Environmental Modelling & Software 83 (2016) 143–154

Contents lists available at ScienceDirect

Environmental Modelling & Software

journal homepage: www.elsevier.com/locate/envsoft

MatSWMM – An open-source toolbox for designing real-time control of urban drainage systems

G. Riaño-Briceño [a, b, *], J. Barreiro-Gomez [a, c], A. Ramirez-Jaime [a], N. Quijano [a], C. Ocampo-Martinez [c]

[a] Departamento de Ingeniería Eléctrica y Electrónica, Universidad de los Andes, Carrera 1 No 18A-10, Bogotá, Colombia
[b] Departamento de Ingeniería Civil, Universidad de los Andes, Carrera 1 No 18A-10, Bogotá, Colombia
[c] Automatic Control Department, Universitat Politècnica de Catalunya, Institut de Robòtica i Informàtica Industrial (CSIC-UPC), Llorens i Artigas, 4-6, 08028 Barcelona, Spain

ARTICLE INFO

Article history:
Received 17 November 2015
Received in revised form
1 March 2016
Accepted 6 May 2016

Keywords:
Open-source toolbox
Urban drainage system integrated model
Real-time control
Matlab
Python
LabVIEW

ABSTRACT

This manuscript describes the MatSWMM toolbox, an open-source Matlab, Python, and LabVIEW-based software package for the analysis and design of real-time control (RTC) strategies in urban drainage systems (UDS). MatSWMM includes control-oriented models of UDS, and the storm water management model (SWMM) of the US Environmental Protection Agency (EPA), as well as systematic-system edition functionalities. Furthermore, MatSWMM is also provided with a population-dynamics-based controller for UDS with three of the fundamental dynamics, i.e., the Smith, projection, and replicator dynamics. The simulation algorithm, and a detailed description of the features of MatSWMM are presented in this manuscript in order to illustrate the capabilities that the tool has for educational and research purposes.
© 2016 Elsevier Ltd. All rights reserved.

Software availability

Name of software: MatSWMM.
Contact address: Gerardo Riaño-Briceño, School of Engineering, Universidad de los Andes, Carrera 1E 19A – 40; ga.riano949@uniandes.edu.co.
Year first available: 2015.
Software requirements: Windows x86 and x64 environments with Matlab 2008 (or higher), Python (2.5), and/or LabVIEW 2012 (or higher).
Program size: 15 Mb.
Availability: Open-source repository, licensed under the GNU General Public License v3.0 (https://github.com/water-systems/MatSWMM).

1. Introduction

Software packages for urban drainage systems (UDS) can be categorized into two classes of tools: commercial and open source. The former ones are popular among industry and utilities because of their all-in-one capabilities and their efficiency. Some of the commercial packages available in the market are CivilStorm, InfoWorks CS, MOUSE, MIKE, SewerCAD, SOBEK-Urban, MicroDrainage, and SIMBA (Price and Vojinović, 2011). Most of these are holistic (Butler and Schütze, 2005), i.e., these packages allow to model not only flow-routing through a drainage network, but also real-time control (RTC), rainfall, runoff, water quality, street flooding (1D and 2D), and many other features. Despite of the completeness of commercial packages, these tools can be inappropriate for educational and research purposes, since they are closed, i.e., they do not allow modifications over the source code or the addition of new algorithms, a feature that is crucial when testing new theories, specially new RTC strategies. Additionally, commercial tools can be expensive despite allowing integration with coding tools.

On the other hand, there are open-source tools aimed to give flexibility to the simulation algorithm, allowing the integration of

* Corresponding author. Departamento de Ingeniería Eléctrica y Electrónica, Universidad de los Andes, Carrera 1 No 18A-10, Bogotá, Colombia.
E-mail addresses: ga.riano949@uniandes.edu.co (G. Riaño-Briceño), j.barreiro135@uniandes.edu.co, jbarreiro@iri.upc.edu (J. Barreiro-Gomez), af.ramirez236@uniandes.edu.co (A. Ramirez-Jaime), nquijano@uniandes.edu.co (N. Quijano), cocampo@iri.upc.edu (C. Ocampo-Martinez).

http://dx.doi.org/10.1016/j.envsoft.2016.05.009
1364-8152/© 2016 Elsevier Ltd. All rights reserved.

several features related to UDS, like the ones stated before (García et al., 2015). Some tools of this type are CITY DRAIN © (Achleitner et al., 2007) and SWMM (Huber et al., 2010). CITY DRAIN © has been developed to operate with Simulink and Matlab in order to enhance its functionality with already developed control systems blocks and toolboxes for Matlab. CITY DRAIN © includes simplified flow-routing models like the Muskingum model (Te Chow, 1959). SWMM is the storm water management model of the US Environmental Protection Agency (EPA). It is oriented to model with high level of precision, using the one-dimensional Saint Venant equations (SVE), runoff processes that take place in UDS. Additionally, SWMM can be useful for rule-based RTC modeling. However, open-source tools can be also limited, since extending and/or modifying SWMM for RTC purposes requires programming skills with a low-level programming language such as C, and adding a more complex model to the CITY DRAIN © framework (like the SVE-based model) can be difficult in some cases. Additionally, Pathirana (Pathirana, 2015) developed a Python-based toolbox to extract easily results from SWMM, attempting to satisfy this need. However, it is not possible to design RTC with it, since that tool only simplifies the extraction of simulation results from SWMM.

