

CONTROL DE POSICION DE UN SEVOMOTOR DE LEGO® MINDSTORMS NXT
A TRAVES DE SEGUIMIENTO FACIAL UTILIZANDO UN SENSOR KINECT® DE
MICROSOFT

YURANNY CASTRO HERNANDEZ
JUAN CARLOS BEJARANO JIMENEZ

UNIVERSIDAD AUTONOMA DEL CARIBE
FACULTAD DE INGENIERIA
PROGRAMA DE INGENIERIA ELECTRONICA Y TELECOMUNICACIONES
PROGRAMA DE INGENIERIA MECATRONICA
BARRANQUILLA
2013

CONTROL DE POSICIÓN DE UN SEVOMOTOR DE LEGO® MINDSTORMS NXT
A TRAVÉS DE SEGUIMIENTO FACIAL UTILIZANDO UN SENSOR KINECT® DE
MICROSOFT

YURANNY CASTRO HERNANDEZ
JUAN CARLOS BEJARANO JIMENEZ

TRABAJO DE GRADO

Director(es):

Ing. José David Posada Aguilar, MSc

Ing. Jair Asir Villanueva Padilla, MSc

Asesor(a) Metodológico(a):

Ing. Mauricio Barrios Barrios, MSc

UNIVERSIDAD AUTONOMA DEL CARIBE
FACULTAD DE INGENIERIA
PROGRAMA DE INGENIERIA ELECTRONICA Y TELECOMUNICACIONES
PROGRAMA DE INGENIERIA MECATRONICA
BARRANQUILLA

2013

RESUMEN

El siguiente informe detalla el proceso de diseño e implementación de un sistema de detección y seguimiento facial desarrollado con un sensor Kinect® de Microsoft y un servomotor de LEGO® Mindstorms NXT utilizando técnicas de visión artificial y el software de Desarrollo LabVIEW; éste obtiene dos ángulos: el primero corresponde a la traslación y el segundo a la rotación de la cabeza del usuario sobre su propio eje, y posteriormente, controla la posición del motor referenciado en uno de los dos ángulos. El prototipo ha sido implementado para brindar a sistemas externos la posibilidad de tomar decisiones basados en seguimiento facial en aplicativos académicos e industriales. Además, se prevé que el modelo motivará el diseño de sistemas basados en visión artificial y reconocimiento de patrones en múltiples aplicaciones, inicialmente en estudiantes de Ingeniería.

Los resultados finales son satisfactorios evidenciando un buen comportamiento del sistema, salvo algunos retardos en su respuesta durante las pruebas. Nuevos retos en la investigación, promueven su implementación en dispositivos más accesibles como una cámara web y motores convencionales (que cuenten con características similares a los utilizados en este proyecto), con el fin de brindar mayor disponibilidad de uso sin tener que recurrir a software y/o controladores privados.

ABSTRACT

The following report details the entire process of design and implementation of a face detection and tracking system, developed with Microsoft ® Kinect sensor and an actuator of the LEGO ® Mindstorms NXT platform, using artificial vision techniques and the development software LabVIEW. This gets two angles referring to the translation of the head and the rotation thereof about its axis, and then controls the position of the engine, with reference on one of the two angles described above. The prototype has been implemented to provide external systems the ability to make decisions based on face tracking in academics and industrial applications. Also the model is expected to motivate the design of systems based on computer vision and pattern recognition in multiple applications, initially in engineering students.

The final results are satisfactory, showing a good overall system performance except for some delays in its response at specific times during testing. To continue in this line, we seek to implement the system based on affordable devices as a camera and conventional engines, but with characteristics similar to those used in this project, in order to increase availability without resorting to software and/or private drivers deprived of freedom to the overall system.

RESUMEN ANALÍTICO EJECUTIVO

(RAE)

TITULO: CONTROL DE POSICIÓN DE UN SERVOMOTOR DE LEGO® MINDSTORMS NXT A TRAVÉS DE SEGUIMIENTO FACIAL UTILIZANDO UN SENSOR KINECT® DE MICROSOFT.

TIPO DE DOCUMENTO: TRABAJO DE GRADO.

AUTORES: YURANNY CASTRO HERNÁNDEZ, JUAN CARLOS BEJARANO JIMÉNEZ.

EMPRESA PATROCINANTE: UNIVERSIDAD AUTÓNOMA DEL CARIBE.

LUGAR Y FECHA PRESENTACIÓN: BARRANQUILLA, 15 DE NOVIEMBRE DE 2013.

PALABRAS CLAVES: Inteligencia artificial, Sistema de control, Detección de Rostro, Seguimiento Facial, Sensor Kinect®.

DESCRIPCIÓN DEL CONTENIDO: DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE DETECCIÓN Y SEGUIMIENTO FACIAL DESARROLLADO CON UN SENSOR KINECT® DE MICROSOFT Y UN SERVOMOTOR DE LEGO® MINDSTORMS NXT, UTILIZANDO TÉCNICAS DE VISIÓN ARTIFICIAL Y EL SOFTWARE DE DESARROLLO LABVIEW.

METODOLOGÍA:

- **FUENTES:**

PRIMARIAS: CORRESPONDEN A LAS SUMINISTRADAS POR PERSONAS O ASESORES CON CONOCIMIENTO TÉCNICO EN EL ÁREA DE APLICACIÓN RELACIONADO CON ESTE PROYECTO.

SECUNDARIAS: LOS LIBROS Y SITIOS WEB QUE BRINDAN INFORMACIÓN SOBRE LAS METODOLOGÍAS RELACIONADAS CON EL SISTEMA A DESARROLLADO.

TIPO DE ESTUDIO: PARA EL DISEÑO E IMPLEMENTACIÓN DE ESTE SISTEMA SE HIZO NECESARIA UNA INVESTIGACIÓN DE CARÁCTER EXPLORATORIA, DEBIDO A QUE REQUIRIÓ LA BÚSQUEDA DE

INFORMACIÓN SOBRE LAS TÉCNICAS CON LAS QUE PODRÍA DESARROLLARSE LA PLATAFORMA PARA LUEGO SELECCIONAR LAS MÁS APROPIADAS PARA LA FASE DE DISEÑO.

METODO DE INVESTIGACIÓN: EL PROCESO INVESTIGATIVO INICIÓ CON LA REVISIÓN DE LITERATURA RELACIONADA CON LA MANIPULACIÓN DE KINECT Y LA IMPLEMENTACIÓN DE MODELOS DE CONTROL CON TIEMPOS DE RESPUESTA EFICIENTES EN TIEMPO REAL.

CONCLUSIONES Y RECOMENDACIONES: EL DESARROLLO DEL SISTEMA FUE EXITOSO GRACIAS A LA CORRECTA SELECCIÓN Y UTILIZACIÓN DE LAS PLATAFORMAS HARDWARE Y SOFTWARE. ADEMÁS LA IMPLEMENTACIÓN DE UNA EFECTIVA ESTRATEGIA DE CONTROL, MINIMIZARON LAS OSCILACIONES DURANTE LAS RESPUESTAS FRENTE A OTROS MODELOS DE CONTROL UTILIZADOS EN LAS PRIMERAS FASES DE PRUEBA. ESTO SE VIO REFLEJADO EN LA CORRECTA ORIENTACIÓN DEL SERVOMOTOR. PARA FURURAS INVESTIGACIONES SE DESEA IMPLEMENTAR EL SISTEMA EN DISPOSITIVOS MÁS ACCESIBLES EN EL MERCADO, CON EL OBJETIVO DE VALIDARLO COMO UNA PLATAFORMA QUE PUEDA SER UTILIZADA EN DIVERSAS APLICACIONES SEGÚN EL REQUERIMIENTO DEL USUARIO.

TABLA DE CONTENIDO

	Pág.
1. INTRODUCCION	10
2. DESCRIPCION DEL PROBLEMA	12
2.1 PLANTEAMIENTO DEL PROBLEMA	12
2.2 IMPACTO ESPERADO	13
2.3 USUARIOS DIRECTOS E INDIRECTOS.....	14
2.4 ESTADO DEL ARTE Y MARCO TEORICO	14
2.5 OBJETIVOS	35
2.5.1 Objetivo General.....	35
2.5.2 Objetivos Específicos.....	35
2.6 METODOLOGIA PROPUESTA.....	36
2.6.1 Fase de detección y seguimiento de rostro	36
2.6.2 Diseño del Algoritmo de control del servomotor	40
2.6.3 Diseño del Algoritmo de comunicación	43
3. ANALISIS DE RESULTADOS Y PROPUESTA INGENIERIL.....	45
3.1 PRUEBAS CON CONTROLADOR DEL MODULO DE LEGO®	45
3.2 PRUEBAS CON CONTROLADOR DIFUSO	48
3.3 DISCUSIÓN DE RESULTADOS	53
4. CONCLUSIONES Y RECOMENDACIONES	56
5. BIBLIOGRAFIA.....	57
ANEXOS.....	61

LISTA DE FIGURAS

	Pág.
Figura 1. Microsoft Kinect	21
Figura 2. Rango de alcance de Kinect®	23
Figura 3. Rango de visión en grados	23
Figura 4. Modo sentado y esqueleto completo	24
Figura 5. Rotación de las juntas.....	25
Figura 6. Sistema de coordenadas para detección facial	26
Figura 7. LEGO® Mindstorms NXT.....	27
Figura 8. Servomotor 9842	28
Figura 9. Entorno de desarrollo de LabVIEW	29
Figura 10. Esquema general de un Sistema Difuso.....	32
Figura 11. Conjunto Difuso inherente a la Altura.	33
Figura 12. Proceso de Fuzzificación	33
Figura 13. Diagrama de bloques del sistema.....	36
Figura 14. Diagrama para Algoritmo de detección y seguimiento de rostro.....	37
Figura 15. Diagrama para medición de distancias en el espacio	39
Figura 16. Funciones de membresía de entrada para el ángulo de traslación	41
Figura 17. Funciones de membresía de entrada para el ángulo de rotación	41
Figura 18. Función de membresía de la salida del sistema	42
Figura 19. Respuesta del controlador PI ante cambios de consigna manuales.....	46
Figura 20. Primera prueba de controlador PI con Kinect	46
Figura 21. Segunda prueba de controlador PI con Kinect	47
Figura 22. Tercera prueba de controlador PI con Kinect	47
Figura 23. Respuesta del controlador difuso ante cambios manuales.....	48
Figura 24. Comportamiento del sistema durante primera la prueba de traslación .	49
Figura 25. Error del sistema durante la primera prueba de traslación	50
Figura 26. Comportamiento del sistema durante segunda prueba de traslación ...	51
Figura 27. Error del sistema durante la segunda prueba de traslación.....	51

Figura 28. Comportamiento del sistema durante prueba de rotación52

Figura 29. Error del sistema durante la prueba de rotación53

1. INTRODUCCION

El desarrollo de nuevas tecnologías demanda el diseño sistemas capaces de integrar soluciones en tiempo real; emular comportamientos que generen la sensación de razonamiento lógico, es uno de los principales retos a los que se enfrentan los desarrolladores modernos.

Una de las principales tendencias que brinda a los sistemas esta posibilidad es la Inteligencia Artificial. La Inteligencia Artificial es un área relativamente moderna de la computación que permite a las máquinas proceder de manera similar a la humana teniendo en cuenta conceptos como racionalidad y razonamiento¹ (el sistema captura y procesa información en función de un objeto patrón). Estos sistemas están representados por modelos que cuentan con habilidades que le brindan autonomía en tiempo real que se ajustan a procesos industriales robustos.

Los procedimientos basados en Inteligencia Artificial brindan a las máquinas la capacidad de intentar emular los pensamientos y métodos de acción humanos; por esta razón son ampliamente utilizados en el diseño, por ejemplo, de modelos basados en robótica comportamental.

La detección y seguimiento facial mejor conocida en inglés como Face Detection and Tracking es uno de los métodos implementados cuando se desea emular comportamientos humanos, debido a que brindan la capacidad a los sistemas de integrar soluciones dinámicas en tiempo real de acuerdo a la localización facial de una persona en el espacio tridimensional (toman decisiones basados en el procesamiento de imágenes del mundo real). Es por esta razón que diseñó e implementó un sistema que determina la ubicación y rotación del rostro de un usuario sobre los ejes X, Y, y Z, y a partir de allí calcula la posición (en grados) a

¹JURADO Francisco et al. Desarrollo de videojuegos: Desarrollo de componentes. 1ra edición. Ciudad real: Bubok, 2012. p. 2.

la que debe orientarse un servomotor de LEGO® Mindstorms NXT por medio de un sistema de control, que se integraría a un sistema externo para desarrollar una función específica.

El desarrollo del sistema se dividió en tres fases:

- Algoritmo de detección y seguimiento de rostro.
- Algoritmo de control.
- Algoritmo de comunicación de los subsistemas anteriores.

El algoritmo de detección y seguimiento de rostro se desarrolló con ayuda de la librería SDK de Kinect® de Microsoft integrada al entorno de desarrollo Microsoft Visual Studio; a su vez este algoritmo se divide en la localización espacial del rostro y el cálculo de su orientación. El algoritmo de control se diseñó en el software de desarrollo LabVIEW y finalmente el algoritmo de comunicación donde se envían y reciben los datos obtenidos del software desde Visual Studio hacia LabVIEW, permite integrar las aplicaciones anteriores.

Se espera que el modelo sea considerado una plataforma de uso general en aplicativos académicos y en sistemas industriales, debido a los campos de acción a los que podría orientarse: prototipos robóticos (dotando a éste de la habilidad de seguir en tiempo real el rostro de una persona), control de acceso en edificaciones y en zonas de alto riesgo, domótica, sistemas de realidad virtual, entre otras.

