2021 IEEE Intelligent Vehicles Symposium (IV)
July 11-17, 2021. Nagoya, Japan

Separated Proportional-Integral Lagrangian for Chance Constrained Reinforcement Learning

Baiyu Peng[1], Yao Mu[1], Jingliang Duan[1], Yang Guan[1], Shengbo Eben Li[1*], Jianyu Chen[2]

***Abstract*— Safety is essential for reinforcement learning (RL) applied in real-world tasks like autonomous driving. Imposing chance constraints (or probabilistic constraints) is a suitable way to enhance RL safety under model uncertainty. Existing chance constrained RL methods like the penalty methods and the Lagrangian methods either exhibit periodic oscillations or learn an over-conservative or unsafe policy. In this paper, we address these shortcomings by elegantly combining these two methods and propose a separated proportional-integral Lagrangian (SPIL) algorithm. We first rewrite penalty methods as optimizing safe probability according to the proportional value of constraint violation, and Lagrangian methods as optimizing according to the integral value of the violation. Then we propose to add up both the integral and proportion values to optimize the policy, with an integral separation technique to limit the integral value within a reasonable range. Besides, the gradient of policy is computed in a model-based paradigm to accelerate training. The proposed method is proved to reduce oscillations and conservatism while ensuring safety by a car-following experiment.**

I. Introduction

Reinforcement Learning (RL) has shown exceptional successes in a variety of domains, from video games [1]–[3] to robotics [4], [5]. As a self-learning method, RL is promising to reduce the massive engineering efforts in autonomous driving. In recent years, there has been a growing interest towards RL in autonomous driving community, such as adaptive cruise control [6], lane-keeping [7], trajectory tracking [8] and multi-vehicle cooperation [9]. However, despite achieving decent performance, these RL methods mostly lack explicit safety constraints, which significantly limits their application in safety-critical autonomous driving.

Recently, some RL researchers begin to investigate including different forms of safety constraints in RL algorithms to improve safety for real-world applications [10]–[13]. One of the most popular forms is the chance constraint, which constrains the possibility of the control policy violating the state constraint below a given level [10], [14], [15]. Chance constraint gives an intuitive and quantitative measure of the safety level of the control policy, so it is suitable to represent the safety demands in real-world systems with uncertainty.

This study is supported by International Science & Technology Cooperation Program of China under 2019YFE0100200, Tsinghua-Toyota Joint Research Institute Cross-discipline Program and Xilinx.

[1]State Key Laboratory of Automotive Safety and Energy, School of Vehicle and Mobility, Tsinghua University, Beijing, China * Corresponding author `lishbo@tsinghua.edu.cn`

[2]Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China `jianyuchen@tsinghua.edu.cn`

Existing strategies used to solve the chance constrained RL problems can be roughly categorized into two approaches. The first solution is the penalty method that gives a large penalty to the objective function for violation of the constraint [9], [14]. Although this approach is very straightforward and simple to implement, it requires the penalty weight to strike a balance between safety and performance correctly. Unfortunately, it is usually difficult to select an appropriate weight. As shown in Fig. 1(a), a large penalty is prone to rapid oscillations and does not converge to a safe policy, while a small penalty cannot satisfy the constraint [16]. The second approach is the Lagrangian method [10], [15], which is widely used in constrained optimization. Actually, it can be regarded as the penalty method with an adaptive weight, which is dynamically adjusted by safety level rather than fixed. Nevertheless, the Lagrangian method suffers from overshooting of Lagrange multiplier under tight chance constraint as shown in Fig. 1(b), which will lead to a over-conservative policy. Besides, it may also have periodic oscillations, resulting from the delay between the optimization of policy and adaptation of the Lagrange multiplier [17], [18].

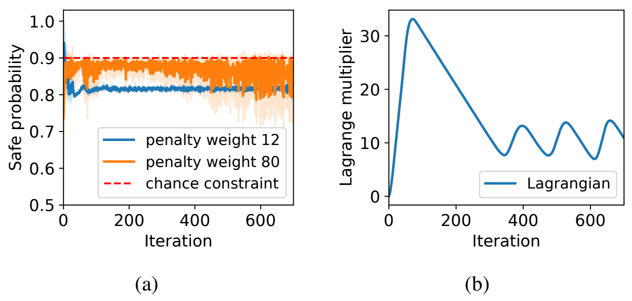


Fig. 1. (a) Penalty method exhibits oscillations and violates the constraint. (b) Lagrangian method exhibits overshooting and oscillations of Lagrange multiplier.

To overcome the drawbacks of previous two methods, we elegantly integrate them and propose a separated proportional-integral Lagrangian (SPIL) method which can fulfill the safety requirements with a steady and fast learning process. Inspired by the dynamical systems view of optimization [18], [19], we first rewrite penalty methods as optimizing safe probability according to the proportional value of constraint violation, and Lagrangian methods as optimizing according to the integral value of the violation. Subsequently, a mixed proportional-integral method is formulated, which combines both methods to get their merits.

