

**INFERENCIA Y VERIFICACIÓN DE PROPIEDADES EN
SISTEMAS CIBER-FÍSICOS ESTOCÁSTICOS**

**TÍTULO EN INGLÉS: INFERENCE AND VERIFICATION
OF PROPERTIES IN STOCHASTIC CYBER-PHYSICAL
SYSTEMS**

DIRECTOR: JOSÉ IGNACIO REQUENO

AUTORES: JAVIER ROMERO FLORES (GIC) & DMYTRO VERNYUK (GII)



TRABAJO DE FIN DE GRADO DEL GRADO EN INGENIERÍA DE COMPUTADORES E
INGENIERÍA INFORMÁTICA

FACULTAD DE INFORMÁTICA

UNIVERSIDAD COMPLUTENSE DE MADRID

Curso 2021-2022

I ain't no physicist but I know what matters.

– Popeye el Marino

Resumen (español)

Blablabla

Palabras Clave: Blablabla

IV

Abstract (English)

Blablabla

keywords: Blablabla

Prefacio

Prefacio.

Índice general

1. Introducción	1
1.1. Sistemas ciberfísicos	1
1.2. Objetivos	2
1.3. Organización del documento	2
2. Nuevos operadores para STL	3
2.1. Signal Temporal Logic	3
2.2. Operador integral	3
2.3. Operador derivada	3
2.4. Integración con ParetoLib	3
3. Interfaz gráfica	5
3.1. Librerías utilizadas	5
3.2. Otros	5
4. Conclusiones	7
4.1. Conclusiones (español)	7
4.2. Conclusions (English)	7
Bibliografía	8

Capítulo 1

Introducción

1.1. Sistemas ciberfísicos

Los sistemas ciberfísicos son un tipo especial de sistemas en tiempo real que combinan un micro controlador o programa software, representado mediante una máquina de estados discretos, con uno o varios sensores que interactúan sobre una variable física. Ejemplos de este tipo de entornos son los ordenadores de a bordo que regulan la velocidad de crucero de un vehículo, o el piloto automático que controla la altitud y trayectoria de un avión.

Garantizar el correcto funcionamiento de estas plataformas es crucial, ya que un mal funcionamiento conlleva una reducción del confort por parte de los usuarios o, incluso, la pérdida de vidas humanas. Una manera de asegurarlo es mediante el uso de técnicas de verificación formal. Las técnicas de verificación en tiempo de ejecución Falcone y col. 2021 (*runtime verification* en inglés) implementan un monitor que supervisa que la ejecución actual cumple con los requisitos especificados. Los monitores actúan como guardianes, alertando al usuario cuando la ejecución se desvía del comportamiento deseado e, incluso, implementando contramedidas para revertir ese hecho y redirigir el sistema hacia un estado saludable.

Existen diversos formalismos para definir los comportamientos deseados y no deseados. Los requisitos funcionales se pueden expresar operacionalmente, mediante algún tipo de autómatas de estados finitos, o declarativamente, mediante una descripción lógico-matemática. Dentro de esta segunda categoría, la lógica temporal es un tipo de lógica modal que expresa propiedades sobre un *estado* en particular del sistema, o sobre los *caminos* (es decir, secuencia de estados) que atraviesa. En este trabajo utilizaremos Signal Temporal Logic Maler y Nickovic 2004, un tipo de lógica temporal enfocada al análisis de señales analógicas.

1.2. Objetivos

Los objetivos de este proyecto son:

- Extender las capacidades de la lógica temporal para expresar propiedades que involucren *tendencias* (derivadas), o *acumulaciones* (integrales),
- Implementar esos nuevos operadores lógicos en las herramientas software actuales, y
- Proporcionar una interfaz de usuario amigable que facilite la interacción con dichas herramientas.

1.3. Organización del documento

El documento está dividido en X capítulos. En el capítulo 2 detallaremos la semántica e implementación de los nuevos operadores lógicos. El capítulo 3 está dedicado a la nueva interfaz gráfica de usuario. Continuamos con las conclusiones más relevantes del proyecto en el capítulo 4. Finalmente

Capítulo 2

Nuevos operadores para STL

2.1. Signal Temporal Logic

Definición, sintaxis y semántica de los operadores.

