Ernesto Torres Vidal
Jesús Rodríguez Ordoñez

Arquitectura del equipo
Eagle Knights de la liga Small Size

Sistema de visión

Hardware utilizado y conexión de las cámaras

El sistema de visión está compuesto por dos cámaras *Guppy G-033C* de la marca *AVT* (*Allied Vision Technology*) y por elementos de conexión que sirven para obtener las señales de video en una computadora *Dell M6300*. De acuerdo con las reglas de la liga[1] las cámaras deben estar montadas a cuatro metros por encima del campo de juego. Las dimensiones del campo de juego son las siguientes:

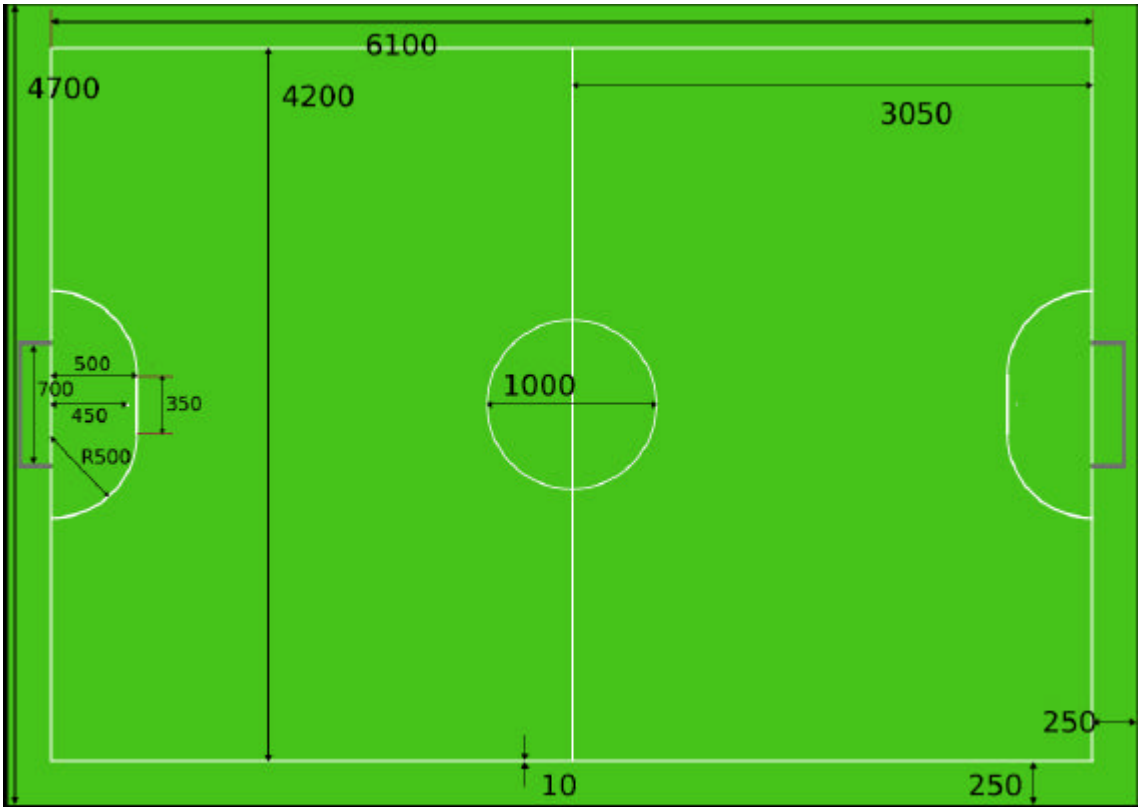


Figura1. Campo de juego

Los elementos de conexión son los siguientes:

- Dos cables *FireWire* (IEEE 1394) de cuatro metros de longitud.
- Dos repetidores *FireWire* (HUBIOG050)
- Dos transformadores de 9V para alimentar los repetidores.
- Un multicontacto con extensión para conectar los transformadores
- Una tarjeta *ExpressCard* con dos puertos *FireWire*

Los repetidores se utilizan para lograr que cada cámara se conecte a una distancia de hasta ocho metros. Si la computadora en donde se corre el sistema de visión es una laptop la cámara no recibe suficiente energía para transmitir la señal por lo que es indispensable el repetidor, en cambio, si se utiliza una PC la cámara recibe alimentación suficiente de tal forma que el repetidor se puede usar sólo como extensión.
La conexión hacia la computadora se puede hacer mediante una tarjeta PCMIA o una *ExpressCard* que tenga dos puertos de *Firewire* dependiendo del tipo de entrada que ésta disponga.
A continuación se muestra un diagrama de la conexión de ambas cámaras con la computadora:

Figura 2 Conexión de las cámaras con la laptop

Instalación del software de visión

El *software* de visión está compuesto de dos programas. El primero es el software provisto por AVT. La finalidad de este sistema es poder visualizar las imágenes de las cámaras conectadas en la computadora y realizar la configuración del formato de la imagen, así como ajustar los parámetros como son el obturador de la cámara, los cuadros por segundo, la ganancia y el brillo de la imagen, entre otros. La instalación se realiza por medio de los discos de instalación o descargando el software de la página web[2].

Una vez instalado este *software* es necesario instalar un *driver* que permita detectar la conexión de las cámaras por medio del siguiente acceso directo:

Ajuste de las cámaras

Para realizar la configuración de las cámaras se ejecuta el software de AVT y se selecciona de la lista la cámara con la que se desea trabajar:

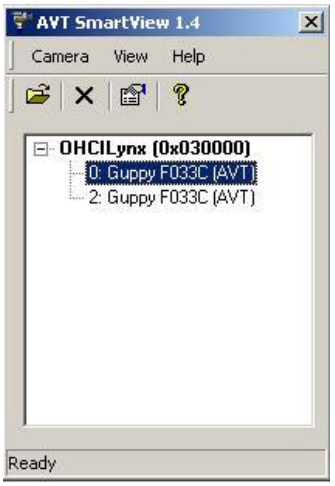


Figura 3. Selección de cámaras

Las cámaras se sostienen utilizando dos brazos que sirven para montarlas a la barra encima del campo de juego. Una vez colocadas y conectadas a la computadora se deben orientar correctamente y ajustar el foco de la imagen. La orientación de la cámara debe permitir ver completamente los límites del campo y una sección extra a partir de la media cancha (al menos la mitad del semicírculo de la media cancha), esto con la finalidad de no perder ningún objeto en el proceso de búsqueda. Es importante también que la imagen se vea lo más derecha posible.

El ajuste del foco se realiza manualmente en el lente gran angular y debe permitir ver los bordes de los parches de los robots de tal forma que los límites sean distinguibles y no se vean borrosos. Además es necesario que el ajuste del foco se realice al mismo tiempo que el obturador (*shutter*) por el cambio en la cantidad de luz que pasa hacia el CCD de la cámara. Este proceso se explicará mas adelante. La siguiente imagen muestra la configuración en el tabulador de formato al seleccionar la opción *"Settings"*:

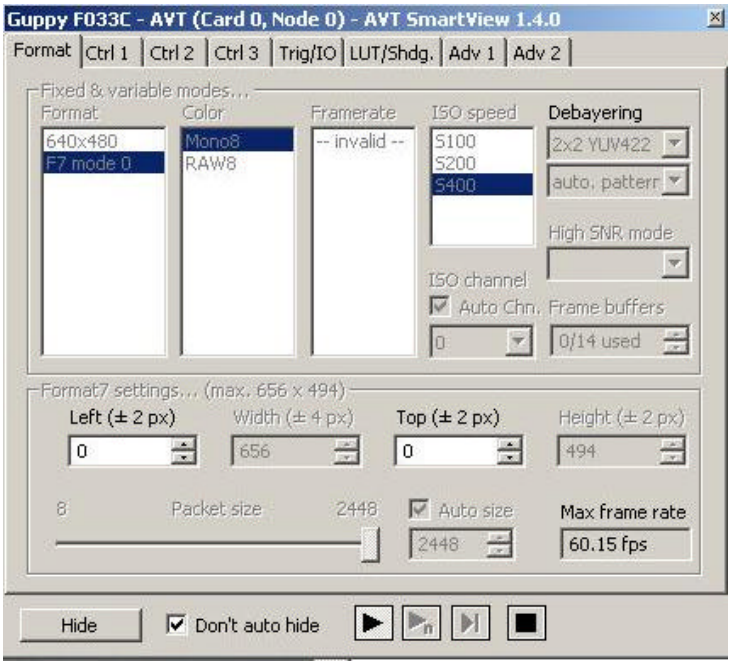


Figura4: Configuración del formato de la imagen

El tabulador **Format** permite configurar la resolución de la imagen la cual se recomienda que sea de 576x444 (*Width*, *Height*) por compatibilidad con el software de EKVision. La tasa máxima de cuadros por segundo que puede entregar cada cámara es de 60.15 fps. Los puntos *left* y *top* son útiles para centrar la imagen.

La siguiente ventana permite ajustar los parámetros más importantes de la imagen. El tabulador *Shutter* permite ajustar el obturador de la imagen modificando en consecuencia los cuadros por segundo a los que se entrega el video. Esto se debe a que mientras menos tiempo permanece cerrado el obturador permite pasar más luz pero es más tardado para entregar cada imagen, en cambio si el tiempo en el que permanece cerrado es mayor, la cantidad de luz es menor y se entregan muchos cuadros por segundo. Se recomienda trabajar con una tasa de aproximadamente 35 cuadros por segundo y con base en este valor ajustar el foco directamente en la cámara. En caso de que la imagen tenga poca iluminación se puede ajustar la ganancia (*Gain*) para mejorar el tono de los colores. Un efecto que tiene esta herramienta es que algunos parches se ven más borrosos. Las demás herramientas permiten realizar algunos cambios en el color de la imagen.

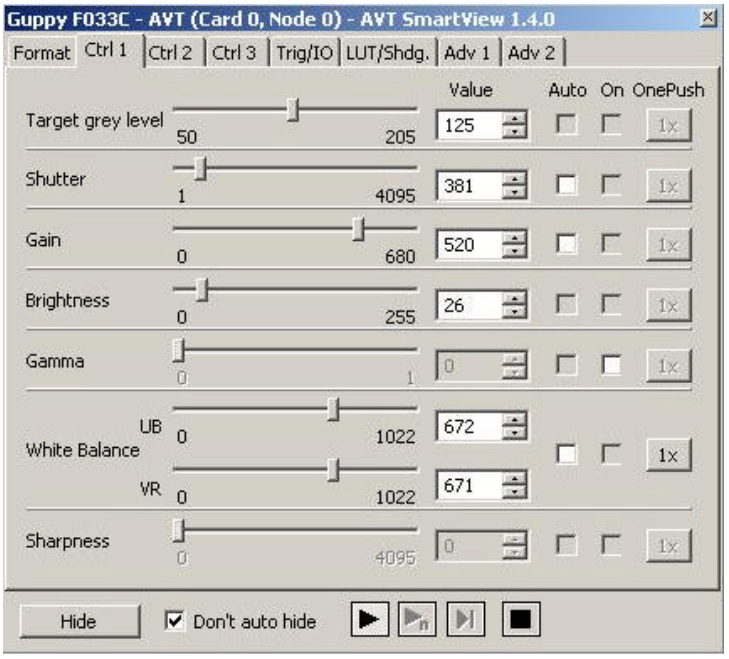


Figura5: Configuración de la imagen

Sistema EKVision

El sistema de visión EKVision es un software programado sobre el ejemplo "*FireDemoProject*" que se encuentra dentro del *FirePackage* del software e AVT. Este software está programado utilizando la interfase gráfica de MFC de Microsoft Visual C++ 6.0 El programa inicial está compuesto por 81 clases y permite resolver el problema de comunicar la aplicación con el hardware de la computadora y así obtener la el buffer de las imágenes, modificar los *settings*, realizar zoom y tomar imágenes ".bmp"

El sistema de visión está compuesto por un menú principal y una serie de ventanas que se encuentran agrupadas en tabuladores. El menú principal permite seleccionar una o dos cámaras en el combo *ActiveCard*. Posteriormente se muestra un listado de las cámaras seleccionadas y otra información que genera el sistema. Además de esta información se cuenta con tres controles principales para el manejo de todo el sistema que son Cámara, Procesamiento y Calibración

