

**INSTITUTO
TECNOLÓGICO
SUPERIOR DE
SAN ANDRÉS TUXTLA**



MATERIA: SIMULACIÓN DE SISTEMAS ROBÓTICOS

PROFESOR: DR. JOSÉ ÁNGEL NIEVES VASQUEZ

ALUMNO: EMMANUEL ROSARIO SOTO

CARRERA: MECATRÓNICA

GRUPO: 811-A

INVESTIGACIONES U2

SAN ANDRÉS TUXTLA, VERACRUZ

23 DE OCTUBRE 2024

CARACTERÍSTICAS Y APLICACIONES

23/10/24

La simulación fuera de línea, también conocida como simulación off-line, es un tipo de simulación que se realiza sin conexión a la planta real o al sistema físico. En este tipo de simulación, se utiliza un modelo matemático o computacional para simular el comportamiento de un sistema o proceso en un entorno virtual. Las características clave de esta incluyen:

- Independencia del sistema real
- Flexibilidad para realizar cambios y pruebas sin afectar el sistema real.
- Capacidad para simular escenarios extremos o peligrosos sin riesgos.
- Posibilidad de realizar análisis y optimizaciones sin interferir con el funcionamiento del sistema real.

La simulación fuera de línea tiene amplia gama de aplicaciones y funcionalidades en diversos campos.

SOFTWARE DE PROGRAMACIÓN

FUERA DE LINEA

Los software de programación fuera de línea permiten a los desarrolladores crear, depurar y probar código sin necesidad de conexión a internet o a un sistema real.

Estos programas ofrecen un entorno de desarrollo integrado (IDE) donde se pueden escribir, compilar y ejecutar programas en un entorno simulado. Algunos ejemplos populares de software de programación en fuera de línea incluyen:

- MPLAB X IDE (Microchip)
- Keil uVision (ARM)
- Code Composer Studio (Texas Instruments)
- IAR Embedded Workbench (IAR Systems)
- Proteus (Labcenter Electronics)

Los software de programación fuera de línea son esenciales para desarrolladores de sistemas embebidos, microcontroladores y aplicaciones de IOT.

CARGAR UN ENTORNO DE TRABAJO

Paso 1: Primero debe de seleccionarse un software de simulación adecuado para tu proyecto.

- Configuración del entorno de trabajo -

- Selección de micro controlador o plataforma objetivo
- Configuración de los parámetros de simulación (frecuencia, reloj, voltaje, etc.)
- Creación de un nuevo proyecto y selección del tipo de proyecto de aplicación, de biblioteca, etc.)
- Configuración de los archivos de código fuente y bibliotecas necesarios
- Establecimiento de los puntos de ruptura y configuración del depurador.

Cargar el entorno de trabajo y simulación.

INSERTAR HERRAMIENTAS DE TRABAJO



Para insertar herramientas de trabajo nos sirve a nosotros, a los ingenieros y diseñadores crear modelos realistas y analizar sistemas complejos de manera eficiente. A continuación, se presentan algunas herramientas comunes utilizadas en entornos de simulación:

- MATLAB/Simulink
- Autodesk Simulation
- ANSYS

Al incorporar estas herramientas en un entorno de simulación, los ingenieros pueden crear modelos precisos, analizar sistemas complejos y optimizar diseños de manera eficiente. Esto mejora la precisión del diseño, reduce costos y tiempo de desarrollo, y optimiza el rendimiento del sistema.

SIMULACIÓN DE MATERIALES DE ENTRADA

La simulación de materiales de entrada es un proceso que utiliza modelos matemáticos y herramientas de simulación para analizar y predecir el comportamiento de los materiales que se utilizan en un sistema o proceso. Esto puede incluir:

1. Simulación de propiedades físicas (resistencia, ductibilidad, conductividad).
2. Simulación de propiedades químicas
3. Simulación de comportamiento mecánico.
4. Simulación de comportamiento térmico

La simulación de materiales de entrada es una herramienta valiosa para los ingenieros y científicos de materiales, ya que permite predecir y optimizar el comportamiento de los materiales antes de su aplicación.

