ISSN: 2007-5448

Volúmen 7 No. 1

RECIBE

Revista electrónica

DE COMPUTACIÓN, INFORMÁTICA, BIOMÉDICA Y ELECTRÓNICA

PUBLICACIONES
Vol.7 No.1 | May 2018 - Oct 2018

Índice

Computación e Informática

Electrónica

Recibido 2 Dic 2017
Aceptado 19 Feb 2018
ReCIBE, Año 7 No. 1, Mayo 2018

Análisis de Estrategias de Gestión de Seguridad Informática con Base en la Metodología Open Source Security Testing Methodology Manual (OSSTMM) para la Intranet de una Institución de Educación Superior

Analysis of Strategies of Computer Security Management Based on the Open Source Security Testing Manual Methodology (OSSTMM) for the Intranet of a Higher Education Institution

Diego Sebastián GORDÓN REVELO[1]
dgordon@uees.edu.ec

Rubén PACHECO VILLAMAR[1]
rpachecov@uees.edu.ec

[1]Universidad de Especialidades Espíritu Santo, Guayaquil, Ecuador

Resumen: El presente estudio se enfocó en tomar como referencia la metodología OSSTMM para aplicar una auditoría de seguridad informática e identificar brechas de seguridad en una Institución de Educación Superior, utilizando como tipo de prueba el Hacking ético. Mediante una investigación de campo, se estableció la situación actual de políticas de la gestión de seguridad informática de la Institución de Educación Superior objeto de estudio, en donde los principales activos de información analizados fueron: el servidor con el sistema de gestión financiera y académica, los laboratorios de informática, salas de docentes y el área administrativa. Con base en la auditoría realizada, se encontró que la institución de educación superior no lleva un control adecuado de políticas de seguridad informática y aplicación de las mismas, obteniéndose como principal hallazgo los valores de evaluación de riesgo (Rav) equivalente al 72,15% de seguridad. En el análisis de seguridad informática llevado a cabo, se concluye que la porosidad y las limitaciones permiten evaluar el nivel de impacto y criticidad de las vulnerabilidades encontradas, las cuales pueden ser mitigadas aplicando estrategias de gestión de seguridad informática y conjuntamente con el aumento de controles de seguridad se puede mejorar la valoración del Rav a una ponderación del 77,00%; de esta manera, se garantiza la confiabilidad, integridad y disponibilidad de la información.

Palabras clave: OSSTMM, seguridad informática, estrategias de seguridad.

Abstract: The present study focused on taking as reference the OSSTMM methodology to apply an auditory of a computer security, and to identify security breaches in a Higher Education Institution, using as a type of test the ethical Hacking. Through a field investigation, it was established the current situation of policies of the computer security management of the Higher Education Institution which is the object of the study, where the main information assets analyzed were: the server with the financial and academic management system, computer labs, teaching rooms and the administrative area. Based on the audit that was done, it was found that the institution of higher superior doesn't carry an adequate control of information security, policies and their application, obtaining as main finding the values of risk assessment (Rav) equivalent to 72.15% of security. In the computer security analysis carried out, it is concluded that the porosity and limitations allow to evaluate the level of impact and criticality of the vulnerabilities found, which can be mitigated by applying computer security management strategies and in conjunction with increased controls the Rav's valuation can be improved to a weighting of 77.00%; in this way, the reliability, integrity and availability of the information is guaranteed.

Keywords: OSSTMM, Informatic security, Security strategies.

1. Introducción

Las tecnologías de información y comunicación (TICs) son un factor de vital importancia en la transformación de la nueva economía global y en los rápidos cambios que están tomando lugar en la sociedad (UNESCO, 2004). Esto ha provocado un crecimiento continuo del papel de la seguridad de la información, considerada, esta última, el activo más valioso de la era digital, y ha obligado a que las infraestructuras tecnológicas tengan que protegerse adecuadamente contra amenazas lógicas y físicas. (Balcerek, Frankowski, Kwiecién, Smutnicki, & Teodorczyk, 2012).

Toda organización es vulnerable a los ataques informáticos y más aún las Instituciones de Educación Superior que poseen información de personal administrativo, docentes y estudiantes. Según ISACA (2015) el número de incidentes de seguridad detectados ha aumentado en un 66%, año tras año, desde el 2009, a su vez, las pérdidas suman 42,8 mil millones de dólares en todo el mundo según se desprende de encuestas y estimaciones realizadas en el año 2014. Por otra parte, Cisco (2016) en su Informe Anual de Seguridad manifiesta que el 92% de las infraestructuras tecnológicas obsoletas y sin actualizaciones, ejecutan software con vulnerabilidades conocidas; es decir, con falencias, que con una correcta disciplina de gestión pudieron haber sido subsanadas.

Así mismo, ESET (2015) menciona que la explotación de las vulnerabilidades es uno de los incidentes de mayor ocurrencia en las empresas grandes y en promedio, una de cada cinco empresas sufrió uno de estos ataques en el 2014. Respecto a este panorama, Toth & Sznek (2014) mencionan, que la necesidad de protección ha impulsado el desarrollo de estándares, métodos, procedimientos y políticas cuyo propósito es obtener información confiable sobre el estado y nivel de preparación en materia de seguridad que se tienen en las organizaciones, con el objetivo final de implementar cambios y mejoras.
Los ataques relacionados con la seguridad informática, con el pasar del tiempo, se han ido ejecutando mediante técnicas más y más sofisticadas, para intentar explotar las vulnerabilidades presentes en cualquier arquitectura. Sobre esto, el Instituto Español de Estudios Estratégicos (2011) manifiesta que, una de las maneras más destacadas de ataques son los programas maliciosos insertados en un sistema operativo para ocultar procesos y archivos.

La mayoría de organizaciones del sector educativo no queda exenta de eventos relacionados con la seguridad informática. En la actualidad, estudiantes, docentes e investigadores requieren de las nuevas tecnologías de la información (TI) para enviar y compartir datos. Hoy en día las nuevas tecnologías evolucionan constantemente y los modelos de seguridad informática regulares que se aplican en las Instituciones de Educación Superior pueden quedar obsoletas rápidamente, por lo que es necesario realizar auditorías que permitan

evaluar el estado actual de su seguridad en las redes de datos. Según ESET (2015) el sector de la educación superior a nivel mundial ocupa el tercer lugar en incidentes de seguridad informática hallándose expuestas en un 60% a contaminación por malware.

En el Manual de Políticas de Seguridad Informática de la Institución de Educación Superior objeto de este estudio, se ha definido como una de las tareas prioritarias, el realizar proyectos encaminados a reforzar la seguridad de su infraestructura tecnológica, mejorando el manejo y almacenamiento de información que se transmite a través de las redes de comunicación, o que se mantiene en bases de datos; además la gran cantidad de información que se envía y recibe a través de la intranet necesita de verificaciones de los sistemas y controles de seguridad con el fin de obtener una correcta funcionalidad de la seguridad operacional (UNIANDES, 2013). Las características del escenario en el cual se desarrolló la investigación, se presentan en la figura 1.

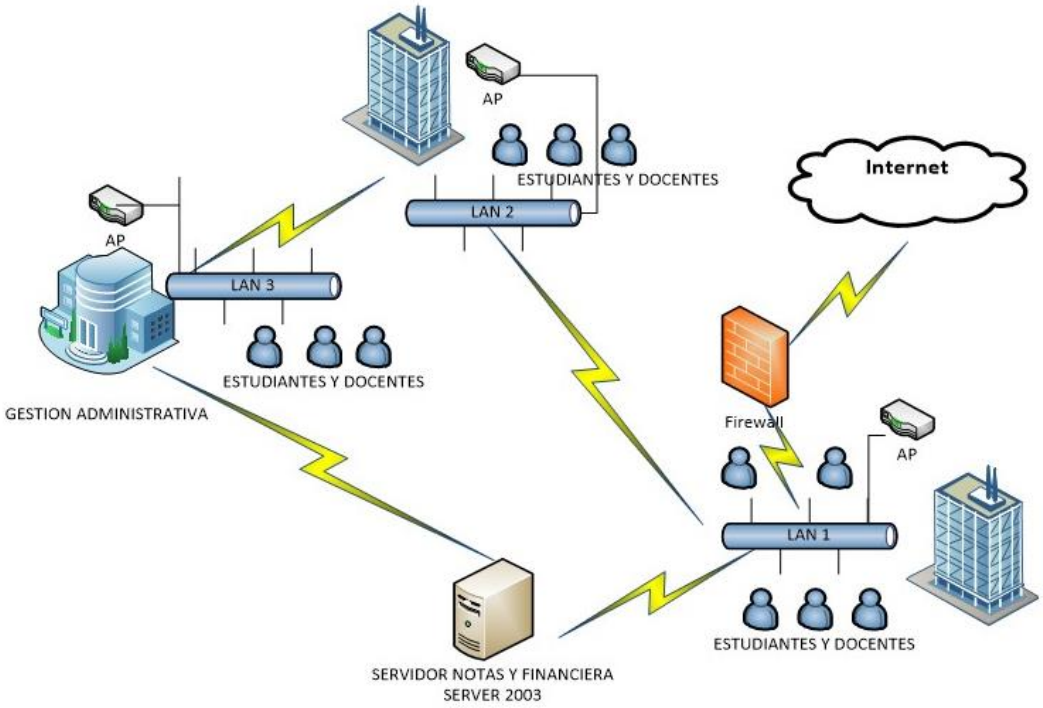

Figura 1. *Diagrama de red Intranet de la Institución de Educación Superior.*

La principal contribución de este trabajo, es la cuantificación de los riesgos y controles que presenta la intranet de la Institución de Educación Superior estudiada en base a la metodología OSSTMM v3, además se pondera el nivel de impacto y criticidad de las vulnerabilidades encontradas.

Coronel (2016) realizó un trabajo, relacionado con la aplicación del hacking ético para la detección de vulnerabilidades mediante herramientas de código abierto

(open source) en las aplicaciones web de una institución de educación superior del Ecuador, siendo el principal resultado el fortalecimiento de todo el escenario de seguridad en cuanto a la estructura de las aplicaciones, demostrado por medio de pruebas y análisis de una serie de herramientas de distribuciones Linux, como son Kali y herramientas de plataformas Windows con licencias libres.

2. Metodología

2.1. Tipo y Alcance de la Investigación

El tipo de investigación empleada en este artículo es de carácter cuantitativo, puesto que la metodología OSSTMM, presenta los resultados representados en métricas o RAV; además con esta metodología, se pretende cubrir la mayoría de los entornos que posee la Institución de Educación Superior objeto de estudio.

El alcance de la investigación es de tipo descriptivo, puesto que su propósito es especificar propiedades, características y rasgos importantes de la auditoría de seguridad informática realizada en una Institución de Educación Superior (Hernández Sampieri, Fernández & Baptista, 2010). En esta investigación se recopiló información para la cuantificación de los riesgos de la seguridad informática tomando en cuenta la metodología OSSTMM y la disciplina de hacking ético, con lo que se pretende conocer los riesgos de seguridad informática de la organización motivo de estudio. La población es una Institución de Educación Superior, y en la muestra se tomó datos de la gestión administrativa, docentes y estudiantes. Las técnicas de recolección de datos aplicados a este estudio fueron encuestas y entrevistas.

2.2. Fases de la Metodología OSSTMM

Con base en ISECOM (2012), la metodología OSSTMM, es un documento que reúne de forma estandarizada y ordenada diversas verificaciones y pruebas que se pueden realizar para una auditoría informática. Esta metodología presenta varias fases, en donde cada una de ellas se asocia con las fases del hacking ético como tipo de prueba que se aplica en este caso de estudio.

2.2.1. Fase de Inducción

El propósito de esta fase es la recolección de datos, tales como: cultura organizacional, reglas, normas y políticas, además permite establecer las limitaciones de la auditoría. Esta fase de la metodología OSSTMM se la aplica conjuntamente con la etapa de recolección de información del hacking ético, para lo cual:

- Se revisó el entorno de la Institución de Educación de Educación Superior objeto de estudio, conociendo la cultura organizacional y políticas de seguridad informática implantadas.
- Se analizaron detalles del canal humano, determinando los horarios en los que laboran o están activos el personal administrativo y los estudiantes.
- Se realizó un check list de verificación, en donde se averiguó la existencia de controles establecidos para mitigar ataques en contra de la seguridad informática.

2.2.2. Fase de Interacción

Esta fase es el núcleo de las pruebas de seguridad informática, en donde se determina el alcance de las interacciones de los activos de información y posibles brechas de seguridad, en esta fase se verifican los accesos a aplicaciones y sistemas y los controles de seguridad establecidos para los mismos.

- Se verificó la visibilidad de los posibles objetivos propensos a ataques de seguridad.
- Se analizaron los puntos de accesos que posee la Institución de Educación Superior; es decir, escaneo de los puertos abiertos.
- Se verificaron los controles que se aplican para garantizar la confidencialidad, integridad y disponibilidad de la información.

2.2.3. Fase de Investigación

En esta etapa se realizan diferentes actividades, tales como la verificación de procesos y exposiciones que puedan provocar algún tipo de interacción, se analiza la información que se descubre; es decir, se ponen a la luz los activos de información que se encuentran mal situados o mal administrados. Además, se recopila información disponible de manera abierta en buscadores utilizando técnicas como google hacking, o análisis de metadata, teniendo como objetivo la verificación de información relevante que estuviera sin ningún tipo de restricción en la red.

2.2.4. Fase de Intervención

En esta fase se determina la efectividad de los controles, el mapeo del impacto del mal uso de los mismos y se realiza una revisión de la auditoría realizada, donde se pretende conocer si la auditoria deja un rastro útil confiable.

- Se expone la seguridad operacional actual de la Institución con el cálculo de RAVs.

- Se define las estrategias para disminuir las limitaciones y se aumenta controles.
- En esta etapa de la metodología OSSTMM, se cuantifica los resultados obtenidos.

3. Análisis de Resultados

Una vez ejecutada la auditoría de seguridad informática, con base en la metodología OSSTMM en la intranet de una Institución de Educación Superior, se destacan los siguientes resultados:

3.1. Fase de Inducción – Recolección de información

3.1.1. Entorno Organizacional

En el área de Tecnologías de la Información de la Institución de Educación Superior estudiada, se establecen y aplican políticas de seguridad informática y de la información de carácter básico, tales como: listas de filtros de contenido en la intranet, firewall perimetral, repositorios externos y controles de acceso lógico.

- Los planes de continuidad no son definidos de manera eficiente puesto que no se disponen de políticas de respaldos tanto de la información como en la infraestructura de suministro y protección eléctrica.
- La seguridad lógica operacional de la organización no dispone de una protección adecuada apoyada con IPS,/IDS, Antivirus bajo licencia para la detección de posibles amenazas.

CHECKLIST DE VERIFICACIÓN DE SEGURIDAD INFORMÁTICA

SEGURIDAD DE LOS DATOS						
INDICADOR	ASPECTO A EVALUAR	CUMPLE		RIESGO		
		SI	NO	Bajo	Medio	Alto
1	La organización tiene definidas políticas de seguridad informática.	/		X		
2	Las políticas de seguridad informática son revisadas periódicamente.		/		X	
3	Se dispone de un inventario de activos tecnológicos.	/		X		
4	Se monitoriza y registra la actualización, instalación de software en equipos de producción.		/			X
5	Se tiene definido perfiles de usuarios para evitar la instalación de cualquier tipo de software en los pc de usuarios finales.		/		X	
6	Dispone implementado listas de control de acceso (ACL)		/	X		
7	Se tiene software antivirus licenciado instalado de cada uno de los computadores que cuenta la organización		/		X	
8	Se tiene instalado antimalware en los equipos de la organización	/		X		
9	Se dispone de repositorios externos para salvaguardar backups y datos relevantes	/		X		
10	Se monitoriza y registra la actividad de las líneas telefónicas.	/		X		
SEGURIDAD DE LA INFRAESTRUCTURA Y SERVICIOS						
INDICADOR	ASPECTO A EVALUAR	CUMPLE		RIESGO		
		SI	NO	Bajo	Medio	Alto
11	Se dispone de Firewall	/		X		
12	Se han definido y documentado perímetros de seguridad (DMZ) en la intranet para equipos con información de alto riesgo.	/		X		
13	Se dispone, implementado, un sistema de protección anti-DDOS		/			X

14	Dispone de redundancia de hardware		/			X
15	Dispone de redundancia de software		/			X
16	La organización cuenta con procesos para brindar mantenimiento preventivo al software	/		X		
17	La organización cuenta con procesos para brindar mantenimiento preventivo al hardware	/		X		
18	Dispone de contratos externos de soporte	/		X		
19	Dispone de UPS en cada estación de trabajo		/		X	
20	Las instalaciones eléctricas cuentan con bajada a tierra	/		X		
CONTROLES DE ACCESO						
INDICADOR	ASPECTO A EVALUAR	CUMPLE		RIESGO		
		SI	NO	Bajo	Medio	Alto
21	Se ha definido e implementado un proceso para la creación de usuarios y contraseñas.		/			X
22	Se ha definido un proceso de altas y bajas de usuarios.		/			X
23	Se dispone de controles de acceso lógico a los servicios críticos de T.I. que dispone la organización	/		X		
24	Se monitoriza y registra la actividad de accesos lógicos en los equipos críticos que dispone.		/			X
25	En los equipos de los usuarios finales dispone de dos cuentas de inicio de sesión una como administrador y otra como usuario normal		/			X
26	Se dispone de controles de acceso físico al data center de la organización		/			X
27	Se monitoriza y registra la actividad de accesos físicos al data center de la organización.		/			X
28	Se monitorea y autentica las conexiones a la red inalámbrica de la organización	/		X		
PLANES DE RESPALDO						
INDICADOR	ASPECTO A EVALUAR	CUMPLE		RIESGO		
		SI	NO	Bajo	Medio	Alto
29	Se tiene establecido políticas de backup en caso de desastres.		/			X
30	Se ha documentado e implantado un proceso para la gestión de incidentes de seguridad informática.		/			X
31	Se ha definido planes de continuidad y de respaldos de información crítica		/			X
32	Se tiene defino planes de continuidad de negocio en la organización		/			X
33	Dispone la organización de respaldos de energía eléctrica en caso de fallas		/			X
34	Dispone de cuartos de acometidas para los servicios provistos por proveedores externos		/		X	
HABITOS SEGUROS Y PREPARACIÓN						
INDICADOR	ASPECTO A EVALUAR	CUMPLE		RIESGO		
		SI	NO	Bajo	Medio	Alto
35	Cuenta con políticas de seguridad de los equipos respecto al consumo de alimentos y bebidas	/		X		
36	Cuenta con planes de capacitación al personal sobre seguridad informática.		/			X
37	Se dispone de un plan de manejo seguro de datos críticos		/			X
38	Se destruyen discos duros catalogados como dañados	/		X		
39	El personal de la empresa se conduce y aplica hábitos seguros de manejo de la información.	/			X	
40	En general, la actitud hacia la aplicación de normas de seguridad es positiva.	/			X	

Figura 2. *Check list de verificación de seguridad informática.*

De acuerdo a la información recopilada en esta fase, como se detalla en la figura 2, la Institución objeto de estudio aplica políticas básicas de seguridad informática; encontrando similitudes de resultados con un estudio reciente realizado por ESET (2017) en donde indica un 74% de las organizaciones en Latinoamérica, incluyendo el Ecuador, ha implementado la creación de políticas de seguridad aplicando controles como antivirus, firewall, controles de acceso entre otros. Con lo indicado anteriormente se hace evidente la necesidad de mejoramiento de los controles de seguridad informática que permitan gestionar de una mejor manera la seguridad de la Institución de Educación Superior estudiada.

3.2. Fase Interacción – Scanning y enumeración

RIESGO	Seguridad Física		Seguridad en el Espectro	Seguridad en las Comunicaciones	Total
	Humano	Físico	Wireless	Redes de datos	
POROSIDAD					
Visibilidad	2	4	3	15	**24**
Acceso	8	10	12	87	**117**
Confianza	1	0	0	4	**5**
Total Porosidad	**11**	**14**	**15**	**106**	**146**

Tabla 1. *Fase de Interacción*

En la tabla 1 se detallan todos los puntos interactivos encontrados en el momento de la evaluación de los canales: humano, físico, wireless, y redes de datos. Se encontró un total de 146 puntos interactivos de acceso y visibilidad, los cuales se evidencian en la tabla 2. Cabe indicar, que estos puntos pueden dar lugar, en algún momento, a un fallo de seguridad informática, y que solo se cuenta con 5 puntos interactivos de confianza. Los resultados anteriores pueden desbordar en una posible red botnet de la Institución de Educación Superior estudiada ya que según ESET (2017), en el Ecuador existe el 46,6% de las organizaciones que, en algún momento, fueron parte de una de estas redes maliciosas a causa de no implementar buenas prácticas de seguridad informática.

Por lo expuesto anteriormente se hace necesario analizar estrategias que permitan regularizar la seguridad lógica operacional de la organización objeto de estudio.

	IP	Tipo	Número	Servicio	Observación	Herramienta	
1	10.10.2.2	TCP	2082	Cpanel	Servicio desactualizado	nessus	GESTIÓN ADMINISTRATIVA
2		TCP	445	microsoft-ds	Puerto abierto	nmap	
3		TCP	135	msrcp	Puerto abierto	nmap	
4	10.10.2.8	TCP	137	netbios-ns	Puerto abierto	nmap	
5		TCP	139	netbios-ssn	Puerto abierto	nmap	
		TCP	1736	desconocido	Puerto abierto	nessus	
6		UDP	5535	DNS-LLMNR	Puerto abierto	nessus	
7	10.10.2.4	TCP	21	ftp	puerto abierto	nmap	
8		TCP	443	ssl	pueto abuerto	nmap	
9		TCP	3306	mysql	Puerto abierto	nessus	
10		TCP	80	http	puero abiero	nmap	
11	10.10.5.1	TCP	135	msrpc	puero abiero	nmap	
12		TCP	139	netbios.ssn	puero abiero	nmap	
13		TCP	445	microsoft-ds	puero abiero	nmap	
14		TCP	1433	ms-sql-s	puero abiero	nmap	
15		TCP	3389	ms-wbt-server	puero abiero	nmap	
16		TCP	49152	desconocido	puero abiero	nmap	
17		TCP	49153	desconocido	puero abiero	nmap	
18		TCP	49154	desconocido	puero abiero	nmap	
19		TCP	49155	desconocido	puero abiero	nmap	
20		TCP	49156	desconocido	puero abiero	nmap	
21		TCP	49157	desconocido	puero abiero	nmap	
22		TCP	135	nsrpc	puero abiero	nmap	
23	10.10.5.24	TCP	139	netbios-ssn	puero abiero	nmap	
24		TCP	445	microsoft-ds	puero abiero	nmap	
25		TCP	2968	empp	puero abiero	nmap	
26		TCP	49155	desconocido	puero abiero	nmap	
27	10.10.5.67	TCP	135	msrpc	puero abiero	nmap	
28		TCP	139	netbios-ssn	puero abiero	nmap	
29		TCP	445	microsoft-ds	puero abiero	nmap	
30		TCP	2968	empp	puero abiero	nmap	
31		TCP	49163	desconocido	puero abiero	nmap	
32	10.10.5.92	TCP	445	microsoft-ds	puero abiero	nmap	
33		TCP	2869	icslap	puero abiero	nmap	
34		TCP	2868	empp	puero abiero	nmap	
35	10.10.5.95	TCP	135	msrpc	puero abiero	nmap	
36		TCP	445	microsoft-ds	puero abiero	nmap	
37		TCP	5000	vnc-http	puero abiero	nmap	
38		TCP	5900	vnc	puero abiero	nmap	
39	10.10.5.120	TCP	80	http	puero abiero	nmap	
40		TCP	443	https	puero abiero	nmap	
41		TCP	902	iss-realsecure	puero abiero	nmap	
42		TCP	912	apex-mesh	puero abiero	nmap	
43	192.168.120.12	TCP	139	netbios-ssn	puerto abierto	nmap	DOCENTES Y ESTUDIANTES
44		TCP	445	microsoft-ds	puerto abierto	nmap	
45		TCP	2869	icslap	puerto abierto	nmap	
46		TCP	2868	empp	puerto abierto	nmap	
47	192.168.120.14	TCP	135	msrpc	puerto abierto	nmap	
48		TCP	139	netbios-ssn	puerto abierto	nmap	
49		TCP	445	microsoft-ds	puerto abierto	nmap	
50	192.168.120.24	TCP	135	msrpc	puerto abierto	nmap	
51		TCP	445	microsoft-ds	puerto abierto	nmap	
52		TCP	2968	empp	puerto abierto	nmap	
53		TCP	49155	desconocido	puerto abierto	nmap	
54	192.168.120.39	TCP	135	msrpc	puerto abierto	nmap	
55		TCP	445	microsoft-ds	puerto abierto	nmap	
56		TCP	2968	empp	puerto abierto	nmap	
57		TCP	49163	desconocido	puerto abierto	nmap	
58	192.168.120.89	TCP	135	msrpc	puerto abierto	nmap	
59		TCP	139	netbios-ssn	puerto abierto	nmap	
60		TCP	445	microsoft-ds	puerto abierto	nmap	
61		TCP	554	rtsp	puerto abierto	nmap	
62		TCP	2869	icslap	puerto abierto	nmap	
63		TCP	5357	wsdapi	puerto abierto	nmap	
64		TCP	10243	desconocido	puerto abierto	nmap	
65	192.168.120.8	TCP	135	msrpc	puerto abierto	nmap	
66		TCP	139	netbios-ssn	puerto abierto	nmap	
67		TCP	445	microsoft-ds	puerto abierto	nmap	
68		TCP	49152	desconocido	puerto abierto	nmap	
69		TCP	49153	desconocido	puerto abierto	nmap	
70		TCP	49154	desconocido	puerto abierto	nmap	
71		TCP	49155	desconocido	puerto abierto	nmap	
72		TCP	49163	desconocido	puerto abierto	nmap	

Tabla 2. *Evidencia de puntos interactivos. Fuente: Nmap, nessus*

RIESGO	Seguridad Física		Seguridad en el Espectro	Seguridad en las Comunicaciones	Total
	Humano	Físico	Wireless	Redes de datos	
CONTROLES					
Clase A (Interacción)					
Autenticación	14	0	3	14	31
Indemnización	1	3	0	1	5
Resistencia	1	0	1	0	2
Subyugación	0	0	0	2	2
Continuidad	0	15	0	8	23
Total Clase A	**16**	**18**	**4**	**25**	**63**
Clase B (Proceso)					
No repudio	0	0	1	1	2
Confidencialidad	1	0	0	3	4
Privacidad	3	0	1	1	5
Integridad	0	0	3	1	4
Alarma	2	1	3	1	7
Total Clase B	**6**	**1**	**8**	**7**	**22**
Total Controles	**22**	**19**	**12**	**32**	**85**

Tabla 3. *Cuantificación de los controles de seguridad en los canales: humano, físico, wireless, redes de datos.*

Al analizar los datos de la tabla 3, en donde se cuantifican los controles encontrados durante la auditoria del tipo *hacking ético*, se puede observar que los controles de interacción o Tipo A son un total de 63 que afectan directamente a la visibilidad, acceso y confianza (porosidad); en cambio, de los controles de proceso o Tipo B, se cuantifica un total de 22, los cuales proporcionan seguridad ante amenazas. En la tabla 4 se evidencian los controles encontrados durante el estudio realizado.