Nota: Inicialmente la imagen de la cámara 1 aparecerá negra, esto es normal

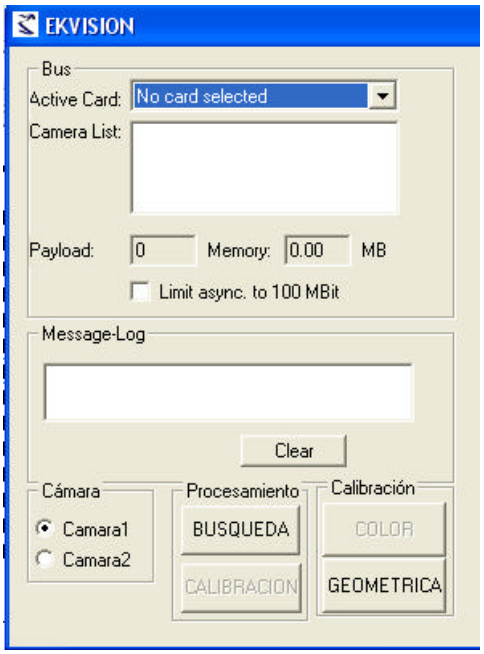
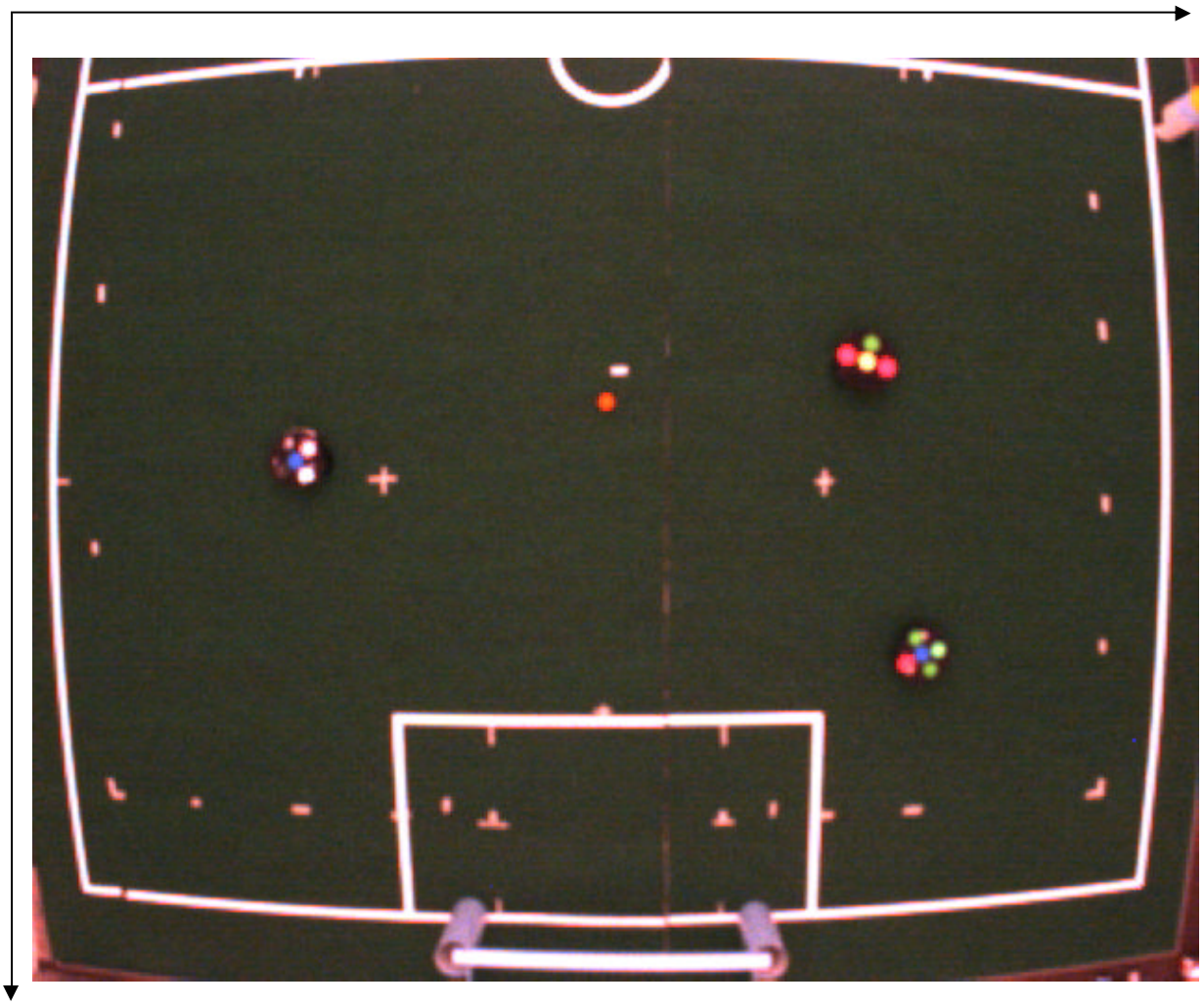


Figura6: Menú principal EKVision

La imagen tiene una resolución de 577 x 444 y las el buffer de la imagen entrega los valores de cada píxel por renglón partiendo de arriba a la izquierda. En la imagen se muestra el sistema de referencia utilizado dentro del sistema para los ejes de las coordenadas.

WIDTH 577 pixeles (Eje X)



HEIGHT 444 pixeles (Eje Y)

Figura 7: Sistema de coordenadas del sistema de visión

Los controles de “Cámara” permiten seleccionar alguna de las dos cámaras sobre la que se quiere trabajar en los procesos de calibración. Los botones de “Procesamiento” permiten cambiar entre la etapa inicial de calibración y el proceso de búsqueda de objetos. Finalmente los botones de “Calibración” permiten seleccionar entre los dos tipos de calibración con los que cuenta el sistema: geométrica y color.

Todos los archivos utilizados por el sistema de visión se encuentran en la carpeta: “C:\Calibraciones”.

El primer paso una vez que se han ajustado los parámetros y la posición de la cámara es realizar la calibración geométrica. El objetivo de esta calibración es poder obtener un sistema de coordenadas correspondientes a las medidas reales en el campo (en el sistema se manejan en milímetros) y la coordenada en pixeles que se lee del buffer de cada una de las imágenes. Además permite resolver los problemas ocasionados por la distorsión del lente gran angular y la intersección existente entre ambas cámaras en la región de la media cancha.

Los pasos para realizar la calibración geométrica son los siguientes:

1. Seleccionar la cámara que se desea calibrar y las opciones de CALIBRACION y GEOMETRICA.
2. Se selecciona el tabulador titulado “**Geométrica**”.

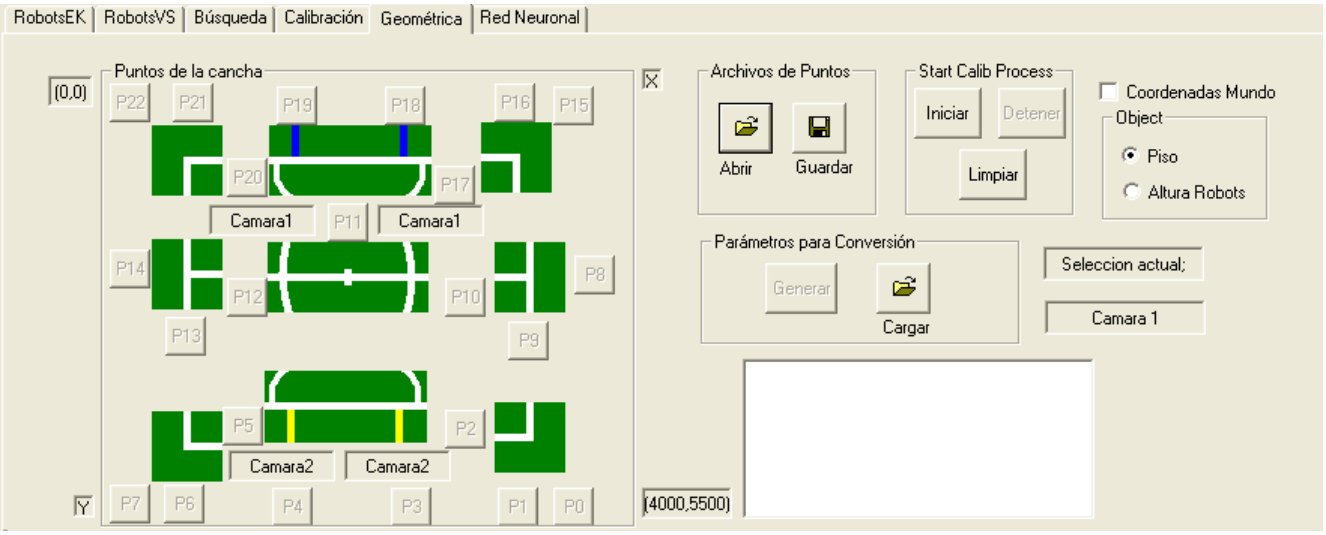


Figura 8: Calibración geométrica

3. En esta ventana se tienen dos opciones: se puede iniciar un nuevo proceso de calibración o se pueden abrir los parámetros de la cámara generados por una calibración anterior. Automáticamente el sistema carga los parámetros de la cámara contenidos en un archivo definido dentro del código (en el constructor de la clase **CImageWorld**) para evitar tener que seleccionarlo cada vez que se corre el software.

4. Para iniciar un nuevo proceso de calibración se selecciona la opción Iniciar dentro del grupo "**Start Calib Process**".

Figura 9: Selección de puntos

5. La opción de iniciar habilita los puntos de referencia que se deben capturar para que la calibración funcione. En este punto se debe verificar que la cámara dos se puede colocar debajo de la cámara uno (como se ve en la imagen) y que el campo de juego no se vea invertido. Las coordenadas correspondientes a cada uno de los puntos que se deben capturar se muestran en la siguiente tabla. La primera columna representa cada uno de los puntos, la segunda las coordenadas reales medidas en milímetros y la tercera en cual de las cámaras se debe considerar.

Punto	**Coordenada (X,Y) mm**	**Cámara que lo utiliza**
P0	4700,6600	Cámara 2
P1	4450,6350	Cámara 2
P2	3025,6350	Cámara 2
P3	2700,6350	Cámara 2
P4	2000,6350	Cámara 2
P5	1675,6350	Cámara 2
P6	250,6350	Cámara 2
P7	0,6600	Cámara 2
P8	4700,3300	Cámaras 1 y 2
P9	4450,3300	Cámaras 1 y 2
P10	3350,3300	Cámaras 1 y 2
P11	2350,3300	Cámaras 1 y 2
P12	1350,3300	Cámaras 1 y 2
P13	250,3300	Cámaras 1 y 2
P14	0,3300	Cámaras 1 y 2
P15	4700,0	Cámara 1
P16	4450,250	Cámara 1
P17	3025,250	Cámara 1
P18	2700,250	Cámara 1
P19	2000,250	Cámara 1
P20	1675,250	Cámara 1
P21	250,250	Cámara 1
P22	0,0	Cámara 1

Tabla 1: Puntos de referencia

6. El proceso de selección de cada punto consiste en seleccionar el botón y posteriormente seleccionar el punto en la imagen en donde se encuentra ese punto. Si la selección es correcta aparece una cruz naranja y se añade a la lista:

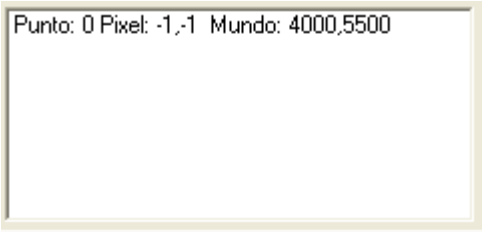


Figura 10: Listado de puntos

7. Al finalizar de seleccionar todos los puntos de una cámara se oprima el botón Detener. En caso de tener algún error se puede reiniciar el proceso con el botón limpiar. Posteriormente se guarda un archivo dentro de la carpeta "**C:\Calibraciones**" en el que automáticamente se escribe en el nombre del archivo la fecha actual:

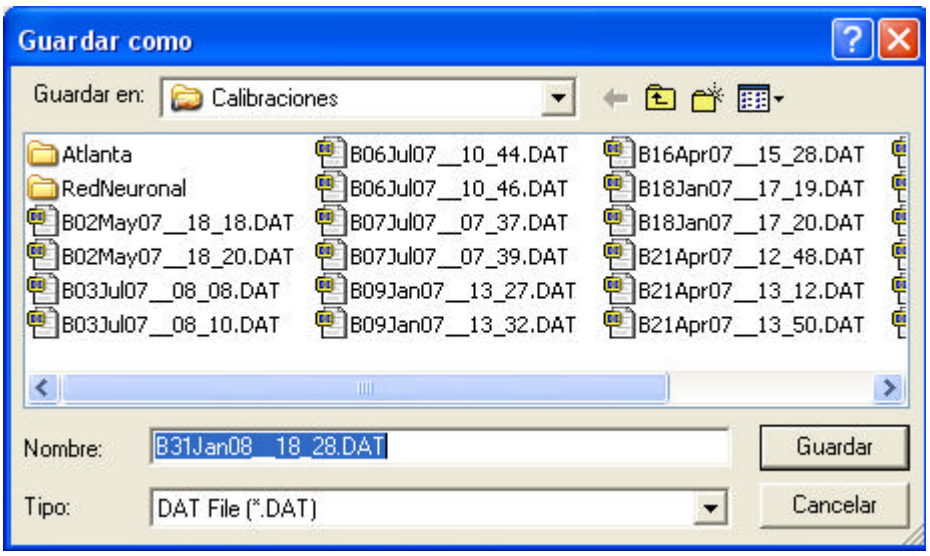


Figura 11: Archivo de puntos

8. Después de generar el archivo con las correspondencias de cada punto se selecciona la opción Generar por medio del cual se obtiene un nuevo archivo con el que a partir del archivo de puntos se generan los parámetros de la cámara por medio del algoritmo *Tsai.*

9. Finalmente se carga este archivo en el grupo "**Parámetros para Conversión**". Para confirmar que la conversión se está realizando correctamente se puede activar el *Checkbox* "Coordenadas Mundo" y automáticamente al pasar el ratón por cualquier punto de la imagen se imprime un cuadro con la coordenada correspondiente en milímetros

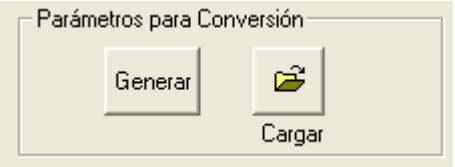


Figura 12 Generación de parámetros

Los pasos para realizar la calibración por color son los siguientes:

1) Seleccionar la cámara que se desea calibrar y las opciones de CALIBRACION y COLOR en los botones del menú principal. En esta opción la imagen de la cámara seleccionada se verá completamente negra debido a que inicialmente no se han clasificado píxeles dentro de ningún color.

