



## Spring-loaded Inverted Pendulum (SLIP)

<พบบเพชร เพชรขำลี, 65340500041>

<วิศยศ โตศิริ, 65340500051>

<อภิขญา ศรีวงศ์, 65340500059>

### บทคัดย่อ

โครงการนี้มีวัตถุประสงค์ในการวิเคราะห์การเคลื่อนที่ของระบบ Spring-loaded Inverted Pendulum (SLIP) ซึ่งเป็นแบบจำลองพื้นฐานสำหรับการกระโดดของหุ่นยนต์สองขาหรือสิ่งมีชีวิต โดยเน้นการใช้ Lagrangian Mechanics เพื่อสร้างสมการการเคลื่อนที่สำหรับการวิเคราะห์แรงและพลังงานที่จำเป็นในระหว่างเฟสการสัมผัสพื้นของการเคลื่อนที่ โครงการนี้จำลองพฤติกรรมของระบบโดยไม่มี Closed-loop control เพื่อตรวจสอบพฤติกรรมการเคลื่อนที่ใน Simulation

การวิเคราะห์ใช้การควบคุมแบบ PD Control เบื้องต้นเพื่อปรับค่าความแข็งของสปริงในบางกรณี ผลการจำลองแสดงถึงพฤติกรรมการเคลื่อนที่และการกระจายของแรงในระบบ SLIP ซึ่งสามารถนำไปเป็นพื้นฐานในการพัฒนาระบบกระโดดของหุ่นยนต์สองขาในอนาคต

**คำสำคัญ:** Spring-loaded Inverted Pendulum (SLIP), Lagrangian Mechanics, PD Control

## 1. บทนำ (Introduction)

ในปัจจุบันการศึกษาและพัฒนาระบบที่เลียนแบบการเคลื่อนที่ของสิ่งมีชีวิตเพื่อใช้ในหุ่นยนต์ขับเคลื่อนกลายเป็นที่นิยม โดยเฉพาะอย่างยิ่งการเคลื่อนที่แบบกระโดดที่ช่วยให้หุ่นยนต์สามารถเคลื่อนที่ในพื้นที่ซับซ้อนได้อย่างคล่องตัว ระบบ Spring-loaded Inverted Pendulum (SLIP) เป็นแบบจำลองพื้นฐานที่มักใช้ในการศึกษาพฤติกรรมการกระโดดของหุ่นยนต์สองขาหรือสิ่งมีชีวิต เนื่องจากเป็นระบบที่แสดงถึงความยืดหยุ่นและการขับเคลื่อนที่สามารถรองรับแรงกระแทกและรักษาสถิตได้ดี การศึกษาพฤติกรรมการเคลื่อนที่และแรงในระบบ SLIP จึงเป็นประโยชน์สำหรับการพัฒนาหุ่นยนต์ที่สามารถเคลื่อนที่แบบกระโดดได้อย่างมีประสิทธิภาพ

### 1.1 จุดประสงค์โครงการ

- 1) เพื่อวิเคราะห์การเคลื่อนที่และพฤติกรรมของระบบ SLIP โดยใช้ Lagrangian Mechanics ในการสร้างสมการการเคลื่อนที่และการคำนวณแรงที่จำเป็นในระหว่างเฟสสัมผัสพื้น
- 2) เพื่อศึกษาการควบคุมเบื้องต้นโดยใช้ PD Control สำหรับปรับค่าความแข็งของสปริงเพื่อรักษาความเสถียรในบางกรณี
- 3) เพื่อสร้างแบบจำลองและการจำลองพฤติกรรมของระบบ SLIP โดยไม่มีการควบคุมแบบปิด (Closed-loop Control) เพื่อตรวจสอบพฤติกรรมของระบบใน Simulation

### 1.2 ขอบเขต

- 1) ศึกษาการเคลื่อนที่ของระบบ Spring-loaded Inverted Pendulum (SLIP) ในรูปแบบ 3 มิติจะครอบคลุมเฉพาะในบางด้าน ดังนี้
  - A. การสร้างสมการการเคลื่อนที่ด้วย Lagrangian Mechanics
    - I. พิจารณาเฉพาะพลังงานจลน์และพลังงานศักย์ของระบบ เพื่อสร้างสมการการเคลื่อนที่
    - II. การวิเคราะห์จะครอบคลุมทั้งเฟสสัมผัสพื้น (Stance Phase) และเฟสลอยตัว (Flight Phase)

