Adaptive Power Management in Solar Energy Harvesting Wireless Sensor Node using Reinforcement Learning

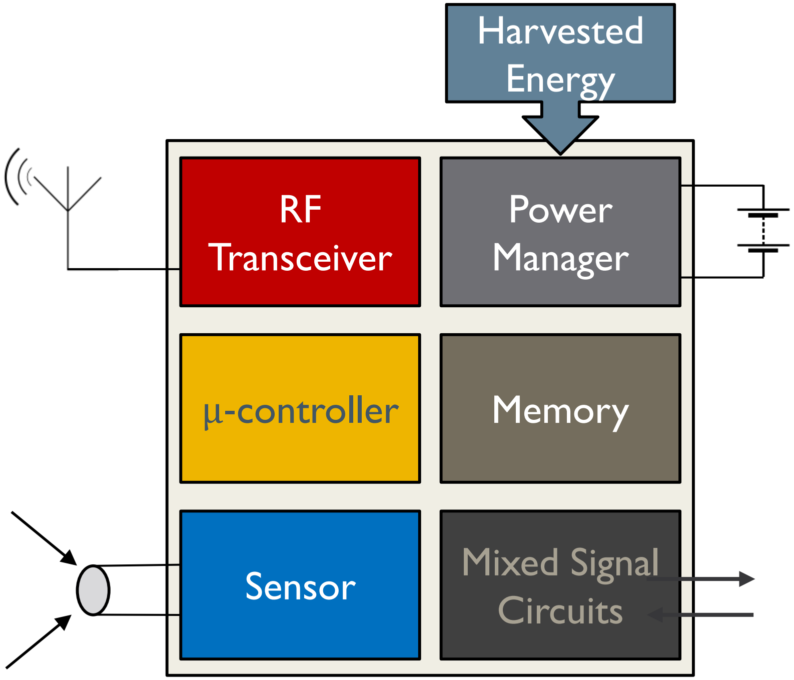
SHASWOT SHRESTHAMALI
MASAAKI KONDO
HIROSHI NAKAMURA

THE UNIVERSITY OF TOKYO

EMBEDDED SYSTEMS WEEK, EMSOFT 2017, SEOUL

INTRODUCTION

- Energy Harvesting Wireless Sensor Nodes (EHWSNs) are wireless sensor nodes with
  - An energy buffer (battery)
  - Energy harvesting module(s) (e.g. solar panels)
- IoT will require (trillions of) diverse sensor nodes deployed in different environments.
- Sensor Nodes should work autonomously and perpetually.
  - Maximize utility of sensor node
  - Sustainable and maintenance-free



Block diagram of an EHWSN

PROBLEM DEFINITION

Perpetual operation and maximization of sensor node utility can be achieved if:

ENERGY HARVESTED = ENERGY CONSUMED

Node Level Energy Neutrality

THE PROBLEM

- **Unreliable energy harvesting**
  - Unpredictable energy profiles
  - Predictions are unreliable
- **Strategies change with changes in environment**
  - Change in location
  - Change in climate
  - Change in device parameters
- **Scaling**
  - Billions/trillions diverse sensors deployed in unique working environments

PREVIOUS APPROACHES TO ACHIEVING NODE LEVEL ENERGY NEUTRALITY

Research	Approach	Limitations
Power management in energy harvesting sensor networks, Kansal et. al (2007)	Predict energy to be harvested and determine duty cycle	Performance dependent on prediction mechanism
Adaptive control of duty cycling in energy-harvesting wireless sensor networks, Vigorito et. al (2007)	Linear Quadratic Control System	Hyper parameters need to be manually adjusted
A learning theoretic approach to energy harvesting communication system optimization, Blasco et. al (2013)	Reinforcement Learning	Applicable for sensor nodes with communications as the only power consuming operation.

SOLUTION

Hand-engineered solutions for all possible scenarios is **impractical**.

We want a **one-size-fits-all** solution i.e. sensor nodes that:

- **learns** the optimal strategy through
  - Context aware action – perception – learning cycle
- **adapts** once they have been deployed in the environment.

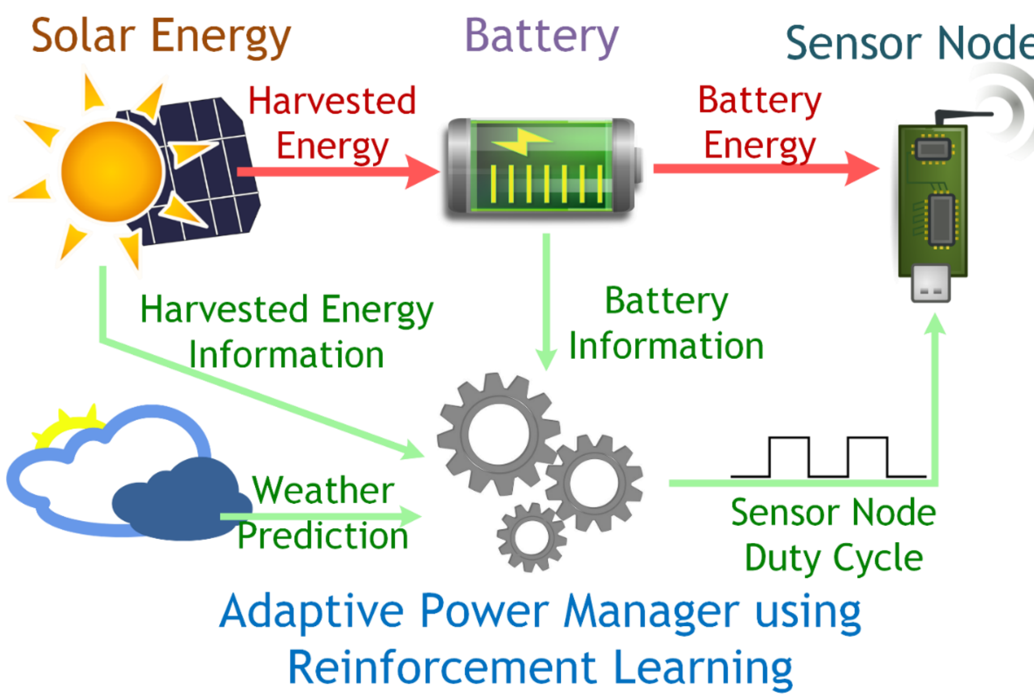
PROBLEM DEFINITION

Perpetual operation and maximization of sensor node utility can be achieved if:

- ENERGY HARVESTED = ENERGY CONSUMED
  - Node Level Energy Neutrality
- Battery is never completely full or depleted
- Sensor node maintains a minimum level of operation at all times
  - Duty Cycling

SYSTEM MODEL

- Solar EHWSN
  - a load that consumes power depending on its duty cycle
  - Higher power consumption implies higher utility
  - sensing/communication functions are irrelevant.
- Use Reinforcement Learning (RL) to arrive at an optimal control policy.

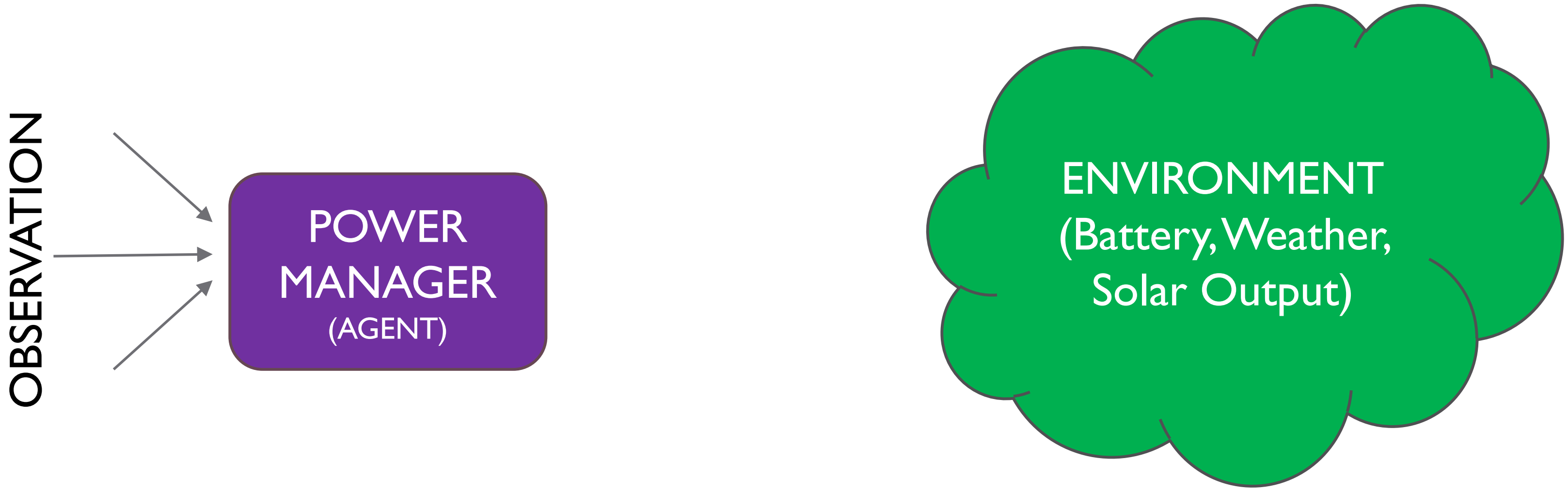


REINFORCEMENT LEARNING

- Type of machine learning based on experience rather than instructions
  - Evaluative feedback instead of Instructive feedback
- Agent interacts with environment to receive rewards. GOAL: Maximize the total (discounted) CUMULATIVE reward.

