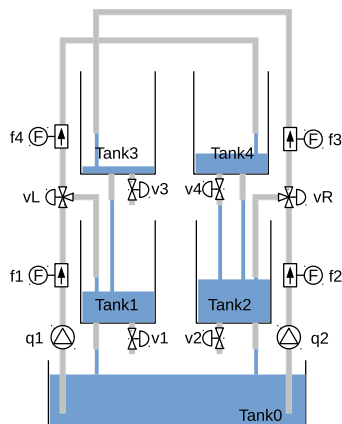


Figura 2.2 Esquema del sistema de cuatro tanques modificado

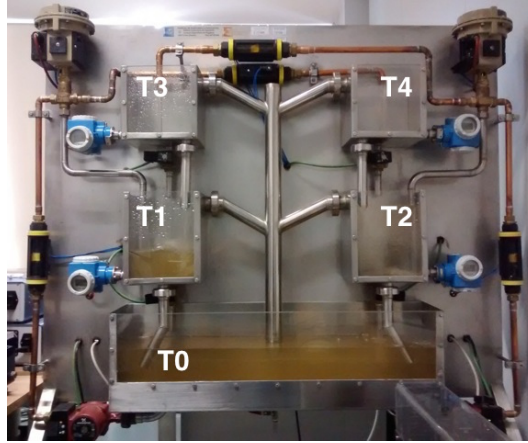


Figura 2.3 Sistema real de cuatro tanques localizado en la UNED

Tabla 2.1 Componentes del sistema de cuatro tanques

Componente	Modelo
Bombas	Grundfos UPE 25-40
Válvulas de tres vías	Samson 3226/2780-2
Válvulas de cierre	SMC
Sensores de altura de agua	Endress & Hauser PCM 731
Sensores de flujo	RS Amidata 257-026
Sistema de adquisición	Opto 22 SNAP Ultimate I/O

Easy Java Simulations (EJS)

Es una herramienta de software libre desarrollada en *Java* que ayuda a los no programadores a crear simulaciones interactivas, principalmente con fines docentes o aprendizaje [7, 14]. *EJS* es parte del proyecto *Open Source Physics* [8]. En la Figura 2.4 se puede ver la interfaz de la aplicación desarrollada con esta herramienta. Esta aplicación puede ser utilizada tanto en simulación como en pruebas de experimentación real. Más concretamente, para poder hacer las pruebas reales, esta aplicación se conecta con el *SCADA* del sistema real, implementado en *Labview*, a través de *JilServer*. Además, para poder ser accesible a través de Internet la aplicación esta accesible en la web del proyecto *Unilabs*.

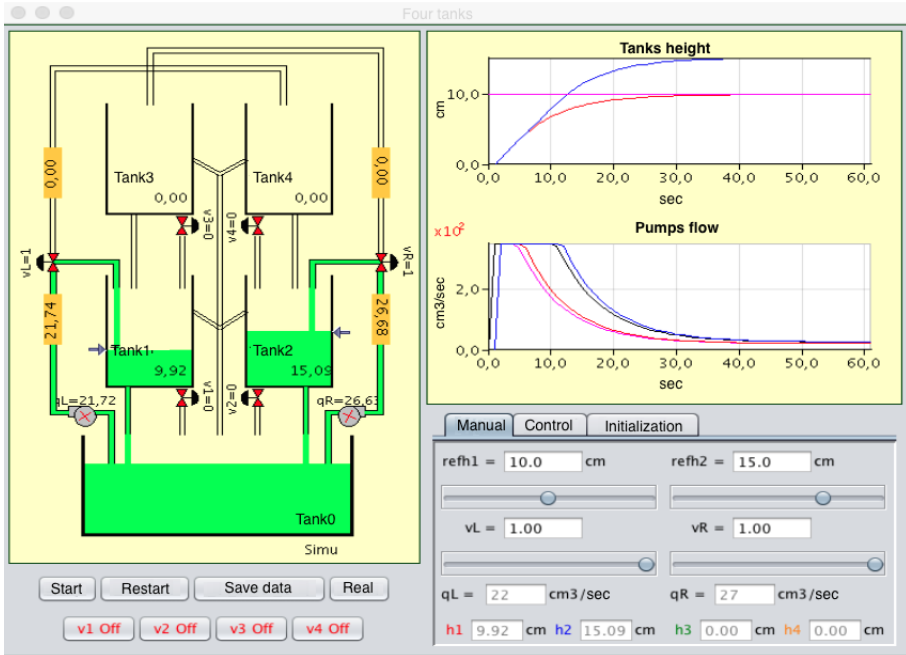


Figura 2.4 Interfaz del sistema real de cuatro tanques en *EasyJava*

JilServer

Se trata de una herramienta implementada en *LabView* que permite la comunicación entre un SCADA desarrollado en *LabView* y una aplicación *EasyJava*. Más en detalle, *JilServer* publica los controles e indicadores existentes en el archivo VI de una aplicación *Labview* y los hace visibles a través de una conexión TCP (*Transmission Control Protocol*) para cualquier aplicación que necesite establecer una comunicación con *Labview* [13].

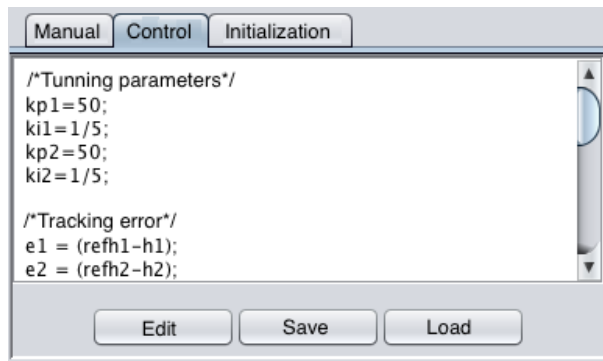
Unilabs

Se trata de una red educativa compuesta por varias universidades que comparten los recursos de sus laboratorios [9]. En este sitio, los estudiantes inscritos pueden encontrar todo el material necesario para realizar experimentos y utilizar las herramientas tanto en modo simulación como real. El acceso al modo real está restringido. Para poder hacer uso del modo real, el usuario debe reservar la instalación a través del sistema de reservas integrado en la web del proyecto utilizando sus credenciales privadas. Esto hace posible que una instalación dada pueda ser accesible por muchas personas desde distintos puntos geográficos sin que haya interferen-

cias en los experimentos. Además, la plataforma *Unilabs* puede almacenar tanto los archivos con el código que se prueba en la instalación como los resultados generados durante los experimentos. Hay que destacar que, en el caso que nos ocupa, los códigos de pruebas son desarrollados utilizando *JavaScript*.

JavaScript

Se trata de un lenguaje de programación interpretado. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo la implementación de páginas web dinámicas. Su uso en aplicaciones externas a la web, como puede ser en documentos PDF o en aplicaciones de escritorio es también significativo [11]. La elección de *JavaScript* como lenguaje para controlar el sistema está motivado por diferentes razones. La más importante es que se trata de un lenguaje sencillo y muy utilizado, esto implica una curva de aprendizaje muy rápida y una gran cantidad de recursos de consulta. Por otro lado, su naturaleza y el que esté pensado para ser ejecutado en un navegador, y que sea interpretado, proporciona seguridad a la hora de ser ejecutado en un sistema *SCADA* remoto. En la Figura 2.5 se puede ver un ejemplo del código *JavaScript* implementado en la herramienta.



```
/*Tunning parameters*/
kp1=50;
ki1=1/5;
kp2=50;
ki2=1/5;

/*Tracking error*/
e1 = (refh1-h1);
e2 = (refh2-h2);
```

Figura 2.5 Código *JavaScript* implementado en la aplicación *EasyJava*

2.3. Desarrollo de una aplicación para la gestión óptima del fertirriego

Android Studio

Es el entorno de desarrollo integrado oficial para la plataforma Android. Fue anunciado el 16 de mayo de 2013 en la conferencia Google I/O, y reemplazó a Eclipse como el IDE oficial para el desarrollo de aplicaciones para Android. La primera versión estable fue publicada en diciembre de 2014.

Está basado en el software IntelliJ IDEA de JetBrains y ha sido publicado de forma gratuita a través de la Licencia Apache 2.0. Está disponible para las plataformas Microsoft Windows, macOS y GNU/Linux. Ha sido diseñado específicamente para el desarrollo de Android [10, 1].

JOptimizer

Es una librería de software libre implementada en Java para la solución de problemas de minimización con restricciones (con igualdades y desigualdades) [4].

SQLite

Es un sistema de gestión de bases de datos relacional, contenida en una relativamente pequeña (275 kiB) biblioteca escrita en C. *SQLite* es un proyecto de dominio público creado por Richard Hipp.

A diferencia de los sistema de gestión de bases de datos cliente-servidor, el motor de *SQLite* no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca *SQLite* se enlaza con el programa pasando a ser parte integral del mismo. El programa utiliza la funcionalidad de *SQLite* a través de llamadas simples a subrutinas y funciones. Esto reduce la latencia en el acceso a la base de datos, debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos. El conjunto de la base de datos (definiciones, tablas, índices, y los propios datos), son guardados como un sólo fichero estándar en la máquina host. Este diseño simple se logra bloqueando todo el fichero de base de datos al principio de cada transacción [12, 6].

2.4. Desarrollo de una librería para el modelado y simulación de fotobiorreactores

Modelica

Es un lenguaje de modelado orientado a objetos que facilita el modelado y la simulación de sistemas físicos. Dicho lenguaje es implementado en herramientas con licencia comercial, como puede ser *Dymola*, y en herramientas de software libre como pueden ser *OpenModelica* o *JModelica*. Las características más importantes de *Modelica* [5] son:

- Descripción causal de modelos, es decir, basada en ecuaciones. La asignación de causalidad es realizada por la herramienta.
- La posibilidad de utilizar algoritmos. Permite la descripción de modelos con una asignación manual de causalidad, así como el uso de funciones que definen la causalidad.
- Características inherentes al lenguaje de modelización orientado a objetos, como la abstracción, la encapsulación, la modularidad, el principio de ocultación, el polimorfismo y las herencias múltiples.
- El uso de anotaciones para la descripción gráfica de los diferentes componentes así como la posibilidad de incluir documentación HTML.
- La posibilidad de implementar modelos híbridos, es decir, con partes continuas y discretas y el uso de eventos de tiempo y estado.
- Los modelos desarrollados en *Modelica* pueden ser conectados a otro software externo utilizando el estándar *FMI* (Functional Mock-up Interface). Esta característica es fundamental desde el punto de vista de la ingeniería de control ya que permite ejecutar cualquier algoritmo de control que se haya implementado utilizando software de propósito general como *Matlab*. *FMI* es un estándar independiente de herramientas que soporta tanto el intercambio de modelos como la co-simulación de modelos dinámicos utilizando una combinación de archivos XML y código C compilado [3].

Dymola

Se trata de una implementación del lenguaje *Modelica* con licencia comercial [2], las principales características de este entorno son:

- Simulación eficiente de modelos. Esta herramienta realiza manipulaciones simbólicas y traduce los modelos al lenguaje C, lo que permite un mejor rendimiento respecto a otros lenguajes interpretados.
- Mejoras de los algoritmos para la manipulación simbólica y la integración con otras herramientas.
- Los resultados se pueden representar mediante resultados gráficos y también mediante simulaciones en 3D.

- Una adaptación más rápida de la herramienta a las variaciones relativamente rápidas del estándar y sus librerías.
- Mayor facilidad de uso en comparación con otras alternativas como *JModelica*.
- Una mayor estabilidad en comparación con otras alternativas como *OpenModelica*.
- Una mayor variedad en la selección de los algoritmos de integración numérica.
- Posibilidad de interconectar los modelos *Dymola* con *Matlab/Simulink* mediante un bloque proporcionado por la propia herramienta.
- Una mayor eficiencia en los cálculos de las alternativas libres, como se muestra en [15].

Referencias

- [1] Web oficial de android studio. Último acceso 23/09/2018. URL <https://developer.android.com/studio/>.
- [2] Web oficial de dymola. Último acceso 23/09/2018. URL <https://www.3ds.com/es/productos-y-servicios/catia/productos/dymola/>.
- [3] Web oficial del estándar fmi. Último acceso 23/09/2018. URL <https://fmi-standard.org/>.
- [4] Web oficial de joptimizer. Último acceso 23/09/2018. URL <http://www.joptimizer.com/>.
- [5] Web oficial de modelica. Último acceso 23/09/2018. URL <https://www.modelica.org/>.
- [6] Web oficial de sqlite. Último acceso 23/09/2018. URL <https://www.sqlite.org/index.html>.
- [7] Web oficial de easyjava. Último acceso 23/09/2018. URL <https://www.um.es/fem/EjsWiki/>.
- [8] Web oficial de open source physics. Último acceso 23/09/2018. URL <http://www.opensourcephysics.org/>.
- [9] Web del proyecto unilabs. Último acceso 23/09/2018. URL <http://unilabs.dia.uned.es/>.
- [10] Entrada en wikipedia de android studio. Último acceso 23/09/2018, . URL https://es.wikipedia.org/wiki/Android_Studio.
- [11] Entrada en wikipedia de javascript. Último acceso 23/09/2018, . URL <https://es.wikipedia.org/wiki/JavaScript>.
- [12] Entrada en wikipedia de sqlite. Último acceso 23/09/2018, . URL <https://es.wikipedia.org/wiki/SQLite>.
- [13] J. Chacón, H. Vargas, G. Farias, J. Sánchez, and S. Dormido. Ejs, jil server, and labview: An architecture for rapid development of remote labs. *IEEE Transactions on Learning Technologies*, 8(4):393–401, 2015.
- [14] F. Esquembre. Easy java simulations: A software tool to create scientific simulations in java. *Computer Physics Communications*, 156(2):199–204, 2004.

- [15] J. Frenkel, C. Schubert, G. Kunze, P. Fritzson, M. Sjölund, and A. Pop. Towards a benchmark suite for modelica compilers: Large models. In *Proceedings of the 8th International Modelica Conference; March 20th-22nd; Technical University; Dresden; Germany*, number 63, pages 143–152. Linköping University Electronic Press, 2011.
- [16] K. H. Johansson. The quadruple-tank process: A multivariable laboratory process with an adjustable zero. *IEEE Transactions on control systems technology*, 8(3):456–465, 2000.
- [17] K. H. Johansson and J. L. R. Nunes. A multivariable laboratory process with an adjustable zero. In *American Control Conference, 1998. Proceedings of the 1998*, volume 4, pages 2045–2049. IEEE, 1998.
- [18] K. H. Johansson, A. Horch, O. Wijk, and A. Hansson. Teaching multivariable control using the quadruple-tank process. In *the 38th IEEE Conference on Decision and Control, 1999, Phoenix, AZ*, pages 807–812. IEEE, 1999.

Capítulo 3

Estrategias, técnicas y decisiones de diseño

3.1. Introducción

En este capítulo se explican las estrategias, técnicas y decisiones de diseño que se han seguido para alcanzar los objetivos que se planteaban en el primer capítulo.

3.2. Desarrollo de una herramienta abierta para experimentación de estrategias de modelado y control en Ingeniería Agrícola

En la Sección 1.2 se mostraba cual era la visión que se pretendía que un usuario del sistema tuviera del mismo. Sin embargo, la arquitectura real del funcionamiento interno del sistema tiene ciertas diferencias, tal y como se puede ver en la Figura 3.1. En dicha figura se pueden destacar tres elementos: el sistema real, explicado en la Sección 2.2; el entorno *SCADA*, cuyo controlador interno se explica en la presente sección; y la herramienta *EasyJava* cuyos puntos más importantes son también descritos a continuación.

3.2.1. Desarrollo del lazo interno de control

Es muy común en los sistemas industriales la necesidad de implementar un esquema de control en cascada [6], tal y como se muestra en la Figura 3.2. La necesidad de dicho control en cascada es para que los actuadores, en este caso las bombas, sigan las consignas del lazo de control superior. Dicho bucle interno será totalmente transparente al usuario. De esta forma, podrá considerar todo el lazo interno como una bomba ideal que responde a la referencia solicitada por el lazo superior, en este caso la referencia de flujo.

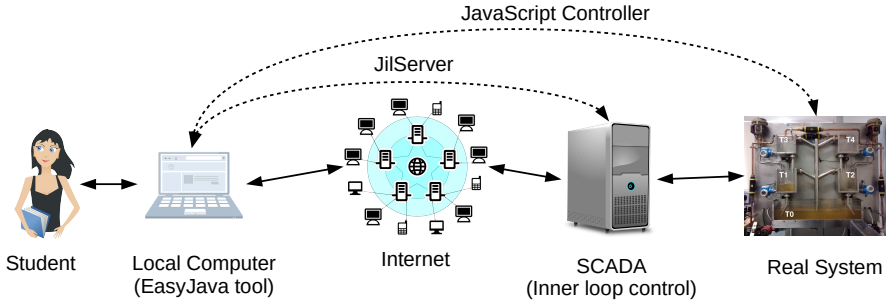


Figura 3.1 Vista esquemática de la implementación interna

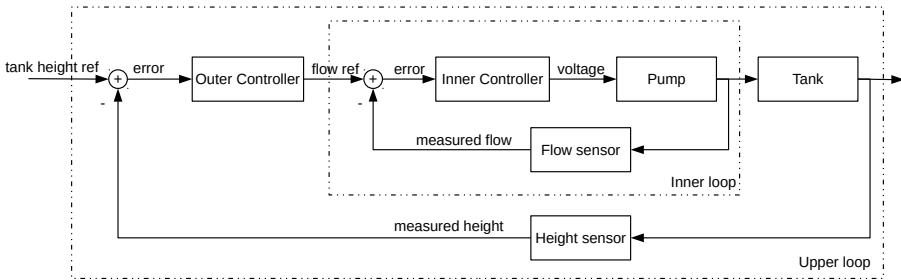


Figura 3.2 Esquema de control en cascada

Para diseñar una estrategia de control de lazo interno apropiada para el flujo de las bombas es necesario analizar el comportamiento de las mismas. Más específicamente, la entrada a cualquiera de las bombas es una tensión comprendida entre 0 y 10 V y la salida está directamente relacionada con esta tensión, la altura del tanque y la posición de las válvulas de tres vías. Es decir, el funcionamiento de las bombas depende de la configuración del sistema y de la cantidad de líquido que haya en el tanque inferior en el momento en que se inicie el experimento. Esto hace necesario un calibrado automático del sistema previo a la prueba de los controladores.

Tras realizar distintos experimentos en el sistema, y como se puede observar en la Figura 3.3, se llega a las siguientes conclusiones: las bombas funcionan a intervalos de 0.4 V estabilizando un único caudal dentro de dichos intervalos; dicha relación entre tensión y voltaje se puede observar que se ajusta con una función polinómica de segundo orden. Por otro lado, se puede ver también que si hacemos uso únicamente de tres puntos: punto de inicio, punto medio y punto de saturación, la nueva función polinómica que se obtiene es bastante similar a la primera.

En base a todos los puntos mencionados anteriormente, se pueden deducir princi-

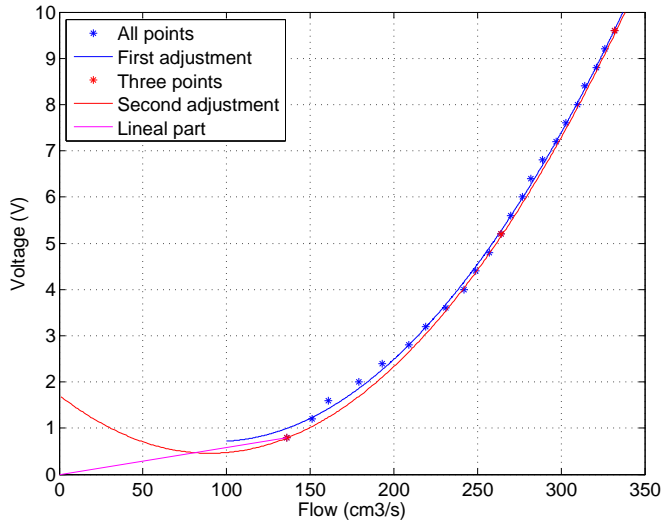


Figura 3.3 Ajuste de la dinámica de las bombas

palmente dos conclusiones. La primera, es que las bombas trabajan como si tuvieran únicamente 23 puntos de operación que proporcionan caudales fijos en función del voltaje de entrada, la posición de las válvulas y la altura del tanque inferior. Además, se puede considerar que la dinámica en cada uno de esos puntos es inmediata en relación al tiempo de retardo en la respuesta de las bombas. La segunda, es que se puede obtener dicha dinámica con únicamente tres puntos de voltaje predefinidos. Por tanto, para obtener la dinámica que tendrán las bombas con una configuración dada del sistema de experimentación, bastara con suministrar tres voltajes diferentes y medir el caudal obtenido. Como último apunte, cabe destacar que para voltajes en la zona muerta de las bombas, y con el objeto de que los controladores funcionen correctamente, se hará uso de una parte lineal.

Conociendo el funcionamiento de las bombas, y teniendo ya un mecanismo para obtener la dinámica cambiante al inicio de los experimentos, se ha desarrollado una estrategia de control específica para el caso que nos ocupa. Se puede establecer una relación entre el control PI y la estrategia de control diseñada. Esta estrategia, cuyo esquema se muestra en la Figura 3.4, puede dividirse en dos bloques diferentes que unidos llevarán al sistema a la referencia de flujo deseada.

El primer bloque, denominado *Polynomial*, obtiene de los polinomios calculados en la sintonización la tensión más cercana posible que proporciona el caudal deseado. Es necesario recordar que sólo se consideran incrementos de 0.4 V, y que los valores intermedios se descartan porque no implican una modificación del funcionamiento

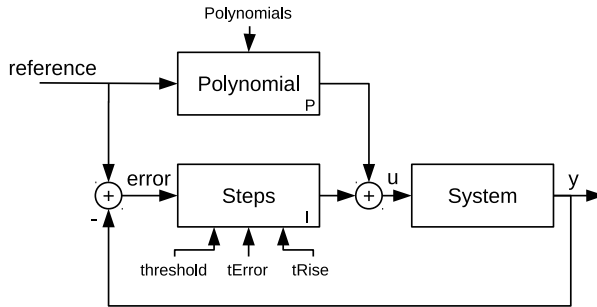


Figura 3.4 Esquema de control interno dentro del control en cascada

de la bomba. Este bloque puede ser equivalente a la parte proporcional de un controlador PI. El segundo bloque, denominado *Steps*, hace subir o bajar la señal de control en pasos de 0.4V en función del error de seguimiento. Este bloque, que puede ser comparable a la acción integral de un controlador PI, integra el tiempo, los períodos de muestreo, donde el error de seguimiento está por encima de un cierto umbral. Si se supera este umbral durante un tiempo, se sumará o restará a la señal de tensión un escalón de 0.4 voltios. Para sintonizar este controlador se utilizan los parámetros *threshold*, *tError* y *tRise* descritos en [9].

En conclusión, la estrategia diseñada conduce, con una fácil sintonización, a un controlador con una velocidad cercana a los límites físicos del actuador y con un error estacionario menor que la diferencia entre dos niveles de tensión. Los inconvenientes que se pueden observar son un retardo sobre la señal de referencia y un pequeño error de estado estacionario. Sin embargo, el retraso es un problema no solventable, si no se conoce el comportamiento futuro, y, el error de estado estacionario, en casi todos los casos, será menor que la sensibilidad de los sensores del medidor de flujo. La Figura 3.5 muestra la respuesta lograda con el controlador propuesto sobre el sistema real.

3.2.2. Interfaz EasyJava

La interfaz *EasyJava* es el único elemento con el que el alumno/investigador interactúa directamente. Usando esta herramienta el usuario puede crear códigos *JavaScript* que se pueden utilizar tanto en modo simulación como en modo real. Además, de forma transparente para el usuario, la herramienta anterior es capaz de comunicarse con la herramienta *SCADA* a través de Internet utilizando *JilServer*. Los motivos para utilizar *EasyJava* son su facilidad de uso, la posibilidad de interconexión con *Labview* a través de *JilServer* y la posibilidad de ser incluido como un

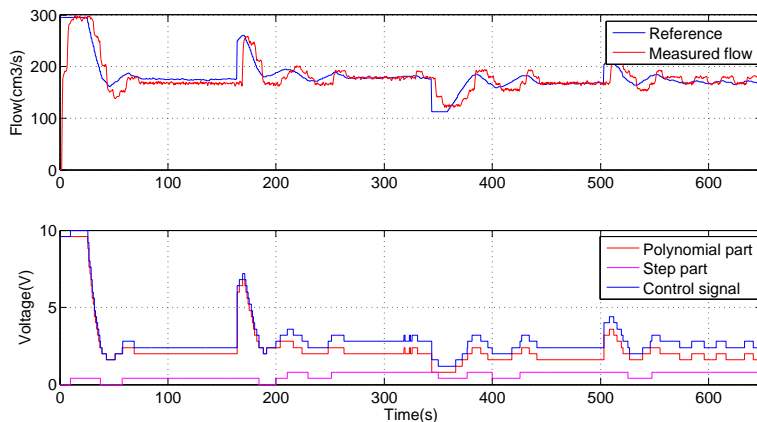


Figura 3.5 Funcionamiento real del lazo interno de control

applet en una web. En esta sección se explica someramente la interfaz y el funcionamiento de la misma.

La Interfaz Gráfica de Usuario (GUI) de la herramienta se divide en dos columnas. En la columna de la izquierda se muestra el sinóptico del sistema, es decir, una representación esquemática del sistema. Esta representación muestra los cambios en tiempo real del sistema, proporcionando los valores medidos, además de representar la altura de los tanques con una animación. Esto ocurre cuando la herramienta trabaja en modo de simulación o conectada al sistema real. Además, cuando funciona conectada al sistema real, se puede observar un vídeo en tiempo real del sistema, como se puede ver en la Figura 3.6. En la parte inferior de esta columna, hay tres botones para iniciar, detener y pausar los experimentos, así como para guardar los datos y cambiar al modo real. Asimismo, hay botones que permiten a los usuarios activar y desactivar las perturbaciones del sistema en cualquier momento del experimento.

Como se puede ver en la Figura 3.7, en la parte superior de la columna derecha hay dos gráficos que representan respectivamente los valores de caudal y altura de los tanques. En la parte inferior de esta columna hay tres pestañas: *Manual*, *Control* e *Inicialización*. Más concretamente, en la pestaña *Manual* se pueden modificar los puntos de ajuste de altura actuales y las posiciones de las válvulas de tres vías, así como los valores de altura y caudal. En las otras dos pestañas, *Inicialización* y *Control*, se puede cargar, guardar y editar el código *JavaScript* para la inicialización y para el bucle de control; tal y como puede observarse en la Figura 3.8. Como medida de seguridad, el código *JavaScript* sólo puede ejecutarse en el sistema real si ha sido previamente probado en modo de simulación.

