# Persistence Length-Based Exploration in Reinforcement Learning

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

BLABLA

## 1 Introduction

We propose exploration algorithm that not only consider the temperature, it also considers the general form of the result trajectory induced from a certain $l_p$.

We also emphasise on the fact that local self-avoidance is achieved only w.r.t. the action-space.

We also emphasize on the comparison between Monte-Carlo sampling and tree backup model with the proposed trajectory based exploration. Since, using $L_p$ we explore trajectories as opposed to the state-action pairs individually.

We emphasise on trajectory-based exploration rather than the state-based exploration as well as a technique that relies on local information about visited state-action pairs as opposed to relying on the statistical summery of visited state-action pairs. The exploration method employs a range of persistence lengths and and and explore a range of low-dimensional manifolds and then explore the selected manifold efficiently. Intuitively we need to show that by choosing proper values for $d_\beta$ one can explore lower dimensional manifolds efficiently and then use results in IJCAI paper and explore the 3D manifold. Therefore two main questions arise here:

1- How to choose $d_B$ to land of the low dimensional manifold, given that such manifold exists.

2- If we assume that we have landed on the $3D$ manifold are we still going to be able to use results from IJCAI and show that they still hold.

## 2 The notion of persistence length: formal treatment

*Maziar:* Mathematically define what a *stiff chain* is.

*Maziar:* Define what end-to-end vector is a stiff chain and show that the end-to-end distribution of a stiff chain is always different than the *freely-jointed* one.

*Maziar:* Computationally show that how the stiffness diminishes over time and change of dimensionality

Note that the strong interaction between bonds reduces the effect of thermal energy.

### 2.1 Stiff chain, a discrete model:

A stiff chain composed of $N$ unit vectors $\{u_1, ..., u_N\}$, in the discrete model is parametrized by the following bending energy:

$$\mathcal{E}_B^N = \frac{\kappa}{2a} \sum_{i=1}^{N} (u_{i+1} - u_i)^2 \tag{1}$$

We also have the distribution function $\nu$ over the first and last bond vectors, $u_f$ and $u_l$ for a stiff chain of $N$ bond vectors. Eq. $15.114$.

$$\mathbb{E}_\nu[u_i.u_j] = e^{\frac{-|i-j|a}{L_p}} \tag{2}$$

$$L_p = \frac{-a}{\log\left[\frac{I_{d/2}(\frac{\kappa}{ak_BT})}{I_{d/2-1}(\frac{\kappa}{ak_BT})}\right]} \tag{3}$$

### 2.2 Stiff chain, a continuous model:

$$\mathcal{E}_B = \frac{\kappa}{2} \int_0^L ds (\partial_s u)^2$$
$$u(s) = \frac{d}{ds} x(s)$$
$$d(s) = \sqrt{dx^2}$$
$$L_p = 2\frac{\kappa}{k_B T()}$$

Our intention is to guarantee the local self-avoidance within a trajectory induced by the exploration policy. We do this, firstly, by assuming that we are in the crossover regime where $L_p \sim D$. This means, that agent always feels that is encompassed within a ball of diameter $D = O(L_p)$. In the crossover regime, the statistical mechanics is governed by the competition of the *Thermal Energy* ($\mathcal{E}_T$) and bending rigidity $\kappa$.

**Note:** We have to note that even when a technique employs both action and state in the exploration phase, we still need to know the reward structure or the observation in order to decide about the next choice of actions.

*Maziar:* Define three different regime and see if the existing categorization extends to higher dimensional regime, clearly the intuition behind all existing definitions come from effort in identifying a parameter that describes three different regimes: 1) Weak confinement, 2) Strong confinement, 3) Crossover.

The physical quantity that captures the behavior of worm-like chain under length constraint (or confinement) is *tangent correlation function*.

In the model proposed thus far, they assume that polymer is a one-dimensional object located in a $d$-dimensional environment. This clearly could be problematic since in our domain trajectory/chain is not necessarily one-dimensional. Thereby, $\kappa^I$ that is assumed to be invariant w.r.t. change in domain dimensionality will not invariant any more.

