Chapter

Rolling Biped Polynomial Motion Planning

Santiago de J. Favela Ortíz and Edgar A. Martínez García

Abstract

This work discloses a kinematic control model to describe the geometry of motion of a two-wheeled biped's limbs. Limb structure is based on a four-bar linkage useful to alleviate damping motion during self-balance. The robot self-balancing kinematics geometry combines with user-customized polynomial vector fields. The vector fields generate safe reference trajectories. Further, the robot is forced to track the reference path by a model-based time-variant recursive controller. The proposed formulation showed effectiveness and reliable performance through numerical simulations.

Keywords: rolling-biped, path planning, motion control, navigation, underactuation

1. Introduction

Motion planning is an essential function and a critical aspect in robotics engineering. Motion planning allows increasing the robot's degree of autonomy. Fundamentally a wide area of robotic tasks needs planning models such as: transportation and vehicular technology, service robotics, search, exploration, surveillance, biomedical robotic applications, spatial deployment, industry, and so forth. Moreover, motion planning is inherently impacted by the degree of holonomy and kinematic constraints in all robotic modalities: robot arms, legged robots, rolling platforms, marine/underwater vehicles, aerial robots, and including their end effectors. Depending on the robotic application, motion planning is designed either global or local. When the robot has an environmental map in advance, it is called global planning even with possibility to globally optimize routes. Alternatively, when there is only robot's local sensor data and the whole environment is unknown, it uses feedback from local observations. Planning methods can be generalized into four types: deterministic (based on mathematical numeric/analytic functions and models) [1–3], stochastic (recursive numerical methods based on probabilistic uncertainties) [4–6], heuristic (algorithms based on logical control and human-heuristic decision-making) [7–10], and mixed planning methods [11–13].

In this chapter, a kinematic motion/path planning method for path tracking of an inverted pendulum self-balancing rolling biped is deduced and discussed. This work is focused on the rolling biped motion modeling and simulation of the robot shown in **Figure 1**. The principal component of a rolling biped is self-balancing by controlling its pitch motion through in-wheel motors that allow rolling motion (inverted-pendulum-like). The robot's yaw motion is accomplished by the angular

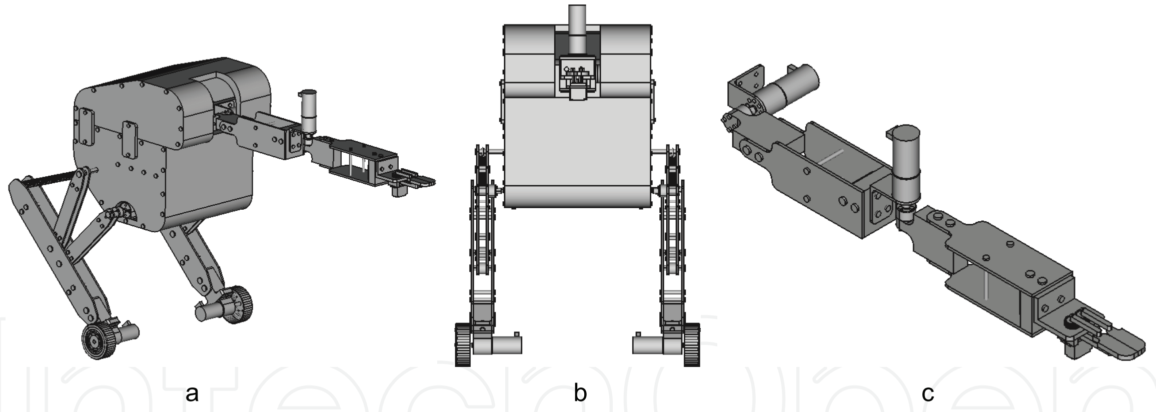

Figure 1.
Rolling biped platform views. (a) Isometric view. (b) Front-view. (c) Onboard robotic arm.