## B. ตัวแปรที่ต้องการพิจารณาในระบบ

I. สามารถปรับค่า parameter ต่างๆ เช่น มวล (mass), ความแข็งของสปริง (K), และระยะสปริงเริ่มต้น ( $r_0$ )

II. การใช้ค่าคงที่ (fixed parameter) สำหรับบางส่วนของระบบ เช่น ตำแหน่งเริ่มต้นของมวล, จำนวนรอบในการกระโดด

## C. การวิเคราะห์และแสดงผลของระบบ

I. วิเคราะห์เฉพาะแรงที่กระทำในทิศทางแนวดิ่ง และไม่รวมแรงในแนวระนาบ

II. การทดลองจะจำกัดเพียงการจำลองใน Simulation เท่านั้น

## 2) การจำลองระบบในสภาพแวดล้อมซอฟต์แวร์

A. การจำลองจะใช้ PyBullet ในการสร้างแบบจำลอง 3 มิติ

B. การสร้าง Toolbox ที่ช่วยในการปรับค่า parameter ต่างๆ เพื่อทดสอบพฤติกรรมของระบบ

## 3) การควบคุมการเคลื่อนที่ (Controller)

A. การใช้ PD Control ในการควบคุมองศาการกระโดดของ Spring-loaded Inverted Pendulum

I. การปรับค่าพารามิเตอร์ PD (Proportional-Derivative) เพื่อทดสอบผลต่อการควบคุมการเคลื่อนที่ของระบบ

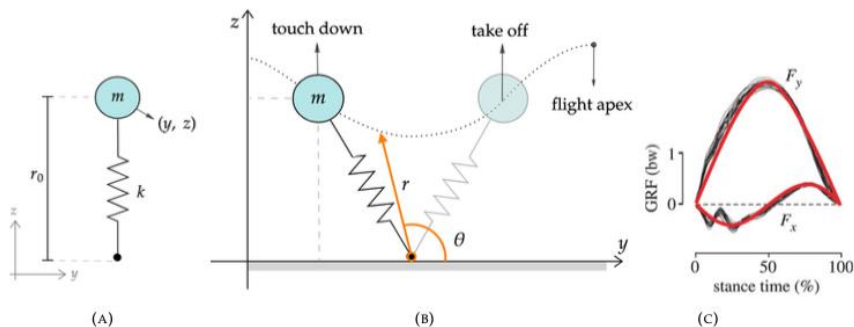
## 2. ทบทวนวรรณกรรม และ ทฤษฎีที่เกี่ยวข้อง (Literature Review)

### 2.1 Guiding policies with the Spring-Loaded Inverted Pendulum (Daniel F., 2021) [1]

#### 2.1.1 SLIP: Spring Loaded Inverted Pendulum

เป็นโมเดลทางพลศาสตร์ที่มีความซับซ้อนต่ำ ซึ่งใช้ในการอธิบายการเคลื่อนไหวของระบบที่มีขา (legged locomotion) ในแนวระนาบ (sagittal plane) โดยโมเดลนี้ช่วยให้สามารถคำนวณการเคลื่อนที่ด้วยความเร็วสูงได้อย่างง่ายดาย โดยไม่ซับซ้อนทางคณิตศาสตร์ ในรูปแบบที่ง่ายที่สุด โมเดลนี้ประกอบด้วย:

- Point-mass ( $m$ ): ใช้แทนจุดศูนย์กลางมวล (Center of Mass - CoM) ของตัววัตถุ
- Massless spring leg: ขาแบบสปริงที่ไม่มีมวล ใช้จำลองความยืดหยุ่นของขาระหว่างการสัมผัสพื้น



รูปที่ 1 การเคลื่อนไหวของ Spring-Loaded Inverted Pendulum (SLIP)

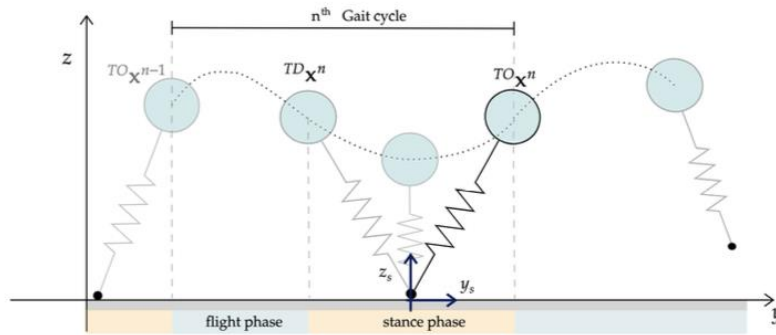
ภาพรวมของการเคลื่อนที่ใน SLIP Model:

- Touch Down: เริ่มต้นการสัมผัสพื้น
- Take Off: ขาหลุดจากพื้นและเริ่มเฟลลอยตัว
- Flight Apex: จุดสูงสุดของการลอยตัวก่อนเริ่มรอบถัดไป