For this reason, in order to facilitate the design and testing processes of real-time controllers for UDS, MatSWMM, an open source tool has been created. MatSWMM is a flexible tool, i.e., a software package that gives the user the possibility to manipulate the simulation results easily for data analysis and/or system edition functionalities, since it has been structured for three high-level programming languages (i.e., Matlab, Python, and LabVIEW), guaranteeing the possibility of implementing easily interfaces, and physical applications, taking advantage of matrix-oriented programming, plotting capabilities, optimization and control toolboxes. The toolbox works as a co-simulation engine, which is based on SWMM and it has been developed as a collection of functions in order to facilitate the expansion of the framework.

An important but often missed issue is that both Matlab and LabVIEW environments are commercial and closed products, thus their kernel and libraries cannot be neither modified nor freely distributed. To allow exchanging ideas effectively improving scientific research, both the toolbox and the platform on which MatSWMM runs should be free (Stallman, 2002). At this aim, MatSWMM can run on Python, which is open source and has a great variety of optimization, data analysis, control and numerical analysis libraries, e.g., NumPy, SciPy and python-control. It is also possible to work with ArcPy, which is a GIS library that can be used to enhance the functionality of MatSWMM in the future, in order to offer the same geographical positioning capabilities of commercial packages such as InfoWorks CS or MIKE.

In this manuscript, a detailed description of the toolbox functionalities is presented and some applications for it are suggested, emphasizing on the RTC design and modeling applications. The remainder of the manuscript is organized as follows. Section 2 introduces some preliminaries related to the simulation algorithm and the MatSWMM environment. Section 3 gives a detailed description of all the functionalities of MatSWMM, which is complemented in Section 4 with the description of the population-dynamics-based controller for UDS that is included into the toolbox. Section 5 gives a brief description of the models that are adapted to MatSWMM in order to implement model-based controllers. Section 6 presents an application example and some realizable applications with the toolbox. Finally, in Section 7, some conclusions are drawn and a discussion is made about further applications of the co-simulation by using this novel tool.

2. Preliminaries

In this section, important aspects for hydraulics computation and RTC with MatSWMM are presented. First, a brief explanation of the SWMM model is given, highlighting the modifications that are done to the SWMM flow-routing algorithm, and how several enhanced functionalities have been adapted to it. Then, it is emphasized how RTC can be designed with MatSWMM, and how it can be scalable for large-scale systems. It is important to consider that MatSWMM can be only executed through command-line instructions when using Matlab and Python. In contrast, if LabVIEW is preferred, functions are called like typical LabVIEW blocks with their corresponding inputs and outputs.

2.1. The SWMM model

SWMM is a dynamic rainfall-runoff simulation model that is used for planning, analysis, and design of infrastructure related to stormwater runoff, combined and sanitary sewers, and other drainage systems in urban areas (Huber et al., 2010). The platform consists of an interface where a drainage network can be created by using elements such as pipes, canals, storage units, subcatchments, among others. Additionally, it has a calculation module that uses the one-dimensional SVE to simulate runoff throughout the network (Te Chow, 1959).

SWMM has been also equipped with dynamic and static control elements. Static elements such as weirs and outlets, whose purpose is to limit water levels along the system, work as barriers that reduce the energy of the flow. Dynamic control elements, such as orifices and gates, can be programmed to operate with either logical or rule-based control. Additionally, local PID controllers can be implemented to set operation points for valves and gates settings in order to manage water levels, cumulative volumes, and flow directions.

The RTC default functionalities of SWMM have become limited for large-scale control of UDS, as the ones developed in (Barreiro-Gomez et al., 2015) (Martin, 2012), and (Ocampo-Martinez, 2010). However, it has a big potential as an open-source software since it is possible to enhance its functionality in order to include new methods that allow users to simulate complex RTC strategies, and to edit systematically any UDS implemented on it.

2.2. Structure of the MatSWMM toolbox

A special structure of folders and work-files, presented in Fig. 1, has been designed for MatSWMM in order to maintain an organized environment while working with the toolbox. The structure of MatSWMM can be categorized in three main parts: the SWMM files, the MatSWMM files, and the simulation results. The SWMM files (i.e., the input, report, and output files), are stored in a single folder called “swmm_files”. In order to run a simulation, it is only required to store the input file created with SWMM, and the path to the input file must be described through code.

The simulation results are stored in four different folders that are related to the simulation time and the three main types of elements in UDS (i.e., links, nodes, and subcatchments). The results of the simulation are stored in “.csv” files that contain information of different attributes depending on the type of object as exposed in Fig. 1.

Moreover, the MatSWMM files that are required to compute a simulation vary depending on the programming language. Since Matlab and Python can be used for object-oriented programming (OOP), the MatSWMM environment has been conceived as a class file with several methods. In contrast, MatSWMM for LabVIEW is composed by a set of “.vi” files related to the functionalities of the

Download English Version:

https://daneshyari.com/en/article/6962406

Download Persian Version:

https://daneshyari.com/article/6962406

Daneshyari.com