Control / Factor	Humano	Físico	Wifi	Redes de Datos
Autenticación	Cada usuario establece contraseña de acceso al PC	No aplica	Validación de contraseña en cada acceso wifi y dirección mac	Autenticaciones puntos wifi
				Autenticaciones sistemas académico y financiero
				Administrador establece contraseña en pc de gestión administrativa
Indemnización	Soporte técnico externo	Posee Seguro de robos	No aplica	Contratos externos de soporte
		Posse seguro de catastros		
		Posee invetarios de activos		
Resistencia	Guardia de seguridad	No aplica	Permite acceso solo a usuarios registrados	No aplica
Subyugación	No aplica	No aplica	No aplica	Intercambio de informacion entre servidor de sistema financiero y académico y host clientes
Continuidad	No aplica	Conectividad a tierra	No aplica	Redundancia de hardware y software
		UPS en estaciones de trabajo administrativo		UPS para servidor de sistema financiero y académico
				UPS rack de redes de datos
No repudio	No aplica	No aplica	Solo se permiten usuarios registrados	Firma electrónica director
Confidencialidad	Firma electrónica dirección	No aplica	No aplica	Se destruyen discos catalogados como dañados
				Personal conoce habitos seguros de manejo de información relevante
				Repositorio externo para respaldos
Privacidad	Politicas de privacidad gestión financiera	No aplica	No aplica	Red de datos con 2 proveedores ISP
Integridad	No aplica	No aplica	Se autentica cada punto wifi	Firma electrónica director
Alarma	Si existen hurtos actuan los guardias de seguridad	Alarma contra incendios	Registro de eventos en puntos wifi	Firewall activo

Tabla 4. *Evidencia de controles encontrados. Fuente: Check list de verificación, nessus, nmap*

3.3. Fase de Investigación – Análisis de Vulnerabilidades.

RIESGO	Seguridad Física		Seguridad en el Espectro	Seguridad en las Comunicaciones	Total
	Humano	Físico	Wireless	Redes de datos	
LIMITACIONES					
Exposición	3	1	**0**	3	7
Vulnerabilidad	0	1	**3**	14	18
Debilidad	2	2	**0**	2	6
Preocupación	0	2	**0**	1	3
Anomalías	0	0	**0**	0	0
Total Limitaciones	5	6	**3**	20	34

Tabla 5. *Fase Investigación – Limitaciones*

En la tabla 5 se expone la cuantificación de las limitaciones, obteniéndose un total de 34, de las cuales 18 son vulnerabilidades que afectan directamente a la confiabilidad, integridad y disponibilidad de la información; en torno a esto ESET (2017) reporta que en Latinoamérica existe un crecimiento en cuanto a infecciones por malware, siendo Nicaragua el país que soporta más ataques de este tipo y apenas un 38% de las organizaciones en Latinoamérica realizan auditorías internas o externas enfocadas a cuantificar los riesgos en cuanto a seguridad informática. En torno a esto en la tabla 6 se evidencian vulnerabilidades más relevantes encontradas en la intranet de la Institución de Educación Superior objeto de estudio.

	IP	Descripción	Observación	Herramienta	Riesgo
Gestión Administrativa	10.10.2.1	Host vulnerable a un buffer overrun en el servicio de acceso remoto	Servidor sistema académico y financiero	nessus	ALTO
		Sistema Operativo Obsoleto - Windows 2003 server		nessus	
	10.10.2.20	Sistema Operativo Obsoleto - Windows XP	Pc posee modulo del sistema financiero	nessus	ALTO
	10.10.2.8	Sistema Operativo con falla en el DNS - S.O Windows 7	S.O sin actualizar	nessus	MEDIO
	10.10.2.4	Servicio OpenSSL obsoleto	version desactualizada Pc posee modulo académico	nessus	ALTO
		XAMPP Obsoleto		nessus	
		Servicio de Apache obsoleto.		nessus	
	10.10.5.1	Autenticacion SMB obsoleta	Actualizar SMB Windows 7	nessus	MEDIO
		Windows afectado por vulnerbilidad de privilegios en protocolos SAM		nessus	
Gestión Docente y estudiantes	192.168.120.23	Pc con sistema Windows XP	S.O obsoleto	nessus	ALTO
		Autenticacion SMB obsoleta		nessus	
	192.168.120.12	Autenticacion SMB obsoleta	Actualizar SMB Windows 7	nessus	MEDIO
	192.168.120.14	PHP V.4	Actualizar PHP v5	nessus	MEDIO
	182.168.120.24	XAMPP Obsoleto	Actualizar aplicativo de xampp server a la versión mas reciente	nessus	MEDIO

Tabla 6. *Vulnerabilidades. Fuente: Nessus*

3.4. Fase de Intervención

En esta fase se da a conocer el estado actual de la seguridad operacional de la Institución de Educación Superior objeto de estudio, una vez concluida la cuantificación de la porosidad, los controles y las limitaciones que se establecen en dicha Institución, los datos obtenidos son ingresados a la matriz de cálculo de RAV, propia de la metodología OSSTMM, obteniendo como resultado el 72,15% de seguridad actual y un 28,04% de brechas de seguridad informática.

4. Estrategias de gestión de seguridad informática con base en la metodología OSSTMM

La metodología OSSTMM propone para la optimización de la seguridad de los activos de información, que se disminuyan las limitaciones entre activos de información a proteger y posibles brechas de seguridad, así como también, la no separación de activos de información y brechas de seguridad informática, dando como resultado la porosidad. Según Herzog (2010) existen cuatro formas para crear separación de activos de información, siendo tres las recomendadas, estas son:

Mover el activo y crear una barrera entre él y las amenazas.
Los controles establecidos en la Institución de Educación Superior estudiada son pocos para toda la seguridad operacional, por lo que es considerable aumentar controles de proceso que permitan gestionar de una mejor manera la intranet con el fin de que los puntos interactivos y las limitaciones encontradas sean minimizados.

Cambiar la amenaza a un estado inofensivo.
Viable para este estudio, puesto que existe una cantidad considerable de puntos interactivos los cuales deben ser reducidos mediante el aumento de controles de confidencialidad, privacidad e integridad que permitan reducir amenazas y vulnerabilidades, aumentando la seguridad operacional de la intranet en la Institución de Educación Superior objeto de estudio.

Destruir la amenaza.
Las amenazas de carácter crítico que fueron halladas, deben ser destruidas para salvaguardar los activos de información, precautelando la confidencialidad, integridad y disponibilidad de la información. Se cataloga como amenazas potenciales a los sistemas operativos, servicios y aplicaciones obsoletas.

5. Mejoramiento del Risk Asessment Values (RAV)

Como se lo ha mencionado anteriormente, el valor agregado de la metodología OSSTMM es la cuantificación de los riesgos que se obtienen al realizar una auditoría informática, para este caso de estudio el resultado de protección en la intranet es de 72,15% de seguridad y de inseguridad es el 28,04%; para mejorar los resultados obtenidos se aplican las tres estrategias antes citadas en los puntos que se detallan en la tabla 7.

RIESGO	Seguridad Física	Seguridad en el Espectro		Seguridad en las Comunicaciones	Total
	Humano	Físico	Wireless	Redes de datos	
POROSIDAD					
Acceso	8	10	12	87	**117**
CONTROLES					
Confidencialidad	1	0	0	3	**4**
Privacidad	3	0	1	1	**5**
Integridad	0	0	3	1	**4**
Alarma	2	1	3	1	**7**
LIMITACIONES					
Vulnerabilidad	0	1	3	14	**18**
Exposición	3	1	0	3	**7**
Debilidad	2	2	0	2	**6**

Tabla 7. *Riesgos y Controles a mejorar*

Los resultados obtenidos de la porosidad en accesos sobrepasan los controles establecidos, para cambiar este estado, es necesario aplicar más controles de confidencialidad y privacidad en cada uno de los canales auditados, sobre todo en el canal redes de datos, puesto que existen puertos abiertos, algunos de manera innecesaria, los cuales se los puede cambiar a un estado inofensivo cerrándolos o controlando de mejor manera para evitar que afecte a la integridad, disponibilidad y confidencialidad de la información.

Así mismo existen algunas vulnerabilidades, exposiciones y debilidades encontradas en la auditoría realizada, que en su mayoría son sistemas operativos, aplicaciones y servicios obsoletos, antivirus sin actualizaciones, entre otros; los cuales se los puede controlar aumentando controles de alarma y de integridad o a su vez actualizar servicios, aplicaciones y sistemas operativos, para cual se debe cambiar las vulnerabilidades a un estado inofensivo.

RIESGO	Seguridad Física		Seguridad en el Espectro	Seguridad en las Comunicaciones	Total
	Humano	Físico	Wireless	Redes de datos	
POROSIDAD					
Acceso	4	5	12	40	**61**
CONTROLES					
Confidencialidad	2	2	3	4	**11**
Privacidad	5	3	6	12	**26**
Integridad	3	3	5	10	**21**
Alarma	4	3	6	10	**23**
LIMITACIONES					
Vulnerabilidad	0	1	3	5	**9**
Exposición	1	1	0	2	**4**
Debilidad	2	2	0	2	**3**

Tabla 8. *Mejoramiento de riesgos y controles*

En la tabla 8 se propone la cuantificación para el mejoramiento de la seguridad informática en la Institución de Educación Superior estudiada, obteniéndose como resultado de los cambios realizados un 77,00% de seguridad, lo cual disminuye el riesgo de inseguridad, teniendo en claro que no existe una seguridad perfecta, ya que el exceso de controles podría desencadenar en otro tipo de fallos que pueden estar ocultos pero actuando activamente sin que el administrador de la intranet se dé cuenta.

6. Conclusiones

En este trabajo se realizó una auditoría de seguridad informática a una institución de educación superior, mediante la aplicación de la metodología OSSTMM y pruebas de hacking ético, estableciendo métricas para evaluar el nivel de impacto y criticidad de las vulnerabilidades encontradas, en donde el principal hallazgo encontrado fue el valor de 72,15% de seguridad, equivalente a una seguridad informática media. Por lo tanto, se propone el mejoramiento de los valores de evaluación de riesgo (RAV) mediante la aplicación de estrategias, tales como: la creación de barreras entre el activo de información y la amenaza, cambiar la amenaza a un estado inofensivo y destruir las amenazas que pueden

vulnerar a la seguridad informática de la intranet, en donde, el punto de equilibrio estratégico es la disminución de la porosidad y de las limitaciones, obteniéndose un aumento del RAV de 77,00%. Adicionalmente, la disminución de las brechas de seguridad debe ser tratada de manera especial para cada activo de información, garantizando la confiabilidad, integridad y disponibilidad de la información.

Además, como parte de la investigación se cuantificaron los riesgos de la seguridad informática en los canales de información mediante la aplicación de la metodología OSSTMM y herramientas adecuadas para la evaluación de cada aspecto de la seguridad operacional, tales como: factores humanos, factores físicos, redes inalámbricas, servicios, aplicaciones, y redes de datos; encontrándose como resultados, que la mayoría de los elementos de la intranet evaluada, tienen riesgos altos de ser vulnerados y de sufrir ataques de seguridad informática.

Referencias

Benchimol. D. (2010). Redes Cisco. Banfield. Argentina: Gradi.

Baldeón, M. & Coronel, C. (2012). Plan maestro de Seguridad Informática para la UTIC de la ESPE con lineamiento en la norma ISO 27002.

Balcerek, B., Frankowski, G., Kwiecién, A. Smutnicki, A., & Teodorczyk, M. (2012). Security best practices: applying defense-in-depth strategy to protect the NGI_PL. Springer Berlin Heidelberg.128-141.

Costas Santos, J. (2010). Seguridad Informática. España: Service Ponit S.A.

Coronel, I. (2016). Aplicar Hackeo Ético para Detección de Vulnerabilidades Mediante Herramientas Open Sourse en las Aplicaciones Web de una Institución de Educación Superior. Disponible en: https://www.dspace.espol.edu.ec/retrieve/97627/D-103391.pdf. (Consultado 05/05/2017).

CISCO. (2016). Informe anual de seguridad de Cisco 2016. San José.

Emiliani, R. Sierra, Y. (2015). Manual Metodológico para pruebas de seguridad OSSTMM 3 y Guía de Pruebas OWASP 4. Disponible en: https://es.scribd.com/document/265102425/Resumen-de-Guias-OSSTMM-OTGv4. (Consultado 25-04-2017).

Enrique, J, & Sánchez, J. (2017). Riesgos de Ciberseguridad en las Empresas. Madrid

ESET Security Report. (2015). ESET Security Report, Latinoamérica 2015.

ESET Security Report. (2017). ESET Security Report, Latinoamérica 2017.

Fuertes, A. (2014). Elaboración de una Metodología de test de intrusión dentro de la Auditoría de Seguridad. Disponible en: http://reunir.unir.net/bitstream/handle/123456789/2331/AntonioFuertesMaestroTFM.pdf?sequence=3&isAllowed=y. (Consultado 25-06-2017).

Guillinta, O. Merino, J (2016). Modelo de Prevención y Defensa contra Ataques Cibernéticos basado en estándares de seguridad internacionales para IT-Expert. Disponible en: repositorioacademico.upc.edu.pe/upc/bitstream/10757/620848/1/MERINO_R_J.pdf. (Consultado 25-05-2017)

Hernández Sampieri, R., Fernández, C., & Baptista, M. (2010). Metodología de la Investigación. Quinta edición. México: McGraw-Hill.

Herzog P, et al. (2001). Open Source Security Testing Methodology. Manual v2.1. Agregar País: Editorial

Herzog P, et al. (2010). Open Source Security Testing Methodology. Manual v3. United Estates: Creative Commons

ISACA. (2015). State of Cybersecurity: Implications for 2015. Usa: Creative Commons

ISECOM. (2012). Hacker Highschool Security Awareness for teens, lección 1. United Estates: Creative Commons.

ISO/IEC 27001. (2013). ISO 27001:2013 Information technology – Security.

Instituto español de estudios estratégicos. (2011) Ciberseguridad, restos y amenazas a la seguridad nacional en el ciberespacio. Barcelona: Ministerio de Defensa.

Jara. H. (2012). Ethical Hacking 2.0. Buenos Aires. Argentina: Fox Andina.

López Santoyo, R. (2015). Propuesta de Implementación de metodología de auditoría de seguridad informática. Madrid: Universidad Autónoma de Madrid.

Maya, E. Jaramillo, D. (2015). Auditoría de Seguridad Informática para el Gobierno Autónomo Descentralizado de Santa Ana de Cotacachi, basada en la norma NTP-ISO/IEC 17799:2007 y la metodología OSSTMM V2. Disponible en: http://repositorio.utn.edu.ec/bitstream/123456789/3774/2/04%20RED%20034%2

0Art%C3%ADculo%20Cient%C3%ADfico%20Espa%C3%B1ol.pdf. (Consultado 05-04-2017).

Morlanes, G. (2012). Seguridad informática, Matanzas. CU. Revista de arquitectura e ingeniería. Vol 6. N° 2. P 1-14.

OWASP. (2013). Owasp Top 10 – 2013. Los 10 Riesgos más Críticos en Aplicaciones Web. Disponible en https://www.owasp.org

Piattini, M. Peso Navarro, E. y Peso Ruiz. M. (2008). Auditoría de tecnologías y sistemas de información. Madrid: RA-MA Editorial.

Portantier, F. (2013). Gestión de la Seguridad Informática. Buenos Aires. Argentina: Fox Andina.

Toth, G. Sznek, J. (2014). Implementación de la guía NIST SP 800-30 mediante la utilización de OSSTMM. Disponible en: https://es.scribd.com/document/323455632/Tesis-Toth-pdf. (Consultado 30-06-2016).

UNESCO. (2004). Las Tecnologías de la Información y la Comunicación en la Formación Docente. Montevideo. Uruguay: Gráfica Futura.

Uniandes. (2013). Manual de Políticas de Seguridad Informática. Ambato.

Yánez, E. (2015). Análisis de las Herramientas para el Proceso de Auditoría de Seguridad Informática Utilizando Kali Linux. Disponible en: http://www.dit.upm.es/~posgrado/doc/TFM/TFMs2014-2015/TFM_Ericka_Yanez_Cedeno_2015.pdf. (Consultado 04-27-2017).

Notas Biográficas:

Ing. Diego Sebastián Gordón Revelo Mgs. Ecuatoriano, nacido en la ciudad de Quito, el 18 de Octubre de 1990. Inició sus estudios universitarios en la Universidad Regional Autónoma de los Andes UNIANDES, obteniendo el título de Ingeniero en Sistemas e Informática y posteriormente continuó sus estudios en la Universidad de Especialidades Espíritu Santo donde obtuvo el título de Magister en Auditoría de Tecnologías de Información. En su vida profesional se ha desempeñado como Asistente de Tics y Analista de Soporte de TI en instituciones gubernamentales. Sus áreas de interés se centran en el estudio y aplicación de seguridad informática y soporte en tecnologías de información

M. Sc. Ing. Rubén Pacheco Villamar. Ecuatoriano, nacido en la ciudad de Guayaquil, el 22 de septiembre de 1965. Inició sus estudios universitarios en la Escuela Superior Politécnica del Litoral (ESPOL) de Guayaquil, y luego, en goce de una beca, continuó en Rusia, en la ciudad de San Petersburgo (entonces Leningrado), en la Universidad de Telecomunicaciones “Bonch Briyevich”. En esta universidad se graduó de Ingeniero en Telecomunicaciones en Transmisión de Datos, y posterior recibió el título de Master of Science en Telecomunicaciones. En su vida profesional se ha desempeñado como: Ingeniero de Soporte en Telecomunicaciones, Ingeniero de Diseño y Gerente de Proyectos para Enlaces de Telecomunicaciones, Redes de Computadoras, e Instalación y Operación de Centros de Datos, como Consultor en Gestión de Proyectos de TI, en Gestión de Procesos y Servicios de TI, y en Seguridad de Redes, y actualmente como Coordinador Nacional de Infraestructura y Producción de TICs en una institución gubernamental. Paralelamente, ha sido docente universitario y de investigación en Protel-ESPOL, en la Universidad San Francisco de Quito y en la Universidad de Especialidades Espíritu Santo; en esta última imparte clases en el pregrado de Ingeniería en Sistemas y Telecomunicaciones, y en posgrado, en la Maestría de Auditoría de Tecnologías de Información. Esporádicamente, realizo traducciones al español de textos en ruso y en inglés, y también se desempeña como Gerente General de una pequeña empresa de servicios de TI, fundada con un par de socios. A sus estudiantes siempre les menciona que “el solo hecho del conocimiento justifica el sacrificio”, pero que si, además, logran generar un cambio positivo en su entorno, se den por muy bien recompensados.

Recibido 1 Feb 2018 *ReCIBE, Año 7 No. 1, Mayo 2018*
Aceptado 2 Abr 2018

Factors Models of Scrum Adoption in the Software Development Process: A Systematic Literature Review

Modelos de Factores en la Adopción de Scrum in el Proceso de Desarrollo de Software. Una Revisión Sistemática de la Literatura

Marilyn Sihuay
msihuayr@unmsm.edu.pe
Escuela de Postgrado, Universidad Nacional Mayor de San Marcos, Lima, Perú

Abraham Dávila
abraham.davila@pucp.edu.pe
Departamento de Ingeniería, Pontificia Universidad Católica del Perú, Lima, Perú

Marcelo Pessoa
mpessoa@usp.br
Polytechnic School, University of Sao Paulo, Sao Paulo, Brazil

Abstract: (Background) The adoption of Agile Software Development (ASD), in particular Scrum, has grown significantly since its introduction in 2001. However, in Lima, many ASDs implementations have been not suitable (uncompleted or inconsistent), thus losing benefits obtainable by this approach and the critical success factors in this context are unknown. (Objective) To analyze factors models used in the evaluation of the adoption of ASDs, as these factors models can contribute to explaining the success or failure of these adoptions. (Method) In this study we used a systematic literature review. (Result) Ten models have been identified; their similarities and differences are presented. (Conclusion) Each model identified consider different factors, however some of them are shared by five of these models, such as team member attributes, engaging customer, customer collaboration, experience and work environment.

Keywords: Software Process Model, Process Adoption, Agile Software Development, Systematic Literature Review.

Resumen: (Antecedentes).La adopción del Desarrollo de Software Agile (DSA), en particular Scrum, ha crecido significativamente desde su introducción en 2001. Sin embargo, en Lima, muchas implementaciones de DSA no han sido adecuadas (incompletas o inconsistentes), perdiendo así los beneficios que se pueden obtener con este enfoque y los factores de éxito críticos en este contexto son desconocidos. (Objetivo) Analizar los modelos de factores utilizados en la evaluación de la adopción de ASD, ya que estos modelos de factores pueden contribuir a explicar el éxito o el fracaso de estas adopciones. (Método) En este estudio, utilizamos una revisión sistemática de la literatura. (Resultado) Diez modelos han sido identificados; sus similitudes y diferencias son presentadas. (Conclusión) Cada modelo identificado considera diferentes factores, sin embargo, algunos de ellos son compartidos por cinco de estos modelos, tales como los atributos del miembro del equipo, el compromiso del cliente, la colaboración del cliente, la experiencia y el entorno laboral.

Palabras Clave: Modelo de Proceso Software, Adopción de Proceso, Desarrollo de Software Agile, Revisión Sistemática de la Literatura.

1. Introduction

The software industry has changed significantly in the way software is developed. The software process has evolved from models with long phases and intermediate results towards models with frequent delivery aimed at satisfying customer needs (Azevedo Santos, 2011) (Dingsøyr, Nerur, Balijepally, & Brede Moe, 2012). With the publication of the Agile Manifesto in 2001, a collection of principles were expressed in order to address the change leaving traditional methods (Agile Manifesto, 2001). These principles emphasized interactions and conceptual simplicity, development-oriented and fast delivery, intense customer collaboration, high quality, low costs and dynamism to face constant changes in the project (Agile Manifesto, 2001). According to Agile Alliance, Agile Software Development (ASD) covers methods and practices based on Agile Manifesto (Agile Alliance, 2016). In the ASD context, the two most important aspects are productivity and quality (Kumar & Kumar Bhatia, 2012). Moreover, according to some authors, traditional software development methods do not conform to the current trends of businesses and technologies, where changes are frequent, affecting the organization and work team, decision-making, management requirements, relationships with partners or suppliers and organizational culture (Hass, 2009), (Chan & Thong, 2007), (Stankovic, Nikolic, Djordjevic, & Cao, 2013), (Chow & Cao, 2008), (Dyba, 2000), (Imreh & Raisinghani, 2011).

Other authors note that ASD are intended to support software development in organizations which aims to introduce and expand their products and services in dynamic markets (Cao, Mohan, Xu, & Ramesh, 2009), (Highsmith & Cockburn, 2001), (Bohem, 2002). The ASD practices have been well received in the industry, favoring its adoption in a growing number of companies (Zhang, Hu, Dai, & Li, 2010). Additionally, agile principles have been introduced in other domains such as project management (Mark, 2011).

Despite the useful principles of ASD, their adoption is a process that involves significant challenges (Pagrut, 2008), (Oyeyipo, 2011). According to the authors experience and preliminary literature review, this adoption also generates problems (Pagrut, 2008), (Dubakow, 2010). Since ASD are also a way to address processes, it is sensitive to people, involved technologies or the market itself; even at the beginning of a process of adoption (Agile Manifesto, 2001). According to Dubakow (Dubakow, 2010), (Mishra & Mishra, 2011), (Hajjdiab & Taleb, 2011), (Kanane, 2014), some problems during the adoption of ASD are: (1) to start using tools or processes before getting familiar with the method; (2) to use it only in development activities; (3) to use it without considering techniques; due to its importance for a balance between architecture and communication; (4) to consider the Scrum master as the project manager who assigns the tasks; (5) to locate team members base on the roles in the project; (6) the coach is not the right person to handle the adoption; (7) to not gather requirements from the

customer resulting in product redefinition; (8) the lack of a self-organized teams; which require a leader to guide the team to a clear objective; overzealous teams do not have the sufficient experience to implement agile methods; (9) to fail in managing sprint issues such as duration, work load, changes and freezing conditions of the sprints; (10) to perform testing on each sprint without considering non-functional requirements verifications; (11) to skip daily meetings because of current work pressure; and (12)) to maintain a traditional culture, bureaucratic structure and old documentation habits.

Scrum adoption must begin with a cultural change of the people involved and its success depends on the hard work and passion of the individuals (Dubakow, 2010). Other reported problems such as lack of information, lack of appropriate architecture planning, low test coverage (Cao, Mohan, Xu, & Ramesh, 2009) and limited knowledge of Scrum (Abrahamsson, Salo, Ronkainen, & Warsta, 2002), (Conboy & Fitzgerald, 2010); and among others are the main inhibitors to conduct an organizational innovations (Daghfous & White, 1994).

In our experience, we can say that Scrum is not successfully implemented due to an inadequate control where there is not a recognized Scrum Master in the organization. There are only some aspects of the framework that are actually implemented such as the daily Scrum meetings, retrospective meetings and the use of user stories, but these are disproportioned among the meetings of the project teams.