โมเดลนี้จำลองการเคลื่อนไหวด้วยการตั้งขึ้น-ลงของมวลบนขาที่มีความยืดหยุ่นแบบสปริง ซึ่งโมเดลคาดการณ์แรงปฏิกิริยาที่เกิดจากพื้นในแนวแกนนอนและแนวแกนตั้ง (GRF) ระหว่างเฟสสัมผัสพื้น โดยกราฟ (C) แสดงการเปรียบเทียบระหว่างแรงที่เกิดขึ้นจริงจากการวิ่งของมนุษย์และสัตว์ กับแรงที่โมเดล SLIP ทำนาย

#### 2.2.2 Model Formulation

โมเดล SLIP เป็นแบบจำลองไดนามิกแบบไฮบริดที่ใช้เพื่ออธิบายการเคลื่อนที่ของศูนย์กลางมวล (Center of Mass) ในระหว่างเฟสการลอยตัว (flight phase) และเฟสสัมผัสพื้น (stance phase) ของรอบการเดินหรือวิ่ง (gait cycle)



รูปที่ 2 Gait Cycle ของ Spring-Loaded Inverted Pendulum (SLIP)

- Flight Phase: ในระหว่างเฟสการลอยตัว โมเดลจะไม่สัมผัสพื้น และการเคลื่อนที่ของศูนย์กลางมวลจะเป็นลักษณะเดียวกับวัตถุที่เคลื่อนที่ในแนวโปรเจกไทล์ โดยมีสมการ

$$\begin{bmatrix} m\ddot{y} \\ m\ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ -mg \end{bmatrix}$$

เมื่อ  $m$  คือมวล

$g$  คือความเร่งเนื่องจากแรงโน้มถ่วง

- Stance Phase: ในระหว่างเฟสสัมผัสพื้น จุดมวลจะติดตัวบนขาที่มีสปริง รองรับการเคลื่อนที่ โดยใช้ฟังก์ชันเชิงเส้นในการอธิบายตำแหน่งของศูนย์กลางมวล จาก Lagrangian จะได้สมการการเคลื่อนที่ดังนี้

$$\begin{bmatrix} m\ddot{r} \\ mr\ddot{\theta} \end{bmatrix} = \begin{bmatrix} mr\dot{\theta}^2 - k(r - r_0) - mg\sin(\theta) \\ -mgr\cos(\theta) - 2mr\dot{\theta} \end{bmatrix}$$

เมื่อ  $r$  และ  $\theta$  คือระยะทางและมุมที่สัมพันธ์กับศูนย์กลางมวล

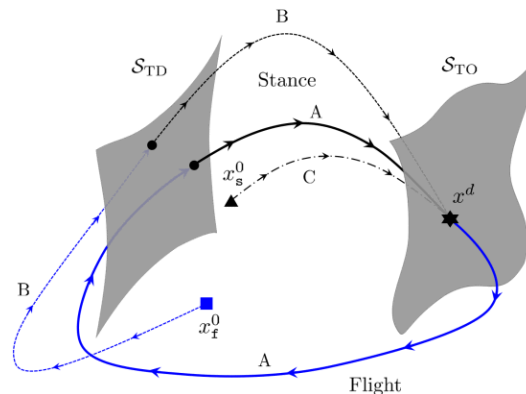
$k$  คือ ค่าความแข็งของสปริง

$r_0$  คือ ความยาวขาขณะพัก

## 2.2 Optimal Control of a Differentially Flat Two-Dimensional Spring-Loaded Inverted Pendulum Model (Hua C. & Patrik M. & Wei Z., 2020) [2]

### 2.2.1 Optimal Control of the Differentially Flat SLIP Model

ควบคุมความเร็วเชิงมุมของขาในช่วง flight และการปรับความยาวขาและแรงบิดที่สะโพกในช่วง stance เพื่อให้การก้าวมีความเสถียร โดยมีวัตถุประสงค์ให้การเริ่มต้นก้าวถัดไปนั้นตรงตามสถานะที่ต้องการ



