SODA: An Adaptive Bitrate Controller for Consistent High-Quality Video Streaming

Tianyu Chen
University of Massachusetts Amherst
Amherst, MA, USA
tianyuchen@umass.edu

Yiheng Lin
California Institute of Technology
Pasadena, CA, USA
yihengl@caltech.edu

Nicolas Christianson
California Institute of Technology
Pasadena, CA, USA
nchristianson@caltech.edu

Zahaib Akhtar
Amazon Prime Video / NCSU
Sunnyvale, CA, USA
akhtz@amazon.com

Sharath Dharmaji
Amazon Prime Video
Sunnyvale, CA, USA
sharatdr@amazon.com

Mohammad Hajiesmaili
University of Massachusetts Amherst
Amherst, MA, USA
hajiesmaili@cs.umass.edu

Adam Wierman
California Institute of Technology
Pasadena, CA, USA
adamw@caltech.edu

Ramesh K. Sitaraman
University of Massachusetts Amherst
Amherst, MA, USA
ramesh@cs.umass.edu

ABSTRACT

The primary objective of adaptive bitrate (ABR) streaming is to enhance users' quality of experience (QoE) by dynamically adjusting the video bitrate in response to changing network conditions. However, users often find frequent bitrate switching frustrating due to the resulting inconsistency in visual quality over time, especially during live streaming when buffer lengths are short. In this paper, we propose a practical smoothness optimized dynamic adaptive (SODA) controller that specifically addresses this problem while remaining deployable. SODA is backed by theoretical guarantees and has shown superior performance in empirical evaluations. Specifically, our numerical simulations show a 9.55% to 27.8% QoE improvement and our prototype evaluation shows a 30.4% QoE improvement compared to the state-of-the-art baselines. In order to be widely deployable, SODA performs bitrate horizon planning in polynomial time compared to brute force approaches that suffer from exponential complexity. To demonstrate its real-world practicality, we deployed SODA on a wide range of devices within the production network of Amazon Prime Video. Production experiments show that SODA reduced bitrate switching by up to 88.8% and increased average stream viewing duration by up to 5.91% compared to a fine-tuned production baseline.

CCS CONCEPTS

• **Information systems** → **Multimedia streaming**; • **Theory of computation** → **Online algorithms**; **Theory and algorithms for application domains**.

ACM SIGCOMM '24, August 4–8, 2024, Sydney, NSW, Australia

ACM ISBN 979-8-4007-0614-1/24/08
https://doi.org/10.1145/3651890.3672260

KEYWORDS

Adaptive bitrate streaming, Smoothed online convex optimization

ACM Reference Format:
Tianyu Chen, Yiheng Lin, Nicolas Christianson, Zahaib Akhtar, Sharath Dharmaji, Mohammad Hajiesmaili, Adam Wierman, and Ramesh K. Sitaraman. 2024. SODA: An Adaptive Bitrate Controller for Consistent High-Quality Video Streaming. In *ACM SIGCOMM 2024 Conference (ACM SIGCOMM '24), August 4–8, 2024, Sydney, NSW, Australia.* ACM, New York, NY, USA, 32 pages. https://doi.org/10.1145/3651890.3672260

1 INTRODUCTION

With the growth of online video streaming, users nowadays stream videos from a highly diverse set of devices, including laptops, mobile devices, smart TVs, set-top boxes, game consoles, etc. These devices span a wide spectrum of hardware capabilities and connect to the Internet in a multitude of ways, e.g., wireless, cellular, cable, etc. To ensure a high quality of experience (QoE) across all devices, video providers utilize adaptive bitrate (ABR) streaming that tailors video delivery to specific devices and network conditions.

The goal of ABR streaming is to deliver a video at the highest sustainable quality over time-varying network conditions. To achieve this, a video source is encoded at different bitrates corresponding to different resolutions, e.g., 720p, 1080p, 1440p, etc. Each encoding is in turn temporally partitioned into a sequence of *segments*, e.g., 2 seconds of video content. An ABR controller inside a user's video player then selects a suitable bitrate for each segment. Finally, downloaded segments are stored in a buffer, till they are rendered.

Past studies have shown that a user's QoE is maximized by delivering the video at the highest possible quality with minimal rebuffering and bitrate switching. It has been shown that a 1% increase in rebuffering time is correlated with a 3-minute reduction in the viewing duration [7] and frequent bitrate switching is strongly correlated with a user abandoning the session [21]. Going beyond correlational studies, the significant *causal* impact of rebuffering and other QoE performance metrics on key measures of user behavior was first established in [9]. However, jointly optimizing all three key components of QoE, i.e., video quality, rebuffering and bitrate

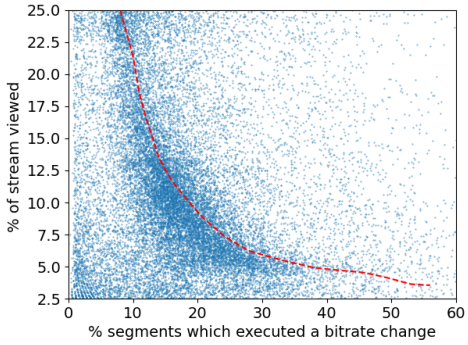

Figure 1: Video stream duration is negatively correlated with bitrate switching rate. Users watch < 10% of the stream when bitrate switching rate is > 20%.

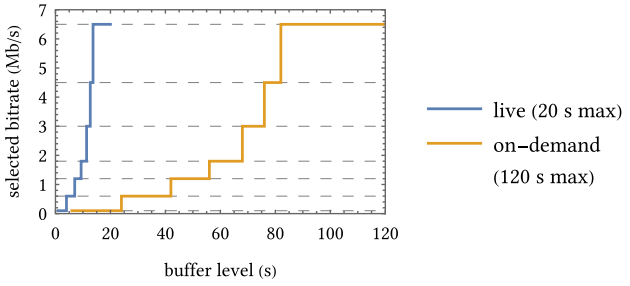

Figure 2: BOLA's [44] decision boundaries are spaced out for on-demand streaming, but tiny fluctuations in buffer level can cause bitrate switching for live streaming.

switching, is non-trivial as they are locked in a three-way trade-off. An ideal ABR controller seeks to push the trade-off boundary and optimize all three QoE components *simultaneously*.

Between on-demand and live streaming, the latter is more challenging as the player buffer is restricted to 10 - 20 seconds (to remain close to actual live action), which is in contrast to 60 - 180 seconds of buffer in on-demand streaming. Consequently, live streaming has higher susceptibility to rebuffering and bitrate switching. To understand the impact of bitrate switching, Figure 1 shows the relationship between the viewing percentage of a stream and bitrate switching rate for a sports event on a large-scale video streaming provider. To minimize potential confounders such as rebuffering and low quality, the plot is focused on short-lived sessions (< 25% of stream viewed) with at least HD quality and no rebuffering. The line of best fit shows that users watch < 10% of the stream when bitrate switching rate is > 20%. While our proposed ABR controller works for both on-demand and live streaming, our evaluations use live streams that represents a more challenging use case.

Our Contributions. We propose a novel **s**moothness **o**ptimized **d**ynamic **a**daptive (SODA) controller that provides theoretical QoE guarantees while exhibiting superior empirical performance in simulation, prototype, and production experiments. We make the following specific contributions:

1) **Theoretical Foundations of ABR Controller Design**. SODA is the first ABR controller to *provably* optimize *all three* key components of QoE, namely, video quality, rebuffering and bitrate switching. Unlike prior work such as BOLA [36, 44] that use Lyapunov methods to optimize the first two components, we use a new framework based on recent advances in smoothed online convex optimization (SOCO) [13, 25, 26, 33–35, 43] to simultaneously optimize all three QoE components. To enable the application of SOCO, we model the rebuffering minimization requirement in a novel fashion using the notion of buffer stability. We prove that SODA is near-optimal and achieves QoE within a small factor of the offline optimal QoE (Theorem 4.1).
2) **Better QoE Across Empirical Evaluations**. We evaluated SODA in three settings: numerical simulations, prototype evaluation, and production deployment within Amazon Prime Video serving actual users. Our numerical simulations show a 9.55% to 27.8% QoE improvement and our prototype evaluation shows a 30.4% QoE improvement compared to the state-of-the-art baselines. Production live streaming experiments in Amazon Prime Video show that SODA reduced bitrate switching by a significant up to 88.8% and increased average stream viewing duration by up to 5.91% (> 5 minutes longer sessions) compared to a fine-tuned production baseline. See Table 1 for a summary of our key findings about SODA as compared to baseline ABR controllers.
3) **Robustness Against Throughput Prediction Errors**. Most ABR controllers rely on and are sensitive to predictions of the future network throughput. Our SOCO framework allows us to design robust ABR controllers that are *provably* robust against prediction errors. Specifically, we show that SODA has the *exponentially decaying perturbation property* [49, 55, 56], i.e., the future impact of prediction errors decay rapidly over time. A key to our proof methodology is that we shifted from the conventional segment-based ABR formulation and adopted a novel *time-based* perspective.
4) **Efficient Implementation for Production Deployment**. ABR controllers deployed in the field need to work on a wide range of client devices, including low-end ones with limited computational resources. Many ABR controllers proposed in the research literature do not meet the efficiency bar for a production deployment and are never implemented in practice. We maximized SODA's runtime and deployment practicality by devising a computationally efficient method to search for near-optimal bitrate decisions, which reduced the runtime complexity from exponential to polynomial, e.g., about 200 iterations max in practice. In addition, we made SODA robust against throughput prediction errors by design, thus eliminating the need for sophisticated computationally-intensive throughput predictors.

This work does not raise any ethical issues.

2 DESIGN GAPS, OPPORTUNITIES, AND REQUIREMENTS

Design Gaps. Live streaming poses the additional constraint of near real-time delivery which makes bitrate adaptation more challenging than that in on-demand streaming. Figure 2 shows the bitrate selection function of BOLA [36, 40, 44], an ABR controller that is widely deployed by video providers and is part of the reference MPEG-DASH video player [64]. Notice that for on-demand

Table 1: A qualitative summary of our key evaluation findings about SODA as compared to baseline ABR controllers.

Controller	Theory[a]	Video Quality	Rebuffering Time	Switching Rate	Deployability
SODA	Q + R + S	high	short	ultra low	high
HYB [24]	none	high	medium	high	high
BOLA [44]	Q + R	high	short	high	high
Dynamic [36]	Q + R	high	short	medium	high
MPC [17]	none	high	long	low	low
Fugu [46]	none	high	medium	low	low
CausalSimRL [60]	none	high	short	high	low

[a]Q, R, S stand for theoretical guarantees for quality, rebuffering, and switching respectively.

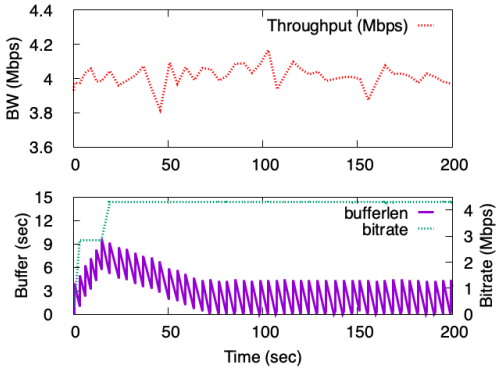

Figure 3: A RobustMPC session where the controller intentionally rebuffers instead of lowering the bitrate.

streaming, a longer buffer of 120 seconds ensures that bitrate jumps are spaced well apart (up to 20 seconds), however, for live streaming with a buffer of 20 seconds, bitrates fluctuate with small deviations of 1 - 3 seconds in the buffer size. This can cause bitrates to switch frequently. While controllers such as MPC [17] and Pensieve [22] offer respite by explicitly penalizing bitrate switching, these controllers suffer from shortcomings of their own:

- Model predictive controllers are hard to deploy at scale because they need to solve a non-linear integer programming problem over a prediction horizon of K segments, e.g., $K = 5$, which is so computationally expensive that it is quicker to download a video segment than to obtain a bitrate decision [17, 19]. Workarounds such as pre-computed lookup tables [17] are impractical for live streaming where the video is not available a priori. In a similar vein, learning-based controllers such as Pensieve [22] work optimally when trained specifically for a given set of bitrates, segment duration, network conditions, etc. Given in-the-wild diversity and its evolving nature, ensuring this specificity imposes a significant operational overhead. Furthermore, even if specifically trained, achieving performance guarantees with these learning-based controllers is shown to be challenging [24].
- Existing ABR controllers naively reduce bitrate switching at the expense of low video quality or more rebuffering. To demonstrate this, Figure 3 shows a RobustMPC session with the exact setup used by [17, 22]. Notice that beyond 70 seconds, RobustMPC repeatedly rebuffers but continues to download the highest bitrate (Figure 3 bottom plot), resulting in 29 rebuffering events over 200 seconds. Strikingly, this behavior is in fact the optimal behavior under RobustMPC's objective function which tolerates rebuffering to prevent bitrate switches. On the surface, this suggests higher rebuffering penalty in the objective function, however, higher penalties only reduce the duration of these *tolerable* rebuffers but do not eliminate them. Indeed, past work has empirically shown that even a 20× buffering penalty has marginal impact [24].
- Variance in network conditions or throughput prediction errors are not well tolerated by existing controllers. Past works have shown that RobustMPC incurs 26% more rebuffering events unless paired with a sophisticated throughput predictor [46, 50]. Similarly, learning-based controllers like Pensieve tend to degrade in performance when trained for realistic network conditions encountered in the wild [24]. In practice, accurate throughput predictions are hard because of several factors, including (i) device and OS level inefficiencies [18], (ii) stop-start nature of video requests which do not interact well with TCP [8, 18], and (iii) volatile network conditions typical in production networks [1, 5, 45]. To make matters worse, sophisticated throughput predictors are themselves not necessarily accurate [16] and are challenging to deploy due to device level bottlenecks [58]. Therefore, in-the-wild performance of these controllers remains questionable.

Opportunities. Outside of the video streaming literature, the interaction of learning and control has blossomed in recent years, leading to new and exciting approaches to controller designs [39, 47, 49, 52, 54]. However, these new approaches have not yet been applied and evaluated in the context of video streaming where classical model predictive and proportional–integral–derivative control have remained the focus, e.g., [17, 23]. In particular, the area of smoothed online convex optimization (SOCO) has seen multiple breakthroughs in recent years [13, 25, 33, 35], including the development of connections to model predictive controllers [26, 47, 49, 52, 56]. SOCO provides a systematic framework to balance an objective function with action switching. It thus lends itself well to video streaming, which needs to jointly optimize video quality, sustained playback, and bitrate smoothness.

Requirements. Driven by the above design gaps and opportunities, we identify three requirements that SODA should deliver. In particular, SODA should (i) achieve bitrate smoothness without

sacrificing video quality or sustained playback, (ii) be robust against volatile network conditions, and (iii) be easy to deploy in practice. Before delving into the details in the remainder of the paper, we provide a brief overview of how SODA satisfies these requirements:

- SODA leverages SOCO to *balance the trade-off* between video quality, sustained playback without rebuffers, and bitrate smoothness without frequent switches. Importantly, SODA focuses on steering the buffer level towards a target rather than weighing video quality against rebuffering duration (see Section 3.1).
- To achieve *robustness* against throughput variability, SODA is designed to satisfy the *exponentially decaying perturbation property*, which guarantees that SODA never operates too far away from the optimal trajectory in the face of prediction errors (see Section 4.2).
- To remain *computationally efficient*, SODA leverages an efficient approximate solver (see Section A.5 for proof and Algorithm 1 for implementation), that only requires evaluation of monotonic bitrate sequences (Section 4.3), which reduces the computational cost by two orders of magnitude over a brute-force solver.

3 SODA OVERVIEW

Given the design gaps, opportunities, and requirements, we set out to design a *theoretically sound* adaptive bitrate streaming (ABR) controller that minimizes bitrate switching without compromising video quality or increasing rebuffering time, thus providing a smooth viewing experience. To accomplish this, we deviate from the conventional segment-based ABR formulation and derive theoretical insights from a time-based ABR formulation. This enables us to incorporate throughput predictions into the controller in a principled way. Taking advantage of recent advancements in smoothed online convex optimization (SOCO), we can theoretically prove that SODA offers a near-optimal quality of experience (QoE) and is robust against throughput prediction errors.

3.1 A Time-Based ABR Formulation

Our time-based ABR formulation treats a video stream as a *continuous flow* rather than a discrete sequence of segments. Consider a streaming session that consists of N time intervals with fixed duration Δt in terms of *clock time* (not video time). The controller's task is to select a bitrate for each time interval from a set of available bitrates $\mathcal{R} \subset [r_{\min}, r_{\max}]$ to optimize for a combination of high quality, short rebuffering, and infrequent bitrate switching.

Let ω_n denote the average throughput during the n^{th} time interval, r_n the selected bitrate for that time interval, and x_n the buffer level immediately after that time interval. Our objective is to minimize the overall cost given as a linear combination of the three QoE components:

$$\sum_{n=1}^{N} \left(v(r_n) \cdot \frac{\omega_n \Delta t}{r_n} + \beta \cdot b(x_n) + \gamma \cdot c(r_n, r_{n-1}) \right), \tag{1}$$

where

- $v(r_n)$ is the **distortion cost**, which should be a positive, strictly decreasing, and convex function that models the encoding distortion, e.g., $v(r_n) = 1/r_n$. It is then weighted by the amount of video downloaded during that time interval, i.e., $\omega_n \Delta t / r_n$ because the controller downloads a *variable* amount of video during each fixed time interval.

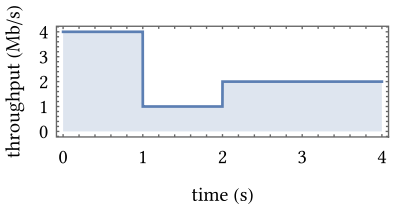

Figure 4: A sample throughput function used to illustrate why our time-based formulation is better for analysis.

- $b(x_n)$ is the **buffer cost**, which aims to stabilize the buffer level around a target level $\bar{x}$, i.e.,

$$b(x_n) = \begin{cases} (\bar{x} - x_n)^2 & x_n \le \bar{x} \\ \epsilon (x_n - \bar{x})^2 & x_n > \bar{x} \end{cases},$$

  where $\epsilon < 1$ is a small constant. Note that we purposely do not model the rebuffering time explicitly to avoid the pitfalls encountered by RobustMPC (Section 2) and as we show later, this helps SODA achieve theoretical performance guarantees (Section 4.2).
- $c(r_n, r_{n-1})$ is the **switching cost** from the previous bitrate to the current bitrate, e.g., $c(r_n, r_{n-1}) = (v(r_n) - v(r_{n-1}))^2$.

Coefficients β and γ are positive weights for the buffer and the switching cost respectively based on user preferences. The choices for the distortion and switching cost functions are flexible.

The *time-based* buffer dynamics are introduced into the optimization problem through the following constraint:

$$x_n = x_{n-1} + \frac{\omega_n \Delta t}{r_n} - \Delta t \in [0, x_{\max}],$$

where $\omega_n \Delta t / r_n$ accounts for the variable amount of video downloaded during a time interval and Δt accounts for the fixed amount of buffer drained during the same time interval. Note that we do not allow the controller to violate the buffer range constraint during the *optimization phase* when determining the bitrate. Of course, due to throughput prediction errors, this may sometimes be inevitable during the *execution phase* when applying the bitrate decision.

Why a Time-Based Formulation? The time-based formulation allows a cleaner theoretical analysis over a given throughput sequence $(\omega_1, \ldots, \omega_N)$. For example, consider the throughput function shown in Figure 4. In the time-based formulation, we naturally have $\omega_1 = 4$, $\omega_2 = 1$, and $\omega_3 = \omega_4 = 2$ Mb/s given $\Delta t = 1$ s. By contrast, in the segment-based formulation, the throughput sequence becomes dependent on the bitrate sequence. Assuming the segment duration is also $L = 1$ s, if the controller chooses $r_1 = 2$ Mb/s and $r_2 = 2.5$ Mb/s, then it takes 0.5 and 1 s to download the first and second segments respectively, resulting in $\omega_1 = 4$ and $\omega_2 = 2.5$ Mb/s. As such, the segment based formulation gets causally biased due to bitrate selection $r_1, ..., r_N$, which in turn makes it difficult to theoretically analyze the design [61].

Why Not Model Rebuffering Directly? Rebuffering is important to minimize from a user's perspective [7, 9]. However, in our optimization problem formulation, we did not explicitly model rebuffering like prior works [17, 46, 50]. Instead, we focus on stabilizing the buffer level around a target level with a smooth roll-off on both sides for the following reasons:

- Minimizing rebuffering directly is not theoretically tractable because it requires a binary penalty function that yields a non-zero penalty exactly when the buffer is empty. Instead, we employ a smoother penalty function that increases in magnitude when the buffer level falls below a desired target level. When there is a network issue, we start to penalize early when the buffer level decreases below the safe target level and we provide the largest penalty when the buffer level is empty. Using a smooth penalty function enables us to guarantee that SODA's optimization is strongly convex, which is key to our theoretical work. Our approach is analogous to the use of control barrier functions to ensure safety properties in control systems [30].
- Modeling rebuffering time directly makes the controller vulnerable to throughput prediction errors. Under a direct rebuffering objective, as long as the buffer level is above zero, there will be no penalty for the controller, even if the buffer level is dangerously close to 0. As a result, even small throughput prediction errors can lead to unexpected rebuffering.

3.2 Incorporating Throughput Predictions

In addition to facilitating theoretical analysis, our time-based formulation is crucial to ensuring the validity of throughput predictions over the prediction horizon. An important observation is that *bitrate decisions have no causal impact on how long the throughput predictions are valid for*. However, segment-based controllers such as MPC [17] and Fugu [46] intertwine throughput predictions and bitrate decisions in non-causal ways. In these designs, the throughput prediction horizon spans shorter periods of clock time when low bitrate is selected compared to when high bitrate is selected. In fact, their underlying assumption about the validity of the throughput prediction horizon can vary by $r_{\max}/r_{\min}$.

By contrast, the way we incorporate throughput predictions into SODA *does not* suffer from this issue. Specifically, just before each time interval, the controller is given access to a (not necessarily accurate) throughput prediction for the next K time intervals from a black-box throughput predictor. It is always assumed that the validity of the throughput prediction is $K\Delta t$, a fixed value. In general, a throughput predictor may output a different value for each of the next K time intervals, i.e., $\hat{\omega}_{n|n-1}, \hat{\omega}_{n+1|n-1}, \ldots, \hat{\omega}_{n+K-1|n-1}$, where $\hat{\omega}_{m|n-1}$ $(m \geq n)$ is the throughput prediction for the m^{th} time interval given previous download information up until the $(n-1)^{\text{th}}$ time interval. In other words, a throughput predictor can output a piecewise constant throughput function for the next $K\Delta t$ time. In practice, though, a typical throughput predictor outputs a single value that corresponds to a constant throughput function.

