

POLITECHNIKA WROCŁAWSKA
WYDZIAŁ ELEKTRONIKI

KIERUNEK: Automatyka i Robotyka (AIR)
SPECJALNOŚĆ: Robotyka (ARR)

PROJEKT INŻYNIERSKI

System symulacji zachowania układów
dynamicznych

Dynamical systems simulation package

AUTOR:
Rafał Skocelas

PROWADZĄCY PROJEKT:
dr inż. Robert Muszyński, I-6

OCENA PROJEKTU:

Spis treści

1	Wprowadzenie	2
2	Metody modelowania układów dynamicznych	3
3	Metody symulacji i wizualizacji	5
4	Przykłady układów dynamicznych	6
4.1	Naelektryzowana masa punktowa na sprężynie	6
4.2	Podwójne wahadło	9
4.3	Odwrócone wahadło na wózku	11
4.4	Noga robota	13
5	Opis aplikacji	15
5.1	Schemat tworzenia aplikacji	15
5.2	Naelektryzowana masa punktowa na sprężynie	16
5.3	Podwójne wahadło	16
5.4	Odwrócone wahadło na wózku	18
5.5	Noga robota	18
6	Podsumowanie	23
	Bibilografia	23

Rozdział 1

Wprowadzenie

Dynamika pojedynczego ciała decyduje o sposobie jego ruchu, jako reakcji na działające na nie siły zewnętrzne. Układ dynamiczny zazwyczaj złożony jest z grupy takich obiektów, połączonych ze sobą więzami kinematycznymi. W rezultacie występują w nim dodatkowe oddziaływania dynamiczne pomiędzy poszczególnymi elementami. W celu opisanego zachowania układu dynamicznego wykorzystujemy znane metody wyprowadzania ich modeli, które pozwalają na otrzymanie układu równań różniczkowych, nazywanych równaniami ruchu [1]. Rozwiązanie tych równań dostarcza nam informacji na temat przemieszczeń, prędkości oraz przyspieszeń występujących w badanym układzie. W procesie konstruowania nowoczesnych układów sterowania, poprawne wyprowadzenie modelu dynamiki jest istotnym krokiem przed doбором oraz implementacją algorytmu sterowania dla danego obiektu [5].

Ponieważ modele układów dynamicznych zwykle są wyprowadzane z przyjęciem co najmniej kilku uproszczeń, bardzo ważna jest umiejętność krytycznego spojrzenia na uzyskany wynik i stwierdzenia, w jakim stopniu otrzymany model imituje rzeczywiste zjawisko. W tym celu pomocne okazują się narzędzia pozwalające na przeprowadzanie symulacji oraz wizualizacji wyniku. I tak równania ruchu można zaimplementować jako równania różniczkowe w środowisku takim jak *Matlab* [3] lub *Mathematica* [11]. Można także zapisać je metodą blokową w pakiecie *Matlab/Simulink* [4]. Inny sposób na zbadanie zachowania obiektu to implementacja opisujących go równań w języku *C/C++* z wykorzystaniem dedykowanych bibliotek jak np. biblioteki *ODE* [6]. Ważnym elementem implementacji jest właściwy dobór interfejsu, pozwalający na dokonanie oceny uzyskiwanych wyników. Rezultaty przeprowadzonych symulacji mogą być przedstawione na wykresach funkcji czasu, portretach fazowych lub animacjach, dzięki czemu możliwe jest niemal natychmiastowe zrozumienie działania badanego układu.

Celem projektu jest przygotowanie systemu umożliwiającego swobodną zmianę parametrów przykładowych układów dynamicznych czy warunków początkowych oraz obsługa symulacji ich zachowania. Zbiór badanych obiektów dobrano na podstawie zawartości kursu „Mechanika analityczna” [8, 7], aplikację zaimplementowano natomiast korzystając ze środowiska *Mathematica* [11] oraz powiązanego z nim projektu *Wolfram Demonstrations Project* [10]. Zamysłem autora było, by wykonany system symulacji układów dynamicznych stanowił pomoc dydaktyczną podczas prowadzenia zajęć ze studentami poznającymi metody mechaniki analitycznej.

Układ niniejszej pracy jest następujący. W rozdziale 2 omówimy podstawowe metody modelowania układów dynamicznych, po czym w rozdziale 3 skupimy się na metodach symulacji oraz wizualizacji ich zachowań. Rozdział 4 zawiera przykłady układów dynamicznych, które zostały wykorzystane w opisanym w rozdziale 5 aplikacji. Rozdział 6 stanowi podsumowanie projektu.

Rozdział 2

Metody modelowania układów dynamicznych

W rozważaniach na temat modelowania układów dynamicznych omówimy trzy podejścia, dzięki którym możemy wyprowadzić równania ruchu. Pierwsze z nich, oparte na mechanice Newtona, polega na wykonaniu rachunku sił działających na dany obiekt. Przy opisie układu szczególnie istotna jest w tym podejściu druga zasada dynamiki, najczęściej zapisywana wzorem

$$F = ma,$$

przy czym $a = \frac{d^2x}{dt^2}$, co można zapisać

$$F = m\ddot{x}, \tag{2.1}$$

gdzie F – wypadkowy wektor sił działających na ciało, x – położenie ciała, m – jego masa, a – wypadkowe przyspieszenie, t – czas. Dla każdego obiektu wchodzącego w skład układu dynamicznego należy napisać równanie, którego lewa strona jest wyrażeniem o postaci (2.1), a prawa strona to suma działających na opisywaną masę sił. Rozwiązaniem jest równanie ruchu, przedstawione jako n równań różniczkowych drugiego rzędu, gdzie n – rozmiar wektora współrzędnych układu [2].

Drugi sposób na wyprowadzenie równań dynamiki polega na wykorzystaniu formalizmu Eulera-Lagrange'a [2]. Korzystając z tej metody, należy znaleźć wyrażenia na całkowitą energię kinetyczną układu E_k oraz jego całkowitą energię potencjalną E_p , na podstawie których wyliczany jest lagranżjan L , według formuły

$$L(q, \dot{q}) = E_k(q, \dot{q}) - E_p(q), \tag{2.2}$$

gdzie $q \in \mathbb{R}^n$ jest wektorem współrzędnych uogólnionych układu. Ostatnim etapem jest napisanie równań ruchu w postaci

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = 0. \tag{2.3}$$

Podobnie jak w metodzie newtonowskiej i tutaj otrzymujemy n równań różniczkowych drugiego rzędu dla n współrzędnych układu.

Trzecie podejście, pozwalające uzyskać równania ruchu, opiera się na mechanice Hamiltona. Podstawowym elementem w wyprowadzeniach jest hamiltonian H , wyrażający energię całkowitą układu w kategoriach położenia q i pędu uogólnionego p , jako

$$H(q, p) = E_k(q, p) + E_p(q).$$

Na jego podstawie można zapisać równania kanoniczne Hamiltona

$$\begin{cases} \dot{q} = \frac{\partial H}{\partial p}, \\ \dot{p} = -\frac{\partial H}{\partial q}. \end{cases}$$

W odróżnieniu od wcześniej opisywanych metod, w tym podejściu równania ruchu wyrażone są jako $2n$ równań różniczkowych pierwszego rzędu, dla n współrzędnych układu [2].

Dla układów mechanicznych otrzymywane równania ruchu można zapisać w ogólnej postaci równań dynamiki [9]

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + D(q) = 0, \quad (2.4)$$

gdzie $M(q)$ – macierz sił bezwładności, $C(q, \dot{q})$ – macierz sił odśrodkowych i Coriolisa, $D(q)$ – wektor sił potencjalnych. Zapis równań ruchu układu robotycznego w formie (2.4) jest powszechnie wykorzystywany w algorytmach sterowania.

W opisanych metodach wyprowadzania modeli układów dynamicznych uwzględniamy działanie tylko kilku czynników rzeczywistych: sił bezwładności, odśrodkowych, Coriolisa czy potencjalnych. Pomijamy natomiast wiele innych zjawisk obecnych w rzeczywistych układach. Najistotniejszym z nich jest występowanie tarcia, dla którego podstawowym modelem jest model tarcia wiskotycznego, zakładający że siła (lub moment sił) tarcia jest proporcjonalna do prędkości między ruchomo połączonymi obiektami. Można ją wyrazić jako

$$F_T = \mu\dot{q},$$

gdzie μ to współczynnik tarcia, a \dot{q} to prędkość ruchu. Opisany w ten sposób model tarcia uwzględniamy w równaniu dynamiki (2.4) otrzymując

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + D(q) + F_T(\dot{q}) = 0. \quad (2.5)$$

Można modelować również inne czynniki rzeczywiste, takie jak nieidealną sztywność obiektów czy obecność ograniczeń fazowych. Wówczas otrzymane równania uwzględniałyby przykładowo zmianę kształtu poruszających się elementów, co z kolei może prowadzić chociażby do powstawania rezonansów. Jednak tego typu efekty są trudne do modelowania [1] i wykraczają poza zakres niniejszej pracy.