To avoid a excessively high integral value (i.e., integral-overshooting), we draw inspiration from PID control again and introduce the integral separation technique, i.e., separate the integrator out when the integral value is large. Such a recipe solves the integral-overshooting problem that is ignored and unsolved in similar works [18]. In addition, we also introduce an analytical gradient of the safe probability with the theoretical basis to optimize policy in a model-based framework. Finally, the experiment of a car-following task demonstrates SPIL succeeds in satisfying the chance constraint while achieving best cumulative reward.

The contributions of this paper are summarize as follows,

1) a separated proportional-integral Lagrangian (SPIL) algorithm is proposed to solve chance constrained RL problems with better performance while satisfying the constraint.
2) an analytical gradient of safe probability is introduced for model-based policy optimization with theoretic basis.

The rest of this paper is organized as follows. The chance constrained RL problem is formulated in Section II. The SPIL method is proposed in Section III. The effectiveness of the method is illustrated by a car-following task in Section IV. Section V concludes this paper.

II. Chance Constrained RL problems

Considering a discrete-time stochastic system, the dynamic with the chance constraint is mathematically described as:

$$\begin{aligned} & x_{t+1} = f(x_t, u_t, \xi_t), \\ & \xi_t \sim p(\xi_t), \\ & \Pr\left\{ \bigcap_{t=1}^{N} \left[h\left(x_t\right) < 0 \right] \right\} \geq 1 - \delta \end{aligned} \tag{1}$$

where t is the current step, $x_t \in \mathcal{X}$ is the state, $u_t \in \mathcal{U}$ is the action, $f(\cdot,\cdot,\cdot)$ is the environmental dynamic model, $\xi_t \in \mathbb{R}^n$ is the uncertainty following an independent and identical distribution $p(\xi_t)$. $h(\cdot)$ is the state constraint function defining a safe state region. We do not make assumptions about the form of $f(\cdot,\cdot,\cdot)$ and $h(\cdot)$, i.e., they can be linear or nonlinear. Note that here the safety constraint takes the form of a joint chance constraint with $1-\delta$ as the required threshold. This form is extensively used in stochastic systems control [20]. Intuitively, it can be interpreted as the probability of the plant staying within a safe region over the finite horizon N is at least $1-\delta$. For simplicity, we only consider one constraint, but our method can readily generalize to multiple constraints by introducing multiple Lagrangian multipliers and updating them respectively.

The objective of chance constrained RL problems is to maximize the expectation of cumulative reward J, while constraining the safe probability p_s:

$$\begin{aligned} & \max_{\pi} J\left(\pi\right) = \mathbb{E}_{x_0,\xi}\left\{ \sum_{t=0}^{\infty} \gamma^t r\left(x_t, u_t\right) \right\} \\ & \text{s.t. } p_s(\pi) = \Pr\left\{ \bigcap_{t=1}^{N} \left[h\left(x_t\right) < 0 \right] \right\} \geq 1 - \delta \end{aligned} \tag{2}$$

where $r(\cdot,\cdot)$ is the reward function, $\gamma \in (0,1)$ is the discounting factor, $\mathbb{E}_{x_0,\xi}(\cdot)$ is the expectation w.r.t. the initial state x_0 and uncertainty ξ. π is the control policy, i.e., a deterministic mapping from state space $\mathcal{X}$ to action space $\mathcal{U}$ with parameters θ, i.e., $u_t = \pi(x_t;\theta)$.

III. Separated PI Lagrangian for Chance Constrained RL Problems

In this section, we will elaborate on the SPIL method for chance constrained problems. Besides, we also introduce an analytical gradient of safe probability and update the policy in a model-based mechanism.

A. Separated PI Lagrangian method

The PI Lagrangian method comes from a control view of the penalty method and the traditional Lagrangian method. It considers the penalty method as a proportional feedback controller and the traditional Lagrangian method as an integral feedback controller, which can be integrated together and lead to a PI Lagrangian method. To see this, we first review the penalty method and traditional Lagrangian method. The penalty method adds a quadratic penalty term in the objective function to force the satisfaction of the constraint:

$$\max_{\pi} J(\pi) - \frac{1}{2}\alpha_p \left((1-\delta-p_s(\pi))^+ \right)^2 \tag{3}$$

where $\alpha_p > 0$ is the penalty weight, $(\cdot)^+$ means $\max(\cdot, 0)$. This unconstrained problem is solved by gradient ascent

$$\theta^k \leftarrow \theta^{k-1} + \alpha_\theta(\nabla_\theta J^k + \alpha_p(1-\delta-p_s^k)^+ \nabla_\theta p_s^k) \tag{4}$$

where k means k-th iteration, $\alpha_\theta > 0$ is the learning rate.

