Uncertainty Modeling in Self-Healing Software Systems

Kim-Pascal Borchart

Machine Learning-based Control of Dynamical Systems

Summer of 2021, Hasso Plattner Institute

Overview and Context

- Two Projects
  - Reinforcement Learning for Self-Repair (Past Seminar)
    - Uncertainty model, where fixes can fail depending on environment
  - Machine Learning Control (This Seminar)
    - Predictive model + python controller to use with mRubis
- Plan
  - Adapt past approach to current architecture
  - Add it to current project
  - Evaluate performance

Architecture Differences

	RL 4 Self-Repair	ML Control
Prediction Goal	Fix order	Fix order
Fix failure	Possible	Never
Fix order based on	$\max(E[P_{success}] * E_{\pi}[R])$	$\max(E_{\pi}[R])$
Issue loading	Batch	„Stream“
Actual Reward	Zero or inferred from table	mRubis / own prediction

Architecture Differences

	RL 4 Self-Repair	ML Control
Prediction Goal	Fix order	Fix order
Fix failure	Possible	
Fix order based on	$\max(E[P_{success}] * E_{\pi}[R])$	
Issue loading		„Stream“
Actual Reward		mRubis / own prediction

Past-Learned Action Failing Probabilities – Transition Matrix

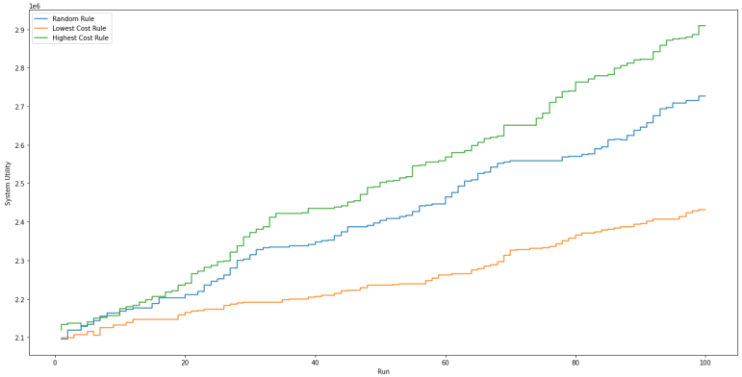
	Sources	Authentication Service	Availability Item Fi...	Bid and Buy Service	Buy Now Item Filter	Category Item Filter	Comment Item Filter	Future Sales Item Filter
1	Authentication Service	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	Availability Item Filter	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	Bid and Buy Service	0.25	0.00	0.00	0.00	0.00	0.00	0.00
4	Buy Now Item Filter	0.00	1.00	0.00	0.00	0.00	0.00	0.00
5	Category Item Filter	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	Comment Item Filter	0.00	0.00	0.00	0.00	1.00	0.00	0.00
7	Future Sales Item Filter	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	Inventory Service	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	Item Management Service	0.33	0.00	0.00	0.00	0.00	0.00	0.00
10	Last Second Sales Item Filter	0.00	0.00	0.00	0.00	0.00	0.00	0.00
11	Past Sales Item Filter	0.00	0.00	0.00	1.00	0.00	0.00	0.00
12	Persistence Service	0.00	0.00	0.00	0.00	0.00	0.00	0.00
13	Query Service	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14	Recommendation Item Filter	0.00	0.00	0.00	0.00	0.00	0.00	1.00
15	Region Item Filter	0.00	0.00	0.00	0.00	0.00	0.00	0.00
16	Reputation Service	0.33	0.00	0.00	0.00	0.00	0.00	0.00
17	Seller Reputation Item Filter	0.00	0.00	0.00	0.00	0.00	1.00	0.00
18	User Management Service	0.33	0.00	0.00	0.00	0.00	0.00	0.00

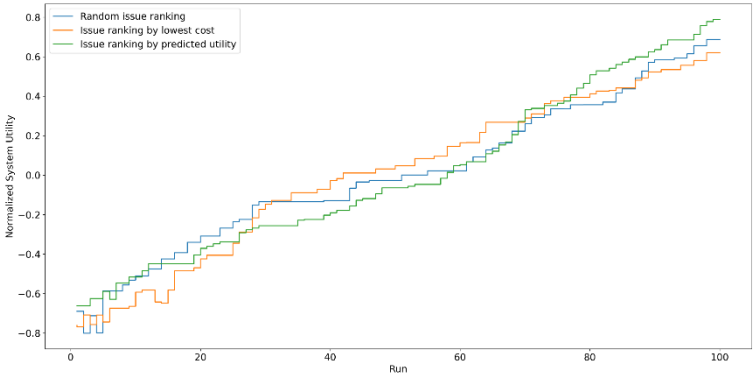
Past-Learned Action Failing Probabilities – Transition Matrix