The adoption of a process model and its subsequent use are subject to a number of factors that influence the obtained results. As per Rogers, there are five factors in the adoption of an innovation (Rogers, 2003): (i) perceived relative advantage among users, (ii) compatibility with their needs and expectations, (iii) simplicity to be understood and implemented, (iv) initial trialability, and (v) observable benefits. Also, Fichman and Kemerer consider that there are limits in the adoption of an innovation such as the lack of diffusion of the innovation regarding their advantages, the complexity and compatibility, in additions to the lack of knowledge of the innovation's application (Fichman & Kemerer, 1999). Other relevant factors perceived in the context of Scrum are: lack of commitment of the management to engage managers (Dyba, 2005), (Goodman, 1996), (Kasse & McQuaid, 2000), (Niazi, Wilson, & Zowghi, 2006), (Powell, 1995), lack of clear objectives (Dyba, 2005), (Kasse & McQuaid, 2000), (El Emam, Goldenson, McCurley, & Herbsleb, 1998) especially inexperienced technical teams (Baddoo & Hall, 2003), (Goodman, 1996), (Kasse & McQuaid, 2000), (Niazi, Wilson, & Zowghi, 2006), lack of training to learn new methods and techniques (Niazi, Wilson, & Zowghi, 2006), (Powell, 1995), pilot syndrome which limits the expansion of benefits just to the pilot project and not the whole company (Goodman, 1996), (Repenning & Sterman, 2001), lack of measurement to demonstrate the success of the adoption (Dyba, 2000), (Niazi, Wilson, & Zowghi,

2006), (Powell, 1995), (Rousseau & McCarthy, 2007), process-oriented rather than results-oriented which include the approval of the managers (Baddoo & Hall, 2003), (Kasse & McQuaid, 2000), commercial pressure where the product owner should be able to prioritize conflicting user requirements (Baddoo & Hall, 2003), (Kasse & McQuaid, 2000), and support tools that will enable better decision (Kasse & McQuaid, 2000).

To analyze factors models used in the assessment of ASDs adoption, we conducted a Systematic Literature Review to collect different factors from scientific data source applying a research method oriented to this objective.

This article presents factors models for the adoption of Scrum methods with a background of agile adoption models. The article is organized as follows: Section 2 presents the systematic literature review protocol; in Section 3 the identified models are introduced; and Section 4 contains the final discussion and future work.

2. Systematic Literature Review

Kitchenham defines a systematic literature review (SLR) as the process to identify, evaluate and interpret all available research relevant to a particular research question, or phenomenon of interest (Kitchenham, 2007), establishing a sequential tailored set of activities, in Figure 1 shows the phases used in this study. In addition, primary studies are contributors of secondary studies (Kitchenham, 2007). In our case, we decided to use a SLR considering that some search results would be case studies or similar and others would be secondary studies. Consequently, we did not use words related to primary or secondary studies and instead we used others related to models factors and ASD.

Some benefits of a SLR are: i) identification of the particular research questions to be investigated; ii) identification of the desired population; iii) intervention, context and outcomes and helps in summarizing the existing research evidence and others (Kitchenham, 2007).

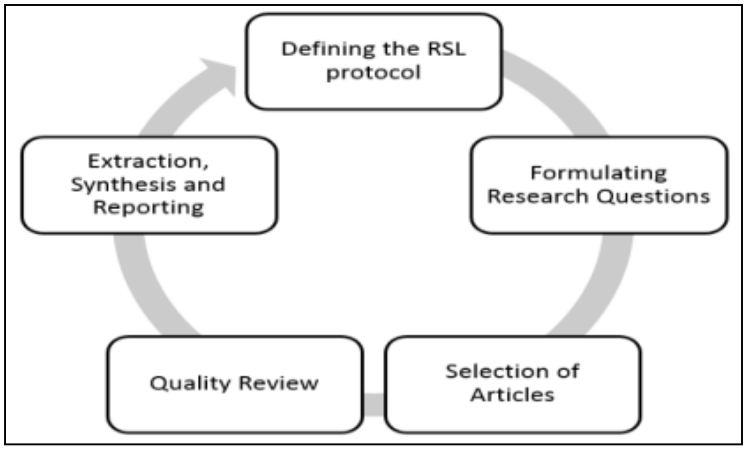

Figure 1. SLR phase, adapted from (Kitchenham, 2007)

2.1. SLR Protocol

The protocol used in this study was adapted from Kitchenham (Kitchenham, 2007) and Sulayman (Sulayman & Mendes, 2009). The phases are:

Defining the SLR protocol: this phase is an iterative flow and covers the overall plan for the SLR process.

Formulating research questions: in this phase the SLR research questions are identified.

Selection of articles: in this phase inclusion and exclusion criteria are considered for selecting primary or secondary articles.

Quality review: in this phase selected articles are submitted to a quality review, where checklists are used to validate the articles filtered in previous phase.

Extraction, synthesis and reporting: in this phase relevant data is extracted from each article. Then, it is added, integrated and summarized in order to answer the research questions in detail.

This SLR aims to consolidate the information about factor models of agile methods adoption in the software development process. The present SRL summarizes existing models which organizes factors in order to explain the influence of these factors on the adoption of agile methods in the software development process and shows common factors shared between all identified models.

2.2. Formulating Research Question

Formulation of research question has been developed following the PICO guidelines presented by Santos (Santos, Pimenta, & Nobre, 2007): Population, Intervention, Comparison and Outcome, as described in Table 1. This research is not focused on what is the intervention compared with, therefore, it is omitted in the subsequent tables and sections. The final search string is the result of some iterations in which each string was refined according to the quantity and quality of articles.

Population: groups of elements that are observed by the intervention. Studies presenting agile methods implementations in software processes and projects.

Intervention: elements to be evaluated within the defined population. These are the factors models or frameworks used in the adoption of agile methods.

Outcome: the result of the information according the investigation. Primary studies about factors models of agile methods adoption in software processes.

Keywords used in the study	
Population	(Agile OR Scrum OR XP) AND ("software process" OR Methodology OR Methods OR "software project" OR "software projects" or "software development")
Intervention	(adoption OR adopt OR implementation OR impact OR adaptation)
Outcome	(model OR framework) AND (Factor OR enable OR disable)
Research Strategy	Population AND Intervention AND Results

Table 1. *Keywords obtained from PICO strategy*

Based on the criteria defined above we established the following research question:

RQ1: Which factors models related to the adoption of ASD are used in the software industry? Models found will be used to identify relevant factors.

RQ2: What factors are common to all the identified models? Factors will be classified based on their presence in all the identified models.

The search was performed in April 2016 and includes studies before that date.

2.3. Studies Selection

The search procedure starts when the search string is built based on PICO criteria, keywords found in the studies, synonyms terms and Boolean terms AND - OR.

Once the search string was defined, this was used on the following online digital libraries to obtain primary studies (Dieste, Grimán, & Juristo, 2009), (Elberzhager, Münch, & Vi, 2012):

ACM Digital Library (http://portal.acm.org)

Proquest (http://www.proquest.com)

Elsevier ScienceDirect (www.sciencedirect.com)

Scopus (http://www.scopus.com)

Thomson Reuters - Web of Knowledge (www.webofknowledge.com)

IEEE (https://ieeexplore.ieee.org/)

Once the results of the search were obtained during the first iteration, titles were analyzed to determine their actual relevance. In addition, during the second iteration, abstracts and conclusions were analyzed. Both iterations were intended to identify those primary studies that answer the research questions.

Inclusion criteria refer to studies that are related to models of factors to adopt agile methods. We used: (CI1) title and abstract related to the object of study; (CI2) studies related to agile methods adoption; (CI3) studies related to Scrum adoption and (CI4) studies related to XP adoption. Exclusion criteria refer to the exclusion of studies that are not focused on factors models or agile method adoption. We used: (CE1) studies which did not relate to factors models adoption and (CE2) duplicate studies.

On the other hand, quality criteria contribute to assess the reliability of the papers. In the first iteration we applied CI1, CI2, CI3 and CI4. In the second, we applied CE1; and finally, on iteration 3, we applied CE2 (see Table 2).

Database	Results	Iterat. 1	Iterat. 2	Iterat. 3
ACM	56	15	7	2
Proquest	24	5	1	0
Science Direct	28	6	4	1
Scopus	122	41	10	6
Web of Knowledge	120	10	8	0
IEEE	3	1	0	1
Total	353	78	30	10

Table 2. *Results of studies Retrieved and Selected*

2.4. Study Quality Assessment

The checklist used to assess the quality of selected studies is listed in Table 3. These qualitative questions were obtained from Sulayma (Sulayman & Mendes, 2009). Based on the answers, each study might have the following qualifications: 1.0 if the answer is '[Y]es', 0.5 if the answer is '[P]artially' and 0.0 if the answer is '[N]o'. Thus, the study could obtain a maximum score of 9. The results of the selected studies and their partial and final scores are presented in Table 4. In our case, (Srinivasan, Dobrin, & Lundqvist, 2009) obtained 5 of 9 points (55%) and represents one of ten models identified. After an individual review and rating than 50%, we decided to include it.

Question
1. Is the methodology used suitable to address the stated research questions?
2. Does the article target the ideal population?
3. Does the article use the research methodology adequately?
4. Does the article discuss any of the previous work/literature?
5. Is the study process specified in the article repeteable?
6. Is the article oriented towards a factors framework, model or techique?
7. Do the findings address the research questions?
8. Does the article document any assumption taken?
9. Does the article document the procedure used to validate its findings?

Table 3. *Quality questions used in our study (Sulayman & Mendes, 2009).*

3. Extraction, Synthesis and Results

The models found in the articles selected as primary studies are described in this Section.

Categories are the same used by Shahane: Organization, Project, Process and People which are based on the five axis polar charts as suggested by Boehm and 3-factor comparison: People, Process, Projects (Shahane, Jamsandekar, & Shahane, 2014). Also, we consolidate in Table 5, agile methods studied in the selected articles.

Study	1	2	3	4	5	6	7	8	9	T
(Ahimbisibwe, Cavana, & Daellenbach, 2015)	Y	Y	Y	Y	Y	Y	Y	Y	P	8.5
(Shahane, Jamsandekar, & Shahane, 2014)	P	P	Y	N	P	Y	Y	P	N	5.5
(Melo, Cruzes, Kon, & Conradi, 2013)	Y	P	P	N	Y	Y	Y	P	Y	6.5
(Lee, 2012)	P	P	P	P	P	Y	Y	N	Y	5.5
(Overhage, Sebastian, Birkmeier, & Miller, 2011)	P	Y	Y	Y	P	Y	Y	P	P	7.0
(Misra, Kumar, & Kumar, 2009)	P	P	P	Y	P	Y	Y	P	Y	6.5
(Chow & Cao, 2008)	Y	Y	Y	Y	P	Y	Y	Y	P	8.0
(Chan & Thong, 2007)	P	P	P	Y	P	Y	Y	N	Y	6.0
(Srinivasan, Dobrin, & Lundqvist, 2009)	P	N	P	Y	N	Y	Y	N	Y	5.0
(Stettina & Heijstek, 2011)	Y	Y	Y	N	P	Y	Y	P	Y	7.0

Table 4. *Quality Result.*

Study	Agile	Scrum	XP	Lean	Other
(Ahimbisibwe, Cavana, & Daellenbach, 2015)	X				
(Shahane, Jamsandekar, & Shahane, 2014)	X	X	X		X
(Melo, Cruzes, Kon, & Conradi, 2013)	X	X	X	X	
(Lee, 2012)	X	X			
(Overhage, Sebastian, Birkmeier, & Miller, 2011)	X	X			
(Misra, Kumar, & Kumar, 2009)	X				
(Chow & Cao, 2008)	X	X	X		X
(Chan & Thong, 2007)	X	X	X		X
(Srinivasan, Dobrin, & Lundqvist, 2009)	X				
(Stettina & Heijstek, 2011)	X	X			
	10	7	4	1	3

Table 5. *Study by Agile Method*

3.1. Ahimbisibwe et. al. - CSF for Software Development Projects

The study of Ahimbisibwe, Cavana, Daellenbach (Ahimbisibwe, Cavana, & Daellenbach, 2015) had the purpose to identify and categorize critical success factors (CSFs) and develop a contingency model to adjust the contrasting perspectives of traditional and agile methodologies (Ahimbisibwe, Cavana, & Daellenbach, 2015).

Based on a previous systematic literature review done by the authors, there were identified 37 CSFs for software development projects within 148 articles, and categorized into three major CSFs: organizational, team and customer factors. The contingency model increases these factors by highlighting the need to match project characteristics and project management methodology to these CSFs.

3.2. Shahane et. al.- Conceptual Framework

The framework Shahane, Jamsandekar and Shahane proposed in 2014 is based on the revision of proposed factors and existing models found in a literature review. The model is oriented to the elements: organization, people, processes and projects, considered as cornerstones (Shahane, Jamsandekar, & Shahane, 2014). In addition, the model is based in one equilateral triangle (pyramid) divided in four equilateral triangle. In this model each triangle in the framework pyramid represents a set of critical factors for the success of any project (Shahane, Jamsandekar, & Shahane, 2014).

3.3. Melo et. al. - Productivity Factors Framework

The framework of Melo et al (Melo, Cruzes, Kon, & Conradi, 2013) presented in 2012 is based on the theoretical model of effectiveness Input-Process-Outcome (IPO) of Cohen and Bailey, Yeatts and Hyten. It is a multiple-case study during six months in three large Brazilian companies, which had used agile methods for

more than 2 years. The study is focused on the main productivity factors perceived by the team members through interviews, retrospectives and documentation (Melo, Cruzes, Kon, & Conradi, 2013). As a result, it was developed a conceptual framework, using thematic analysis to understand the possible mechanisms behind these productivity factors. Agile team's management proved to be the most influential factor in the achievement of agile team productivity. For intra-team level, the main productivity factors were team design (structure and allocation of work) and employee turnover (Melo, Cruzes, Kon, & Conradi, 2013). For inter-team level, the main productivity factors were the ability to coordinate effectively through appropriate interfaces, avoiding delays of the software.

3.4. Lee - Scrum Performance Dynamic

The Dynamic Performance Scrum is a framework designed by Rich Lee in 2012, based on a qualitative research, led by observation and interviews of two teams (Lee, 2012). The categories used were covered based on the review of existing literature, such as: personal attributes, characteristics of user stories, capacity of project team, team autonomy, team diversity, change response, efficiency and performance team in software development (Lee, 2012).

3.5. Overhage et. al. - Framework of Drivers and Inhibitors to Developer Acceptance

Overhage, Schlauderer, Birkmeier framework was defined in 2011 as a group of drivers and inhibitors to the developer acceptance of Scrum. This framework is based on the Extended Technology Acceptance Model (TAM), applicable to developer acceptance of this methodology (Overhage, Sebastian, Birkmeier, & Miller, 2011).

Their results were based on six qualitative interviews applied to six Scrum experienced experts of a German company. The general determinants defined in the TAM were refined with several factors that have influence in the willingness of developers to use Scrum (Overhage, Sebastian, Birkmeier, & Miller, 2011).

3.6. Misra et. al. - Success Factors Framework

This framework was developed in 2009 and it is based on a literature review. It establishes 14 factors derived from its hypothesis, and the most important factors were determined based on questionnaires, where nine factors were related to success: satisfaction customer, customer collaboration, customer commitment, decision time, corporate culture, control, characteristics of people, culture social, and learning and training (Misra, Kumar, & Kumar, 2009). Multiple regression models were used to test the relations between the success factors (Misra, Kumar, & Kumar, 2009).

3.7. Chow and Cao - Factors Model

The Chow-Cao Model was defined in 2007 (Chow & Cao, 2008). It has the following characteristics: i) it is based on a quantitative analysis; ii) it identified 12 critical factors grouped into 4 categories of project success: quality, scope, time and cost; iii) the surveys were distributed to professionals of the agile community in 109 projects in 25 countries; iv) to validate the model they used regression techniques such as complete model and optimized model and 10 out of 48 hypotheses were verified; v) only three factors should be considered critical: delivery strategy, agile software engineering techniques and team capabilities.

3.8. Chan and Thong - Conceptual Framework

The conceptual framework of Chan and Thong was developed in 2007, based on previous empirical studies about the acceptance of Systems Development Method (SDM) in the organization/individual and studies on agile methodologies, where potential factors were identified (Chan & Thong, 2007). These factors were classified as: (i) individual factors associated with software developers; (ii) organizational factors associated with the management and organization; (iii) factors associated with agile methodology; and (iv) factors associated with the relationship creator-client (Chan & Thong, 2007).

3.9. Srinivasan et. al.– Technical factors Framework

In this paper, Srinivasan, Dobrin and Lundqvist (Srinivasan, Dobrin, & Lundqvist, 2009) have found that there are technical issues (requirement management, and testing), as well as organizational issues (process tailoring, knowledge sharing and transfer, culture change and infrastructure support). They were considered as a framework for preliminary guidance.

3.10. Stettina and Heijstek. Five Agile Factors.

This study is based on the qualitative model of Moe et al., Stettina and Heijstek (Srinivasan, Dobrin, & Lundqvist, 2009), who developed a quantitative questionnaire organized among five dimensions of agile teamwork, analogous to the Five Factor Model in contemporary psychology. This survey was conducted with 79 individuals and eight international Scrum teams.

4. Final Discussion and Future Work

Some of the studies found evaluate factors related to the method itself, and others explore factors related with the organizational practice taking into consideration different factors. Consequently, there are factors, which were studied individually and others that were classified in the same category by the selected studies. Categories are the same used by Shahane: Organization, Project, Process and People.
The amount of factors per category is shown in Figure 2.

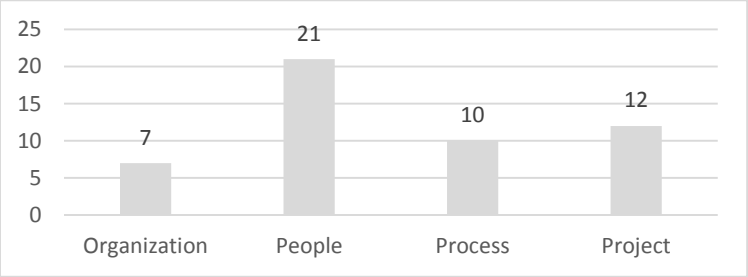

Figura 2. *Factors Quantity per Coincidence*

Then those factors are listed by article in Table 6, showing which factors have more concurrence. The higher the concurrence is better the comparison analysis between models. Only eight factors had the higher concurrence, greater than three. They are: user participation, team orientation, team qualified attributes, training and learning, teamwork, team's experience, user support, and requirements generation.

User participation and User support. Both enables software developers and customers to work towards a common objective in the most effective and agile way. Therefore, software developers are more likely to adopt agile methodology when they have a good understanding with its customers (Ahimbisibwe, Cavana, & Daellenbach, 2015), (Shahane, Jamsandekar, & Shahane, 2014), (Melo, Cruzes, Kon, & Conradi, 2013), (Misra, Kumar, & Kumar, 2009), (Srinivasan, Dobrin, & Lundqvist, 2009), (Chow & Cao, 2008), (Chan & Thong, 2007).

Team orientation. Team goals are prioritized over individual goals. This increases individual responsibility (Ahimbisibwe, Cavana, & Daellenbach, 2015), (Melo, Cruzes, Kon, & Conradi, 2013), (Lee, 2012), (Overhage, Sebastian, Birkmeier, & Miller, 2011), (Chan & Thong, 2007), (Srinivasan, Dobrin, & Lundqvist, 2009).

Team qualified attributes. The technical and business skills are the most relevant (Lee, 2012), (Misra, Kumar, & Kumar, 2009), (Chow & Cao, 2008), (Chan & Thong, 2007). According to the authors, agile method is best adopted by developers when they have received adequate training.

Training and learning. Shared knowledge that allows its maintenance (Ahimbisibwe, Cavana, & Daellenbach, 2015), this allows an organization to develop knowledge and to be better prepared to implement the methodology (Ahimbisibwe, Cavana, & Daellenbach, 2015), (Shahane, Jamsandekar, & Shahane, 2014), (Lee, 2012), (Chan & Thong, 2007)

Teamwork. A work environment based on collaboration and an accurate social pressure; these elements will allow the developer an early adoption of the agile method (Melo, Cruzes, Kon, & Conradi, 2013), (Lee, 2012), (Overhage, Sebastian, Birkmeier, & Miller, 2011), (Misra, Kumar, & Kumar, 2009). When developers work in a communicative, dynamic and progressive environment, they are prepared for success (Misra, Kumar, & Kumar, 2009).

In the people category, coincidentally, the most relevant factors are oriented to the team, team attributes, teamwork and team experience (Ahimbisibwe, Cavana, & Daellenbach, 2015), (Melo, Cruzes, Kon, & Conradi, 2013), (Stankovic, Nikolic, Djordjevic, & Cao, 2013), (Lee, 2012), (Overhage, Sebastian, Birkmeier, & Miller, 2011), (Misra, Kumar, & Kumar, 2009), (Chow & Cao, 2008), (Chan & Thong, 2007), (Srinivasan, Dobrin, & Lundqvist, 2009), which allow the team to be empowered.

Requirements generation. The functional specification of the final product is very important. It should be consolidated in an iterative way with continued participation of the user. A good definition will allow proper implementation (Shahane, Jamsandekar, & Shahane, 2014), (Lee, 2012), (Chan & Thong, 2007), (Srinivasan, Dobrin, & Lundqvist, 2009).

Category	Factor	Ahimbisibwe et. al.	Shahane et. al.	Melo et. al.	Lee	Overhage et. al.	Misra et. al.	Chow and Cao	Chan and Thong	Srinivasan et. al.	Stettina and Heijstek	Coincidence
People	User participation	X	X				X	X	X	X		6
People	Team orientation	X		X	X	X			X		X	6
People	Team qualified attributes				X		X	X	X			4
People	Training and Learning	X	X				X				X	4
People	Teamwork			X	X	X	X					4
People	Team's Experience	X	X		X				X			4
People	User support	X		X				X		X		4
Process	Requirements generation		X		X				X	X		4
People	Work Environment						X	X	X			3
People	Autonomy					X				X	X	3
Process	Simplicity		X					X	X			3
Process	Methodology		X			X		X				3
Process	Process Adaptation					X			X	X		3
Project	Complexity	X	X		X							3
Project	On-Site / Off * Shore		X		X				X			3
Project	Project Size	X	X				X					3
Project	Urgency	X	X		X							3
Organization	Management Support		X							X		2
Organization	Vision and mission	X	X									2
Organization	Culture Change		X							X		2
Organization	Shared Leadership	X									X	2
People	Team Size		X	X								2
People	Diversity team			X	X							2
People	Efficiency				X				X			2
People	Communication	X							X			2
People	Team composition	X		X								2
Process	Knowledge, sharing & transfer					X				X		2
Process	Dynamism		X					X				2
Process	Maturity					X			X			2
Project	Project planning and controlling	X				X						2
Project	Technological uncertainty	X								X		2
Organization	Security		X									1
Organization	Stability		X									1
Organization	Organizational culture	X										1
People	Learning curve				X							1
People	Career opportunities								X			1
People	Transparency					X						1
People	Staff turnover			X								1
People	Location			X								1
People	Team's expertise	X										1
People	Customer experience	X										1
Process	Perception of Use								X			1
Process	Redundancy										X	1
Process	Testing Approaches									X		1
Project	Change management	X										1
Project	Cost				X							1
Project	Client satisfaction						X					1
Project	Rules								X			1
Project	Project Specification Changes	X										1
Project	Project criticality	X										1

Table 6. *Factors per Author*

The appearance of these models came from 2007 to 2015, models Chan – Thong (Chan & Thong, 2007) and Chow – Cao (Chow & Cao, 2008) are the oldest and the most recent is Ahimbisibwe et.al.
Table 7 indicates the study technique (research) that was used for the selection of factors, which were corroborated using questionnaires, surveys or using empirical and analytical case studies.

Study	Literature review	Questionaires	Case study	Exploratory Expert Opinion
(Ahimbisibwe, Cavana, & Daellenbach, 2015)	X			
(Shahane, Jamsandekar, & Shahane, 2014)			X	
(Melo, Cruzes, Kon, & Conradi, 2013)			X	
(Lee, 2012)				X
(Overhage, Sebastian, Birkmeier, & Miller, 2011)	X	X		
(Misra, Kumar, & Kumar, 2009)	X	X		
(Chow & Cao, 2008)	X		x	
(Chan & Thong, 2007)	X			
(Srinivasan, Dobrin, & Lundqvist, 2009)	X			
(Stettina & Heijstek, 2011)		X		

Table 7. *Characteristic of Models Found*

Finally, it was found empirical work about impact factors analysis in the adoption of agile methods, such as Mann (Mann & Maurer, 2005), Imreh (Imreh & Raisinghani, 2011), Kumar (Kumar & Kumar Bhatia, 2012) Pagrut (Pagrut, 2008), Mark (Mark, 2011), Stankovic (Stankovic, Nikolic, Djordjevic, & Cao, 2013), among others. Those articles may be considered for future research related to impact factors. In addition, as a future research, we will consider a case study to evaluate those models in a local context.

Acknowledgment

This work is framed within ProCal-ProSer Contract N° 210-FINCYT-IA-2013 (Innovate Perú): “Productivity and Quality Relevance Factors in small software development or service organizations adopted ISO standards”, the Software Engineering Development and Research Group and the Department of Engineering of Pontificia Universidad Católica del Perú.

References

Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2002). Agile Software development methods. Review and analysis. Finland: VTT Publications.

Agile Alliance. (06 de 04 de 2016). What is Agile Software Development? Obtenido de https://www.agilealliance.org/agile101/what-is-agile/

Agile Manifesto. (2001). Recuperado el 8 de Aug de 2015, de http://agilemanifesto.org

Ahimbisibwe, A., Cavana, R. Y., & Daellenbach, U. (2015). A contingency fit model of critical success factors for software development projects: A comparison of agile and traditional plan-based methodologies. Journal of Enterprise Information Management, 28(1), 7 - 33.

Azevedo Santos, M. d. (2011). Agile Practices: An Assessment of Perception of Value of Professionals on the Quality Criteria in Performance of Projects. Journal of Software Engineering and Applications - Scientific Research, 4(12), 700-709.

Baddoo, N., & Hall, T. (Apr de 2003). De-motivators for Software Process Improvement: An Analysis of Practitioners' Views. The Journal of Systems and Software, 66(1), 23–33.

Bohem, B. (Jan de 2002). Get ready for agile methods, with care. Computer, 35(1), 64-69.

Cao, L., Mohan, K., Xu, P., & Ramesh, B. (August de 2009). A framework for adapting agile development methodologies. European Journal of Information Systems, 18(4), 332–343.

Chan, K., & Thong, J. (2007). An Integrated Frame-work of Individual Acceptance of Agile Methodologies. Pacific Asia Conference on Information Systems (PACIS) (pág. Paper 154). Auckland, New Zealand: AISEL.

Chow, T., & Cao, D.-B. (Jun de 2008). A survey study of critical success factors in agile software projects. Journal of Systems and Software, 81(6), 961-971.

Conboy, K., & Fitzgerald, B. (June de 2010). Method and developer characteristics for effective agile method tailoring: A study of XP expert opinion. ACM Trans. Softw. Eng. Methodol, 20(1), Paper 2, 30p.