3.3 Control Mechanism

Inspired by the model predictive control framework, SODA selects a bitrate for each time interval by optimizing over the next K time intervals and then committing to the bitrate decision for the immediate next time interval, i.e., minimizing

$$\sum_{m=n}^{n+K-1} \left(v(r_m) \cdot \frac{\hat{\omega}_{m|n-1}\Delta t}{r_m} + \beta \cdot b(x_m) + \gamma \cdot c(r_m, r_{m-1}) \right) \quad (2a)$$

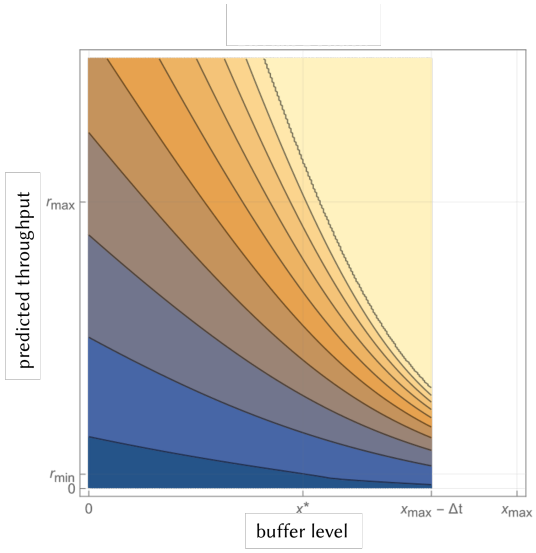

Figure 5: SODA's bitrate decision as a function of buffer level and predicted throughput. Dark blue to light orange represent low to high bitrate decisions. Notice that SODA becomes more aggressive in selecting higher bitrates as the buffer grows. The rightmost region is blank since SODA makes no downloads to prevent a buffer overflow.

$$\text{subject to} \quad x_m = x_{m-1} + \frac{\hat{\omega}_{m|n-1}\Delta t}{r_m} - \Delta t, \quad (2b)$$

$$x_m \in [0, x_{\max}], \quad r_m \in \mathcal{R}, \quad (2c)$$

with respect to variables $r_n, \ldots, r_{n+K-1}$ and then committing to only the first bitrate decision r_n. The behavior of SODA is visualized as a bitrate decision diagram in Figure 5 to provide readers with intuition about how SODA selects bitrates in practice.

As discussed in Section 2, solving this optimization problem is computationally expensive, furthermore, it is unclear what prediction horizon should be used and how accurate throughput predictions must be in order for SODA to perform well. We first analyze these questions theoretically (Section 4) and then present a practical implementation of SODA that answers these concerns (Section 5).

4 THEORETICAL DESIGN INSIGHTS

Our design of SODA is motivated by recent theoretical advances at the interface of learning and control [28, 38, 49, 54] and smoothed online convex optimization [25, 33, 55]. In particular, we design SODA to satisfy an *exponentially decaying perturbation property* that has been shown to ensure efficient and robust use of predictions in model predictive control policies [49, 56]. Intuitively, this property describes the behavior of the solution to the optimization problem defining SODA (Equation 2) as a function of problem parameters, including bandwidth predictions $\{\hat{\omega}_{m|n-1}\}_{n \leq m < n+K}$ and the previous buffer level/action pair (x_{n-1}, u_{n-1}). Here, we define the *actions* as the inverse of the bitrates (i.e., $u_\tau = 1/r_\tau$ for all time step τ) and do a change of the variables to make the dynamics linear for the theoretical analysis. Under this property, when $\{\hat{\omega}_{m|n-1}\}_{n \leq m < n+K}$ are fixed, the optimal trajectory of (Equation 2) under the initial

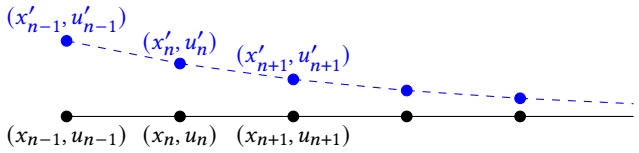

Figure 6: Illustration of the exponentially decaying perturbation property: When $\{\hat{\omega}_{m|n-1}\}_{n \le m < n+K}$ are fixed, the optimal trajectories of Equation 2 under different initial buffer/action pairs converge exponentially toward each other.

buffer/action pair (x'_{n-1}, u'_{n-1}) converges exponentially toward the optimal trajectory under the pair (x_{n-1}, u_{n-1}) (see Figure 6 for an illustration). On the other hand, when the initial buffer/action pair is fixed, the impact of perturbing a prediction $\hat{\omega}_{m|n-1}$ on the first action u_n decays exponentially with respect to their temporal distance $(m-n)$. The formal definition of exponentially decaying perturbation generalizes the intuition above to consider the impact of perturbing any parameters on the entire optimal trajectory (see Definition A.1 in Appendix A).

Two metrics that we use to measure SODA's performance theoretically are *dynamic regret* and *competitive ratio*, which are standard in the literature of online optimization [25, 28, 33, 49, 54]. Specifically, let cost(ALG) denote the total cost incurred by an online algorithm ALG and cost(OPT) denote the offline optimal cost (Equation 1) an agent can incur if it has exact knowledge of all future bandwidth at the beginning. We say ALG achieves a dynamic regret of R if cost(ALG) − cost(OPT) $\le R$ always holds, and ALG achieves a competitive ratio of C if cost(ALG) $\le C \cdot$ cost(OPT) always holds.

The key idea underlying our theoretical analysis is to leverage the exponential decay property to bound (i) the error that SODA incurs at every intermediate time step n due to its limited prediction power ($\hat{\omega}_{m|n-1} \neq \omega_m, K \ll N$), and (ii) the aggregation of such errors over the whole horizon N. Specifically, we define the notion of *per-step error* at a time step n as the distance between SODA's buffer/action pair and the optimal buffer/action pair that one could reach with exact predictions of all future bandwidths $\omega_n, \omega_{n+1}, \ldots, \omega_N$ given the previous buffer/action pair (x_{n-1}, u_{n-1}) (Definition A.2). Using the principle of optimality, we reformulate the optimal buffer/action pair as an entry of the optimal trajectory from time n to $(n+K-1)$ so that we can directly compare it with SODA's buffer/action pair under the exponentially decaying perturbation. Thus, we establish a bound on the per-step error that depends on the errors of predicting future bandwidths and the prediction horizon K (Lemma A.4). On the other hand, we also show that the aggregation of per-step errors does not grow linearly in time because the exponentially decaying perturbation guarantees that the impact of each previous per-step error vanishes exponentially over time (Lemma A.5). We present a proof outline and the detailed proofs in Appendix A. To prove the exponentially decaying perturbation, we require a technical assumption that guarantees the controller can "reach" any desired buffer level by choosing the largest/smallest bitrate (see Assumption A.1 in Appendix A for the formal statement). This assumption is used to eliminate extreme boundary cases in the analysis, but we find SODA empirically performs very well even when this assumption is not strictly satisfied.

In this section, we set $\Delta t = 1$, $\mathcal{R} = [r_{\min}, r_{\max}]$, and $v(r) = 1/r$. Our results can apply to other distortion cost functions, e.g., $v(r) = \log(r_{\max}/r)$, as long as certain regularity conditions hold; see Appendix B for a discussion.

4.1 Exact Predictions

When the bandwidth predictions are accurate, a small prediction horizon is sufficient for SODA to achieve near-optimal performance. In practice, it is desirable to use a relatively small prediction horizon for a predictive controller like SODA because prediction errors grow dramatically as we predict further into the future. Fortunately, the exponential decay property that ensures good performance with only a few predictions. More formally, we present a theorem showing that a small prediction horizon is sufficient for SODA to achieve near-optimal performance when the predictions within this window are accurate (i.e., $\hat{\omega}_{m|n-1} = \omega_m$ for $m = n, \ldots, n+K-1$).

Theorem 4.1. *[Informal] When the predictions of the bandwidth in future K steps are exact (i.e., $\hat{\omega}_{m|n-1} = \omega_m$ for $m = n, \ldots, n+K-1$) and the prediction horizon $K \ge O(1)$, SODA achieves a dynamic regret of $O(\rho^K N)$ and a competitive ratio of $1 + O(\rho^K)$, where $\rho < 1$ is the decay factor of the exponentially decaying perturbation property.*

The formal statement of Theorem 4.1 is given in Theorem A.3 in Appendix A. This result implies that SODA's performance approaches that of the optimal sequence of decisions *exponentially fast* in the prediction horizon size K; thus, only a small prediction horizon length is necessary to obtain good performance.

4.2 Inexact Predictions

We now relax the exact prediction assumption to prove SODA's robustness to a certain level of prediction errors thanks to its exponentially decaying perturbation property.

Theorem 4.2. *[Informal] Suppose the prediction error at each step is bounded above. The buffer level of* SODA *will never hit the constraint boundary, i.e., $0 < x_n < x_{max}$. Further, define $\mathcal{E} = \rho^{2K} N + \sum_{\kappa=1}^{K} \rho^{\kappa} E_{\kappa}$, where E_{κ} is the total squared error for predicting κ steps into the future.* SODA *achieves a dynamic regret of $O(\sqrt{\mathcal{E}N} + \mathcal{E})$.*

The formal statement of Theorem 4.2 is given in Theorem A.8 in Appendix A. Theorem 4.2 shows that, if the buffer costs are "steep" and the prediction errors on the bandwidth are relatively small, SODA can achieve a sequence of buffer levels that stay safely away from the boundaries of buffer constraint $[0, x_{\max}]$. The dynamic regret of SODA depends on the magnitude of the prediction errors and the regret improves when the errors become smaller. SODA acquires this guarantee thanks to its maintenance of the buffer near a target level $\bar{x}$. In contrast, RobustMPC [17] doesn't offer the same performance guarantee, thus even small bandwidth prediction errors can cause the video to rebuffer if the buffer level is near zero.

4.3 Computational Efficiency

Solving the predictive optimization problem to determine the exact optimal solution can be unrealistic in the application of adaptive bitrate streaming, where each decision needs to be made in the

minimum possible time. A critical observation underlying the implementation of SODA is that it is sufficient to search only for bitrate sequences that are increasing or decreasing monotonically. We provide a theoretical justification in the following theorem.

Theorem 4.3. *[Informal] Suppose* SODA *is given the predictions that satisfy* $\hat{\omega}_{n|n-1} = \cdots = \hat{\omega}_{n+K-1|n-1}$ *at an intermediate time step* n*. Then, the bitrate trajectory solved by* SODA *can be approximated by a feasible monotonic bitrate trajectory with an error of* $O\left(K/\sqrt{\gamma}\right)$.

The formal statement of Theorem 4.3 is given in Theorem A.9 in Appendix A. Theorem 4.3 shows that the true optimal solution becomes closer to monotonic as the weight γ of switching costs increases. While the theoretical bound can be conservative, we find that even with moderate γ, the (discrete) decision made under the monotonic heuristic is usually identical to the true optimal solution on a real trajectory (see Figure 8).

5 IMPLEMENTATION DETAILS

Given the theoretical design insights, we now discuss the practical implementation of the high-level design described in Section 3. There are three practical concerns that require discussion: (i) how to translate the time-based design to the segment-based schema; (ii) how to incorporate throughput predictions robustly; and (iii) how to solve the predictive optimization problem efficiently.

5.1 Segment-Based Schema

SODA is intrinsically a time-based controller, but in practice, a video must be downloaded segment by segment according to the MPEG-DASH standard. To reconcile with this requirement, we keep the optimization phase as is in the time-based format and empirically set Δt to be equal to the segment length. This choice is justified by the fact that in the steady state, the download time of a video segment is expected to be close to the segment length or much less than that [29]. To further minimize the likelihood of committing to a bitrate for significantly longer than Δt, we introduce another heuristic that the controller must select a bitrate no higher than $\min\{r \in \mathcal{R} : r \geq \hat{\omega}\}$.

5.2 Incorporating Predictions Robustly

According to Section 4.2, SODA is robust against prediction errors by design as long as there is no systematic bias in prediction errors. Given the diverse network conditions in the wild, we prefer simple throughput predictors which makes SODA highly deployable since there is no dependence on complex throughput predictors. In practice, we observe that prediction accuracy degrades as the prediction horizon increases (see Figure 7). Therefore, we limit the prediction horizon length to at most 10 s. This is also supported by our finding in Section 4.1 that a longer prediction horizon yields diminishing returns.

5.3 Efficient Approximate Solver

At SODA's core is the predictive optimization problem described in Section 3.3. Unfortunately, solving this problem on the fly is computationally challenging. One may propose enumerating all combinations of discretized throughputs, buffer levels, and previous bitrates in the form of an offline computed lookup table, as is the case in FastMPC [17], however, this is neither flexible nor scalable in practice. A lookup table is specific to a particular set of bitrates, maximum player buffer, segment durations and byte sizes *etc.* thus needs to be recomputed when any of these quantities change. Furthermore, computing this lookup in live streaming is undesirable due to the additional computational and latency overhead it incurs. Instead, we opt for an efficient approximate solver.

SODA's approximate solver is designed to take advantage of the structure of the optimal solution presented in Section 4.3. Instead of searching through all possible bitrate sequences in the prediction horizon, the approximate solver only considers *monotonic* bitrate sequences, i.e., it imposes an additional constraint that $r_{n-1} \leq r_n \leq \ldots \leq r_{n+K-1}$ or $r_{n-1} \geq r_n \geq \ldots \geq r_{n+K-1}$. The pseudocode for a recursive implementation is shown in Algorithm 1.

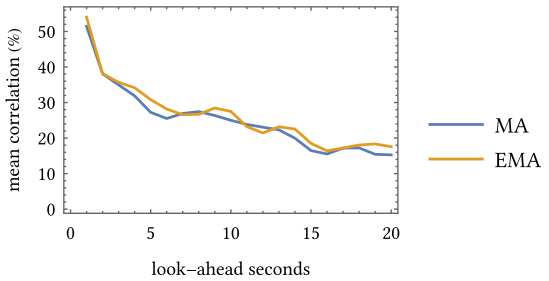

Figure 7: We profiled the performance of the two throughput predictors shipped with `dash.js` [64], i.e., moving average predictor and exponential moving average predictor. Both predictors have a high mean correlation (around 50%) in the immediate future but a very low mean correlation (around 15%) in the far future.

Algorithm 1: SODA's efficient approximate optimization solver. SearchDown is omitted for brevity due to symmetry. The current buffer level and the previous bitrate are denoted by x_0 and r_0 respectively.

function Search($\hat{\omega}, x_0, r_0, K$)
  $(r^*_{\text{up}}, \text{obj}^*_{\text{up}}) \leftarrow$ SearchUp($\hat{\omega}, x_0, r_0, K$)
  $(r^*_{\text{down}}, \text{obj}^*_{\text{down}}) \leftarrow$ SearchDown($\hat{\omega}, x_0, r_0, K$)
  return $r^*_{up} \neq \texttt{null} \wedge obj^*_{up} < obj^*_{down}\ ?\ r^*_{up} : r^*_{down}$
function SearchUp($\hat{\omega}, x_0, r_0, K$)
  $r^*_1 \leftarrow \texttt{null}, \text{obj}^* \leftarrow \infty$
  foreach $r_1 \in \{r \in \mathcal{R} : r > r_0\}$
    $x_1 \leftarrow x_0 + \hat{\omega}\Delta t / r_1 - \Delta t$
    if $x_1 < 0$ **then continue**
    $\text{obj} \leftarrow v(r_1) \cdot \hat{\omega}\Delta t / r_1 + \beta \cdot b(x_1) + \gamma \cdot c(r_1, r_0)$
    if $K > 1$ **then**
      $(r^*_2, \Delta\text{obj}^*) \leftarrow$ SearchUp($\hat{\omega}, x_1, r_1, K-1$)
      if $r^*_2 = \texttt{null}$ **then continue**
      $\text{obj} \leftarrow \text{obj} + \Delta\text{obj}^*$
    if $obj < obj^*$ **then** $r^*_1 \leftarrow r_1, \text{obj}^* \leftarrow \text{obj}$
  return (r^*_1, obj^*)

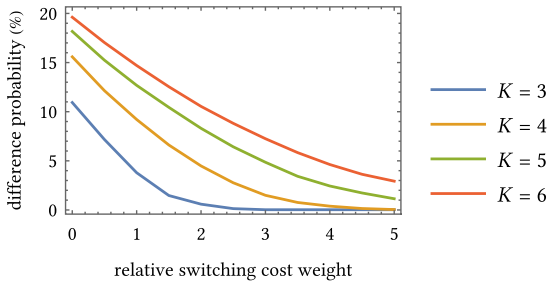

Figure 8: The probability that the bitrate decision produced by the approximate solver is different from that produced by the brute-force solver quickly converges to 0 as switching cost weight increases.

The approximate solver reduces the time complexity from $O(|\mathcal{R}|^K)$ (exponential in K) in the case of a brute-force search over all possible bitrate sequences in the prediction horizon down to $O\left(\binom{|\mathcal{R}|+K}{K}\right)$ (polynomial in K) and has a space complexity of $O(K)$ only. The time complexity can be further reduced by limiting extreme bitrate switches. In practice, SODA searches through at most around 200 bitrate sequences. According to our production deployment experience, the approximate solver did not impose a runtime burden even on low-end devices such as set-top boxes, which shows that SODA is highly practical.

Empirical results are shown in Figure 8 to validate the near-optimality of bitrate decisions produced by the approximate solver. For each algorithm configuration, we uniformly sample a million situations with different throughputs, buffer levels, and previous bitrates. Then, we count the probability that the bitrate decision produced by the approximate solver is different from that produced by the brute-force solver. The difference is negligible for a reasonable switching cost weight, e.g., below 5% for $K = 4$ and a relative switching cost weight of 2. Throughout the evaluation sections, we use this efficient implementation of SODA.

6 EVALUATION

To thoroughly evaluate SODA's performance, we conducted three levels of empirical evaluation: (i) large-scale numerical simulations, (ii) prototype evaluation in Puffer [46], and (iii) production deployment in Amazon Prime Video. This funnel approach allowed us to first systematically evaluate SODA against a variety of baselines in a wide range of controlled environments. Later, we narrowed the comparison target to a deployed and fine-tuned ABR controller in production using A/B tests on real user sessions.

Performance Metrics. To maintain consistency in terms of performance metrics with prior works such as [17, 22, 24, 46], a similar definition of QoE is adopted that consists of mean utility, rebuffering ratio, and switching rate. These correspond to the three main desired properties of adaptive bitrate streaming, i.e., high video quality, shorter rebuffering time, and less bitrate switching. All three QoE components are normalized between 0 and 1 for ease of interpretation. The precise definitions are as follows:

- **Mean Utility**: Unless otherwise noted, we use the commonly-used logarithmic utility function:

$$\bar{v} = \frac{1}{N}\sum_{i=1}^{N}\frac{\log(r_i/r_{\min})}{\log(r_{\max}/r_{\min})}.$$

- **Rebuffering Ratio**: The ratio of the total rebuffering time to the session duration, i.e., $\rho_{\text{rebuf}} = T_{\text{rebuf}}/T$.
- **Switching Rate**: Bitrate switch count divided by segment count minus one, i.e., $p_{\text{switch}} = N_{\text{switch}}/(N-1)$.

The QoE score is simply a linear combination of the three QoE components, i.e., $\text{QoE} = \bar{v} - \beta \cdot \rho_{\text{rebuf}} - \gamma \cdot p_{\text{switch}}$. In this work, we chose $\beta = 10$ and $\gamma = 1$ to reflect the high importance of minimizing rebuffering time. To establish fair comparisons, we report the individual QoE components along with the QoE score.

6.1 Numerical Simulations

To perform large-scale numerical simulations, we implemented a highly optimized ABR simulator in C++ derived from `Sabre` [36]. The simulation accuracy of `Sabre` has been empirically validated against `dash.js` [64], the reference player for MPEG-DASH. We configured the simulator to allow a maximum buffer length of 20 seconds to replicate the typical live streaming conditions.

6.1.1 Experimental Setup. Our network dataset consists of about 38,000 hours of throughput traces compiled from the following three public sources:

- **Puffer Dataset** [46]: We downloaded and parsed all throughput traces from the Puffer platform during the time period of January 2023 to June 2023.
- **5G Dataset** [41]: A 5G network dataset from a major Irish mobile operator under both static and moving scenarios while downloading online content.
- **4G Dataset** [27]: A 4G network dataset from two major Irish mobile operators under both static and moving scenarios while downloading online content.

For all three datasets, we filtered out sessions shorter than 10 minutes and divided long sessions into consecutive 10-minute sessions, resulting in 230,322 sessions from the Puffer dataset, 88 sessions from the 5G dataset, and 187 sessions from the 4G dataset. Figure 9 illustrates the wide range of network conditions covered by these datasets. In general, the Puffer dataset represents better network conditions than the 5G and 4G datasets. The latter have much lower mean throughput and higher variance, thus posing a bigger challenge for ABR controllers.

To fully exercise our datasets, we considered a high-frame-rate 4K video encoded according to the YouTube recommended settings (1.5, 4, 7.5, 12, 24, and 60 Mb/s) [65] with a segment length of 2 seconds. For the 5G and 4G datasets, we considered the same video with the two highest bitrates removed. Finally, for throughput prediction, we opted for the exponential moving average (EMA) predictor, the default throughput predictor in `dash.js`.

6.1.2 Baseline ABR Controllers. We compared SODA against the following ABR controllers representative of each of the common ABR controller categories, i.e., throughput-based, buffer-based, and hybrid. They were tuned to our best efforts for our network datasets.

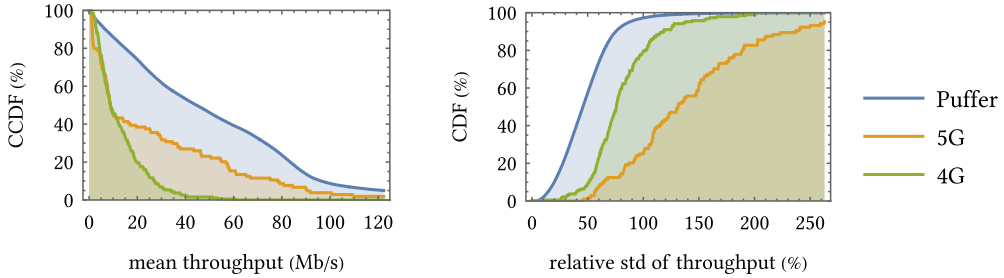

Figure 9: The mean throughput of the Puffer, 5G, and 4G datasets are 57.1, 31.3, and 13.0 Mb/s. The mean relative standard deviations of throughput of the Puffer, 5G, and 4G dataset are 47.2%, 133%, and 80.6%.

- **HYB** [24]: A heuristic throughput-based ABR controller that selects the highest bitrate without rebuffering.
- **BOLA** [36]: A buffer-based ABR controller derived from Lyapunov optimization. It provides theoretical guarantees about utility and rebuffering time only.
- **Dynamic** [44]: A production version of BOLA that dynamically switches between buffer mode and throughput mode in response to changes in network conditions. Additionally, it has low-buffer safety heuristic to reduce rebuffering and a switching avoidance heuristic to mitigate bitrate switching. It is the default ABR controller in `dash.js`.
- **MPC** [17]: One application of model predictive control to adaptive streaming that models utility, rebuffering time, and bitrate switching, without theoretical guarantees.

6.1.3 QoE Performance. The aggregate statistics for QoE scores and individual QoE components under each network dataset are shown in Figure 10. To better understand how the performance of different ABR controllers react to the intrinsic volatility of network conditions, we split the Puffer dataset into four quarters according to the relative standard deviation of throughput (Q1 represents the most stable network conditions, while Q4 represents the most volatile network conditions). In general, the more volatile network conditions are, the more the QoE performance of any ABR controller degrades, as evidenced by the trend in Figure 10 from left to right. Nonetheless, SODA consistently outperforms baseline ABR controllers under all network conditions. The improvement in terms of mean QoE scores compared to the best baseline across different network datasets ranges from 9.55% to 27.8%, which mainly stems from improvement in terms of smoothness (shorter rebuffering time and less bitrate switching). We discuss the improvement of SODA over each baseline ABR controller below:

- **SODA vs HYB**. HYB is not as robust as SODA under volatile network conditions. In addition, it switches up to 215% more since it does not consider bitrate switching.
- **SODA vs BOLA & Dynamic**. As mainly buffer-based ABR controllers, BOLA and Dynamic are fairly robust against volatile network conditions. Dynamic's performance is what one would expect in a typical production environment. Nonetheless, SODA is able to achieve similar mean utilities without sacrificing mean rebuffering ratios, proving its outstanding robustness as a hybrid ABR controller. Where SODA really shines though is its significantly lower mean switching rates. Despite Dynamic's switching avoidance heuristic, SODA cuts down mean switching rates by as much as 70.4%, which demonstrates the superiority of theoretically-sound design.
- **SODA vs MPC**. MPC has high mean utilities and low mean switching rates under stable network conditions (see Puffer (Q1 variance) in Figure 10). However, the performance of MPC is tightly coupled with the intrinsic volatility of network conditions. Specifically, MPC suffers a lot in terms of mean rebuffering ratios especially under mobile network conditions. By contrast, SODA does not have this issue since it is robust against prediction errors by design, making it much more suitable for production deployment.