*Maziar:* We show that a chain is locally homeomorphic to Euclidean 1-space therefore it is a 1-dimensional object. The main idea comes from the fact that trajectory is a time dependent entity and time is the only free parameter and therefore we are not going to have notions like figure eight objects.

*Maziar:* I need to define the *internal* and *external* dimension.

*Maziar:* Prove where $L_P^d$ formula is coming from.

Fro a chain with one internal dimension and $d$ external dimension, $L_p^d$ represents the persistence length in $d$-dimensional space. We define the intrinsic persistence length $L_p^i$ of a chain as a property that captures the inherent competition between *Thermal Energy* and *Rigidity*.

$$L_p^i \propto \frac{\kappa^i}{\mathcal{E}_T} \tag{4}$$

Where $\kappa^i$ represents the intrinsic rigidity required to de-correlate two adjacent bond vectors (vector that connect adjacent actions) in a chain lying on a 3-dimensional manifold and $\mathcal{E}_T$ represents the thermal energy. One can show $L_p^d = \frac{2L_p^i}{d-d_B}$, where $d = d_T + d_B$ (or $d \geq d_T + d_B$) and $d_T$ is the dimension of space governed by entropic energy and $d_B$ represent the dimension of space governed by the bending energy.

## 3 Algorithm:Old

**Algorithm 1:** PolyRL

**Input**: $\mathcal{A}, \mathcal{S},$ Intrinsic Rigidity$\kappa^i$, temperature $\mathcal{E}_T$, $l_B$, $l$, Step-size $b_o$, $\phi$, $\gamma$, $\rho$, $\epsilon$, $\alpha$, $\sigma$, $T$
**Output**: $\hat{w}$

```
Construct the intrinsic L_p^i ← κ^i / E_T ;
L_p^l ← 2L_p^i / (l − l_B);
μ ← cos^{-1}(e^{−b_o / L_p});
Sample A_0 and S_0 w.r.t ρ;
Let w_0 = 0;
for t = 1 to t = T do
    Convert A_t to its spherical equivalent A_t^{sp};
    for i = 1 to l_β do
        Sample θ_o w.r.t. N(μ, σ^2);
        Set A_t^{sp}(i) w.r.t. θ_o and ε (ε greedy) // A_t(i) is the ith coordinate of A_t
        ;
    for i = l_β + 1 to l do
        Sample θ_i w.r.t. U([0, 2π]);
        Choose A_t^{sp}(i) w.t.t. θ_i and ε (ε greedy);
    OR;
    Convert A_t^{sp} to its Cartesian equivalent A_t;
    Take action A_t and observe R_{t+1} and S_{t+1};
    Update ŵ with respect to the update rule in equation (5);
Return ŵ;
```

## 4 Algorithm: 2D and 3D action-space

**Note**, here we are exploring (chain components) the action space therefore the step-size $b_o$ represents the length of the bond vector between actions.

Given the following rotation operators,

$$\prod\nolimits_2^\theta = \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix}$$

$$\prod\nolimits_3^\theta = \begin{bmatrix} \cos\theta + u_x^2\left(1-\cos\theta\right) & u_x u_y\left(1-\cos\theta\right) - u_z\sin\theta & u_x u_z\left(1-\cos\theta\right) + u_y\sin\theta \\ u_y u_x\left(1-\cos\theta\right) + u_z\sin\theta & \cos\theta + u_y^2\left(1-\cos\theta\right) & u_y u_z\left(1-\cos\theta\right) - u_x\sin\theta \\ u_z u_x\left(1-\cos\theta\right) - u_y\sin\theta & u_z u_y\left(1-\cos\theta\right) + u_x\sin\theta & \cos\theta + u_z^2\left(1-\cos\theta\right) \end{bmatrix}$$

around point $(0, 0, 0)$ and unit axis $(u_x, u_y, u_z)$. Update rule,

$$\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t + \alpha\big[R_{t+1} + \gamma \max_a Q_{\boldsymbol{w}_t}(S_{t+1}, a) - Q_{\boldsymbol{w}_t}(S_t, A_t)\big]\nabla_{\boldsymbol{w}_t} Q_{\boldsymbol{w}_t}(S_t, A_t) \tag{5}$$

