Wearable Mechatronics-Enable Control Software Framework

Open-Source Project Details

The Wearable Biomechatronics Laboratory, Western University

1. Framework Motivation

The Wearable Mechatronics Control Software (WearMECS) framework is a software framework that has been developed to support the creation of control software for wearable mechatronic devices that assist humans with motion. More specifically, the framework is a software design and implementation tool. The current scope of the framework is targeted at the development of control systems for motion tasks between humans and wearable mechatronic devices. Upon examining the literature, it was clear that a control software framework that was specifically designed for this domain did not exist.

In 2014, Dr. Tyler Desplenter, with the assistance of Dr. Ana Luisa Trejos, began to construct the framework from the control system literature and the control systems that were developed to control a wearable mechatronic elbow brace [1]. The purpose of the WearMECS framework was to provide a platform that enables more efficient control system development and comparison between variations of control system components. This framework was the basis of the control system designs and implementations that were developed as part of the Dr. Desplenter's doctoral research [2].

In order to assist other control system developers, Dr. Desplenter and Dr. Trejos decided to release the WearMECS framework as an open-source software project. This project includes a series of Visual Studio applications containing the framework and examples that were developed using the framework. The examples included in this project were developed for control of devices during motion tasks that fall within the realm of musculoskeletal rehabilitation. However, the framework is not limited to this domain. Developers should adopt and modify the framework as needed to fit their applications.

The remainder of this document is structured as follows: Section 2 contains a brief description of the framework and Section 3 provides descriptions of the Visual Studio projects that are included within the open-source repository.

2. Framework Component Definition

The framework has been designed on the concept that motion-assistance control systems are all required to implement similar high-level functionality. Based on examination of the research literature, it was found that there are three major areas of functionality that these control systems must implement: task control, estimation control, and actuation control [2] (see Table 1 for definitions). The framework is built upon this commonality, which can be seen in the class diagram (Fig. 1). Each of the three control functionality groups are represented by a single class. The classes are implemented as interfaces to provide templates, while allowing developers to customize their own implementation of each of the provided functions.

Until the creation of this framework, the ability to classify and compare control functionality was difficult. This is due to the variance between the views and definitions of each developer, along with no standard to follow. Since control systems are commonly arranged in hierarchies, control functionality is often classified as a series of two or more levels. However, the number and arrangement of levels are chosen arbitrarily by each developer. In order to accommodate this variance, the framework does not provide any relationships between the base classes. This enables the developers to implement multiple versions of each interface and arrange their relationships to meet the unique requirements of their control system project.

Table 1: Brief descriptions of the three control functionality groups that form the foundation of the WearMECS framework.

Control Functionality Group	Description
Task control	The set of functions responsible for orchestrating the execution of the desired motion tasks. The focus of this functionality is to control the interactions of system objects in order to execute the motion task within the expected timing schedule.
Estimation control	The set of functions responsible for manipulating and estimating system parameters. The main focus of this functionality is to provide estimates of parameters that are crucial to the function of the rest of the system, such as parameters used in the production of motion.
Actuation control	The set of functions responsible for performing the desired motion of the actuation systems. The focus of this functionality is to produce physical interactions between the device, the user, and the environment.

The framework includes three classes, which are intended to centralize different portions of the control system behaviour. If the developer centralizes each of the control functionality groups within each of these base classes, it is easy to swap controllers without restructuring a portion or all of the control architecture. This makes it easier to focus modification efforts or perform comparisons across multiple variations of a particular controller. However, the developer is not constrained to this centralized structure and may spread functionality across multiple base classes or additional classes. This formulation of the framework was chosen to maximize creative freedom, while providing templates for common behaviours.

In summary, the WearMECS framework provides many benefits for control system developers, including common behaviour templates, non-restricted class arrangement, and more efficient comparison and evolution. In addition, the framework can be integrated into a larger number of existing control solutions due to the lack of definition of class relationships. This becomes especially important when developers want to compare existing solutions. The framework has also been designed and used as both a software design and implementation tool. Examples of using the framework as a software design tool can be found in [2–4], while the implementation examples are included within this open-source software project. The source code for the framework has also been provided as part of this project. This allows customization and evolution of the framework to include new behaviours that are common to motion-assistance control systems.