6.1.4 Intrinsic Sensitivity to Prediction Accuracy. In an effort to improve throughput prediction accuracy, several prior works have focused on designing more sophisticated throughput predictors such as C2SP [20], Fugu [46], and Xatu [50]. While these throughput predictors may offer higher prediction accuracy, they are complex and difficult to deploy, especially on compute or memory constrained devices [58]. In Section 4.2, we have showed that SODA is robust against prediction errors by design and does not require a sophisticated throughput predictor. We now demonstrate this empirically.

First, we replaced the throughput predictor used in simulations with a perfect short-term throughput predictor. Next, we gradually introduced more and more white noise to the perfect throughput predictions and observed how different ABR controllers behave accordingly. This experiment was conducted on a random subset of our network datasets with a size of 10,000 sessions. Note that throughput prediction discounts were turned off for all ABR controllers to reveal their *intrinsic* robustness.[1]

The results are shown in Figure 11, from which we observe that all hybrid ABR controllers that take throughput predictions into account will inevitably be affected by prediction errors to some extent (BOLA is not affected since it is purely buffer-based). Nonetheless, SODA still consistently outperforms all baseline ABR controllers up to a noise level of 50%. For reference, EMA predictor has an empirical noise level of about 30% on the same sessions. More importantly, the QoE degradation of SODA is minimal up to the reference point of EMA predictor, i.e., about 10%, which reinforces

[1]The ranking between different ABR controllers in this section may be different from that in Figure 10, which reveals that the robustness of certain ABR controllers should be attribute to throughput prediction discounts instead of intrinsic designs.

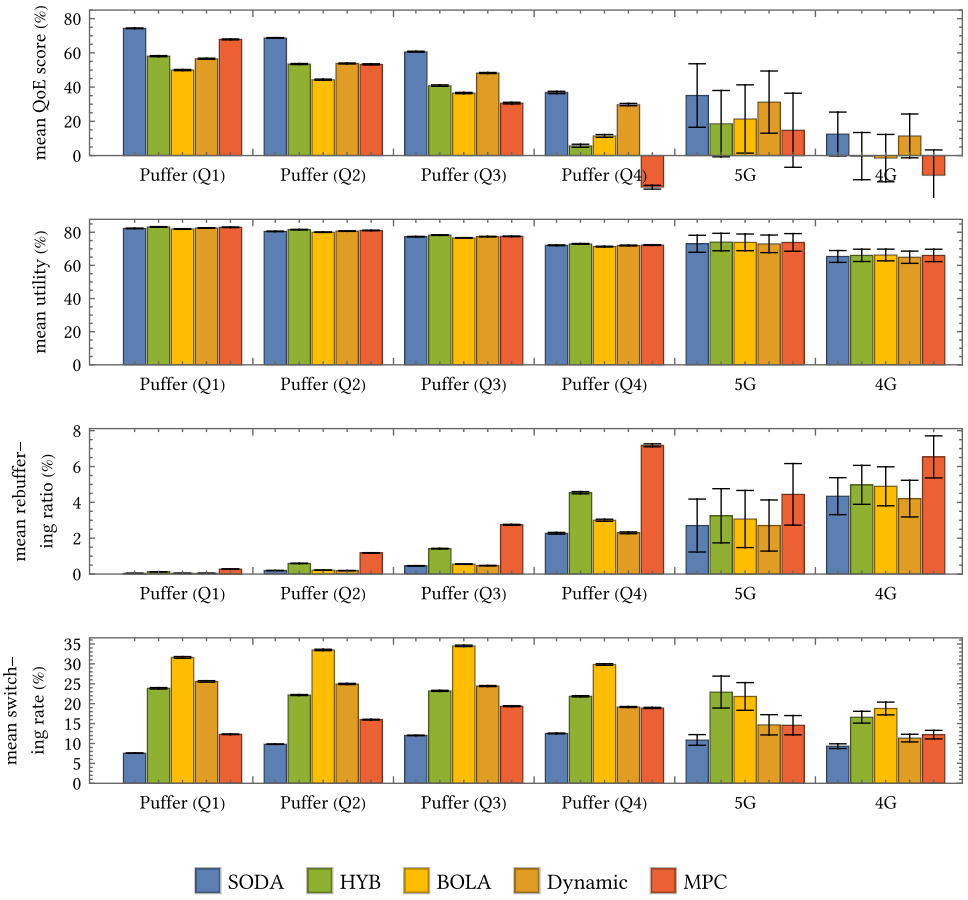

Figure 10: The mean QoE scores, utilities, rebuffering ratios and switching rates of SODA and baseline ABR controllers under each network dataset. The Puffer dataset is split into four quarters according the throughput variance (Q1 being lowest while Q4 being highest). SODA has consistently higher mean QoE scores and lower switching rates than all baseline ABR controllers under all network conditions. (Error bars represent 95% confidence intervals.)

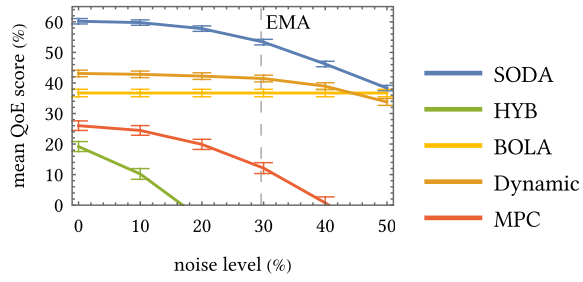

Figure 11: The mean QoE scores for SODA and baseline ABR controllers under variable amounts of white noise. (The error bars represent 95% confidence intervals.)

the idea that a practical deployment of SODA *does not* require a sophisticated throughput predictor.[2]

[2]In practice, we observe that EMA predictor is actually much better than a perfect short-term predictor with 30% white noise because the noise patterns are different, which means that real gap is less than 10%.

6.2 Prototype Evaluation

We next present emulation results from our local client-server deployment where we implemented SODA in the Puffer platform [46]. Thanks to Chrome DevTools' new capability to throttle WebSocket requests [59], we could replay our network datasets directly in Chrome using WebDriver [63]. The results are intended to highlight the robustness of different ABR controllers under actual browser-based playback. For these experiments, we allowed a maximum buffer length of 15 seconds, as set by Puffer.

6.2.1 Experimental Setup. The video source was a news clip encoded in five different resolutions (426 × 240, 640 × 360, 854 × 480, 1280 × 720, and 1920 × 1080) with a constant rate factor of 26 and a segment length of 2 seconds. To be fair to those learning-based ABR controllers trained specifically for the Puffer platform, we only considered the Puffer dataset. Since the average bitrate of the highest resolution is only about 2 Mb/s, we take a random subset of the Puffer dataset with a size of 1,000 sessions whose mean throughput is below 2 Mb/s to create challenging scenarios.

6.2.2 Baseline ABR Controllers. In response to the growing interest in learning-based throughput predictors and ABR controllers in the research community, we included two representative learning-based ABR controllers for local deployment on top of the major baseline ABR controllers from numerical simulations:

- **Fugu** [46]: Developed as part of the Puffer project, it features a learning-based stochastic throughput predictor, while its underlying control algorithm is similar to MPC.
- **CausalSimRL** [60]: A modern implementation of a reinforcement learning (RL)-based ABR controller Pensieve [22]. It is trained using CausalSim for the Puffer platform.

6.2.3 QoE Performance. Puffer employs structure similarity index measures (SSIM) [4] to quantify utility, thus to compare fairly using Puffer, we adapt mean utility to normalized mean SSIM, i.e., $\bar{v} = \overline{\text{SSIM}}/\text{SSIM}_{\max}$. The definitions of rebuffering ratio, switching rate, and QoE score remain the same. The aggregate statistics or QoE scores and individual QoE components across all sessions are shown in Figure 12. SODA outperforms the best baseline (Fugu) by 30.4% in terms of mean QoE score. More importantly, SODA is the only ABR controller that achieves low mean rebuffering ratio and switching rate *simultaneously*, which translates to superior smoothness of adaptive streaming. We highlight comparisons with the new baseline ABR controllers below:

- **SODA vs MPC & Fugu**. MPC and Fugu are grouped together since, apart from the more sophisticated stochastic throughput predictor, Fugu shares a similar underlying control algorithm with MPC. While they both achieve slightly higher mean utilities than SODA and reasonably low mean switching rates, these benefits are overshadowed by worse mean rebuffering ratios (230% and 104% worse respectively). Although Fugu partially mitigates the rebuffering issue due to its stochastic throughput predictor, it is still not robust enough for challenging network conditions.
- **SODA vs CausalSimRL**. CausalSimRL achieves slightly higher mean utility than SODA and a reasonably low mean rebuffering ratio. However, it switches bitrates 86.3% more often than SODA. Due to the black-box nature of RL-based ABR controllers, it is hard to reason why this is the case. In addition, there exists no straightforward way to tune an RL-based controller in favor of one particular QoE component without a complete retraining. In a production environment, it is highly desirable that the trade-off between different QoE components is tunable.

6.3 Production Deployment

We now describe the results from deploying SODA for live streams delivered on Amazon Prime Video. The bitrate ladder for these video streams had the following bitrate rungs {0.2, 0.45, 0.8, 1.2, 1.8, 2, 4, 5, 6.5, 8.0} Mb/s. This range of available bitrates fully exercised SODA's bitrate adaptation capability as well as tested its runtime feasibility on actual devices. The experiment was run on three device families, including (i) desktops/laptops (HTML5 browsers), (ii) smart TVs, and (iii) set-top boxes. On all three platforms, SODA used a simple sliding window-based throughput predictor. All devices were 20 seconds behind live action, so they could accumulate at most 20 seconds of buffer. To compare performance with a production tuned baseline, we conducted large-scale A/B experiments where customers were randomly assigned SODA or the production baseline controller. The experiment ran for more than 1 week with live streams delivered to more than 10 countries. In total, SODA sessions logged more than 50,000 streaming hours.

Figure 13 shows SODA's performance relative to the production deployed and tuned controller. First, notice that SODA consistently improves all the metrics across all device families, reducing the frequency of bitrate switching on set-top boxes by 88.8%. SODA really shines on HTML5 browsers where it reduced the mean rebuffering ratio by up to 53.0% in addition to 81.8% reduction in switching. This is because HTML5 browsers experience more volatility in network conditions compared to smart TVs and set-top boxes and thus present greater opportunity for improvement. Finally, notice that on all three platforms, the average duration of session increased, with 5.91% improvement on set-top boxes. Live streaming sessions for sports events routinely span multiple hours (e.g., 2-hour soccer broadcast, 3.5-hour cricket broadcast), so a 5.91% increase translates to more than 5 minutes duration.

Takeaways from Production Deployment. The production deployment shows that SODA is practical and can be widely deployed across different device types and network connections. Furthermore, to achieve its significant performance gains, it is sufficient for SODA to use simple sliding window-based throughput predictors.

7 RELATED WORK

7.1 Adaptive Bitrate Streaming

Bitrate adaptation has received significant attention from the multimedia research community. Buffer-based controllers like BBA [15] and BOLA [36, 44] make bitrate decisions based on buffer occupancy, while hybrid controllers like HYB [24], MPC [17] and DYNAMIC [44] combine throughput predictions with buffer occupancy to make decisions. SODA belongs to the latter category. There are also learning-based controllers such as Pensive [22] that utilize reinforcement learning to learn a bitrate selection strategy. Another relevant stream of work focuses on improving the accuracy of throughput predictions, including CS2P [20], Fugu [46], and Xatu [50]. Our work makes no assumption on the quality of throughput predictions and does not require a sophisticated throughput predictor. Past works have also considered upgrading the downloaded segments through replacement [36] which we do not consider in this paper.

7.2 Video Quality of Experience

The advent of video content delivery networks [2, 6] in the late 1990's led to efforts in industry to define and measure quality metrics for video delivery. Since then the quality of video delivery is a well studied topic with early work on the Akamai Stream Analyzer system which defined metrics such as startup time, rebuffer ratio, bitrate and failures etc and measured these metrics using data derived from video players deployed around the world [3, 66]. Subsequently, [7] showed that a 1% increase in rebuffering correlated with a 3-minute reduction in the amount of time users streamed live content. A study on YouTube [21] found that bitrate fluctuations strongly correlate with a user abandoning the session. Beyond correlations, the first study [9] to establish a causal relationship between video quality and user behavior used quasi-experimental

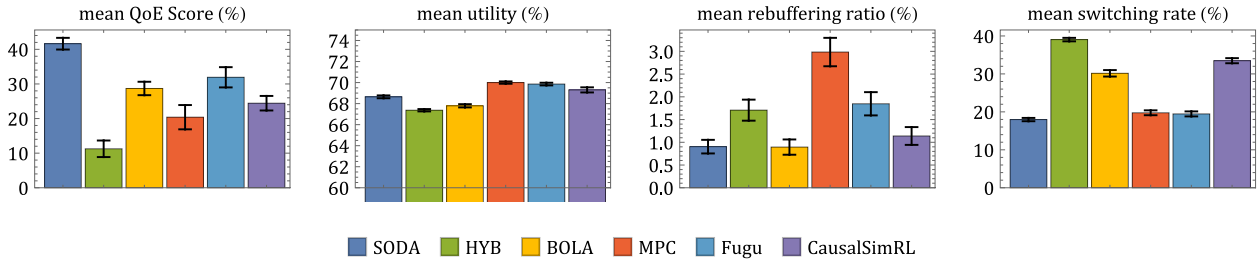

Figure 12: The mean QoE scores, utilities, rebuffering ratios, and switching rates from local deployment. SODA again has the highest mean QoE score and unlike all other baselines, *simultaneously* achieves ultra low mean rebuffering ratio and switching rate. (Error bars represent 95% confidence intervals.)

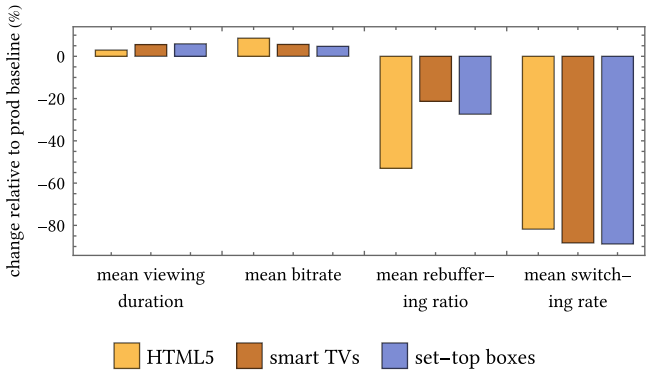

Figure 13: The change in mean viewing durations (higher is better), bitrates (higher is better), rebuffering ratios (lower is better), and switching rates (lower is better) of SODA compared to the production baseline.

designs (QEDs) to quantify the causal (adverse) impact of startup delay, rebuffering, and failures on user engagement, abandonment, and repeat viewership. A related work [11] built predictive models for user engagement based on QoE metrics. Our work leverages insights from these works in our ABR controller design.

7.3 Smoothed Online Convex Optimization

Our algorithm builds on recent developments in smoothed online convex optimization (SOCO), a variant of online optimization that penalizes switching between consecutive decisions via a "switching cost." [25, 33, 34]. In recent years, the design and analysis of algorithms for SOCO has received considerable attention, e.g., [14, 25, 31, 32, 34, 37], with optimal online algorithms emerging in various settings [33, 42, 48, 62] and a variety of applications receiving attention [10, 12, 26, 43, 53, 55, 57]. SOCO's switching cost model inspires our design of SODA for video streaming.

Our mathematical formulation of adaptive video streaming can be viewed as a specific example of online (optimal) control [54]. Similar to online optimization, online control seeks to design a controller to minimize the total cost incurred over a finite horizon. The theoretical bounds in this paper are most related to works that study how future predictions can improve online controller performance [39, 47, 49, 52]. Our proofs follow an analytic framework for studying MPC-based algorithms via exponentially decaying perturbation bounds [49, 55, 56]. Our work shows that this decay property holds under our model of adaptive video streaming, allowing us to establish performance guarantees for SODA.

8 LIMITATIONS AND FUTURE WORK

An emerging genre (but, still a small fraction) of live streaming is ultra-low latency live streams where the delay between the capture of an event and its display to the user is required to be of the order of a few seconds, as opposed to 10 to 20 seconds for the traditional live streams used in our current work. In future work, we would like to study if our SOCO-based strategy can be adapted for ultra-low latency live streams with buffer lengths in the order of a few seconds. The main challenge with ultra-small buffer sizes is that it is harder to prevent rebuffering and bitrate switching in this regime as the ABR controller needs to react to network fluctuations in a significantly shorter amount of time.

9 CONCLUSION

In this work, we propose a smoothness-optimized dynamic adaptive (SODA) controller that addresses this issue in a theoretically sound way. Thanks to SODA's robustness against prediction errors and low runtime complexity, it is readily deployable in a wide range of production environments. Through numerical simulations and prototype evaluation, we show that SODA consistently outperforms the state-of-the-art baselines. More importantly, we deployed SODA in a major video streaming provider where SODA significantly reduced bitrate switching by up to 88.8% compared to a fine-tuned production baseline. SODA's novel time-based ABR formulation and theoretical insights shed new light on how to achieve consistent high-quality video streaming.

ACKNOWLEDGMENTS

We thank the anonymous reviewers and our shepherd for their valuable feedback, as well as colleagues at Amazon Prime Video for their support with the production deployment of SODA. This work was funded by NSF under grants CAREER-204564, CCF-2325956, CNS-1763617, CNS-1901137, CNS-2102963, CNS-2106299, CNS-2106403, CNS-2106463, CNS-2146814, CPS-2136197, and NGSDI-2105648, as well as an Amazon Research Award. The research of Yiheng Lin was additionally supported by Amazon AI4Science Fellowship and PIMCO Graduate Fellowship in Data Science.

REFERENCES

[1] Mun Choon Chan and Ramachandran Ramjee. 2002. TCP/IP Performance over 3G Wireless Links with Rate and Delay Variation. In *Proceedings of the 8th Annual International Conference on Mobile Computing and Networking* (MobiCom '02). Atlanta, Georgia, USA, 71–82. ISBN: 158113486X. DOI: 10.1145/570645.570655.

[2] John Dilley, Bruce M. Maggs, Jay Parikh, Harald Prokop, Ramesh K. Sitaraman, and William E. Weihl. 2002. Globally Distributed Content Delivery. *IEEE Internet Computing*, 6, 5, 50–58.

[3] R.K. Sitaraman and R.W. Barton. 2003. Method and apparatus for measuring stream availability, quality and performance. US Patent 7,010,598. (Feb. 2003).

[4] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. 2004. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13, (Apr. 2004), 600–612, 4, (Apr. 2004). DOI: 10.1109/TIP.2003.819861.

[5] Junxian Huang, Qiang Xu, Birjodh Tiwana, Z. Morley Mao, Ming Zhang, and Paramvir Bahl. 2010. Anatomizing Application Performance Differences on Smartphones. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services* (MobiSys '10). San Francisco, California, USA, 165–178. ISBN: 9781605589855. DOI: 10.1145/1814433.1814452.

[6] E. Nygren, Ramesh K. Sitaraman, and J. Sun. 2010. The Akamai Network: A platform for high-performance Internet applications. *ACM SIGOPS Operating Systems Review*, 44, 3, 2–19.

[7] Florin Dobrian, Vyas Sekar, Asad Awan, Ion Stoica, Dilip Joseph, Aditya Ganjam, Jibin Zhan, and Hui Zhang. 2011. Understanding the Impact of Video Quality on User Engagement. In *Proceedings of the ACM SIGCOMM 2011 Conference* (SIGCOMM '11). Toronto, Ontario, Canada, 362–373. ISBN: 9781450307970. DOI: 10.1145/2018436.2018478.

[8] Te-Yuan Huang, Nikhil Handigol, Brandon Heller, Nick McKeown, and Ramesh Johari. 2012. Confused, timid, and unstable. In *Proceedings of the 2012 Internet Measurement Conference*. ACM, New York, NY, USA, (Nov. 2012), 225–238. ISBN: 9781450317054. DOI: 10.1145/2398776.2398800.

[9] S. Shunmuga Krishnan and Ramesh K. Sitaraman. 2012. Video Stream Quality Impacts Viewer Behavior: Inferring Causality Using Quasi-Experimental Designs. In *Proceedings of the 2012 Internet Measurement Conference* (IMC '12). Boston, Massachusetts, USA, 211–224. ISBN: 9781450317054. DOI: 10.1145/2398776.2398799.

[10] Minghong Lin, Zhenhua Liu, Adam Wierman, and Lachlan LH Andrew. 2012. Online algorithms for geographical load balancing. In *Proceedings of the International Green Computing Conference (IGCC)*, 1–10.

[11] Athula Balachandran, Vyas Sekar, Aditya Akella, Srinivasan Seshan, Ion Stoica, and Hui Zhang. 2013. Developing a predictive model of quality of experience for internet video. *SIGCOMM Comput. Commun. Rev.*, 43, 4, (Aug. 2013), 339–350. DOI: 10.1145/2534169.2486025.

[12] Minghong Lin, Adam Wierman, Lachlan L. H. Andrew, and Eno Thereska. 2013. Dynamic Right-Sizing for Power-Proportional Data Centers. *IEEE/ACM Transactions on Networking*, 21, 5, (Oct. 2013), 1378–1391. DOI: 10.1109/TNET.2012.2226216.

[13] Masoud Badiei, Na Li, and Adam Wierman. 2015. Online convex optimization with ramp constraints. In *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE, 6730–6736.

[14] Nikhil Bansal, Anupam Gupta, Ravishankar Krishnaswamy, Kirk Pruhs, Kevin Schewior, and Cliff Stein. 2015. A 2-Competitive Algorithm For Online Convex Optimization With Switching Costs. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2015)* (Leibniz International Proceedings in Informatics (LIPIcs)). Naveen Garg, Klaus Jansen, Anup Rao, and José D. P. Rolim, (Eds.) Vol. 40. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 96–109. DOI: 10.4230/LIPIcs.APPROX-RANDOM.2015.96.

[15] Te-Yuan Huang, Ramesh Johari, Nick McKeown, Matthew Trunnell, and Mark Watson. 2015. A Buffer-Based Approach to Rate Adaptation: Evidence from a Large Video Streaming Service. *ACM SIGCOMM Computer Communication Review*, 44, (Feb. 2015), 187–198, 4, (Feb. 2015). DOI: 10.1145/2740070.2626296.

[16] Yan Liu and Jack Y. B. Lee. 2015. An Empirical Study of Throughput Prediction in Mobile Data Networks. In *2015 IEEE Global Communications Conference (GLOBECOM)*, 1–6. DOI: 10.1109/GLOCOM.2015.7417858.

[17] Xiaoqi Yin, Abhishek Jindal, Vyas Sekar, and Bruno Sinopoli. 2015. A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*. ACM, New York, NY, USA, (Aug. 2015), 325–338. ISBN: 9781450335423. DOI: 10.1145/2785956.2787486.

[18] Mojgan Ghasemi, Partha Kanuparthy, Ahmed Mansy, Theophilus Benson, and Jennifer Rexford. 2016. Performance Characterization of a Commercial Video Streaming Service. In *Proceedings of the 2016 Internet Measurement Conference*. ACM, New York, NY, USA, (Nov. 2016), 499–511. ISBN: 9781450345262. DOI: 10.1145/2987443.2987481.

[19] he1enh. 2016. Reproducing Network Research. (May 2016). https://reproducingnetworkresearch.wordpress.com/2016/05/30/cs244-16-failed-experiments-with-fastmpc-integrating-rate-based-adaptive-streaming-into-vlc/.