Rozdział 3

Metody symulacji i wizualizacji

W celu przeprowadzenia symulacji zachowania układu dynamicznego korzystamy z środowisk dostarczających metod implementacji oraz rozwiązywania równań różniczkowych. Przykładami programów, w których równania dynamiki można rozwiązać korzystając z ich matematycznej postaci są *Mathematica* [11] oraz *Matlab* [3]. Możliwe jest również zaimplementowanie równania dynamiki w postaci blokowej korzystając z pakietu *Matlab/Simulink* [4], czy opracowanie własnego oprogramowania przy wykorzystaniu popularnych języków programowania i dostępnych dla nich bibliotek (np. *ODE* [6] w *C/C++*). Warto zauważyć, że rezultat może być różny w zależności od przyjętej metody rozwiązywania równań różniczkowych czy całkowania numerycznego [1].

W zależności od wybranego środowiska pracy, rozwiązania równań różniczkowych mogą być otrzymane w różnej postaci. *Matlab* jako efekt swojego działania podaje wektory rozwiązań oraz czasu, podczas gdy *Mathematica* zwraca funkcje interpolujące rozwiązania. Należy o tym pamiętać podczas ustawiania parametrów symulacji.

Jak wspomniano we wstępie uzyskane wyniki można wizualizować na wiele sposobów. Podstawowy z nich to wykres zależności funkcji od czasu, pozwalający na obserwowanie położenia, prędkości oraz przemieszczeń dynamicznego obiektu. Niestety korzystając z tego sposobu wizualizacji, interpretacja zachowania układu zazwyczaj wymaga wnikliwej analizy wykresów wspomnianych funkcji od czasu.

Innym znanym sposobem wizualizacji rozwiązań symulacyjnych jest portret fazowy. Uzyskuje się go rysując zestaw wykresów zależności pomiędzy współrzędnymi stanu układu. Każdy z nanoszonych wykresów przedstawia zachowanie układu dla wybranych warunków początkowych. Portret fazowy można również interpretować jako poziomice hamiltonianu na trójwymiarowym wykresie zależności energii całkowitej układu (oś *Z*) od dwóch badanych wielkości (osie *X*, *Y*).

Rozwiązania symulacji mogą być także przedstawione w formie animacji. W rezultacie otrzymujemy pewnego rodzaju połączenie wykresów położenia, prędkości oraz przyspieszeń od czasu. Wizualizowanie wyników symulacji poprzez animację pozwala w bardzo intuicyjny sposób przedstawić dynamiczne zachowanie układu.

Rozdział 4

Przykłady układów dynamicznych

Rozwiązywanie zadań związanych z układami dynamicznymi jest istotną częścią kursów wprowadzających studentów w świat robotyki. Poniżej, na podstawie notatek sporządzonych podczas ćwiczeń z Mechaniki analitycznej, opisano kolejne kroki prowadzące do rozwiązania przykładowych zadań [7].

4.1 Naelektryzowana masa punktowa na sprężynie

Zadanie

Naelektryzowana masa punktowa m o ładunku q została zawieszona na liniowej sprężynie o współczynniku sprężystości k nad ładunkiem punktowym Q , przeciwnym do q , umieszczonym w odległości l (zobacz rys 4.1). Wyznaczyć równanie ruchu takiego układu i narysować jego portret fazowy.

Rozwiązanie

Oznaczmy chwilowe położenie naładowanej masy jako x (rys 4.1). Wówczas korzystając z drugiej zasady dynamiki Newtona (2.1) oraz przeprowadzając rachunek sił działających na masę m otrzymujemy

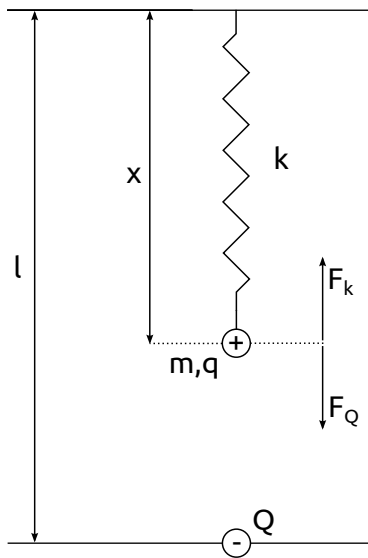
$$m\ddot{x} = F_Q - F_k, \quad (4.1)$$

gdzie $F_Q = \frac{-Qq}{(l-x)^2}$ jest siłą oddziaływania ładunków elektrycznych, zaś $F_k = kx$ siłą działającą na masę pochodzącą od sprężyny. Należy zauważyć, że układ sił (4.1) obowiązuje wyłącznie dla $x < l$, a znak „-” przy F_Q wynika z przeciwnych wartości ładunków. Po podstawieniu do równania (4.1) wyrażen na F_Q i F_k oraz przeniesieniu wyrazów na jedną stronę, otrzymujemy równanie różniczkowe 2 rzędu, o postaci

$$m\ddot{x} + kx + \frac{Qq}{(l-x)^2} = 0. \quad (4.2)$$

Zadanie wyznaczania równań ruchu można również rozwiązać korzystając z formalizmu Eulera-Lagrange’a. W celu znalezienia langranżjanu dla badanego układu, obliczamy wyrażenia na jego całkowitą energię kinetyczną

$$E_k = \frac{m\dot{x}^2}{2},$$



Rysunek 4.1 Rysunek do zadania „Naelektryzowana masa punktowa na sprężynie”

oraz całkowitą energię potencjalną

$$E_p = \frac{kx^2}{2} + \frac{Qq}{l-x},$$

i podstawiamy je do wzoru (2.2) uzyskując

$$L = \frac{m\dot{x}^2}{2} - \frac{kx^2}{2} - \frac{Qq}{l-x}.$$

Następnie obliczamy elementy składowe równania (2.3), kolejno otrzymując

$$\begin{aligned} \frac{\partial L}{\partial \dot{x}} &= m\dot{x}, \\ \frac{d}{dt} \frac{\partial L}{\partial \dot{x}} &= m\ddot{x}, \\ \frac{\partial L}{\partial x} &= -kx - \frac{Qq}{(l-x)^2}. \end{aligned}$$

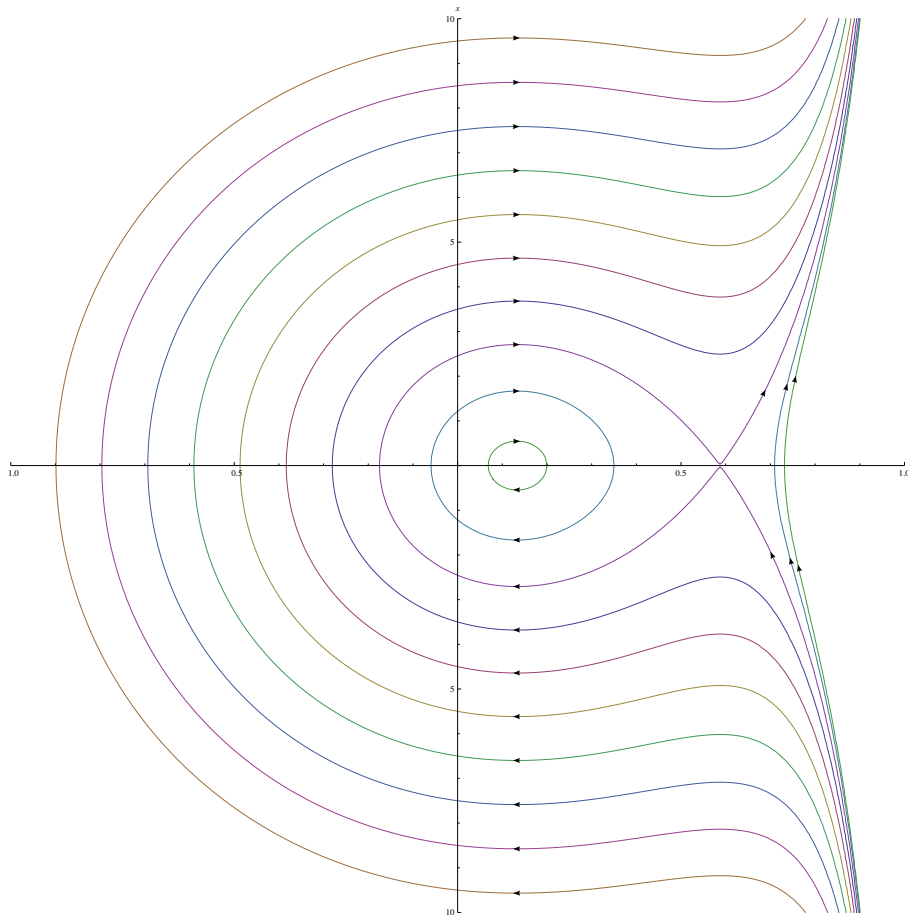
Zapisanie otrzymanych wyrażeń w formie równania (2.3) daje ostatecznie równanie (4.2).