ESTRUCTURA DE SIMULACIÓN



La estructura de simulación se refiere a la organización y diseño de los componentes y proceso que intervienen en una simulación como el modelo de sistema, modelo de comportamiento y modelo de interfaz. Además, comprende componentes como entradas, procesos, salidas y controladores que regulan y monitorean la simulación.

La estructura lógica de una simulación puede ser secuencial, concurrente o jerárquica, y puede ser clasificada en tipos de simulación como continua, discreta o híbrida. Las herramientas de simulación utilizadas pueden variar desde software especializado como MATLAB y Simulink hasta lenguajes de programación como C++ y Python.

DESARROLLO DE SIMULACIÓN

El desarrollo de una simulación implica varios pasos para crear un modelo efectivo y eficiente. Primero, se define el problema y el objetivo de la simulación, identificando el sistema a simular y estableciendo requisitos y limitaciones. Luego, se diseña la simulación (continua, discreta, o híbrida) y las herramientas y software adecuados.

Se crea el modelo matemático del sistema, definiendo variables y parámetros y validando el modelo. Posteriormente, se implementa la simulación, codificando el modelo del software mencionado, configurando la simulación y realizando pruebas y depuración.

Finalmente, se optimiza y refina el modelo y la simulación, identificando áreas de mejora, ajustando parámetros y variables, y refinando el modelo.

Esto permite mejorar la precisión y eficiencia.

GUIA DE OBSERVACIÓN EXPOSICIÓN

Simulación de Sistemas Robóticos.



Nombre del estudiante: Rosario Soto Emmanuel.

Tema: Simulación de materiales de entrada.

Explicación	10 %	10 %
Dominio del tema	10 %	10 %
Avance del trabajo	5 %	5 %
Presentación en tiempo y forma	5 %	3 %
Total	30 %	28 %

SIMULACIÓN DE DATOS DE ENTRADA

Profesor: Dr. José Ángel Nieves Vasquez

Alumno: Emmanuel Rosario Soto

Materia: Simulación de sistemas robóticos



Objetivo de la simulación

Predicción de fallas y fracturas

Optimización de diseño y materiales

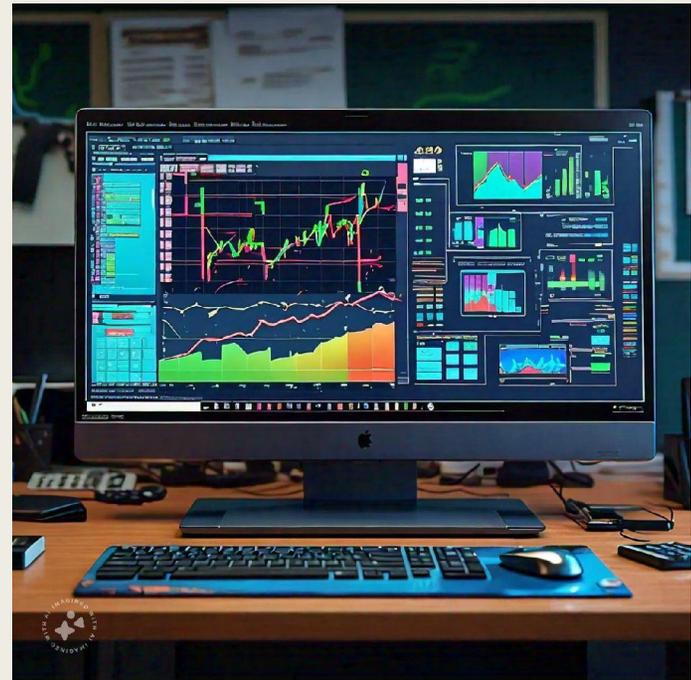
Reducción de costos y tiempo de desarrollo

Mejora de la seguridad

Desafíos

Interacción compleja entre componentes

Incertidumbre en materiales y condiciones



Simulación de Materiales de Entrada

Análisis de comportamiento de materiales bajo condiciones específicas



Aplicaciones en Robótica

Diseño de robots más eficientes
Optimización de rendimiento
Análisis de seguridad y fiabilidad



Desafíos a futuro

Integración de IA y aprendizaje automático

Simulación de sistemas complejos

Desarrollo de materiales avanzados

Aplicaciones en industrias emergentes

Estándares y regulaciones

Conclusión

La simulación de materiales de entrada es esencial para impulsar la innovación y competitividad en la robótica.