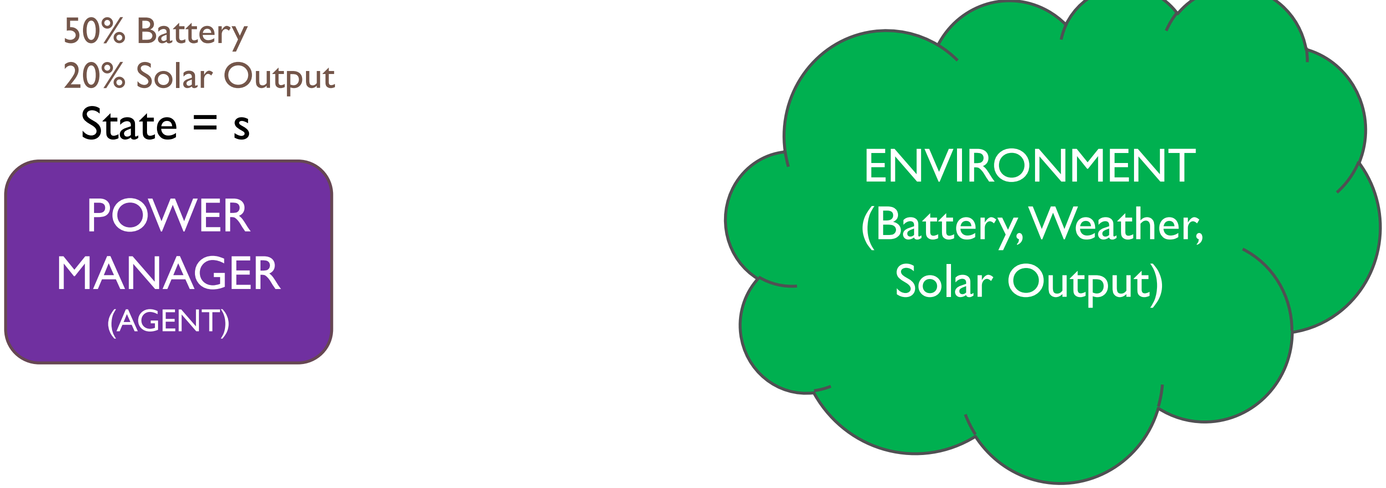
REINFORCEMENT LEARNING

- Type of machine learning based on experience rather than instructions
  - Evaluative feedback instead of Instructive feedback
- Agent interacts with environment to receive rewards. GOAL: Maximize the total (discounted) CUMULATIVE reward.



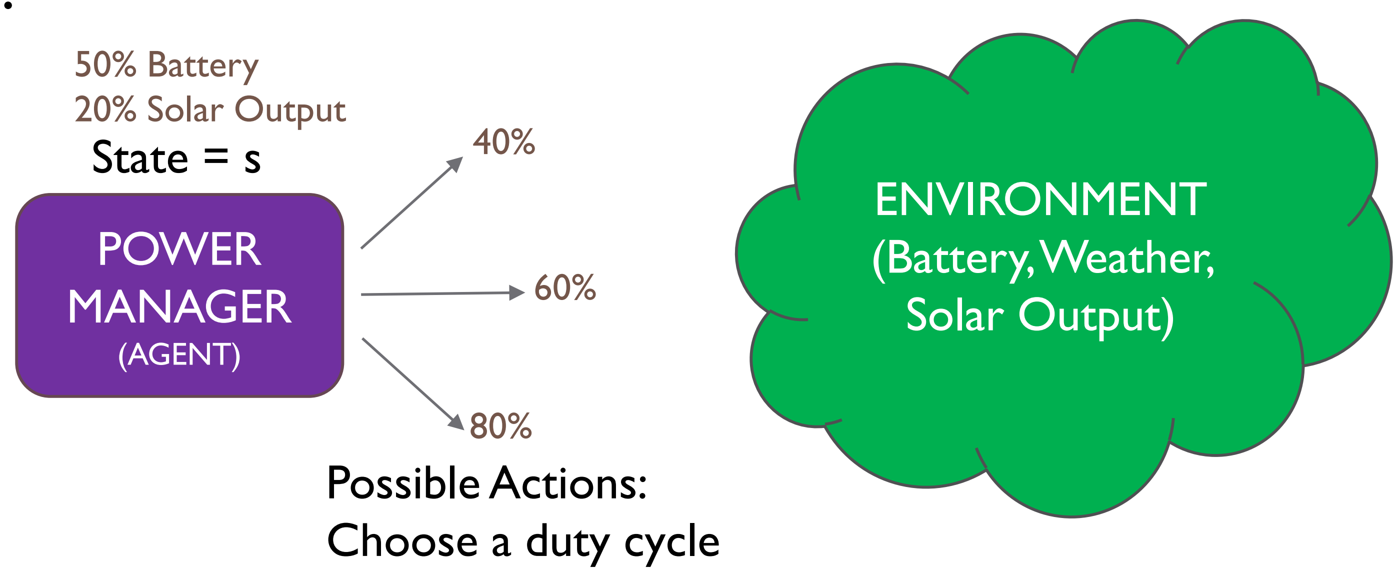
REINFORCEMENT LEARNING

- Type of machine learning based on experience rather than instructions
  - Evaluative feedback instead of Instructive feedback
- Agent interacts with environment to receive rewards. GOAL: Maximize the total (discounted) CUMULATIVE reward.



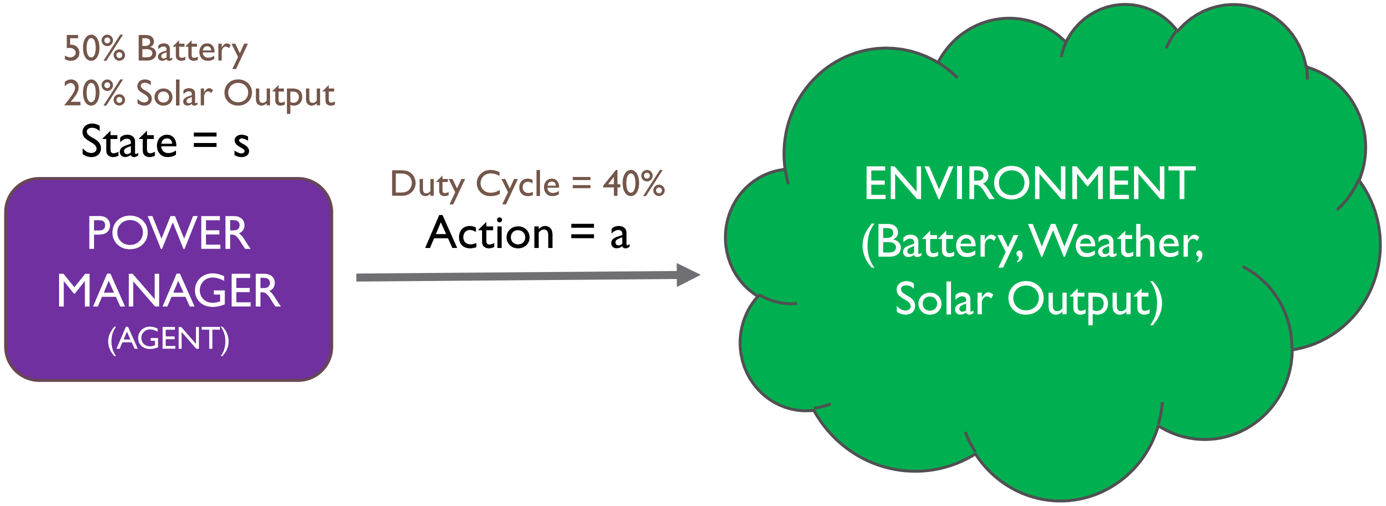
REINFORCEMENT LEARNING

- Type of machine learning based on experience rather than instructions
  - Evaluative feedback instead of Instructive feedback
- Agent interacts with environment to receive rewards. GOAL: Maximize the total (discounted) CUMULATIVE reward.



