

# Learning Bicycle Stunts

Jie Tan, Yuting Gu, C.Karen Liu, Greg Turk

Pali Kunovský & Lucka Ďurikovičová

# Úvod

- Článok prezentuje simuláciu a ovládanie človeka, ktorý riadi bicykel
- Dva hlavné komponenty systému sú
  - offline learning
  - online simulácia



# Motivácia

- Už malé deti sa dokážu naučiť

- Dostať sa na bicykel
- Udržať balans
- Riadenie bicykla po nerovnomernom povrchu.

čo vyžaduje týždne praxe.



- Motiváciou bolo zistiť či je možné použiť aj počítač na naučenie tohoto procesu

- Základne manévry
- Skákať cez prekážky
- Zdvihnutie jedného kolesa a balansovať
- Vykonávanie rôznych rizikových bicyklových kúskov

# Problematika

- Interakcia medzi človekom (cyklistom) a bicyklom (komplexný problém)
- Kým cyklista aktívne riadi každú časť bicykla, bicykel je pasívny systém, ktorý môže byť iba kontrovaný človekom
- Ak chceme riadiť jeden stupeň voľnosti (DOF) bicykla, vyžaduje koordinované pohyby s niekoľkými stupňami voľnosti
- Ťažkosťou v pohybe je, ak pri pôsobení externých síl na bicykel chceme ovládať systém
- Všetky kontrolné sily jazdca musia najprv prekonať dynamiku bicykla a reakčné sily s povrchom zeme a naopak (komplexný problém)
- Balans na bicykli je náročný vzhľadom k jeho šírke pneumatiky
- Človek stráca slobodu v pohybe rôznych častiach tela, aby nestratil rovnováhu
  - Ruky držia riadidlá a nohy sú na pedáloch

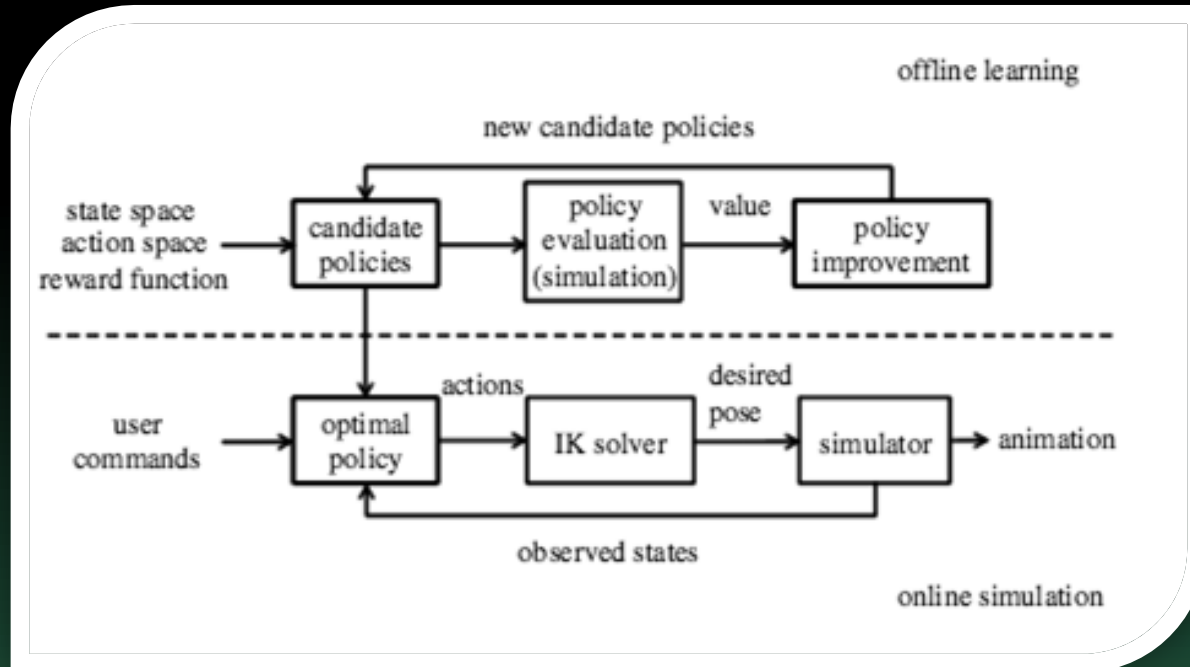
# Problematika

- Balance počas zastavenia na bicykli je oveľa náročnejší, než je obvyklé jazdenie.
- Ako príklad uvidíme obrázok, v ktorom jazdec zdvihne zadné koleso bicykla a udržuje rovnováhu na prednom kolese.