2. DESCRIPCION DEL PROBLEMA

2.1 PLANTEAMIENTO DEL PROBLEMA

En años recientes se han desarrollado sistemas capaces de capturar las distintas partes cuerpo humano y a partir de sus movimientos faciales, producir una respuesta en función de ello ya sea solo para emularlos, o para generar una reacción en éste que promueva la interacción hombre-máquina. Son muchas las aplicaciones que pueden diseñarse para una plataforma con estas características; así por ejemplo en robótica comportamental, donde a partir de la detección y el seguimiento facial pueden producirse eventos como saber si el sujeto ha establecido contacto visual con éste, si lo está mirando fijamente, o incluso que el robot sea capaz de reaccionar ante expresiones faciales, lo que permite dotar al sistema robótico de habilidades para actuar de manera muy similar como lo haría un humano sometido a las mismas condiciones. Mientras tanto en el sector industrial, un sistema que ubique y capture el rostro de una persona es útil en plantas de procesos donde deban ejecutarse acciones preventivas en zonas altamente riesgosas.

En el mercado existen diversos modelos que cumplen funciones basadas en detección y seguimiento facial por lo que se está popularizando su uso. Sin embargo, encontrar uno capaz de integrar diversas aplicaciones en un mismo sistema no es tarea sencilla debido a que los costos de producción que se ven incrementados. La dificultad para desarrollar estos sistemas se debe a factores como:

- Tipo de hardware y de software.
- Calidad de los materiales.
- Costos de adquisición y mantenimiento.

Para empezar, los dispositivos que se emplean en una aplicación u otra pueden no ser compatibles con varias tecnologías o solo están predeterminadas para una en específico; además, es probable que los componentes del dispositivo no soporten ciertas condiciones para su correcta utilización en otro tipo de aplicación. Todos estos factores complejizan el desarrollo de un módulo multifuncional que es más usual encontrarlo en sistemas robustos, lo que supondría elevados costos de adquisición y mantenimiento.

Con la problemática descrita, se propone el diseño y la implementación de un sistema de bajo costo capaz de determinar la ubicación y rotación del rostro una persona en el espacio para orientar un servomotor de LEGO Mindstorms NXT, que a futuro y con las modificaciones idóneas, será implementado en aplicaciones donde el seguimiento facial sea requerimiento funcional del sistema.

La problemática y solución descritas anteriormente generan principalmente dos interrogantes que han sido resueltos gracias a la exitosa implementación del sistema que se propone. Éstos son:

¿Cómo controlar la posición de un servomotor LEGO® Mindstorms NXT a través de seguimiento facial en tiempo real?

Y a partir de esto: ¿Cómo brindar la capacidad a un sistema de detectar y seguir el rostro de una persona en tiempo real en el espacio?

Estas preguntas serán resueltas con el objetivo de brindar un comportamiento inteligente y funcional a cualquier que sistema que se integre al modelo que se propone para cumplir una función específica.

2.2 IMPACTO ESPERADO

El modelo que se propone espera ser aceptado como un prototipo de bajo costo que integre sistemas dirigidos a la interacción con humanos (por ejemplo robots); además gracias a su flexibilidad, a nivel industrial por ejemplo, permitiría

emprender acciones preventivas al tomar decisiones basados en la localización espacial de rostros de personas en zonas de alto riesgo donde esté restringido el acceso. Así mismo, y de acuerdo a los lineamientos investigativos que pretende abordar la facultad de Ingeniería de la Universidad Autónoma del Caribe, se espera que el modelo motive el diseño de nuevas aplicaciones basadas en Inteligencia Artificial y la utilización de técnicas de Control avanzado en procesos industriales robustos en estudiantes de la institución.

2.3 USUARIOS DIRECTOS E INDIRECTOS

De acuerdo a los lineamientos investigativos que están siendo explorados por el Programa de Ingeniería Electrónica y Telecomunicaciones de la Universidad Autónoma Del Caribe y su iniciativa de desarrollar proyectos que involucren diversas ramas de la ingeniería como la electrónica, la mecánica, la ingeniería del software, entre otras, los usuarios directos de esta propuesta de investigación son aquellos proyectos que deban integrar soluciones en tiempo real basadas en la localización espacial de rostros.

Así mismo los estudiantes de diversos programas académicos dentro y fuera de la institución y el sector industrial a través de sus aplicaciones en sistemas de seguridad, hacen parte de los usuarios indirectos del presente modelo de investigación.

2.4 ESTADO DEL ARTE Y MARCO TEORICO

El sistema de control de gestos y voz de Microsoft llamado Kinect®, ha probado que puede llegar mucho más lejos de lo que se esperaba. Desde su lanzamiento en noviembre de 2010 ingenieros, investigadores, músicos y científicos, comprendieron el potencial de este dispositivo constituido por dos cámaras, un sensor infrarrojo y una matriz de micrófonos, por lo que de inmediato comenzaron a utilizarlo. En menos de dos años después de su lanzamiento y con el software

adecuado, este sensor fue utilizado en tratamientos contra el Alzheimer, la esclerosis múltiple, el autismo, entre otros padecimientos.

Therapin2 es un sistema de terapia integral propuesto por Lucila Saccone, que busca apoyar los procesos de rehabilitación integral en niños autistas. El modelo fue ganador del Startup Weekend Villahermosa en 2012, evento reúne a emprendedores y diseñadores para explotar ideas de negocio. Therapin2 “es un muñeco virtual que va a dar terapia a los niños con autismo en tiempo real y de manera simultánea con el terapeuta que puede estar en otro estado o en otro país incluso” .Esta iniciativa promueve el desarrollo de habilidades en los menores como cepillarse los dientes y vestirse de forma autónoma bajo supervisión remota, lo que es ventajoso cuando no se cuenta con la presencia de un especialista. Kinect ha sido empleado además como herramienta de diagnóstico; así por ejemplo investigadores del Instituto de Desarrollo Infantil de la Universidad de Minnesota decidieron apoyarse en la tecnología que ofrece Kinect para diagnosticar el trastorno: mediante un experimento, los investigadores utilizaron cinco de estos sensores dentro de un salón de clases con grupos de niños entre tres y cinco años de edad para recopilar información inherente a sus movimientos corporales y de esta manera calcular la predisposición de padecimiento².

Lakeside Center for Autism es un centro asistencial para niños con autismo que utiliza Kinect® y otras herramientas para contribuir en las terapias específicas dentro del trastorno³. Otro claro ejemplo es “Baila conmigo”, un proyecto que se basa en las necesidades de la Corporación Síndrome de Down en Bogotá con la intención de desarrollar nuevas técnicas que fortalezcan la motricidad gruesa en pacientes con este padecimiento. “Baila Conmigo” ha sido puesto en marcha en 2013 por ingenieros del Grupo de Investigación Osiris de la Universidad del

² PSICOUTITERS.COM. Therapin2, juego de Kinect para niños con autismo [En línea]. 2013 [Citado 6-Dic-2013].

³ XBOX.COM. El efecto Kinectct: Lakeside Center forAutism [En línea]. 2013 [Citado 19-May-2013].

Bosque y utiliza un Kinect® para el control del juego que permite el reconocimiento del usuario frente a una pantalla que lee sus movimientos; con esto se consigue que niños entre 6 y 10 años de edad que padecen Síndrome de Down puedan continuar trabajando en la evolución de su motricidad gruesa a medida que van realizando actividades que los entretienen y divierten.

Baila conmigo incluye rutinas de ejercicios y secuencias de baile acompañados de música libre e instrumental aprobadas por especialistas de la corporación. Esta técnica se conoce como FNP (Facilitación Neuromuscular Propioceptiva), y es una metodología terapéutica para la obtención de respuestas precisas gracias a la estimulación de los propioceptores orgánicos, encargados de detectar el movimiento y posición de las articulaciones. Para iniciar la interacción el paciente debe ubicarse frente a la pantalla y una vez Kinect capture sus movimientos, éste podrá seguir la rutina que le dicta una figura de color negro (visualizada en la pantalla), que a su vez le indica cómo debe manejar los brazos y sus piernas, mientras el infante baila. Sus movimientos son detectados por una mujer que sigue el ritmo del menor⁴.

Por otro lado Royal Berkshire Hospital a través del programa de rehabilitación neurológica, ha incluido en 2012 el uso de esta plataforma en terapias de pacientes que han sufrido accidentes cerebrovasculares u otros traumatismos que han reducido su movilidad; con Kinect se proporciona un modelo integral que al reaccionar a comandos de voz y gestos corporales ha contribuido a la recuperación de la coordinación, la movilidad y el balance de los pacientes⁵.

Kinect ha sido utilizado no sólo en el desarrollo de aplicaciones para ordenador, sino además en el control de movimiento de sistemas externos. Tal es el caso de Quadrotor Micro-air Vehicle (MAV), un sistema de odometría visual en tiempo real

⁴ COLPRENSA, Un videojuego, nueva terapia para niños con síndrome de Down [En línea]. En: El universal. Bogotá, 10-Abr-2013.

⁵ REVISTA DINERS, El asombroso Kinect [En línea]. 2012 [Citado 6-Dic-2013]

desarrollado en 2011 por Albert S. Huang, Abraham Bachrach, Peter Henry, Michael Krainin, Daniel Maturana, Dieter Fox, Nicholas Roy que hace uso de este sensor para estimar y precisar trayectorias que son implementadas en vuelos autónomos. Inicialmente los primeros algoritmos de vuelo de MAV utilizaban LIDAR; LIDAR es la integración de tecnologías GPS, Unidad de Medición Inercial y sensor láser, implementado en la recolección de datos de altitud para definir la superficie del terreno y generar Modelos Digitales de Elevación (MDE), empleados por ejemplo en vehículos aéreos. Un sensor LIDAR transmite rayos láser hacia un objetivo mientras se mueve a través de rutas de topografía específicas. El reflejo del láser del objetivo es detectado y analizado por los receptores en el sensor LIDAR. Estos receptores registran el tiempo preciso desde que el pulso láser dejó el sistema hasta cuando regresó para calcular la distancia límite entre el sensor y el objetivo. Combinado con la información posicional (GPS e INS), estas medidas de distancia se transforman en medidas de puntos tridimensionales reales del objetivo reflector en el espacio del objeto. El mayor inconveniente que presentaba esta técnica es su incapacidad de detectar objetos que no se intersecan el plano de detección. Por esta razón decidió implementarse un sistema con cámaras RGB-D que iluminan una escena con un patrón de luz estructurada para estimar profundidades aún en zonas con poca visibilidad⁶.

La odometría visual se refiere a la estimación 3D del movimiento de un vehículo desde imágenes visuales. Esta técnica basada en el algoritmo básico utilizado por Moravec para detectar características de interés en cada fotograma de la cámara para estimar profundidades de objetos del entorno; con esto se reduce al mínimo el error entre píxeles para una reproyección en lugar de 3D.

Una reciente aplicación de la tecnología que ofrece Kinect es UBI Interactive, puesta a la venta en 2013; ésta brinda soporte táctil a una imagen creada a través

⁶ HUANG Albert et al, Visual Odometry and Mapping for Autonomous Flight Using an RGB-D Camera. En: International Symposium on Robotics Research (ISRR). 2011

de un proyector. Su requerimiento funcional sólo necesita un sensor Kinect, un proyector de imágenes y un ordenador con Windows 8. Esta aplicación funciona exitosamente gracias a que es capaz de reconocer gestos cotidianos: pellizcar para hacer acercamientos, deslizar, tocar un icono para activarlo, entre otros. Esta técnica puede implementarse en el sector empresarial, y en su versión básica permite la manipulación de pantallas de hasta 45", mientras que con una licencia paga pueden controlarse pantallas de hasta 100 pulgadas⁷.

La integración de Kinect con otros sistemas no se detiene. Actualmente está disponible de manera gratuita una versión del SDK de Kinect especialmente para el desarrollo en robótica. Este software denominado "Kinect Services for RDS" está disponible desde la página de Microsoft Research y como requerimiento funcional, debe instalarse en el ordenador donde será desarrollada la aplicación. Kinect Service incluye un ejemplo relacionado con la utilización del sensor en sistemas robóticos que se desplazan y evitan obstáculos. Servicios Kinect para RDS está construido en la cima de la Kinect para Windows SDK y hace uso de las funcionalidades de Kinect para que pueda implementarse con el modelo de programación CCR / DSS⁸.

Microsoft® Robotics Developer Studio 4 (RDS 4) es una versión más actual de su antecesor Microsoft® Robotics Developer Studio 4 (RDS 4) y presenta modificaciones como:

- Se ha actualizado a .NET 4.0.
- Simulador actualizado a XNA 4.0.
- Soporte para el sensor de Kinect a través de Kinect para Windows SDK V1.

⁷ ENGADGET, Ubi Interactive usa Kinect para convertir a cualquier superficie en una pantalla táctil [En línea]. 2013 [Citado 6-Dic-2013]

⁸ MICROSOFT ROBOTICS BLOG, Announcing Kinect Services for RDS 2008 R3 [En línea]. 2011 [Citado 6-Dic-2013]

- Una nueva referencia de diseño de la plataforma mediante el sensor de Kinect con servicios asociados, incluyendo un nuevo Dashboard Robot.
- Servicios Kinect simulados (RGB, profundidad e inclinación, más reconocimiento de voz utilizando el micrófono de matriz).
- Una plataforma de referencia simulada.
- Un ejemplo de implementación de un servicio de evitación de obstáculos Drive basado en Kinect, más sensores IR y Sonar opcionales
- A CCR para DLL de Silverlight 4.0.
- Nuevos Métodos de extensión para el CCR.
- Un nuevo servicio DSS Log Analyzer.
- Logging estructurado⁹.

RDS Microsoft Robotics Developer Studio está basado en CCR (Concurrency and Coordination Runtime) ; CCR es una biblioteca de código administrado que puede accederse usando .NET; está orientado a la manipulación de aplicaciones que cuentan con operaciones asíncronas frente a la concurrencia , explotar hardware paralelo y lidiar con el fracaso parcial; CCR permite el diseño de aplicaciones para que los módulos o componentes de software a través de un perfecto acople, que eventualmente significa que pueden desarrollarse de forma independiente y hacer suposiciones mínimas sobre su entorno de ejecución y otros componentes¹⁰. presentas notables ventajas con lo que respecta a la concurrencia, el fracaso y el aislamiento de una manera consistente.