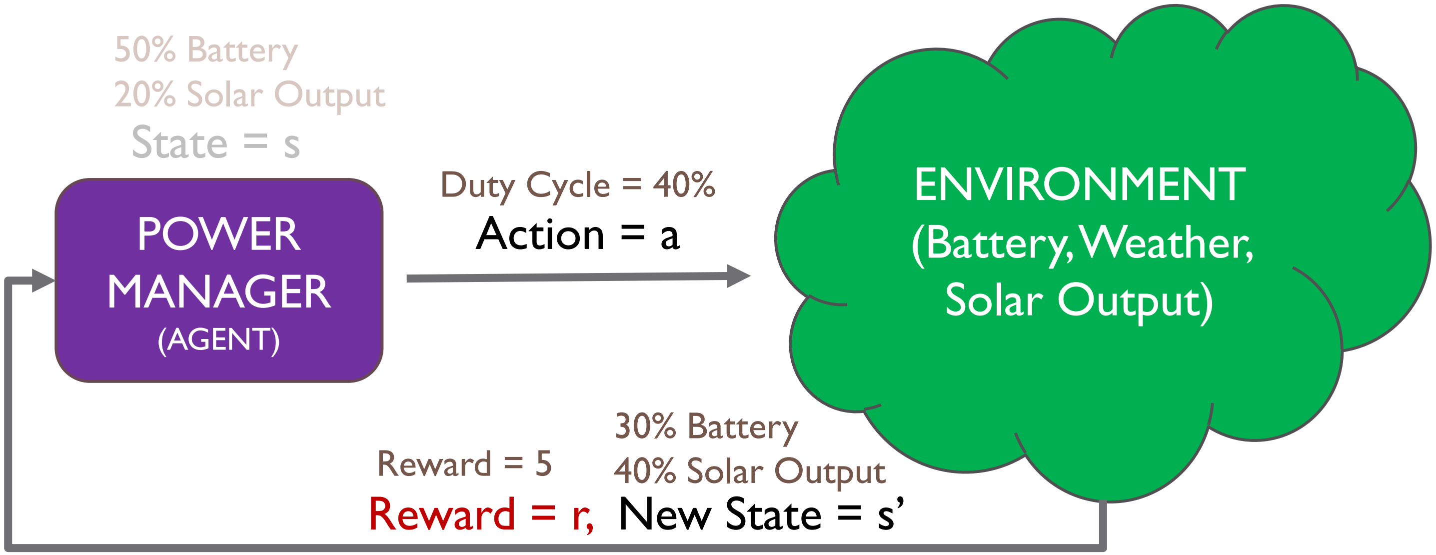
REINFORCEMENT LEARNING

- Type of machine learning based on experience rather than instructions
  - Evaluative feedback instead of Instructive feedback
- Agent interacts with environment to receive rewards. GOAL: Maximize the total (discounted) CUMULATIVE reward.



REINFORCEMENT LEARNING

- Type of machine learning based on experience rather than instructions
  - Evaluative feedback instead of Instructive feedback
- Agent interacts with environment to receive rewards. GOAL: Maximize the total (discounted) CUMULATIVE reward.



STATE DEFINITION

$$\text{State at epoch } t_k = \left(S_{dist}(t_k), S_{batt}(t_k), S_{eharvest}(t_k), S_{day}(t_k) \right)$$

Distance from energy neutrality, $S_{dist}(t_k)$	Battery, $S_{batt}(t_k)$	Harvested Energy, $S_{eharvest}(t_k)$	Weather Forecast, $S_{day}(t_k)$
- 20000 mWh	Low (< 20%)	0 mWh	Very little sun
- 19000 mWh	Mid (20% to 80%)	0 - 100 mWh	Overcast
⋮	High (> 80%)	100 mWh - 500 mWh	Partly Cloudy
0 mWh		500 mWh - 1000 mWh	Fair
⋮		1000 mWh - 1500 mWh	Sunny
19000 mWh		1500 mWh - 2000 mWh	Very Sunny
20000 mWh		> 2000 mWh	

ACTION SPACE

Choose duty cycle of the sensor node

$$A = a(t_k) \in \{1,2,3,4,5\}$$

ACTION $a(t_k)$	DUTY CYCLE (%)	ENERGY CONSUMED PER HOUR (mWh)
1	20	100
2	40	200
3	60	300
4	80	400
5	100	500

REINFORCEMENT LEARNING

- Battery Level (3)
- Weather Forecast (6)
- Harvested Energy (7)
- Energy Neutral Performance (ENP) (41)
  - Current battery – Optimal battery level

Calculated using statistical data about the energy harvesting environment

State = s

Discrete Duty Cycles (20%, 40%, 60%, 80%, 100%)

POWER MANAGER
(AGENT)

Action = a

An action is executed every hour

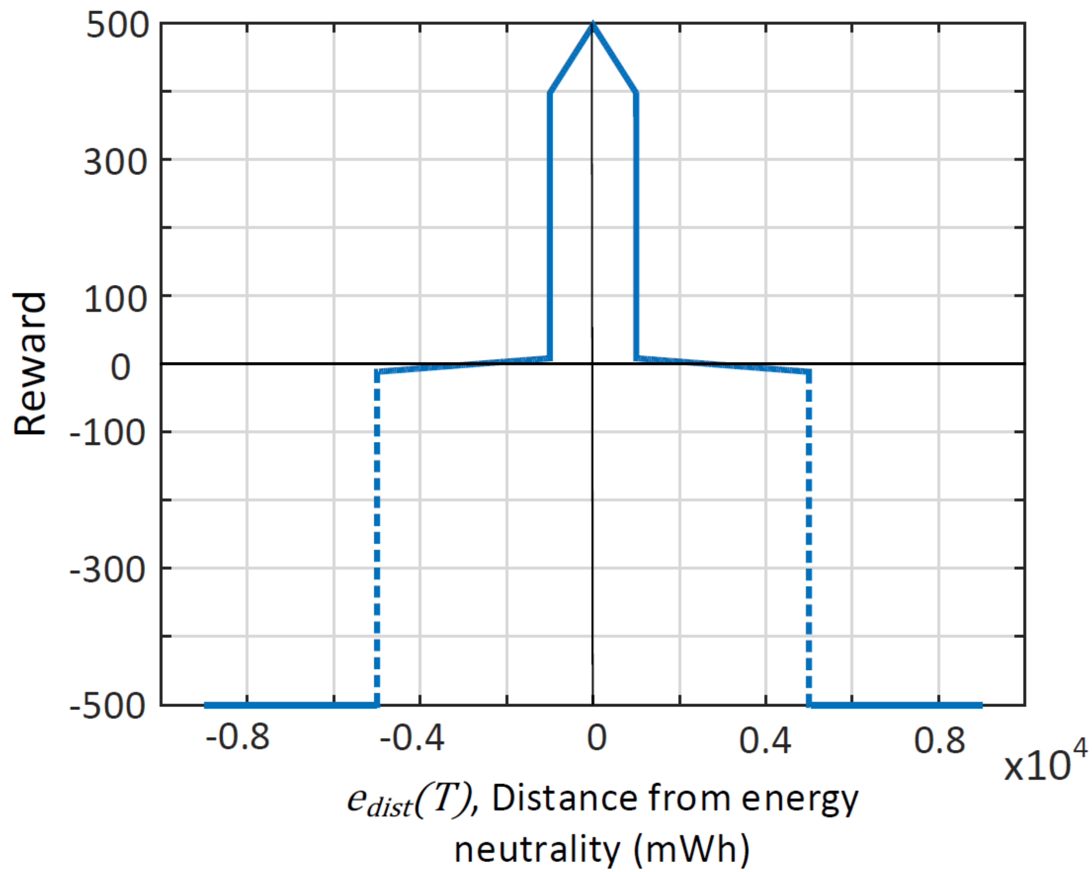
ENVIRONMENT
(Battery, Weather, Solar Output)

1 hour = 1 EPOCH
24 epochs = 1 EPISODE

- SINGLE scalar value
- Rewarded at the end of a day (episode)
- Based on deviation of battery from optimal value

Reward = r

Reward Function



- Awarded at the end of an episode (day).
- Ideally, difference between initial and final battery levels = 0
- Reward scheme depends on **Terminal Energy Neutral Performance (TENP)** i.e. ENP at the end of the episode.
  - Terminal Energy Neutral Performance is defined here as

$$|Initial\ battery\ level - Final(current)battery\ level|$$

THE LEARNING PROCESS

- Simulate using historical weather data for Tokyo, 2010.
- Agent tries various strategies, learns which policies are best and remembers them.
- Learning Algorithm: SARSA(λ) Learning
- Compare with Offline Policy for 2011
  - Offline Policy is calculated using assuming an omniscient solar energy predictor and Linear Programming methods.
  - Gives the optimal policy.
  - This is not a realistic solution as it requires perfect information about the future.
    - Only for comparison purposes