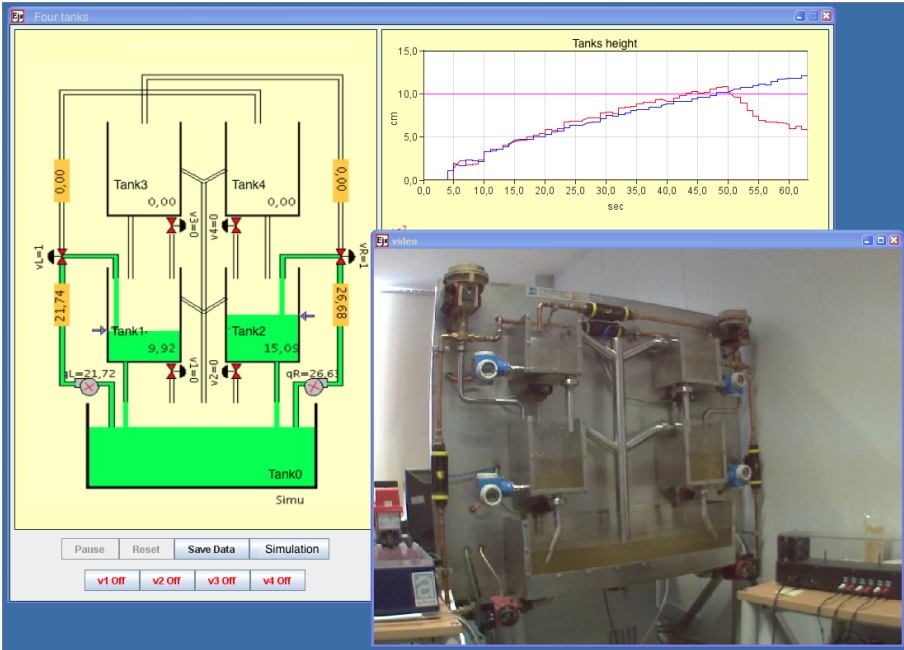


Figura 3.6 Funcionamiento de la interfaz en modo real

En lo que refiere al funcionamiento de la herramienta, cuando se activa el modo real, la herramienta *EasyJava* establece una conexión con el *SCADA* del sistema real de cuatro tanques y le envía el código *JavaScript* que el usuario desea ejecutar en el sistema real. Posteriormente, este sistema *SCADA* comienza a ejecutar el código *JavaScript* y envía información sobre el estado de los sensores y actuadores a la herramienta *EasyJava* cada vez que se realiza un muestreo. Además, los valores de la válvula de perturbación y de la válvula de cierre también se envían, cada período de muestreo. Por lo tanto, en cada período de muestreo, los datos recibidos se representan y se escriben en un archivo *Matlab*. La comunicación entre la herramienta *EasyJava* y el *SCADA* del sistema real se realiza mediante la herramienta *JilServer* [1, 2, 4]. Más específicamente, el período de muestreo de este sistema es igual a un segundo, tanto para la adquisición como para el envío de datos. No obstante, la velocidad de actualización de los datos y del streaming de vídeo dependerá del estado de la red y de la distancia del usuario al sistema real, es decir, el usuario recibirá la información en tiempo real o con un pequeño retraso.

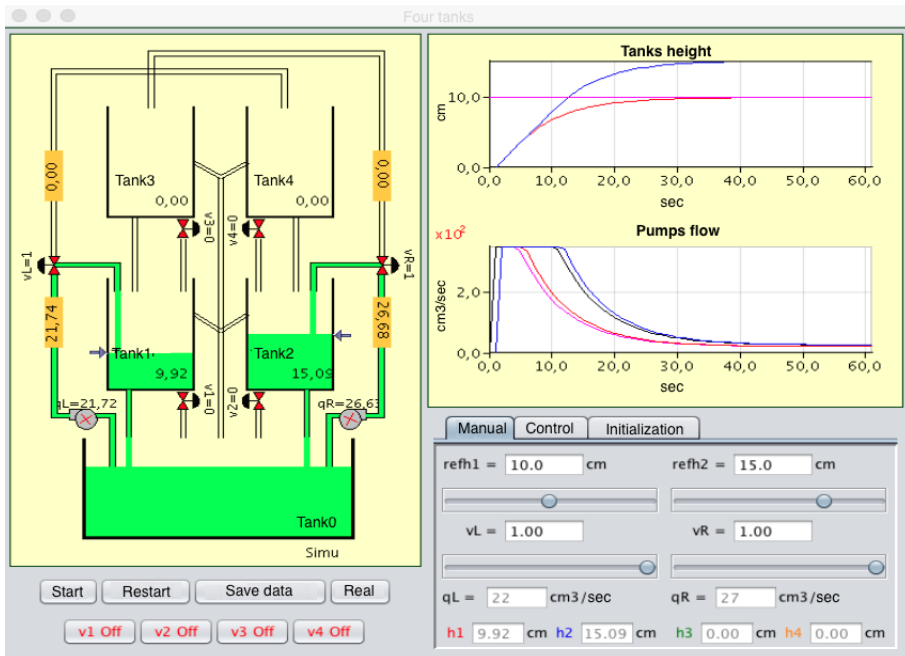


Figura 3.7 Vista general de la interfaz *EasyJava*

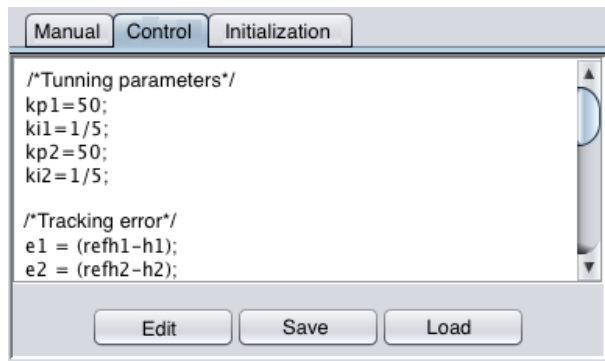


Figura 3.8 Introducción de código *JavaScript* en la herramienta

3.3. Desarrollo de una aplicación para la gestión óptima del fertirriego

La aplicación móvil presentada en este trabajo ofrece varias posibilidades y funcionalidades que mejoran la aplicación de fertirrigación y la experiencia del usuario. El manejo de este tipo de herramientas y procesos es generalmente tedioso y complejo, por lo que los agricultores se han quejado de la dificultad y el tiempo que les lleva su uso. El objetivo principal en el desarrollo de esta aplicación, véase Figura 3.9.a, es facilitar la comprensión de la tarea de suministro de agua y fertilizantes a los agricultores, estudiantes e ingenieros agrónomos. Es por esto que se da gran importancia al diseño y facilidad de uso de la herramienta y es, precisamente, dicho diseño y modo de uso lo que se explica en esta sección. Para otros apartados importantes como pueden ser las ecuaciones utilizadas o configuración del optimizador se puede consultar [8].

El primer paso para el sistema de decisión es definir la simulación, véase Figura 3.9.b. En esta etapa, la información principal se ordena por medio de diferentes niveles: el primer nivel se asocia al nombre del agricultor, lo que ayudará a identificar sus experimentos almacenados en la base de datos. El segundo nivel está compuesto por los diferentes invernaderos relacionados con el agricultor que se define en el primer nivel. Este nivel incluye las principales características tanto del invernadero (necesarias para el modelo de abastecimiento de agua) como del equipo de riego, véase Figura 3.9.c. A continuación, el tercer nivel establece la temporada de cultivo, véase Figura 3.10.a, con el fin de obtener información sobre el crecimiento de los cultivos: la variedad y la densidad de las plantas, y la solución óptima establecida para ello, véase Figura 3.10.b. Al final, se genera un informe final que incluye los principales resultados.

Además, el análisis del agua de riego, véase Figura 3.10.c, se introduce manualmente en la aplicación y se almacena en la base de datos para que pueda utilizarse con más de un cultivo o invernadero. El análisis del agua de riego y la solución óptima se utilizan para obtener los diferentes componentes de nutrientes que deben ser suministrados por el sistema de fertirrigación ; estos se utilizan para obtener los resultados finales ayudando a establecer el peso de los diferentes fertilizantes.

Finalmente, es posible generar un informe resumido, véase Figura 3.11.a, que contiene un resumen de los valores numéricos estimados asociados al peso de los diferentes fertilizantes y los resultados obtenidos del análisis de drenaje, destacando los problemas más significativos en relación con el suministro o la absorción de los diferentes nutrientes. Por otro lado, la Figura 3.11.b muestra la distribución de los diferentes fertilizantes en tanques teniendo en cuenta las incompatibilidades, el porcentaje de distribución en los tanques y el cálculo del peso del fertilizante.

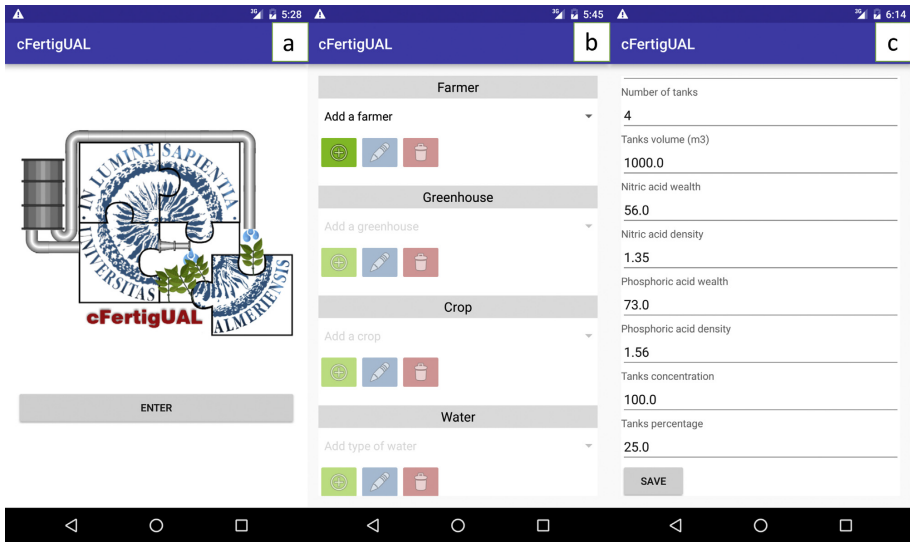


Figura 3.9 Algunas funcionalidades de la aplicación

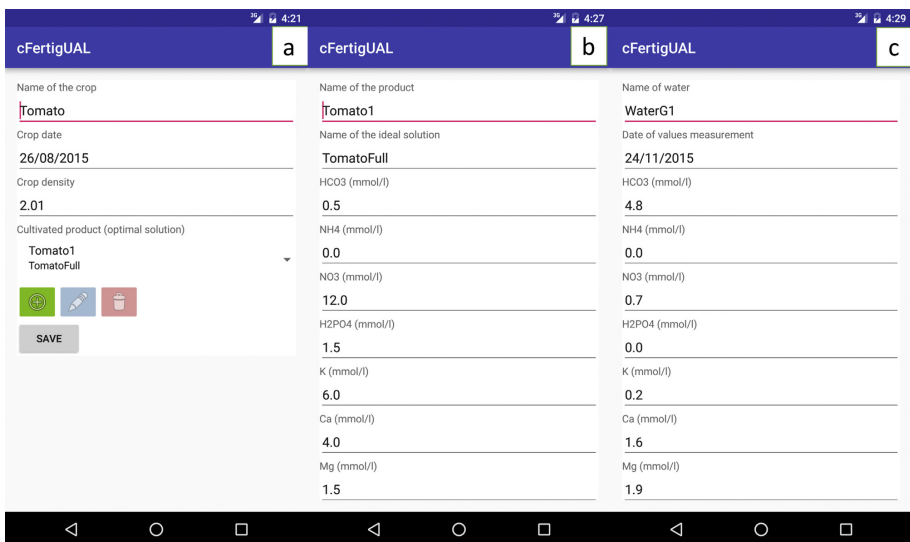


Figura 3.10 Principales características y resultados



Figura 3.11 Resultados obtenidos por la aplicación

3.4. Desarrollo de una librería para el modelado y simulación de fotobiorreactores

A continuación se explican las decisiones tomadas a cabo para el desarrollo de la librería intentando hacer uso del potencial que aportan un lenguaje como *Modelica* o una herramienta como *Dymola*. Para una descripción completa de las ecuaciones que son utilizadas para el modelado de cada uno de los fenómenos y componentes se puede consultar [7]. Dichas ecuaciones guardan gran similitud con las utilizadas en [3, 5], ya que son las que se han utilizado como base para desarrollar la librería.

Para optimizar el diseño de la librería ha sido necesario aprovechar las propiedades orientadas a objetos del lenguaje de modelado. Los fundamentos de esta estrategia pueden resumirse en los principios de modelado modular y jerárquico. El modelado modular se basa en la idea de “divide y vencerás”. Por lo tanto, un sistema puede descomponerse en diferentes sub-sistemas y sub-modelos, por ejemplo, dos sub-modelos diferentes podrían ser una sección de tubería y un tanque. Las principales ventajas que aporta la modularidad son una mayor facilidad para el desarrollo y validación de modelos, así como la posibilidad de composición, reutilización y abstracción, que permite utilizar cada submodelo sin conocer sus detalles internos. Para la composición de los distintos submodelos, es necesario establecer una comunicación entre ellos a través de interfaces. Más concretamente, estas interfaces se diseñarán en función de los distintos ámbitos descritos en Sección 3.4.1. Por otro lado, el modelado jerárquico nos permite construir modelos y submodelos de bajo a alto nivel, como se muestra en la Sección 3.4.2. Por lo tanto, a partir de submode-

los básicos o atómicos, es posible componer modelos cada vez más complejos hasta construir el modelo de sistema completo. En nuestro caso, los modelos atómicos son los que describen los fenómenos físico-químicos descritos por las ecuaciones básicas. El diseño de la biblioteca se puede describir desde tres perspectivas diferentes: los dominios implementados, el esquema de composición de clases y la arquitectura de los paquetes.

3.4.1. Dominios

Se han considerado cuatro dominios: gas, (*Gas*), líquido, (*Liq*), calor, (*Heat*) y líquido con biomasa, (*Bio*). Por lo tanto, los conectores, las interfaces, las clases básicas y algunas partes se describen en función de estos dominios. Además, algunos dominios pueden incluir otros, es decir, incluir relaciones de herencia que hacen que algunas variables de dominios estén presentes en otros. Por lo tanto, el dominio *Heat* se incluye en el dominio *Liq* y éste se incluye en el dominio *Bio*. Lo anterior está bien representado en los conectores diseñados, cuyos pares de variables de potencial y flujo, variables *across* y *through* respectivamente, se muestran en la Tabla 3.1.

Tabla 3.1 Variables de los conectores

Connector	Across variable	Through variable
<i>GasCon</i>	Gas volume - $V_g [m^3]$	Gas flow - $fV_g [m^3 s^{-1}]$
	O_2 concentration - $O_{2g} [mol m^{-3}]$	O_2 flow - $fO_{2g} [mol s^{-1}]$
	CO_2 concentration - $CO_{2g} [mol m^{-3}]$	CO_2 flow - $fCO_{2g} [mol s^{-1}]$
	N_2 concentration - $N_{2g} [mol m^{-3}]$	N_2 flow - $fN_{2g} [mol s^{-1}]$
<i>HeatCon</i>	Temperature - $T [K]$	Heat flow - $fQ [J s^{-1}]$
<i>LiqCon</i>		
<i>BioCon</i>		
<i>LiqCon</i>	Liquid volume - $V_l [m^3]$	Liquid flow - $fV_l [m^3 s^{-1}]$
<i>BioCon</i>		
<i>BioCon</i>	Biomass concentration - $C_b [Kg m^{-3}]$	Biomass flow - $fCb [Kg s^{-1}]$
	O_2 concentration - $O_2 [mol m^{-3}]$	O_2 flow - $fO_2 [mol s^{-1}]$
	CO_2 concentration - $CO_2 [mol m^{-3}]$	CO_2 flow - $fCO_2 [mol s^{-1}]$
	C_T concentration - $C_T [mol m^{-3}]$	C_T flow - $fC_T [mol s^{-1}]$
	H^+ concentration - $H^+ [mol m^{-3}]$	H^+ flow - $fH^+ [mol s^{-1}]$

3.4.2. Esquema de composición

El esquema de composición de clases, representado en la Figura 3.12, se divide en cuatro niveles. En el nivel 0 están los conectores, (*Connectors*), e interfaces,

(*Interfaces*), que se han creado a través de los primeros. Las interfaces describen las conexiones con el exterior de una clase. Además, en este nivel también se encuentra la representación gráfica, los iconos y las ecuaciones que describen la dinámica de cada clase.

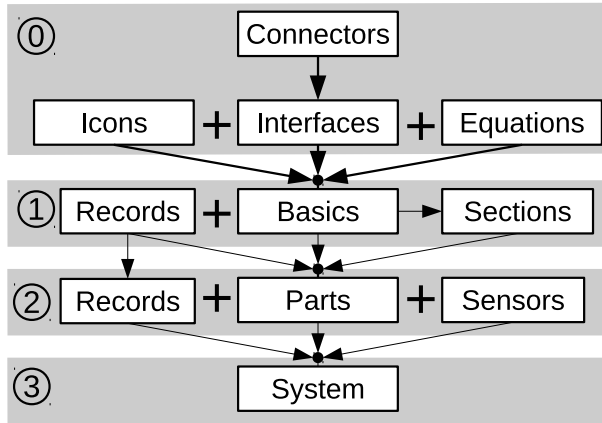


Figura 3.12 Composición de clases

Mediante un conjunto de ecuaciones, una interfaz para la transferencia de información más un icono para su descripción gráfica, se crea cada clase básica, (*Basics*). Este tipo de clases describen los volúmenes de control y los fenómenos de transporte. Estas clases están en el nivel 1 con las clases de secciones, (*Sections*). Las clases de secciones son clases compuestas que no pueden ser consideradas como partes. Las clases de registros, (*Records*), también en el nivel 1, son conjuntos de parámetros y constantes. En la Figura 3.13 se muestra la composición de una sección tubular. Más concretamente, en referencia a su conexión al exterior, este apartado consta de cuatro conectores que se utilizan para conectar a otros bloques y formar un tubo completo. Asimismo, consta de dos entradas que proporcionan información sobre la temperatura exterior y la radiación solar. El bloque rojo realiza los cálculos de transferencia de calor con el medio ambiente. Los bloques verdes estiman, de izquierda a derecha, la cantidad o concentración de diversos elementos en el medio acuoso, la generación de biomasa por fotosíntesis y el transporte desde el medio acuoso hasta los componentes conectados respectivamente. Finalmente, el bloque bicolor calcula la transferencia de masa entre el medio acuoso y el gaseoso.

Para la composición de las partes, (*Parts*), se utilizan las clases básicas, secciones y otras partes. Además, para la configuración de los parámetros se puede utilizar un conjunto de registros, que pueden estar compuestos por otro registro del nivel anterior. Todo lo anterior, junto con la clase de sensores, crean el nivel 2 del esquema de composición. Finalmente, el nivel 3 del esquema de composición define sistemas

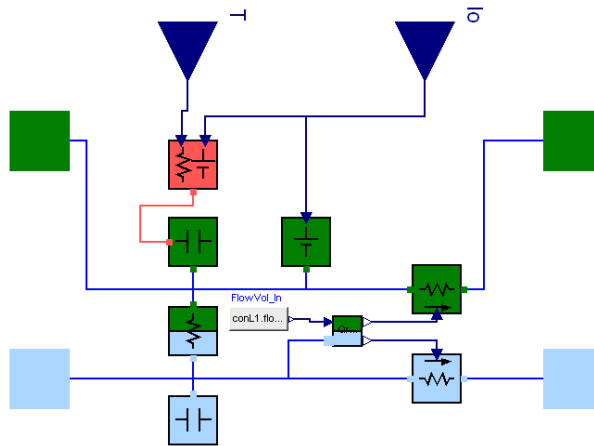


Figura 3.13 Composición interna de una sección de tubería

complejos, (*Systems*), como puede ser un fotobiorreactor cerrado. Estos sistemas están contruidos con piezas y sensores para la medición de potenciales y caudales, y se configuran mediante registros. En la Figura 3.14 se muestra la construcción interna de un fotobiorreactor completo usando componentes de niveles inferiores.

3.4.3. *Arquitectura*

A continuación se presenta un resumen del contenido de los principales paquetes que forman la librería:

- *Basic*: Este paquete contiene los elementos básicos.
- *Examples*: Este paquete contiene el ejemplo de un fotobiorreactor completo para probar la biblioteca.
- *Icons*: Descripciones gráficas usadas en el resto de paquetes.
- *Interfaces*: Conectores y el conjunto de modelos parciales que describen las interfaces.
- *Parts*: Aquellos modelos similares a los componentes físicos reales, es decir, una columna, un bucle, bombas, tanques, etc.
- *Users Guide*: Información más relevante sobre la biblioteca y su uso.
- *Utilities*: Conjunto de herramientas para facilitar las simulaciones, por ejemplo, herramientas para leer y procesar archivos de datos.

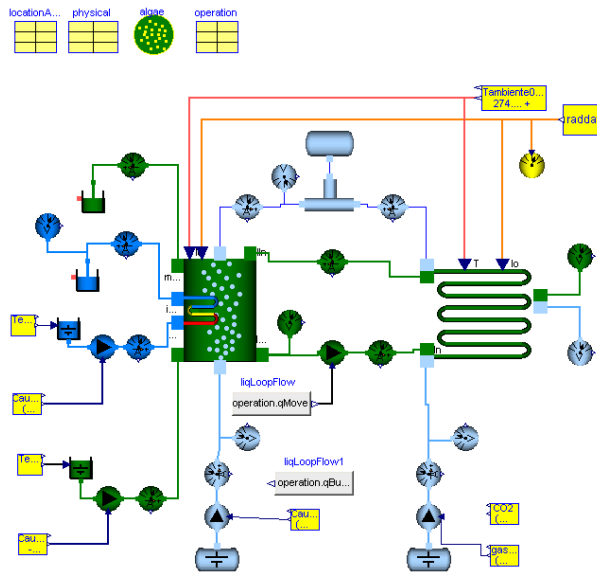


Figura 3.14 Composición interna de un fotobiorreactor completo

Referencias

- [1] Web oficial de jilserver. Último acceso 23/09/2018. URL <https://github.com/UNEDLabs/jil-server/>.
- [2] J. Chacón, H. Vargas, G. Farias, J. Sánchez, and S. Dormido. Ejs, jil server, and labview: An architecture for rapid development of remote labs. *IEEE Transactions on Learning Technologies*, 8(4):393–401, 2015.
- [3] T. Costache, F. G. A. Fernández, M. Morales, J. Fernández-Sevilla, I. Stamatini, and E. Molina. Comprehensive model of microalgae photosynthesis rate as a function of culture conditions in photobioreactors. *Applied microbiology and biotechnology*, 97(17):7627–7637, 2013.
- [4] L. de la Torre, M. Guinaldo, R. Heradio, and S. Dormido. The ball and beam system: A case study of virtual and remote lab enhancement with moodle. *IEEE Transactions on Industrial Informatics*, 11(4):934–945, 2015.
- [5] I. Fernández, F. Ación, J. Fernández, J. Guzmán, J. Magán, and M. Berenguel. Dynamic model of microalgal production in tubular photobioreactors. *Biore-source technology*, 126:172–181, 2012.
- [6] B. A. Ogunnaike and W. H. Ray. *Process dynamics, modeling, and control*, volume 1. Oxford University Press New York, 1994.
- [7] A. Pérez-Castro, J. Sánchez, F. G. Ación, I. Fernández, and J. L. Guzmán. Photobiolib: una librería de modelica para el modelado y simulación de fotobiorreactores. In *XXXV Jornadas de Automática (Valencia)*, pages 3–10, 2014.

- [8] A. Pérez-Castro, J. Sánchez-Molina, M. Castilla, J. Sánchez-Moreno, J. Moreno-Úbeda, and J. Magán. cfertigual: A fertigation management app for greenhouse vegetable crops. *Agricultural Water Management*, 183:186–193, 2017.
- [9] A. Perez-Castro, J. Sanchez-Moreno, and M. d. M. Castilla. Development of an open experimentation tool based on javascript for the control of a four-tank plant. *Computer Applications in Engineering Education*, 26(2):228–238, 2018.

Capítulo 4

Conclusiones y futuros trabajos

4.1. Introducción

En este capítulo se explican las conclusiones derivadas de cada uno de los artículos publicados durante el desarrollo de esta tesis doctoral así como los futuros trabajos que se pueden desprender de los mismos.

4.2. Desarrollo de una herramienta abierta para experimentación de estrategias de modelado y control en Ingeniería Agrícola

En este trabajo se muestra el desarrollo de una herramienta que permite a investigadores y estudiantes acceder a un sistema de experimentación real, concretamente un sistema acoplado de cuatro tanques, y que permite la prueba de controladores o hacer experimentos para identificación de modelos. Además, para el lazo interno de control se ha diseñado una estrategia de control adaptativa basada en la interpolación de polinomios para la regulación de las bombas de caudal. Tal y como se explica en el artículo, ésta tiene ciertas similitudes con un control PI clásico.

Más concretamente, la herramienta desarrollada permite trabajar en modo simulación y operación real y, lo que es más importante, el código implementado para simulación puede ser directamente usado en el modo operación real sin realizar cambio alguno en el mismo. Además, la herramienta recibe y muestra, en tiempo real, los datos sensorizados en el sistema de experimentación como resultado de la ejecución del controlador así como una señal de video en directo. Por otro lado, los datos recibidos también pueden ser guardados en un fichero para su posterior análisis.

Para la programación por parte del usuario en la herramienta desarrollada se ha

elegido el lenguaje *JavaScript*. Gracias al uso de dicho lenguaje es posible desarrollar cualquier tipo de controlador y de esta forma el usuario no está limitado, como puede ocurrir con herramientas similares, a la sintonización de parámetros de un controlador ya establecido. Esto permite un gran abanico de posibilidades con una única instalación. De esta forma, es posible realizar un gran número de experimentos en un mismo sistema y sin necesidad de cambiar de herramienta. Dichos experimentos abarcan desde la sintonización de parámetros de un controlador dado, al diseño e implementación de una estrategia de control predictivo basada en modelo pasando por la programación de controladores multivariables o la realización de experimentos para su utilización posterior con técnicas de identificación de sistemas. Por lo tanto, los límites están marcados por las posibilidades que nos brinda el sistema real y no por la herramienta o el lenguaje usado.

En lo referente a posibles mejoras o futuros trabajos hay dos líneas principales. La primera de ellas es permitir el uso de otros lenguajes de programación, como por ejemplo *Matlab*, para la implementación de controladores más complejos por parte de investigadores, haciendo innecesario la conversión de los programas ya implementados en *Matlab* a *JavaScript*. La segunda línea es la incorporación de funciones *JavaScript* de alto nivel prediseñadas con el fin de facilitar el desarrollo de estrategias de control más complejas, como por ejemplo, funciones de optimización para ser utilizadas en controladores predictivos basados en modelo.