### รูปที่ 3 Phase Diagram of SLIP Model Showing Stance and Flight Phases with Optimal Control Trajectories

- $S_{TO}$  (Stance Take-Off) และ  $S_{TD}$  (Stance Touch-Down): พื้นที่สี่เหลี่ยมทางขวาและซ้ายแทนสถานะ take-off และ touch-down ของขาในช่วงการเคลื่อนที่ เมื่อขาอยู่ในสถานะใดสถานะหนึ่งจะเกิดการเปลี่ยนเฟสระหว่างการสัมผัสพื้น (stance) และการลอยตัว (flight)
- Trajectory A, B, C: เส้นทางการเคลื่อนที่ที่ต่างกัน
  - เส้นทาง A (Flight Phase): แสดงด้วยเส้นสีน้ำเงิน ซึ่งแสดงถึงช่วงการลอยตัวของขา เริ่มต้นจากจุดสัมผัสพื้นแล้วต่อเนื่องไปยังจุด take-off
  - เส้นทาง B: เส้นประสีดำ เป็นอีกหนึ่งวิธีการเคลื่อนที่ของระบบที่มีลักษณะต่างจากเส้นทาง A
  - เส้นทาง C: เส้นประสีดำจุด-ขีด ซึ่งแสดงถึงการเปลี่ยนแปลงที่เกิดขึ้นระหว่างการเคลื่อนไหว
- $x_f^0$  และ  $x_s^0$  : ตำแหน่งเริ่มต้นในช่วง flight และ stance ซึ่งระบุตำแหน่งของขา ณ จุดเริ่มต้นของการเคลื่อนที่ในแต่ละเฟส
- $x_d$ : จุดมุ่งหมายของสถานะที่ต้องการให้ไปถึงในช่วง take-off

#### Stance Phase Optimal Control - Flatness-Based Solution

ควบคุมในช่วง Stance Phase ด้วยวิธี Flatness-Based Solution มีวัตถุประสงค์เพื่อหาค่าที่เหมาะสมที่สุดของสัญญาณการกระจัด (linear actuator displacement) และแรงบิดที่สะโพก (hip torque) สำหรับสถานะเริ่มต้นที่กำหนด โดยมีการตั้งสมการควบคุมดังนี้:

##### 1) สมการเบื้องต้น

เริ่มจากการตั้งปัญหาการหาค่าที่เหมาะสมในช่วง Stance Phase ตามสมการ

$$\min_{T_s, u} \int_0^{T_s} c(x_s, u) dt + c_T(x_s(T_s))$$

โดยเงื่อนไข คือ

$$\begin{aligned} x_s &= f_s^{\cdot}(x_s, u), \\ x_s(0) &= x_s^0, u(0) = u^0, \\ (x_s(T_s), u(T_s)) &\in S_{TO} \end{aligned}$$

ในสมการข้างต้น:

- $T_s$  คือช่วงเวลา (time horizon) ที่ใช้ในการคำนวณในช่วง Stance Phase
- $u$  คือ input (สัญญาณควบคุม) ที่จะใช้ควบคุมระบบ
- $c(x_s, u)$  คือค่าใช้จ่าย (cost function) ที่ขึ้นกับความแตกต่างของค่าปัจจุบันจากค่าอ้างอิง
- $c_T(x_s(T_s))$  คือค่าใช้จ่ายตอนจบช่วงเวลา ซึ่งกำหนดเพื่อให้ใกล้เคียงกับสถานะที่ต้องการในช่วง Take-off

##### 2) การใช้ Differential Flatness

ระบบ SLIP สามารถแปลงสมการที่ซับซ้อนให้อยู่ในรูปของ flat outputs ได้ ซึ่งทำให้สามารถแปลงปัญหาการควบคุมไปเป็นการหาค่าที่เหมาะสมใน flat space แทน ทำให้การคำนวณง่ายขึ้นและสามารถแก้ไขได้ด้วยวิธีการเชิงตัวเลขอย่าง Quadratic Programming (QP)

##### 3) สมการควบคุมใน Flat Space

- ใช้ Polynomial Approximation เพื่อประมาณค่า flat outputs และอนุพันธ์ของมัน ทำให้สามารถหาค่าควบคุมที่เหมาะสมในรูปของพหุนาม ซึ่งง่ายต่อการคำนวณและปรับค่าให้สอดคล้องกับ trajectory ที่กำหนด
- การตั้งปัญหาให้อยู่ในรูปของ QP ช่วยให้ได้ค่าควบคุมที่เหมาะสมที่สุดได้อย่างมีประสิทธิภาพ

##### 4) การตั้งค่า Trajectory และค่าเป้าหมาย

ค่าอ้างอิงและค่าเป้าหมายที่ใช้ใน flat space จะกำหนดล่วงหน้า เพื่อให้ SLIP เคลื่อนที่ได้อย่างมีประสิทธิภาพและบรรลุเป้าหมายการเคลื่อนที่อย่างปลอดภัยในแต่ละช่วงของ Stance Phase

#### Flight Phase Optimal Control

เน้นการหาค่าควบคุมที่ดีที่สุดเพื่อให้ระบบมีการเคลื่อนที่ที่เสถียรและบรรลุเป้าหมายที่กำหนด โดยในช่วงบินนั้น โครงสร้างของปัญหาจะเป็นการหามุมการสัมผัสพื้น (touch down angle) ของขาที่สัมพันธ์กับสถานะการบิน (flight state) เริ่มต้น  $x_f^0$  และยังต้องหาค่าของความเร็วเชิงมุมของขา ระหว่างบิน  $\omega$  ที่สอดคล้องกัน