Bellmoe, Ptucha y Savakis en 2011 diseñaron un sistema de visualización interactiva en tiempo real guiada por expresiones faciales y la orientación del rostro de un observador humano, donde el sensor Kinect® detecta y sigue las

⁹ MICROSOFT DEVELOPER NETWORK, New and Changed Features: Robotics Developer Studio 4 [En línea]. 2012 [Citado 6-Dic-2013]

¹⁰ MICROSOFT DEVELOPER NETWORK, CCR Introduction [En línea]. 2012 [Citado 6-Dic-2013]

articulaciones del sujeto, mientras una cámara RGB realiza un análisis facial más detallado. La visualización consiste en regiones activas que el observador puede manipular con gestos corporales y regiones aisladas, que a su vez son estimuladas a través de la orientación de la cabeza y expresiones faciales¹¹. Mientras tanto en 2012 se desarrolló un sistema de Face Detection and Tracking teniendo en cuenta la información de color de los rostros y su profundidad, lo que mejoró significativamente el rendimiento del algoritmo implementado¹². Como versión más óptima de esta aplicación, ese mismo año se diseñó un sistema de seguimiento facial a través de redes neuronales artificiales que detecta no solo las coordenadas en X y Y, sino además la orientación de un rostro, lo que permite determinar, por ejemplo, si el sujeto está mirando de frente o de perfil, especificando la ubicación de ojos, boca, entre otros¹³.

Una de las aplicaciones comerciales más conocidas de Kinect es 'MagicMirror' de Intel. Esta experiencia de compra virtual utiliza a un avatar del cliente vestido con las prendas que éste desea probarse. Una pantalla muestra el avatar que sigue fielmente los movimientos del sujeto en tiempo real y cambia las dimensiones del cuerpo a través gestos. Magic Mirror propone un sistema de espejo para los probadores que combina un sensor Kinect®, una tablet, una pantalla de 60" y un espejo¹⁴.

¹¹ BELLMORE Collin, Ptucha Raymond, Savakis Andreas. Interactive Display using depth and RGB sensors for face and gesture control. En: Western New York Image Processing Workshop. 2011. p. 5-8

¹² YANG Fie et al. Robust face tracking with a consumer depth camera. En: 19th IEEE International Conference on Image Processing. Orlando, FL, 2012. p. 561-564

¹³ ACHMAD Bañza, Firdausy Kartika. Neural Network-based Face Pose Tracking for Interactive Face Recognition System. En: International Journal on Advanced Science, Engineering and Information Technology, 2012, Vol. 2, No. 1. 105-108

¹⁴ INTEL. MagicMirror: Gesture Controlled Parametric Body Modeling [En línea]. 2013 [Citado 19-May-2013]

Ha quedado evidenciado el potencial de Kinect a través de las múltiples plataformas donde ha sido implementada su tecnología. Sin embargo, debe tenerse claro para su correcta utilización desde su constitución interna y su funcionamiento, hasta los requerimientos de uso del sistema. Este sensor que fue desarrollado por la compañía Microsoft y lanzado al mercado en 2010 , inicialmente se promocionó como un dispositivo adicional a la consola de juegos XBOX 360 también de la misma empresa, que permite la interacción con video juegos mediante control de gestos hechos por el usuario sin tener que sostener algún mando físico.

Su funcionamiento se divide en dos partes: la primera se basa en la detección y reconocimiento en un espacio 3D de personas, gestos, objetos, entre otros a través la obtención de mapas de profundidad y la segunda consiste en reconocimiento de voz a través de una matriz de micrófonos. El sistema de reconocimiento visual está constituido por un proyector láser y dos cámaras CMOS: el primero es una cámara monocroma que captura la luz infrarroja que permite el trazado de los mapas de profundidad, y la segunda es una cámara RGB que toma imágenes a color para la interfaz de usuario.

Figura 1. Microsoft Kinect



Fuente: EL TIPOGRAFO, Apple habría adquirido la compañía creadora de la tecnología de movimiento Kinect [En línea]. 2010 [Citado 6-Dic-2013]

Vistas las posibilidades que ofrece este sensor no solo en videojuegos sino además en el desarrollo de interfaces de usuario, Microsoft lanzó gratuitamente un Kit de Desarrollo de Software (Software development kit o SDK) para controlar todas las características del Kinect® desde un computador con Windows 7 o Windows 8, utilizando el entorno de desarrollo Microsoft Visual Studio (se recomienda la versión 2010 en adelante). Este SDK permite diseñar nuestras propias interfaces de usuario haciendo uso de los requerimientos sin requerir conexión del Kinect®. Algunas de sus características más destacadas son:

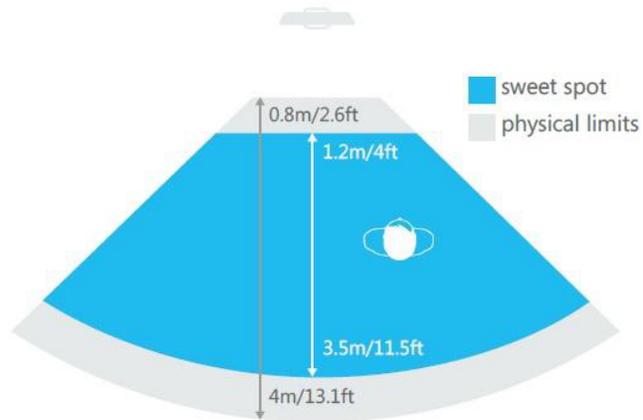
- Entendimiento del cuerpo humano: el SDK de Kinect® cuenta con una variedad de funciones donde se destaca un entendimiento profundo del ser humano, permitiéndole realizar seguimiento facial y de esqueleto, control de gestos, reconocimiento de voz, entre otros¹⁵.
- Manipulación completa de los datos: el SDK permite obtener en un arreglo los datos correspondientes a los mapas de profundidad, color de la cámara RGB y los esqueletos detectados por el sensor para su posterior manipulación.
- Flexibilidad de hardware: permite conectar hasta 4 sensores a la vez e implementarse en múltiples máquinas virtuales.

Si lo que se desea es iniciar las funciones de reconocimiento que brinda el sensor, es necesario que el usuario se ubique dentro del rango ideal (Sweet Spot 1,2 mts-3,5 mts) debido a que por fuera de estos rangos físicos, no puede garantizarse una correcta medición. La figura 2 muestra el rango de visión en el que el sensor puede detectar al usuario. Kinect® for Windows permite detectar y seguir hasta 6 sujetos a la vez y obtener los datos de 2 esqueletos enteros para seguimiento¹⁶.

¹⁵MICROSOFT. Kinect for Windows Features [En línea]. 2013 [Citado 20-Mar-2013].

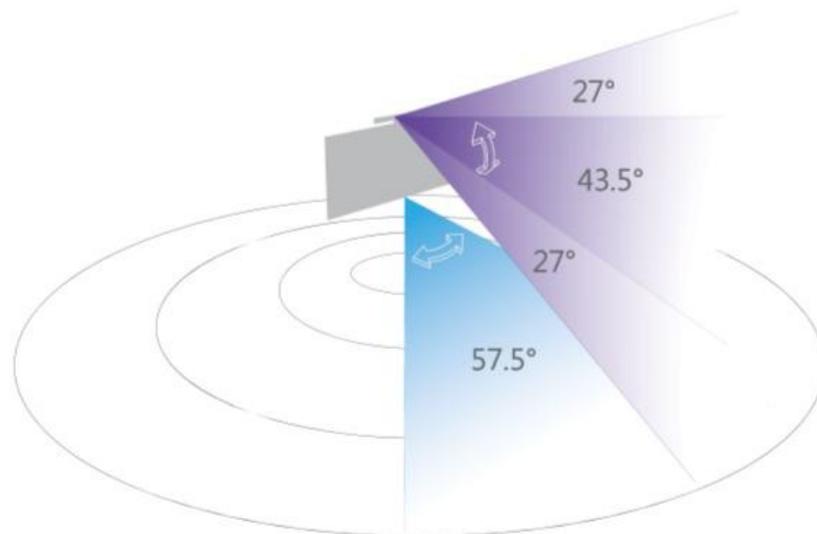
¹⁶MICROSOFT. Kinect for Windows: Human interface Guidelines v1.7. 2013. Pag 7.

Figura 2. Rango de alcance de Kinect®



Fuente: MICROSOFT. Kinect for Windows: Human interface Guidelines v1.7. 2013. Pag 7.

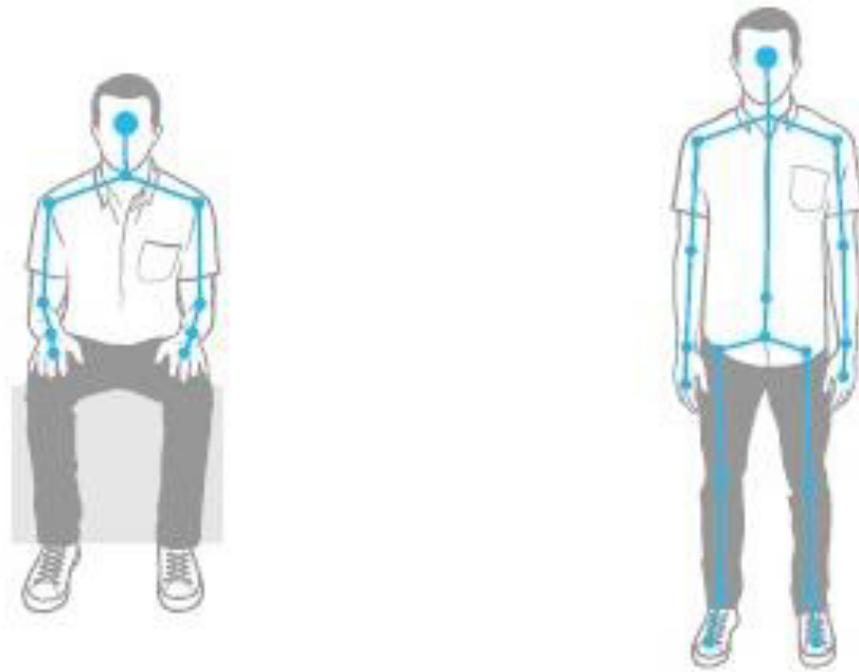
Figura 3. Rango de visión en grados



Fuente: MICROSOFT. Kinect for Windows: Human interface Guidelines v1.7. 2013. Pag 7.

Las funciones de entendimiento del cuerpo humano que ofrece Kinect a través del SDK de Kinect® for Windows permite la detección de máximo 20 articulaciones en modo completo o “full Skeleton mode”; además existe un segundo modo llamado “Modo sentado” o “Seated mode” donde se detectan las 10 articulaciones superiores del usuario¹⁷. A partir de esto pueden calcularse parámetros como la posición de cierta articulación en el área de trabajo de Kinect® (Sweet spot), orientación de algún hueso o junta a partir de los huesos conectados entre sí o la rotación absoluta del esqueleto completo (las rotaciones tanto de las juntas como del esqueleto completo siempre son sobre el eje Y del plano cartesiano, véase figura 5)¹⁸.

Figura 4. Modo sentado y esqueleto completo

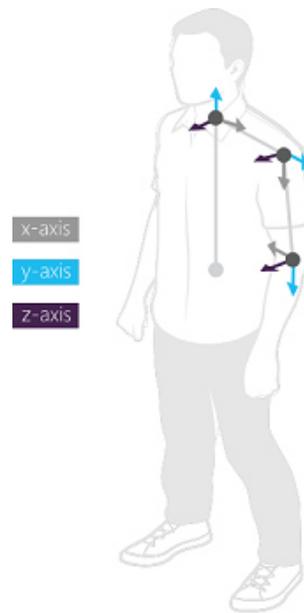


Fuente: MICROSOFT. Kinect for Windows: Human interface Guidelines v1.7. 2013. Pag 8

¹⁷Ibid., p. 8

¹⁸MICROSOFT DEVELOPMENT NETWORK. Joint Orientation [En línea]. 2013 [Citado 16-May-13].

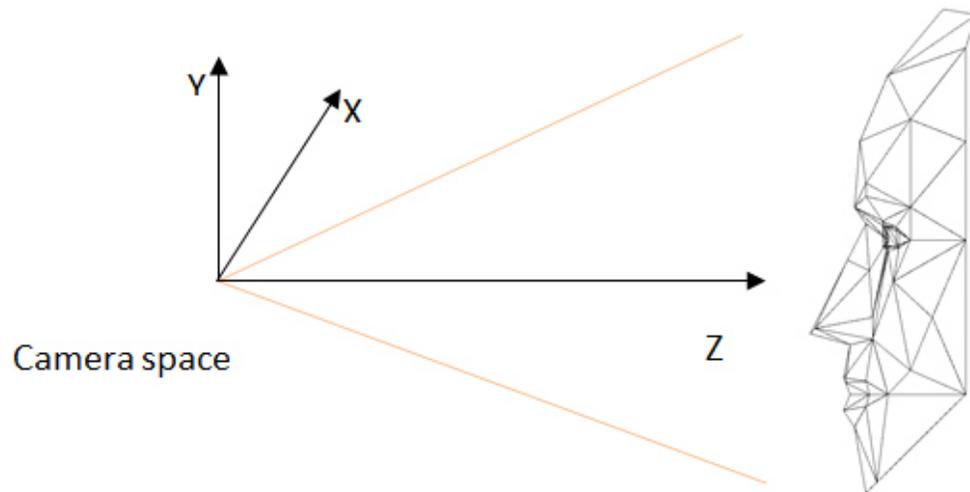
Figura 5. Rotación de las juntas



Fuente: MICROSOFT DEVELOPMENT NETWORK. Joint Orientation [En línea]. 2013 [Citado 16-May-13].