LISTA DE COTEJO DE INVESTIGACIÓN

U2 SIMULACIÓN DE SISTEMAS ROBÓTICOS

Nombre del estudiante: Rosario Soto Emmanuel.

Tema: Simulación fuera de línea.

Portada	2 %	0 %
Temas completos	20 %	20 %
Entrega en tiempo y forma	8 %	8 %
Total	30 %	28 %



LISTA DE COTEJO DE PRÁCTICAS

U2 SIMULACIÓN DE SISTEMAS ROBÓTICOS

Nombre del estudiante: Rosario Soto Emmanuel

Tema: Simulación fuera de línea.

Portada	4 %	4 %
Introducción	5 %	5 %
Desarrollo	10 %	10 %
Conclusiones	5 %	5 %
Referencias	5 %	5 %
Simulación	5 %	5 %
Entrega en tiempo y forma	6 %	1 %
Total	40 %	35 %



**INSTITUTO
TECNOLÓGICO
SUPERIOR DE SAN
ANDRÉS TUXTLA**



ALUMNO: EMMANUEL ROSARIO
SOTO

PROFESOR: DR. JOSÉ ÁNGEL
NIEVES VASQUEZ

MATERIA: SIMULACIÓN DE
SISTEMAS ROBÓTICOS

CARRERA: MECATRÓNICA



INTRODUCCIÓN

ROBO DK es un kit de robótica educativa desarrollado por el ingeniero brasileño, Felipe Sanches, que permite a estudiantes y entusiastas de la robótica crear y programar sus propios robots. El kit está diseñado para ser fácil de usar y accesible, incluso para aquellos sin experiencia previa en robótica.

Características y componentes

ROBO DK incluye una variedad de componentes y características, como:

- Placa base con microcontrolador PIC o Arduino
- Sensores (ultrasonido, infrarrojo, tacto)
- Motores DC y servomotores
- LCD y teclado
- Conectores para expansión
- Software de programación (ROBO DK Studio). Estos componentes permiten crear robots que pueden interactuar con su entorno y realizar tareas complejas.

Aplicaciones y beneficios

ROBO DK se utiliza en diversas áreas, como:

- Educación (enseñanza de programación y robótica)
- Investigación (desarrollo de prototipos y prueba de conceptos)
- Hobby (creación de robots personales)
- Competencias robóticas. Los beneficios de utilizar ROBO DK incluyen:
- Desarrollo de habilidades en programación y robótica
- Fomento de la creatividad y innovación
- Preparación para carreras en tecnología y ciencia.

Programación y compatibilidad

ROBO DK Studio es el software de programación oficial, que ofrece una interfaz gráfica intuitiva y compatibilidad con lenguajes de programación como C, C++ y Python. Además, ROBO DK es compatible con otras plataformas de robótica y microcontroladores, lo que permite una amplia gama de posibilidades de expansión y personalización.

Plugin RoboDK para Rhino

El plugin de RoboDK para Rhino le permite combinar el modelado paramétrico de Rhino con RoboDK para la simulación de robots y la programación fuera de línea.

Con el plugin de RoboDK para Rhino puede cargar fácilmente modelos 3D creados con Rhino en RoboDK. Este plugin le permite programar más de 30 fabricantes de robots diferentes y 300 robots directamente desde Rhino.