#### 1) ตั้งสมการ Optimization Problem

เพื่อหาค่า  $\omega$  และ  $T_F$  ที่เหมาะสม โดยมีวัตถุประสงค์ในการลดการเบี่ยงเบนของสถานะจากค่าอ้างอิง (Reference) ด้วยการลดค่า Running Cost  $c(x_f, \omega)c(x_f, \omega)$  ซึ่งคำนวณจาก  $\|x - x_{ref}\|_Q^2 dt + \|\omega - \omega_{ref}\|_R^2$  และ Terminal Cost  $V_S$  ซึ่งคำนวณจากฟังก์ชันค่า (Value Function) ของช่วงสัมผัสพื้น  $V_S(T_{fs}(x_s(T_F)))$

#### 2) ข้อจำกัดของระบบ

- สมการการเคลื่อนที่ในช่วงบิน  $\dot{x}_f = f_f(x_f, \omega)$
- ช่วงค่าของความเร็วเชิงมุม
- สถานะสุดท้าย  $x_f(T_F) \in S_{TD}$  ต้องอยู่บนแมนิโฟลด์สัมผัสพื้น (Touch Down Manifold)

#### 3) การคำนวณระยะเวลา

$T_F$ : ระยะเวลาที่ใช้ในช่วงบิน  $T_F$  นั้นถูกคำนวณจากสมการ

$$\frac{1}{2}gT_F^2 - x_{f,4}^0 T_F + l_0 \sin\left(x_{f,5}^0 + \int_0^{T_F} \omega(t)dt\right) - x_{f,3}^0 = 0$$

ซึ่งเป็นสมการเชิงพาราโบลาแบบล้อม (Implicit Quadratic Equation) ที่ซับซ้อน ทำให้การแก้ปัญหานี้มีความท้าทายในการหาเวลาที่เหมาะสม  $T_F$

#### 4) ข้อสรุปจาก Proposition

จากการสังเกตข้อจำกัดของการสัมผัสพื้น (Touch Down Manifold) พบว่ามีการจำกัดมิติของแมนิโฟลด์สัมผัสพื้นไว้เพียง 1 มิติ ที่สามารถใช้พารามิเตอร์มุมสัมผัส  $\theta_f$  เป็นตัวแปรหลักในการกำหนดเงื่อนไขการสัมผัสพื้นได้

- ด้วยการเปลี่ยนตัวแปรให้เป็น  $\theta_f = x_{f,5}^0 + \int_0^{T_F} \omega dt$
- เราสามารถคำนวณระยะเวลาช่วงบิน TTF ได้โดยการแก้สมการ ดังนี้

$$T_F = \frac{1}{g}(x_{f,4}^0 + \sqrt{(x_{f,4}^0)^2 - 2g(l_0 \sin(\theta_f) - x_{f,3}^0)})$$

#### 5) การหาค่าที่เหมาะสมของมุมสัมผัสพื้น $\theta_f^*$

เพื่อให้การคำนวณง่ายขึ้น ปัญหาการหาค่าที่เหมาะสมนี้ถูกย่อยออกเป็นปัญหาการหาค่าเพียง 1 มิติสำหรับ  $\theta_f$  โดยค่า  $\theta_f^*$  ที่เหมาะสมจะถูกค้นหาวัยวิธี Numerical Optimization เช่น Gradient Descent หลังจากได้ค่า  $\theta_f^*$  แล้ว ความเร็วเชิงมุมของขา  $\omega$  สามารถคำนวณได้ดังนี้

$$\omega = \min\{\underline{\omega}, \max\{\underline{\omega}, \frac{\theta_f^* - x_{f,5}^0}{T_F}\}\}$$

## 2.3 ALGORITHMS FOR WALKING, RUNNING, SWIMMING, FLYING, AND MANIPULATION (RUSS TEDRAKE, 2024) [3]

### 2.3.1 LINEAR QUADRATIC REGULATOR (LQR)

LQR ถูกสร้างขึ้นมาเพื่อให้ระบบเชิงเส้นที่ไม่เปลี่ยนแปลงตามเวลา มีความเสถียรเข้าสู่จุดกำเนิด โดยใช้สำหรับแก้ปัญหา DYNAMIC PROGRAMMING สำหรับ CONTINUOUS SYSTEMS โดยส่วนใหญ่แล้วกรณีเหล่านี้จะเกี่ยวข้องกับระบบที่มีพลศาสตร์เชิงเส้นและต้นทุนที่เป็นฟังก์ชันนูน (เช่น ฟังก์ชันกำลังสอง)