As for the traditional Lagrangian method, it first transforms the original chance constrained problem (2) into an dual problem by introduction of a the Lagrange multiplier λ [21]:

$$\max_{\lambda \geq 0} \max_{\pi} \mathcal{L}(\pi, \lambda) = J(\pi) - \lambda\left(1-\delta-p_s(\pi)\right) \tag{5}$$

The problem (5) is solved by iteratively updating the Lagrange multiplier and primal variables:

$$\lambda^k \leftarrow (\lambda^{k-1} + \alpha_\lambda(1-\delta-p_s^k))^+ \tag{6}$$

$$\theta^k \leftarrow \theta^{k-1} + \alpha_\theta(\nabla_\theta J^k + \lambda^k \nabla_\theta p_s^k) \tag{7}$$

where $\alpha_\lambda > 0$ is the learning rate. Comparing the policy update rule of penalty method (4) with that of Lagrangian method (7), one may find they are surprisingly similar. Both gradients is the weighted sums of $\nabla_\theta J^k$ and $\nabla_\theta p_s^k$. The only difference lies in that the weight $(1-\delta+p_s^k)^+$ in penalty method is the constraint violation at k-th iteration, while the weight λ in Lagrangian method is the constraint violation accumulated in the previous k iterations. This insight builds the bridge between optimization and feedback control. One can view the optimization as dynamic systems control, where the weight of $\nabla_\theta p_s$ is the control input, p_s is the control output, $1-\delta$ is the desired output and $1-\delta-p_s^k$ is the feedback error. Consequently, the penalty method becomes a proportional controller with coefficient α_p, while the Lagrangian method becomes an integral controller with coefficient α_λ. Considering constrained optimization in such a control perspective, one can immediately understand the merits and faults of these two methods. For the penalty method, a large penalty α_p is prone to oscillations, while a small penalty leads to steady-state errors, i.e., not satisfying the constraint. For the Lagrangian method, it suffers from periodic oscillations from a delayed feedback.

Subsequently, we naturally formulate a proportional-integral Lagrangian method to realize fast and steady learning process with no steady-state error. The update rule is a combination of previous two methods:

$$\Delta^k \leftarrow 1-\delta-p_s^k \tag{8}$$
$$I^k \leftarrow (I^{k-1}+\Delta^k)^+ \tag{9}$$
$$\lambda^k \leftarrow (K_P\Delta^k+K_I I^k)^+ \tag{10}$$
$$\theta^k \leftarrow \theta^{k-1}+\alpha_\theta(\nabla_\theta J^k+\lambda^k\nabla_\theta p_s^k) \tag{11}$$

where Δ, I are proportional and integral values, respectively, with K_P, K_I denoting their coefficients. The proportional term Δ serves as an immediate feedback of the constraint violation. The integral term I eliminates the steady-state error at convergence. In such a framework, the penalty method and traditional Lagrangian method can be regarded as two special cases of PI Lagrangian with $K_P>0, K_I=0$ and $K_P=0, K_I>0$, respectively. Actually, the proportional and integral terms together will achieve better performance in RL, just as the PI controller works well in control area.

However, if the chance constraint is very tight and the initial policy is relatively unsafe, the integral terms usually increase rapidly since Δ is large, which will cause the overshooting of λ. With a large λ in (11), the policy tends to be over-conservative since the weight of $\nabla_\theta p_s^k$ is relatively large. Even worse, since the maximal safe probability is 1, the overshooting and conservatism problems will not recover by themselves. For e.g., if $1-\delta=0.999, p_s=1.0$ and the λ^k is already overshooting, the integral term I only decreases very slowly with the speed of $\Delta^k=-0.001$. Therefore, the policy optimization in such a case will be decelerated. This challenge is also not well recognized and resolved in previous similar works like [18]. In this paper, we draw inspiration from some anti-saturation methods in PID control [22], and introduce the integral separation technique. It reshapes the integrator in (9) into:

$$\begin{aligned} &I^k \leftarrow (I^{k-1}+K_S\Delta^k)^+,\\ &K_S=\begin{cases}0 & \varepsilon_1<\Delta^k\\ \beta & \varepsilon_2<\Delta^k<\varepsilon_1\\ 1 & \Delta^k<\varepsilon_2\end{cases}\end{aligned} \tag{12}$$

where K_S is the separation function, $1>\beta>0, \varepsilon_1>\varepsilon_2>0$ are the parameters. Obviously, the piecewise function K_S separates the integrator out or slows it down if the error is relatively large. Such a recipe prevents the occurrence of integral-overshooting, greatly improving the performance under tight constraint as shown in our experiments.

The framework of proposed method is summarized Fig. 2.

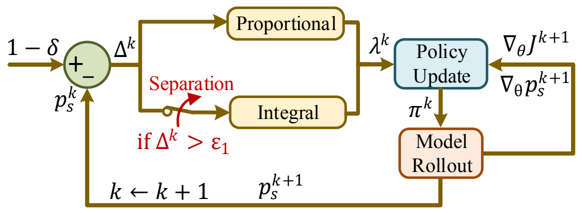


Fig. 2. The framework of SPIL method.

B. Analytical Gradient of Safe Probability

In the previous subsection, we have derived the main update rules of our SPIL method. The following parts discuss about how to calculate p_s and $\nabla_\theta p_s$ in the update equations.