velocity resulting from the differential lateral speeds, which is a nonholonomic constrained model. The robot's design is purposed for missions to collect solid garbage in outdoors (a park), similar to other works [14]. Nevertheless, the robotic mission/task is out of this chapter's scope, instead a detailed geometry of motion is described in three parts: (i) motion planning for biped's balance, (ii) navigational path generation, and (iii) path tracking control. The work [15] proposed a balancing and dynamic speed control of a unicycle robot based on variable structure and linear quadratic regulator to follow a desired trajectory. The work [16] modeled a wheeled bipedal robot with analytic solutions of closed-form expressions in kinematic control loops. The work [17] reported a self-balancing two-wheeled robot with a manipulator on-board, using auto-balancing system to maintain force equilibrium. The work [18] applied a proportional integral derivative (PID) control and active disturbance rejection control to balance and steer a two-wheeled self-balancing robot modeled by Lagrange formula. In [19], an adaptive robust control of a self-balancing two-wheeled underactuated robot to estimate uncertainty bound information, using deterministic system performance by Lyapunov method, was reported. In [20], a navigational two-wheeled self-balancing robot control using a PD-PI controller based on the Kalman filter algorithm was reported. Similarly, [21] used variable structure combining proportional integral differential controllers for balance and locomotion deriving a kinematic model based on the center of gravity constraint. Lagrangian-based with Kane's approach for dynamic balancing was reported in [22]. Moreover, [23] conducted a study using model predictive control for trajectory tracking of an inverted-pendulum wheeled robot. The work [24] reported a self-balancing robot controller using Euler-Lagrange and geometric control, and planar motion is controlled by logarithmic feedback and Lie group exponential coordinates. The work [25] developed a balancing and trajectory tracking system for an inverted-pendulum wheeled robot using a Lagrange-based backstepping structure variable method. This work's main contributions are an original design of limbs based on four-bar linkages to alleviate damping motion yielded from irregular terrains, from which a kinematic balancing condition is deduced. Further, polynomial vector fields with limit conditions are deduced from user-customized interpolation functions as path-generator models to yield safe routes in advance. Moreover, a recursive time-varying kinematic controller forces the robot to track resulting routes. The proposed system is demonstrated at the level of simulation. This chapter is organized in the following sections. Section 2 deduces the limb kinematics and its effects in the biped's balance. Section 3 presents the polynomial approach to trajectory generation by directional fields. Section 4 describes a navigation recursive controller for path tracking control. Finally, Section 5 presents the work's conclusions.

2. Balancing motion planning model

The main issue of an inverted-pendulum-like rolling biped is its capability to self-balance by controlling its pitch angle through the wheels velocity longitudinally (**Figure 2(b)**). This section deduces the balancing kinematics of the limb's planar linkage shown in **Figure 2(a)**. As a difference from other biomechanical inspired muscle-tendon limbs [26], in this work each limb is comprised of a four-bar linkage operating as a double crank, where bars r and d are linked by a coupling link l with limited rotary angles.

The expressions provided in Proposition 2.1 are obtained from deductions in Appendix A. The passive joint point $\mathbf{P}_1 = (x_1, y_1)^\top$ mutually depends on the analytic model of the joint located at $\mathbf{P}_2 = (x_2, y_2)^\top$ to describe a rotary planar motion, proposed by

Proposition 2.1 Limbs motion estimation. *The four-bar linkage's passive joint* $\mathbf{P}_2$ *is analytically described by.*

$$\mathbf{P}_2 = d\begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix} + \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}. \tag{1}$$

Therefore, based on Definitions 4.1–4.3 of Appendix A, the model solution $\mathbf{P}_2$ *is*

$$\begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = d\begin{pmatrix} \cos\left(2\arctan\left(\dfrac{-Q \pm \sqrt{Q^2 - 4PR}}{2P}\right)\right) \\ \sin\left(2\arctan\left(\dfrac{-Q \pm \sqrt{Q^2 - 4PR}}{2P}\right)\right) \end{pmatrix} + \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \tag{2}$$

From Eq. (56) deduced in Appendix A, there are two possible solutions for θ, described as opened or crossed motion. For this work, the type of motion should be $\sqrt{Q^2 - 4PR}\forall Q^2 \leq 4PR$ that must be satisfied. Therefore, by using expressions (47), (52)–(56), the plots shown in 3 are obtained.