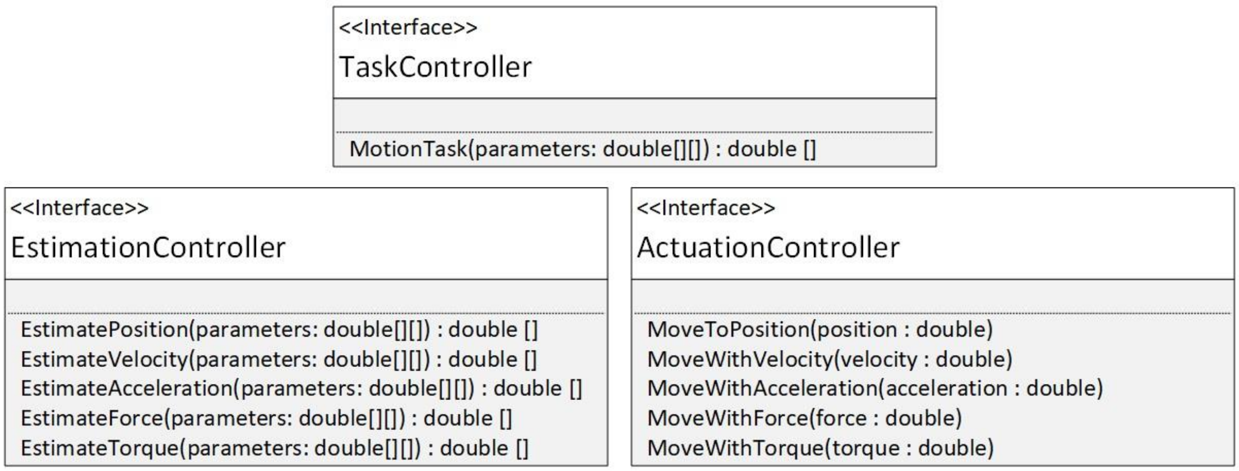

Fig. 1: Class diagram of the WearMECS framework consisting of three interfaces. Each interface represents one of three control functionality groups.

3. Source Code

The open-source repository contains source code that either defines the framework or uses the framework to implement a control system for a wearable mechatronic device. The repository contains six projects, all of which were developed using Visual Studio 2015. The WearMECSF_CSharp and WearMECSF_CPlusPlus projects contain definitions of the WearMECS framework. The WearMECSF_Modified contains an example of appending the framework with additional classes. The ActiveAGearControlSystem, WearMEBraceControlSystem, and WearMEBenchTopControlSystem projects contain examples of control systems built upon the WearMECS framework. The source code contained in these projects was written exclusively by Dr. Desplenter, with the exception of some portions of the EposController.cpp file which was contributed by Abelardo Escoto.

WearMECSF_CSharp

The WearMECSF_CSharp project contains a definition of the WearMECS framework in the C# language. This project contains the three base classes of the framework and was used in the development of control software for the WearME elbow brace, such as the WearMEBenchTopControlSystem project.

WearMECSF_CPlusPlus

The WearMECSF_CPlusPlus project contains a definition of the WearMECS framework in the C++ language. This project contains the three base classes of the framework and was used in the development of control software for the WearME elbow brace [1], such as the WearMEBraceControlSystem and ActiveAGearControlSystem projects.

WearMECSF_Modified

The WearMECSF_Modified project is an example of extending the WearMECS framework and is based on the WearMECSF_CSharp project definition of the framework. This project contains classes to represent human bodies and muscles. Controlling wearable mechatronic devices involves interacting with the human body and muscles, making these classes a natural extension of the framework. Furthermore, many of the existing control system models for estimating human motion require information about the human body and muscles in order to produce more accurate estimates. This version of the framework was used for control software that implemented these types of biological models.