Po wprowadzeniu zmiennej pomocniczej $y = \dot{x}$ możemy zapisać równanie dynamiki wahadła (4.2) w formie układu dwóch równań różniczkowych 1 rzędu, a mianowicie

$$\begin{cases} \dot{x} = y, \\ \dot{y} = -\frac{k}{m}x - \frac{Qq}{m(l-x)^2}, \end{cases}$$

co stanowi model analizowanego układu zapisany w jego przestrzeni stanu z wektorem stanu $(x, y)^T$. Dzieliąc drugie z otrzymanych w ten sposób równań przez pierwsze dostajemy

$$\frac{dy}{dx} = -\frac{kx}{my} - \frac{Qq}{my(l-x)^2}.$$



Rysunek 4.2 Portret fazowy do zadania „Naelektryzowana masa punktowa na sprężynie”

Następnie mnożymy nasze równanie obustronnie przez ydx , aby uzyskać

$$ydy = \left(-\frac{kx}{m} - \frac{Qq}{m(l-x)^2} \right) dx,$$

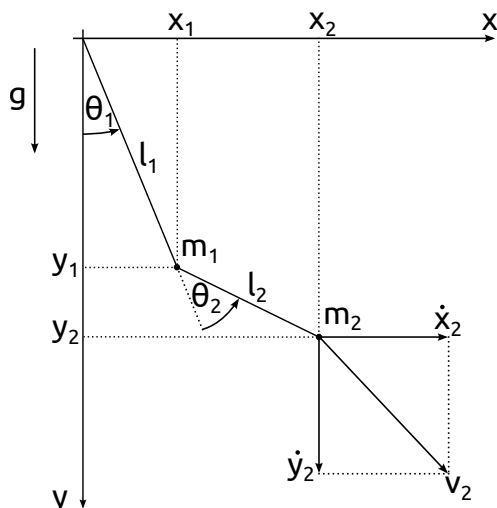
które w kolejnym kroku całkujemy, otrzymując

$$\frac{y^2}{2} = -\frac{kx^2}{2m} - \frac{Qq}{m(l-x)} + \frac{C}{2},$$

gdzie C – stała całkowania o wartości wynikającej z warunków początkowych. Ostatecznie, otrzymujemy wyrażenie o postaci

$$y^2 + \frac{kx^2}{m} + \frac{2Qq}{m(l-x)} = C, \quad (4.3)$$

ponieważ $y = \dot{x}$ jest prędkością ruchu masy punktowej m zaś x jej położeniem, wyznaczona formuła pozwala na wykreślenie rodziny krzywych (dla różnych wartości parametru C) stanowiących razem portret fazowy analizowanego układu, pokazany na rysunku 4.2. Strzałki znajdujące się na narysowanych trajektoriach wskazują kierunek zmian stanu układu.



Rysunek 4.3 Rysunek do zadania „Podwójne wahadło”

4.2 Podwójne wahadło

Zadanie

Dane jest podlegające działaniu siły grawitacji podwójne wahadło, składające się z ramion o długościach l_1 , l_2 oraz punktowych mas m_1 , m_2 umieszczonych na końcach ramion. Korzystając z formalizmu Eulera-Lagrange’a wyznaczyć równania ruchu wahadła.

Rozwiązanie

W celu wyprowadzenia modelu wahadła oznaczymy położenie końca pierwszego ramienia jako (x_1, y_1) , a drugiego (x_2, y_2) , tak jak pokazano na rysunku 4.3. Dla opisywanego układu kąty θ_1 , θ_2 jednoznacznie określające konfigurację układu tworzą wektor jego położenia $q = (\theta_1, \theta_2)^T$. Wówczas położenie mas układu jest opisane jako

$$\begin{aligned}(x_1, y_1) &= (l_1 s_1, l_1 c_1), \\ (x_2, y_2) &= (l_1 s_1 + l_2 s_{12}, l_1 c_1 + l_2 c_{12}),\end{aligned}$$

gdzie dla zachowania przejrzystości przyjęto oznaczenia $s_1 = \sin \theta_1$, $c_1 = \cos \theta_1$, $s_2 = \sin \theta_2$, $c_2 = \cos \theta_2$, $s_{12} = \sin(\theta_1 + \theta_2)$ oraz $c_{12} = \cos(\theta_1 + \theta_2)$. W celu zastosowania metody Eulera-Lagrange’a wyliczamy całkowitą energię kinetyczną i potencjalną układu. Do tego najpierw znajdujemy składowe prędkości liniowej obydwu mas punktowych

$$\begin{aligned}(\dot{x}_1, \dot{y}_1) &= (l_1 c_1 \dot{\theta}_1, -l_1 s_1 \dot{\theta}_1), \\ (\dot{x}_2, \dot{y}_2) &= \left(l_1 c_1 \dot{\theta}_1 + l_2 c_{12} (\dot{\theta}_1 + \dot{\theta}_2), -l_1 s_1 \dot{\theta}_1 - l_2 s_{12} (\dot{\theta}_1 + \dot{\theta}_2) \right).\end{aligned}$$

W kolejnym kroku obliczamy kwadrat prędkości liniowej masy pierwszej

$$v_1^2 = \dot{x}_1^2 + \dot{y}_1^2 = (l_1 c_1 \dot{\theta}_1)^2 + (-l_1 s_1 \dot{\theta}_1)^2 = l_1^2 \dot{\theta}_1^2,$$

oraz drugiej

$$\begin{aligned} v_2^2 = \dot{x}_2^2 + \dot{y}_2^2 &= \left(l_1 c_1 \dot{\theta}_1 + l_2 c_{12} (\dot{\theta}_1 + \dot{\theta}_2) \right)^2 + \left(-l_1 s_1 \dot{\theta}_1 - l_2 s_{12} (\dot{\theta}_1 + \dot{\theta}_2) \right)^2 = \\ &= l_1^2 \dot{\theta}_1^2 + 2l_1 l_2 \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) c_2 + l_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2, \end{aligned}$$

co ostatecznie prowadzi do uzyskania wyrażenia na całkowitą energię kinetyczną układu daną wzorem

$$\begin{aligned} E_k &= \frac{1}{2} (m_1 v_1^2 + m_2 v_2^2) = \frac{1}{2} m_1 l_1^2 \dot{\theta}_1^2 + \frac{1}{2} m_2 \left(l_1^2 \dot{\theta}_1^2 + 2l_1 l_2 \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) c_2 + l_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 \right) = \\ &= \frac{1}{2} (m_1 + m_2) l_1^2 \dot{\theta}_1^2 + m_2 l_1 l_2 c_2 \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) + \frac{1}{2} m_2 l_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2. \end{aligned}$$

Wyliczenie całkowitej energii potencjalnej prowadzi do wyrażenia

$$E_p = m_1 g y_1 + m_2 g y_2 = m_1 g l_1 c_1 + m_2 g (l_1 c_1 + l_2 c_{12}) = (m_1 + m_2) g l_1 c_1 + m_2 g l_2 c_{12}.$$

Dla tak określonych energii lagranżjan (2.2) przyjmuje postać

$$L = \frac{1}{2} (m_1 + m_2) l_1^2 \dot{\theta}_1^2 + m_2 l_1 l_2 c_2 \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) + \frac{1}{2} m_2 l_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 - (m_1 + m_2) g l_1 c_1 - m_2 g l_2 c_{12}.$$

W rozpatrywanym przypadku dla wektora położenia $q = (\theta_1, \theta_2)^T$ równania (2.3) tworzą następujący układ równań

