

TRABAJO DE GRADO

Diseño e implementación de funciones de interacción emocional y sensorial para el robot Baxter en el contexto del proyecto Human-Robot Scaffolding

Gabriel Alberto Díaz Guevara
Santiago Sarmiento Navarro

Diseño e implementación de funciones de interacción emocional y sensorial para el robot Baxter en el contexto del proyecto Human-Robot Scaffolding

Pontificia Universidad Javeriana
Facultad de Ingeniería
Pregrado en Ingeniería Electrónica

Gabriel Alberto Díaz Guevara
Santiago Sarmiento Navarro
Director: Eduardo Gerlein PhD
Coodirector: Enrique González Guerrero PhD

Bogotá D.C., Abril 9 de 2018

Índice general

1. Introducción	1
2. Antecedentes y Marco Teórico	3
2.1. Introducción	3
2.2. Antecedentes	3
2.3. Marco Teórico	7
2.3.1. Robot Antropomórfico	7
2.3.2. Relación Robot-Humano	8
2.3.3. Módulos	9
2.3.4. ROS	13
3. Funciones de expresión emocional	14
3.1. Introducción	14
3.2. Descripción detallada de las funciones Expresión Emocional	14
3.2.1. HRS Arquitectura	16
3.2.2. Bloque Función de expresión emocional	17
3.3. Validación de Resultados	25
4. Funciones de percepción	34
4.1. Introducción	34
4.2. Captura	36
4.3. Pre-Procesamiento	38
4.4. Segmentación	41
4.5. Interpretación	42
4.6. Descripción detallada funciones de percepción	44
4.6.1. Bloques Función de Percepción	45
4.6.2. Bloque Cámara Kinect	48
4.6.3. Proceso de Activación	48
4.7. Validación de resultados	49
4.8. Conclusiones	52
5. Funciones de movimiento	53
5.1. Introducción	53
5.2. Descripción Robot Baxter	53
5.3. Descripción detallada funciones de movimiento	55
5.3.1. Calibración Coordenadas de percepción con movimiento	56
5.3.2. Posicionamiento	57
5.3.3. Nodos Implementados	58
5.4. Validación de resultados	60
5.5. Conclusiones	61
6. Conclusiones	62
7. Anexos	65

Índice de figuras

1.1. Robot antropomórfico Baxter	1
2.1. Robot Baxter en función de recoger bloques [28]	4
2.2. Control de movimiento Baxter [12]	4
2.3. Control de orientación Robot Baxter el cono que observa el usuario [25]	5
2.4. Interacción entre Baxter y el mago [17]	5
2.5. Robot Kismet [2]	6
2.6. Robot asistencial [30]	6
2.7. Robot Paro [18]	7
2.8. Cadena articulada brazos del robot Baxter [20]	7
2.9. Robot Baxter [22]	8
2.10. Cinemática Directa y Cinemática Inversa [4]	9
2.11. Parámetros de ubicación relativa de dos sistemas de coordenadas [8]	10
2.12. Expresiones emocionales	12
2.13. Tópicos en ROS [14]	13
2.14. Servicios en ROS [13]	13
3.1. Diagrama de bloques de las funciones de expresión emocional	15
3.2. Representación de las funciones de expresión emocional según sus parámetros de entrada(Control y Aceptación)	15
3.3. Frame de imágenes para caracterizar como habla el robot	16
3.4. Frame de imágenes para caracterizar como guiña el robot	16
3.5. Diagrama específico de los nodos y tópicos en ROS	17
3.6. Transición de imágenes	17
3.7. Función escalonada	18
3.8. Anillo de leds que se activan con el nodo Emociones Leds Baxter [23].	22
3.9. Ubicación Ventilador Baxter [23].	23
3.10. Anillo led que se controlan por el nodo Emociones Frecuencia Baxter [23].	24
3.11. Interacción Baxter con la primera estudiante.	25
3.12. Interacción Baxter con la segunda estudiante.	25
3.13. Interacción Baxter con la tercera estudiante.	26
3.14. Interacción Baxter con la cuarta estudiante.	26
3.15. Interacción Baxter con la quinta y sexta estudiante.	26
3.16. Resultados Primera Pregunta.	27
3.17. Resultados Segunda Pregunta.	28
3.18. Resultados Tercera Pregunta.	29
3.19. Resultados Cuarta Pregunta.	30
3.20. Resultados Quinta Pregunta.	30
3.21. Resultados Sexta Pregunta.	31
3.22. Resultados Séptima Pregunta.	32
3.23. Resultados Octava Pregunta.	33
4.1. Diagrama de Bloques Visión Artificial para el reconocimiento de colores	35
4.2. Captura de la Cámara con visión enfocada en los cubos y espacios vacíos ubicados en el tablero	36
4.3. Captura de la Cámara con los puntos de referencia identificando los cubos y espacios vacíos dentro del tablero	36

4.4. Captura de cámara en modelo HSV	38
4.5. Proceso para detectar colores	40
4.6. Proceso para detectar colores	41
4.7. Captura de la imagen con los puntos de cada punto del tablero con su respectivo color	42
4.8. Tablero con las posiciones de la matriz	43
4.9. Numero al que pertenece cada color	43
4.10. Diagrama de bloques de las funciones de percepción	44
4.11. Diagrama detallado de los nodos y tópicos en ROS	45
4.12. Recorrido del cubo por el tablero	49
4.13. Cuadrícula que representa el tablero y compara los resultados teóricos con los resultados experimentales del Cubo Verde	50
4.14. Cuadrícula que representa el tablero y compara los resultados teóricos con los resultados experimentales del Cubo Rojo	51
4.15. Cuadrícula que representa el tablero y compara los resultados teóricos con los resultados experimentales del Cubo Amarillo	51
4.16. Cuadrícula que representa el tablero y compara los resultados teóricos con los resultados experimentales del Cubo Azul	52
5.1. Ubicación de los 7 motores representados en ángulos de uno de los brazos del robot Baxter [20]	53
5.2. Diagrama de bloques detallado de las funciones de movimiento	55
5.3. Diagrama de bloques detallado de las funciones de movimiento	56
5.4. Coordenadas del Robot Baxter	56
5.5. Diagrama de bloques general de las funciones de movimiento	57
5.6. Movimiento del cubo	60

Capítulo 1

Introducción

El grupo de investigación SIRP de la Pontificia Universidad Javeriana abrió recientemente una línea de investigación relacionada con el uso de robots en contextos educativos. Dentro de la línea de robótica educativa se encuentra el proyecto de tesis doctoral llamado *Human-Robot Scaffolding* orientado a utilizar un robot Baxter ¹ para promover el aprendizaje de estrategias de solución de problemas en niños de 10-13 años. La Figura 1.1 muestra el robot Baxter disponible en la Pontificia Universidad Javeriana. Para validar la arquitectura propuesta del proyecto *Human-Robot Scaffolding* utilizando a Baxter se requiere la construcción de librerías que integren ROS² (*Robot Operating System*) con la plataforma RoboAct ³, que permitan realizar acciones de movimiento, expresión emocional y percepción del ambiente de la actividad durante la interacción con el usuario. El trabajo de grado propuesto desarrolló los módulos de movimiento en contexto emocional enmarcados en el proyecto *Human-Robot Scaffolding* que a su vez se encargará de la toma de decisiones durante la interacción de los niños y el robot.



Figura 1.1: Robot antropomórfico Baxter

El trabajo propuesto responde a las necesidades de interacción de Baxter en contextos educativos. Esta interacción requiere funciones que coordinen acciones de movimiento, expresión emocional y percepción del ambiente de la actividad. El desarrollo de las librerías mencionadas contribuye en la construcción de procesos de interacción humano-robot, a través de movimientos más naturales y expresiones verbales y no-verbales del robot durante su interacción con el usuario. Si bien, actualmente hay propuestas para el uso de robots en contextos educativos, RoboAct tiene una visión innovadora porque asigna la categoría de agente social a un

¹Baxter es un robot antropomórfico con las siguientes características: a) 7 grados de libertad GDL en cada brazo. b) 1 GDL en la cabeza. c) Sensores de posición y fuerza. d) 3 cámaras. e) Sensores de percepción, aceleración y distancia. f) Compensador de gravedad cero [21].

²ROS es un conjunto de librerías y herramienta que permiten el control de un robot [24].

³RoboAct es una plataforma desarrollada por la PUJ para utilizar los robots como actores de teatro en distintos contextos como los educativos. Está compuesta por cuatro módulos: sensores, creencias, deseos, intenciones. Su desarrollo, inspirado en el concepto de Sistemas Multi-Agente, integra las plataformas BESA, RationalBESA y BDIBESA.

robot para promover el aprendizaje.

En el proyecto de grado se plantearon los siguientes objetivos:

1. Establecer los requerimientos físicos, tecnológicos y pedagógicos que permitan diseñar las funciones para el robot Baxter y apoyen el proceso de enseñanza-aprendizaje en el contexto del proyecto de tesis doctoral *Human-Robot Scaffolding*.
2. Diseñar la arquitectura que integre las plataformas de desarrollo ROS y RoboAct para realizar funciones de movimiento, expresión emocional y percepción del ambiente para el robot Baxter.
3. Implementar en ROS las funciones definidas para movimiento de actuadores, expresión emocional y percepción del ambiente en el robot Baxter.
4. Evaluar el desempeño de la librería en términos de coordinación, integración entre ROS y RoboAct y eficiencia durante la interacción del robot Baxter con un usuario.

Las funciones de expresión emocional, percepción y manipulación presentadas en el trabajo de grado, han sido diseñadas de acuerdo a los requerimientos de la arquitectura Human-Robot Scaffolding. Esta arquitectura fue diseñada en JAVA de acuerdo a la metodología de programación orientada a agentes. El funcionamiento de los agentes está fundamentado en el paradigma BDI *Belief, Desire, Intention*. Así, las funciones de percepción, contribuyen en la configuración de creencias del agente relacionado con el estado de la tarea, específicamente con la posición de los bloques para cada uno de los tres juegos mencionado (*Apilar Bloques, Juego de la Escalera, Cuatro Caballos*). Para utilizar las funciones desarrolladas en la tesis de ingeniería en electrónica, fue necesario articular desde ROS y JAVA, las condiciones de comunicación mediante un WebSocket. De la misma manera, las funciones de expresión emocional que integran la pantalla del robot Baxter, los ventiladores, un parlante externo, las luces del robot y la velocidad de movimiento de los brazos, han sido diseñadas para que la estrategia de activación reconozca los valores de *Control y Aceptación* que son enviados mediante el WebSocket desde la arquitectura en JAVA. La semántica de los ejes de control y aceptación han sido redefinidos en términos del comportamiento de tutor, par o aprendiz como se explicará en el capítulo respectivo. Finalmente, las funciones de movimiento diseñadas presentan una condición de activación que son recibidas desde JAVA.

En general, las funciones presentadas en el trabajo de grado son activadas de acuerdo a los requerimientos temporales de las intenciones del agente que controla el comportamiento de robot. Si bien, las intenciones del agente cuentan con 35 planes de acción para que el robot promueva el proceso de aprendizaje en los niños, las funciones desarrolladas en el trabajo de grado permiten básicamente articular tres aspectos: expresión emocional, movimiento del robot y percepción. Los 35 planes utilizan las funciones desarrolladas en el contexto del trabajo de grado, para que el robot ayude al aprendiz a solucionar el problema planteado a través de acciones de manipulación de los cubos; ya sea para avanzar en el problema o para retroceder.

Las funciones que se presentan en el trabajo de grado son las funciones de expresión emocional, percepción y movimiento. El capítulo 3 comenzamos por describir las funciones de expresión emocional, las cuales han sido diseñadas para establecer la relación humano-robot; adicionalmente se aplicó una encuesta para validar los resultados, la cual fue implementada en la plataforma Google para llegar a diferentes sectores de la población y lograr un muestreo significativo en los resultados. El capítulo 4 abarca las funciones de percepción, las que permiten establecer la visión artificial del robot Baxter; el capítulo 5 desarrolla las funciones de movimiento, como su nombre lo indica son las funciones que permiten el movimiento del robot Baxter. En cada capítulo encontraremos procesos, validación de resultados y conclusiones que nos permitirán ver el logro de los objetivos planteados.

El documento finaliza con las conclusiones que resaltan las contribuciones del trabajo de grado a la tesis de doctorado y a los procesos de interacción humano-robot en el proceso de enseñanza-aprendizaje. El desarrollo de este trabajo contribuyó de manera significativa en nuestros procesos de formación académica y dio la oportunidad de interactuar con un robot de última tecnología y conocer las alternativas de control más utilizadas por la comunidad de investigadores en robótica a nivel internacional.

Capítulo 2

Antecedentes y Marco Teórico

2.1. Introducción

En este capítulo se presentan los fundamentos teóricos y epistemológicos que han configurado los procesos de investigación en la interacción humano robot, haciendo énfasis en el contexto educativo. Según *Institute of Electrical and Electronics (IEEE)*, el uso de robots con propósitos educativos, es una de las áreas de mayor crecimiento en investigación en la última década [29]. Los fundamentos mencionados están relacionados con cuatro aspectos. Primero, las funciones de expresión emocional están inspiradas en estudios psicológicos, sociales, fenomenológicos y educativos. Estas expresiones están relacionadas con gestos faciales para evocar en los usuarios sensaciones como alegría, tristeza, sorpresa, etc. Segundo, las funciones de expresión corporal están fundamentadas en los avances semióticos relacionados con las variables de expresión no verbal en procesos de interacción social que complementan las acciones verbales y regulan los actos comunicativos entre robots y humanos. Tercero, la expresión emocional de robots en contextos educativos está relacionada con los requerimientos de expresión verbal y no verbal que orientan los procesos de aprendizaje de los estudiantes durante la solución de problemas. Las acciones de retroalimentación de los robots como: movimiento de brazos, expresión emocional a través de pantallas, cambio de luces o intensidad de las mismas, presentación de sonidos o habla; son las características más implementadas, dados los recursos electrónicos y mecánicos con los que habitualmente son construidos los robots como Nexi, KOBIAN, Meet Milo, KASPAR, Kismet, NAO, iCub, Pepper, Roboy, Nabaztag, Robovie, PARO, Leonardo, Pleo, iCat, Mamoru y BAXTER. Cuarto, la percepción de robots en ambientes educativos está relacionada con aspectos como: la intencionalidad de intervención del robot, las condiciones de la tarea, las condiciones del ambiente de aprendizaje, las capacidades de percepción y la acción de los robots. Para visión artificial se utilizan sensores como OKAO Sensor, Kinect2. Para el reconocimiento de imagen y habla se utilizan plataformas como: Kaldi speech recognition toolkit, OpenCV, NITE, CMU PocketSphinx, Object Tracking System. Otros desarrollos utilizan dispositivos como: Tablets, Touchscreen, Radio frequency ID, Flexible Tactile Sensor.

2.2. Antecedentes

Baxter es un robot antropomórfico a partir del cual desarrollamos las funciones de percepción, funciones expresión emocional y funciones de movimiento.

Para iniciar los trabajos implementados en Baxter, fue necesario definir el control del movimiento de los brazos y pinzas mediante ROS (*Véase figura 2.1*); cuya función era recoger bloques de madera con un orden preestablecido y ordenarlos de manera vertical para crear una torre. Las librerías fueron implementadas en AR-tag para activar las cámaras que están incorporadas en el robot y definir la ubicación de los bloques. [28]



Figura 2.1: Robot Baxter en función de recoger bloques [28]

Por otro lado, en [12] se realizó una exploración de los movimientos posibles para Baxter, y las librerías que se desarrollaron en ROS fueron orientadas a movimiento de brazos con respecto a la posición en la cual estaba ubicado el objeto (Véase figura 2.2).

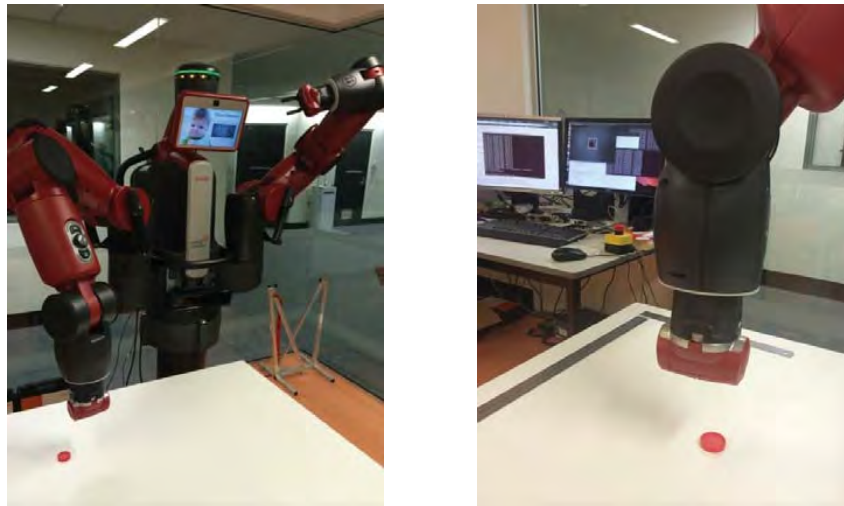


Figura 2.2: Control de movimiento Baxter [12]

Otro trabajo realizado con Baxter [25] fue recoger una de las copas de vidrio ubicadas en una mesa (Véase figura 2.3), con base en la interpretación de la mirada del usuario. Las librerías que se implementaron en ROS fueron orientadas a desarrollar las trayectorias del giro del brazo y agarre suave de la copa.

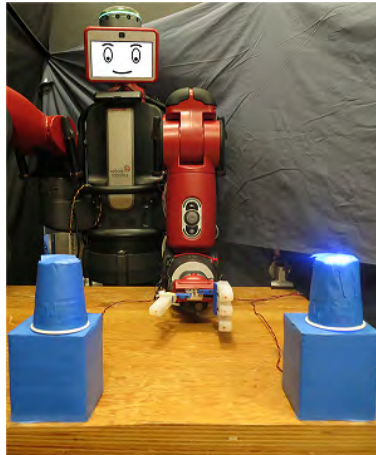


Figura 2.3: Control de orientación Robot Baxter el cono que observa el usuario [25]

No obstante Baxter también puede ser utilizado para otras aplicaciones, por ejemplo en [[17]], el robot fue programado para interpretar el personaje de un mago en una obra de teatro y presentó interacción con un usuario en una secuencia de movimientos y expresiones coreografiadas. Se desarrollaron en ROS los movimientos del robot en secuencia para el acto de magia (Véase figura 2.4).



Figura 2.4: Interacción entre Baxter y el mago [17]

Para la interacción humano-robot no solo existe Baxter. Michaud et al. Describen en [[19]] un robot diseñado para comparar la interacción humano-humano con la interacción humano-robot. Las pruebas consistían en un juego de interacción de selección e identificación de platos de colores. Se desarrollaron en ROS las funciones de identificación del entorno y movimiento del robot para señalar los diferentes platos.

Una función importante para el movimiento es discutida en [[12]], donde se expone el funcionamiento de movimiento de los brazos de Baxter, así como la identificación de un objeto. Estas funciones harán parte de una fase inicial de apropiación del uso y programación del robot.

Uno de los robots pioneros en la interacción humano robot es el robot Kismet desarrollado por Cynthia Bretzel del MIT. En [[2]] Se presenta a Kismet con capacidades de aprendizaje a partir percepciones visuales, y es utilizado para interacción con niños. En el proyecto, usuarios adultos señalaban acciones que no debían ser realizadas por los niños, el robot posteriormente mediante el dialogo y expresiones faciales se comunica con los niños para orientarlos. ROS fue utilizado desarrolló de acciones en contexto emocional (Véase figura 2.5). A partir de la investigación desarrollada con Kismet, se consolidad el campo de trabajo en robot sociales en la década de los noventa.

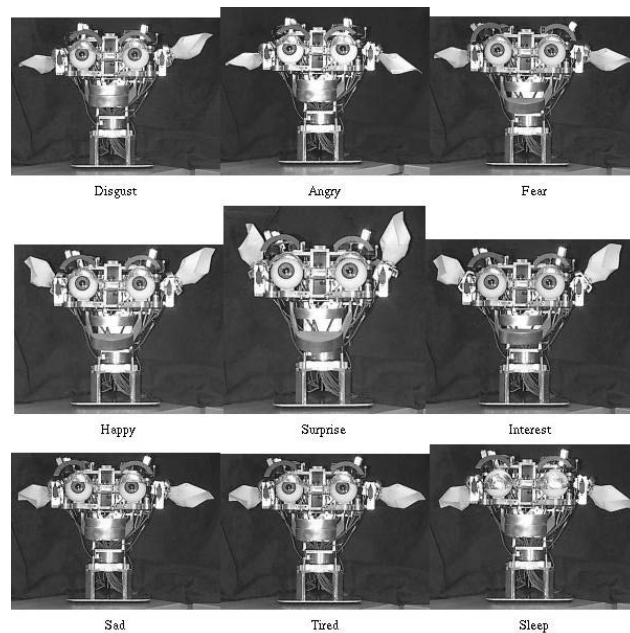


Figura 2.5: Robot Kismet [2]

Por otro lado, en [30] se discute el diseño de un robot antropomórfico y se presentan métodos y técnicas para que el robot interactúe con múltiples usuarios al mismo tiempo. El robot es utilizado en un aula de clases para generar empatía, familiaridad y aceptación de los usuarios en tal contexto (Véase figura 2.6).