[20] Yi Sun, Xiaoqi Yin, Junchen Jiang, Vyas Sekar, Fuyuan Lin, Nanshu Wang, Tao Liu, and Bruno Sinopoli. 2016. CS2P: Improving Video Bitrate Selection and Adaptation with Data-Driven Throughput Prediction. In *Proceedings of the 2016 ACM SIGCOMM Conference*. ACM, New York, NY, USA, (Aug. 2016), 272–285. ISBN: 9781450341936. DOI: 10.1145/2934872.2934898.

[21] Christos George Bampis, Zhi Li, Anush Krishna Moorthy, Ioannis Katsavounidis, Anne Aaron, and Alan Conrad Bovik. 2017. Study of Temporal Effects on Subjective Video Quality of Experience. *IEEE Transactions on Image Processing*, 26, 11, 5217–5231. DOI: 10.1109/TIP.2017.2729891.

[22] Hongzi Mao, Ravi Netravali, and Mohammad Alizadeh. 2017. Neural Adaptive Video Streaming with Pensieve. In ACM, (Aug. 2017), 197–210. ISBN: 9781450346535. DOI: 10.1145/3098822.3098843.

[23] Yanyuan Qin, Ruofan Jin, Shuai Hao, Krishna R. Pattipati, Feng Qian, Subhabrata Sen, Bing Wang, and Chaoqun Yue. 2017. A control theoretic approach to ABR video streaming: A fresh look at PID-based rate adaptation. In *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*. IEEE, (May 2017), 1–9. ISBN: 978-1-5090-5336-0. DOI: 10.1109/INFOCOM.2017.8057056.

[24] Zahaib Akhtar, Yun Seong Nam, Ramesh Govindan, Sanjay Rao, Jessica Chen, Ethan Katz-Bassett, Bruno Ribeiro, Jibin Zhan, and Hui Zhang. 2018. Oboe: Auto-Tuning Video ABR Algorithms to Network Conditions. In ACM, (Aug. 2018), 44–58. ISBN: 9781450355674. DOI: 10.1145/3230543.3230558.

[25] Niangjun Chen, Gautam Goel, and Adam Wierman. 2018. Smoothed Online Convex Optimization in High Dimensions via Online Balanced Descent. In *Proceedings of Conference On Learning Theory (COLT)*, 1574–1594.

[26] Yingying Li, Guannan Qu, and Na Li. 2018. Online Optimization with Predictions and Switching Costs: Fast Algorithms and the Fundamental Limit. (2018). arXiv: 1801.07780v3 [math.OC].

[27] Darijo Raca, Jason J. Quinlan, Ahmed H. Zahran, and Cormac J. Sreenan. 2018. Beyond Throughput: A 4G LTE Dataset with Channel and Context Metrics. In *Proceedings of the 9th ACM Multimedia Systems Conference*. ACM, New York, NY, USA, (June 2018), 460–465. ISBN: 9781450351928. DOI: 10.1145/3204949.3208123.

[28] Naman Agarwal, Brian Bullins, Elad Hazan, Sham Kakade, and Karan Singh. 2019. Online control with adversarial disturbances. In *International Conference on Machine Learning*. PMLR, 111–119.

[29] Zahaib Akhtar, Yaguang Li, Ramesh Govindan, Emir Halepovic, Shuai Hao, Yan Liu, and Subhabrata Sen. 2019. AViC: A Cache for Adaptive Bitrate Video. In *Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies* (CoNEXT '19). Association for Computing Machinery, Orlando, Florida, 305–317. ISBN: 9781450369985. DOI: 10.1145/3359989.3365423.

[30] Aaron D Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. 2019. Control barrier functions: theory and applications. In *2019 18th European control conference (ECC)*. IEEE, 3420–3431.

[31] C.J. Argue, Sébastien Bubeck, Michael B. Cohen, Anupam Gupta, and Yin Tat Lee. 2019. A Nearly-Linear Bound for Chasing Nested Convex Bodies. In *Proceedings of the 2019 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. Proceedings. Society for Industrial and Applied Mathematics, (Jan. 2019), 117–122. DOI: 10.1137/1.9781611975482.8.

[32] Sébastien Bubeck, Bo'az Klartag, Yin Tat Lee, Yuanzhi Li, and Mark Sellke. 2019. Chasing Nested Convex Bodies Nearly Optimally. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms (SODA)*. Proceedings. Society for Industrial and Applied Mathematics, (Dec. 2019), 1496–1508. DOI: 10.1137/1.9781611975994.91.

[33] Gautam Goel, Yiheng Lin, Haoyuan Sun, and Adam Wierman. 2019. Beyond online balanced descent: An optimal algorithm for smoothed online optimization. *Advances in Neural Information Processing Systems*, 32.

[34] Gautam Goel and Adam Wierman. 2019. An Online Algorithm for Smoothed Regression and LQR Control. In *Proceedings of the Machine Learning Research*. Vol. 89, 2504–2513. http://proceedings.mlr.press/v89/goel19a.html.

[35] Ming Shi, Xiaojun Lin, and Lei Jiao. 2019. On the value of look-ahead in competitive online convex optimization. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 3, 2, 22.

[36] Kevin Spiteri, Ramesh Sitaraman, and Daniel Sparacio. 2019. From Theory to Practice: Improving Bitrate Adaptation in the DASH Reference Player. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 15, 2s, (Apr. 2019), 1–29. DOI: 10.1145/3336497.

[37] C. J. Argue, Anupam Gupta, and Guru Guruganesh. 2020. Dimension-Free Bounds for Chasing Convex Functions. In *Proceedings of Thirty Third Conference on Learning Theory*. PMLR, (July 2020), 219–241. Retrieved Feb. 4, 2022 from.

[38] Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. 2020. On the sample complexity of the linear quadratic regulator. *Foundations of Computational Mathematics*, 20, 4, 633–679.

[39] Yingying Li, Guannan Qu, and Na Li. 2020. Online optimization with predictions and switching costs: Fast algorithms and the fundamental limit. *IEEE Transactions on Automatic Control*, 66, 10, 4761–4768.

[40] Emily Marx, Francis Y. Yan, and Keith Winstein. 2020. Implementing BOLA-BASIC on Puffer: Lessons for the use of SSIM in ABR logic, (Nov. 2020).

[41] Darijo Raca, Dylan Leahy, Cormac J. Sreenan, and Jason J. Quinlan. 2020. Beyond Throughput, The Next Generation: A 5G Dataset with Channel and Context Metrics. In *Proceedings of the 11th ACM Multimedia Systems Conference*. ACM, New York, NY, USA, (May 2020), 303–308. ISBN: 9781450368452. DOI: 10.1145/3339825.3394938.

[42] Mark Sellke. 2020. Chasing convex bodies optimally. In *Proceedings of the Thirty-First Annual ACM-SIAM Symposium on Discrete Algorithms* (SODA '20). Society for Industrial and Applied Mathematics, USA, (Jan. 2020), 1509–1518. Retrieved Oct. 15, 2021 from.

[43] Guanya Shi, Yiheng Lin, Soon-Jo Chung, Yisong Yue, and Adam Wierman. 2020. Online optimization with memory and competitive control. *Advances in Neural Information Processing Systems*, 33, 20636–20647.

[44] Kevin Spiteri, Rahul Urgaonkar, and Ramesh K. Sitaraman. 2020. BOLA: Near-Optimal Bitrate Adaptation for Online Videos. *IEEE/ACM Transactions on Networking*, 28, 4, (Aug. 2020), 1698–1711. DOI: 10.1109/TNET.2020.2996964.

[45] Dongzhu Xu, Anfu Zhou, Xinyu Zhang, Guixian Wang, Xi Liu, Congkai An, Yiming Shi, Liang Liu, and Huadong Ma. 2020. Understanding Operational 5G: A First Measurement Study on Its Coverage, Performance and Energy Consumption. In (SIGCOMM '20). Virtual Event, USA, 479–494. ISBN: 9781450379557. DOI: 10.1145/3387514.3405882.

[46] F.Y. Yan, H. Ayers, C. Zhu, S. Fouladi, J. Hong, K. Zhang, P. Levis, and K. Winstein. 2020. Learning in Situ: A Randomized Experiment in Video Streaming. In *Proceedings of the 17th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2020*, 495–511. ISBN: 9781939133137. https://www.usenix.org/conference/nsdi20/presentation/yan.

[47] Chenkai Yu, Guanya Shi, Soon-Jo Chung, Yisong Yue, and Adam Wierman. 2020. The power of predictions in online control. *Advances in Neural Information Processing Systems*, 33, 1994–2004.

[48] C. J. Argue, Anupam Gupta, Ziye Tang, and Guru Guruganesh. 2021. Chasing Convex Bodies with Linear Competitive Ratio. *Journal of the ACM*, 68, 5, 1–10. DOI: 10.1145/3450349.

[49] Yiheng Lin, Yang Hu, Haoyuan Sun, Guanya Shi, Guannan Qu, and Adam Wierman. 2021. Perturbation-based Regret Analysis of Predictive Control in Linear Time Varying Systems. *arXiv preprint arXiv:2106.10497*.

[50] Yun Seong Nam, Jianfei Gao, Chandan Bothra, Ehab Ghabashneh, Sanjay Rao, Bruno Ribeiro, Jibin Zhan, and Hui Zhang. 2021. Xatu: Richer Neural Network Based Prediction for Video Streaming. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 5, 3, (Dec. 2021), 1–26. DOI: 10.1145/3491056.

[51] Sungho Shin and Victor M. Zavala. 2021. Controllability and Observability Imply Exponential Decay of Sensitivity in Dynamic Optimization. *arXiv preprint arXiv:2101.06350*.

[52] Runyu Zhang, Yingying Li, and Na Li. 2021. On the regret analysis of online LQR control with predictions. In *2021 American Control Conference (ACC)*. IEEE, 697–703.

[53] Nicolas Christianson, Christopher Yeh, Tongxin Li, Mahdi Torabi Rad, Azarang Golmohammadi, and Adam Wierman. 2022. Robustifying machine-learned algorithms for efficient grid operation. In *NeurIPS 2022 Workshop on Tackling Climate Change with Machine Learning*. https://www.climatechange.ai/papers/neurips2022/19.

[54] Elad Hazan and Karan Singh. 2022. Introduction to online nonstochastic control. *arXiv preprint arXiv:2211.09619*.

[55] Yiheng Lin, Judy Gan, Guannan Qu, Yash Kanoria, and Adam Wierman. 2022. Decentralized Online Convex Optimization in Networked Systems. In *International Conference on Machine Learning*. PMLR, 13356–13393.

[56] Yiheng Lin, Yang Hu, Guannan Qu, Tongxin Li, and Adam Wierman. 2022. Bounded-Regret MPC via Perturbation Analysis: Prediction Error, Constraints, and Nonlinearity. *arXiv preprint arXiv:2210.12312*.

[57] Weici Pan, Guanya Shi, Yiheng Lin, and Adam Wierman. 2022. Online optimization with feedback delay and nonlinear switching cost. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 6, 1, 1–34.

[58] Talha Waheed, Ihsan Ayyub Qazi, Zahaib Akhtar, and Zafar Ayyub Qazi. 2022. Coal Not Diamonds: How Memory Pressure Falters Mobile Video QoE. In (CoNEXT '22). Roma, Italy, 307–320. ISBN: 9781450395083. DOI: 10.1145/3555050.3569120.

[59] Jecelyn Yeen. 2022. What's New In DevTools (Chrome 99). (Feb. 2022). https://developer.chrome.com/en/blog/new-in-devtools-99/.

[60] A. Alomar, P. Hamadanian, A. Nasr-Esfahany, A. Agarwal, M. Alizadeh, and D. Shah. 2023. CausalSim: A Causal Framework for Unbiased Trace-Driven Simulation. In 1115–1147. ISBN: 9781939133335. https://www.usenix.org/conference/nsdi23/presentation/alomar.

[61] Chandan Bothra, Jianfei Gao, Sanjay Rao, and Bruno Ribeiro. 2023. Veritas: Answering Causal Queries from Video Streaming Traces. In *Proceedings of the ACM SIGCOMM 2023 Conference* (ACM SIGCOMM '23). New York, NY, USA, 738–753. DOI: 10.1145/3603269.3604828.

[62] Nicolas Christianson, Junxuan Shen, and Adam Wierman. 2023. Optimal Robustness-Consistency Tradeoffs for Learning-Augmented Metrical Task Systems. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*. PMLR, (Apr. 2023), 9377–9399.

[63] MDN contributors. 2023. WebDriver. (June 2023). https://developer.mozilla.org/en-US/docs/Web/WebDriver.

[64] DASH Industry Forum. 2023. dash.js: A Reference Client Implementation for the Playback of MPEG DASH via JavaScript and Compliant Browsers. https://github.com/Dash-Industry-Forum/dash.js.

[65] YouTube. 2023. YouTube Recommended Upload Encoding Settings. https://support.google.com/youtube/answer/1722171.

[66] Akamai. [n. d.] Stream Analyzer Service Description. https://groups.cs.umass.edu/ramesh/wp-content/uploads/sites/3/2023/10/Stream_Analyzer_Service_Description.pdf. ().

Appendices are supporting material that has not been peer-reviewed.

A PROOF OUTLINE

In this section, we present an outline of our theoretical analysis for SODA. As we discussed in Section 4, our proof is based on an exponentially decaying perturbation bound that relates the behavior of the solution to the optimization problem defining SODA as a function of problem parameters. This section is organized as follows: We first introduce the modeling of SODA that we use to establish theoretical results in Section A.1. Then, we introduce the exponentially decaying perturbation bound, its implications, and the proof idea in Section A.2. Next, we present the outlines for proving SODA's performance guarantees with the help of exponentially decaying perturbation bounds in Sections A.3 and A.4. Finally, we will discuss some sufficient conditions under which the optimal bitrate sequence can be approximated by a monotonic sequence in Section A.5.

A.1 Theoretical Problem Setting

We first introduce the notation used to define the performance metrics and the variant of SODA studied in our theoretical analysis. To make the formulation of the video streaming problem closer to a classic control problem, we define the "control action" u_t as the inverse of the bitrate (i.e., $u_t = \frac{1}{r_t}$). Recall that we set $v(r) = \frac{1}{r}$ in our theoretical analysis. Thus, we can write down a general form of the optimization problem solved by SODA and use $\psi_t^{t+p}\left((\sigma_{t-1}, v_{t-1}); \hat{\omega}_{t:t+p}; F\right)$ to denote its optimal solution:

$$\underset{x_{t:t+p}, u_{t+1:t+p}}{\arg\min} \sum_{\tau=t}^{t+p} \hat{\omega}_\tau u_\tau^2 + \beta \sum_{\tau=t}^{t+p} b(x_\tau) + \gamma \sum_{\tau=t}^{t+p+1} |u_\tau - u_{\tau-1}|^2 + F(x_{t+p}, u_{t+p+1}) \tag{3a}$$

$$\text{s.t. } x_\tau = x_{\tau-1} + \hat{\omega}_\tau u_\tau - 1, \text{ for } \tau = t, \ldots, t+p, \tag{3b}$$

$$0 \le x_\tau \le x_{\max}, \frac{1}{r_{\max}} \le u_\tau \le \frac{1}{r_{\min}}, \text{ for } \tau = t, \ldots, t+p, \tag{3c}$$

$$x_{t-1} = \sigma_{t-1}, u_{t-1} = v_{t-1}. \tag{3d}$$

Here, $\psi_t^{t+p}\left((\sigma_{t-1}, v_{t-1}); \hat{\omega}_{t:t+p}; F\right)$ is defined to be a vector that contains the states $x_{t:t+p}$ and control actions $u_{t+1:t+p}$ in the optimal solution. The initial condition (σ_{t-1}, v_{t-1}), bandwidth sequence $\hat{\omega}_{t:t+p}$, and terminal cost function F are the parameters of the optimization problem. For the terminal costs, we consider two types of functions: (1) The zero function $F = \mathbf{0}$, i.e., $F(x, u) = 0$ for all x, u; (2) The indicator function $F = \mathbb{I}_{\sigma,v}$, which is defined as

$$F(x,u) = \mathbb{I}_{\sigma,v}(x,u) = \begin{cases} 0 & \text{if } x = \sigma, u = v, \\ +\infty & \text{otherwise.} \end{cases}$$

The first type of terminal cost will be used to define the performance metrics (competitive ratio and dynamic regret), and the second type will be used in the algorithm design. Since we will use the indicator terminal cost frequently, we introduce the shorthand $\tilde{\psi}_t^{t+p}\left((\sigma_{t-1}, v_{t-1}); \hat{\omega}_{t:t+p}; (\sigma_{t+p}, v_{t+p+1})\right)$, which denotes $\tilde{\psi}_t^{t+p}\left((\sigma_{t-1}, v_{t-1}); \hat{\omega}_{t:t+p}; \mathbb{I}_{\sigma_{t+p}, v_{t+p+1}}\right)$. We use $\iota_t^{t+p}\left((\sigma_{t-1}, v_{t-1}); \hat{\omega}_{t:t+p}; F\right)$ to denote the optimal objective value of the optimization problem (3).

The model of SODA that we consider in the theoretical analysis is summarized in Algorithm 2. The major difference from the SODA algorithm discussed in Section 3.3 is that we include the indicator terminal cost (in line 5) so that the last two states in the predictive trajectory are equal to the target buffer level. This terminal constraint is important for our competitive ratio result in Theorem 4.1, for which we need to bound the squared distance between the trajectories of SODA and the offline optimal controller by a part of the offline optimal cost.

Algorithm 2: SODA (for theoretical analysis)

Require: Prediction horizon K.
1: **for** $t = 1, 2, \ldots, N$ **do**
2: Set $t' = \min\{t + K - 1, N\}$.
3: Receive predictions $\hat{\omega}_{t+1:t'|t}$.
4: **if** $t' < N$ **then**
5: Set terminal cost $F_{t'} = \mathbb{I}_{x^*, 1/\hat{\omega}_{t'|t}}$.
6: **else**
7: Set terminal cost $F_{t'} = \mathbf{0}$.
8: **end if**
9: Commit $u_t = \psi_t^{t'-1}\left((x_{t-1}, u_{t-1}); \hat{\omega}_{t:t'|t}; F_{t'}\right)$.
10: **end for**

Using the notations above, we can formally define the performance metrics we employ: Let cost(OPT) denote the offline optimal cost one can achieve when exact predictions of all future bandwidth are available at the start of the problem, i.e., $\text{cost}(\texttt{OPT}) = \iota_1^N\left((x_0, u_0); \omega^*_{1:N}; \mathbf{0}\right)$. Then,

- *Dynamic regret* is an upper bound on the difference cost(SODA) − cost(OPT);
- *Competitive ratio* is an upper bound on the ratio cost(SODA)/cost(OPT).

A.2 Exponentially Decaying Perturbations

Exponentially decaying perturbations is a critical property of the finite-time optimal control problem that our analysis builds upon. We define this property formally in Definition A.1.

Definition A.1 (Exponentially Decaying Perturbation Bound)**.** *We say the exponentially decaying perturbation bound holds if there exists uniform constants $C > 0, \rho \in (0,1)$ such that the following inequalities hold:*

$$\begin{aligned}
&\left|\psi_t^{t+p}\left((\sigma_{t-1}, v_{t-1}); \hat{\omega}_{t:t+p}; \mathbf{0}\right)_{x_\tau} - \psi_t^{t+p}\left((\sigma'_{t-1}, v'_{t-1}); \hat{\omega}'_{t:t+p}; \mathbf{0}\right)_{x_\tau}\right| \\
&\le C\rho^{\tau-t+1}\left(\left|\sigma_{t-1} - \sigma'_{t-1}\right| + \left|v_{t-1} - v'_{t-1}\right|\right) + C\sum_{j=t}^{t+p}\rho^{|\tau-j|}\left|\hat{\omega}_j - \hat{\omega}'_j\right|, \qquad (4)\\
&\left|\tilde{\psi}_t^{t+p}\left((\sigma_{t-1}, v_{t-1}); \hat{\omega}_{t:t+p}; (\sigma_{t+p}, v_{t+p+1})\right)_{x_\tau} - \psi_t^{t+p}\left((\sigma'_{t-1}, v'_{t-1}); \hat{\omega}'_{t:t+p}; (\sigma'_{t+p}, v'_{t+p+1})\right)_{x_\tau}\right| \\
&\le C\rho^{\tau-t+1}\left(\left|\sigma_{t-1} - \sigma'_{t-1}\right| + \left|v_{t-1} - v'_{t-1}\right|\right) + C\sum_{j=t}^{t+p}\rho^{|\tau-j|}\left|\hat{\omega}_j - \hat{\omega}'_j\right| + C\rho^{t+p-\tau}\left(\left|\sigma_{t+p} - \sigma'_{t+p}\right| + \left|v_{t+p+1} - v'_{t+p+1}\right|\right). \qquad (5)
\end{aligned}$$

Intuitively, the exponential decay property (Definition A.1) holds if the impact of a perturbation on the initial condition (σ_{t-1}, v_{t-1}), prediction $\hat{\omega}_j$, or terminal constraint $(\sigma_{t+p}, v_{t+p+1})$ on the component x_τ in the optimal trajectory decays exponentially with respect to the absolute difference between their corresponding time indices.

Due to its importance for the theoretical analysis of MPC-based algorithms, many previous works have established exponentially decaying perturbation bounds for various cases of online optimization with switching costs [49], optimal control with unconstrained dynamics [49, 56], and online optimization in networked systems [55]. In contrast to previous work, however, the video streaming problem (3) that we consider is a constrained optimal control problem. To this point, there has been limited success in establishing exponentially decaying perturbation bounds for general constrained optimal control problems, and existing results that provide sufficient conditions for their validity are difficult to verify [51, 56].

In this work, we leverage the special structure of the video streaming problem to show the exponentially decaying perturbation bound holds in this setting. We require the following assumption about the buffer constraints, bandwidth, and the bitrate range.

Assumption A.1. *There exists uniform constants $\omega_{max} > \omega_{min} > 0$ such that for any time step t, we have that $\omega_{min} \le \omega_t \le \omega_{max}$ holds. We also assume that $\omega_{min}/r_{min} \ge x_{max}$, and $\omega_{max}/r_{max} - 1 \le -\delta$ holds for a fixed constant $\delta > 0$.*

Intuitively, Assumption A.1 guarantees that the controller can always fill up the buffer at the cost of choosing the smallest bitrate or decrease the buffer level by choosing the largest bitrate. As we discussed in Section 4, this assumption is used to eliminate extreme boundary cases in the analysis, but SODA empirically performs well even when Assumption A.1 is not strictly satisfied. Using this assumption, we show the exponentially decaying perturbation property holds for the video streaming problem in Theorem A.1.

Theorem A.1. *Under Assumption A.1, the exponentially decaying perturbation bound holds with constants*

$$\rho = \left(1 - \frac{2}{1 + \sqrt{1 + \frac{\max\{6\omega_{min}(\omega_{min}+3), 4x_{max}(\omega_{min}+8\gamma)\}}{\omega_{min}^3 \epsilon\beta}}}\right)^{\frac{1}{3(3+\lceil x_{max}/\delta\rceil)}}$$

and

$$C = \frac{(1+\omega_{max})\left(3\beta\omega_{min}^3 + \max\{6\omega_{min}(\omega_{min}+3), 4x_{max}(\omega_{min}+8\gamma)\}\right)}{\omega_{min}^3 \rho^{3+\lceil x_{max}/\delta\rceil}}.$$

While the exponentially decaying property (Definition A.1) bounds the impact of parameter perturbations on the states, we extend the definition to the control actions and show that this variant holds as a corollary of Theorem A.1.