Daghfous, A., & White, G. R. (1994). Information and innovation: a comprehensive representation. Research Policy, 23(3), 267-280.

Dieste, O., Grimán, A., & Juristo, N. (2009). Developing search strategies for detecting relevant experiments. Empirical Software Engineering, 14(5), 513-539.

Dingsøyr, T., Nerur, S., Balijepally, V., & Brede Moe, N. (Jun de 2012). A decade of agile methodolo-gies: Towards explaining agile software development. Journal of Systems and Software, Elsevier, 85(6), 1213–1221.

Dubakow, M. (10 de 2010). 10 Most Common Mistakes in Agile Adoption. Recuperado el 06 de 04 de 2016, de Targetprocess Inc: https://www.targetprocess.com/blog/2010/10/10-most-common-mistakes-in-agile-adoption-part-i/

Dyba, T. (Set-Oct de 2000). Improvisation in Small Software Organizations. IEEE Software, 17(5), 82–87.

Dyba, T. (May de 2005). An empirical investigation of the key factors for success in software process improvement. IEEE Transactions on Software Engineering, 31, 410-424.

El Emam, K., Goldenson, D., McCurley, J., & Herbsleb, J. (1998). Success or Failure? Modeling the Likelihood of Software Process Improvement. International Software Engineering Research Network, ISERN-98-15. ISERN-98.

Elberzhager, F., Münch, J., & Vi, T. (Jan de 2012). A systematic mapping study on the combination of static and dynamic quality assurance techniques. Information and Software Technology, 54(1), 1-15.

Fichman, R. G., & Kemerer, C. F. (1999). The Illusory Diffusion of Innovations: An Examination of Assimilation Gaps. Information Systems Research, 10(3), 255 - 275.

Goodman, P. (1996). The Practical Implementation of Process Improvement Initiatives. En N. Fenton, R. Whitty, & Y. Lizuka (Edits.), Software Quality Assurance and Measurement: A Worldwide Perspective (pág. 315). International Thomson Computer Press.

Hajjdiab, H., & Taleb, A. (Sep de 2011). Adopting Agile Software Development: Issues and Challenges. International Journal of Managing Value and Supply Chains (IJMVSC), 2(3).

Hass, K. B. (10 de March de 2009). The Blending of Traditional and Agile Project Management. Recuperado el 06 de September de 2015, de PM Times for Projects Managers: https://www.projecttimes.com/articles/the-blending-of-traditional-and-agile-project-management.html.

Highsmith, J., & Cockburn, A. (Sep de 2001). Agile software development: the business of innovation. Computer, 34(9), 120-127.

Imreh, R., & Raisinghani, M. S. (Oct de 2011). Impact of Agile Software Development on Quality within Information Technology Organizations. Journal of Emerging Trends in Computing and Information Sciences, 2(10).

Kanane, A. (2014). Challenges related to the adoption of Scrum. Case study of a financial IT company. UMEA University, Department of informatics. IT management master program. UMEA University.

Kasse , T., & McQuaid, P. A. (Aug de 2000). Factors Affecting Process Improvement Initiatives. Crosstalk: the Journal of De-fense Software Engineering, 13(8), 4-8.

Kitchenham, B. (2007). Guidelines for performing Systematic Literature Reviews in Software Engineering. EBSE Technical Report. EBSE-2007-01, Keele University , School of Computer Science and Mathematics.

Kumar, G., & Kumar Bhatia, P. (2012). Impact of Agile Methodology on Software Development Process. International Journal of Computer Technology and Electronics Engineering (IJCTEE), 2(4), 46-50.

Lee, R. (2012). The Success Factors of Running Scrum: a Qualitative Perspective. Journal of Software Engineering and Applications, 5(6), 367-374.

Mann, C., & Maurer, F. (2005). A Case Study on the Impact of Scrum on Overtime and Customer Satisfaction. Agile Development Conference (ADC'05), (págs. 70-79).

Mark, P. (2011). On empirical research into scrum. Institute for Software Research Carnegie Mellon University, Pittsburgh. Obtenido de http://www.cs.cmu.edu/~mcp/agile/oersa.pdf

Melo, C., Cruzes, D. S., Kon, F., & Conradi, R. (Feb de 2013). Interpretative case studies on agile team productivity and management. Information and Software Technology, 55(2), 412–427.

Mishra, D., & Mishra, A. (Jan de 2011). Complex software project development: agile methods adoption. Journal of Softwrae Maintenance and Evolution: Research and Practice, 549–564.

Misra, S. C., Kumar, V., & Kumar, U. (Nov. de 2009). Identifying some important success factors in adopting agile software development practices. Journal of Systems and Software, 82(11), 1869-1890.

Niazi, M., Wilson, D., & Zowghi, D. (2006). Critical Success Factors for Software Process Improvement Implementation: An Empirical Study. Software Process: Improvement and Practice, 11(2), 193-211.

Overhage, S., Sebastian, S., Birkmeier, D., & Miller, J. (2011). What Makes IT Personnel Adopt Scrum? A Framework of Drivers and Inhibitors to Developer Acceptance. 44th Hawaii International Conference on System Sciences (HICSS), (págs. 1-10). Kauai.

Overhage, S., Sebastian, S., Birkmeier, D., & Miller, J. (2011). What Makes IT Personnel Adopt Scrum? A Framework of Drivers and Inhibitors to Developer Acceptance. 44th Hawaii International Conference on System Sciences (HICSS), (págs. 1-10). Kauai.

Oyeyipo, E. (2011). An empirical study of requirements management in an agile-Scrum development enviroment. Thesis Master of Sciemce, Universidad San Marcos, Department of Computer Science, Texas.

Pagrut, D. (2008). The Impact of an Agile Scrum on Software Testing: A Case Study of Tech Mahindra Limited. 5th International Conference On Software Testing. STeP-IN SUMMIT.

Powell, T. C. (1995). Total Quality Management as Competitive Advantage: A Review and Empirical Study. Strategic Management Journal, 16, 15-37.

Repenning, N. P., & Sterman, J. D. (Summer de 2001). Nobody Ever Gets Credit for Fixing Problems that Never Happened: Creating and Sustaining Process Improvement. California Management Review, 43(4), 64-88.

Rogers, E. M. (2003). Diffusion of Innovations (Fifth Edition ed.). New York: The Fnree Press.

Rousseau, D. M., & McCarthy, S. (Mar de 2007). Educating Managers From an Evidence-Based Perspective. Academy of Management Learning & Education, 6(1), 84-101.

Santos, C., Pimenta, C., & Nobre, M. (2007). A estratégia PICO para a construção da pergunta de pesquisa e busca de evidências. Revista Latino-Americana de Enfermagem, 15(3), 508-511.

Shahane, D., Jamsandekar, P., & Shahane, D. (2014). Factors influencing the agile methods in practice - Literature survey & review. International Conference on Computing for Sustainable Global Development (INDIACom), (págs. 556-560). New Delhi.

Srinivasan, J., Dobrin, R., & Lundqvist, K. (2009). 'State of the Art' in Using Agile Methods for Embedded Systems Development. 33rd IEEE International Annual Conference on Computer Software and Applications, (págs. 522-527). Seattle.

Stankovic, D., Nikolic, V., Djordjevic, M., & Cao, D.-B. (June de 2013). A survey study of critical success factors in agile software projects in former Yugoslavia IT companies. Journal of Systems and Software, 86(6), 1663-1678.

Stettina, C., & Heijstek, W. (2011). Five Agile Factors: Helping Self-management to Self-reflect. 18th European Conference on Systems, Software and Service Process Improvement, EuroSPI 2011, (págs. 84-96). Roskilde: EuroSPI .

Sulayman, M., & Mendes, E. (2009). A Systematic Literature Review of Software Process Improvement in Small and Medium Web Companies. International Conference on Advanced Software Engineering and Its Applications, ASEA. 59, págs. 1-8. Jeju Island: Springer.

Zhang, X., Hu, T., Dai, H., & Li, X. (2010). Software Development Methodologies, Trends, and Implications. Southern Association for Information Systems Conference (pág. Paper 31). SAIS

Notas Biográficas:

Marilyn Sihuay es Titulada y Colegiada en Ingeniería de Sistemas, con más de 17 años de experiencia como Project Manager, Test Manager, Analista de Calidad de software y Analista de Sistemas. Con amplios conocimientos en Procesos de Implementación y Aplicación de Mejores Prácticas en SCRUM, ITIL, ISTQB, CMMI, Gestión de Proyectos PMP, ISO 9001:2000, Cloud Computing. Líder de proyectos clásicos y ágiles, de mejora continua y desarrollo de software, con amplio manejo del presupuesto, estrategia, planificación, ejecución y cierre; cumpliendo los objetivos del proyecto con éxito en tiempo y calidad, compromiso, honestidad, perseverancia y sentido de urgencia.

Abraham Dávila es investigador y profesor principal de la Pontificia Universidad Católica del Perú (PUCP) desde el 2000. Dirige y es investigador principal del proyecto ProCalProSer (2013-2016 Fase I y 2017-2018 Fase II) y miembro fundador de GIDIS-PUCP. Posee el grado de bachiller en ciencias con mención en Ingeniería Mecánica y magister en Informática por la PUCP. Miembro del grupo de trabajo de la ISO/IEC que elabora la norma ISO/IEC 29110. Sus principales áreas de interés son calidad en informática (a nivel de proceso software, productos y gestión de servicios) y educación en ingeniería de software.

Marcelo Pessoa es Ingeniero Electrónico, tiene una maestría, doctorado y libre docencia por la Universidad Politécnica de San Pablo – Brasil. Profesor del Dpto. de Ingeniería de la Producción desde 1987. Tiene experiencia e investigaciones en las áreas de sistemas de operaciones, computación, electrónica, telecomunicaciones y automatización. Miembro de la Comisión del Estudio de

Procesos del Ciclo de Vida de Software de la ABNT en el área de Ingeniería de Software para la elaboración de normas nacionales e internacionales en la ISO. Coordinador del CEGPTI Curso de Especialización en Gestión de la TI desde 2008. Coordinador del curso Análisis de Negocio basado en BABOK. Fue Director-Presidente de la Fundación Carlos Alberto Vanzolini en el periodo 2002-2005 y después miembro del Consejo Curador de las misma Fundación. Actualmente es vice-Jefe del Dpto. de Ingeniería de la Producción 2015/2017. Trabaja como investigador en los laboratorios eLabSoft donde realiza investigación sobre Fábrica de Software y Proceso Software. También es investigador de LADOS (Laboratorio de Análisis, Desarrollo y Operaciones de Sistemas donde desarrolla investigación sobre sistemas tecnológicos avanzados, combinando software y servicios tecnológicos para la generación de innovaciones, desarrollo de nuevos productos y servicios tecnológicos para la re-estructuración de los procesos productivos.

Recibido 8 Feb 2018 *ReCIBE, Año 7 No. 1, Mayo 2018*
Aceptado 3 Abr 2018

Adoptability of Test Process Models: ISO/IEC 29119, TMMI y TPI from the small organization perspective

Adoptabilidad de Modelos de Proceso de Pruebas: ISO/IEC 29119, TMMI, TPI, desde la perspectiva de una pequeña organización

Cecilia García[1]
garcia.cecilia@pucp.edu.pe

Karin Meléndez[2]
kmelendez@pucp.edu.pe

Abraham Dávila[2]
abraham.davila@pucp.edu.pe

[1] Escuela de Graduados, Pontificia Universidad Católica del Perú, Lima, Perú
[2]Departamento de Ingeniería, Pontificia Universidad Católica del Perú, Lima, Perú

Abstract: Testing is considered to be an important stage in the software development process. Therefore, there are different proposals regarding how to perform testing activities at a project level and at organizational level. However, in the context of very small organizations where there are ad-hoc proposals in software and system engineering, the ease of the adoption of influential models such as ISO/IEC 29119, TMMI and TPI is not yet clear. The objective of this work is to compare these models with the characteristics of the resources and finances of small organizations established in the standardization guidelines of ISO. This study conducted a comparative analysis of test process models with respect to the characteristics of very small organizations. A comparative chart was obtained with the answers of the analysis of each model with respect to the criteria considered. It can be inferred from the analysis, that the assessed test process models are not easily adoptable by very small organizations.

Keywords: TMMi, TAMAR, ISO/IEC 29119-2, ISO/IEC 33063, TPI, MPS, software test process model, comparative analysis.

Resumen: La prueba de software es una etapa importante en el proceso de desarrollo de software. Por lo tanto, hay diferentes propuestas sobre cómo realizar actividades de prueba a nivel de proyecto y a nivel organizacional. Sin embargo, en el contexto de organizaciones muy pequeñas donde existen propuestas ad hoc en software e ingeniería de sistemas, la facilidad de adopción de modelos influyentes como ISO/IEC 29119, TMMI y TPI aún no está clara. El objetivo de este trabajo es comparar estos modelos desde las características de los recursos y las finanzas de las pequeñas organizaciones establecidas en las pautas de estandarización de ISO. Este estudio realizó un análisis comparativo de modelos de procesos de prueba con respecto a las características de organizaciones muy pequeñas. Se obtuvo un cuadro comparativo con las respuestas del análisis de cada modelo con respecto a los criterios considerados. Del análisis se puede inferir que los modelos evaluados de procesos de prueba no son fácilmente adoptables por organizaciones muy pequeñas.

Palabras clave: TMMi, TAMAR, ISO/IEC 29119-2, ISO/IEC 33063, TPI, MPS, modelo de proceso de prueba de software, análisis comparativo.

1. Introduction

Software testing is a stage of the development process, the application of which is complex in different projects (NIST, 2002). The concept of testing has evolved over time, it developed from a debugging-oriented activity, where the majority of organizations did not differentiate between testing and debugging; to one aimed at preventing (Gelperin & Hetzel, 1988), (Luo, 2001). On the other hand, according to (Meerts, 2016), Myers first publications (Myers, 1979) prepared the scenario for the modern practices of software testing. Trends in the software market impact on the discipline of testing by introducing new elements, such as: agile testing, mobile testing, crowdtesting, test factories, automation testing, context-driven testing (Kulkarni, 2006), among others.

Software testing is also considered a key approach for risk mitigation in the development of software (Swinkels, 2000). However, there are different proposals, such as standards, process models, techniques and process descriptions in various industries that describe how to perform testing at a project level and at an organizational level (García & Dávila, 2012). In a systematic literature review of test process models, (García, Dávila, & Pessoa, 2014) 23 models were found and it was noted that the most reported and used ones are TMMi, proposed by the TMMi Foundation (TMMI Foundation, 2016) and TPI (Visser, et al., 2013), a proprietary process model of the company Sogeti. Finally, the recently published standard ISO/IEC 29119 (ISO/IEC, 2013) can also be included in this group.

On the other hand, the definition of small organizations is not universally accepted, therefore, various institutions around the world, governmental or not, have developed their own definition based on criteria such as number of employees, level of sales and investment, among the most common (ECS, 2010). Some examples of this variety of definitions are: (i) United States. International Trade Commission refers to small and medium-sized enterprises (SMEs) as firms with less than 500 U.S.-based employees (USITC, 2010), or (ii) Official Journal of the European Union (European Union, 2003), which determines a limit of 250 employees for the category of micro, small and medium-sized enterprises (SMEs).

In addition, as noted in (ISO/IEC, 2011), small and micro-businesses are very superior in number to the rest. The software industry is no stranger to this reality and many of the software companies are small (Cernant, Norman-Lopez, & Duch T-Figueras, 2014), (United Nations, 2012), (Laporte, Houde, & Marvin, 2014). In this context, the new family of standards ISO/IEC 29110 - Systems and Software Life Cycle Profiles and Guidelines for Very Small Entities (VSE), defines a small organization, as an enterprise, an organization, a department or a project having up to 25 people (ISO/IEC, 2011).

VSEs have been recognized as an important factor in the economics of the software industry, either producing independent software components or components integrated into larger systems (Laporte, Houde, & Marvin, 2014). VSE needs are often neglected in the development of models and standards recognized in the market. That translates into lack of simplicity, low flexibility and difficulty among other things, for the VSE (ISO/IEC, 2011). It also implies high costs for the adoption of models (ISO/IEC, 2011), which could lead to the exclusion of the VSE from the market and to a distortion of fair competition (ECS, 2010), (Laporte, Houde, & Marvin, 2014).

In the context of the international standardization, guide has been published for writing standards taking into account the needs of micro, small and medium-sized enterprises (ECS, 2010) (ISO/IEC, 2016) and in the field of information technology (IT), several documents of the series ISO/IEC 29110 representing profiles of life cycle processes for systems and software domains, ad-hoc for VSE, has also been published (ISO/IEC, 2011). In particular, the ISO/IEC 29110-5-1-2 is a profile that presents two processes based on the ISO/IEC 12207 but adapted to the reality of the VSE (ISO/IEC, 2011) for software development. The same is the case with the ISO/IEC 29110-5-6-2, which is based on the ISO/IEC 15288 systems (ISO/IEC, 2014) domain. At a technical level the ISO/IEC 29110-5-1-2 covers a small set of practices of the ISO/IEC 12207, which can be considered as "indispensable" to ensure the success of small projects. This new effort of the ISO opens the possibility of extending the analysis of the need for models ad-hoc for other processes of software development where there are specialized models (Dávila, 2012). It also recognizes the need for the VSE to have certifications of any kind (in particular process certifications), which represents better opportunities for them to enter the market.

This work aims to perform a comparative assessment of the most representative test process models TMMI, TPI and ISO/IEC 29119 using relevant assessment criteria from the perspective of the VSE. The article is organized as follows: Section 2 presents a description of each of the models to compare; Section 3 shows the criteria used, comparative analysis is shown in Section 4, and finally Section 5 presents a final discussion and future work.

2. Test Process Models

In the software industry, from the perspective of processes, there are two concepts to consider: (i) a process model (of the life cycle) is a framework of processes, activities and artifacts that facilitates communication and understanding of the different ways of organizing them (ISO/IEC, 2010); and (ii) organizational maturity is understood as the degree to which the processes that contribute to the current and future business goals have been implemented consistently across an organization (ISO/IEC, 2015). These concepts provide a

frame to the technological development from the perspective of processes and must also be considered for software testing discipline. In a previous work of the authors (García, Dávila, & Pessoa, 2014) various process models of software testing were identified, of which models ISO/IEC 29119, TMMi and TPI are the most representative and are described below:

2.1. ISO/IEC/IEEE 29119-2: Test Processes

The family of standards ISO/IEC/IEEE 29119 is developed by the ISO/IEC in cooperation with IEEE (ISO/IEC, 2013). The parts of the standard published to date are four: (i) concepts (ISO/IEC, 2013), (ii) test process model (ISO/IEC, 2013), (iii) documentation (ISO/IEC, 2013), and (iv) testing techniques (ISO/IEC, 2015). In particular, Part 2 refers to a test process model that describes itself as generic, which can be used as part of any software life cycle model, applied to any test type and to organizations of any size (Reid, 2013). Organizations may not need to use all these processes, so process implementation usually involves a selection of the most suitable set of processes for the specific organization or project (ISO/IEC, 2013) (ISO/IEC, 2008), i.e. an adaptation procedure. However, the commercial pressure to obtain certifications means having to meet certain requirements. There are two ways in which an organization can comply with Part 2 of the standard: full compliance (proving that all the "musts" of all processes are completed), or adapted compliance (the selection and use of certain processes is justified and it is proven that all the "musts" of the selected processes are completed) (ISO/IEC, 2013) but usually this is determined by who offers the certification.

ISO/IEC 29119 includes eight processes, which implies 89 base practices and 40 artifacts according to ISO/IEC 33063 (ISO/IEC, 2015). It is organized into 5 levels of process capability: Performed, Managed, Established, Predictable and Innovating. ISO/IEC 29119 can be seen as a staged or continuous model. The architecture and assessment are presented below.

2.1.1. Architecture of ISO/IEC 29119-2

ISO/IEC 29119-2 classifies the processes into organizational test process group, test management process group and dynamic test process group (ISO/IEC, 2013). Each process is described in terms of purpose, outcome, activities, tasks and information items, as indicated in the guide for processed description ISO/IEC TR 24774 (ISO/IEC, 2013).

2.1.2. Assessment of ISO/IEC 29119-2

According to ISO/IEC 15504-2 (ISO/IEC, 2004) and ISO/IEC 33001 (ISO/IEC, 2015), a process reference model (PRM), as the one described in ISO/IEC/IEEE 29119-2 requires a process assessment model (PAM). A PAM (ISO/IEC, 2004)

allows a reliable and consistent assessment of process capability. In addition, a PAM extends a PRM, incorporating indicators to be considered when interpreting the intent of the PRM. ISO/IEC 33063 (ISO/IEC, 2015) provides an example of a PAM which takes as its basis the PRM i.e. ISO/IEC/IEE29119-2, for using in performing a conformant assessment in accordance with the requirements of ISO/IEC 33002 (ISO/IEC, 2015). ISO/IEC 33063 determines a degree of process capability from a set of capability indicators applicable to the PRM under a formal and rigorous process (ISO/IEC, 2015).

2.2. Test Process Improvement TPI®

The process improvement methodology TPI® was developed in 1998 by Koomen and Pol (Tim & Martin, 1999) based on practical knowledge and experience of the test process of Sogeti (Sogeti Web Site, 2015). In 2009 Sogeti publishes TPI Next® (Visser, et al., 2013), an improved version of the previous model. The model offers a viewpoint on the maturity of the test processes within the organization. Based on this understanding the model helps to define gradual and controllable test process improvement steps (Visser, et al., 2013). Below is the architecture and how to evaluate the model.

2.2.1. Architecture of TPI

The TPI model defines 16 Key areas, which are not processes in the strict sense, but together they cover all aspects of the testing process (Visser, et al., 2013) . The key areas are classified into three groups: Stakeholder Relations, Test Management and Test Profession (Visser, et al., 2013) . Every key area can be classified into one of four levels of maturity: Initial, Controlled, Efficient and Optimized. To verify the classification into levels, one or more checkpoints are assigned to each level. If a test process passes all the checkpoints of a certain level, then the process is classified at that level. TPI has 157 checkpoints and 13 pre-defined improvement steps, also called base clusters which have between 10 and 14 checkpoints.

For each key area, the model presents (Visser, et al., 2013): i) an overview of each maturity level of that key area; ii) checkpoints by level; iii) enablers by level; and iv) improvement suggestions by maturity level of the key area. TPI is a staged model; its architecture lays out the steps for improvement or base cluster to increase process maturity. In this type of representation it is assumed that all key areas have the same relevance. For an organization to use the model in a continuous way, a prioritization of key areas based on business goals is required. The checkpoints of the base cluster are re-arranged producing new clusters, and verifying that the dependencies of checkpoints are not transgressed; otherwise another re-arrangement of checkpoints is needed. Finally, each new cluster load is balanced to end up with a similar number of checkpoints.

2.2.2. Assessment of TPI

The testing process in the organization is assessed using the same elements presented in the previous section. TPI proposes a four level maturity model (Visser, et al., 2013): Initial, Controlled, Efficient and Optimized. These maturity levels are achieved in stages, when the 16 key areas are at that maturity level (Visser, et al., 2013). A particular level of maturity can be reached only if the previous level of maturity has been reached (Visser, et al., 2013). To achieve each level, TPI defines the element Cluster (11 clusters in total), in order to establish small improvement steps to the desired level of maturity; and where each cluster consists of a fixed number of checkpoints from several key areas (Visser, et al., 2013).

The model structure allows the organization to implement TPI in a staged or continuous way, the latter means that it is expected that certain key areas contribute more to the business goals than others (Alone & Glocksien). For this, a preliminary analysis of prioritization is required, so the improvement steps are not those set by the model by default, but according to the needs of each organization.

2.3. TMMi: Test Maturity Model Integration

Test Maturity Model Integration is a maturity model developed by the TMMi Foundation (TMMI Foundation, 2016) . According to (Tmmi Foundation, 2012), the main differences with other models are its independence, its compliance with international standards and its complementary relationship with the CMMI maturity model; the latter being a model of software process maturity well positioned in the IT industry (Rasking, 2011).

The terminology used in TMMi is based on the glossary of terms established by ISTQB (International Software Testing Qualifications Board) (ISTQB, 2016), so in this model the tests are considered a dynamic and static activity; and, unlike ISO/IEC 29119-2, it does include static verification techniques such as Reviews. TMMi is a process model with a staged representation: 16 process areas, 173 specific practices, and 12 generic practices organized into five maturity levels: Initial, Managed, Defined, Quantitatively Managed and Optimized. Currently TMMi has not posted a continuous representation of the model. Below is the architecture of TMMi and how to assess it:

2.3.1. Architecture of TMMi

The TMMi model consists of several components which are categorized (Tmmi Foundation, 2012) as: i) required.- components necessary to achieve a certain level of maturity; ii) expected.- components usually performed by organizations to meet a specific process area, although an acceptable alternative can also be

recognized by the evaluator; and iii) informative.- such as sub-practices, work products examples, notes, references, etc., to provide ideas to the organization on how to address the required components.

To TMMi, a maturity level is a degree of quality of a testing process of the organization and defines it as an improvement step the organization must follow in stages (Tmmi Foundation, 2012). To reach a level you must meet all the objectives (general and specific) of the desired level and previous levels (Tmmi Foundation, 2012). The model has five levels of maturity that contain fixed process areas. All organizations start at Maturity Level 1 since this does not define objectives to be met (Tmmi Foundation, 2012).

2.3.2. Assessment of TMMi

In 2014, the TMMi Foundation published TAMAR TMMi Assessment Method Application Requirements (Tmmi Foundation, 2009), a set of requirements for implementing an evaluation method based on the process model TMMi. This is an adaptation of the process assessment standard ISO/IEC 15504-2 (ISO/IEC, 2004).

Assessments methods that comply with TAMAR (Tmmi Foundation, 2009) can be accredited by TMMi Foundation, if they meet those requirements. Such methods could be used to identify the strengths and weaknesses of the current testing process and determine rankings based on the TMMi maturity model.

3. Criteria for the Comparative Analysis

In the context of international standards, ISO (ISO/IEC, 2016) as well as the European Committee for Standardization (ECS, 2010) has published a guide providing advice and recommendations to write standards that consider the needs of small businesses. Both documents can be downloaded for free. A Very Small Entity (VSE) is defined as an enterprise, an organization, a department or a project formed by up to 25 people, according to ISO/IEC/29110 (ISO/IEC, 2011). In addition, ISO/IEC 29110-4 (ISO/IEC, 2011) contains in section 6.3 a set of characteristics, needs and suggested competencies for the basic profile of a small organization that develops software. This approach is used to develop new documents related to the series ISO/IEC 29110 (ISO/IEC, 2011).