Otra posible línea de trabajo futuro sería la de extrapolar la herramienta diseñada a otros sistemas de experimentación. Esto provocaría dos grandes ventajas, el primer efecto sería el de proporcionar a dichos sistemas una interfaz sencilla y polivalente para la experimentación en los mismos. La segunda ventaja hace referencia a la posibilidad de usar la misma metodología de trabajo en distintas plantas, e incluso reutilizar código. De esta forma, el usuario podrá centrarse en las técnicas a desarrollar, sin tener que aprender cada vez el funcionamiento de la herramienta, y poder ver cómo las distintas estrategias desarrolladas se adaptan a distintos sistemas.

4.3. Desarrollo de una aplicación para la gestión óptima del fertirriego

En esta parte de la tesis se ha desarrollado una aplicación para la gestión del fertirriego en invernaderos. Más concretamente, se trata de una aplicación desarrollada de forma nativa para dispositivos Android con el objetivo de ayudar a la gestión del fertirriego de cultivos en invernaderos. Esto es, el cálculo óptimo de los nutrientes que se han de aportar al agua de riego para, conociendo las cualidades del agua y las necesidades del cultivo, garantizar las necesidades de los mismos con el mínimo coste posible. Los cálculos óptimos son realizados mediante un algoritmo de programación lineal que ha sido implementado haciendo uso de la librería *Java JOptimizer*. Además, en el artículo que corresponde a este trabajo se ha incluido un

caso práctico de estudio. En dicho ejemplo, la aplicación calcula los requerimientos de agua y fertilizantes para un cultivo de tomate bajo las condiciones de crecimiento en un invernadero. La aplicación desarrollada, *cFertigUAL*, ofrece otras ventajas como la configuración de diferentes sistemas de fertirrigación en función de la complejidad del sistema, la posibilidad de generar una base de datos para el análisis de aguas de riego, drenaje y suelos, la introducción y almacenamiento de diferentes soluciones mixtas óptimas para cada cultivo y la estimación del agua a sustituir por el riego.

Pero quizás la mayor ventaja del sistema implementado no está en la optimización de los cálculos. La mayor ventaja de la herramienta implementada es que proporciona una interfaz muy sencilla y permite su utilización en dispositivos móviles. Esto permite que el sistema no sea solo accesible a expertos en la materia sino a cualquier usuario potencial. De poco sirve desarrollar el sistema más potente del mercado si las necesidades de conocimientos, costes o acceso al mismo hacen que no pueda ser utilizado.

Los posibles trabajos futuros estarían sobretodo orientados a la inclusión de nuevas funcionalidades a la aplicación. La que quizás suponga una mayor diferenciación sería la de incluir una lista de fertilizantes editables con sus precios asociados. Con dicha lista la aplicación podría calcular no solo el uso óptimo de fertilizantes sino que también se podría hacer una optimización multiobjetivo considerando variables económicas, así como hacer predicciones de costes a futuro. Otra posibilidad es la inclusión de datos climáticos externos proporcionados por estaciones meteorológicas, incluyendo un pronóstico de las principales variables climáticas externas. Dichos datos permitirían obtener predicciones del suministro de agua. Por último, otra posible mejora sería la inclusión de diferentes modelos de absorción de nutrientes para ayudar a decidir la solución óptima de los mismos.

4.4. Desarrollo de una librería para el modelado y simulación de fotobiorreactores

Como resultado de esta parte de la tesis se ha creado una librería que permite el modelado y simulación de fotobiorreactores. Las principales ventajas que proporciona dicha librería se comentan a continuación, algunas de las mismas serán resultado de las decisiones de diseño que se han tomado y de su implementación y otras vendrán heredadas del lenguaje y herramientas seleccionados para su desarrollo.

Como ya se ha comentado, algunas de las ventajas que tiene la librería desarrollada están muy influenciadas por el lenguaje utilizado para su desarrollo. Quizás se pueden destacar dos: Una gran velocidad de simulación; y su facilidad para la interconexión con otros modelos o librerías. En lo que se refiere a la velocidad de simu-

lación, el uso del lenguaje *Modelica* y, más específicamente, del entorno *Dymola*, permite simular modelos con miles de ecuaciones y largos periodos de simulación bajo pequeños tiempos de muestreo. Esto es posible principalmente gracias a las tareas de preprocesamiento que realiza *Dymola* sobre las ecuaciones diferenciales previo a su traducción a lenguaje C sin necesidad de la intervención del usuario. No obstante, en el buen desempeño de la librería también tienen bastante importancia la calidad de la implementación, el buen uso de algoritmos y el control del instante en que producen los eventos de estado.

Respecto a la facilidad para la interconexión con otros modelos y librerías, los modelos creados con la librería *PhotoBioLib* pueden ser conectados a otros componentes disponibles en *Modelica*, como pueden ser bloques para el procesado de señales o controladores, o con otras librerías, como por ejemplo, una que modele la temperatura en el interior de un invernadero para simular el comportamiento del fotobiorreactor. Esto además está facilitado por los sensores y actuadores que se han implementado en la propia librería, así como por las interfaces de más bajo nivel.

Las decisiones de diseño son también responsables de algunas de las virtudes de la librería. Una de dichas decisiones fue el uso de modelos basados en primeros principios. Dichos modelos tienen ventajas sobre el uso de modelos basados en funciones de transferencia, modelos de cajas negras u otras estrategias. Por ejemplo, permiten el acceso a información de estados internos. Otro punto a destacar es lo que respecta a las virtudes del diseño de la librería es su alto potencial de configuración. El que se implementen componentes que sean representación de partes físicas del sistema, a través de una construcción jerárquica, tiene dos grandes ventajas. Por un lado, se pueden usar dichos componentes para montar sistemas como se haría en la realidad. Por otro lado, se pueden construir nuevos componentes o hacer variaciones de los existentes de una forma muy sencilla. Lo anterior conduce a otro de los puntos fuertes de la librería, y es la gran similitud con la realidad. Como ya se ha comentado, se pueden construir sistemas añadiendo secciones y sensores como si se tratara de un sistema real. Esto permite tanto simular sistemas reales como el crear nuevos prototipos para evaluarlos antes de construirlos.

En conclusión, se ha desarrollado una librería con un gran potencial y facilidad de uso que permite que no solo los expertos en la materia le saquen partido, sino que pueda ser usada por cualquier usuario sin ningún conocimiento previo sobre programación, ya que solo es necesario arrastrar y conectar los diferentes componentes para crear modelos de sistemas reales.

En lo que se refiere a futuros trabajos estos estarían relacionados con la ampliación de la librería. Quizás la ampliación más importante sería el desarrollo de nuevos componentes y ejemplos para la simulación de otro tipo de fotobiorreactores. La librería actual tiene los componentes necesarios y un ejemplo de un fotobiorreactor tubular cerrado. El siguiente paso natural sería la inclusión de las partes necesarias para la creación de un ejemplo de fotobiorreactor abierto.

Otro punto importante sería el de proporcionar a los modelos de fotobiorreactores la información externa necesaria para su funcionamiento. Esto es, de información externa al sistema, como puede ser la inclusión de modelos para la simulación de la radiación solar y la temperatura exterior en cualquier parte del mundo en una fecha dada. Pero también de información interna de decisión del sistema, como puede ser la implementación de un paquete de controladores. Especialmente un paquete para de control predictivo basado en modelo.

Por último, algo que ayudaría bastante al uso práctico y real de la librería sería la inclusión de herramientas y procesos concretos que facilitarían la sintonización de parámetros y la validación de modelos con datos reales de una forma más o menos automatizada.

4.5. Conclusiones generales

A lo largo de la tesis, y en los *preprints* al final de la misma, se han presentado tres desarrollos que suponen una contribución al modelado, simulación y control en ingeniería agrícola. Aunque en principio puede parecer que sus temáticas son totalmente diferentes, los tres trabajos que se han presentado en esta tesis tienen un hilo conductor, y es que están orientados a la transferencia de conocimiento y la mejora en educación en Ingeniería Agrícola. Además de esto, según el uso que se le da a las herramientas podrán suponer una herramienta para el sector productivo o incluso herramientas de experimentación para probar desarrollos en investigación. No obstante, no hay que olvidar que, aunque en este caso se haya buscado una orientación más práctica, nada de esto habría sido posible sin la adquisición previa de los conocimientos relacionados con las técnicas usadas en el área del modelado, simulación y control, así como la realización de una serie de tareas de investigación que han permitido decidir cuáles son las soluciones más eficientes para los problemas abordados en el desarrollo de esta tesis doctoral.

Un punto que no se suele discutir en las publicaciones científicas, y que, en opinión del autor, debería tener una mayor relevancia, es la explicación de los problemas que han surgido durante la investigación. En otras palabras, la explicación de los trabajos o las líneas de investigación que se han descartado porque se llegó a la conclusión de que no se alcanzaría una solución factible. En este caso, tampoco se han incluido en las publicaciones principales. Más concretamente, el motivo de que no se hayan incluido es que, al tratarse de trabajos prácticos, dichas aseveraciones podrían haber quedado desfasadas antes incluso de la publicación final de los trabajos. Las dos líneas principales que se han abandonado han estado relacionadas con el trabajo en la librería *PhotoBiolib*.

La primera de estas líneas abandonadas fue la del uso de herramientas libres como

OpenModelica. El motivo de buscar el uso de dicha herramienta en el desarrollo de la librería era las grandes ventajas que esto aporta. Más detalladamente, la primera gran ventaja es el ahorro en licencias en el desarrollo. De la anterior deriva otra gran ventaja, y es que la difusión de una librería que necesita de una herramienta de un coste elevado para ser utilizada es más complicada precisamente por los costes que supone el acceso al uso de la misma. Por último, el uso de herramientas libres permite el desarrollo de herramientas adicionales como la que se hace referencia en la segunda línea abandonada. A todo lo anterior hay que añadir lo que es quizás una consideración personal del doctorando, y es que toda investigación en el ámbito público debe intentar, en la medida de lo posible, huir del uso de herramientas propietarias que supongan tener a un público cautivo de dichas herramientas a la hora de usar los avances en investigación subvencionados con dinero público.

Si bien es verdad que *OpenModelica* es un proyecto bastante avanzado y que es de elogiar el trabajo que hacen sus desarrolladores, no es menos cierto que no es fácil competir con herramientas comerciales que tengan tras de ellas grandes empresas. No obstante, en el desarrollo de la presente tesis se han encontrado problemas que han hecho abandonar el uso de dicha herramienta en la implementación de la librería. Lo más importante han sido errores en la implementación de elementos básicos que en algunas versiones provocaban que incluso los ejemplos de la librería estándar de *Modelica* no funcionaran correctamente. Esto tenía un importante efecto secundario, y es que lo anterior unido a la dificultad que supone la depuración en *Modelica*, creaba la desconfianza de no saber si los errores durante la implementación eran del propio desarrollo de la librería o de problemas con la librería. Por otro lado, aunque menos importante, los algoritmos de simulación están menos optimizados que en su alternativa comercial y esto es algo que también frena el desarrollo y depuración de la librería. Todo lo anterior hay que tenerlo en cuenta considerando que se está hablando del estado de la herramienta en el momento en que se inició la presente tesis y que en este tiempo la herramienta puede haber mejorado mucho, y lo que aquí se expone puede haber cambiado bastante.

La segunda línea que se ha abandonado fue la del desarrollo de herramientas de educación e investigación a partir de librería *PhotoBioLib*. La idea era la de exportar modelos de ejemplo desarrollados con la librería a herramientas web implementadas en *JavaScript*, dejando acceso a la modificación de parámetros para ver como los mismos afectaban a la simulación. Esto habría permitido la creación de un catálogo de herramientas de educación y experimentación relacionadas con los fotobiorreactores con una gran accesibilidad y facilidad de desarrollo. Más concretamente, la línea se basaba en el uso del código de este proyecto en *GitHub* [2]. El funcionamiento de dicho proyecto se basa en el uso de *Emscripten* [1] para traducir el código C generado por *OpenModelica* a código *JavaScript*. Esta línea se abandono porque, aunque en la página del proyecto tienen varias ejemplos que funcionan a la perfección, tras hacer varias pruebas no se consiguió reproducir los resultados. Tras hablar con el creador del proyecto se llegó a la conclusión de que para poder reproducirlo se necesitaban unas condiciones muy especiales de versiones de *OpenModelica*, que

además no estaban disponibles en el sitio oficial. A lo anterior había que añadir los problemas de *OpenModelica* que se ha comentado anteriormente.

Como conclusión a lo anterior la recomendación del autor es que, aunque se debe converger al uso de herramientas libres en cualquier ámbito de la investigación, en el caso del desarrollo de librerías de *Modelica* es mejor asumir el coste de licencias versus el coste que supone en tiempo el desarrollo en alternativas libres. Sin embargo, una vez el desarrollo este completo se puede intentar exportarlo a herramientas libres de forma que sea más sencilla su divulgación y su uso en educación.

Adicionalmente a lo que se ha presentado a lo largo de la tesis, en el Capítulo 5 se pueden consultar otras publicaciones, que si bien no se han considerado con bastante entidad como para ser destacadas en el desarrollo de la presente tesis, suponen también pequeños aportes al modelado, simulación y control de sistemas. Entre estos aportes se puede destacar el del diseño de un mecanismo para la corrección de modelos no lineales variantes en el tiempo [3]. Dicha estrategia puede tener usos polivalentes, desde su aplicación en el control predictivo, hasta sistemas de supervisión pasando por su uso para la creación de sensores virtuales.

Referencias

- [1] Web oficial de emscripten. Último acceso 23/09/2018. URL <http://kripken.github.io/emscripten-site/>.
- [2] Web oficial del proyecto openmodelica to javascript. Último acceso 23/09/2018. URL <https://github.com/tshort/openmodelica-javascript>.
- [3] A. Pérez-Castro, J. Sánchez, and J. L. Guzman. Model correction mechanism for nonlinear time variant systems as support to predictive control strategies. *IFAC Proceedings Volumes*, 47(3):5975–5980, 2014.

Capítulo 5

Publicaciones

5.1. Introducción

En este capítulo se puede encontrar un pequeño resumen de todas las publicaciones realizadas durante el desarrollo de la tesis doctoral. Además, al final de este capítulo se incluyen los *preprints* de las publicaciones más relevantes.

Algunas de las publicaciones que se pueden encontrar en este capítulo son trabajos previos a los artículos principales de esta tesis, otros se corresponden al inicio de los trabajos futuros, y por último, alguno de ellos abarcan la introducción del doctorando al mundo de la investigación y la publicación científica o a la exploración de técnicas que serán usadas para futuros trabajos. Como ya se ha comentado, se incluye un pequeño resumen de todos los artículos, la relación que el mismo tiene con los artículos principales de esta tesis así como la referencia a los mismos para poder consultarlos.

5.2. Publicaciones principales

Se enumeran a continuación cada una de las publicaciones con un pequeño resumen y los datos para poder acceder a cada uno de los artículos.

Development of an open experimentation tool based on JavaScript for the control of a four-tank plant

Actualmente, uno de los principales problemas a los que nos podemos enfrentar en el ámbito de la educación en ingeniería es la disponibilidad de instalaciones industriales reales que permitan tener un entorno de experimental real. La importancia de este tipo de sistemas de experimentación radica en permitir a los estudiantes

el aplicar los conceptos teóricos aprendidos en la resolución de problemas reales. Motivado por lo anterior, en el artículo citado a continuación se presenta una herramienta remota para educación. En concreto, se muestra un sistema de cuatro tanques, el diseño de su bucle interno de control y la implementación de una aplicación *EasyJava* que permite la ejecución de código *JavaScript* tanto en simulación como en el sistema real. Como resultado, se obtiene una herramienta que permite realizar experimentos sobre el sistema en tiempo real y de forma remota, también se permite hacer experimentos en simulación y usar dicho código para los experimentos reales. En la herramienta desarrollada, los límites de uso de la instalación van a estar impuestos por el sistema real y no por la propia herramienta.

- A. Perez-Castro, J. Sanchez-Moreno, M. Castilla. Development of an open experimentation tool based on JavaScript for the control of a four-tank plant. *Comput Appl Eng Educ.* 2018;26:228-238. <https://doi.org/10.1002/cae.21879>
- **Índice de impacto (JCR):** 1.153 (2017)
- **Posición:** 50/86 - Q3 (Categoría Engineering, Multidisciplinary) - 29/41 - Q3 (Categoría Education, Scientific Disciplines) - 84/105 - Q4 (Categoría Computer Science, Interdisciplinary Applications)

CFertigUAL: A fertigation management app for greenhouse vegetable crops

El proceso de fertirriego es el método más utilizado para suministrar agua y nutrientes a cultivos con o sin suelo en invernaderos en Almería. El principal objetivo del trabajo contenido en este artículo es el desarrollo de una aplicación móvil, en concreto una aplicación nativa *Android* que cumpla la función de calcular la cantidad de fertilizantes y riego para ser suministrados a los cultivos más importantes. El uso de esta aplicación móvil permitirá a ingenieros agrícolas, agricultores y estudiantes un mejor uso de los recursos existentes y de esta forma asegurar el suministro apropiado de agua y nutrientes. El desarrollo de esta plataforma se ha dividido en dos partes: 1) Determinar la cantidad de agua perdida por transpiración por medio de sensores virtuales que monitorizan las condiciones externas del invernadero, 2) Determinar las fertilizantes que se han de aplicar por litro de agua en los tanques de fertirrigación (kg/l). Una vez se ha configurado la aplicación, la misma funciona de forma automática usando algoritmos de optimización para proporcionar la solución final, en la que el equilibrio estimado satisface las necesidades de nutrientes del cultivo. Además, es posible operar la aplicación manualmente siguiendo las instrucciones para lograr un mejor control de la gestión de la fertirrigación y, en muchos casos, aprender sobre los cálculos de fertilizantes en función del perfil del sistema. Esta plataforma proporciona las cantidades de suministro de los distintos fertilizantes, así como la conductividad eléctrica y el pH, la posible presencia de precipitación en la solución final, y un drenaje o análisis del suelo para ayudar a optimizar la estrategia de nutrientes. Como resultado, la aplicación desarrollada funcionará tanto como un sistema de aprendizaje como una herramienta de toma de decisiones para los sistemas de fertirrigación, facilitando el acceso a la metodología utilizada para

determinar los parámetros óptimos de fertirrigación dentro de cualquier sistema de cultivo.

- A. Pérez-Castro, J.A. Sánchez-Molina, M. Castilla, J. Sánchez-Moreno, J.C. Moreno-Úbeda, J.J. Magán, cFertigUAL: A fertigation management app for greenhouse vegetable crops, *Agricultural Water Management*, Volume 183, 2017, Pages 186-193, <https://doi.org/10.1016/j.agwat.2016.09.013>.
- **Índice de impacto (JCR):** 3.182 (2017)
- **Posición:** 10/87 - Q1 (Categoría Agronomy) - 12/90 - Q1 (Categoría Water Resources)

PhotoBioLib: A Modelica library for modeling and simulation of large-scale photobioreactors

El trabajo presentado en este artículo describe la librería *PhotoBioLib*, una librería desarrollada en el lenguaje *Modelica* para el modelado y simulación de fotobiorreactores tubulares industriales sin la necesidad de un conocimiento previo de programación o de las dinámicas involucradas. Además, se consiguen modelos muy similares a los sistemas reales, con un gran potencial de configuración, con la posibilidad de interconexión con otras librerías y con una gran velocidad de simulación. Esta librería ha sido validada con datos reales.

- A. Perez-Castro, J. Sanchez-Moreno, M. Castilla, PhotoBioLib: A Modelica library for modeling and simulation of large-scale photobioreactors, *Computers & Chemical Engineering*, Volume 98, 2017, Pages 12-20, <https://doi.org/10.1016/j.compchemeng.2016.12.002>.
- **Índice de impacto (JCR):** 3.113 (2017)
- **Posición:** 24/105 - Q1 (Categoría Computer Science, Interdisciplinary Applications) - 35/135 - Q2 (Categoría Engineering, Chemical)

5.3. Resto de publicaciones

Se enumeran a continuación cada una de las publicaciones con un pequeño resumen y los datos para poder acceder a cada uno de los artículos.

Modelado y control de un vehículo eléctrico mediante una estrategia de control predictivo basado en modelo

En este trabajo se muestra el diseño de un controlador de la dinámica longitudinal de un vehículo eléctrico, teniendo como objetivo final el optimizar la eficiencia energética del mismo. Debido a las características del sistema a controlar y al

objetivo perseguido, se hace uso de un control predictivo basado en modelo, más concretamente se hace uso de la técnica PNMPC (*Practical Non Linear Model Predictive Control*). La variación de los parámetros, función de coste y restricciones permite al controlador gestionar el comportamiento del vehículo. Como paso previo al diseño del sistema de control se desarrolló un modelo del vehículo basado en primeros principios. Dicho modelo es usado tanto para la simulación de la dinámica del sistema real como para las predicciones que usa el controlador.

Este trabajo fue el primero que se desarrolló en la tesis y su cometido fue el de adentrarse en el modelado basado en primeros principios haciendo uso de ecuaciones diferenciales y la introducción al control predictivo basado en modelo. El controlador utilizado en este artículo es la base de las estrategias de control desarrolladas más adelante y que tienen como objetivo final ser aplicadas a los fotobiorreactores.

- A. Pérez-Castro, J.E. Normey-Rico, M. Berenguel, Modelado y control de un vehículo eléctrico mediante una estrategia de control predictivo basado en modelo, XXXIV Jornadas de Automática, Terrasa 2013 330-337

Estrategia para la corrección de modelos no lineales variantes en el tiempo mediante estimación de parámetros: Aplicación al control predictivo y a la diagnosis de sistemas

Este trabajo recibió el Primer Premio en Ingeniería de Control en las XXXIV Jornadas de Automática de Terrasa (2013).

En lo que respecta a su contenido, como su título indica, se presenta una estrategia para la corrección de modelos de predicción no lineales variantes en el tiempo. Dicha corrección se lleva a cabo mediante el recálculo de los parámetros internos que modifican la dinámica del sistema al que representan, así como con la actualización de sus estados. Esta corrección de modelos tiene una doble funcionalidad; por un lado, permite un mejor desempeño en controladores predictivos no lineales que actúen sobre sistemas variantes en el tiempo; por otro lado, sirve como mecanismo de diagnosis de sistemas, ya que proporciona información relevante sobre el estado en el que se encuentran los mismos. Para el desarrollo de esta estrategia se parte de un controlador predictivo no lineal y es la optimización de dicho control la encargada de recalcular los parámetros del modelo.

Este trabajo se centra en el desarrollo de mecanismos para la corrección de modelos con la vista puesta en su aplicación a los fotobiorreactores. Dicho corrector de modelos se cree necesario ya que al ser organismos vivos y vivir en un entorno dinámico con condiciones ambientales también variables sus dinámicas son cambiantes. Además, esta técnica puede servir también como el inicio de un mecanismo de diagnosis de errores y de control tolerante a fallos, así como de una técnica de control predictivo no lineal basado en eventos.

- A. Pérez-Castro, J. Sánchez, J.L. Guzmán, Estrategia para la corrección de modelos no lineales variantes en el tiempo mediante estimación de parámetros: Aplicación al control predictivo y a la diagnosis de sistemas, XXXIV Jornadas de Automática, Terrasa 2013, 338-345

Model correction mechanism for nonlinear time variant systems as support to predictive control strategies

Este trabajo es una mejora del anterior y fue presentado en el *19th World Congress of the International Federation of Automatic Control*. En el mismo se presenta una estrategia para estimar y corregir las variaciones en la dinámica de sistemas no lineales variantes en el tiempo. Esta corrección es llevada a cabo a través de la estimación de los parámetros internos del proceso y determinando las diferencias con un modelo no lineal del sistema y sus salidas reales. La estrategia propuesta tiene una doble funcionalidad; por un lado, permite un mejor desempeño en el uso de modelos no lineales para el control, como puede ser para su uso en controladores predictivos no lineales; por otro lado, puede ser adoptado como mecanismo de diagnosis ya que proporciona información relevante sobre el estado actual del sistema. De esta forma, con el objetivo de usar esta técnica con sistemas no lineales variantes en el tiempo, se ha recurrido a una estrategia de control predictivo basada en modelo.

A diferencia de en la primera aproximación enviada a las Jornadas de Automática, en este artículo se presentan resultados experimentales sobre un sistema real de cuatro tanques. Se demuestra cómo la técnica es capaz de detectar problemas de modelado adicionales así como la variación dinámica del orificio de salida de los tanques.

- A. Pérez-Castro, J. Sánchez, J.L. Guzmán, Model correction mechanism for nonlinear time variant systems as support to predictive control strategies, IFAC Proceedings Volumes, Volume 47, Issue 3, 2014, Pages 5975-5980, <https://doi.org/10.3182/20140824-6-ZA-1003.00766>.

PhotoBioLib: una librería de Modelica para el modelado y simulación de fotobiorreactores

Este artículo presenta la librería *PhotoBioLib*. Dicha librería, implementada en *Modelica*, permite el modelado y simulación de fotobiorreactores construyéndolos como si de sistemas reales se trataran, sin necesidad de conocimientos de programación ni de las dinámicas involucradas. Así, se obtienen como resultado modelos con gran parecido con la realidad, con gran potencial de configuración, con la posibilidad de interconexión con otras librerías y con grandes velocidades de simulación.

Se trata de una primera aproximación a uno de los tres artículos que componen el núcleo de esta tesis.

- A. Pérez-Castro, J. Sánchez, J.L. Guzmán, PhotoBioLib: una librería de Modelica para el modelado y simulación de fotobiorreactores A. Pérez-Castro, J. Sánchez, F.G. Ación, I. Fernández, J.L. Guzmán XXXV Jornadas de Automática, Valencia 2014, 3-10

Desarrollo de un entorno abierto de experimentación basado en *JavaScript* para el control de una maqueta de cuatro tanques

Este artículo presenta la puesta en marcha de un sistema de cuatro tanques, el diseño del control interno de la misma y la puesta en marcha de una herramienta *EasyJava* para ejecución de código *JavaScript*, tanto en simulación como en experimentación real. Como resultado se obtiene una herramienta que permite la experimentación sobre un sistema real, además de su equivalente en simulación, y donde los límites de uso no los impone la herramienta sino la propia instalación física.

Igual que en el caso anterior, se trata de una primera aproximación a uno de los tres artículos que componen el núcleo de esta tesis.

- A. Pérez-Castro, J. Sánchez, D. Chaos, Desarrollo de un entorno abierto de experimentación basado en JavaScript para el control de una maqueta de cuatro tanques, XXXVI Jornadas de Automática, Bilbao 2015, 616-623

CRIG: Una aplicación móvil para cálculos de fertirriego

El objetivo de este trabajo es el desarrollo de una aplicación móvil que ayude a realizar el cálculo del aporte de fertilizantes y riego para todos los cultivos en diferentes sistemas de cultivo. Este sistema permitirá a técnicos, agricultores y estudiantes aprovechar mejor los recursos existentes para una aplicación adecuada de nutrientes y agua. La aplicación desarrollada servirá como sistema de aprendizaje y de ayuda a la toma de decisiones para un sistema tan crítico como es el fertirriego. Por lo tanto, el facilitar la accesibilidad al procedimiento utilizado para determinar los parámetros óptimos de fertirriego en cualquier explotación agrícola es un problema de vital importancia que debe ser estudiado. En este trabajo, se propone una aplicación para dispositivos móviles que estima el valor idóneo de los parámetros de fertirriego en función de las características de la explotación agrícola y el tipo de cultivo. Asimismo, permite llevar un control de varias explotaciones agrícolas de forma simultánea mediante el uso de bases de datos persistentes.

