Now, Remove Mother Board

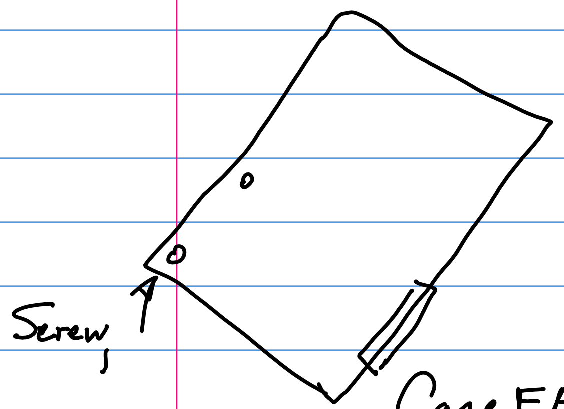


Fig 12

Jan 27 (Wed) TX2 SPI Driver

Minh, HL, YY, NP.

1) Background O.S. Kernel Image

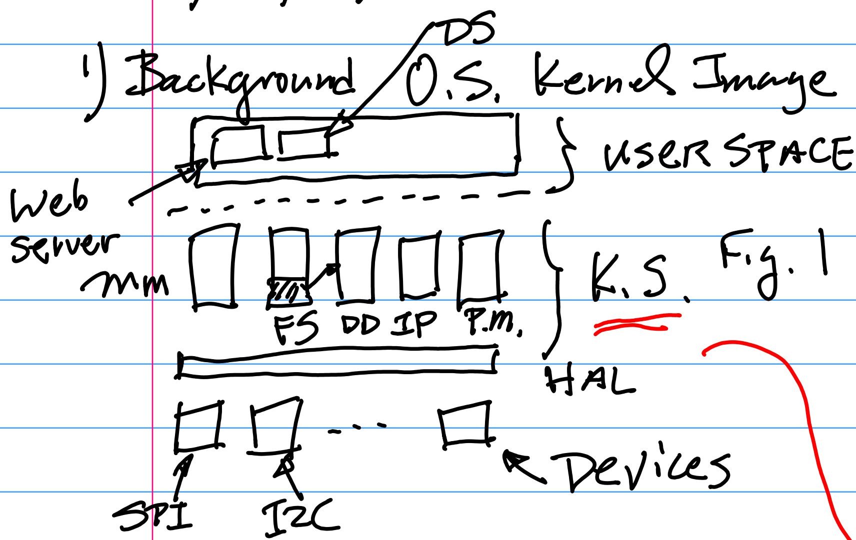
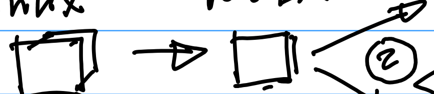


Fig. 1

2) Open Source Source Code for O.S. as Distribution

Linux NVDA



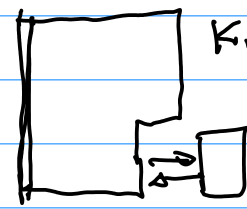
① Binary Distribution → Optimization

② Source Code Distribution

Note:

1° Check the module (SPI) if it is a part of the Kernel, lsmod, insmod

Action 1: Tool for K.S. module manipulation

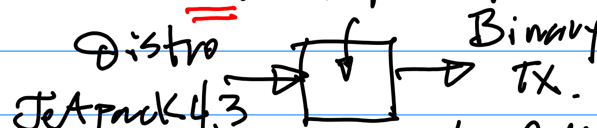


Insmod → When PWR Down Reexecuted & modification of Boot Script.

Action 2: Build O.S. Image From Distro

2.1. Locate the Distro (NVDA Dev. Site)

2.2. Cross-Compiler (NVDA)



On x86 platform

2.3 Upload the Kernel to TX2. Connection (Cable) for writing the Kernel, Ethernet TFTP

Optimization Speed, Foot Print (size) Functionality BootLoader

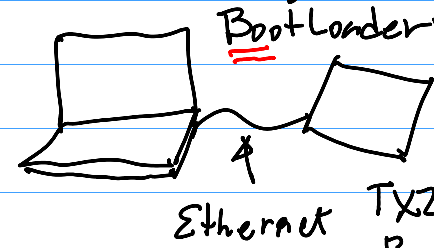


Fig 2

Find/DownLoad Boot Loader from NVDA → In JetPack 4.3?

Connectivity Table, plus S/W, Boot Sequence

Action 3. Boot Sequence

Feb 3, Minh, YY, HL.

To Add Driver:

- Kernel Distro (Source) — Device Driver (Part of O.S) — O.S.
  - Binary Executable
- Tool Chain

Feb 6.

$$f(x) = f(x_0) + \frac{f'(x)}{1!}(x - x_0) + R_n(x) \quad \ldots (1)$$

$$R_n(x) = \frac{f''(x)}{2!}(x-x_0)^2 + \frac{f^{(3)}(x)}{3!}(x-x_0)^3 + \cdots$$

So,

$$f(x) \simeq f(x_0) + f'(x)(x - x_0)$$

Let $f(x) = 0$ (Because it is at a root).

$$0 \simeq f(x_0) + f'(x)(x - x_0)$$

$$-\frac{f(x_0)}{f'(x_0)} = x - x_0$$

$$X = X_0 - \frac{f(x_0)}{f'(x_0)}$$

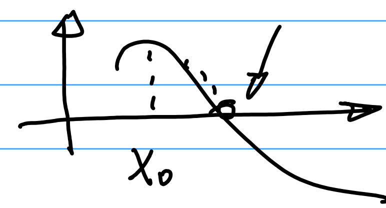
Let $\Delta X = -f(x_0)/f'(x_0)$

Therefor,

$$X = X_0 + \Delta X$$

OR

$$X^{n+1} = X^n + \Delta X$$



Feb 10, Feasibility of RISC-V (FPGA) Interface with CAM module.

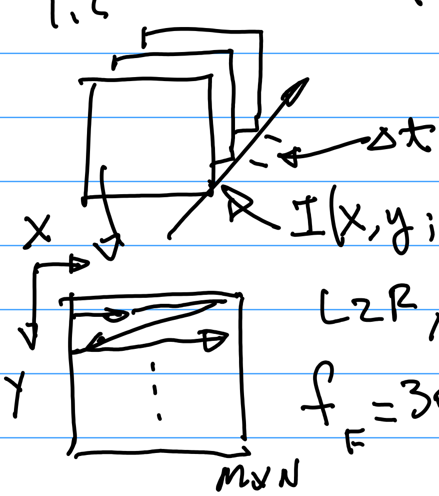
1. Clock Rate

Image Sensor Clock Rate:

1.1 Resolution of the Image: $2mp \simeq 2 \times 2^{20}$ ...(1)

FRAME Rate: 30 FPS

1.2 M×N (1024 × 768)



L2R, T2B

$$f_F = 30\ Hz \ldots (2)$$

M: No. of Pixels Per Row (x) 8/

N: No. of Rows Per frame. y

$f_H = N f_F$ ... (3)

Horizantal

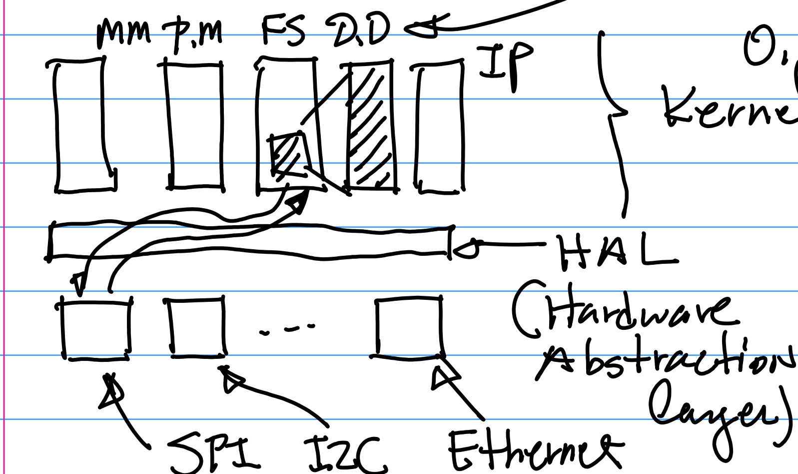
$N = 768$, $f_H = 768 f_F$

$f_D = M f_H = 1024 f_H = 1024 \cdot 768 f_F$

pixel clock

Feb 12 Regarding SPI Driver

Device Driver as a part of FS



O.S Kernel Space

(Hardware Abstraction Layer)

Driver. spi.c

spi-Tegra114.c

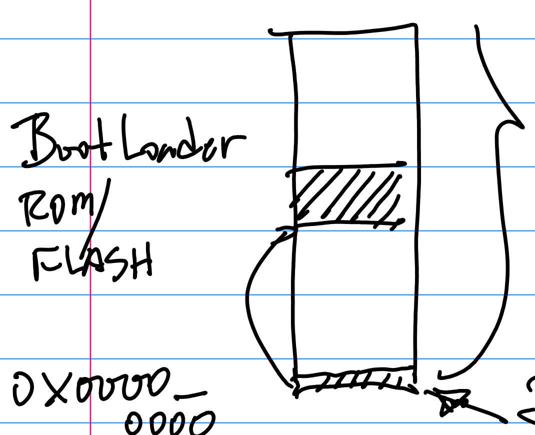
To Be Modified by NVDA Developer

Hardware Specific NVDA

Kernel Image Build & Flashing.

Boot Load: Interface

{ a Load O.S
{ b Interface to Developer



2^{32}
memory map
4 GB
OR
64 Bits

Jmp/Branch Instruction

Objectives:

1. Ethernet Based O.S. Flashing TFTP

NSF OR 2° module

9/

Note

r31 Feb15, Monday. YY, HL r0

0x43d0_0000
plus offset

SPI_
SPR : Command_0
Offset 0x0
Base ?

SPI_CMD0[30]=1 : SPI master

SPI_CMD0[29:28]=01 : mode 1

SPI_CMD0[27:26]=00 : SPI CS0
Chipselect

SPI_CMD0[25:22]=1111 By Default

:

SPI_CMD0[4:0]=11111=0x1f
32 Bit Transfer

0x43d0_0000

0100 0011 1101

4

SPI_CMD0[31]

SPI_CMD0[30]

SPR

Clocks : Sys_CLK → P_CLK

SPI_CLK ← STR

Patent Applications Feb17 (Th)

Deep Reinforcement Learning + Food (Noodle-Cooking)

Autonomous System

a AIV100 { Computer Vision
AI

b FD-100 6 DoF

c AGV2000
4000

FLK 4000

Welcome Chee!

1. Schedule : M-F 20 hrs/week
10:00 Am – 3:00 p.m.
Lunch Hour 1 hr (12:00 - 1:00 pm)

2. AGV2000

2.1 Sensor Fusion

Proximity Sensor Array (6)

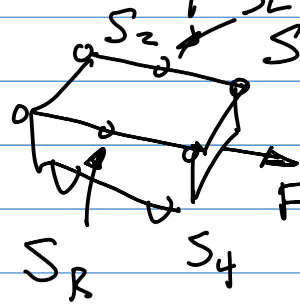
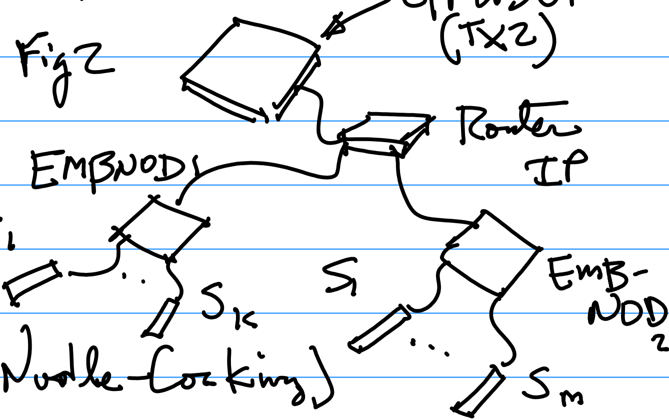


Fig 1.