Figura 2.6: Robot asistencial [30]

El robot PARO, como vemos en [18] es un robot que se empezó a desarrollar en 1993 por la compañía japonés AIST, y fue usado hasta el año 2001 que fue presentado al público; este es uno de los robots que mejor muestra la importancia de la interacción humano-robot; fue usado para simular la interacción con un animal, una foca bebe como podemos ver en la figura 2.7; posteriormente se usó en personas con problemas mentales, niños hospitalizados, personas con alguna discapacidad física, personas de la tercera edad, etc. Además, se ha demostrado que PARO ayuda a reducir el estrés en los pacientes, estimular la relación del paciente con su cuidador y demás pacientes, también ha demostrado tener un efecto psicológico positivo motivando y

relajando a sus usuarios.



Figura 2.7: Robot Paro [18]

La arquitectura de Baxter presenta grandes potenciales para la interacción con humanos debido a su particular característica de gravedad cero ¹ en los actuadores y la capacidad de presentar emociones faciales mediante una pantalla. El movimiento de los diferentes actuadores en un contexto emocional [28] implica la parametrización de variables tales como la velocidad y la estabilidad de la articulación dependiendo de la emoción o estado de ánimo requerido. Por ejemplo, estados de ánimo positivos tales como *feliz* implican mayores velocidades de movimiento, por el contrario, estados de ánimo negativos como “nervioso“ implicarán una velocidad más lenta y una vibración en el movimiento. Adicionalmente, el de grado propuesto pretende la interacción del robot con usuarios, lo que a su vez implica que Baxter tendrá que reconocer el entorno y ayudarlo al usuario en su tarea de aprendizaje de resolución de problemas. Se pretende que el desarrolló de librerías de movimiento en contexto emocional genere familiaridad de los usuarios con el robot, de tal forma que se disminuya la prevención para desarrollar actividades de interacción con el robot.

2.3. Marco Teórico

2.3.1. Robot Antropomórfico

El termino Antropomórfico es aplicado a cualquier objeto que tenga el aspecto físico similar al del ser humano; los robots antropomórficos son máquinas que se asemejan al ser humano, principalmente en la parte del hombro, brazo y muñeca como es el caso de Baxter (el robot de nuestro proyecto), y sirven para ponerse en una posición en el espacio y orientarse para realizar un determinado trabajo [5]. La figura 2.8 representa la cadena articulada del brazo del robot Baxter.

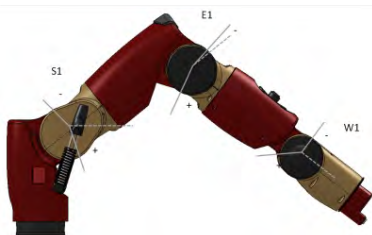


Figura 2.8: Cadena articulada brazos del robot Baxter [20]

¹El robot mantiene sus brazos en posición pero sin ofrecer resistencia cuando el operador los mueve.

Robot Baxter

Baxter es un robot industrial (*Ver figura 2.9*), que posee dos brazos y tiene una Tableta que actúa como la cara del robot. Fue fabricado por *Rethink Robotics*, y se introdujo al mercado en el año 2013, con el fin de ayudar en el trabajo a empresas industriales de tamaño mediano y pequeño, realizando labores de precisión en la manipulación, carga, descarga y clasificación de materiales.

Tiene una estatura de 1,60 m con pedestal y de 1 m sin pedestal y su peso es de aproximadamente de 80 kg, posee una pantalla que le permite mostrar emociones, 7 grados de libertad GDL en cada brazo, sensores de posición y fuerza, tiene 3 cámaras (Una en la cabeza y dos en los brazos), sensores de ultrasonido, dos anillos de leds, aceleración y distancia, compensador de gravedad cero, permitiendo que el robot se acople a su entorno, es decir, nunca tendrán una colisión entre sus brazos o con las personas, evitando futuros accidentes.

En principio el robot Baxter está orientado para el campo industrial. Sin embargo, Baxter también puede estar en otros campos como el campo educativo. Este es el campo de implementación para este proyecto.



Figura 2.9: Robot Baxter [22]

2.3.2. Relación Robot-Humano

La relación humano-robot no es un tema nuevo ya que ha venido evolucionado con el tiempo, por medio de diferentes tipos de robots entre ellos los robots sociables, por lo general tiene una apariencia humanoide generando una combinación entre tecnología y estructura humana. Son capaces de interactuar entre sí y con personas humanas involucrando emociones, sentimientos, rodeados por un ambiente con el cual puedan percibir y reaccionar a él.

Al hablar de la interacción de humanos con robots los primeros proyectos se enfocan más al movimiento de estos como tal; pero al pasar el tiempo se ha venido incorporando comportamientos similares al ser humano, podemos ajustar al del ser humano, podemos ver la interacción entre estos dos cuando encontramos robots para el cuidado de ancianos, niños, discapacitados ya que se evidencio la necesidad de crear este tipo de robots para poder ayudar a esta población. Al observar esta interacción usamos el robot como instrumento académico en nuestro proyecto; en el ámbito educativo el factor tecnológico es una herramienta útil al momento de adquirir conocimientos, debido a que causa gran impacto emocional a las personas y las incentiva a desarrollar este aprendizaje [6].

A medida que va pasando el tiempo se busca que la interacción humano-robot este más integrada al aspecto de la vida humana como en actividades cotidianas, al igual que comportamientos percibidos por estos y que se puedan reaccionar a este tipo de comportamientos.

2.3.3. Módulos

Movimiento

La descripción cinemática de los robots está en función de sus características físicas y especialmente de los actuadores. Para el caso de los robots manipuladores, el posicionamiento del robot reconoce la relación de ángulos y distancia de los eslabones. El posicionamiento del actuador utiliza dos modelos: el modelo de cinemática inversa y el modelo de cinemática directa.

El modelo de cinemática inversa tiene como entrada las coordenadas de posiciones al gripper en el espacio de trabajo. El modelo de cinemática directa requiere los valores de los ángulos y la distancia de los eslabones.

Ahora, en el contexto del trabajo de grado, el robot Baxter es considerado un robot antropomórfico con dos brazos. El posicionamiento de cada brazo puede hacerse con cualquiera de los dos modelos mencionados anteriormente. Sin embargo, dada las condiciones de percepción del ambiente con la cámara de profundidad que reconoce los objetos en el espacio de trabajo, la función de movimiento de ambos brazos se realiza mediante el modelo de cinemática inversa. La cinemática se divide en cinemática directa y cinemática inversa [29], la figura 2.10 representa el modelo de estas dos cinemáticas.

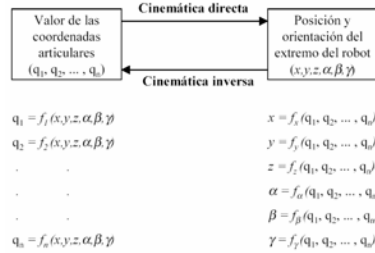


Figura 2.10: Cinemática Directa y Cinemática Inversa [4]

Las ecuaciones básicas del modelo de cinemática directa son: La ubicación de una articulación de un manipulador la cual debe incluir necesariamente un término de posición y otro de orientación.

Las matrices de rotación solo permiten representar cambios de orientación entre dos sistemas de coordenadas, dado que un manipulador necesita representar tanto cambios rotacionales como traslacionales, se definen las matrices de transformación homogénea observada en la ecuación 2.1.

$$T = \begin{bmatrix} Rotación_{3x3} & Posición_{3x1} \\ perspectiva_{1x3} & escalado_{1x1} \end{bmatrix} \quad (2.1)$$

La matriz de Transformación Homogénea para Articulaciones rotacionales es la siguiente:

$$\begin{bmatrix} \cos\theta_i & -\cos\alpha_i * \text{sen}\theta_i & \text{sen}\alpha_i * \text{sen}\theta_i & a_i * \cos\theta_i \\ \text{sen}\theta_i & \cos\alpha_i * \cos\theta_i & -\text{sen}\alpha_i * \cos\theta_i & a_i * \text{sen}\theta_i \\ 0 & \text{sen}\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

Denavit y Hartenberg proponen un método para describir y representar la geometría espacial de los elementos de un brazo con respecto a un sistema de coordenadas fijo a través de matrices de transformación de 4x4, y reducen el problema directo a encontrar la relación entre el desplazamiento espacial del sistema de coordenadas del efector final y el sistema de coordenadas de referencia [8].

Finalmente, el problema de la cinemática directa queda resuelto al multiplicar las matrices de transformación homogénea que relacionan el sistema de coordenadas i -ésimo con el sistema de coordenadas $(i-1)$ -ésimo.

$$T_n^0 = A_1^0 * A_2^1 * \dots * A_n^{n-1} \quad (2.3)$$

Los parámetros de ubicación relativa de dos sistemas de coordenadas son:

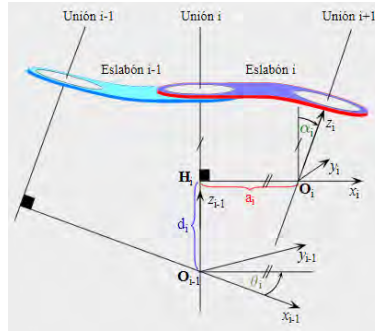


Figura 2.11: Parámetros de ubicación relativa de dos sistemas de coordenadas [8]

θ_i Es el ángulo comprendido entre los ejes x_{i-1} y x_i , medido en torno al eje z_{i-1} , en el sentido de la regla de la mano derecha.

d_i Es la distancia comprendida entre el origen O_{i-1} y el punto H_i .

a_i Es la longitud de la normal común $H_i O_i$.

α_i Es el ángulo comprendido entre los ejes z_{i-1} y z_i , medido en torno al eje x_i , en sentido de la regla de la mano derecha.

La siguiente grafica representa los parámetros antes mencionados.

Las ecuaciones básicas del modelo cinemática inversa son:

La cinemática inversa consiste en encontrar los valores que deben adoptar las coordenadas articuladas del robot $q=[q_1, q_2, \dots, q_n]$ para que su brazo se posicione y se oriente según la localización espacial determinada.

A la hora de resolver el problema de cinemática inversa es adecuado encontrar una relación matemática de la siguiente forma:

$$q_k = f_k(x, y, z, \alpha, \beta, \gamma) \quad (2.4)$$

$$k = 1 \dots n$$

En la mayoría de los casos se usa métodos geométricos, los cuales consisten en utilizar relaciones trigonométricas y resolución de triángulos formados por las articulaciones del robot. Un método para resolver el problema es recurrir a las ecuaciones de cinemática directa. Esto significa que a partir de la relación entre la matriz de transformación y las ecuaciones en funciones de las articulaciones es posible despejar las n variables

de las articulares q_i en función de las componentes de los vectores n , o , a y p [8]:

$$T_n^0 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

Emociones

Kemper define la emoción como “una experiencia corporal viva, veraz, situada y transitoria que impregna el flujo de conciencia de una persona; es percibida en el interior, va recorriendo el cuerpo, y durante el transcurso de su vivencia, sume a la persona y a sus acompañantes en una realidad nueva y transformada, la realidad de un mundo constituido por la experiencia emocional” [9].

Las emociones tienen diferentes funciones:

1. **Función adaptativa:** Es cuando se ejecuta una conducta de acuerdo a las condiciones ambientales dirigiendo la conducta hacia un objetivo determinado por ejemplo ira: conducta que se adquiere de destrucción; sorpresa: conducta que se adquiere por ejemplo exploración; miedo: conducta que se adquiere es protección; estas son algunas emociones que podemos ver en Baxter en el momento de realizar la interacción y la toma de decisiones del estudiante con el robot.
2. **Función social:** Permite la expresión de la emoción y poder presidir el comportamiento asociado a la misma; este lo podemos ver en los movimientos que realiza el robot cuando se está mostrando la emoción que quiere transmitir al estudiante
3. **Función Motivacional:** Es la experiencia que se mantiene presente en cualquier tipo de actividad, esta se observa cuando se adquieren los conocimientos que quería transmitir el robot al estudiante.

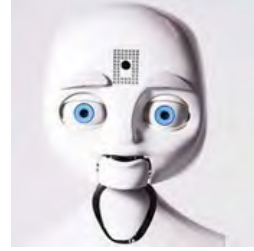
Los robots pueden ser programados para que entiendan y transmitan las emociones que se les quieran transferir al estudiante; gracias al avance que se ha tenido en el campo de la robótica se puede encontrar software que ayuda a entender y reaccionar ante las emociones humanas como la figura 2.12.



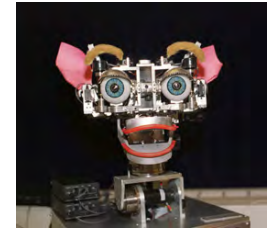
(a) Triste(kobian) [11]



(b) Bravo(cozmo) [1]



(c) Feliz(nexi) [16]



(d) Sorprendido(kismet) [10]



(e) Neutral(Milo) [15]



(f) Dudoso(Baxter)

Figura 2.12: Expresiones emocionales

Sonidos

Cuando se habla de la aplicación del sonido en la enseñanza nos dice Moreno que “la utilización del medio sonoro facilita diversas formas de expresión, entendiendo esta como la manifestación de procesos de reflexión que implican la capacidad de concentración y de adquisición de conocimientos, motivados a su vez por la percepción multisensorial y la experiencia de cada individuo” [7].

El sonido puede generar diferentes formas de comunicación con el estudiante que mejoran su capacidad de desarrollo y conocimiento, al usar el sonido en el ámbito educativo podemos observar algunos beneficios que trae como ampliar el conocimiento teórico y práctico con desarrollo de pensamientos más profundos, impulsar la imaginación en el momento en el que se genera los diferentes tipos de sonido, desarrollar sensibilidad auditiva en el momento de diferenciar los diferentes sonidos, mejorar la actitud de los estudiantes captando la atención y su interés por medio de este y tener diferentes alternativas de comunicación con el estudiante.

Percepción

Los robots pueden percibir objetos mediante dos componentes:

1. **Ambiente Controlado:** En el ambiente controlado, todos los objetos están siempre ubicados en la misma posición, por tanto si sabemos el eje de coordenadas del robot y la distancia de todos los objetos el robot podrá moverse libremente en esa ubicación y podrá recoger cualquier objeto en ese ambiente. Un ejemplo es el robot Dave, este es un robot que tiene un ambiente controlado para poder servir helados [26]; para servir los helados es necesario que Dave sepa donde están todos los ingredientes, de modo que él los pueda manipular y entregar el helado al usuario.

2. **Sensores** : Sensores de ultrasonido, infrarrojo y cámaras son herramientas que permiten al robot tener una visión sobre su espacio de trabajo; la combinación de cámara con un sensor infrarrojo como es el Kinect permiten al robot tener la visión y ubicación espacial de estos objetos, lo único que se debería desarrollar es igualar las coordenadas de la cámara y el sensor infrarrojo con el centro de masa del robot. Este mecanismo de percibir los objetos es el más parecido a la percepción de las personas, esto se debe a que antes de manipular los objetos primero debemos saber dónde están ubicados.

2.3.4. ROS

El sistema operativo de robot ROS (*Robot Operating System*), es un *framework* flexible para escribir el software de un robot. Tiene una gran variedad de herramientas, librerías, etc, que permiten simplificar la tarea de crear sistemas complejos para robots robustos. [24]

Los elementos que se desarrollan en ROS son: nodos, servicios y tópicos que fueron implementados en el trabajo de grado.

1. **Nodo**: Un nodo es un archivo ejecutable dentro de los paquetes que vienen incluidos en ROS. Dentro de ROS hay diferentes tipos de mensajes que permiten la comunicación entre nodos, estos nodos pueden enviar mensajes mediante tópicos y servicios.

Los nodos son ejecutables que utilizan diferentes lenguajes de programación, estos lenguajes son: Python, C++, C, Java, etc. El lenguaje desarrollado para nuestro proyecto de grado fue Python.

Mediante mensajes se puede entender la información entre lenguajes de programación, es decir, que si tengo un nodo en C++ y un nodo en Python el sistema de comunicación se puede establecer por medio de mensajes.

2. **Tópico**: Un tópico es un tipo de mensaje que permite la comunicación entre nodos; no obstante, un tópico es un nodo al cual se pueden enviar o recibir mensajes. La figura 2.13 representa la comunicación entre nodos mediante tópicos.



Figura 2.13: Tópicos en ROS [14]

3. **Servicio**: Un servicio es otro medio de comunicación entre nodos, los servicios son bidireccionales, es decir, que los nodos que envían un mensaje para comunicarse con otros nodos pueden recibir un mensaje de los nodos receptores. La figura 2.14 representa la comunicación entre nodos mediante un servicio.

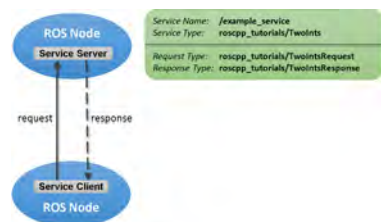


Figura 2.14: Servicios en ROS [13]

Capítulo 3

Funciones de expresión emocional

3.1. Introducción

En este capítulo se describe el proceso de diseño e implementación de las funciones de expresión emocional del robot Baxter. Inicia con una descripción general del modelo teórico para identificar los ejes del estado emocional del robot: control y aceptación, que regulan las acciones emocionales del robot. Con los parámetros identificados describiremos las funciones de expresión emocional que se activan con el uso de estos ejes, para que finalmente activen los actuadores del robot expresando las emociones reflejadas en los ejes de control y aceptación

El comportamiento emocional del robot considera dos ejes que han sido ampliamente abordados desde el campo educativo [27]. El eje aceptación indica la aprobación que tiene el tutor acerca de las acciones del aprendiz durante el proceso de aprendizaje. El eje control surge de las acciones de compensación emocional que adelanta el tutor para mantener motivado al estudiante durante el proceso de aprendizaje. De acuerdo con la literatura, la relación entre estos dos aspectos condiciona el comportamiento del experto como tutor, par o aprendiz. El comportamiento emocional del tutor está orientado a retroalimentar las acciones del aprendiz mediante acciones emocionales que insinúan el desempeño del estudiante. De manera similar, las acciones del par están orientadas a desarrollar comportamientos semejantes a los del aprendiz; esta condición rompe con la relación vertical tradicional entre el tutor y aprendiz y hace de ésta una relación horizontal. Finalmente, los ejes de control y aceptación en la condición del rol de aprendiz sugieren un comportamiento emocional del robot orientado a que el aprendiz sienta la necesidad emocional de orientar al robot en el aprendizaje.

3.2. Descripción detallada de las funciones Expresión Emocional

En el contexto de este documento, las funciones de expresión emocional están definidas como los actos no verbales que ejecuta el robot Baxter para transmitir emociones. Las funciones involucran los siguientes recursos físicos: pantalla, parlante, ventiladores, anillos de LEDs y velocidad de los actuadores. La figura 3.1 presenta un diagrama de bloques que articula el uso de la arquitectura HRS con ROS. La arquitectura HRS ha sido diseñada para dar soporte emocional al aprendiz durante el proceso de solución de problemas, de transformación con bloques como: juego de la escalera, apilar bloques y juego de los cuatro caballos.