	Sources	Authentication Service	Availability Item Fi...	Bid
1	Authentication Service	0.00	0.00	0.00
2	Availability Item Filter	0.00	0.00	0.00
3	Bid and Buy Service	0.25	0.00	0.00
4	Buy Now Item Filter	0.00	1.00	0.00
5	Category Item Filter	0.00	0.00	0.00
6	Comment Item Filter	0.00	0.00	0.00
7	Future Sales Item Filter	0.00	0.00	0.00
8	Inventory Service	0.00	0.00	0.00
9	Item Management Service	0.33	0.00	0.00
10	Last Second Sales Item Filter	0.00	0.00	0.00
11	Past Sales Item Filter	0.00	0.00	0.00
12	Persistence Service	0.00	0.00	0.00
13	Query Service	0.00	0.00	0.00
14	Recommendation Item Filter	0.00	0.00	0.00

Refresher: Machine Learning Control

- Best Rule-Picking Strategy: Highest cost (vs. lowest cost, random)
- Best Issue Ranking Strategy: Highest utility (vs. lowest rule cost, random)



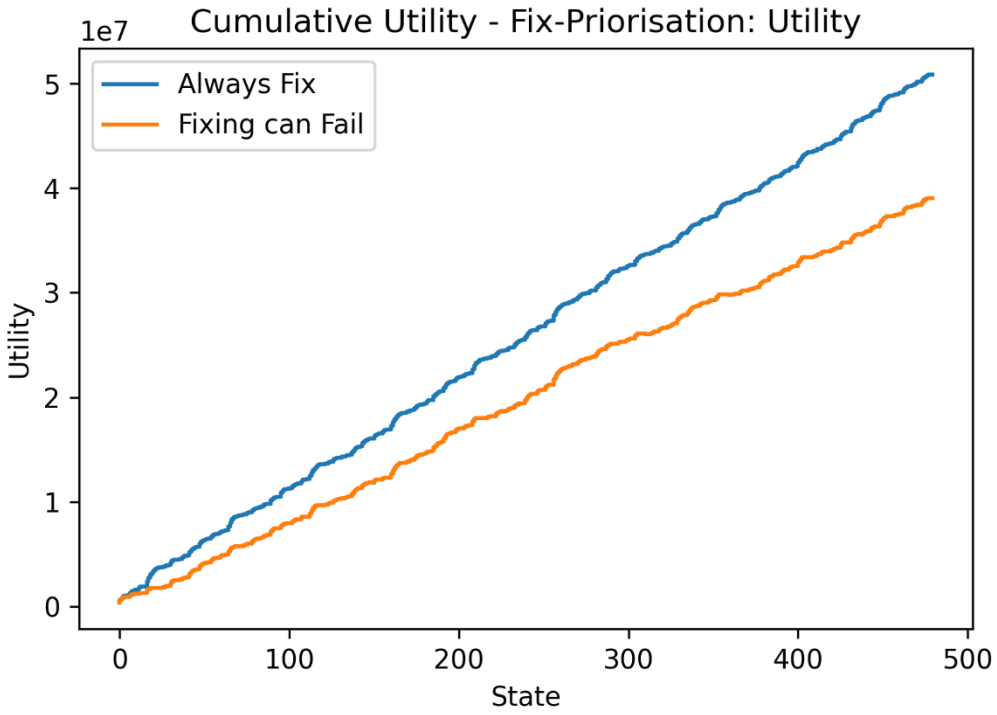


Research Questions

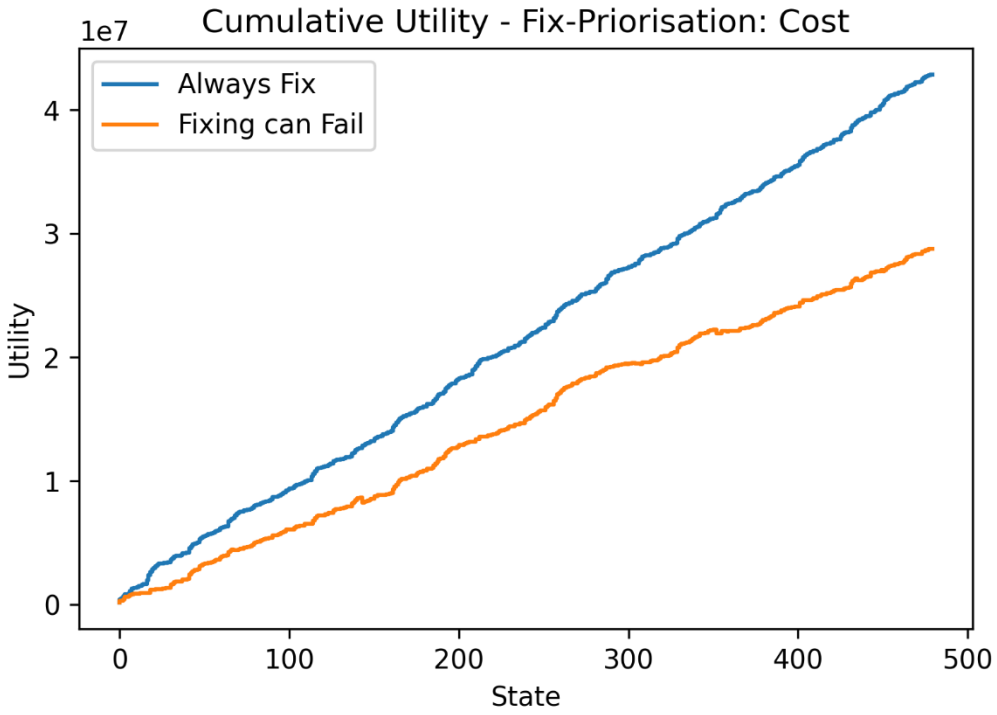
- RQ1
  - Are the two architecture approaches compatible?