Otra de las funciones más destacadas de la tecnología que ofrece Kinect es la Detección y Seguimiento de Rostro. La librería Face Tracking dentro del SDK de Kinect® permite a partir de la detección del esqueleto, localizar el rostro del usuario dentro del área de visión del sensor; para ello es necesario además de la localización del esqueleto correspondiente a la cabeza, las tramas y los formatos tanto de imagen obtenidas de la cámara RGB como de profundidad por parte del sensor infrarrojo. Una vez detectado el rostro, se almacena toda la información dentro de una trama única y dedicada para éste, con la que pueden realizarse acciones como dibujar máscaras a partir de triángulos o completas que reemplacen el rostro del usuario en pantalla.

Figura 6. Sistema de coordenadas para detección facial



Fuente: MICROSOFT DEVELOPMENT NETWORK. Face Tracking [En línea]. 2013 [Citado 2-Nov-13]

Dentro de los grupos de información almacenados en las tramas de seguimiento facial se encuentra la posición del rostro dentro del espacio y su orientación, ambos compartiendo el mismo sistema de coordenadas ilustrado en la figura 6¹⁹.

Una vez asegurada la etapa de adquisición de datos y la ubicación espacial de las coordenadas del rostro y su orientación, es necesario definir las plataformas hardware y software que soportarán el modelo de control. En cuanto al hardware la plataforma utilizada es LEGO® Mindstorms NXT. Ésta es la sucesora de LEGO® RIS (Robotics Invention System) y salió al mercado en 2006 con diversas novedades en cuanto a las herramientas a las que permite manipular: USB, Bluetooth, LCD más grande, microcontrolador de 32 bits, además de sensores

¹⁹MICROSOFT DEVELOPMENT NETWORK. Face Tracking [En línea]. 2013 [Citado 2-Nov-13].

más precisos²⁰. El kit básico contiene la consola principal llamada bloque o ladrillo, 4 sensores, 3 servomotores y piezas para construir estructuras mecánicas que pueden acoplarse a otras piezas de otros productos LEGO®.

Figura 7. LEGO® Mindstorms NXT



Fuente: LEGOINFO. LEGO: historia [En línea]. [Citado 10-Dic-2013]

El servomotor incluido en el kit es de la referencia 9842, y es quién ejecuta las órdenes del sistema de control. Este servomotor es un actuador de alta precisión que posee un encoder interno con una sensibilidad de un grado por paso; parámetros como el ángulo y la velocidad de rotación pueden ser configurados desde el software que lo controle²¹.

²⁰CALVO Isidro, Perianez Gorka. Uso conjunto de la plataforma LEGO MINDSTORMS NXT y metodologías PBL en Informática Industrial. En: Ikastorratza, e-Revista de didáctica. 2010, No. 6. p. 3

²¹YANG Rouchen. Complex Behavior from a Simple Rule: Demonstration with Lego Mindstorms NXT Kit. Trabajo de grado (opta) Columbus: The Ohio State University. Department of Electrical and Computer Engineering, 2013. p. 12

Figura 8. Servomotor 9842



Fuente: YANG Rouchen. Complex Behavior from a Simple Rule: Demonstration with Lego Mindstorms NXT Kit. Trabajo de grado (opta) Columbus: The Ohio State University. Department of Electrical and Computer Engineering, 2013. p. 12

Mientras tanto *National Instruments* ha potencializado la programación de la plataforma *LEGO®* a través de su toolkit de LabVIEW para *LEGO® Mindstorms NXT²²*; gracias a este toolkit es posible programar la consola principal desde el entorno de programación gráfico característico de LabVIEW, haciendo mucho más fácil la comprensión del algoritmo a implementar a diferencia de otros lenguajes de programación basados en códigos.

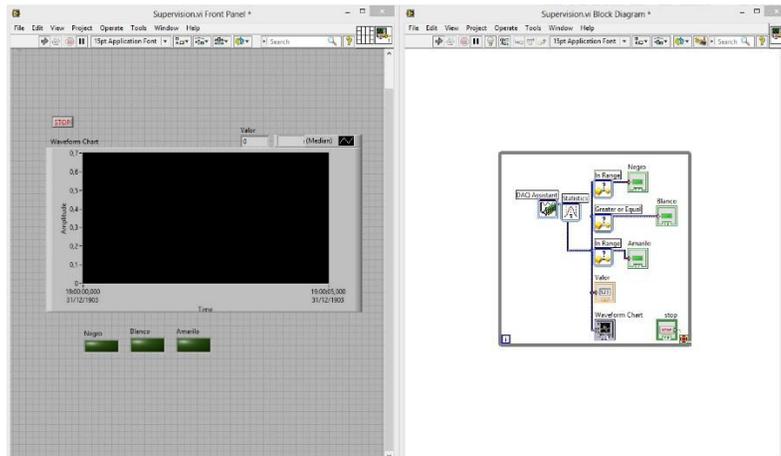
Laboratory Virtual Instrument Engineering Workbench mejor conocido como LabVIEW, es un lenguaje y entorno de programación gráfico que permite el desarrollo de aplicaciones con suma facilidad. Originalmente estaba orientado al control de instrumentación electrónica.

A los ficheros diseñados en LabVIEW se les denomina instrumentos virtuales (Virtual Instrumento VI) y tienen dos ventanas principales: la primera llamada Panel Frontal es la interfaz de usuario y cuenta con botones, pantallas y otros

²²NATIONAL INSTRUMENTS. NI LabVIEW para LEGO® MINDSTORMS® [En línea]. 2013. [Citado 18-May-2013]

controles; la segunda recibe el nombre de Diagrama de Bloques, y es donde se aloja el algoritmo del instrumento (homologo a la circuitería interna de éste).²³.

Figura 9. Entorno de desarrollo de LabVIEW



Fuente: Los autores

Las dos ventanas de cualquier archivo VI están entrelazadas a través de terminales que representan los elementos utilizados por el instrumento; estos tienen un bloque específico en la ventana de diagrama de bloques. El algoritmo implementado actuará sobre el terminal alojado en la ventana del panel frontal.

Las dos ventanas de cualquier archivo VI están entrelazadas a través de terminales que representan los elementos utilizados por el instrumento; éstos tienen un bloque específico en la ventana de diagrama de bloques. El algoritmo implementado actuará sobre el terminal alojado en la ventana del panel frontal.

La selección de LabVIEW como la plataforma hardware que soportará el algoritmo de control, se fundamenta en las herramientas que suministra para la manipulación de todos los componentes LEGO. Estas herramientas permiten la

²³LAJARA José, Pelegrí José. LabVIEW Entorno gráfico de programación. 1ra edición. Mexico D.F: Alfaomega, 2007. p. 4.

interacción del lenguaje código desde Visual Studio (entorno de desarrollo utilizado en la etapa de adquisición con Kinect) y los diagramas de bloques característicos de LabVIEW; una de los instrumentos que permite esta interacción es el protocolo UDP. Este protocolo pertenece a la capa de transporte del modelo TCP/IP y es la versión simplificada de TCP (Transmission Control Protocol). No tiene control de flujo, fidelidad o corrección de errores en el envío de datos, en cambio, se convierte en un protocolo rápido, ideal para aplicaciones con baja transmisión de información sin retorno (Comunicación Simplex)²⁴.

Para realizar una conexión UDP es necesario conocer la dirección IP y el puerto al que están asociadas las aplicaciones. LabVIEW dispone de una paleta de funciones para comunicación TCP y UDP donde se introduce la IP y los datos a enviar en forma de caracteres (los datos recibidos también son del mismo tipo)²⁵. Así mismo, Microsoft provee librerías para transmisión y recepción de datagramas a través de Visual Studio; esto brinda la posibilidad de establecer comunicación entre ambas aplicaciones permitiendo transmitir datos de algoritmos hechos en código en Visual Studio hacia el diagrama de bloques en LabVIEW y viceversa.

Además de la selección de las plataformas hardware y software que soportarán la aplicación desde la fase de adquisición de datos hasta la orientación del servomotor, es necesario consolidar la estrategia de control a implementar. Uno de los algoritmos de control tradicionalmente implementados en la industria con casi 60 años de antigüedad, es el control Proporcional–Integral–Derivativo (PID). El control PID ofrece una buena alternativa para controlar procesos dinámicos con tiempos de respuesta rápidos y un proceso de ajuste relativamente simple. Sin embargo, al asumir no linealidades se hace más complejo la implementación de esta configuración.

²⁴Ibid. p. 190

²⁵Ibid. p. 191

Como una alternativa eficiente al tradicional uso de controladores PID surgió el Control Lógico Difuso. La Lógica Difusa, conocida en inglés como *Fuzzy Logic* es un modelo matemático desarrollado por el ingeniero Lotfi Zaded de la Universidad de Berkeley (California- USA) que permite resolver problemas de manera imprecisa a través del uso de términos lingüísticos; se basa en conceptos imprecisos que permite a las máquinas proceder de manera similar a la humana. Se basa en dos principios fundamentales:

- Permite presentar a las máquinas una alternativa de solución de problemas de forma similar a la humana, lo que ofrece un bajo coste computacional
- La mayor parte de elementos son representados por grados de pertenencia, dentro de unos conjuntos definidos: el concepto de Lógica Difusa, permite la existencia de valores intermedios (en la teoría de conjuntos tradicional, un elemento pertenece absolutamente o no pertenece a él), sin embargo, en este modelo matemático un elemento puede pertenecer a distintos conjuntos con diferentes grados de pertenencia; así por ejemplo en el intervalo $[0,1]$, 1 representa la pertenencia absoluta, mientras 0 el caso contrario²⁶.

La implementación de este modelo matemático, se viene desarrollando con éxito desde 1.965 en aplicaciones²⁷, tales como:

- Sistema de control inteligente de aires acondicionados.
- Electrodomésticos.
- Control industrial
- Control electrónico.
- Sistemas de escritura.
- Mejora de la eficiencia en el consumo de combustible en vehículos.
- Desarrollo de videojuegos.

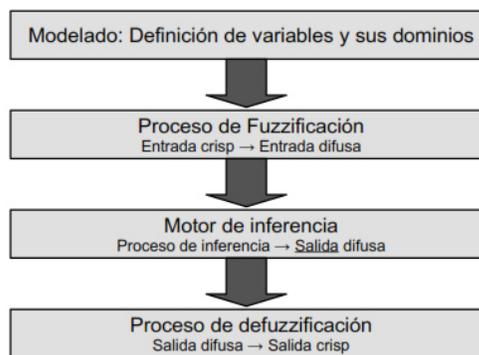
²⁶ Jurado. Op. Cit., p. 14.

²⁷ Ibid., p. 16.

- Comercio electrónico.
- *Tecnologías informáticas.*
- *Sistemas expertos que emulan comportamientos humanos.*

En esencia, un sistema difuso, consta de 4 partes: una de modelado, una de fuzzificación, una de interferencia y una de desfuzzificación²⁸.

Figura 10. Esquema general de un Sistema Difuso.



Fuente: Jurado Francisco et al. Desarrollo de videojuegos: Desarrollo de componentes. 1ra edición. Ciudad real: Bubok, 2012. p. 19.

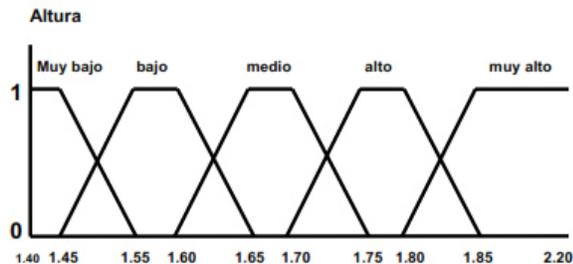
- Fase de Modelado

Es aquí donde son definidas las variables de entrada que más adelante serán utilizadas en la fase de Interferencia. Es necesario definir el dominio de cada una de las variables, que es especificado dentro de rango de valores numéricos que estas pueden tomar y los conjuntos difusos. Los conjuntos difusos abarcan un rango de valores y se les asocia una etiqueta lingüística; en función de las características del problema a modelar, los conjuntos difusos pueden adoptar diferentes formas: triangulares, trapezoidales, curvilíneos, entre otras.

²⁸ Ibid., p. 18.

En la Figura 4 se muestra un conjunto Difuso inherente a la altura de una persona. Aquí es definido un conjunto difuso para la variable altura.

Figura 11. Conjunto Difuso inherente a la Altura.

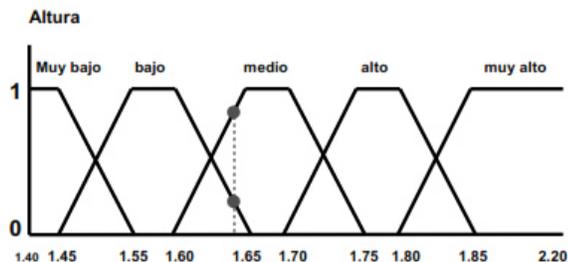


Fuente: Jurado Francisco et al. Desarrollo de videojuegos: Desarrollo de componentes. 1ra edición. Ciudad real: Bubok, 2012. p. 19.

- Fase de Fuzzificación

Se desarrolla una vez son definidas las variables y sus conjuntos de pertenencia. Es en esta fase donde son evaluados los valores de pertenencia en función de los valores de entrada. Así en la figura 3 es posible inferir que por ejemplo una persona con una altura de 1.65 mts puede ser considerada con cierto grado de pertenencia como Baja o de estatura Media.

Figura 12. Proceso de Fuzzificación



Fuente: Jurado Francisco et al. Desarrollo de videojuegos: Desarrollo de componentes. 1ra edición. Ciudad real: Bubok, 2012. p. 20.

- Fase de Interferencia

Una vez es evaluada la pertenencia de las variables de entrada a los conjuntos difusos, inicia el proceso de razonamiento o interferencia que permitirá tomar decisiones en función de ellas. En esta etapa, un motor muy usual de razonamiento se basa en el uso de Reglas Difusas de tipo SI-ENTONCES²⁹. Este tipo de reglas están constituidas por un conjunto de antecedentes conectados a través de operadores lógicos (AND, OR, NOT) y uno o varios consecuentes; cada consecuente consiste en una variable difusa que el rango lingüístico que esta debería tomar para que sea activada la regla.

