PAPER • OPEN ACCESS

Controlling optical-cavity locking using reinforcement learning

To cite this article: Edoardo Fazzari *et al* 2024 *Mach. Learn.: Sci. Technol.* **5** 035027

IOP Publishing Mach. Learn.: Sci. Technol. 5 (2024) 035027 https://doi.org/10.1088/2632-2153/ad638f

MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
25 February 2024

REVISED
23 June 2024

ACCEPTED FOR PUBLICATION
15 July 2024

PUBLISHED
24 July 2024

PAPER

Controlling optical-cavity locking using reinforcement learning

Edoardo Fazzari[1,*], **Hudson A Loughlin**[2] and **Chris Stoughton**[3]

[1] The BioRobotics Institute, Sant'Anna School of Advanced Studies, Viale Rinaldo Piaggio 56025 Pontedera, Italy
[2] LIGO Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139, United States of America
[3] Fermi National Accelerator Laboratory, Batavia, IL 60510, United States of America
[*] Author to whom any correspondence should be addressed.

E-mail: edoardo.fazzari@santannapisa.it

Keywords: reinforcement learning, machine learning, laser control, optics, q-learning

Abstract

This study applies an effective methodology based on Reinforcement Learning to a control system. Using the Pound–Drever–Hall locking scheme, we match the wavelength of a controlled laser to the length of a Fabry-Pérot cavity such that the cavity length is an exact integer multiple of the laser wavelength. Typically, long-term drift of the cavity length and laser wavelength exceeds the dynamic range of this control if only the laser's piezoelectric transducer is actuated, so the same error signal also controls the temperature of the laser crystal. In this work, we instead implement this feedback control grounded on Q-Learning. Our system learns in real-time, eschewing reliance on historical data, and exhibits adaptability to system variations post-training. This adaptive quality ensures continuous updates to the learning agent. This innovative approach maintains lock for eight days on average.

1. Introduction

Reinforcement Learning (RL) algorithms [1] have emerged as a pivotal force reshaping the scientific landscape by offering unprecedented capabilities to learn optimal actions leading to eventual success in uncharted environments, all without the need for external supervision. This transformative paradigm has found application across a spectrum of domains, ranging from self-driving cars [2, 3], industrial automation [4, 5], trading and finance [6], healthcare [7], gaming [8, 9], to the intricacies of optics [10–12].

Previous works utilizing RL in optics have primarily focused on the alignment of mirrors and waveplate angles. [13, 14]. In this work, we align optics manually and begin with well-aligned system components. The laser wavelength is matched to the Fabry–Pérot cavity length using a Pound–Drever–Hall (PDH) lock [15, 16]. At this resonance condition, power builds up inside the cavity. The wavelength of the laser light depends on the physical size of the laser's crystal. This is controlled in two ways: a high-bandwidth, low-dynamic-range control of the laser crystal' s shape with a piezo crystal [17, 18]; and a low-bandwidth, high-dynamic-range control of its temperature [19–21]. Fast (piezo) actuation of the laser wavelength has sufficient bandwidth to suppress most environmental noises and maintain lock between the laser and cavity. However, the fast actuator has a limited dynamic range, so slow (temperature) actuation is necessary for longer duration experiments [22].

The cavity operates in three modes throughout the RL process.

(i) Acquiring lock. This is done initially and repeated when the system loses the lock. This procedure is described in section 2.3
(ii) Training the Q-Learning agent (section 2.4). A Q-Learning agent is trained to operate the slow control.
(iii) Operating with the Q-Learning agent (section 3.2). The trained agent is tested to maintain lock using the best actions it learnt.

Our goal is to use Q-Learning for the Red Pitaya's analog output, which controls the laser crystal temperature. This large dynamic range control of laser wavelength allows stable lock maintenance.

In section 2, we provide a comprehensive overview of our experimental setup, detailing the optical system configuration, the Red Pitaya connection, and the RL paradigm employed. Additionally, we present a detailed account of the integration process within our system. In section 3, we present the outcomes of our experiments along with a thorough discussion of the findings. Finally, our conclusions are outlined in section 4.

2. Materials & methods

In this section, we detail the configuration of the optical setup and outline the implementation of our methodological modes. The initial step is to achieve the PDH lock [23], which necessitates setting the crystal's temperature such that a TEM_{00} mode [24] of the optical cavity lies within the wavelength range accessible by actuating the laser's piezo. To facilitate this, we devised a software-based approach spanning from an initial temperature to an endpoint, systematically checking for the presence of the desired mode. Upon successful detection, the PDH lock [23] is executed. After lock acquisition, we tested two ways to operate on the slow output: 1) keep the crystal' s temperature set at the point where the TEM_{00} mode appeared (we called this method *baseline*); 2) exploit Q-Learning [25] to change the temperature by small, predefined variations.

2.1. Optical setup

The optical setup was built at the *CryoModule Test Facility* (CMTF) at Fermilab and is shown in figure 1. We employed a 2 W Mephisto laser and used a highly reflective mirror and a beam stop to limit the power propagating towards the phase modulator and Faraday isolator to approximately 65 mW. After attenuating the power, a lens focuses the beam so it is substantially smaller than the phase modulator's 2 mm aperture. The phase modulator interfaces with a 25 MHz RF signal from the Red Pitaya and produces phase modulation sidebands on the laser light at this frequency. Another lens refocuses the laser beam and places a waist in the center of the Faraday. Between the lens and Faraday, a pair of $\lambda/4$ and $\lambda/2$ waveplates adjust the light's polarization to optimize the power transmitted through the Faraday isolator. Residual polarization mismatches result in 5 mW of light in the rejected polarization, which is dumped on a beam stop.