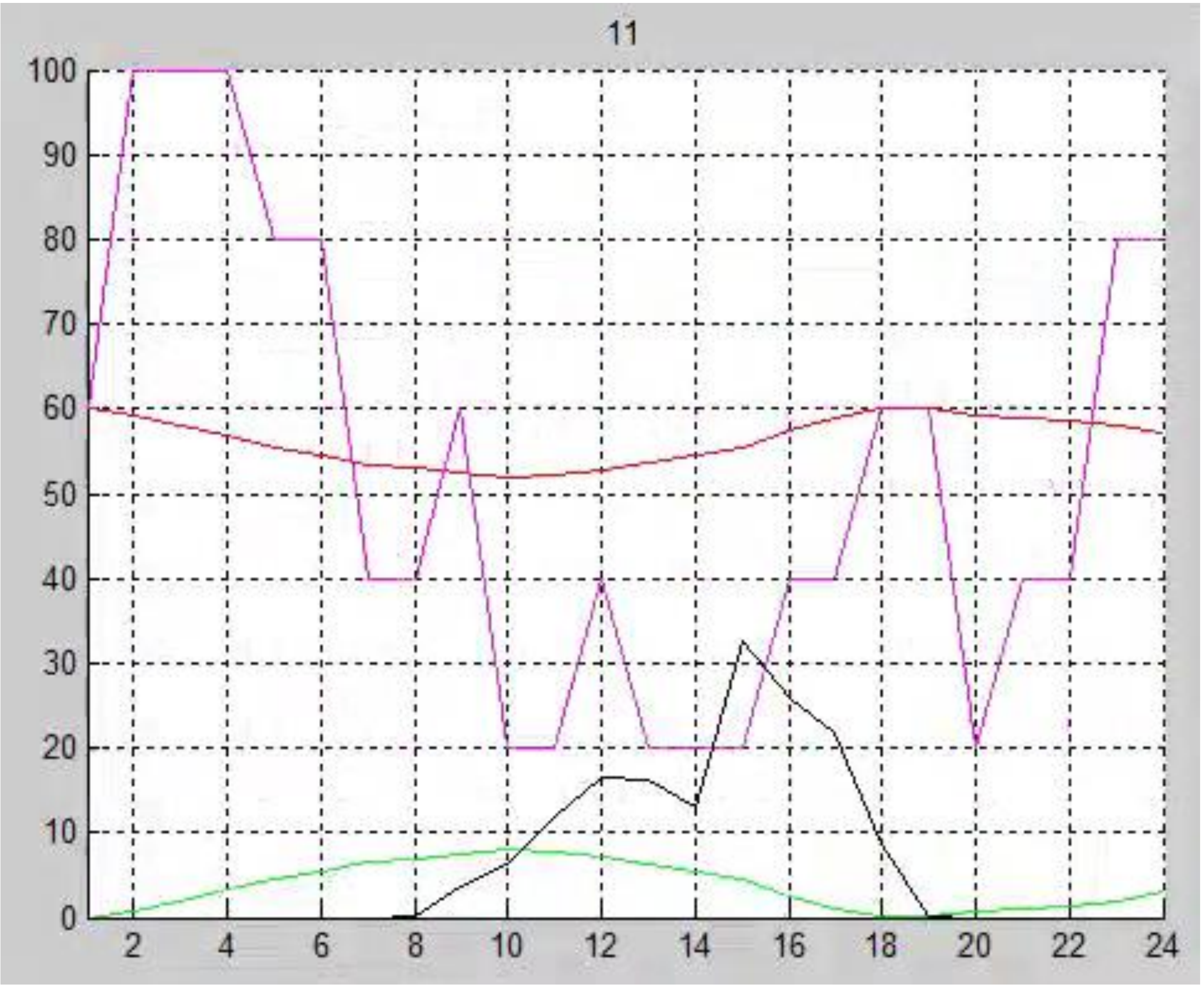
LEARNING

$$\begin{aligned}
\alpha \text{ (learning rate)} &= 0.1 \\
\epsilon \text{ (exploration ratio)} &= 0.1 \\
\gamma \text{ (discount factor)} &= 0.8 \\
\lambda \text{ (trace}-\text{decay parameter)} &= 0.8 \\
N \text{ (number of iterations)} &= 10^4
\end{aligned}$$

TRAINING DAYS

DAY	Total Energy Received (mWh)	Best Duty Cycle
265	13296.25	110.80%
80	11990.00	99.92%
101	10800.63	90.00%
37	9625.00	80.21%
69	8415.00	70.13%
343	7218.75	60.16%
329	6050.00	50.42%
53	4716.25	39.30%
277	3575.00	29.79%
61	2433.75	20.28%
102	1244.38	10.37%
303	515.625	4.30%