Así por ejemplo: un videojuego donde deba determinarse la fuerza con la que se deba atacar al enemigo dependiendo de la distancia a la que se encuentre de este. Para que la regla “SI distancia {corta o muy corta} y Fuerza_enemigo {baja, muy baja} ENTONCES Ataque {agresivo} sea activada, la distancia entre el enemigo y el personaje principal debe tener un grado de pertenencia a cero en los conjuntos corta, muy corta. Así mismo la fuerza del enemigo debe clasificar en el caso alta, media o baja; finalmente, en caso de ser afirmativo, el personaje atacará con cierto grado de fuerza en función de la distancia de su enemigo³⁰.

- Fase de Defuzzificación

Solo es necesario si se desea conocer un valor real de salida en el sistema.

En aplicaciones como robótica, es útil el diseño de controladores cuya fuente de información proviene del procesamiento de imágenes del mundo tridimensional; y precisamente esta rama de la tecnología, una de las que hace uso de todos los procedimientos expuestos con anterioridad.

²⁹ Ibid., p. 21.

³⁰ Ibid., p. 21.

Una vez identificadas las plataformas hardware y software y seleccionada la estrategia de control a implementar, se procederá al diseño del sistema orientado al control de posición de un servomotor basado en detección y seguimiento facial utilizando lógica difusa.

2.5 OBJETIVOS

2.5.1 Objetivo General

Diseñar un sistema de control que oriente la posición de un servomotor LEGO® MINDSTORM NXT a través de la detección y seguimiento del rostro de una persona utilizando un sensor Kinect® de Microsoft.

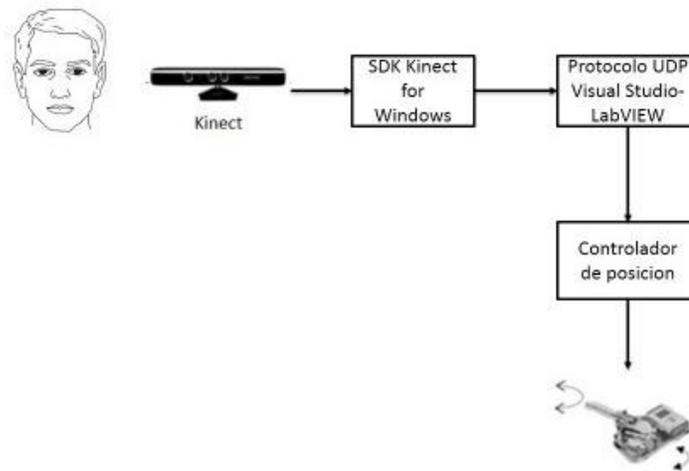
2.5.2 Objetivos Específicos

- Analizar las funcionalidades requeridas para el desarrollo del sistema de control de posición de un servomotor de LEGO® MINDSTORMS NXT.
- Analizar y determinar la plataforma tecnológica más adecuada para la implementación del modelo de control.
- Diseñar e implementar un algoritmo que haga uso de las herramientas provistas por el sensor Kinect® de Microsoft para calcular el ángulo correspondiente a la posición del rostro de una persona en un rango comprendido entre 1 m y 2.5 m.
- Diseñar e implementar un algoritmo de control que oriente un servomotor LEGO® MINDSTORM NXT hacia una posición angular dada.
- Validar y retroalimentar el funcionamiento de la estrategia en un entorno controlado.
- Documentar de manera detallada el proceso general de diseño y montaje de los equipos, herramientas y software para futuros participantes en proyectos similares.

2.6 METODOLOGIA PROPUESTA

El desarrollo del sistema se divide en 3 fases: diseño del algoritmo de detección y seguimiento de rostro, diseño del algoritmo de control del servomotor y diseño del algoritmo de integración de las dos primeras aplicaciones a través del software de desarrollo LabVIEW y el protocolo UDP (*User Datagram Protocol*). A continuación serán descritas cada una de ellas.

Figura 13. Diagrama de bloques del sistema



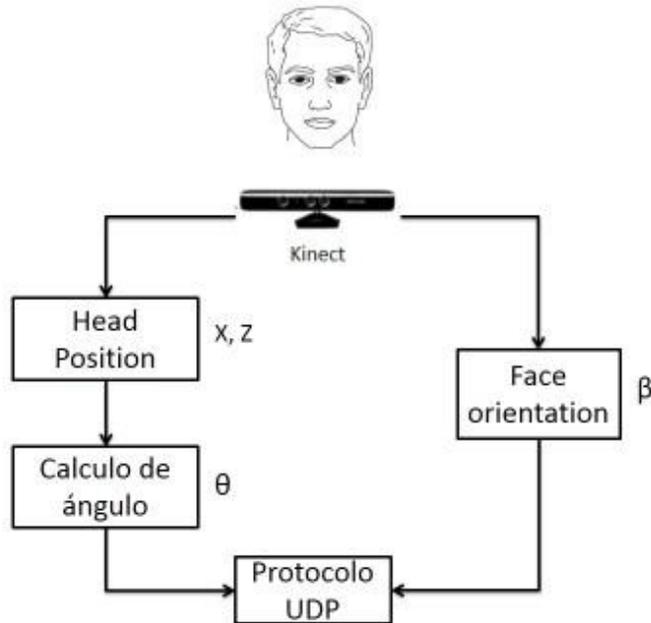
Fuente: Los autores

2.6.1 Fase de detección y seguimiento de rostro

La figura 14 ilustra el algoritmo de detección y seguimiento de rostro que se divide en dos etapas: la primera consiste en la localización del rostro en el espacio cartesiano (en los ejes X y Z) y en la segunda se procede a determinar el ángulo de rotación sobre el eje Y según el sistema cartesiano del sensor. Se desarrolló un algoritmo en C# tomando como referencia la aplicación Face Tracking Basics³¹ disponible en el SDK de Kinect®.

³¹MICROSOFT DEVELOPMENT NETWORK. Face Tracking Basics-WPF C# Sample [En línea]. 2013 [Citado 16-May-13].

Figura 14. Diagrama para Algoritmo de detección y seguimiento de rostro



Fuente: los autores

El código implementado utiliza la librería de *Face Tracking* para localizar la articulación del esqueleto detectado correspondiente a la cabeza del usuario y el desarrollo posterior de las funciones principales mencionadas con anterioridad; Kinect® tiene la capacidad de cambiar los esqueletos a detectar con un máximo de dos de forma simultánea; una vez alcanzado ese límite, no se permite detectar más esqueletos (en este proyecto solo se detectará un esqueleto para iniciar el seguimiento).

2.6.1.1 Localización espacial del rostro

La primera labor que debe llevar a cabo el sensor es encontrar el esqueleto del usuario; seguido a esto, es posible obtener la posición dentro del área de trabajo del Kinect® (*Sweet Spot*) de una articulación determinada o del esqueleto completo; el SDK provee las herramientas para cifrar las distancias en el plano

cartesiano; la figura 15 muestra la disposición del sistema de coordenadas, el usuario idealmente se puede encontrar a una distancia de 0 metros en el eje paralelo al sensor, es decir el eje X, independiente de que tanto se encuentre alejado la persona, esa distancia longitudinal es marcada por el eje Z, esta posición la cual fue denominada A, garantizaría un ángulo inicial de 0 grados, debido a que se encontraría en la misma línea de referencia del Kinect. Seguidamente el usuario puede desplazarse a una nueva posición denominada A', esta vez con una distancia determinada en el eje X y en el eje Z, la distribución de ambos ejes en el espacio formarían los catetos de un triángulo imaginario, en el que para completarlo se necesitaría calcular la hipotenusa (la distancia desde la posición exacta del usuario hasta la ubicación del sensor) por medio de la ecuación 1, y completado el triángulo se utiliza la ecuación 2 para calcular el ángulo del vértice entre el eje Z y la hipotenusa; este ángulo denominado θ es el que representa el movimiento de la cabeza de la persona a lo largo y ancho del espacio de trabajo del sensor Kinect y será utilizado como la primera referencia para controlar el servomotor de LEGO Mindstorms NXT.

Ecuación 1. Teorema de Pitágoras para calcular la hipotenusa

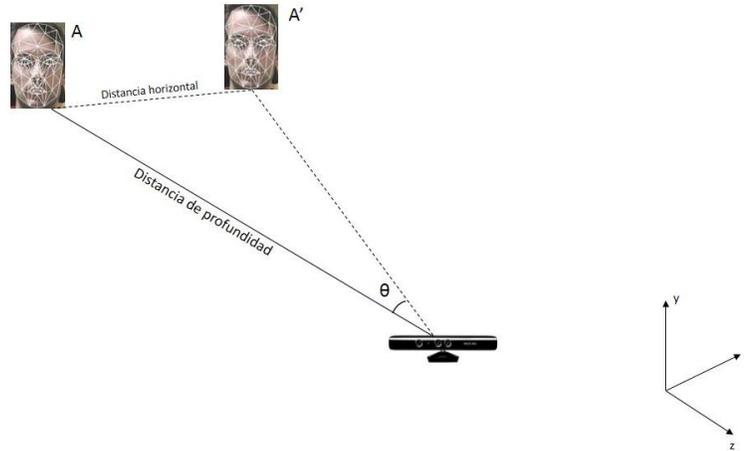
$$H = \sqrt{X^2 + Z^2}$$

Una vez detectado el rostro y su ubicación (coordenadas en X y Z), se tomará parte la librería de Face Tracking Basics para dibujar la máscara sobre el rostro del usuario en la interfaz de usuario y para dar inicio el seguimiento facial.

Ecuación 2. Calculo del ángulo de rotación

$$\theta = \sin^{-1} \frac{X}{H}$$

Figura 15. Diagrama para medición de distancias en el espacio



Fuente: Los autores

2.6.1.2 Orientación del rostro

Una vez calculado el ángulo correspondiente a la ubicación espacial del rostro dentro del rango de visión del Kinect®, se procede a determinar el ángulo de orientación de la cabeza cuando gira sobre su propio eje (en la figura 6 se ilustró el sistema de coordenadas que se emplea para la detección y seguimiento facial por parte de Kinect®); el eje es paralelo al cuello, por tanto es sobre el eje Y sobre el que se gira la cabeza, lo que supondría obtener este ángulo con la información contenida en la trama de seguimiento facial.

El anexo A1 muestra el código completo del proceso escrito en C#; primero se obtiene el ángulo de rotación en grados a través de la propiedad *Rotation* en la trama que entrega un vector con la información de los tres ejes (éste permite seleccionar cualquiera de los ejes; se escogió el eje Y). Seguidamente se elige una junta del esqueleto obtenido (en este caso la cabeza), y se obtiene su ubicación en los ejes X y Z mediante la propiedad *Position* determinando su distancia en metros. Finalmente se procede a utilizar las ecuaciones 1 y 2 para

determinar el ángulo de traslación inicialmente en radianes, por lo que se convierte inmediatamente a grados para su posterior uso en los controladores.

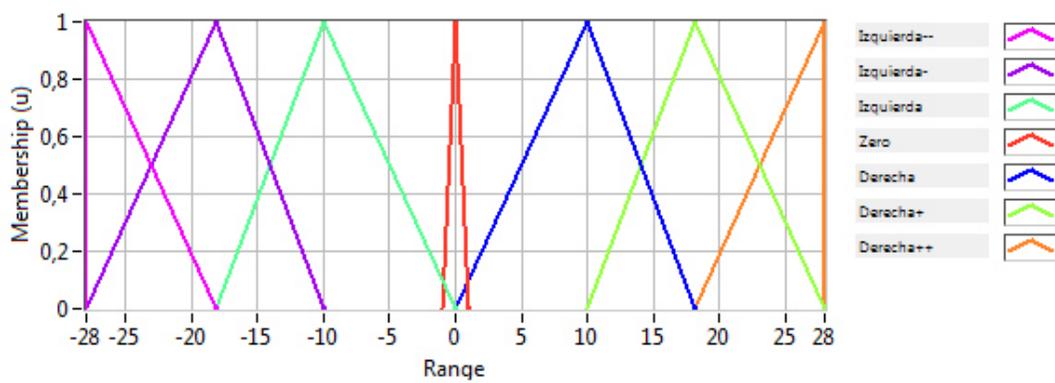
2.6.2 Diseño del Algoritmo de control del servomotor

Como primera alternativa para la etapa de control, se implementó primero un controlador PI ofrecido por el módulo de LEGO Mindstorms NXT, el diagrama de bloques puede verse en el anexo B2; se realizaron las pruebas correspondientes y se comprobó que aunque el sistema no siempre muestra el rendimiento esperado, marca un buen camino y un punto de referencia para el diseño del siguiente controlador. La decisión de diseñar un sistema de control a partir de la lógica difusa se basó en que los sistemas difusos tienen características importantes como una mejor versatilidad en el manejo de datos que un controlador con configuración PID convencional, que nos permite decidir no solo los valores tanto de las entradas como de las salidas del sistema, sino también determinar cómo será el comportamiento de las salidas con relación al de una o unas entradas específicas; presenta una mejor robustez para controlar plantas donde se presenten retrasos en la señal y además es un sistema de control ideal para implementarse en plataformas que empleen la lógica comportamental.