The characterization of VSEs presented in 29110-4 (ISO/IEC, 2011) is classified in four categories: (i) Finance and Resources, (ii) Customer Interface, (iii) Internal Business Processes, and (iv) Learning and Growth. This article introduces a progress on the analysis conducted for the category Finance and Resources. Therefore, information related to other categories has been intentionally omitted. Table 1 shows the criteria and the characteristics that have been considered as

the basis for this analysis. The codes that identify each characteristic in the table are used throughout this work to associate or reference them in the analysis.

4. Comparative Analysis

The comparative analysis is a technique introduced in Psychology by Ragin and can be used in Software Engineering for the synthesis of information (Genero Bocco, Cruz-Lemus, & Piattini Velthuis, 2014). In our case, the analysis is conducted by setting a question about the degree of adoption (adoptability) of the discussed models in the context of a VSE, and responding from the model features. The responses have been obtained by expert judgment of software improvement professionals. The "Yes/No" value means that the standard or model meets or does not meet the characteristic. The "NA" value means that it was not enough evidence to make the decision. The analysis is carried out for the criteria and the characteristics listed in Section 3. Summary tables are presented for each criterion and subsequently, argument tables.

Criteria	Characteristics
c1. Finance and Resources	c11. Small number of professionals (up to 25 engineers)
	c12. Short term cash flow of each project may be critical for a VSE
	c13. Low budget projects that last a few months and involve few people to develop small products
	c14. Dependence on successful project completion within schedule and budget
	c15. Limited internal resources to perform management, support and organizational processes like risk management, training, quality management, process improvement and reuse

Tabla 1. *Criteria and characteristics adapted from ISO/IEC 29110 (ISO/IEC, 2011)*

Table 2 provides an overview of the analysis results of the models with respect to the questions of adoptability of each criterion. The first column of this table (labeled "C") refers to the characteristics, the second column contains the question of analysis and the remaining columns, grouped in pairs, show the Yes / No answers to the questions concerning the analysis of each model. Tables 3, 4, 5, 6 and 7 contain the justification for the criteria c11, c12, c13, c14 and c15, respectively. Each justification table has a front row with the question and a summary of ideas that apply to that feature.

C	Question:	29119 (i)		TPI (j)		TMMI (k)	
Is model X easily adoptable considering that a VSE							
c11	has a small number of professionals?	No	i11	No	j11	No	k11
c12	has a limited and short-term cash-flow?	No	i12	No	j12	No	k12
c13	has low budget projects that last a few months and involve few people to develop small products?	No	i13	No	j13	No	k13
c14	depends on successful project completion within schedule and budget?	No	i14	No	j14	No	k14
c15	has limited internal resources to perform several organizational and support processes such as risk management, training, process improvement, etc.?	No	i15	Yes	j15	Yes	k15

Table 2. *Comparison of the Criterion Finance and Resources*

Qstn	Is model X easily adoptable considering that a VSE has a small number of professionals?
r11	Several studies such as (Dyba, 2003), (Dyba, 2005) and (Dyba, 2002) conclude that, based on empirical evidence taken from the industry and literature, the lack of resources to be assigned to Software Process Improvement (SPI) is a barrier to their success, regardless of the process model used. This barrier is even more important for a VSE, which due to its nature, has a small number of professionals.
i11	ISO/IEC 29119 has a high number of process elements (see Section 2.A) and a VSE has a limited number of technical staff devoted to software development. For a VSE, the latter means having a low number of resources to allocate to SPI (almost zero or null) and, as a result, this affects their adoptability.
j11	TPI presents a large number of model elements (see Section 2.B). This task requires a high effort considering that a VSE has a limited number of professionals, dedicated primarily to software development.
k11	Of the 3 models, TMMI has the highest number of process elements (see Section 2.C). This is unfavorable for adoption, considering that the technical staff is intensively engaged in the development of software. However, the similarity of its structure with CMMi may favor in some cases its adoption if the VSE is familiar with it.

Table 3. *Qualification of Criterion c11*

Qstn	Is model X easily adoptable considering that a VSE has a limited and short-term cash-flow?
r12	Overall costs associated with the implementation of a model or standard processes can vary based on: i) the cost of the model or standard and the training in its use, ii) the cost of external consultancy, iii) the cost of the technology, iv) the cost of the time invested by the employee of the organization, v) the cost for certification, and vi) if the evaluation is formal or informal.
i12	ISO/IEC 29119 standard is not free. To date, a quality seal or certification of their process based on this standard is not offerd to organizations. No pricing information for external consulting has been found.
j12	TPI is a proprietary model from Sogeti (Sogeti Web Site, 2015), which recently offers a quality seal or certification (Sogueti, 2016). No information on external consultancy fees has been found. However since it is a very specialized model and not well known in some contexts, it is expected to be more expensive compared to other models which are in ample supply such as ISO 9001.
k12	TMMi is a model available for free download. At the time of writing the article, 12 companies have been found accredited to issue TMMi certifications (Foundation, 2016). However, the costs are also significant, as mentioned in r12.

Table 4. *Qualification of Criterion c12*

5. Discussion of Results and Future Work

This comparative analysis of the considered test process models reveals directly (and qualitatively) that the considered test process models are not aligned to the underlying needs of a small business with respect to the characteristics of the category Resources and Finance. The main reason is substantially associated to the size that very small entities exhibit (less than 25 professionals) as defined by ISO/IEC 29110-1.

Taking the above-mentioned analysis into consideration, it can be established that the compared models present a high level of abstraction. That is, they are generic models that attempt to cover all the needs of the testing process for different types of tests, different software life cycles and different domains. Therefore, they do not provide a high level of specificity. Whereas this level of abstraction enables adaptability to different contexts, in practice, it turns out to be more complicated for VSEs due to the constraints they present, as reflected in the analysis. As it has occurred in other contexts, the VSEs needs have not been considered explicitly.

Qstn	Is model X easily adoptable considering that a VSE has low budget projects that last a few months and involve few people to develop small products?
r13	In connection with criterion r11 (Table 3), models with more resulting products become more difficult to adopt.
i13	The ISO/IEC 29119 establishes the adoption of 40 artifacts, an action that involves more time in software development projects, specially in small projects where the increase of workload is more noticeable.
j13	In the same way, having so many artifacts to produce on TPI can be excessive for small projects (for example 2 or 3 people, for few months). This problem requires the VSE to have to go through the simplification of the model, as for example, considering the same artifact as an evidence of more than one checkpoint or reducing the documentation among other actions.
k13	TMMi has a large number of practices increasing the number of hours to be allocated to be able to evidence them. This may impose an overload on small projects. It is likely that the organization has to define an adoption method of TMMi for small projects. However, this would be an additional work that would have to be generated by the VSE.

Table 5. *Qualification of Criterion c13*

One aspect to highlight is that these three models describe themselves as risk-based testing models. The concept of risk is used to define and limit the testing process in the context of scope, cost and schedule. This is an aspect from which the VSE could benefit but it is not easy to perceive since the models are extensive in terms of number of documents and practices that should be covered.

Qstn	Is model X easily adoptable considering that a VSE depends on successful project completion within schedule and budget?
r14	In general the adoption of best practices, standards or similar formalities, always contribute to the productivity of the organizations (and to the compliance with various attributes of the project). However, the adoption of a new model involves usual activities such as PDCA (plan-do-check-act) that are done against the processes that people perform, increasing the risk of not meeting the deliverables. It is also correct to indicate that the degree of impact also depends on other factors such as the level of adaptability to change (Dyba, 2000).
i14	The workload that the 40 artifacts or the 89 model practices (for small projects) may involve can lead to a situation where compliance with the schedule or budget is very difficult.
j14	TPI proposes a structured way to improve the process as follows: i) raising awareness, ii) determining objectives, scope and focus of improvement, iii) assessing the current situation, iv) establishing a plan of action, v) implementing improvement actions and vi) evaluating and re-directing. These activities represent a workload for these roles and consequently, impact the schedule of the development project.
k14	Of the three models, TMMi is the one with more specific practices. Depending on the scope of the process improvement program, the amount of man-hours invested in the implementation of these new procedures would be increased proportionally to the number of hours of practice indicated by the model.

Table 6. *Qualification of Criterion c14*

Without impairing what has been stated on the generic aspect of the models, a more appropriate approach for a VSE can be a gradual adoption where a simplified and inexpensive test process model is available for VSEs in early stages. This model should allow to experience the progress quickly and to provide short-term benefits, which can enable the VSE to adopt a culture of quality and continuous improvement on aspects related to software testing processes. Subsequently, the VSE can fully adopt, as a result of its evolution (growth in people and resources), a test process model as the ones presented in the analysis.

Whereas VSE has a minimal infrastructure and limited resources, they can focus on process improvement. The existence of appropriate implementation guides, assessment tools and templates, among others, could be an important enabler for the adoption of the analyzed models. In that regard, TPI offers free downloadable tools such as evaluation matrix and templates presented as a toolkit. For the other two models, ISO/IEC 29119 and TMMi, these tools must be developed by the VSE itself or be made available by communities of interest that support these models.

The need for a test process model suitable for VSEs may be experienced in small contexts such as CMMi (Garcia, 2005), proprietary models as MPS.Br in Brazil (Softex, 2016), MoProSoft in Mexico (NYCE, 2016). At a ISO level, work has been carried out on lifecycle profiles for VSEs that develop software (ISO/IEC, 2011) (ISO/IEC, 2011) (ISO/IEC, 2014). All this indicates a need to make available more appropriate processes for VSEs and supports the view that current models are perceived as costly and complex. A current case of a process model specialized in software testing for VSE is MPT-Br from Brazil (Carvalho,

Wanderley, Carneiro, & Honório, 2012). However, there is little evidence of its benefits and results in literature. Another case is the Spanish model TestPAI which claims that it attempts to solve the problem of small and medium Spanish companies that demand a framework for improve testing processes in a simple and inexpensive way, and which proposes a test process model compatible with CMMI (Sanz, Saldaña, Garcia, & Gaitero, 2008). In the case of the latter, there is not sufficient empirical evidence of their results, except for the model validation case study presented by the same authors (Sanz, Saldaña, Garcia, & Gaitero, 2008).

The discussion presented in this work, from the perspective of a VSE, has been developed around one of ISO (ISO/IEC, 2011) criteria. It is intended as part of this project to complete the analysis in the rest of the criteria and the characteristics. It also opens the opportunity to propose a model of software testing process specifically oriented to VSE and to evaluate its feasibility and usefulness in the context of the industry itself. The model to be developed should be based on the ISO/IEC 29119 (software testing) and inspired by the philosophy underlying the ISO/IEC 29110 (profiles VSE life cycle).The expectation that guides this idea is that the VSE can use a more adoptable model with short-term benefits. It also will allow them to take an interest in other models such as the ISO/IEC 29119.

Qstn	Is model X easily adoptable considering that a VSE has limited internal resources to perform several organizational and support processes such as risk management, training, process improvement, etc.?
r15	In (Dyba, 2003) it is suggested that, given resource constraints for implementing SPI, the VSE should capitalize on employee participation and exploration of new knowledge, complementing the use of formal processes with interpersonal and informal coordination of practices. The informative elements of the model contribute to this purpose, favoring its adoption. On the other hand, according to (Dyba, 2000) 67% of project managers want guidance on how to implement the process of improvement rather than the practices of the reference model. Furthermore, in (ECS, 2010) the inclusion of implementation guidelines for complex norms that cannot be simplified is recommended.
i15	ISO/IEC 33063 includes notes and examples for some practices of the process and refers to another part of the standard: 29119-3: Test Documentation to define the elements of the work products. However, these may be insufficient. For example, in order to achieve level 1 of capacity for the organizational test process, the organization requires to develop, monitor compliance and maintain organizational test specifications. Nevertheless, 29119 does not provide sufficient informative elements such as notes and examples on how to implement these practices required to achieve the level 1 of this process. In addition, the family of standards ISO/IEC 29119 does not include a deployment guide. However, ISO has other standards.
j15	With respect to the amount of informative elements of the process, TPI provides for each pair of key-area / maturity-level, suggestions for improvement to meet the checkpoints. According to (Visser, et al., 2013), these suggestions for improvement are equivalent to base practices, generic practices and work products of the ISO models as well as to the specific or generic practices of TMMi. However, in the TPI model,work products are not clearly determined. They should be easier to identify encouraging the VSE to find the information required to achieve a given level of maturity in a simpler way. On the other hand, the wording of the model does not identify a 1:1 relationship between a suggestion for improvement and a checkpoint. For example, the key area Test Organization has checkpoints that are required to achieve a controlled maturity level. However, the suggestions for improvement are not enumerated and there is no mapping for each of the checkpoints. A strength of the model structure is the inclusion of the elements Maturity Matrix and Enablers. The Maturity Matrix is a good way of visual communication of the current and desired state of the process. Enablers are recommendations on how to relate the testing process with other processes of the software life cycle. In (Visser, et al., 2013) further examples of model adaptation are shown in different situations: 1) iterative development, 2) agile development, 3) multiple testing processes in distributed organizations, 4) management of testing services 5) outsorcing of testing services, etc; although each of these examples do not have the level of detail required for easy adoption.
k15	TMMi provides informative elements such as examples of work products. This may help a VSE to have not only processes definitions, but also an explorative learning (Dyba, 2002). For example, a specific practice of the process Test Strategy and Policy which belongs to the maturity level called Managed is to "distribute the test strategy to stakeholders". For this specific practice the model lists examples on how to do the distribution: 1) document it in a quality system, 2) present it in project meetings, 3) reference it in publications in murals on the wall, 4) make it part of the introduction program to the department, and 5) facilitate its access throught a web portal.

Table 7. *Qualification of Criterion c15*

Acknowledgments

This work is framed within ProCal-ProSer Contract N° 210-FINCYT-IA-2013 (Innovate Perú): “Productivity and Quality Relevance Factors in small software development or service organizations adopted ISO standards”, the Software Engineering Development and Research Group and the Department of Engineering of Pontificia Universidad Católica del Perú.

References

Alone, S., & Glocksien, K. (s.f.). Evaluation of Test Process Improvement approaches: An industrial case study. Recuperado el 14 de abril de 2016, de University of Gothenburg: https://gupea.ub.gu.se/bitstream/2077/38986/1/gupea_2077_38986_1.pdf

Carvalho Cavalcanti Furtado, A., Wanderley Gomes, M., Carneiro Andrade, E., & Honório de Farias Junior, I. (27-29 Aug. 2012). MPT.BR: A Brazilian Maturity Model for Testing. 2012 12th International Conference on Quality Software. Xi'an, Shaanxi: IEEE.

Cernant, L., Norman-Lopez, A., & Duch T-Figueras, A. (September de 2014). SMEs are more Important than you think! Challenges and Opportunities for EU Exporting SMEs. DG TRADE Chief Economist.

Dávila, A. (2012). Nueva aproximación de procesos en pequeñas organizaciones de Tecnología de Información. Magazine IEEEPeru. La revista tecnológica de la Sección Perú de IEEE, Vol 1(Nro 2), pp 30-31.

Dyba, T. (December de 2000). An Instrument for Measuring the Key Factors of Success in Software Process Improvement. Empirical Software Engineering, págs. 357-390.

Dyba, T. (December de 2002). Enabling Software Process Improvement: An Investigation of the Importance of Organizational Issues. Empirical Software Engineering, 7, págs. 387-390.

Dyba, T. (2003). Factors of software process improvement success in small and large organizations: an empirical study in the scandinavian context. Proceedings of the 9th European software engineering conference held jointly with 11th ACM SIGSOFT international symposium on Foundations of software engineering (ESEC/FSE-11) (págs. 148-157). New York, USA: ACM.

Dyba, T. (Mayo de 2005). An empirical investigation of the key factors for success in software process improvement. IEEE Transactions on Software Engineering, 31, 410-424.

ECS. (2010). Guidance for writing standards taking into account micro, small and medium-sized enterprises (SMEs) needs. European Committee for Standardization (CEN) and European Committee for Electrotechnical Standardization (CENELEC). Brussels: European Committee for Standardization.

European Union. (20 de May de 2003). Commission Recommendation of 6 May 2003 concerning the definition of micro, small and medium-sized enterprises. Official Journal of the European Union, 46, págs. p. 36–41.

Foundation, T. (s.f.). Find an accredited Assessor. Recuperado el 14 de Abril de 2015, de http://www.tmmi.org/?q=assesor

García, C., & Dávila, A. (2012). Mejora del proceso de pruebas usando el modelo TPI® en proyectos internos de desarrollo con Scrum. Caso Alfa-Lim.

García, C., Dávila, A., & Pessoa, M. (2014). Test process models: Systematic literature review. Software Process Improvement and Capability Determination (págs. 84-93). Springer.

Garcia, S. (2005). Thoughts on Applying CMMI in Small Settings. Pittsburgh: Carnegie-Mellon University. Software Engineering Institute.

Gelperin, D. H., & Hetzel, B. (June de 1988). The growth of software testing. Communications of the ACM., 31(6), 687-695.

Genero Bocco, M., Cruz-Lemus, J. A., & Piattini Velthuis, M. (2014). Métodos de investigación en Ingeniería de Software. Ra-Ma Editorial.

ISO/IEC. (2004). ISO/IEC 15504-2:2003/Cor 1:2004 Information technology -- Process assessment -- Part 2: Performing an assessment. Geneva, Switzerland: International Organization for Standardization.

ISO/IEC. (2008). ISO/IEC 12207:2008 Systems and software engineering -- Software life cycle processes. Geneva, Switzerland: International Organization for Standardization.

ISO/IEC. (2010). ISO/IEC TR 24774:2010 Systems and software engineering -- Life cycle management -- Guidelines for process description. Geneva, Switzerland: International Organization for Standarization.

ISO/IEC. (2011). ISO/IEC 29110-1: Software engineering — Lifecycle profiles for Very Small Entities (VSEs) — Part 1: Overview. Geneva, Switzerland: International Organization for Standardization.

ISO/IEC. (2011). ISO/IEC 29110-4-1:2011 Software engineering -- Lifecycle profiles for Very Small Entities (VSEs) Part 4-1: Profile specifications: Generic profile group. Geneva, Switzerland: International Organization for Standardization.

ISO/IEC. (2011). ISO/IEC TR 29110-5-1-2:2011 Software engineering -- Lifecycle profiles for Very Small Entities (VSEs) -- Part 5-1-2: Management and engineering

guide: Generic profile group: Basic profile. Geneva, Switzerland: International Organization for Standardization.

ISO/IEC. (2013). ISO/IEC/IEEE 29119-1:2013 Software and systems engineering — Software testing — Part 1: Concepts and definitions. Geneva, Switzerland: International Organization for Standarization.

ISO/IEC. (2013). ISO/IEC/IEEE 29119-2: Software and systems engineering — Software testing — Part 2: Test process. Geneva, Switzerland: International Organization for Standarization.

ISO/IEC. (2013). ISO/IEC/IEEE 29119-3:2013 Software and systems engineering — Software testing — Part 3: Test Documentation. Geneva, Switzerland: International Organization for Standardization.

ISO/IEC. (2014). ISO/IEC TR 29110-5-6-2:2014 Systems and software engineering -- Lifecycle profiles for Very Small Entities (VSEs) -- Part 5-6-2: Systems engineering -- Management and engineering guide: Generic profile group: Basic profile. Geneva, Switzerland: International Organization for Standarization.

ISO/IEC. (2015). ISO/IEC 33001 Information technology — Process assessment — Concepts and terminology. Geneva, Switzerland: International Organization for Standardization.

ISO/IEC. (2015). ISO/IEC 33002:2015 Information technology -- Process assessment -- Requirements for performing process assessment. Geneva, Switzerland: International Organization for Standardization.

ISO/IEC. (2015). ISO/IEC FDIS 33063 Information technology -- Process assessment -- Process assessment model for software testing. Geneva, Switzerland: International Organization for Standardization.

ISO/IEC. (2015). ISO/IEC TR 15504-7:2008 Information technology -- Process assessment -- Part 7: Assessment of organizational maturity. Geneva, Switzerland: International Organization for Standardization.

ISO/IEC. (2015). ISO/IEC/IEEE 29119-4:2015 Software and systems engineering -- Software testing -- Part 4: Test techniques. Geneva, Switzerland: International Organization for Standardization.

ISO/IEC. (2016). ISO/IEC Guide 17:2016 Guide for writing standards taking into account the needs of micro, small and medium-sized enterprises. Geneva, Switzerland: International Organization for Standardization.

ISO/IEC. (2016). ISO/IEC Guide 17:2016 Guide for writing standards taking into account the needs of micro, small and medium-sized enterprises. Geneva, Switzerland: International Organization for Standardization.

ISTQB. International Software Testing Qualifications Board. (s.f.). Recuperado el 14 de abril de 2015, de http://www.istqb.org

Kulkarni, S. (2006). Test process maturity models–yesterday, today and tomorrow. Proceedings of the 6th Annual International Software Testing Conference, Delhi, India. Citeseer.

Laporte, C., Houde, R., & Marvin, J. (2014). 6.4.2 Systems Engineering International Standards and Support Tools for Very Small Enterprises. INCOSE International Symposium, (págs. 551–569). Las Vegas, NV.

Luo, L. (2001). Software testing techniques. PA: Institute for software research international Carnegie mellon university Pittsburgh.

Meerts, J. (04 de 14 de 2016). http://www.testingreferences.com/testinghistory.php. Obtenido de http://www.testingreferences.com/testinghistory.php.

Mexico, N. (s.f.). Tecnología de la información - Software - Modelos de procesos y evaluación para desarrollo y mantenimiento de software - Parte 2: Requisitos y procesos (MoProSoft). Recuperado el 14 de April de 2015, de http://www.tecnyce.com.mx/index.php/proceso-verif/moprosoft.html

Myers, G. J. (1979). Art of Software Testing. New York, USA: John Wiley & Sons, Inc. NIST. (2002). The Economic Impacts of Inadequate Infrastructure for Software Testing. Washington: National Institute of Standards & Technology.

Rasking, M. (2011). Experiences Developing TMMi® as a Public Model. 11th International Conference, SPICE 2011 (págs. pp 190-193). Dublin, Ireland: Springer Berlin Heidelberg.

Reid, S. (s.f.). ISO/IEC/IEEE 29119:The New International Software Testing Standards. London UK: Testing Solution Group Ltd.

Sanz, A., Saldaña, J., Garcia, J., & Gaitero, D. (Diciembre de 2008). TestPAI Un área de proceso de pruebas integrada con CMMI. Revista Española de Innovación, Calidad e Ingeniería del Software REICIS, Volumen 4(No 4).

Softex. (s.f.). Melhoria de Proceso do Software Brasileiro. Recuperado el 14 de April de 2015, de http://www.softex.br/mpsbr/

Sogeti Web Site. (14 de April de 2015). Obtenido de SOGETI: http://www.sogeti.com/

Sogueti. (s.f.). TPI NEXT® Process Certification. Recuperado el 14 de April de 2015, de http://www.sogeti.es/globalassets/spain/soluciones/testing/certificacion-tpi-next-sogeti.pdf

Swinkels, R. (2000). A comparison of TMM and other test process improvement models.

Tim , K., & Martin, P. (1999). Test Process Improvement: A practical step-by-step guide to structured testing. Addison Wesley Longman INC.

TMMI Foundation. (04 de 06 de 2016). Recuperado el 14 de Abril de 2015, de http://www.tmmi.org/

Tmmi Foundation. (2012). Test Maturity Model TMMi Release 1.0. Ireland: Erik van Veenendaal .

Tmmi Foundation. (s.f.). TMMi Assessment Method Application Requirements (TAMAR) version 2.0. Andrew Goslin .

United Nations. (2012). Information Economy Report 2012 - The Software Industry and Developing Countries. En L. Cernat (Ed.), United Nations Conference on Tradea and Development UNCTAD, (pág. 14). New York and Geneva.

USITC. (2010). Small and MediumSized Enterprises: Characteristics and Performance. Washington, DC 20436: United States International Trade Commission. Publication 4189.

Visser, B., de Vries, G., Linker, B., Wilhelmus, L., van Ewijk, A., & van Oosterwijk, M. (2013). TPI® NEXT - Business Driven Test Process Improvement (1 edition (November 17, 2009) ed.). UTN Publishers

Notas Biográficas:

Cecilia García is a leader of software testing projects. Certified by ISTQB and CAPM of the PMI. She is computer engineer of the Pontifical Catholic University of Peru (PUCP) with a master's degree in software engineering. She is member of the research group in software engineering (GIDIS-PUCP). Her research area is the improvement of software testing processes.

Karin Meléndez is a professor at the Pontifical Catholic University of Peru (PUCP), a researcher in the ProCal-ProSer Project (2013-2016)(2017-2019) and

a software quality consultant. She is member of the Technical Committee for Standardization in Software Engineering. She is member of the research group in software engineering (GIDIS-PUCP). Master in Strategic Business Administration at Centrum Católica, business school of the PUCP (2013), and computer engineer of the PUCP (2003). Her research areas are the process management for software development and information technology services.

Abraham Dávila is a researcher and professor at the Pontifical Catholic University of Peru (PUCP) since 2000. He is a principal research of the ProCal-ProSer Project (2013-2016 Phase I and 2017-2019 Phase II) and he is a founding member of the research group in software engineering (GIDIS-PUCP). Master's degree in Computer Science from PUCP and bachelor's degree in science with a major in Mechanical Engineering. He is member of the ISO/IEC working group of ISO/IEC 29110 standards. Their main areas of interest are computer quality (at the level of software process, products and service management) and education in software engineering.

Recibido 5 Feb 2018 *ReCIBE, Año 7 No. 1, Mayo 2018*
Aceptado 21 Feb 2018

Control Robusto de un Doble Péndulo Invertido

Robust control of a Double Inverted Pendulum

Eduardo Ruiz-Velázquez[1]
eduardo.ruiz@cucei.udg.mx

Gustavo Daniel Vega-Magdaleno[1]
gus-dany91@hotmail.com

Julio Alberto García-Rodríguez[1]
julio.garciar@alumnos.udg.mx

[1]División de electrónica y Computación, CUCEI,
Universidad de Guadalajara, Guadalajara, México

Resumen: Este trabajo presenta el diseño de un controlador robusto H∞ para la estabilización de un Doble Péndulo Invertido (DPI). El modelo matemático del DPI es obtenido vía linealización del sistema diseñado en la Toolbox de simulación virtual SimscapeTM MultibodyTM de MathWorks® . Se realiza una comparación de la ejecución entre el controlador incluido por defecto en la Toolbox mencionada y el controlador propuesto. La experimentación virtual demuestra que el algoritmo de control robusto H∞ exhibe un mejor desempeño en escenarios donde existen variaciones paramétricas como era de esperarse. En este sentido, la aportación de este trabajo es mostrar la síntesis del controlador y resaltar la importancia de la Toolbox que proporciona un entorno de simulación 3D para múltiples sistemas de ingeniería mecánica y robótica. De igual manera, resulta ser una excelente herramienta didáctica para el aprendizaje de los sistemas de control retroalimentado.