- RQ2
  - Is the decision-making *significantly* worse when using uncertainty penalties?
- RQ3
  - How much utility do we lose with the uncertainty penalties?

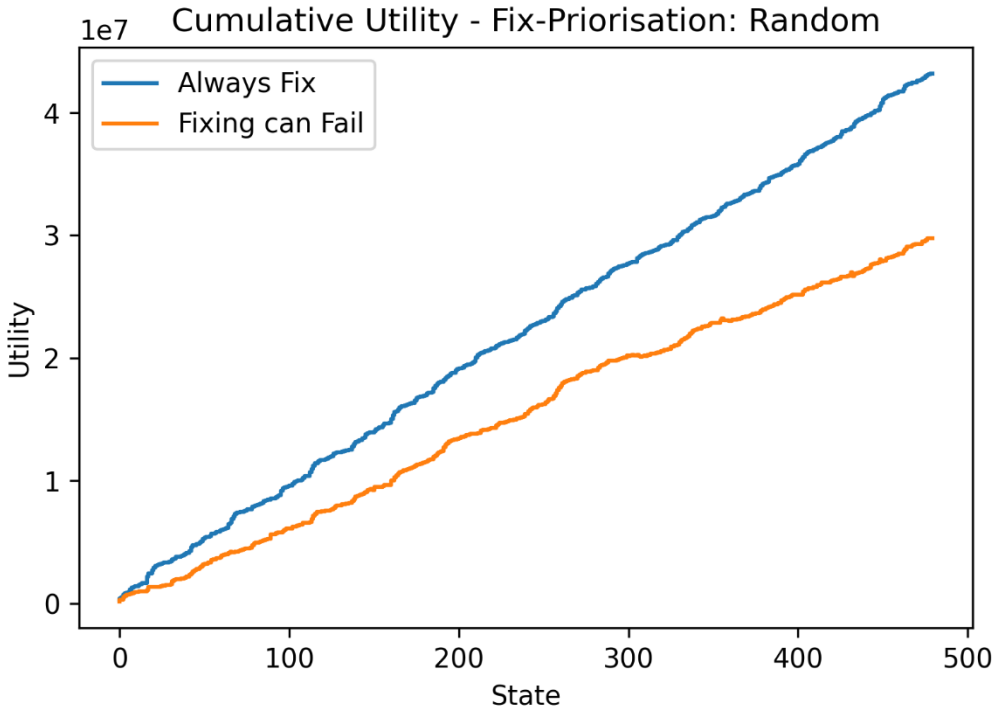
RQ 2 –
Significance of Uncertainty Penalties



RQ 2 –
Significance of Uncertainty Penalties



RQ 2 –
Significance of Uncertainty Penalties



RQ 2 –
Statistical Significance of Uncertainty Penalties

***H0**: The reward penalties have no impact on the actual rewards.*

=> t-test

```
```{r}
t.test(fixesCantFail$predicted_optimal_utility, fixesCanFail$predicted_optimal_utility)
```
```

```
        Welch Two Sample t-test

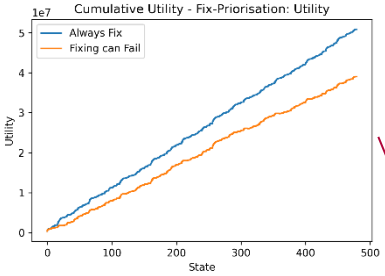
data:  fixesCantFail$predicted_optimal_utility and fixesCanFail$predicted_optimal_utility
t = 7.3209, df = 911.75, p-value = 5.408e-13
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 5240.121 9078.668
sample estimates:
mean of x mean of y
 26241.12  19081.73
```