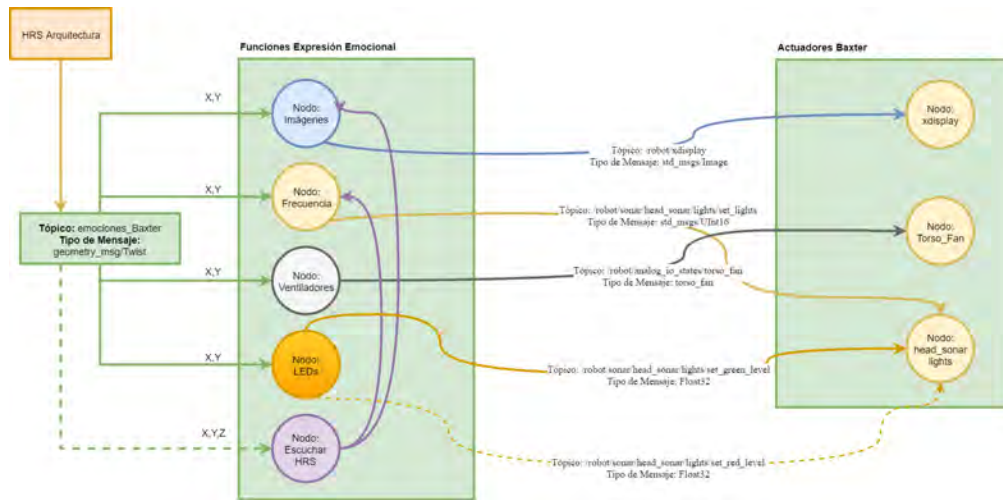


Figura 3.1: Diagrama de bloques de las funciones de expresión emocional

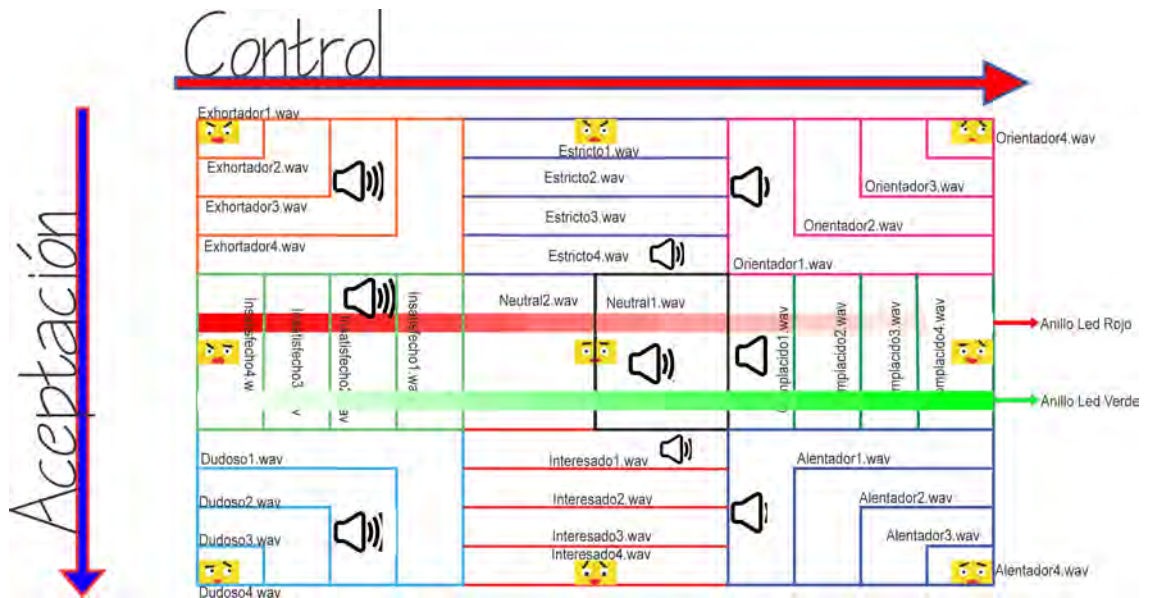


Figura 3.2: Representación de las funciones de expresión emocional según sus parámetros de entrada (Control y Aceptación)

Las funciones de expresión emocional presentan movimientos, sonidos y colores asociados al estado emocional del robot, en función de los ejes emocionales asociados al proceso de enseñanza: control y aceptación (Ver figura 3.2). El primero, indica el rol del robot como tutor, par o aprendiz durante la actividad de aprendizaje. El segundo, está relacionado con la evaluación emocional que hace el robot de las acciones del aprendiz para solucionar el problema. Las funciones de expresión emocional permiten el control de los actuadores del robot de acuerdo con las variables enviadas por la arquitectura. Por ejemplo, si la arquitectura HRS estima que el comportamiento del robot en un instante es presentar emociones de felicidad, entonces recibirá dos parámetros tipo entero para modular los actuadores.

3.2.1. HRS Arquitectura

En la figura 3.1, el bloque llamado Human-Robot Scaffolding, representa de manera general la arquitectura desarrollada en el contexto de la investigación de tesis doctoral. La arquitectura identifica el estado emocional del aprendiz, el estado cognitivo relacionado con el conocimiento del problema y el estado de la tarea de los bloques. Con la información percibida, la arquitectura estima las acciones emocionales y cognitivas apropiadas para orientar el proceso de aprendizaje del estudiante. Las acciones emocionales del robot buscan retroalimentar las acciones del estudiante y mantenerlo motivado durante el proceso de aprendizaje. Para el diseño de las funciones de expresión emocional, se analizaron posibilidades físicas de intervención del robot como movimiento de los brazos, posibilidades de expresión en la pantalla, los dispositivos de luz que tiene integrado y los demás actuadores que permitan dar indicios de estados emocionales del robot. Por ejemplo, la activación de los ventiladores indica que el robot está estricto.

La salida del bloque HRS muestra el tipo de mensaje que se envía a los nodos que están suscritos a este mensaje. El mensaje del HRS es una variable tipo Twist y se define de la siguiente manera:

$$Twist : \begin{cases} Linear : \begin{cases} float x \\ float y \\ float z \end{cases} \\ Angular : \begin{cases} float x \\ float y \\ float z \end{cases} \end{cases} \quad (3.1)$$

En la figura 3.1 se muestra que el tipo de mensaje que conecta el bloque HRS con los nodos del bloque de funciones de expresión emocional es un Twist. Los valores del mensaje que se enviaron para activar los nodos fueron: linear.x, linear.y y linear.z.

En las variables linear de Twist X es el control, Y es la aceptación y Z se usa solamente en el nodo Emociones imágenes Baxter. Después del recorrido de las imágenes, para mayor atención e interacción con el niño, se añadieron dos funciones dentro de este nodo, de modo que el robot también pueda hablar o guiñar el ojo como lo muestran las figuras 3.3 y la figura 3.4.



Figura 3.3: Frame de imágenes para caracterizar como habla el robot



Figura 3.4: Frame de imágenes para caracterizar como guiña el robot

Las funciones de expresión emocional han sido diseñadas para controlar de manera asíncrona los actuadores del robot. Deben ser entendidas como una configuración que transmite al aprendiz estados emocionales del robot asociados al aprendizaje. Si bien, existen diferentes alternativas para transmitir emociones, las funciones diseñadas consideran dos aspectos: el sistema operativo ROS (*Robotic Operative System*) y las características técnicas del robot Baxter. ROS nos permite diseñar las funciones emocionales como nodos que se comunican con los nodos de la capa de control mediante la suscripción a tópicos para cada uno de los actuadores. Por ejemplo, si la intención es controlar la pantalla de robot Baxter, la función emocional debe suscribirse al tópico: `/robot/xdisplay`.

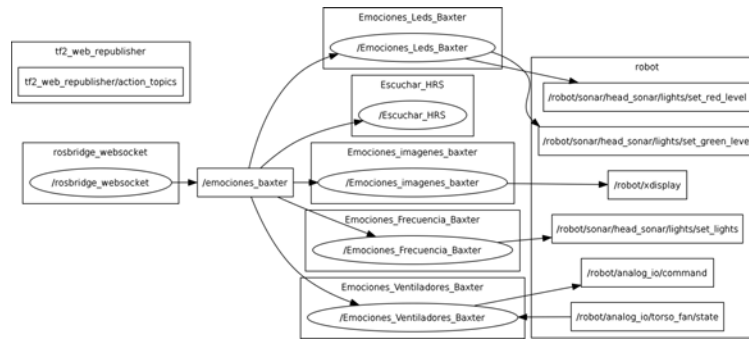


Figura 3.5: Diagrama específico de los nodos y tópicos en ROS

La figura 3.5, es el diagrama en tiempo real que tiene implementado ROS, el diagrama que tenemos al comienzo es nuestra conexión con el bloque HRS mostrado en la figura 3.1; la diferencia en estas figuras es que en el diagrama de ROS no se pueden observar los servicios.

3.2.2. Bloque Función de expresión emocional

Las salidas del bloque que representa la arquitectura para el soporte emocional están agrupadas en los nodos-ROS llamados: Emociones Imágenes Baxter, Emociones Leds Baxter, Emociones Ventiladores Baxter, Emociones Frecuencia Baxter.

1. **Nodo Emociones Imágenes Baxter:** es una función que recibe los parámetros de control y aceptación, estos parámetros se describen en esta función como una imagen. Los puntos varían dependiendo del HRS, por tanto, cuando hay un cambio de parámetros no se muestran 2 imágenes, sino la transición de la imagen de los parámetros anteriores, con los parámetros nuevos.



Figura 3.6: Transición de imágenes

La figura 3.6 es un ejemplo de las transiciones que pueden hacer las imágenes, en donde los extremos son el punto de inicio/partida. El pseudocódigo 1 representa el nodo Imágenes Baxter, el cual determina a partir de los estados de control y aceptación las imágenes que representan las caras del robot Baxter. De acuerdo al pseudocódigo 1 las imágenes establecidas están representadas como la figura 3.2.

Algorithm 1 Nodo Imágenes Baxter

```
while exista control and aceptación do  
  Función(emociones iniciales(x,y), emociones finales(x,y))  
  Función escalonada → emociones iniciales a emociones finales  
  Return(recorrido coordenadas(x,y))  
  Vector imágenes → recorrido de coordenadas  
  for 0: longitud de imágenes do  
    Publicar tópico de Vector imágenes  
    if Robot Habla then  
      Escuchar sonido mientras robot habla  
    else if Robot Guiña ojo then  
      Transición de imágenes con ilustración del guiño del robot  
    end if  
  end for  
end while
```

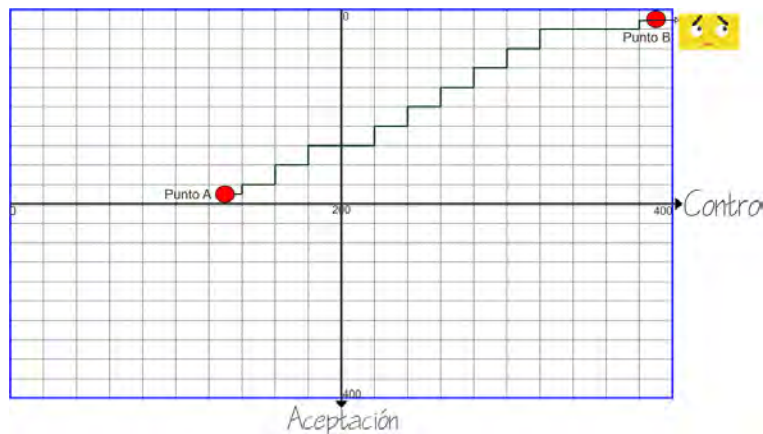


Figura 3.7: Función escalonada

La figura 3.7 es un ejemplo de la función escalonada; esta función representa el recorrido más corto, se debe realizar para llegar del punto inicial al punto final que son representados por los parámetros de entrada (control y aceptación); para definir el camino más corto se calculó la pendiente del punto inicial al punto final, mediante la ecuación:

$$m = \frac{y_{final} - y_{inicial}}{x_{final} - x_{inicial}} \quad (3.2)$$

Después de hallar la pendiente m de la ecuación 3.2 se reemplaza la pendiente en la siguiente función:

$$f(x) = m \cdot x + b \quad (3.3)$$

Con la ecuación 3.3 hallamos los puntos mínimos que se tienen que recorrer del punto inicial al punto final, representando las coordenadas de los números de las imágenes para las transiciones de las caras del robot.

Los puntos de transición para llegar al punto final representan el número de la imagen que se debe mostrar en la cara de Baxter, por tanto, la función retorna los números que representan la transición de las imágenes, desde el número de la imagen inicial hasta el número de la imagen final.

Los valores de la función escalonada serán almacenados en un vector de imágenes; no obstante, antes de almacenarlos necesitamos que cada uno de los números que retorna la función, tengan la extensión que nos permite abrir la imagen en la cara del robot.

Después de guardar todos los números de las imágenes con su extensión en el vector imágenes, se crea un ciclo para mandarle un mensaje a Baxter diciéndole que muestre la transición de imágenes desde el punto de inicio hasta el punto final.

Después de que se publican cada una de las imágenes, solo queda mostrada la imagen final en la cara del robot.

Finalmente, la imagen final será ahora la imagen inicial.

Algorithm 2 Código Función Escalonada

```
#!/usr/bin/python2
...
def send_image(path):
path = '/home/z420/ros_ws/src/baxter_examples/share/images/'+path
img = cv2.imread(path)
newimage = cv2.resize(img,(1024,600))
msg = cv_bridge.CvBridge().cv2_to_imgmsg(newimage, encoding="bgr")
pub = rospy.Publisher('/robot/xdisplay', Image, latch=True, queue_size=1)
pub.publish(msg)
...
..
.

Espacio_Emociones = np.reshape(np.arange(400),(20,20))
a=1
for x in range(0, 20):
for y in range(0, 20):
Espacio_Emociones[x][y] = a
a = a+1
im="" for x in range(1,401)
nombre=".png"

while (x1 < x2) & (y1 < y2):
if x1 < x2:
x1 = x1+1
if y1 < y2:
y1 = y1+1
im[array]=str(Espacio_Emociones[y1,x1])+nombre
array=array+1
...
..
.
imageess= ["" for x in range(0,array)]
for x in range(array):
imageess[x]=im[x]
for path in imageess:
send_image(path)
```

Los puntos X1 y Y1 representan el punto de inicio (Punto A), es decir la primera imagen que realizara el recorrido para llegar a la imagen final, y los puntos X2 y Y2 representan el punto final (Punto B), es decir la última imagen del recorrido.

Para cumplir la menor distancia del Punto A al Punto B, los puntos están en un eje de coordenadas como el de la figura 3.7; existen 4 condiciones para obtener el menor recorrido de la imagen inicial hasta

la imagen final, estas condiciones son:

$$\left\{ \begin{array}{l} \textit{Primera Condición:} (x1 < x2) \textbf{ and } (y1 < y2) \\ \textit{Segunda Condición:} (x1 > x2) \textbf{ and } (y1 > y2) \\ \textit{Tercera Condición:} (x1 < x2) \textbf{ or } (y1 > y2) \\ \textit{Cuarta Condición:} (x1 > x2) \textbf{ or } (y1 < y2) \end{array} \right. \quad (3.4)$$

Por lo tanto, son necesarios cuatro ciclos, donde básicamente se realiza el procedimiento del código mencionado anteriormente; si los puntos X1 o Y1 son menores, se aumentará el valor de estas variables en una unidad; por el contrario si los puntos son mayores, se restará el valor de estas variables en una unidad.

1.1 Sonidos Robot Baxter: Los sonidos se introdujeron en el robot Baxter para tener una mayor percepción de las emociones del robot Baxter; los sonidos son una herramienta fundamental para la expresión de emociones, porque al producir un sonido el robot puede llamar la atención del usuario permitiendo una interacción humano-robot.

Los sonidos fueron obtenidos con recursos de la tesis doctoral mediante la página [31], la cual permite la compra de los sonidos del robot Bleep. Esta librería contiene gran variedad de emociones, por tanto, las emociones del robot Bleep fueron implementadas en el robot Baxter.

La figura 3.2, representa la ubicación de los sonidos dependiendo en que eje emocional se encuentra, por cada emoción tendremos cuatro sonidos. Las líneas que están pintadas por cada eje emocional representan el espacio que tiene cada sonido en el plano cartesiano.

El pseudocódigo 3 representa el funcionamiento de los sonidos.

Algorithm 3 Sonidos Bleep para el robot Baxter

```
if (Control >= 0 and Control < 134) and (Aceptación >= 0 and Aceptación < 134) then
  Sonidos Pertenecientes Exhortador (Nombres con Exhortador.wav), y su distribución como la figura
  3.2
  while Reproduzca Sonido do
    Robot Habla
  end while
end if
if (Control >= 134 and Control < 267) and (Aceptación >= 0 and Aceptación < 134) then
  Sonidos Pertenecientes Estricto (Nombres con Estricto.wav), y su distribución como la figura 3.2
  while Reproduzca Sonido do
    Robot Habla
  end while
end if
if (Control >= 267 and Control < 400) and (Aceptación >= 0 and Aceptación < 134) then
  Sonidos Pertenecientes Orientador (Nombres con Orientador.wav), y su distribución como la figura 3.2
  while Reproduzca Sonido do
    Robot Habla
  end while
end if
if (Control >= 0 and Control < 134) and (Aceptación >= 134 and Aceptación < 267) then
  Sonidos Pertenecientes Insatisfecho (Nombres con Insatisfecho.wav), y su distribución como la figura
  3.2
  while Reproduzca Sonido do
    Robot Habla
  end while
end if
if (Control >= 134 and Control < 267) and (Aceptación >= 134 and Aceptación < 267) then
  Sonidos Pertenecientes Neutral (Nombres con Neutral.wav), y su distribución como la figura 3.2
  while Reproduzca Sonido do
    Robot Habla
  end while
end if
if (Control >= 267 and Control < 400) and (Aceptación >= 134 and Aceptación < 267) then
  Sonidos Pertenecientes Complacido (Nombres con Complacido.wav), y su distribución como la figura
  3.2
  while Reproduzca Sonido do
    Robot Habla
  end while
end if
if (Control >= 0 and Control < 134) and (Aceptación >= 267 and Aceptación < 400) then
  Sonidos Pertenecientes Dudoso (Nombres con Dudoso.wav), y su distribución como la figura 3.2
  while Reproduzca Sonido do
    Robot Habla
  end while
end if
if (Control >= 134 and Control < 267) and (Aceptación >= 267 and Aceptación < 400) then
  Sonidos Pertenecientes Interesado (Nombres con Interesado.wav), y su distribución como la figura 3.2
  while Reproduzca Sonido do
    Robot Habla
  end while
end if
if (Control >= 267 and Control < 400) and (Aceptación >= 267 and Aceptación < 400) then
  Sonidos Pertenecientes Alentador (Nombres con Alentador.wav), y su distribución como la figura 3.2
  while Reproduzca Sonido do
    Robot Habla
  end while
end if
```

Debido a que los valores de control y aceptación varían de 0-400, los sonidos son repartidos sobre todo el eje cartesiano y según sus valores de control y aceptación, el sonido que sea reproducido dependerá de estos parámetros.

Mientras el robot emita un sonido, las imágenes del robot realizarán la transición de los gestos de hablado como el de la figura 3.3, y las imágenes dependerán de los parámetros de control y aceptación

2. **Nodo Emociones Leds Baxter** es una función que depende de los parámetros de entrada (control y aceptación), con estos parámetros se enviarán mensajes al robot que le permitan activar el anillo de leds.

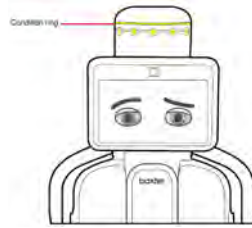


Figura 3.8: Anillo de leds que se activan con el nodo Emociones Leds Baxter [23].

Los anillos de leds pueden ser de color rojo o verde, esta función controlará la iluminación del anillo de leds entorno a sus dos colores. El pseudocódigo 4 representa el nodo Leds, este nodo está asociado con los anillos de leds de la figura 3.8, controla la iluminación de este anillo de leds dependiendo de los parámetros de entrada.

Algorithm 4 Nodo Leds Baxter

```

while exista control and aceptación do
  Anillo Verde = (Control*100)/400
  Anillo Rojo = (Aceptación*100)/400
  Publicar tópico iluminación Anillo led verde según el valor de Anillo Verde
  Publicar tópico iluminación Anillo led rojo según el valor de Anillo Rojo
end while

```

El anillo de LEDs tiene los colores rojo y verde; no obstante, están definidos por diferentes tópicos, pero el actuador es el mismo.

El control de iluminación del anillo de LEDs está representado por las siguientes ecuaciones:

$$AnilloVerde = \frac{Control \cdot 100}{400} \quad (3.5)$$

$$AnilloRojo = \frac{Aceptacion \cdot 100}{400} \quad (3.6)$$

Dependiendo de los parámetros de control, la iluminación del robot será diferente, la figura 3.2, representa como será la iluminación de cada anillo de LEDs si se varía el Control, es decir el eje x.

Si el punto que se envía en el eje x está en la parte izquierda del plano cartesiano de la figura 3.2, el robot presentará inconformidad con el desempeño del usuario y el anillo de LEDs rojo tendrá su máxima iluminación; el anillo de LEDs verde tendrá una iluminación baja, por tanto, el anillo de LEDs que se observara será de color rojo.

3. **Nodo Emociones Ventiladores Baxter** se activa con los parámetros de control y aceptación, este nodo envía mensajes al ventilador que está en el torso como se puede observar en la siguiente figura.



Figura 3.9: Ubicación Ventilador Baxter [23].

Esta función permite controlar la velocidad del ventilador; dependiendo de los parámetros de entrada el ventilador sonara rápido o no sonara. El pseudocódigo 5 representa el nodo Ventilador Baxter, este nodo dependiendo de sus parámetros de entrada (control y aceptación), enciende y controla la velocidad del ventilador que tiene incorporado Baxter.

Algorithm 5 Nodo Ventilador Baxter

```

while exista control and aceptación do
  if (Control >= 150 and Control <= 250) and (Aceptación >= 150 and Aceptación <= 250) then
    0.3 * Velocidad Máxima del Ventilador
  else if (Control >= 250) and (Aceptación >= 150 and Aceptación <= 250) then
    0* Velocidad Máxima del Ventilador
  else if (Control > 250) and (Aceptación >= 0 and Aceptación <= 150) then
    0.2* Velocidad Máxima del Ventilador
  else if (Control > 250) and (Aceptación >= 250 and Aceptación <= 400) then
    0.1* Velocidad Máxima del Ventilador
  else if (Control < 150) and (Aceptación > 150 and Aceptación < 250) then
    Velocidad Máxima del Ventilador
  else if (Control < 150) and (Aceptación > 0 and Aceptación <= 150) then
    0.8* Velocidad Máxima del Ventilador
  else if (Control < 150) and (Aceptación >= 250 and Aceptación <= 400) then
    0.6* Velocidad Máxima del Ventilador
  end if
end while

```

La figura 3.2 representa un plano cartesiano; para el eje emocional, cabe mencionar que los puntos en el eje x, y en el eje y es de 0-400, donde su punto de origen es (200,200).

