Introduction to AI Robotics

4 - Software Organization ~~and~~ of Atonomy

I Overview

This chapter attempts to answer the questions

- What is the programming style for Autonomy?
- Can intelligence be added in layers?

II The Three types of Software Architectures

An architecture provides a principled way of organizing a control system. In addition to providing structure, an architecture imposes constraints on the way the control problem can be solved

A) Types of Architectures

This was defined by the Air Force, so pay attention!

- Operational Architecture: describes what a system does ~~describes how a system is decomposed into major subsystems~~. This talks about how the seven distinct areas of artificial intelligence are pieced together and can be used to determine if a design provides the intended functionality.

- Systems Architecture: describes how a system is decomposed into major subsystems. This can be used to determine if ~~of~~ a design meets good software engineering principles, especially modularity and extensibility.

- Technical Architecture: describes the implementation details of the system. This specifies algorithms, languages that the modules are programmed in. It can be used to ·

to determine if the design is using the most appropriate algorithms. Technical architectures are always changing, reflecting the advances in artificial intelligence.

B) Architectures Reinforce Good Software Engineering Principles

Thinking of the design of an autonomous ~~robot~~ robot in terms of the the three architectures encourages the creation of designs that meet the four general principals of software engineering.

- Abstraction: Operational and System ~~architects~~ architectures allow designers to ignore details of to focus on overall organization ~~and~~ of intelligence

- Modularity: Systems and technical architectures encourage the designer to think in terms of object oriented programming. High cohesion and low coupling supports unit testing and ~~to~~ debugging.

- Anticipation of change with incrementality: Related to modularity, the system and technical architectures provide the designer with insights whether the code is engineered to support upgrading an algorithm and adding new capabilities without requiring the designer to rewrite the code.

- Generality: The operational, ~~and~~ system, and technical architectures determine if the basic organization, subsystems, and implementation will allow the code to be used for other applications.

Arkin adds two more software engineering principles for AI robotics.

- niche targetability: How well the robot works for its intended application. This is often at odds with generality.

- Robustness: identifies where the system is ~~vul~~ vulnerable, and how the system design intrinsically reduces vulnerability.

III Canonical AI Robotics Operational Architecture

Over the years, the AI robotics community converged on the hybrid deliberative/reactive operational architecture. This architecture has three layers:

- Reactive
- Deliberative
- Interactive

This has obvious biologically inspired metaphors. Biological intelligence can be thought of having four major functions

- Reaction
- Deliberation
- Conversion of signals into symbols, and
- interaction

These 3 layers distinct ~~the~~ different styles of computing for each layer. Not only do these layers encapsulate a philosophy each layer adheres to, they have five attributes which influence computing

- Primitives,
- Perceptual ability,
- Planning Horizon,
- Time Scale, and
- Use of models.

A) Attributes for Describing Layers

1) Primitives

AI robotics views autonomous capabilities as consisting four primitives:

- SENSE,
- PLAN,
- ACT, and
- LEARN.

a) SENSE

SENSE takes information from ~~other sensors or~~ sensors and producing an output useful by other functions

b) PLAN

If the function takes information from either from sensors or its internal knowledge and produces tasks for the robot to perform, then it is a PLAN function.

c) ACT

These are functions that output commands to motors and actuators.

d) LEARN

A mechanism by which an agent maximizes its chances for success, but transcends the other three primitives. An agent can learn to sense or plan better, acquire more acts or skills, or even learn from a specific mission what to sense, what to plan for, and how to act.

2) Perceptual Ability

Perception can either be direct or requireing recognition. That is, the system works directly with the signal or the signal requires conversion to a symbol, respectively.

3) Planning Horizon

The planning horizon for each capability can have present, past, present, or future, past, present.

4) Time Scale

Dictates the rate at which the function must run. It can be slow for long-term complex planning or very fast for quick reaction to immediate threats. It can be used to determine asynchronous operation of the software components.

5) World Model

AI robotics divides capabilities into those that use information for only that function (local world model), and those that rely on all sensor information to first be processed into a global world model.

B) The Reactive Layer

The reactive layer consits of functionality constructed from SENSE and ACT. That is, this layer is analogous to an animal's reaction where the action is generated directly by sensed or internal stimuli. There is no awareness of the larger world or mission considerations

SENSE & ACT being so ~~thik~~ tightly coupled produce behaviors which produce capabilities.