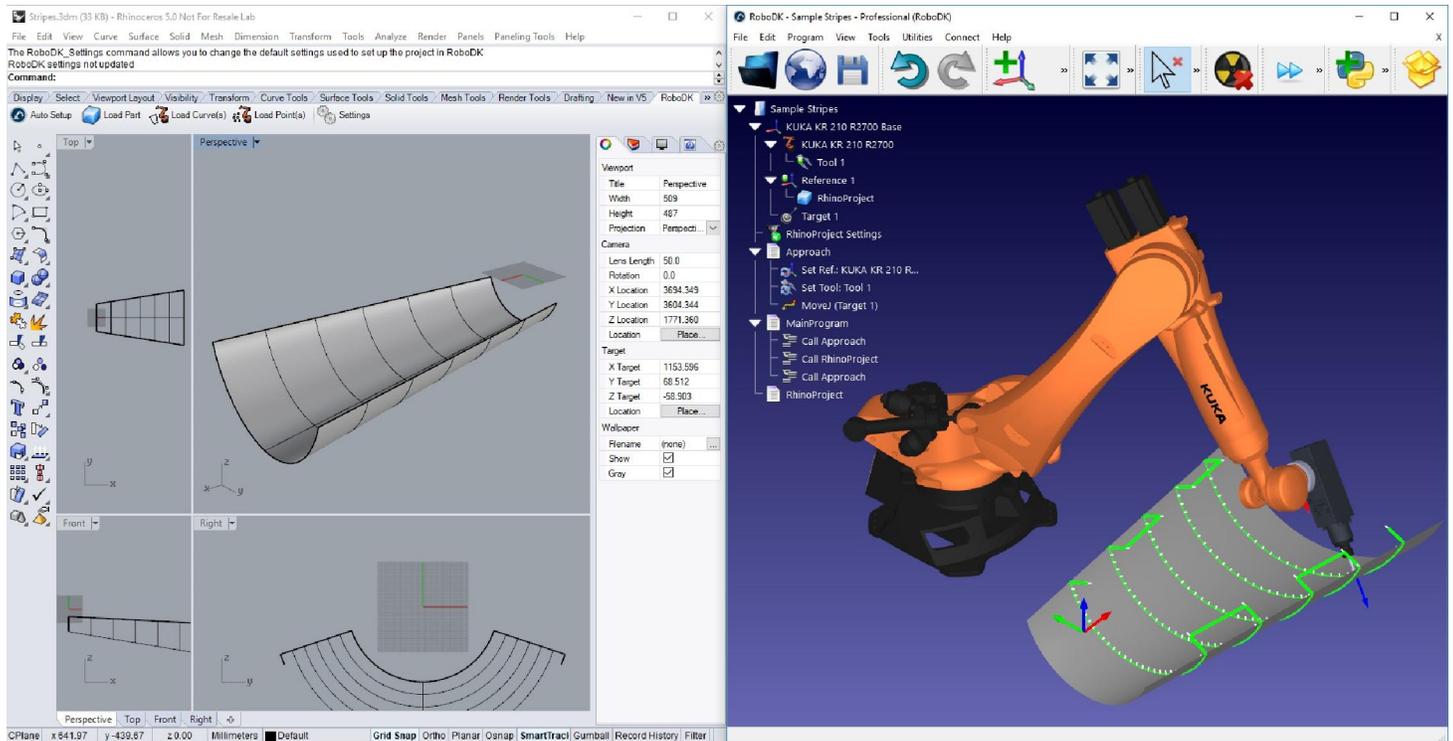
Los programas de robot se pueden generar directamente a partir de un grupo de curvas, puntos o programas CAM (como archivos APT o archivos de código G). Los robots se pueden programar fácilmente como máquinas de 5 ejes para una amplia variedad de aplicaciones de fabricación, como taladrado, soldadura, recorte, impresión 3D o mecanizado robótico. Más información disponible en la sección de mecanizado robótico.

La [sección de post procesadores](#) de robots muestra una lista de las marcas de robots y los controladores compatibles. También es posible modificar o crear nuevos post procesadores para personalizar la generación de programas.

Vídeo: Descripción general del plugin de RoboDK para Rhino: <https://www.youtube.com/watch?v=oDM8EuQw5JA>.

Nota: Lea esta [Guía Básica](#) si es su primera vez utilizando RoboDK.

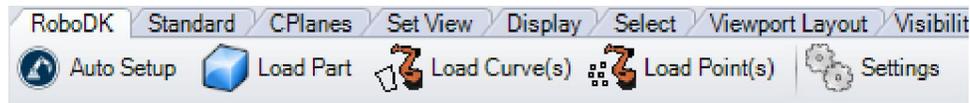
El plugin de RoboDK para Rhino es compatible con Rhino 5 y Rhino 6. El plugin de RoboDK para Rhino es gratuito, no obstante, ciertas opciones en RoboDK requieren la compra de una licencia.