**Algorithm 2:** PolyRL: 2D

**Input**: $\mathcal{A}$, $\mathcal{S}$, $\prod_d^\bullet$, Intrinsic Rigidity$\kappa^i$, temperature $\mathcal{E}_T$, $d$, Step-size $b_o$, $\phi$, $\gamma$, $\rho$, $\epsilon$, $\alpha$, $\sigma$, $T$
**Output**: $\hat{w}$

```
Construct L_p^(3) ← κ^i / E_T ;
L_p^(d) ← 2L_p^(3) / (d−1);
μ ← cos^{-1}(e^{−b_o / L_p^(d)});
Let w_0 = 0;
for each epoch do
    if t == 0 then
        Sample A_0 and S_0 w.r.t ρ;
        θ ← uniform draw from (−π, π);
    else if t == 1 then
        A_1 ← (A_0(0) + b_o cos(θ), A_0(1) + b_o sin(θ));
    else
        Draw a sample θ from N(μ, σ);
        θ_t ← toss a coin and choose between θ and −θ;
        H_{t−1} ← A_{t−1} − A_{t−2};
        A_t ← A_{t−1} + apply ∏_i^{θ_t} on H_{t−1};
    if A_t is valid then
        Apply step function on action A_t and observe R_{t+1} and S_{t+1};
    else
        End the episode and re-start the chain;
    Update ŵ with respect to the update rule in equation (5);
Return ŵ;
```

**Algorithm 3:** PolyRL:Higher Dimesnion $d \geq 3$

**Input**: $\mathcal{A}$, $\mathcal{S}$, Persistence Length $L_p$, Action Space Dimension $d$, Step-size $b_o$, $\gamma$,
Initial State-Action Distribution $\rho$, $\epsilon$, $\alpha$, $\sigma$, Episode Length $T$

**Output**: $\hat{w}$

$\mu \leftarrow cos^{-1}(e^{\frac{-b_o}{L_p^{(d)}}})$;
remainder $\leftarrow d \mod 3$;
**if** *remainder==0* **then**
    Subchain=d/3;
    Group actions coordinates to “subchain” number of groups;
**else**
    Subchain=(d+(3-remainder))/3;
    Group actions to “subchain” number of groups; add total of “3-remainder” dummy dimensions such that each group has at most “1” dummy dimension;
Let $w_0 = 0$;
**for** *each epoch till T* **do**
    **for** *i=1:Subchain* **do**
        **if** $t == 0$ **then**
            Sample $A_0(i)$ and $S_0(i)$ w.r.t $\rho$;
            $\varphi(i) \leftarrow$ uniform draw from $(0, 2\pi)$;
            $\theta(i) \leftarrow$ uniform draw from $(-\pi/2, \pi/2)$;
        **else if** $t == 1$ **then**
            $A_1(i) \leftarrow (A_0(i,1) + b_o \cos(\theta(i)) \sin(\varphi(i)), A_0(i,2) + b_o \sin(\theta(i)) sin(\varphi(i)), A_0(i,3)) + b_o \cos(\varphi(i))$;
        **else**
            Draw a sample $\theta(i)$ from $\mathcal{N}(\mu, \sigma)$;
            $\theta_t(i) \leftarrow$ toss a coin and choose between $\theta(i)$ and $-\theta(i)$;
            $\varphi_t(i) \leftarrow$ uniform draw from $(0, 2\pi)$;
            $A_t(i) \leftarrow (A_{t-1}(i,1) + b_o \cos(\theta_t(i)) \sin(\varphi_t(i)), A_{t-1}(i,2) + b_o \sin(\theta_t(i)) sin(\varphi_t(i)), A_{t-1}(i,3)) + b_o \cos(\varphi_t(i))$;
    **if** $A_t$ *is valid* **then**
        Apply step function on action $A_t$ and observe $R_{t+1}$ and $S_{t+1}$;
    **else**
        End the episode and re-start the chain;
    Update $\hat{w}$ with respect to the update rule in equation;
    (5) Return $\hat{w}$;