C) The Deliberative Layer

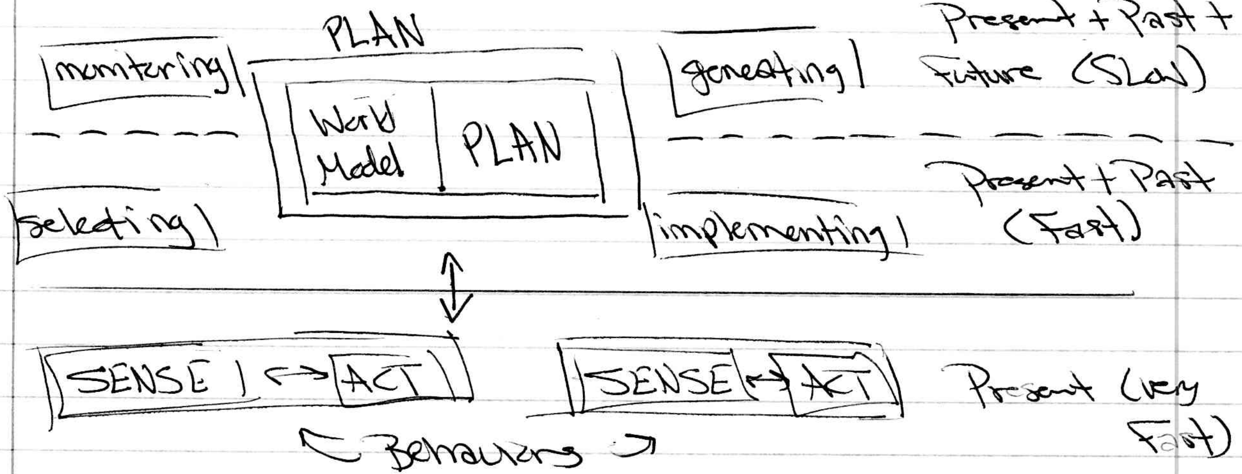
This layer corresponds to the cognitive loop associated with the cortex of the brain. This part independently takes as inputs the same signals used by the reactive layer and adds additional sensor processing to make conscious decisions. As such, this layer tends to run slower and asynchronously from the reactive layer.

build from the input signal!

The deliberative layer rely on symbols that has been synthesized, labeled, and combined with memory and knowledge. Reaction is based solely on the signal from sensors.

The deliberative layer hosts the PLAN activities of the robot. The Planner generates a plan in reference to the world model and constructs the correct SENSE-ACT behaviors ~~to~~ in the reactive layer to execute the plan.

The deliberative layer is divided into two sublayers connected by the world model that contains four deliberative functions.



The upper most layer from the deliberative layer contains mission generation and monitering functionality. This is the most "cognitively" ~~[illegible]~~ demanding layer which is the slowest.

The second (lower) layer from the deliberative ~~model~~ layer contains the selection and implementation of the behaviors. This layer has to make decisions based on what is ahead of it, so it typically is less time intensive.

These two sublayers work asyncronously of one another!

Functions within the deliberative layer are often programmed in a functional language such as LISP, which simplifies planning and reasoning.

↳ How?

D) The Interactive Layer

This layer is required to interact with other agents, either people, other robots, or software agents. This is a topic of research, so there is no clear defined method of representing the structure of its implementation.

What there is an idea of is that there are reactive interactions called social intelligence. There is also deliberative reactions such as security and privacy decisions.

E) Canonical Operational Architecture ~~Design~~ Diagram

This is basically the same figure as before, but with an interactive layer on top of the deliberative.

IV Other Operational Architectures

This section reviews other architectures and discusses how they relate to canonical AI robot architectures.

A) Levels of Automation

Traditionally, this community has labeled the state of automation at any given time in terms of what functions a human has currently delegated to the computer. This naturally leads to a heiarchy, or levels of automation that the system may perform at.

The resulting layers of automation is often used as an operational architecture with the thought that additional layers will be added making the robot more autonomous.

The major advantage of this method is that the four functions and levels offer more specific modularity than the 3 layers in the biological method.

The issue with this method is that it was designed as a method of identifying the current state of autonomous capability, not as an operational architecture. That is, the goal is full automation and thus interactivity is often neglected.

B) Autonomus Control Levels (ACL)
This is sometimes used by the DoD for developing Unmanned systems. ACL has 10 levels reflecting increasing sophistication to the observe, orient, decide, and act (OODA) loop, as well as their value for categories of military missions. Each level can be divided into 3 functions:

- Perception/Situational Awareness,
- Analysis/Decision making, and
- Communikation/Cooperation.

The levels are

9) Battle space Swarm Intelligence
8) Battlespace Cognizance
7) Battlespace Knowledge
6) Real-time Multi-Vehicle Cooperation
5) Real-Time Multi-Vehicle Coordination
4) Fault/Event Adaptive Vehicle
3) Robust Response to Real-Time Faults/Events
2) Changeable Mission

1) Execute Preplanned Mission
0) Remotely Piloted Vehicle

The problem with this method is that the ad-hoc selection of the levels is unclear.

C) Levels of Initiative

Initiative is conceptualized by roles rather than by functions. The five levels of initiative are

1) No autonomy: Follow rigid programming
2) Process Autonomy: The robot can choose the algorithm or process to achieve its task goal.
3) System State Autonomy: The robot can ~~change its goals to meet the intent of the roles.~~ generate and choose between options to achieve its goal
4) Intentional Autonomy: The robot can change its goals to meet the intent of its roles
5) Constraint Autonomy: The robot can create its own roles and goals.