The remaining light reflects off of a pair of steering mirrors is strategically placed roughly 90 degrees apart in Gouy phase [26]. Three mode-matching lenses follow, aligning the incident laser beam's mode with the optical cavity's fundamental TEM_{00} mode. A camera and photodetector (PD) are used to monitor light transmitted by the cavity. These instruments facilitate alignment tuning and allow adjustments to the laser's temperature to ensure that the TEM_{00} mode falls within the frequency adjustment range of the laser's piezo-actuated, fast frequency control. The light reflected from the cavity goes back towards the mode-matching lenses and alignment mirrors and interacts with the circulator (Faraday isolator). The circulator reflects the returning beam, which is then focused through a lens and detected with a fast (125 MHz) photodetector, PDref. The photodetector's radio frequency (RF) output signal is sent to the Red Pitaya.

2.2. Red pitaya board

A Red Pitaya STEMlab 125-14 digitizer board was used to control the PDH lock acquisition, lock maintenance, and laser crystal temperature. The board has two RF inputs and RF outputs and four analog inputs and outputs, but we made use of only one analog output as described below (figure 2 summarizes the configuration):

(i) **RF Input 1** is the photodetector's output, which measures the beam reflected by the optical cavity. This signal contains the PDH error signal on top of a 25 MHz carrier frequency. To recover the error signal, the signal on *input 1* is demodulated at the 25 MHz carrier frequency.
(ii) **RF Input 2** is the optical cavity's transmitted power measured on the photodetector PDcav.
(iii) **RF Output 1** goes to the laser's fast frequency actuator for fast control and to the oscilloscope to monitor this signal. This output is generated by the demodulated PDH error signal from *input 1* fed to a Proportional-Integral-Derivative (PID) block in the Red Pitaya, which converts the error signal into a control signal. The implemented PID controller uses only an integrator, i.e. there are no proportional or differentiation parts. Since the actuator accepts signals between –100 V and +100 V[4] and the range of the Red Pitaya is between –1 V and +1 V, we send *Output 1* to a Piezo driver with a 100x amplification factor.

[4] For reference check page 5 in the Mephisto Laser documentation, downloadable here https://wiki.nikhef.nl/gravwav/images/6/6f/Innolight_Mephisto_Laser.pdf.

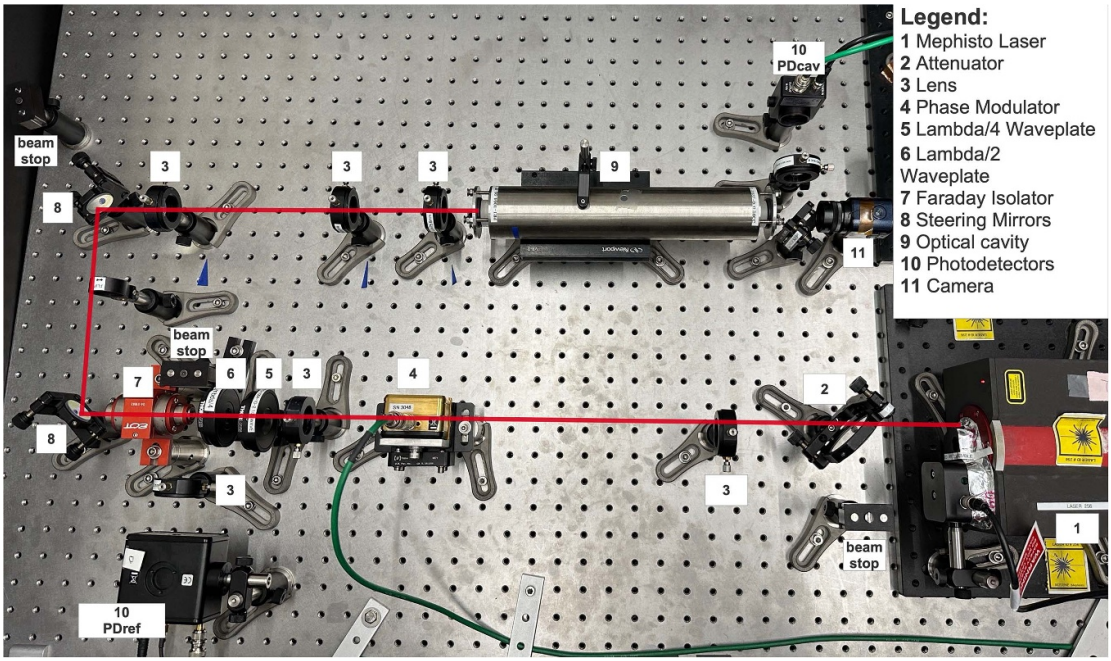

Figure 1. Optical system setup.