The safe probability p_s can be directly estimated through Monte-Carlo sampling. To be specific, we rollout M trajectories with current policy π through the dynamic model. Suppose there are m safe trajectories, then the safety probability is $p_s \approx \frac{m}{M}$. Note that this rollout procedure will not impose much extra computation burden since these trajectories are also necessary for the update of actor-critic as we will discuss in III-C.

However, it turns out that the gradient $\nabla_\theta p_s(\pi)$ is rather difficult to compute, which is also a major challenge in chance constrained problems [20], [23]. Previous researchers usually replace $\nabla_\theta p_s(\pi)$ with the gradient of a lower bound of p_s without sufficient theoretical basis [14], [15]. In this paper, we introduce an analytical gradient from the recent research in stochastic optimization [23]. To the best of our knowledge, this is the first time such a gradient is used in RL.

We first define an indicator-like function $\phi(z,\tau)$:

$$\begin{aligned}&\phi(z,\tau)=\frac{1+a_1\tau}{1+a_2\tau\exp(-\frac{z}{\tau})},\\ &0<a_2<\frac{a_1}{1+a_1}, 0<\tau<1\end{aligned} \tag{13}$$

where z and τ are scalar variables of the function, a_1, a_2 are the parameters. The expected production $\Phi(\pi,\tau)$ is defined as:

$$\Phi(\pi,\tau)=\mathbb{E}_{x_0,\xi}\left\{\prod_{t=1}^{N}\phi\left(-h(x_t),\tau\right)\right\} \tag{14}$$

As shown in Fig. 3, $\phi(z,\tau)$ can be intuitively regarded as a differentiable approximation of indicator function for constraint violation, and its expected product $\Phi(\pi,\tau)$ approximates joint safe probability. The parameter τ controls

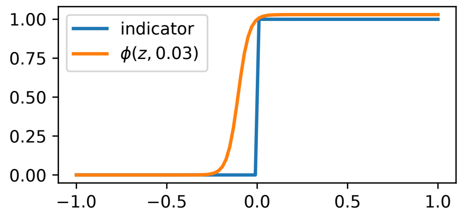


Fig. 3. Comparison of indicator function and $\phi(z, 0.03)$.

how well the indicator function is approximated. Regardless of the nonlinearity and convexity of $h(x_t)$, the gradient of $\Phi(\pi, \tau)$ is proved to converge to the gradient of joint safe probability $p_s(\pi)$ as τ approaches 0 under mild assumptions [23].

$$\lim_{\tau \to 0+} \sup_{\theta \in \Theta} \nabla_\theta \Phi(\pi, \tau) = \nabla_\theta p_s(\pi) \tag{15}$$

where Θ is a ball in the policy parameter space. The equation (15) shows that one can use the gradient of a differentiable function $\Phi(\pi, \tau)$ to approximate $\nabla_\theta p_s(\pi)$ if τ is small enough. For simplicity, we do not provide more mathematical details; interested readers are recommended to refer to [23] for a rigorous explanation. In practice, one only needs to pick a small fixed τ and compute $\nabla_\theta \Phi(\pi, \tau)$ with any autograd package, where the expectation is substituted by sampling average. According to our experiments, the algorithm performance is not sensitive to τ, and the range from 0.001 to 0.1 is acceptable. Finally, the order of magnitude of $\nabla_\theta J$ and $\nabla_\theta p_s$ are usually different. To better balance them, the gradient $\nabla_\theta p_s$ is re-scaled to match the scale of $\nabla_\theta J$:

$$\nabla_\theta p_s \leftarrow \frac{\|\nabla_\theta J\|}{\|\nabla_\theta p_s\|} \nabla_\theta p_s \tag{16}$$

C. Model-based Actor-Critic with Parameterized Functions

In this subsection, the main focus is on how to learn a parameterized policy and state-action value function in the model-based framework, where the gradient of the dynamic model will be utilized to attain an accurate ascent direction and thus improve the convergence rate compared with model-free RL algorithms [24], [25].

For an agent behaving according to policy π, the values of the state-action pair (x, u) are defined as follows:

$$Q^\pi(x, u) = \mathbb{E}_\xi \left\{ \sum_{t=0}^{\infty} \gamma^t r(x_t, u_t) \Big| x_0 = x, u_0 = u \right\} \tag{17}$$

Consequently, the expected cumulative reward J can be expressed as a N-step form:

$$J(\pi) = \mathbb{E}_{x_0, \xi} \left\{ \sum_{t=0}^{N-1} \gamma^t r(x_t, u_t) + \gamma^N Q^\pi(x_N, u_N) \right\} \tag{18}$$

For large and continuous state spaces, both value function and policy are parameterized, as shown in (19). The parameterized state-action value function with parameter w is usually named the "critic", and the parameterized policy with parameter θ is named the "actor" [25].

$$Q(x, u) \cong Q(x, u; w), \quad u \cong \pi(x; \theta) \tag{19}$$

The parameterized critic is trained by minimizing the average square error (20):

$$J_Q^k = \mathbb{E}_{x_0, \xi} \left\{ \frac{1}{2} \left(Q_{\text{target}} - Q(x_0, u_0; w^k) \right)^2 \right\} \tag{20}$$

where $Q_{\text{target}} = \sum_{t=0}^{N-1} \gamma^t r(x_t, u_t) + \gamma^N Q\left(x_N, u_N; w^k\right)$ is the N-step target. Note that the rollout length N is identical to the horizon of chance constraint.