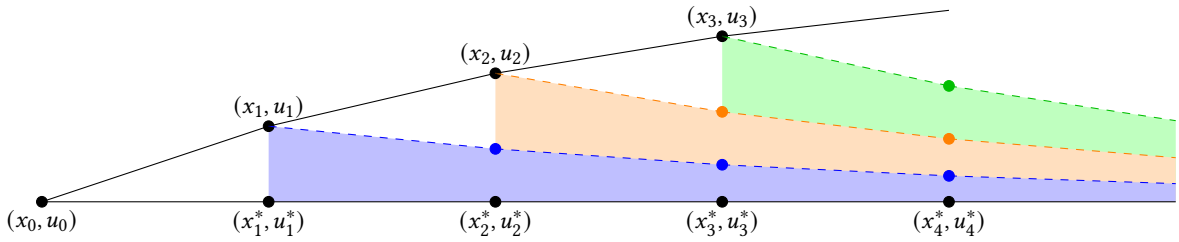

Figure 14: Illustration of the aggregations of per-step errors. In the figure, $\{(x_t^*, u_t^*)\}_{t=1,2,\dots}$ denotes the offline optimal states and control actions, and $\{(x_t, u_t)\}_{t=1,2,\dots}$ denotes the buffer level achieved by SODA. The dashed trajectory from (x_t, u_t) denotes the clairvoyant optimal trajectory from (x_t, u_t). At time t, the per-step error e_t leads to the deviation of the actual trajectory of SODA with the clairvoyant optimal trajectory. The impact of the per-step error e_1 at a future time step t is the height of blue area, which decays exponentially fast with respect to t when exponentially decaying perturbation holds. Therefore, although a per-step error occurs at every time step, the distance between (x_t, u_t) and (x_t^*, u_t^*) is still uniformly bounded.

Corollary A.2. *Under Assumption A.1, for the control action u, we also have that*

$$\begin{aligned}
&\left|\psi_t^{t+p}\left((\sigma_{t-1}, v_{t-1}); \hat{\omega}_{t:t+p}; \mathbf{0}\right)_{u_\tau} - \psi_t^{t+p}\left((\sigma'_{t-1}, v'_{t-1}); \hat{\omega}'_{t:t+p}; \mathbf{0}\right)_{u_\tau}\right| \\
&\le C'\rho^{\tau-t+1}\left(\left|\sigma_{t-1} - \sigma'_{t-1}\right| + \left|v_{t-1} - v'_{t-1}\right|\right) + C'\sum_{j=t}^{t+p}\rho^{|\tau-j|}\left|\hat{\omega}_j - \hat{\omega}'_j\right|, \\
&\left|\tilde{\psi}_t^{t+p}\left((\sigma_{t-1}, v_{t-1}); \hat{\omega}_{t:t+p}; (\sigma_{t+p}, v_{t+p+1})\right)_{u_\tau} - \psi_t^{t+p}\left((\sigma'_{t-1}, v'_{t-1}); \hat{\omega}'_{t:t+p}; (\sigma'_{t+p}, v'_{t+p+1})\right)_{u_\tau}\right| \\
&\le C'\rho^{\tau-t+1}\left(\left|\sigma_{t-1} - \sigma'_{t-1}\right| + \left|v_{t-1} - v'_{t-1}\right|\right) + C'\sum_{j=t}^{t+p}\rho^{|\tau-j|}\left|\hat{\omega}_j - \hat{\omega}'_j\right| + C'\rho^{t+p-\tau}\left(\left|\sigma_{t+p} - \sigma'_{t+p}\right| + \left|v_{t+p+1} - v'_{t+p+1}\right|\right),
\end{aligned}$$

where the decay factor ρ is the same as Theorem A.1, and the constant C' is given by

$$C' = \frac{C(1+\rho)r_{min} + \rho}{\omega_{min} r_{min} \rho}.$$

Here, C is the same as Theorem A.1.

To establish the exponentially decaying perturbation property, we first reduce the video streaming problem to a more general online optimization problem with memory and inequality constraints. Then, we consider each possible combination of active inequality constraints separately and show that the exponentially decaying perturbation property holds in each case. This only requires considering optimization problems with equality constraints with second-order differentiable objectives. Lastly, we show that the exponential decay properties for these separate cases can be combined to establish the exponential decay property for the original video streaming problem.

A.3 Proof Outline for Exact Predictions

We provide the formal version of Theorem 4.1 that gives the dynamic regret and competitive ratio for SODA with specific coefficients in Theorem A.3.

Theorem A.3. *Under Assumption A.1, consider* SODA *with the terminal constraints $x_{t+K-1} = \bar{x}, r_{t+K-1} = \hat{\omega}_{t+K-1|t-1}$. Define the weight C and the decay factor ρ to be the same as Theorem A.1, and the coefficient C' is given by Corollary A.2. Suppose all predictions are exact (i.e., $\hat{\omega}_{m|n-1} = \omega_m$ for $m = n, \dots, n+K-1$) and the prediction horizon K satisfies*

$$K \ge \frac{1}{4}\ln\left(\frac{16}{1-\rho}\cdot\left(1 + \frac{(C+C')^2}{1-\rho}\right)\cdot\left(C^2 + (C')^2\right)^2\right)/\ln\left(\frac{1}{\rho}\right) = O(1).$$

Here, the coefficients C, C' and the decay factor ρ are given by Theorem A.1 and Corollary A.2. Then, SODA achieves a dynamic regret of $C_1\rho^{K-1}cost(\texttt{OPT}) = O(\rho^K N)$ and a competitive ratio of $1 + C_1\rho^{K-1} = 1 + O(\rho^K)$. Here, the coefficient C_1 is given by

$$C_1 = 8\left(2(4\gamma + \beta + \omega_{max})\cdot\frac{1}{1-\rho}\cdot\left(1 + \frac{(C+C')^2}{1-\rho}\right)\left(C^2 + (C')^2\right)\cdot\frac{4 + \omega_{min}^2}{\epsilon\beta\omega_{min}^2}\right)^{1/2}.$$

and the notation $O(\cdot)$ hides polynomial dependence on system parameters ϵ, β, γ and d.

The proof outline of Theorem A.3 contains two parts: (1) Bounding the per-step error of SODA at each time step when compared against the hindsight optimal policy; (2) Showing that the past per-step does not accumulate to be unbounded over time.

Bounding the Per-step error. We introduce the concept of *per-step error* to characterize the decision error of SODA at each time step due to its limited prediction power. While the prediction power of SODA is limited because it only has exact predictions of future bandwidths within a finite horizon K, the idea of per-step error also extends to inexact predictions (Section A.4). We provide the formal definition of the per-step error in Definition A.2.

Definition A.2. *The per-step error of* SODA *at time step t (denoted as e_t) is defined as the sum of the difference between the actual state/action pair of* SODA (x_t, u_t) *and the clairvoyant optimal next state from (x_{t-1}, u_{t-1}), i.e.,*

$$e_t \coloneqq \left| x_t - \psi_t^N\left((x_{t-1}, u_{t-1}); \omega_{t:N}; \mathbf{0}\right)_{x_t} \right| + \left| u_t - \psi_t^N\left((x_{t-1}, u_{t-1}); \omega_{t:N}; \mathbf{0}\right)_{u_t} \right|$$

Intuitively, starting from the state/action pair (x_{t-1}, u_{t-1}), we compare the actual next state/action pair (x_t, u_t) of SODA with the clairvoyant optimal next state/action a controller would take if it had the exact predictions of all future bandwidths after time step t. We define the magnitude of this difference as the per-step error of SODA.

When the predictions of future bandwidths are exact, we leverage the exponentially decaying perturbation property to bound the per-step error of SODA in Lemma A.4. We defer the proof of Lemma A.4 to Section C.1.

Lemma A.4. *When the predictions for the future bandwidth are exact, the per-step error of* SODA *satisfies*

$$e_t^2 \le 16\rho^{4K-2}\left(C^2 + (C')^2\right)^2\left(\left|x_{t-1} - x_{t-1}^*\right|^2 + \left|u_{t-1} - u_{t-1}^*\right|^2\right) + 8\rho^{2K-2}\left(C^2 + (C')^2\right)\frac{(2+\omega_{min}^2)b(x_{t+K-1}^*) + 2b(x_{t+K-2}^*)}{\epsilon\omega_{min}^2}.$$

The exponentially decaying coefficients ρ^{4K-2} and ρ^{2K-2} suggest that the per-step error improves exponentially fast as the prediction horizon K grows. Although one can simplify the expression by bounding the terms $\left|x_{t-1} - x_{t-1}^*\right|^2$, $\left|u_{t-1} - u_{t-1}^*\right|^2$, $b(x_{t+K-1}^*)$, and $b(x_{t+K-2}^*)$ with some uniform constants, we keep these terms because the careful treatment is required to show the competitive ratio result.

Bounding the accumulation of past errors. Besides bounding the per-step errors, another important consequence of the exponentially decaying perturbation bounds is that it guarantees the impact of a previous per-step error decays quickly over time. Therefore, when we bound the total difference between SODA's trajectory $\{(x_t, u_t)\}_{t=1}^N$ and the offline optimal trajectory $\{(x_t^*, u_t^*)\}_{t=1}^N$, the aggregated contribution of any per-step error term e_τ is up to a constant factor that depends on the decay factor rather than growing linearly with respect to the total horizon length N (see Figure 14 for an illustration). We state this result formally in Lemma A.5 and defer its proof to Section C.2.

Lemma A.5. *The trajectory of* SODA $\{(x_t, u_t)\}_{t=1}^N$ *satisfies that*

$$\sum_{t=1}^N\left(\left|x_t - x_t^*\right|^2 + \left|u_t - u_t^*\right|^2\right) \le \frac{1}{1-\rho}\cdot\left(1 + \frac{(C+C')^2}{1-\rho}\right)\sum_{t=1}^N e_t^2,$$

where $\{(x_t^*, u_t^*)\}_{t=1}^N$ denotes the offline optimal trajectory.

By combining Lemma A.4 and Lemma A.5, we bound the total squared distance between SODA's trajectory and the offline optimal trajectory by a part of the offline optimal cost times a coefficient of the order $O(\rho^{2K})$. Since the cost functions for adaptive video streaming are well-conditioned, we can convert the bound on the total squared distance between the two trajectories into the competitive ratio bound and the dynamic regret bound to finish the proof of Theorem A.3.

A.4 Proof Outline for Inexact Predictions

Compared with the case when all predictions are exact, a major challenge when the predictions are inexact is that one of SODA's decisions may cause the next state to violate the state constraint. In this section, we show in two steps that SODA's decision trajectory will not violate the state constraints. First, by increasing the coefficient β of the buffer cost, one can guarantee that the offline optimal trajectory stays arbitrarily close to the offline optimal trajectory (see Lemma A.6). Then, we show a bound on the per-step error (Definition A.2) which depends on the prediction error (see Lemma A.7). Recall that the exponentially decaying perturbation bounds allow us to bound the distance between the SODA and the offline optimal trajectories. Therefore, we can combine these results to show that under some mild assumptions on the coefficient β and the prediction errors, SODA will not violate any constraints, and moreover, it also satisfies a dynamic regret bound (see Theorem A.8).

We first show that for any $\zeta > 0$, one can select the coefficient β to be sufficiently large so that the offline optimal trajectory stays within a margin of ζ around the target buffer level $\bar{x}$. We state this result formally in Lemma A.6 and defer its proof to Section D.1.

Lemma A.6. *Suppose $\zeta \le \min\{\bar{x}, x_{max} - \bar{x}\}$ is positive number and $x_0 \le \bar{x} + \zeta$, if the coefficient β for the buffer cost is sufficiently large such that*

$$\beta \ge \frac{1}{\epsilon\zeta}\cdot\left(1 + \frac{4\gamma}{\omega_{min}}\right)\cdot\left(\frac{1}{r_{min}} - \frac{1}{r_{max}}\right).$$

Then, the offline optimal trajectory satisfies $x_t^* \in [\bar{x} - \zeta, \bar{x} + \zeta]$ holds for all time step t.

Intuitively, Lemma A.6 holds because increasing β makes staying close to the target buffer level more important. In the extreme case that β tends to $+\infty$, the offline optimal will ignore the distortion/switching cost and select actions so that the buffer level always equal to $\bar{x}$.

Recall that the per-step error of SODA is defined in Definition A.2. We bound the per-step error in Lemma A.7 and defer its proof to Section D.2.

Lemma A.7. *When the predictions for the future bandwidth are inexact, the per-step error of* SODA *satisfies*

$$e_t \le (C+C')\rho^K\left(x_{max}+\frac{1}{r_{min}}-\frac{1}{r_{max}}\right)+(C+C')\cdot E(t-1,K)+\frac{\left|\omega_t-\hat{\omega}_{t|t-1}\right|}{r_{min}},$$

where $E(t-1,K) \coloneqq \sum_{\tau=t}^{t+K-1}\rho^{\tau-t}\left|\hat{\omega}_{\tau|t-1}-\omega_\tau\right|$.

Similar to the proof outline for the exact prediction case in Section A.3, we can apply Lemma A.5 to bound the accumulation of past errors.

With the help of Lemma A.6 and Lemma A.7, we show our main result for SODA when the predictions of the future bandwidths are inexact in Theorem A.8. We defer the proof of Theorem A.8 to Section D.3.

Theorem A.8. *Under Assumption A.1, consider* SODA *with the terminal constraints* $x_{t+K-1}=\bar{x}, r_{t+K-1}=\hat{\omega}_{t+K-1|t-1}$. *Let* $D \coloneqq \min\{\bar{x}, x_{max}-\bar{x}\}$. *Suppose the weight* β, *the prediction horizon* K, *and the prediction errors satisfy that*

$$\beta \ge \frac{3}{\epsilon D}\cdot\left(1+\frac{4\gamma}{\omega_{min}}\right)\cdot\left(\frac{1}{r_{min}}-\frac{1}{r_{max}}\right),\ \textit{and}$$

$$E(t,K)+\rho^K \le \frac{(1-\rho)D}{3C(1+C+C')\left(1+x_{max}+\frac{1}{r_{min}}-\frac{1}{r_{max}}\right)},$$

where, recall, $E(t,K)=\sum_{\tau=t+1}^{t+K}\rho^{\tau-t-1}\left|\hat{\omega}_{\tau|t}-\omega_\tau\right|$. *Then, the buffer levels in the* SODA*'s decision trajectory never hits the constraint boundary, i.e.,* $0<x_t<x_{max}$ *for* $t=1,\dots,N$. *Further,* SODA *achieves a dynamic regret of*

$$\frac{2\left(1+\frac{1}{r_{min}}+C+C'\right)^2\left(1+x_{max}+\frac{1}{r_{min}}-\frac{1}{r_{max}}\right)}{(1-\rho)^{3/2}}\cdot\sqrt{4\gamma+\beta+\omega_{max}}\cdot\sqrt{\mathcal{E}\cdot cost(\mathtt{OPT})}+$$

$$\frac{\left(1+\frac{1}{r_{min}}+C+C'\right)^4\left(1+x_{max}+\frac{1}{r_{min}}-\frac{1}{r_{max}}\right)^2(4\gamma+\beta+\omega_{max})}{(1-\rho)^3}\cdot\mathcal{E},$$

where $\mathcal{E}=\rho^{2K}N+\sum_{\kappa=1}^{K}\rho^\kappa E_\kappa$. *Here* $E_\kappa \coloneqq \sum_{t=1}^{N}\left|\hat{\omega}_{t+\kappa|t}-\omega_{t+\kappa}\right|^2$.

Note that the dynamic regret bound shown in Theorem A.8 is in the order of $O(\sqrt{\mathcal{E}N}+\mathcal{E})$, since $\text{cost}(\mathtt{OPT})=O(N)$. Intuitively, from the form of $E_t(K)$, we see that predicting the future bandwidth ω_τ accurately at time step t becomes less important as $(\tau-t)$ increases.

A.5 Proof Outline for Efficient Structure

In this section, we show that optimal solution of the finite-time optimal control problem solved by SODA can be approximated well by a monotonic sequence of bitrates when the coefficient γ of the switching cost is sufficiently large (see Theorem A.9). Although this result is shown for the continuous variable case, it also provides some insight as to why the efficient approximate solver in Algorithm 1 can provide identical decisions to the brute-force solver with relatively high probabilities, as shown in Figure 8.

Theorem A.9. *Let* $\hat{\omega}_{\times K}$ *denote the sequence* $\{\hat{\omega},\dots,\hat{\omega}\}$ *with length* K. *For any* $\lambda>0$, *when the coefficient* γ *is sufficiently large such that*

$$\gamma \ge \frac{K^2}{\lambda^2}\left(\hat{\omega}\left(\frac{1}{r_{min}^2}-\frac{1}{r_{max}^2}\right)+\beta\max\{\bar{x}^2,\epsilon(x_{max}-\bar{x})^2\}\right),$$

we have that the following inequality holds for all $\tau\in\{t,t+1,\dots,t+K-1\}$:

$$\left|\hat{\psi}_t^{t+K-1}\left((\sigma_{t-1},v_{t-1});\hat{\omega}_{\times K};\mathbf{0}\right)_{u_\tau}-\hat{\phi}_t^{t+K-1}\left((\sigma_{t-1},v_{t-1});\hat{\omega};\mathbf{0}\right)_{u_\tau}\right|\le\lambda.$$

Note that $\hat{\phi}_t^{t+K-1}\left((\sigma_{t-1},v_{t-1});\hat{\omega};\mathbf{0}\right)_{u_\tau}$ *is monotonic by Lemma A.10.*

We defer the formal proof of Theorem A.9 to Section E.2. The theoretical insight provided by Theorem A.9 aligns with our empirical result in Figure 8. Specifically, if we increase the coefficient γ while keeping the prediction horizon K fixed, the decision made by the efficient monotonic approximation approach (Algorithm 1) is more likely to be identical with the brute-force solver. On the other hand, if we increase K and fix γ, it is more challenging for Algorithm 1 to match the decision of the brute-force solver.

To show Theorem A.9, we first consider a setting where the objective function only contains the switching cost terms (i.e., the distortion cost and the buffer cost are removed.) This can be viewed as the extreme case when γ tends to $+\infty$ so that both α and β are negligible. In this scenario, we show the optimal sequence of the inverse bitrates is monotonic. We state this result formally in Lemma A.10 and defer its proof to Section E.1.

Lemma A.10. *Under the same assumption as Theorem A.9, consider the optimal solution to the optimization problem*

$$\begin{aligned}\hat{\phi}_t^{t+K-1}\left((\sigma_{t-1}, v_{t-1}); \hat{\omega}; \mathbf{0}\right) \coloneqq \underset{u_{t:t+K-1}}{\arg\min} & \sum_{\tau=t}^{t+K-1} \gamma \cdot (u_t - u_{t-1})^2 \\ s.t.\ & x_\tau = x_{\tau-1} + \hat{\omega} u_\tau - 1, \ \textit{for } \tau = t, \ldots, t+K-1, \\ & x_\tau \in [0, x_{max}], u_\tau \in \left[\frac{1}{r_{max}}, \frac{1}{r_{min}}\right], \ \textit{for } \tau = t, \ldots, t+K-1, \\ & x_{t-1} = \sigma_{t-1}, u_{t-1} = v_{t-1}. \end{aligned} \tag{6}$$

The solution satisfies that: If $v_{t-1} > 1/\hat{\omega}$, then the sequence $v_{t-1}, \hat{\phi}_t^{t+K-1}\left((\sigma_{t-1}, v_{t-1}); \hat{\omega}; \mathbf{0}\right)$ is monotonically decreasing; If $v_{t-1} < 1/\hat{\omega}$, then the sequence $v_{t-1}, \hat{\phi}_t^{t+K-1}\left((\sigma_{t-1}, v_{t-1}); \hat{\omega}; \mathbf{0}\right)$ is monotonically increasing; If $v_{t-1} = 1/\hat{\omega}$, the optimal solution is $u_t = u_{t+1} = \cdots = u_{t+K-1} = v_{t-1} = 1/\hat{\omega}$.

The key observation that allows us to generalize Lemma A.10 to the case where the distortion/buffer costs are non-negligible is the following: If we change the variable of (6) to $a_t = u_t - u_{t-1}$, which denotes the increments of the control actions, the objective of (6) is a γ-strongly convex function of $(a_t, \ldots, a_{t+K-1})$. Any deviation from the optimal solution of (6) will cause a loss on the total switching costs that grows with γ. When γ is sufficiently large, a feasible solution cannot use its gain on the distortion/buffer costs to cancel the loss on the total switching cost if it deviates too much from the optimal solution of (6).

B PROOFS OF THE EXPONENTIALLY DECAYING PERTURBATION BOUNDS

In this section, we establish the critical exponentially decaying perturbation bounds (Definition A.1). Instead of just focusing on the video streaming application itself, we establish the perturbation bound for a more general SOCO with memory framework.

Specifically, we consider the following finite-time optimal control problem with memory H.

$$\psi(y, z; \mu, w, \delta) = \underset{x_{-H+1:p+H-1}}{\arg\min} \sum_{t=0}^{p} f_t(x_t; \mu_t) + \sum_{t=0}^{p+H-1} c_t(x_{t:t-H+1}; w_t) \tag{7a}$$

$$\text{s.t. } x_t \in [0, x_{\max}] \subseteq \mathbb{R}, \forall 0 \le t \le p, \tag{7b}$$

$$x_t - x_{t-1} \ge -\delta_t, \forall 0 \le t \le p+1, \tag{7c}$$

$$x_{-H+1:-1} = y, x_{p+1:p+H-1} = z, \tag{7d}$$

where $y, z \in [0, x_{\max}]^{H-1}$, $\mu \in [0, x_{\max}]^{p+1}, w \in \mathcal{W}^{p+H}, \delta \in \Delta^{p+2}$. Here, the objective function (7a) contains the hitting costs $f_t(x_t; \mu_t)$ (parameterized by μ_t) and the switching costs $c_t(x_{t:t-H+1}; w_t)$ (parameterized by w_t). For the constraints, (7b) imposes a box constraint on each decision variable x_t; (7c) imposes a constraint on how much x_t can decrease at each time step; and (7d) specifies the boundary conditions of the optimization problem.

In the special case of video streaming, the decision is on the buffer level x_t. Given the buffer levels, the inverse of the bitrate $u_t \coloneqq 1/r_t$ is uniquely decided by the equation

$$u_t = (x_t - x_{t-1} + 1)/\omega_t,$$

where ω_t denotes the bandwidth. The memory length $H = 3$. For the hitting cost, we have $\mu_t \equiv \bar{x}$, and

$$f_t(x; \mu_t) = \beta b(x) = \begin{cases} \beta(x - \bar{x})^2, & \text{if } x \le \bar{x}, \\ \epsilon\beta(x - \bar{x})^2, & \text{otherwise.} \end{cases}$$

For the switching cost, we have $w_t = (\omega_t, \omega_{t-1})$ and

$$\begin{aligned} c_t(x_{t:t-2}; w_t) &= \omega_t u_t^2 + \gamma(u_t - u_{t-1})^2 \\ &= \frac{(x_t - x_{t-1} + 1)^2}{\omega_t} + \gamma \frac{(\omega_{t-1} x_t + \omega_t x_{t-2} - (\omega_t + \omega_{t-1}) x_{t-1} + (\omega_{t-1} - \omega_t))^2}{\omega_t^2 \omega_{t-1}^2}. \end{aligned}$$

The first constraint $x_t \in [0, x_{\max}]$ of (7) matches the buffer constraint of the video streaming problem exactly.

The second constraint $x_t - x_{t-1} \ge -\delta_t$ corresponds to the constraint that $u_t \ge \frac{1}{r_{\max}}$ in (3). Thus, when applying (7) to video streaming, we have $\delta_t = 1 - \frac{\omega_t}{r_{\max}}$. By Assumption A.1, we have $\delta_t \ge \delta > 0$.