Como en los dos casos anteriores, se trata de una primera aproximación a uno de los tres artículos que componen el núcleo de esta tesis.

- A. Perez-Castro, J.A. Sanchez, M. Castilla, J. Sanchez-Moreno, CRIG: Una aplicación móvil para cálculos de fertirriego, II Simposio Nacional de Ingeniería Hortícola, Almería 2016, 241-244

5.4. Preprints de las publicaciones principales

En esta sección se añaden los tres *preprints* de los principales artículos de la presente tesis.

Development of an open experimentation tool based on JavaScript for the control of a four tank plant

A. PEREZ-CASTRO¹, J. SANCHEZ-MORENO¹, M. CASTILLA²

¹ Dept. of Computer Sciences and Automatic Control, Universidad Nacional de Educación a Distancia (UNED), Madrid, Spain

² System Engineering and Automation Department, School of Engineering, Universidad de Sevilla 41092 Seville, Spain

Abstract

At present, one of the main problems which can be found within chemical engineering education is the availability of real industrial plants able to provide an experimentation framework. The importance of this kind of plants lies in allowing students to apply the learned theoretical concepts to solve real problems. In this paper, a remote tool for educational purposes is presented. More specifically, the implementation of a four tank plant, its internal control design and the implementation of an EasyJava tool for executing JavaScript code in both simulation and real experimental tests have been widely described. As a result, a tool that allows to perform experimental tests over the plant in both real and simulation modes has been obtained. In this tool, the limits of use are imposed by the plant and not by the own tool.

Keywords: Four tank, EasyJava, Virtual laboratory, Remote laboratory, JavaScript

1. Introduction

Most part of processes within chemical engineering should satisfy strict conditions of temperature, humidity, etc. Therefore, different control strategies are implemented in order to guarantee that

1
2 these conditions are satisfied. For this reason, control education is a cornerstone for the students of
3
4 chemical engineering degree.
5
6
7

8
9 Although a theoretical development is necessary from both researchers and students points of view,
10 it is also true that real experimentation provides information that sustains the theoretical knowledge
11 and supplies the required training to the student in order to apply their knowledge in any
12 engineering field [8, 18].
13
14
15
16
17
18
19

20
21 Nevertheless, real experimentation frequently deals with two big problems: the high cost that
22 represents to have real plants available for experimentation [7, 17] and, the distance that the
23 students can perceive from the theoretical knowledge to its practical use. The previously mentioned
24 issues are being extensively analysed, and thus, it is possible to find in literature different papers
25 related to this topic. One of the widespread solutions used to solve these problems is the
26 development of virtual and remote laboratories. For instance, in [2] a review about the main issues
27 involved with virtual and remote laboratories which can be accessed via Internet is performed. In
28 [6], a discussion about the experience obtained from the development of virtual and remote
29 laboratories along the last years, including some advices for people who want to develop new
30 virtual and remote laboratories, is performed. Finally, in [19], an analysis of the main variables
31 which influence the acceptance of this kind of laboratories by students and their relations is done.
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49

50 Therefore, the main objective of this work is to solve, in the better possible way, the problems
51 which have been mentioned previously. Concretely, the cost problem is addressed by decreasing the
52 relative cost of the plant, through, on the one hand, doing the plant accessible to more public
53 through the Internet and, on the other hand, providing more flexibility for the experimentation. In
54 regard to the perception of the theoretical knowledge to its practical use, it is solved by linking up
55
56
57
58
59
60

1
2 the use of a virtual laboratory with its real representation, allowing that the code implemented for
3
4 the virtual laboratory can be directly used in the real one.
5
6
7

8
9
10 As a result of the work presented in this paper, a scheme similar to the one displayed Figure 1 has
11
12 been obtained. More specifically, the student uses a computer in the local network to experiment
13
14 with both simulation and real systems. Hence, the student can implement any code for that system;
15
16 such as a basic controller, an identification process, a multivariable controller, an optimal controller,
17
18 etc. Afterwards, the student can verify its performance in simulation and, without any change, test it
19
20 in the real system obtaining both real time data and video signal.
21
22
23
24
25

26 The rest of the paper is organized as follows: Section 2 provides a description of the real plant
27
28 implementation; Section 3 explains some details about the developed SCADA (Supervisory Control
29
30 And Data Acquisition) system and the internal control loop of the plant; in Section 4 the EasyJava
31
32 tool designed for the experiments is presented and the integration of JavaScript within this tool is
33
34 widely described; and, finally, Section 5 shows the main conclusions and future works.
35
36
37
38
39
40

41 2. Implemented system: The four tank plant

42 The real plant which has been chosen to develop an educational/investigation tool has been the four
43
44 tank system. This is one of most relevant systems used in the multivariable control teaching
45
46 framework. The basic process scheme was developed by Eric Johansson from the Lund Institute of
47
48 Technology (Sweden) and published among others in [11, 12, 13].
49
50
51
52
53
54

55 This system, which basic scheme is shown in Figure 2, is composed of four tanks, two pumps and
56
57 two three-way valves. More specifically, its operation principles are: the pumps extract liquid from
58
59
60

1 the lower drain tank, and this flow is distributed between the four tanks by the three-way valves. In
2 addition, these tanks have a constant drainage area at their bases and thus the lower tanks receive
3 flow from both the upper tanks and the three-way valves. It is necessary mentioning that the
4 pipelines over the upper tanks are crossed and this causes that the two pumps have influence over
5 the two lower tanks. Therefore, the position of the three-way valves will modify the system
6 dynamics. For a complete description of the system and a discussion of its dynamics and
7 multivariable control applications [13] can be consulted.
8
9
10
11
12
13
14
15
16
17
18
19

20 The proposed scheme, see Figure 2.b, is quite similar to the Johansson's one. The main differences
21 are the addition of four on-off valves, for disturbances generation, and, four flow meters to measure
22 the flow in each branch of the plant. All sensors and actuators are monitored and controlled by a
23 Programmable Logic Controller (PLC), which is managed by a SCADA tool which has been
24 developed for the work presented in this paper. The real plant is located at the Department of
25 Computer Sciences and Automatic Control (UNED), see Figure 3. Table 1 shows a list of the
26 system components.
27
28
29
30
31
32
33
34
35
36
37
38

39 In Figure 1 the user's view of the implemented tool is shown. Nevertheless, Figure 4 shows how
40 internally works the proposed tool. Three elements can be highlighted: The real system, which has
41 been explained on this section; the SCADA environment, which will be explained on Section 3;
42 and, the EasyJava tool, which will be described on Section 4. The last one is the only element with
43 which the student directly interacts. Using this tool the student can create JavaScript codes both for
44 simulation and real operation modes. Furthermore, in a transparent way for the student, the above
45 tool is able to communicate with the SCADA tool through the Internet by using the JilServer (see
46 Section 4). Moreover, the SCADA tool is the responsible to handle the data flow to/from the real
47 plant and, also, to control the necessary inner control loops.
48
49
50
51
52
53
54
55
56
57
58
59
60

3. Development of the SCADA

It is very common in industrial plants the necessity of implement a cascade control scheme [15], such as that one which is shown in Figure 5. Hence, the actuators, in this case pumps, follow in the right way the signals requested by the upper loop controller. Without this internal loop it would not be possible a good performance of the upper loop. For that, it is necessary to implement an internal control loop transparent for the student. In this way, the whole inner loop can be considered as an ideal pump that responds to the reference requested by the upper loop, in this case the flow reference, and thus the student has only to implement the tank height controller.

As it was mentioned previously, a SCADA tool has been developed in LabVIEW [9] for data acquisition and the internal control loop, that is, the flow control of the pumps. Besides, it also allows to configure the plant behaviour, through the configuration of the three-way valves, and to generate several disturbances, by means of the on-off valves, which are located at the bottom of each tank. Moreover, the SCADA tool includes a block for JavaScript code execution. This block will be in charge to control the upper loop, i.e., the tanks height.

3.1. Analysis of the pump behaviour

In order to design an appropriate internal loop control strategy for the pumps flow, it is necessary to analyse the pumps behaviour. More specifically, the input to any of the pumps is a voltage between 0 and 10 volts and the output is directly related to this voltage, the tank height, and the position of three-way valves.

Therefore, to find out the pump dynamics it has been performed several experiments that increase

1
2 the voltage in 0.1 volts in successive steps, as it can be observed in Figure 6. As can be seen, not all
3 input variations cause output changes; solely some specific points cause alterations and, besides,
4 these points are not equidistant. Also, it can be observed that the pump provides a relatively stable
5 flow, through a pressure difference, for each one of these points. In Figure 6 only the flows over 120
6 cm^3 are shown since for lower levels the flow is equal to 0.
7
8
9
10
11

12
13
14
15
16 By studying the points to which the pump responds, it has been concluded that if 0.4 volts steps are
17 considered is possible to cover all the dynamic range with a minimal set of points. Furthermore, the
18 start point is located, in almost all cases, in the 0.8 volts and, the bigger response is in the 9.6 volts,
19 being the same response for 10 volts.
20
21
22
23
24

25
26
27
28 Moreover, once that the above points have been marked, it can be established that by defining a
29 relation between the flow and the necessary voltage to achieve it, these points can be adjusted
30 nicely with a second order polynomial, as can be observed in Figure 7. Likewise, if only three
31 points are used (the first, the intermediate and the last one) and the second order polynomial is
32 recalculated, the obtained polynomial is quite similar to the first one.
33
34
35
36
37
38
39
40
41
42

43 Finally, it is important to remark, that, as it can be observed in Figure 6, there is a relatively
44 important delay between the voltage and the flow provided by the pump if it is compared with the
45 pump dynamics.
46
47
48
49
50
51
52

53 3.2. Obtaining the pump dynamics

54 As it has already been described in the previous Section, the pump dynamics depends on the liquid
55 level in tank 0, which varies by evaporation and, on the position of the three-way valves, i.e., on the
56
57
58
59
60

1
2 plant configuration.

3
4
5
6
7 As can be observed in Figure 7, by interpolating using only three points the dynamic behaviour can
8
9 be precisely captured. Therefore, the procedure to obtain the pump dynamics can be summarized as
10
11 follows:

- 12
13
14 1. Place the three-way valves as a function of the desired plant configuration and open all the
15
16 on-off valves in order to avoid any possible error caused by lack of water in tank 0.
- 17
18
19 2. Provide the three operation points to the pump; 0.8, 5.2 and 9.6 volts; leaving enough time
20
21 for the flow stabilization.
- 22
23
24 3. The minimum and maximum flow are registered, these points are necessary, for example,
25
26 for an anti-windup control strategy.
- 27
28
29 4. The second order polynomial that better fit to the above three points is calculated. Also, a
30
31 first order polynomial is calculated using the minimum flow and flow 0. This polynomial is
32
33 necessary because the quadratic function can grow up and provide incorrect values, as can
34
35 be seen in Figure 7.
- 36
37
38 5. Switch to a normal operation mode of the plant.

39
40
41
42
43
44 The tuning process will be realized before to whatever experiment unless the user indicates the
45
46 opposite. The above can take place when the same configuration will be used and there are not
47
48 water lost in tank 0, for example, due to evaporation.

55 3.3. Internal loop control strategy

56
57 It can be established a relation between the PI control and the designed control strategy. This
58
59 strategy, whose scheme is shown in Figure 8, can be split into two different parts which together
60

1
2 are able to obtain the final set-point.
3
4

5
6
7 The first block, named the 'Polynomial' block, obtains from the calculated polynomials in the
8 previous tuning, the most nearby voltage possible that provides the desired flow. It is necessary to
9 remember that only 0.4 volts increments are considered, and the intermediate values are discarded
10 because they don't involve a modification of the pump operation. This block can be equivalent to
11 the proportional part of a PI controller.
12
13
14
15
16
17
18
19
20

21 The second block, denominated 'Steps' block, makes ascend or descend the control signal in 0.4
22 volts steps as a function of the tracking error. This block, which is maybe comparable to the integral
23 action of a PI controller, integrates the time, sampling periods, where the follow error is over a
24 certain threshold. If this threshold is exceeded for a time, it will be added or subtracted to the
25 voltage signal one 0.4 volts step. To tune this controller the next parameters will be used:
26
27
28
29
30
31
32

- 33 • threshold: It is the level above which the tracking error is considered. This threshold must be
34 a value bigger than the flow difference between two pump operation points.
35
- 36 • tError: It is the time that is employed to estimate the mean of the tracking error. That mean
37 is the value taken into account to decide if the error will be or not considered.
38
- 39 • tRise: It is the time integrated to jump to the next or the previous voltage step. When a
40 number of sampling periods equivalent to this time will be accumulated the voltage level
41 will change. When the set-point signal cuts the actual flow, the counter, that is, the
42 integrator, will be reset. The rising time must be bigger than the establishment time
43 following a change in voltage level, since, otherwise, the control signal will oscillate.
44
45
46
47
48
49
50
51
52
53
54
55
56
57

58 The aforementioned strategy leads up, with an easy tuning, to a controller with a speed near to the
59
60

1 actuator physical limits and with a stationary error minor than the difference between two voltage
2 levels. The drawbacks that can be observed are a delay over the reference signal and a small steady
3 state error. Nevertheless, the delay is a non-solvable problem, if the future behaviour is not known,
4 and, the steady state error, in almost all cases, will be less than the flow meter sensors sensitivity.
5
6
7
8
9
10
11 Figure 9 shows the response achieved with the proposed controller over the real system.

12
13
14
15
16 Therefore, it has been shown that the designed control strategy has a good performance and an easy
17 tuning procedure. Furthermore, there is another reason to use this strategy and not another one like,
18 for example, a PI controller. This is that the pumps do not have a continuous dynamic behaviour,
19 i.e., they only provide, through an internal control, a particular flow for each range of voltages; and,
20 the pump delay time is dominant over the pump internal dynamics. Hence, the pumps work like if
21 they had only 23 possible configuration points, with a nearly instantaneous dynamics in comparison
22 with the delay time. Therefore, the main function of the proposed controller will be to find the
23 appropriate voltage point for the demanded flow with a minimal tuning effort.
24
25
26
27
28
29
30
31
32
33
34
35
36
37

38 3.4. Testing the external loop performance

39 As it has already been already mentioned, the SCADA tool includes a block for the execution of
40 JavaScript code. It will be responsible for calculating the control signals of the outer loop, i.e., to
41 control the height of the tanks.
42
43
44
45
46
47
48
49

50 In order to test the correct performance of the whole system a control for the two lower tanks has
51 been developed. More specifically, a PI control with anti-windup, tuned by the open loop Ziegler-
52 Nichols method combined with a Smith Predictor, and, discretised by the Tustin method [1, 14], has
53 been implemented. The obtained results can be seen in Figure 10. The control signals, that is the
54 flow signals that are shown in Figure 9, correspond to tank 2. Moreover, in Listings 1 and 2 the
55
56
57
58
59
60

1
2 used JavaScript code is shown.
3
4
5
6
7

8 **4. Easy Java Simulations experimentation tool**

9 Easy Java Simulations, also known as EJS, is a free authoring tool written in Java that helps non-
10 programmers to create interactive simulations in Java, mainly for teaching or learning purposes [4,
11
12
13
14 5]. EJS is part of the Open Source Physics project [16].
15
16

17
18
19 In order to perform experiments and as a teaching tool, an EasyJava based tool has been developed.
20
21 This tool allows using a four tank plant for both simulation and real experimentation tests. The tool
22
23
24 interface is shown in Figure 11.
25
26

27
28
29 The Graphical User Interface (GUI) of the developed tool is divided into two columns. On the left
30
31
32 column, the synoptic, that is, a schematic representation of the plant, is shown. This representation
33
34 shows the changes on real-time, providing the measured values, in addition to depicting the height
35
36 of the tanks with an animation. This will occur when the tool work in simulation mode or connected
37
38 to the real system; moreover, in the last one, a real-time video of the system will be shown, as can
39
40 be seen in Figure 12. On the bottom of this column, there are three buttons to start, to stop, and to
41
42 pause the experiments, as well as to save the data and to change to the real mode. Also, there are
43
44 some buttons that allow users to activate and deactivate the system disturbances in any moment.
45
46
47
48
49

50
51 At the top of the right column there are two graphs representing respectively the flow and height
52
53 values of the tanks. At the bottom of this column are three tabs: Manual, Control and Initialization.
54
55 More specifically, in the 'Manual' tab the current height set-points and the three-way valve positions
56
57 can be modified, also, the height and flow values are shown. In the other two tabs, 'Initialization'
58
59
60

1 and 'Control', the JavaScript code for the initialization and for the control loop can be loaded, saved
2 and edited; as can be seen in Figure 13. As a security precaution, the JavaScript code only can be
3
4 and edited; as can be seen in Figure 13. As a security precaution, the JavaScript code only can be
5
6 executed in the real system if it has been previously tested in simulation mode.
7

8
9
10
11 When the real-mode is activated, the EasyJava tool establishes a connection to the SCADA of the
12 real four tank system and sends to it the JavaScript code that the user wants to execute in the real
13 plant. Afterwards, this SCADA system begins to execute the JavaScript code and it sends
14 information about sensors and actuators' state to the EasyJava tool every sampling time. In addition,
15 the disturbance valve values and on-off valve, are also sent, every sampling period. Therefore, at
16 each sampling period, the received data, are represented and are written in a Matlab file. The
17 communication between the EasyJava tool and the SCADA of the real system is performed by the
18 JilServer tool [3, 10, 20]. More specifically, the sampling period of this system is equal to one
19 second for both acquiring and send data. Nevertheless, the data and video streaming updating rate
20 will depend of the state of the network and the distance of the user to the real plant, that is, the user
21 will receive the information at real time or with a small delay.
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38

39 The JilServer is a LabView application that allows the communication between a SCADA
40 developed in LabView and an EasyJava tool. More specifically, JilServer is able to publish the
41 controls and indicators existing in the LabView VI file and do it visible through a Transmission
42 Control Protocol (TCP) connection for any application that needs to establish a communication with
43 LabView.
44
45
46
47
48
49
50
51
52
53

54 For using the tool as a teaching environment, it has been uploaded to the Unilabs site [21]. Unilabs
55 is an educational network composed by several universities which share their laboratories resources.
56 In this site, the enrolled students can find all the necessary material to do experiments and use the
57
58
59
60

1
2 tool in both simulation and real modes. However, the access to the real mode, that is, to use the real
3
4 four tank system is restricted. Hence, to use it in real mode, the user must book the facility through
5
6 the reservation system integrated in Unilabs using his/her private credentials. Therefore, only one
7
8 user can be connected to the real facility at every moment. Moreover, the Unilabs platform can store
9
10 the files with the control code and the results generated during the experiments.
11
12
13
14
15

16 4.1. About the JavaScript use

17 The choice of JavaScript as language to control the plant is motivated by the following reasons: it is
18
19 a widely used and simple language, which implies that it is a very easy to learn and thus, there is a
20
21 lot of available resources. Moreover, it is an interpreted language whatever it provides security for
22
23 its execution in a remote environment. The last point is very important because the control code, in
24
25 the real mode, will be executed directly by the SCADA system. The execution in the SCADA
26
27 system prevents faults cause by the delay in the sending of the control signals.
28
29
30
31
32
33
34

35 A set of variables is provided to the user for the interaction with the tool for both simulation and
36
37 remote connection. Through these variables it can obtain sensors data as well as write the flow set-
38
39 points, the sampling time or the minimal and maximum flow ranges. In Table 2 these variables are
40
41 described.
42
43
44
45
46

47 Apart from the provided variables, the user can create new auxiliary variables. The JavaScript code
48
49 of the implemented controller can be seen in Listings 1 and 2. Moreover, in the initialization code it
50
51 is shown how to create an extra auxiliary variable.
52
53
54
55
56
57
58
59
60

5. Conclusions and future works

In this paper, it has been shown the development of a tool to allow both researchers and students to access to a real system and test their controllers, over a four-tank system, and the design of a polynomial adaptive control strategy to regulate the pump flow. The designed tool allow both real and simulation operation modes, and, the more important thing, the implemented code for the simulation modes can be directly used over the real system. Furthermore, it is possible to receive at real time the controller results data and a video signal of the real plant as well as to save the data for a later analysis.

In addition, due to the use of the JavaScript language is possible to develop almost any kind of controller and so user is not limited, for example, to do the tuning of a preselected controller parameters. This fact provides a wide range of possibilities in just one facility. So it is possible for example to tune the parameters for a given controller, to design and to implement a model predictive control strategy or to implement a multivariable controller to perform system identification experiments. The limits are set by the real system and not by the tool or the used language.

There are two main lines for future improvements. The first one is the possibility of using another programming languages, for example, Matlab, for the implementation of more complex controllers by researchers, being not necessary to translate already implemented programs Matlab into JavaScript code. The second one is to incorporate high-level JavaScript functions to ease the development of more complex control strategies just as, for example, optimization functions for their use in a model predictive controller.

Acknowledgements

The authors acknowledge the financial support of the Ministry of Economy and Competitiveness under DPI2014-55932-C2-2-R and DPI2011-27818-C02-02 research projects and the FPI scholarship framed in the latter. M. Castilla is a fellow of the Spanish “Juan de la Cierva-Formación” contract program.

References

1. K. J. Åström, T. Hägglund, *Advanced PID control*, ISA-The Instrumentation Systems and Automation Society; Research Triangle Park, NC 27709, 2006.
2. B. Balamuralithara, P. C. Woods, Virtual laboratories in engineering education: The simulation lab and remote lab, *Computer Applications in Engineering Education*, 17, 108–118 (2007).
3. J. Chacon, H. Vargas, G. F. Castro, J. S. Moreno, S. Dormido, Ejs, JilServer and Labview: How to build a remote lab in the blink of an eye, *IEEE Transactions on Learning Technologies*, 8 (4), 393-401 (2015).
4. EasyJava site. URL <http://www.um.es/fem/EjsWiki/>. Last Accessed on: 15th July, 2017.
5. F. Esquembre, Easy java simulations: A software tool to create scientific simulations in java, *Computer Physics Communications*, 156 (2), 199–204 (2004).
6. F. Esquembre, Facilitating the Creation of Virtual and Remote Laboratories for Science and Engineering Education, *IFAC Workshop on Internet Based Control Education IBCE15*, 48 (29), 49-58 (2015).
7. J. Fayolle, C. Gravier, N. Yankelovich, E. Kim, Remote lab in virtual world for remote control of industrial processes, *2011 IEEE International Conference on Multimedia and Expo (ICME)*, 1–4 (2011).

- 1
2 8. L. D. Feisel, A. J. Rosa, The Role of the Laboratory in Undergraduate Engineering
3 Education, *Journal of Engineering Education*, 94, 121–130 (2005).
- 4
5
6
7 9. T. Jeffrey, J.Kring, *LabVIEW for everyone: graphical programming made easy and fun*,
8 Prentice-Hall, 2007.
- 9
10
11 10. JilServer site. URL <https://github.com/UNEDLabs/jil-server/>. Last Accessed on: 15th July,
12 2017.
- 13
14
15
16 11. K. H. Johansson, J. L. R. Nunes, A multivariable laboratory process with an adjustable zero,
17 *Proceedings of the American Control Conference*, 4, 2045–2049 (1998).
- 18
19
20
21 12. K. H. Johansson, A. Horch, O. Wijk, A. Hansson, Teaching multivariable control using the
22 quadruple-tank process, *Proceedings of the 38th IEEE Conference on Decision and Control*,
23 1, 807– 812 (1999).
- 24
25
26
27 13. K. H. Johansson, The quadruple-tank process: a multivariable laboratory process with an
28 adjustable zero, *IEEE Transactions on Control Systems Technology*, 8 (3), 456–465 (2000).
- 29
30
31
32 14. K. Ogata, *Discrete-time control systems*, Vol. 2, Prentice Hall Englewood Cliffs, NJ, 1995.
- 33
34
35
36 15. B. A. Ogunnaike, W. H. Ray, *Process dynamics, modelling, and control*, Oxford University
37 Press, New York, 1994.
- 38
39
40
41 16. Open source physics site. URL <http://www.opensourcephysics.org/>. Last Accessed on: 15th
42 July, 2017.
- 43
44
45
46 17. J. Rossiter, Y. Shokouhi, Developing virtual laboratories for introductory control, *UKACC*
47 *International Conference on Control (CONTROL)*, 1025–1030 (2012).
- 48
49
50
51 18. M. Stefanovic, D. Tadic, S. Nestic, A. Djordjevic, An assessment of distance learning
52 laboratory objectives for control engineering education, *Computer Applications in*
53 *Engineering Education*, 23, 191–202 (2015).
- 54
55
56
57 19. R. Tirado, R. S. Herrera, M. A. Márquez, A. Mejías, M.J. Andújar, Comparing Remote
58
59
60

1
2 Laboratories from the Student Perspective, *IFAC Workshop on Internet Based Control*
3
4 *Education IBCE15*, 48 (29), 176-181 (2015).
5

6
7 20. L. De La Torre, M. Guinaldo, R. Heradio, S. Dormido, The ball and beam system: a case
8
9 study of virtual and remote lab enhancement with moodle, *IEEE Transactions on Industrial*
10
11 *Informatics*, 11 (4), 934–945 (2015).
12

13
14 21. Unilabs site. URL <http://unilabs.dia.uned.es/>. Last Accessed on: 15th July, 2017.
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

For Peer Review

Listing 1: JavaScript initialization code

```
1  /*Control signals*/
2  u[0] = 0;
3  u[1] = 0;
4  a1 = [0,0];
5  a2 = [0,0];
6  e1 = [0,0];
7  e2 = [0,0];
8  v = [0,0];
9
10 /*Discrete model parameters for the Smith Predictor*/
11 t1b0 = 0.001519;
12 t1b1 = 0.001519;
13 t1a0 = 1;
14 t1a1 = -0.9753;
15 t2b0 = 0.001244;
16 t2b1 = 0.001244;
17 t2a0 = 1;
18 t2a1 = -0.9852;
19
20 /*Past heigh and flow values*/
21 h1ant = 0;
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

```
22 h2ant = 0;
23 u1ant = 0;
24 u2ant = 0;
25
26 /*Outputs memory for the Smith Predictor*/
27 hm1 = [0,0,0,0,0,0,0,0,0,0,0,0];
28 hm2 = [0,0,0,0,0,0,0,0,0,0,0,0];
29
30 /*ZN control tuning*/
31 kp1 = 36;
32 Ti1 = 26;
33 Tr1 = 1*Ti1;
34 kp2 = 36;
35 Ti2 = 33;
36 Tr2 = 1*Ti2;
37
38 ki1 = kp1/Ti1;
39 kr1 = 1/Tr1;
40 ki2 = kp2/Ti2;
41 kr2 = 1/Tr2;
42
43 q01 = kp1+(dt/2)*ki1;
44 q11 = -kp1+(dt/2)*ki1;
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

```

45  qA1 = kr1*dt/2;
46  qO2 = kp2+(dt/2)*ki2;
47  qI2 = -kp2+(dt/2)*ki2;
48  qA2 = kr2*dt/2;