Vision (CAM) Video (4)
1 Stereo Vision

2.2 Architecture

Fig2



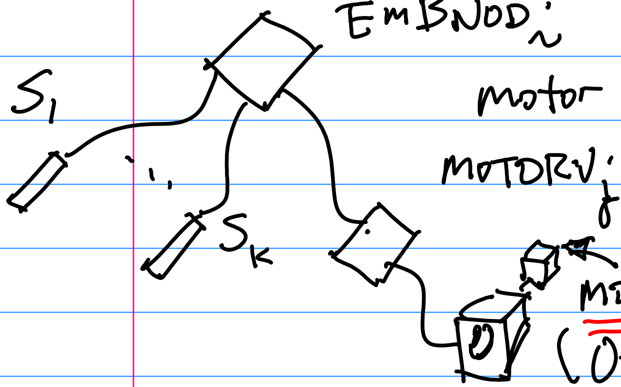
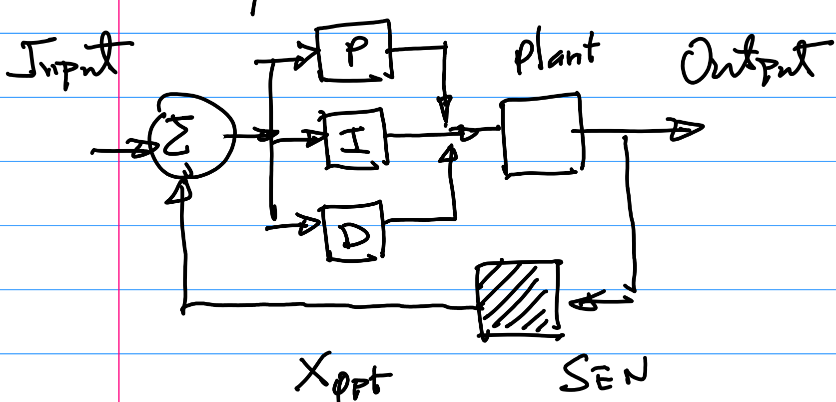


Fig3

I/F: $S_1 \cdots S_K \cdots S_m$
Digital I/F I2C, SPI
MOTSEN/OPTSEN



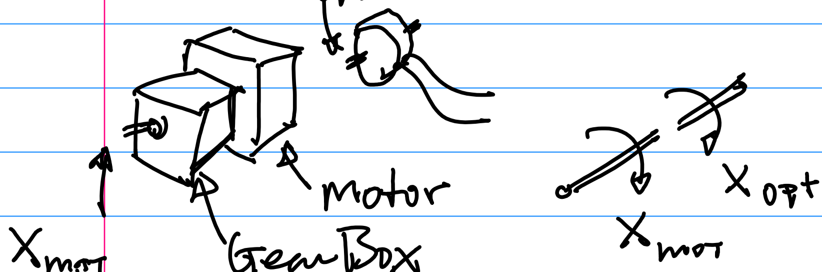


Fig. 4

2.3 TX2 (NVDA GPU)

Deep learning (User Space)
Path planning

Task 1 Kernel/Device Driver Programming

① Kernel Source ② ToolChain ③

Document Datasheet

Task2: Unity platform to test out Self-Driving Algorithm.

Unity Installed → Tutorial "Run"

Karting (Car) GAME

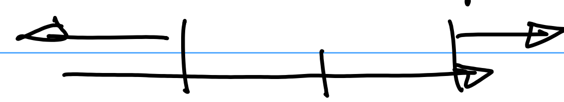
Record: Your Action
Motion + Event

4 ± Days

Benchmarking with Google Deepmind Team.

Feb 19 (Friday) YY, NP, HL

1. Year 2021 Priority



2019 2020 2021

AGV2000
FD100 ✓
AIV100 ✓

✓CAMSUR2 RFU
AIV100 (Remote Firmware Update)
β-Test for CAMSUR2

Demo System 1000 units
CHN. (NJ) ~100 units
EU + Africa Not operational By End of 2020

2021: AI + Embedded + Robotics Market / NSF, NASA

CTI

11/

AIU100 — Productization/Demo
~~CHN~~
* EU Shipping – Mar 15–17
AF/BU — Demo *
SA

CAMSUR2 — Production Ready.
Phase I R&D Completed

AI + Robotics + Medical → Health (NSF)
Health Care → General Application
Construction J.S

1–2 yrs
Patent(s) ←
Airport
CHN HaiNan Hospitals F.G.

Focus on AI + Autonomous System (AGV 2000)

1° 2~6 weeks
CAMSUR2
AIU100 * (Demo)

2° AGV2000 – Jan 2021
Self-Driving
Lifting Cap.

(1) Full Embedded
Hardware/Device Drivers.
Akshat, Chee, NP, HL
RTOS/Ubuntu OS App.
Web Server.
Akshat, Chee, NP, HL
YY

(2)
GPUSUR
Deep Learning Implementation
Communication/managing
EMBNOD
YY, HL, NP?

(3) Deep Learning
HL, Graduate Research Assistants.
YY

(4)
System Software
GUI, Web, ~~Data~~ Base
(AIU100 Retool, Reuse)

Tasks:

1° Proximity Sensor On Line. (6)

2° LSM303 sensor (1)
On Line
X-Y-Z Acceleration
Magnetic Sensor

3° Optical Encoder
Back on Line (2)

4° 6+1+2 = 9+2
EmbNOD = 11
~~NANO~~
~~Pie3b+~~

5° Force Sensor (2)
(ADC) FD100

"Super Loop" → sequential Delay

CT1

12/

Feb22 (mon) SS, YY, HL

1. Visit Road map & Feedback

Concerned on the progress. (Behind)

a) Research Survey.

Chapter 1 By Wed.

Chapter 2. Survey. By Friday { Unity AI Repo. / Google Team's 6DoF DRL

Chapter 3. Methodology

Section 3.1 Unity API

3.2 Repo. Google Robot Control

Chapter 4. Implementation

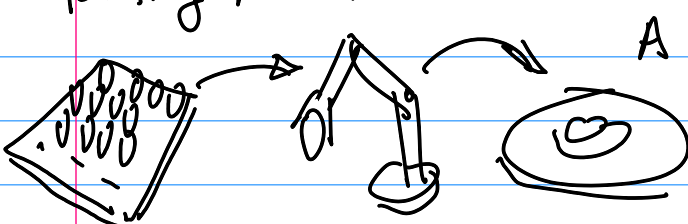
4.1 FD100 Robot

⋮

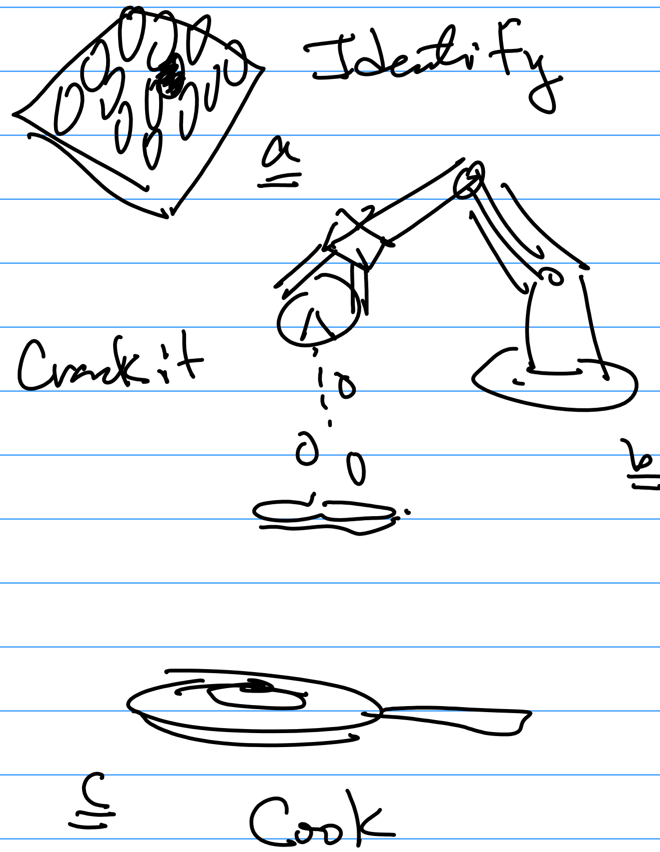
4.x? Experiment

Chapter 5 Experiments

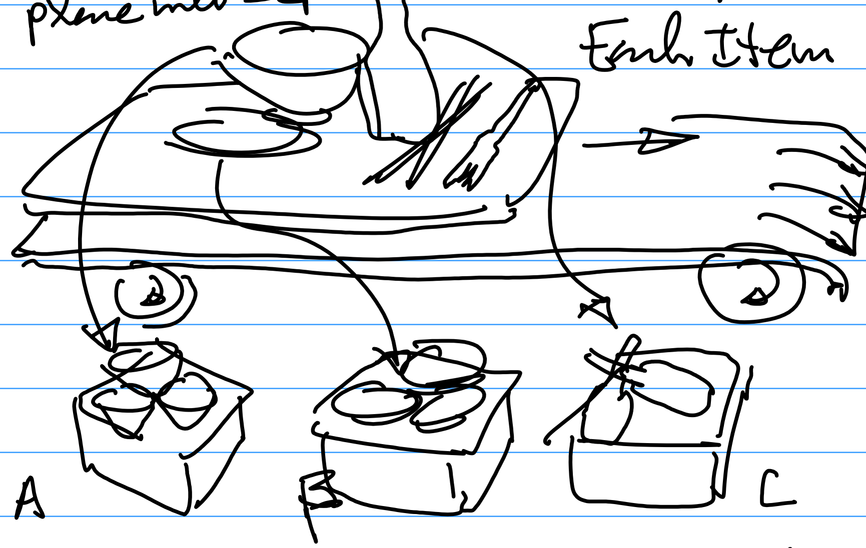
Testing Goal:



Hardware Side: { Gripper / Force Sensor



Option: Dump the food scrap plane into Separate bins Robot Grabs Each Item



Action 1: Repo Robot working Code

Action: Chapter 1 – Chapter 3 By Friday

CTI

131

Action 3 : Google paper
Repo — Code Testing

AGV2000 Teach mode

Feb 22. Embedded Group
NP, Akshit, Chee, YY, HL
Huan.