The semi-gradient of the critic is

$$\nabla_\omega J_Q^k = \mathbb{E}_{x_0, \xi} \left\{ \left(Q(x_0, u_0; w^k) - Q_{\text{target}} \right) \frac{\partial Q(x_0, u_0; w^k)}{\partial w} \right\} \tag{21}$$

As discussed in (5), the parameterized actor aims to maximize Lagrangian function $\mathcal{L}$ via gradient ascent. The analytical gradient $\nabla_\theta \mathcal{L}$ is composed of $\nabla_\theta J$ and $\nabla_\theta p_s$, which are computed via backpropagation though time with the dynamic model [25]. In practice, they can be easily obtained by any autograd package. Finally, the pseudo-code of proposed algorithm is summarized in Algorithm 1. Note that, to maintain a relatively consistent step size, the update rules for θ in (11) is re-scaled by $\frac{1}{1+\lambda^k}$.

Algorithm 1 SPIL algorithm

```
Initialize $x_0 \in \mathcal{X}$, $k = 0$
repeat
    Rollout $M$ trajectories by $N$ steps via dynamic model
    Estimate safe probability
        $p_s^k \leftarrow \frac{m}{M}$
    Update $\lambda$ via PI Lagrangian rules
        $\Delta^k \leftarrow 1 - \delta - p_s^k$
        $I^k \leftarrow (I^{k-1} + K_S \Delta^k)^+$
        $\lambda^k \leftarrow (K_P \Delta^k + K_I I^k)^+$
    Update critic according to (21):
        $\omega^k \leftarrow \omega^{k-1} + \alpha_\omega \nabla_\omega J_Q^k$
    Update actor:
        $\theta^k \leftarrow \theta^{k-1} + \alpha_\theta \nabla_\theta \mathcal{L}$
        $\nabla_\theta \mathcal{L} = \frac{1}{1+\lambda^k} \left( \nabla_\theta J^k + \lambda^k \nabla_\theta p_s^k \right)$
    $k \leftarrow k + 1$
until $|Q^k - Q^{k-1}| \le \zeta$ and $|\pi^k - \pi^{k-1}| \le \zeta$
```

IV. Numerical Experiment

A. Experiment Setup

In this section, the proposed SPIL is applied to a car-following scenario as shown in Fig. 4, where the ego car expects to drive fast and closely with the front car to reduce wind drag [26], while keeping a minimum gap between the two cars at a high probability. Concretely, the ego car and front car follow the kinematics model, where the front car is assumed to drive with a randomly varying velocity (e.g., due to the varying road grade, wind drag).

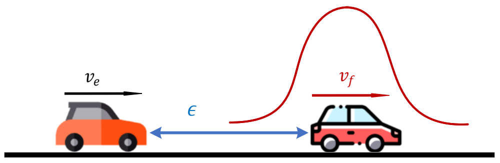


Fig. 4. Car-following scenario.

The discrete-time stochastic system is

$$
\begin{aligned}
x_{t+1} &= Ax_t + Bu_t + D\xi_t \\
A &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -T & T & 1 \end{bmatrix}, \\
B &= [T, 0, 0]^\top, \quad D = [0, T, 0]^\top
\end{aligned}
\tag{22}
$$

The system state vector is $x = [v_e \quad v_f \quad \epsilon]^\top$, where v_e denotes the velocity of ego car, v_f is the velocity of front car, and ϵ is the gap between the two cars. The control input $u \in (-4, 3)$ is the acceleration of ego car. The disturbance $\xi_t \sim \mathcal{N}(0, 0.7)$ is truncated in the interval $(-7, 7)$. $T = 0.1s$ is the simulation time step. With a chance constraint on the gap, the chance constrained RL problem is defined as

$$
\begin{aligned}
&\max_\pi \sum_{t=0}^{\infty} \gamma^t (0.2v_{e,t} - 0.1\epsilon_t - 0.02u_t^2) \\
&\text{s.t. } \Pr\left\{ \bigcap_{t=1}^{N} (\epsilon_t > 2) \right\} \geq 1 - \delta
\end{aligned}
\tag{23}
$$

where $v_{e,t}$ denotes the ego car velocity at step t.

B. Implementation Details

We implement SPIL algorithm on the problem above. Our parameterized actor and critic are both fully-connected neural networks. Each network has two hidden layers using rectified linear unit (ReLU) as activation functions, with 64 units per layer. We adopt the Adam method to update the networks [27]. The main hyper-parameters are listed in Table I.