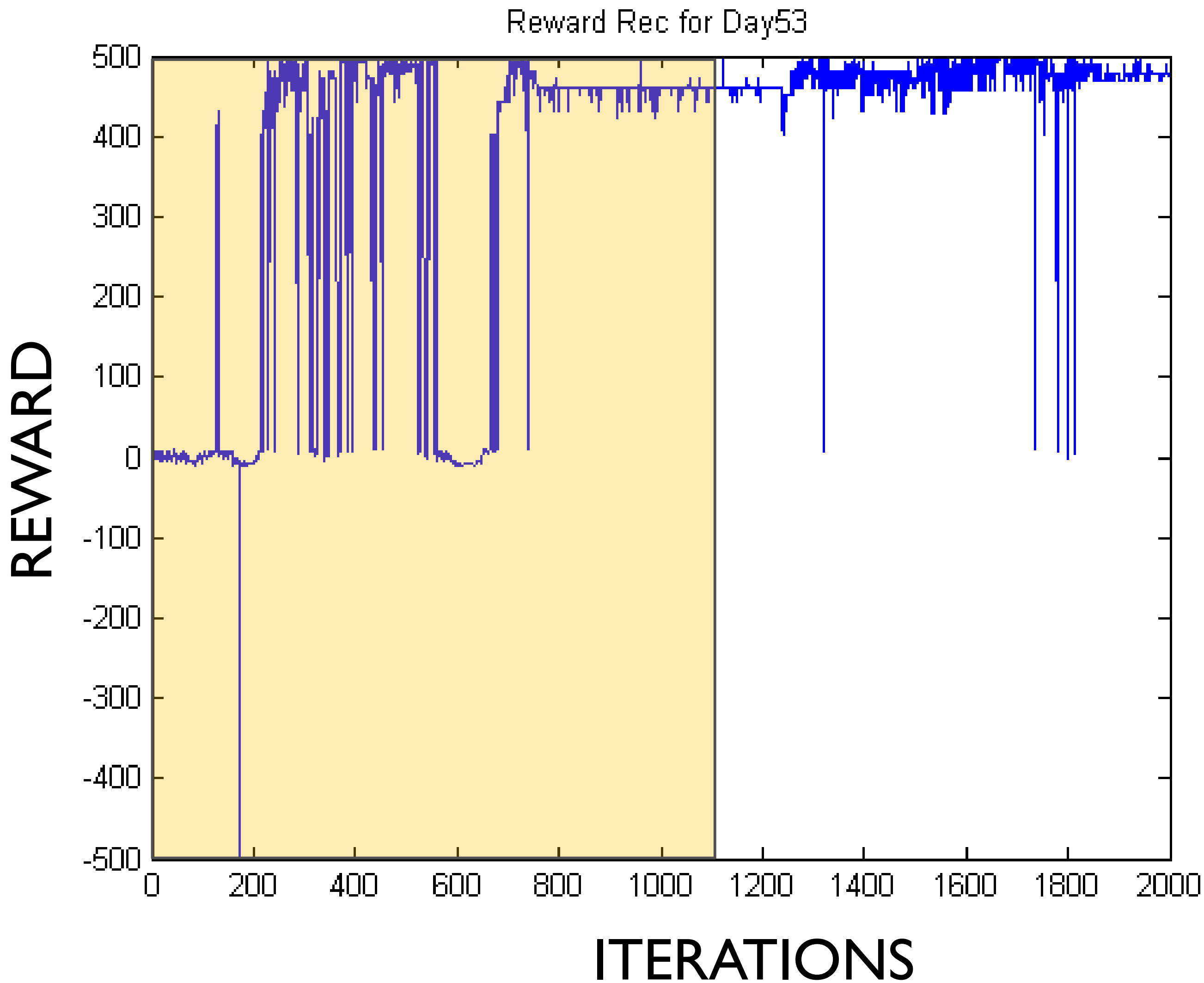
LEARNING



SARSA(λ) Learning

DAY 53, 2010 Tokyo

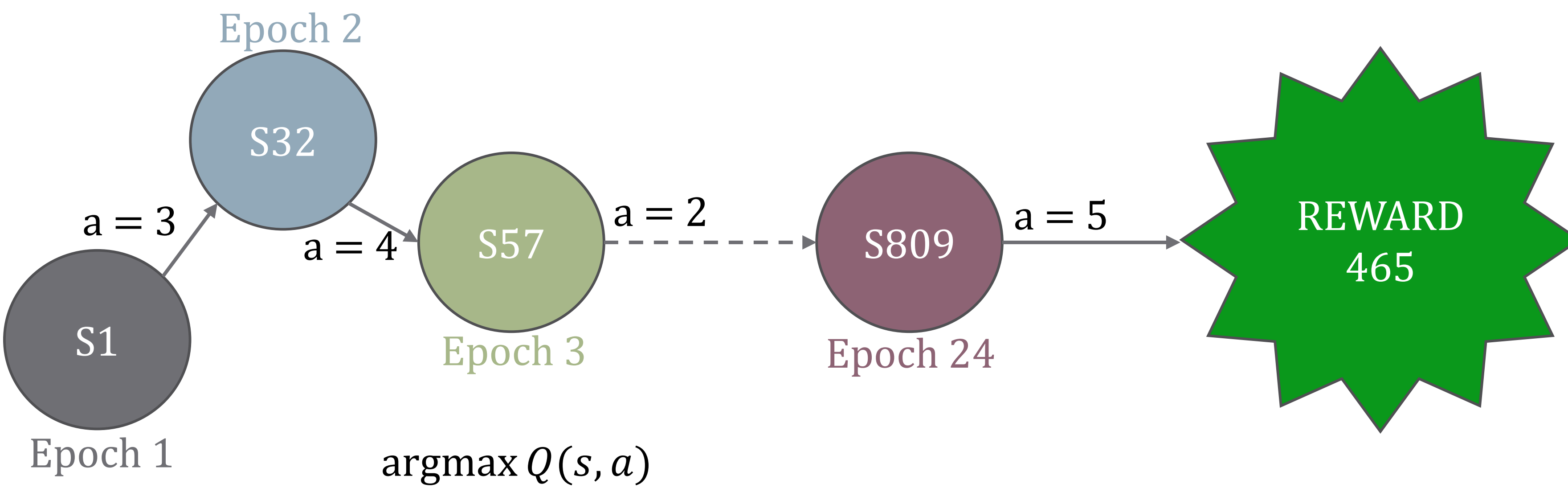
LEARNING



SARSA(λ) Learning

DAY 53, 2010 Tokyo

Q-Values



$\underset{a}{\operatorname{argmax}}\, Q(s,a)$

$Q(s,a)$	-2.5	-1.7	0.5	-5.8	0.1	...	30.7	...	121.3	...	483.7	…
(s,a)	(1,1)	(1,2)	(1,3)	(1,4)	(1,5)	...	(32,4)	...	(57,2)	...	(809,5)	…

Each state-action pair (s,a) is associated with a Q-value $Q(s,a)$ for a particular policy π.

$Q(s,a)$ is the expected cumulative reward if you take action a at state s and follow π.

SARSA(λ)

- Each state action pair is initialized to an eligibility value (trace), $e(s, a) = 0$
  - Every time (s, a) is visited, $e(s, a) = e(s, a) + 1$
  - Otherwise, $e(s, a)$ decays by a factor of $\gamma\lambda$.
  - The value of $e(s, a)$ determines how *influential* that state-action pair was in obtaining the reward at the end of an episode.

- Agent starts at state s_k and takes some action a_k according to policy π.
- It receives a reward r_k and is transported to a new state s_{k+1}.
- The agent *considers* taking the next action a_{k+1}.
- The Q-value $Q^{\pi}(s_k, ak)$ is then updated as:

$$Q^{\pi}(s_k, a_k) \leftarrow Q^{\pi}(s_k, a_k) + \alpha e(s, a)[r_k + \gamma Q^{\pi}(s_{k+1}, a_{k+1}) \ - Q^{\pi}(s_k, a_k)]$$

- ε-greedy policy is used i.e. random actions are taken with probability ε to allow exploration. Otherwise greedy actions are executed.

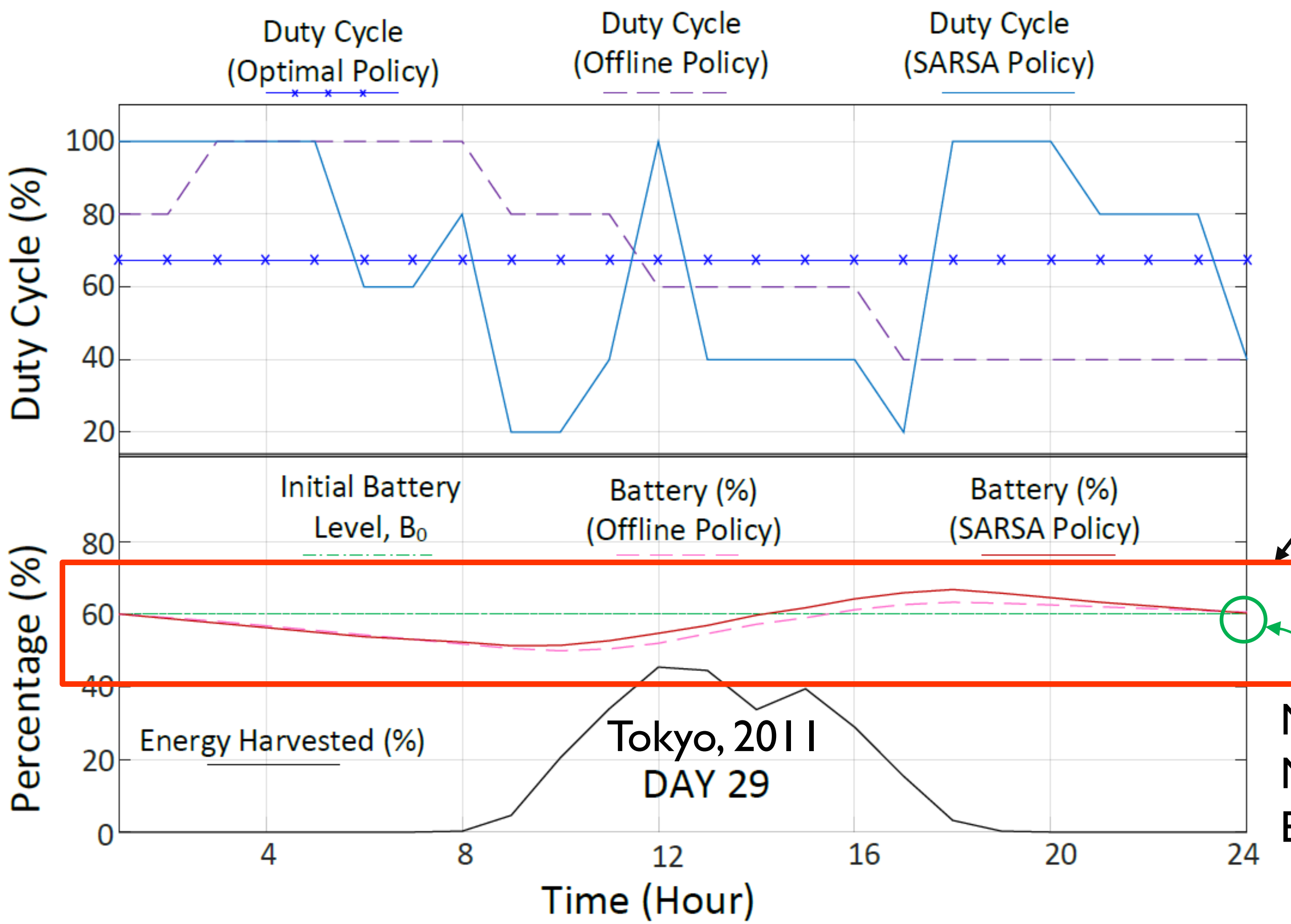
IMPLEMENTATION

- Wakkanai
- Much colder climate
- Average Annual Temp = 6.2°C
- Observe behavior at a location that has never been experienced

- Tokyo
- Training grounds
- Average Annual Temp = 15.6°C

RESULTS

- Comparison with omniscient Offline Policy
- Near Perfect Energy Neutral Performance