2) Se selecciona el tabulador titulado "**Calibración**"

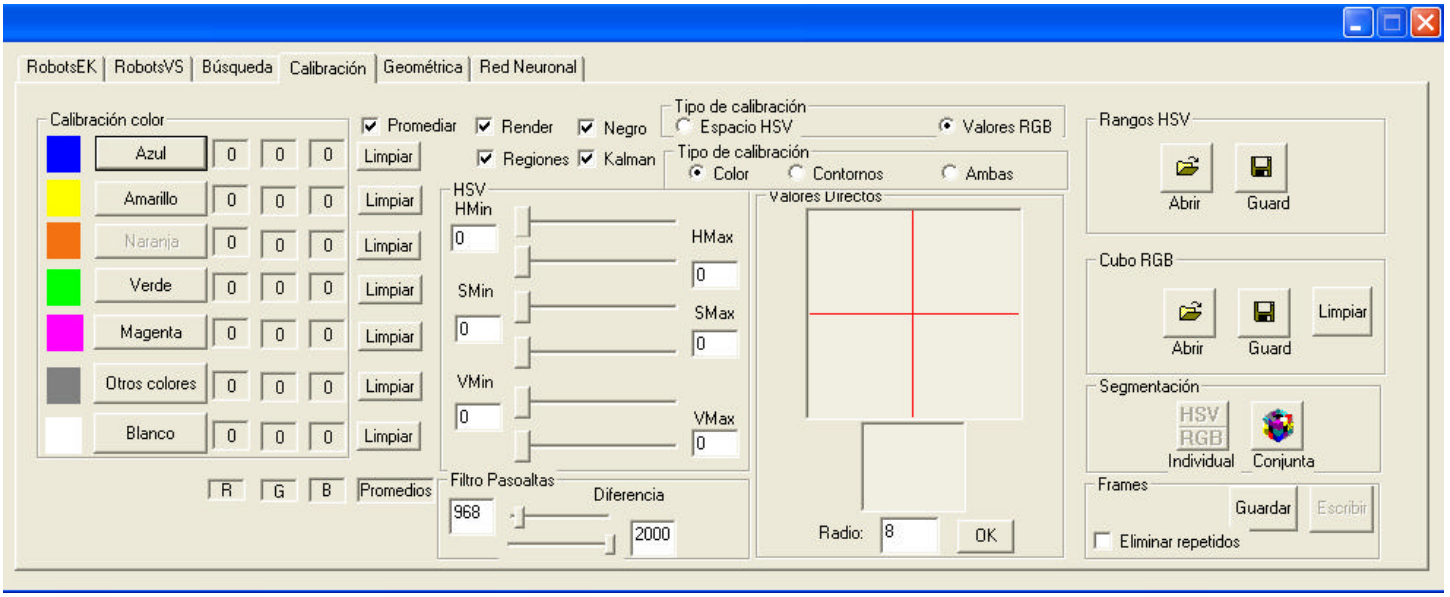


Figura 13: Calibración por color

3) El control está compuesto por una lista de los colores que se quieren calibrar. Para ver la imagen sin sobreponer el resultado de la segmentación se desactiva la opción de "**Render**". Si sólo se desea desactivar el color negro se desactiva la opción "**Negro**".

Existen dos métodos para calibrar por color. Para cambiar entre ambos se utilizan los siguientes controles:

RGB

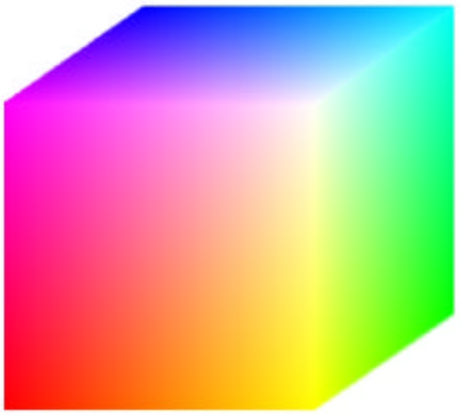


Figura 14: Cubo RGB

4) Para empezar a calibrar se selecciona el color y se seleccionan los píxeles que se quieren capturar directamente en la imagen con el botón izquierdo del ratón. Si se desea eliminar una selección se oprime el botón derecho del ratón. Esta calibración se realiza directamente sobre el espacio de color RGB que es el entregado por las imágenes. En caso de que un grupo de píxeles formen una región se dibujará un círculo punteado alrededor de la región. Para desactivar esta opción se utiliza el *checkbox* "Regiones". Cada color calcula un promedio del valor total de los píxeles seleccionados, este promedio se puede limpiar utilizando el botón correspondiente o se puede desactivar utilizando el *checbox* "Promediar".

HSV

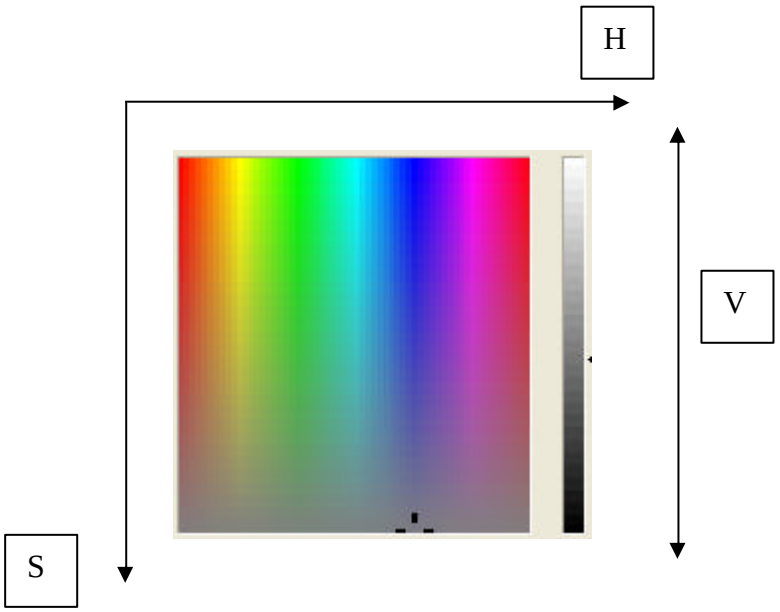


Figura 15: Espacio HSV

5) Este segundo método de calibración (no es necesario si se utiliza el descrito en el punto anterior) utiliza las barras del grupo HSV. Este espacio tiene tres dimensiones, cada una con un rango acotado entre un mínimo y un máximo. Los colores utilizados en la imagen se pueden separar fácilmente por medio de estos rangos ya que contienen un alto nivel de saturación, un rango extenso de

iluminación (para ser más flexible con cambios en la iluminación) y un rango acotado de tono para determinar el color del objeto.

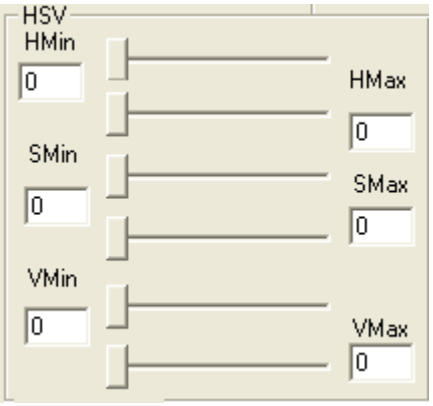


Figura 16: Rangos HSV

6) Una vez finalizada la calibración de todos los colores en HSI es necesario realizar una conversión del equivalente de estos rangos al espacio RGB debido a que las cámaras entregan la información en este espacio. Esto se realiza seleccionando la opción "**Conjunta**". Este proceso puede tardar varios segundos. En caso de querer realizar alguna modificación sobre el resultado se puede hacer utilizando la calibración RGB.

Figura 17: Conversión HSV a RGB

7) Es posible guardar o abrir el resultado de una calibración o limpiar completamente una calibración utilizando los siguientes controles:

Figura 18: Manejo de archivos de calibración de color

Identificación de robots y pelota y ajuste de parámetros de búsqueda.

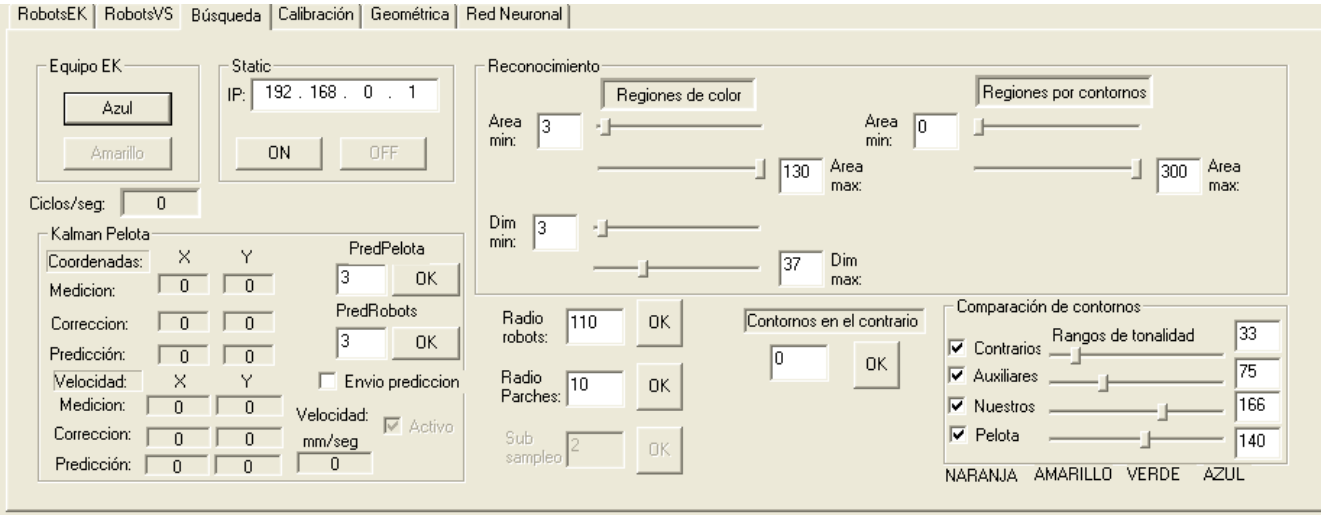


Figura 19: Búsqueda de objetos

Al seleccionar el tabulador "**Búsqueda**" se despliega una serie de controles que sirven para ajustar los parámetros de búsqueda. En primer lugar se define el color con el que juegan los robots en "**Equipo EK**". Una vez realizada la segmentación de la imagen se utiliza el algoritmo RLE para posteriormente poder formar regiones que representen a los parches y a la pelota. El tamaño es aproximadamente el mismo por lo que se restringe el área que ocupan las regiones de interés (principalmente el área mínima) para eliminar ruido, además se validan la dimensión mínima en los ejes "x" y "y" que miden las regiones de interés. Estos ajustes se hacen con los siguientes controles:

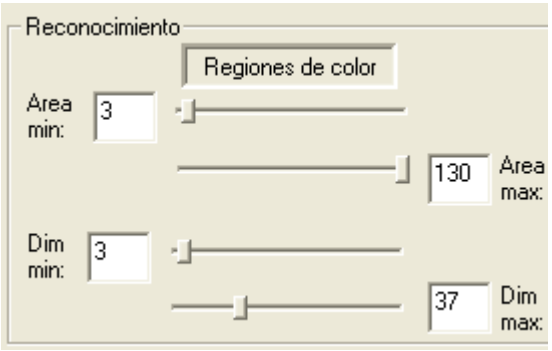


Figura 20: Validación de regiones

Es importante también definir el radio de los robots y el radio de los parches los cuales varían dependiendo de la altura de las cámaras:

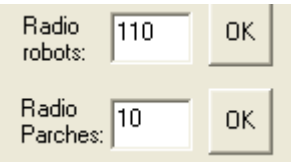


Figura 21: Parámetros de búsqueda

Información de los robots y la pelota

La información del resultado de la búsqueda de la pelota se encuentra en el tabulador "**Búsqueda**". El sistema de visión obtiene la coordenada real, y el resultado del filtro de Kalman: corrección y predicción. Además de la posición se obtiene también la velocidad de movimiento (Vx y Vy) y el resultado del filtro de Kalman. El resultado de la predicción puede ser obtenido en función del número de cuadros que se seleccionen.