พิจารณาระบบเชิงเส้นที่ไม่เปลี่ยนแปลงตามเวลาในรูปแบบของสถานะ (STATE-SPACE FORM)

$$\dot{x} = Ax + Bu$$

โดยมี COST FUNCTION แบบอินทิกรัลไม่สิ้นสุด (INFINITE-HORIZON COST FUNCTION) ให้โดย

$$J = \int_0^{\infty} (x^T Q x + u^T R u) dt$$

โดย  $Q = Q^T \geq 0$  และ  $R = R^T > 0$

เป้าหมายของเราคือการหา COST FUNCTION ที่เหมาะสมที่สุด  $J^*(x)$  ซึ่งต้องสอดคล้องกับสมการ HAMILTON-JACOBI-BELLMAN (HJB) ดังนี้

$$\forall x, 0 = \min_u [x^T Q x + u^T R u + \frac{\partial J^*}{\partial x} (Ax + Bu)]$$

มีขั้นตอนสำคัญหนึ่งขั้นตอน คือ ในปัญหานี้ COST FUNCTION ที่เหมาะสมที่สุดคือควอดราติก (QUADRATIC) จึงกำหนดให้

$$J^*(x) = x^T S x, S = S^T \geq 0$$

เกรเดียนต์ของฟังก์ชันนี้คือ

$$\frac{\partial J^*}{\partial x} = 2x^T S$$

เนื่องจากเราได้กำหนดให้ค่าในฟังก์ชัน  $\min$  นั้นเป็นควอดราติกและเป็นนูน (เนื่องจาก  $R > 0$ ) เราจึงสามารถหาค่าต่ำสุดโดยตรงได้โดยการหาจุดที่เกรเดียนต์ของค่าเหล่านั้นเป็นศูนย์

$$\frac{\partial}{\partial x} = 2u^T R + 2x^T S B = 0$$

ดังนั้น เราจะได้เงื่อนไขการควบคุมที่เหมาะสมที่สุด (optimal policy) เป็น

$$u^* = \pi^*(x) = -R^{-1} B^T S x = -K x$$

เมื่อแทนค่ากลับเข้าไปในสมการ HJB และจัดรูปจะได้ว่า

$$0 = x^T [Q - S B R^{-1} B^T S + 2SA] x$$

โดยคำทั้งหมดนี้เป็นสมมาตร ยกเว้น  $2SA$  แต่เนื่องจาก  $x^T S A x = x^T A^T S x$  เราจึงสามารถเขียนได้ว่า

$$0 = x^T [Q - S B R^{-1} B^T S + SA + A^T S] x$$

เนื่องจากเงื่อนไขนี้ต้องเป็นจริงสำหรับทุก  $x$  เราจึงสามารถพิจารณาสมการเมทริกซ์

$$0 = SA + A^T S - S B R^{-1} B^T S + Q$$

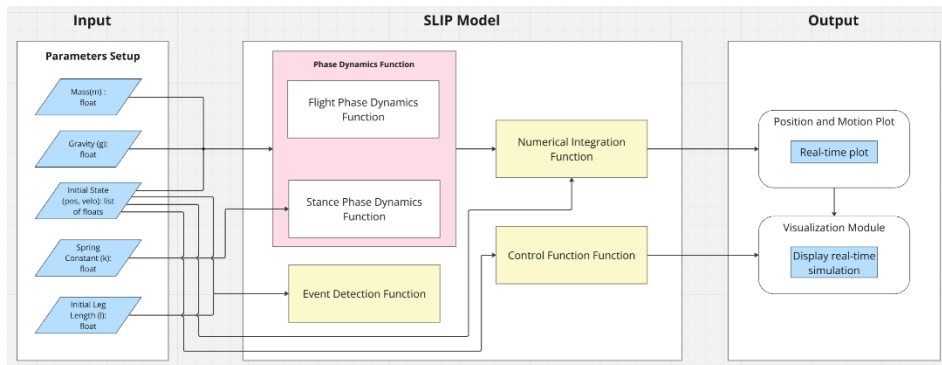
สมการนี้เป็นเวอร์ชันหนึ่งของสมการ RICCATI เชิงพีชคณิต (ALGEBRAIC RICCATI EQUATION) ซึ่งควอดราติกใน  $S$  ทำให้การหาค่าผลลัพธ์ไม่ง่าย แต่เป็นที่ทราบกันดีว่าสมการนี้มีผลลัพธ์ที่เป็น POSITIVE-DEFINITE เพียงหนึ่งเดียวก็ต่อเมื่อระบบสามารถควบคุมได้ นอกจากนี้ยังมีวิธีการทางตัวเลขที่ดีในการหาค่าผลลัพธ์นี้ แม้ในกรณีที่มีมิติสูง ทั้งเงื่อนไขการควบคุมที่เหมาะสมที่สุดและฟังก์ชันต้นทุนที่เหมาะสมที่สุดสามารถหาได้จาก DRAKE โดยเรียก  $(K, S) = \text{LinearQuadraticRegulator}(A, B, Q, R)$  ถ้ารูปแบบควอดราติกของฟังก์ชันต้นทุนดูเหมือนจะน่าฉงน ลองพิจารณาการแก้สมการของระบบเชิงเส้น  $\dot{x} = (A - BK)x$  ซึ่งมีรูปแบบ  $x(t) = e^{(A-BK)t}x(0)$  แล้วแทนค่ากลับเข้าไปในฟังก์ชันต้นทุนเชิงอินทิกรัล จะเห็นได้ว่าต้นทุนมีรูปแบบเป็น  $J = x^T(0)Sx(0)$