Se diseñaron dos controladores difusos en LabVIEW con la ayuda del toolkit de control PID y lógica difusa (el diagrama de bloques correspondiente puede apreciarse en el anexo B1); uno para la traslación de la cabeza del usuario en el espacio y otro para la rotación de la misma en su propio eje, Una vez recibidos los ángulos por parte del algoritmo de detección y seguimiento de rostro, los controladores difusos obtienen su entrada a partir el error del sistema, es decir, la diferencia entre la referencia (ángulo de traslación o de rotación de la cabeza) y la lectura del encoder del servomotor manipulado, y envían como salida hacia el motor la potencia a la que éste debe moverse. El primer paso para el diseño de los controladores fue determinar la forma de fuzzificación de los datos. Para este proceso las funciones de membresía son del tipo triangular, ya que se buscó un comportamiento en la salida donde esta incrementara o disminuyera su valor

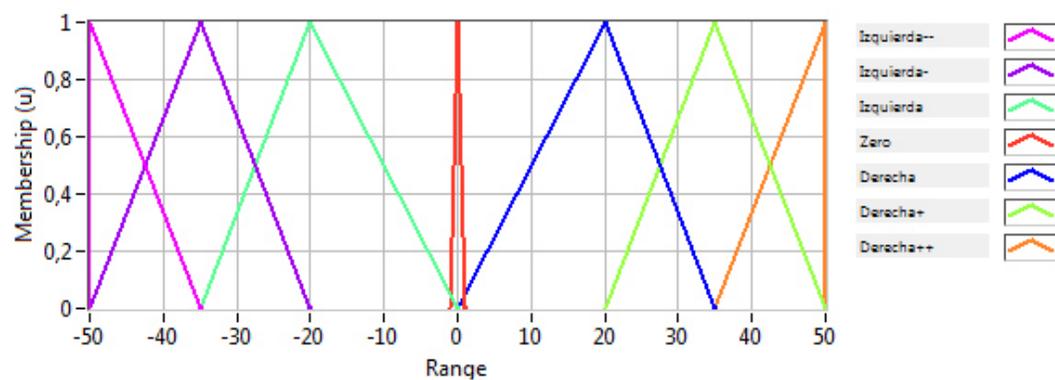
gradualmente dependiendo de la lectura de las variables de entrada, quienes también tiene este tipo de comportamiento, las figuras 16 y 17 muestran las dos variables de entrada del controlador de traslación y de rotación respectivamente con sus funciones de membresía, son muy parecidas en su forma pero su principal diferencia radica en los rangos de acción medido en grados, mientras que el rango de entrada del controlador para traslación es de ± 28 , el de rotación es de ± 50 .

Figura 16. Funciones de membresía de entrada para el ángulo de traslación



Fuente: Los autores

Figura 17. Funciones de membresía de entrada para el ángulo de rotación



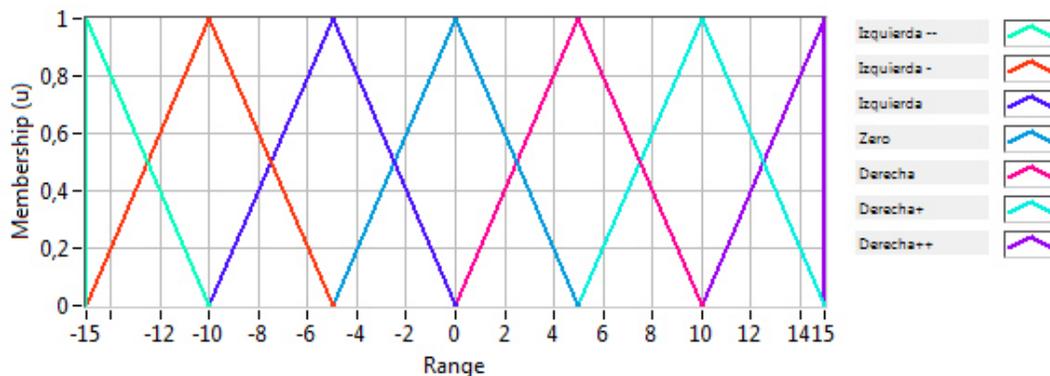
Fuente: Los autores

El siguiente paso fue definir las funciones de membresía de la variable de salida, las cuales se pueden apreciar en la figura 18 y posteriormente establecer las reglas difusas para definir el comportamiento del sistema; la tabla 1 muestra las diferentes reglas donde básicamente consiste en una selección simple de la función de membresía según el valor de la variable de entrada, al dictar las reglas de esta manera, el valor de la variable de salida será directamente proporcional al valor de la variable de entrada. Para los dos controladores se emplea el mismo sistema de reglas.

Tabla 1. Reglas difusas para los controladores

Si Entrada es	I--	I-	I	Z	D	D+	D++
Entonces Salida es	I--	I-	I	Z	D	D+	D++

Figura 18. Función de membresía de la salida del sistema



Fuente: Los autores

La figura 18 muestra las funciones de membresía de la salida junto a su rango de trabajo que representa la potencia a la que debe ser activado el motor (en una escala de 0 a 100 según el módulo de LEGO). Se escogieron rangos bajos para darle una mejor respuesta al sistema ante cambios rápidos. En el caso de las entradas se eligieron esos rangos para asegurar su funcionamiento dentro del

área de visión de Kinect®, en caso que el usuario se encuentre por fuera de este rango no será detectado por el sensor. Además de lo anterior, puede observarse que en la entrada la función de membresía central es más angosta que las demás (Función de membresía Zero); esto se debe a que la potencia mínima para activar el motor es aproximadamente ± 4.5 (lo que puede variar de acuerdo al peso que sostenga el motor) y debido a que esta se relaciona directamente con la función de membresía Zero de la salida, se decidió hacer más estrecho su rango de acción con el objetivo de alcanzar la potencia mínima al detectar el error más bajo (± 1 grado).

El método de desfuzzificación utilizado fue el método *Center of Area* o *Centroide*, el cual es uno de los más utilizados para transformar los valores difusos resultantes en el valor final de la variable de salida utilizado para el actuador del sistema

Tanto el error como la salida provista por el controlador difuso son multiplicados por una ganancia determinada a manera de sintonización con el objetivo de compensar los valores estos, y así corregir desperfectos en el sistema; ésta puede variar de acuerdo a factores como las especificaciones técnicas del procesador a utilizar.

2.6.3 Diseño del Algoritmo de comunicación

En el marco teórico se mencionó que *User Datagram Protocol* protocolo UDP es un mecanismo de envío de datagramas para comunicación entre dos aplicaciones de software. Por tanto se diseñó un algoritmo, que a través de este protocolo, se enviarán y recibirán los datos obtenidos con ayuda del SDK en Visual Studio hacia LabVIEW. El algoritmo de seguimiento de rostro tendrá su etapa de emisión de ángulos hacia el algoritmo de control en LabVIEW, donde la etapa de recepción tomará los datos para ser manipulados en el software, lo que permite la transferencia de datos entre ambas aplicaciones. Los códigos en Visual Studio y LabVIEW deben compartir el mismo puerto (puerto 6557 para el ángulo de traslación y puerto 6558 para ángulo de rotación), para establecer con éxito la

comunicación. El código completo y sus cabeceras están disponibles en el Anexo B2 y B3 respectivamente.

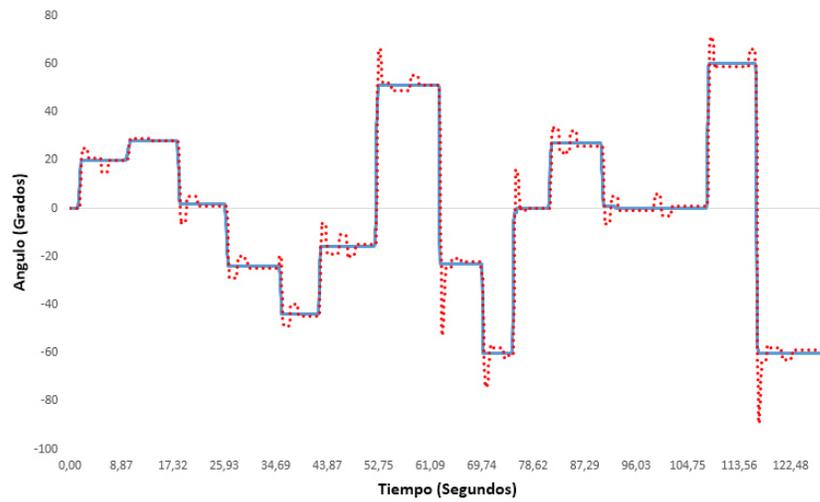
3. ANALISIS DE RESULTADOS Y PROPUESTA INGENIERIL

Se realizaron dos pruebas principales para validar el funcionamiento del sistema: tres muestras para un controlador proporcionado por el módulo de LEGO® para LabVIEW y otras tres muestras con el controlador difuso desarrollado como se explicó en la metodología de diseño, tomando como referencia el ángulo de traslación y una tomando como referencia el ángulo de rotación de la cabeza del usuario en el espacio. Para cada una de las pruebas, se tomaron las muestras en tiempo real desde el algoritmo de control de servomotor desarrollado en LabVIEW durante su ejecución.

3.1 PRUEBAS CON CONTROLADOR DEL MODULO DE LEGO®

El módulo de LEGO® Mindstorms NXT para LabVIEW ofrece un controlador PI (Proporcional-Integral) de posición para los servomotores; el anexo B2 muestra el diagrama de bloques del algoritmo de control implementado. Para comprobar que funciona correctamente independientemente de cualquier añadido, se realizó una prueba cambiando manualmente la consigna, es decir, el ángulo de rotación al cual debe ubicarse el servomotor; la figura 19 muestra el comportamiento del sistema por medio de la línea punteada, contra la referencia mostrada con una línea continua (este tipo de gráfica será igual en todas las figuras que evidencien desempeño del sistema). Se evidencia un buen rendimiento salvo sobre impulsos producto de la alta velocidad a la que gira el motor y no se detiene a tiempo antes de alcanzar el 0 en el error, es decir cuando alcanza el ángulo de referencia. También se puede apreciar que su respuesta es casi inmediata, ya que casi se superponen ambas líneas, por lo que se habla bien de la velocidad con la que responde el controlador en cambios manuales.

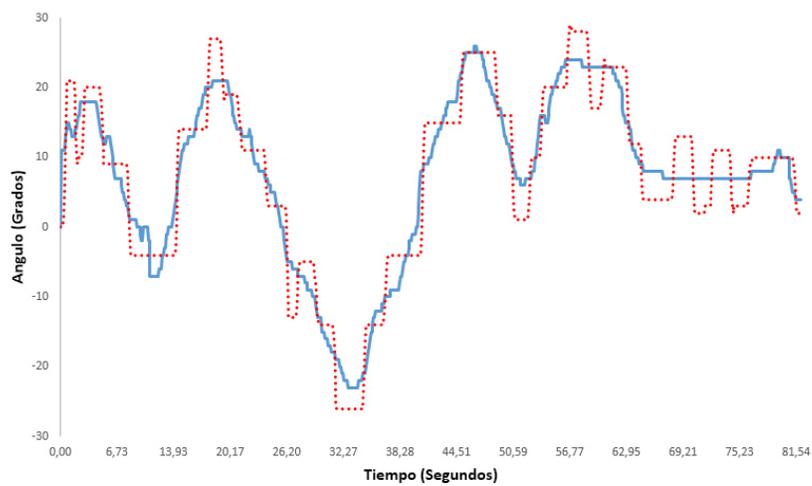
Figura 19. Respuesta del controlador PI ante cambios de consigna manuales



Fuente: Los autores

Siguiente a la primera prueba con resultados aceptables, se realizaron tres pruebas esta vez utilizando el sensor Kinect y la transferencia de datos desde Visual Studio hasta LabVIEW por medio del protocolo UDP. Las tres pruebas presentan resultados muy inferiores a los que se vieron en la figura 19.

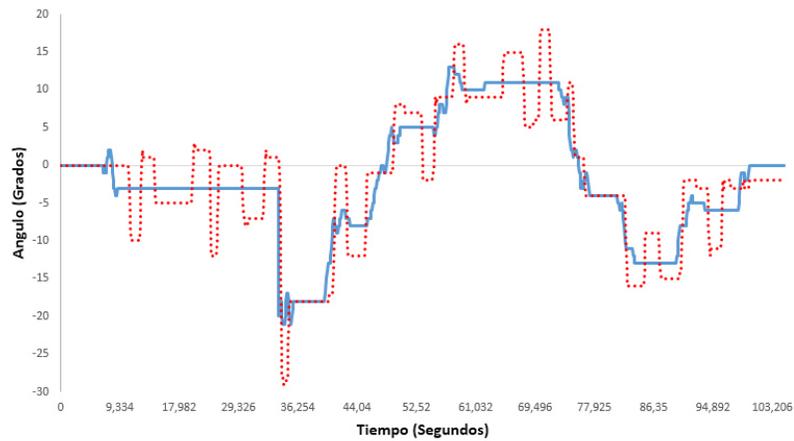
Figura 20. Primera prueba de controlador PI con Kinect



Fuente: Los autores

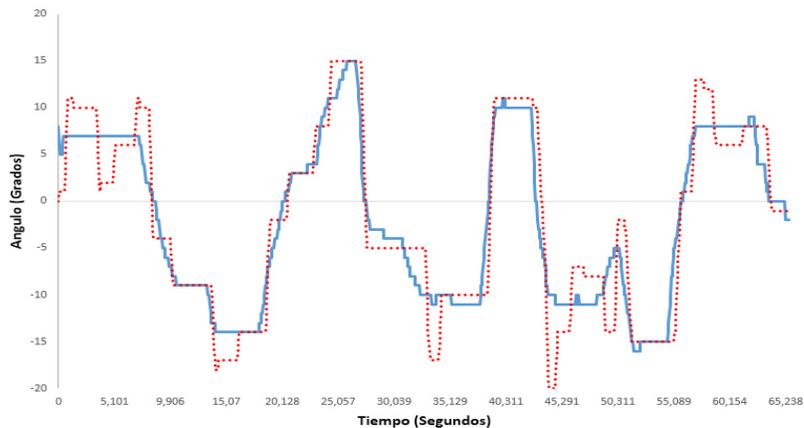
En la figura 20 el sistema trata de seguir la línea de la referencia, pero tienen muchas oscilaciones aun cuando no hay cambios en la consigna, lo que da a pensar que la potencia de salida al servomotor es mucho más alta de lo que se requiere en el momento de alcanzar el ángulo deseado. Los resultados de la segunda prueba se pueden ver en la figura 21, donde son mucho más evidentes las oscilaciones incluso cuando la referencia esta quieta.