- Malá kontaktná oblasť kolesa
- Zdvihlo sa ťažisko
- Predné koleso je neschopné zrýchlenia
- Nestabilná póza, obtiažnosť rovnováhy počas zastavenia



# Riešenie problematiky



- Na vstupe **offline learning** máme state space, action space a reward function pre každú úlohu bicykla
- Jeho podsystem si vstupy rozdelí na množiny vhodných kandidátov a iteratívne ich vyhodnocuje a vyvíja pomocou CMA alebo NEAT
- Dokiaľ sa nenájde optimálna taktika na simuláciu

Pod systém online simulácie si najprv extrahuje pozorovateľné stavy, ako je uhol náklonu, rýchlosť padania a uhly nastavené používateľom pri riadidlách a pedáloch.

Následne sa použije optimálna taktika (vypočítaná v offline learning). Systém udržiava rovnováhu bicykla aj pri zadávaní príkazov od užívateľa (otáčanie riadidlami, pohyb).

Inverzná kinematika zabezpečuje dynamiku a pohyb jazdca na bicykli, aby simuloval skutočnosť.

# Bicycle and Rider Simulation

- Náš simulátor je založený na Open Dynamic Engine (ODE)
- Rozšírili sme ho na cyklistu a bicykel
- Každé tuhé teleso má 6 stupňov voľnosti

- $M$  - mass matrix
- $I$  - inertia tensor

$$\begin{bmatrix} M & \mathbf{0} \\ \mathbf{0} & I \end{bmatrix} \begin{bmatrix} \dot{\mathbf{v}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} = \begin{bmatrix} m\mathbf{g} \\ -I\dot{\boldsymbol{\omega}} \end{bmatrix} + \mathbf{J}^T \begin{bmatrix} \mathbf{f} \\ \boldsymbol{\tau} \end{bmatrix}$$

- $\mathbf{v}$  a  $\boldsymbol{\omega}$  sú lineárne a uhlová rýchlosť
- $\mathbf{f}$  a  $\boldsymbol{\tau}$  sú väzbové sily a točiacé momenty, ktoré pochádzajú od reťaze bicyklov, kĺbov a kontakty.
- $\mathbf{J}^T$  je transponovaná Jacobiho matica, ktorá mapuje väzbu síl a točiace momenty na telo

- Prenos sily z pedálov do zadného kolesa na bicykli
- $A$  a  $B$  - sú pedále a zadné koleso
- $\mathbf{n}$  - smer vychádzajúci z osi otáčania
- $\alpha$  – reprezentuje prevod rýchlosti