The method is similar to ACL in prioritizing adaptability of the robot in emerging situations, but is more focused on the one to many relationship.

The disadvantage is that it lacks specificity.

V Five Subsystems in Systems Architectures

Software for an intelligent robot normally has at least five subsystems, which are encapsulated as object libraries or similar reusable programming repositories.

System architectural design encourages designers to think in terms of creating libraries of algorithms and data structures for each subsystem.

A) Planning

This subsystem is associated with generating and monitoring overall mission objectives and passing along the geospatial component of those objectives to the navigation subsystem, selecting the motor and perceptual resources, implementing or instatiating those resources, and montering the performance of the mission.

B) Navigation

Associated with generating and monitering paths of movement and selecting the resources to accomplish the movements. This subsystem may contain the path planning algorithms. These algorithms may also be stored in the Cartographer ~~state~~ subsystem.

C) Cartographer

Also known as the World Model or World Map. This subsystem is responsible for the construction and maintenance of knowledge representations about the world. While containing geospatial maps, it can also include more abstract symbols of information, such as beliefs about the state of the world or the intent of other agents

In deliberative robot, this subsystem bridges the gap of signals and symbols.

D) Motor Schema

Also known as the Motor Schema Library (or the behavioral subsystem), ~~a~~ is associated with selecting the best motor routines and implementing actions.

E) Perception

Also known as the sensing, perceptual, ~~st~~ Schema, or Perceptual Schema Library, is associated with selecting the best sensors for the motor actions and implementing sensor processing alg

There are two important notes.

- The subsystems are not independent and do not represent a sequence of programming actions. They ~~are~~ represent major objects or classes that reflect software engineering decisions about how to group similar functions and data.

- The list of subsystems is not necessarily complete

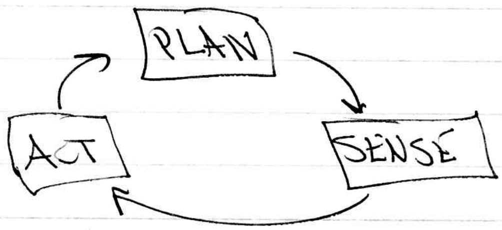
VI Three Systems Architecture Paradigms

Historically, systems architectures for AI Robotics fall into 3 categories called paradigms:

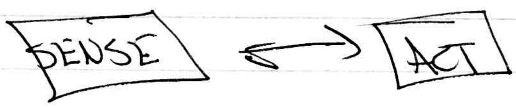
- heirarchical,
- Reactive, or
- hybrid

These paradigms arrange the flow of data and control within the robot. These are described uniquely by the interaction between SENSE, PLAN, and ACT and the route of sensing.

A) Trait 1: Interaction Between Primitives
There are 3 styles of interaction



Go through the cycle of SENSE, PLAN, and ACT



- Sense and act communicate, but run concurrently

PLAN is ran asynchronously to the SENSE ↔ ACT concurrency.

B) Trait 2: ~~SEN~~ Sensing Route
There are 3 routes sensing routes. The routes represent design choices in computational sophistication and costs, latency, and whether to mimic sensing organization in animals

1) Local Sensing
Sensing goes directly from the sensor to the behavior function. The function is responsible for the interpretation and is thus, local.

2) Global Sensing

Data originating from all the sensors is transmitted to a function that transforms and fuses the data into a global World Model.

3) Hybrid Sensing

This, as the name suggests, is just a combination of the previous two. That is, data from a sensor is passed to a local and global sensing method.

C) ~~Hst~~ Hierarchical Systems Architecture Paradigm

Hierarchical systems are used for implementations ~~that~~ where the mission is well-understood and further upgrades are not expected. These are also used when there is little expectation for code reuse or functionality expansion.

This method utilizes the SENSE-PLAN-ACT cycle to create decisions. The world is perceived in SENSE, the data is fused into the World Model. From this new data, a ~~plan~~ PLAN is created or adjusted which the provides actions for ACT to complete the plan.

This method can be really slow and there ~~are~~ is no method for reactions that are quick. It is, however intuitive and easy to follow

D) Reactive System Paradigms
This only uses perception and the motor schema (as ~~discu~~ discussed before). Behaviors have a number of advantages, mostly of them also being consistent with good software engineering principles.

Behaviors are inherently modular and easy to test separate from the system. They also support incremental expansion.

All the behaviors (at least multiple) may run simultaneously. It is the ACT phase that coordinates these ~~functions~~ behaviors to output the true desired behavior.

E) Hybrid Deliberative/Reactive Systems Paradigm
Planning occurs over long periods of time and requires a global knowledge. That is why it is decoupled from the real-time execution of the reactive part.

This is the most versatile paradigm.

The best example of this is the 3T (three tier) architecture used by NASA. It has 3 layers, a purely deliberative, a purely reactive, and an interface layer.

VII Execution Approval and Task Execution
A human may be set as an executive decision maker to allow or disallow ~~as~~ a plan to be initiated. This, however, has not been very successful for high performance robots.