Furthermore, the wheels position is tracked by $\mathbf{P}_f$ (**Figure 2(a)**). This design assumes the robot's center of gravity (cog) located at the same length $f \equiv a$

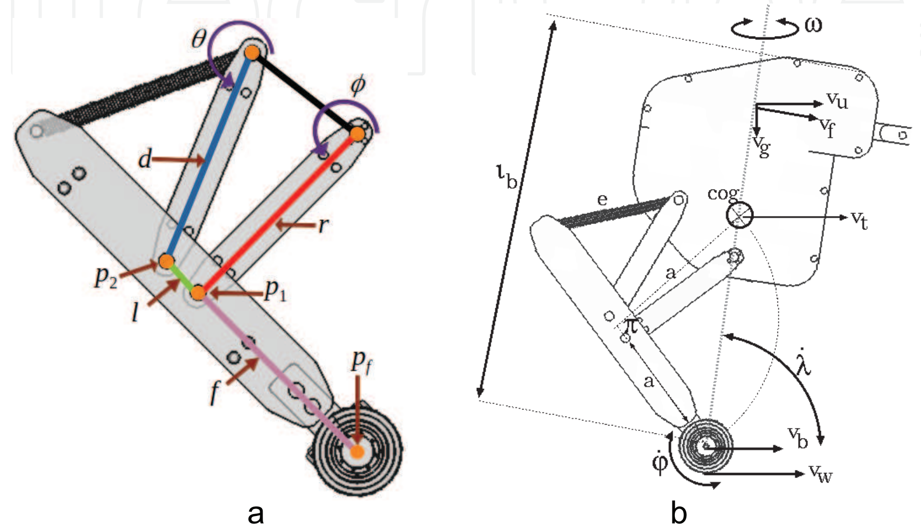

Figure 2.
Rolling biped kinematic constrains. (a) Limb's planar linkage parameters. (b) Balancing parameters.

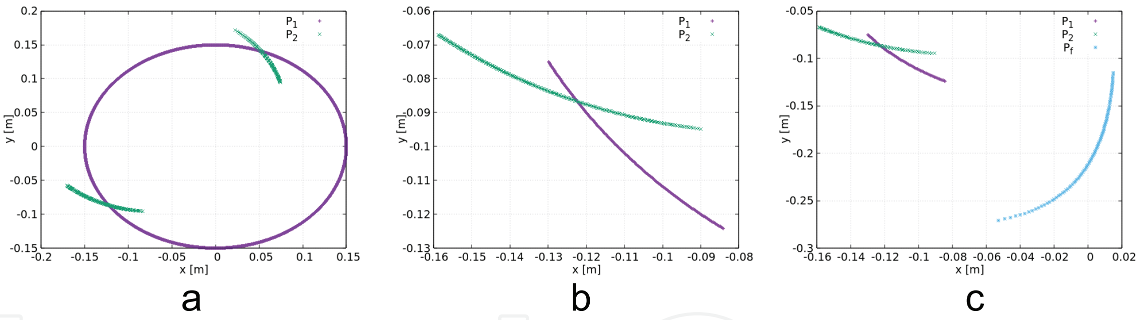

Figure 3.
Limb motion planning produced with parameters r = *0.15 m,* d = *0.15 m,* l = *0.03 m,* Δx = *−0.07 m,* Δy = *0.05387 m,* ∀201° ≤ ϕ ≤ 236°.