AGV2000 α-Version 2.0 prototype

Deep Learning
Physical Implementation

Kernel/Driver
LSM303
Pie 4
Step 1.

multi-threading Step 2
12~20 Threads

1° Proximity Sensors

Front

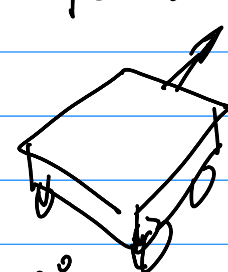


Fig 2

Step 1

Unity Paper → ML-Toolkit
↓
Pytorch (Ubuntu 18.04) Readme.
ML-Toolkit
↓
Run "Hallway Environment" + Kenting (Outdoor)
Unity

Note: HL to share Design Note

120°

Top

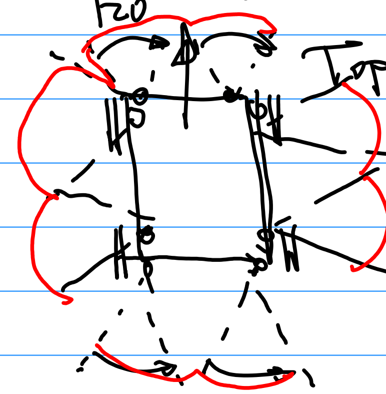
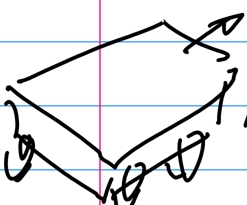
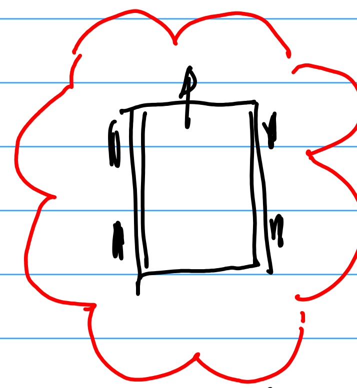


Fig 1.



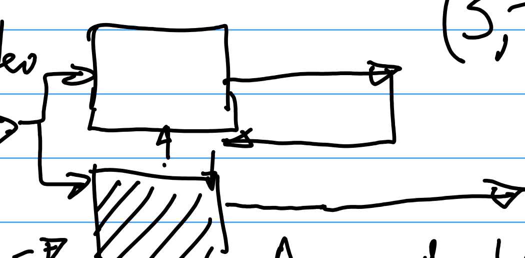
AGV2000 → In-door
Walk-Way 10-15%
80~85% Hallway



Capability Development "Dual-DRL"

Step 2

Video



Part of AIV100 ...

(S, T, A, R)
↑
Reward

Augmented DDRL

4 (sides) x 2 (sensor) = 8

Task 1. Pie 3B/4 I/F SEN

Task 2. multi-threading

C+I

Python Code for multi threading NP

Feb 25th Thur.

YY, Chee, Huan, Akshat

Embedded Group.

a Sensor Make/model HL.

b Program HL ARDUINO Code

c PCB (update) Add mounting HL Holes.

Fig 1.

d Enclosure + mounting Standardization RJ45

Home Depot, Junction Box NP $3.

- Pre-fab'ed
- Pre-Built Holes/Openings
- Size (PCB)

e Cabling — Embedded/mechanical Tubing/Pipe .../Zip Tie

Note: Connectors Have to Be provided RJ45 If fitted

1° Paper/Document Ready On-going Process

1.1 PPP: Template — Standardized.

Title; ID of the Document

Hardware:

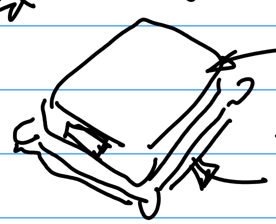
a System Block Diagram

b Schematics Connectivity Table

c Implementation Details

d photo of the System

f mounting/3D Printing Design. NP/Human



e Testing (Validation & Verification)

Optical Encoder

a-f Table for Each Sensor/Person/Time

b Program {Reference Code: Adafruit/SPARKFUN

Note 1° Ref URL should be provided

motor

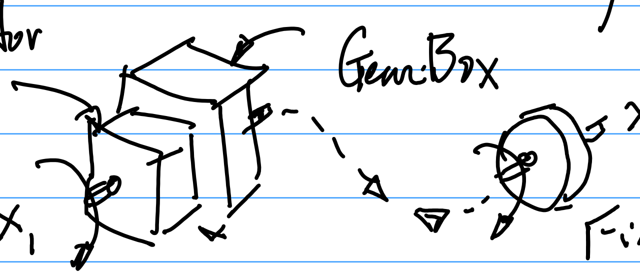


Fig 2-1

Coupling HL → Human

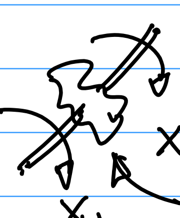


Fig 2-2

Coupler

Quality Control:
Item by Item Checking
Show & Tell (Group) — Presentation
Friday, Feedback: 1, 2, 3 (Highest)
One-on-One meeting (Akshat 1:30pm) → File for Reference

1.2 White Paper
Requirements
- Latex/Tex format, IEEE Template
- Document: Conference paper quality
- Corporate Internal Technical Report → C.V.

a. Detailed Information Match up P.R.T.
b. Sufficient Info for the work to Be Reproduced/Verified without the 2nd or 3rd Reference Source
c. Software
  - i. Verbalized Algorithm (~5 Steps)
  - ii. Flow chart(s) IEEE. Label Each Block. Match up to your algorithm
  - iii. Pseudo Code "must"

"I" (idea) → "L" (logic Flow) → "C" (Coding)

1.3 Source Code Joint Session Development/Debugging
- i. Readme.txt
- ii. Distribution of the Code
- iii. Environment/Tools/ Setup prep etc (Libs, packages)

Ranking: 1, 2, 3

1.4 Coordination/Implementation of the process
1.1 → 1.2 → 1.3
HL / YY / Group Leader
Embedded NP

- Optical Sensor / Sensor Interface Design
  - Feb 26. Akshat
- LSM303
  - Chee March 2nd.

→ Presentation, Show & Tell
Ranking: 1, 2, 3
Below, meet, Exceed

2. Path Planning — Deep Learning Group
PPT / White Paper / Revisit Code

Note: Create Tables for Naming Convention (Each System, Sensor).

① Situation Awareness (SA) → ② Motion/Trajectory Guidelines

Situation Selection (SS)

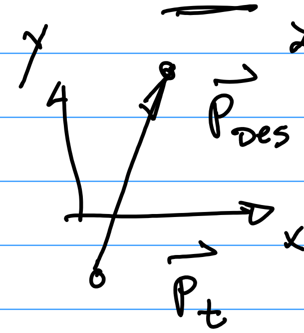
e.g. Hospital,
Indoor/Outdoor { Walkway, ⋮ }
→ Hospital
Nursing Home
Apartment

ID	Sub-ID	SubSub-ID
Ind 1	Apartment	
⋮	Nursing	
Out 1	Hospital	

Set/Define
1° Destination
$\vec{P}_{Des}(x, y, z)$
$\vec{P}_t(x, y, z)$
Current position

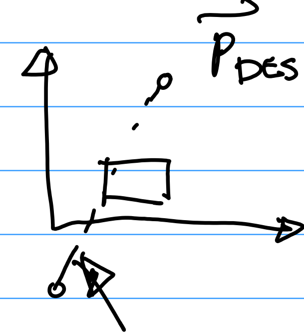
2° Shortest Distance Principle
SDP

Let $z = z_0$



3° Obstacle Avoidance

Step 1. Detection of Obstacle
↓
Step 2.



Traj shortest

Watch → Move → Eval
W–M–E

Note: Need 3D Environment mapping from Sensor Array

- Visual
  - Plain Color CAM (4-6)
  - Stereo CAM (2) STRCAM
  - LIDAR
- Non-Visual
  - Opt Sensor (2)
  - Proximity Sensor (8)
  - LSM Sensor (2)
  - ⋮

Blind Spots!

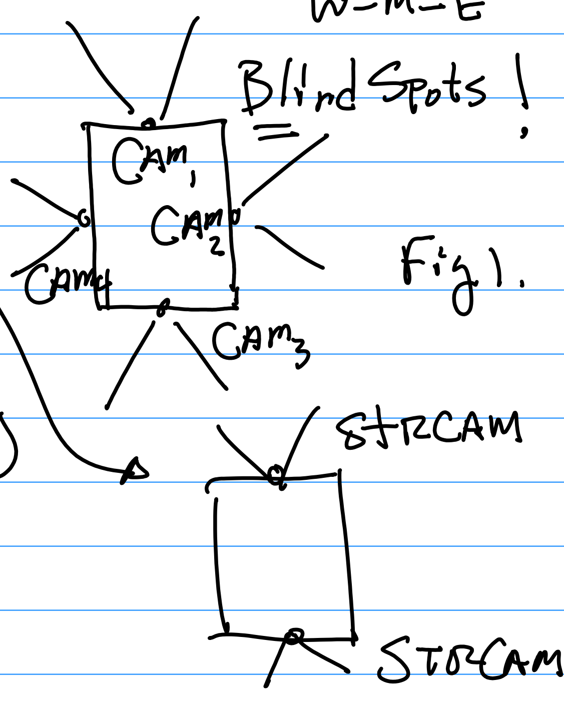


Fig 1.

W-M-E

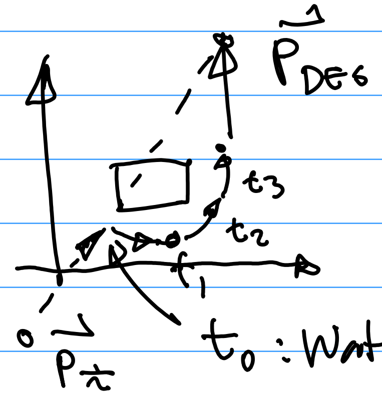
Move (Action), Consequence
Reward Function

Motion (Action)	Consequence (Reward)
A_1 : Right	
A_2 : Left	
A_3 : :	
A_4 : :	

1, 2, 3
Big, Medium, Small (Angles, $\alpha_B, \alpha_m, \alpha_S$)
Speed (fast, medium, Slow, Very fast), S_1, S_2, S_3, S_4

$$B \times S = \{\alpha_1, \alpha_2, \alpha_3\} \times \{S_1, S_2, S_3, S_4\}$$

$$= \{\alpha_1 S_1, \alpha_1 S_2, \alpha_1 S_3, \alpha_1 S_4; \; \ldots\ldots \; \alpha_3 S_1, \ldots \; , \alpha_3 S_4\}$$



Deep Reinforcement Learning Framework

Action	Reward	Heuristic
$\alpha_1 S_1$	R_{11}	H_{11}
$\alpha_1 S_2$	R_{12}	H_{12}
⋮	⋮	
$\alpha_3 S_4$	R_{34}	H_{34}

Stochastic Process

$$\sum_j \sum_i \mathrm{Prob}(R_{ij}) = 1$$

Note: Shortest Time is Competing Criterion

Unit - ML Toolkit

Feb 26 (Fri) NP, HL

1° Sequential Blocking Problems
System Configuration
Fig 3 pp. 10

Notation: EMBNOD — Embedded Node

$OPTSEN_i$, $i = 1, 2, \ldots, K$

Root: Sensor

Optical Sensor

LAN Environment

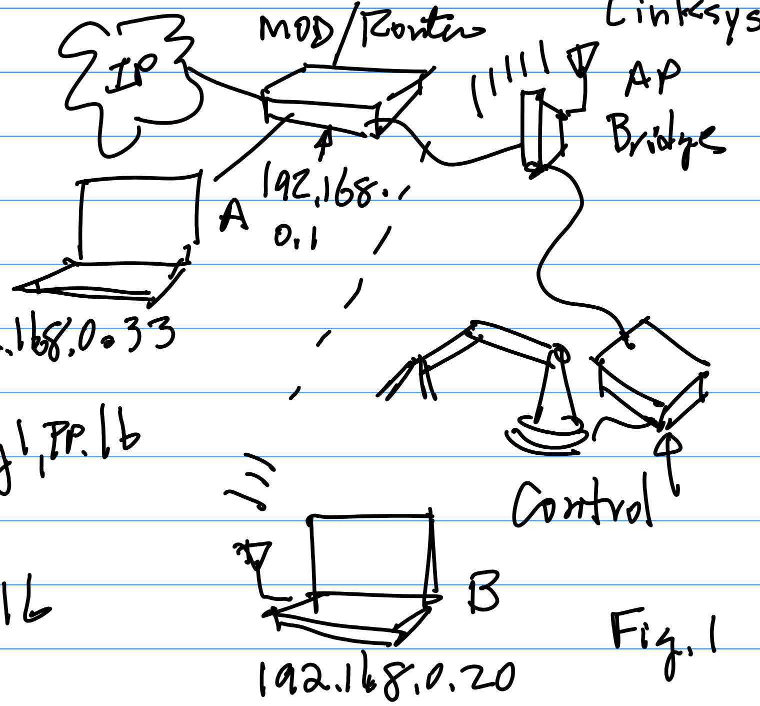


Fig. 1

$PRXSEN_i$, $i = 1, 2, \ldots, 6$;