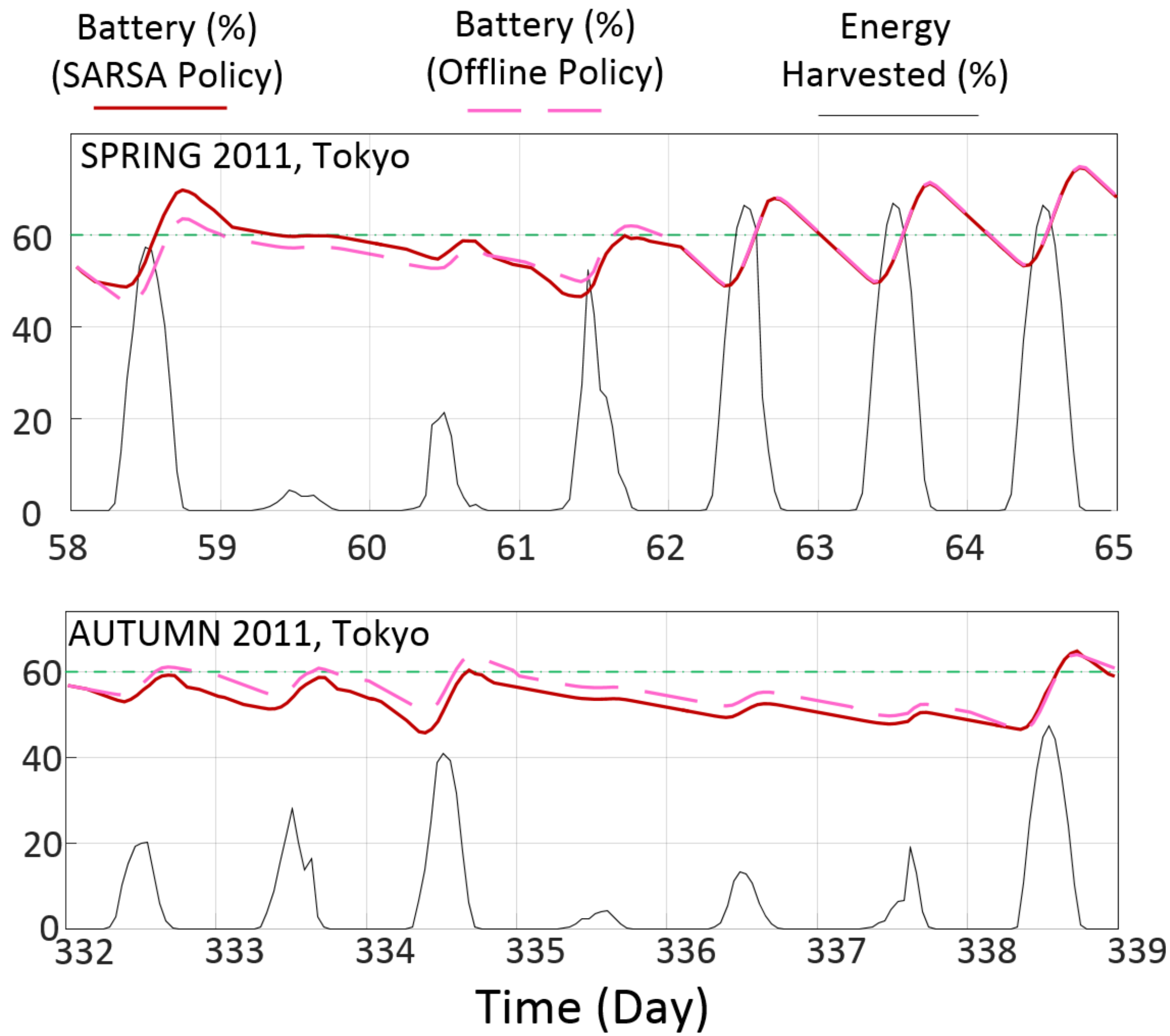


Battery profiles for SARSA and Optimal Policy are very similar

Near Perfect Node Level Energy Neutrality

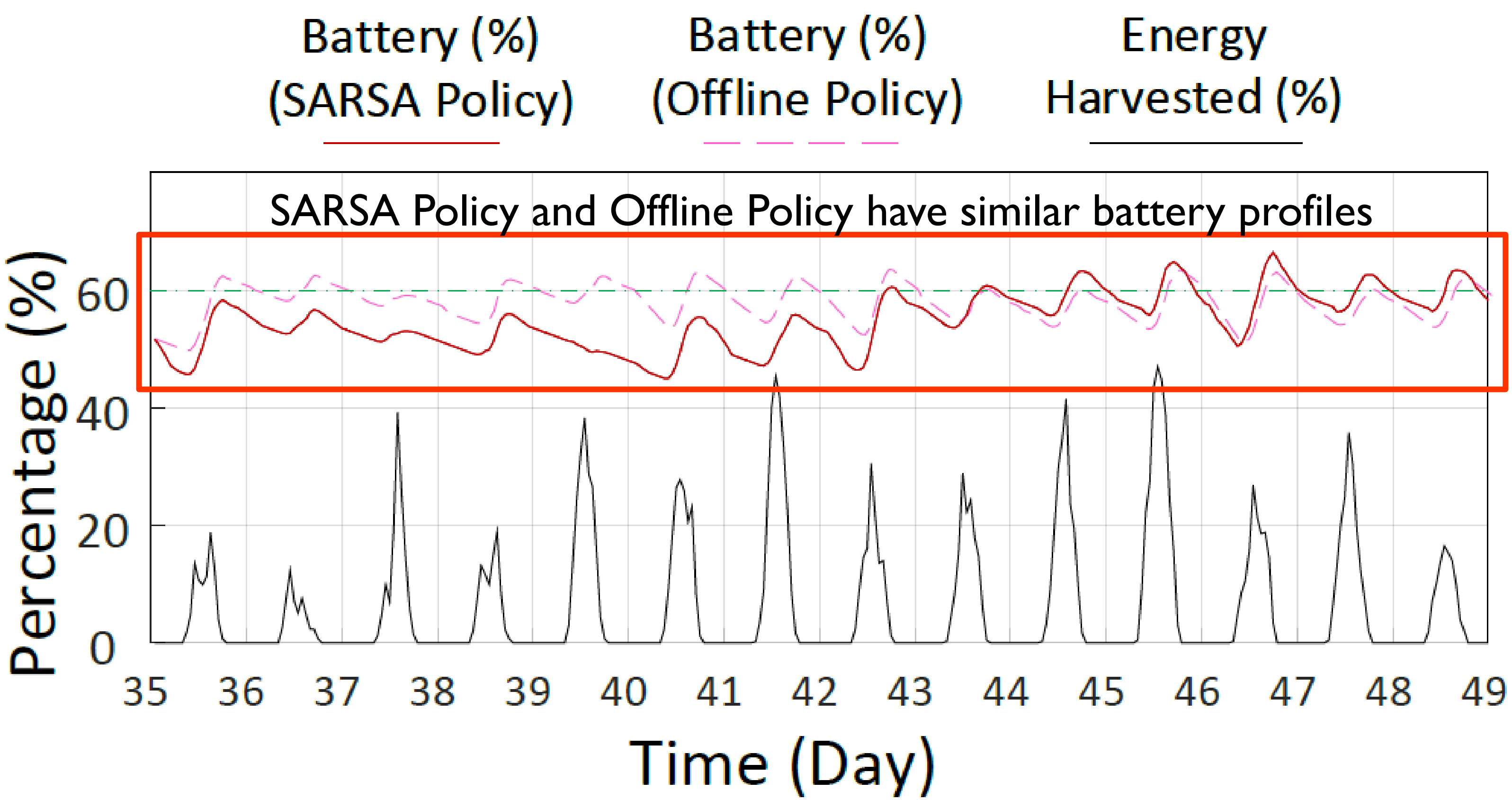
RESULTS

- Trained in Tokyo, 2010
- Implemented in Tokyo, 2011
- Adaptation to change in weather