Given the relationship between SOCO with memory problem and adaptive video streaming problem, we only need to establish the exponentially decaying perturbation bound for the more general SOCO with memory problem. To show this perturbation bound, we need the following assumption about the objective function and constraints:

Assumption B.1. *We need the following assumption on the optimization problem* (7) *for the exponentially decaying perturbation property to hold:*

1) *$f_t(\cdot;\mu_t):\mathbb{R}\to\mathbb{R}$ is strongly convex for all t and $\mu_t\in[0,x_{max}]$. We further assume there exists two m_f-strongly convex and ℓ_f-smooth functions $f_t^{(0)}(\cdot;\mu_t), f_t^{(1)}(\cdot;\mu_t):\mathbb{R}\to\mathbb{R}$ in C^2 such that $f_t(x_t;\mu_t)=f_t^{(0)}(x_t;\mu_t)$ for $x_t\in[0,\mu_t]$ and $f_t(x_t)=f_t^{(1)}(x_t;\mu_t)$ for $x_t\in[\mu_t,x_{max}]$. We also assume that for $j=1,2$, $f_t^{(j)}$ satisfies that for all $x_t,\mu_t\in[0,x_{max}]$,*

$$\left\|\nabla_{x_t} f_t^{(j)}(x_t;\mu_t)\right\|+\left\|\nabla_{\mu_t} f_t^{(j)}(x_t;\mu_t)\right\|\le L_f,\ \text{and}\ \left\|\nabla_{\mu_t}\nabla_{x_t} f_t^{(j)}(x_t;\mu_t)\right\|\le \ell_\mu.$$

2) *$c_t(\cdot;w_t):\mathbb{R}^H\to\mathbb{R}$ is convex and ℓ_c-smooth for all t and $w_t\in\mathcal{W}\subset\mathbb{R}^q$. $c_t(\cdot;w_t)$ is in C^2 on $[0,x_{max}]^H$. We also assume that for all $w_t\in\mathcal{W}$ and feasible $x_{t:t-H+1}$, we have*

$$\left\|\nabla_{x_{t:t-H+1}} c_t(x_{t:t-H+1};w_t)\right\|+\left\|\nabla_{w_t} c_t(x_{t:t-H+1};w_t)\right\|\le L_c,\ \text{and}$$
$$\left\|\nabla_{w_t}\nabla_{x_{t:t-H+1}} c_t(x_{t:t-H+1};w_t)\right\|\le \ell_w.$$

3) *We have $\delta_t\in\Delta$ holds for all t, where Δ is a closed interval on $\mathbb{R}$ and is bounded below by some positive constant δ. Denote $d:=\lceil x_{max}/\delta\rceil$.*

In the special case of the video streaming problem, Assumption B.1 is satisfied with the parameters $m_f=\epsilon\beta$, $\ell_f=\ell_\mu=\beta$, $\ell_c=\frac{2(\omega_{\min}+3)}{\omega_{\min}^2}$, $\ell_w=\frac{4x_{\max}(\omega_{\min}+8\gamma)}{\omega_{\min}^3}$. In addition, both L_f and L_c are bounded.

We state the exponentially decaying perturbation bound for the SOCO with memory problem formally in Theorem B.1 and defer its proof to Appendix B.1.

Theorem B.1. *Under Assumption B.1, if $p\ge d$, the inequality*

$$\begin{aligned}&\left\|\psi(y,z;\mu,w,\delta)_t-\psi(y',z';\mu',w',\delta')_t\right\|\\ &\le C\left(\rho^t\left\|y-y'\right\|+\rho^{p-t}\left\|z-z'\right\|\right)+C\left(\sum_{\tau=0}^{p}\rho^{|t-\tau|}\left|\mu_\tau-\mu'_\tau\right|+\sum_{\tau=0}^{p+H-1}\rho^{|t-\tau|}\left\|w_\tau-w'_\tau\right\|+\sum_{\tau=0}^{p+1}\rho^{|t-\tau|}\left\|\delta_\tau-\delta'_\tau\right\|\right)\end{aligned}\tag{8}$$

holds for all $t\in[0,p]$ and $y,z\in[\underline{x},\overline{x}]^{H-1}$. Here,

$$\rho=\left(1-\frac{2}{1+\sqrt{1+(\underline{\ell}/m_f)}}\right)^{\frac{1}{H(H+d)}},\ C=\frac{2\bar{\ell}}{m_f\rho^{(H-2)(H+d)}},$$

where $\underline{\ell}:=\max\{H\ell_c,\ell_w\}$ and $\bar{\ell}:=\max\{H\ell_f,\ell_\mu,\underline{\ell}\}$.

In the special case of the video streaming, we see that

$$\underline{\ell}=\max\{3\ell_c,\ell_w\}=\frac{\max\{6\omega_{\min}(\omega_{\min}+3),4x_{\max}(\omega_{\min}+8\gamma)\}}{\omega_{\min}^3}.$$

Therefore, we have

$$\rho=\left(1-\frac{2}{1+\sqrt{1+\frac{\max\{6\omega_{\min}(\omega_{\min}+3),4x_{\max}(\omega_{\min}+8\gamma)\}}{\omega_{\min}^3\epsilon\beta}}}\right)^{\frac{1}{3(3+\lceil x_{\max}/\delta\rceil)}}.$$

The coefficient C is bounded by

$$C\le\frac{3\beta\omega_{\min}^3+\max\{6\omega_{\min}(\omega_{\min}+3),4x_{\max}(\omega_{\min}+8\gamma)\}}{\omega_{\min}^3\rho^{3+\lceil x_{\max}/\delta\rceil}}.$$

Discussion about different distortion costs. Note that Assumption B.1 still holds if we replace the distortion cost function $v(r)=\frac{1}{r}$ by $v(r)=\log(r_{\max}/r)$. This is because the new switching cost

$$\begin{aligned}c'_t(x_{t:t-2};w_t)&=\omega_t u_t\log(r_{\max}u_t)+\gamma(u_t-u_{t-1})^2\\&=(x_t-x_{t-1}+1)\log\left(\frac{r_{\max}(x_t-x_{t-1}+1)}{\omega_t}\right)\\&\quad+\gamma\frac{(\omega_{t-1}x_t+\omega_t x_{t-2}-(\omega_t+\omega_{t-1})x_{t-1}+(\omega_{t-1}-\omega_t))^2}{\omega_t^2\omega_{t-1}^2}\end{aligned}$$

also satisfies Assumption B.1 for any $w_t=(\omega_t,\omega_{t-1})\in[\omega_{\min},\omega_{\max}]^2$ and feasible $x_{t:t-2}$.

B.1 Proof of Theorem B.1

To show Theorem B.1, we first need to define *indicators of active constraints*, denoted as $\xi \in \{0,1\}^{4p+5}$. Specifically, given the unique optimal solution $x_{0:p} = \psi(y, z; \mu, w, \delta)$ under a tuple of parameters $(y, z; \mu, w, \delta)$, we consider whether the following equality conditions hold:

$$\begin{aligned}
&\xi_{1,t} = \mathbf{1}\{x_t = 0\}, \forall 0 \le t \le p;\\
&\xi_{2,t} = \mathbf{1}\{x_t = x_{\max}\}, \forall 0 \le t \le p;\\
&\xi_{3,t} = \mathbf{1}\{x_t = \mu_t\}, \forall 0 \le t \le p;\\
&\xi_{4,t} = \mathbf{1}\{x_t - x_{t-1} = -\delta_t\}, \forall 0 \le t \le p+1.
\end{aligned}$$

And we define *indicators of the sides* (denoted as $\sigma \in \{0,1\}^{p+1}$) as the following:

$$\sigma_t = \mathbf{1}\{x_t \in [\mu_t, x_{\max}]\}, \forall 0 \le t \le p.$$

To simplify the notation, we let $\theta \coloneqq (\mu, w, \delta) \in \Theta \coloneqq [0, x_{\max}]^{p+1} \times \mathcal{W}^{p+H} \times \Delta^{p+2}$. While $\psi(y, z; \theta)$ can decide a unique pair of (ξ, σ), we can also define a new equality-constrained optimization problem using $(y, z; \theta)$ and (ξ, σ):

Definition B.1. *We define the equality-constrained optimization problem $\hat{\psi}(y, z; \theta; \xi, \sigma)$ as*

$$\hat{\psi}(y, z; \theta; \xi, \sigma) = \underset{x_{-H+1:p+H-1}}{\arg\min} \sum_{t=0}^{p} f_t^{(\sigma_t)}(x_t; \mu_t) + \sum_{t=0}^{p+H-1} c_t(x_{t:t-H+1}; w_t) \tag{9a}$$

$$s.t.\ x_t = \begin{cases} 0, & if\ \xi_{1,t} = 1 \\ x_{max}, & if\ \xi_{2,t} = 1 \\ \mu_t, & if\ \xi_{3,t} = 1 \end{cases}, \forall 0 \le t \le p, \tag{9b}$$

$$x_t - x_{t-1} = -\delta_t,\ if\ \xi_{4,t} = 1, \forall 0 \le t \le p+1, \tag{9c}$$

$$x_{-H+1:-1} = y, x_{p+1:p+H-1} = z. \tag{9d}$$

Note that it is possible that the optimization problem $\hat{\psi}(y, z; \theta; \xi, \sigma)$ for some parameters and constraint configurations. We use $\hat{\iota}(y, z; \theta; \xi, \sigma)$ to denote the optimal value of this optimization problem. The following lemma states that the optimal solution of (7) will not change if we remove all inactive inequality constraints and leave active constraints as equality constraints.

Lemma B.2. *Suppose Assumption B.1 holds and $p \ge d$. For $y, z \in [0, x_{max}]^{H-1}$ and $\theta \in \Theta$, let ξ, σ be the corresponding indicators of active constraints/sides. Then, we have*

$$\psi(y, z; \theta) = \hat{\psi}(y, z; \theta; \xi, \sigma) \text{ and } \iota(y, z; \theta) = \hat{\iota}(y, z; \theta; \xi, \sigma).$$

Proof of Lemma B.2. Note that

$$\iota(y, z; \theta) \ge \hat{\iota}(y, z; \theta; \xi, \sigma)$$

because the optimization problem on the RHS has less constraints. If the inequality holds with equality, we must have $\psi(y, z; \theta) = \hat{\psi}(y, z; \theta; \xi, \sigma)$ since the optimal solution for the LHS is feasible for the RHS by the assumption on active constraints, and the optimization problem on the RHS has a unique solution. Otherwise, we must have

$$\psi(y, z; \theta) \ne \hat{\psi}(y, z; \theta; \xi, \sigma), \text{ and } \iota(y, z; \theta) > \hat{\iota}(y, z; \theta; \xi, \sigma).$$

Consider the convex combination $\zeta(\eta)$ for $\eta \in [0, 1]$ defined as

$$\zeta(\eta) = (1 - \eta)\psi(y, z; \theta) + \eta\hat{\psi}(y, z; \theta; \xi, \sigma).$$

Note that $\zeta(\eta)$ satisfies all the active constraints and sides as specified by (ξ, σ) because they are active for all $\eta \in [0, 1]$. Since the constraints of (7) that are not in (ξ, σ) are inactive at $\eta = 0$, there must exist $\eta > 0$ such that $\zeta(\eta)$ is also feasible for (7). $\zeta(\eta)$ achieves a strictly smaller objective than $\zeta(0) = \psi(y, z; \theta)$, which leads to a contradiction. □

Lemma B.2 establishes that given any feasible tuple of $(y, z; \theta)$, one can find at least one pair of (ξ, σ) such that $\psi(y, z; \theta) = \hat{\psi}(y, z; \theta; \xi, \sigma)$, while there can be other (ξ', σ') that satisfies $\psi(y, z; \theta) = \hat{\psi}(y, z; \theta; \xi', \sigma')$.

Lemma B.3. *Suppose Assumption B.1 holds and $p \ge d$. If both $\hat{\psi}(y, z; \theta; \xi, \sigma)$ and $\hat{\psi}(y', z'; \theta'; \xi, \sigma)$ exist for $y, z, y', z' \in [0, x_{max}]^{H-1}$ and (ξ, σ), then we have*

$$\begin{aligned}
&\left\|\hat{\psi}(y, z; \theta; \xi, \sigma)_t - \hat{\psi}(y', z'; \theta'; \xi, \sigma)_t\right\| \\
&\le C\left(\rho^t \left\|y - y'\right\| + \rho^{p-t}\left\|z - z'\right\|\right) + C\left(\sum_{\tau=0}^{p} \rho^{|t-\tau|}\left|\mu_\tau - \mu'_\tau\right| + \sum_{\tau=0}^{p+H-1} \rho^{|t-\tau|}\left\|w_\tau - w'_\tau\right\| + \sum_{\tau=0}^{p+1} \rho^{|t-\tau|}\left|\delta_\tau - \delta'_\tau\right|\right),
\end{aligned} \tag{10}$$

where

$$\rho = \left(1 - \frac{2}{1+\sqrt{1+(\underline{\ell}/m_f)}}\right)^{\frac{1}{H(H+d)}}, C = \frac{2\bar{\ell}}{m_f \rho^{(H-2)(H+d)}}.$$

Here, $\underline{\ell} \coloneqq \max\{H\ell_c, \ell_w\}$ *and* $\bar{\ell} \coloneqq \max\{H\ell_f, \ell_\mu, \underline{\ell}\}$.

Proof of Lemma B.3. We do a variable change to eliminate all constraints in the equality-constrained optimization problem. After the elimination, we get an unconstrained optimization problem with the free variables $x_{t_0}, x_{t_1}, \ldots, x_{t_q}$ where the indices satisfy $0 \le t_0 < t_1 < \ldots < t_q \le p$. To simplify the notation, we let $t_{-1} = -1$ and $t_{q+1} = p+1$. For τ that satisfies $t_i < \tau < t_{i+1}$, we have either $x_\tau = x_{t_i} - \sum_{\gamma=t_i+1}^{\tau} \delta_\gamma$ or x_τ is some constant. Without loss of generality, we can assume $t_{i+1} \le t_i + d + H$, because otherwise we can find $\tau \in (t_i, t_{i+1} - H]$ such that $x_{\tau:\tau+H-1}$ are constants, which means the free variables after $x_{t_{i+1}}$ will not change, regardless of how we perturb y, and the free variables before x_{t_i} will not change, regardless of how we perturb z. Thus, we can decompose the perturbation to the left side and the right side and derive them separately.

After the change of variable, the objective becomes a function $\hat{h}$ of $x_{t_0}, x_{t_1}, \ldots, x_{t_q}$. To simplify the notation, we let $\hat{x}_\tau \coloneqq x_{t_\tau}$, where $\tau = 0, \ldots, q$. We can decompose $\hat{h}$ as

$$\hat{h}(\hat{x}_{0:q}; \zeta) = \hat{h}_a(\hat{x}_{0:q}; \mu) + \hat{h}_b(\hat{x}_{0:q}; \zeta),$$

where $\zeta = (y, z, \theta)$, $\hat{h}_a$ is the sum of the original hitting costs minus $\frac{m_f}{2}\left\|\hat{x}_{0:q}\right\|^2$, and $\hat{h}_b$ is the sum of the original switching costs plus $\frac{m_f}{2}\left\|\hat{x}_{0:q}\right\|^2$. By Assumption B.1, we see that

$$\nabla^2_{\hat{x}_{0:q}} \hat{h}_a(\hat{x}_{0:q}; \mu) \succeq 0, (m_f + H\ell_c) I \succeq \nabla^2_{\hat{x}_{0:q}} \hat{h}_b(\hat{x}_{0:q}; \zeta) \succeq m_f I. \tag{11}$$

We also note that $\nabla^2_{\hat{x}_{0:q}} \hat{h}_a(\hat{x}_{0:q}; \mu)$ is a diagonal matrix and $\nabla^2_{\hat{x}_{0:q}} \hat{h}_b(\hat{x}_{0:q}; \zeta)$ is a $2H$-banded matrix.

We can follow a similar procedure as Theorem 3.1 in [49] to show

$$\begin{aligned}
&\left\|\hat{\psi}(y, z; \theta; \xi, \sigma)_{t_\tau} - \hat{\psi}(y', z'; \theta'; \xi, \sigma)_{t_\tau}\right\| \\
&\le C_0 \left(\rho_0^\tau \left\|y - y'\right\| + \rho_0^{q-\tau} \left\|z - z'\right\|\right) + C_0 \left(\sum_{i=0}^{p} \rho_0^{|\phi(i)-\tau|} \left|\mu_i - \mu_i'\right| + \sum_{i=0}^{p+H-1} \rho_0^{|\phi(i)-\tau|} \left\|w_i - w_i'\right\| + \sum_{i=0}^{p+1} \rho_0^{|\phi(i)-\tau|} \left\|\delta_i - \delta_i'\right\|\right),
\end{aligned} \tag{12}$$

where $\phi(i)$ denotes the integer j that satisfies $t_j \le i < t_{j+1}$ and

$$\rho_0 = \left(1 - \frac{2}{1+\sqrt{1+(\underline{\ell}/m_f)}}\right)^{\frac{1}{H}}, C_0 = \frac{2\bar{\ell}}{m_f \rho_0^{H-2}}.$$

Here, $\underline{\ell} \coloneqq \max\{H\ell_c, \ell_w\}$ and $\bar{\ell} \coloneqq \max\{H\ell_f, \ell_\mu, \underline{\ell}\}$. For completeness, we give the detailed proof below: Let e be a vector such that both ζ and $\zeta + e$ are in $\mathcal{Y} \times \mathcal{Z} \times \Theta$. Consider the function

$$\overline{\psi}(\zeta + \eta e) \coloneqq \hat{\psi}(\zeta + \eta e; \xi, \sigma)_{t_{0:q}},$$

which is implicitly determined by the equation

$$\nabla_{\hat{x}_{0:q}} \hat{h}(\overline{\psi}(\zeta + \eta e), \zeta + \eta e) = 0.$$

By the implicit function theorem we know that the function $\overline{\psi}$ is differentiable. Taking the derivative with respect to θ gives that

$$\begin{aligned}
\nabla^2_{\hat{x}_{0:q}} \hat{h}(\overline{\psi}(\zeta + \eta e), \zeta + \eta e) \frac{d}{d\eta} \overline{\psi}(\zeta + \eta e) = &- \nabla_y \nabla_{\hat{x}_{0:q}} \hat{h}(\overline{\psi}(\zeta + \eta e), \zeta + \eta e) e_y - \nabla_z \nabla_{\hat{x}_{0:q}} \hat{h}(\overline{\psi}(\zeta + \eta e), \zeta + \eta e) e_z \\
&- \sum_{t=0}^{p} \nabla_{\mu_t} \nabla_{\hat{x}_{0:q}} \hat{h}(\overline{\psi}(\zeta + \eta e), \zeta + \eta e) e_{\mu_t} - \sum_{t=0}^{p+H-1} \nabla_{w_t} \nabla_{\hat{x}_{0:q}} \hat{h}(\overline{\psi}(\zeta + \eta e), \zeta + \eta e) e_{w_t} \\
&- \sum_{t=0}^{p} \nabla_{\delta_t} \nabla_{\hat{x}_{0:q}} \hat{h}(\overline{\psi}(\zeta + \eta e), \zeta + \eta e) e_{\delta_t}.
\end{aligned}$$

To simplify the notation, we define

$$\begin{aligned}
M &\coloneqq \nabla^2_{\hat{x}_{0:q}} \hat{h}(\overline{\psi}(\zeta+\eta e),\zeta+\eta e), \text{which is a } (q+1)\times(q+1) \text{ matrix},\\
R^{(y)} &\coloneqq -\nabla_y \nabla_{\hat{x}_{0:q}} \hat{h}(\overline{\psi}(\zeta+\eta e),\zeta+\eta e), \text{which is a } (q+1)\times(H-1) \text{ matrix},\\
R^{(z)} &\coloneqq -\nabla_z \nabla_{\hat{x}_{0:q}} \hat{h}(\overline{\psi}(\zeta+\eta e),\zeta+\eta e), \text{which is a } (q+1)\times(H-1) \text{ matrix},\\
R^{(\mu_t)} &\coloneqq -\nabla_{\mu_t} \nabla_{\hat{x}_{0:q}} \hat{h}(\overline{\psi}(\zeta+\eta e),\zeta+\eta e), \text{which is a } (q+1)\times 1 \text{ matrix},\\
R^{(w_t)} &\coloneqq -\nabla_{w_t} \nabla_{\hat{x}_{0:q}} \hat{h}(\overline{\psi}(\zeta+\eta e),\zeta+\eta e), \text{which is a } (q+1)\times d \text{ matrix},\\
R^{(\delta_t)} &\coloneqq -\nabla_{\delta_t} \nabla_{\hat{x}_{0:q}} \hat{h}(\overline{\psi}(\zeta+\eta e),\zeta+\eta e), \text{which is a } (q+1)\times 1 \text{ matrix}.
\end{aligned}$$

Hence we can write

$$\frac{d}{d\theta}\overline{\psi}(\zeta+\eta e) = M^{-1}\left(R^{(y)}e_y + R^{(z)}e_z + \sum_{t=0}^{p} R^{(\mu_t)}e_{\mu_t} + \sum_{t=0}^{p+H-1} R^{(w_t)}e_{w_t} + \sum_{t=0}^{p} R^{(\delta_t)}e_{\delta_t}\right).$$

Recall that $R^{(y)}, R^{(z)}$ are $(q+1)\times(H-1)$ matrices. For $R^{(y)}$, only the first $H-1$ rows are non-zero. For $R^{(z)}$, only the last $H-1$ rows are non-zero. Hence we see that

$$\begin{aligned}
\frac{d}{d\eta}\overline{\psi}(\zeta+\eta e)_\tau = {} & (M^{-1})_{\tau,0:H-2} R^{(y)}_{0:H-2,:} e_y + (M^{-1})_{\tau,q-H+2:q} R^{(z)}_{q-H+2:q,:} e_z\\
& + \sum_{j=0}^{q}\sum_{i=t_j}^{t_{j+1}-1} (M^{-1})_{\tau,j} R^{(\mu_i)}_{j,:} e_{\mu_i} + \sum_{j=0}^{q+1}\sum_{i=t_j}^{t_{j+1}-1} (M^{-1})_{\tau,j-H+1:j+H-1} R^{(w_i)}_{j-H+1:j+H-1,:} e_{w_i}\\
& + \sum_{j=0}^{q}\sum_{i=t_j}^{t_{j+1}-1} (M^{-1})_{\tau,j} R^{(\delta_i)}_{j,:} e_{\delta_i}.
\end{aligned} \tag{13}$$

Recall that $\bar{\ell} \coloneqq \max\{H\ell_c, H\ell_f, \ell_\mu, \ell_w\}$. We know that the norms of

$$R^{(y)}_{0:H-2,:}, R^{(z)}_{q-H+2:q,:}, R^{(\mu_i)}_{j,:}, R^{(w_i)}_{j-H+1:j+H-1,:}, \text{ and } R^{(\delta_i)}_{j,:}$$

are all upper bounded by $\bar{\ell}$. Taking norm on both sides of (13) gives

$$\begin{aligned}
\left\|\frac{d}{d\theta}\overline{\psi}(\zeta+\eta e)_\tau\right\| \le {} & \bar{\ell}\left\|(M^{-1})_{\tau,0:H-2}\right\|\|e_y\| + \bar{\ell}\left\|(M^{-1})_{\tau,q-H+2:q}\right\|\|e_z\|\\
& + \bar{\ell}\sum_{j=0}^{q}\sum_{i=t_j}^{t_{j+1}-1}\left\|(M^{-1})_{\tau,j}\right\|\|e_{\mu_i}\| + \bar{\ell}\sum_{j=0}^{q+1}\sum_{i=t_j}^{t_{j+1}-1}\left\|(M^{-1})_{\tau,j-H+1:j+H-1}\right\|\|e_{w_i}\|\\
& + \bar{\ell}\sum_{j=0}^{q}\sum_{i=t_j}^{t_{j+1}-1}\left\|(M^{-1})_{\tau,j}\right\|\|e_{\delta_i}\|.
\end{aligned} \tag{14}$$

Note that M can be decomposed as $M = M_a + M_b$, where

$$\begin{aligned}
M_a &\coloneqq \nabla^2_{\hat{x}_{0:q}} \hat{h}_a(\overline{\psi}(\zeta+\eta e),\zeta+\eta e),\\
M_b &\coloneqq \nabla^2_{\hat{x}_{0:q}} \hat{h}_b(\overline{\psi}(\zeta+\eta e),\zeta+\eta e).
\end{aligned}$$

Since M_a is a diagonal $(q+1)\times(q+1)$ matrix and satisfies $M_a \succeq 0$, and M_b is $2H$-banded and satisfies $(m_f + \underline{\ell})I \succeq M_b \succeq m_f I$, we obtain the following with Lemma B.1 in [49]:

$$\begin{aligned}
\left\|(M^{-1})_{\tau,0:H-2}\right\| &\le \frac{2}{m_f}\rho_0^{\tau-(H-2)}, \left\|(M^{-1})_{\tau,q-H+2:q}\right\| \le \frac{2}{m_f}\rho_0^{q-\tau-(H-2)}\\
\left\|(M^{-1})_{\tau,j}\right\| &\le \frac{2}{m_f}\rho_0^{|\tau-j|}, \left\|(M^{-1})_{\tau,j-H+1:j+H-1}\right\| \le \frac{2}{m_f}\rho_0^{|\tau-j|-(H-1)},
\end{aligned}$$

where $\rho_0 \coloneqq (\sqrt{cond(M_b)}-1)/(\sqrt{cond(M_b)}+1) = 1 - 2\cdot\left(\sqrt{1+(\underline{\ell}/\mu)}+1\right)^{-1}$.