Figura 21. Segunda prueba de controlador PI con Kinect



Fuente: Los autores

Figura 22. Tercera prueba de controlador PI con Kinect



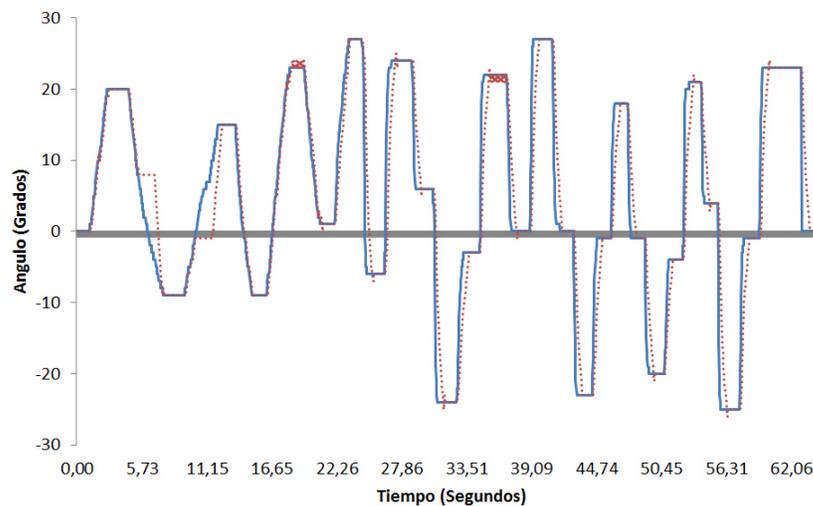
Fuente: los autores

Finalmente la figura 22 muestra resultados mejores durante la tercera prueba, mostrando un rendimiento un poco más cercano al de la prueba con cambio manual de consigna, pero sin ser aun satisfactorios para cumplir con el objetivo.

3.2 PRUEBAS CON CONTROLADOR DIFUSO

El diseño del controlador difuso explicado partió de la necesidad de resolver los problemas encontrados durante las pruebas con el controlador PI. En general fue mucho más ventajoso el diseño de este tipo de sistema de control en particular, ya que se tenía conocimiento de las variables de entrada y salida del sistema, además de poder determinar de qué manera se activara la salida en función del tipo de entrada que se reciba, algo muy difícil de lograr en un controlador PI aun diseñado desde cero.

Figura 23. Respuesta del controlador difuso ante cambios manuales



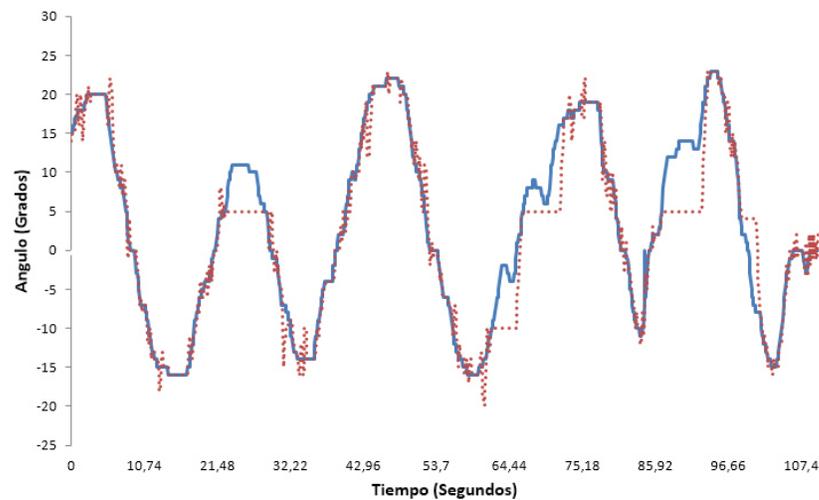
Fuente: Los autores

Al igual que con el controlador PI, se hizo una primera prueba con cambios manuales en la referencia; la figura 23 muestra los resultados de esta. El rendimiento del controlador es parecido al anterior, pero con la ausencia de sobre impulsos y unos retardos al inicio de la prueba, también se ve que cuando está a

punto de llegar a alcanzar la referencia, se ve un pequeño desvío en la línea punteada, ya que el controlador fue diseñado para bajar a la velocidad mínima cuanto más cerca este de tener cero grados en el error del sistema. A partir de aquí se procedió a realizar tres pruebas con el controlador difuso agregando el sensor Kinect.

La primera prueba corresponde a la traslación de la cabeza del usuario, que se mantuvo a una distancia de 1.5m del Kinect® (eje Z), desplazándose horizontalmente (eje X) sin dejar de mirar el sensor y con los multiplicadores del error y salida del sistema en 1 (manteniendo el mismo error y salida diseñados en el controlador difuso); al mismo tiempo el servomotor recibía la acción de control para orientarse en el mismo ángulo que suministraba el algoritmo de detección facial. La figura 24 muestra el comportamiento del servomotor comparándolo con su referencia o *Set Point*.

Figura 24. Comportamiento del sistema durante primera la prueba de traslación

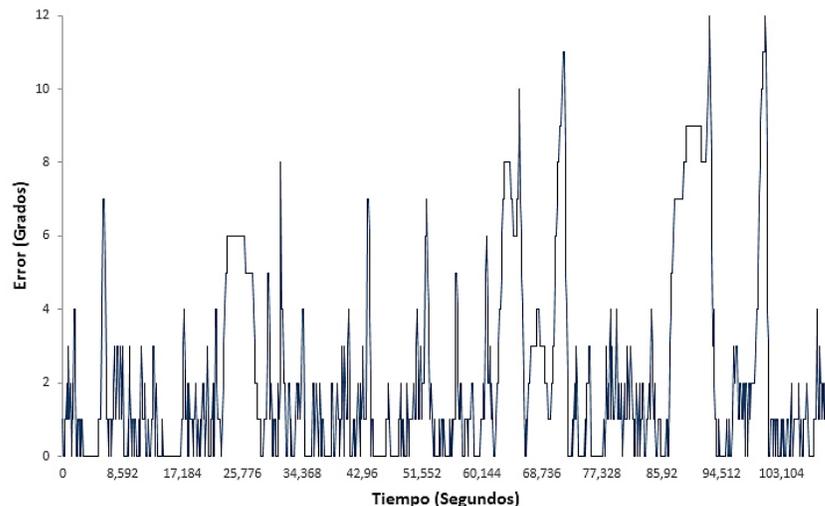


Fuente: Los autores

La falta de una afinación previa del controlador provocó continuas pero leves oscilaciones en la mayoría de los casos, aunque en general el servomotor siguió la referencia entregada durante toda la prueba.

Luego de recibir todas las muestras se obtiene un error promedio de 2,099°. La figura 25 muestra cuantos grados de error alcanzó el sistema durante todo el proceso; puede observarse que la mayoría de los picos en la gráfica se encuentran en el orden de los 2º y 3º, dando veracidad al cálculo anterior de error promedio.

Figura 25. Error del sistema durante la primera prueba de traslación

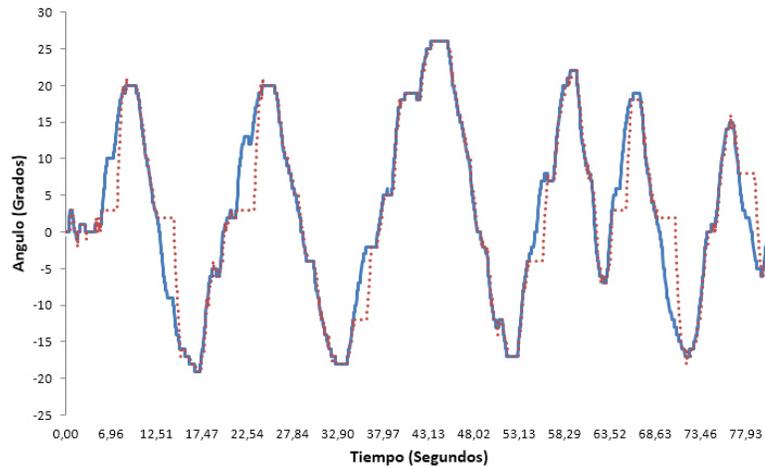


Fuente: Los autores

Los resultados de la primera a pesar de ser buenos eran mejorables, por lo que se procedió a afinar o sintonizar el controlador, en este caso lo que se hizo fue modificar los multiplicadores tanto del error como de la salida del sistema, ayudando a compensar su respuesta haciéndola más eficiente, en este caso se dejaron ambos multiplicadores con valores de 1.1. Durante la prueba la ubicación del usuario fue a 2 metros del sensor; se realizó el mismo procedimiento que

durante la primera prueba logrando mejores resultados en el comportamiento del sistema como puede observarse en la figura 26.

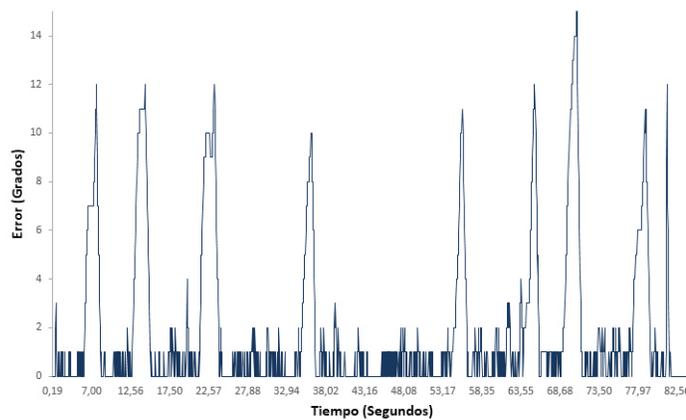
Figura 26. Comportamiento del sistema durante segunda prueba de traslación



Fuente: Los autores

Se evidenció mejor resultado en el cálculo del error promedio, ahora de 1,846° (aunque en la figura 27 muestra que la mayoría de veces durante la prueba el error era de 1 grado).

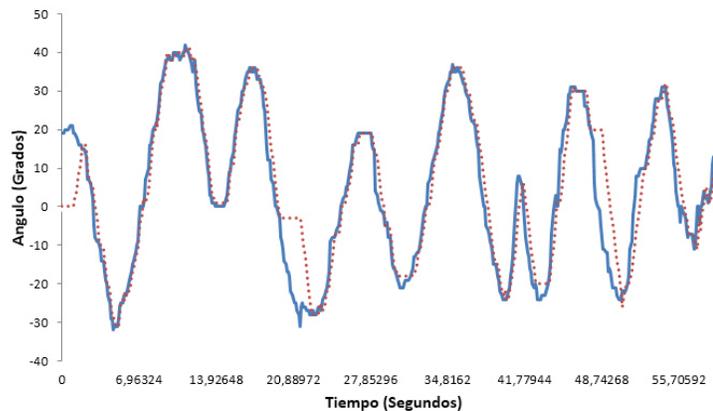
Figura 27. Error del sistema durante la segunda prueba de traslación



Fuente: Los autores

Con los resultados obtenidos durante la prueba de traslación era hora de poner a prueba el controlador para la rotación de la cabeza. Durante esta prueba el usuario se ubicó a una distancia aproximada 1 metro del sensor. Éste giraba la cabeza con la precaución de que la aplicación de detección y seguimiento facial no perdiera de vista su rostro; la adquisición de muestras y la evaluación del rendimiento del sistema se llevaron a cabo de la misma manera que en las dos pruebas anteriores. Las figuras 28 y 29 evidencian el rendimiento del controlador, esta vez tomando como referencia el ángulo de rotación de la cabeza del usuario que sigue con más eficiencia la referencia comparado con los dos controladores.

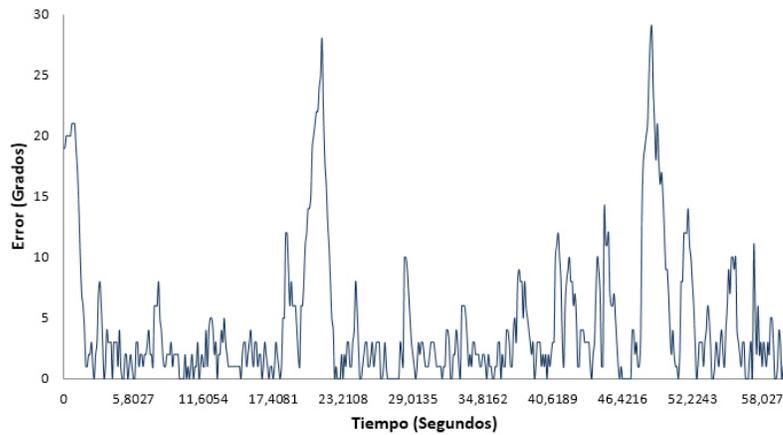
Figura 28. Comportamiento del sistema durante prueba de rotación



Fuentes: Los autores

Al igual que en las dos pruebas anteriores, se calculó el error promedio basados en las muestras adquiridas durante el proceso, obteniendo un valor de $4,506^\circ$; éste es un error más alto comparado con las pruebas anteriores, pero debe tenerse en cuenta que el rango de acción del controlador de rotación es de $\pm 50^\circ$ mientras que el de traslación es de $\pm 28^\circ$ (por esta razón el error es considerado como muy aceptable); la figura 29 muestra el progreso con respecto a la eficiencia del sistema durante toda la prueba.

Figura 29. Error del sistema durante la prueba de rotación



Fuente: Los autores

3.3 DISCUSIÓN DE RESULTADOS

Una de los detalles más significativos que se pudieron apreciar durante las pruebas es la clara diferencia en el rendimiento con y sin el sensor Kinect, principalmente durante el uso del controlador PI; El principal inconveniente de utilizar este sensor es que el computador que se va a utilizar debe ser de altas prestaciones, el procesamiento de imagen que utiliza el Kinect, el instrumento virtual funcionando en LabVIEW y el protocolo UDP que transfiere los datos de una aplicación a otra demandan mucha potencia de procesamiento al computador que se utilizó para las pruebas, como resultado se produjeron retardos en la ejecución del sistema y la actualización de los datos, y debido a que el controlador de LEGO mantiene la salida del sistema actual independientemente del equipo de cómputo, lo más probable es que se llegue a la referencia sin bajar la velocidad como se había previsto, provocando los sobre impulsos e incluso las oscilaciones apreciadas por ejemplo en las pruebas con el controlador PI.