(both in **Figure 3**) that is projected within the biped's body. Therefore, the cog is described by Definition 2.1:

Definition 2.1 Robot's center of gravity (cog). *The robot's cog is assumed to be located at Cartesian body's position,*

$$x_{cog} = \left(\frac{x_2 - x_1}{2} + a\right) \cos\left(\pi + \arctan\left(\frac{y_2 - y_1}{x_2 - x_1}\right)\right) \tag{3}$$

and

$$y_{cog} = \left(\frac{y_2 - y_1}{2} + a\right) \sin\left(\pi + \arctan\left(\frac{y_2 - y_1}{x_2 - x_1}\right)\right). \tag{4}$$

Hence, in accordance to Definition 2.1, it follows that the robot's falling speed v_f due to vertical unbalance is

$$v_f = \sqrt[2]{\left(\frac{dx_{cog}}{dt}\right)^2 + \left(\frac{dy_{cog}}{dt}\right)^2}. \tag{5}$$

Taking into account that the robot is a dual differential drive kinematic structure, where v_R and v_L are the right-sided and left-sided velocities, respectively

$$\frac{ds}{dt} = \frac{r}{2}\left(\frac{d\varphi_R}{dt} + \frac{d\varphi_L}{dt}\right), \tag{6}$$

where s is the robot displacement over the ground, and r is the wheel's radius [m]. Likewise, the dual differential velocity is

$$v_{diff} = \frac{d\varphi_R}{dt} - \frac{d\varphi_L}{dt}. \tag{7}$$

Hence, the robot's angular velocity ω [rad/s] in terms of its differential speed v_{diff} and constrained by its wheels lateral baseline distance b_l and v_{diff} is substituted to yield:

$$\omega = \frac{2v_{diff}}{b_l} = \frac{2r_w}{b_l}\left(\frac{d\varphi_R}{dt} - \frac{d\varphi_L}{dt}\right). \tag{8}$$

The biped's falling angle and angular falling speed are λ [rad] and $\dot{\lambda}$ [rad/s], respectively. Let v_f [m/s] be the falling velocity affecting the robot's balance

$$v_f = \ell_b \dot{\lambda}_t, \tag{9}$$

which makes the robot's pitch turn around the wheel's center. Unbalancing velocity v_u [m/s] is an horizontal component such that

$$v_u = v_f \cos\left(\frac{\pi}{2} - \lambda\right) = \ell_b \dot{\lambda} \cos\left(\frac{\pi}{2} - \lambda\right). \tag{10}$$

The wheel's axis point moves at balancing speed v_b that is parallel to the ground. Wheel's tangential speed is $v_w = r\dot{\varphi}$, and linear motion speed is the balancing velocity underneath the biped's body at the wheel's axis,

$$v_b = \frac{v_w}{2} = \frac{r\dot{\varphi}_w}{2} \tag{11}$$

Therefore, the balancing condition is described by Definition 2.2:

Definition 2.2 Kinematic balancing condition. *The robot's vertical equilibrium condition is assumed for balancing and unbalancing velocities of equal magnitudes (or very approximated). Thus,*

$$v_u - v_b = 0, \tag{12}$$

and redefining, let $\dot{\varphi}_B$ be the wheel's balancing angular speed,

$$\ell_b \dot{\lambda} \cos\left(\frac{\pi}{2} - \lambda\right) - \frac{r_w \dot{\varphi}_B}{2} = 0. \tag{13}$$

Therefore, it is of interest to find the wheel's rotary speed that balances the biped motion and from Definition 2.2, the following Proposition 2.2 arises,

Proposition 2.2 Robot's balancing speed. *The wheel's rolling velocity to satisfy the robot's body balance is proposed as*

$$\dot{\varphi}_B = \frac{2\ell_b}{r_w} \dot{\lambda}_t \cos\left(\frac{\pi}{2} - \lambda\right). \tag{14}$$

and if and only if $\dot{\varphi}_R = \dot{\varphi}_L$ must exist, assuming that in such period of time, the robot's yaw $\omega_t \approxeq 0$ allowing longitudinal balancing equilibrium. Thus,

$$v_t = \begin{cases} r\dot{\varphi}_B, & \dot{\varphi}_R \approx \dot{\varphi}_L, \quad v_u - v_b \neq 0 \\ \frac{r}{2}(\dot{\varphi}_R + \dot{\varphi}_L), & v_u - v_b \approx 0 \end{cases} \tag{15}$$