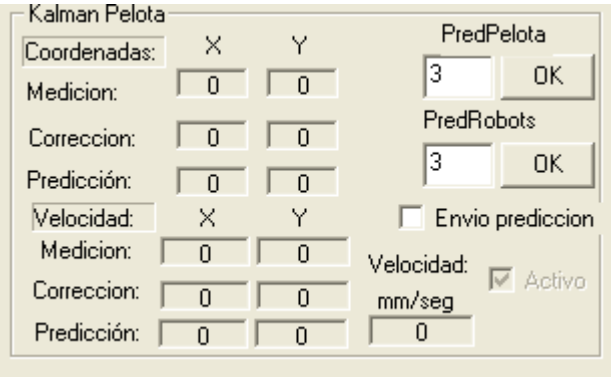


Figura 22: Filtro de Kalman para la pelota

La información de los robots se encuentra en el tabulador "**RobotsEK"** para nuestro equipo y "**RobotsVS"** para los contrarios. La funcionalidad de ambas ventanas es la misma excepto porque no se conoce la orientación de los robots contrarios.
Los principales datos que se obtienen son: la coordenada, la orientación del robot y la velocidad de movimiento. En esta interfase se activa el control "**Encontró**" si el sistema pudo identificar correctamente a los robots, muestra también la configuración de parches utilizada por cada uno de los robots

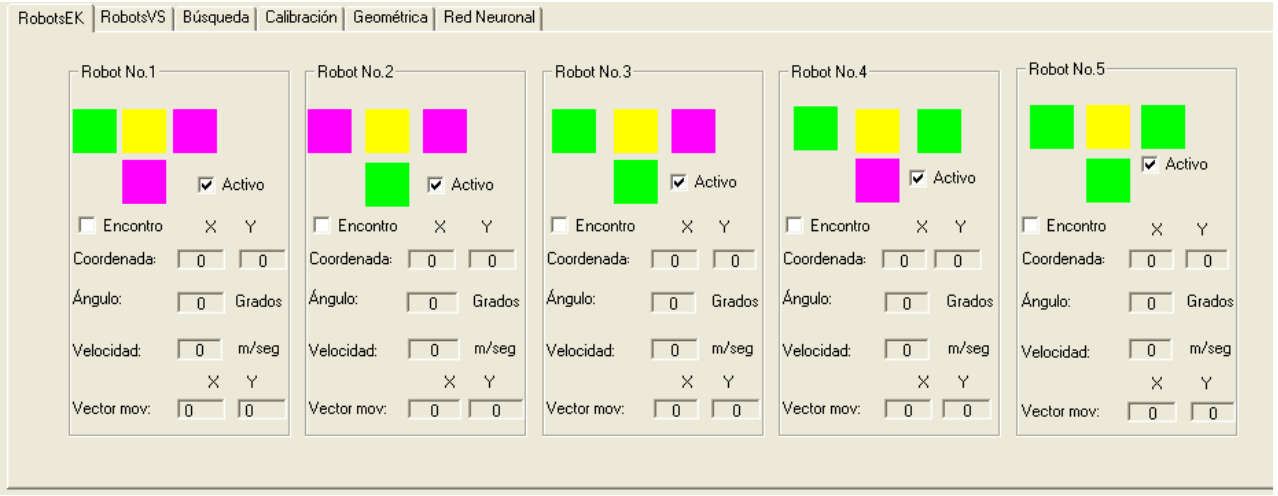


Figura 23: Información de los robots

Comunicación EKVision y EKIntel

La comunicación entre el sistema EKVision y el sistema EKIntel se realiza por medio de una comunicación directa en red. Generalmente cada sistema se ejecuta en una computadora diferente por lo que se conectan por medio de un cable cruzado (pueden

conectarse utilizando un *acces point*, pero para evitar problemas de es mejor la conexión por *ethernet*).
La configuración de las direcciones de cada una de las computadoras debe ser las siguientes:

Computadora	**Dir IP**	**Máscara de Red**	**Socket**
EKVision	192.168.0.2	255.255.255.0	5001
EKIntelSSL	192.168.0.1	255.255.255.0	5001

Tabla 2: Comunicación EKVision y EKIntel

La dirección IP y la máscara de red deben configurarse seleccionando el puerto *ethernet* que se va a utilizar en las "Conexiones de red" y posteriormente en propiedades de "Protocolo Internet (TCP/IP)

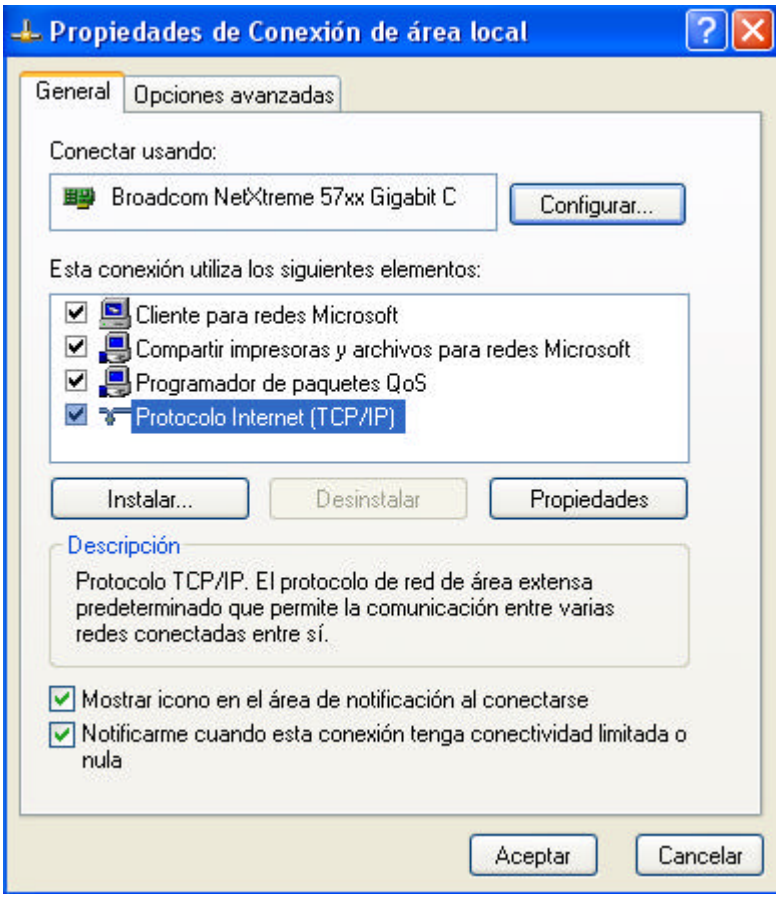


Figura 24: Configuración de la dirIP y la máscara de Red

Sistema de Inteligencia Artifical

Hardware y conexión con periféricos

El Sistema de inteligencia artificial utiliza diversas conexiones con dispositivos externos. La primera conexión se realiza con el sistema de visión. La configuración de esta conexión se explicó en el apartado "Comunicación EKVision y EKIntel".

La segunda conexión se realiza por medio del puerto paralelo con la base del *transceiver* la cual sirve para comunicar al sistema con los robots utilizando un cable de impresora, la base debe estar alimentada por medio de un transformador de 9volts:

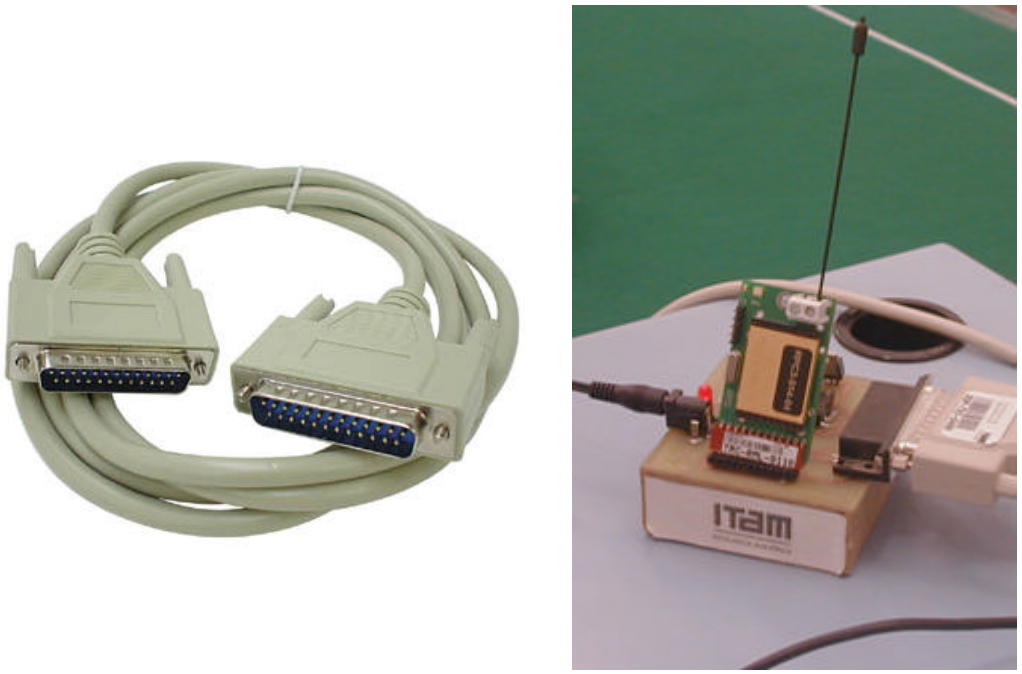


Figura 25: Base del transceiver

La tercera conexión se realiza con el "*Referee box*" por medio de una conexión LAN por *ethernet*. Debido a que la laptop Dell M6300 solo tiene un puerto *ethernet* se utiliza el siguiente convertidor *USB-Ethernet*:

Figura 26: Convertidor USB-Ethernet

La configuración de la dirección IP y el Socket UDP se puede encontrar en la página dedicada al Referee Box en el sitio oficial de la liga *Small Size*[4].

Anteriormente se utilizaba una interfase serial para recibir los comandos por lo que esa funcionalidad se mantiene y se realizaba utilizando un convertidor USB-Serial.

Finalmente el sistema EKIntel utiliza un *joystick* para probar el control de los robots. Es posible conectar cualquier tipo pero la respuesta a la sensibilidad del *joystick* y los botones a utilizarse están programados para el modelo inalámbrico *"Logitech® Freedom™ 2.4 Cordless Joystick"* que se conecta por un puerto USB:

Figura 27: Joystick inalámbrico

El sistema EKIntel puede trabajar sin necesidad de tener todos los periféricos conectados. Sin embargo algunas funcionalidades se ven restringidas si no se activan estos dispositivos.

Instalación del software de Inteligencia artificial

El sistema de inteligencia artificial consiste en un solo software "EKIntel" el cual es programado sobre MFC de Microsoft Visual C++ 6.0. El programa puede estar ubicado en cualquier carpeta del sistema y utiliza archivos de texto de la carpeta "C:\Robotica".

a) Configuración del puerto paralelo para el uso del transceiver.