To demonstrate the advantages of SPIL, we compare the performance of SPIL with the penalty method (amounts to proportional-only SPIL) and traditional Lagrangian method (amounts to integral-only SPIL). The coefficients of SPIL are $K_P = 15, K_I = 0.6$. The penalty method is trained on two different weights $K_P = 12$ and 80. Actually, $K_P = 80$ already has large oscillation, so we do not choose a larger weight even through $K_P = 80$ cannot totally satisfy the constraint. For the traditional Lagrangian, we initially tested on small $K_I = 0.6$ but it diverged, so a large $K_I = 18$ is chosen. The cumulative reward and safe probability in horizon N are compared under two chance constraint thresholds 90.0% and 99.9%, i.e., $\delta = 0.1$ and $\delta = 0.001$.

C. Evaluation Results

1) Overall Performance: The learning curves are plotted in Fig. 5, where each curve is averaged over five independent experiments. We emphasize that any comparison between the

TABLE I
HYPER-PARAMETERS

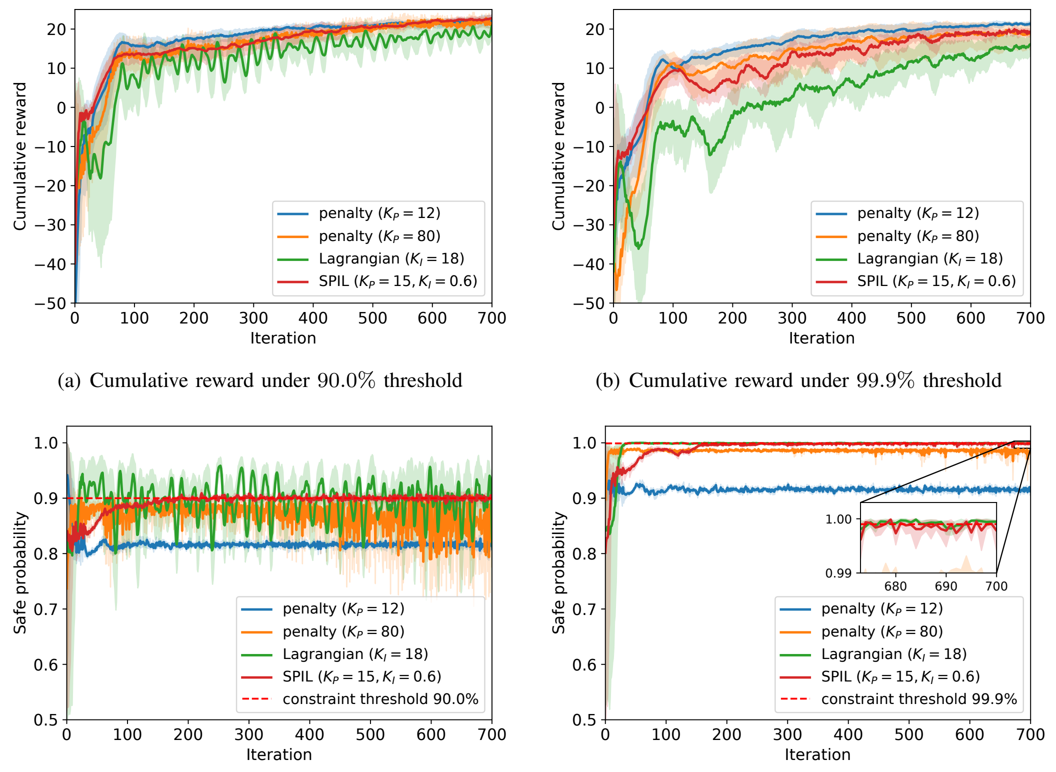
Parameters	Symbol	Value
trajectories number	M	4096
constraint horizon	N	40
discounting factor	γ	0.99
learning rate of policy network	α_θ	3e-4
learning rate of value network	α_ω	2e-4
parameters of K_S	β	0.3
parameters of K_S	ε_1	0.2
parameters of K_S	ε_2	0.05
parameters of $\phi(\cdot)$	τ	1e-3
parameters of $\phi(\cdot)$	a_1	0.45
parameters of $\phi(\cdot)$	a_2	1

methods should consider both safety and reward-winning, and the reader should not draw conclusions only from one aspect. Generally, our SPIL not only succeeds to satisfy the chance constraint without periodic oscillations, but also achieves best cumulative reward among methods which meet the safety threshold.

Observing the safe probability curves in Fig. 5(c) and Fig. 5(d), the proposed SPIL satisfies the chance constraint in both settings. On the contrary, the penalty method with $K_P = 12$ fails to achieve the required threshold due to small penalty weight. Although one can improve the penalty size and raise K_P to reduce this error (i.e., set $K_P = 80$), large K_P also brings about rapid oscillations as a side effect, especially when the threshold is 90.0%. This is because with a large K_P, a small change of Δ will cause a dramatic change of λ, resulting in a radical policy update. The Lagrangian method does not have steady-state errors (i.e., constraint violation), but suffers from periodic oscillations under 90.0% threshold. In a word, the proposed SPIL combines the advantages of integral and proportional methods, leading to a stable learning process with no steady-state errors. Interestingly, these phenomena are quite similar to conclusions in PID control, which exhibits the beauty of understanding optimization from the control perspective.