Thus, let us redefine the state variables as $x_1 = \lambda$, $x_2 = \dot{x}_1$, $\dot{x}_2 = x_2^2 \tan\left(\frac{\pi}{2} - x_1\right)$. For an stability analysis for the balancing case when $v_t = r_w \dot{\varphi}_B$, and equilibrium condition $\ddot{\varphi}_B = 0$ occurs,

$$\frac{2\ell_b}{2}\left(\dot{x}_2 \cos\left(\frac{\pi}{2} - x_1\right) + \dot{x}_2{}^2 \sin\left(\frac{\pi}{2} - x_1\right)\right) = 0 \tag{16}$$

and dropping off the highest-order derivative of the system

$$\ddot{\lambda} = x_2^2 \tan\left(\frac{\pi}{2} - x_1\right) \tag{17}$$

The system total energy $E_k + E_p$ (kinetic plus potential) is a positive function, which is used as a Lyapunov candidate function, where the robot's translation motion $r_w \dot{\varphi}_B$ for equilibrium is

$$E_T = E_k + E_p = \frac{1}{2} m r_w^2 \dot{\varphi}_B^2 + mg\frac{\ell_b}{2}(1- \cos(\lambda)) \tag{18}$$

and substituting $\dot{\varphi}_B$,

$$E_T = 2m\ell_b^2\dot{\lambda}^2 \cos\left(\frac{\pi}{2}-\lambda\right) + mg\frac{\ell_b}{2}(1- \cos(\lambda)). \tag{19}$$

For $V(x) = E_T$ and finding out that v(x) and $\dot{V}(x)$ fulfill $\dot{V}(0) = 0$. In addition, assuming that $-2\pi \ll \lambda \ll 2\pi$, then $\dot{V}(0) > \in D - 0$. Assuming that $V : D \to \mathbb{R}$ is a continuous differentiable function. Finally, the following Lyapunov criterion is satisfied (20),

$$\left(\frac{\partial V(x)}{\partial x_1}, \frac{\partial V(x)}{\partial x_2}\right) \cdot (x_2, \dot{x}_2)^\top = 0 \tag{20}$$

3. Polynomial vector fields

This section introduces the proposed path planning strategy to dynamically generate local paths. Two polynomial models are enhanced as vector fields to exert attractive and repulsive robot's accelerations. In principle, the desired acceleration functions are designed from the interpolating attractive/repulsive accelerations with respect to (w.r.t.) distance. The method used to fit points is the Lagrange formula:

$$a(\delta) = \sum_i \left(\prod_j \frac{\delta - \delta_i}{\delta_i - \delta_j}\right) a_i, \tag{21}$$

where the human-user establishes a desired numerical acceleration a_i w.r.t. a desired distance δ_i, from either goal or obstacle. Therefore, the following general cubic polynomial forms attractive a_A and repulsive a_R are obtained with their respective numeric coefficients λ_i and β_i,

$$a_A(\delta) = \lambda_0 + \lambda_1\delta + \lambda_2\delta^2 + \lambda_3\delta^3, \quad a_R(\delta) = \beta_0 + \beta_1\delta + \beta_2\delta^2 + \beta_3\delta^3 \tag{22}$$

It follows that, Definition 3.1 establishes the acceleration path planning model toward a goal of interest.

Definition 3.1 Planning toward a goal. *Given the kinematic parameters* $\lambda_0 = 0$, $\lambda_1 = \frac{18}{75}[s^{-2}]$, $\lambda_2 = \frac{18}{75}[m^{-1}s^{-2}]$, $\lambda_3 = \frac{3}{75}[m^{-2}s^{-2}]$, *the attraction path model* $-\nabla_\delta \kappa f^A(\delta)$ *is defined by*

$$f^A(\delta_g) = -\nabla_\delta \frac{3}{75}\left(6\delta_g + 6\delta_g^2 - \delta_g^3\right). \tag{23}$$

Assuming that the distance between the goal and the robot $\delta_g = \sqrt[2]{x^2 + y^2}$, *such that* $x \doteq x_g - x_r$ *and* $y \doteq y_g - y_r$.