$$\mathbf{n}^T (\alpha \boldsymbol{\omega}_A - \boldsymbol{\omega}_B) = 0$$

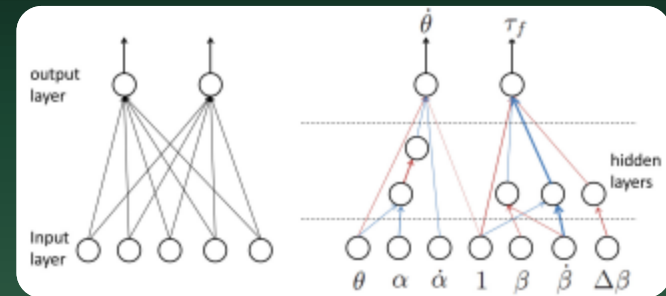
# Learning To Ride Bicycle

## Markov Decision Process

- MDP predstavuje tuple  $(S, A, R, D, P_{sas'}, \gamma)$
- $S$  – state space
- $A$  – action space
- $R$  – reward function
- $D$  – distribúcia začiatného stavu  $s_0$
- $P_{sas'}$  – pravdepodobnosť prechodu
- $\gamma$  – faktor vylúčenia, jeho hodnoty sú medzi 0 a 1
- Príklad – pri úlohe balansu na bicykli, si môžeme vybrať state space  $S$ , ktorý obsahuje uhol náklonu, rýchlosť naklonenia a uhol riadidiel bicykla. Môžeme vybrať akciu  $A$ , ktorá otáča riadidlami pri nejakej rýchlosti. Vyberáme potom reward function, ktorá povie, či daný stav vyhovuje – bicykel ostáva v spriamenej polohe, alebo nevyhovuje.
- Začiatkový stav je náhodný, ale taký aby bicykel bol v spriamenej polohe. Následne podľa faktoru vylúčenia vyhodnotíme, či tento stav vylúčiť alebo nie za použitia simulácie.

## Policy Parametrization

- Parametrizácia určuje potenciálnu kvalitu optimálnej taktiky. Pre nájdenie oboch štruktúr (neurónovú sieť, súčasťnú hmotnosť) použijeme NEAT, pretože nájde lepšie taktiky ako keby sme mali prehľadávať celú sieť fixne.



- Vľavo vidíme neurónovú sieť s input a output vrstvami, ktoré sú priamo prepojené. V pravo naučená neurónová sieť s použitím nášho algoritmu pre balansovanie na prednom kolese. Modré šípky reprezentujú negatívne váhy a červené pozitívne. Šírka šípok kóduje veľkosť závažia.



# Learning To Ride Bicycle

## Policy Evaluation

- Pri zhodnotení taktiky používame nasledovnú reward function v nasledujúcej forme

$$R(s) = R_t(s) + wR_r(s)$$

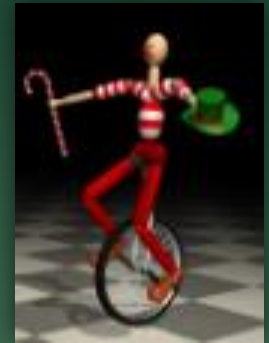
- Kde  $R_t$  a  $R_r$  sú špecifické úlohy a regulačné podmienky a  $w$  je váha.
- Táto reward function metóda vracia „ohodnotenie“ pre každú špecifickú úlohu iné (rôzne triky vyžadujú rôzne nastavenie)

## Policy Improvement

- Mnoho metód zlepšovania taktiky využíva gradient politiku, ktorá vykonáva iteratívne vzostupné operácie.
- Avšak, naša simulácia bicyklových trikov/kúskov zahŕňa časté, diskkrétne akcie, ako je zavedený a rozpínací kontakt, ktorý znehodnocuje informáciu gradientu. Z tohto dôvodu, používame vzorky na báze stochastickej optimalizačnej techniky. Aplikujeme CMA k hľadaniu radičov posuvov. Používame NEAT k hľadaniu regulátorov spätnej väzby, vrátane štruktúry závažia v neurónové sieti. NEAT má mnoho podobností do genetických algoritmov, ale je prispôsobený na vytváranie neurónových sietí.

# Výsledky

- Systém je implementovaný v C++
- Používame ODE na simulovanie obydvoch bicykla a cyklistu
- Simulácia beží v reálnom čase s 2.26GHz CPU a 4GB pamäti
- Bolo vygenerovaných 90 príkladov
- Finálny výsledok bol dosiahnutý pri 50 iteráciach
- Testy prebehli na rôznych typoch bicykloch
  - Road bikes
  - BMX bike
  - High wheelers
  - Unicycle