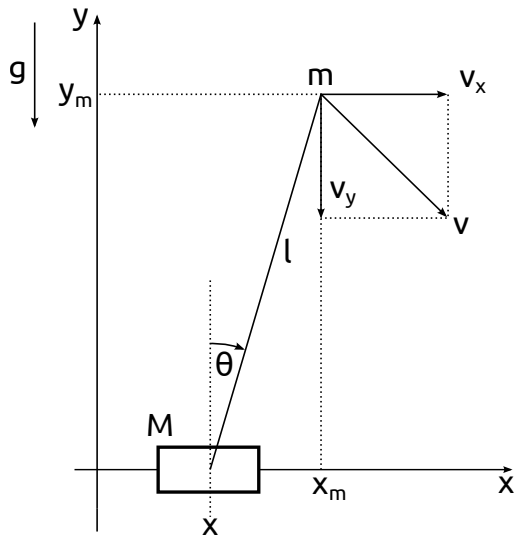
$$\begin{cases} \frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_1} - \frac{\partial L}{\partial \theta_1} = 0, \\ \frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_2} - \frac{\partial L}{\partial \theta_2} = 0. \end{cases} \quad (4.4)$$

Poszczególne elementy składowe równania (4.4) wyrażone są jako

$$\begin{aligned} \frac{\partial L}{\partial \dot{\theta}_1} &= (m_1 + m_2) l_1^2 \dot{\theta}_1 + m_2 l_1 l_2 c_2 (2\dot{\theta}_1 + \dot{\theta}_2) + m_2 l_2^2 (\dot{\theta}_1 + \dot{\theta}_2), \\ \frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_1} &= (m_1 + m_2) l_1^2 \ddot{\theta}_1 - m_2 l_1 l_2 s_2 \dot{\theta}_2 (2\dot{\theta}_1 + \dot{\theta}_2) + m_2 l_1 l_2 c_2 (2\ddot{\theta}_1 + \ddot{\theta}_2) + m_2 l_2^2 (\ddot{\theta}_1 + \ddot{\theta}_2), \\ \frac{\partial L}{\partial \theta_1} &= (m_1 + m_2) g l_1 s_1 + m_2 g l_2 s_{12}, \\ \frac{\partial L}{\partial \dot{\theta}_2} &= m_2 l_1 l_2 c_2 \dot{\theta}_1 + m_2 l_2^2 (\dot{\theta}_1 + \dot{\theta}_2), \\ \frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_2} &= -m_2 l_1 l_2 s_2 \dot{\theta}_2 \dot{\theta}_1 + m_2 l_1 l_2 s_2 \ddot{\theta}_1 + m_2 l_2^2 (\ddot{\theta}_1 + \ddot{\theta}_2), \\ \frac{\partial L}{\partial \theta_2} &= -m_2 l_1 l_2 s_2 \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) + m_2 g l_2 s_{12}. \end{aligned}$$

Ostatecznie, po podstawieniu składowych, równanie (4.4) przyjmuje postać

$$\begin{cases} \left((m_1 + m_2) l_1^2 + 2m_2 l_1 l_2 c_2 + m_2 l_2^2 \right) \ddot{\theta}_1 + \left(m_2 l_1 l_2 c_2 + m_2 l_2^2 \right) \ddot{\theta}_2 + \\ \quad - m_2 l_1 l_2 s_2 (2\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_2^2) - (m_1 + m_2) g l_1 s_1 - m_2 g l_2 s_{12} = 0, \\ \left(m_2 l_1 l_2 s_2 + m_2 l_2^2 \right) \ddot{\theta}_1 + m_2 l_2^2 \ddot{\theta}_2 - m_2 l_1 l_2 s_2 (-\dot{\theta}_1^2) - m_2 g l_2 s_{12} = 0. \end{cases} \quad (4.5)$$



Rysunek 4.4 Ilustracja zadania „Odwrócone wahadło na wózku”

Na podstawie macierzowej postaci równań dynamiki (2.4) oraz układu równań (4.5) można dla analizowanego obiektu wyznaczyć macierze

$$M(q) = \begin{bmatrix} (m_1 + m_2)l_1^2 + 2m_2l_1l_2c_2 + m_2l_2^2 & m_2l_1l_2c_2 + m_2l_2^2 \\ m_2l_1l_2c_2 + m_2l_2^2 & m_2l_2^2 \end{bmatrix},$$

$$C(q, \dot{q}) = -m_2l_1l_2s_2 \begin{bmatrix} 2\dot{\theta}_2 & \dot{\theta}_2 \\ -\dot{\theta}_1 & 0 \end{bmatrix},$$

oraz wektor

$$D(q) = -g \begin{pmatrix} (m_1 + m_2)l_1s_1 + m_2l_2s_{12} \\ m_2l_2s_{12} \end{pmatrix}.$$

4.3 Odwrócone wahadło na wózku

Zadanie

Dane jest podlegające działaniu siły grawitacji odwrócone wahadło o długości l , zakończone masą punktową m , umieszczone na przemieszczającym się w poziomie wózku o masie M . Korzystając z formalizmu Eulera-Lagrange’a wyznaczyć równania ruchu.

Rozwiązanie

Dla układu wahadła można wprowadzić wektor położenia postaci $q = (x, \theta)^T$, gdzie x – pozycja wózka, θ – kąt nachylenia wahadła tak jak pokazano na rysunku 4.4. Oznaczmy położenie końca wahadła jako (x_m, y_m) . Można je wyrazić w funkcji wektora położenia q jako

$$(x_m, y_m) = (x + l \cos \theta, l \sin \theta). \quad (4.6)$$

Analogicznie, wprowadzając współrzędne położenia wózka (x_M, y_M) możemy napisać

$$(x_M, y_M) = (x, 0). \quad (4.7)$$

W celu znalezienia wyrażenia na prędkości mas układu liczymy pochodną położenia chwilowego końca wahadła (4.6)

$$(\dot{x}_m, \dot{y}_m) = (\dot{x} - l \sin \theta \dot{\theta}, l \cos \theta \dot{\theta}),$$

a następnie pochodną położenia chwilowego wózka (4.7)

$$(\dot{x}_M, \dot{y}_M) = (\dot{x}, 0).$$

Teraz możemy napisać wyrażenie na całkowitą energię kinetyczną układu jako

$$E_k = \frac{1}{2}M(\dot{x}_M^2 + \dot{y}_M^2) + \frac{1}{2}m(\dot{x}_m^2 + \dot{y}_m^2) = \frac{1}{2}M\dot{x}^2 + \frac{1}{2}m(\dot{x}^2 - 2\dot{x}l \sin \theta \dot{\theta} + l^2\dot{\theta}^2). \quad (4.8)$$

Wyrażenie na całkowitą energię potencjalną wygląda następująco

$$E_p = mg\eta = mgl \sin \theta.$$

Mając wyliczone E_k i E_p oraz korzystając ze wzoru (2.2) możemy wyznaczyć lagranżjan

$$L = \frac{1}{2}(M + m)\dot{x}^2 + \frac{1}{2}m(l^2\dot{\theta}^2 - 2\dot{x}l \sin \theta \dot{\theta}) - mgl \sin \theta.$$

W kolejnym kroku znajdujemy składowe równania Eulera-Lagrange'a (2.3)

$$\begin{aligned} \frac{\partial L}{\partial \dot{x}} &= (M + m)\dot{x} - ml \sin \theta \dot{\theta}, \\ \frac{d}{dt} \frac{\partial L}{\partial \dot{x}} &= (M + m)\ddot{x} - ml \cos \theta \dot{\theta}^2 - ml \sin \theta \ddot{\theta}, \\ \frac{\partial L}{\partial x} &= 0, \\ \frac{\partial L}{\partial \dot{\theta}} &= ml^2\dot{\theta} - m\dot{x}l \sin \theta, \\ \frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}} &= ml^2\ddot{\theta} - m\ddot{x}l \sin \theta - m\dot{x}l \cos \theta \dot{\theta}, \\ \frac{\partial L}{\partial \theta} &= -m\dot{x}l \cos \theta \dot{\theta} - mgl \cos \theta. \end{aligned}$$

Ostatecznie zapisujemy równanie ruchu w postaci

$$\begin{cases} (M + m)\ddot{x} - ml \sin \theta \ddot{\theta} - ml \cos \theta \dot{\theta}^2 = 0, \\ -ml \sin \theta \ddot{x} + ml^2\ddot{\theta} + mgl \cos \theta = 0. \end{cases} \quad (4.9)$$