1. Asegurarse de que el archivo io.dll ese encuentre en la carpeta Debug del proyecto.
2. Entrar al menú del BIOS y configurar el puerto paralelo de la siguiente forma:

Modo: ECP
Base I/O Address: 378
Intemp: IRQ 7
DMA: DMA0

(Las direcciones de memoria de cada línea del puerto se establecen a partir de la 0x379.)

b) Configuración de gráficas con OpenGl

1. Descargar los siguientes archivos: glut.h, glut.lib, glut.dll y guardarlos en las siguientes carpetas:

glut.h	C:\Program Files\Microsoft Visual Studio\VC98\Include\GL
glut.lib	C:\Program Files\Microsoft Visual Studio\VC98\Lib
glut.dll	*C:\WINDOWS\system32*

2. Dentro del proyecto en visual c++ ir a:

Project/Settings/Link
Añadir los siguientes archivos en la parte de Object/library modules

opengl32.lib glu32.lib glut32.lib

(el último no se utiliza en el proyecto pero es conveniente para otras aplicaciones de gráficas en opengl)

c) Configuración del uso del joystick con DirectX

1. Descargar el paquete completo de DirectX 9.0 Sdk e instalarlo (no solo un update sino todo el paquete)
2. Dentro del proyecto en visual c++ ir a:

Project/Settings/Link

Añadir el siguiente archivo en la parte de Object/library modules

dinput8.lib

Finalmente es necesario añadir las siguientes rutas para los directorios que se usarán en el siguiente orden:

Carpeta de directX

C:\DXSDK\INCLUDE

Arhivos generales de visual c++ (por default)

C:\Archivos de programa\Microsoft Visual Studio\VC98\INCLUDE
C:\Archivos de programa\Microsoft Visual Studio\VC98\MFC\INCLUDE
C:\Archivos de programa\Microsoft Visual Studio\VC98\ATL\INCLUDE

Carpeta de opengl

C:\ARCHIVOS DE PROGRAMA\MICROSOFT VISUAL STUDIO\VC98\INCLUDE\GL

d) Uso de otras fuentes en la interfase gráfica del programa.

1. Descargar el tipo de letra deseado en la carpeta:

C:\WINDOWS\Fonts

Sistema EKIntel

El sistema de inteligencia artificial es un software que tiene múltiples funcionalidades. El menú principal permite seleccionar que periféricos se desean utilizar. Los controles de "**Comunicación**" se utilizan para activar la recepción de información del sistema de visión la recepción de comandos del *referee*, el uso del *joystick* para control de los robots y el uso del *transceiver* para comunicarse con los robots.

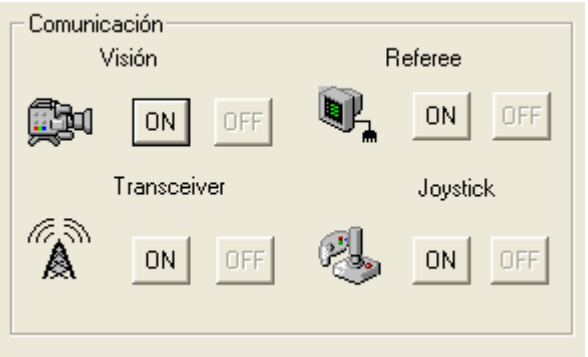


Figura 28: Control de comunicación

En caso de que alguno de los dispositivos de conexión no esté correctamente conectado e instalado el sistema no funcionará apropiadamente.

Otros controles que se encuentran en la ventana principal son los que muestran la información principal acerca del funcionamiento del sistema:

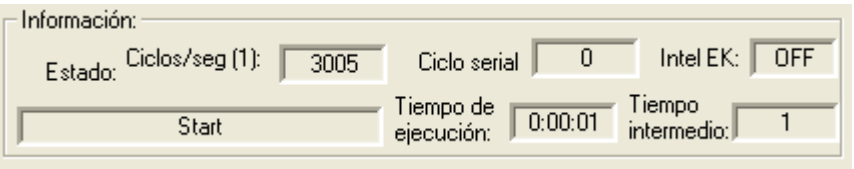


Figura 29: Información del Sistema

Dentro de este recuadro se informa acerca de la velocidad de ejecución en Ciclos/Seg, los ciclos por segundo del *thread* utilizado para la lectura de comandos por medio del puerto serial, el estado de juego actual, el tiempo de ejecución del sistema y el número de tiempo intermedio, además de indicar si el control "**IntelEK**" está encendido.

Un tercer recuadro informa acerca del estatus del *josytick*, es decir, que número de robot se controla y de que equipo (el número de robot se puede modificar por medio del *slider* del *joystick*).

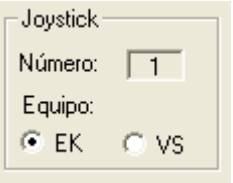


Figura 30: Configuración del joysitick

Finalmente se muestra un cuarto recuadro en donde se encuentran los controles generales para elegir el color de los robots y el sentido en el que juegan (esto ocasione que el color de las porterías cambie automáticamente). También se incluye el control de simulación.

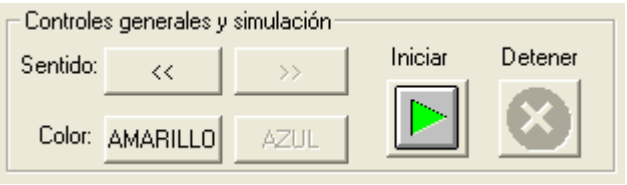


Figura 31: Controles generales

La configuración de estos controles (sentido y color) se leen y guardan automáticamente al iniciar y cerrar el sistema en la ruta "**C:\Robotica\Controles**" en el archivo "Controles.txt"

El sistema de referencia utilizado en el Sistema EKIntel es distinto al utilizado en EKVision y se muestra en la siguiente imagen (todas las mediciones se manejan en mm):

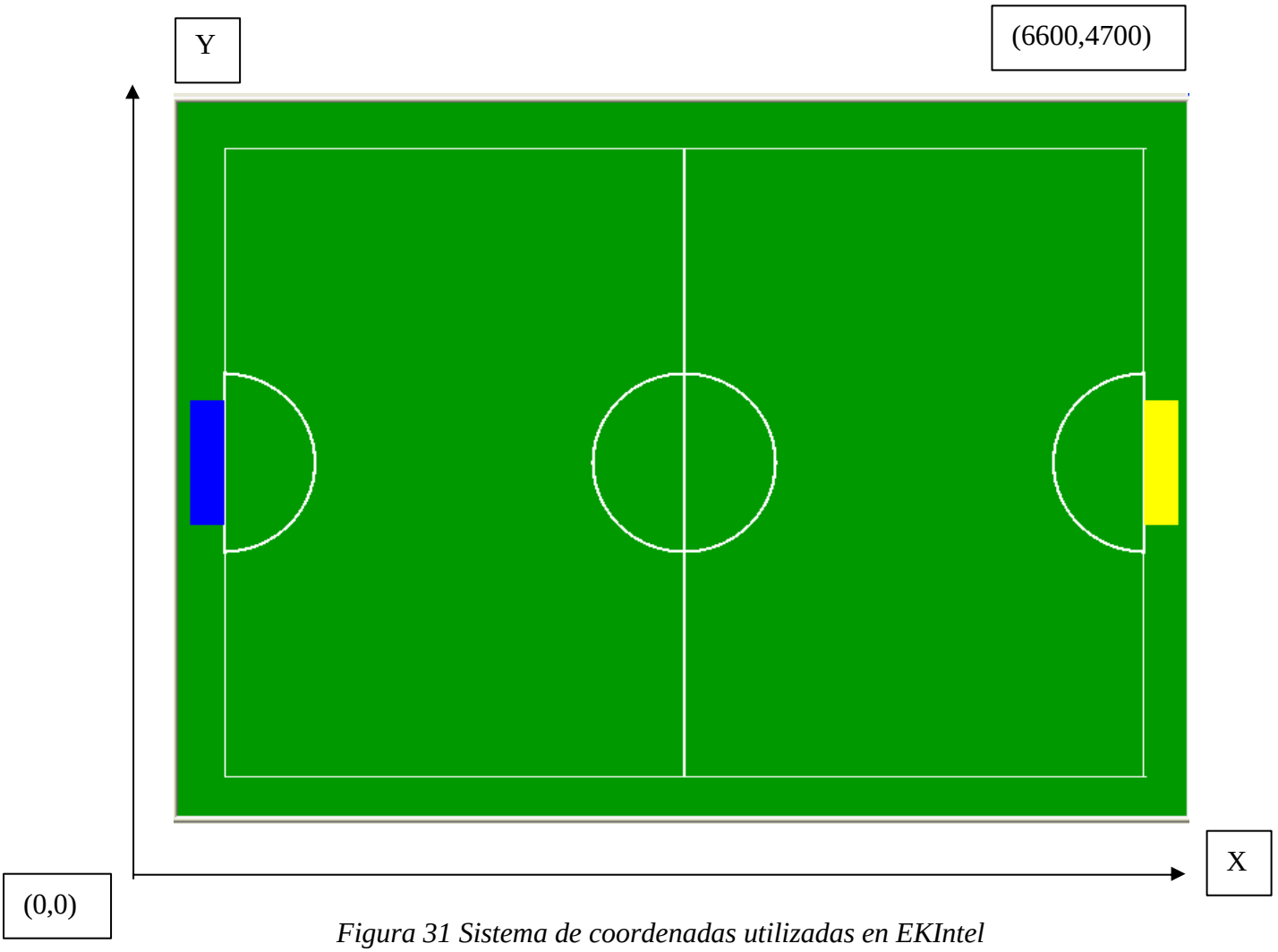


Figura 31 Sistema de coordenadas utilizadas en EKIntel

Además de los controles principales y de la representación digital del campo de juego el sistema tiene ocho tabuladores para controlar y configurar su funcionamiento.

- **Robots EK y contrarios**

Existen dos controles para controlar a diez robots. Aunque en un partido se controlan únicamente a los cinco de un equipo, es muy útil poder manipular a robots de ambos equipos al momento de programar los comportamientos de cada robot (Ej 2 contra 3, 2 contra 2). Además en caso de contar con más de cinco robots, con sólo aumentar el número de tramas que se envían en la comunicación con los robots, el sistema puede controlar hasta 10 robots jugando 5 contra 5.

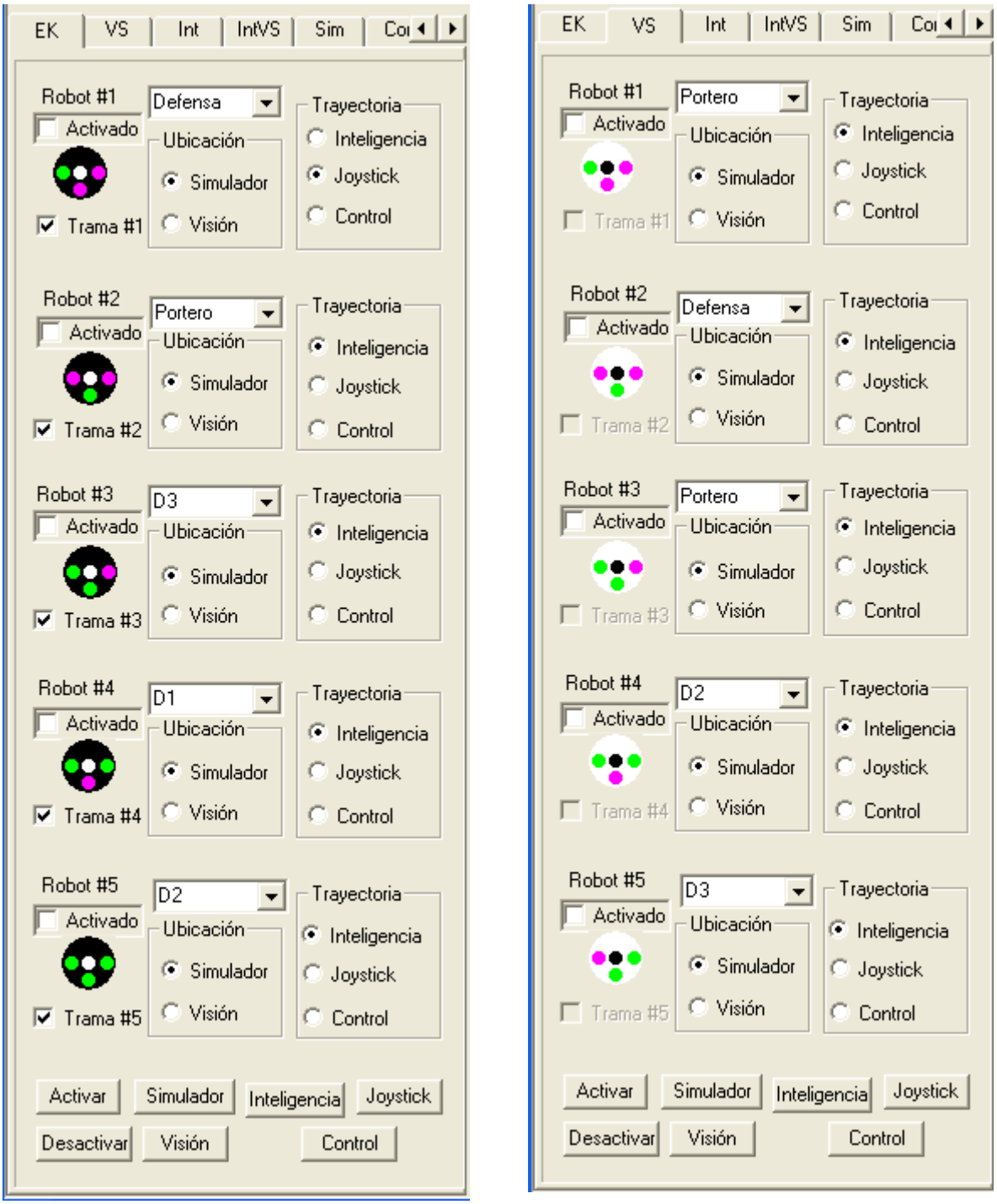


Figura 32: Configuración de robots

Cada uno de los robots contiene la siguiente información:

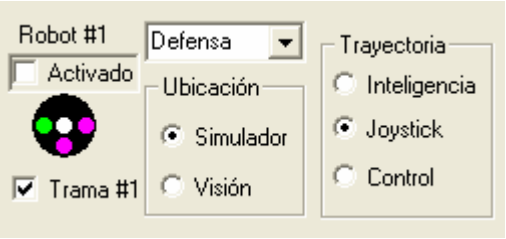


Figura 33: Controles por robot

Se indica el número de robot y el equipo (en caso de tener la cubierta negra significa que es nuestro y la cubierta blanca que es contrario). La imagen del robot muestra también el código de colores utilizado en la identificación del sistema de visión. Debajo del número de robot se puede activar o desactivar con la finalidad de no intentar controlarlo en caso de que el robot no esté encendido o jugando en ese momento por medio del *checkbox* "**Activado"**. Finalmente en el último control de la primera columna es el *checkbox* "**Trama #**" el cual es útil para decidir si el número de trama correspondiente al robot será la del robot nuestro o la del robot contrario. En la segunda columna se cuenta con un *combo-box* con el que se define el rol que va a tomar el robot. Abajo en el recuadro **Ubicación** se indica si en cada iteración la posición actual del robot se obtiene por medio del sistema de visión o por el simulador. Finalmente el cuadro **Trayectoria** define la forma en que se va a controlar el robot: Inteligencia, por medio del *joystick* o por la interfase de control. En caso de controlarse pode medio del *joystick*, este debe configurarse en el menú principal (descrito en la Figura 30), si se utiliza la interfase de control se tiene que usar el tabulador "Con" que se explicará más adelante y si se quiere controlar por el algoritmo de inteligencia artificial se utiliza alguno de los tabuladores **Int** o **IntVS** dependiendo del equipo al que corresponda el robot.

Los botones que se encuentran hasta debajo de la ventana permiten realizar una selección rápida que se aplique a todos los robots para los controles de activación l ubicación y trayectoria.

Toda la información correspondiente a cada uno de los robots se guarda y se carga automáticamente en los archivos de texto "RobotsEK.txt" y "RobotsVS.txt" ubicados en "C:\Robotica\Robots" en donde se incluye otra información como la forma en que se actualizan las velocidades lineal, de giro y de pateo en caso de simular el movimiento del robot. Esto se explicará en siguiente parte.

- **Simulador y control**

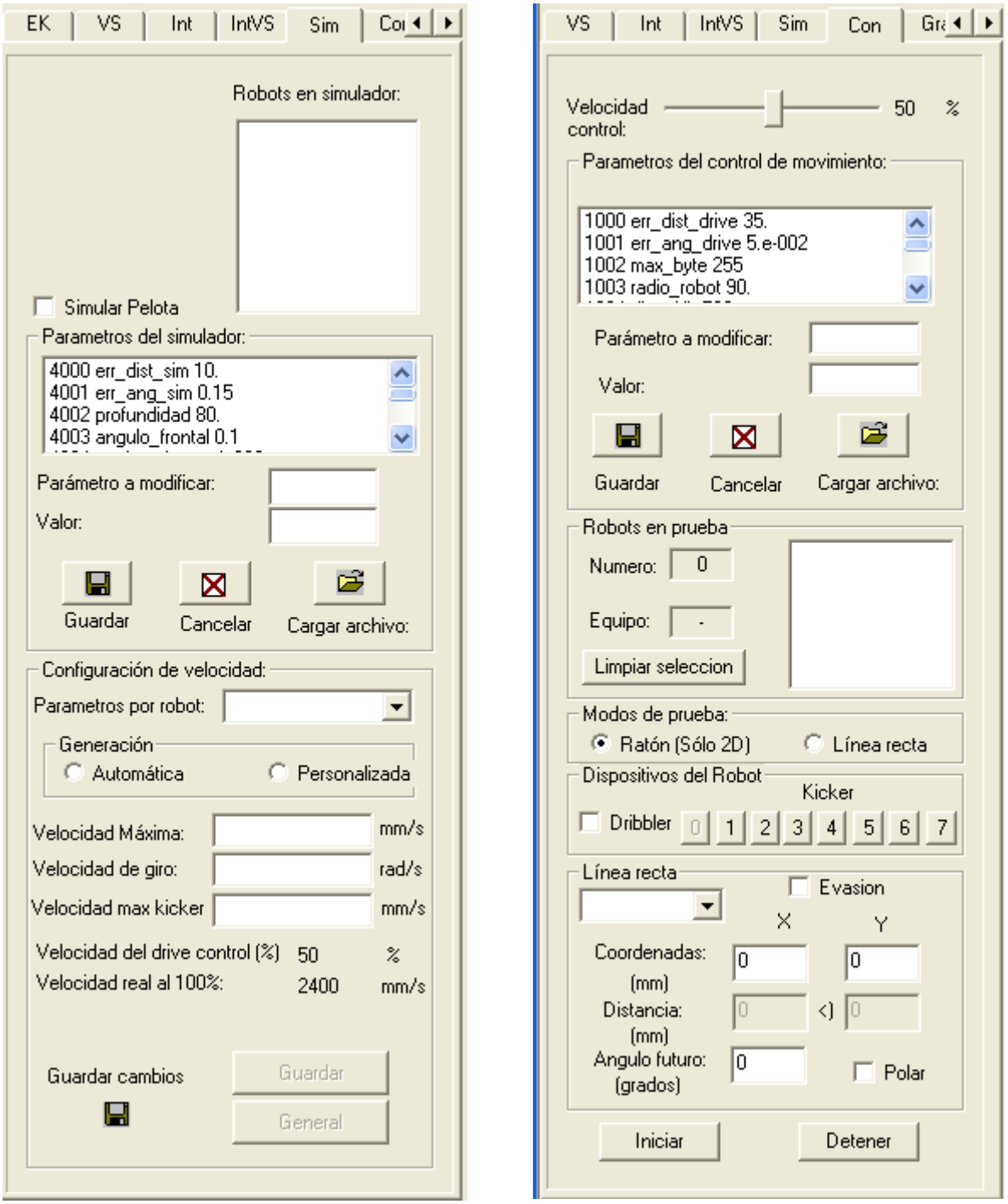


Figura 34: Configuración del simulador y del control de los robots

Simulador

La ventana de Simulación permite modificar cualquiera de los parámetros utilizados dentro del simulador de los robots Todos tienen un identificador que define el tipo de parámetro (40 para simulador) y posteriormente el número de parámetro.

A continuación se describen algunos de los utilizados actualmente:

err_dist_sim: Rango utilizado para decidir si un robot ha llegado a su posición final.
err_ang_sim: Rango utilizado para decidir si un robot ha llegado a su orientación final.
profundidad: Rango utilizado para decidir si se puede manipular la pelota con el kicker y el dribbler.
angulo_frontal: Ángulo de apertura para definir si la pelota está en frente del robot
aceleracion_pel: Desaceleración de la pelota por la fricción de la alfombra
tiempo: Periodo que define la velocidad de simulación.
elasticidad: Entre [0,1] define el porcentaje de velocidad que mantiene la pelota después de chocar con algún robot o portería.
vel_50%: Velocidad de simulación cuando la velocidad de control está al 50%
giro_50%: Velocidad de giro cuando la velocidad de control está al 50%

Es importante aclarar que el simulador utiliza no sólo los parámetros descritos en esta lista sino también algunos parámetros de la lista de control.

Estos parámetros se pueden modificar desde la interfase del sistema. Cada vez que se guarda algún cambio, al cerrar el programa, se actualiza el archivo "Simulador.txt" ubicado en "C:\Robotica\Simulador". También es posible cargar archivos con otros nombres utilizando el botón "**Cargar Archivo**". El número total de parámetros y el nombre puede modificarse siempre y cuando sea congruente con lo utilizado dentro del código del simulador.

Además de los parámetros generales está la "**Configuración de velocidad**" que permite ajustar la velocidad de cada uno de los robots por separado. El *combo-box* llamado "**Parámetros por robot**" permite seleccionar a alguno de los diez robots. Existen dos formas de ajustar las tres velocidades utilizadas en el simulador: máxima, de giro y de pateo. La primera es la generación "**Personalizada**" en donde cada uno de los valores se ingresa directamente en el sistema. La generación "**Automática**" permite ajustar las tres velocidades utilizando el *slider* de "**Velocidad control**" que se encuentra en el tabulador de control. De esta forma la relación entre velocidad real (mm por segundo) y el porcentaje definido por el *slider* se realiza por los parámetros de simulación vel_50% y giro_50% como se muestra al final de la ventana ("**Velocidad del dirve control (50%)"** y **"2Velocidad real 100%"**). La velocidad de pateo no se modifica de forma automática.

Estos cambios se pueden guardar y se abren automáticamente por medio de los archivos "RobotsEK.txt" y "RobotsVS.txt" en la ruta "C:\Robotica\Robots" junto con otros atributos de los robots descritos en la sección anterior.

Control

La ventana de control tiene dos finalidades.

En primer lugar permite configurar los parámetros que son necesarios para controlar a los robots cuando estos se controlan de forma autónoma cuando todo el sistema está conectado (visión y comunicación con los robots). Estos parámetros son utilizados tanto para realizar diferentes comportamientos como para definir la velocidad a la que debe ir

el robot y el valor que se envía por el transceiver. Además permite ajustar ciertos parámetros utilizados cuando se desea controlar a los robots por medio del *joystick*

Los parámetros tienen un identificador inicial de dos números (igual que ocurre con la lista de parámetros del simulador) en donde (10) se usa para los parámetros de control, (20) para los parámetros de inteligencia artificial, (30) para los parámetros del joystick. Dependiendo de cómo son programados los comportamientos de IA, algunos requieren leer algún parámetro de esta listo y otros manejan internamente sus parámetros por lo que a continuación se explicarán los parámetros de control y del joystick.

Parámetros de control

Los parámetros utilizados actualmente para el control de los robots son (esta lista puede aumentar o ser modificada dependiendo de la evolución del sistema):

err_dist_drive: Margen para decidir si un robot ha llegado a su posición final (considerando cierto error inducido por el sistema de visión).
err_ang_dirve: Margen para decidir si un robot ha llegado a su orientación final
max_byte: Máximo valor que puede tomar un dato que se va a enviar por el transceiver
radio_robot: Radio del robot (puede variar dependiendo de la altura de la cámara)
dist_ref: Distancia que debe moverse la pelota a partir de un punto de referencia para cambiar automáticamente de estado de juego.
err_radio: Parámetro de error para definir un cambio de estado por cercanía a un punto
dist_fren: Distancia de frenado y aceleración para la velocidad lineal
ang_fren: Ángulo de frenado para la velocidad de giro

Los parámetros de control se guardan y se abren automáticamente en el archivo de texto "ControlMovimiento.txt" en la ruta "**C:\Robotica\Control**".

Parámetros del joystick

ang_sum_joy: Ángulo que se aumenta al ángulo actual para definir la orientación final por medio del joystick
division_ejes: Número de niveles de velocidad que puede tener cada eje del joystick
rango_giro: Rango de sensibilidad para anular el movimiento de giro del joystick
sensibilidad: Rango de sensibilidad para anular el movimiento del joystick

Los parámetros de control se guardan y se abren automáticamente en el archivo de texto "Joystick.txt" en la ruta "**C:\Robotica\Joystick**".

En segundo lugar la ventana de control permite realizar movimientos sencillos y precisos que son útiles al probar el movimiento omnidireccional del robot. Es necesario seleccionar de la lista "**Robots en prueba**" (aparecen los robots activos en modo de control configurados individualmente para cada robot) al robot que se desea probar dandole doble click. Una vez seleccionado el robot se establece el método de prueba: "**Ratón**" y "**Línea Recta**".