Substituting this into (14), we see that

$$\left\|\frac{d}{d\theta}\overline{\psi}(\zeta+\theta e)_\tau\right\| \le C_0\left(\rho_0^\tau\|e_y\| + \rho_0^{q-\tau}\|e_z\| + \sum_{i=0}^{p}\rho_0^{|\phi(i)-\tau|}\|e_{\mu_i}\| + \sum_{i=0}^{p+H-1}\rho_0^{|\phi(i)-\tau|}\|e_{w_i}\| + \sum_{i=0}^{p}\rho_0^{|\phi(i)-\tau|}\|e_{\delta_i}\|\right).$$

Hence we obtain

$$\begin{aligned}\left\|\overline{\psi}(\zeta)_\tau - \overline{\psi}(\zeta+e)_\tau\right\| &= \left\|\int_0^1 \frac{d}{d\eta}\overline{\psi}(\zeta+\eta e)_\tau d\eta\right\| \\ &\le \int_0^1 \left\|\frac{d}{d\eta}\overline{\psi}(\zeta+\eta e)_\tau\right\| d\eta \\ &\le C_0\left(\rho_0^\tau \left\|e_y\right\| + \rho_0^{q-\tau}\left\|e_z\right\| + \sum_{i=0}^{p}\rho_0^{|\phi(i)-\tau|}\left\|e_{\mu_i}\right\| + \sum_{i=0}^{p+H-1}\rho_0^{|\phi(i)-\tau|}\left\|e_{w_i}\right\| + \sum_{i=0}^{p}\rho_0^{|\phi(i)-\tau|}\left\|e_{\delta_i}\right\|\right).\end{aligned}$$

This finishes the proof of (12). Recall that we have $t_i < t_{i+1} \le t_i + d + H$. Therefore, (12) implies (10). □

In the next lemma, we show a continuity property of the "equality-constrained labeling" method.

Lemma B.4. *Suppose Assumption B.1 holds and $p \ge d$. For a pair of (ξ, σ), if any tuple in the sequence $\{(y_q, z_q; \theta_q)\}_{q=1}^{\infty}$ satisfies $\psi(y_q, z_q; \theta_q) = \hat{\psi}(y_q, z_q; \theta_q; \xi, \sigma)$ and $\lim_{q\to\infty}(y_q, z_q, \theta_q) = (y, z, \theta)$, then we have*

$$\psi(y, z; \theta) = \hat{\psi}(y, z; \theta; \xi, \sigma).$$

Proof of Lemma B.4. Note that the perturbation bound in Lemma B.3 also establishes the continuity of the function $\hat{\psi}(\cdot,\cdot;\cdot;\xi,\sigma)$. Therefore, we see that

$$\lim_{q\to\infty}\psi(y_q, z_q; \theta_q) = \lim_{q\to\infty}\hat{\psi}(y_q, z_q; \theta_q; \xi, \sigma) = \hat{\psi}(y, z; \theta; \xi, \sigma).$$

Since the constraint set of (7) is closed, we know $\hat{\psi}(y, z; \theta; \xi, \sigma)$ is a feasible solution of (7).

For the sake of contradiction, we assume $\psi(y, z; \theta) \ne \hat{\psi}(y, z; \theta; \xi, \sigma)$. In this case, since $\hat{\psi}(y, z; \theta; \xi, \sigma)$ is feasible for (7), we must have

$$\iota(y, z; \theta) < \hat{\iota}(y, z; \theta; \xi, \sigma).$$

Define the optimality gap as $\Lambda := \hat{\iota}(y, z; \theta; \xi, \sigma) - \iota(y, z; \theta)$.

Since $\lim_{q\to\infty}(y_q, z_q; \theta_q) = (y, z; \theta)$, for an arbitrary small positive real number ϵ, we can find a positive integer q such that

$$\left\|y_q - y\right\| + \left\|z_q - z\right\| + dist(\theta, \theta_q) < \epsilon,$$

where $dist(\theta, \theta') = \sum_{i=0}^{p}\left|\mu_i - \mu_i'\right| + \sum_{i=0}^{p+H-1}\left\|w_i - w_i'\right\| + \sum_{i=0}^{p+1}\left|\delta_i - \delta_i'\right|$. Based on $x_{-H+1:p+H-1} := \psi(y, z; \theta)$, we construct a feasible solution $x'_{-H+1:p+H-1} =: x'$ for the optimization problem (7) with parameters $(y_q, z_q; \theta_q)$ as following: Let $x'_{0:p} = x_{0:p}, x_{-H+1:-1} = y, x_{p+1:p+H-1} = z$. For $t = 0, 1, \ldots$, if $x'_t - x'_{t-1} < -\delta_t^{(q)}$, we increase x'_t such that $x'_t = x'_{t-1} - \delta_t^{(q)}$. Then, for $t = p, p-1, \ldots$, if $x'_{t+1} - x'_t < -\delta_{t+1}^{(q)}$, we decrease x'_t such that $x'_t = x'_{t+1} + \delta_{t+1}^{(q)}$. Note that this procedure can guarantee that x' is a feasible solution for (7), and their distance are upper bounded by

$$\left\|\psi(y, z; \theta) - x'\right\| \le (2d+1)\epsilon. \tag{15}$$

Since the objective function of (7) is Lipschitz in (x, y, z, θ), by (15), we know there exists some positive constant c_0 such that

$$\iota(y_q, z_q; \theta_q) - \iota(y, z; \theta) \le c_0\left(\left\|x' - \psi(y, z; \theta)\right\| + \epsilon\right) \le (2d+2)c_0\epsilon. \tag{16}$$

On the other hand, by Lemma B.3, we see that

$$\left\|\hat{\psi}(y_q, z_q; \theta_q; \xi, \sigma) - \hat{\psi}(y, z; \theta; \xi, \sigma)\right\| \le \left(\frac{C}{1-\rho} + 1\right)\epsilon. \tag{17}$$

Since the objective function of (7) is smooth in (x, y, z, θ), by (17), we see that

$$\left|\hat{\iota}(y_q, z_q; \theta_q; \xi, \sigma) - \hat{\iota}(y, z; \theta; \xi, \sigma)\right| \le c_0\left(\frac{C}{1-\rho} + 2\right)\epsilon. \tag{18}$$

Therefore, we see that

$$\begin{aligned}\hat{\iota}(y_q, z_q; \theta_q; \xi, \sigma) - \iota(y_q, z_q; \theta_q) &\ge -\left|\hat{\iota}(y_q, z_q; \theta_q; \xi, \sigma) - \hat{\iota}(y, z; \theta; \xi, \sigma)\right| + (\hat{\iota}(y, z; \theta; \xi, \sigma) - \iota(y, z; \theta)) + (\iota(y, z; \theta) - \iota(y_q, z_q; \theta_q)) \\ &\ge -c_0\left(\frac{C}{1-\rho} + 2\right)\epsilon + \Lambda - c_0(2d+2)\epsilon \qquad (19a) \\ &= \Lambda - c_0\left(\frac{C}{1-\rho} + 2d + 4\right)\epsilon,\end{aligned}$$

where we used (16) and (18) in (19a). Let $\epsilon := \frac{1}{2}\Lambda c_0^{-1}\left(\frac{C}{1-\rho} + 2d + 4\right)^{-1}$ leads to a contradiction with the assumption that $\hat{\iota}(y_q, z_q; \theta_q; \xi, \sigma) = \iota(y_q, z_q; \theta_q)$. Therefore, we have shown that $\psi(y, z; \theta) = \hat{\psi}(y, z; \theta; \xi, \sigma)$. □

With the above technical lemmas, we are ready to finish the proof of Theorem B.1.

Proof of Theorem B.1. Consider the segment $((1-\eta)y+\eta y', (1-\eta)z+\eta z'; (1-\eta)\theta+\eta\theta'), \eta \in [0,1]$. Note that since $(1-\eta)\psi(y,z;\theta)+\eta\psi(y',z';\theta')$ is a feasible solution for the optimization problem (7) parameterized by

$$\left((1-\eta)y+\eta y', (1-\eta)z+\eta z'; (1-\eta)\theta+\eta\theta'\right),$$

we know that the corresponding optimization problem is feasible. With some slight abuse of notation, we use $(\xi,\sigma)(\eta) \subseteq \Xi \times \Sigma$ to denote the set of indicators of active constraints and sides such that

$$\begin{aligned}&\psi\left((1-\eta)y+\eta y', (1-\eta)z+\eta z'; (1-\eta)\theta+\eta\theta'\right)\\ =&\hat{\psi}\left((1-\eta)y+\eta y', (1-\eta)z+\eta z'; (1-\eta)\theta+\eta\theta'; \xi,\sigma\right), \forall (\xi,\sigma) \in (\xi,\sigma)(\eta).\end{aligned}$$

By Lemma B.2, we know this set is not empty for any $\eta \in [0,1]$.

We can divide the interval $[0,1]$ into $0=\eta_0<\eta_1<\ldots<\eta_q=1$ for some positive integer $q \le 2^{5p+6}$ such that there exists a sequence of *different* indicators of active constraints and sides $(\xi,\sigma)_{0:q-1}$ which satisfies

$$\begin{aligned}\psi\left((1-\eta_i)(y,z;\theta)+\eta_i(y',z';\theta')\right) &= \hat{\psi}\left((1-\eta_i)(y,z;\theta)+\eta_i(y',z';\theta'); (\xi,\sigma)_i\right),\\ \psi\left((1-\eta_{i+1})(y,z;\theta)+\eta_{i+1}(y',z';\theta')\right) &= \hat{\psi}\left((1-\eta_{i+1})(y,z;\theta)+\eta_{i+1}(y',z';\theta'); (\xi,\sigma)_i\right)\end{aligned}$$

for all $0 \le i \le q-1$. Note that this requires $(\xi,\sigma)(\eta_i)$ to contain both $(\xi,\sigma)_{i-1}$ and $(\xi,\sigma)_i$ for $i=1,\ldots,q-1$. To construct the sequence $\eta_{0:q}$ and $(\xi,\sigma)_{0:q-1}$, we first have $\eta_0=0$ and let $(\xi,\sigma)_0$ be any pair $(\xi,\sigma) \in (\xi,\sigma)(\eta_0)$ such that

$$\sup\{\eta \in [0,1] \mid \psi\left((1-\eta)(y,z;\theta)+\eta(y',z';\theta')\right) = \hat{\psi}\left((1-\eta)(y,z;\theta)+\eta(y',z';\theta'); \xi,\sigma\right)\} > 0,$$

and let η_1 be the supremum value above. Since $0=\inf(0,1]$ and $(\xi,\sigma)(\eta) \subseteq \Xi\times\Sigma$ is nonempty for every $\eta \in (0,1]$, we know such $(\xi,\sigma)_0$ exists by Lemma B.4. Suppose we have already constructed $\eta_{0:i}$, $(\xi,\sigma)_{0:i-1}$, and $\eta_i<1$. Then we select $(\xi,\sigma)_i$ to be any pair (ξ,σ) such that

$$\sup\{\eta \in [0,1] \mid \psi\left((1-\eta)(y,z;\theta)+\eta(y',z';\theta')\right) = \hat{\psi}\left((1-\eta)(y,z;\theta)+\eta(y',z';\theta'); \xi,\sigma\right)\} > \eta_i,$$

and let η_{i+1} be the supremum value above. We can repeat this construction and stop when $\eta_{i+1}=1$. By the construction, we know all pairs in the sequence $(\xi,\sigma)_{0:i-1}$ are distinct, thus the construction will terminate in finite time. Hence, we have a finite index q such that $\eta_q=1$.

By Lemma B.3, we know that

$$\begin{aligned}&\left\|\psi\left((1-\eta_i)(y,z;\theta)+\eta_i(y',z';\theta')\right)_t - \psi\left((1-\eta_{i+1})(y,z;\theta)+\eta_{i+1}(y',z';\theta')\right)_t\right\|\\ \le\ &(\eta_{i+1}-\eta_i)C\left(\rho^t\left\|y-y'\right\| + \rho^{p-t}\left\|z-z'\right\|\right) + (\eta_{i+1}-\eta_i)C\left(\sum_{\tau=0}^{p}\rho^{|t-\tau|}\left|\mu_\tau-\mu'_\tau\right| + \sum_{\tau=0}^{p+H-1}\rho^{|t-\tau|}\left\|w_\tau-w'_\tau\right\| + \sum_{\tau=0}^{p+1}\rho^{|t-\tau|}\left\|\delta_\tau-\delta'_\tau\right\|\right). \quad (20)\end{aligned}$$

Summing (20) over $i=0,1,\ldots,q-1$ finishes the proof. □

C PROOFS FOR EXACT PREDICTIONS

C.1 Proof of Lemma A.4

To simplify the notation, we introduce the shorthand

$$x^*_{\tau|t} = \psi_t^N\left((x_{t-1},u_{t-1}); \omega_{t:N}; \mathbf{0}\right)_{x_\tau}, u^*_{\tau|t} = \psi_t^N\left((x_{t-1},u_{t-1}); \omega_{t:N}; \mathbf{0}\right)_{u_\tau}, \forall \tau \ge t.$$

And we use $\{(x_t^*,u_t^*)\}_{t=1}^N$ to denote the offline optimal trajectory.

For time step $t < N-K+1$, we see that

$$\begin{aligned}&\left|x_t - \psi_t^N\left((x_{t-1},u_{t-1}); \omega_{t:N}; \mathbf{0}\right)_{x_t}\right|^2\\ \le\ &\left(C\rho^K\left|x^*_{t+K|t}-\bar{x}\right| + C\rho^{K-1}\left|u^*_{t+K-1|t} - \frac{1}{\omega_{t+K-1}}\right|\right)^2 \quad (21a)\\ \le\ &\left(C\rho^K\left(\left|x^*_{t+K|t}-x^*_{t+K}\right| + \left|x^*_{t+K}-\bar{x}\right|\right) + C\rho^{K-1}\left(\left|u^*_{t+K-1|t}-u^*_{t+K-1}\right| + \left|u^*_{t+K-1} - \frac{1}{\omega_{t+K-1}}\right|\right)\right)^2 \quad (21b)\\ \le\ &4C^2\rho^{2K}\left|x^*_{t+K|t}-x^*_{t+K}\right|^2 + 4C^2\rho^{2K-2}\left|u^*_{t+K-1|t}-u^*_{t+K-1}\right|^2 + 4C^2\rho^{2K}\left|x^*_{t+K-1}-\bar{x}\right|^2 +\\ &+ 4C^2\rho^{2K-2}\left|u^*_{t+K-1} - \frac{1}{\omega_{t+K-1}}\right|^2. \quad (21c)\end{aligned}$$

where in (21a), we use

$$x_t = \psi_t^{t+K-1}\left((x_{t-1},u_{t-1});\omega_{t:t+K-1};(\bar{x},\frac{1}{\omega_{t+K-1}})\right)_{x_t},$$
$$\psi_t^N\left((x_{t-1},u_{t-1});\omega_{t:N};\mathbf{0}\right)_{x_t} = \psi_t^{t+K-1}\left((x_{t-1},u_{t-1});\omega_{t:t+K-1};(x^*_{t+K|t},u^*_{t+K-1|t})\right)_{x_t},$$

and the exponentially decaying perturbation bound. We use the triangle inequality in (21b) and rearrange the terms in (21c).

We note that for the first term in (21), we have

$$\left|x^*_{t+K|t}-x^*_{t+K}\right| \le C\rho^{K+1}\left(\left|x_{t-1}-x^*_{t-1}\right|+\left|u_{t-1}-u^*_{t-1}\right|\right). \tag{22}$$

For the second term, we have

$$\left|u^*_{t+K-1|t}-u^*_{t+K-1}\right| \le C'\rho^{K}\left(\left|x_{t-1}-x^*_{t-1}\right|+\left|u_{t-1}-u^*_{t-1}\right|\right). \tag{23}$$

For the third term, we have

$$\left|x^*_{t+K}-\bar{x}\right|^2 \le \frac{1}{\epsilon\beta}b(x^*_{t+K}). \tag{24}$$

For the last term, we see that

$$\left|u^*_{t+K-1}-\frac{1}{\omega_{t+K-1}}\right|^2 \le \frac{(x^*_{t+K-1}-x^*_{t+K-2})^2}{\omega^2_{t+K-1}} \le \frac{2(x^*_{t+K-1}-\bar{x})^2+2(\bar{x}-x^*_{t+K-2})^2}{\omega^2_{t+K-1}} \le \frac{2b(x^*_{t+K-1})+2b(x^*_{t+K-2})}{\epsilon\beta\omega^2_{\min}}. \tag{25}$$

Substituting (22), (23), (24), (25) into (21) gives that

$$\begin{aligned}\left|x_t-\psi_t^N\left((x_{t-1},u_{t-1});\omega_{t:N};\mathbf{0}\right)_{x_t}\right|^2 &\le 8C^4\rho^{4K+2}\left(\left|x_{t-1}-x^*_{t-1}\right|^2+\left|u_{t-1}-u^*_{t-1}\right|^2\right)+8(C')^2C^2\rho^{4K-2}\left(\left|x_{t-1}-x^*_{t-1}\right|^2+\left|u_{t-1}-u^*_{t-1}\right|^2\right)\\ &\quad+4C^2\rho^{2K-2}\frac{(2+\omega^2_{\min})b(x^*_{t+K-1})+2b(x^*_{t+K-2})}{\epsilon\beta\omega^2_{\min}}\end{aligned} \tag{26}$$

Similarly, we can obtain that

$$\begin{aligned}\left|u_t-\psi_t^N\left((x_{t-1},u_{t-1});\omega_{t:N};\mathbf{0}\right)_{u_t}\right|^2 &\le 8(C')^2C^2\rho^{4K+2}\left(\left|x_{t-1}-x^*_{t-1}\right|^2+\left|u_{t-1}-u^*_{t-1}\right|^2\right)+8(C')^4\rho^{4K-2}\left(\left|x_{t-1}-x^*_{t-1}\right|^2+\left|u_{t-1}-u^*_{t-1}\right|^2\right)\\ &\quad+4(C')^2\rho^{2K-2}\frac{(2+\omega^2_{\min})b(x^*_{t+K-1})+2b(x^*_{t+K-2})}{\epsilon\beta\omega^2_{\min}}\end{aligned} \tag{27}$$

Therefore, combining (26) and (27) gives that

$$\begin{aligned}e_t^2 &\le 2\left|x_t-\psi_t^N\left((x_{t-1},u_{t-1});\omega_{t:N};\mathbf{0}\right)_{x_t}\right|^2+2\left|u_t-\psi_t^N\left((x_{t-1},u_{t-1});\omega_{t:N};\mathbf{0}\right)_{u_t}\right|^2\\ &\le 16\rho^{4K-2}\left(C^2+(C')^2\right)^2\left(\left|x_{t-1}-x^*_{t-1}\right|^2+\left|u_{t-1}-u^*_{t-1}\right|^2\right)+8\rho^{2K-2}\left(C^2+(C')^2\right)\frac{(2+\omega^2_{\min})b(x^*_{t+K-1})+2b(x^*_{t+K-2})}{\epsilon\omega^2_{\min}}.\end{aligned}$$

C.2 Proof of Lemma A.5

We see the distance between the trajectories of SODA and the offline optimal at an intermediate time step can be bounded by

$$\begin{aligned}\left|x_t-x^*_t\right|+\left|u_t-u^*_t\right| &= \left|x_t-\psi_1^N\left((x_0,u_0);\omega_{1:N};\mathbf{0}\right)_{x_t}\right|+\left|u_t-\psi_1^N\left((x_0,u_0);\omega_{1:N};\mathbf{0}\right)_{u_t}\right|\\ &\le \left|x_t-\psi_t^N\left((x_{t-1},u_{t-1});\omega_{t:N};\mathbf{0}\right)_{x_t}\right|+\left|u_t-\psi_t^N\left((x_{t-1},u_{t-1});\omega_{t:N};\mathbf{0}\right)_{u_t}\right|\\ &\quad+\sum_{\tau=1}^{t-1}\left|\psi_\tau^N\left((x_{\tau-1},u_{\tau-1});\omega_{\tau:N};\mathbf{0}\right)_{x_t}-\psi_{\tau+1}^N\left((x_\tau,u_\tau);\omega_{\tau+1:N};\mathbf{0}\right)_{x_t}\right|\\ &\quad+\sum_{\tau=1}^{t-1}\left|\psi_\tau^N\left((x_{\tau-1},u_{\tau-1});\omega_{\tau:N};\mathbf{0}\right)_{u_t}-\psi_{\tau+1}^N\left((x_\tau,u_\tau);\omega_{\tau+1:N};\mathbf{0}\right)_{u_t}\right| &\text{(28a)}\\ &\le e_t+(C+C')\sum_{\tau=1}^{t-1}\rho^{t-\tau}e_\tau. &\text{(28b)}\end{aligned}$$

We use the triangle inequality in (28a). In (28b), we note that $\psi_\tau^N\left((x_{\tau-1},u_{\tau-1});\omega_{\tau:N};\mathbf{0}\right)_{x_t}$ can be written as $\psi_{\tau+1}^N\left((x^*_{\tau|\tau-1},u^*_{\tau|\tau-1});\omega_{\tau+1:N};\mathbf{0}\right)_{x_t}$, where

$$x^*_{\tau|\tau-1}=\psi_\tau^N\left((x_{\tau-1},u_{\tau-1});\omega_{\tau:N};\mathbf{0}\right)_{x_\tau},\ \text{and}\ u^*_{\tau|\tau-1}=\psi_\tau^N\left((x_{\tau-1},u_{\tau-1});\omega_{\tau:N};\mathbf{0}\right)_{u_\tau}.$$

Thus, we can apply the exponentially decaying perturbation bound and Lemma A.4 to obtain

$$\left|\psi_\tau^N\left((x_{\tau-1},u_{\tau-1});\omega_{\tau:N};\mathbf{0}\right)_{x_t}-\psi_{\tau+1}^N\left((x_\tau,u_\tau);\omega_{\tau+1:N};\mathbf{0}\right)_{x_t}\right|\le C\rho^{t-\tau}e_\tau.$$

Similarly, we obtain that

$$\left|\psi_\tau^N\left((x_{\tau-1},u_{\tau-1});\omega_{\tau:N};\mathbf{0}\right)_{u_t}-\psi_{\tau+1}^N\left((x_\tau,u_\tau);\omega_{\tau+1:N};\mathbf{0}\right)_{u_t}\right|\le C'\rho^{t-\tau}e_\tau.$$

Therefore, we see that

$$\left|x_t-x_t^*\right|^2+\left|u_t-u_t^*\right|^2\le\left(1+\frac{(C+C')^2}{1-\rho}\right)\sum_{\tau=1}^t\rho^{t-\tau}e_\tau^2.$$