Similarly, Definition 3.2 establishes the acceleration path planning model that avoids obstacles zones.

Definition 3.2 Planning obstacles avoidance. *Given the kinematic parameters* $\beta_0 = 5$, $\beta_1 = \frac{109}{440}$, $\beta_2 = -\frac{18}{55}[m^{-1}s^{-2}]$, $\beta_3 = \frac{3}{110}[m^{-2}s^{-2}]$, *the avoidance path model* $-\nabla_\delta f^R(\delta)$ *is defined by*

$$f^R(\delta_o) = -\nabla_\delta \kappa_R \left(5 + \frac{109}{440}\delta_o - \frac{18}{55}\delta_o^2 + \frac{3}{11}\delta_o^3\right), \tag{24}$$

Assuming that the distance between the obstacle and the robot $\delta_0 = \sqrt[2]{x^2+y^2}$, *such that* $x \doteq x_o - x_r$ *and* $y \doteq y_o - y_r$.

Figure 4(a) shows the robot's instantaneous acceleration toward a goal of interest. The attraction acceleration starts when the goal-robot distance $\delta_g < 9$m. The planner allows start from $f^A \approxeq 0$ to realistically provide speeds physically possible. **Figure 4(b)** shows the robot's instantaneous acceleration away from obstacles. The avoidance acceleration starts when the obstacle-robot distance $\delta_o < 8$m. This avoidance planner f^R is faster than f^A to increase confidence against obstacles.

Therefore, extending to two-dimension Cartesian space, let us deduce the following algebraic process for the goal-attraction planner f^A:

$$\frac{\partial f^A}{\partial x} = -\kappa_A \left(\frac{6x}{\sqrt[2]{x^2+y^2}} + 12x - \frac{3x}{2}\sqrt[2]{x^2+y^2}\right), \tag{25}$$

as well as

$$\frac{\partial f^A}{\partial y} = -\kappa_A \left(\frac{6y}{\sqrt[2]{x^2+y^2}} + 12y - \frac{3y}{2}\sqrt[2]{x^2+y^2}\right). \tag{26}$$

Similarly, for the robot's acceleration to avoid obstacles, let us develop the following f^R, by substituting the functional form of δ_o into the gradient function

$$f^R(x,y) = -\nabla \kappa_R \left(5 + \frac{109}{440}\sqrt[2]{x^2+y^2} - \frac{18}{55}(x^2+y^2) + \frac{3}{110}(x^2+y^2)^{3/2}\right), \tag{27}$$

subsequently by applying the gradient operator,

$$\frac{\partial f^R}{\partial x} = -\frac{109x}{440\sqrt[2]{x^2+y^2}} + \frac{18x}{55} - \frac{9x}{110}\sqrt[2]{x^2+y^2} \tag{28}$$

and

$$\frac{\partial f^R}{\partial y} = -\frac{109y}{440\sqrt[2]{x^2+y^2}} + \frac{18y}{55} - \frac{9y}{110}\sqrt[2]{x^2+y^2} \tag{30}$$