# Výsledky

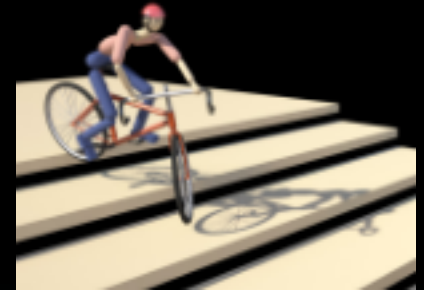
## Balance and steering

- Jazda na bicykli vyžaduje rovnováhu a riadenie. Vyváženie môže byť udržiavané pomocou riadenia smerom k padajúcemu smeru, čo vytvára odstredivú silu, aby sa bicykel dostal do vzpriamenej polohy.
- Testovali sme robustnosť balansu na nerovnom teréne
  - Cyklista stále dokáže vyrovnať a pozorne sleduje požadovaný smer.
  - Nohy sa učia politike rovnováhy, použitá IK pre genrovanie full-body motion



## Going down stairs

- Dole schodami bez spadnutia z bicykla
- Každý schod má 0.15m výšku a 0.8m široké
- Balance controller ako v predošlom príklade
- Náročnosť: častá strata kontaktu a náhla kolízia medzi pneumatikou a podlahou
- Cyklista musí robiť veľké manévry s volantom, aby udržal rovnováhu, lebo rýchlosť vpredu je nízka
- Ak bicykel ide rýchlejšie, opravy sa znižujú a sú stabilnejšie



# Výsledky



## Endo

- Endo zdvíhá zadné koleso a bicykel balansuje na prednom kolese
  - Vyžaduje úplne iné zručnosti a prostredie
  - Obvykle vykonávané v miernom zostupnom sklone, čo zabezpečuje potrebný pohyb vpred, keď zadné koleso je zdvihnuté nad zemou
  - Použitý sklon je 2.5 stupňa
  - Výsledný kontrolér pomaly pohybuje panvu cyklistu dozadu a potom rýchlo sa vrhne dopredu
- Cyklista neustále stláča alebo uvoľňuje prednú brzdú na udržanie rovnováhy a riadi riadidlá, aby udržal rovnováhu

## Front wheel pivot

- Predné koleso (pivot) je rýchly spôsob ako otočiť bicykel o 180 stupňov na prednom kolese.
- Použili sme feed-forward kontroler a aplikovali sme ho na 2 samostatné úlohy počas 2 fáz tohoto pohybu.
- Prvá funkcia maximalizuje uhol otáčania počas pivot fázy
- Potom ako sa zadné koleso dotkne zeme, prepneme kontroler na naučenie sa balansovať a meráme, ako dlho koleso vydrží balansovať
- Bez druhej reward funkcie, politika "optimality" môže vytvárať veľké role v priebehu pivot fázy, po ktorom cyklista nemôže obnoviť rovnováhu.
- V animácií, po otočení sa cyklistu ostro do ľava na prednom kolese, vykoná endo.
  - Výsledkom je otočenie sa cyklistu na prednom kolese okolo 180 stupňov, ktoré sa ukončí do 3 sekúnd

# Výsledky



## Bunny hop

- Cyklista skáče s bicyklom cez prekážky
- Tesne pred skokom, cyklista sa najprv nakloní dopredu a potom sa jeho panva rýchlo pohne do zadnej časti bicykla
- Tento výrazný pohyb naklápa koleso smerom nahor. Cyklista potom skáče s bicyklom cez klauna ležiacého na zemi.
- Optimálna politika pre bunny hop pohybu obsahuje fázu, ktorá nakláňa bicykel hore
  - je nevyhnutný pre zoskoky pre väčšie výšky a vzdialenosti



## Riding a unicycle

- Jednokolky sa nemusia riadiť riadidlami. Ak chceme riadiť jednokolku, cyklista musí krútiť jeho hornú časť tela.
- Ak chceme začať na jednokolobežke jazdiť, najprv začneme pedálovať dozadu, čo nakláňa kolobežku dopredu. Potom zrýchlime až kým nedosiahneme požadovanú rýchlosť.
- Počas jazdenia, točíme pásom aby bicykel našiel rovnováhu, čo spôsobuje, že jednokolobežka jazdí po sinusoide

P. Kunovský, L. Ďurikovičová

**ĎAKUJEME ZA POZORNOST**