

Detección de posición angular con sensor Kinect.

Rubén Hernández Alemán

Universidad Autónoma de Nuevo León
Facultad de Ingeniería Mecánica y Eléctrica

25 Febrero 2015

Contenido

- 1 Introducción
- 2 Planteamiento del problema
- 3 Objetivos
- 4 Hipótesis
- 5 Justificación
- 6 Metodología
- 7 Cronograma
- 8 Bibliografía

Contenido

- 1 Introducción
- 2 Planteamiento del problema
- 3 Objetivos
- 4 Hipótesis
- 5 Justificación
- 6 Metodología
- 7 Cronograma
- 8 Bibliografía

Contenido

- 1 Introducción
- 2 Planteamiento del problema
- 3 Objetivos
- 4 Hipótesis
- 5 Justificación
- 6 Metodología
- 7 Cronograma
- 8 Bibliografía

Contenido

- 1 Introducción
- 2 Planteamiento del problema
- 3 Objetivos
- 4 Hipótesis
- 5 Justificación
- 6 Metodología
- 7 Cronograma
- 8 Bibliografía

Contenido

- 1 Introducción
- 2 Planteamiento del problema
- 3 Objetivos
- 4 Hipótesis
- 5 Justificación
- 6 Metodología
- 7 Cronograma
- 8 Bibliografía

Contenido

- 1 Introducción
- 2 Planteamiento del problema
- 3 Objetivos
- 4 Hipótesis
- 5 Justificación
- 6 Metodología
- 7 Cronograma
- 8 Bibliografía

Contenido

- 1 Introducción
- 2 Planteamiento del problema
- 3 Objetivos
- 4 Hipótesis
- 5 Justificación
- 6 Metodología
- 7 Cronograma
- 8 Bibliografía

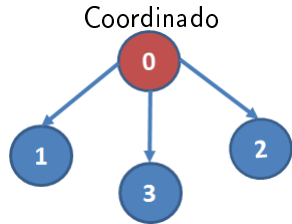
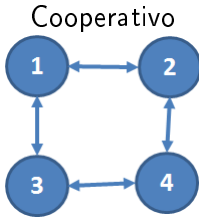
Contenido

- 1 Introducción
- 2 Planteamiento del problema
- 3 Objetivos
- 4 Hipótesis
- 5 Justificación
- 6 Metodología
- 7 Cronograma
- 8 Bibliografía

Sistema Multi-Agente (SMA)

¿Que es?

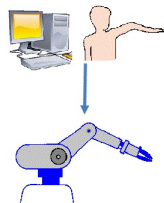
Un sistema multi-agente puede operar bajo los esquemas:



Sistema Multi-Agente (SMA)

Sistema Teleoperado:

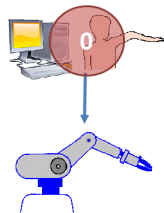
- Un líder V_0
- $\mathcal{V} = \{V_1, V_2, \dots, V_N\}$
- $\hat{\varepsilon} = \{(j, i) \in \mathcal{V} \times \mathcal{V}\}$



Sistema Multi-Agente (SMA)

Sistema Teleoperado:

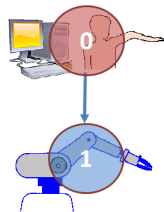
- Un líder V_0
- $\mathcal{V} = \{V_1, V_2, \dots, V_N\}$
- $\hat{\epsilon} = \{(j, i) \in \mathcal{V} \times \mathcal{V}\}$



Sistema Multi-Agente (SMA)

Sistema Teleoperado:

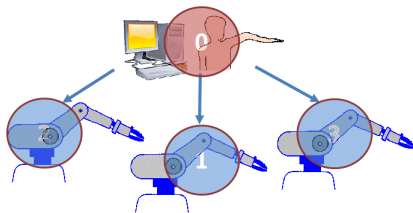
- Un líder V_0
- $\mathcal{V} = \{V_1, V_2, \dots, V_N\}$
- $\hat{e} = \{(j, i) \in \mathcal{V} \times \mathcal{V}\}$



Sistema Multi-Agente (SMA)

Sistema Teleoperado:

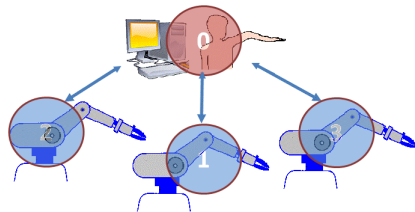
- Un líder V_0
- $\mathcal{V} = \{V_1, V_2, \dots, V_N\}$
- $\hat{e} = \{(j, i) \in \mathcal{V} \times \mathcal{V}\}$



Sistema Multi-Agente (SMA)

Sistema Teleoperado:

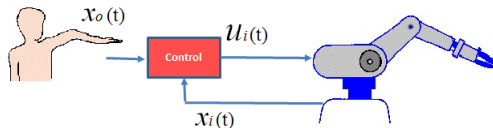
- Un líder V_0
- $\mathcal{V} = \{V_1, V_2, \dots, V_N\}$
- $\hat{e} = \{(j, i) \in \mathcal{V} \times \mathcal{V}\}$



Control en SMA

Arquitectura de control en SMA:

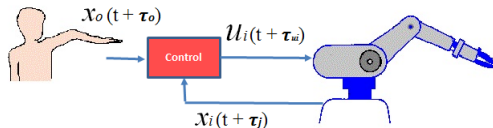
- Requiere información local
- Retardos de tiempo
- Ruido en medición



Control en SMA

Arquitectura de control en SMA:

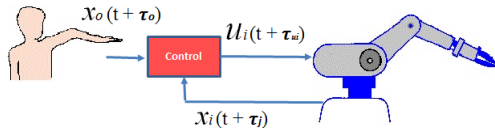
- Requiere información local
- Retardos de tiempo
- Ruido en medición



Control en SMA

Arquitectura de control en SMA:

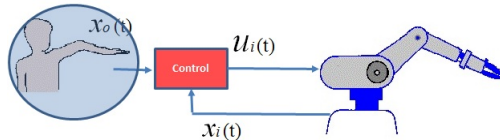
- Requiere información local
- Retardos de tiempo
- Ruido en medición



Control en SMA

Arquitectura de control en SMA:

- Requiere información local
- Retardos de tiempo
- Ruido en medición



Planteamiento del problema

Realizar una medición de los desplazamientos angulares de alguna extremidad humana en tiempo real.

Objetivos

- Diseñar una herramienta de software, basada en técnicas de vision computacional, que permita medir los desplazamientos angulares.
- Acotar los desplazamientos medidos para evitar trayectorias no deseadas.
- Establecer alguna regla, de modo que la función de cada trayectoria sea continuamente diferenciable.

Hipótesis

Es posible medir los desplazamientos angulares mediante el uso de técnicas de visión computacional.

Con ayuda del procesamiento de los frames obtenidos, es posible discriminar movimientos no deseados.

Justificación

- Reducción de costos, sólo se requiere de una cámara o sensor.
- Reduce ruido en la medición, que es muy común en sensores fijos al cuerpo: potenciómetros o giroscopios

Justificación

- Reducción de costos, sólo se requiere de una cámara o sensor.
- Reduce ruido en la medición, que es muy común en sensores fijos al cuerpo: potenciómetros o giroscopios

Metodología

- Revisión bibliográfica de técnicas existentes.
- Revisión de trabajos existentes o con dicha orientación y de librerías:
 - SciPy
 - NumPy
 - OpenCv
 - OpenNI
 - SensorKinect
- Realizar pruebas de visión computacional con sensor **Kinect**.
- Diseñar e implementar software de detección de posición.
- Realizar pruebas finales.

Cronograma

	<i>Enero</i>	<i>Febrero</i>	<i>Marzo</i>	<i>Abril</i>	<i>Mayo</i>	<i>Junio</i>
Rev. bibliografica						
Rev. Trabajos y Librerías						
Pruebas previas						
Implementación						
Pruebas finales						

Bibliografía

- [1] Joseph Howse. "*OpenCv Computer Vision with Python*". PACKT Publishing, 2013.
- [2] Jan Erik Solem "*Programming Computer Vision with Python*". Creative Commons 2012.
- [3] Daniel Lelis Baggio "*Mastering OpenCV with Practical Computer Vision Projects*". PACKT Publishing. 2012
- [4] E. R. Davies (2008). "*Computer and Machine Vision: Theory, Algorithms, Practicalities*". Fourth Edition, Elsevier 2012.