Proximity Sensor

$USBCAM_i$, $i = 1, 2, 3, 4$ Fig 1, pp. 16

IPx CAM

$STRCAM_i$, $i = 1, 2$, Fig. 1, pp 16

Timing for Each Sensor

Let T_{OPTSEN} Time I/F OPTSEN

Sequential I/F

.10
.27
.33 A ✓
.11

T_{OPTSEN_1} T_{OPTSEN_2}

$$T_{\Sigma, OPTSEN} \cong 2T_{OPTSEN_1} \quad \ldots (1)$$

$$T_{PRXSEN_1} + T_{PRXSEN_2} + \cdots + T_{PRXSEN_6}$$

$$T_{\Sigma, PRXSEN} \cong 6T_{PRXSEN_i}\Big|_{i=1} \quad \ldots (2)$$

$$T_{USBCAM_1} + \cdots + T_{USBCAM_4}$$

$$T_{\Sigma, USBCAM} \cong 4T_{USBCAM_1} \quad \ldots (3)$$

Feb 26 (Friday) FD100

March 1, Monday.

Training ON DRL

Requirements

1° Homework ; 2°

QUIZ ON Each Lecture

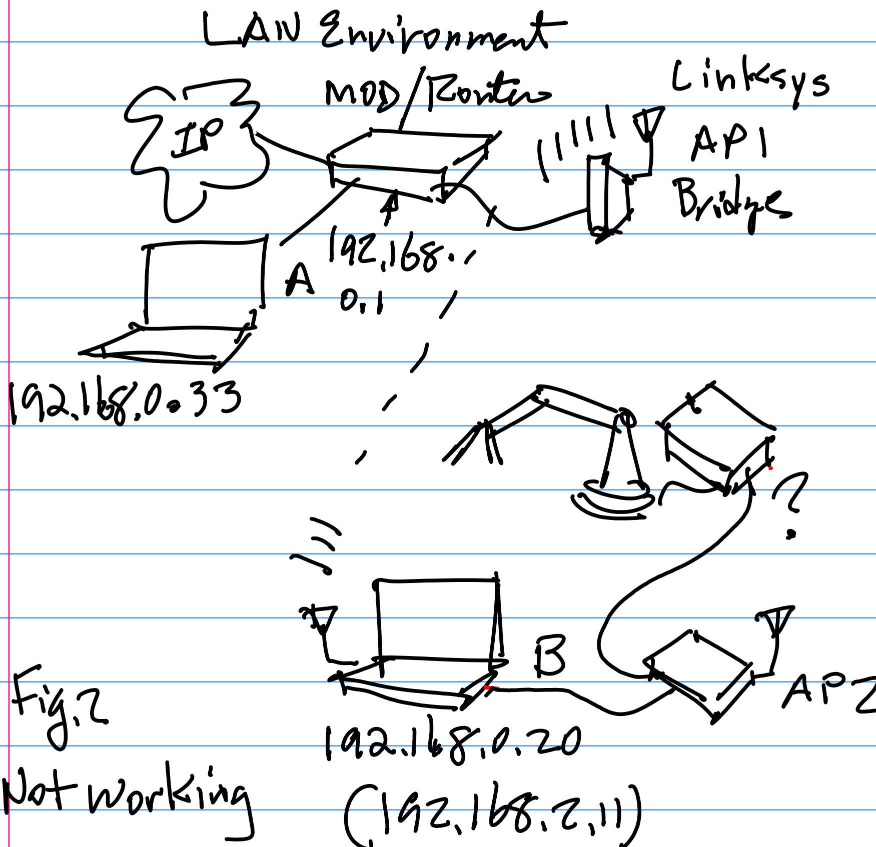


Fig.2
Not working

Objectives :

Optimize "π" Policy for Decision making.

a_i : Action, i index take I_n

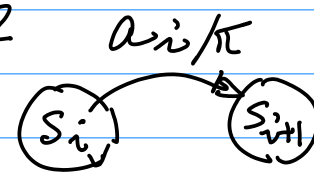


Fig. 1

S_i : State of the Robot

$\pi(a_i|S_i) \to S_{i+1}$

Action   Current State

T : Transition, function

r : reward associated with $\pi(a_i|S_i)$

Note : MDP → Markov Process { a memory, b "short" memory

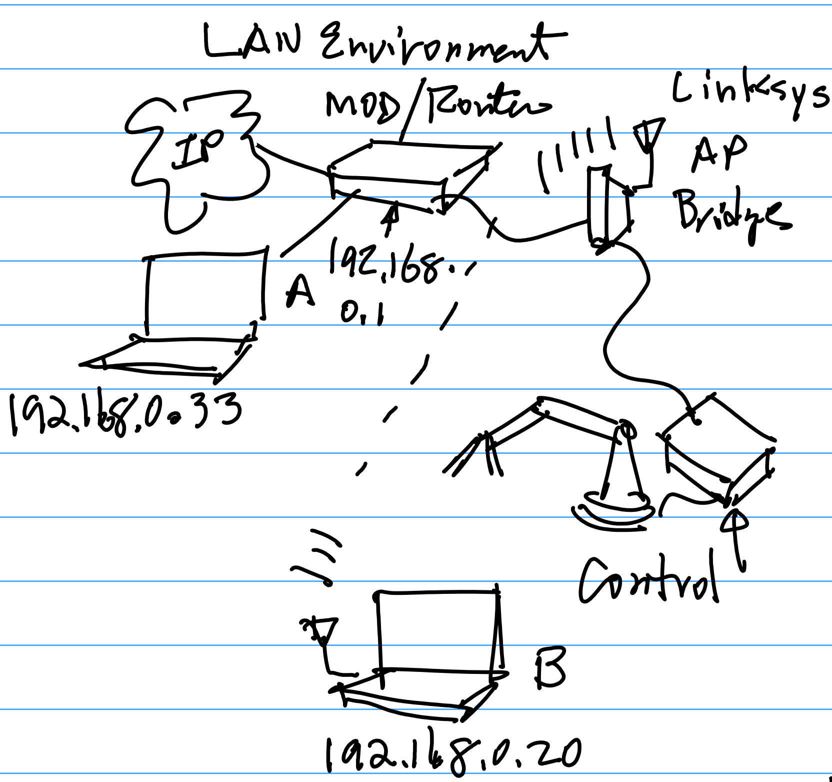


Fig.3

Note 2 :

$S = (s_1, s_2, \ldots, s_N)$ N: No. of states

S_1: init State, System Powered up

S_i: Robot Near an Object → Joints

$\text{Joint}_i \begin{cases} \text{Position} \\ \text{Speed} \\ \text{Torque} \end{cases}$ ← Joint_i ←

etc.

S_K: OPTSEN Reading.

"$<S, T, A, R>$" MDP

Pick + place → Joints → Joint_i

Motion Drive ← Embedded System EMBNOD$_1$ ←

Stepper motor

Drive the motor Action a_i

$\begin{cases} \text{PWM} & S_i \\ \text{OPTSEN} & S_{i+p} \end{cases}$.

$\underline{a_i \; S_i}$

$S \times A$ ... (1)

Notation $\text{Prob}_T(S, a, S')$ ... (2)

$$S \xrightarrow{a} S'$$

Current State   Next State

Question: Python Code for this

$\pi : S \to D$ (Final Destination State)

mapping

$\pi(a|s)$

$R(S_K, a_K)$ K time

Reward

γ: Discounted factor

$$\sum_{K=0}^{H-1} \gamma^K R(S_K, a_K) \quad \ldots (3)$$

$$= \gamma^0 R(S_0, a_0) + \gamma^1 R(S_1, a_1) + \gamma^2 R(S_2, a_2) + \cdots + \gamma^{H-1} R(S_{H-1}, a_{H-1})$$

$$\text{Exp}\left[\sum_{K=0}^{H-1} \gamma^K R(S_K, a_K)\right] \quad \ldots (4)$$

"$\underset{\pi \in \Omega_\pi}{\arg\max}$" $\{ \ldots \}$

Maximization "Max"

Indep Variable π from all Ω_π

Note: f function, from Eqn (1)   Notes 105

Max(f)

↓

$1/f$ ← min$\{f\}$

Shifu: 1° Chapter 3 math. Formulation of DRL (MDP)

2° Experiments:

$S = (S_1, S_2, \cdots, S_n)$

Class 1: Joint Angle, Joint Speed.

$S_{jx,\alpha}$

Joint x

$x = 1, 2, \cdots, 6$

$S_{jx,spd}$

[10%, 50%]

Class

$S_{cv\,x_0}$ — Joint x OR Entire System

Computer Vision, Pose

3D Environment

Upper
middle
Lower
Position

1. Joy: Biz Dev.

Mike: 34 yrs old

Website

2. Demo, Video

Capability Demo

3. Ask what Services to provid.

Ares — Contract 40x80, 80x120

Inspection

Drone. Video Streams, OpenCV, ROS

Cracks, Falling Part

4. Professional Experience

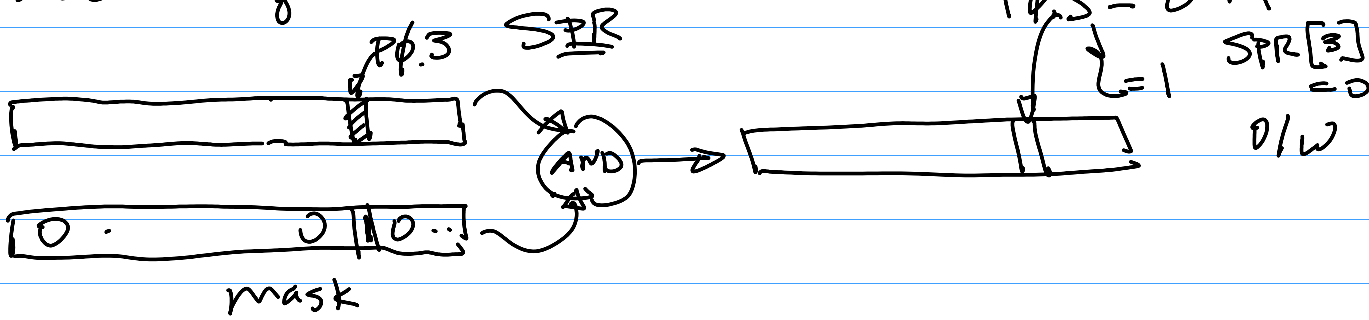
5. Service Biz, Hardware

GPU Portable System

Robotics

6

Thesis On Daily Basis



Farai Munyoro 津巴布维,纳米比亚 March 2nd (Tue)

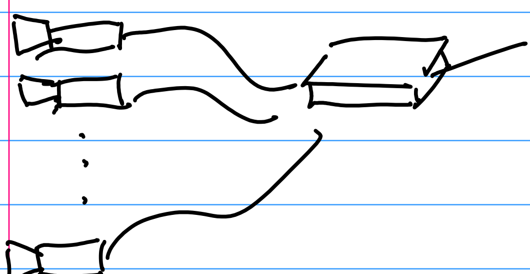
Electrical Contractors
CCTV
Water Utility
Telecom
Tourist Industry

1° Demo: Dual Systems
Sunnyvale site, Site 1
Palo Alto, Site 2
2° Prepare Tutorial/Training
Network Setup.