```

Listing 2: JavaScript control loop code

```

1  /*Estimation of the fitting error*/
2  e1[0] = refh1 - (h1+hm1[1]-hm1[9]);
3  e2[0] = refh2 - (h2+hm2[1]-hm2[11]);
4
5  /*Control signals increment*/
6  uAdd1 = qO1*e1[0]+qI1*e1[1]+qA1*(a1[0]+a1[1]);
7  uAdd2 = qO2*e2[0]+qI2*e2[1]+qA2*(a2[0]+a2[1]);
8
9  /*New control signals*/
10 v[0] = v[0]+uAdd1;
11 v[1] = v[1]+uAdd2;
12 u[0] = v[0];
13 u[1] = v[1];
14
15 /*Filtering and control signal saturation*/
16 if (u[0]>max1) u[0] = max1;

```

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
```

```
17  if (u[0]<min1) u[0] = min1;
18  if (u[1]>max2) u[1] = max2;
19  if (u[1]<min2) u[1] = min2;
20
21  /*Smith Predictor model output*/
22  hm1[0] = -t1a1*h1ant + t1b0*u[0] + t1b1*u1ant;
23  hm2[0] = -t2a1*h2ant + t2b0*u[1] + t2b1*u2ant;
24
25  for (i = 10; i <= 0; i--) {
26      hm1[i+1]=hm1[i];
27      hm2[i+1]=hm2[i];
28  }
29
30  /*Storage of past values*/
31  e1[1] = e1[0];
32  a1[1] = a1[0];
33  a1[0] = u[0]-v[0];
34  e2[1] = e2[0];
35  a2[1] = a2[0];
36  a2[0] = u[1]-v[1];
37  u1ant = u[0];
38  u2ant = u[1];
39  h1ant = h1;
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

```
40 h2ant = h2;
```

For Peer Review

Component	Model
Pumps	Grundfos UPE 25-40
Three-way valves	Samson 3226/2780-2
On-off valves	SMC
Tank height sensors	Endress & Hauser PCM 731
Flow sensors	RS Amidata 257-026
Adquisition system	Opto 22 SNAP Ultimate I/O

Table 1. Components of the four tank plant

83x30mm (150 x 150 DPI)

Variable	I/O	Description
$h1$	Output	Tank 1 height
$h2$	Output	Tank 2 height
$refh1$	Output	Tank 1 reference
$refh2$	Output	Tank 2 reference
$u[0]$	Input	Pump 1 flow (qL)
$u[1]$	Input	Pump 2 flow (qR)
dt	Output	Sample time
$min1$	Output	Pump 1 min flow
$min2$	Output	Pump 2 min flow
$max1$	Output	Pump 1 max flow
$max2$	Output	Pump 2 max flow

Table 2. Set of configurable variables by the user

66x52mm (150 x 150 DPI)

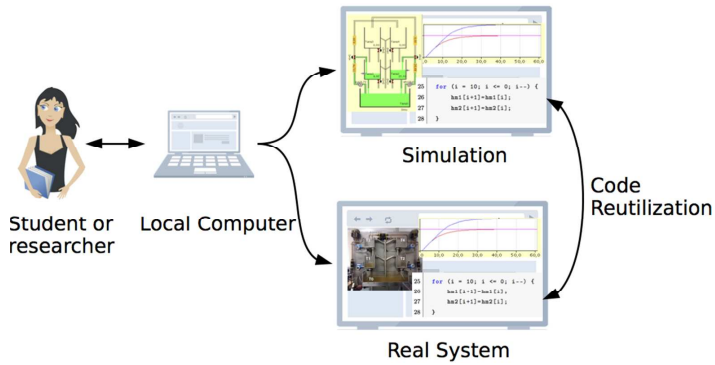


Figure 1. Student scheme view

187x96mm (150 x 150 DPI)

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

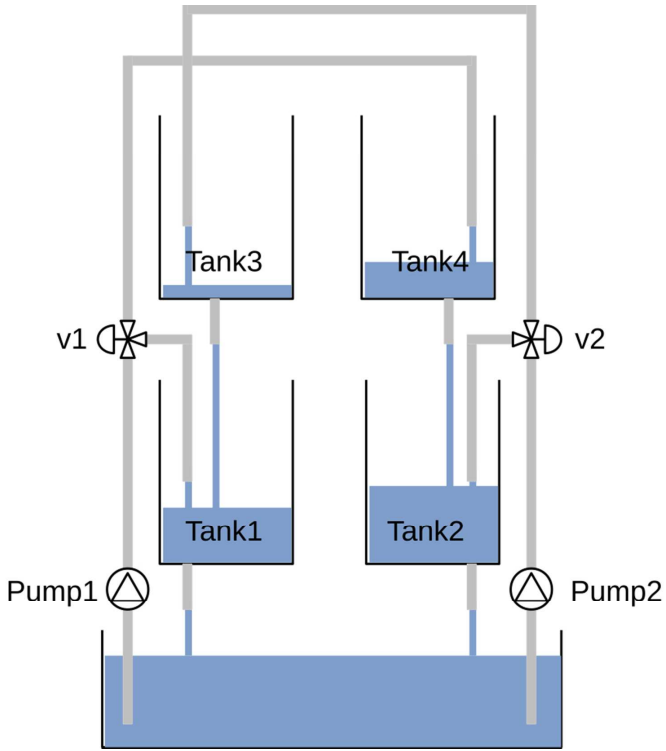


Figure 2a. Four tank scheme

153x162mm (150 x 150 DPI)

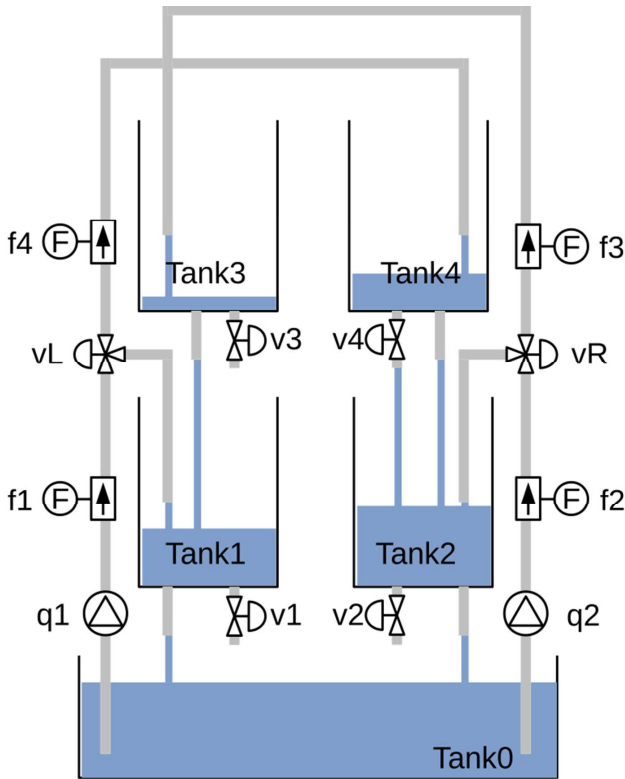


Figure 2b. Four tank scheme

146x163mm (150 x 150 DPI)

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

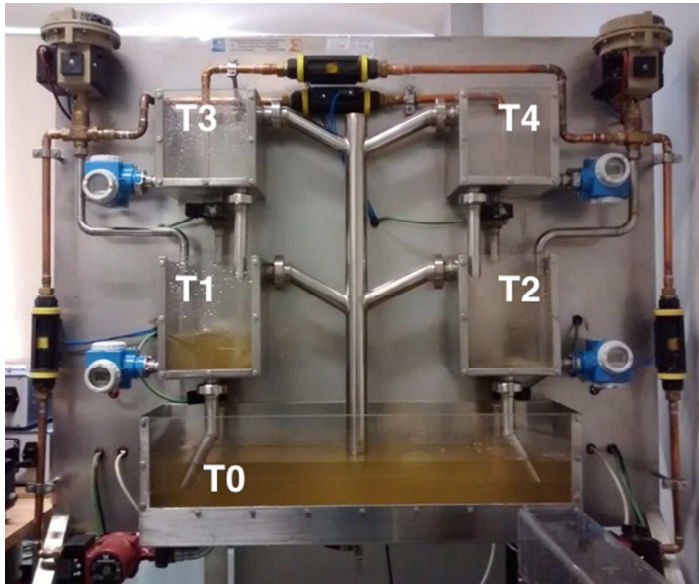


Figure 3. Four tank plant (Located at UNED)

203x169mm (72 x 72 DPI)

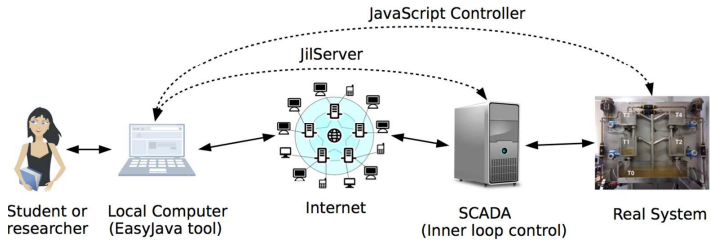


Figure 4. Internal implementation of the experimentation tool

244x81mm (150 x 150 DPI)

Peer Review

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

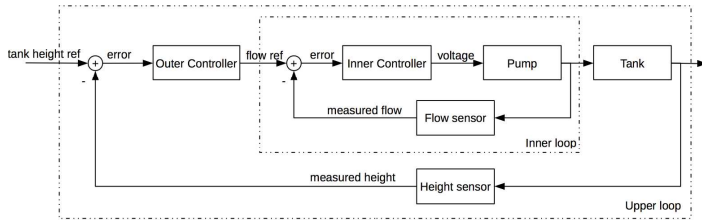


Figure 5. Scheme of the cascade controller implemented in the real four tank plant

387x118mm (150 x 150 DPI)

Or Peer Review

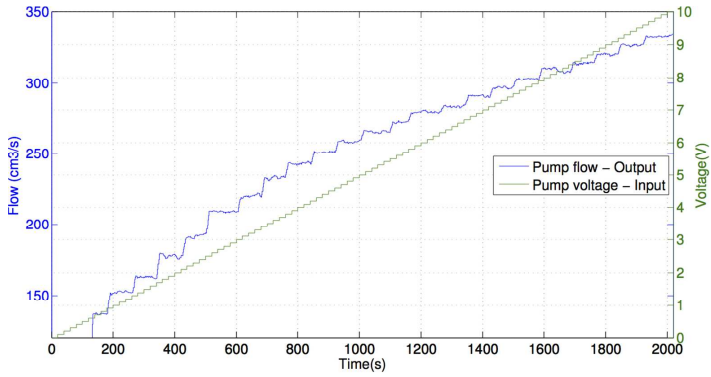


Figure 6. Pump flow as a function of the input voltage

246x129mm (150 x 150 DPI)

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

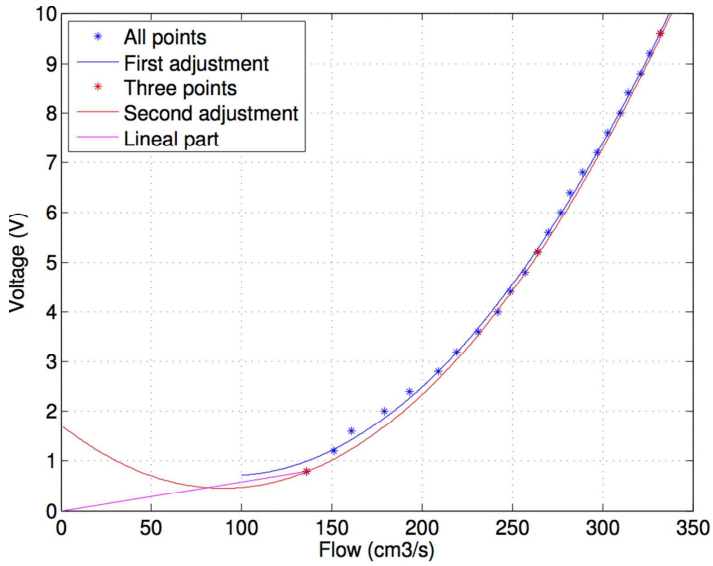


Figure 7. Adjustment of the pump dynamics

169x134mm (150 x 150 DPI)

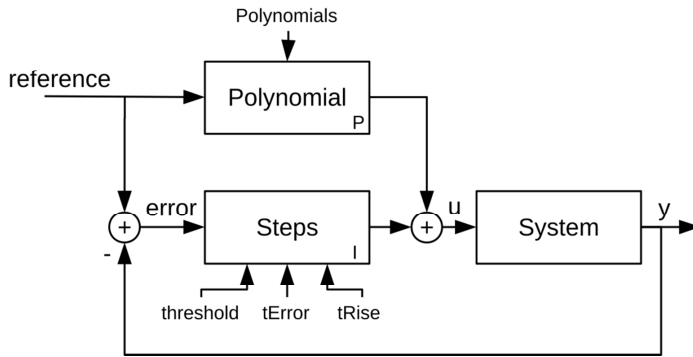


Figure 8. Control scheme of the pump

183x93mm (150 x 150 DPI)

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

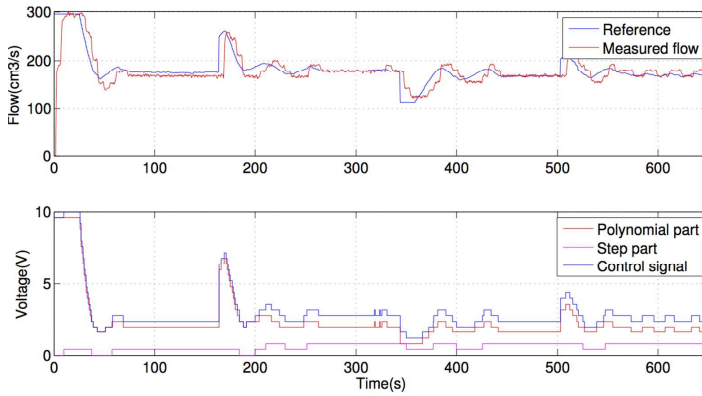


Figure 9. Tracking of the flow reference (upper) and its associate control signal (botton)

234x129mm (150 x 150 DPI)

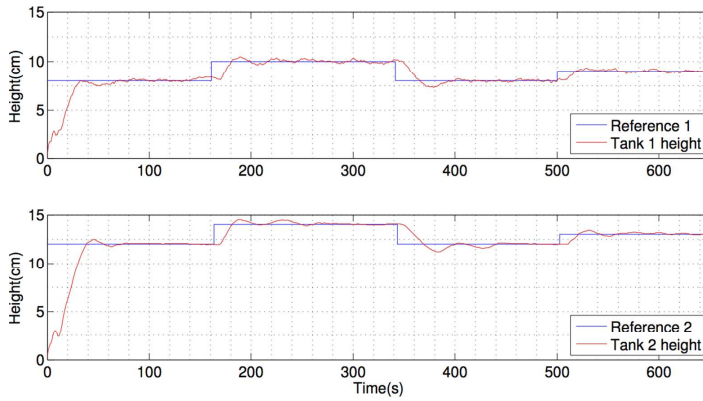


Figure 10. Tracking of the height set-points in the two lower tanks

231x129mm (150 x 150 DPI)

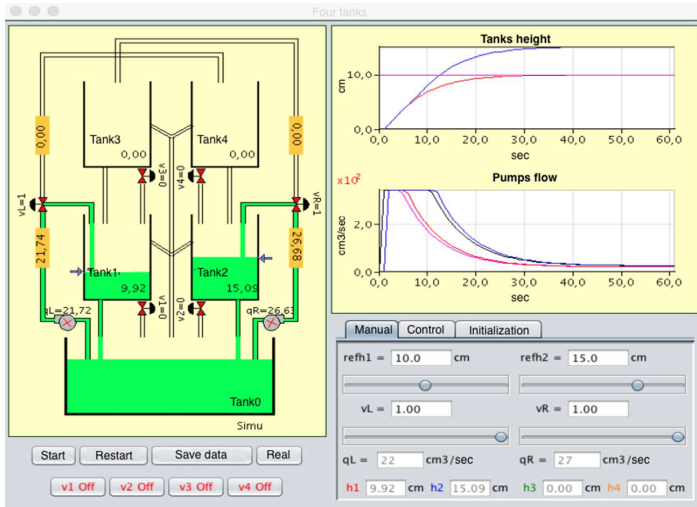


Figure 11. Interface of the four tank EasyJava tool

299x217mm (72 x 72 DPI)

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

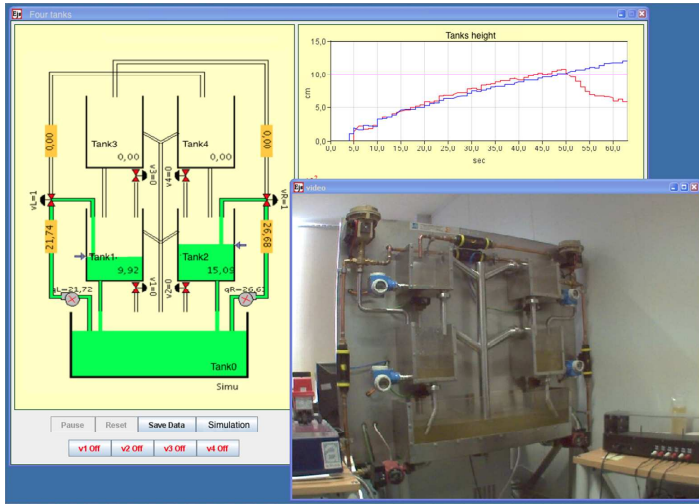


Figure 12. EasyJava tool connected to the real plant

388x277mm (72 x 72 DPI)

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

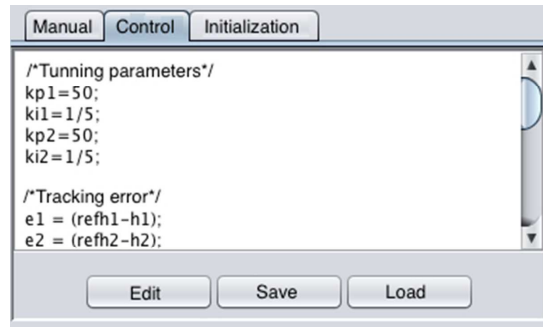


Figure 13. Example of JavaScript code typed in the EasyJava tool

136x81mm (72 x 72 DPI)

***Manuscript**[Click here to download Manuscript: Perez - AGWAT 2016 V6.pdf](#)[Click here to view linked References](#)

cFertigUAL: Fertigation Management App

A. Pérez-Castro¹, J.A. Sánchez-Molina², M. Castilla³, J. Sánchez-Moreno¹, J.C. Moreno-Úbeda² y J.J.

Magán⁴

¹ Dept. of Computer Sciences and Automatic Control, Universidad Nacional de Educación a Distancia (UNED), Madrid,

Spain E-mail: agustinperezcastro@bec.uned.es / jsanchez@dia.uned.es

² Department of Informatics, University of Almería, ceiA3, CIESOL, 04120 Almería (Spain), Spain. E-mail: {jorgesanchez,

jcmoreno}@ual.es

³ Department of System Engineering and Automation, School of Engineering, University of Seville, 41092 Seville, Spain.

E-mail: mcastilla4@us.es

⁴ Experimental Station of Cajamar Foundation, "Las Palmerillas", El Ejido (Almería, Spain) E-mail:

juanjosemagan@fundacioncajamar.com

1 Abstract

2 *Fertigation process is the most common way to supply water and nutrients in Almería's greenhouses*
3 *with soil-less or soil culture. Therefore, the main objective of this work is to develop, using Android*
4 *Studio software, a mobile application (app) that helps with the calculation of fertilizers and irrigation*
5 *to be supplied for the most important types of crops, different crop growing systems and for a bunch of*
6 *fertigation technologies. The use of this mobile application (app) will allow agricultural engineers,*
7 *farmers and students to perform a better use of the existing resources for an appropriate application*
8 *of nutrients and water. The development of this platform has been divided into two different parts: (1)*
9 *Obtaining the amount of water lost by transpiration from virtual sensors that use the external*
10 *greenhouse conditions to estimate this energy cost, and to establish the irrigation necessities, and (2)*
11 *Determining of the fertilizers that have to be applied per liter of water in fertigation tanks (kg/l). Once*
12 *that the system configuration data has been obtained, the app will operate automatically using some*
13 *optimization algorithms that provide a final solution in which the estimated equilibrium allows to meet*
14 *the needs of nutrients of the crops. In addition, it is also possible to operate the app by hand following*
15 *its instructions in order to get a better control over fertigation management and, in many cases, to*

16 *learn about the fertilization calculations as a function of system profile. At the end, this platform*
17 *provides a final weight of the different fertilizers to supply, including electrical conductivity and pH,*
18 *the possible presence of precipitation in the final solution, and a study of the drainage or soil analysis*
19 *trying to optimize the nutrient approach. Hence, the developed application will work as both a*
20 *learning system and a decision making tool for fertigation systems, making easier the accessibility to*
21 *the methodology used to determine the optimal fertigation parameters within any cropping system.*
22 *Keywords: Fertilizers, Irrigation, Optimization, Precipitation, Mobile application, Android*

23 **1. Introduction**

24 Greenhouse-based vegetable growth system is one of the main economic sectors in the province of
25 Almería (Spain), where there is the largest concentration of greenhouses in the world. Concretely,
26 more than 37,000 hectares of simple plastic Mediterranean greenhouses are used for vegetable crop
27 production (Castilla and Hernández, 2005). This can be translated into a production of more than 3
28 million tons, including tomato which is the most important type of crop. The export volume comprises
29 almost 70% of the total production, with a business of 2 billion Euros (Sanchez et al., 2012).
30 Nevertheless, this substantial economic development has increased the use of resources such as land
31 and water, indeed eighty per cent of the irrigation water used comes from underground sources,
32 leading to localized overexploitation of aquifers (Fernandez et al., 2007). This underground resources
33 exploitation is made worse due to the recent rapid expansion of the surface area occupied by
34 greenhouses in the Mediterranean Basin (Sanchez-Molina et. al, 2015). Consequently, it has also led to
35 considerate water as an important factor in the sustainability of the greenhouse-based vegetable
36 growth system. For that reason, several institutions have worked to optimize the amount of water and
37 fertilizers used in irrigation, reducing the environmental problems associated with fertigation process
38 in order to mitigate the severe structural water deficit and the leakage of considerable amounts of
39 fertilizers (Gallardo et al., 2009) which deplete the aquifers and reduce the water quality (Sánchez-
40 Martos et al., 1999; Carvajal et al., 2014).

41 Thus, as in other arid and semi-arid areas of the world, Almeria has promoted the drip fertigation,
42 water collection as a secondary source, the reuse of drainage water, and the development of new
43 technologies related to the efficiency of water and nutrients used, such as, advanced fertigation
44 controllers. The aim of these systems is to precisely supply water and fertilizers, since too much leads
45 to an excessive washing out of fertilizers (Ansorena, 1994). In turn, this might result in radicular
46 asphyxiation, substrate flooding or subterranean water contamination (Pardossi et al., 2004).
47 Conversely, a nutrients or hydrological deficit may be provoked if the irrigation process fails to
48 provide enough water. Hence, this can lead to a decrease in production and might even be dangerous
49 to crop growth (Domeño et al, 2009). Therefore, automatic control systems and management tools are
50 fundamental mechanisms to supply water and nutrients to any crop at the required amount and
51 frequency (Magan et al., 2008).

52 Consequently, the efficiency in the amount of water and nutrients supplied is a practical problem of
53 considerable interest, which has a significant economic importance. The requirements for fertigation
54 control systems must be determined as a function of the desired crop performance, the irrigation and
55 drainage water analyses, the soil/substrate nutrients content and the type of irrigation system (Gieling,
56 2001). Accordingly, an adequate management tool can help in this purpose, since an efficient
57 management of these systems is quite complicated, and thus, it requires a deep knowledge of the crop,
58 the distribution of water and nutrients and other chemical phenomena like precipitation, nutrients
59 absorption, etc. In the market, there are different examples of Decision Support System (DSS) which
60 can help to make easier the fertigation calculations like Ecofert (Bueno-Delgado et al. 2016), Smart-
61 fertilizer (Smart-fertilizer, 2012), SSCA (SSCA, 2014), NPK 2015 (NPK, 2015), Optifer (Pagán et al.,
62 2015) and some excel tools like Homo-Agricola (Homo-Agricola, 2011). More in detail, Ecofert offers
63 similar options and calculations than cFertigUAL, but the tool presented in this paper, cFertigUAL,
64 provides farmers, students and agricultural engineers a greater degree of configuration and analysis as:
65 the configuration of different fertigation equipments, from most advanced fertigation controllers with
66 until twelve fertigation tanks to simplest fertigation systems; a prediction of the water required for

67 crops including drainage water analyses; and automatic results estimated as a function of an
68 optimization algorithm based on agronomical criteria (expert based optimization system). Therefore,
69 the availability of a software application (app) which can be used with mobiles or tablets can help
70 farmers, students or agricultural engineers in the making decision process, allowing them both the
71 access from any place and use of data (through internal and external databases), previous calculations
72 and a detailed knowledge of the fertigation process.

73 In this paper, an app for mobile devices able to estimate the suitable parameters for the fertigation
74 problem as a function of both, the main characteristics of the cropping system and the type of crop, is
75 proposed. More specifically, the work describes an app that allows the user to perform an appropriate
76 management of fertilizers with the main aim of providing optimal conditions for the crop growth
77 optimizing, at the same time, the use of water and fertilizers. Additionally, it also allows keeping a
78 simultaneous control of several cropping systems by means of persistent databases. In this line, the use
79 of this app will provide agricultural engineers, farmers and students with a good support tool for
80 decision-making systems in fertigation management. The development of this platform has been
81 divided into two different parts: (1) Obtaining the amount of water lost by transpiration from virtual
82 sensors that use external greenhouse conditions to estimate this energy cost. This calculation must be
83 made to determine the amount of dry matter that must be produced and that replaced through
84 irrigation. Concretely, it is carried out through physical models that allow obtaining the inside
85 greenhouse conditions (temperature, humidity and radiation) from external conditions and a virtual
86 sensor. This virtual sensor needs the inside greenhouse conditions as inputs in order to obtain the
87 amount water lost by transpiration. Therefore, it allows both to obtain data from meteorological
88 stations and to establish the irrigation necessities; (2) Determining of the fertilizers that have to be
89 applied per liter of water in fertigation tanks (kg/l). This calculation starts with different analyses (soil,
90 irrigation water and, in the case of soil-less culture, drainage) and an ideal solution for each crop and
91 growth state is obtained. This optimal solution is included in the greenhouse/farmer profile. To do that,
92 further information related to the type of fertigation system, size and number of fertigation tanks,

93 fertilizers concentration, nitric and phosphoric acids characteristics, etc. is required; Moreover, the
 94 evaluation of different drainage and soil analyses in order to modify the optimal solution as a function
 95 of the obtained results. Hence, the developed application will work as both, a learning system and a
 96 decision-making tool for fertigation systems, which increases the accessibility to the methodology
 97 used to determine the optimal fertigation parameters of any cropping system.
 98 The rest of the paper is organized as follows: Section 2 performs a detailed explanation of the different
 99 materials and methods which have been used to develop the app proposed in this paper. Section 3 is
 100 devoted to discuss the functionalities of the proposed app and a real example of use is included and
 101 widely commented. Finally, in Section 4, the main conclusions and future works are shown.