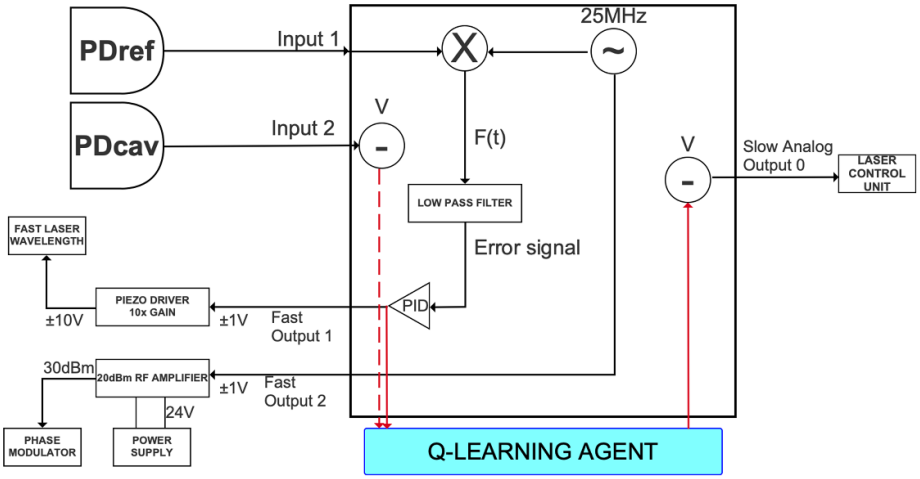

Figure 2. Red Pitaya's inputs and outputs configuration schema. The Q-Learning agent takes two inputs: the signal coming from the PID block, which is used as state, and the voltage from PDcav, used for checking lock and for rewarding the agent during training.

(iv) **RF Output 2** is a 25 MHz RF signal with a 2 V peak-to-peak amplitude. This output is amplified by a high-power RF amplifier, powered by an external +24 V power supply, before going to the phase modulator, which imprints phase modulation sidebands on the laser beam at this frequency for the PDH lock.

(v) **Analog Output 0** is used to control the temperature of the laser crystal, and it is connected to the rear panel of the Mephisto control electronic unit. For each increment of 0.01 V, the temperature increases 1° C.

The Red Pitaya was programmed using the Python language and the *Pyrpl* library[5].

[5] https://pyrpl.readthedocs.io/en/latest/.

Algorithm 1. Scanning Temperature.

```
procedure ScanTemperature(ϵ = 1000)
    for i in np.arange(0, 0.3, 0.00 025)do
        SetDAC2i
        _, input2 ← orderscope trace (self)
        half_trace ← int (input2.shape[0]/2)
        input2_peak_idx ← np.where (input2 == input2.max)
        input2_peak_idx ← input2_peak_idx[0][0]
        if half_trace − ϵ < input2_peak_idx < half_trace + ϵ and input2.max > 0.95
    then
            Return True
        end if
    end for
    Return False
end procedure
```

2.3. Software automatic lock

Our objective is to manipulate the laser crystal's temperature using the Red Pitaya, thereby modifying the laser's wavelength to find a TEM_{00} mode. To attain this goal, it is crucial to employ procedures capable of autonomously identifying the optimal mode by scanning through a range of temperatures. Simultaneously, a distinct procedure is necessary to establish and maintain the cavity lock, with subsequent checks to ensure its stability, supported by detailed analyses. These two critical tasks are outlined in the following paragraphs.

2.3.1. Initial temperature discovery

Determining the precise temperature alignment corresponding to the TEM_{00} mode can be done by manually adjusting the Mephisto's laser control and watching the traces on the oscilloscope. We automated this process as follows, with the 'Scanning Temperature' algorithm. This automated method initializes the temperature at 22° C and invokes a computational routine outlined in algorithm 1. In essence, the Red Pitaya device generates a progressively increasing voltage through its slow analog output channel. With each incremental voltage adjustment, accomplished by setting the `DAC2` value, i.e. Digital-to-Analog Converter Number 2, of the Red Pitaya, a data acquisition is performed, capturing two critical parameters: the signal received from the PDcav and the voltage applied to the laser's rapid frequency tuning port (i.e. RF Output 1). The nomenclature `orderedScopeTrace` aptly describes this method because it acquires a scope trace and subsequently restructures the data to ensure that the voltage ramp exhibits continuous progression without abrupt discontinuities, thereby positioning the trace's midpoint at the center of the voltage ramp. The algorithm exclusively relies on the `input2` signal, discerning its peak position to determine the termination condition of the iteration. If the peak occurs approximately at the midpoint of the ramp, within a specified tolerance threshold denoted as ϵ, the algorithm concludes, yielding a *True* result. In this case, PDH lock [23] can be performed to lock the cavity. Conversely, if the peak lies outside this central range, the algorithm continues to increment the voltage value, augmenting it by a fixed increment of 0.00 025 V. If the voltage value reaches 0.3 V without satisfying the termination criterion, the algorithm exits the loop and returns *False*. In such cases, the procedure entails a restart.

2.3.2. Automatic (re)lock

One straightforward approach to establishing lock and reestablishing it when it is lost is to execute Algorithm 1 each time the lock is disengaged. This procedure is elaborated upon in Algorithm 2, which keeps the system as described in figure 2 but does not employ the Q-Learning agent. It functions as follows:

(i) Initially, the Red Pitaya's outputs are reset, and the fast input is configured to produce a ramp signal by invoking the function `RampPiezo`, as described in Algorithm 3. This function sets the Arbitrary Signal Generator module 0 (`ASG0`) output of the Red Pitaya to a ramp waveform with the specified frequency, designed to be slow enough so the system responds accurately and fast enough that there are not significant changes during a scan (256 is the decimation factor), and the 'quadrature I' and 'quadrature Q' low-pass filter 0' (`IQ0`) to a quadrature signal in accordance with the input phase.