2.2. Operador integral

2.3. Operador derivada

2.4. Integración con ParetoLib

Capítulo 3

Interfaz gráfica

3.1. Librerías utilizadas

Primera sección.

3.2. Otros

Capítulo 4

Conclusiones

Para acabar este trabajo presentamos algunas conclusiones sobre los temas tratados, tanto teóricos como prácticos, y discutimos algunas ideas propicias para la mejora y continuidad del proyecto.

4.1. Conclusiones (español)

En vista a los resultados del capítulo anterior ...

4.2. Conclusions (English)

In view of the results of the previous chapter, ...

Bibliografía

- Bakhirkin, Alexey y Nicolas Basset (2019). «Specification and Efficient Monitoring Beyond STL». En: *Tools and Algorithms for the Construction and Analysis of Systems - 25th International Conference, TACAS 2019, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2019, Prague, Czech Republic, April 6-11, 2019, Proceedings, Part II*. Ed. por Tomás Vojnar y Lijun Zhang. Vol. 11428. Lecture Notes in Computer Science. Springer, págs. 79-97. DOI: 10.1007/978-3-030-17465-1_5. URL: https://doi.org/10.1007/978-3-030-17465-1_5.
- Bakhirkin, Alexey, Akshay Mambakam y José-Ignacio Requeno Jarabo (2022). *StlEval*. VERIMAG Git Repository. Ver. 2.X. URL: <https://gricad-gitlab.univ-grenoble-alpes.fr/verimag/tempo/StlEval>.
- Bakhirkin, Alexey y col. (2019). «ParetoLib: A Python Library for Parameter Synthesis». En: *Formal Modeling and Analysis of Timed Systems - 17th International Conference, FORMATS 2019, Amsterdam, The Netherlands, August 27-29, 2019, Proceedings*. Ed. por Étienne André y Mariëlle Stoelinga. Vol. 11750. Lecture Notes in Computer Science. Springer, págs. 114-120. DOI: 10.1007/978-3-030-29662-9_7. URL: https://doi.org/10.1007/978-3-030-29662-9_7.
- Basset, Nicolas y col. (2020). «Learning Specifications for Labelled Patterns». En: *Formal Modeling and Analysis of Timed Systems - 18th International Conference, FORMATS 2020, Vienna, Austria, September 1-3, 2020, Proceedings*. Ed. por Nathalie Bertrand y Nils Jansen. Vol. 12288. Lecture Notes in Computer Science. Springer, págs. 76-93. DOI: 10.1007/978-3-030-57628-8_5. URL: https://doi.org/10.1007/978-3-030-57628-8_5.
- Buyukkocak, Ali Tevfik, Derya Aksaray y Yasin Yazicioglu (2021). «Control Synthesis using Signal Temporal Logic Specifications with Integral and Derivative Predicates». En: *2021 American Control Conference, ACC 2021, New Orleans, LA, USA, May 25-28, 2021*. IEEE, págs. 4873-4878. DOI: 10.23919/ACC50511.2021.9482651. URL: <https://doi.org/10.23919/ACC50511.2021.9482651>.

- Falcone, Yliès y col. (2021). «A taxonomy for classifying runtime verification tools». En: *Int. J. Softw. Tools Technol. Transf.* 23.2, págs. 255-284. DOI: 10.1007/s10009-021-00609-z. URL: <https://doi.org/10.1007/s10009-021-00609-z>.
- Maler, Oded y Dejan Nickovic (2004). «Monitoring temporal properties of continuous signals». En: *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*. Springer, págs. 152-166.
- Mambakam, Akshay y José-Ignacio Requeno Jarabo (2022). *ParetoLib*. VERIMAG Git Repository. Ver. 2.X. URL: https://gricad-gitlab.univ-grenoble-alpes.fr/verimag/tempo/multidimensional_search.
- Nickovic, Dejan y col. (2020). «AMT 2.0: qualitative and quantitative trace analysis with extended signal temporal logic». En: *Int. J. Softw. Tools Technol. Transf.* 22.6, págs. 741-758. DOI: 10.1007/s10009-020-00582-z. URL: <https://doi.org/10.1007/s10009-020-00582-z>.