As for the cumulative reward shown in Fig. 5(a) and Fig. 5(b), excluding the unsafe penalty method ($K_P = 12$), SPIL achieves the best cumulative reward in both thresholds among the other three methods, which confirms the excellent performance of SPIL.

2) Integral Separation: Subsequently, we demonstrate that the integral separation technique in SPIL is essential to prevent integral-overshooting and avoid policy over-conservatism. We manually select five initially unsafe random seeds, i.e., the safe probability of initial policy $p_s < 0.5$, and train the policy under 99.9% threshold using SPIL with and without integral separation. The learning curves of cumulative reward J, safe probability p_s, integral value I are plotted in Fig. 6. If the integral separation is removed, the integral value I in Fig. 6(c) will have a sharp rise at the beginning. Then the policy rapidly learns to satisfy the constraint with safe probability becoming 1. However, since $\Delta = -0.001$ in (9), the decrease of I is quite slow. With the excessively large I and λ, the policy keeps conservative for



(a) Cumulative reward under 90.0% threshold

(b) Cumulative reward under 99.9% threshold

(c) Safe probability under 90.0% threshold

(d) Safe probability under 99.9% threshold

Fig. 5. Comparison of performance among SPIL (separated PI Lagrangian), penalty method and traditional Lagrangian method.

a long time and wins few rewards. On the contrary, with the help of integral separation, I will not overshoot at the start and the policy successfully strikes a good balance between performance and safety, i.e., achieves more rewards while satisfying the constraint. We stress the integral-overshooting problems are not well recognized and solved in similar RL works [18] and we are the first to discuss about it and give a solution. Note that the results in Fig. 5 and Fig. 6 are not comparable since the latter are conducted under manually chosen bad initial policies.

3) Sensitivity Analysis: A potential drawback of SPIL is that it introduces additional tuning parameters such as K_P, K_I and β. However, we find that the performance of SPIL is relatively insensitive to the choice of these parameters. To demonstrate this, we conduct a series of experiments across different values of K_P , K_I and β while keeping all other parameters fixed. The results under 90.0% threshold are summarized in Table II, III and IV, with the best parameters shown in bold. Even the worst case only leads to 1.8% degradation in safe probability and 6.5% degradation in cumulative reward.

TABLE II
COMPARISON OF PERFORMANCE FOR DIFFERENT K_P

Value of K_P	3.75	7.5	**15**	30	60
Cumulative reward	22.27	21.86	**23.01**	21.91	21.49
Safe probability	89.5%	90.2%	**90.0%**	91.6%	88.2%

TABLE III
COMPARISON OF PERFORMANCE FOR DIFFERENT K_I

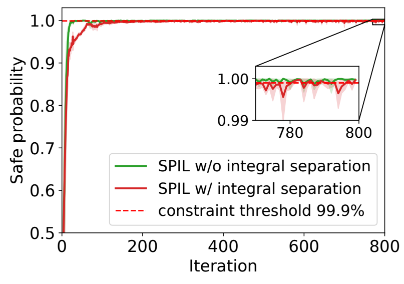
Value of K_I	0.15	0.3	**0.6**	1.2	2.4
Cumulative reward	21.60	22.18	**23.00**	22.77	21.87
Safe probability	89.9%	89.96%	**90.0%**	90.52%	89.5%

TABLE IV
COMPARISON OF PERFORMANCE FOR DIFFERENT β

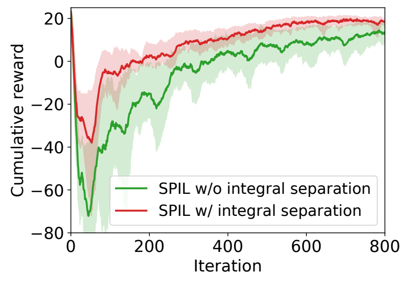
Value of β	0.1	0.2	**0.3**	0.4	0.5
Cumulative reward	21.87	21.25	**23.00**	21.90	21.74
Safe probability	90.0%	90.9%	**90.0%**	89.9%	90.1%

V. CONCLUSION

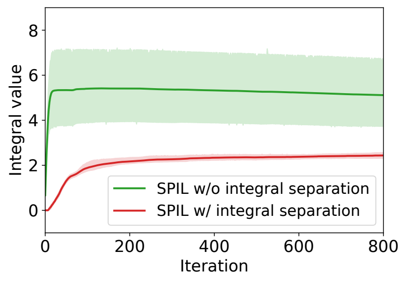
This paper proposed the separated proportional-integral Lagrangian (SPIL) method for chance constrained RL. While previous methods either utilized the proportional or the integral values of constraint violation to optimize the safe probability, SPIL instead utilized both the proportional and integral value to combine their merits, and enhanced it with an integral separation technique to limit the integral value in a reasonable range. In addition, SPIL also introduced an analytical gradient of safe probability for model-based policy optimization. The benefits of SPIL were demonstrated in simulations of a car-following task. It achieved high cumulative reward while satisfying the chance constraint



(a) Safe probability under 99.9% threshold



(b) Cumulative reward under 99.9% threshold



(c) Integral value under 99.9% threshold

Fig. 6. Comparison of performance of SPIL with and without integral separation.

under different thresholds. The application of SPIL to more general environmental dynamics will be investigated in the future.