(ii) The `ScanTemperature` function is called, and if it returns *True*, signifying the successful identification of the correct mode, a 10 s waiting period is initiated. This waiting period serves two crucial purposes: firstly, it allows for more stable temperature conditions to be established within the crystal, and secondly, it enables the centering of the peak of the PDcav signal on the center of the ramp. The

Algorithm 2. Automatic Relock Function.

```
1: procedure AutoRelock
2:     Reset
3:     RampPiezo
4:     if ScanTemperature(500) then
5:         Sleep(10)
6:         LockCavity
7:         while True do
8:             Sleep(10)
9:             fast_out1, input2 ← Scope
10:            if Max(input2) < 0.95 then
11:                Break
12:            end if
13:        end while
14:    end if
15:    AutoRelock
16: end procedure
```

Algorithm 3. Ramping the Piezo.

```
1: procedure RampPiezo(phase = 15)
2:     Reset(self)
3:     ScanPiezo(freq = 1/(8E − 9 ∗ (2 ∗ ∗14) ∗ 256))
4:     setIQ0(phase = phase)
5: end procedure
```

centering process is facilitated by monitoring the cavity's transmitted optical power. Through careful observation using a cavity transmission camera, it was discovered that a TEM_{00} mode is achieved when the voltage on the cavity transmission photodetector, PDcav, exceeds 0.95 V.

(iii) Subsequently, the cavity is locked using the PDH technique [23].

(iv) At intervals of 10 s, the lock status is monitored by analyzing the voltage level of `input2`.

We denoted this process as our 'baseline' because it represents the most straightforward method to acquire lock. This is attributed to the fact that no further action is executed following the `LockCavity` function. In cases where conditions remain stable, the outlined approach proves adequate for locking a laser to a cavity. However, real-world scenarios often involve long term drifts that exceed the fast frequency actuator's dynamic range. For example, in this system, temperature variations in the room change the physical length of the cavity, which can be beyond the dynamic range of the fast control. This is usually handled by applying a frequency-domain filter to the PDH error signal and sending the resulting signal to the laser's slow wavelength control. This implementation replaces the standard technique with a Q-Learning agent.

2.4. Q-Learning

Q-Learning [25] stands out as an off-policy Temporal Difference (TD) [27] algorithm, within the realm of RL, designed to ascertain an optimal policy for an agent navigating its environment. TD methods exhibit a unique capability to glean insights directly from raw experiential data, obviating the need for a predefined model of the environment's dynamics. This distinguishes them from approaches such as Monte Carlo [28], as TD methods update their estimates by leveraging information from other learned estimates, allowing for a form of bootstrapping. This characteristic proves advantageous, particularly in scenarios with extended episodes, such as our experimental setup, where waiting until the episode's end for learning is impractical and hampers efficiency. In the case of Q-Learning, the estimation is encapsulated by the action-value function Q. Notably, this function directly approximates the optimal action-value function, irrespective of the specific policy being pursued, hence its classification as an off-policy algorithm. The action-value function in Q-Learning undergoes updates according to the following formula:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left(R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(s,a) \right), \tag{1}$$

Algorithm 4. General Q-Learning.

Parameters: $\alpha \in (0,1]$, small $\epsilon > 0$
Initialize $Q(s,a)$, $\forall s \in \mathcal{S}^+$, $a \in \mathcal{A}(s)$, arbitrarily except that $Q(terminal,\cdot) = 0$
for each episode **do**
    Initialize S
    for each step of episode **do**
        Choose A from S using policy derived from Q (e.g. ϵ-greedy)
        Take action A, observe R, S_{t+1}
        $Q(s,a) \leftarrow Q(s,a) + \alpha[R_{t+1} + \gamma \max_a Q(S_{t+1},a) - Q(s,a)]$
        $S \leftarrow S_{t+1}$
    end for
end for

where s represents the current state of the environment, a denotes the action taken by the agent in that state, R_{t+1} signify the reward received for executing that action in the given state, and S_{t+1} stand for the subsequent state following s. The parameters α and γ correspond to the learning rate and discount factor, respectively. These parameters play a pivotal role in the agent's decision-making process, as α governs the rate at which the agent incorporates new information into its understanding of the environment, while γ influences the discounting of future rewards. Higher values of γ give more weight to future rewards, thereby encouraging the agent to prioritize long-term gains in its decision-making.

The policy still has an effect in that it determines which state–action pairs are visited and updated. However, all that is required for correct convergence is that all pairs continue to be updated, e.g. taking advantage of an ϵ-greedy strategy [29]. The procedure is described in Algorithm 4.

Adapting the approach to our specific case necessitated the precise definition of our *states* and *actions*. The voltage values obtained from the PDcav fall within the range of –1 to +1 volts. To facilitate the utilization of Q-Learning, we discretized this voltage range into intervals of 0.1 volts, yielding a total of 21 distinct states. Similarly, the temperature underwent a similar treatment, with a set of actions defined for the slow analog input. These actions induced temperature changes in increments ranging from -0.001 V to 0.001 V, with a step size of 0.0005 V, resulting in a total of 5 actions. In addition to defining states and actions, other crucial parameters were established. α was set to 0.4, while the *discount factor*, represented as γ, was assigned a value of 0.99. To balance exploration and exploitation, an ϵ-*greedy policy* was adopted. Initially, ϵ was set to 0.7 to favor exploration, but after 1000 episodes, it was adjusted to 0.3. We define an episode from the moment the system undergoes lock to the unlock.