De

Barra de Herramientas de Rhino

La barra de herramientas predeterminada proporcionada por Rhino consta de 5 botones:



-  Configuración Automática – Este botón le permite seleccionar cualquier geometría (curvas o puntos) y cargarla en RoboDK junto con el modelo 3D. Este botón corresponde al comando RoboDK_ConfiguraciónAutomática.
-  Cargar Pieza – Cargue el modelo 3D de Rhino a RoboDK. Las características como curvas o puntos no se cargarán. Este botón corresponde al comando RoboDK_PiezaaRobot.
-  Cargar Curva(s) – Cargue todas las curvas seleccionadas en RoboDK como un elemento nuevo. Todas las superficies seleccionadas se utilizarán para calcular las curvas normales. Estas normales son importantes ya que definirán el eje de aproximación del robot. Este botón corresponde al comando RoboDK_CurvaaRobot.
-  Cargar Punto(s) – Cargue todos los puntos seleccionados en RoboDK como un nuevo objeto. Todas las superficies seleccionadas se utilizarán para calcular puntos normales. Estas normales son importantes ya que definirán el eje de aproximación del robot. Este botón corresponde al comando RoboDK_PuntosaRobot.
-  Configuraciones – Abra la ventana [Configuración predeterminada](#). Esta es la misma ventana de configuración que se muestra para cada componente grasshopper. Este botón corresponde al comando RoboDK_ConfiguraciónPredeterminada.

La principal diferencia entre Configuración automática y Cargar curva(s) o Cargar Punto(s) es que la Configuración automática carga la pieza y crea un nuevo proyecto de seguimiento de curva/punto. El uso de Cargar Curva(s) o Cargar Puto(s) es más rápido y solo actualizará las características geométricas existentes en RoboDK, manteniendo la configuración definida previamente.

Nota: Cada uno de los botones corresponde a un comando. Todos los comandos de RoboDK tienen el prefijo RoboDK_.

Nota: Si desea cargar más de un objeto de Rhino, debe cambiar el nombre de los objetos una vez que se hayan cargado en RoboDK.

Barra de Herramientas de Grasshopper

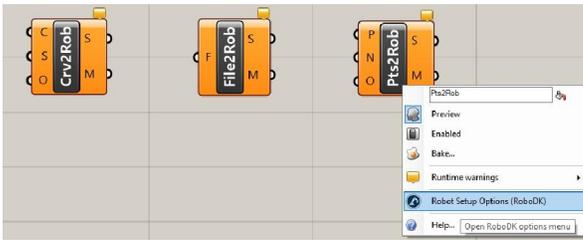
El plugin de RoboDK plugin para Rhino incluye 3 componentes de Grasshopper:



-  Curva a robot (Crv2Rob) – Carga una curva (C) o un grupo de curvas en RoboDK. Opcionalmente, se pueden proporcionar una o más superficies (S) para extraer superficies normales para cada punto de la curva.
-  Archivo a robot (File2Rob) – Cargue un archivo (F) en RoboDK. Se debe proporcionar la ruta de ese archivo. RoboDK admite la carga de archivos APT o de código G destinados a la fabricación de 5 ejes, como la impresión 3D o el mecanizado de robots. Lea la sección de [mecanizado robótico](#) para obtener más información.

- 
 Puntos a robot (Pts2Rob) – Cargue un punto (P) or un grupo de puntos en RoboDK. Opcionalmente, también puede proporcionar los vectores normales (o vectores de aproximación) correspondientes para cada punto (N)

Todos los componentes vuelven a un estado (S) como porcentaje (100% significa que el robot puede completar el programa con éxito). Además, se puede leer un mensaje descriptivo (M) de cada componente.



Puede hacer clic con el botón derecho en cualquier componente y seleccionar Opciones de configuración del robot (RoboDK) para personalizar la configuración específica de un componente.