Palabras clave: control robusto, doble péndulo invertido, SimscapeTM MultibodyTM.

Abstract: This paper presents the design of a robust H∞ controller for the stabilization of a Double Inverted Pendulum (DIP). The mathematical model of the DIP is obtained via linearization of the system designed in the virtual simulation toolbox SimscapeTM MultibodyTM of MathWorks®. A comparison of the execution is made between the controller included by default in the aforementioned Toolbox and the proposed controller. Virtual experimentation shows that the H∞ robust control algorithm exhibits better performance in scenarios where there are parametric variations as expected. In this sense, the contribution of this work is to show the synthesis of the controller and highlight the importance of the Toolbox that provides a 3D simulation environment for multiple mechanical and robotic engineering systems. In the same way, it turns out to be an excellent didactic tool for the learning of feedback control systems.

Keywords: robust control, double inverted pendulum, SimscapeTM MultibodyTM.

1. Introducción

El péndulo invertido es uno de los problemas más interesantes en la teoría de control moderno y que ha sido abordado por la comunidad científica desde diversos enfoques como en los trabajos de (Patil, M. & Kurode, S. 2017), (Raj, S. 2016), (Bettayeb, M., Boussalem, C., Mansouri, R., & Al-Saggaf, U.M. 2014), (Aoustin, Y. & Formal, A. 2011), (Glück, T., Eder, A., & Kugi, A. 2013),(Park, M., & Chwa, D. 2009). El Doble Péndulo Invertido (DPI) es la combinación de dos péndulos individuales para crear un sistema altamente no-lineal que presenta un comportamiento caótico (Li-jie, Chen 2011). El control de este tipo de sistemas, llamados subactuados, es actualmente un área de investigación muy relevante por sus aplicaciones en robótica, sistemas aeroespaciales y vehículos marinos como en (Hyla, Paweł 2012), (Giua, A., Seatzu C., & Usai G. 1999), (Tomofumi Okada, Kenji Tahara 2014), (Bogdanov, Alexander 2004). Desde el punto de vista de los sistemas dinámicos, el DPI montado sobre un carro que puede desplazarse en un eje tiene un punto de operación inestable. De esta manera, es un buen punto de partida para evaluar el desempeño en lazo cerrado de algoritmos de control.

Los métodos newtonianos, leyes de energía cinética y potencial permiten modelar este problema clásico de ingeniería. El modelo matemático del DPI consiste en un sistema no-lineal de seis ecuaciones diferenciales. Estas ecuaciones relacionan la evolución de la posición y velocidad angular tanto del péndulo inferior como las del superior. Así mismo, se relaciona la posición y velocidad lineales del carro sobre el eje. La complejidad matemática, la gran cantidad de parámetros e incertidumbres en éstos, permiten que las técnicas de control robusto sean factibles para la experimentación con el DPI. Particularmente el diseño de controladores robustos por H∞ pueden tratar con sistemas de tales características. Por otra parte, la Toolbox SimscapeTM MultibodyTM de Mathworks® proporciona un entorno de simulación 3D para múltiples sistemas de ingeniería mecánica y robótica (Simscape Multibody, 2016). Los cuales pueden ser modelados por bloques que representan cuerpos, articulaciones, restricciones, elementos de fuerza y sensores. La programación y solución de todas las ecuaciones del modelo no-lineal del DPI se encuentran integradas en el software. Además, es posible obtener las matrices A, B, C y D de la notación en espacio de estados para el modelo lineal del DPI.

En esta aportación se presenta la comparación de la ejecución de dos controladores. El primero es diseñado usando la conocida fórmula de Ackerman o también conocida como ubicación de polos. Esta técnica, se utiliza para la enseñanza de técnicas introductorias de control moderno en cursos de pregrado. Cabe destacar que este controlador es provisto en SimscapeTM MultibodyTM y

está disponible para mostrar la efectividad de la realización del DPI en ese ambiente de simulación y de su control en lazo cerrado. El segundo controlador, que es el que se presenta en este trabajo es diseñado usando la técnica de control H∞. Para ésto, el diseño del controlador es presentado en dos secciones: un controlador nominal y otro robusto. En el primero se incluye solamente el valor nominal de todos los parámetros del DPI para el planteamiento generalizado de control. En el segundo se incluyen funciones de transferencia de peso para caracterizar las incertidumbres paramétricas de la planta y obtener un controlador robusto en el sentido de la norma H∞. La evaluación del lazo cerrado demuestra que el algoritmo robusto presenta mejoras significativas cuando hay variaciones en los parámetros del modelo.

2. Formulación del problema

Este trabajo aborda el diseño de un controlador robusto H∞ para la estabilización de un DPI que está montado en un carro y se desplaza en un eje. Cuando se trata con problemas de ingeniería desde el punto de vista de la teoría de control H∞ pueden definirse márgenes de variación en el sistema debido a errores de modelado o variaciones de los parámetros del sistema. Por lo tanto, la idea del control robusto es que el sistema en lazo cerrado permanezca estable a pesar de tales variaciones.

Además, en este tipo de enfoque también se engloban las perturbaciones externas, las cuales pueden afectar directamente a la entrada, salida o estados del sistema. Por otra parte, el único actuador al cual se tiene acceso para estabilizar ambos péndulos es el torque que desplaza el carro sobre el eje. Este desplazamiento se efectúa de forma minuciosa para que ambos péndulos se mantengan en posición vertical para todo el tiempo de ejecución. Es importante mencionar que la longitud del eje es acotada. Por lo tanto, la acción de control debería ser lo suficientemente suave para que el desplazamiento no sobrepase ese margen.

Las técnicas tradicionales de control clásico no son capaces de exhibir un desempeño aceptable en este tipo de problemas con dinámicas complejas y no lineales. Generalmente requieren del conocimiento de muchas variables del sistema para efectuar control. Por ejemplo, el control por retroalimentación de estados requiere que las posiciones como velocidades angulares de los péndulos sean medibles. De igual manera, la posición y velocidad lineal del carro. Si las variables no están disponibles, pueden ser determinadas por un observador de estados. Sin embargo, algunas variables pudieran ser no observables o que los sensores fueran de un costo elevado. Cabe mencionar, que el control por retroalimentación de estados no es robusto ante variaciones en los parámetros (Zhou, K., & Doyle, J.C. 1998). Por lo tanto, cualquier dinámica

no modelada, variación paramétrica o incertidumbre puede desestabilizar el sistema.

Las metodologías de control clásico Proporcional-Integral-Derivativo (PID), tiene únicamente una salida para la retroalimentación. Para el problema del DPI, la salida consta de dos partes: La posición angular del péndulo superior y la posición del carro dentro de su eje. Por lo tanto, el control PID es limitado para el presente problema. Los controladores no-lineales pueden ser excelentes herramientas cuando se desea tratar con problemas de control sofisticados como el del DPI. Sin embargo, la realización puede convertirse en una tarea muy desafiante para el diseñador debido a que involucra una gran complejidad matemática y experiencia en la implementación de lazo cerrado.

Por otro lado, la presente propuesta muestra que modelar el DPI mediante un esquema en Simscape™ Multibody™ permite obtener un modelo matemático lo bastante adecuado para el diseño del controlador sin pasar por la tarea del modelado. El diseño por la técnica de control H∞ permite considerar algunas condiciones de desempeño tales como la forma de la acción de control e incluso contemplar incertidumbres del tipo no-estructuradas. Esto se logra con la inclusión de funciones de transferencia de peso integradas en el diseño del controlador.

3. Diseño de control por H∞

3.1 Preliminares

Considere que la planta *G* y el controlador *K* del sistema descrito en el diagrama a bloques de la Figura 1 poseen funciones de transferencia propias y racionales reales.

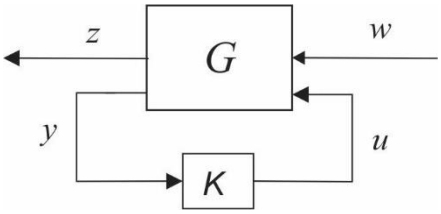

Figura 1. *Configuración generalizada de Transformación Lineal Fraccional (TLF).*

Es posible asumir que se tiene acceso a los modelos en espacio de estados que representan a *G* y *K*. También se asume que estas realizaciones son estabilizables y detectables. Con el fin de desarrollar la síntesis de control, es necesario que exista una $\gamma > 0$ tal que un controlador subóptimo *K*, si existe,

debe satisfacer que $||T_{zw}||_\infty < \gamma$ (Zhou, K., & Doyle, J.C., 1998). La función de transferencia a minimizar está dada por:

$$T_{zw} = F_l(G, K) = G_{11} + G_{12}K(I - G_{22})^{-1}G_{21} \quad (1)$$

en su representación mediante una Transformación Lineal Fraccional (TLF). Donde *G*(*s*) se conoce denomina matriz de transferencia generalizada.

3.2 Diseño del controlador nominal H∞

Para realizar la síntesis de un controlador nominal por H_∞ es necesario linealizar el modelo en un punto de equilibrio. Para este problema el equilibrio está situado en el origen, es decir, se busca que el carro se encuentre en la parte central del riel. Además que ambos péndulos tengan un ángulo de cero grados con respecto a su vertical. La linealizacion es llevada a cabo con el comando *linmod* de Matlab®. Este comando linealiza cada bloque de manera individual empleando un método basado en la obtención de matrices Jacobianas para dichos bloques. Así, las matrices A, B, C, D para la representación en espacios de estado son obtenidas. Es importante señalar que las salidas del sistema son: la posición de carro y el ángulo del péndulo superior.

La planta nominal generalizada *G* representada en la Figura 1 se obtiene mediante su representación por TLF del diagrama a bloques de la Figura 2. Donde P_{nom} es la matriz de transferencia de la planta con valores nominales, la cual relaciona las salidas con la entrada de control. El vector de salida [z_1, z_2, y]' se debe minimizar respecto al vector de entrada [d,u]' en el sentido de la norma H_∞, como en la Ecuación 1. Por lo tanto, la planta generalizada se describe de la siguiente manera:

$$\begin{bmatrix} z_1 \\ z_2 \\ y \end{bmatrix} = G \begin{bmatrix} d \\ u \end{bmatrix} = \begin{bmatrix} W_e & W_e P_{nom} \\ 0 & W_u \\ I & -P_{nom} \end{bmatrix} \begin{bmatrix} d \\ u \end{bmatrix} \quad (2)$$

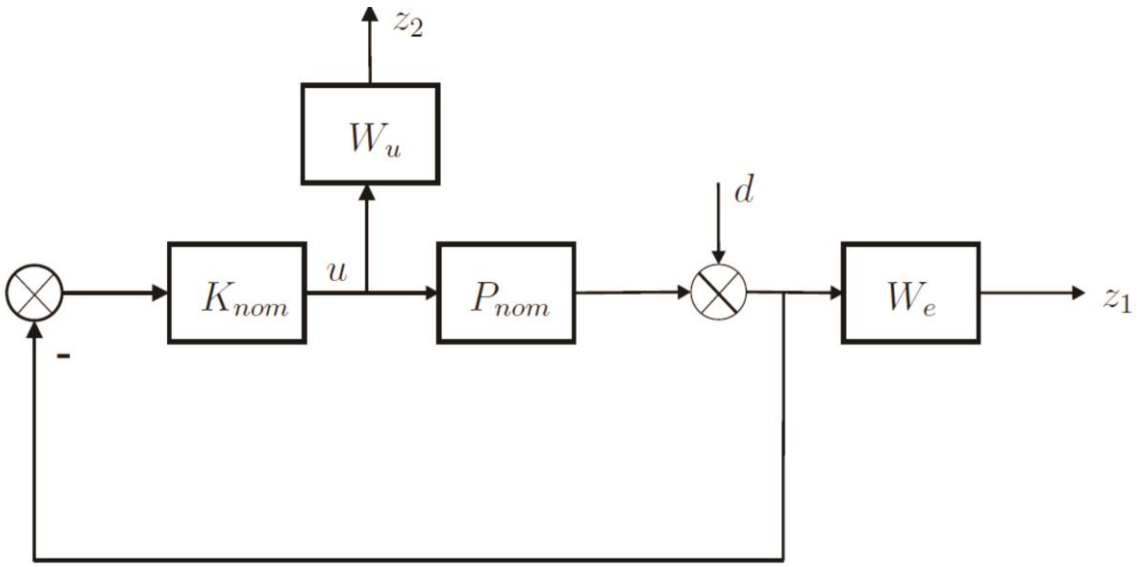

Figura 2. *Planteamiento del problema de control nominal.*

La matriz de transferencia de peso W_e contiene funciones de transferencia de primer orden y su objetivo es incluir los requisitos de desempeño deseados para la estabilización del DPI. W_u es la función de peso que da forma a la señal de control. Las funciones de transferencia están dadas por las siguientes ecuaciones:

$$G_{\theta u} = \frac{-13{,}13s^2 - 2{,}332e^{-14}s + 0{,}006545}{s^4 - 1{,}332^{-14}s^3 - 302{,}3s^2 + 1{,}592e^{-12}s + 1{,}42e^4} \tag{3}$$

$$G_{xu} = \frac{1{,}537s^4 - 2{,}13e^{-13}s^3 - 421{,}3s^2 + 1{,}853e^{-11}s + 1{,}56e^4}{s^6 - 1{,}332e^{-14}s^5 - 302{,}3s^4 + 1{,}592e^{-12}s^3 + 1{,}42e^4s^2} \tag{4}$$

donde estas dos últimas ecuaciones son elementos de la matriz de transferencia P_{nom}. Por último se tienen:

$$W_e = \begin{bmatrix} \frac{0{,}00028571(s+1{,}75)}{s+0{,}05} & 0 \\ 0 & \frac{0{,}033333(s+0{,}9)}{s+0{,}03} \end{bmatrix} \tag{5}$$

$$W_u = 0{,}0010 \tag{6}$$

Se utilizó la función *hinfsyn* del Toolbox™ Robust Control de Matlab® con la finalidad de llevar a cabo la síntesis del controlador H∞. El controlador resultante es de orden 8 con un valor γ = 0.033$. Previo a la implementacion, se realiza un análisis de los valores singulares de Hankel con el propósito de reducir el orden del controlador. La instrucción *reduce* de Matlab® reduce el orden del controlador a 6. La matriz de transferencia del controlador se muestra a continuación:

$$K_{nom} = \begin{bmatrix} \frac{3{,}476e^4 s^5 + 9{,}945e^5 s^4 + 8{,}481e^6 s^3 + 2{,}239e^7 s^2 + 9{,}976e^5 s + 1{,}928e^4}{s^6 + 2262 s^5 + 2{,}05e^5 s^4 + 2{,}377e^6 s^3 + 4{,}488e^6 s^2 - 2{,}51e^6 s - 2{,}044e^7} \\ \\ \frac{-3{,}887e^5 s^5 - 1{,}328e^7 s^4 - 1{,}452e^8 s^3 - 5{,}098e^8 s^2 - 2{,}039e^7 s + 5{,}235e^4}{s^6 + 2262 s^5 + 2{,}05e^5 s^4 + 2{,}377e^6 s^3 + 4{,}488e^6 s^2 - 2{,}51e^6 s - 2{,}044e^7} \end{bmatrix} \quad (7)$$

3.3 Diseño de controlador robusto en H∞

El controlador de la sección anterior considera que todos los parámetros del DPI son nominales. El diseño de un controlador robusto implica tomar en cuenta los parámetros que presentan incertidumbres, *e.g*. por errores de modelado. En esta aportación se consideran incertidumbres del tipo no estructuradas y que deben ser representadas por matrices de transferencia como funciones de peso. Una familia de plantas se obtuvo al variar todos los parámetros del modelo nominal linealizado del DPI. Esta familia de plantas puede ser representada utilizando incertidumbres de tipo multiplicativas,

$$P = (I + W_1 \Delta W_2) P_{nom} \quad (8)$$

con $||\Delta||_\infty \leq 1$ (Zhou, K., & Doyle, J.C., 1998). P_{nom} es la planta nominal. W_1 en y W_2 son matrices de funciones de peso cuya respuesta en frecuencia simboliza la familia de plantas con incertidumbres. Por lo tanto, se puede tener la siguiente representación:

$$W_1 \Delta W_2 = \left| \frac{P - P_{nom}}{P_{nom}} \right| \quad (9)$$

Así, para cada frecuencia se tiene que

$$W_1(jw) W_2(jw) = \max_w \left| \frac{P(jw) - P_{nom}(jw)}{P_{nom}(jw)} \right| \quad (10)$$

De esta forma, W_1 y W_2 son la máxima respuesta en frecuencia de la familia de plantas con variación paramétrica. Se identifican las siguientes de funciones de transferencia con el comando *fitmag* de Matlab®:

$$W_1 = \begin{bmatrix} \frac{2{,}24e^{-5}s^2+3{,}85e^{-5}s+0{,}0009644}{s^2+0{,}9962s+0{,}8858} & 0 \\ 0 & \frac{7{,}218e^{-7}s+2{,}504e^{-5}}{s+0{,}09563} \end{bmatrix} \quad (11)$$

$$W_2 = \begin{bmatrix} 0{,}0450 & 0 \\ 0 & 0{,}0450 \end{bmatrix} \quad (12)$$

La ecuaciones 11 Y 12 representan la incertidumbre paramétrica no modelada de la planta *P*. Por lo tanto, el planteamiento del problema de control robusto se modifica como se observa en la Figura 3. Finalmente, para la síntesis de este controlador se agregó la función W_d de peso que limita la magnitud de las perturbaciones externas dentro del rango de frecuencias de interés. La función de transferencia que la representa es:

$$W_d = \begin{bmatrix} \frac{0{,}0001(s+1e^8)}{s+1e^6} & 0 \\ 0 & \frac{0{,}0001(s+1e^8)}{s+1e^6} \end{bmatrix} \quad (13)$$

La planta generalizada para el caso robusto se plantea a continuación:

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ y \end{bmatrix} = P(s) \begin{bmatrix} d \\ d_1 \\ u \end{bmatrix} = \begin{bmatrix} W_dW_e & W_2W_e & W_eG_o \\ 0 & 0 & W_u \\ 0 & 0 & W_1G_o \\ -W_d & -W_2 & -G_o \end{bmatrix} \begin{bmatrix} d \\ d_1 \\ u \end{bmatrix} \quad (14)$$

Se realiza la síntesis H_∞ y se obtiene un controlador de orden 13 con $\gamma = 0.0461$. El análisis de los valores singulares de Hankel permite una reducción a orden 8. La matriz de transferencia del controlador reducido se muestra a continuación:

$$K_{rob} = \begin{bmatrix} \frac{-0{,}5268s^8-1{,}429e^6s^7-1{,}123e^{12}s^6-8{,}857e^{11}s^5-2{,}42e^{15}s^4-2{,}723e^{16}s^3-9{,}338e^{15}s^2+1{,}381e^{16}s+7{,}841e^{14}}{s^8+2{,}627e^6s^7+1{,}649e^{12}s^6+1{,}643e^{14}s^5+4{,}299e^{15}s^4+1{,}953e^{16}s^3+4{,}742e^{16}s^2+8{,}89e^{16}s+5{,}439e^{16}} \\ \\ \frac{-105s^8-3{,}024e^8s^7-2{,}153e^{14}s^6-1{,}329e^{16}s^5-2{,}378e^{17}s^4-1{,}463e^{18}s^3-2{,}707e^{18}s^2-1{,}785e^{18}s+9{,}634e^{12}}{s^8+2{,}627e^6s^7+1{,}649e^{12}s^6+1{,}643e^{14}s^5+4{,}299e^{15}s^4+1{,}953e^{16}s^3+4{,}742e^{16}s^2+8{,}89e^{16}s+5{,}439e^{16}} \end{bmatrix} \quad (15)$$

4. Simulaciones Numéricas

La representación en diagrama de bloques de Simulink® y de Simscape™ Multibody™ que representa al modelado del DPI puede apreciarse en la Figura 4. El software permite la visualización del sistema en 3 dimensiones y simula su comportamiento con respecto al tiempo, como puede verse en la Figura 5. En esta sección se muestran 3 pruebas con diferentes perturbaciones a la posición angular del segundo péndulo. De la misma forma, se realizan variaciones paramétricas para probar la robustez del controlador. En cada uno de los experimentos se compara el desempeño del control robusto H_∞ con el controlador incluido por defecto en el simulador.

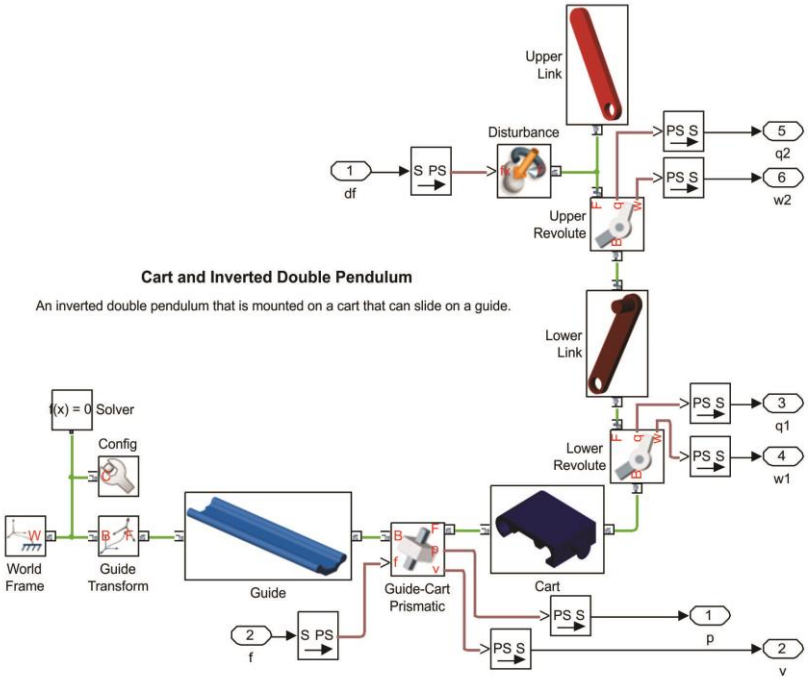

Figura 4. *Modelado del DPI en Simscape™ Multibody™.*

Cabe mencionar que el controlador por ubicación de polos requiere la retroalimentación de todos los estados, es decir, las posiciones y velocidades angulares de ambos péndulos. Esto puede verse en la Figura 5.

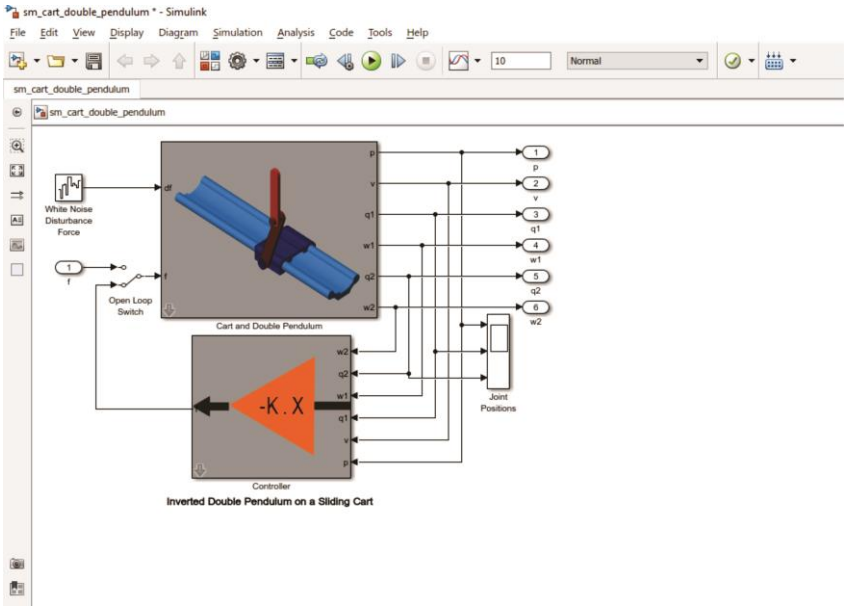

Figura 5. *Lazo cerrado para el control del DPI por retroalimentación de estados.*

El controlador por ubicación de polos fue diseñado eligiendo polos de lazo cerrado estables por supuesto y que la estabilización del DPI se obtuviera con poco sobreimpuslo en un tiempo corto. El vector de ganancias para este controlador es presentado a continuación:

$$K = [-11 - 10 - 10 - 10 - 10 - 10] \tag{16}$$

La implementación de los controladores nominal y robusto H_∞ se llevó acabo al sustituir el bloque naranja de la retroalimentación de estados por un bloque de función de transferencia en la Figura 5. Recordando que a esta función de transferencia únicamente se lo proporciona el ángulo del péndulo superior así como la posición del carro.

4.1 Experimentación con parámetros nominales

En esta prueba se considera como perturbación al ruido de magnitud $5e^{-6}$ en el ángulo del péndulo superior. Las longitudes nominales de ambos péndulos son las siguientes:

- Longitud del péndulo inferior: 18cm.
- Longitud del péndulo superior: 14cm.

Las condiciones iniciales fueron:

- Ángulo del péndulo superior: 5°.
- Posición del carro: -1cm.
- Restantes: 0.

En la Figura 6 se presentan las respuestas del controlador nominal, robusto y por ubicación de polos con parámetros nominales en el DPI. Aquí se puede apreciar que el controlador incluido por defecto en la Toolbox exhibe un mejor desempeño que los controladores diseñados por H_∞. El ángulo del péndulo superior es similar para los tres controladores pero la posición del carro exhibe un comportamiento mejor para el diseñado con la fórmula de Ackerman. Sin embargo hay que recordar que para este controlador se está suponiendo que se cuenta con todo el estado para la retroalimentación.