En el método de "**Ratón**" se puede dar click automáticamente a cualquier punto de la cancha para definir la coordenada a la que se desea mover el robot son modificar su orientación.

En la "**Línea Recta**" se define el vector de movimiento de forma polar o cartesiana (*checkbox* **polar**). En caso de estar activo se escribe la distancia y el ángulo en el que se quiere mover al robot, así como la orientación final. En caso de que no esté activa se escribe cualquier coordenada del campo de juego. Para iniciar el movimiento o detenerlo se utilizan los botones **Iniciar** y **Detener** en ambos métodos de prueba.

Es posible también probar el funcionamiento del *dribbler* y del *kicker* por medio de los controles de "**Dispositivos del Robot**". El dribbler se controla por medio del checkbox "**Dribbler**" y existen siete posibles niveles de pateo "**Kicker**".

Dentro de los métodos de prueba se puede activar o desactivar la evasión de obstáculos con el *checkbox* "**Evasión**"

El valor de la velocidad del *slider* y el modo de prueba se mantienen en el archivo "ControlMovimientoInicializacion.txt" en la ruta "**C:\Robotica\Controles**"

- **Gráficas**

El sistema EKIntel utiliza una representación gráfica del campo de juego la cual está programada utilizando *OpenGL*. La visualización puede ser en dos o en tres dimensiones, esto se ajusta con los controles "**Tipo de gráficas**". Es posible también desactivar la visualización para saber el retraso que pudieran estar metiendo al sistema completo.

Dentro del cuadro de opciones se seleccionan los atributos que se quieren mostrar en el campo de juego:

Vector de movimiento: El vector de movimiento del robot se compone de una línea que indica la trayectoria en línea recta que debe seguir y un circulo que representa la orientación que debe tomar el robot

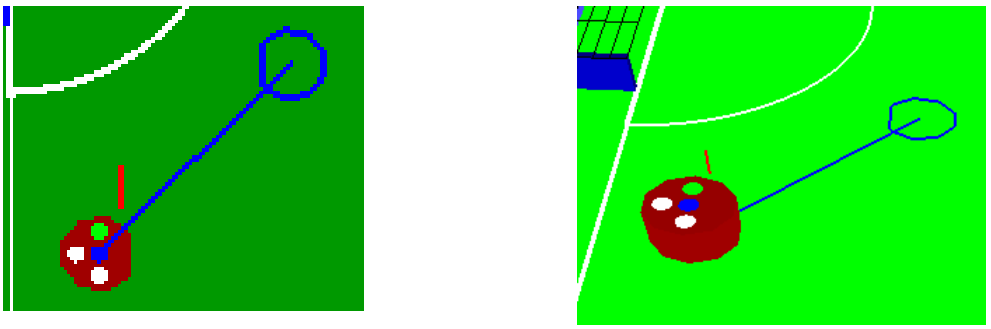


Figura 35: Vector de movimiento

Ángulo actual: 0 grados
Ángulo final: 180 grados
Ángulo de movimiento: 45 grados
Distancia a moverse: 600 mm

Potencia del *kicker*: La potencia de pateo se indica con variaciones entre los colores amarillo y rojo en donde amarillo es el nivel 1 y rojo es el nivel 7

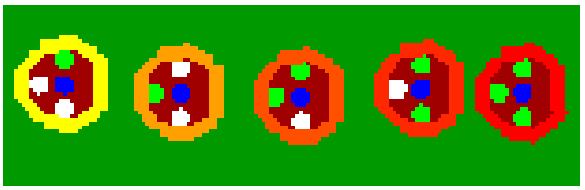


Figura 36: Potencias de pateo (niveles 1,3,5,6 y 7)

Uso del dribbler: La activación del dirbbler se dibuja con un círculo morado:

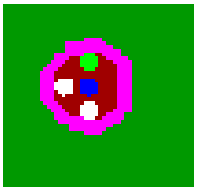


Figura 37: Activación del dribbler

Árbol de evasión: El árbol creado para la evasión de obstáculos es un conjjnto de puntos y líneas que describen los nodos y las extensiones:

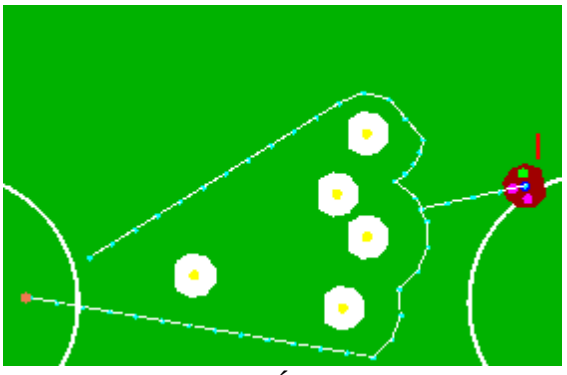
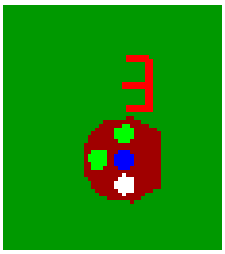


Figura 38: Árbol de evasión

Número de robot:



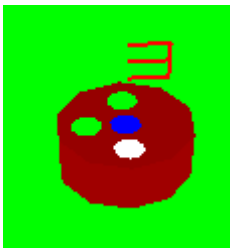


Figura 39: Número del robot

Trayectorias: Con esta opción se puede ver los últimos n puntos por los que ha pasado el robot. Es útil para saber si el robot se está moviendo adecuadamente y para ver la distancia que hay entre cada medición del sistema EKVision.

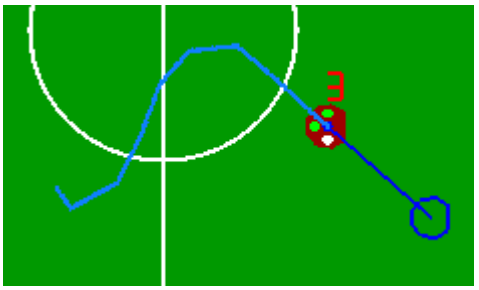


Figura 40: Trayectoria del robot

Pelota: La pelota se dibuja como un círculo naranja (o esfera en tres dimensiones). Además se dibuja una línea que representa la predicción que entrega el sistema de visión por medio del filtro de Kalman o el obtenido por el simulador.

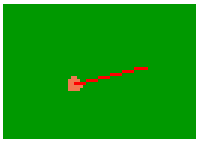


Figura 41: Predicción de la pelota

El control de la cámara en el modo de tres dimensiones se realiza seleccionando la cancha y utilizando las siguientes teclas:

Flecha arriba: Avanzar hacia adelante
Flecha abajo: Avanzar hacia atras
Flecha derecha: Girar a la derecha
Flecha izquierda: Girar a la izquierda
Re Pag: Girar hacia abajo
Av Pag: Girar hacia arriba

Dentro del cuadro de "**Información**" se puede ver toda la información de los diez robots, así como de la pelota. Esta. Esta información es la que se utiliza para controlar a los robots y se escribe en la pantalla antes de ser enviada por el transceiver. Se muestra la posición actual, la posición futura, el nivel de pateo, el uso del dribbler, la velocidad de movimiento (descompuesta en cada eje), la velocidad de giro, la velocidad de movimiento lineal, la velocidad medida por el sistema de visión por robot y la posición de la pelota.

Es importante aclarar que los valores de las velocidades del robot que se muestran en esta pantalla son afectados por el modelo de aceleración y desaceleración del robot y por las evaluaciones realizadas para poder controlar al robot.

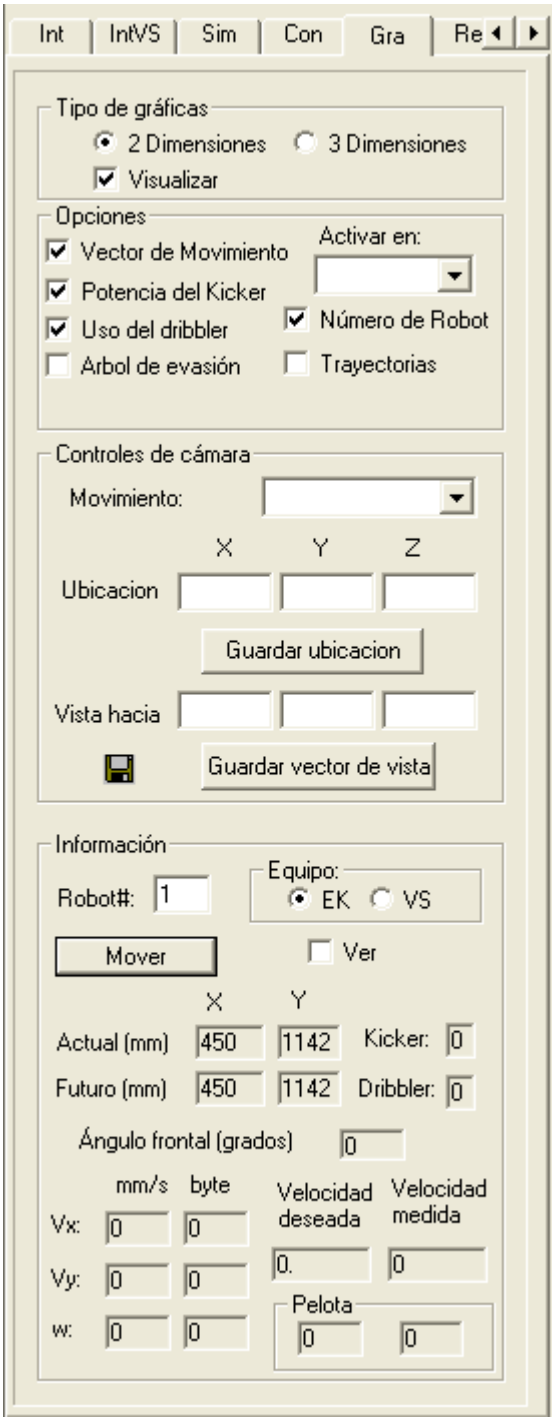


Figura 42: Configuración de las gráficas en OpenGL

La visualización del campo entero en dos dimensiones se puede ver en la figura 32. A continuación se muestra una imagen tomada del entorno en tres dimensiones. Dentro del entorno se dibujan las paredes, las cámaras, las porterías, el borde de la cancha, los robots y la pelota.

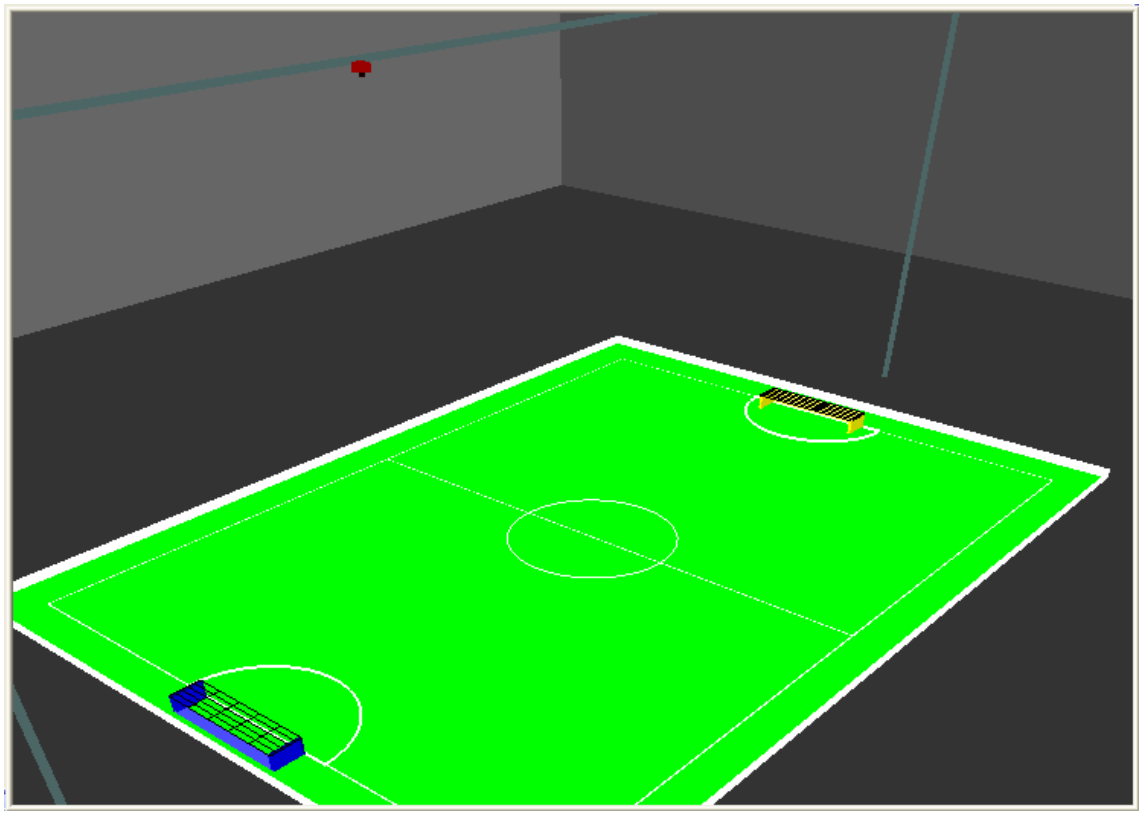


Figura 43: Campo de juego en tres dimensiones

Las dimensiones del campo de juego utilizadas dentro de todo el sistema se leen automáticamente al iniciar el sistema en la ruta "**C:\Robotica\Cancha**".