REFERENCES

[1] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, *et al.*, "Grandmaster level in starcraft ii using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.

[2] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. A. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, 2015.

[3] M. Hessel, J. Modayil, H. V. Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. G. Azar, and D. Silver, "Rainbow: Combining improvements in deep reinforcement learning," in *AAAI*, 2018.

[4] T. Kurutach, I. Clavera, Y. Duan, A. Tamar, and P. Abbeel, "Model-ensemble trust-region policy optimization," in *International Conference on Learning Representations*, 2018.

[5] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International Conference on Machine Learning*, pp. 1861–1870, PMLR, 2018.

[6] Y. Lin, J. McPhee, and N. L. Azad, "Longitudinal dynamic versus kinematic models for car-following control using deep reinforcement learning," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 1504–1510, IEEE, 2019.

[7] J. Duan, S. E. Li, Y. Guan, Q. Sun, and B. Cheng, "Hierarchical reinforcement learning for self-driving decision-making without reliance on labelled driving data," *IET Intelligent Transport Systems*, vol. 14, no. 5, pp. 297–305, 2020.

[8] Y. Mu, B. Peng, Z. Gu, S. Li, C. Liu, B. Nie, J. Zheng, and B. Zhang, "Mixed reinforcement learning for efficient policy optimization in stochastic environments," *2020 20th International Conference on Control, Automation and Systems (ICCAS)*, pp. 1212–1219, 2020.

[9] Y. Guan, Y. Ren, S. Li, Q. Sun, L. Luo, and K. Li, "Centralized cooperation for connected and automated vehicles at intersections by proximal policy optimization," *IEEE Transactions on Vehicular Technology*, vol. 69, pp. 12597–12608, 2020.

[10] Y. Chow, M. Ghavamzadeh, L. Janson, and M. Pavone, "Risk-constrained reinforcement learning with percentile risk criteria," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 6070–6120, 2017.

[11] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *International Conference on Machine Learning*, pp. 22–31, PMLR, 2017.

[12] T.-Y. Yang, J. Rosca, K. Narasimhan, and P. J. Ramadge, "Projection-based constrained policy optimization," in *International Conference on Learning Representations*, 2019.

[13] J. Garcıa and F. Fernández, "A comprehensive survey on safe reinforcement learning," *Journal of Machine Learning Research*, vol. 16, no. 1, pp. 1437–1480, 2015.

[14] B. Peng, Y. Mu, Y. Guan, S. Li, Y. Yin, and J. Chen, "Model-based actor-critic with chance constraint for stochastic system," *ArXiv*, vol. abs/2012.10716, 2020.

[15] S. Paternain, M. Calvo-Fullana, L. F. O. Chamon, and A. Ribeiro, "Learning safe policies via primal-dual methods," *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 6491–6497, 2019.

[16] C. Tessler, D. J. Mankowitz, and S. Mannor, "Reward constrained policy optimization," in *International Conference on Learning Representations*, 2018.

[17] B. W. Wah, T. Wang, Y. Shang, and Z. Wu, "Improving the performance of weighted lagrange-multiplier methods for nonlinear constrained optimization," *Information Sciences*, vol. 124, no. 1-4, pp. 241–272, 2000.

[18] A. Stooke, J. Achiam, and P. Abbeel, "Responsive safety in reinforcement learning by pid lagrangian methods," in *International Conference on Machine Learning*, pp. 9133–9143, PMLR, 2020.

[19] W. An, H. Wang, Q. Sun, J. Xu, Q. Dai, and L. Zhang, "A PID controller approach for stochastic optimization of deep networks," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8522–8531, 2018.

[20] A. Mesbah, "Stochastic model predictive control: An overview and perspectives for future research," *IEEE Control Systems*, vol. 36, pp. 30–44, 2016.

[21] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[22] L. Jia and X. Zhao, "An improved particle swarm optimization (PSO) optimized integral separation PID and its application on central position control system," *IEEE Sensors Journal*, vol. 19, pp. 7064–7071, 2019.

[23] A. Geletu, A. Hoffmann, and P. Li, "Analytic approximation and differentiability of joint chance constraints," *Optimization*, vol. 68, pp. 1985 – 2023, 2019.

[24] M. Deisenroth and C. E. Rasmussen, "Pilco: A model-based and data-efficient approach to policy search," in *Proceedings of the 28th International Conference on machine learning (ICML-11)*, pp. 465–472, Citeseer, 2011.

[25] S. E. Li, "Reinforcement Learning and Control." Tsinghua University: Lecture Notes. http://www.idlab-tsinghua.com/thulab/labweb/publications.html, 2020.

[26] F. Gao, S. E. Li, Y. Zheng, and D. Kum, "Robust control of heterogeneous vehicular platoon with uncertain dynamics and communication delay," *IET Intelligent Transport Systems*, vol. 10, no. 7, pp. 503–513, 2016.

[27] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.