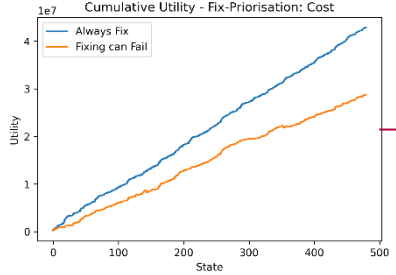
*p = 0.00, **H0** rejected.*

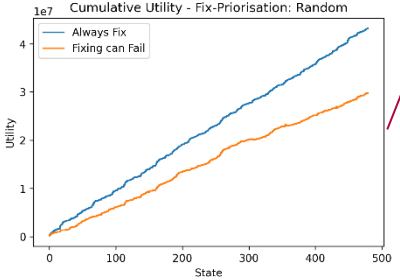
Research Questions

- RQ1
  - Are the two architecture approaches compatible?
- RQ2
  - Is the decision-making *significantly* worse when using incorrect reward penalties?
- RQ3
  - How much utility do we lose with the uncertainty penalties?

RQ3 – Uncertainty Model Utility Loss







	condition	normal	uncertain	rel.loss	abs.loss
1	utility	50825934	39037362	0.2319401	-11788573
2	cost	42859026	28787352	0.3283246	-14071674
3	random	43180494	29778061	0.3103816	-13402432

Research Questions

- RQ1
  - Are the two architecture approaches compatible?
- RQ2
  - Is the decision-making *significantly* worse when using incorrect reward penalties?
- RQ3
  - How much utility do we lose with the uncertainty penalties?
    - 23% - 32%

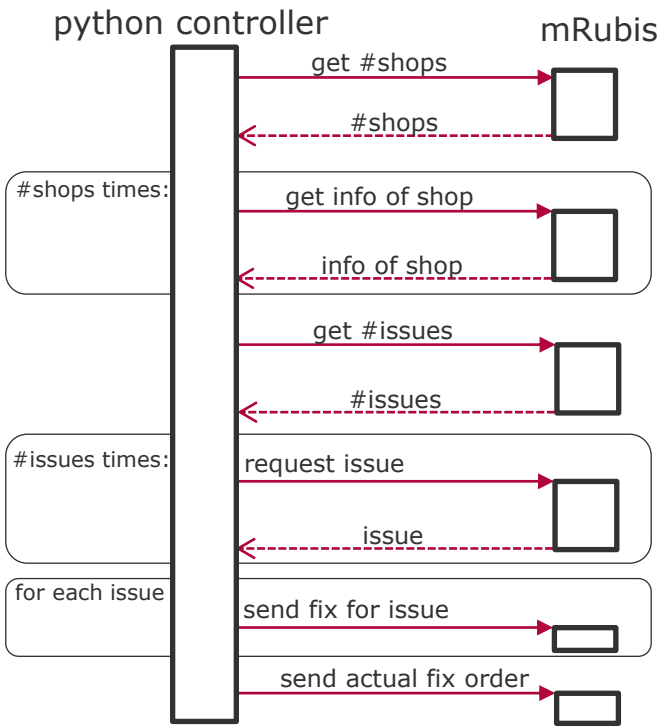
Discussion and Future Work

```
# Side effect: self.mrubis_state[shop][component]['predicted_optimal_utility']
self._predict_optimal_utility_of_fixed_components()
```

- Python
  - Currently a lot of functions with side effects
  - => Harder to understand data-transformations
  - Learn real fix-fail probabilities
  - Learn longer state horizon
- mRubis extensions
  - Needs „execution plan“ but ignores it ☺
  - Cumulative utility in mRubis
  - Fix-fail possibility
  - Publish-Subscribe architecture to communicate

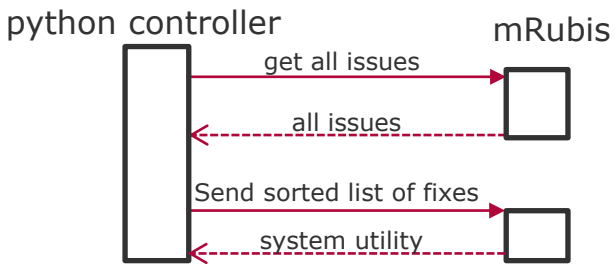
Disscussion - mRubis and Python Communication

Current



→ 20-40 messages

Ideal



2 messages

- ! Requires Message Broker that can handle long messages (RabbitMQ, ...)

Thank you ☺
Questions + Discussion?

Kim-Pascal Borchart

Machine Learning-based Control of Dynamical Systems

Summer of 2021, Hasso Plattner Institute