102 **2. Materials and methods**

103 2.1. Estimation of water necessities from external conditions

104 The water necessities are established using the modified version (Sánchez et al., 2012) of the
 105 transpiration virtual sensor proposed by (Jolliet and Baille, 1992). This equation essentially combines
 106 the equation for heat transfer between the crop and the mass of the surrounding air to describe the
 107 transpiration ($M_{tp,cr}$) as a process based on two main variables: the solar radiation ($X_{rad,int}$) arriving at a
 108 particular depth in the canopy plant, and the vapor pressure deficit (V_{VPD}). The reduced virtual sensor
 109 is shown in the following equation:

$$110 \lambda M_{tp,cr} = C_A (1 - e^{-C_K X_{LAI}}) X_{rad,int} + C_B V_{VPD} X_{LAI} \quad (1)$$

$$111 \lambda M_{tp,cr} = Q_{tp,cr} \quad (2)$$

112 where $Q_{tp,cr}$ is the transpiration used for energy balances in Wm^{-2} , X_{LAI} is the Leaf Area Index (m^2_{leaf}
 113 m^{-2}_{soil}), C_K is the extinction coefficient and the coefficients C_A and C_B are parameters dependent on the
 114 type of crop.

115 Thus, to obtain V_{VPD} is necessary to estimate the greenhouse air temperature, Eq. (3), and relative
 116 humidity, Eq. (4), from external conditions. It has been assessed by means of energy and mass balance
 117 equations proposed by (Rodriguez et al., 2015):

$$118 C_{sph,a} C_{den,a} C_{vol,g} C_{area,s}^{-1} X_{t,a} = Q_{env,aero-a} + Q_{env,heat-a} + Q_{env,cv-a} + Q_{env,ss-a} - Q_{ven} - Q_{tp,cr} \quad (3)$$

119 where $Q_{\text{env,aero-a}}$ represents the heat fluxes with the pipe-based heating system, $Q_{\text{env,heat-a}}$, the heat fluxes
 120 with the aerothermal heating system, $Q_{\text{env,cv-a}}$ is the convective flux with the cover, $Q_{\text{env,ss-a}}$ is the
 121 convective flux with the soil surface, Q_{ven} is the heat lost by natural ventilation and the heat lost by
 122 infiltration losses, $Q_{\text{tp,cr}}$ is the latent heat effect from crop transpiration, $C_{\text{sph,a}}$ is the product of specific
 123 air heat, $C_{\text{den,a}}$ is the air density and $C_{\text{vol,g}}$ and $C_{\text{area,s}}$ represent the effective height of the greenhouse
 124 (that is, the greenhouse volume and soil surface area). Furthermore, the relative humidity ($X_{\text{hr,a}}$) can be
 125 estimated as it is shown in Eq. (4):

$$126 \quad C_{\text{den,a}} C_{\text{vol,g}} C_{\text{area,s}}^{-1} X_{\text{hr,a}} = M_{\text{tra,cu}} + M_{\text{ls,ss}} - M_{\text{lc,b-a}} - M_{\text{ven,a-e}} - M_{\text{per,a-e}} \quad (4)$$

127 where $M_{\text{tra,cu}}$ is the amount of water lost by transpiration, $M_{\text{ls,ss}}$ represents the amount of water vapor
 128 generated by soil evaporation, $M_{\text{lc,b-a}}$ is the water condensation on the plastic cover, and finally, $M_{\text{ven,a-}}$
 129 e and $M_{\text{per,a-e}}$ are the humidity lost by ventilation and infiltration, respectively.

130 The Leaf Area Index (X_{LAI} , $\text{m}^2_{\text{[leaves]}} \text{m}^{-2}_{\text{[soil]}}$) was obtained through a mechanistic model based on the
 131 works of (Jones et al., 1999; Ramírez et al., 2004):

$$132 \quad X_{\text{LAI}} = \rho \delta e^{(\beta(X_n - N_b))} / (1 + e^{(\beta(X_n - N_b))}), \quad \text{if } X_{\text{LAI}} > C_{\text{LAI,max}} \quad (5)$$

$$133 \quad X_{\text{LAI}}/dt = 0, \quad \text{if } X_{\text{LAI}} > C_{\text{LAI,max}} \quad (6)$$

134 where ρ is the density of plants, β is an empirical coefficient, δ is the maximum expansion of leaf area
 135 per node, N_b is a daily-average temperature-dependent linear function, and X_n is the number of nodes:

$$136 \quad dX_n/dt = N_m f_N(X_{T_d}) \quad (7)$$

137 In the previous equation, N_m is the node appearance rate and $f_N(X_{T_d})$ is a temperature linear function.

138 The appearance of nodes is modelled considering temperature as the main regulating element.

139 It is important to notice that to obtain the water necessities from this model, an external climate data
 140 pattern-year stored in the system was used. Therefore, in this case, there is no need of obtaining data
 141 measured from meteorological stations. This option will be included in following versions.

142

143 2.2. APP development software

144 In this paper, an improved version of the mobile application presented in (Pérez-Castro et al., 2016) is
145 introduced. More specifically, the previous version was developed using Mit App Inventor 2 (App
146 Inventor, 2016). On the contrary, the app presented in this paper has been implemented using Android
147 Studio 2.1. which is a free development environment for Android platform. Besides, the new version
148 of the app includes some improvements in comparison with its previous one. For instance, the
149 inclusion of a linear-programming optimization algorithm to estimate the ideal amount of fertilizers
150 which allows both, setting to zero the error associated with some components, and to establish a
151 minimum amount of certain fertilizers by defining several constraints in the optimization algorithm.
152 This algorithm has been implemented using an open-source library named JOptimizer (JOptimizer
153 library, 2016), which has been developed in pure Java programming Language. Additionally, a
154 persistent database has been implemented using SQLite which is an embedded SQL database engine
155 that implements a self-contained, server-less, zero-configuration, transactional SQL data engine
156 (SQLite library, 2016). The main advantage of this SQL data engine in comparison with other
157 approaches is that it does not have a separate server process.

158 As it was mentioned previously, this new version of the mobile application can be used with any
159 device with Android Operating System (OS) and it has been developed using Android Studio 2.1.
160 (Android Studio, 2016), which is the official IDE for Android app development and is based on
161 IntelliJ IDEA (IntelliJ IDEA, 2016). The main reason to choose Android as OS has been its associated
162 market share which was approximately equal to 82.8% in the second four-month period of 2015 (IDC
163 Corp, 2016). Moreover, this app was carried out by using the free Android Software Development Kit
164 (SDK) of Google (Google Inc., 2016) since it supplies the necessities tools and Application
165 Programming Interface (API) libraries in order to build and test any application for any device with
166 Android OS. Besides, the app presented in this paper has been designed to be compatible with a wide
167 range of devices and several versions of Android OS. Concretely, it is compatible with version 4.03
168 (API Level 15) since 97.7% of Android based devices have this version or higher, including the latest
169 available, version 6.0 (Android dashboards, 2016).

170 The developed fertigation management app provides to agricultural engineers, farmers and students
171 the possibility to obtain, in an automatic and transparent way, some optimized parameters related to
172 the fertigation process by means of a friendly and easy to use interface. A detailed description of the
173 methodology used to estimate these parameters has been included in this paper. Furthermore, the
174 installation and configuration process of this app is very simple, and moreover, it is able to adapt itself
175 to the language selected in the mobile device between English (default language) and Spanish.

176

177 2.3. Fertigation calculation criteria

178 Drip fertigation consists on the application of water and fertilizers through emitters (drippers)
179 originating from distribution tubes where water flows with a certain pressure in order to ensure the
180 irrigation uniformity along the system. Furthermore, these emitters have a low flow (around 3 l h^{-1}) in
181 order to maintain an appropriate soil or substrate water content by short and frequent water supply.
182 Moreover, this system allows both including and localizing the fertilizers nearby the crop roots, and
183 thus, the water content in soil/substrate is close to field capacity. The fertigation equipment helps in
184 this task, since the use of this type of systems allows to maintain a certain pressure (impulsion
185 system), one or more fertilizers tanks, water filtration systems, different nutrients solutions
186 (fertilization tanks), distribution tubes and drippers. Although, there are also many divergences in the
187 different systems which provoke that the optimal distribution of the fertilizers in tanks depends on the
188 specific characteristics of the system:

189 - Full automatic fertigation system: the use of this equipment allows the automatic EC and pH
190 control by regulating the amount of fertigation water. Generally, these systems are composed
191 of a programmable unit (a computer or a controller), injection pumps or venturi flow meters,
192 mixing tanks (normally 3 tanks, but there can be many configurations, and thus, in this app
193 they have been limited to a maximum of 12 tanks) which allows to separate the different
194 fertilizers avoiding incompatibilities problems (Table 1), a tank for nitric acid to control the
195 pH and, finally, as a function of the type of system, is possible to find different sensors that

196 assist to the equipment (pH, flow and EC) and elements that contribute to the irrigation
197 amount decision (evapotranspiration models, demand tray, evaporation tank, etc.). The
198 fertigation process starts setting the desired EC and pH values and the used percentage of each
199 fertigation tank obtained from the optimal configuration decided for the farmer, normally, in a
200 system with two tanks this percentage is equal to 50% for each tank, and equal to 25% in the
201 case of a system with four tanks. These fertilizers use distribution help in order to establish the
202 same level of tank emptying.

203 - Automatic fertigation system without nitric acid tank: This system is similar to the previously
204 commented, but in this case, there is not a dedicated nitric acid tank. Hence, the nitric acid
205 should be incorporated in one of the other tanks, and thus, the quantity of nitric acid must be
206 obtained with the optimal solution.

207 - Classical fertigation system: The fertigation system is composed of one or two tanks, a simple
208 distribution system and manometers without a specific place for the nitric acid.

209 Table 1. Incompatibilities among fertilizers, adapted from (Smart-fertilizer, 2012).

210

211 As it was mentioned above, drip fertigation offers significant advantages. However, despite this fact,
212 many Almeria's farmers are neglecting these advantages both by obviating soil or substrate, or even
213 drainage analyses, and by avoiding an optimal nutrients solution according to the specific necessities.
214 On the contrary, they usually make this operation based on their own experience and without using
215 analytical measurements. These analyses should be done periodically in order to determine, in a
216 precise way, the ideal nutrients solution to the crop necessities at each crop growth stage. Even so,
217 these operations have not often been set back for many years, wasting water and fertilizers. It can be
218 translated into money loss and underground water contamination. The problem is that these operations
219 require specific knowledge that, in many cases, have not been acquired or applied by farmers or
220 agricultural engineers. There is no doubt that these engineers have acquired this knowledge but, maybe
221 for lack of time or tools that assist in this task, they recommend some fertigation based on experience

222 or the recommendation provided by fertilizers sellers which frequently is not adjusted to the
 223 agronomical necessities. Therefore, an adequate fertigation consists in providing the anticipated
 224 quantity of nutrients that can be absorbed by the crop as essential elements which it needs. These
 225 elements have a great influence over the crop growth at each period of the crop cycle and over the
 226 final harvest collected. Moreover, these nutrients necessities are estimated based on the crop
 227 requirements and modified according to the soil or drainage (in the case of soil-less culture) nutrients
 228 content. These analyses give information about the nutrients absorption as a function of the
 229 environmental conditions and the type of crop.

230 Furthermore, in modern facilities is imposing the use of a simple approach which consists in adding
 231 fertilizers directly in the different tanks solutions with an effective injection. This approach makes this
 232 operation cheaper and easier, and allows reaching a huge variety of different final solutions. In this
 233 work, only the most used solid fertilizers are taken into account to carry out this operation: (i) nitric
 234 acid (HNO_3), strong acid whose main function, apart from supplying nitrogen to crop, is acidifying
 235 irrigation water for an optimal pH control by removing the bicarbonates. In the most advanced system,
 236 this acid is injected from a separate tank to the rest of fertilizers tanks. This injection is controlled
 237 through the measurements provided by a pH meter. (ii) phosphoric acid (H_3PO_4), as an acid, is also
 238 involved in the destruction of bicarbonates (they have a high concentration in Almeria's underground
 239 water used for irrigation) and in the pH control, Eq. (5), similarly to as nitric acid does and (iii) calcic
 240 nitrate ($5[\text{Ca}(\text{NO}_3)_2 \cdot 2\text{H}_2\text{O}] \text{NH}_4\text{NO}_3$), ammoniac nitrate (NH_4NO_3), ammoniac sulfate ($(\text{NH}_4)_2\text{SO}_4$),
 241 monoammoniac fosfate ($\text{NH}_4\text{H}_2\text{PO}_4$), monopotassium fosfate (KH_2PO_4), potassium nitrate (KNO_3),
 242 potassium sulfate (K_2SO_4), magnesium sulfate ($\text{MgSO}_4 \cdot 7\text{H}_2\text{O}$), sodium chloride (ClNa) and potassium
 243 chloride (ClK), which supply different nutrients to the final solution.

244 $\Sigma(\text{C}_{\text{HCO}_3, \text{wat}} - \text{C}_{\text{acids}}) = \text{C}_{\text{HCO}_3, \text{opt}}$, without pH problems (5)

245 where $\text{C}_{\text{HCO}_3, \text{wat}}$ is the bicarbonates concentration in the irrigation water and C_{acids} is the concentration
 246 of the different acids included in the final solution.


```

277      0 0 0 0 0 0.5 1 0 0 0 0.5 0 0;
278      0 0 0 0 0 0 0 0 0 0 0 1 1;
279      0 0 0 0 0 0 0 0 0 0 0 0 1;
280      ]

```

```

281  Aerror = -[
282      1 0 0 0 0 0 0 0 0 0;
283      0 1 0 0 0 0 0 0 0 0;
284      0 0 1 0 0 0 0 0 0 0;
285      0 0 0 1 0 0 0 0 0 0;
286      0 0 0 0 1 0 0 0 0 0;
287      0 0 0 0 0 1 0 0 0 0;
288      0 0 0 0 0 0 1 0 0 0;
289      0 0 0 0 0 0 0 1 0 0;
290      0 0 0 0 0 0 0 0 1 0;
291      0 0 0 0 0 0 0 0 0 1;
292      ]

```

```

293  for i=1...23, xi >= 0          (6)

```

294 In the previous equations A_{eq} is the coefficient matrix, x_i are the decision variables, B_{eq} are the
 295 independent terms and A_{error} is the starting error matrix. However, as the optimal solution is considered
 296 as the minimum objective to fulfil along the supply process, the decision variables will never be
 297 negative, Eq. (6).

298 Once the final equilibrium has been obtained, the weight associated with the different fertilizers
 299 previously mentioned has to be calculated. Afterwards, the final quantity and distribution of the
 300 fertilizers depend on the characteristics of both the system and the fertilizers: incompatibilities among
 301 fertilizers (Table 1), the type of installed fertigation system, the number of tanks, the concentration of
 302 the fertilizers in the tanks and the configuration of the system in the case of having more than one tank
 303 (percent of emptying associated with each tank).

304 Finally, the results validation is mandatory due to several problems that could provoke problems in the
 305 crop or even in the installed equipment. This validation is established by the app and provided to the
 306 user in the final report: pH problems caused by the bicarbonates concentration in the irrigation, Eq.
 307 (5), and solution precipitation risk in tanks due to the different fertilizers concentration and the
 308 solubility (S_x) associated to the temperature (Table 2), the weight of the different fertilizers (P_x) and
 309 the volume of the mixed tank (V_{tank}):

$$S_{fer} = \Sigma(P_x/V_{tank}) < S_x \quad (7)$$

Table 2. Solubility (S_x) of the different fertilizers as a function of the temperature

Finally, obtaining water necessities from external conditions can be useful for farmers in order to make decisions regarding the programming of greenhouse fertigation to maintain an optimal crop growth. This estimation is established by taking into account the different crops necessities as a function of the external climate variables prediction.

3. Results and discussion

The mobile application presented in this paper offers several possibilities and functionalities in order to boost the usefulness of fertigation and the user experience. Moreover, the use of the proposed app allows the farmers a mechanism to make aware themselves about the advantages of installing modern fertigation control systems. The management of this type of tools and processes is, in general, tedious and complex, and thus, farmers have complained about the difficulty and amount of time it takes to carry out them. The development of this app (Fig. 1a) was performed with the main objective of making the water and fertilizers supply task easier to understand by the farmers, students and agricultural engineers. The different irrigation equipments and crops possible configuration provided by this app make it an interesting and valid tool to be used with them. This variety of options allows users to try out different combinations and compare the obtained results. The app displays valuable information by means of a results report.

Fig. 1. Some functionalities of app: a) Logo and first screen of the tool, b) Main interface, calculation definition and c) greenhouse and irrigation system description.

The first step for the DSS is to define the simulation (Fig. 1b). In this stage, the main information is ordered by means of different levels: the first level is associated with the farmer name, which will help to identify his/her experiments stored in the data base. The second level is composed of the different

336 greenhouses related to a farmer defined in the first level. This level includes the main characteristics of
337 both the greenhouse, since they are necessary for the water supply model and, the irrigation equipment
338 (Fig. 1c). For instance, some of the main characteristics included in this level are: type of irrigation
339 equipment (Full automatic fertigation system for this example), greenhouse surface (1000m^2), drips
340 flow (3 l h^{-1}), nitric and phosphoric acids richness and density (56% and 1.35g cc^{-1} , and 73% and 1.56g
341 cc^{-1} , respectively), number of irrigation tanks (4), tanks volume (1000l), tanks concentration (100
342 concentrations) and tanks percentage distribution (25% each). Afterwards, the third level establishes
343 the crop season (Fig. 2a) in order to obtain the information related to the crop growth: variety and
344 plant density and the optimal mixed solution established for it (Fig. 2b, Table 3). In the end, the final
345 report with the main results is obtained.

346 Fig. 2. Fertigation main characteristics and results: a) Crop season configuration b) Optimal
347 fertigation solution for tomato crops in greenhouses and c) Analysis of the irrigation water.

348 Table 3. Optimal solution for tomato crop fertigation (mmol/l)

349 Moreover, the analysis of the irrigation water (Fig. 2c, Table 4) is introduced manually to the app and
350 stored in the DB to be used with one or more crops or greenhouses. The irrigation water analysis and
351 the optimal solution are used to obtain the different ionic nutrient components that should be supplied
352 by the fertigation system (Table 5) and that are used to obtain the final results (Table 6) that assist
353 establishing the weight of the different fertilizers (Table 7).

354 Table 4. Analysis of the irrigation water (mmol/l)

355

356 Table 5. Nutrients that must be supplied by fertigation (mmol/l)

357

358 Table 6. Nutrients calculated by the app (mmol/l)

359

360 Table 7. Final fertilizers weight and water necessities estimated by the app (kg)

361

362 Finally, it is possible to generate a summary report (Fig. 3a) which contains a summary of the
363 estimated numeric values associated with the weight of the different fertilizers (Table 7) and the
364 results obtained from the analysis of the drainage (Table 8) highlighting the most significant problems
365 with the supply or absorption of the different nutrients. On the other hand, Fig. 3b shows the
366 distribution of the different fertilizers in tanks taking into account the incompatibilities (Table 1), the
367 percentage of distribution in tanks and the fertilizers weight calculation.
368 Fig. 3. Result obtain with the app: a) Final weight of the different fertilizers and b) Tank distribution
369 of the different fertilizers.

370 Table 8. Analysis of the drainage (mmol/l)

371

372 **4. Conclusions and Future Works**

373 This work consists in the development of an app for the fertigation management in greenhouses using
374 different combinations of equipments and crops. More specifically, an app named cFertigUAL was
375 developed for android mobile devices in order to help with the fertigation (fertilizers + irrigation)
376 management on greenhouse crops. Overall, the proposed optimization method showed a strong
377 agreement with the agronomical criteria of crop fertigation. This optimization is solved by means of a
378 linear programming algorithm which has been implemented using the library JOptimizer which is able
379 to solve equations with constraints as inequations. Moreover, a practical case of use has been included
380 in this paper. To do that, data from a greenhouse and a pattern-year of external climate data have been
381 used. The proposed app calculated the fertilizers and water requirements for a tomato crop in
382 greenhouse growing conditions.

383 In addition to this, cFertigUAL has several advantages, like the configuration of different fertigation
384 systems as a function of the system complexity, the possibility of generate a database of irrigation
385 water, drainage and soil analyses, the introduction and storage of different optimal mixed solutions for
386 each crop and the estimation of the water that must be restored by irrigation.

387 As future works, the app will be provided of new functionalities: an optimization process based in
388 economical criteria. To do that, an editable fertilizers list will be included with their associated prices
389 (updated manually by user or automatically by internet), external climate data provided by
390 meteorological stations which will include forecast of the main external climate variables that will
391 allow to obtain water supply predictions and, finally, the inclusion of different models of nutrient
392 absorption to assist in the decision of the optimal nutrient solution.

393

394 **Acknowledgements**

395 This research has been funded by the 'Controlcrop' Project, P10-TEP-6174, project framework,
396 supported by the Andalusian Ministry of Economy, Innovation and Science (Andalusia, Spain), and by
397 the National R+D+i Plan Project DPI2014-56364-C2-1-R of the Spanish Ministry of Economy and
398 Competitiveness and ERDF funds. M. Castilla is a fellow of the Spanish "Juan de la Cierva -
399 Formación" contract program.

400

401 **Bibliography**

402 Android Studio. 2016. Available online at:

403 <http://developer.android.com/intl/es/tools/studio/index.html>. Last access 09th may, 2016.

404 Android dashboards. 2016. Available online at:

405 <http://developer.android.com/intl/es/about/dashboards/index.html>. Last access 09th may, 2016.

406 Ansorena M.J. 1994. Substrates. Properties and characterization. Mundi Prensa. Pp 172. (In

407 Spanish)

408 App Inventor. 2016. Available online at: <http://appinventor.mit.edu/explore/about-us.html>. Last

409 access 09th may, 2016.

410 Bueno-Delgado, M.V., Molina-Martínez, J.M., Correoso-Campillo, R. Pavón-Mariño, P. 2016.

411 Ecofert: An Android application for the optimization of fertilizer cost in fertigation, Computers and

412 Electronics in Agriculture, 121, 32-42.

- 413 Carvajal, F., Agüera, F., Sánchez-Hermosilla, J., 2014. Water balance in artificial on-farm
414 agricultural water reservoirs for the irrigation of intensive greenhouse crops, *Agricultural Water*
415 *Management*. 131, 146-155.
- 416 Domeño, I., Irigoyen, N., Muro, J., 2009. Evolution of organic matter and drainages in wood fibre
417 and coconut fibre substrates. *Scientia Horticulturae*.122 (2), 269-274.
- 418 Fernández, M.D., González, A.M., Carreño, J., Pérez, C., Bonachela S., 2007. Analysis of on-
419 farm irrigation performance in Mediterranean greenhouses, *Agricultural Water Management*, 89 (3),
420 251-260.
- 421 Gieling, Th. H., 2001. Control of water supply and specific nutrient application in closed growing
422 systems. Ph. D. Thesis. Wageningen University. The Netherlands. 176 pp.
- 423 Gallardo, M., Thompson, R., Rodriguez, J., Rodriguez, F., Fernandez, M., Sanchez, J., Magan, J.,
424 2009. Simulation of transpiration, drainage, N uptake, nitrate leaching, and N uptake concentration in
425 tomato grown in open substrate. *Agricultural Water Management*. 96, 1773–1784.
- 426 Google Inc Android Software Development Kit (SDK). 2016. Available online at:
427 <http://developer.android.com/index.html>. Last access 09th may, 2016.
- 428 Homo-Agricola, 2011. Fertigation excel tool. Tool available on line at:
429 <https://sites.google.com/site/revistahomoagricola/home>. Last access 17th may, 2016.
- 430 IDC Corp Worldwide mobile statistics. 2016. Available online at:
431 <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>. Last access 09th may, 2016.
- 432 IntellJ IDEA. 2016. Available online at: <https://www.jetbrains.com/idea/>. Last access 09th may,
433 2016.
- 434 Jolliet, O., Bailey, B., 1992. The effect of climate on tomato transpiration in greenhouses:
435 measurements and models comparison. *Agricultural and Forest Meteorology*. 58, 43–62.
- 436 Jones, J.W., Kenig, A., Vallejos, C.E., 1999. Reduced state-variable tomato growth model. *T*
437 *ASAE*. 42, 255-265.

- 438 JOptimizer library. 2016. Available online at: <http://www.joptimizer.com/>. Last access 11th may,
439 2016.
- 440 Kramer, P., Boyer, J., 1995. Water relations of plants and soils. Academic Press. pp. 495.
- 441 Magán, C. J. J., Moreno, N., Meca, D., Cánovas, F., 2004. Response to salinity of a tomato crop
442 in Mediterranean greenhouse climate conditions. *Acta Horticulturae*. 644, 479–484.
- 443 Pagán, F.J., Fernández-Villena, M., Fernández-Pacheco, D.G., Rosillo, J.J., Molina-Martínez, J.M.
444 2015. Optifer: an application to optimize fertiliser costs in fertigation *Agricultural Water Management*.
445 151, 19–29
- 446 Pérez-Castro, A., Sánchez, J.A., Castilla, M., Sánchez-Moreno, J., Berenguel, M., Magán, J.J.
447 2016. CRIG: A mobile application for fertirrigation calculations (in Spanish). In proceedings of the II
448 Simposio Nacional de Ingeniería Hortícola. Automatización y TICs en Agricultura, Almería, (Spain).
- 449 Ramírez, A., Rodríguez, F., Berenguel, M., Heuvelink, E., 2004. Calibration and validation of
450 complex and simplified tomato growth models for control purposes in the southeast of Spain. *Acta*
451 *Horticulturae*. 654, 147–154.
- 452 Sánchez, J. A., Rodríguez, F., Guzmán, J. L., Arahal, M. R., 2012. Virtual Sensors for Designing
453 Irrigation Controllers in Greenhouses. *Sensors*. 12, 15244-15266.
- 454 Sánchez-Martos, F., Pulido-Bosch, A., Molina-Sánchez, L., Vallejos-Izquierdo, A., 2007.
455 Identification of the origin of salinization in groundwater using minor ions (Lower Andarax,
456 Southeast Spain); *The Science of the Total Environment*. 297, 43-58.
- 457 Sánchez-Molina, J., Reinoso, J., Ación, F., Rodríguez, F., López, J., 2014. Development of a
458 biomass-based system for nocturnal temperature and diurnal CO₂ concentration control in
459 greenhouses. *Biomass Bioenergy* 67, 60 – 71.
- 460 Smart-fertilizer, 2012. Web browser tool, available on line at: [<https://web.smart-fertilizer.com/>](https://web.smart-fertilizer.com/).
461 Last access 13th may, 2016.
- 462 Smart-fertilizer-chart: fertilizers incompatibility chart. Available on line at: [<http://www.smart-](http://www.smart-fertilizer.com/)
463 [fertilizer.com/>](http://www.smart-fertilizer.com/). Last access 13th may, 2016.