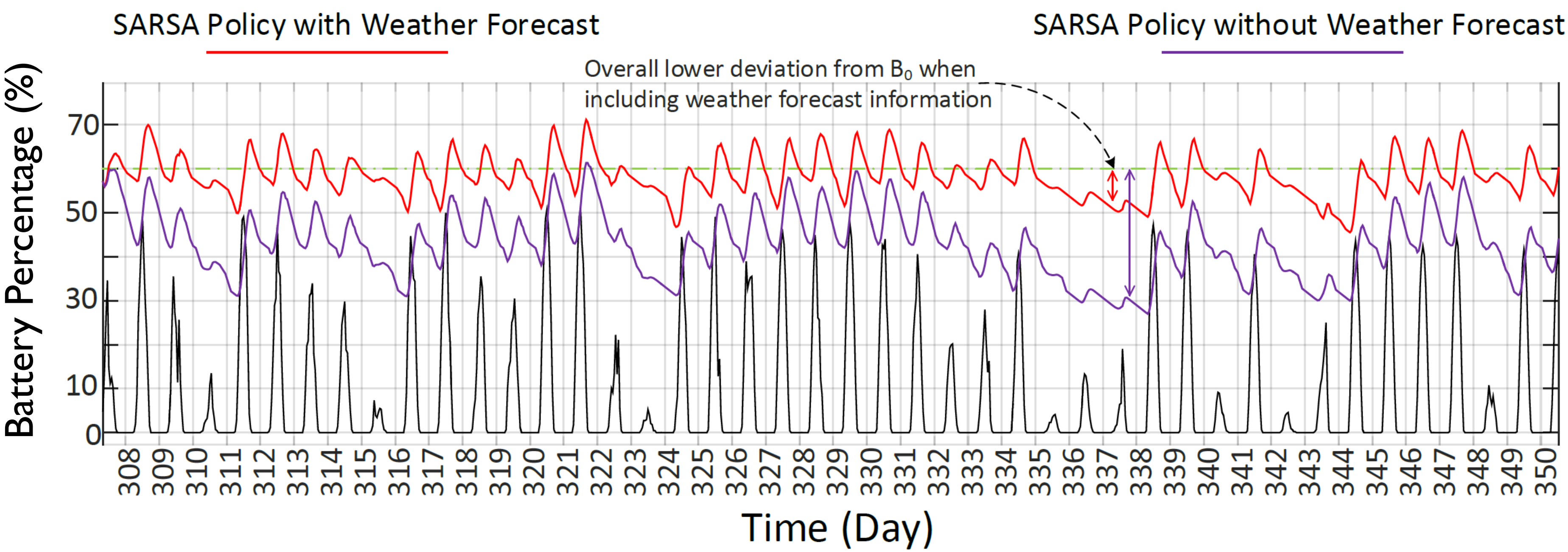
RESULTS

- Trained with Tokyo 2010 weather
- Implemented in Wakkanai, 2011



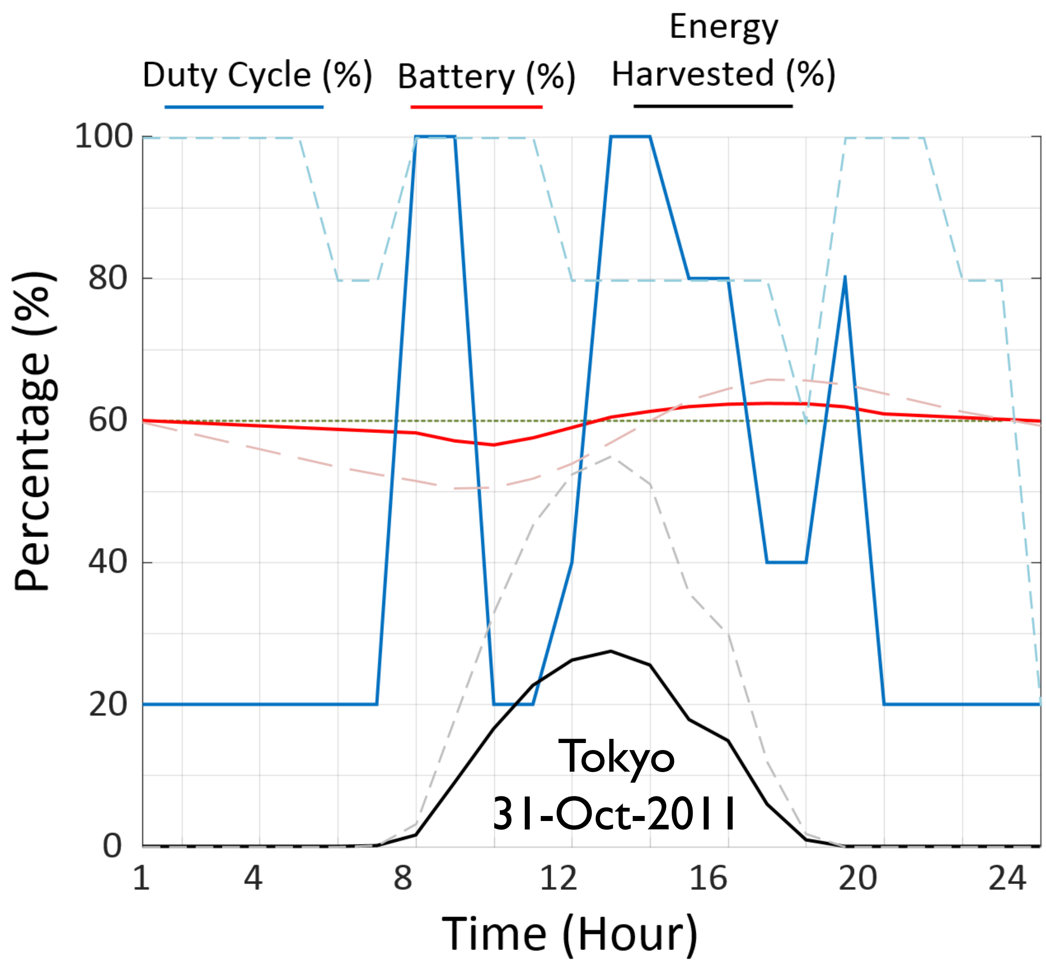
RESULTS

- Trained in Tokyo, 2010
- Implemented in Wakkanai, 2011
- Weather Forecast enhances perfomance



RESULTS

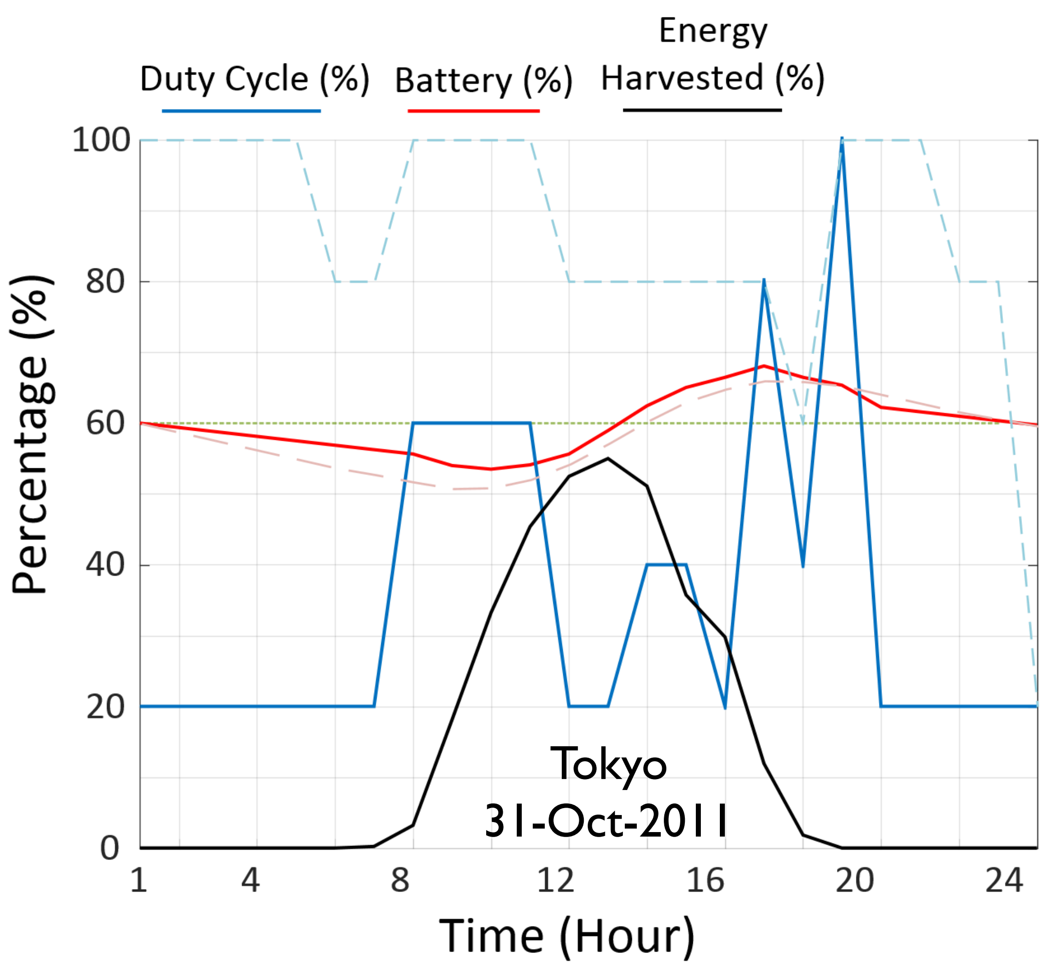
- Half Solar Panel Capacity
- After training for 1000 iterations with α = 0.1 and ϵ = 0.7