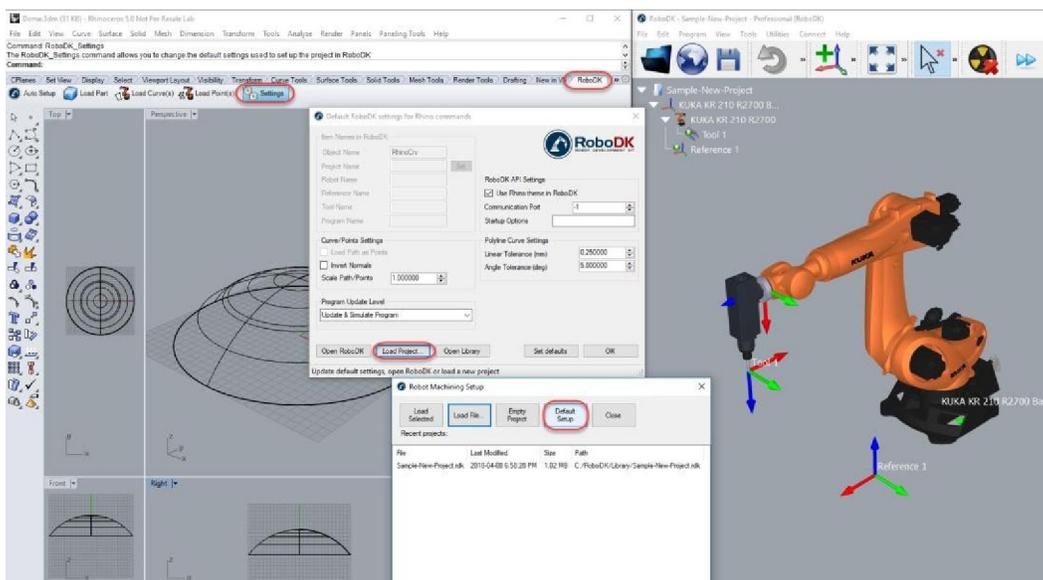
Ejemplos

Esta sección muestra ejemplos básicos para cargar un proyecto de Rhino en RoboDK para fines de simulación de robots y programación fuera de línea.

Ejemplo Rhino

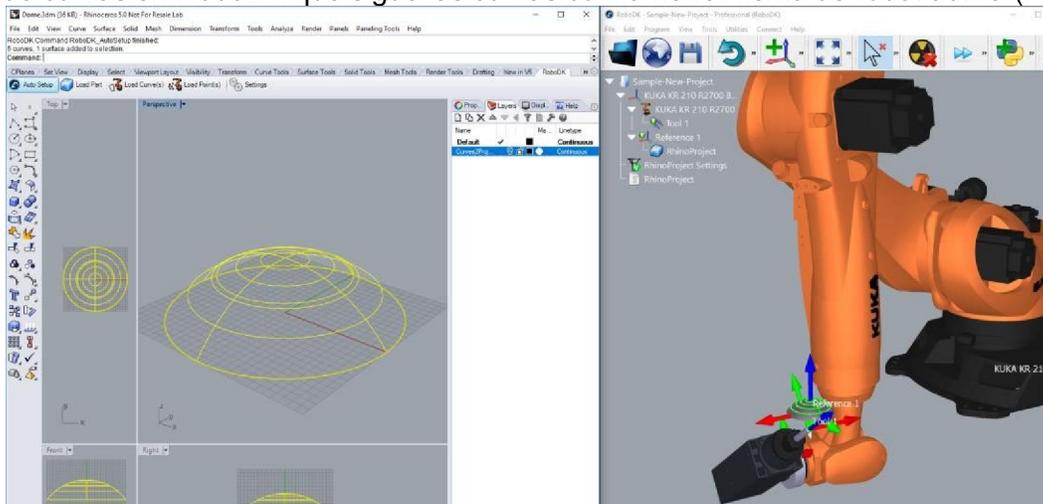
Este ejemplo muestra cómo cargar una estación RoboDK de muestra (configuración predeterminada) y seguir un conjunto de curvas definidas en Rhino con el robot.