We defined the reward system such that a value of 1 was assigned for each time step during which the system successfully maintained lock. Conversely, if the lock was not sustained, the reward was set to 0, indicating the conclusion of the episode and necessitating a restart. This intricate process involves the iterative search for the initial temperature for the TEM_{00} mode, the execution of the PDH lock [23], and the application of the Q-Learning algorithm to ensure the ongoing maintenance of lock. For a comprehensive understanding of the procedure, Algorithm 5 provides a detailed overview. Essentially, it incorporates the Q-Learning technique into Algorithm 2. The `test` parameter is a Boolean that regularizes the agent's configuration parameters to be exclusively greedy.

3. Results & discussion

3.1. Software lock maintenance (baseline)

To appraise the efficacy of the baseline approach, our experimental setup involved running the code for an approximate period of 18 h. Within this timeframe, the system lost and restablished lock 32 times. Notably, the locks were successfully maintained for an average duration of half an hour, ranging from a minimum of 10s to a maximum of 2 h and 4 min. This demonstrated that we needed additional control to maintain lock for a longer period.

The behavior of the RF Output 1 signal is depicted in figure 3. It is evident from the illustration that deviations in voltage beyond certain thresholds result in the loss of lock. Specifically, when the signal drifts towards excessively high voltage values, exceeding 0.75 V, or too low, reaching approximately –0.4 V, lock is lost. This observation suggests that optimal performance is achieved when the signal hovers around its operational mean value, minimizing deviations. Notably, lock is consistently maintained during periods when the signal remains close to the mean value, emphasizing the importance of maintaining proximity to the optimal operating range.

Algorithm 5. Q-Learning.

```
procedure QLearning(episode)
    while episode < MAX_EPISODE do
        if episode == 1000 and not test then
            ϵ = .3
        end if
        Reset
        RampPiezo
        if ScanTemperature(500) then
            Sleep(10) & LockCavity
            systemUnlock ← False
            fast_out1, _ ← Scope
            state ← StateIndex(fast_out1)
            while do
                Sleep(1)
                if Rand < ϵ and not test then
                    action ← RandomChoise(Actions)
                else
                    action ← ArgMax(Q[state, :])
                end if
                SetDAC2(currentDAC2 + action)
                Sleep(0.1)
                fast_out1, input2 ← Scope
                reward ← 1
                if Max(input2) < 0.95 then:
                    systemUnlock ← True
                    reward ← 0
                end if
                nextState ← StateIndex(fast_out1)
                if not test then
                    maxNextQ ← Max(Q[nextState, :])
                    update Q
                end if
                state ← nextState
                if systemUnlock then
                    break
                end if
            end while
        end if
    end while
end procedure
```

3.2. Q-Learning training and testing

Our Q-Learning agent underwent an initial training phase on a CPU consisting of 5000 episodes, lasting for a total of approximately 5 days, during which the algorithm iteratively refined the Q-matrix in real-time while the laser was active. Throughout this process, the agent strategically explored various actions, occasionally resorting to random selections to assess its efficacy over the long term. Although this approach resulted in intermittent loss of stability due to suboptimal choices, it ultimately facilitated the learning agent's ability to discern and avoid unfavorable actions in specific states. In figure 4, we present a visual representation of four episodes during the training period. Notably, the RF Output 1 exhibits frequent fluctuations between high and low values. The mean voltage of RF Output 1 throughout the training phase was approximately 0.297 V.

Following the successful training of our Q-Learning agent, we subjected it to thorough testing by deploying a purely greedy strategy, where it prioritized actions with higher values in the action-value matrix corresponding to specific states. During the testing phase, the agent displayed remarkable performance, consistently maintaining a prolonged state of lock for 136 times longer than the previous case. For the most extended test run, we closely monitored the RF Output signal 1 of the Red Pitaya, with figure 5 depicting a part of the signal dynamics from the longest episode lasting approximately 12 days. The mean signal value throughout this extended period was 0.321 V. Figure 5 clearly illustrates that the agent's actions strategically steered the signal toward the mean value, effectively avoiding lock-loss states. This observation was consistent across various runs.

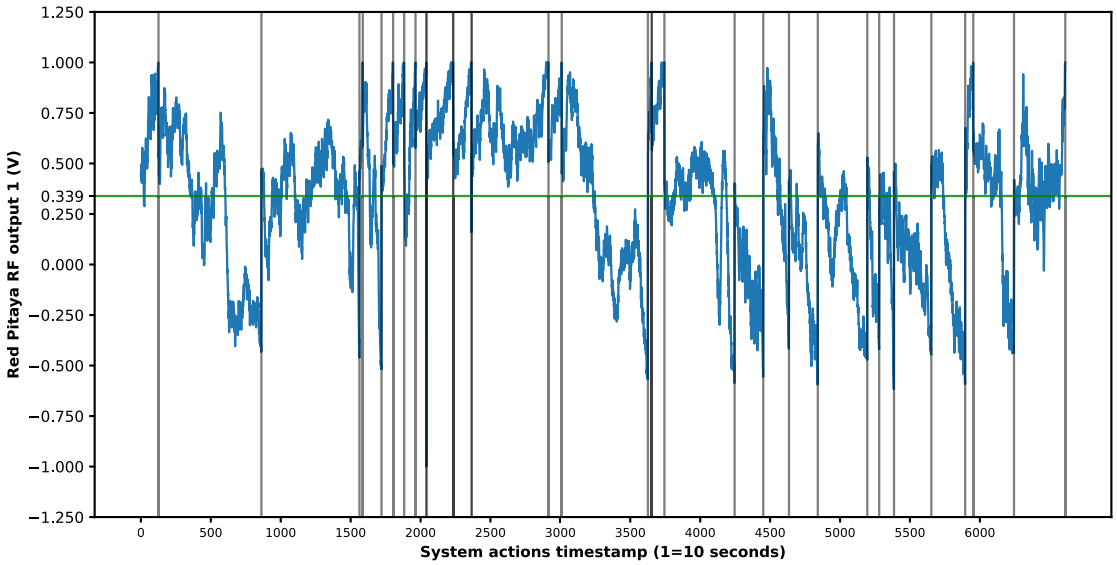

Figure 3. Visualization of RF output 1 signal during the testing phase of the software lock (baseline). Unlock events are marked by black vertical lines. The horizontal green line indicates the mean value. The *x*-axis indicates a counter for when the system takes a scope, which is done every 10 s.

