Real-Time Systems

HW 1: Distributed RTS Design

Due: April 27, 2017 (Before Class)

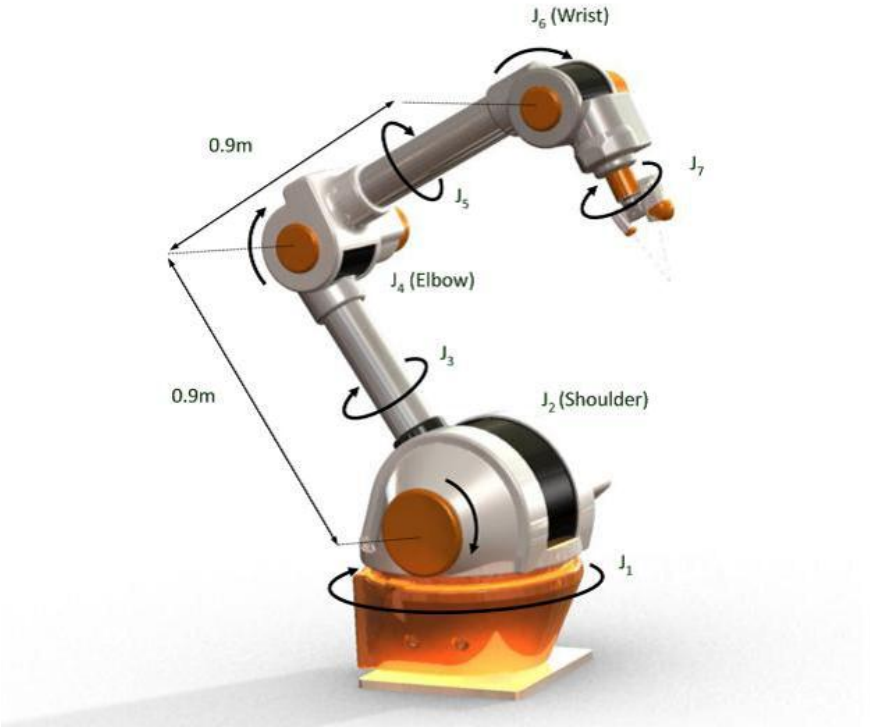
Introduction

The objective of this homework is to solidify the concepts of distributed real-time system modeling and how they relate to good system design. In this homework you will design the high-level distributed RTS for an articulated robot arm using the ideas we've been discussing. From a grading perspective I want to see that you've put thought into the design. I'm less concerned with the precise details and design choices you make and more concerned with the justification you make for your design choices.

The System

The system in question (see the figure below) is an articulated robotic arm. It has many joints consisting of motors, sensors, etc. To get started, break the system down into its constituent motors, sensors, etc. Then select the components you might need and group them into clusters (if it's helpful). Then set about designing messages, interfaces, etc. Don't worry about any intelligence, control, planning, or guidance algorithms – assume someone else is designing those around whatever RTS you design.

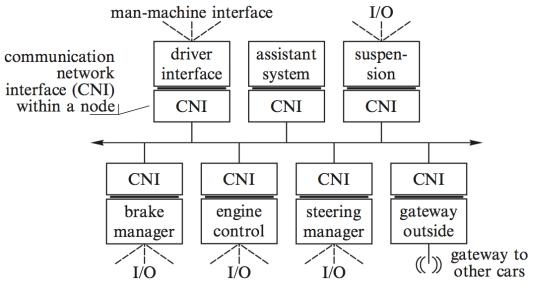
At one extreme, your design could have a single master component architecture where every sensor, motor, etc. is connected to that one master component via a local interface and there is no LIF, cluster, or interaction with other components. This isn't an acceptable design choice since it isn't really a distributed RTS. At the other extreme you could have a separate component for every single motor, sensor, etc. This is also not an acceptable design choice. While it is a distributed RTS, it isn't effective or efficient.