Cam
Hotel
Shop

Fig 1. AIV100 Network
8 Ports
16-24 ports
Router

Small System ≤16 CAMs
medium System ≤255
Large System ≤1024



a Network Topology. b Equipment
Routers + PoE (Power Over Ethernet)
CAMs, AIV Software, GPUSVR

Teaching — E-faculty.
March 19 (Fri)
Post Tenure Review
SOTE, Other Material
Periodic Evaluation of Tenured Faculty 97-5
Magda, Harry, Weider

1st, Not 0.

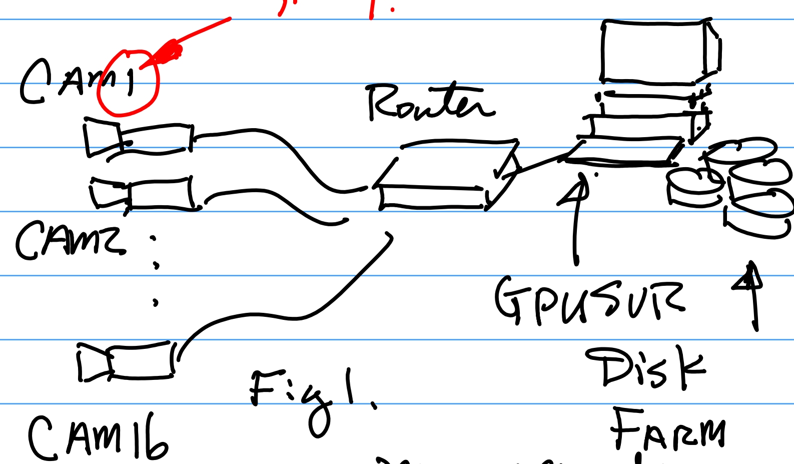


Fig 1.

management

EU Conference Talk
(1) Security/Surveillance
(2) IoT Engine Room/Control Room, Visual Feedback
(3) Analog meter/Smart meter

CTI

23

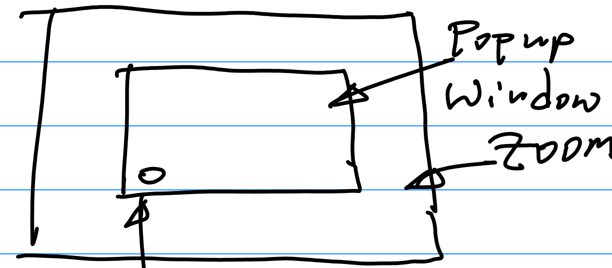
(4) RUL (Remain Useful Life)
Diagnosis/Prognosis
Based On Large Sensor Array.
Kalmain Filter Based Approach
Maintainene/Failure Prevention
Parts Replacement/Evaluation etc.

YY/HL Evaluation for Displacement Detection

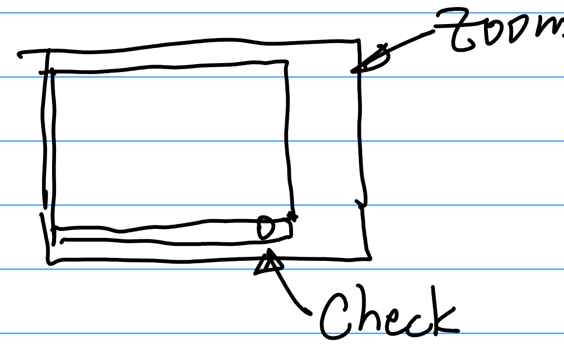
	Features	Bugs/Interruption
SCORE 3	7 features (all)	No, NO
2	6 or more features	No Bug/No Interruption (One minor Bug or One minor Interruption)
1	Less than Half of the features	More One Bugs/Interruption

Today's Demo : 1/3 { (1) Displacement Bug; (2) Audio Issue }

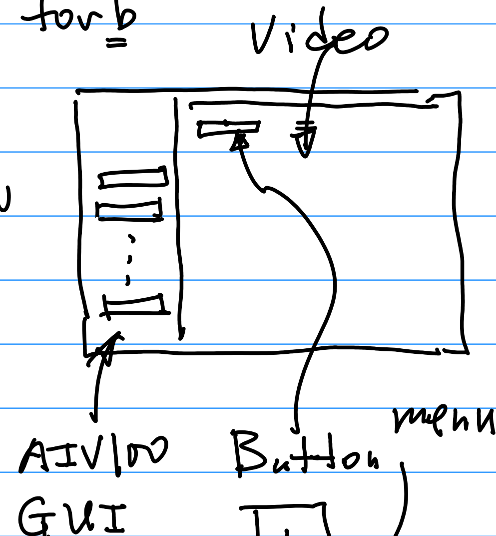
Audio Issue
{ a Share → then Select Audio
{ b Speaker @ player is ON