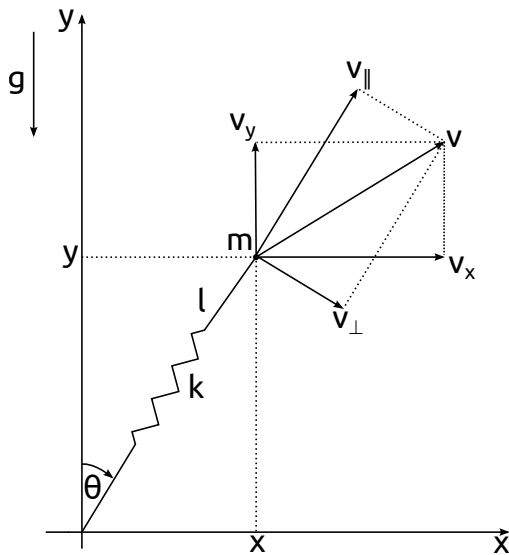
Na podstawie otrzymanych równań (4.9), znając macierzową postać równań dynamiki (2.4), można dla analizowanego układu wyliczyć macierze

$$M(q) = \begin{bmatrix} M + m & -ml \sin \theta \\ -ml \sin \theta & ml^2 \end{bmatrix},$$

$$C(q, \dot{q}) = \begin{bmatrix} 0 & -ml \cos \theta \dot{\theta} \\ 0 & 0 \end{bmatrix},$$

oraz wektor

$$D(q) = \begin{pmatrix} 0 \\ mgl \cos \theta \end{pmatrix}.$$



Rysunek 4.5 Ilustracja zadania „Noga robota”

4.4 Noga robota

Zadanie

Dana jest noga robota o długości nominalnej l_0 , współczynnika sprężystości k oraz masie m , w fazie kontaktu z podłożem. Korzystając z formalizmu Eulera-Lagrange'a wyznaczyć równania jej ruchu.

Rozwiązanie

Przyjęto wektor położenia $q = (l, \theta)$, gdzie l – długość nogi, a θ – kąt nachylenia w stosunku do podłoża. Współrzędne końca nogi oznaczono (x, y) , jak pokazano na rysunku 4.5. Możemy napisać następujące wyrażenie na położenie końca nogi

$$(x, y) = (l \cos \theta, l \sin \theta),$$

następnie licząc jego pochodną otrzymujemy

$$(\dot{x}, \dot{y}) = (\dot{l} \cos \theta - l \sin \theta \dot{\theta}, \dot{l} \sin \theta + l \cos \theta \dot{\theta}),$$

co po podniesieniu do kwadratu i zsumowaniu składowych daje kwadrat prędkości liniowej

$$v^2 = \dot{l}^2 + l^2 \dot{\theta}^2.$$

Natychmiastowo otrzymujemy wyrażenie na całkowitą energię kinetyczną układu o postaci

$$E_k = \frac{1}{2} m (\dot{l}^2 + l^2 \dot{\theta}^2).$$

Całkowita energia potencjalna jest równa energii sumie energii potencjalnych grawitacji oraz sprężystości, co zapisujemy

$$E_p = \frac{1}{2} k (l - l_0)^2 + mgl \sin \theta,$$

gdzie l_0 – nominalna długość nogi. Korzystając ze wzoru (2.2) wyznaczamy lagranżjan

$$L = \frac{1}{2}m(\dot{l}^2 + l^2\dot{\theta}^2) - \frac{1}{2}k(l - l_0)^2 - mgl \sin \theta,$$

po czym obliczamy składowe równania (2.3)

$$\begin{aligned} \frac{\partial L}{\partial \dot{l}} &= m\dot{l}, \\ \frac{d}{dt} \frac{\partial L}{\partial \dot{l}} &= m\ddot{l}, \\ \frac{\partial L}{\partial l} &= m\dot{\theta}^2 - k(l - l_0) - mg \sin \theta, \\ \frac{\partial L}{\partial \dot{\theta}} &= ml^2\dot{\theta}, \\ \frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}} &= ml^2\ddot{\theta} + 2ml\dot{\theta}, \\ \frac{\partial L}{\partial \theta} &= -mgl \cos \theta. \end{aligned}$$

Ostatecznie równanie (2.3) przyjmuje postać

$$\begin{cases} m\ddot{l} - m\dot{\theta}^2 - k(l - l_0) + mg \sin \theta = 0, \\ ml^2\ddot{\theta} + 2ml\dot{\theta} + mgl \cos \theta = 0, \end{cases} \quad (4.10)$$

z którego możemy dla analizowanego obiektu wyznaczyć macierze

$$M(q) = \begin{bmatrix} m & 0 \\ 0 & ml^2 \end{bmatrix},$$

$$C(q, \dot{q}) = \begin{bmatrix} 0 & -ml\dot{\theta} \\ 2ml\dot{\theta} & 0 \end{bmatrix},$$

oraz wektor

$$D(q) = \begin{pmatrix} -k(l - l_0) + mg \sin \theta \\ mgl \cos \theta \end{pmatrix}.$$

Rozdział 5

Opis aplikacji

System symulacji zachowania układów dynamicznych przygotowano w środowisku *Mathematica* [11]. Ponieważ przy implementacji każdego z interaktywnych interfejsów wykonywano wiele analogicznych operacji, poniżej przedstawiony zostanie najpierw ogólny schemat tworzenia tego typu aplikacji, a dopiero w dalszej części zaprezentowane zostaną poszczególne rozwiązania. Podstawowa trudność przy tworzeniu takich interfejsów polega na znalezieniu kompromisu pomiędzy funkcjonalnością, a przejrzystością aplikacji. Dając użytkownikowi zbyt wiele parametrów do kontroli sprawiamy, że może poczuć się on zdezorientowany. Równie niekorzystne odczucia wywołuje ograniczanie jego możliwości oddziaływania z programem.

5.1 Schemat tworzenia aplikacji

Przed przystąpieniem do implementacji interfejsu należy przygotować wyrażenia opisujące wektor zmiennych stanu, kinematykę wszystkich elementów obiektu oraz macierze równania dynamiki (2.4). Następnie należy określić, które spośród parametrów układu będą kontrolowane przez użytkownika oraz jaka metoda wizualizacji zostanie użyta.

Implementację zaczynamy od napisania funkcji inicjalizującej. Jej zadaniem jest wprowadzenie wszystkich zmiennych (w znaczeniu programistycznym) odpowiadających parametrom układu, stałym fizycznym oraz wektorowi współrzędnych. Następnie stworzymy funkcję pozwalającą na rozwiązanie równań dynamiki z uwzględnieniem zadanych warunków początkowych i wybranych przez użytkownika parametrów układu.

Otrzymane rozwiązanie jest gotowe do przedstawienia na wykresie w zależności od czasu. Inne metody wizualizacji wyników wymagają dodatkowych operacji. Jeżeli wykonujemy animację, to potrzebna jest lista grafik otrzymana przez wielokrotne podstawienie (w zadanych odstępach czasu) rozwiązania równania dynamiki do macierzy kinematyki układu. Natomiast jeżeli chcemy uzyskać portret fazowy, to należy wykorzystać funkcję do rysowania wykresów funkcji uwikłanych, w której argumentem podajemy równania opisujące poziomice hamiltonianu w zależności od interesujących nas wielkości.

W kolejnym kroku przystępujemy do implementacji interaktywnej części aplikacji. W tym celu wykorzystujemy funkcję *Manipulate*. Jej składnia pozwala na podanie obiektu i metody wizualizacji oraz dodanie suwaków i przycisków wraz z opisami, dzięki którym użytkownik może przygotować układ dynamiczny do wizualizacji oraz kontrolować proces symulacji. Napisane funkcje przenosimy następnie do specjalnie przygotowanego notatnika programu *Mathematica* o nazwie *demonstration*, po czym wysyłamy go na stronę projektu „Wolfram Demonstrations Project” [10], która zamienia go na gotowy interfejs cdf.