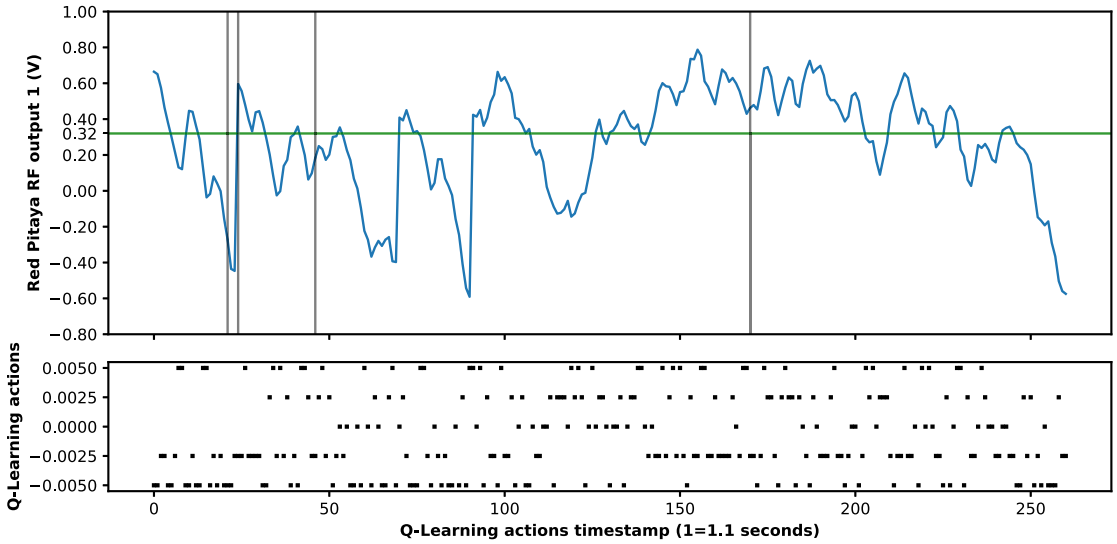

Figure 4. Visualization of RF output 1 (rapid signal) across 5 training episodes, showcasing diverse actions at each time step through distinct colors. Unlock events are marked by black vertical lines. The *x*-axis indicates a counter for when the system takes a scope trace, which is done every 1.1 s.

In-depth analysis of the agent's behavior during the extended episode is detailed in table 1, providing insights into the frequency of each action taken. In instances when the voltage exhibited fluctuations of less than 30mV and the agent refrained from taking action, we categorize the agent behavior as 'stable.' Agent actions that yield a >3mV increase in voltage are classified as 'positive,' and agent actions that yield a >3mV decrease in voltage are categorized as 'negative.' Notably, even in the absence of overt actions, the signal displayed a subtle decreasing trend, likely attributed to environmental factors. Fluctuations in ambient light or interference from external sources may have influenced the signal received by the photodetector.

The relationship between action and rapid signal changes is examined in table 1 by considering all actions executed during the testing phase, encompassing all testing episodes. An intriguing aspect emerged in the infrequent utilization of the action involving a 2.5 mV change. Examination of the Q-matrix revealed that this specific action was selectively executed when the signal fell within the –20 mV to –10 mV ranges, providing additional context to the agent's decision-making process and its nuanced response to specific signal conditions. This comprehensive analysis underscores the effectiveness and reliability of the trained Q-Learning agent in maintaining a long-lasting-lock state. In table 2, we present differences in lock time