Por tanto, en el pseudocódigo 4 se representa la velocidad de movimiento del ventilador; dependiendo de los parámetros de entrada; es decir, por cada eje emocional la velocidad del ventilador cambiara. En la figura 3.2 hay un parlante, el cual representa el sonido del ventilador, en la parte izquierda del plano cartesiano la velocidad del ventilador será máxima, y el sonido del ventilador será muy fuerte; es decir, que en el plano cartesiano el sonido del parlante esta al máximo. Por otro lado, en la parte central del plano cartesiano los parlantes se muestran con la mitad del volumen, esto quiere decir que la velocidad del ventilador estará a la mitad de su máxima potencia. Por último, en la parte derecha del plano cartesiano los parlantes aparecen casi sin líneas de sonidos, entonces la velocidad del parlante será nula o muy pequeña.

Cada estado de velocidad del ventilador representa el estado de cada eje emocional, donde el eje izquierdo representa inconformidad, el eje central representa un estado neutro y el eje derecho representa un esta alegre.

4. **Nodo Emociones Frecuencia Baxter**, es una función que envía un mensaje para activar el anillo de led como se observa en la siguiente figura:

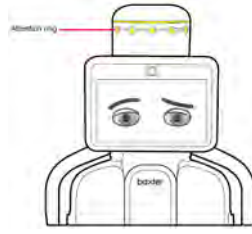


Figura 3.10: Anillo led que se controlan por el nodo Emociones Frecuencia Baxter [23].

Los parámetros de entrada (control y aceptación), controlan la frecuencia de encendido y apagado de los LEDs; ejemplo si el robot está tranquilo entonces el anillo de leds se apagará y se prenderá intermitentemente muy despacio, si por el contrario el robot está furioso el anillo de LEDs se apagará y prenderá intermitentemente, muy rápido.

El pseudocódigo 6 representa la Frecuencia LEDs, este nodo está asociado con los anillos LEDs de la figura 3.10 y controla la frecuencia de iluminación de este anillo de leds dependiendo de los parámetros de entrada.

Algorithm 6 Nodo Frecuencia Baxter

```

while exista control and aceptación do
  Frecuencia = 10.1 - (Control * 10) / 400
  Publicar tópico Frecuencia del parpadeo de los leds según Frecuencia
end while

```

La frecuencia del parpadeo de los LEDs está definida por:

$$Frecuencia = \frac{10,1 - (Control \cdot 10)}{400} [Hz] \quad (3.7)$$

En la ecuación 3.7, el valor de 10.1 es la máxima frecuencia en la que se encenderán y se apagaran los leds (cuando está en el eje insatisfecho de la figura 3.2, significa que la frecuencia será máxima), es decir, el parpadeo será mucho más rápido que cuando está en el eje complacido, donde su frecuencia será mucho más lenta.

5. **Actuadores:** El bloque de actuadores del robot Baxter representa la capa de control del robot. Esta capa contiene los nodos, tópicos y servicios asociados al control de cada uno de los actuadores del robot en funciones como: movimiento de los actuadores, control de velocidad del movimiento, sistema de sensores ultrasónicos para la percepción del ambiente, control y uso de las cámaras ubicadas en cada uno de los brazos, sensores infrarrojos ubicados en cada brazo. El diseño de las funciones de expresión emocional requiere el desarrollo de nodos de acuerdo con las características de los tópicos de ROS y del robot Baxter.

Las funciones de expresión emocional explicadas anteriormente, son enviadas cada una de ellas mediante mensajes, estos mensajes son llamados tópicos. Los tópicos de las funciones implementadas se subscriben a los nodos de los actuadores de Baxter.

La figura 3.1 en el diagrama de actuadores del robot indican cuales nodos de Baxter son usados.

- 5.1 **Xdisplay:** es el nodo que permite activar la pantalla del robot Baxter y mostrar la cara (o transición de imágenes) que la función Emociones imágenes Baxter halla mandado.
- 5.2 **Torso fan:** es el nodo que controla los ventiladores de Baxter, por lo tanto, la función de Emociones Ventiladores Baxter activa y controla la velocidad de los ventiladores.
- 5.3 **Head sonar lights:** es el nodo que viene conectado a los dos anillos de la cabeza de Baxter, y dependiendo de que tópico se halla enviado activaran los anillos led de la figura 3.8 o el de la figura 3.10. Si se activan los de la figura 3.8, entonces se controlará la iluminación del anillo de led rojo y verde; en cambio sí se activan el anillo de led de la figura 3.10 se prenderán y encenderán el anillo de led a cierta frecuencia.

3.3. Validación de Resultados

La validación de resultados involucra 2 actividades.

1. El análisis de la interacción de las 6 estudiantes de la carrera de licenciatura en pedagogía infantil de la Pontificia Universidad Javeriana. En esta experiencia las estudiantes interactúan con el robot durante el proceso de solución del problema de la escalera.

Aquí, el robot fue controlado mediante la técnica del *Mago de Oz*, lo que indica que un operador humano controlaba las emociones y movimientos del robot que fueron diseñados de acuerdo al objetivo de este trabajo de grado.

La información que se les dio a los estudiantes fue que el robot iba a ser un tutor para ellos en la solución de un problema, además ellas sabían que el robot iba a expresar emociones según su desempeño a la prueba, y finalmente el usuario que manejo el robot les explicaba el problema a resolver. Donde Baxter les ayudaba cuando ellos pedían ayuda.

Para la primera estudiante se observa en la figura 3.11, que al desarrollar la actividad prestaba atención a las expresiones emocionales de Baxter, por tanto, cuando el robot estaba alegre ella reconocía que está haciendo bien las cosas y cuando Baxter estaba enojado ella reconocía que tenía que replantear el problema y pensar mejor la solución. Esto indica que las emociones diseñadas transmiten la información asertiva al usuario como mecanismo de retroalimentación emocional.

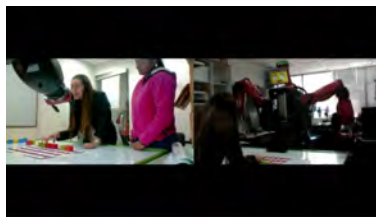


Figura 3.11: Interacción Baxter con la primera estudiante.

La segunda estudiante realizó el problema sin prestarle atención a Baxter como se muestra en la figura 3.12, esto se debe a que algunas personas cuando están desarrollando una tarea no quieren ser ayudados por terceras personas, haciendo que Baxter pase totalmente inadvertido. Sin embargo, la actitud de la estudiante no sugiere un rechazo a las emociones diseñadas, simplemente es una estrategia de la estudiante para lograr mayor concentración en el problema.

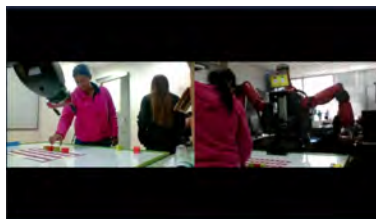


Figura 3.12: Interacción Baxter con la segunda estudiante.

Un primer acercamiento con un robot la mayoría de las personas demuestran un sentimiento de miedo como se muestra en la figura 3.13; la tercera estudiante cuando empezó a desarrollar el problema se alejaba de Baxter pensando que este podía golpearla cuando él se movía. Pero al ver que Baxter solo expresaba sus emociones y que este se movería solo si ella necesitaba ayuda en el problema, empezó a desarrollar el problema con menos miedo y al final termino el problema de la escalera sin ningún inconveniente.

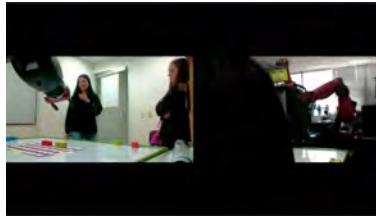


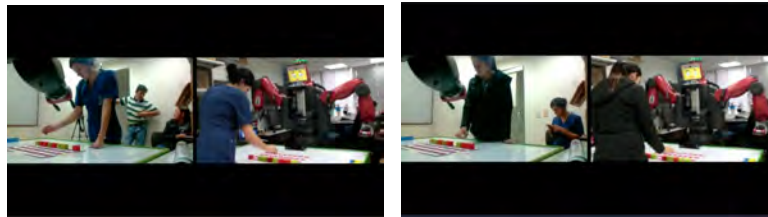
Figura 3.13: Interacción Baxter con la tercera estudiante.

La cuarta estudiante fue la de mayor interacción con Baxter, ella se percataba de las emociones que hacía Baxter y como se muestra en la figura 3.14, sabía que no estaba realizando bien las cosas y sintió frustración. Finalmente, con la ayuda de Baxter ella pudo resolver el problema de la escalera.



Figura 3.14: Interacción Baxter con la cuarta estudiante.

La quinta y sexta estudiante, al igual que la segunda se enfocaron más en la solución del problema que en la intervención de Baxter para desarrollar el problema como se observa en la figura 3.15.



(a) Interacción Baxter con la quinta estudiante. (b) Interacción Baxter con la sexta estudiante.

Figura 3.15: Interacción Baxter con la quinta y sexta estudiante.

La interacción humano-robot dentro del proceso de enseñanza-aprendizaje nos permitió encontrar diferentes reacciones de los usuarios frente al robot:

- a. Dos estudiantes interactuaron de manera fluida con el robot
- b. Tres estudiantes ignoraron la presencia del robot, pues su objetivo era realizar la actividad
- c. Una estudiante mostró temor a la interacción con el robot.

Las experiencias pedagógicas marcan la conducta del educando en los diferentes ambientes naturales y/o tecnológicos.

2. Para validar los resultados se realizó una rutina de movimientos, donde el robot tenía que seguir una demostración de movimientos predeterminados, esta secuencia se repitió 8 veces debido a que cada demostración representaba una emoción distinta (Complacido, orientador, estricto, exhortador, insatisfecho, dudoso, interesado, alentador).

La encuesta fue aplicada a través de la plataforma Google, contenía el video del robot realizando las 8 demostraciones pertenecientes a cada emoción. Cada emoción estaba explicada en la encuesta. Las preguntas estaban enfocadas a establecer la emoción que se hizo en cada demostración, la encuesta se encuentra en anexos y las opciones de respuesta fueron:

1. Complacido.
2. Orientador.
3. Estricto.
4. Exhortador.
5. Insatisfecho.
6. Dudoso.
7. Interesado.
8. Alentador.
9. Ninguna de las anteriores.

Las respuestas esperadas, es decir, las emociones que se diseñaron para cada demostración son:

1. Exhortador.
2. Estricto.
3. Orientador.
4. Insatisfecho.
5. Dudoso.
6. Complacido.
7. Interesado.
8. Alentador.

La encuesta fue aplicada a 22 personas de diferentes edades y formación académica; los resultados que arrojó la encuesta fueron:

a) Primera Pregunta:

Pregunta: En la demostración 1 de Baxter cual considera que es la emoción predominante.

228n6sp,respuestas

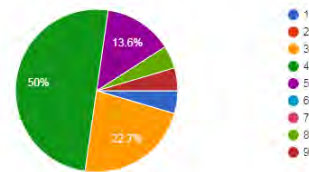


Figura 3.16: Resultados Primera Pregunta.

Emociones	Porcentaje de respuesta de los encuestados de la emoción predominante (Pregunta 1)	Personas que contestaron la pregunta 1
Complacido	0 %	0
Orientador	0 %	0
Estricto	22.7 %	5
Exhortador	50 %	11
Insatisfecho	13.6 %	3
Dudoso	0 %	0
Interesado	0 %	0
Alentador	0 %	0
Ninguna de las Anteriores	0 %	0

Cuadro 3.1: Resultados Pregunta 1

De acuerdo con la figura 3.16 y a la tabla 3.1 podemos observar que la emoción exhortadora fue la emoción predominante para la mayoría de los encuestados (once personas) en la primera pregunta. La primera pregunta diseñada teóricamente fue para la emoción exhortadora.

El error de la emoción exhortadora fue:

$$Error \ Emoción \ Exhortadora = \frac{22 - 11}{22} = 0,5 = 50\% \quad (3.8)$$

El error entre el resultado teórico y experimental fue del 50 %, este porcentaje de error se debe a que la emoción de exhortador tiene una gran similitud con las emociones de estricto e insatisfecho, esto obedece a que las caras del robot manifestaban una emoción de rabia, por tanto, para los usuarios esta emoción representaba sentimientos de enojo.

b) Segunda Pregunta:

Pregunta: En la demostración 2 de Baxter cual considera que es la emoción predominante.

228 respuestas

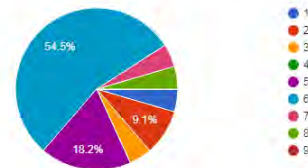


Figura 3.17: Resultados Segunda Pregunta.

Emociones	Porcentaje de respuesta de los encuestados de la emoción predominante (Pregunta 2)	Personas que contestaron la pregunta 2
Complacido	0 %	0
Orientador	9.1 %	2
Estricto	0 %	0
Exhortador	0 %	0
Insatisfecho	18.2 %	4
Dudoso	54.5 %	12
Interesado	0 %	0
Alentador	0 %	0
Ninguna de las Anteriores	0 %	0

Cuadro 3.2: Resultados Pregunta 2

De acuerdo con la figura 3.17 y a la tabla 3.2 se observa que la emoción de duda fue la emoción predominante que los encuestados fácilmente identificaron. La segunda pregunta diseñada teóricamente fue para la emoción: estricto.

El error de la emoción cuando el robot estaba estricto fue:

$$Error\ Estricto = \frac{22 - 0}{22} = 1 = 100\% \quad (3.9)$$

La emoción estricta no fue percibida por ninguno de los encuestados, en esta emoción los encuestados percibieron una emoción de duda, insatisfacción y orientación.

c) **Tercera Pregunta:**



Figura 3.18: Resultados Tercera Pregunta.

Emociones	Porcentaje de respuesta de los encuestados de la emoción predominante (Pregunta 3)	Personas que contestaron la pregunta 3
Complacido	0 %	0
Orientador	0 %	0
Estricto	0 %	0
Exhortador	0 %	0
Insatisfecho	13.6 %	3
Dudoso	40.9 %	9
Interesado	13.6 %	3
Alentador	13.6 %	3
Ninguna de las Anteriores	0 %	0

Cuadro 3.3: Resultados Pregunta 3

De acuerdo con la figura 3.18 y a la tabla 3.3 la emoción de duda fue la predominante que los encuestados identificaron. La tercera pregunta diseñada teóricamente fue para la emoción orientadora. El error de la emoción orientadora fue:

$$Error\ Emoción\ Orientadora = \frac{22 - 0}{22} = 1 = 100\% \quad (3.10)$$

De los encuestados ninguno respondió la emoción orientadora, al igual que la segunda pregunta la emoción duda fue percibida en la mayoría de los usuarios, las otras emociones que percibieron fueron insatisfecho. En la segunda y tercera pregunta se observa que los encuestados coinciden con una emoción de duda muy diferente a la definida en la encuesta, es decir, la expresión del robot se interpretó como duda y no como la función establecida.

d) **Cuarta Pregunta:**

Pregunta: En la demostración 4 de Baxter cual considera que es la emoción predominante.

228 respuestas

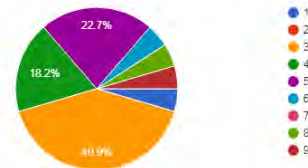


Figura 3.19: Resultados Cuarta Pregunta.

Emociones	Porcentaje de respuesta de los encuestados de la emoción predominante (Pregunta 4)	Personas que contestaron la pregunta 4
Complacido	0 %	0
Orientador	0 %	0
Estricto	40.9 %	9
Exhortador	18.2 %	4
Insatisfecho	22.7 %	5
Dudoso	0 %	0
Interesado	0 %	0
Alentador	0 %	0
Ninguna de las Anteriores	0 %	0

Cuadro 3.4: Resultados Pregunta 4

Los resultados de la figura 3.19 y a la tabla 3.4 muestran que la emoción predominante en la cuarta pregunta fue: estricto.

En la cuarta pregunta encontramos que la emoción diseñada teóricamente fue para: Insatisfecho.

El error de Insatisfecho fue:

$$Error\ Insatisfecho = \frac{22 - 5}{22} = 0,77 = 77\% \quad (3.11)$$

Las respuestas encontradas nos permiten concluir que se vio una confusión de similitud con la emoción de estricto, debido a que estas emociones representan sentimientos de furia e ira.

e) **Quinta Pregunta:**

Pregunta: En la demostración 5 de Baxter cual considera que es la emoción predominante.

228 respuestas

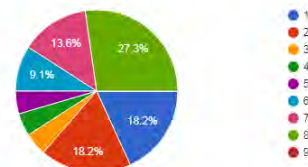


Figura 3.20: Resultados Quinta Pregunta.

Emociones	Porcentaje de respuesta de los encuestados de la emoción predominante (Pregunta 5)	Personas que contestaron la pregunta 5
Complacido	18.2 %	4
Orientador	18.2 %	4
Estricto	0 %	0
Exhortador	0 %	0
Insatisfecho	0 %	0
Dudoso	9.1 %	2
Interesado	13.6 %	3
Alentador	27.3 %	6
Ninguna de las Anteriores	0 %	0

Cuadro 3.5: Resultados Pregunta 5

Los resultados de la figura 3.20 y a la tabla 3.5 muestran que la emoción predominante en la quinta pregunta es: alentador.

En la quinta pregunta encontramos que la emoción diseñada teóricamente fue para: Dudoso.

El error de la emoción de duda fue:

$$\text{Error emoción de duda} = \frac{22 - 2}{22} = 0,91 = 91 \% \quad (3.12)$$

En esta quinta pregunta se concluye que hubo dispersión en las respuestas al no presentarse ninguna emoción destacada, por tanto, esta emoción no fue comprendida.

f) Sexta Pregunta:

Pregunta: En la demostración 6 de Baxter cual considera que es la emoción predominante.

228 respuestas

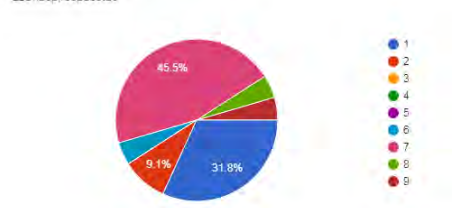


Figura 3.21: Resultados Sexta Pregunta.

Emociones	Porcentaje de respuesta de los encuestados de la emoción predominante (Pregunta 6)	Personas que contestaron la pregunta 6
Complacido	31.8 %	7
Orientador	9.1 %	2
Estricto	0 %	0
Exhortador	0 %	0
Insatisfecho	0 %	0
Dudoso	0 %	0
Interesado	45.5 %	10
Alentador	0 %	0
Ninguna de las Anteriores	0 %	0

Cuadro 3.6: Resultados Pregunta 6

Los resultados de la figura 3.21 y a la tabla 3.6 muestran que la emoción predominante en la sexta pregunta fue: Interesado.

En la sexta pregunta encontramos que la emoción diseñada teóricamente fue para: Complacido.

El error de Complacido fue:

$$Error\ Complacido = \frac{22 - 7}{22} = 0,68 = 68\% \quad (3.13)$$

En la sexta pregunta las dos emociones que resaltaron fue complacido e interesado, por tanto, se concluye que estas dos emociones tienen gran similitud debido a que los encuestados en su mayoría escogieron estas respuestas.

g) Séptima Pregunta:



Figura 3.22: Resultados Séptima Pregunta.

Emociones	Porcentaje de respuesta de los encuestados de la emoción predominante (Pregunta 7)	Personas que contestaron la pregunta 7
Complacido	22.7 %	5
Orientador	18.2 %	4
Estricto	0 %	0
Exhortador	0 %	0
Insatisfecho	0 %	0
Dudoso	22.7 %	5
Interesado	18.2 %	4
Alentador	9.1 %	2
Ninguna de las Anteriores	0 %	0

Cuadro 3.7: Resultados Pregunta 7

Los resultados de la figura 3.22 y a la tabla 3.7 muestran que las emociones predominantes en la séptima pregunta fueron: complacido y dudoso.

En la séptima pregunta encontramos que la emoción diseñada teóricamente fue para: Interesado.

El error de Interesado es:

$$Error\ Interesado = \frac{22 - 4}{22} = 0,82 = 82\% \quad (3.14)$$

En conclusión, a esta pregunta la emoción de Interesado no fue percibida por los encuestados, adicionalmente hubo amplia dispersión en las respuestas al no presentarse ninguna emoción predominante.

h) Octava Pregunta:

Pregunta: En la demostración 8 de Baxter cual considera que es la emoción predominante.

228 respuestas

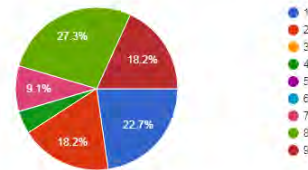


Figura 3.23: Resultados Octava Pregunta.