1. Cargue el ejemplo de Dome en Rhino.
2. Seleccione la pestaña RoboDK y seleccione  Configuración.
3. Seleccione Cargar proyecto...
4. Seleccione Configuración predeterminada. RoboDK iniciará y cargará un proyecto de muestra con un robot KUKA, una herramienta (un husillo como Herramienta 1) y un sistema de referencia (Referencia 1).



5. Cierre la Ventana Configuración o seleccione Aceptar.
6. Seleccione el botón  Configuración automática en Rhino.
7. Seleccione todas las curvas y superficies y presione Entrar (o haga clic con el botón derecho del mouse). El proyecto se cargará en RoboDK como se muestra en la siguiente imagen.

Deberíamos ver la pieza cargada en el sistema de referencia activo (Referencia 1) y un nuevo proyecto de Seguimiento de curvas en RoboDK que sigue las curvas con la herramienta de robot activa (Herramienta 1).



Consejo: Presione la tecla + o – para hacer que los sistemas de referencia sean más grandes o más pequeños.

También es posible que las superficies normales se hayan extraído en la dirección opuesta. Siga estos pasos para resolver este problema:

8. Seleccione  Configuración.
9. Marque la opción Invertir Normales.
10. Seleccione Aceptar.
11. Repita los pasos 6 y 7.

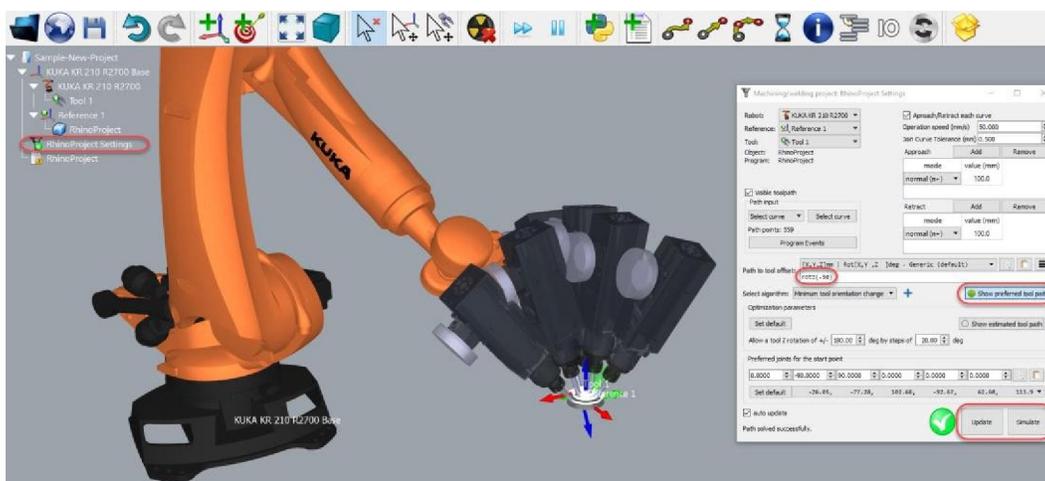
Ahora deberíamos ver las normales de la superficie invertidas y que el movimiento de aproximación proviene de la parte superior de la pieza. El robot debería poder moverse a lo largo de la trayectoria sin ningún problema.

Consejo: Mantenga pulsada la tecla Alt para mover un sistema de referencia o el robot en la vista 3D.

Haga doble clic en  Configuración de RhinoProject desde el árbol de RoboDK para abrir más opciones y personalizar la trayectoria seguida por el robot.

Por ejemplo, podemos seleccionar Mostrar trayectoria preferida de la herramienta para ver y modificar la orientación predeterminada de la herramienta en relación a la pieza. Cambie el valor de Desplazamiento de trayectoria a herramienta para definir una rotación adicional. Para hacerlo, puede añadir un nuevo valor o simplemente usar la rueda del mouse para ver una vista previa rápida del resultado.

Más información para cambiar estos ajustes está disponible en la sección de [mecanizado robótico](#).



Finalmente, cuando la simulación produzca el resultado deseado podemos generar el programa o exportar la simulación:

12. Haga clic derecho en el programa RhinoProject
13. Seleccione Generar programa de robot (F6). Se generará y mostrará el programa del robot.