การพิจารณารูปแบบของเงื่อนไขการควบคุมที่เหมาะสมเป็นสิ่งที่มีความสำคัญ เนื่องจากฟังก์ชันมูลค่าตัวแทนต้นทุนที่เหลือน้อย การกระทำที่เหมาะสมจะนำไปในทิศทางที่ลดต้นทุนนี้ให้เร็วที่สุดเท่าที่จะเป็นไปได้ โดยทั่วไป  $-Sx$  เป็นทิศทางของการลดลงที่เร็วที่สุดของฟังก์ชันมูลค่า แต่ไม่ใช่ว่าทิศทางทั้งหมดจะเป็นไปได้ในการควบคุมระบบเชิงสถานะ เมทริกซ์  $-B^T Sx$  แสดงถึงการฉายภาพของการลดลงที่เร็วที่สุดเข้าสู่พื้นที่ของการกระทำ และเป็นการลดลงที่เร็วที่สุดที่สามารถทำได้ผ่านอินพุตควบคุม  $u$  ในที่สุด การปรับสเกลล่วงหน้าโดยเมทริกซ์  $R^{-1}$  จะทำให้การลดลงในทิศทางของการลดลงสะท้อนถึงการกำหนดน้ำหนักต้นทุนที่เรากำหนดในอินพุตควบคุมต่าง ๆ

### 3. เนื้อหาในรายวิชาที่เกี่ยวข้อง

- 1) Spatial Transformations
- 2) Forward Kinematics
- 3) Differential Kinematics
- 4) Dynamics

### 4. System Diagram / System Overview (Function and Argument)



รูปที่ 4 System Diagram

#### 1. โมดูลตั้งค่าพารามิเตอร์เริ่มต้น (Input Parameters Module)

- **วัตถุประสงค์:** กำหนดค่าพารามิเตอร์ทางกายภาพของระบบ SLIP
- **Argument:**
  - $m$  : มวลของจุดที่แทนร่างกาย
  - $k$  : ความแข็งของสปริงที่ขา
  - $l$  : ความยาวธรรมชาติของขาเมื่อไม่ถูกแรงกด
  - $g$  : ค่าความเร่งเนื่องจากแรงโน้มถ่วง
  - $y0$  : ตำแหน่งเริ่มต้นในแนวนอน
  - $z0$  : ความสูงเริ่มต้นจากพื้น

- $vy0$  : ความเร็วเริ่มต้นในแนวตั้ง
- $vx0$  : ความเร็วเริ่มต้นในแนวนอน

## 2. ฟังก์ชันคำนวณพลศาสตร์ในเฟสลอยตัว (Flight Phase Dynamics Function)

- วัตถุประสงค์: กำหนดสมการการเคลื่อนที่ระหว่างเฟสลอยตัว (เมื่อขาลอยจากพื้น)
- Argument:
  - $state$  : แสดงสถานะปัจจุบัน ประกอบด้วย  $y, z, y', z'$
  - $g$  : ค่าความเร่งเนื่องจากแรงโน้มถ่วง

## 3. ฟังก์ชันคำนวณพลศาสตร์ในเฟสสัมผัสพื้น (Stance Phase Dynamics Function)

- วัตถุประสงค์: กำหนดสมการการเคลื่อนที่ระหว่างเฟสสัมผัสพื้น (เมื่อขาสัมผัสพื้น)
- Argument:
  - $state$  : แสดงสถานะปัจจุบัน ประกอบด้วย  $r, \theta, r', \theta'$
  - $m$  : มวลของจุดที่แทนร่างกาย
  - $k$  : ความแข็งของสปริงที่ขา
  - $g$  : ค่าความเร่งเนื่องจากแรงโน้มถ่วง