5.2 Naelektryzowana masa punktowa na sprężynie

Głównym założeniem interfejsu do zadania „Naelektryzowana masa punktowa na sprężynie” jest jednoczesna wizualizacja położenia punkowej masy na animacji oraz odpowiadających mu stanów na portrecie fazowym. Wykorzystane elementy interakcji z użytkownikiem to:

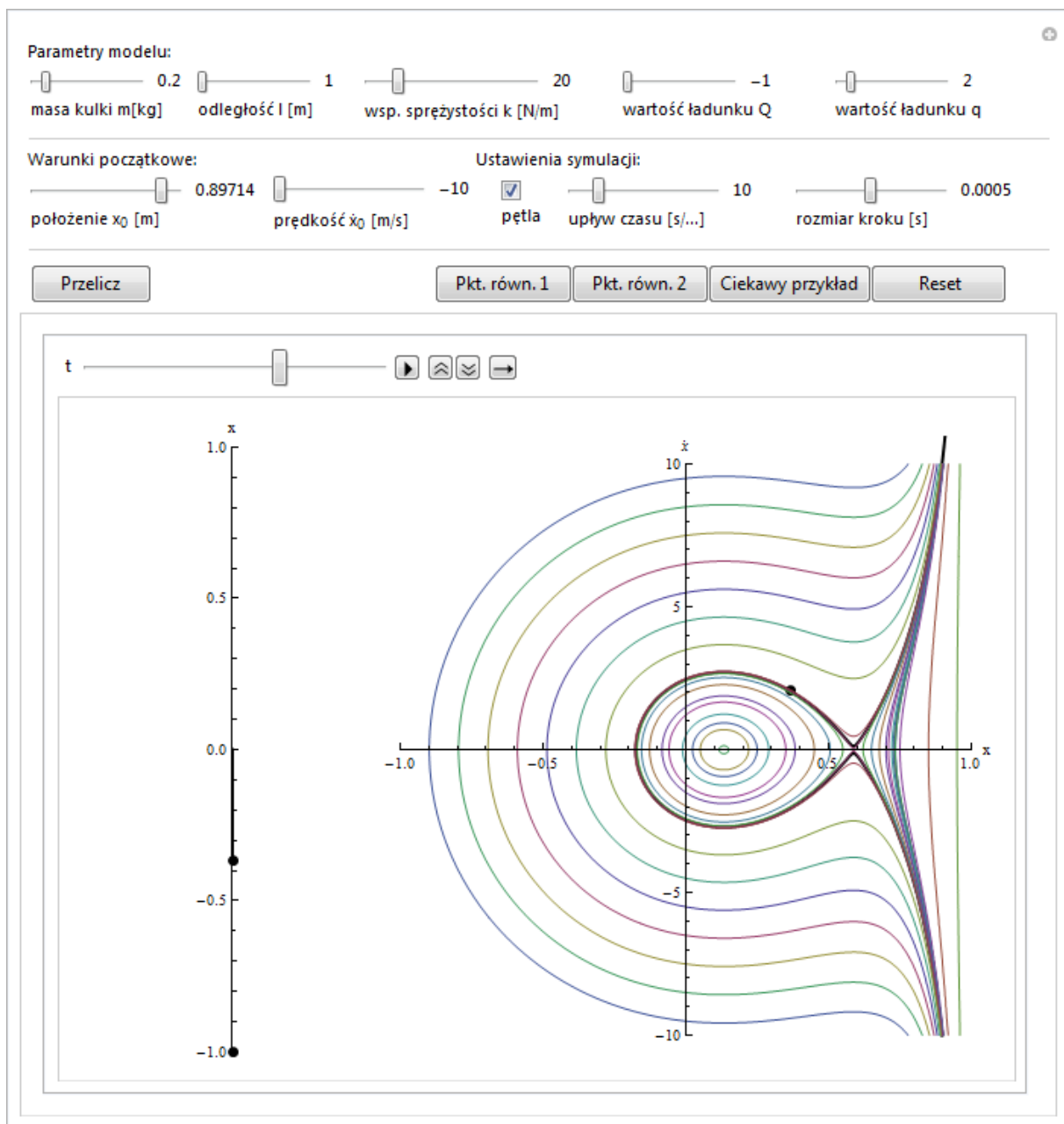
- suwaki parametrów modelu: masa m , odległość l , współczynnik sprężystości k , wartość ładunku nieruchomego Q , wartość ładunku masy punktowej q ,
- suwaki warunków początkowych: położenie początkowe masy x_0 , prędkość początkowa masy \dot{x}_0 ,
- kontrola symulacji: znacznik odtwarzania symulacji w pętli, krotność zwolnienia czasu symulacji w stosunku do czasu rzeczywistego, różnica czasu między kolejnymi klatkami animacji wyrażona w sekundach,
- przyciski: ponowne wykonanie symulacji z uwzględnieniem wartości ustawionych suwakami (przelicz), przykładowa niestabilna konfiguracja początkowa układu, dla której masa punktowa dwukrotnie niemal całkowicie zatrzymuje się (ciekawo), sytuacja, w której kulka znajduje się w pierwszym punkcie równowagi (pkt. równ. 1), sytuacja, w której kulka znajduje się w drugim punkcie równowagi (pkt. równ. 2), przywrócenie konfiguracji startowej (reset).

Przyjęta konfiguracja startowa to następujące ustawienie parametrów w chwili uruchomienia aplikacji: $m = 0.2$, $l = 1$, $k = 20$, $Q = -1$, $q = 2$ oraz warunków początkowych: $x_0 = 0.5$, $\dot{x}_0 = 0$. Na rysunku 5.1 przedstawiono wygląd zaimplementowanego interfejsu.

5.3 Podwójne wahadło

Celem implementacji interfejsu do zadania „Podwójne wahadło” było pokazanie zachowania dynamicznego obiektu za pomocą animacji oraz umożliwienie użytkownikowi wpływania na następujące elementy symulacji:

- suwaki parametrów modelu: długość pierwszego ramienia l_1 , masa pierwszego ramienia m_1 , długość drugiego ramienia l_2 , masa drugiego ramienia m_2 , współczynnik tarcia μ ,
- suwaki dynamiki obiektu: wzmocnienie sił bezwładności (poza diagonalą macierzy $M(q)$), wzmocnienie sił odśrodkowych i Coriolisa (macierz $C(q, \dot{q})$), wzmocnienie sił potencjalnych (wektor $D(q)$),
- kontrola symulacji: znacznik odtwarzania symulacji w pętli, czas trwania animacji, różnica czasu między kolejnymi klatkami animacji wyrażona w sekundach, liczba wyświetlanych poprzednich klatek animacji,
- przyciski: ponowne wykonanie symulacji z uwzględnieniem wartości ustawionych suwakami (przelicz), sytuacja, w której obiekt jest rozprzęgnięty, tzn. poszczególne ramiona zachowują się jak dwa dynamicznie niezależne pojedyncze wahadła ($M = I.M$), sytuacja, w której na zachowanie obiektu wpływają siły tarcia (tarcie), sytuacja, w której wykorzystano parametr pozwalający na obserwację śladu ruchu obiektu (ślad), przywrócenie konfiguracji startowej (reset).



Rysunek 5.1 Wygląd interfejsu do zadania „Naelektryzowana masa punktowa na sprężynie”

Przyjęta konfiguracja startowa to następujące ustawienie parametrów w chwili uruchomienia aplikacji: $l_1 = 1$, $l_2 = 1$, $m_1 = 1$, $m_2 = 1$, $\mu = 0$ oraz warunków początkowych: $\theta_{10} = \pi$, $\dot{\theta}_{10} = 0$, $\theta_{20} = 0$, $\dot{\theta}_{20} = -0.01$. Na rysunku 5.2 przedstawiono wygląd interfejsu w trakcie animacji.

5.4 Odwrócone wahadło na wózku

W kolejnym interfejsie wizualizowano za pomocą animacji „Odwrócone wahadło na wózku”. Elementy interakcji udostępnione użytkownikowi to:

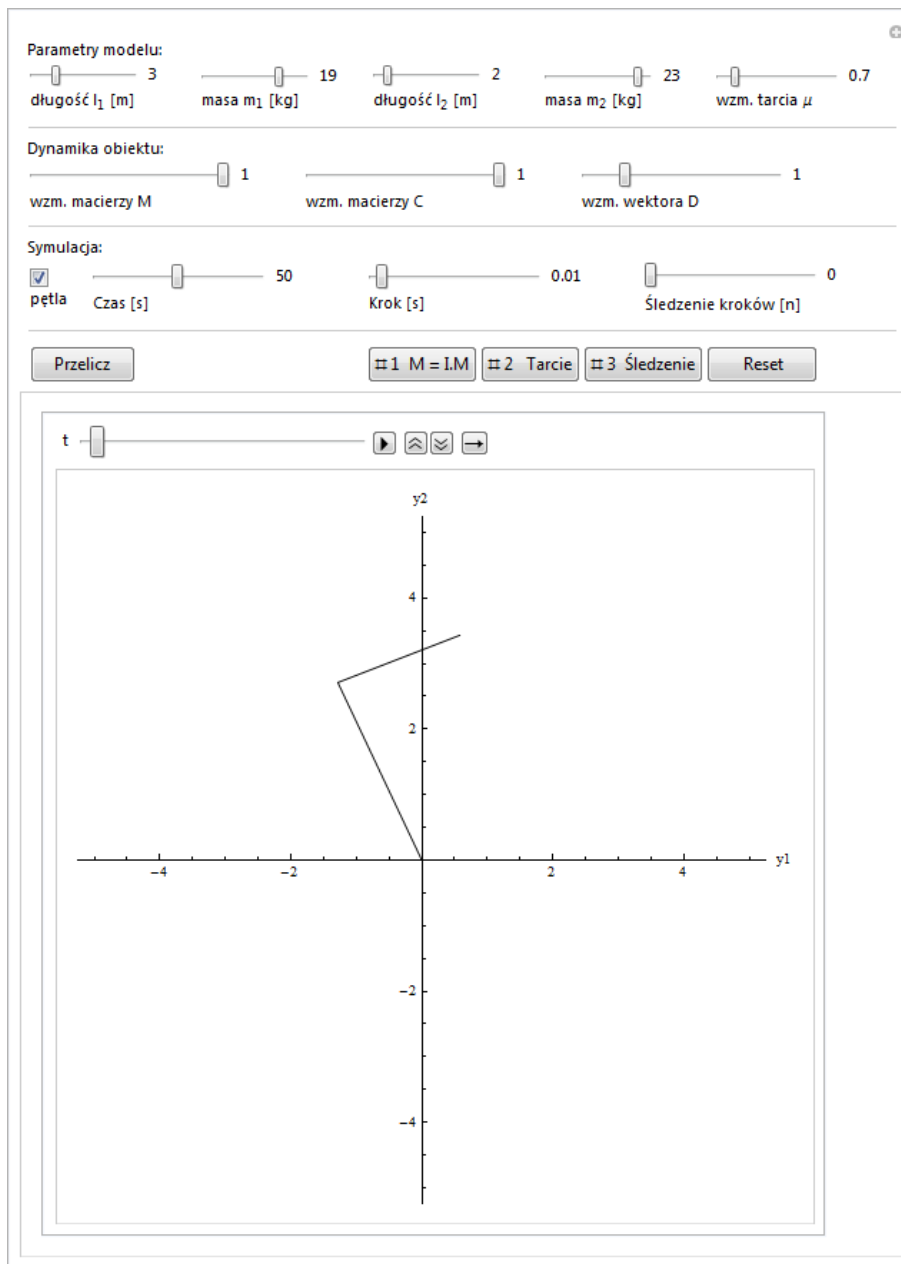
- suwaki parametrów modelu: długość wahadła l , masa wahadła m , masa wózka M , współczynnik tarcia μ ,
- suwaki dynamiki obiektu: wzmocnienie sił bezwładności (poza diagonalą macierzy $M(q)$), wzmocnienie sił odśrodkowych i Coriolisa (macierz $C(q, \dot{q})$), wzmocnienie sił potencjalnych (wektor $D(q)$),
- kontrola symulacji: znacznik odtwarzania symulacji w pętli, czas trwania animacji, różnica czasu między kolejnymi klatkami animacji wyrażona w sekundach, liczba wyświetlanych poprzednich klatek animacji, (x'_0) prędkość początkowa wózka,
- przyciski: ponowne wykonanie symulacji z uwzględnieniem wartości ustawionych suwakami (przelicz), sytuacja, w której obiekt jest rozprzęgnięty, tzn. wózek i wahadło zachowują się jak dwa dynamicznie niezależne ciała ($M = I.M$), sytuacja, w której na zachowanie obiektu wpływają siły tarcia (tarcie), sytuacja, w której wykorzystano parametr pozwalający na obserwację śladu ruchu obiektu (ślad), przywrócenie konfiguracji startowej (reset).

Przyjęta konfiguracja startowa to następujące ustawienie parametrów w chwili uruchomienia aplikacji: $m = 1$, $M = 10$, $l = 0.5$, $\mu = 0$ oraz warunków początkowych $x_0 = 0$, $\dot{x}_0 = 0.1$, $\theta_0 = 0$, $\dot{\theta}_0 = -0.1$. Na rysunku 5.3 pokazano zaimplementowany interfejs.

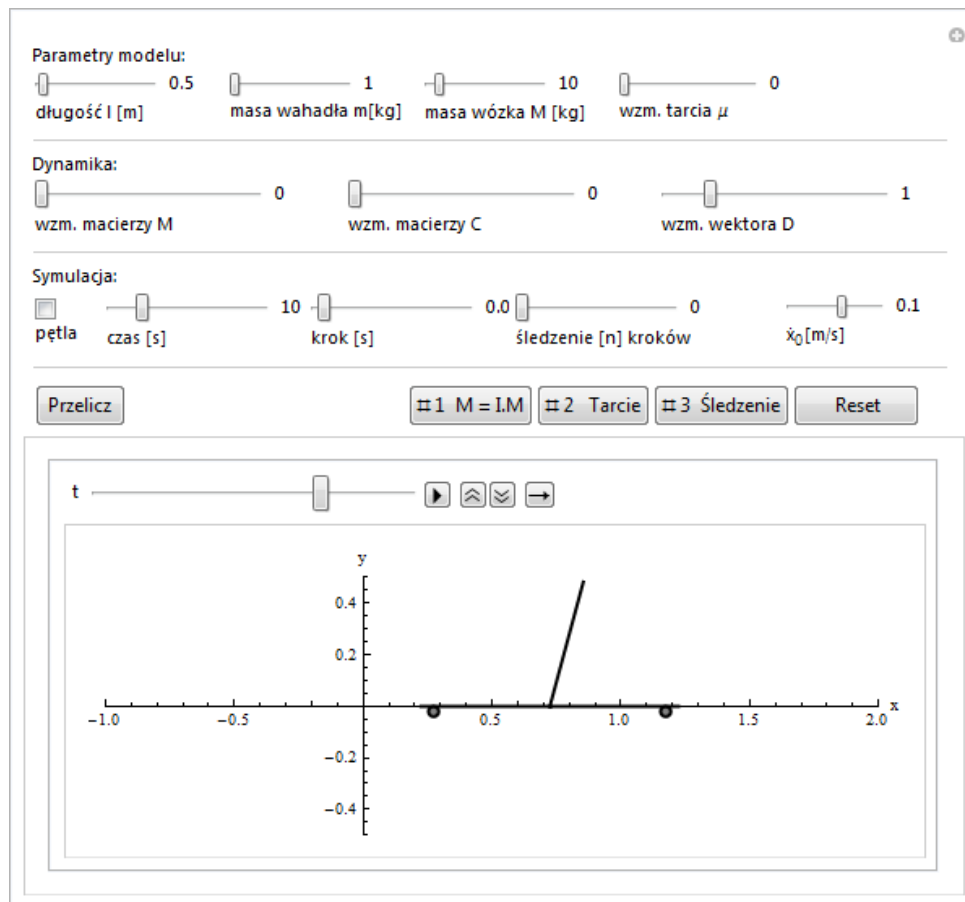
5.5 Noga robota

W ramach implementacji interfejsu przygotowano wizualizację do zadania „Noga robota” korzystając z funkcji animacji. Przygotowano następujące elementy interakcji z użytkownikiem:

- suwaki parametrów modelu: długość nogi l , masa nogi m , współczynnik sprężystości k , współczynnik tarcia μ ,
- suwaki dynamiki obiektu: wzmocnienie sił odśrodkowych i Coriolisa (macierz $C(q, \dot{q})$), wzmocnienie sił potencjalnych (wektor $D(q)$),
- suwaki warunków początkowych: nachylenie początkowe nogi robota θ_0 , wektor prędkości liniowej stawianej nogi v_0 ,
- kontrola symulacji: znacznik odtwarzania symulacji w pętli, ilość klatek animacji składających się na całość wizualizacji ruchu nogi robota, ilość wyświetlanych poprzednich klatek animacji,



Rysunek 5.2 Wygląd interfejsu do zadania „Podwójne wahadło”



Rysunek 5.3 Wygląd interfejsu do zadania „Odwrócone wahadło na wózku”

- przyciski: ponowne wykonanie symulacji z uwzględnieniem wartości ustawionych suwakami (przelicz), sytuacja, w której robot stawia nogę stojąc (stojąc), sytuacja, w której obserwujemy stawianie nogi idącego robota (idąc), konfiguracja przedstawiająca sytuację, w której robot stawia nogę w biegu (biegnąc).