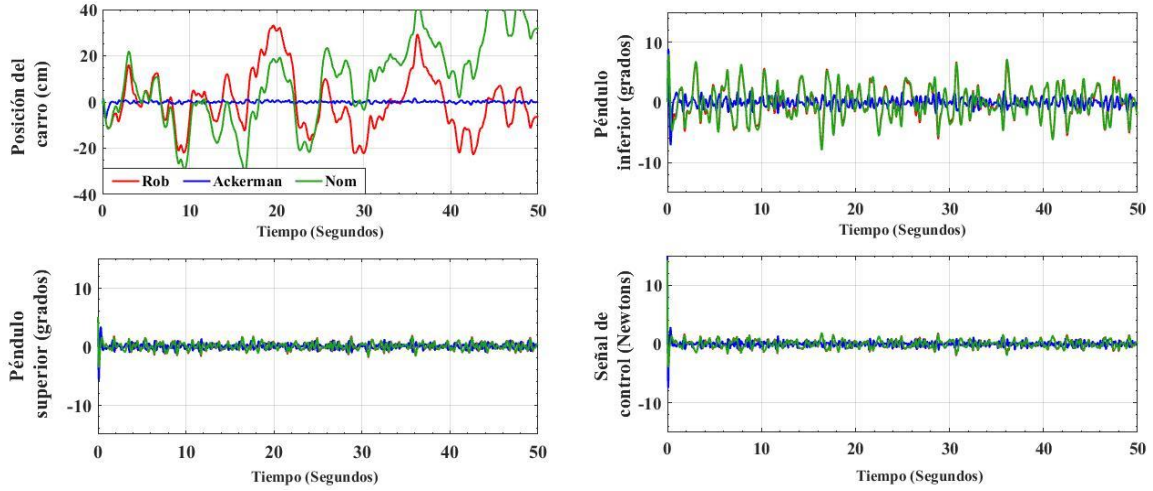

Figura 6: *Respuesta de controladores Nominal, Robusto y por ubicación de polos con parámetros nominales en el DPI.*

4.2 Prueba con péndulos de longitud mayor a la nominal

La variación parámetrica se lleva a cabo al sustituir ambos péndulos por otros con longitud mayor a la nominal. Se considera un ruido de perturbación igual al caso anterior. Se compara únicamente el controlador robusto con el de Ackerman.

Las longitudes de los péndulos en esta ocasión fueron:

- Longitud del péndulo inferior: 37cm.
- Longitud del péndulo superior: 30cm.

Las condiciones iniciales son:

- Ángulo del péndulo superior: 5°.
- Posición del carro: -1cm.
- Restantes: 0.

La Figura 7 muestra el desempeño del controlador robusto. Se aprecia que existe un transitorio de aproximadamente 10 segundos para que el péndulo superior se estabilice alrededor de los cero grados. Por otra parte, en la Figura 8 se puede observar que el controlador incluido por defecto no puede estabilizar al DPI e incluso la respuesta diverge por lo que el lazo cerrado se vuelve inestable.

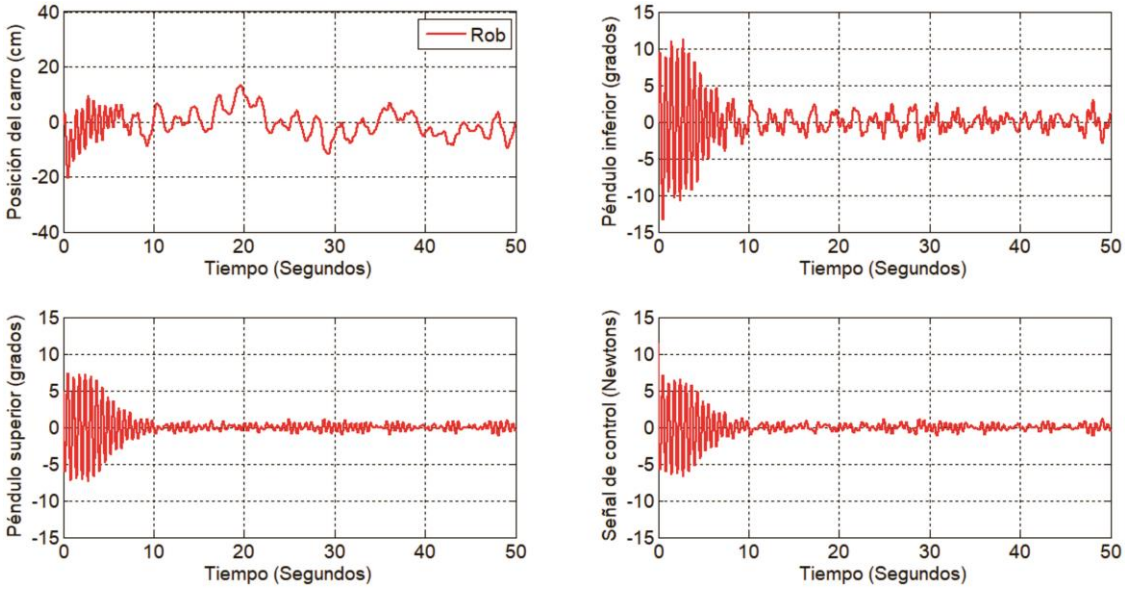

Figura 7. *Respuesta del controlador robusto con variación paramétrica en la longitud de los péndulos.*

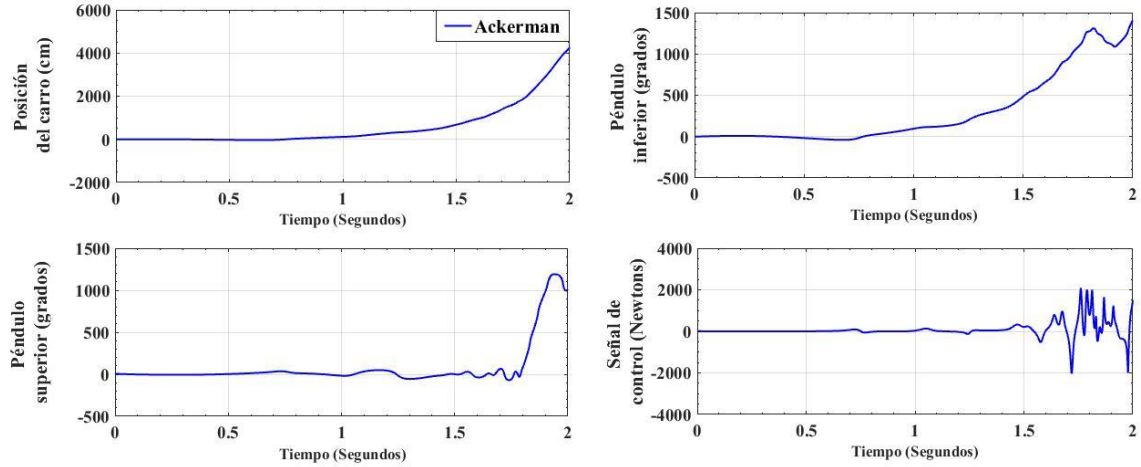

Figura 8. *Respuesta del controlador por ubicación de polos con variación paramétrica en la longitud de los péndulos.*

4.3 Prueba con péndulos de longitud menor a la nominal

Similar a la situación anterior, ahora se plantea que ambos péndulos sean sustituidos por otros de longitud menor a la nominal. Se considera el mismo ruido de perturbación que en los dos caso anteriores.
Las longitudes de los péndulos son:

- Longitud del péndulo inferior: 10cm.
- Longitud del péndulo superior: 10cm.

Las condiciones iniciales son:

- Ángulo del péndulo superior: 5°.
- Posición del carro: -1cm.
- Restantes: 0.

Similar al desempeño mostrado cuando los péndulos aumentan de su longitud nominal, cuando se disminuyen, el controlador robusto puede estabilizar al péndulo superior con buena precisión (Figura 9). Por otro lado, el controlador por Ackerman no puede estabilizar al DPI y de esta manera el sistema de lazo cerrado se vuelve inestable en poco tiempo de iniciada la simulación (Figura 10).

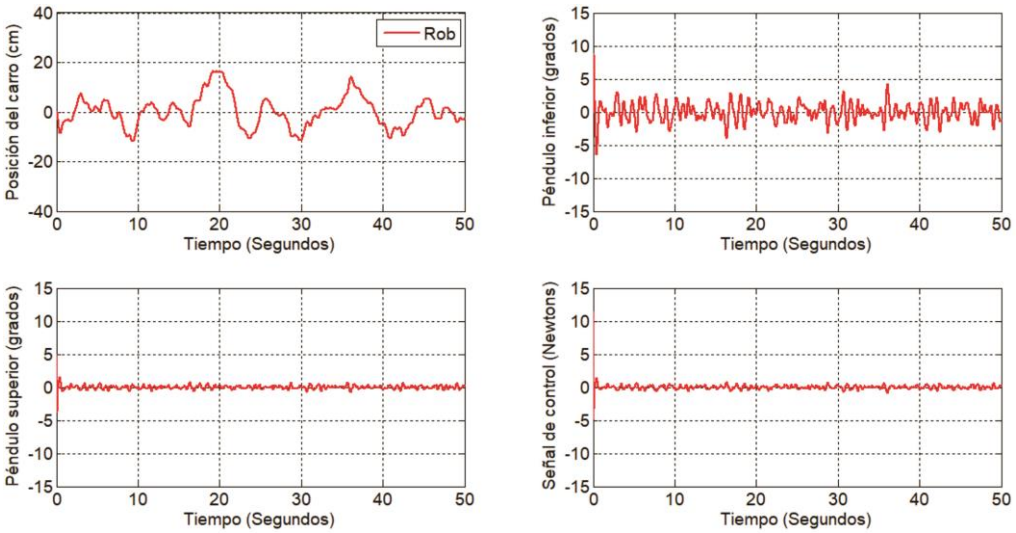

Figura 9. *Respuesta del controlador robusto con variación paramétrica. Longitud menor a la nominal.*

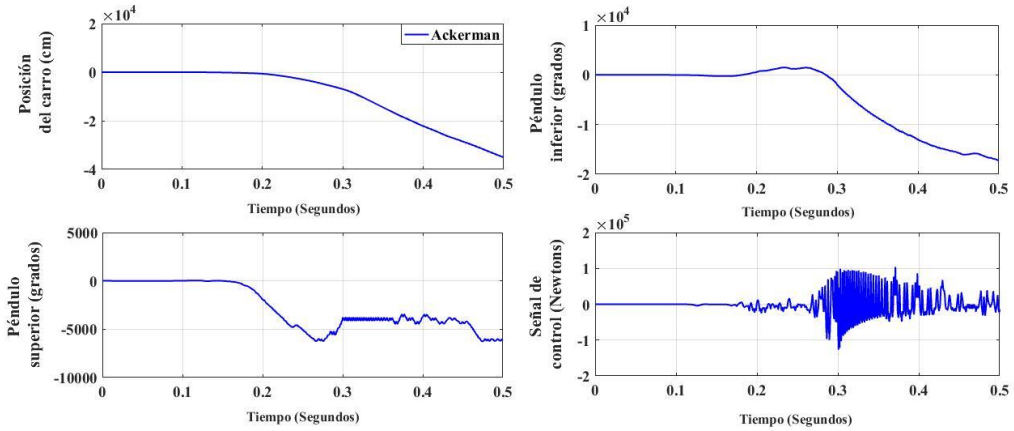

Figura 10. *Respuesta del controlador por ubicación de polos con variación paramétrica. Longitud menor a la nominal.*

5. Conclusiones

En el presente trabajo se muestra el diseño de dos controladores por H_∞ para la estabilización de un sistema de doble péndulo invertido DPI montado en un carro. El primer controlador se diseñó considerando valores nominales en el modelo del DPI. El segundo, denominado robusto, se obtuvo considerando variaciones paramétricas, incertidumbres no estructuradas, en los parámetros del DPI. Estos controladores fueron comparados con un controlador diseñado por la técnica de ubicación de polos, que viene acompañada con la demostración del modelado del DPI en el ambiente Simulink® y de Simscape™ Multibody™. Es conocido de la literatura que el controlador diseñado por el método de ubicación de polos no tiene la capacidad de manejar incertidumbres en los modelos de la plantas a controlar. De esta manera, el presente trabajo tiene como objetivo demostrar ese conocimiento previo con este ejemplo y, además ilustrar que el modelado de sistemas físicos por medio de la herramienta de simulación mencionada es una alternativa de gran utilidad. Esto último refiere a que es posible obtener modelos matemáticos aproximados de sistemas físicos de manera directa sin pasar por la tarea de aplicar leyes físicas que los gobiernan.

Referencias

Aoustin, Y. & Formal, A. (2011) Pendubot: combining of energy and intuitive approaches to swing up , stabilization in erected pose. Myltibody Syst. Dyn., 25: 65–80.

Bettayeb, M., Boussalem, C., Mansouri, R., & Al-Saggaf, U.M. (2014) Stabilization of an inverted pendulum-cart system by fractional PI-state feedback, ISA Trans. 53, 508–516.

Bogdanov, Alexander (2004) Optimal Control of a Double Inverted Pendulum on a Cart. Department of Computer Science and Electrical Engineering, OGI School of Science and Engineering, OHSU, Technical Report CSE 04-006.

Giua, A., Seatzu C., & Usai G. (1999) Observer-controller design for cranes via Lyapunov equivalence. Automatica, vol. 35, no 4, pp. 669-678.

Glück, T., Eder, A., & Kugi, A. (2013) Swing-up control of a triple pendulum on a cart with experimental validation. Automatica. 49, 801–808.

Hyla, Paweł (2012) The crane control systems: A survey. 2012 17th International Conference on Methods and Models in Automation and Robotics (MMAR).

Li-jie, Chen (2011) Research on the nonlinear dynamical behavior of double pendulum. 2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC), 1637 - 1640.

Park, M., & Chwa, D. (2009) Swing-Up and Stabilization Control of Inverted-Pendulum Systems via Coupled Sliding-Mode Control Method. IEEE Trans. Industrial Electronics, Vol. 56, No. 9, 3541–3555.

Patil, M. & Kurode, S. (2017) Stabilization of Rotary Double Inverted Pendulum using Higher Order Sliding Modes. Asian Control conference (ASCC), 1818–1823.

Raj, S. (2016) Reinforcement Learning based Controller for Stabilization of Double Inverted Pendulum. IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES-2016), 1–5.

Simscape Multibody (2016), The Mathworks Inc., 2016a.
Tomofumi Okada, Kenji Tahara (2014) Development of a two-link planar manipulator with continuously variable transmission mechanism. 2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM).

Zhou, K., & Doyle, J.C. (1998) Essentials of Robust Control. Prentice-Hall, Upper Saddle River, NJ, 1998.

Notas Biográficas

Eduardo Ruiz Velázquez. Recibió los grados de: Ingeniero en Electrónica (1998), Maestro en Ingeniería Eléctrica (Control Automático, 1999), Doctorado en Ingeniería Eléctrica (Control Automático, 2003) todos por la Universidad Autónoma de San Luis Potosí. Actualmente es Profesor-Investigador Titular en el Departamento de Electrónica, CUCEI, de la Universidad de Guadalajara. Sus principales intereses de investigación son: Modelado y Control de sistemas biomédicos, Control de glucosa en diabetes, identificación de sistemas, Control Robusto.

Gustavo Daniel Vega Magdaleno. Obtuvo el grado de Ing. en Comunicaciones y Electrónica por la Universidad de Guadalajara (2016). Actualmente estudia el grado de M.C. en Ingeniería Electrónica y Computación (Sistemas inteligentes) en la Universidad de Guadalajara. Sus intereses de investigación están enfocados al uso de redes neuronales, identificación de sistemas, algoritmos evolutivos, modelo de control predictivo y control robusto.

Julio Alberto García Rodríguez. Recibió el grado de Ingeniero en Electrónica (2010) por el Instituto Tecnológico de Ciudad Guzmán, Maestro en Ciencias en Ingeniería Electrónica y Computación (2012) por el CUCEI de la Universidad de Guadalajara. Actualmente es estudiante activo del Doctorado en Ciencias de la Electrónica y la Computación. Sus principales áreas de investigación son: Sistemas de control para Diabetes Tipo 1, modelado matemático de sistemas biomédicos, control robusto.

Recibido 6 Feb 2018
Aceptado 1 Mar 2018
ReCIBE, Año 7 No. 1, Mayo 2018

Control µ-Síntesis para Estabilización de un Helicóptero de 3 Grados de Libertad (GDL)

µ-Synthesis Control for Stabilization of a Helicopter of 3 Degrees of Freedom (DOF)

José Manuel Jiménez-Mora[1]
jomaji007@hotmail.com

Eduardo Ruiz-Velázquez[1]
eduardo.ruiz@cucei.udg.mx

Gualberto Solís-Perales[1]
gualberto.solis@cucei.udg.mx

Julio Alberto García-Rodríguez[1]
julio.garciar@alumnos.udg.mx

[1]División de electrónica y Computación, CUCEI, Universidad de Guadalajara, Guadalajara, México

Resumen: Este artículo presenta el diseño y puesta en marcha de un controlador robusto para estabilizar un helicóptero de 3 Grados De Libertad (GDL). Esta plataforma de experimentación es desarrollada por la empresa Quanser Inc. La caracterización de algunos parámetros del helicóptero, las incertidumbres estructuradas y la inclusión de funciones de peso permiten obtener un controlador por µ-síntesis. La estabilización del helicóptero es llevada a cabo por el control de los tres ángulos más importantes: elevación, cabeceo y viaje. Los resultados experimentales muestran que el diseño de control por la técnica de µ -síntesis proporciona un buen desempeño en sistemas con dinámicas altamente no lineales.

Palabras clave: µ-síntesis, helicóptero de 3 GDL, control robusto.

Abstract: This work presents the design and implementation of a robust controller to stabilize a helicopter of 3 Degrees of Freedom (DOF). This experimentation platform is developed by the company Quanser Inc. The characterization of some parameters of the helicopter, the structured uncertainties and the inclusion of weight functions allow to obtain a controller by µ-synthesis. The stabilization of the helicopter is carried out by the control of the three most important angles: elevation, pitch and travel. The experimental results show that the control design by the technique of µ-synthesis provides a good performance in systems with highly non-linear dynamics.

Keywords: µ-synthesis, 3 DOF helicopter, robust control.

1. Introducción

En años recientes el uso de vehículos aéreos han tenido un rápido crecimiento en varias aplicaciones: militar, industrial, investigación científica e incluso a nivel lúdico. Se utilizan ampliamente para monitoreo ambiental, detección de incendios forestales, monitoreo del tráfico, la construcción, prevención del delito, búsqueda, rescate, cartografía, vigilancia, reconocimiento aéreo, entre otras (Raptis, A. & Valavanis, P. 2011). De todas las clases de Vehículos Aéreos No Tripulados (VANT) el uso de helicópteros tienen ventajas sobre las aeronaves de ala fija. Estos helicópteros pueden aterrizar y despegar verticalmente en un área relativamente restringida, no requieren una pista y tienen la capacidad de volar en altitudes bajas (Wang, X., Chen, Y., Lu, G., & Zhong Y. 2013). El helicóptero de 3 GDL fabricado por la empresa Quanser Inc. es una plataforma de experimentos de control aeroespacial. Se basa en el modelo de un helicóptero de rotor tándem. Es usado para investigar y desarrollar leyes de control para un vehículo que tenga dinámicas representativas de un helicóptero de cuerpo rígido de doble rotor o cualquier dispositivo con dinámica similar (Quanser Consulting Inc. 2012).

Las propuestas de control para VANT's han sido desarrolladas usando diferentes técnicas para estabilizar helicópteros en la posición fluctuante, denominada en inglés *hover position*. Por mencionar algunos de estos trabajos, como en (Vázquez, E., Pérez, E., & Hernández, D. 2017) donde presentan un diseño de controladores por modos deslizantes, Super-Twisting y de terminal continua para el control del helicóptero de 3 GDL de Quanser Inc. Los resultados mostrados en dicho artículo son alentadores, notando que a pesar de ello, la dinámica del ángulo elevación presenta un reto importante. En (Zhou, F., Li, D., & Xia, P. 2009) proponen un control para seguimiento en el helicóptero de 3 GDL basado en lógica difusa. El sistema pudo seguir la posición del eje de elevación exhibiendo estabilidad y robustez en lazo cerrado. Por su parte en (Wang, X., Chen, Y., Lu, G., & Zhong Y. 2013) diseñan un control robusto no-lineal combinado con una red neuronal. Las simulaciones numéricas muestran que es efectivo tanto para elevación como para el sistema en cascada *pitch-travel*.

En el trabajo (Ríos, H., Rosales, J.A., Ferreira, A. & Iriarte, R. 2009) muestran la idea de un control Proporcional Integral Derivativo (PID) y un compensador de perturbaciones. Además integran un observador por modos deslizantes para mejorar la robustez del sistema real. Así mismo, un algoritmo adaptativo integral backstepping es publicado en (Fang, Z., Gao, W., & Zhang, L. 2012) para el control robusto de un helicóptero de 3 GDL. Este algoritmo puede estimar las incertidumbres del modelo en línea y mejorar el desempeño del sistema para seguimiento de una trayectoria. Desde la perspectiva de la ingeniería los VANT son de gran interés por ser un problema altamente no lineal, con incertidumbres paramétricas y perturbaciones externas. En este artículo abordaremos el diseño

de un control basado en la técnica de μ-síntesis ya que ofrece una gran ventaja sobre otras técnicas, debido a que puede trabajar bajo discrepancias en el modelado matemático e incertidumbres paramétricas.

2. Modelado matemático

El helicóptero de 3 GDL de Quanser Inc. es un sistema altamente inestable con dinámicas complejas relacionadas entre sí. Existe dependencia lineal entre sus tres ángulos principales: elevación (ε), cabeceo (ρ) y viaje (τ). La tensión aplicada a los dos motores permite que pueda despegar de su posición de reposo y hacer movimientos en tres dimensiones. La diferencia de presión de aire entre la parte superior e inferior y la fuerza perpendicular generada, hacen posible que el helicóptero se mueva en las tres direcciones angulares (Ríos, H., Rosales, J.A., Ferreira, A. & Iriarte, R. 2009).

Como se mencionó anteriormente, diversas técnicas de control lineal han sido diseñadas para implementarse en esta plataforma de helicóptero de 3 GDL de Quanser Inc. La popularidad de este experimento se debe en gran parte a la compatibilidad con Matlab® de MathWorks Inc. Esto conlleva a una minimización significativa en procesamiento computacional y tiempo de diseño. En la Figura 1 se muestra el diagrama de cuerpo libre de todas las distribuciones del helicóptero de 3 GDL.

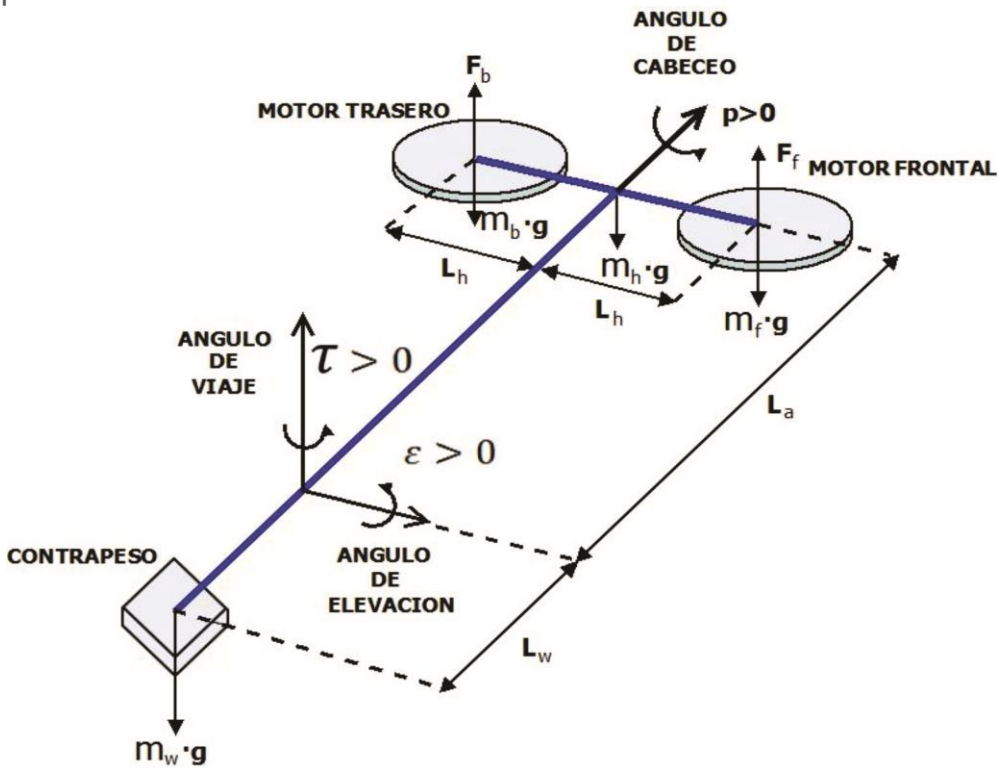

Figura 1. Diagrama de cuerpo libre del helicóptero de 3 GDL (Quanser Consulting Inc. 2012).

A continuación se muestra el modelo matemático para la dinámica de los 3 GDL del helicóptero.