La configuración de las opciones utilizadas en esta parte se guardan y leen automáticamente al terminar e iniciar el sistema respectivamente en la ruta: "**C:\Robotica\Controles**" en el archivo "Graficas.txt"

- **Referee**

El tabulador del referee "**Ref**" muestra una ventana que permite generar comandos en caso de que el Referee-box no esté conectado a la computadora. El uso del árbitro es indispensable para la programación de los comportamientos en la IA.

A continuación se muestran los comandos con su interpretación

```
Command type    Command Description     Command
================================================================================
Control commands
        Halt                            H
        Stop                            S
        Ready                           ' '
        Start                           s
Game Notifications
        Begin first half                1
        Begin half time                 h
        Begin second half               2
        Begin overtime half 1           o
        Begin overtime half 2           O
        Begin penalty shootout          a

Command type    Command Description     Yellow Team Command     Blue Team Command
================================================================================
Game restarts
        Kick off                        k                       K
        Penalty                         p                       P
        Direct Free kick                f                       F
        Indirect Free kick              i                       I
Extras
        Timeout                         t                       T
        Timeout end                     z                       z
        Goal scored                     g                       G
        decrease Goal score             d                       D
        Yellow Card                     y                       Y
        Red Card                        r                       R
        Cancel                          c
```

Figura 44: Protocolo el Referee

El sistema permite ingresar estos comandos y generar algunos automáticamente como combinación de dos botones consecutivos. Por ejemplo KickoffReady a partir de *Kickoff* seguido de *Ready.* La interpretación de todas las combinaciones de comandos para control y reinicio de juego son:

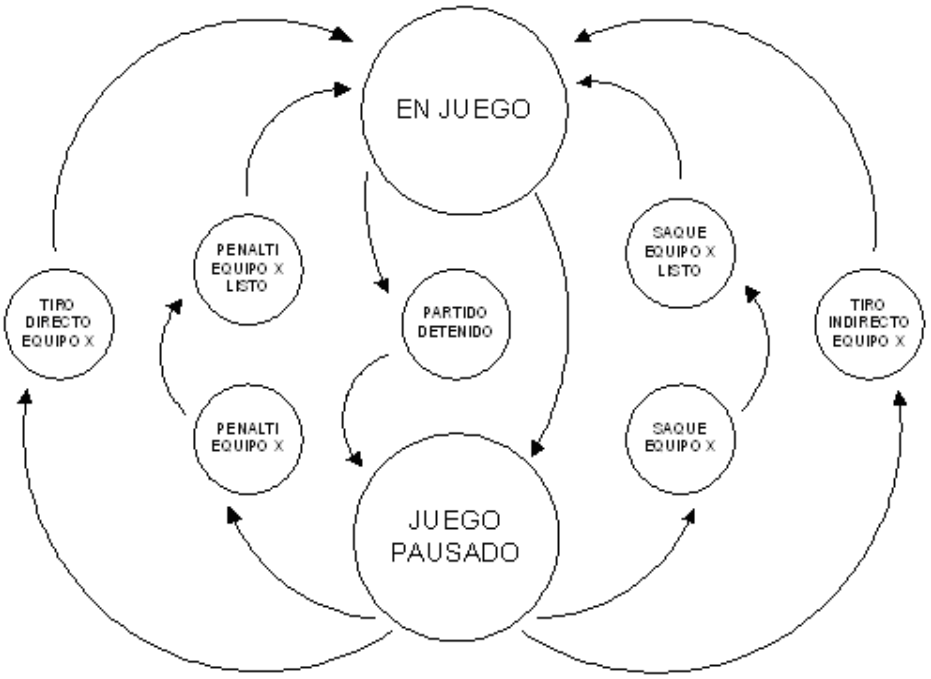


Figura 45: Diagrama de estados del Referee-bo

La ventana del Referee se muestra a continuación:

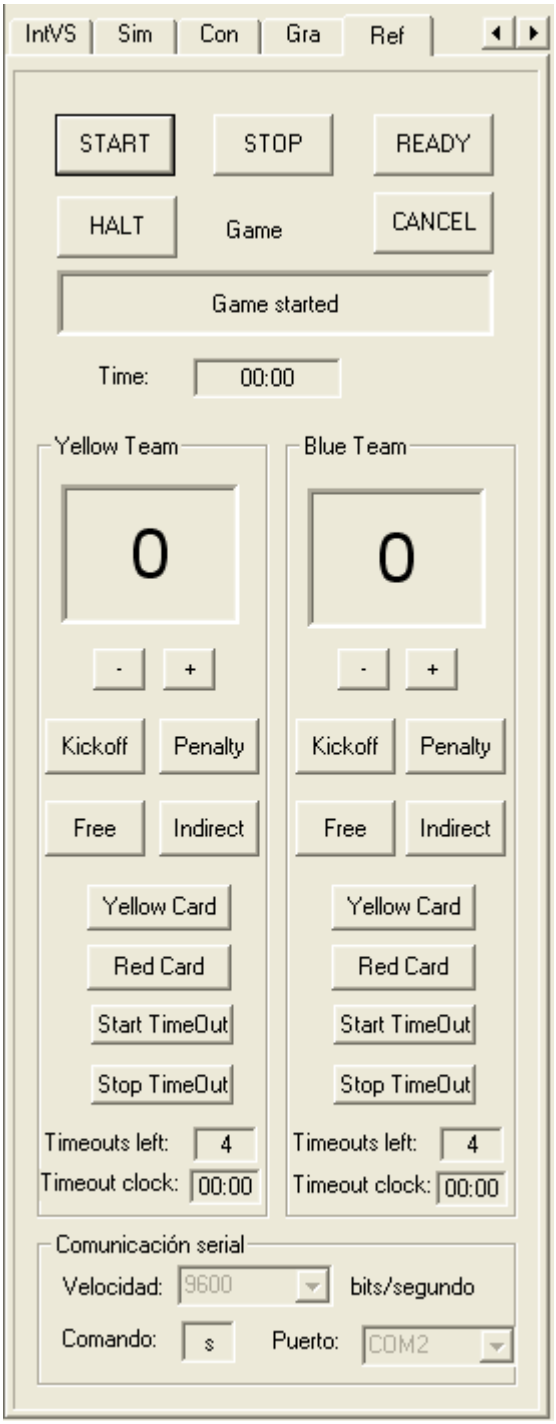


Figura 46: Referee integrado al sistema EKIntel

En la parte de Comunicación serial se muestra como debe configurarse la interfase: **velocidad** (baudios), **puerto** en el que está conectado el cable seria y último **comando** recibido.

- **Inteligencia EK y contrarios**

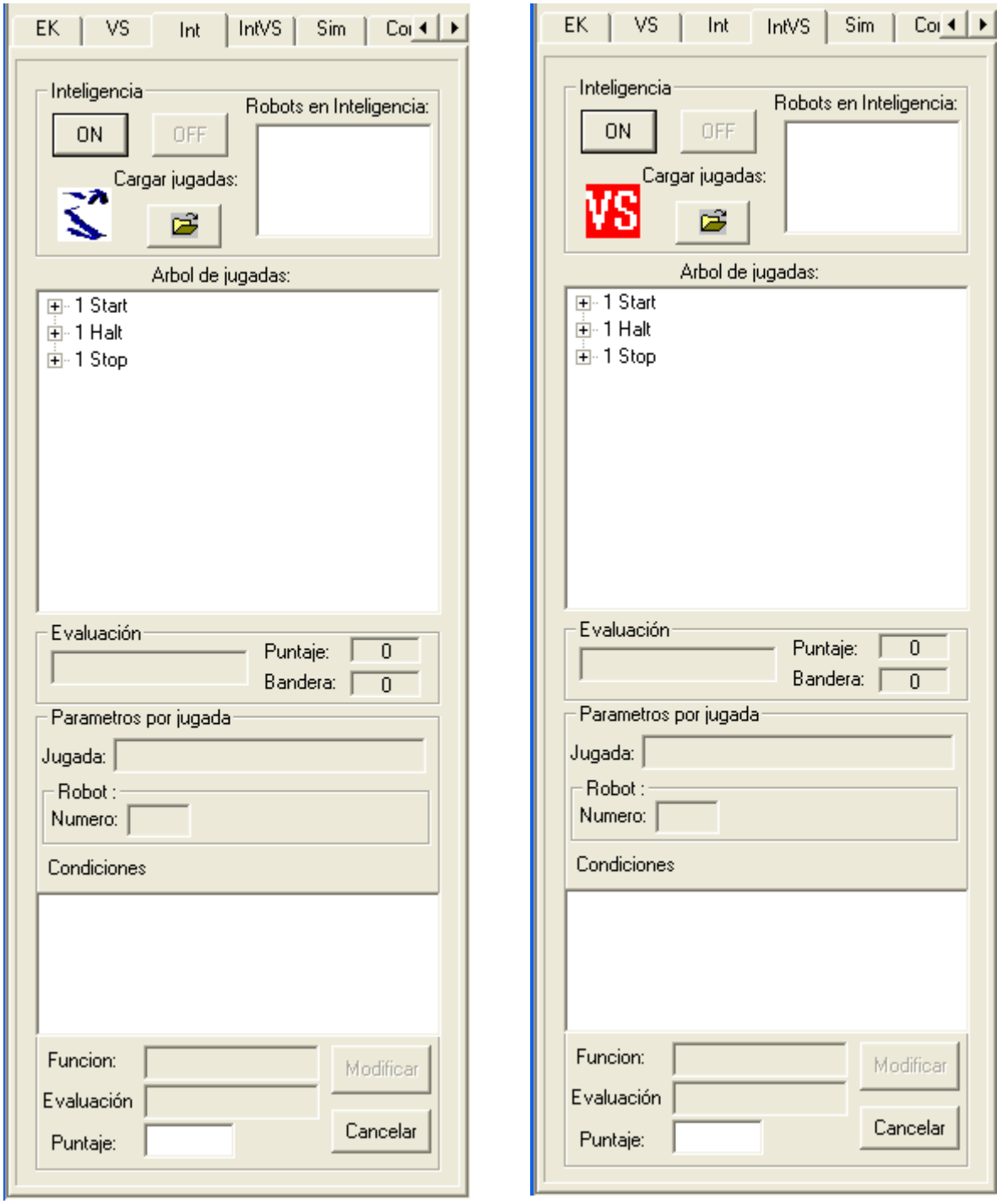


Figura 47: Configuración de la inteligencia artificial

La inteligencia artificial del sistema EKIntel es el núcleo del sistema y es el que permite que controlar a los robots de manera autónoma. El algoritmo utilizado para determinar los comportamientos de los robots y para planificar la ruta que debe tomar se explicará a detalle en otro documento.

El sistema EKIntel permite controlar de forma autónoma hasta diez robots. El control de cada equipo se maneja por separado con su propia ventana, pero tienen la misma funcionalidad.

Los controles de "**Inteligencia"** permiten activar el uso del algoritmo de IA y visualizar que robots están seleccionados en esta opción. Todo el árbol utilizado para definir los comportamientos se guarda en un archivo de texto en la ruta "**C:\Robotica\Inteligencia**". Con el nombre "ArbolEK[árbol a usar].txt".

Para que el sistema pueda generar el árbol requiere de una serie de archivos ubicados en la carpeta "**C:\Robotica\Inteligencia\JugadasEK**".

Una vez cargado el árbol que se va a utilizar se puede ver su jerarquía y el nombre de sus nodos en el control "**Árbol de Jugadas**". Al seleccionar alguna jugada se pueden ver los parámetros utilizados para su evaluación en el cuadro "**Condiciones**"

Referencias

1. Reglas de la liga SSL http://small-size.informatik.uni-bremen.de/rules:main
2. AVT http://www.alliedvisiontec.com/
3. Sitio del laboratorio de robótica ITAM http://robotica.itam.mx
4. Referee Box SSL http://small-size.informatik.uni-bremen.de/referee:start
5. Sitio official SSL http://small-size.informatik.uni-bremen.de/start
6. Sitio official RoboCup http://www.robocup.org/