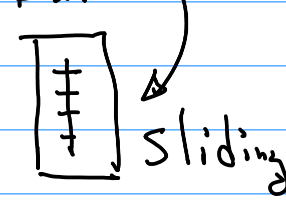


Select Share Audio for a

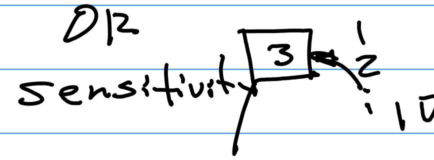


for b



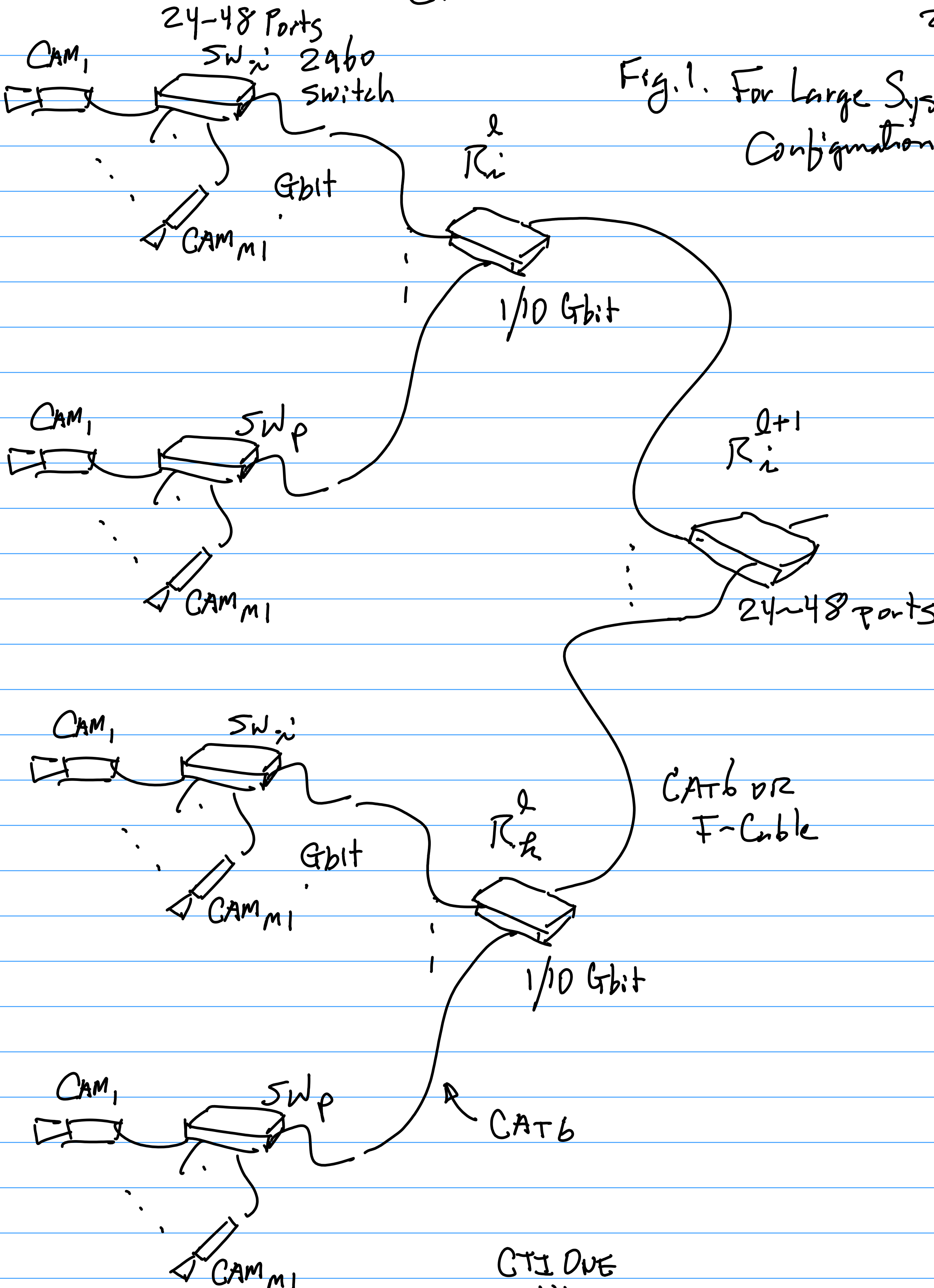


OR



CTI

Fig. 1. For Large System Configuration 3

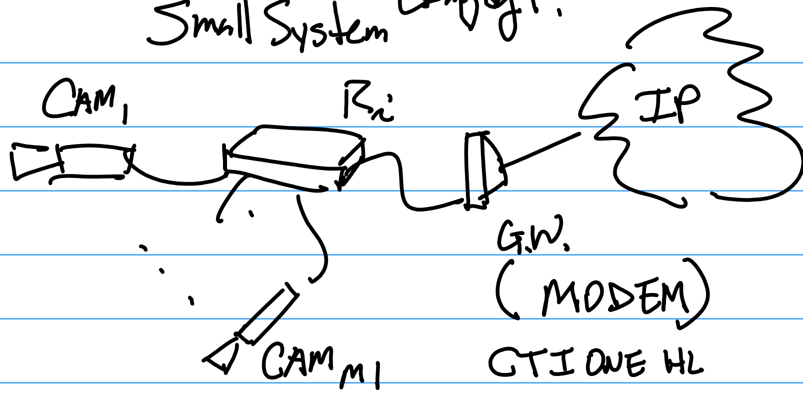


CTI ONE
ITL

ZS8pAz

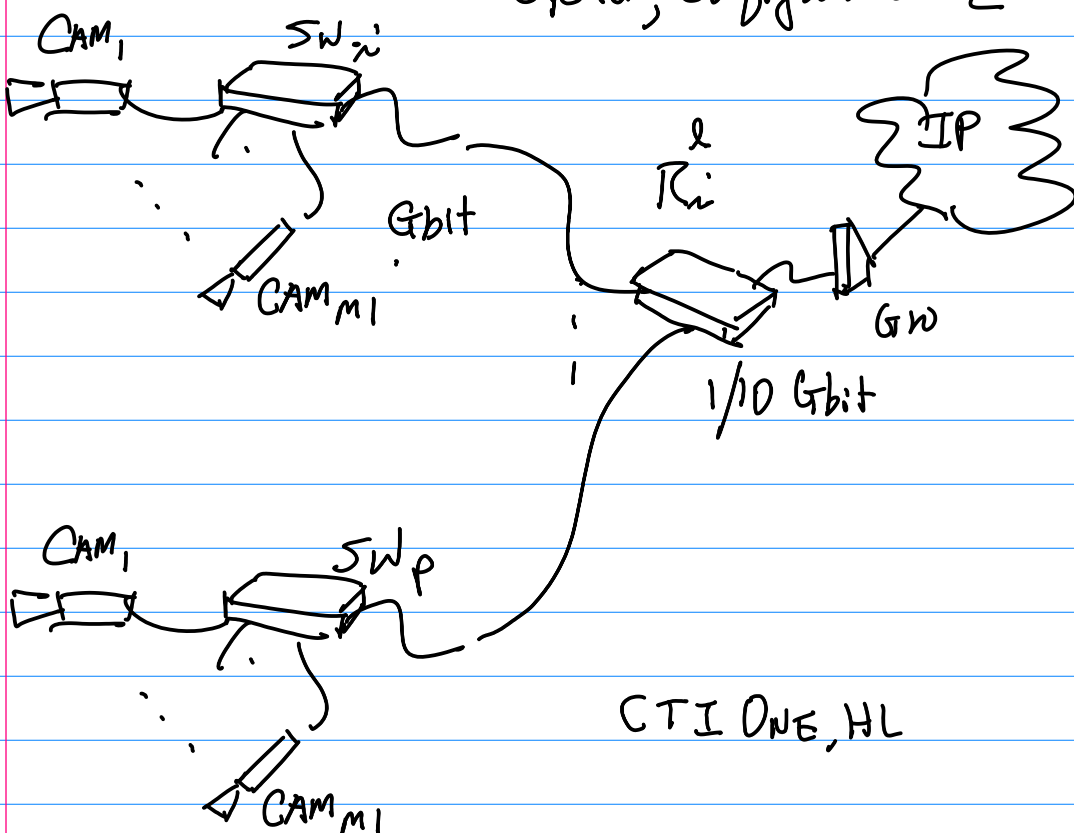
CTI

Small System Config 1.



CTI ONE HL

medium System, Configuration 2



CTI ONE, HL

Video CAM Setups for FD100

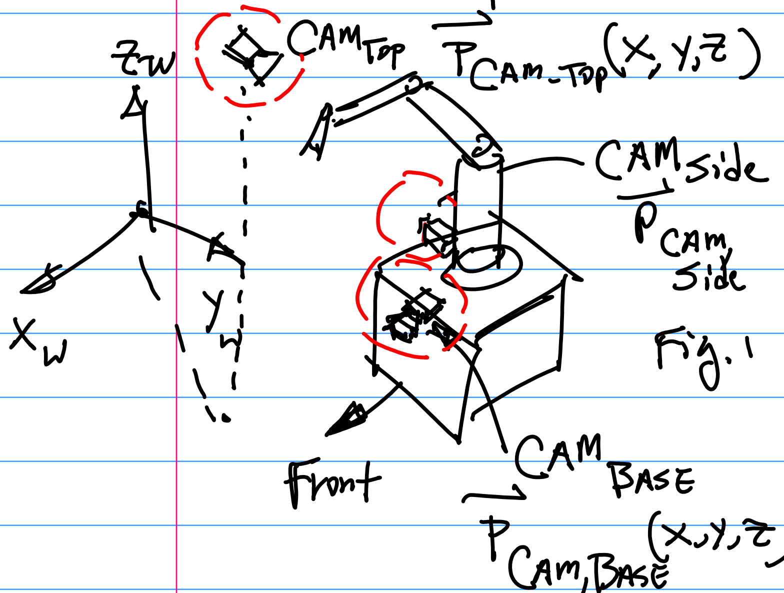


Fig. 1

F Stop, $F \triangleq f_d / D$ ...(1)

f_d : focal Length

D : Lens Diameter

March 3rd YY, HL

Fig. 1

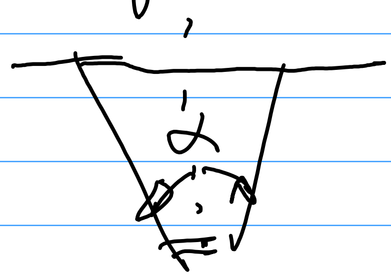
FD Interface to BD

AIV 100 CAM Distance

Optic Lense Selection

a. Focal Length mm

b. Field of View Angle



c. Field of View

Lense Type

C-type (in 32 thread)

CS-type

ALC (Automatic Lense Control)

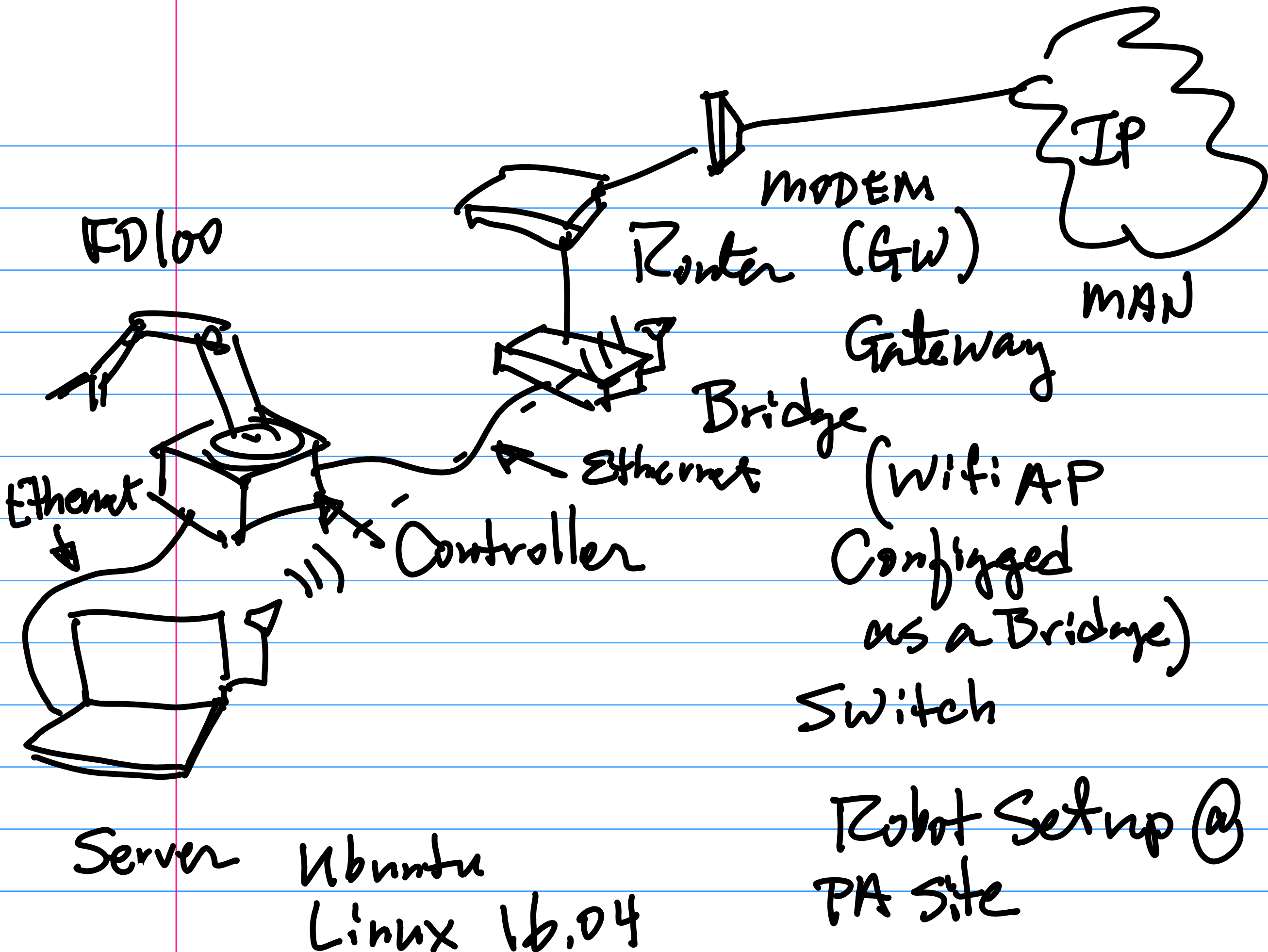
MS Thesis meeting w/Shifa

FD100 Robot Side

1° Python Networked Control Testing Done Last Friday, HL

2° White paper — Python Code, HL, Shared

3° Run Remotely



Robot Setup @ PA Site

Fig 1

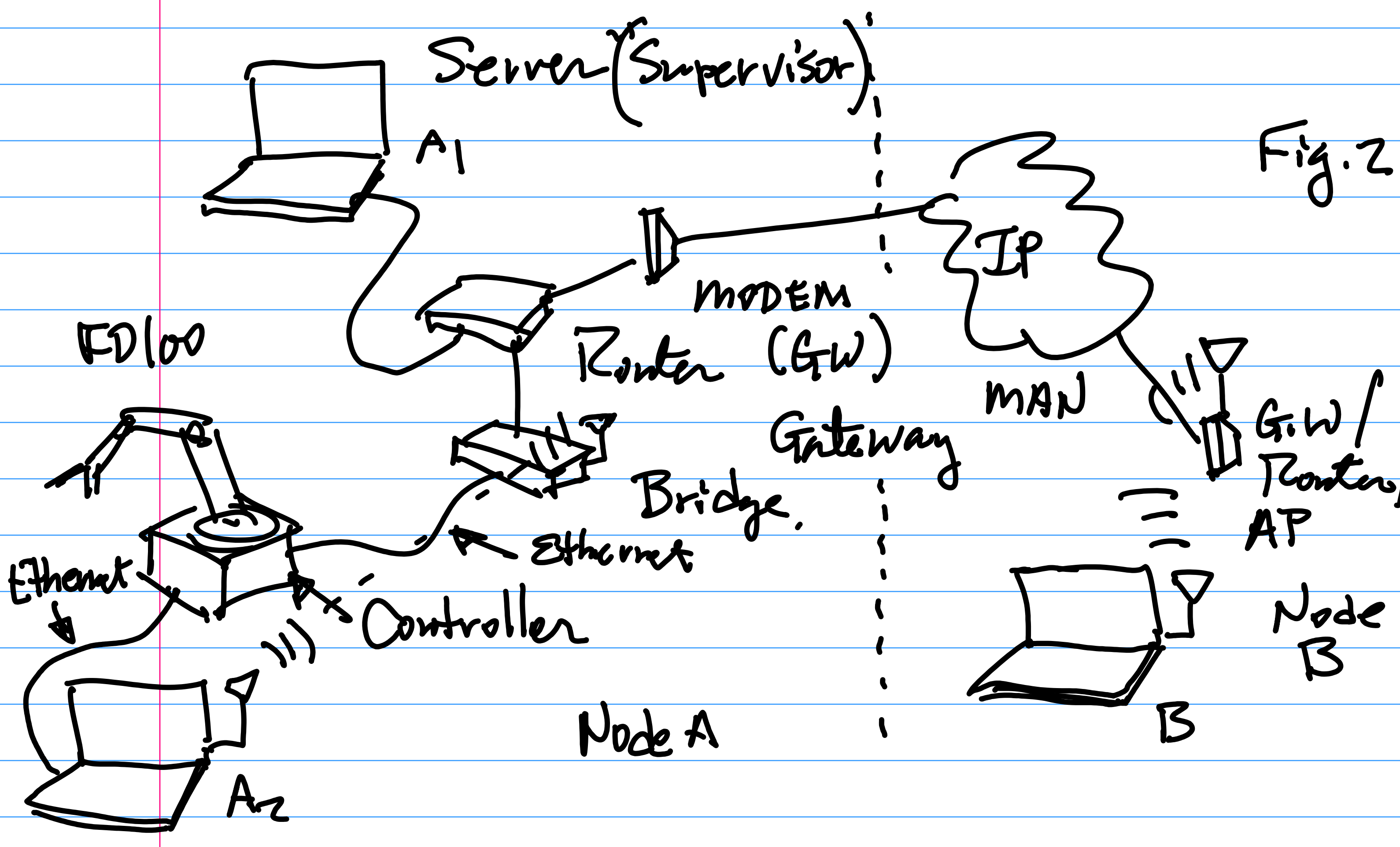


Fig. 2

Node A: (Laptop A_2)

a NM - Program (NoMachine) Running to establish Connection to $A_1 \rightleftarrows A_2$ Interface A_1 to control A_2

b Python Program to Run FD100 Robot

Laptop A_1

a NM Program to Control Laptop A_2

b Zoom to B1

c B1 → A_1 → A_2 → FD100

4° Objectives DRL Code Running github

5° { 1.3 Thesis
1.4 Combine them

{ 2.4
2.5 Combine, → AI HVR Development

3.3. Rework

Note:

Run Github Simple DRL Code

{ option 1. Code Reference 1
option 2 Code from Reference 2

March 3rd CmpE240

SPI LCD Pin Connection

1° Datasheet

2° github Reference Design

3° Lecture Notes

↓

Basic Concepts

a C/D Commands
Data

b multiplexed

Toggle B/W "0" & "1" to select One of the two functions

c GTP

d Target Board { P1
P2

↓

Simple Code

RISC-V Research Roadmap.
EE242/EE297

A ⟶ B

↑ Computer Vision AI

{ Ethernet Controller — Wifi ✓ Controller (N/A)
\ 2 IP cores Existing
Graphics Engine
Computer Vision { General Computing — Floating Processor
CAM → Digital CAM I/F → Hardware
OV7670 }

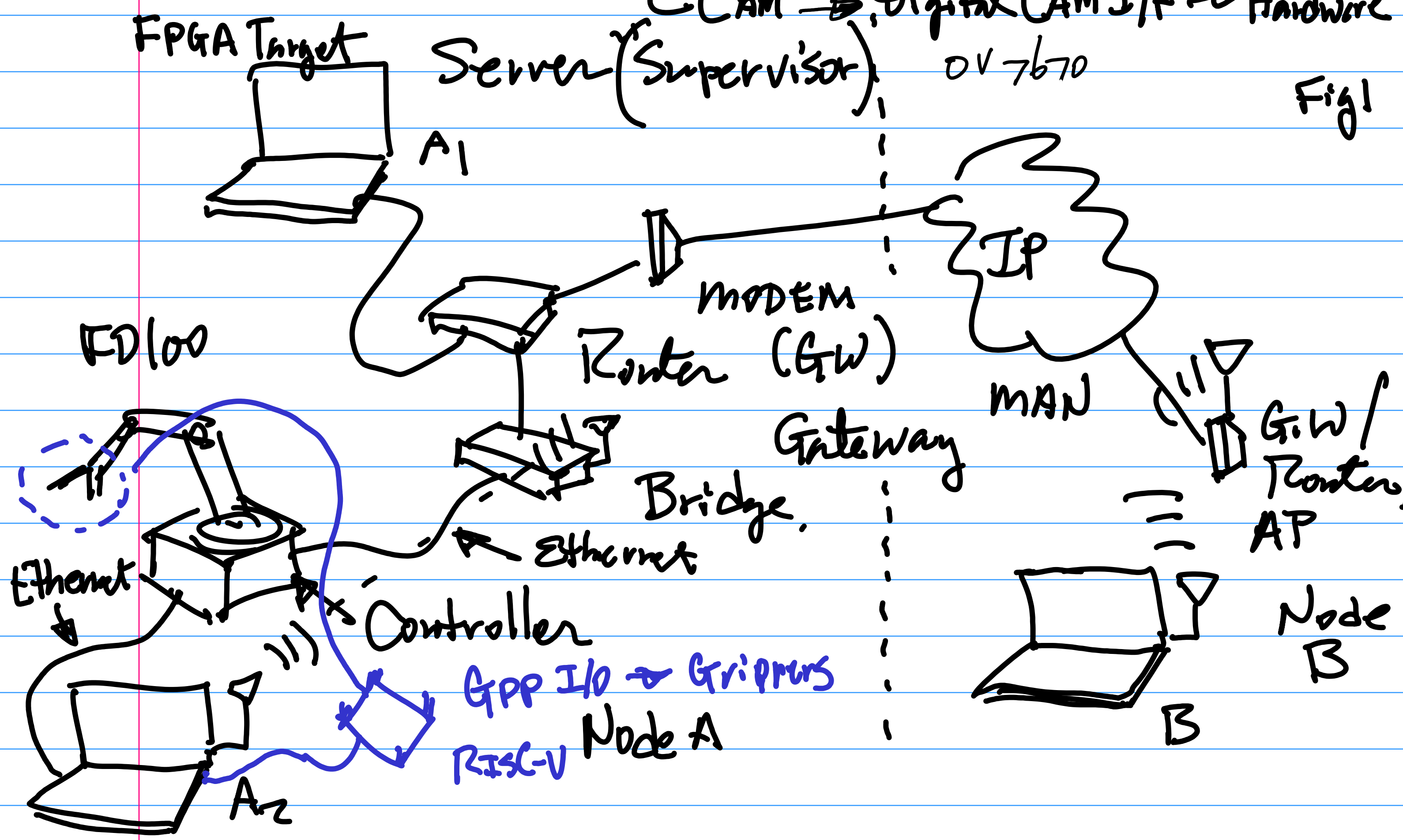


Fig 1

Video CAM Setups for FD100

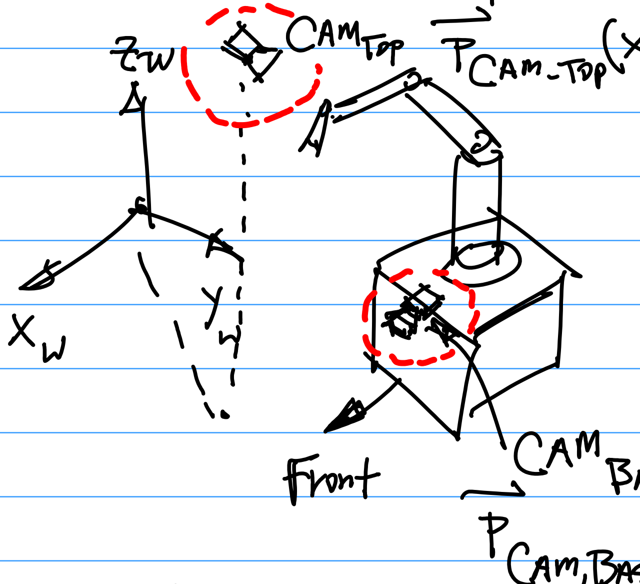
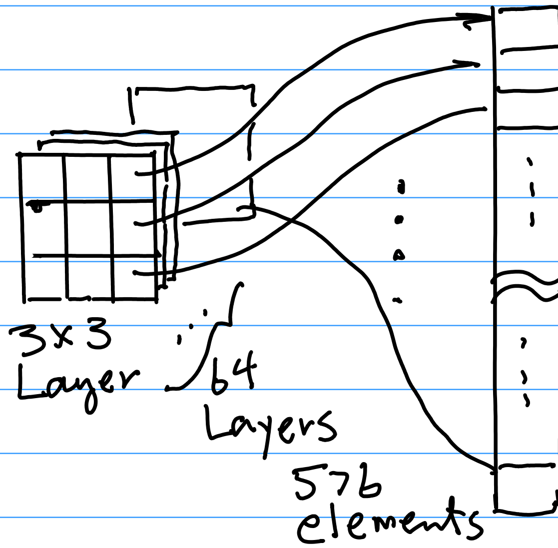


Fig. 1

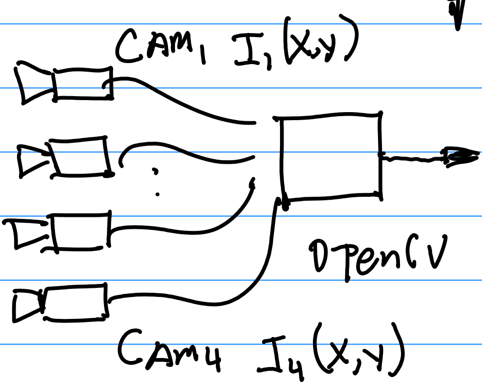
MNIST flatten Layer

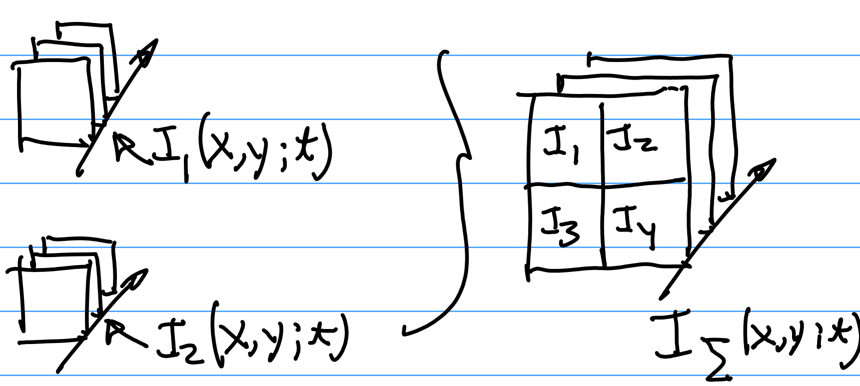


multi-CAMs

RISC-V → FPGA

Python OpenCV
↓





March 5 (Fri)

1° DeadLine HL Check
2° Template HL Junie/Urbano
3° E-faculty — Brandi;
4° Persons to Reviewed (5) Jerry, Simon, Rod, mo, Ahmet
5° Committee Evaluation
Comments — GuideLine
University policy

Next meeting: Next Monday 5:30pm. Zoom

March 6 AI IA
Zach, VanKand
Indexing AI Effort

March 7 (Sun) DPG
Deterministic Policy Gradient
Google DeepMind Team, London
2016 Paper ICLR2016 by
Timothy P. Lillicrap et al

Define E as a environment in Discrete TimeStep t

x_t as an Observation,
a_t as an action, such that real valued $a_t \in \mathbb{R}^N$
r_t as a scaler reward
And state is defined as an action pair

$$S_t = (x_1, a_1, \dots, a_{t-1}, x_t) \quad \dots(1)$$

Environment is fully observed, so

$$S_t = x_t \quad \dots(1\text{-a})$$

π: Policy to define agent behavior

$$\pi : S \to P(A) \quad \dots(2)$$

States (State Space); Probability distribution of Action

The mapping is modeled as Markov Decision Process, (MDP)

Action Space $A = \mathbb{R}^N$ ...(3)

initial state distribution

$$Prob(S_1) \quad \dots(4)$$

transation distribution

$$Prob(S_{t+1} | S_t, a_t) \quad \dots(5)$$

$$S_t \xrightarrow{a_t} S_{t+1}$$

Reward $r(s_t, a_t)$

$$R_t = \sum_{i=t}^{T} \gamma^{(i-t)} r(s_i, a_i) \quad \ldots (6)$$

Start distribution

$$J = E_{r_i, s_i \sim E, a_i \sim \pi}[R_1] \quad \ldots (7)$$

And visitation distribution for a policy π as

$$\rho^{\pi} \quad \ldots (8)$$

Action Value Function

$$Q^{\pi}(s_i, a_i) = E_{r_{i \geq t}, s_{i > t} \sim E, a_{i > t} \sim \pi}[R_t \mid s_t, a_t] \quad \ldots (9)$$

"$\sim$" \sim : Exact Asymptotic equality.