Ángulo de elevación (ε): El movimiento vertical es producido por la fuerza de los motores en las hélices. El comportamiento de este movimiento se modela a través la siguiente ecuación diferencial:

$$\ddot{\varepsilon} = \frac{-(m_f + m_b)gL_a\cos(\varepsilon) + m_\omega L_\omega \cos(\varepsilon)g - \eta_\varepsilon \dot{\varepsilon} + K_f L_a(V_f + V_b)\cos(\rho)}{J_\epsilon} \quad (1)$$

Ángulo de cabeceo (ρ): Este movimiento se genera debido a la diferencia de tensión aplicada a cualquiera de los motores, es decir, por un desequilibrio de fuerzas entre los pares de torsión del motor que causa una inclinación en el eje de paso. La dinámica en el eje de cabeceo es:

$$\ddot{\rho} = \frac{-(m_f + m_b)gL_h\cos(\rho) - \eta_\rho \dot{\rho} + K_f L_h(V_f - V_b)}{J_\rho} \quad (2)$$

Ángulo de viaje (τ): Es un movimiento alrededor de la base principal generado por el componente horizontal de las hélices. La siguiente ecuación diferencial muestra el ángulo de viaje dinámico:

$$\ddot{\tau} = \frac{-\eta_\tau \dot{\tau} - K_f L_a(V_f + V_b)\sin(\rho)}{J_\tau} \quad (3)$$

La notación de los parámetros de acuerdo al manual de instructor (Quanser Consulting Inc. 2012) se define en la Tabla 1 como:

Símbolo	Nomenclatura	Rango-Unidad
V_f, V_b	Voltage aplicado a cada uno respectivamente	0-24 V
m_h, m_ω	Masa del helicóptero y del contrapeso	1.15 Kg y 1.87 Kg
m_f, m_b	Masa de la propela frontal y trasera	0.575 Kg y 0.713 Kg
L_ω	Distancia desde el eje del ángulo de viaje hasta el contrapeso	0.4699 m.
L_a	Distancia desde el eje del ángulo de viaje hasta las propelas del helicóptero	0.6604 m.
L_h	Distancia desde el ángulo de cabeceo hasta al centro de la hélice	0.1778 Kg
K_f	Constante fuerza-empuje de la hélice	0.1778 N/V
g	Constante gravitacional	9.8 m/s^2

Tabla 1. *Nomenclatura de los parámetros del modelo.*

Para el diseño de control por μ-síntesis se propone una linealización del modelo no lineal utilizando el jacobiano en el punto cero, donde todos los ángulos se aproximan a ese valor. Para este problema el equilibrio es tomado como una

posición de vuelo deseada, alrededor de cero. El sistema no lineal se representa por el conjunto de ecuaciones (4) y (5) donde las funciones *f* y *h* representan la dinámica entrada-salida excitadas por una entrada *u*(*t*).

$$\begin{aligned} \dot{x} &= f(x,t) + g(x,t)u(t), \qquad x(t_0) = x_0 \\ y(t) &= h(x(t)) \end{aligned} \tag{4)-(5)}$$

La representación en espacio de estados es descrita de la siguiente forma:

$$\begin{aligned} \dot{x} &= [\varepsilon\ \rho\ \tau\ \dot{\varepsilon}\ \dot{\rho}\ \dot{\tau}]^T \\ u &= [V_f\ V_b]^T \\ y &= [\varepsilon\ \rho\ \tau]^T \end{aligned} \tag{6)-(8)}$$

De esta manera la representación lineal del modelo del helicóptero de 3 GDL contiene las siguientes matrices:

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -10{,}42e^{-4} & 0 & 0 \\ 0 & 0 & 0 & 0 & -137{,}53e^{-3} & 0 \\ 0 & -2{,}1092 & 0 & 0 & 0 & -10{,}42e^{-4} \end{bmatrix} \tag{9}$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 817{,}51e^{-4} & 817{,}51e^{-4} \\ 581{,}01e^{-3} & -581{,}01e^{-3} \\ 0 & 0 \end{bmatrix} \tag{10}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{11}$$

$$D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{12}$$

3. Diseño de control por μ-síntesis

Uno de los requerimientos más importantes para un sistema de control es la característica de robustez. En la realidad, todos los sistemas están expuestos a perturbaciones externas, incertidumbres paramétricas y efectos de ruido (Da-Wei, G., Petko H. & Konstantinov, M. 2005). El helicóptero de 3 GDL es una plataforma de experimentos que permite evaluar el desempeño de algoritmos de control ante estas circunstancias. De esta manera se elige *μ*-síntesis por ser una herramienta de control robusto, que es aplicada al modelo lineal o linealizado de la planta. Además, permite caracterizar las incertidumbres paramétricas del modelo de la planta como estructuradas. Esto es, como errores bien definidos en los parámetros del helicóptero. De esta manera, el algoritmo sintetizado puede desempeñarse satisfactoriamente incluso cuando ocurra el *peor de los casos* en la variación de los parámetros inciertos o en la peor perturbación.

El esquema generalizado de control puede presentarse como en la Figura 2. El bloque principal *M* describe el modelo matemático del sistema con parámetros nominales. El bloque Δ caracteriza todos los elementos externos de incertidumbre del modelo *M*. Esta configuración llamada Transformación Lineal Fraccional (TLF) representa los efectos del vector de salida *z* debido al vector de entrada *w* (Da-Wei, G., Petko H. & Konstantinov, M. 2005).

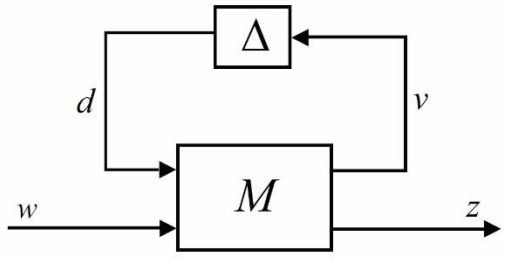

Figura 2. *Esquema generalizado para control por μ-síntesis.*

Por lo tanto, el objetivo general de control es diseñar una ley que estabilice internamente el helicóptero de 3 GDL con modelo *M* y que exhiba un desempeño robusto incluso en la presencia de incertidumbres externas estructuradas caracterizadas como Δ. Esta técnica se basa en la minimización del Valor Singular Estructurado (VSE) *μ* expresado matemáticamente como:

$$\mu_{\Delta}^{-1}(M) = \min_{\Delta \in \mathbf{\Delta}} \{\bar{\sigma}(\Delta) : \det(I - M\Delta) = 0\} \tag{13}$$

La definición del VSE *μ* se basa en encontrar la Δ estructurada más pequeña medida en términos del $\bar{\sigma}(\Delta)$ la cual hace a la matriz $(I - M\Delta)$ singular (Zhou, K., & Doyle, J.C. 1998).

En la Figura 3 se muestra la configuración a bloques para el diseño del controlador por μ-síntesis que incluye incertidumbres estructuradas y matrices de transferencia de peso.

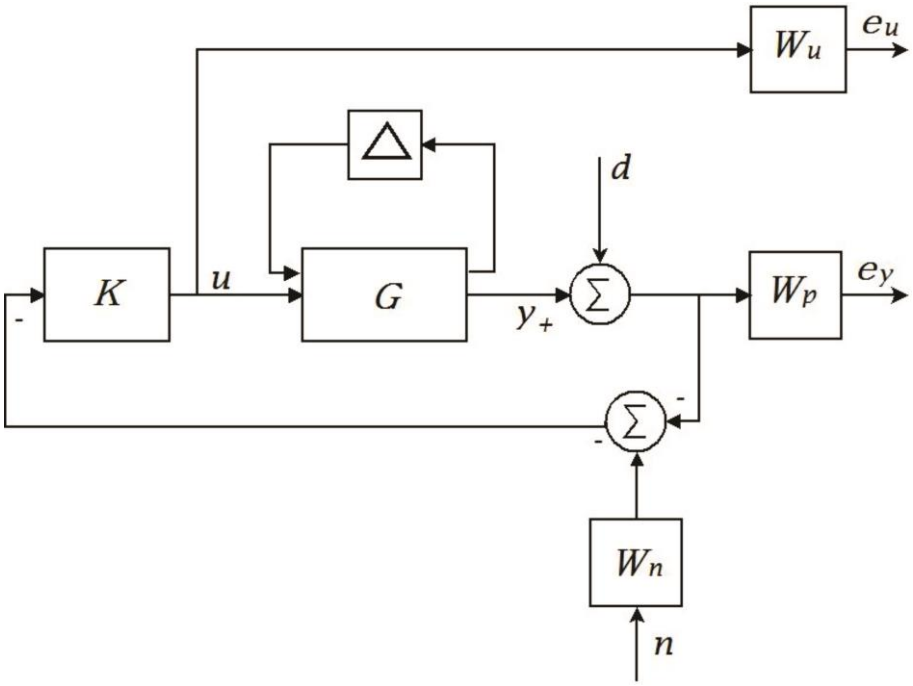

Figura 3. *Diagrama de bloques 3 GDL con incertidumbres estructuradas.*

El bloque Δ de incertidumbres del tipo estructuradas es expresado matemáticamente como:

$$\Delta = diag\{\delta_i\} = \begin{bmatrix} \delta_1 & 0 & 0 & 0 \\ 0 & \delta_2 & 0 & 0 \\ 0 & 0 & \delta_3 & 0 \\ 0 & 0 & 0 & \delta_4 \end{bmatrix} \quad (14)$$

Los parámetros inciertos en el presente trabajo se definieron de acuerdo a la posibilidad de representarlos en el experimento real. Así las masas del helicóptero y del contrapeso se pueden variar. De la misma manera la distancia del eje del ángulo de viaje hasta el contrapeso puede variar su longitud. La constante fuerza-empuje de la hélice se definió pensando a que el tiempo de uso puede llegar a variar de su valor nominal. El conjunto de parámetros considerados como incertidumbre estructurada en el modelo del helicóptero de 3 GDL se describen en la Tabla 2.

Parámetros	Valor nominal	Porcentaje de variación
K_f	0.1188 N/V	$\pm$ 5 %
m_h	1.15 Kg	$\pm$ 10 %
m_ω	1.87 Kg	$\pm$ 10 %
L_ω	0.4699 m	$\pm$ 20 %

Tabla 2. *Parámetros inciertos en el helicóptero de 3 GDL.*

En el diagrama de bloques de la Figura 3, W_p es una matriz de transferencia que incluye información sobre el rendimiento de los tres ángulos del helicóptero. Se seleccionaron entonces los elementos de esta matriz de tal manera que se obtuvieran los desempeños deseados de respuesta transitoria para los ángulos del helicóptero. W_p está definida de la siguiente manera:

$$W_p = \begin{bmatrix} \frac{0{,}15s + 0{,}99}{4{,}41s + 0{,}63} & 0 & 0 \\ 0 & \frac{0{,}1s + 0{,}97}{2{,}4s + 2{,}37} & 0 \\ 0 & 0 & \frac{0{,}13s + 0{,}9}{11{,}41s + 0{,}63} \end{bmatrix} \quad (15)$$

La posición (1,1) de W_p corresponde a la función de peso del ángulo de elevación, la posición (2,2) caracteriza al ángulo de cabeceo y finalmente la posición (3,3) al ángulo de viaje del helicóptero.

El bloque W_u es una matriz de transferencia de peso que caracteriza a la señal de control u. Esto es, da forma a la acción del controlador según los requerimientos del sistema. La matriz de transferencia W_n caracteriza los errores que los sensores pueden tener en el momento de la medición. La selección para la matriz de transferencia de peso W_u es:

$$W_u = \begin{bmatrix} \frac{15{,}4e^{-4}s + 41{,}42e^{-4}}{2s + 17{,}8} & 0 \\ 0 & \frac{15{,}4e^{-4}s + 41{,}42e^{-4}}{2s + 17{,}8} \end{bmatrix} \quad (16)$$

Una de las aplicaciones más comunes de los filtros pasa-altas es eliminar o reducir la información no deseada dentro de un ancho de banda especificado. Por esta razón, se incluye en el diseño del controlador una matriz de transferencia de peso W_n tipo filtro pasa-altas descrita como:

$$W_n = \begin{bmatrix} \frac{0{,}01s}{10s+0{,}7} & 0 & 0 \\ 0 & \frac{0{,}01s}{10s+0{,}7} & 0 \\ 0 & 0 & \frac{0{,}01s}{10s+0{,}7} \end{bmatrix} \quad (17)$$

La posición (1,1) del peso de W_n caracteriza al transductor de elevación. Por su parte el de (2,2) tiene relación con el encoder de cabeceo. Finalmente la función de transferencia en (3,3) se relaciona con la posición de codificación del ángulo de viaje.

4. Resultados

En el diseño por μ-síntesis se encuentra el controlador que minimiza un problema de optimización en términos del VSE. Sin embargo, no existe un método directo para encontrar tal controlador. El procedimiento llamado iteraciones *D-K* combina la síntesis H_∞ y el μ-análisis para sintetizar este controlador robusto. La función *dksyn* de Matlab® lleva a cabo la μ-síntesis a través de las iteraciones *D-K* para el modelo linealizado del helicóptero de 3 GDL. Los resultados de las iteraciones se presentan en la Tabla 3. En la tercera iteración se obtiene un controlador de orden 132 con una gamma de 0.709. Se propone una reducción de orden por truncación balanceada para obtener un orden 16 (Zhou, K., & Doyle, J.C. 1998). La respuesta en frecuencia del controlador de orden completo y el de orden reducido se muestra en la Figura 4. Puede observarse que ambos controladores poseen las mismas características en frecuencia para todas las entradas y salidas. Así, se implementa el controlador de orden 16 en la plataforma del helicóptero de 3 GDL de Quanser Inc.

Número de iteración	1	2	3
Orden del controlador	14	112	132
Orden total de escalamiento D	0	98	118
Gama alcanzada	12.707	0.924	0.709
Valor μ pico	0.791	0.837	0.691

Tabla 3. *Resumen de las iteraciones D-K.*

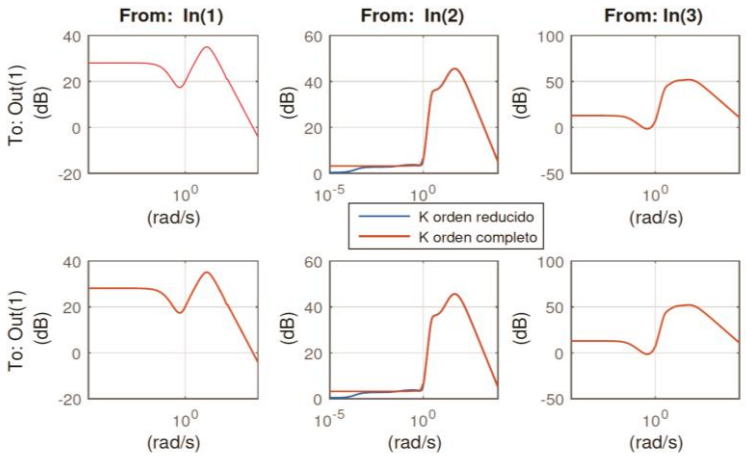

Figura 4. *Respuesta en frecuencia para el controlador de orden completo y reducido.*

El sistema de control en lazo cerrado para el helicóptero está desarrollado en Simulink® de MathWorks Inc. A través del software Quarc® se consigue descargar el controlador a bajo nivel de tal manera que pueda ser utilizado con una tarjeta de adquisición de datos. De esta manera el controlador se puede probar en tiempo real con el equipo experimental del helicóptero de 3 GDL (Quanser Consulting Inc. 2012). El controlador LQR incluido en el demo del equipo en Simulink® es reemplazado por el μ-controlador propuesto. Por lo tanto, se puntualiza que los resultados que se muestran en las siguientes figuras corresponden a experimentos en el equipo experimental.

Se buscó entonces que el controlador estabilice al helicóptero en la "*hover position*"}, es decir en la posición donde sus tres ángulos son cero. En la Figura 5 se presenta la estabilización del ángulo de elevación.

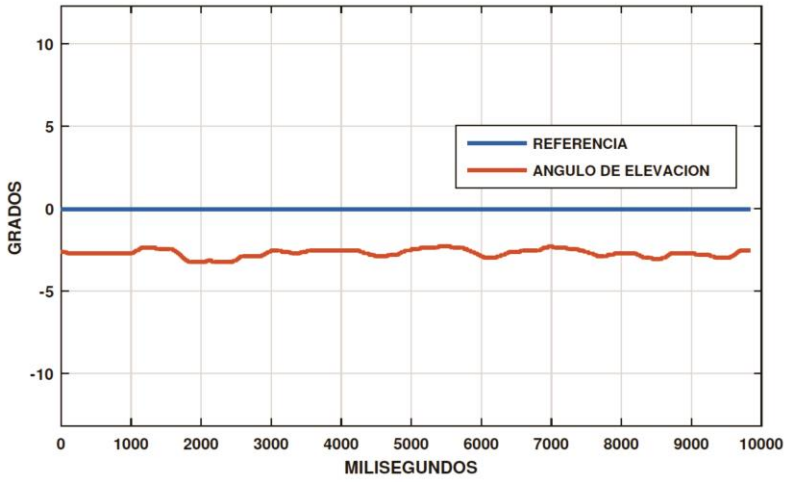

Figura 5. *Ángulo de elevación ε.*

Así mismo, en la Figura 6 se pueden encontrar los datos para el ángulo de cabeceo o inclinación. Como puede verse el ángulo de cabeceo es uno de los ángulos más difíciles de controlar. Para este caso, no es posible estabilizar este ángulo alrededor de los cero grados. Debido a que los dos motores propulsores están configurados para girar en la misma dirección. Como resultado la inclinación presenta un ángulo entre 4 y 6 grados durante la normal operación del sistema. De esta manera en los experimentos se notó que con el tiempo este ángulo tiende a acercarse a los 6 grados.

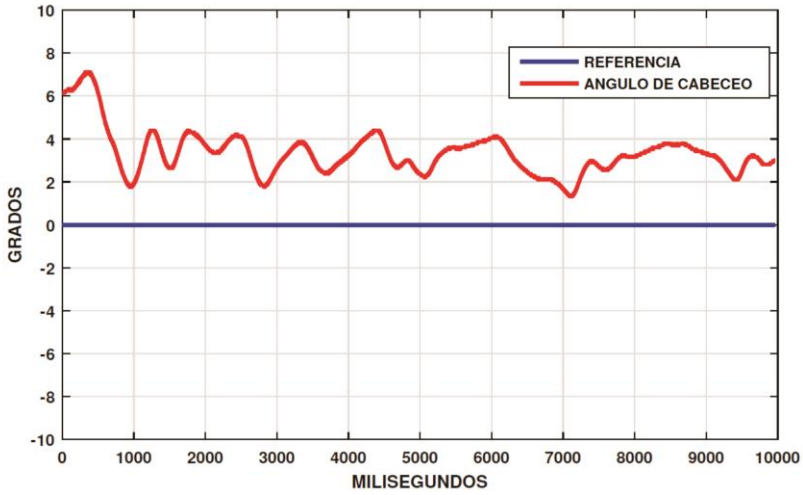

Figura 6. *Ángulo de cabeceo ρ.*

En la Figura 7 se aprecia la medición del ángulo de viaje. Estos resultados muestran que el ángulo de viaje es cercano a cero grados con respecto al ángulo de referencia.

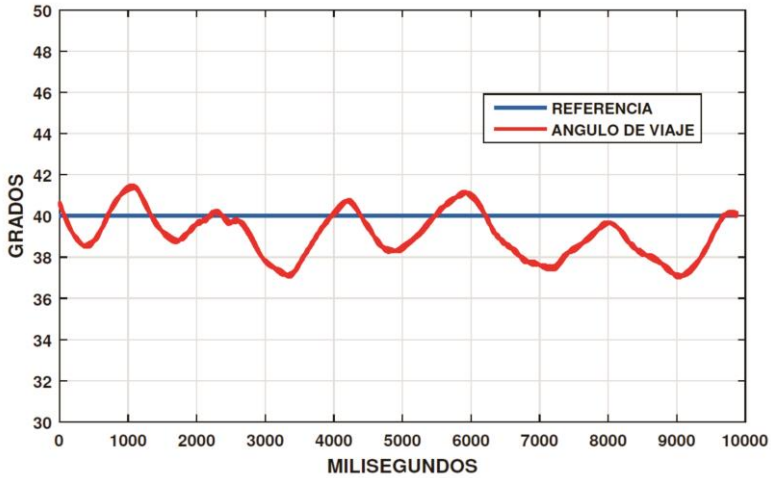

Figura 7. *Ángulo de viaje τ.*

Finalmente, la acción de control se grafica en la Figura 8. Los voltajes de los motores frontal y trasero, calculados por el controlador diseñado, se encuentran entre 15 y 16 voltios. Debe mencionarse que es complicado mostrar más tiempo

de ejecución en los experimentos ya que esto está limitado por la memoria de almacenamiento de los mismos.

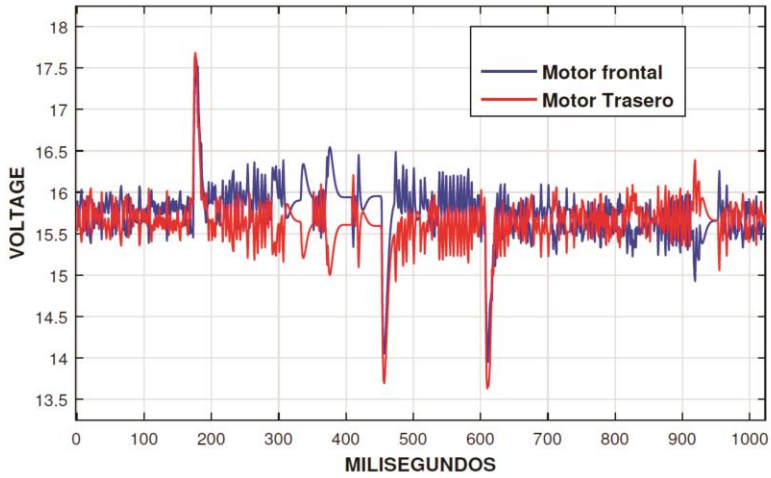

Figura 8. *Acción de control para ambos motores.*

5. Conclusiones

Este trabajo propone un controlador robusto utilizando la técnica de μ-síntesis para estabilizar el equipo experimental denominado Helicóptero de 3 GDL de Quanser Inc. Se definieron algunos parámetros del modelo del helicóptero con incertidumbres estructuradas. El diseño además contempla la inclusión de funciones de peso para reflejar los requisitos del sistema de control en lazo cerrado. Debe notarse que se trata de un sistema multivariable de 2 entradas y 3 salidas, lo que implica que las funciones de peso son matrices de transferencia. Por ejemplo, el peso W_p está compuesto en su diagonal principal de funciones de transferencia que reflejan las características de tiempo de asentamiento, porcentaje de sobredisparo para cada uno de los ángulos. Así, las tres funciones de transferencia son diferentes ya que los requisitos son distintos para cada ángulo. También se debe notar que aunque en la literatura existen diversas propuestas de controladores para este experimento, son muy pocas aquellas que muestran sus resultados en el equipo experimental. Así, la implementación en el prototipo de Quanser Inc. del controlador propuesto, permite constatar que este controlador diseñado por μ-síntesis exhibe un buen desempeño satisfactorio para la estabilización del helicóptero. Como trabajo próximo se trabaja en el diseño de un nuevo controlador que permita seguir no solo la estabilización del ángulo de viaje. Esto es, que alcance al menos entradas tipo escalón.

Referencias

Apkarian, Jacob & Lévis, Michel & Fulford, Cameron (2012) 3 DOF Helicopter Experiment for MATLAB/Simulink Users. Quanser Inc. Address: 119 Spy Court, Markham, Ontario, Canada. https://www.quanser.com/

Boukhnifer, M., Chaibet, A., & Larouci C. (2012) H-infinity robust control of 3-DOF helicopter. Proc. International Multi-conference on systems, Signals and Devices. Pages: 1-6. DOI: 10.1109/SSD.2012.6198011.

Da-Wei, G., Petko H. & Konstantinov, M. (2005) Robust Control Design with MATLAB®. Second Edition. Springer Publishing Company, Incorporated. ISBN: 1852339837, 978-1-85233-983-8.

Fang, Z., Gao, W., & Zhang, L. (2012) Robust Adaptive Integral Backstepping Control of a 3-DOF Helicopter. International Journal of Advanced Robotic Systems. Vol. 9, no. 79, Pages 1-8. DOI: 10.5772/50864.

Papageorgiou, George & Glover, Keith (1999) H-infinity loop shaping - Why is it a sensible procedure for designing robust flight controllers? American Institute of Aeronautics & Astronautics.

Quanser Consulting Inc. (2012) 3 DOF Helicopter, Workbook. Instructor Version.
Raptis, A. & Valavanis, P. (2011) Linear and Nonlinear Control of Small-Scale Unmanned Helicopters. International Series on Intelligent Systems, Control and Automation: Science and Engineering. Springer. DOI: 10.1007/978-94-007-0023-9_10.

Ríos, H., Rosales, J.A., Ferreira, A. & Iriarte, R. (2009) Control robusto por salida basado en un Observador por Modos Deslizantes para un Helicóptero de 3 grados de libertad. Congreso Anual 2009 de la Asociación de México de Control Automático. Páginas 1-6.

Vázquez, E., Pérez, E., & Hernández, D. (2017) Continuous Sliding-Mode Controllers for a 3-DOF Helicopter. 2017 14th International Conference on Electrical Engineering, Computing Science and Automatic Control. Pages 1-6.

Wang, X., Chen, Y., Lu, G., & Zhong Y. (2013) Robust flight control of small-scale unmanned helicopter. Proceedings 32nd Chinese Control Conference. Pages 2700-2705.

Zhou, F., Li, D., & Xia, P. (2009) Research of Fuzzy Control for Elevation Attitude of 3-DOF Helicopter. 2009 International Conference on Intelligent Human-Machine Systems and Cybernetics. Pages 367-370. DOI: 10.1109/IHMSC.2009.100.

Notas Biográficas

José Manuel Jiménez Mora. Recibió de la Universidad de Guadalajara el título de Ingeniero en Electrónica y Comunicaciones en el año 2006. Desde ese mismo año hasta el 2013 colaboró en diversas industrias en el sector privado, en el ámbito de la automatización. Actualmente es profesor de asignatura en los Departamentos de Electrónica y Matemáticas del CUCEI de la Universidad de Guadalajara. Sus intereses actuales incluyen: vehículos aéreos no tripulados, sistemas biomédicos, sistemas bio-inspirados y sistemas de control.

Eduardo Ruiz Velázquez. Recibió los grados de: Ingeniero en Electrónica (1998), Maestro en Ingeniería Eléctrica (Control Automático, 1999), Doctorado en Ingeniería Eléctrica (Control Automático, 2003) todos por la Universidad Autónoma de San Luis Potosí. Actualmente es Profesor-Investigador Titular en el Departamento de Electrónica, CUCEI, de la Universidad de Guadalajara. Sus principales intereses de investigación son: Modelado y Control de sistemas biomédicos, Control de glucosa en diabetes, identificación de sistemas, Control Robusto.

Gualberto Solís Perales. Obtuvo el grado de Ingeniero en Electrónica en la Facultad de Ciencias en la Universidad Autónoma de San Luis Potosí en 1995. Los grados de Maestría y Doctorado en Ingeniería Eléctrica en la Universidad Autónoma de San Luis Potosí en 1999 y 2002 respectivamente. Sus intereses de investigación incluyen: la Teoría de Control No Lineal y sus aplicaciones al control de Sistemas Complejos y la sincronización de Caos. Más precisamente, el control y sincronización de redes complejas y sus aplicaciones.

Julio Alberto García Rodríguez. Recibió el grado de Ingeniero en Electrónica (2010) por el Instituto Tecnológico de Ciudad Guzmán, Maestro en Ciencias en Ingeniería Electrónica y Computación (2012) por el CUCEI de la Universidad de Guadalajara. Actualmente es estudiante activo del Doctorado en Ciencias de la Electrónica y la Computación. Sus principales áreas de investigación son: Sistemas de control para Diabetes Tipo 1, modelado matemático de sistemas biomédicos, control robusto.