W zaimplementowanej aplikacji warunkiem początkowym przy rozwiązywaniu równań ruchu są pary (l_0, θ_0) oraz $(\dot{l}_0, \dot{\theta}_0)$. Jednak, o ile parametry l_0 oraz θ_0 mają oczywistą interpretację w kontekście nogi robota, o tyle wielkości \dot{l}_0 i $\dot{\theta}_0$ nie są już tak intuicyjne. W związku z powyższym zaproponowano skorzystanie z pomocniczej prędkości początkowej v_0 , którą interpretujemy jako wektor prędkości liniowej masy m . Wzorując się na rysunku 4.5 można napisać

$$v_0 = v_{0\perp} + v_{0\parallel},$$

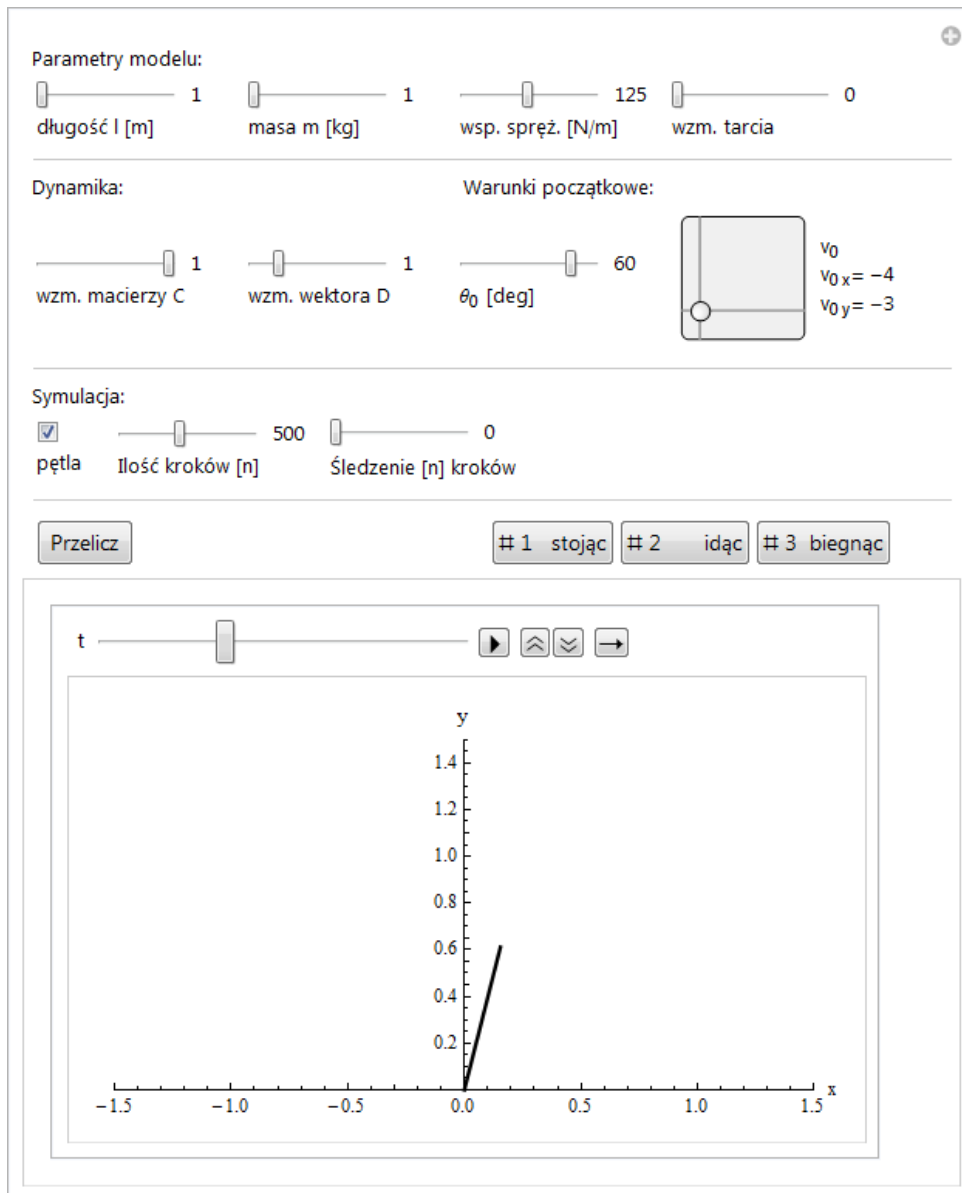
przy czym składowe prostopadła i równoległa prędkości mogą być opisane jako

$$\begin{cases} v_{0\perp} = |v_0| \sin(\arctan v_0 - \theta_0), \\ v_{0\parallel} = |v_0| \cos(\arctan v_0 - \theta_0). \end{cases}$$

Ostatecznie można zapisać

$$\begin{cases} \dot{l}_0 = v_{0\parallel} = |v_0| \cos(\arctan v_0 - \theta_0), \\ \dot{\theta}_0 = \frac{v_{0\perp}}{l_0} = \frac{|v_0| \sin(\arctan v_0 - \theta_0)}{l_0}. \end{cases}$$

Na rysunku 5.4 przedstawiono interfejs będący rezultatem procesu implementacji.



Rysunek 5.4 Wygląd interfejsu do zadania „Noga robota”

Rozdział 6

Podsumowanie

Podjęty projekt polegał na opracowaniu systemu symulacji zachowania przykładowych układów dynamicznych. W celu zrealizowania postawionego zadania opisano metody modelowania takich układów. W dalszej części przedstawiono metody symulacji i wizualizacji zachowań omawianych układów dynamicznych. Następnie zapisano rozwiązania zadań, które w dalszej części zaimplementowano w środowisku *Mathematica* [11]. Ostatecznie opisano wypracowany schemat tworzenia interfejsów i przedstawiono elementy utworzonej aplikacji, tym samym realizując w pełni postawione zadanie. Podczas prac szczególnie pomocne okazały się pozycje [1, 2, 9] oraz notatki z kursu „Mechanika analityczna” [7].

Dalsze rozwijanie aplikacji mogłoby polegać na utworzeniu bardziej złożonych interfejsów lub przygotowaniu ogólnych rozwiązań dla tych już wykonanych. Dla przykładu zamiast symulować zachowanie podwójnego wahadła można przygotować rozwiązanie dla dowolnej liczby ramion dobieranej przez użytkownika. W dalszych pracach nad utworzoną aplikacją należałoby przeprowadzić optymalizację interfejsu dla zadania „Naelektryzowana masa punktowa na sprężynie”. Aktualne rozwiązanie zakłada stałą wartość parametru *PlotPoints* w funkcji rysującej portret fazowy. Jest to spowodowane niepoprawnym rysowaniem dla domyślnej wartości parametru *PlotPoints*. Wadą zastosowanego rozwiązania jest znacznie większy czas obliczeń oraz niezapewnienie poprawnego działania w każdym przypadku.

W celu ostatecznego rozstrzygnięcia konfliktu między funkcjonalnością a przejrzystością interfejsu należałoby umożliwić rozwijanie i chowanie suwaków kontroli parametrów lub utworzyć osobną, uproszczoną wersję aplikacji. Niezależnie od obranej drogi, należy zwrócić szczególną uwagę na intuicyjność interfejsu.

Oczywistym rozwinięciem projektu jest utworzenie systemu symulacji sterowania układami dynamicznymi. W ramach takiego zadania należałoby zaimplementować odpowiednie algorytmy sterowania w oparciu o wyprowadzone modele dynamiki. Wskazane byłoby porzucenie idei kontrolowania wszystkich własności obiektu na rzecz funkcjonalności związanych z doбором różnych trajektorii oraz parametrów sterowania.

Bibliografia

- [1] J. Craig, J. Knapczyk. *Wprowadzenie do robotyki: mechanika i sterowanie*. Wydawn. Naukowo-Techniczne, 1995.
- [2] H. Goldstein, J. Charles P Poole, S. John L Safko. *Classical Mechanics*. Addison Wesley, 2002.
- [3] Mathworks. Matlab. <http://www.mathworks.com/products/matlab/>.
- [4] Mathworks. Matlab/Simulink. <http://www.mathworks.com/products/simulink/>.
- [5] R. Murray, Z. Li, R. Sastry. *Robotic Manipulation: A Mathematical Introduction to*. CRC PressINC, 1994.
- [6] R. Smith. Open dynamics engine. <http://www.ode.org/>.
- [7] K. Tchoń. Notatki z kursu mechanika analityczna. Zapisane przez autora pracy na ćwiczeniach oraz wykładach prowadzonych przez Profesora Krzysztofa Tchońa.
- [8] K. Tchoń. Strona kursu mechanika analityczna. <http://zapisy.pwr.wroc.pl/dydaktyka/katalog/index.php?kodkursu=AREK002>.
- [9] K. Tchoń, A. Mazur, I. Dulęba, R. Hossa, R. Muszyński. *Manipulatory i roboty mobilne: modele, planowanie ruchu, sterowanie*. Problemy Współczesnej Nauki, Teoria i Zastosowania. Robotyka. Akademicka Oficyna Wydawnicza PLJ, 2000.
- [10] Wolfram Research. Wolfram Demonstrations Project. <http://demonstrations.wolfram.com/>.
- [11] Wolfram Research. Wolfram Mathematica 8. <http://www.wolfram.com/mathematica/>.