WearMEBenchTopControlSystem

The WearMEBenchTopControlSystem project is a C# control software application built upon the WearMECS framework, which was designed for the comparison of different elbow motion estimation models. Each of these models is implemented as an EstimationController framework object. The project contains a graphical user interface prototype with separate panels for connecting to Trigno Wireless Lab (Delsys, Massachusetts, U.S.A.), EPOS 24/5 Position Controller (Maxon Motor AG, Switzerland), and Collection Arm (see [2] for details) devices. Through this interface, the user is able to control the WearME elbow brace, via the EPOS 24/5 Position Controller, based on the EMG and elbow position signals collected through the Trigno Wireless Lab and Collection Arm, respectively. This control software also records data representing both the user's motion signals and the resultant motion of the brace. The WearMEBenchTopControlSystem project also contains a biological elbow motion model. This model was not used during the experiment that involved the WearMEBenchTopControlSystem project, but is included as an example.

WearMEBraceControlSystem

The WearMEBraceControlSystem project is a control software application built upon the WearMECS framework for controlling the WearME elbow brace and written in C++. This software was designed to control the position of the WearME elbow brace in an offline remote-controlled manner, meaning that the user's data were recorded and used at a later time to control the device, which was located remotely from the user. As a result, the project contains functionality to parse data files that contain the user's motion signals. The software controls the device through a library that interfaces with an EPOS 24/5 Position Controller, which is connected to the motor of the WearME elbow brace.

ActiveAGearControlSystem

The ActiveAGearControlSystem project is a control software application built upon the WearMECS framework for controlling the Active A-Gear arm support [4] and written in C++. This control software is a modification of the WearMEBraceControlSystem project. There are two main differences between these projects. First, the file parser is modified to account for variations in the data collection tools that were used to control the Active A-Gear. Second, this software controls the velocity of the Active A-Gear's elbow joint, as opposed to controlling the position of the elbow joint. Interaction between the software and the device is achieved through a library that interfaces with an EPOS2 24/2 Position Controller (Maxon Motor AG Switzerland), which is connected to the elbow joint motor of the Active A-Gear arm support.

Acknowledgements

We would like to thank Dr. Ken McIsaac and Dr. Ilia Polushin for their help with the formulation of the framework and Anastasiia Kyrylova, Myles Lidka, Abelardo Escoto, Dr. Joan Lobo-Prat, Dr. Arno Stienen, and Yue Zhou for their help in the testing of the framework, which was essential to demonstrate the efficacy of the framework as a control software design and implementation tool. We would also like to Dr. Michael Naish for providing some of the equipment used in the experiments involving the framework and Shrikant Chinchalkar for providing insights into patient–therapist interactions during rehabilitation.

References

[1] T. Desplenter, A. Kyrylova, T.K. Stanbury, S. Chinchalkar, A. Escoto and A.L. Trejos. "A Wearable Mechatronic Brace for Arm Rehabilitation," in Proceedings of *IEEE International Conference on Biomedical Robotics and Biomechatronics*, Sao Paulo, Brazil, 2014, pp. 491-496.

[2] T. Desplenter. "Development of Digital Control Systems for Wearable Mechatronic Devices: Applications in Musculoskeletal Rehabilitation of the Upper Limb." PhD thesis, *Electronic Thesis and Dissertation Repository*, University of Western Ontario, 2018.

[3] T. Desplenter, J. Lobo-Prat, A.H.A. Stienen and A.L. Trejos. "Extension of the WearME Framework for EMG-driven Control of a Wearable Assistive Exoskeleton." *IEEE International Conference on Advanced Intelligent Mechatronics*, Banff, Canada, 2016, pp. 288–293.

[4] T. Desplenter, A. Kyrylova and A.L. Trejos. "Development of an EMG-Driven Control System for a Wearable Mechatronic Elbow Brace." in *Proceedings of IEEE International Conference on Biomedical and Health Informatics,* Las Vegas, USA, 2016, pp. 501–504.