Summing the above inequality over $t=1,2,\ldots,T$ gives that

$$\sum_{t=1}^N\left(\left|x_t-x_t^*\right|^2+\left|u_t-u_t^*\right|^2\right)\le\frac{1}{1-\rho}\cdot\left(1+\frac{(C+C')^2}{1-\rho}\right)\sum_{t=1}^N e_t^2.$$

C.3 Proof of Theorem A.3

Combining Lemmas A.4 and A.5, we see that

$$\begin{aligned}\sum_{t=1}^N\left(\left|x_t-x_t^*\right|^2+\left|u_t-u_t^*\right|^2\right)\le{}&\frac{1}{1-\rho}\cdot\left(1+\frac{(C+C')^2}{1-\rho}\right)\cdot16\rho^{4K-2}\left(C^2+(C')^2\right)^2\sum_{t=1}^N\left(\left|x_{t-1}-x_{t-1}^*\right|^2+\left|u_{t-1}-u_{t-1}^*\right|^2\right)\\&+\frac{1}{1-\rho}\cdot\left(1+\frac{(C+C')^2}{1-\rho}\right)\cdot8\rho^{2K-2}\left(C^2+(C')^2\right)\frac{(4+\omega_{\min}^2)\sum_{t=1}^N b(x_t^*)}{\epsilon\beta\omega_{\min}^2}.\end{aligned}\tag{29}$$

Since the prediction horizon K satisfies

$$K\ge\frac{1}{4}\ln\left(\frac{16}{1-\rho}\cdot\left(1+\frac{(C+C')^2}{1-\rho}\right)\cdot\left(C^2+(C')^2\right)^2\right)/\ln\left(\frac{1}{\rho}\right),$$

we see that

$$\sum_{t=1}^N\left(\left|x_t-x_t^*\right|^2+\left|u_t-u_t^*\right|^2\right)\le\frac{16\rho^{2K-2}}{1-\rho}\cdot\left(1+\frac{(C+C')^2}{1-\rho}\right)\left(C^2+(C')^2\right)\frac{(4+\omega_{\min}^2)\sum_{t=1}^N b(x_t^*)}{\epsilon\omega_{\min}^2}.\tag{30}$$

On the other hand, we also see that for any $\eta>0$, we have

$$\begin{aligned}\mathrm{cost}(\mathtt{SODA})={}&\sum_{t=1}^N\omega_t u_t^2+b(x_t)+\gamma(u_t-u_{t-1})^2\\={}&\sum_{t=1}^N\omega_t(u_t^*+(u_t-u_t^*))^2+\sum_{t=1}^N b(x_t^*+(x_t-x_t^*))\\&+\sum_{t=1}^N\gamma(u_t^*-u_{t-1}^*+(u_t-u_t^*)-(u_{t-1}-u_{t-1}^*))^2\\\le{}&(1+\eta)\sum_{t=1}^N\left(\omega_t(u_t^*)^2+b(x_t^*)+\gamma(u_t^*-u_{t-1}^*)^2\right)\\&+\left(1+\frac{1}{\eta}\right)\sum_{t=1}^N\left(\omega_t(u_t^*-u_t)^2+\beta(x_t^*-x_t)^2+2\gamma(u_t^*-u_t)^2+2\gamma(u_{t-1}^*-u_{t-1})^2\right)\qquad(31a)\\\le{}&(1+\eta)\mathrm{cost}(\mathtt{OPT})+\left(1+\frac{1}{\eta}\right)(4\gamma+\beta+\omega_{\max})\sum_{t=1}^N\left(\left|x_t-x_t^*\right|^2+\left|u_t-u_t^*\right|^2\right),\qquad(31b)\end{aligned}$$

where we use the quadratic form of the cost functions and the AM-GM inequality in (31a); we use (30) in (31b).

Substituting (30) into (31) gives that

$$\mathrm{cost}(\mathtt{SODA}) - \mathrm{cost}(\mathtt{OPT}) \le \left(\eta + \left(1+\frac{1}{\eta}\right)(4\gamma+\beta+\omega_{\max}) \cdot \frac{16\rho^{2K-2}}{1-\rho} \cdot \left(1+\frac{(C+C')^2}{1-\rho}\right)\left(C^2+(C')^2\right) \cdot \frac{4+\omega_{\min}^2}{\epsilon\beta\omega_{\min}^2}\right)\mathrm{cost}(\mathtt{OPT}).$$

Letting $\eta = 4\left(2(4\gamma+\beta+\omega_{\max}) \cdot \frac{1}{1-\rho} \cdot \left(1+\frac{(C+C')^2}{1-\rho}\right)\left(C^2+(C')^2\right) \cdot \frac{4+\omega_{\min}^2}{\epsilon\beta\omega_{\min}^2}\right)^{1/2}$ finishes the proof.

D PROOFS FOR INEXACT PREDICTIONS

D.1 Proof of Lemma A.6

Suppose $\{x_t\}_{1\le t\le N}$ is a feasible trajectory of the buffer levels and $\{x_t\}_{t_1\le t\le t_2}$ is a sub-trajectory such that $x_{t_1-1} \ge \bar{x}-\zeta, x_t < \bar{x}-\zeta, \forall t = t_1,\dots,t_2$, and $x_{t_2+1} \ge \bar{x}-\zeta$ where $1 \le t_1 < t_2 < N$.

For $\lambda \ge 0$, consider the trajectory $\{x'_t(\lambda)\}_{1\le t\le N}$ constructed by

$$x'_t(\lambda) = \begin{cases} x_t, & \text{if } t < t_1 \text{ or } t > t_2 \\ x_t + \lambda, & \text{otherwise.} \end{cases}$$

Note that under this construction, $\{x'_t(0)\}_{1\le t\le N}$ is identical with the original trajectory $\{x_t\}_{1\le t\le N}$. Let $\Upsilon(\lambda)$ denote the total cost of this trajectory. For sufficiently small $\lambda \ge 0$, we see that

$$\begin{aligned}\Upsilon(\lambda) - \Upsilon(0) = \beta\sum_{t=t_1}^{t_2}\left((x_t+\lambda-\bar{x})^2-(x_t-\bar{x})^2\right) &+ \omega_{t_1}\left(u'_{t_1}(\lambda)^2-u_{t_1}^2\right) + \omega_{t_2+1}\left(u'_{t_2+1}(\lambda)^2-u_{t_2+1}^2\right)\\ &+\gamma\left((u'_{t_1}(\lambda)-u_{t_1-1})^2-(u_{t_1}-u_{t_1-1})^2\right)+\gamma\left((u_{t_1+1}-u'_{t_1}(\lambda))^2-(u_{t_1+1}-u_{t_1})^2\right)\\ &+\gamma\left((u'_{t_2+1}(\lambda)-u_{t_2})^2-(u_{t_2+1}-u_{t_2})^2\right)+\gamma\left((u_{t_2+2}-u'_{t_2+1}(\lambda))^2-(u_{t_2+2}-u_{t_2+1})^2\right),\end{aligned}$$

where $u'_{t_1}(\lambda) = u_{t_1} + \frac{\lambda}{\omega_{t_1}}$ and $u'_{t_2+1}(\lambda) = u_{t_2+1} - \frac{\lambda}{\omega_{t_2+1}}$.

Therefore, we see that

$$\begin{aligned}\frac{d}{d\lambda}\Upsilon(\lambda)\Big|_{\lambda=0^+} &= \frac{d}{d\lambda}\left(\Upsilon(\lambda)-\Upsilon(0)\right)\Big|_{\lambda=0^+}\\ &= 2\beta\sum_{t=t_1}^{t_2}(x_t-\bar{x}) + 2u_{t_1} - 2u_{t_2+1} + \frac{2\gamma}{\omega_{t_1}}(2u_{t_1}-u_{t_1-1}-u_{t_1+1}) + \frac{2\gamma}{\omega_{t_2+1}}(-2u_{t_2+1}+u_{t_2}+u_{t_2+2})\\ &< -2\beta\zeta + \left(2+\frac{8\gamma}{\omega_{\min}}\right)\left(\frac{1}{r_{\min}}-\frac{1}{r_{\max}}\right) \le 0.\end{aligned}$$

Thus, we know that there exists $\lambda > 0$ such that $\{x'_t(\lambda)\}_{1\le t\le N}$ is feasible and $\Upsilon(\lambda)$ is less than the total cost of $\{x_t\}_{1\le t\le N}$. Therefore, the offline optimal trajectory cannot contain a sub-trajectory such that $x_{t_1-1} \ge \bar{x}-\zeta, x_t < \bar{x}-\zeta, \forall t = t_1,\dots,t_2$, and $x_{t_2+1} \ge \bar{x}-\zeta$ where $1 \le t_1 < t_2 < N$. Using similar techniques, we can extend this claim to include $t_2 = N$ and/or $t_1 = t_2$. Thus, the buffer levels in the offline optimal trajectory do not go below $\bar{x}-\zeta$. By symmetry, we can show that the offline optimal trajectory also does not exceed $\bar{x}+\zeta$.

D.2 Proof of Lemma A.7

To simplify the notation, we introduce the shorthand

$$x^*_{\tau|t} = \psi_t^N\left((x_{t-1},u_{t-1});\omega_{t:N};\mathbf{0}\right)_{x_\tau}, u^*_{\tau|t} = \psi_t^N\left((x_{t-1},u_{t-1});\omega_{t:N};\mathbf{0}\right)_{u_\tau}, \forall\tau \ge t.$$

And we use $\{(x_t^*,u_t^*)\}$ to denote the offline optimal trajectory.

For time step $t < N-K+1$, we see that

$$\left|x_t - \psi_t^N\left((x_{t-1},u_{t-1});\omega_{t:N};\mathbf{0}\right)_{x_t}\right| \le C\rho^K\left|x^*_{t+K|t}-\bar{x}\right| + C\rho^{K-1}\left|u^*_{t+K-1|t}-\frac{1}{\omega_{t+K-1}}\right| + C\sum_{\tau=t}^{t+K-1}\rho^{\tau-t}\left|\hat{\omega}_{\tau|t-1}-\omega_\tau\right| + \frac{\left|\omega_t-\hat{\omega}_{t|t-1}\right|}{r_{\min}} \tag{32a}$$

$$\le C\rho^K\left(x_{\max}+\frac{1}{r_{\min}}-\frac{1}{r_{\max}}\right) + C\cdot E(t-1,K) + \frac{\left|\omega_t-\hat{\omega}_{t|t-1}\right|}{r_{\min}}, \tag{32b}$$

where in (32a), we use the facts that

$$x_t = \psi_t^{t+K-1}\left((x_{t-1}, u_{t-1}); \hat{\omega}_{t:t+K-1|t-1}; (\bar{x}, \frac{1}{\omega_{t+K-1}})\right)_{x_t} + (\omega_t - \hat{\omega}_{t|t-1})u_t,$$
$$\psi_t^N\left((x_{t-1}, u_{t-1}); \omega_{t:N}; \mathbf{0}\right)_{x_t} = \psi_t^N\left((x_{t-1}, u_{t-1}); \omega_{t:t+K-1}; (x^*_{t+K|t}, u^*_{t+K-1|t})\right)_{x_t},$$

and apply the exponentially decaying perturbation bound. In (32b), we apply the worst-case bound for the first two terms and use the definition of $E_{t-1}(K)$.

Note that we can show (32) also holds for $t \geq N - K + 1$ with the same approach.

Similarly, we can show that

$$\left|u_t - \psi_t^N\left((x_{t-1}, u_{t-1}); \omega_{t:N}; \mathbf{0}\right)_{u_t}\right| \leq C'\rho^K\left(x_{\max} + \frac{1}{r_{\min}} - \frac{1}{r_{\max}}\right) + C' \cdot E(t-1, K). \tag{33}$$

Combining (32) and (33) finishes the proof of Lemma A.7.

D.3 Proof of Theorem A.8

We first use induction to show that SODA's entire trajectory satisfies the buffer level constraints strictly. To see this, note that for $t = 1$, we have

$$\left|x_1 - x_1^*\right| \leq e_1 \leq C\rho^K\left(x_{\max} + \frac{1}{r_{\min}} - \frac{1}{r_{\max}}\right) + C \cdot E(t-1, K) + \frac{\left|\omega_t - \hat{\omega}_{t|t-1}\right|}{r_{\min}} \leq \frac{D}{3}.$$

By Lemma A.6, we know that $\left|x_1^* - \bar{x}\right| \leq \frac{D}{3}$. Thus, we have

$$|x_1 - \bar{x}| \leq \left|x_1 - x_1^*\right| + \left|x_1^* - \bar{x}\right| \leq \frac{2D}{3}.$$

Therefore, we see that $0 < x_1 < x_{\max}$. Supposing that $0 < x_\tau < x_{\max}$ holds for $\tau = 1, \ldots, t-1$, we see that

$$\left|x_t - x_t^*\right| + \left|u_t - u_t^*\right| \leq e_t + (C + C')\sum_{\tau=1}^{t-1}\rho^{t-\tau}e_\tau \tag{34a}$$
$$\leq \frac{(1 + C + C')^2}{1-\rho}\left(x_{\max} + \frac{1}{r_{\min}} - \frac{1}{r_{\max}}\right) \cdot \rho^K + \left(1 + \frac{1}{r_{\min}} + C + C'\right)^2 \sum_{\tau=1}^{t}\rho^{t-\tau}E(\tau - 1, K), \tag{34b}$$

In (34a), we use (28) in the proof of Lemma A.5. We use Lemma A.7 in (34b).

Thus, we obtain that $\left|x_t^* - x_t\right| \leq \frac{D}{3}$. By Lemma A.6, we see that

$$|x_t - \bar{x}| \leq \left|x_t - x_t^*\right| + \left|x_t^* - \bar{x}\right| \leq \frac{2D}{3}.$$

Therefore, we have shown that $0 < x_t < x_{\max}$ holds for all time steps t by induction.

By (34), we see that

$$\left|x_t - x_t^*\right|^2 + \left|u_t - u_t^*\right|^2 \leq \frac{\left(1 + \frac{1}{r_{\min}} + C + C'\right)^4}{1-\rho}\left(1 + x_{\max} + \frac{1}{r_{\min}} - \frac{1}{r_{\max}}\right) \cdot \left(\frac{1}{1-\rho}\left(x_{\max} + \frac{1}{r_{\min}} - \frac{1}{r_{\max}}\right) \cdot \rho^{2K} + \sum_{\tau=1}^{t}\rho^{t-\tau}E(\tau - 1, K)^2\right). \tag{35}$$

Therefore, by summing (35) over t, we obtain that

$$\sum_{t=1}^{N}\left(\left|x_t - x_t^*\right|^2 + \left|u_t - u_t^*\right|^2\right) \leq \frac{\left(1 + \frac{1}{r_{\min}} + C + C'\right)^4}{(1-\rho)^2}\left(1 + x_{\max} + \frac{1}{r_{\min}} - \frac{1}{r_{\max}}\right) \cdot \left(\left(x_{\max} + \frac{1}{r_{\min}} - \frac{1}{r_{\max}}\right) \cdot N\rho^{2K} + \sum_{t=0}^{N-1}E(t, K)^2\right). \tag{36}$$

By (31), we see that for any $\eta > 0$, we have

$$\begin{aligned}\text{cost}(\texttt{SODA}) &\le (1+\eta)\text{cost}(\texttt{OPT}) + \left(1+\frac{1}{\eta}\right)(4\gamma+\beta+\omega_{\max})\sum_{t=1}^{N}\left(\left|x_t - x_t^*\right|^2 + \left|u_t - u_t^*\right|^2\right)\\ &\le (1+\eta)\text{cost}(\texttt{OPT}) + \left(1+\frac{1}{\eta}\right)(4\gamma+\beta+\omega_{\max})\cdot\\ &\quad \frac{\left(1+\frac{1}{r_{\min}}+C+C'\right)^4}{(1-\rho)^2}\left(1+x_{\max}+\frac{1}{r_{\min}}-\frac{1}{r_{\max}}\right)\cdot\\ &\quad \left(\left(x_{\max}+\frac{1}{r_{\min}}-\frac{1}{r_{\max}}\right)\cdot N\rho^{2K} + \sum_{t=0}^{N-1}E(t,K)^2\right).\end{aligned}$$

Note that $N\rho^{2K} + \sum_{t=0}^{N-1} E(t,K)^2 \le \frac{1}{1-\rho}\mathcal{E}$. Setting

$$\eta = \frac{\left(1+\frac{1}{r_{\min}}+C+C'\right)^2\left(1+x_{\max}+\frac{1}{r_{\min}}-\frac{1}{r_{\max}}\right)}{(1-\rho)^{3/2}}\cdot\sqrt{4\gamma+\beta+\omega_{\max}}\cdot\sqrt{\frac{\mathcal{E}}{\text{cost}(\texttt{OPT})}}$$

finishes the proof.

E PROOFS FOR EFFICIENT STRUCTURES

E.1 Proof of Lemma A.10

We first consider the case when $v_{t-1} > 1/\hat{\omega}$. To simplify the notation, we use $\check{u}_{t:t+K-1}$ to denote the sequence of control actions in $\hat{\phi}_t^{t+K-1}((\sigma_{t-1}, v_{t-1}); \hat{\omega}; \mathbf{0})$.

We first show that $\check{u}_\tau \ge 1/\hat{\omega}$ for all $\tau \in \{t, \dots, t+K-1\}$. For the sake of contradiction, let $\check{u}_{t_1}$ be the first action such that $u_{t_1-1} \ge 1/\hat{\omega}$ and $\check{u}_{t_1} < 1/\hat{\omega}$. Note that resetting the sequence $\check{u}_{t_1:t+K-1}$ to $u_{t_1} = u_{t_1+1} = \cdots = u_{t+K-1} = 1/\hat{\omega}$ will strictly decrease the total cost and the whole sequence remains feasible. This contradicts with the optimality of $\check{u}_{t:t+K-1}$. Thus, we have $\check{u}_\tau \ge 1/\hat{\omega}$ for all $\tau \in \{t, \dots, t+K-1\}$.

We next show that $\check{u}_\tau \le v_{t-1}$ for all $\tau \in \{t, \dots, t+K-1\}$. To see this, for all u_τ such that $u_\tau > v_{t-1}$, we can reset them to $u_\tau = v_{t-1}$ to decrease the total switching cost strictly without violating any feasibility constraints.

Since $\check{u}_\tau \in [1/\hat{\omega}, v_{t-1}]$ for all $\tau \in \{t, \dots, t+K-1\}$, we know that the buffer level sequence is monotonically increasing. Thus, if $\check{u}_{t:t+K-1}$ is not monotonically decreasing, we can permute it to make it monotonically decreasing. This change will strictly decrease the total switching cost without violating any feasibility constraints. Therefore, we have shown Theorem A.10 holds for the case $v_{t-1} > 1/\hat{\omega}$.

Using similar techniques, we can show Lemma A.10 also holds for the case $v_{t-1} < 1/\hat{\omega}$ and $v_{t-1} = 1/\hat{\omega}$.

E.2 Proof of Theorem A.9

We can rewrite the optimization problem (6) as

$$\begin{aligned}\min_{a_{t:t+K-1}} \quad & \sum_{\tau=t}^{t+K-1} \gamma \cdot a_t^2\\ \text{s.t. } & x_\tau = x_{\tau-1} + \hat{\omega}u_\tau - 1, \text{ for } \tau = t, \dots, t+K-1,\\ & u_\tau = u_{\tau-1} + a_\tau, \text{ for } \tau = t, \dots, t+K-1,\\ & x_\tau \in [0, x_{\max}], u_\tau \in \left[\frac{1}{r_{\max}}, \frac{1}{r_{\min}}\right], \text{ for } \tau = t, \dots, t+K-1,\\ & x_{t-1} = \sigma_{t-1}, u_{t-1} = v_{t-1}.\end{aligned} \tag{37}$$

We use $\{(\check{a}_\tau, \check{u}_\tau, \check{x}_\tau)\}_{\tau=t,\dots,t+K-1}$ to denote the optimal solution of (37).

Similarly, we can rewrite the optimization problem $\hat{\psi}_t^{t+K-1}\left((\sigma_{t-1}, \nu_{t-1}); \hat{\omega}; \mathbf{0}\right)$ as

$$\begin{aligned}
\min_{a_{t:t+K-1}} \quad & \sum_{\tau=t}^{t+K-1} \gamma \cdot a_t^2 + \hat{\omega} u_t^2 + \beta b(x_t) \\
\text{s.t. } & x_\tau = x_{\tau-1} + \hat{\omega} u_\tau - 1, \text{ for } \tau = t, \ldots, t+K-1, \\
& u_\tau = u_{\tau-1} + a_\tau, \text{ for } \tau = t, \ldots, t+K-1, \\
& x_\tau \in [0, x_{\max}], u_\tau \in \left[\frac{1}{r_{\max}}, \frac{1}{r_{\min}}\right], \text{ for } \tau = t, \ldots, t+K-1, \\
& x_{t-1} = \sigma_{t-1}, u_{t-1} = \nu_{t-1}.
\end{aligned} \tag{38}$$

We use $\{(\hat{a}_\tau, \hat{u}_\tau, \hat{x}_\tau)\}_{\tau=t,\ldots,t+K-1}$ to denote the optimal solution of (38).

For the sake of contradiction, we assume there exists $\tau \in \{t, t+1, \ldots, t+K-1\}$ such that

$$\left|\hat{\psi}_t^{t+K-1}\left((\sigma_{t-1}, \nu_{t-1}); \hat{\omega}; \mathbf{0}\right)_{u_\tau} - \hat{\phi}_t^{t+K-1}\left((\sigma_{t-1}, \nu_{t-1}); \hat{\omega}; \mathbf{0}\right)_{u_\tau}\right| > \lambda.$$

By the strongly convexity of the constrained optimization problem (37), we see that

$$\sum_{\tau=t}^{t+K-1} \gamma \hat{a}_t^2 - \sum_{\tau=t}^{t+K-1} \gamma \check{a}_t^2 \geq \gamma \sum_{\tau=t}^{t+K-1} (\hat{a}_t - \check{a}_t)^2 > \frac{\gamma \lambda^2}{K}. \tag{39}$$

On the other hand, we have that

$$\sum_{\tau=t}^{t+K-1} \left(\hat{\omega} \hat{u}_t^2 + \beta b(\hat{x}_t)\right) - \sum_{\tau=t}^{t+K-1} \left(\hat{\omega} \check{u}_t^2 + \beta b(\check{x}_t)\right) \geq K \left(\hat{\omega}\left(\frac{1}{r_{\max}^2} - \frac{1}{r_{\min}^2}\right) - \beta \max\{\bar{x}^2, \epsilon (x_{\max} - \bar{x})^2\}\right)$$

By the optimality of $\{(\hat{a}_\tau, \hat{u}_\tau, \hat{x}_\tau)\}_{\tau=t,\ldots,t+K-1}$ in (38), we see that

$$\begin{aligned}
0 \geq & \sum_{\tau=t}^{t+K-1} \left(\hat{\omega} \hat{u}_t^2 + \beta b(\hat{x}_t) + \gamma \hat{a}_t^2\right) - \sum_{\tau=t}^{t+K-1} \left(\hat{\omega} \check{u}_t^2 + \beta b(\check{x}_t) + \gamma \check{a}_t^2\right) \\
> & \frac{\gamma \lambda^2}{K} + K \left(\hat{\omega}\left(\frac{1}{r_{\max}^2} - \frac{1}{r_{\min}^2}\right) - \beta \max\{\bar{x}^2, \epsilon (x_{\max} - \bar{x})^2\}\right),
\end{aligned}$$

which contradicts our assumption that

$$\gamma \geq \frac{K^2}{\lambda^2}\left(\hat{\omega}\left(\frac{1}{r_{\min}^2} - \frac{1}{r_{\max}^2}\right) + \beta \max\{\bar{x}^2, \epsilon (x_{\max} - \bar{x})^2\}\right).$$