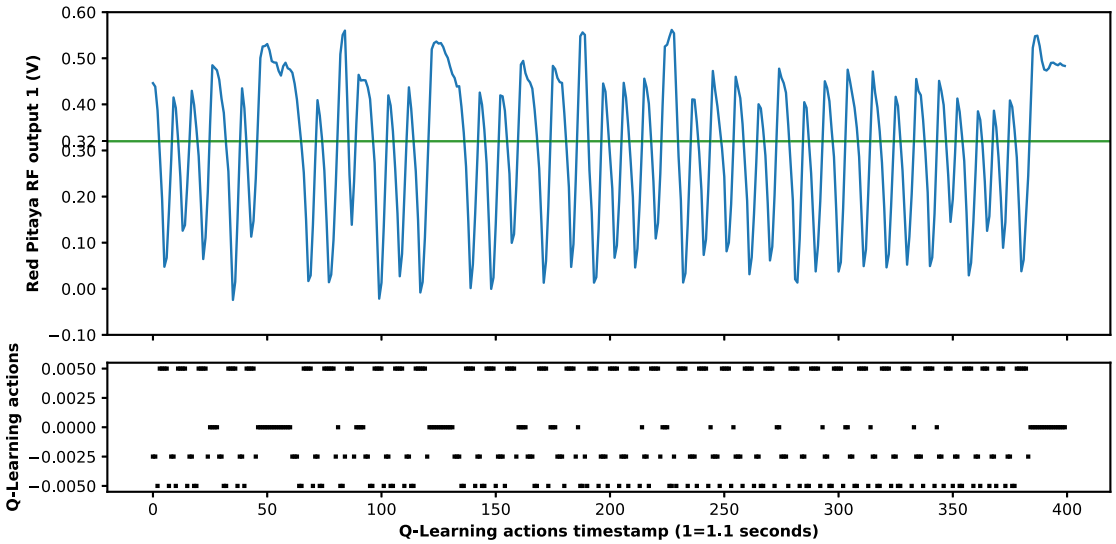

Figure 5. A segment of the RF Output Signal 1 of the Red Pitaya from the longest run during the testing phase. The dots denote the actions executed by the Q-Learning agent at distinct time points t, determined by the state s_t of the signal. The x-axis indicates a counter for when the system takes a scope, which is done every 1.1 s.

Table 1. The relationship between action and rapid signal changes is examined by considering all actions executed during the testing phase, encompassing all testing episodes. Average and standard deviation voltages are computed as the mean and standard deviation of voltage differences ($s_{t+1} - s_t$) for each action. In the 'stable' category, fluctuations are values < 30 mV. An increase happens when s_{t+1} exceeds s_t by > 3 mV, and vice versa for a decrease.

Action (mV)	Increase	Stable	Decrease	Avg (V)	Std (V)
–5	79 191	19 009	130 113	–0.016 606	0.10 763
–2.5	58 470	72 116	88 633	–0.02 504	0.06 991
0	18 238	178 888	38 696	–0.01 149	0.03 380
2.5	2	0	1	0.01 859	0.05 003
5	168 787	32 582	135 472	0.03 485	0.10 916

Table 2. Comparison of execution times between the baseline method, where the temperature is fixed at the value used during PDH lock, and Q-Learning. We present average (left column), minimum (central column) and maximum (right column) lock duration of the baseline and Q-Learning methods. The baseline method ran for 18.5 h, and the Q-Learning method ran for two weeks.

	Average (s)	Minimum (s)	Maximum (s)
Baseline	2087.64	10.08	7437.11
Q-Learning	686 820.06	179 730.63	1014 133.55

duration between our Q-Learning agent and the natural lock condition exhibited in the software automatic lock section. The baseline model demonstrated an average duration of 2087.64 s, equivalent to 34 min and 47 s, with a peak duration of 7437.11 s, corresponding to 124 minutes. Regrettably, due to its inherent instabilities, the baseline model exhibited a minimal lock duration of a mere 10 s. Conversely, the Q-Learning approach revealed a minimum lock duration of 179 730.63 s (approximately 2 days), underscoring its robustness and commendable performance. Notably, the average duration is remarkably impressive, showcasing 686 820.06 (approximately 8 days), with a maximum duration of 1014 133.55 (approximately 12 days). This highlights the efficacy of the Q-Learning approach as a superior strategy compared to the baseline, which lacks the benefits of RL.

4. Conclusion

In this study, we presented a novel methodology utilizing Q-Learning to sustain lock a PDH lock between a laser and an optical cavity, leveraging the Red Pitaya as a crucial interface between the laser system and our intelligent learning agent. Additionally, we introduced a software-based approach, harnessing the oscilloscope capabilities of the Red Pitaya board, for the recognition of the TEM_{00} Gaussian mode. Our investigation focused on the efficacy of the trained RL agent in enhancing lock maintenance over the natural course, during which laser parameters are traditionally fixed post-PDH lock. Our results vividly demonstrate

a substantial improvement in the duration of stable lock on the TEM_{00} mode, with an impressive increase from an average of 34 minutes to an outstanding eight days.

Importantly, our system exhibits the capability to learn in real-time while the laser is operational, capitalizing on the principles of Q-Learning–a Temporal-Difference method. The adaptive nature of our model ensures continuous improvement with each action undertaken by the agent-controlled Red Pitaya. Notably, the training process is seamlessly executed on a standard CPU, thanks to the simplicity of the update rule, which relies solely on efficient matrix multiplications. Given the simplicity of the method and resources required, we find this solution practical and scalable.

This technique holds paramount significance in high-sensitivity physics experiments. These experiments demand exceptionally stable conditions, particularly when grappling with the inherent challenge of managing small fluctuations in the emitted light from a laser source. Conversely, the light emanating from the cavity post-lock exhibits enhanced stability [30]. Consequently, the meticulous processes of establishing and maintaining the lock emerge as pivotal aspects in experiments of great precision, such as those focused on the detection of gravitational waves [31, 32]. This research was conducted within the scope of the Gravity from the Quantum Entanglement of Space-Time (GQuEST) experiment [33]. The primary objective of GQuEST is to enhance sensitivity beyond the conventional quantum limit by precisely counting individual photons, the fundamental energy packets constituting light.

Given its remarkable ability to swiftly adapt to evolving conditions and its facile training process, the presented model stands as a significant advancement in the realm of maintaining lock in optical systems. We posit that our approach serves as a foundational model, paving the way for further developments in enhancing the stability and performance of optical systems through intelligent learning agents.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: https://github.com/edofazza/cavity-lock-control.

Acknowledgments

This work was supported by the EU Horizon 2020 Research and Innovation Program under the Marie Sklodowska-Curie Grant Agreement No. 822185.