## 4. โมดูลตรวจจับเหตุการณ์ (Event Detection Module)

- วัตถุประสงค์: ตรวจจับการเปลี่ยนเฟสระหว่างเฟสลอยตัวและเฟสสัมผัสพื้น เช่น เหตุการณ์การสัมผัสพื้นและการลอยขึ้น
- Argument:
  - $state$  : แสดงสถานะปัจจุบัน ประกอบด้วย
  - $l$  : ความยาวธรรมชาติของขาเมื่อไม่ถูกแรงกด ใช้สำหรับตรวจจับเฟสสัมผัสพื้น

## 5. โมดูลการอินทิเกรตเชิงตัวเลข (Numerical Integration Module)

- วัตถุประสงค์: ทำการอินทิเกรตสมการการเคลื่อนที่ตามเวลา โดยคำนึงถึงการตรวจจับเหตุการณ์สำหรับการสลับเฟส
- Argument:
  - `dynamics_function` (ฟังก์ชันพลศาสตร์): ฟังก์ชันสำหรับเฟสปัจจุบัน (ลอยตัวหรือสัมผัสพื้น)
  - `initial_state` (เวกเตอร์สถานะ): ค่าสถานะปัจจุบัน
  - `timestep` (ช่วงเวลา): ระยะเวลาของแต่ละขั้นในการอินทิเกรต

## 6. โมดูลฟังก์ชันควบคุม (Control Function Module - Optional)

- วัตถุประสงค์: ใช้กลยุทธ์ควบคุมเพื่อรักษาหรือปรับเสถียรภาพ เช่น ปรับมุมสัมผัสพื้นหรือลดแรงสปริง
- Argument:
  - `state` (เวกเตอร์สถานะปัจจุบัน): ค่าตำแหน่งและความเร็วปัจจุบัน
  - `target_values` (ค่าที่ต้องการ): ความสูงเป้าหมาย มุม หรือเกณฑ์เสถียรภาพที่ต้องการ

## 7. โมดูลแสดงผล (Output Module - Simulation Results)

- วัตถุประสงค์: เก็บข้อมูลการจำลองและแสดงผลลัพธ์ เช่น เส้นทางการเคลื่อนที่และการเปลี่ยนเฟส
- Argument:
  - `Display`: real-time simulation



5. ผลการศึกษาที่คาดหวัง

- 1) Toolbox ที่สามารถใช้ในการคำนวณ Dynamics ของระบบ SLIP โดยใช้ Lagrangian Mechanics เพื่อสร้างสมการการเคลื่อนที่และคำนวณแรงที่เกิดขึ้นในระบบได้
- 2) สามารถสร้าง Toolbox ที่รองรับการควบคุมด้วย PD Control สำหรับการปรับค่าความแข็งของสปริง เพื่อทดสอบผลกระทบต่อการเคลื่อนที่และเสถียรภาพของระบบ SLIP
- 3) สามารถสร้าง Toolbox สำหรับการจำลอง (Simulation) การเคลื่อนที่ของระบบ SLIP ในเฟสสัมผัสพื้น (Stance Phase) และเฟสลอยตัว (Flight Phase) รวมถึงการแสดงผล Visualization ของการเคลื่อนที่และการกระจายพลังงานในระบบ

6. รายละเอียดโครงการ

ลำดับ	การดำเนินงาน	สัปดาห์ที่ 1	สัปดาห์ที่ 2	สัปดาห์ที่ 3	สัปดาห์ที่ 4	สัปดาห์ที่ 5	สัปดาห์ที่ 6
1	ศึกษาข้อมูลเกี่ยวกับ Spring-loaded Inverted Pendulum (SLIP)						
2	ศึกษาข้อมูลเกี่ยวกับระบบควบคุม						
3	คำนวณสมการที่เกี่ยวข้องและกำหนดโมเดลทางคณิตศาสตร์						
4	ออกแบบ PD Controller เพื่อควบคุมการเคลื่อนไหว						
5	ทดสอบระบบและปรับพารามิเตอร์ให้เหมาะสม						
6	วิเคราะห์ข้อมูลจากการจำลองและสรุปผล						

7. เอกสารอ้างอิง (References)

[1] Ordoñez Apraez, D. F. (2021). Learning to run naturally: Guiding policies with the Spring-Loaded Inverted Pendulum(Master’s thesis). Universitat Politècnica de Catalunya.

[2] Hua C. & Patrik M. & Wei Z. (2020). *Optimal control of a differentially flat two-dimensional spring-loaded inverted pendulum model*. Southern University of Science and Technology, Shenzhen, China

[3] RUSS TEDRAKE. (2024). *Algorithms for Walking, Running, Swimming, Flying, and Manipulation*. Underactuated Robotics