Bellman Recursive Equation to Solve Action Value function in Eqn (9)

(see Next Page)

$$Q^{\pi}(s_t, a_t) = E_{r_t, s_{t+1} \sim E}\left[r(s_t, a_t) + \gamma E_{a_{t+1} \sim \pi}\left[Q^{\pi}(s_{t-1}, a_{t-1})\right]\right] \quad \ldots (10)$$

$t-1$ (Note google Deepmind Typo) pp.3

$Q^{\pi}(s_t, a_t)$ Action Value function Depends on Policy π, So

$$Q(s_t, a_t \mid \pi) \quad \ldots (11)$$

In deterministic Case, (Policy)

$$\mu : S \to A \quad \ldots (12)$$

Policy (Deterministic)

mapping to Action

Note Google Deepmind Typo PP3

So Action Value function

Q^{π} Rewritten as Q^{μ} ... (13a)

And

$$Q^{\pi}(s_t, a_t) \to Q^{\mu}(s_t, \mu(s_t))$$

Deterministic Policy, Eqn (10) Changes
to Action-Value Function: (e.g., Reward Function)

$$Q^{\mu}(s_t, a_t) = E_{r_t, s_{t+1} \sim E}\left[r(s_t, a_t) + \gamma Q^{\mu}(s_{t+1}, \mu(s_{t+1}))\right] \ldots (14)$$

So, the Objective is to find Policy μ, such that $Q(s,a)$ maximized

$$\mu(s) = \mathrm{argmax}_{\underline{a}}\, Q(s,a) \quad \ldots (15)$$

Deterministic Policy

As a function of State Space

maximize Reward, e.g. Action-Value function Q over Action Space, or action a

Note: Not very good notation by Google Deepmind Team, Because they are using Watkins & Dayan 1992 Notation. I think.

Note: μ is not a policy here, it should be written as "Q_{max}" Reward, e.g., Action-Value function.

To find Q_{max}, e.g. Maximize Reward function, Let's define Objective function, e.g. Loss function:

$$\text{Loss Function} \triangleq \left(\overset{\text{unknown}}{\text{Max Reward Function}}\right) - \left(\text{Function Proximator Function}\right) \quad \ldots (16)$$

$Q^{\mu}(s_t, \mu(s_t))$ $\qquad y_t$

Rewrite $Q^{\mu}(s_t, \mu(s_t)) = Q(s_t, \mu(s_t) \mid \mu)$
as in Eqn (11), pp. 31. And let y_t function Approximator
Parameterized by θ^{Q}, so $\mu \triangleq \theta^{Q}$

So,

$$\text{Loss Function} \triangleq \begin{pmatrix}\text{unknow}\\ \text{max. Reward}\\ \text{Function}\end{pmatrix} - \begin{pmatrix}\text{Function}\\ \text{Approximator}\\ \text{Function}\end{pmatrix}$$

$$= Q(s_t, a_t) | \theta^Q) - y_t \quad \cdots (17)$$

$\stackrel{a}{=}$ Square it to take care of Potential +, − Cancellation

$$\left(Q(s_t, a_t | \theta^Q) - y_t\right)^2 \quad \cdots (18)$$

$\stackrel{b}{=}$ Statistic Sense, Expectation,

$$E_{s_t \sim \rho^\beta, a_t \sim \beta, r_t \sim E}\left[Q(s_t, a_t | \theta^Q) - y_t\right]^2 \quad \cdots (19)$$

Denote Loss function as

$$L(\theta^Q) \triangleq E_{s_t \sim \rho^\beta, a_t \sim \beta, r_t \sim E}\left[Q(s_t, a_t | \theta^Q) - y_t\right]^2 \cdots (19a)$$

//

March 8

Roadmap: (A) Unity AI { Robot / Self-Driving

↓

Algorithm Enhancement

↓

Github Coding/Program

↓

Formulation

(B) Testing Platform ←

FD100 Vision/Python Control

Allen, Working

Today's Topics:

DRL → Google Paper

DPG

Deterministic Policy Gradient

Ref: 1° White paper

102-5 FD100, HL

2° Github code ✓

3° Paper from Google Deepmind Team

Soft-Actor Critic Algorithm
SAC

Note: URS_DRL → Source Code
DDPG.py → Torch from Facebook

Note 1° Action a_i, time i

Actual New Action is from a_i, "hard update"

2° Actions a_i, a_i'

$\tau a_i + (1-\tau) a_i' \rightarrow$ New Action

$\tau \in [0,1]$ weighting factor

$\tau = 0.3$

$0.3 a_i + 0.7 a_i'$

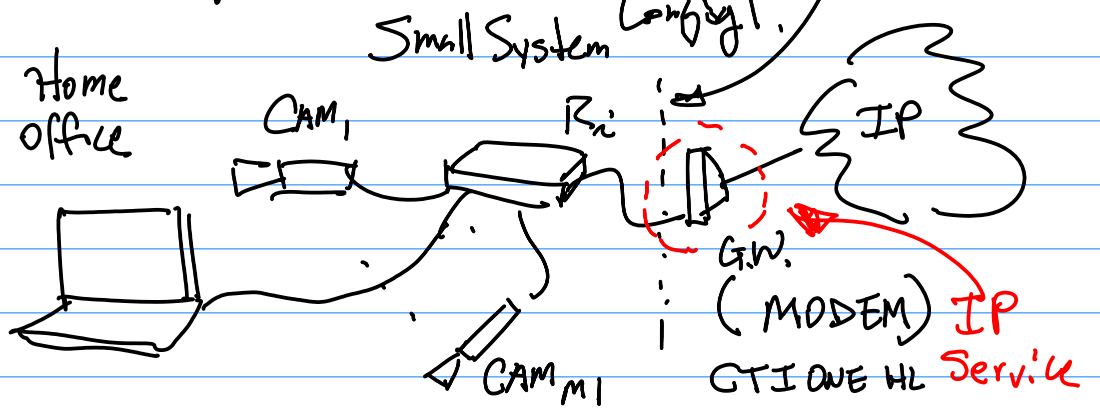
$\tau = 0.5$, Equal Contribution from Both.

Now, Computational perspective for DPG.

From Eqn (19a)

March 9 (Tue)

AIV100



Small System Config 1.

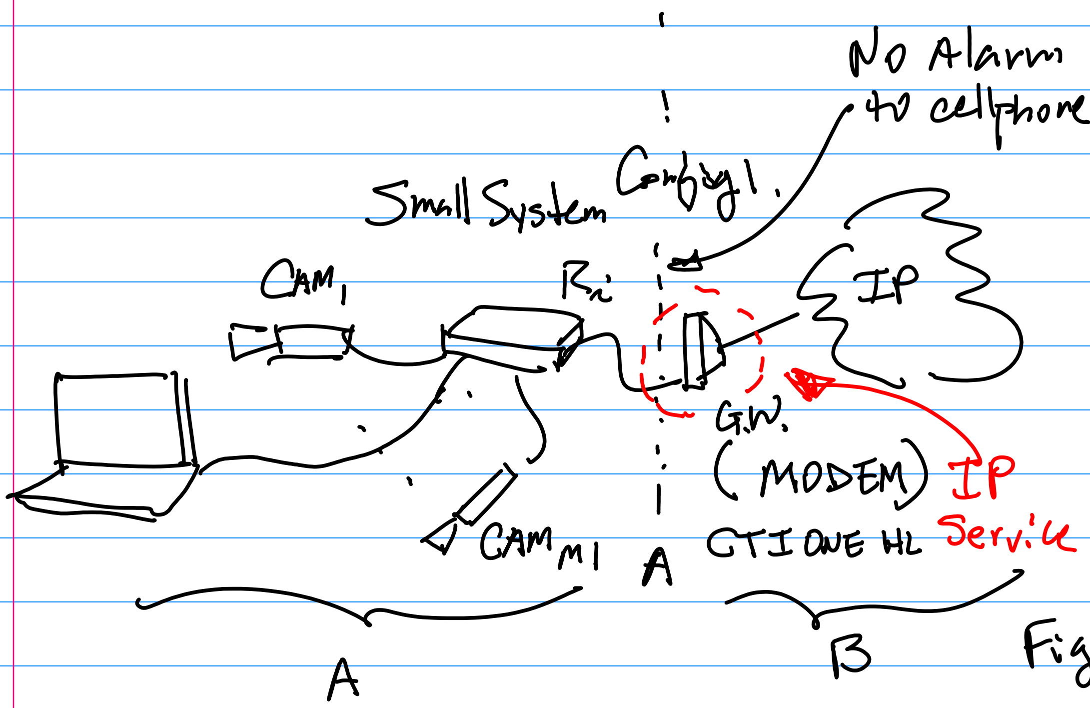


Fig 1

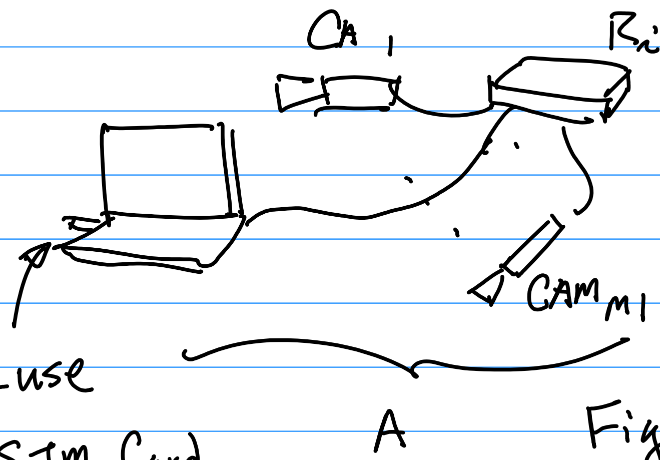


Fig 2

Pay_per_use
USB+SIM-Card
I/F

* B.O.M.
2 CAMs
1 AIV100
1 S/W
1 USB-SIM
Storage Capacity
Cam Features + Configuration
Switch/Router
Cables

$40 for 4~5 G Data for over a year

* CheckList
* Storage 1TB, ...
* Security Solution
* Use-Case for 2,3 Branch offices

To check memory. Need to let user know how — Description/photo yy

Actions
1. Engineering Notes
   a. Small 2 CAMs household Application HL
   b. 2-3 Branch offices Application HL
   c. Debugging — show Log
   System monitoring tool →

2. GUI Main page photo — Color YY
3. Displacement Sensitivity Testing
4. Upgrading Sensitivity 1~2 hrs.
   a To let user know, HL
   b Deep Learning to change Sensitivity level HL, YY
5. USB-Tangle SIM-Card Solution. HL
6. Recording/Downloading Video YY
7. Fix the program start up issu. GUI Start with 127.0.0.0:8080 ⇓ Then GUI Button to start manage.py.

Software
(1) Algorithms 5 step
(2) Flowcharts
(3) Pseudo Code
(4) Code Segment
(5) V & V Validation & Verification

Action Item:
1. 2 CAM multi-Threading Testing Case ~120 FPS
2. Integration with Deep Lerning Testing Program

Yolo4 github

March 10 (Wed)

Embedded Group, Nitin, AKshat, HL

Multi-Threading → "Super Loop" Sequential Blocking

Nitin: Implement Multi-Threading Function — white

Hardware (1) System Block Diagram
(2) SCH
(3) Waveforms, Photos