Watermarked, dashed lines are corresponding values for full solar panel capacity

RESULTS

- Node Power Consumption increases by 2.5 times
- After training for 1000 iterations with $\alpha = 0.1$ and $\epsilon = 0.7$



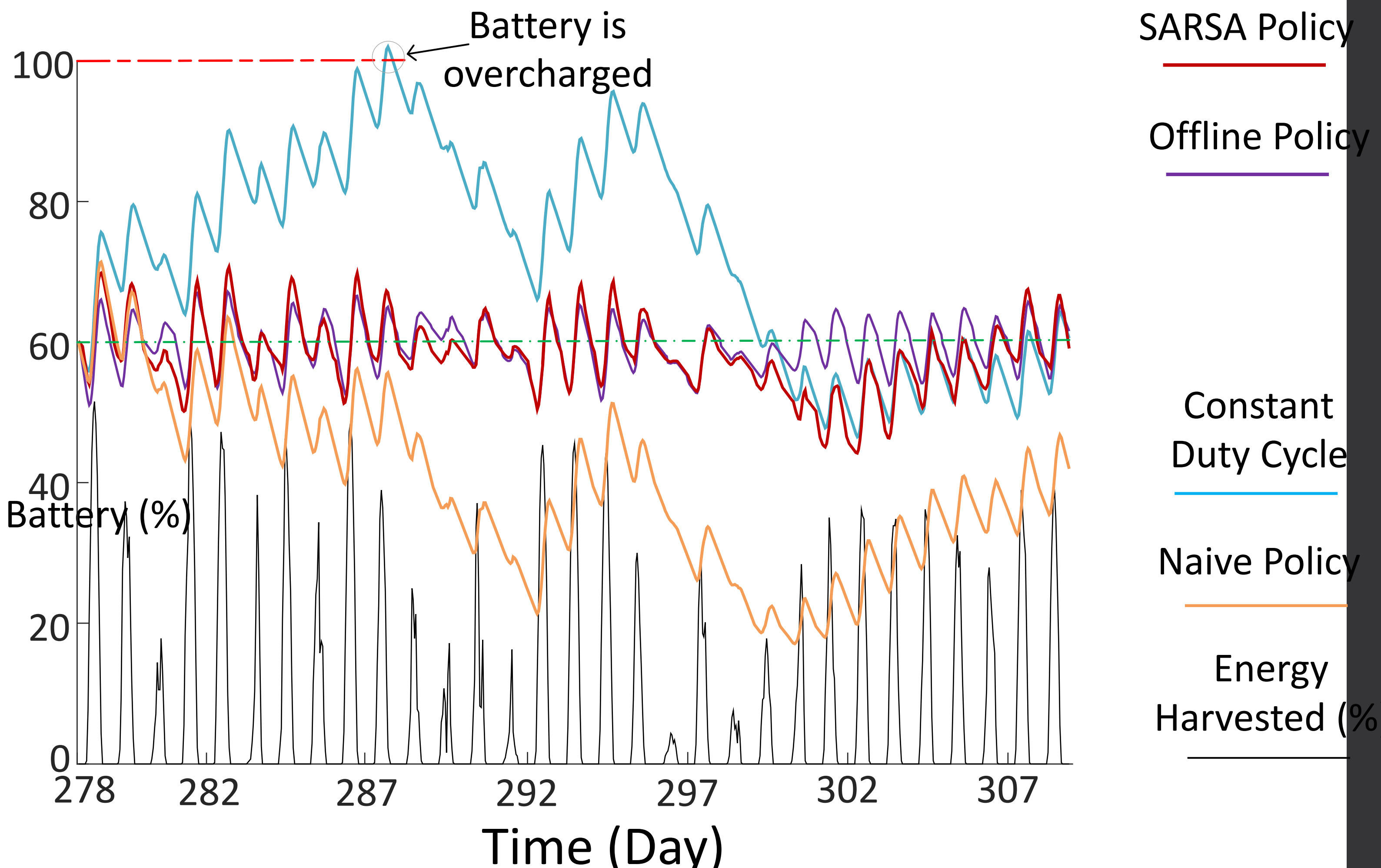
Watermarked, dashed lines are corresponding values for full solar panel capacity

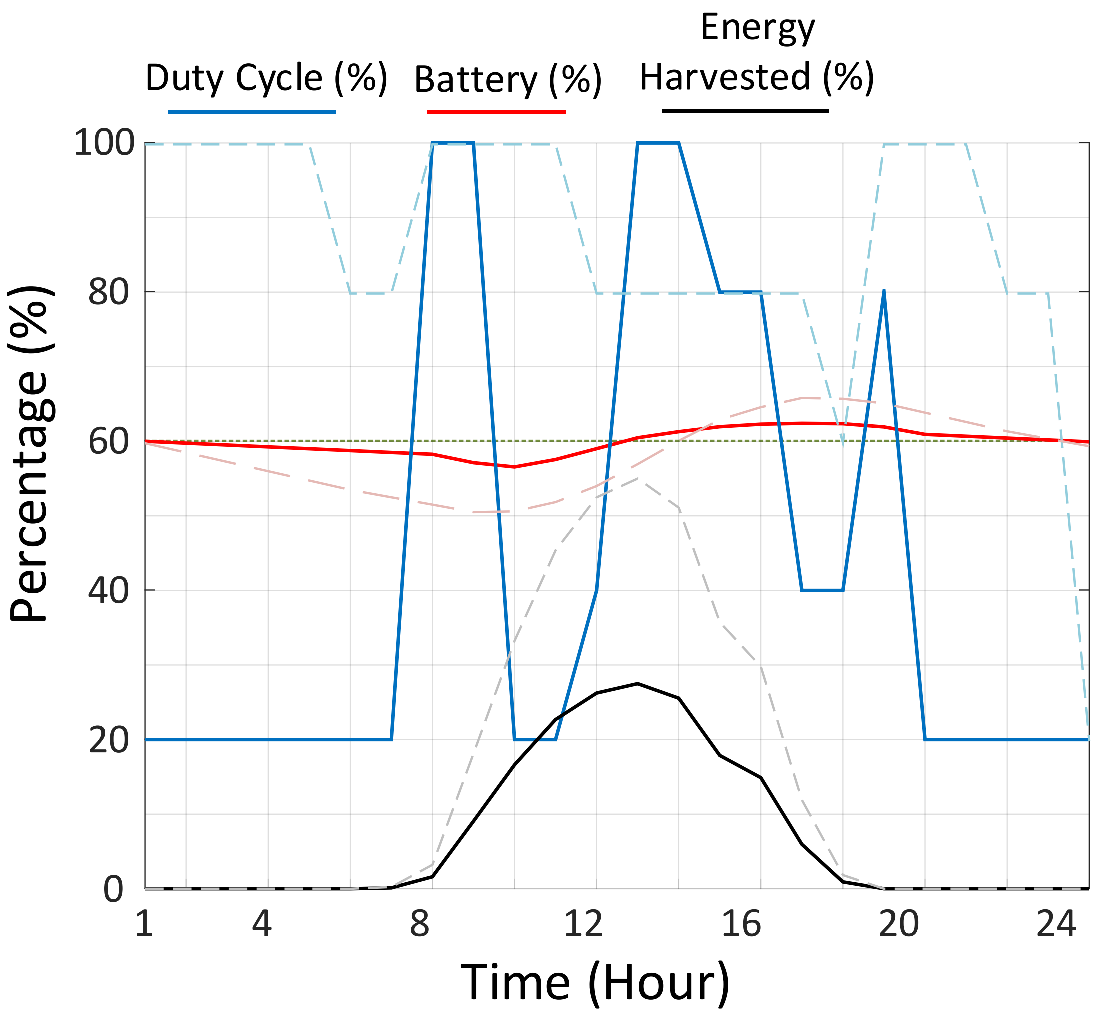
CONCLUSION

- Reinforcement Learning using SARSA(λ) is capable of attaining near-perfect node level energy neutrality.
- SARSA(λ) is able to learn from its working environment and adapt accordingly to achieve near-perfect node level energy neutrality.
- Inclusion of weather forecast information helps in achieving node level energy neutrality

THANK YOU FOR LISTENING

shaswot@hal.ipc.i.u-tokyo.ac.jp


Battery is overcharged
100
80
60
40
20
0
Battery (%)
278
282
287
292
297
302
307
Time (Day)
SARSA Policy
Offline Policy
Constant Duty Cycle
Naive Policy
Energy Harvested (%)


Duty Cycle (%)
Battery (%)
Energy Harvested (%)
100
80
60
40
20
0
1
4
8
12
16
20
24
Percentage (%)
Time (Hour)