464 SQLite library. 2016. Available online at: <https://www.sqlite.org/>. Last access 09th may, 2016.

465 SSca, 2014. Fertilizer Blend Calculator. App tool available on line at

466 <<http://www.ssc.ca/index.php/28-last-added-projects/71-ssca-fertilizer-blend-calculator>>.

467

468

Table 1
[Click here to download Tables: Table 1.pdf](#)

Table 1. Incompatibilities among fertilizers, adapted from (Smart-fertilizer, 2012).

	Ammonium nitrate	Ammonium sulfate	Calcium nitrate	Potassium nitrate	Potassium chloride	Potassium sulfate	Ammonium phosphate	Magnesium sulfate	Phosphoric acid	Nitric acid
Ammonium Nitrate	✓									
Ammonium sulfate	✓	✓								
Calcium Nitrate	✓	✗	✓							
Potassium Nitrate	✓	✓	✓	✓						
Potassium Chloride	✓	✓	✓	✓	✓					
Potassium sulfate	✓	R	✗	✓	R	✓				
Ammonium Phosphate	✓	✓	✗	✓	✓	✓	✓			
Phosphoric Acid	✓	✓	✗	✓	✓	✓	✓	✓	✓	
Nitric Acid	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

✓ = Compatible.

✗ = Incompatible.

R = Reduced solubility.

Table 2[Click here to download Tables: Table 2.pdf](#)

Table 2. Solubility (S_s) of the different fertilizers as a function of the temperature (diverse origin)

Fertilizer	Solubility (g l^{-1})			
	0°C	10°C	20°C	30°C
Ammonium Nitrate	1180	1500	1920	2420
Ammonium sulfate	710	730	754	780
Calcium Nitrate	1020	1150	1290	1530
Potassium Nitrate	133	209	316	458
Potassium Chloride	282	218	342	495
Potassium sulfate	74	93	111	131
Potassium phosphate	143	188	227	303
Ammonium Phosphate	220	280	365	458

Table 3[Click here to download Tables: Table 3.pdf](#)

Table 3. Optimal solution for tomato crop fertigation (mmol/l)

HCO_3^-	NH_4^+	NO_3^-	H_2PO_4^-	K^+	Ca^{2+}	Mg^{2+}	SO_4^{2-}	Cl^-	Na^+	CE (dS/m)
0.50	0.00	12	1.50	6	4	1.50	1.50	0	0	3.50

Table 4[Click here to download Tables: Table 4.pdf](#)

Table 3. Analysis of the irrigation water (mmol/l)

HCO_3^-	NH_4^+	NO_3^-	H_2PO_4^-	K^+	Ca^{2+}	Mg^{2+}	SO_4^{2-}	Cl^-	Na^+	CE (dS/m)
4.80	0.00	0.70	0.00	0.20	1.60	1.90	1.30	6.20	4.90	1.17

Table 5[Click here to download Tables: Table 5.pdf](#)

Table 5. Nutrients that must be supplied by fertigation (mmol/l)

HCO_3^-	NH_4^+	NO_3^-	H_2PO_4^-	K^+	Ca^{2+}	Mg^{2+}	SO_4^{2-}	Cl^-	Na^+	CE (dS/m)
-4.30	0.00	11.30	1.50	5.80	2.40	-0.40	0.20	-6.20	-4.90	1.17

Table 6[Click here to download Tables: Table 6.pdf](#)

Table 6. Nutrients calculated by the app (mmol/l)

HCO_3^-	NH_4^+	NO_3^-	H_2PO_4^-	K^+	Ca^{2+}	Mg^{2+}	SO_4^{2-}	Cl^-	Na^+	CE (dS/m)
-4.3	0.48	11.58	2.48	5.80	2.40	0	0.66	0	0	1.24

Table 7

[Click here to download Tables: Table 7.pdf](#)

Table 7. Final fertilizers weight and water necessities estimated by the app (kg)

H ₃ PO ₄	Ca(NO ₃) ₂	KNO ₃	NH ₄ NO ₃	K ₂ SO ₄	MgSO ₄	NH ₄ H ₂ PO ₄	KH ₂ PO ₄	Mg(NO ₃) ₂	(NH ₄) ₂ SO ₄	ClK	ClNa	Water necessities (l)
59.35	365.18	173.04	0	154.26	0	0	75.58	0	0	0	0	3890
Tank 3	Tank 1	Tank 2		Tank 4			Tank 2					

Table 8[Click here to download Tables: Table 8.pdf](#)

Table 8. Analysis of the drainage (mmol/l)

HCO ₃ ⁻	NO ₃ ⁻	H ₂ PO ₄ ⁻	K ⁺	Ca ²⁺	Mg ²⁺	SO ₄ ²⁻	Cl ⁻	Na ⁺	CE (dS/m)	pH
1.71	16.90	3.21	18.05	27.75	5.42	8.42	23.84	27.75	7.1	6.88

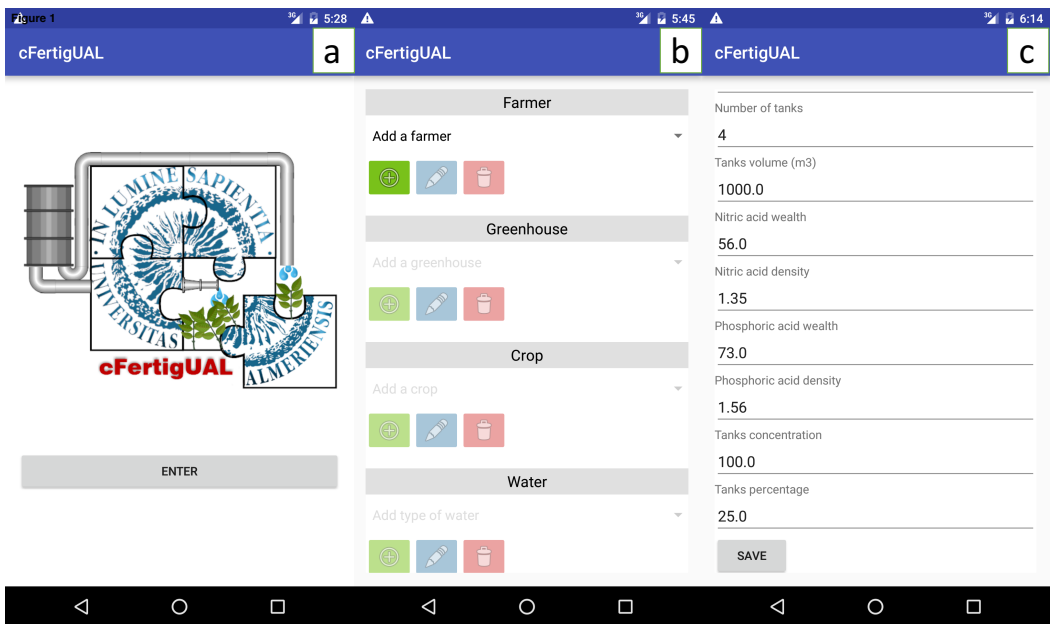


Figure 2

a




cFertigUAL

Name of the crop
Tomato

Crop date
26/08/2015

Crop density
2.01

Cultivated product (optimal solution)
Tomato1
TomatoFull

SAVE

b

cFertigUAL

Name of the product
Tomato1

Name of the ideal solution
TomatoFull

HCO3 (mmol/l)
0.5

NH4 (mmol/l)
0.0

NO3 (mmol/l)
12.0

H2PO4 (mmol/l)
1.5

K (mmol/l)
6.0

Ca (mmol/l)
4.0

Mg (mmol/l)
1.5

c

cFertigUAL

Name of water
WaterG1

Date of values measurement
24/11/2015

HCO3 (mmol/l)
4.8

NH4 (mmol/l)
0.0


NO3 (mmol/l)
0.7

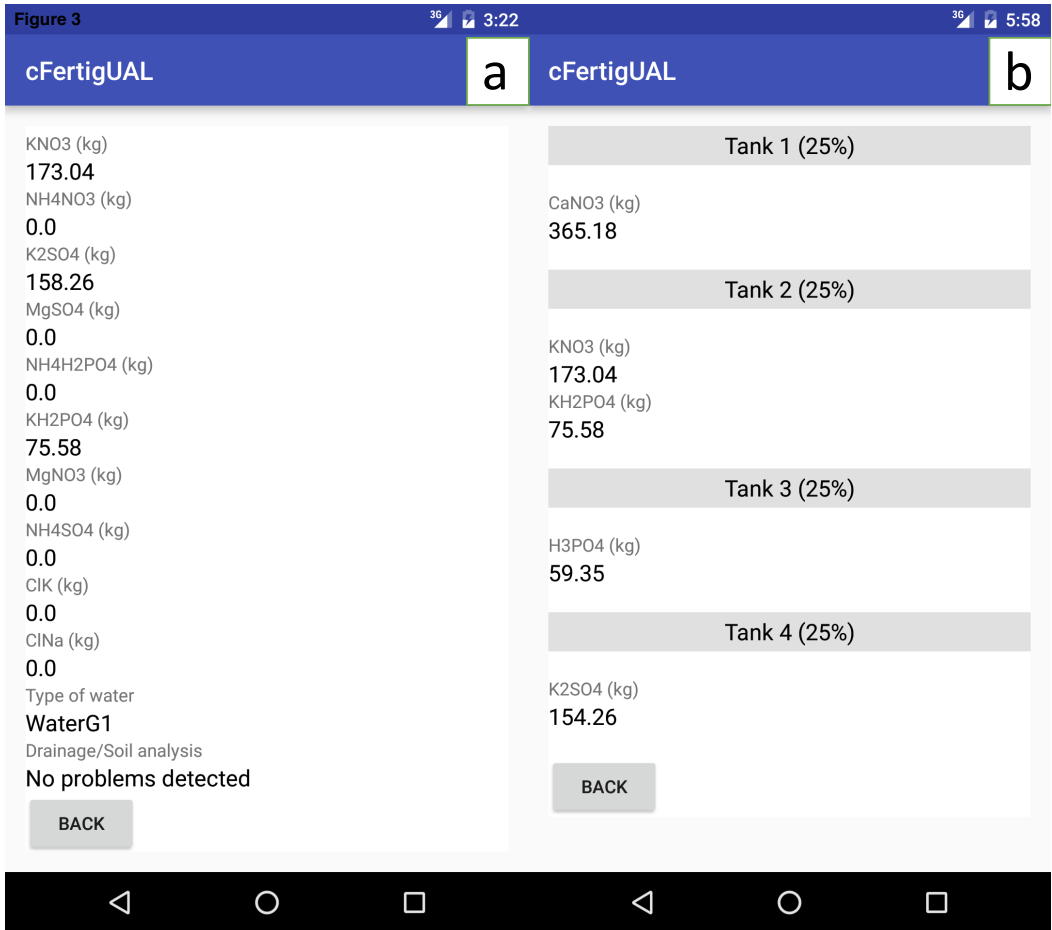
H2PO4 (mmol/l)
0.0

K (mmol/l)
0.2

Ca (mmol/l)
1.6

Mg (mmol/l)
1.9





PhotoBioLib: a Modelica library for modeling and simulation of large-scale photobioreactors

A. Perez-Castro^{a,*}, J. Sanchez-Moreno^a, M. Castilla^b

^a*Dept. of Computer Sciences and Automatic Control, Universidad Nacional de Educación a Distancia (UNED), Madrid, Spain*

^b*Department of System Engineering and Automation, School of Engineering, University of Seville, Seville, Spain.*

Abstract

This paper presents *PhotoBioLib*, a library developed in *Modelica* language for modeling and simulation of industrial tubular photobioreactors without necessity of a previous knowledge of programming neither the involved dynamics. Therefore, it provides models very similar to the real ones, with a high configuration potential, interconnection possibilities with other libraries and a high simulation speed. The library has been validated with real data.

Keywords: Photobioreactors, Modeling, Simulation, Modelica, PhotoBioLib

1. INTRODUCTION

In recent years, the cultivation of microalgae for production of energy as an alternative to fossil fuel and biofuel has grown in interest [1, 2]. Furthermore, it also becomes relevant for the production of biochemical components for food or medicine [3].

The cultivation of microalgae requires adequate facilities for their culture [4, 5], which can be classified into two different groups, open and closed photobioreactors.

The main feature of open photobioreactors (raceways) is that they are open to the air. As open system can be in the form of a pond, lake, or some large

*Principal corresponding author

Email addresses: agustinperezcastro@bec.uned.es (A. Perez-Castro), jsanchez@dia.uned.es (J. Sanchez-Moreno), mcastilla4@us.es (M. Castilla)

open vessel. The main benefit of open photobioreactors lies in their easier construction and that they are cheaper and easier to maintain than closed ones. On the other hand, their main disadvantages are: the water can evaporate, crop contamination is a risk to be considered, large land areas are required, and in relatively deep crops the light distribution can be a problem.

In closed photobioreactors, the light, air, and nutrients can be controlled in order to achieve an optimal growth. Their main drawbacks are the costs associated with their construction and maintenance. Regarding their advantages they can limit contamination problems, higher production rates than open ones, small evaporation of the medium, and an easier control and handling of the crop. Within the closed photobioreactors there are different geometries for efficient distribution of light, the most common ones are the flat plate, the annular and the tubular ones [6].

The design and development of photobioreactors in order to maximize the production qualitatively and quantitatively involves various aspects ranging from radiation, form and distribution, transfer of matter, scalability, nutrition, temperature, pH, and, of course, operating conditions and thus, the control strategies necessary to maintain the desired conditions. Therefore, and mainly due to the facts that have just been mentioned, it is necessary the use of tools to model and simulate the different aspects that can affect the performance of photobioreactors. It will allow researchers to evaluate designs and strategies that improve productivity, and serve moreover, as a simulation or learning platform prior to the construction of real plants.

Although there is not a recent interest in tubular photobioreactors, it does not exist methods or specific software for their modeling [7]. The complexity of the modeling and the parameters identification, the difficulties in the measurement of biological systems and the existing diversity in the configuration of photobioreactors are somewhat the reasons of the lack of specific tools or libraries for the modeling and simulation of these systems. Hence, the *PhotoBioLib* library has been developed in the *Modelica* language. Although the aforementioned library has the ultimate objective of allowing the development

of any kind of photobioreactor model, for the first version, it has been decided to implement a closed tubular photobioreactor. However, the equations which are included in this paper and their implementation are intended for reuse in any kind of photobioreactor.

The rest of the paper is organized as follows. The next section refers to the state of the art on modeling and simulation of photobioreactors and the main features of the *Modelica* language and the *Dymola* environment are enumerated. In Section 2 the modeled phenomena and the equations that describe it are exposed. Later, Section 3 discusses the design of the library from the perspective of the considered domains, its composition scheme, and its packages architecture. The obtained results using the library are presented and widely discussed in Section 4. Finally, the main features of the library along with possible improvements are enumerated in Section 5.

1.1. STATE OF THE ART OF MODELING AND SIMULATION OF PHOTOBIOREACTORS

In the literature within photobioreactors area, it is only possible to find applications developed from scratch, without the support of specific libraries; or the use of more or less generic modeling and simulation programs to model and simulate some partial aspects, for example, fluid dynamics or heat transfer processes. Some of the applications used in the literature are listed below, explaining briefly the scope where they have been used and their specific utilization.

Aspen Plus is used in [8], where a study of the economic viability of open and a closed photobioreactor is developed. In this paper, *Aspen Plus* is used to model these photobioreactors by means of mass and energy balances.

In [9], where a vertical photobioreactor designed for the production of oxygen at laboratory scale is presented, *COMSOL Multiphysics* is used to study the heat transfer properties.

In [10], *Labview* is used to evaluate the potential of biomass generation and CO_2 capture taking into account different types of algae species, photobiore-

actor types, and installation locations. As a result, the *EvAlgae* tool has been developed.

In [11] *ANSYS Fluent* is used to perform a numerical simulation of the stirring performance, the historic light intensity received by algae, and the fluid patterns in different types of photobioreactor designs.

Using *Matlab* and *Simulink* as generic programming, modeling, and simulation tools; in [12] a dynamic model of the microalgae cultivation is implemented, simulated, calibrated, and validated. This model takes into account the mass and energy transfer processes and the biological phenomena.

MeteoNorm is used in [13] to generate the meteorological data required for simulate and develop a closed photobioreactor temperature model to research the parameters that influence its thermal behavior.

In [14], *SuperPro Designer* is used to modeling the CO_2 capture process, considering both the biomass growth in addition to the oil separation process. The main aim is to develop a sustainable technology at pilot scale in order to capture the flue gas of a power plant for the production of biodiesel by the use of microalgae.

The *Tracepro* software is used to design an optimal light plate, that based on a solar collector patent, uses optical fiber wires to distribute the light through the photobioreactor [15].

Also in literature it possible to find papers where combinations of the previous tools are used. For example, in [16], *COMSOL Multiphysics* and *Aspen Plus* are used to study the fluid dynamics and the CO_2 intake in a helicoidal photobioreactor.

The above tools only describe partial aspects of the complete process, and moreover, some of them, do not allow the calculation of transients, showing only static relationships. Furthermore, the library developed in *Modelica* allows the interconnection with controllers in order to test their performance or to perform more complete simulations through the use of other models can be interconnected, such as solar radiation or environmental conditions in specific environments like, for example, a greenhouse.

1.2. MODELICA

Modelica is an object-oriented modeling language that makes easier the modeling and simulation of physical systems. *Dymola* is an environment that implements such language, maybe the most important one that uses it. *Dymola* has a commercial license, but also it is possible to use free tools like *OpenModelica* or *JModelica*.

The most important features of *Modelica* are:

- Acausal description of models, that is, based on equations. The causality assignment is performed by the tool.
- The possibility of using algorithms. It allows the description of models with both a manual assignation of causality as well as the use of functions which define causality.
- Inherent characteristics of an object-oriented modeling language, like: abstraction, encapsulation, modularity, concealment principle, polymorphism and multiple inheritance.
- The use of annotations for the graphical description of the different components just as the possibility of including HTML documentation.
- The possibility of implementing hybrid models, that is, with continuous and discrete parts and the use of time and state events.
- The models which have been developed with *Modelica* can be connected to other external software using the FMI (Functional Mock-up Interface) standard. This characteristic is fundamental from a control engineering point of view since it allows to execute any control algorithm which has been implemented using general purpose software such as Matlab. FMI is a tool independent standard to support both model exchange and co-simulation of dynamic models using a combination of XML-files and compiled C-code [17].

Moreover, according to the *Dymola* environment, the following features should be highlight:

- Efficient simulation of models. This tool makes symbolic manipulations and translates the models into C language, which allows a better performance regarding other interpreted languages.
- Improvements of the algorithms for the symbolic manipulation and integration with other tools.
- Results can be represented by means of both graphical results and 3D simulation representations.

The use the commercial tool *Dymola* instead of their free variants such as *OpenModelica* or *JModelica* has been motivated by different reasons:

- A faster adaptation from the tool to the relatively rapid variations in the standard and their libraries.
- Ease of use in comparison with other alternatives such as *JModelica*.
- A greater stability compared to other alternatives like *OpenModelica*.
- A greater variety in the choice of numerical integration algorithms.
- Possibility to interconnect *Dymola* models with *Matlab/Simulink* through a block provided by the own tool. This block is used in the present paper to calculate the model parameters using genetic algorithms in a similar way as it is shown in [18].
- A greater efficiency in the calculations regarding the free alternatives, as shown in [19].

2. MODELING ASSUMPTIONS

In this section, the modeling phenomena and the most important equations are described. For a detailed description [20] can be referred.

More specifically, the model of the system will be described by means of physical/chemical equations related to the processes involved in an independent way. These equations are those that make up the atomic sub-models in Section 2. With these sub-models and by means of a hierarchical composition, more complex models will be built, without the necessity of adding more equations or modify the operation of these atomic sub-models.

Therefore, the equations described below, although in this work are used to model a closed photobioreactor, may be reused, as well as the basic components that incorporate, to construct other photobioreactors through their composition.

The equations used in this library are inspired in [21] and [12]. The modeling is based on mass and energy balances, biological phenomena and some equilibrium conditions that must be satisfied.

The first phenomenon to be considered is the photosynthetic rate. This rate indicates the matter generation and consumption by the microalgae. This variation will be marked by the average radiation in a section of the photobioreactor, equation (1). In addition, the average radiation will be estimated with: the solar radiation in the horizontal surface without any obstacle, $I_0(t)$; the light trajectory, d_t ; the biomass concentration, $C_b(t)$; the biomass extinction coefficient, K_a ; and the distribution factor, α , that represents the fraction of solar radiation available in a particular area of the reactor.

$$I_{av}(t) = \frac{\alpha I_0(t)}{K_a d_t C_b(t)} (1 - \exp(-K_a d_t C_b(t))) \quad (1)$$

The O_2 generation and the CO_2 consumption, as a function of the available irradiance, $P_{O_2}(t)$ y $P_{CO_2}(t)$, are described by equations (2) y (3), where: $P_{O_2, max}$ is the maximum photosynthetic rate; m , n , K_i y z are form parameters; A_1 , A_2 , B_1 y B_2 are pre-exponential factors; C_1 , C_2 , Ea_1 y Ea_2 are activation energies; the temperature is represented by $T(t)$; $O_2(t)$ is the concentration of dissolved O_2 ; R is the ideal gas constant; the pH is represented by $pH(t)$; K_{O_2} is the oxygen inhibition constant, and; r is the respiration rate. The biomass production is showed in equation (4), where: Y_{bo} is the yield coefficient of biomass

and V_l is the liquid volume.

$$P_{O_2}(t) = \frac{P_{O_2,max} I_{av}(t)^n}{K_i \exp(I_{av}(t)m) + I_{av}(t)^n}$$

$$\left(A_1 \exp\left(\frac{-Ea_1}{R T(t)}\right) - A_2 \exp\left(\frac{-Ea_2}{R T(t)}\right) \right)$$
(2)

$$\left(B_1 \exp\left(\frac{-C_1}{pH(t)}\right) - B_2 \exp\left(\frac{-C_2}{pH(t)}\right) \right)$$

$$\left(1 - \left(\frac{O_2(t)}{K_{O_2}}\right)^z \right) - r P_{O_2,max}$$

$$P_{CO_2}(t) = -P_{O_2}(t)$$
(3)

$$P_{Cb}(t) = V_l P_{O_2}(t) Y_{bo}$$
(4)

The total inorganic carbon, C_T , is the sum of inorganic carbon species. Therefore, considering the equilibrium constants K_w , K_1 y K_2 it can be established a relation between the concentration variation of $C_T(t)$ and the concentration of $CO_2(t)$, as it is shown in equation (5).

$$\frac{dC_T(t)}{dt} = \left(1 + \frac{K_1}{H^+(t)} + \frac{K_1 K_2}{H^+(t)^2} \right) \frac{dCO_2(t)}{dt}$$

$$- CO_2(t) \left(\frac{K_1}{H^+(t)^2} + \frac{2K_1 K_2}{H^+(t)^3} \right) \frac{dH^+(t)}{dt}$$
(5)

With respect to $CO_2(t)$ and $H^+(t)$ concentrations, the electroneutrality condition have to be satisfied which leads to equation (6).

$$\frac{dH^+(t)}{dt} = \frac{\frac{K_1}{H^+(t)} + \frac{2K_1 K_2}{H^+(t)^2}}{1 + \frac{K_w}{H^+(t)^2} + \frac{K_1 CO_2(t)}{H^+(t)^2} + 4 \frac{2K_1 K_2}{H^+(t)^3}} \frac{dCO_2(t)}{dt}$$
(6)

In the culture a matter transfer occurs between the gas and the liquid phases. However, this transfer is only modeled for the O_2 and CO_2 since the N_2 is not soluble in water. Hence, equations (7) and (8) show those matter transfers as a function of the liquid volume, $V_l(t)$; the equilibrium concentration of O_2 and CO_2 in the gas phase, $O_2^*(t)$ and $CO_2^*(t)$; and the volumetric mass transfer coefficients for O_2 and CO_2 , $K_{laO_2}(t)$ and $K_{laCO_2}(t)$. These last parameters

are calculated based on the volume ratios and the surface speed between the liquid and gas phases.

$$fO_{2,gl}(t) = V_l(t)K_{laO_2}(t)(O_2^*(t) - O_2(t)) \quad (7)$$

$$fCO_{2,gl}(t) = V_l(t)K_{laCO_2}(t)(CO_2^*(t) - CO_2(t)) \quad (8)$$

The concentrations at an equilibrium state, equations (9) and (10), are described in terms of the Henry constants for O_2 and CO_2 , H_{O_2} y H_{CO_2} ; the liquid pressure, P_t ; and the molar fractions of O_2 and CO_2 , $y_{O_2}(t)$ and $y_{CO_2}(t)$.

$$O_2^*(t) = H_{O_2}P_t y_{O_2}(t) \quad (9)$$

$$CO_2^*(t) = H_{CO_2}P_t y_{CO_2}(t) \quad (10)$$

To do that, three different mechanisms are considered for heat transfer process: convection, radiation and heat transfer produced by mass transfer. These mechanisms are described by equations (11), (12) and (13), where h is the thermal conductivity, S_{conv} is the contact surface, $T_1(t)$ and $T_2(t)$ are the temperatures of the contact elements, $Q_l(t)$ is the volumetric flow, C_v is the heat capacity, $I_c(t)$ is the solar radiation, S_{rad} is the area that receives that radiation and aR is the absorptivity of the solar radiation.

$$fQ_c(t) = hS_{conv}(T_1(t) - T_2(t)) \quad (11)$$

$$fQ_r(t) = \alpha I_c(t)S_{rad} aR \quad (12)$$

$$fQ_m(t) = Q_l(t)C_vT(t) \quad (13)$$

Besides the phenomena previously described, it will take place flow variations in the concentrations of both liquid and gas phases mainly due to the gas and liquid flows.

3. LIBRARY DESIGN

In order to optimize the design of the library, it has been necessary to take advantage of the object-oriented properties of the modeling language. The basics of this strategy can be summarized in the modular and hierarchical modeling principles.

The modular modeling is based on the idea of "divide and conquer". Hence, a system can be decomposed into different sub-systems and sub-models, for example, within the system modeled in this paper, two different sub-models could be a pipe section and a tank. The main advantages provided by the modularity are a greater ease for the development and validation of models, as well as the possibility of composition, reuse, and abstraction, that enables to each sub-model to be used without knowing their internal details. For the composition of the various sub-models, it is necessary to establish a communication between them through interfaces. More specifically, these interfaces will be designed according to the different domains described in Section 3.1.