Emociones	Porcentaje de respuesta de los encuestados de la emoción predominante (Pregunta 8)	Personas que contestaron la pregunta 8
Complacido	22.7 %	5
Orientador	18.2 %	4
Estricto	0 %	0
Exhortador	0 %	0
Insatisfecho	0 %	0
Dudoso	0 %	0
Interesado	9.1 %	2
Alentador	27.3 %	6
Ninguna de las Anteriores	18.2 %	4

Cuadro 3.8: Resultados Pregunta 8

Los resultados de la figura 3.23 y a la tabla 3.8 muestran que la emoción predominante en la octava pregunta fue: Alentador.

En la octava pregunta encontramos que la emoción diseñada teóricamente fue para: Alentador.

El error de Alentador es:

$$Error\ Alentador = \frac{22 - 6}{22} = 0,72 = 72\% \quad (3.15)$$

La emoción alentadora fue la más percibida en la octava pregunta; no obstante, hubo amplia dispersión en las respuestas y en esta pregunta muchos encuestados respondieron ninguna de las anteriores haciendo que el porcentaje de error fuera elevado.

Podemos concluir que las emociones son subjetivas y dependen de la vivencia y el ambiente en que se desarrolla el ser humano; las emociones cambian a medida que se avancen los procesos, muchas veces comienza en el temor y/o curiosidad hasta llegar a una interacción completa con el recurso, en este caso el robot Baxter.

Capítulo 4

Funciones de percepción

4.1. Introducción

El capítulo 4 describe la implementación y diseño de las funciones de percepción del robot Baxter mediante el uso de la cámara de profundidad. Para este desarrollo se utilizó un Kinect Xbox 360. Inicia con el proceso de calibración que desarrolla la cámara para poder distinguir los colores en los cubos que serán usados en los juegos de interacción con el usuario. Después se describen detalladamente las funciones implementadas con el proceso de calibración mediante el diseño de nuevos nodos, tópicos y servicios. Este capítulo permite comprender el proceso de detección de colores de objetos en un ambiente controlado.

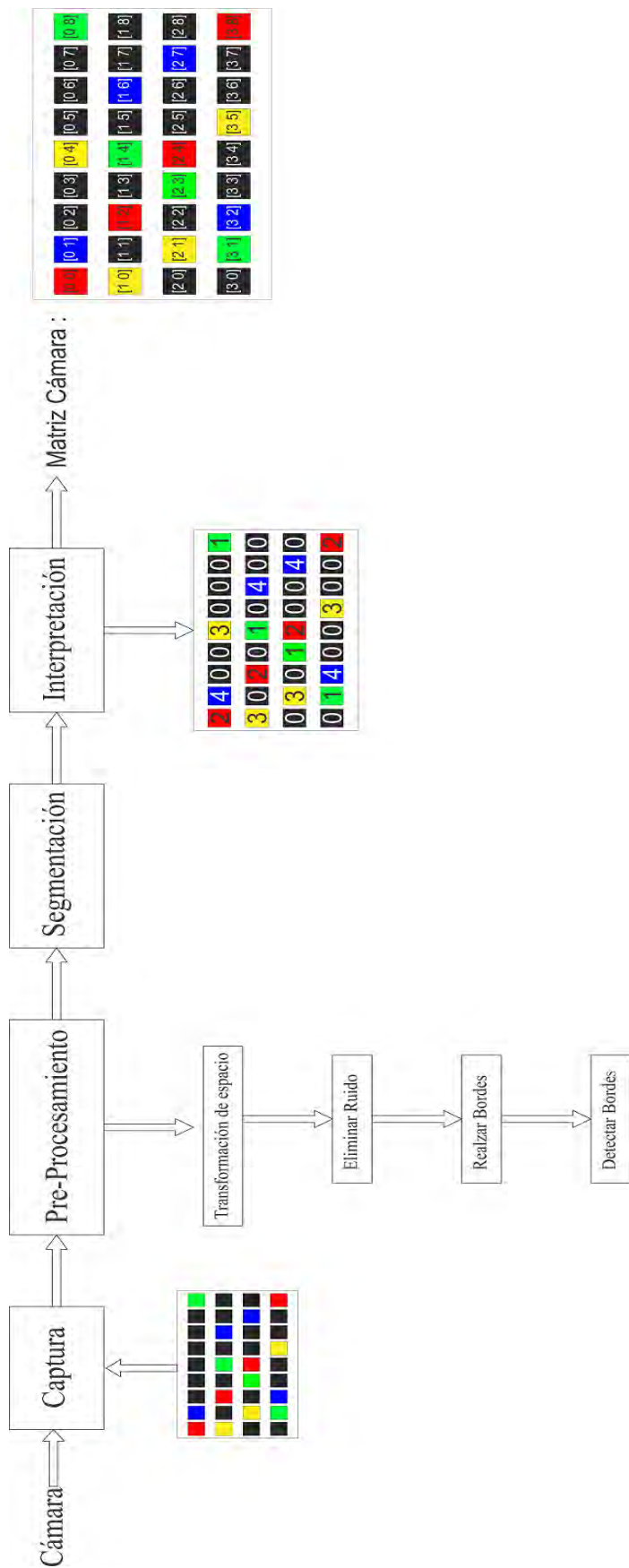


Figura 4.1: Diagrama de Bloques Visión Artificial para el reconocimiento de colores

La figura 4.1, nos muestra el diagrama de bloques de la visión Artificial del robot mediante la cámara de marca kinect, este diagrama de bloques nos permite identificar los colores que están en el rango de vista de la cámara.

4.2. Captura

El bloque de captura es el primer paso para el proceso de identificación de colores. La cámara capturará un tablero, dentro de este se ubicarán cubos de diferentes colores; con estos cubos podremos desarrollar los juegos de interacción con el usuario. Para el proceso de reconocimiento de colores es necesario ubicar los cubos como aparecen en la figura4.2.



Figura 4.2: Captura de la Cámara con visión enfocada en los cubos y espacios vacíos ubicados en el tablero

Los cubos fueron repartidos por todo el espacio para obtener la variación del rango del color de cada cubo, esto se debe a que la iluminación varía por todo el tablero, por tanto, es necesario tener en el proceso de calibración los cubos repartidos por todo el tablero. En el proceso de iluminación no se tenía en cuenta el efecto del reflejo de la luz, lo importante en este proceso es que durante el proceso de calibración y durante el desarrollo de pruebas se debe mantener la misma intensidad de luz.

El proceso de calibración que desarrollamos fue después de realizar pruebas de calibración manual, es decir, se ajustaban los parámetros de cada color por ensayo y error. No obstante, este proceso requería mucho tiempo y no se tenía en cuenta el proceso de automatización. Por tanto, el modelo de calibración escogido se acopla a las condiciones del diseño de automatización y clasificador de colores.

Los espacios en negro que están dentro del tablero (Ver figura 4.2) tienen que ser identificados al igual que los cubos, esto se debe a que el sistema requiere tener un seguimiento de los cubos dentro del tablero.

Después de capturar la imagen con el tablero y los cubos ubicados alrededor de este, se etiquetaron puntos de referencia en el centro de los cubos y en el centro de los espacios vacíos dentro del tablero de forma manual. La figura 4.3 muestra la captura de la imagen con los puntos de referencia dentro del tablero.

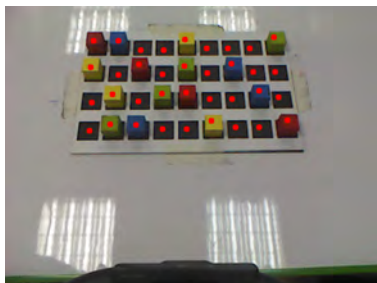


Figura 4.3: Captura de la Cámara con los puntos de referencia identificando los cubos y espacios vacíos dentro del tablero

Con los puntos de referencia y los cubos puestos siempre en la posición de la figura 4.3, podemos identificar el color al cual pertenece el cubo o si es un espacio vacío; el siguiente código muestra como fueron guardados los 36 puntos del tablero.

Algorithm 7 Variables de las casillas del tablero

```

...
..
.
x_red1 , y_red1=centro (150 ,60 , frame)
x_blue1 , y_blue1=centro (192 ,60 , frame)
x_obs1 , y_obs1=centro (230 ,76 , frame)
x_obs2 , y_obs2=centro (269 ,76 , frame)
x_orange1 , y_orange1=centro (305 ,58 , frame)
x_obs3 , y_obs3=centro (342 ,73 , frame)
x_obs4 , y_obs4=centro (378 ,73 , frame)
x_obs5 , y_obs5=centro (415 ,73 , frame)
x_green1 , y_green1=centro (456 ,54 , frame)
x_orange2 , y_orange2=centro (145 ,105 , frame)
x_obs6 , y_obs6=centro (192 ,118 , frame)
x_red2 , y_red2=centro (227 ,108 , frame)
x_obs7 , y_obs7=centro (267 ,117 , frame)
x_green2 , y_green2=centro (304 ,103 , frame)
x_obs8 , y_obs8=centro (343 ,115 , frame)
x_blue2 , y_blue2=centro (384 ,99 , frame)
x_obs9 , y_obs9=centro (421 ,113 , frame)
x_obs10 , y_obs10=centro (458 ,113 , frame)
x_obs11 , y_obs11=centro (147 ,165 , frame)
x_orange3 , y_orange3=centro (182 ,151 , frame)
x_obs12 , y_obs12=centro (226 ,163 , frame)
x_green3 , y_green3=centro (266 ,150 , frame)
x_red3 , y_red3=centro (307 ,148 , frame)
x_obs13 , y_obs13=centro (347 ,159 , frame)
x_obs14 , y_obs14=centro (385 ,159 , frame)
x_blue3 , y_blue3=centro (431 ,145 , frame)
x_obs15 , y_obs15=centro (464 ,156 , frame)
x_obs16 , y_obs16=centro (143 ,213 , frame)
x_green4 , y_green4=centro (178 ,202 , frame)
x_blue4 , y_blue4=centro (222 ,203 , frame)
x_obs17 , y_obs17=centro (267 ,210 , frame)
x_obs18 , y_obs18=centro (308 ,210 , frame)
x_orange4 , y_orange4=centro (350 ,196 , frame)
x_obs19 , y_obs19=centro (391 ,207 , frame)
x_obs20 , y_obs20=centro (431 ,207 , frame)
x_red4 , y_red4=centro (479 ,195 , frame)
...
..
.

```

Los ejes de coordenadas X, Y que se guardan en todas las variables del código anterior, son los puntos de referencia que están de color rojo (Ver figura 4.3). Para el proceso de identificación de los colores es necesario tener siempre la misma organización de los cubos dentro del tablero. Como se puede observar en el pseudocódigo 7, las variables se almacenan con su punto de referencia, pero el nombre de estas variables se nombra dependiendo del nombre del color del cubo o los espacios negros; por ejemplo, el primer cubo en el extremo superior izquierdo de la figura 4.3 es un cubo rojo, por tanto la variable que corresponde a este

punto es:

```
x_red1 , y_red1=centro (150 ,60 , frame )
```

Los puntos (150,60) son el eje de coordenadas en el que está ubicado el cubo rojo con respecto a la visión de la cámara, estos valores se establecieron manualmente para tener el centro de la cara superior del cubo.

4.3. Pre-Procesamiento

El Pre-Procesamiento de imágenes, es el proceso que permite obtener información de la captura de imágenes. El Pre-procesamiento de imágenes se divide en 4 pasos: Transformación de espacio, eliminar ruido, realzar bordes y detectar bordes.

1. **Transformación de espacio:** En el proceso de detectar colores es necesario cambiar el formato de la imagen, el formato de la captura de las imágenes está por defecto en el formato RGB; este modelo está representado por coordenadas euclidianas; no obstante, el modelo HSV está representado en coordenadas cilíndricas, este modelo representa mejor la clasificación de imágenes por colores, esto se debe a que los humanos perciben los colores como el modelo HSV.

La siguiente ecuación representa el cambio de coordenadas del modelo RGB al modelo HSV [3].

$$V = \max(R, G, B) \quad (4.1)$$

$$S = \begin{cases} \frac{V - \min(R, G, B)}{V} & \text{if } V \neq 0 \\ 0 & \text{de otra manera} \end{cases} \quad (4.2)$$

$$H = \begin{cases} \frac{60(G-B)}{V - \min(R, G, B)} & \text{if } V = R \\ \frac{120 + 60(B-R)}{V - \min(R, G, B)} & \text{if } V = G \\ \frac{240 + 60(R-G)}{V - \min(R, G, B)} & \text{if } V = B \end{cases} \quad (4.3)$$

La figura 4.4 es la captura de la imagen en el modelo HSV, esta figura es el primer paso en el proceso de clasificar colores.

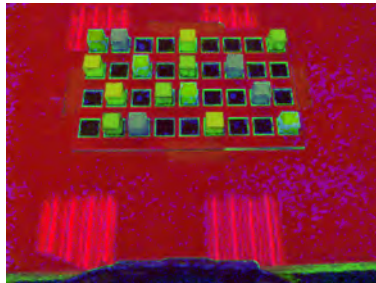


Figura 4.4: Captura de cámara en modelo HSV

2. **Eliminar ruido:** Para el proceso de eliminar el ruido para detectar los colores dentro del tablero, es necesario que en el modelo de HSV obtengamos los valores máximos y mínimos de H, S y V. Para detectar estos valores son necesarios los puntos de referencia de la figura 4.3. El siguiente algoritmo define como se encontraron los valores máximos y mínimos de HSV.

Algorithm 8 Pre-Procesamiento de eliminar el ruido de la captura

```
crear 36 matrices de cada posición del tablero
for x=0 hasta 36 do
  for y=0 hasta 200 do
    matriz [x]= HSV[y](cada matriz guardara 200 el dato del HSV debido a que el valor cambia)
  end for
end for
for x=0 hasta 36 do
  minH=1000
  minS=1000
  minV=1000
  maxH=0
  maxS=0
  maxV=0
  for y=0 hasta 200 do
    if minH > H de HSV[y] then
      minH=H de HSV[y]
    end if
    if maxH < H de HSV[y] then
      maxH=H de HSV[y]
    end if
    if minS > S de HSV[y] then
      minS=S de HSV[y]
    end if
    if maxS < S de HSV[y] then
      maxS=S de HSV[y]
    end if
    if minV > V de HSV[y] then
      minV=V de HSV[y]
    end if
    if maxV < V de HSV[y] then
      maxV=V de HSV[y]
    end if
  end for
  máscara[x]=cv2.inRange(hsv,(minH,minS,minV),(maxH,maxS,maxV))
end for
```

En el pseudocódigo 8, se crean 36 matrices que representan cada punto del tablero, por cada matriz se guardarán 200 valores de HSV. Estos valores se toman de los puntos de referencia de la figura 4.3. Después de obtener los valores de HSV, por cada matriz se crean los valores máximos y mínimos de los 200 valores de HSV obtenidos anteriormente.

Con estos valores se obtendrán la máscara de cada punto del tablero. La máscara, se produce por la siguiente ecuación:

$$mascara = \min(H, S, V) \leq (H, S, V) \leq \max(H, S, V) \quad (4.4)$$

La máscara nos permite extraer de la captura de la imagen, el color de los valores máximos y mínimos por cada color; es decir, esta máscara obtiene el color de cada punto de referencia del tablero, y finalmente como el orden dentro del tablero quedo establecido en la figura 4.2, entonces estas máscaras finalmente se sumarán y por cada color obtendremos una máscara.

Las máscaras obtenidas se representan en la figura 4.5

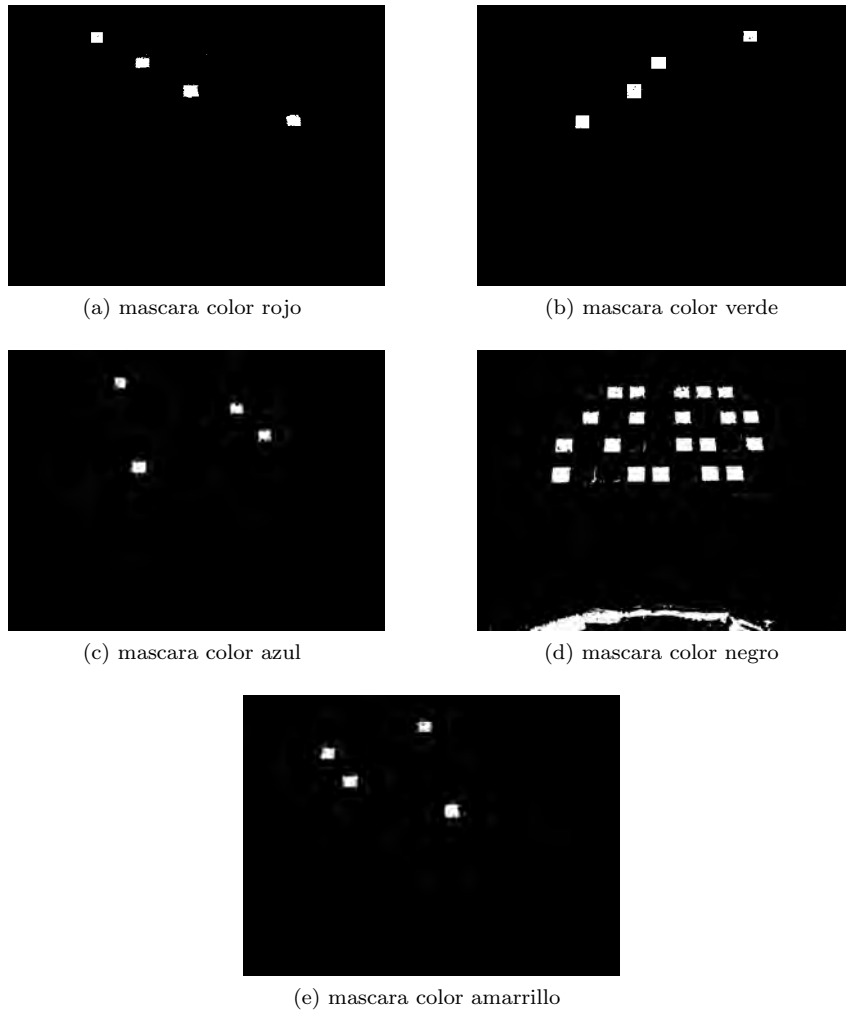


Figura 4.5: Proceso para detectar colores

Las imágenes de cada máscara representan los colores que hay en el tablero, por tanto, en el tablero hay 5 colores: rojo, verde, azul, amarillo y negro. Los primeros 4 colores representan los cubos y el negro representa los espacios vacíos que se encuentran en el tablero, como son máscaras se sustraen las imágenes en blanco y negro.

3. **Realzar Bordes:** Con las máscaras diseñadas para cada color el proceso de realzar los bordes es poder tener el contorno de la imagen por cada color, para este proceso se compara la captura original de la imagen y las máscaras, y en cada imagen obtendremos la intersección entre estas imágenes, la figura 4.6 contiene todas las imágenes de intersección de cada color.

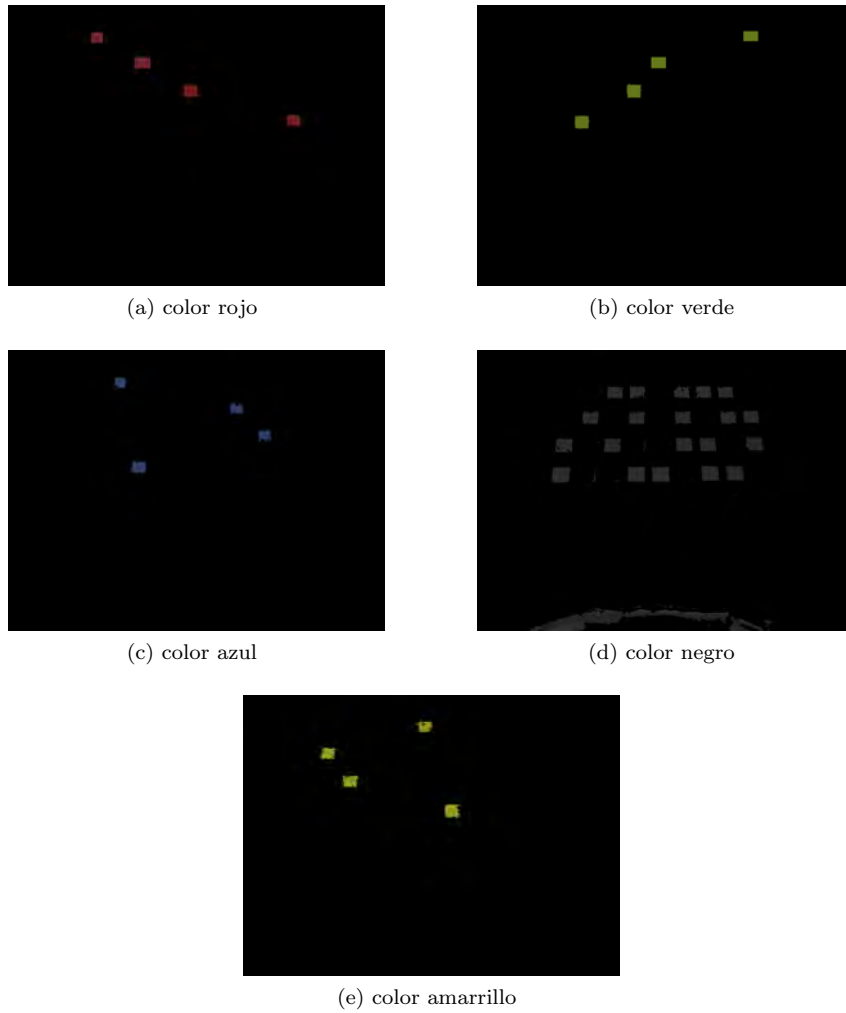


Figura 4.6: Proceso para detectar colores

Con estos valores, podemos obtener el contorno y la figura de la imagen, estos valores son los que finalmente se enviaran al bloque de segmentación, los cuales manipularan los valores del filtro de cada color.