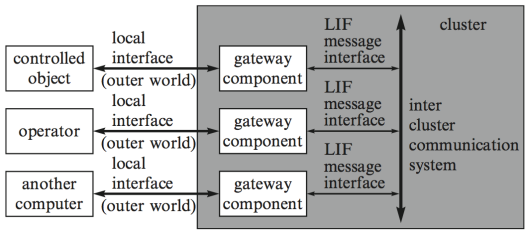


What to do - 100 points

In the document you type up, provide a section for each of the main bullets (i.e. Components, Messages, Interfaces, System Sketch) so I can follow along easily.

- **Components – 25 points** – identify the components and clusters you will need and justify the reasoning
  - Give each component a clear purpose
    - Suggest 1 or 2 RTOS tasks (algorithms) that would run on each component you identify
  - Identify any clusters of components (it might simplify the design)
    - You don't have to have clusters, but if you do explain your design choice
- **Messages – 25 points** – identify the messages each component will send and receive and justify the reasoning
  - Make sure to specify for each message whether it will be event-triggered or time-triggered. Will it be an atomic message, or broken up (as in TDMA (time-division multiple access))?
    - Give an outline of what the message will contain (what will be in the header, data, and trailer)
      - This will clearly connect strongly to the next part - the LIF and TII - but here you are listing the messages without the specific interface spec
  - Since you don't have information about the actual autonomous algorithms involved (or their timing) make some assumptions about the timing requirements and indicate them for purposes of your design. For example, assume the motor controller must update at a 30 Hz rate and incorporate that into your timing specs. What does this mean for how often you must read the sensors?
- **Interfaces – 25 points** – design the LIF, TII interfaces for each component (don't worry about the TDI and local interfaces as those are implementation specific)
  - Tell me about the level of detail at which you've chosen to apply the real-time model abstraction and why you chose it. This means I want you to tell me what details are you excluding, what assumptions are you making about the system, etc.
  - It's possible the TII will be very similar (or even exactly the same) for many of the components.
  - Use the "Linking Interface Specification" section 4.6 in the Kopetz book [1] (posted to the internal course website) to help you design the interfaces
    - Provide the transport spec, operational spec, and meta-level spec
- **System Sketch – 25 points** – describe the system and provide a sketch of the system layout similar to ones we've seen in class (see the figures below).
  - Provide a high-level description (one or two paragraphs) of how the system will operate.
  - You may need multiple sketches (depending on your design choices), one showing components, one showing interfaces, one showing clusters – but it's up to you. These are examples from the book:





Protips

- Part of the purpose in our RT model is the ability to abstract a component of the system so we can reuse it. I strongly suggest you design components that you can group into clusters and reuse those components and clusters in appropriate points in the robot. This shows good design.
- Remember, system design is a subjective thing...there isn't a **right** answer. However, I will grade based on your application of the RT model we've been discussing. I will **not** dock you points for any design decisions you make **as long as you describe why you made the decision** and it's not unreasonable. You must have all the elements: components, messages, and interfaces, but you need not provide so much detail that this assignment requires 40 hours worth of work.
- If you don't know what exactly would go into an articulated robot arm do some Googling to figure out how it might work. At a bare minimum there must be a motor and a sensor for each rotating part. Consider the following sources:
  - `https://en.wikipedia.org/wiki/Industrial_robot#Robot_programming_and_interfaces`
  - `http://www.societyofrobots.com/robot_arm_tutorial.shtml`
- You are **not** designing any of the intelligence in this robot. It doesn't matter how the robot moves, what algorithms plan its trajectory, or how the controller works...someone else designed that. You are **solely** focused on the real-time model for the system.
- Some suggested programs for "drawing" any diagrams are: PowerPoint, Keynote, Visio. But feel free to use whatever program you want.

What to submit

1. **(100 points)** - A PDF (or Word) document with your typeset solution to this design problem. There is no page limit – though my happiness will be inversely proportional to the number of pages of your document so long as you meet the requirements. You need enough detail to persuade me you've thought about it, but not so much detail it takes up all your time. Upload all of this into "handin" at `https://cse-apps.unl.edu/handin`

References

[1] Hermann Kopetz. *Real-time systems: design principles for distributed embedded applications*. Springer Science & Business Media, 2011.