On the other hand, hierarchical modeling allows us to build models and sub-models from low to high level, as it is shown in Section 3.2. Therefore from basic or atomic sub-models, it is possible to compose increasingly complex models up to build the complete system model. In our case, the atomic models are those that describe the physical/chemical phenomena which have been explained in Section 2.

The design of the library can be described from three different perspectives: the implemented domains, the classes composition scheme, and the packages architecture.

3.1. DOMAINS

Four domains have been considered: gas, *Gas*; liquid, *Liq*; heat, *Heat*; and liquid with biomass, *Bio*. Therefore, connectors, interfaces, the basic classes and some parts are described as a function of these domains. In addition, some domains can include others, that is, for example, to include inheritance relations

that make some domains variables be present in others. Thus, the *Heat* domain is included in the *Liq* domain and this is included in the *Bio* domain. The above is well represented in the designed connectors, whose pairs of potential and flow variables, *across* and *through* variables, are showed in Table 1.

Table 1: Variables of the connectors

Connector	<i>Across</i> variable	<i>Through</i> variable
<i>GasCon</i>	Gas volume - V_g [m^3]	Gas flow - fV_g [$m^3 s^{-1}$]
	O_2 concentration - O_{2g} [$mol m^{-3}$]	O_2 flow - fO_{2g} [$mol s^{-1}$]
	CO_2 concentration - CO_{2g} [$mol m^{-3}$]	CO_2 flow - fCO_{2g} [$mol s^{-1}$]
	N_2 concentration - N_{2g} [$mol m^{-3}$]	N_2 flow - fN_{2g} [$mol s^{-1}$]
<i>HeatCon</i>	Temperature - T [K]	Heat flow - fQ [$J s^{-1}$]
<i>LiqCon</i>		
<i>BioCon</i>		
<i>LiqCon</i>	Liquid volume - V_l [m^3]	Liquid flow - fV_l [$m^3 s^{-1}$]
<i>BioCon</i>		
<i>BioCon</i>	Biomass concentration - C_b [$Kg m^{-3}$]	Biomass flow - fCb [$Kg s^{-1}$]
	O_2 concentration - O_2 [$mol m^{-3}$]	O_2 flow - fO_2 [$mol s^{-1}$]
	CO_2 concentration - CO_2 [$mol m^{-3}$]	CO_2 flow - fCO_2 [$mol s^{-1}$]
	C_T concentration - C_T [$mol m^{-3}$]	C_T flow - fC_T [$mol s^{-1}$]
	H^+ concentration - H^+ [$mol m^{-3}$]	H^+ flow - fH^+ [$mol s^{-1}$]

3.2. COMPOSITION SCHEME

The classes composition scheme, represented by Figure 1, is divided into four levels.

At level 0 there are the *connectors* and *interfaces* which have been created through the first ones. The interfaces describe connections to the outside of a class. Furthermore, at this level, it is also located the graphical representation, *icons*, and the equations that describe the dynamics of each class.

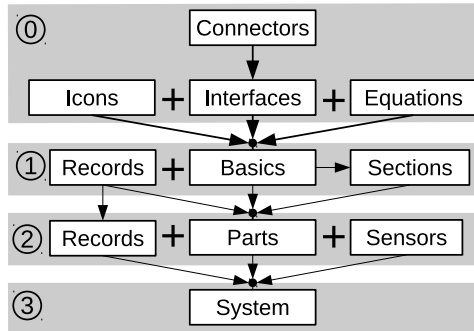


Figure 1: Composition scheme of classes

By means of a group of equations, an interface for the information transference plus an icon for its graphical description, each *basic* class is created. These types of classes describe the control volumes and the transport phenomena, which is reflected in their icons, as it can be observed in Figure 2. These classes are at level 1 with the *sections* class. *Sections* classes are compound classes that can not be considered as *parts*. The *register* classes are also at level 1, these are sets of parameters and constants.

In Figure 3 the composition of one tubular section is showed. In reference to its connection to the outside, this section consists of four connectors that are used to connect to other blocks and to form a complete tube, and two inputs which provide information about the outside temperature and the solar radiation. The red block makes the calculations of heat transfer with the environment. The green blocks estimate, from left to right, the amount or concentration of various elements in the aqueous medium, the generation of biomass by photosynthesis, and the transport from the aqueous medium to the connected components. Finally, the bicolor block calculates the mass transfer between the aqueous and gaseous medium.

For the composition of the *parts*, the basic classes, sections, and other parts are used, as it can be seen in Figure 4. Also, for the parameters configuration a set of registers be used, these can be compounded by another register of the

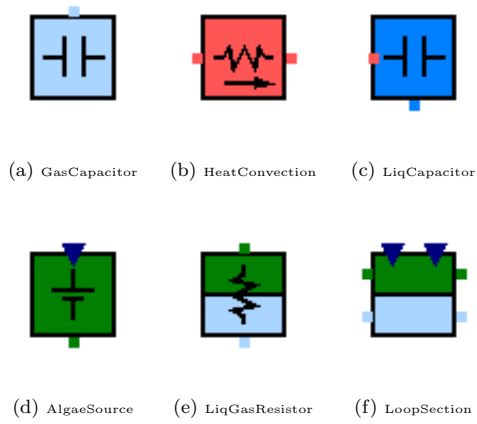


Figure 2: Some of *basic* classes

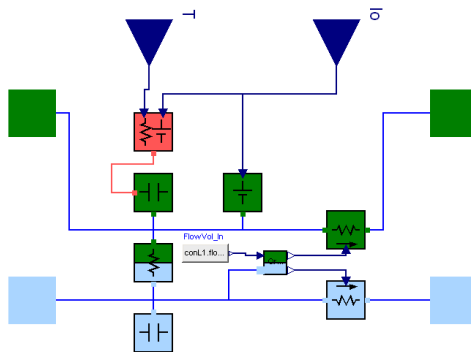


Figure 3: Composition of a loop section

previous level. All the above, together with the *sensors* class, from the level 2 of the composition scheme. In Figure 5 the graphical representation of some implemented parts is showed.

As in Figure 3, Figure 4 shows some connectors and data inputs. Furthermore, in this case, there is another type of blue connector that represents the connection of the aqueous medium that flows into the heat exchanger. In reference with the three internal blocks, the multicolor loop is a heat exchanger that is connected to the bicolor block that represents a tank with the aqueous and the gaseous medium, and, finally, the green block is responsible for calculating the overflow of the aqueous medium in the bubble column.

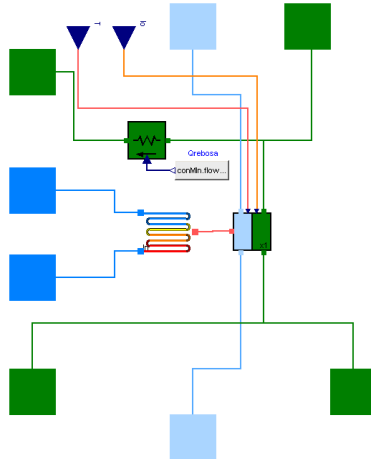


Figure 4: Composition of a bubble column with a heat exchanger

Finally, the level 3 of the composition scheme defines complex systems like a real photobioreactor. These systems are built using parts and sensors for the measure of potentials and flows, and are configured using registers. As an example of a complex system, in Figure 6 the composition of a complete closed photobioreactor is showed.

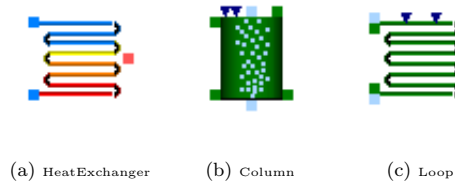


Figure 5: Some parts

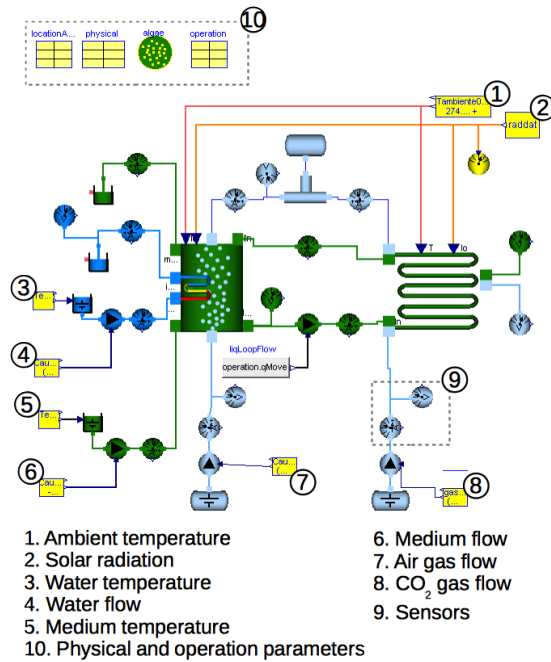


Figure 6: Composition of a complete closed photobioreactor

3.3. LIBRARY ARCHITECTURE

Following a summary of the content of the main packages is presented.

- *Basic*: This package contains the basic elements.
- *Examples*: This package contains some complete photobioreactor systems designed for testing the library.
- *Icons*: Graphical descriptions.
- *Interfaces*: Designed connectors and also the set of partial models that described the interfaces.
- *Parts*: Models similar to real physical components are in this package, i. e. a column, a loop, pumps, tanks, etc.
- *Records*: Classes of type *record* that contain constants, parameters and initial values for the basic components and parts.
- *Sensors*: Sensors are separated from *Parts* because, although are physically similar, they do not have the same behavior. Sensors have no dynamics and are only an interface to extract the internal data of the modeled systems and to communicate with the outside. They are useful for the log of data and for feeding the controllers.
- *Users Guide*: This package incorporates the most relevant information about the library and its use.
- *Utilities*: There is a set of tools to make easier the simulations, for example, tools for reading and processing data files.

4. RESULTS

In Figure 6 the composition of a closed photobioreactor is showed, this photobioreactor is similar to the one described in [12]. More specifically, it represents

a real system located in the *Experimental Station of "Las Palmerillas"*, Almería ($36^{\circ}47'36.5''N$ $2^{\circ}43'11.9''W$).

The plant, which has been mentioned above, is compounded by a bubble column, a heat exchanger, a loop, several pumps, gas injectors and sensors. The bubble column, in addition, to mix the culture and extract the oxygen by means of the injected air in its base, is the point where both the medium is added and the biomass is extracted. Inside the column is also located the heat exchanger that heats and cools the culture as a function of the liquid that flows inside it. The loop is connected to the input and output of the column, and it is composed of a winding pipe in which an injection of CO_2 feeds the photosynthesis process to result in the generation of biomass and O_2 . In Figure 7a a schematic representation of the modeled tubular photobioreactor is shown and in Figure 7b can be seen a capture of the real plant.

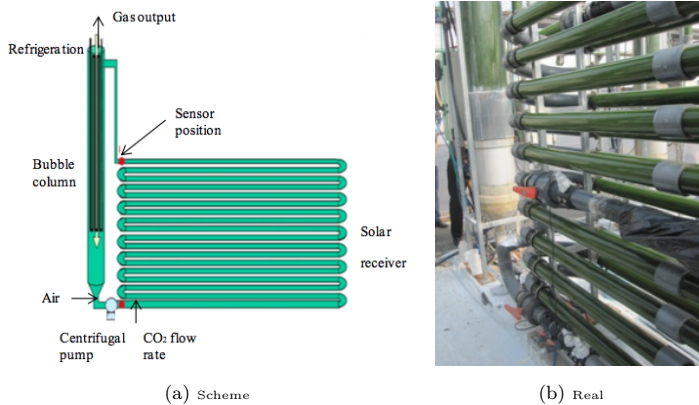


Figure 7: Schematic representation and physical facility

Figure 8 shows the solar radiation which affected to the photobioreactor the 4th February 2014, in Figure 9 the CO_2 flow injected to the loop during the experiment is depicted and, finally, Figure 10 shows some measured temperatures.

Through the use of the previously described inputs, some different variables have been measured in order to validate if the model built with the library

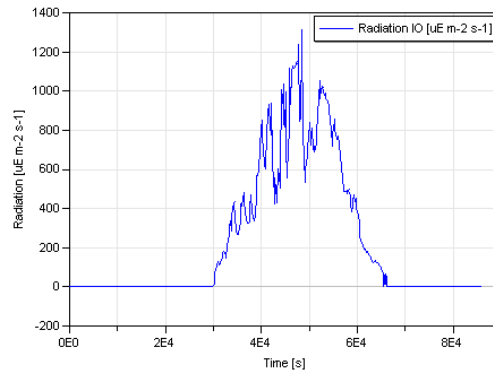
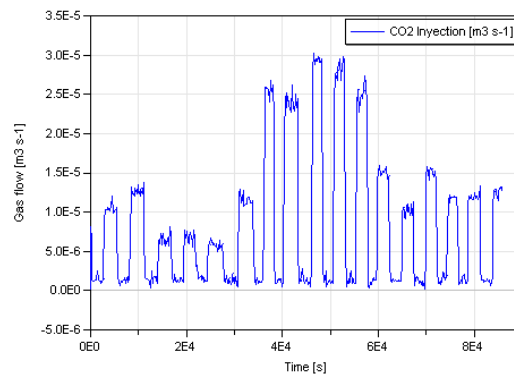


Figure 8: Solar radiation at 4th February 2014

Figure 9: CO_2 injection in the loop

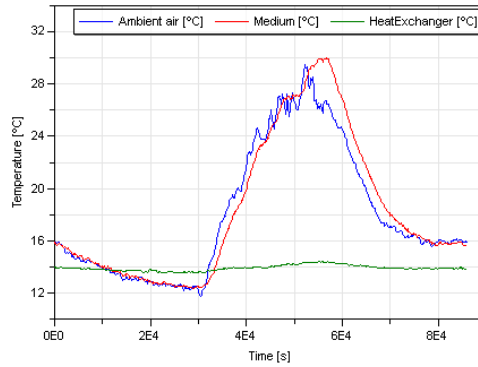
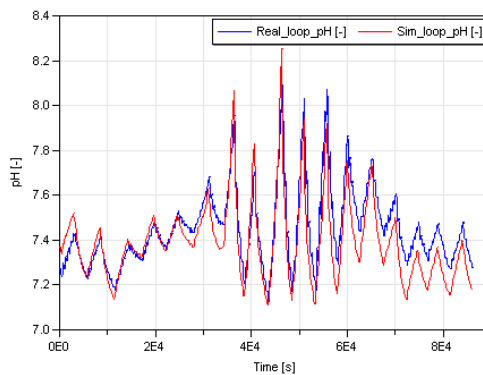


Figure 10: Photobioreactor temperatures

makes a correct calculation. More specifically, two of the validated variables are shown: pH at the output of the loop (Figure 11) and culture temperature at the input of the loop (Figure 12). Moreover, Figure 13 shows the evolution of biomass concentration at both the input and the output of the loop. Furthermore, as this value had not been measured when the experiment was performed, the proposed model provides an estimation that otherwise would be difficult to obtain.

Figure 11: Culture pH at the output of the loop

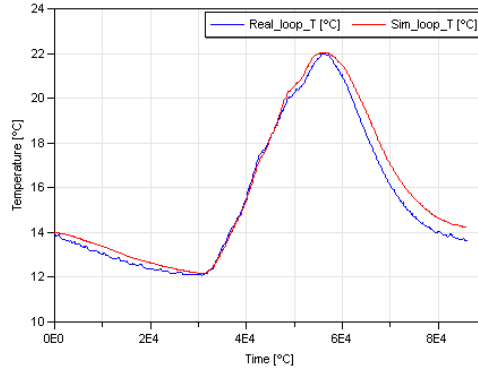


Figure 12: Culture temperature at the input of the loop

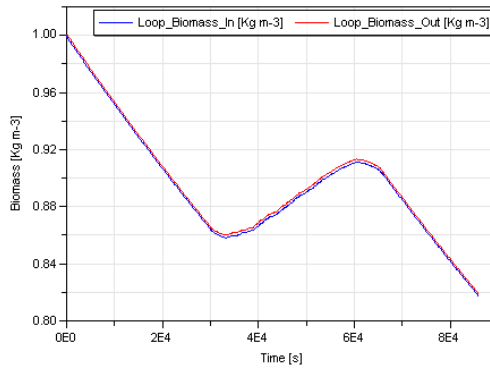


Figure 13: Biomass concentration at both the input and the output of the loop

Figures 11 and 12 show the library is able to simulate accurately the behavior of a real photobioreactor. Moreover, one of the great advantages of this library is the simulation time. Therefore, to illustrate the simulation time obtained with the proposed library, a study has been performed using four popular numerical solvers. Specifically, two of them (DASSL and LSODAR) are based on multi-step methods where DASSL is more conservative in relation to the step size control. On the contrary, the other two numerical solvers are based on

single-step Runge-Kutta methods, one implicit (RADAU IIA) and other explicit (DOPRI5). A more in detail description of the numerical solvers mentioned can be found in [22]. Hence, Table 2 displays the simulation times obtained for the example exposed in Section 3.2 and described by Figure 6 using the previously commented numerical solvers. All simulations have been performed on *Dymola* 6.1, over a *Windows XP* virtual machine, using only one processor. The virtual machine runs on *MacBook Pro* laptop with a first generation *i7* processor. As it can be observed in Table 2 the best simulation time has been reached using LSODAR numerical solver (for multi-step methods) and DOPRI5 (for single-step methods). Furthermore, apart from the differences among simulation times, there is not an appreciable variation among the results provided by the models using the previously mentioned numerical solvers and they have detected the same number of events. The simulation plots that are showed in this section use the *DOPRI5* algorithm with a 10^{-10} sensibility.

Table 2: Simulation times in function of algorithm and sensibility

Algorithm	Sensibility	Simulation time
<i>DASSL</i>	10^{-5}	125.00 <i>s</i>
	10^{-10}	300.00 <i>s</i>
<i>LSODAR</i>	10^{-5}	3.19 <i>s</i>
	10^{-10}	9.42 <i>s</i>
<i>DOPRI5</i>	10^{-5}	2.61 <i>s</i>
	10^{-10}	10.09 <i>s</i>
<i>Radau IIA</i>	10^{-5}	39.76 <i>s</i>
	10^{-10}	119.14 <i>s</i>

Additionally, the performance of the proposed model has been analysed by means of statistical analysis. More in detail, it is based on the following indexes: number of samples (N), absolute variation of the modelled variables (*range*), Mean Absolute and Relative errors (*MAE* and *MRE*, respectively),

maximum absolute error (AE_{max}), standard deviation (S_N), and Normalized Mean Absolute Error ($NMAE$). The mathematical formulation related to these indexes can be observed in (14)-(19). Table 3 shows the results provided by the statistical analysis.

$$range = |max(\mathbf{x}) - min(\mathbf{x})| \quad (14)$$

$$MAE = \frac{1}{N} \sum_{i=1}^N |\mathbf{x}(i) - \hat{\mathbf{x}}(i, \Psi)| \quad (15)$$

$$MRE = \frac{1}{N} \sum_{i=1}^N \frac{|\mathbf{x}(i) - \hat{\mathbf{x}}(i, \Psi)|}{\mathbf{x}(i)} \quad (16)$$

$$AE_{max} = \max(AE(\mathbf{x}, \hat{\mathbf{x}}(\Psi))) \quad (17)$$

$$S_N = \sqrt{\frac{1}{N} \sum_{i=1}^N (\hat{\mathbf{x}}(i, \Psi) - \bar{\hat{\mathbf{x}}}(\Psi))^2} \quad (18)$$

$$NMAE = \frac{MAE \times 100}{Rng} = \frac{\left(\frac{1}{N} \sum_{i=1}^N |\mathbf{x}(i) - \hat{\mathbf{x}}(i, \Psi)|\right) \times 100}{|max(\mathbf{x}) - min(\mathbf{x})|} \quad (19)$$

where x represents the real values of the modelled variables, \hat{x} and \bar{x} are the results provided by the model and the mean value of them, respectively.

Table 3: Statistical analysis of the results

	<i>pH</i>	<i>Temperature</i>
<i>MAE</i>	0.0931 [-]	0.4845 [K]
<i>MRE</i>	0.0124 [-]	0.0316 [K]
<i>AE_{max}</i>	0.4442 [-]	1.3962 [K]
<i>S_N</i>	0.1257 [-]	3.2068 [K]
<i>NMAE</i>	9.6240 [%]	4.9002 [%]

5. CONCLUSIONS AND FUTURE WORK LINES

In this paper, the *PhotoBioLib* library, designed for the modeling and simulation of photobioreactors has been presented. The main advantages of this library versus other options are:

- Models based on first principles. This type of models has some advantages regarding the use of models based on transfer functions, black box models or other strategies.
- High configuration potential. From the predefined parts and the basic classes, the construction of a prototype is simple.
- Easy to use. It can be used by any user without previous knowledge about programming, since it is only necessary to drag and connect the different components.
- Models have a great similarity to reality. You can build systems by adding parts and sensors as if it were a real system. On the one hand, it allows to simulate existent real systems and, on the other hand, to create new systems for prior evaluation before proceeding to a real implementation.
- Simulation speed. The use of *Modelica* language, and more specifically, *Dymola* environment, allows us to simulate models with thousands of equations and large simulation times under small sampling times, like the showed in Figure 6. This is possible mainly due to the preprocessing task that *Dymola* performs over the differential equations previously to translate them into *C* language, without the user knowledge.
- Easy to interconnect with other models or libraries. The models created with the *PhotoBioLib* library can be connected to other components already available in *Modelica*, as blocks for signal processing or controllers, or other libraries, for example, one that models the temperature inside a greenhouse to simulate the behavior of one photobioreactor.

As future works, improvements that can be included in the library are the following:

- To develop new components and examples for the simulation of other types of photobioreactors. Currently, the library is being increased for the inclusion of open raceway photobioreactors.
- To include models for the simulation of solar radiation and outdoor temperature from anywhere in the world at any desired date. In the next version of the library, it will be included a model for the simulation of the solar radiation based on the date, longitude, and latitude.
- To implement a controllers package. Especially a model predictive control package.
- The possibility of tuning parameters and validating models from data recorded at real systems more or less automatically.

Acknowledgements

The authors acknowledge the financial support of the Ministry of Economy and Competitiveness under DPI2014-55932-C2-2-R and DPI2011-27818- C02-02 research projects and the FPI scholarship framed in the latter. M. Castilla is a fellow of the Spanish “Juan de la Cierva - Formación” contract program.

References

- [1] O. Bernard, Hurdles and challenges for modelling and control of microalgae for co₂ mitigation and biofuel production, *Journal of Process Control* 21 (10) (2011) 1378–1389.
- [2] P. K. Campbell, T. Beer, D. Batten, Life cycle assessment of biodiesel production from microalgae in ponds, *Bioresource technology* 102 (1) (2011) 50–56.

- [3] J. J. Milledge, Commercial application of microalgae other than as biofuels: a brief review, *Reviews in Environmental Science and Bio/Technology* 10 (1) (2011) 31–41.
- [4] A. Richmond, *Handbook of microalgal culture: biotechnology and applied phycology*, John Wiley & Sons, 2008.
- [5] C. Ugwu, H. Aoyagi, H. Uchiyama, Photobioreactors for mass cultivation of algae, *Bioresource technology* 99 (10) (2008) 4021–4028.
- [6] A. P. Carvalho, L. A. Meireles, F. X. Malcata, Microalgal reactors: a review of enclosed system designs and performances, *Biotechnology progress* 22 (6) (2006) 1490–1506.
- [7] Š. Papáček, V. Štumbauer, D. Štys, K. Petera, C. Matonoha, Growth impact of hydrodynamic dispersion in a couette–taylor bioreactor, *Mathematical and Computer Modelling* 54 (7) (2011) 1791–1795.
- [8] R. Davis, A. Aden, P. T. Pienkos, Techno-economic analysis of autotrophic microalgae for fuel production, *Applied Energy* 88 (10) (2011) 3524–3531.
- [9] K. J. Sink, Design and construction of a lighting system to illuminate a photobioreactor, Ph.D. thesis, Ohio University (2011).
- [10] J. Miñon Martínez, Modelo de biomasa algal para la captura de dióxido de carbono y su desarrollo en un software de evaluación, Ph.D. thesis, Universidad de Valladolid (2012).
- [11] L. Wu, Z. Li, Y. Song, Hydrodynamic conditions in designed spiral photobioreactors, *Bioresource technology* 101 (1) (2010) 298–303.
- [12] I. Fernández, F. Acién, J. Fernández, J. Guzmán, J. Magán, M. Berenguel, Dynamic model of microalgal production in tubular photobioreactors, *Bioresource technology* 126 (2012) 172–181.

- [13] V. Goetz, F. Le Borgne, J. Pruvost, G. Plantard, J. Legrand, A generic temperature model for solar photobioreactors, *Chemical Engineering Journal* 175 (2011) 443–449.
- [14] P. Iancu, V. Pleşu, S. Velea, Flue gas co2 capture by microalgae in photobioreactor: a sustainable technology, *CHEMICAL ENGINEERING* 29.
- [15] B. Tamburic, F. W. Zemichael, P. Crudge, G. C. Maitland, K. Hellgardt, Design of a novel flat-plate photobioreactor system for green algal hydrogen production, *international journal of hydrogen energy* 36 (11) (2011) 6578–6591.
- [16] K. Cheenkachorn, N. Choosri, A. Chutrapukdeekul, T. Kangsadan, Computational modeling of microalgae culture using a helical photobioreactor, in: *IASME/WSEAS INTERNATIONAL CONFERENCE ON HEAT TRANSFER, THERMAL ENGINEERING AND ENVIRONMENT*, 9th, 2011, pp. 300–305.
- [17] Functional mock-up interface (fmi) for model exchange and co-simulation, <https://www.fmi-standard.org>, accessed: 2016-11-25.
- [18] K. Hongesombut, Y. Mitani, K. Tsuji, An incorporated use of genetic algorithm and a modelica library for simultaneous tuning of power system stabilizers, in: *Proceedings of the 2nd international Modelica Conference*, 2002, pp. 89–98.
- [19] J. Frenkel, C. Schubert, G. Kunze, P. Fritzson, M. Sjölund, A. Pop, Towards a benchmark suite for modelica compilers: Large models, in: *8th International Modelica Conference (Modelica'2011)*, Dresden, Germany, March 20-22, 2011, 2011.
- [20] A. Pérez-Castro, PhotoBioLib: una librería de Modelica para el modelado y simulación de fotobiorreactores, Master's thesis, UNED (2014).
- [21] T. Costache, F. G. A. Fernández, M. Morales, J. Fernández-Sevilla, I. Stamatina, E. Molina, Comprehensive model of microalgae photosynthesis rate

as a function of culture conditions in photobioreactors, *Applied microbiology and biotechnology* 97 (17) (2013) 7627–7637.

- [22] L. Liu, F. Felgner, G. Frey, Comparison of 4 numerical solvers for stiff and hybrid systems simulation, in: 2010 IEEE 15th Conference on Emerging Technologies Factory Automation (ETFA 2010), 2010, pp. 1–8.