4.4. Segmentación

Con los contornos de los 5 colores (*rojo, verde, azul, amarillo, negro*) que están dentro del tablero, se necesitan analizar las imágenes para poder obtener el centro de cada área de cada punto del tablero.

Algorithm 9 Segmentación

```
if existe contorno then
  n=cantidad de contornos
  for x=0 hasta n do
    c= contorno[x]
    área= área del contorno c
    y= punto en el eje y del contorno c
    if area > 100 and y < 387 then
      center=x,y puntos x,y del contorno
      dibujar en la coordenada center de la captura de la imagen
    end if
  end for
end if
```

El pseudocódigo 9 muestra el proceso de filtrado de la imagen, existen 5 contornos que representan los 5 colores que hay en el tablero, no obstante, para obtener el área y la ubicación de cada posición del tablero se debe realizar un ciclo por cada contorno, esto se debe a que dentro de cada contorno están todos los objetos que representan el color de dicho contorno, por tanto, el contorno se debe separar por cada objeto.

Debido a que las dimensiones de la cámara son superiores a la dimensión del tablero, se creó una variable que filtra todos los objetos que se encuentren fuera del tablero; el área es para filtrar los puntos que se pueden detectar dentro del tablero, estos puntos suelen ser un ruido interno de la cámara, para eliminar este ruido solo se pone un condicional para que solo detecte los colores del tablero, esta área es superior a 100.

Finalmente, cuando hallamos obtenido la ubicación de las coordenadas de los puntos del tablero, cada punto del tablero se pintará al color que corresponde, un ejemplo sería la primera posición del tablero que se encuentra en la parte superior izquierda, este punto tiene un cubo rojo, por tanto en la captura de la imagen se pintará un punto rojo en el centro de la ubicación de este objeto.

La figura 4.7 muestra los puntos pintados en cada punto del tablero.



Figura 4.7: Captura de la imagen con los puntos de cada punto del tablero con su respectivo color

4.5. Interpretación

Con las casillas del tablero clasificadas se crearon dos matrices, las posiciones de las dos matrices se definieron como se muestra en la figura 4.8.



Figura 4.8: Tablero con las posiciones de la matriz

La primera matriz tendrá las coordenadas de cada punto del tablero, la segunda matriz como conocemos todas posiciones y los colores de cada punto del tablero entonces se cambiará las coordenadas de cada punto por un número que representa un color, es decir, se etiqueta cada color por un número, la figura 4.9 muestra el cómo sería la etiqueta de cada posición del tablero según el color del cubo o el espacio negro.

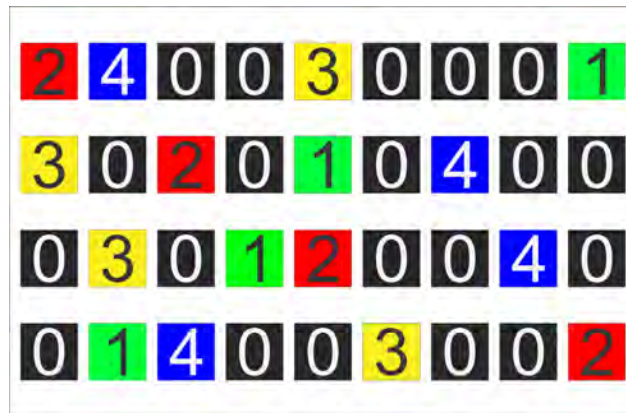


Figura 4.9: Numero al que pertenece cada color

Donde 1 es verde, 2 rojo, 3 amarillo, 4 azul y 0 negro. Estas 2 matrices son enviadas mediante un **Servicio ROS** el cual se llama **InfoCoor**. ROS envía matrices de la siguiente forma:

$$array[]^{n \times 1} \tag{4.5}$$

Debido a esto las matrices son enviadas:

Tipo de Variable	Nombre de la Variable	Descripción
Float32[]	matriz_1x	todos los puntos de eje x de la matriz de coordenadas del tablero desde la [0 0] hasta la [0 8] de la figura 4.8
Float32[]	matriz_2x	todos los puntos de eje x de la matriz de coordenadas del tablero desde la [1 0] hasta la [1 8] de la figura 4.8
Float32[]	matriz_3x	todos los puntos de eje x de la matriz de coordenadas del tablero desde la [2 0] hasta la [2 8] de la figura 4.8
Float32[]	matriz_4x	todos los puntos de eje x de la matriz de coordenadas del tablero desde la [3 0] hasta la [3 8] de la figura 4.8
Float32[]	matriz_1y	todos los puntos de eje y de la matriz de coordenadas del tablero desde la [0 0] hasta la [0 8] de la figura 4.8
Float32[]	matriz_2y	todos los puntos de eje y de la matriz de coordenadas del tablero desde la [1 0] hasta la [1 8] de la figura 4.8
Float32[]	matriz_3y	todos los puntos de eje y de la matriz de coordenadas del tablero desde la [2 0] hasta la [2 8] de la figura 4.8
Float32[]	matriz_4y	todos los puntos de eje y de la matriz de coordenadas del tablero desde la [3 0] hasta la [3 8] de la figura 4.8
Int32[]	matriz_5	matriz con la etiqueta de colores desde la [0 0] hasta la [0 8] de la figura 4.8
Int32[]	matriz_6	matriz con la etiqueta de colores desde la [1 0] hasta la [1 8] de la figura 4.8
Int32[]	matriz_7	matriz con la etiqueta de colores desde la [2 0] hasta la [2 8] de la figura 4.8
Int32[]	matriz_8	matriz con la etiqueta de colores desde la [3 0] hasta la [3 8] de la figura 4.8

Cuadro 4.1: Contenido del servicio **Infocoor** que contiene la información de las matriz para ser enviadas a otros nodos

La primera matriz es necesaria para los brazos del robot Baxter y la segunda matriz es necesario para el seguimiento de los movimientos del usuario.

4.6. Descripción detallada funciones de percepción

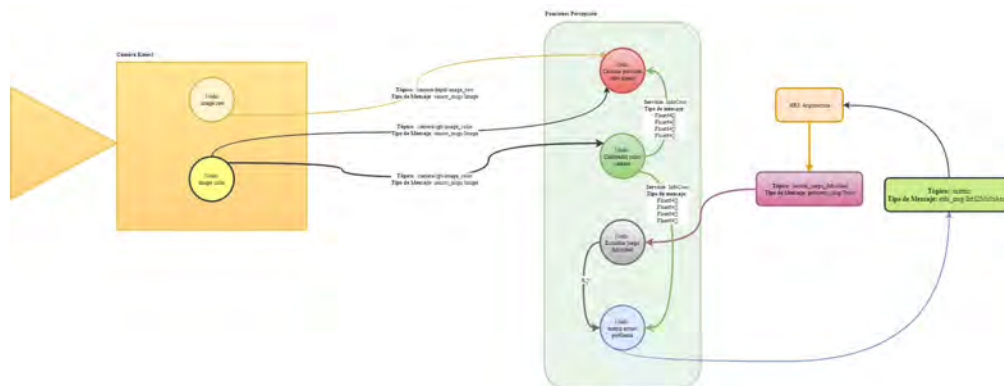


Figura 4.10: Diagrama de bloques de las funciones de percepción

La figura 4.10 representa las funciones de percepción, estas funciones tienen como objetivo implementar la visión artificial del robot Baxter.

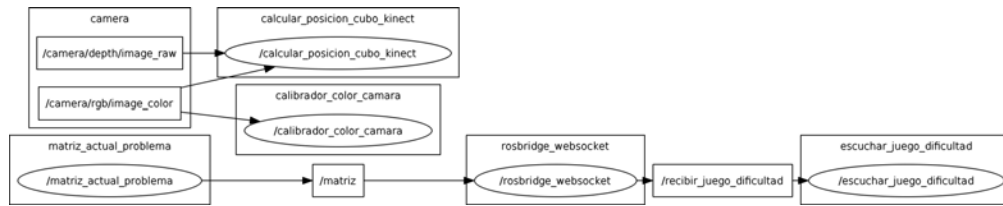


Figura 4.11: Diagrama detallado de los nodos y tópicos en ROS

La figura 4.11 es el diagrama detallado del funcionamiento de las funciones de percepción, el bloque que se llama *rosbridge websocket* es el bloque que manda la información al bloque HRS de la figura 4.10.

4.6.1. Bloques Función de Percepción

Los nodos que se desarrollarán para implementar la visión artificial del robot fueron: Calcular posición cubo Kinect, Calibrador color cámara, Escuchar juego dificultad y matriz actual del problema.

1. **Nodo Calibrador color cámara:** Este nodo está en la explicación desde la sección 4.2 hasta la sección 4.5. Tiene como parámetro de entrada la cámara del Kinect y como parámetro de salida 2 matrices.
2. **Nodo Calcular posición cubo Kinect:** Permite determinar la profundidad del cubo con respecto a la cámara del Kinect, para poder determinar la posición de los cubos en metros. Esta función es necesaria para el movimiento de los brazos del robot Baxter.

Los parámetros de entrada son las casillas del tablero (si tiene un cubo en la casilla o espacio vacío) que se envían del calibrador de color, y los otros parámetros de entrada son la imagen y la profundidad del Kinect.

La primera matriz del nodo Calibrador color cámara contiene las coordenadas de todos los puntos del tablero de la captura de la imagen, en el servicio InfoCoor solamente se manipularán los datos correspondientes a las variables float 64.

La cámara Kinect tiene implementado además un sensor infrarrojo, este sensor nos permite obtener la profundidad de los objetos, y además obtendremos las coordenadas de todos los puntos del tablero, pero su unidad será ahora en metros, debido a las coordenadas que recibimos del servicio originalmente tienen sus unidades en píxeles.

El pseudocódigo 10 representa el nodo Posición Cubos Tablero.

Algorithm 10 nodo Calcular posición cubo Kinect

```

recibir servicio infoCoor(todos los float64[]) de Calibrador color cámara
Crear Variable que representa la visualización del Kinect
Crear Variable que representa la profundidad del Kinect
for 0 hasta longitud matriz servicio infoCoor(todos los float64[]) do
    profundidad = pixel de las coordenadas de la posición del objeto
    x metros tablero[] = -1*(x pixel*0.00026458*profundidad/1000)
    y metros tablero[] = -1*(y pixel*0.00026458*profundidad/1000)
end for
Enviar servicio con los arreglos de x metros tablero, y metros tablero

```

En cada punto del tablero tendremos una variable de profundidad, la cual nos permitirá obtener el cambio de variables para poder cambiar las unidades de las coordenadas de los puntos del tablero.

Con la variable de profundidad de cada punto del tablero y sabiendo que 1 píxel equivale a 0.00026458 metros. entonces para todos los puntos del tablero se usó la siguiente ecuación:

$$x_{\text{coordenadametros}} = \frac{x_{\text{píxeles}} \cdot 0,00026458 \cdot \text{profundidad}}{1000} \quad (4.6)$$

$$y_{\text{coordenadametros}} = \frac{y_{\text{píxeles}} \cdot 0,00026458 \cdot \text{profundidad}}{1000} \quad (4.7)$$

Las ecuaciones 4.6 y 4.7 son el cambio de coordenadas que nos permiten pasar de píxeles a metros, esto se debe a que las unidades que representan el movimiento del robot Baxter son metros, por tanto, era necesario este cambio.

Con los cambios realizados se creará una matriz con las coordenadas de los puntos del tablero pero con sus unidades en metros, la organización de la matriz es como el de la figura 4.8.

Finalmente se enviará un servicio llamado *Coordinates*, el cual será enviado de la siguiente forma:

Tipo de Variable	Nombre de la Variable	Descripción
Float32[]	xc1	todos los puntos de eje x de la matriz de coordenadas del tablero desde la [0 0] hasta la [0 8] de la figura 4.8
Float32[]	xc2	todos los puntos de eje x de la matriz de coordenadas del tablero desde la [1 0] hasta la [1 8] de la figura 4.8
Float32[]	xc3	todos los puntos de eje x de la matriz de coordenadas del tablero desde la [2 0] hasta la [2 8] de la figura 4.8
Float32[]	xc4	todos los puntos de eje x de la matriz de coordenadas del tablero desde la [3 0] hasta la [3 8] de la figura 4.8
Float32[]	yc1	todos los puntos de eje y de la matriz de coordenadas del tablero desde la [0 0] hasta la [0 8] de la figura 4.8
Float32[]	yc2	todos los puntos de eje y de la matriz de coordenadas del tablero desde la [1 0] hasta la [1 8] de la figura 4.8
Float32[]	yc3	todos los puntos de eje y de la matriz de coordenadas del tablero desde la [2 0] hasta la [2 8] de la figura 4.8
Float32[]	yc4	todos los puntos de eje y de la matriz de coordenadas del tablero desde la [3 0] hasta la [3 8] de la figura 4.8

Cuadro 4.2: Contenido del servicio **Coordinates** que contiene la información de las matriz para ser enviadas a otros nodos

3. **Nodo Escuchar juego dificultad:** Es el nodo que escucha cual es el problema que se va a desarrollar con el usuario, este bloque es dependiente de la información recibida del bloque HRS, la información obtenida será que tipo de juego será y cual será su dificultad, los juegos serán los siguientes:

$$\text{Juego : } \begin{cases} 1 = \text{Escalera} \\ 2 = \text{Apilar Bloques} \\ 3 = \text{Juego de los caballos} \end{cases}$$

La dificultad de la escalera es:

$$\text{Escalera : } \begin{cases} 1 = 2 \text{ cubos rojos y } 2 \text{ cubos verdes} \\ 2 = 3 \text{ cubos rojos y } 3 \text{ cubos verdes} \\ 3 = 4 \text{ cubos rojos y } 4 \text{ cubos verdes} \end{cases}$$

Apilar bloques:

$$\text{Apilar bloques : } \begin{cases} 1 = 2 \text{ cubos del mismo color} \\ 2 = 4 \text{ cubos , pero 2 del mismo color} \\ 3 = 6 \text{ cubos , pero 2 del mismo color} \\ 4 = 8 \text{ cubos , pero 2 del mismo color} \end{cases}$$

Juego de los caballos:

$$\text{Juego de los caballos : } \begin{cases} 1 = 4 \text{ cubos , pero 2 del mismo color} \\ 2 = 8 \text{ cubos , pero 4 del mismo color} \end{cases}$$

4. **Nodo matriz actual del problema:** Es el nodo que se encarga de indicar los cambios de los puntos del tablero, es decir, si se mueve un cubo que estaba ubicado dentro del tablero a un espacio vacío dentro de este, entonces este nodo debe notificar el cambio de estado. Los parámetros de entrada son: Juego y dificultad que son enviados del nodo Escuchar juego dificultad y la tercera entrada es la matriz del servicio *infoCoor* con las variables `int32[]`, las cuales contienen las etiquetas de las posiciones del tablero, esta matriz está organizada como la de la figura 4.8. Como parámetro de salida es una matriz, la cual solo se mandará mediante dos condiciones: la primera que la cantidad y color de cubos correspondan con el juego y dificultad del nodo Escuchar Juego y dificultad. La segunda condición es el cuándo un cubo cambia de posición en el tablero, un ejemplo de la matriz que se envía es:

$$Matriz\ tablero : \begin{cases} 2, 4, 0, 0, 3, 0, 0, 0, 1 \\ 3, 0, 2, 0, 1, 0, 4, 0, 0 \\ 0, 3, 0, 1, 2, 0, 0, 4, 0 \\ 0, 1, 4, 0, 0, 3, 0, 0, 2 \end{cases}$$

Donde 0 es negro, 1 verde, 2 rojo, 3 amarillo y 4 azul.

Dependiendo del juego y dificultad del problema, el nodo matriz actual del problema enviara una matriz con las etiquetas de los bloques que observa el Kinect al bloque HRS.

El pseudocódigo 12 representa la función del nodo matriz actual del problema.

Algorithm 11 nodo matriz actual del problema

recibir servicio 2 de Calibrador color cámara

while exista juego **and** dificultad **do**

if (juego **and** dificultad) = servicio *infoCoor*(todos los `int32[]`) **then**

if primera vez que cumple condición **then**

 publicar matriz servicio *infoCoor*(todos los `int32[]`)

else if Algún cubo del tablero cambia de posición **then**

 publicar matriz servicio *infoCoor*(todos los `int32[]`)

else no publicar nada

end if

else no publicar nada

end if

end while

Un ejemplo para que juego y dificultad sea igual a la matriz con la información del tablero seria si juego fuera 1 y dificultad 3, los cuales representan: juego escalera, dificultad 4 cubos rojos y 4 cubos verdes. La matriz que cumpla esta condición puede ser la siguiente:

$$Matriz\ tablero : \begin{cases} 1, 1, 1, 1, 0, 2, 2, 2, 2 \\ 0, 0, 0, 0, 0, 0, 0, 0, 0 \\ 0, 0, 0, 0, 0, 0, 0, 0, 0 \\ 0, 0, 0, 0, 0, 0, 0, 0, 0 \end{cases}$$

Pero esta no es la única condición para que la matriz y esta dificultad se cumplan, lo único que importa para este juego y dificultad es que en el tablero hallan 4 cubos verdes (1) y 4 cubos rojos (2).

Al cumplirse por primera vez esta condición publicara un mensaje con la posición del tablero que ve actualmente el Kinect; sin embargo, para evitar mandar siempre la misma matriz lo que se desarrolló fue que cuando el usuario cambie el cubo de posición se vuelva a enviar el mensaje, pero informando que el cubo cambio de posición es decir si tenemos el tablero de esta manera:

$$Matriz\ tablero : \begin{cases} 1, 1, 1, 1, 0, 2, 2, 2, 2 \\ 0, 0, 0, 0, 0, 0, 0, 0, 0 \\ 0, 0, 0, 0, 0, 0, 0, 0, 0 \\ 0, 0, 0, 0, 0, 0, 0, 0, 0 \end{cases}$$

Como este capítulo son las funciones de percepción y la finalidad es obtener el resultado de los cambios de estados de lo que observa la captura de la cámara este paso a paso solo mencionara la segunda matriz que envía el detector de colores.

2. **Nodo matriz actual del Problema:** Cuando se seleccione el nivel y dificultad del problema se enviara un mensaje de tipo Twist al nodo Escuchar juego dificultad y este enviara el mensaje al Nodo matriz actual del problema con el juego y dificultad, una vez recibidos entonces este nodo recibirá el servicio *InfoCoor* y como solo es necesario la matriz que contenga las etiquetas de los puntos del tablero, entonces solo usaremos las variables *int32[]*, con estas variables tendremos en tiempo real los puntos del tablero organizados por su color de la captura de la cámara del Kinect y si el usuario cambia la posición del cubo de lugar enviara una matriz con las etiquetas de los puntos del tablero con el cambio de posición, este mensaje se enviara mediante un tópic llamado matriz, el cual tiene tipo de mensaje *int32 multiarray* y que sirve para enviar la matriz con los 36 puntos del tablero, donde finalmente el bloque HRS recibirá la información con el cambio de estado del tablero debido al cambio de posición de alguno de los cubos

4.7. Validación de resultados

Para validar las funciones de percepción, fue comprobar el recorrido de un cubo por todo el tablero, el recorrido fue el de la figura 4.12.

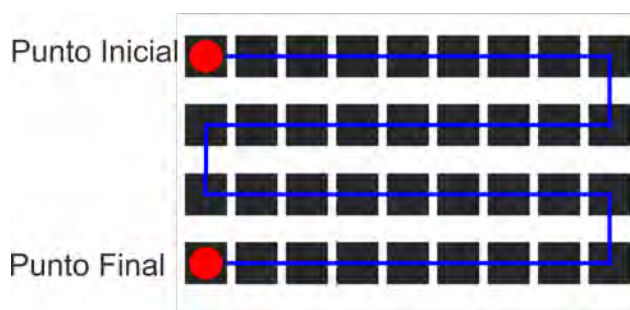


Figura 4.12: Recorrido del cubo por el tablero

El modelo Teórico que se desarrolló para validar las funciones de percepción es el recorrido de la figura 4.12, este recorrido fue desarrollado en la vida real, el movimiento del cubo para cambiar de posición fue hecho por los brazos del robot Baxter. El cubo de color verde fue el primer objeto en recorrer todas las posiciones del tablero, después fue el cubo rojo, amarillo y azul.

El cuadro 4.3, representa el color del cubo el número de percepciones exitosas, es decir, el número de veces que la cámara captó el cambio de cubo del tablero como el modelo teórico, y por último el número de percepciones fallidas.