Es necesario afinar el sistema antes de ponerlo en funcionamiento para evitar problemas como oscilaciones (aunque no ejercía gran efecto en el rendimiento

general, visualmente podría llegar a ser molesto para el usuario ver vibrar el sistema continuamente).

Las gráficas que evidencian el comportamiento del sistema durante las pruebas tienen una característica en común: presentan retardos en determinados momentos del proceso que están representados por desvíos en la línea punteada en las gráficas donde se compara la respuesta del controlador con su referencia; además pueden observarse picos mucho más altos en las gráficas donde se visualiza el error del controlador durante el proceso. Esto se debe a dos factores: el primero es la potencia mínima a la que debe activarse el servomotor; existen ocasiones en las que esta potencia no alcanza a poner en movimiento el actuador lo que da inicio al retardo. El segundo factor corresponde al controlador difuso, que está diseñado para que la potencia de salida se incremente mediante rangos de error determinados, lo significa que si la potencia correspondiente al rango de error mínimo no es capaz de mover el servomotor, el error seguirá aumentando hasta pasar al segundo rango, incrementando la potencia de salida lo suficiente para activarlo. La solución más sencilla sería incrementar el valor de la potencia de salida mínima, pero al hacerlo estaríamos provocando de una forma diferente el problema mencionado al inicio de esta discusión, el servomotor debido a la potencia que se le entregaría, llegaría con una alta velocidad a alcanzar la referencia y sin tiempo para detenerse a tiempo, provocando sobre impulsos u oscilaciones como los vistos en las pruebas con el controlador PI.

Estos retrasos en la respuesta interfieren en el error promedio del sistema; si se revisa por ejemplo la figura 27, la gráfica muestra que durante la mayoría del tiempo el error es de 1 grado, pero el error promedio calculado fue 1.846 grados, esto demuestra que a pesar del buen rendimiento del controlador se podría mejorar aún más su diseño; esto para acercar el valor promedio al que realmente se ilustra en el diagrama.

Si se comparan las figuras 24 y 26 puede observarse diferencias en el comportamiento de estos desvíos en las dos pruebas. En la segunda prueba se presentaron más oscilaciones durante lapsos más cortos, mientras que en la primera fue más prolongada la duración del retardo; puede inferirse que una buena afinación además de reducir las vibraciones también permite disminuir la duración de los retrasos en la respuesta. Esta sintonización que consiste en multiplicar la entrada y la salida del sistema por un determinado número puede manipularse para activar el servomotor con una potencia mayor que la especificada en el diseño del controlador difuso.

Dos detalles positivos que vale la pena mencionar es la ausencia de ruido en la transmisión de datos y la robustez del sistema LEGO, es importante destacar que el protocolo UDP se implementa en el mismo equipo de cómputo, lo cual garantiza casi en su totalidad la recepción de la información a pesar de no tener control de flujo, aun así, en caso de perderse la ubicación de la cabeza por parte de la aplicación de detección y seguimiento, se marca como cero la recepción de datos en LabVIEW. Con respecto al servomotor de LEGO, este tiene la suficiente robustez para asegurarse de que se medirá correcta y constantemente el ángulo de rotación por medio de su encoder interno; sumados estos dos detalles, el sistema se encuentra prácticamente libre de ruido en su señal.

4. CONCLUSIONES Y RECOMENDACIONES

Se desarrolló un sistema que permite localizar y seguir el rostro de una persona en un rango comprendido entre 1m y 2,5 m. El proceso de diseño e implementación así como las pruebas de la fase final fueron desarrollados con resultados satisfactorios.

Debido a que la aplicación de detección y seguimiento facial fue diseñada con base en el código ejemplo *Face Tracking Basics WPF*, deben tenerse en cuenta las recomendaciones hechas por Microsoft para el correcto uso del programa. Además es importante destacar que el sistema es tan efectivo dependiendo del equipo de procesamiento donde sea ejecutado, así que es recomendable que sean atendidas las sugerencias acerca de los requerimientos de sistema para un óptimo uso del sensor Kinect®.

Una vez analizados los resultados obtenidos durante las pruebas, se planea el diseño de un controlador que solucione el conflicto de los retrasos en la respuesta del sistema ante situaciones donde la potencia de salida no es suficiente para mover el servomotor de LEGO®; en lo posible se mantendrá el controlador difuso implementado en esta propuesta para el próximo diseño.

Se evaluará la posibilidad de implementar el sistema en dispositivos más accesibles en el mercado, con el objetivo de validarlo como una plataforma que pueda ser utilizada en diversas aplicaciones según el requerimiento del usuario. Para ello es necesario un estudio completo sobre las técnicas que utiliza el algoritmo de detección y seguimiento en el sensor Kinect® y del nuevo controlador para determinar las compatibilidades de uso y finalmente proceder al diseño del nuevo prototipo.

5. BIBLIOGRAFIA

ACHMAD Bañza, Firdausy Kartika. Neural Network-based Face Pose Tracking for Interactive Face Recognition System. En: International Journal on Advanced Science, Engineering and Information Technology, 2012, Vol. 2, No. 1. 105-108

BELLMORE Collin, Ptucha Raymond, Savakis Andreas. Interactive Display using depth and RGB sensors for face and gesture control. En: Western New York Image Processing Workshop. 2011. p. 5-8

CALVO Isidro, Perianez Gorka. Uso conjunto de la plataforma LEGO MINDSTORMS NXT y metodologías PBL en Informática Industrial. En: Ikastorratza, e-Revista de didáctica. 2010, No. 6.

HUANG Albert et al, Visual Odometry and Mapping for Autonomous Flight Using an RGB-D Camera. En: International Symposium on Robotics Research (ISRR). 2011

JURADO Francisco et al. Desarrollo de videojuegos: Desarrollo de componentes. 1ra edicion. Ciudad real: Bubok, 2012.

LAJARA José, Pelegrí José. LabVIEW Entorno gráfico de programación. 1ra edicion. Mexico D.F: Alfaomega, 2007.

YANG Fie et al. Robust face tracking with a consumer depth camera. En: 19th IEEE International Conference on Image Processing. Orlando, FL, 2012. p. 561-564

YANG Rouchen. Complex Behavior from a Simple Rule: Demonstration with Lego Mindstorms NXT Kit. Trabajo de grado (opta) Columbus: The Ohio State University. Department of Electrical and Computer Engineering, 2013.

CIBERGRAFIA

COLPRENSA, Un videojuego, nueva terapia para niños con síndrome de Down [En línea]. En: El universal. Bogotá, 10-Abr-2013. Disponible en internet: <http://www.eluniversal.com.co/cartagena/vida-sana/un-videojuego-nueva-terapia-para-ninos-con-sindrome-de-down-115535>

ENGADGET, Ubi Interactive usa Kinect para convertir a cualquier superficie en una pantalla táctil [En línea]. 2013 [Citado 6-Dic-2013]. Disponible en internet: <http://es.engadget.com/2013/08/15/ubi-interactive-kinect-pantalla-tactil/>

INTEL. Magic Mirror: Gesture Controlled Parametric Body Modeling [En línea]. 2013 [Citado 19-May-2013]. Disponible en internet: <http://www.intel.com/content/www/us/en/research/intel-labs-research-at-intel-magic-mirror-demo-video.html>

MICROSOFT DEVELOPMENT NETWORK. Joint Orientation [En línea]. 2013 [Citado 16-May-13]. Disponible en internet: <http://msdn.microsoft.com/en-us/library/hh973073.aspx>

MICROSOFT DEVELOPMENT NETWORK. Utilizar servicios UDP [En línea]. 2013 [Citado 18-May-13]. Disponible en internet: [http://msdn.microsoft.com/es-es/library/tst0kwb1\(v=vs.80\).aspx](http://msdn.microsoft.com/es-es/library/tst0kwb1(v=vs.80).aspx)

MICROSOFT DEVELOPMENT NETWORK. Face Tracking [En línea]. 2013 [Citado 2-Nov-13]. Disponible en internet: <http://msdn.microsoft.com/en-us/library/jj130970.aspx>

MICROSOFT DEVELOPMENT NETWORK. Face Tracking Basics-WPF C# Sample [En línea]. 2013 [Citado 16-May-13]. Disponible en internet: <http://msdn.microsoft.com/en-us/library/jj131044.aspx>

MICROSOFT DEVELOPER NETWORK, New and Changed Features: Robotics Developer Studio 4 [En línea]. 2012 [Citado 6-Dic-2013]. Disponible en internet: <http://msdn.microsoft.com/en-us/library/cc998522.aspx>

MICROSOFT DEVELOPER NETWORK, CCR Introduction [En línea]. 2012 [Citado 6-Dic-2013]. Disponible en internet: <http://msdn.microsoft.com/en-us/library/bb648752.aspx>

MICROSOFT. Kinect for Windows: Human interface Guidelines v1.7. 2013.

MICROSOFT. Kinect for Windows Features [En línea]. 2013 [Citado 20-Mar-2013]. Disponible en internet: <http://www.microsoft.com/en-us/kinectforwindows/discover/features.aspx>

MICROSOFT ROBOTICS BLOG, Announcing Kinect Services for RDS 2008 R3 [En línea]. 2011 [Citado 6-Dic-2013]. Disponible en internet: <http://blogs.msdn.com/b/msroboticsstudio/archive/2011/07/13/announcing-kinect-services-for-rds-2008-r3.aspx>

NATIONAL INSTRUMENTS. NI LabVIEW para LEGO® MINDSTORMS® [En línea]. 2013. [Citado 18-May-2013]. Disponible en internet: <http://www.ni.com/academic/mindstorms/esa/>

PSICOTUITERS.COM. Terapin2, juego de Kinect para niños con autismo [En línea]. 2013 [Citado 6-Dic-2013]. Disponible en internet: <http://psicotuiters.wordpress.com/2012/10/01/terapin2-juego-de-kinect-para-ninos-con-autismo-via-webadictos/>

REVISTA DINERS, El asombroso Kinect [En línea]. 2012 [Citado 6-Dic-2013]. Disponible en internet: http://revistadiners.com.co/articulo/62_523_el-asombroso-kinect

XBOX.COM. El efecto Kinect: Lakeside Center for Autism [En línea]. 2013 [Citado 19-May-2013]. Disponible en internet: <http://www.xbox.com/es-ES/Kinect/Kinect-Effect>

ANEXOS

Anexo A. Algoritmo del sistema

A1. Algoritmo para detectar y obtener los ángulos de traslación y rotación

```
Private void Angulo_de_traslacion_y_rotacion(FaceTrackFrame Trama, Skeleton
esqueleto)
{
RotY = Trama.Rotation.Y;
Joint cabeza = esqueleto.Joints[JointType.Head];
TransX = cabeza.Position.X*100;
TransZ = cabeza.Position.Z*100;
Hipotenusa = Math.Sqrt((Math.Pow(TransX, 2)) + (Math.Pow(TransZ, 2)));
Angulo = ((Math.Asin(TransX / Hipotenusa)) * 180) / Math.PI;
}
```

A2. Algoritmo de comunicación Visual Studio-LabVIEW

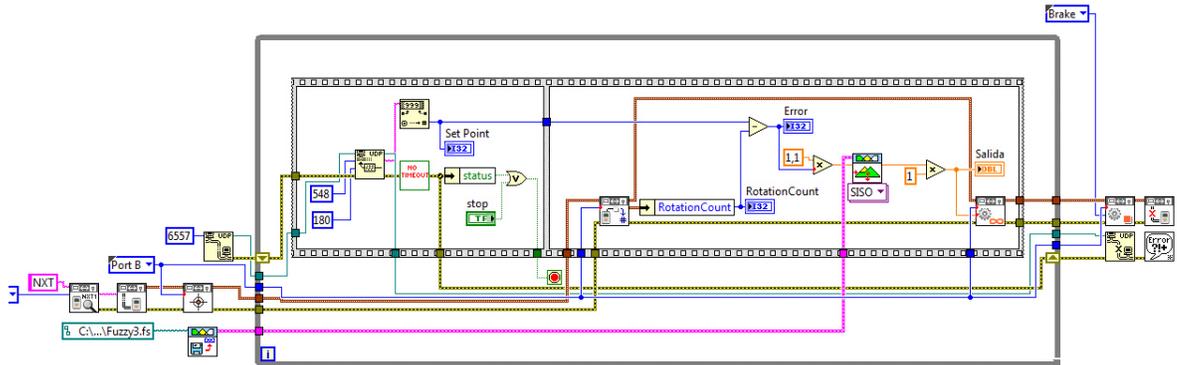
```
Private void LabVIEW_UDP(double angulo, double roty)
{
Socket s = new Socket(AddressFamily.InterNetwork, SocketType.Dgram,
ProtocolType.Udp);
IPAddress broadcast = IPAddress.Parse("127.0.0.1");
IPEndPoint ep = new IPEndPoint(broadcast, 6557);
IPEndPoint ep2 = new IPEndPoint(broadcast, 6558);
byte[] sendbuf = Encoding.ASCII.GetBytes(angulo.ToString());
byte[] sendbuf2 = Encoding.ASCII.GetBytes(roty.ToString());
s.SendTo(sendbuf, ep);
s.SendTo(sendbuf2, ep2);
}
```

A3. Cabeceras del algoritmo de detección y seguimiento facial

```
using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Media;
using Microsoft.Kinect;
using Microsoft.Kinect.Toolkit.FaceTracking;
using System.Net;
using System.Net.Sockets;
using System.Text;
```

ANEXO B. Diagramas de bloques

B1. Diagrama de bloques del controlador difuso



B2. Diagrama de bloques del controlador PI