This manuscript has been authored by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U S Department of Energy, Office of Science, Office of High Energy Physics.

H A L gratefully acknowledges the support of the National Science Foundation through the LIGO operations cooperative agreement PHY18671764464.

We are very thankful to Aleksandra Ćiprijanović and Adam Schreckenberger for help in reviewing and editing this work.

Author contributions

E Fazzari: conceptualization, data curation, formal analysis, investigation, methodology, project administration, software, supervision, visualization, writing of original draft; H A Loughlin: optics setup and alignment, PDH lock implementation, electronics and controls, writing (editing sections relevant to PDH locking); C Stoughton: conceptualization, project administration, resources, supervision, writing (review and editing).

ORCID iDs

Edoardo Fazzari https://orcid.org/0000-0002-4570-4170
Hudson A Loughlin https://orcid.org/0000-0002-1160-8711

References

[1] Barto A G 1997 *Reinforcement Learning Neural Systems for Control* (Elsevier) pp 7–30
[2] Wang L, Yang S, Yuan K, Huang Y and Chen H 2023 *Chin. J. Mech. Eng.* **36** 80
[3] Sun J, Fang X and Zhang Q 2023 Reinforcement learning driving strategy based on auxiliary task for multi-scenarios autonomous driving *2023 IEEE 12th Data Driven Control and Learning Systems Conf. (DDCLS)* (IEEE) pp 1337–42
[4] Farzanullah M, Vu H V and Le-Ngoc T 2022 Deep reinforcement learning for joint user association and resource allocation in factory automation *2022 IEEE Wireless Communications and Networking Conf. (WCNC)* (IEEE) pp 2059–64
[5] Nambiar S, Wiberg A and Tarkian M 2023 *Front. Manuf. Technol.* **3** 1154263
[6] Malibari N Katib I and Mehmood R 2023 arXiv:2305.07466

[7] Abdellatif A A, Mhaisen N, Mohamed A, Erbad A and Guizani M 2023 *IEEE Internet Things J.* **10** 21982–2007
[8] Hu C, Wang Z, Shu T, Tong H, Togelius J, Yao X and Liu J 2022 *IEEE Trans. Games* **15** 202–16
[9] Souchleris K, Sidiropoulos G K and Papakostas G A 2023 *Appl. Sci.* **13** 2443
[10] Pou B, Ferreira F, Quinones E, Gratadour D and Martin M 2022 *Opt. Express* **30** 2991–3015
[11] Natalino C and Monti P 2020 The optical rl-gym: an open-source toolkit for applying reinforcement learning in optical networks *2020 22nd Int. Conf. on Transparent Optical Networks (ICTON)* (IEEE) pp 1–5
[12] Praeger M, Xie Y, Grant-Jacob J A, Eason R W and Mills B 2021 *Mach. Learn.: Sci. Technol.* **2** 035024
[13] Sun C, Kaiser E, Brunton S L and Kutz J N 2020 *Mach. Learn.: Sci. Technol.* **1** 045013
[14] Chang S C, Chang C P, Wang Y C and Chu C C 2023 *Tehnicki glasnik* **17** 268–72
[15] Drever R, Ford G, Hough J, Kerr I, Munley A, Pugh J, Robertson N and Ward H 1983 *General Relativ. Grav.* **94** 265
[16] Drever R W, Hall J L, Kowalski F V, Hough J, Ford G, Munley A and Ward H 1983 *Appl. Phys.* B **31** 97–105
[17] Ray A, Bandyopadhyay A, De S, Ray B and Ghosh P N 2007 *Opt. Laser Technol.* **39** 359–67
[18] Okamura H 2010 *Opt. Lett.* **35** 1175–7
[19] Petrenko A, Mikhailovskii G, Kotova E, Petrov A and Bugrov V 2019 *J. Phys.: Conf. Ser.* **1236** 012076
[20] Ma Y, Li Y, Tian K, Yang J, Dou X, Xu H, Han W and Liu J 2018 *Opt. Laser Technol.* **108** 360–3
[21] Němec M, Boháček P, Švejkar R, Šulc J, Jelínková H, Trunda B, Havlák L, Nikl M and Jurek K 2020 *Opt. Mater. Express* **10** 1249–54
[22] Mueller G, McNamara P, Thorpe I and Camp J 2005 Frequency stabilization for lisa *Technical Report* NASA Goddard Space Flight Center
[23] Black E D 2001 *Am. J. Phys.* **69** 79–87
[24] Svelto O and Hanna D C *et al* 2010 *Principles of Lasers* vol 1 (Springer)
[25] Watkins C J and Dayan P 1992 *Mach. Learn.* **8** 279–92
[26] Siegman A E 1986 *Lasers* (University Science Books)
[27] Sutton R S 1988 *Mach. Learn.* **3** 9–44
[28] Singh S P and Sutton R S 1996 *Mach. Learn.* **22** 123–58
[29] Watkins C J C H 1989 Learning from Delayed Rewards King's College
[30] Martin M J and Ye J 2012 *Optical Coatings and Thermal Noise in Precision Measurement G M* Harry, T Bodiya, R DeSalvo and eds pp 237–58
[31] Li D, Lee V S,H, Chen Y and Zurek K M 2023 *Phys. Rev.* D **107** 024002
[32] Verlinde E P and Zurek K M 2021 *Phys. Lett.* B **822** 136663
[33] Vermeulen S M *et al* 2024 arXiv:2404.07524