Color Cubo	Percepciones Exitosas	Percepciones fallidas
Verde (1)	31	5
Rojo (2)	35	1
Amarillo (3)	35	1
Azul (4)	36	0

Cuadro 4.3: Resultados Percepción por cada color de cubo

Las percepciones exitosas se representarán de color verde en la casilla del tablero, esto significa que si la percepción de la cámara es igual a el modelo teórico (figura 4.12) entonces la casilla tendrá un color verde; sin

embargo si la percepción es fallida, es decir, que la percepción de la cámara fue diferente al modelo teórico, por tanto la casilla tendrá un color rojo.

Al tener la cuadrícula obtendremos por cada color del cubo su error de percepción, es decir, el rango de error de cada color de cubo por cada casilla del tablero, la ecuación es la siguiente:

$$Error\ Percepción\ Cubo = \frac{Modelo_{Teórico} - Modelo_{Experimental}}{Modelo_{Teórico}} \quad (4.9)$$

Las cuadrículas que representan el tablero con sus percepciones exitosas y fallidas para cada cubo con su Error de percepción son:

1. **Cubo Verde (1):**

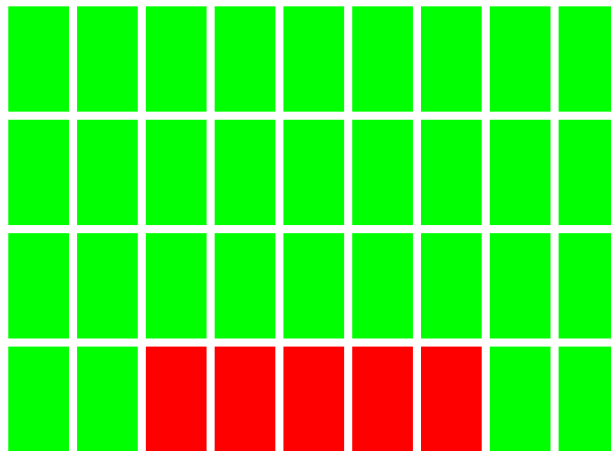


Figura 4.13: Cuadrícula que representa el tablero y compara los resultados teóricos con los resultados experimentales del Cubo Verde

La figura 4.13, muestra que la matriz se envió mal 5 veces para el cubo verde en la parte inferior del tablero.

Al reemplazar los datos de la ecuación 4.9 por los datos obtenidos del modelo teórico y experimental tenemos que el error de Percepción del cubo verde es:

$$Error\ Percepción\ Cubo\ Verde = \frac{36 - 31}{36} = 0,1388 = 13,8\% \quad (4.10)$$

Por tanto el error de percepción del cubo es del 13%, no obstante, este error se produce sobre todo en la parte inferior del tablero, debido a esto en la manipulación de los cubos es preferible usar la parte superior del tablero para no tener un error en la toma de datos.

2. **Cubo Rojo (2):**

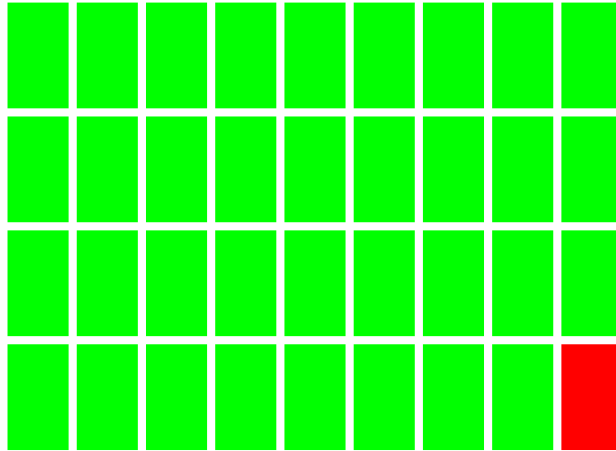


Figura 4.14: Cuadrícula que representa el tablero y compara los resultados teóricos con los resultados experimentales del Cubo Rojo

La figura 4.14, muestra que la matriz solo obtuvo un error en el extremo inferior derecho del tablero. Reemplazando los valores de la ecuación 4.9, el error de percepción del cubo rojo es:

$$Error\ Percepción\ Cubo\ Rojo = \frac{36 - 35}{36} = 0,0277 = 2,77\% \quad (4.11)$$

El cubo rojo solo obtuvo un error en la parte inferior derecha del tablero, este error equivale al 2%. El cubo de color rojo tiene menor rango de percepción que el cubo verde, por tanto el rango tiene un mejor rango de manipulación sobre las casillas del tablero.

3. Cubo Amarillo (3):

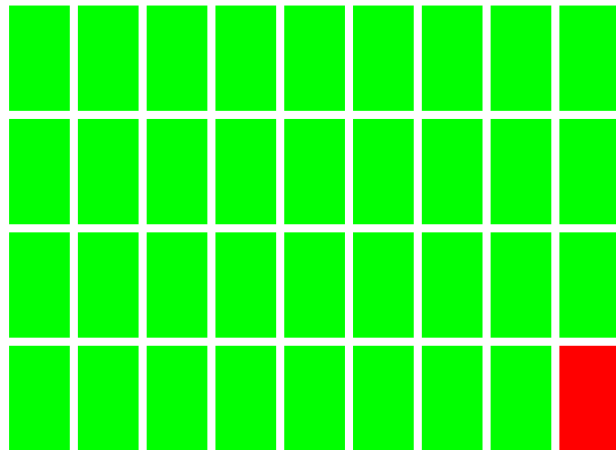


Figura 4.15: Cuadrícula que representa el tablero y compara los resultados teóricos con los resultados experimentales del Cubo Amarillo

La figura 4.15, muestra que la matriz solo obtuvo un error en el extremo inferior derecho del tablero. Reemplazando los valores de la ecuación 4.9, el error de percepción del cubo amarillo es:

$$Error\ Percepción\ Cubo\ Amarillo = \frac{36 - 35}{36} = 0,0277 = 2,77\% \quad (4.12)$$

Las cuadrículas resultantes del cubo rojo y el cubo amarillo son las mismas, por tanto, el error de percepción del cubo amarillo al igual que el cubo rojo es del 2.77%, lo que significa que la percepción de la cámara

en el extremo derecho inferior del tablero no corresponde al modelo teórico, por tanto, para los cubos verde, rojo y amarillo, no es conveniente utilizar esa posición. En los problemas que se desarrollaran con el usuario esa posición es la de menor uso, esto no solo se debe al error presentado por los cubos, también se debe a que el movimiento de los brazos del robot Baxter en la parte inferior del tablero dificulta más en esa área de trabajo, por tanto, el área de trabajo será en su mayoría en la parte superior del tablero.

4. **Cubo Azul (4):** El resultado experimental del cubo azul recorriendo todo tablero es el siguiente:

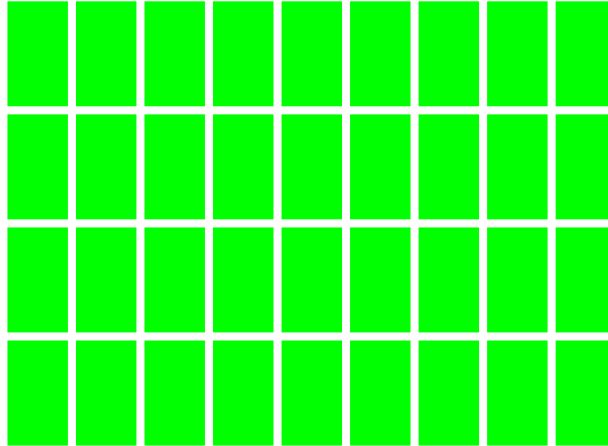


Figura 4.16: Cuadrícula que representa el tablero y compara los resultados teóricos con los resultados experimentales del Cubo Azul

La figura 4.15, muestra que la matriz no obtuvo error en recorrer el tablero, por tanto, se envió correctamente todas las posiciones del tablero en comparación al modelo teórico.

Reemplazando los valores de la ecuación 4.9, el error de percepción del cubo azul es:

$$Error\ Percepción\ Cubo\ Azul = \frac{36 - 36}{36} = 0 \tag{4.13}$$

El error de percepción del cubo azul fue 0%, esto significa que cuando tengamos un cubo azul en el tablero y lo cambiemos de casilla, entonces la matriz siempre será enviada correctamente, es decir, sus resultados serán confiables.

4.8. Conclusiones

1. Para obtener el menor error de percepción para cada color del cubo no se debe cambiar el sistema de iluminación, es decir, si las luces se apagan o disminuye la iluminación de las lámparas, el calibrador deja de funcionar, debido a que en el sistema HSV depende del control de iluminación en el ambiente, por tanto, si varía el rango de iluminación entonces la calibración no serviría y no se detectarían los colores.
2. Como conclusión para detectar los colores es necesario que el tablero no tenga efecto de sombra ya sea por disminución de luz o porque el usuario genera una sombra enfrente del tablero, esto genera un cambio en la calibración enorme, por ejemplo, si se genera sombra sobre un color rojo, entonces la cámara percibirá color negro porque la sombra que se genera hace que la cámara detecta es el color de la sombra que el color del cubo.

Capítulo 5

Funciones de movimiento

5.1. Introducción

Este capítulo describe el diseño e implementación de las funciones de movimiento de los brazos del robot Baxter. El capítulo comienza por una descripción física del robot Baxter, después tendrá la descripción detallada del movimiento de los brazos del robot, los cuales están definidos en 3 partes: calibración de las coordenadas de la percepción de la cámara con las coordenadas del robot Baxter, Posicionamiento que describe el procedimiento de cómo se movería el robot Baxter y cuál es la función que describe su movimiento, y finalmente una descripción detallada de los nodos que formaron parte del proceso de implementación de las funciones de movimiento.

5.2. Descripción Robot Baxter

El robot Baxter es un robot antropomórfico, el cual posee 2 brazos, cada brazo está compuesto por 7 motores los cuales están ubicados como la figura 5.1, estos motores representan los ángulos de los motores.



Figura 5.1: Ubicación de los 7 motores representados en ángulos de uno de los brazos del robot Baxter [20]

Además de poseer 7 motores en representación de los brazos de una persona, tiene en cada brazo un gripper, el cual simula una mano. Estos grippers están diseñados para poder manipular objetos sobre su espacio; no obstante, las dimensiones de este objeto no deben pasar de los 5cm de largo y ancho, debido a que las manos del gripper serian menor que el área de los objetos a atrapar.

En el tronco del robot podemos encontrar un ventilador, el cual es utilizado en las funciones de expresión emocional del capítulo 3.

La parte superior del robot se encuentra una tableta, la cual simula la cara del robot y al crear y editar gran variedad de imágenes (mínimo 400) se puede crear emociones al robot Baxter.

Para este capítulo de funciones de movimiento las partes más importantes del robot Baxter son los brazos, esto se debe a que este capítulo describe el movimiento de los brazos del robot Baxter entorno a los cubos que están ubicados en las casillas del tablero.

Existen 5 nodos implementados en ROS que representan el movimiento del robot Baxter: left gripper, right gripper, right limb, left limb y state robot.

1. El nodo left gripper es el nodo que permite calibrar, abrir o cerrar el gripper de la mano izquierda, envía al nodo Baxter pick and place 2 mensajes los cuales sirven para brindar información sobre el estado y las propiedades del gripper y sus datos de entrada son los mensajes que permiten el accionamiento de este gripper para su movilidad.
2. El nodo right gripper es el nodo que permite calibrar, abrir o cerrar el gripper de la mano derecha, sus parámetros de salida al nodo Baxter pick and place son la información del estado y las propiedades que tiene el gripper actualmente; por otro lado, sus parámetros de entrada son los mensajes que permiten el accionamiento de los actuadores del gripper.
3. El nodo right limb es el nodo que permite el movimiento del brazo derecho del robot, este nodo permite activar los 7 ángulos que contienen los 7 actuadores como la siguiente gráfica:
Los parámetros de entrada que recibe este nodo para activar los actuadores son la posición en la que se deben mover los motores, el tiempo que transcurre y la velocidad a la que deben andar los motores. Como parámetro de salida este nodo envía un mensaje al nodo Baxter pick and place con la información del estado actual de los brazos, es decir, la información de cada uno de los ángulos de las variables de la figura 5.1.
4. El nodo left limb es el nodo que activa el movimiento del brazo izquierdo del robot, su funcionamiento es igual que el nodo right limb.
5. El nodo state robot es el nodo que contiene la información básica perteneciente al robot, este nodo activa al robot Baxter y envía como parámetro de salida información del estado del robot y el estado de los brazos del robot.

5.3. Descripción detallada funciones de movimiento

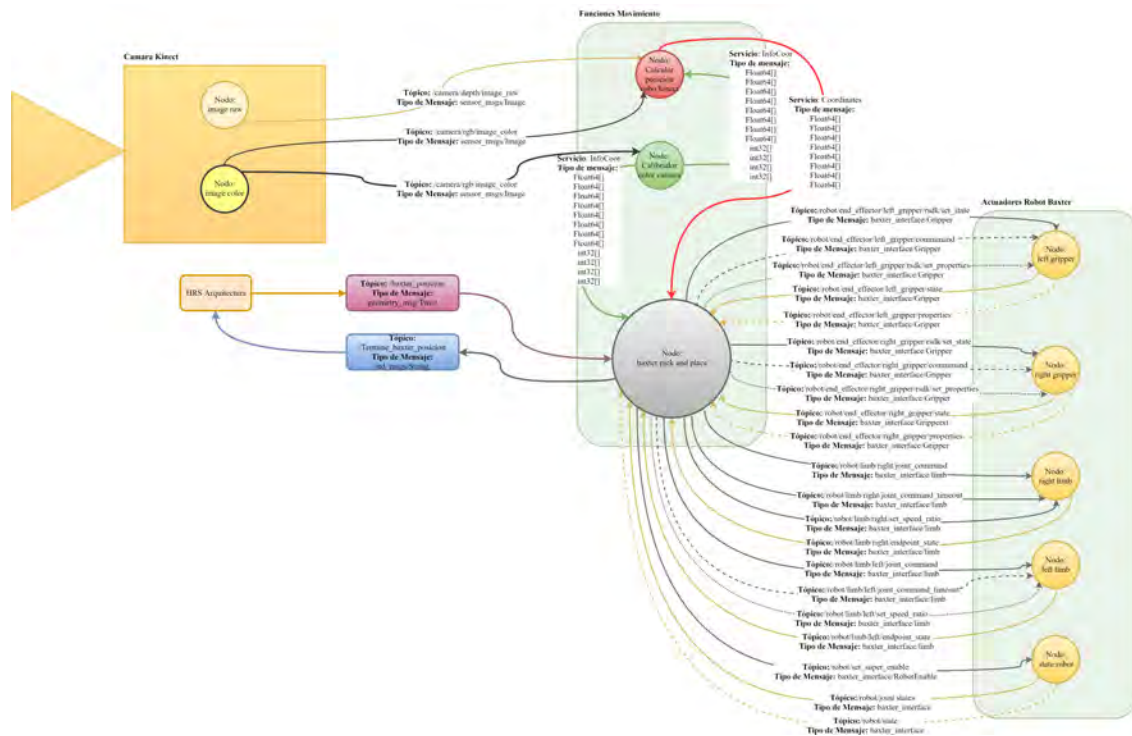


Figura 5.2: Diagrama de bloques detallado de las funciones de movimiento

La 5.2 representa el diagrama de bloques detallado de las funciones del movimiento, en este diagrama están incorporados los mensajes y servicios.

Los nodos Calibrador color bloque y Calcular posición cubo Kinect fueron descritos detalladamente en el capítulo 4; no obstante, es necesario ilustrarlos en la figura 5.2, debido a que para que Baxter logre moverse es necesario tener la visión artificial del Kinect, ya que estos nos proporcionan la ubicación de los cubos que se ubican en el tablero.

El funcionamiento de los nodos en ROS (sin incluir servicios) es la figura 5.3, la cual representa la conexión entre nodos de las funciones de percepción.

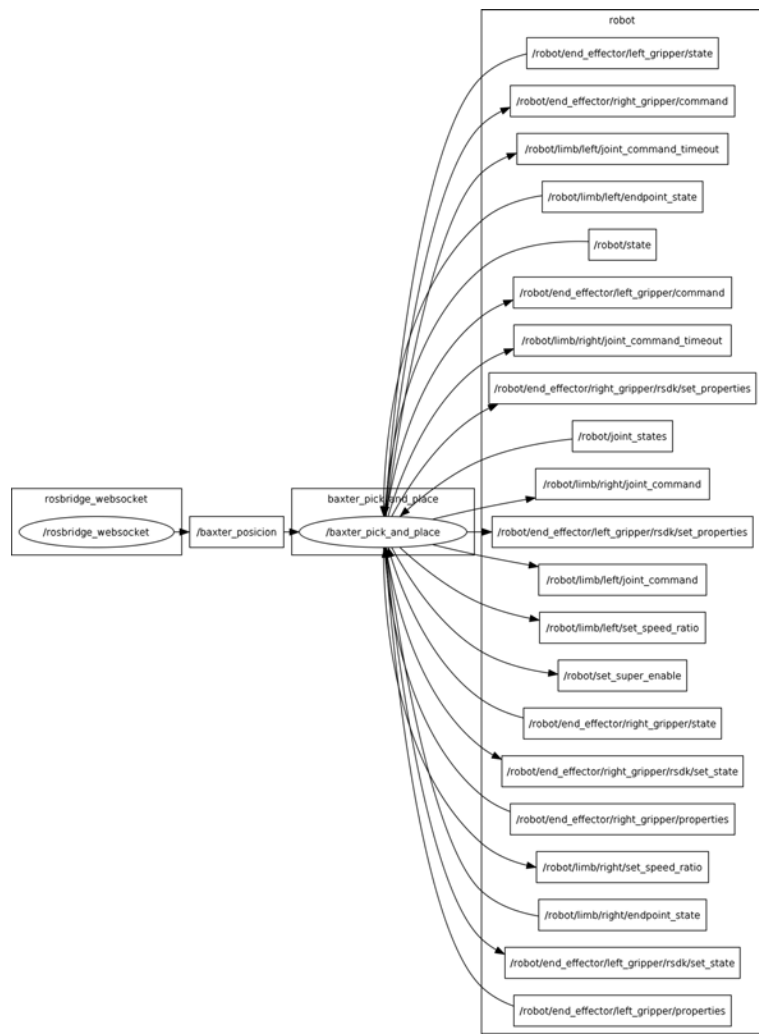


Figura 5.3: Diagrama de bloques detallado de las funciones de movimiento

5.3.1. Calibración Coordenadas de percepción con movimiento

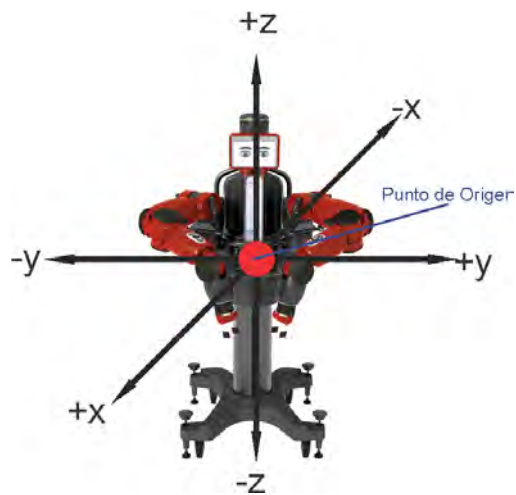


Figura 5.4: Coordenadas del Robot Baxter

Las coordenadas del robot Baxter está en metros, y de las funciones de percepción tenemos que en el nodo Calcular Posición Cubo Kinect del capítulo 2 tenemos la posición de cada casilla del tablero en metros, por tanto, para igualar las coordenadas de la cámara del Kinect con el robot Baxter fue necesario calibrar las coordenadas del robot Baxter con la cámara del Kinect manualmente como se muestra en las siguientes ecuaciones.

$$x_{\text{CoordenadasRobotBaxter}} = x_{\text{CoordenadasCamaraKinect}} - 0,0004 \quad (5.1)$$

$$y_{\text{CoordenadasRobotBaxter}} = y_{\text{CoordenadasCamaraKinect}} + 0,4547 \quad (5.2)$$

Con estas ecuaciones las coordenadas del robot Baxter y la cámara del Kinect estarán iguales, por tanto, podremos recoger los cubos que están en las casillas del tablero.

5.3.2. Posicionamiento

Para realizar movimiento del robot se implementó en el robot cinemática inversa, la cual permite determinar el movimiento del robot a partir de las coordenadas x,y de las casillas del tablero podemos determinar los ángulos de los actuadores del brazo robot para lograr el movimiento de forma correcta. La función viene implementada en ROS mediante la función IKService, cuya función esta predeterminada solo para el robot Baxter, la cual permite que mediante los puntos x,y,z (0.20 m distancia del centro de coordenadas hasta la mesa), tenga como resultado los 7 ángulos de los actuadores de los brazos del robot Baxter.

