

Práctica 03: Detección en tiempo real - RIA 2025/2026

- Autores: Pablo Hernández Martínez e Iván Moure Pérez

1. INTRODUCCIÓN Y ARQUITECTURA DEL SISTEMA

1.1 Objetivo

Esta práctica implementa un sistema de control del robot Robobo mediante detección de poses corporales en tiempo real usando modelos YOLO. El sistema opera en dos fases secuenciales: primero, una fase de **teleoperación** donde el usuario controla el robot mediante gestos con los brazos para acercarlo hacia un objetivo; posteriormente, una fase de **aproximación autónoma** donde la política de RL aprendida en la P.01 toma el control para completar la tarea: llegar al objetivo.

1.2 Arquitectura del sistema

El sistema integra múltiples componentes de visión por computador y control robótico: los modelos de detección YOLO, PPO para la política de RL, y las librerías de Robobo.

En cuanto al flujo de datos, la webcam del PC captura al usuario y alimenta YOLOv8-pose para clasificar sus gestos, mientras que la cámara del robot alimenta YOLOv8 para detectar el objeto objetivo. Los gestos clasificados se traducen en comandos de control para los motores del Robobo, y cuando se detecta el objetivo, se activa automáticamente la política de RL aprendida.

1.3 Sistema de control gestual

La detección de poses considera algunos de los 17 keypoints corporales del modelo YOLOv8-pose. El sistema de control se basa en la posición relativa de las muñecas respecto a los hombros:

Gesto	Acción del robot
Ambos brazos arriba	Avanzar (20, 20)
Brazo derecho arriba	Girar izquierda (0, 25)
Brazo izquierdo arriba	Girar derecha (25, 0)
Sin brazos arriba	Detener motores

Decisiones de diseño: El umbral de 20 píxeles entre muñeca y hombro proporciona robustez ante ruido en la detección de poses de YOLO. La inversión del control (brazo derecho → giro izquierda) resulta más intuitiva para el operador, ya que el robot gira hacia donde "apunta" el brazo elevado desde la perspectiva del usuario.

2. MODOS DE DETECCIÓN Y TRANSICIÓN A POLÍTICA RL

2.1 Detección del objetivo

El sistema soporta dos modos de detección configurables mediante la variable **MODE**. El modo BLOB (para simulador) utiliza la detección de blobs de Robobo para identificar objetos rojos. Por otro lado, el modo YOLO (para el robot real) emplea YOLOv8n para detectar objetos arbitrarios definidos por la variable **TARGET** (por defecto "bottle"). El sistema realiza inferencia con el modelo frame a frame hasta detectar el objetivo (umbral de confianza mínimo de 0.5).

2.2 Integración con política de RL

Condiciones de activación: La política se activa cuando: (1) el objetivo es detectado por la cámara del robot, y (2) el usuario ha

Ejecución de la política:

completado al menos 5 acciones de teleoperación.

Adaptación de observaciones: La política espera observaciones discretizadas en 6 sectores. Se adaptan las coordenadas de YOLO normalizando al rango [0, 100] y se mapean a 0-6 (0-1-2-3-4 indican orientación y 5 indica no detectado).

```
action, _ = model.predict(observation,
deterministic=True)

if action == 0: # Avanzar
    rob.moveWheelsByTime(10, 10, 0.5)
elif action == 1: # Izquierda
    rob.moveWheelsByTime(10, 0, 0.5)
elif action == 2: # Derecha
    rob.moveWheelsByTime(0, 10, 0.5)
```

Condición de éxito: El episodio finaliza cuando el área del objeto detectado supera 70000 píxeles (modo YOLO) o el blob supera 8000 con posición centrada (modo BLOB).

3. RESULTADOS Y CONCLUSIONES

3.1 Demostración del sistema

El funcionamiento del sistema se demuestra en dos vídeos complementarios. El primero ([pov_ordenador.mp4](#)) es en una grabación de pantalla del ordenador que muestra las dos cámaras activas durante la ejecución: la del móvil montado sobre el Robobo, que transmite la perspectiva del robot, y la webcam del ordenador donde el usuario realiza los gestos para dirigir el robot. Este vídeo muestra las anotaciones de gestos de YOLOv8-pose, el objeto detectado por YOLOv8n (si lo hay) las acciones ejecutadas y la activación de la política RL.

El segundo vídeo ([pov_persona.mp4](#)) ofrece un punto de vista externo del Robobo durante la ejecución. Se puede observar de nuevo como navega hacia el objetivo: primero bajo el control del usuario y posteriormente de forma autónoma con RL.

3.2 Análisis del sistema

Entre las **fortalezas** del sistema destaca la fusión fluida de los componentes de visión artificial (YOLO) y el control gestual, permitiendo una operación intuitiva. La reutilización de la política de RL aprendida en la P.01 demuestra la transferibilidad del aprendizaje entre contextos, mientras que la arquitectura modular con modos BLOB y YOLO intercambiables facilita el desarrollo en simulador y el despliegue en el robot real. El feedback visual proporciona al operador información continua sobre el estado del sistema en tiempo real.

Respecto a las **limitaciones**, la latencia inherente al streaming de vídeo desde el móvil puede introducir retrasos perceptibles en la respuesta del robot ante los comandos. La detección de poses muestra sensibilidad a las condiciones de iluminación del entorno, y el umbral fijo de confianza de 0.5 podría requerir ajustes según las características específicas de cada escenario. Además, la política de RL entrenada exclusivamente en simulador puede manifestar el fenómeno *reality gap* al ejecutarse en el robot físico.

3.3 Conclusiones

Este trabajo demuestra la viabilidad de sistemas híbridos que combinan teleoperación gestual con control autónomo. La arquitectura modular permite alternar entre control humano y política de refuerzo de forma transparente, aprovechando las fortalezas de cada modo: la intuición humana para navegación de largo alcance y la precisión del agente entrenado para la aproximación final. La integración de YOLOv8 para detección de poses y objetos proporciona una base robusta y extensible para aplicaciones de robótica interactiva en tiempo real.