El robot tiene limitaciones para recorrer todas las casillas del tablero con un solo brazo, por tanto, el movimiento de los brazos será como el de la figura 5.5

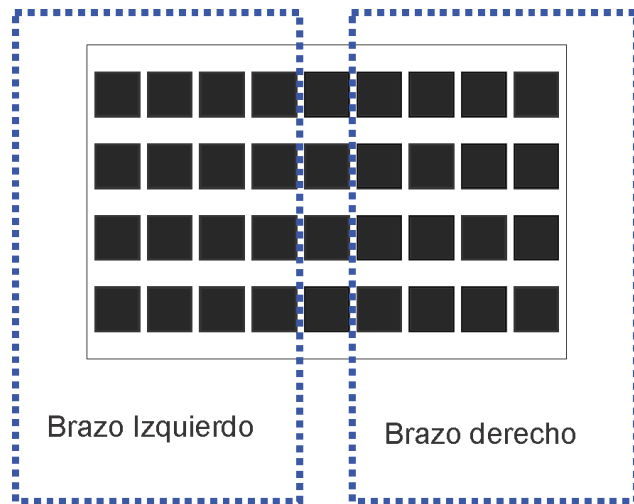


Figura 5.5: Diagrama de bloques general de las funciones de movimiento

La columna que no está resaltada en la figura 5.5, se debe a la zona en donde el brazo dejara el cubo, es decir, si la posición final del cubo está en las casillas de la región del brazo izquierdo del robot, entonces el robot moverá el brazo izquierdo; pero si la posición final del cubo se encuentra en la región derecha del tablero entonces el brazo que se moverá será el derecho.

El servicios **Coordinates** con el cambio de coordenadas nos da todas las posiciones permitidas del robot Baxter en su espacio de trabajo(tablero), estos valores están guardados en una matriz, por tanto si queremos mover el cubo de una casilla a otra solo es necesario enviar las posiciones de la matriz para poder mover el cubo, esta matriz es igual a la matriz del tablero (*Vease figura 4.8*).

5.3.3. Nodos Implementados

La figura 5.2 representa el diagrama de bloques detallado de las funciones de movimiento, los nodos que fueron desarrollados para la implementación del movimiento de los brazos de Baxter fueron: Calibrar color cámara, Calcular posición cubo Kinect y Baxter pick and place.

1. Nodo Calibrar color cámara se mencionó en el capítulo 4, debido a que hace parte de las funciones de percepción, no obstante, como la visión artificial es necesaria para realizar el movimiento del robot Baxter. Esto se debe a que este nodo contiene la información pertinente a la ubicación de los cubos y los colores que están ubicados en las casillas del tablero.
2. Nodo Calcular posición cubo Kinect se mencionó en el capítulo 4. Como se mencionó anteriormente este nodo depende de la matriz que envía el nodo *Calibrador color Cámara*. Donde su salida es una matriz que contiene las coordenadas en metros de las casillas del tablero.
3. Nodo pick and place es el nodo en donde el robot Baxter mueve sus brazos, para mover sus brazos son necesarias dos condiciones. La primera que el bloque HRS envíe un mensaje con los movimientos que Baxter debe realizar, los parámetros que envía el robot son: posición en donde debe agarrar el cubo, la posición donde se debe dejar el cubo y la velocidad en la que se debe mover el robot. La segunda condición es la posición de todos los puntos del tablero de la figura 4.3, los cuales se reciben del servicio enviado por el nodo Calcular posición cubo Kinect.

Cuando estas condiciones se cumplan el robot procederá a moverse para recoger el cubo y dejarlo en otra posición del tablero, por tanto, las posiciones donde el robot deja el cubo deben estar estrictamente dentro del tablero.

Al terminar el movimiento Baxter, este mandará un mensaje al bloque HRS confirmando que ha terminado de moverse, este mensaje es muy importante, debido a que los brazos del robot son muy grandes y es necesario tener precauciones con el usuario que interactúa con el robot, por tanto, cuando el robot este en movimiento los usuarios no deben intervenir en el tablero que es el ambiente de trabajo del robot.

El pseudocódigo 30, representa el funcionamiento del nodo Baxter pick and place.

Algorithm 12 nodo Baxter pick and place

```
Recibir servicio Coordinates de Calcular posición cubo kinect
posición x de baxter=x servicio - 0.0004
posición y de baxter=y servicio + 0.4547
while pick and place do
  x = posición x de Baxter
  y = posición y de Baxter
  z = 0.20
  Recibir servicio InfoCoor de Calibrador Color Camara
  Con los arreglos del servicio InfoCoor crear matriz de 4x9
  if Cubo está en la 5(número ubicado en la matriz de 4x9) casilla del tablero then
    if place > 5 then
      pick and place mano derecha
    end if
    if place < 5 then
      pick and place mano izquierda
    end if
  else if pick es brazo izquierdo then
    if si place está en la posición 6,7,8 o 9 del tablero then
      place brazo derecho
    else
      place brazo izquierdo
    end if
  else if pick es brazo derecho then
    if si place está en la posición 1,2,3 o 4 del tablero then
      place brazo izquierdo
    else
      place brazo derecho
    end if
  end if
end while
```

Con los datos del servicio **Infocoor** de la tabla 4.1 se creará una matriz con la etiqueta de los colores de las casillas del tablero de la siguiente forma:

$$Camara^{4 \times 9} = \begin{bmatrix} matriz_5[0] & matriz_5[1] & \cdots & matriz_5[8] \\ matriz_6[0] & matriz_6[1] & \cdots & matriz_6[8] \\ matriz_7[0] & matriz_7[1] & \cdots & matriz_7[8] \\ matriz_8[0] & matriz_8[1] & \cdots & matriz_8[8] \end{bmatrix} \quad (5.3)$$

El servicio **Coordinates** tiene las coordenadas de cada casilla del tablero (si hay un cubo en esta posición o no) en metros. La tabla 4.2 muestra cómo se envían los datos por este servicio.

Como las dimensiones de todos los arreglos son la misma entonces se creará una matriz, en donde el contenido estará como el siguiente ejemplo:

$$matrizPosiciones_{00} = (xc1[0], yc1[0]) \quad (5.4)$$

Esto quiere decir que el contenido de la primera matriz el eje de coordenadas X, Y. Por tanto, la matriz de posiciones de los cubos en metros estará organizada de la siguiente manera:

$$matrizPosiciones^{4 \times 9} = \begin{bmatrix} (xc1[0], yc1[0]) & \cdots & (xc1[8], yc1[8]) \\ (xc2[0], yc2[0]) & \cdots & (xc2[8], yc2[8]) \\ (xc3[0], yc3[0]) & \cdots & (xc3[8], yc3[8]) \\ (xc4[0], yc4[0]) & \cdots & (xc4[8], yc4[8]) \end{bmatrix} \quad (5.5)$$

Por último, es necesario igualar el eje de coordenadas del robot Baxter con el eje de coordenadas de la cámara del Kinect. Para esto utilizamos las ecuaciones 5.1 y 5.2. Con estas ecuaciones el robot Baxter ya tiene todos los puntos en los cuales puede moverse dentro del tablero. Como se puede observar en la matriz de la ecuación 5.3 y 5.5, son de las mismas dimensiones, por tanto, con la información de los cubos que hay en el tablero según la matriz cámara podemos decir que si queremos mover algún cubo solo tenemos que dar a través de la matriz Posiciones la instrucción para que el robot realice el movimiento hacia donde el usuario lo desee. Donde el usuario sería el Bloque HRS.

Para activar el movimiento el Bloque HRS manda un mensaje de tipo Twist que se explicó en el capítulo 3, para realizar el movimiento con Baxter los datos `linear.x` y `linear.y` son las coordenadas (x,y) de la posición que el robot Baxter debe agarrar el cubo, además los datos `linear.z` establecen la velocidad del robot. Mientras que los datos `angular.x` y `angular.y` son las coordenadas (x,y) de la posición que el robot debe dejar el cubo dentro del tablero.

Los nodos que permiten activar los actuadores de los brazos de Baxter para realizar el movimiento de Baxter son los siguientes: `left gripper`, `right gripper`, `right limb`, `left limb`, `state robot`.

5.4. Validación de resultados

En la validación de las funciones de movimiento se realizó una rutina del movimiento de los brazos en la cual el robot se encargaba de mover un cubo de una casilla del tablero a la que se encuentra su lado, así hasta terminar con la primera columna y pasar a la siguiente, asegurándonos de que el robot coloque el cubo en cada una de las posiciones del tablero; repitiendo estos movimientos con un cubo por color (verde, rojo, amarillo y azul), estas 4 rutinas fueron grabados para luego revisarlas y ver la eficiencia del robot en el movimiento de los cubos, en la figura 5.6 podemos ver unas cuantas escenas donde se aprecia la secuencia de movimiento del cubo.

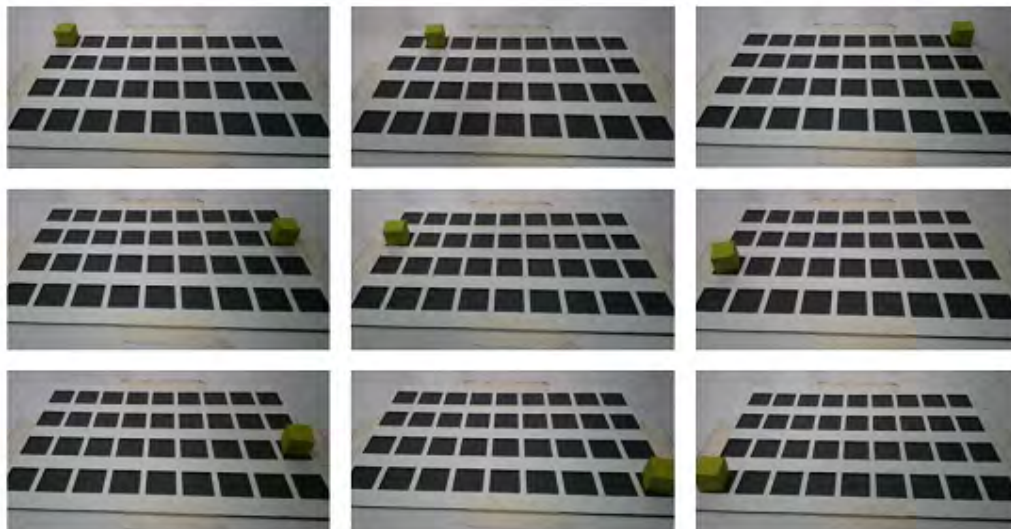


Figura 5.6: Movimiento del cubo

Luego de revisar cada video y contar el número de aciertos del robot se obtuvo como resultado:

Color Cubo	Movimientos Exitosos	Movimientos Fallidos	Porcentaje de Movimiento del Cubo
Verde (1)	25	10	71.42 %
Rojo (2)	31	4	88.57 %
Amarillo (3)	28	7	80 %
Azul (4)	31	4	88.57 %
Total	115	25	82.14 %

Cuadro 5.1: Resultados Validación Función Movimiento

Los movimientos se consideran fallidos si el robot Baxter deja el cubo tocando con los bordes del tablero, es decir, que el cubo este sobre la superficie del tablero.

5.5. Conclusiones

1. Analizando los resultados obtenidos en la revisión de los videos, se puede concluir que la ubicación del cubo en cualquier casilla del tablero debe estar centrado en el centro de la casilla, de esta forma se asegura que el movimiento del cubo por todo el tablero tenga menor porcentaje de error. Esto se debe a que si el cubo no queda centrado en la casilla entonces el gripper no tendrá buen agarre con el cubo y tendremos un corrimiento en el agarre y quedará un desfase al dejar el cubo en la casilla destinada, donde ese desfase quedará para todos los movimientos.
2. Un error del Robot Baxter fue el proceso de calibración de los brazos, este problema ocurrió debido a que los brazos deben calibrarse mensualmente, pero por problemas de espacio no se pudo realizar esta calibración, por tanto, cuando se combinaban las coordenadas de la cámara con Baxter se creaba un desfase entre estas coordenadas, para solucionar este problema fue necesario mover la tabla para resolver el problema de desfase.
3. En los resultados obtenidos, cuando el robot dejaba un cubo, este no quedaba en el centro de la casilla, sino que tenía un desfase en milímetros, pero este desfase fue el mismo en todo el recorrido de los cubos por el tablero, en conclusión, se obtuvo precisión mas no exactitud, es decir, el cubo nunca estuvo en el centro de la cuadrícula; pero el desfase milimétrico que tuvo el cubo en su recorrido fue el mismo.

Capítulo 6

Conclusiones

1. Para establecer los requerimientos físicos, tecnológicos y pedagógicos las funciones de expresión emocional, percepción y movimiento fueron diseñadas de acuerdo con las condiciones de la tarea de aprendizaje, de las posibilidades técnicas del robot Baxter y de las condiciones de enseñanza de acuerdo a la arquitectura Human-Robot Scaffolding. Las funciones de expresión emocional se diseñaron de acuerdo con las restricciones educativas en las cuales, los agentes tutores expresan emociones con el objetivo de orientar el aprendizaje del estudiante. Es decir, las emociones no buscan transmitir las emociones están asociadas a aspectos instruccionales para que los niños reconozcan al robot como un agente colaborador. Las condiciones de diseño de las funciones de percepción están en función de la matriz de 4x9 porque este espacio de trabajo es ideal para que los niños manipulen los bloques con propósitos de aprendizaje. Finalmente, los requerimientos para las funciones de movimiento fueron implementados a través del servicio IK Service mediante el cual no es necesario hacer los cálculos cinemáticos, sino llamar la función que tiene como parámetros de entrada la posición final del actuador.
2. En el diseño de la arquitectura que integro las plataformas de ROS y RobotAct se diseñaron las funciones de movimiento, expresión emocional y percepción de acuerdo a los principios de diseño de ROS: El trabajo explica como fue el proceso de diseño de cada función de acuerdo a las características técnicas. Como se puede observar en los capítulos previos, en los videos de interacción con niños y con estudiantes universitarios, las funciones de expresión, ahora controladas desde JAVA, se integran de manera exitosa a los requerimientos de la arquitectura RobotAct. Como es presentado en los capítulos 2,3,4 y 5, el diseño de los nodos ROS se ajusta de manera eficaz a las necesidades de la plataforma.
3. La implementación en ROS con las funciones definidas se puede afirmar que fueron diseñados y construidos los nodos para las funciones de expresión emocional, con sus respectivos tópicos para la comunicación entre nodos ROS y la comunicación con los nodos y tópicos de ambientes externos como JAVA en la cual está desarrollada la arquitectura RoboAct. Además, fueron diseñados y construidos los nodos para las funciones de percepción, con los respectivos tópicos para enviar la información del estado de la tarea y de cada uno de los bloques de diferentes colores; información que es útil para el comportamiento del robot durante el proceso de aprendizaje del estudiante que interactúa. Finalmente, para las funciones de movimiento fueron diseñados e implementados los nodos con sus respectivos tópicos y protocolo de comunicación entre ROS y con JAVA. Como se aprecia en los videos, que no deben ser publicados, solo para los propósitos de la sustentación de la tesis, cada uno de los movimientos del robot se desarrolla de acuerdo con las condiciones de diseño y las restricciones del ambiente. El robot se posiciona con cada uno de sus dos brazos de acuerdo con la distribución espacial de los bloques y la distancia con el robot.
4. Para poder validar la metodología de validación de las funciones de movimiento presentadas, demuestran la eficacia de cada una de las funciones. Es importante resaltar que las funciones también fueron probadas con usuarios entre 10 y 16 años y como es evidente en los videos tomados en cada una de las experiencias, los comportamientos de Baxter a nivel emocional, de percepción y de movimiento, tienen a desarrollarse asertivamente y el comportamiento errático no es frecuente como se presenta en la tabla 4.3. Además, las funciones fueron desarrolladas, como se muestra en el repositorio, utilizando el lenguaje nativo de ROS.

Bibliografía

- [1] anki. kozmo. url <https://co.pinterest.com/pin/203013895682632899/>.
- [2] Cynthia Breazeal and Juan Velásquez. Toward teaching a robot 'infant' using emotive communication acts. In *Proceedings of the 1998 Simulated Adaptive Behavior Workshop on Socially Situated Intelligence*, pages 25–40. Citeseer, 1998.
- [3] Open CV. Image documentation. url <https://docs.opencv.org>.
- [4] Wiki de Robótica. Cinemática de robots. url <http://wiki.robotica.webs.upv.es/wiki-de-robotica/cinematica/>.
- [5] Luis Ayala y Cristhian Bermeo Francisco Araujo. Robots antropomórficos. url https://arhatarahant.files.wordpress.com/2014/07/robots_antropomorficos.pdf.
- [6] Juan J González and Jovani Alberto Jiménez Builes. La robótica como herramienta para la educación en ciencias e ingeniería. *IE Comunicaciones: Revista Iberoamericana de Informática Educativa*, (10):31–36, 2009.
- [7] Josefina Guzmán, Jorge Mata, and María Gómez. El sonido como herramienta tecnológica de apoyo a la educación. 2. 2008.
- [8] Colorado-Montaña J Jaramillo Botero A, Acosta Mejía J. *ROBÓTICA: UN ENFOQUE TEÓRICO Y COMPUTACIONAL (ROBOTICS: a theoretical and computational approach)*. McGraw-Hill, 2016.
- [9] Theodore D Kemper. How many emotions are there? wedding the social and the autonomic components. *American journal of Sociology*, 93(2):263–289, 1987.
- [10] kismet. Can robots feel your pain. url https://greatergood.berkeley.edu/article/item/can_robots_feel_your_pain.
- [11] kobian. emo robot. url <http://nothingnerdy.wikispaces.com/emo+robot>.
- [12] H Liang. Controlling robots using vision and bio sensors. pages 6–8, 2014.
- [13] MathWorks. Call and provide ros services. url <https://www.mathworks.com/help/robotics/examples/call-and-provide-ros-services.html>.
- [14] MathWorks. Exchange data with ros publishers and subscribers. url <https://www.mathworks.com/help/robotics/examples/exchange-data-with-ros-publishers-and-subscribers.html>.
- [15] milo. Milo the robot helps students with autism work on social skills. url http://www.nonpareilonline.com/news/faces/education/milo-the-robot-helps-students-with-autism-work-on-social/article_36665856-3e7e-11e7-b4f7-033021f298b3.html.
- [16] nexi. Nexi now for sale. url <https://robotliving.com/robot-news/nexi-now-for-sale/>.
- [17] David Nuñez, Marco Tempest, Enrico Viola, and Cynthia Breazeal. An initial discussion of timing considerations raised during development of a magician-robot interaction. In *ACM/IEEE International Conference on Human-Robot Interaction (HRI) Workshop on Timing in HRI*, 2014.
- [18] Robot Paro. Playing with a robot. url http://www.bostonhigashi.org/about_us/BHSresearch/research_initiatives/play

- [19] Charles Rich, Brett Ponsler, Aaron Holroyd, and Candace L Sidner. Recognizing engagement in human-robot interaction. In *Human-Robot Interaction (HRI), 2010 5th ACM/IEEE International Conference on*, pages 375–382. IEEE, 2010.
- [20] Rethink Robotics. Baxter arms specifications. url http://sdk.rethinkrobotics.com/wiki/Hardware_Specifications.
- [21] Rethink Robotics. Baxter hardware specification architecture datasheet. url <https://www.active8robots.com/wp-content/uploads/Baxter-Hardware-Specification-Architecture-Datasheet.pdf>, 2013.
- [22] Rethink Robotics. Robot baxter. url <https://johnbrooks.co.nz/Baxter-Robot>, 2013.
- [23] Rethink Robotics. Robot baxter hardware. url http://mfg.rethinkrobotics.com/wiki/Robot_Hardware, 2013.
- [24] ROS.org. Powering the world’s robots. url <http://www.ros.org/>.
- [25] Andres F Salazar-Gomez, Joseph DelPreto, Stephanie Gil, Frank H Guenther, and Daniela Rus. Correcting robot mistakes in real time using eeg signals. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 6570–6577. IEEE, 2017.
- [26] Industry Tap. Dave ice cream robot. url <http://www.industrytap.com/ice-cream-making-robot-serves-game-changing-ai/21196>.
- [27] Jeannine E Turner, Ralph M Waugh, Jessica J Summers, and Crissie M Grove. Implementing high-quality educational reform efforts: An interpersonal circumplex model bridging social and personal aspects of teacher’s motivation. In *Advances in teacher emotion research*, pages 253–271. Springer, 2009.
- [28] ucbBaxterTower. Baxter builds a tower. url <http://ucbbaxtertower.github.io/BaxterTower/>, 2016.
- [29] John Jairo Páez Vela, Luis Carlos Sarmiento y Rodriguez. Robótica. In *Cinemática de Robots Manipuladores con Matlab*, pages 1–105. Autores, 2007.
- [30] Paulina Andrea Vélez and Alex Quingatuña. Robot socialmente asistencial y de interacción multimodal autónoma. *INGE@ UAN-Tendencias en la Ingeniería*, 5(9), 2014.
- [31] Sonic Worlds. Bleep - cute robot voice. url <https://www.asoundeffect.com/sound-library/bleep-cute-robot-voice/>, 2016.

Capítulo 7

Anexos

Todos los archivos y anexos del trabajo de grado se encuentran en el repositorio https://bitbucket.org/GADG94/anexos_tesis_ingenieria/src, incluyendo el documento del Trabajo de grado. Los archivos son:

1. Informe final.
2. Proyecto ROS.
3. 2000 Imágenes.
4. Sonidos del robot Baxter.
5. Formato de la encuesta.
6. Enlace de los videos.