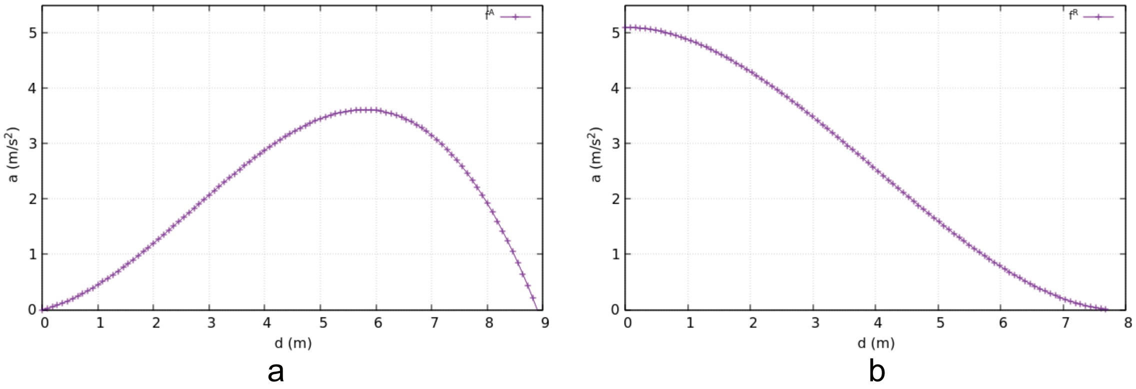

Figure 4.
Navigational robot's motion planning, desired acceleration w.r.t. distance. (a) Goal attractive motion. (b) Obstacle avoidance motion.

Thus, by substituting terms in both local planners, the general attractive motion planning is a negative function with amplitude coefficient $\kappa_A = \frac{3}{75}\left[s^{-2}\right]$ is represented by

$$\begin{pmatrix} \dfrac{\partial f^A}{\partial x} \\ \dfrac{\partial f^A}{\partial y} \end{pmatrix} = -\kappa_A \left(\frac{6}{\delta_t} + 12 - \frac{3}{2}\delta_g \right) \begin{pmatrix} x_g - x_t \\ y_g - y_t \end{pmatrix} \tag{31}$$

The attractive field is a positive function working within limits $0 \le \delta_g < 9$, thus

$$\lim_{\delta_g \to 9} f^R\left(\delta_g\right) = 0; \quad \text{and} \quad \lim_{\delta_g \to 6} f^R\left(\delta_g\right) = 3.75 \tag{32}$$

and the repulsive with $\kappa_R = \frac{1}{440}\left[s^{-2}\right]$

$$\begin{pmatrix} \dfrac{\partial f^R}{\partial x} \\ \dfrac{\partial f^R}{\partial y} \end{pmatrix} = -\kappa_R \left(\frac{109}{\delta_o} - 288 + 36\delta_o \right) \begin{pmatrix} x_o - x_t \\ y_o - y_t \end{pmatrix}. \tag{33}$$

The repulsive field is a negative function that works within range $0 \le \delta_o \le 8$,

$$\lim_{\delta_o \to 8} f^R(\delta_o) = 0; \quad \text{and} \quad \lim_{\delta_o \to 0} f^R(\delta_o) = 5. \tag{34}$$

The direction fields produced are shown in **Figure 5**, where for both cases the coordinates origin represents either goal or obstacle locations.

Moreover, when neither goals of interest nor obstacles are within the observation field of the robot, it must keep navigating along a prior route plan. The routing plan is map comprised of a sequence of Cartesian points $\mathbf{g}_k \in \mathbb{R}^2$, $\mathbf{g}_k = (x, y)^\top$.

When the robot accomplishes either f^A or f^R, it continues toward the following route point, expressed in terms of unit vectors:

$$\mathbf{a}_t^o = a^o \frac{\mathbf{g}_{k+1} - \mathbf{g}_k}{\left\|\mathbf{g}_{k+1} - \mathbf{g}_k\right\|}, \tag{35}$$

where a^o is an ideal or averaged acceleration toward the next route point $\mathbf{g}_2$ from $\mathbf{g}_1$. Therefore, the total path planner mode is given by Proposition 3.1.

Proposition 3.1 Total mission path planning. *The total path planning model subjected to adaptive environmental changes is proposed as a vector fields sum:*

$$\mathbf{a}_t = \mathbf{a}_t^o + \sum_g \left[\kappa_A \left(\frac{3\delta_g}{2} - \frac{6}{\delta_g} - 12 \right) \begin{pmatrix} x_g - x_t \\ y_g - y_t \end{pmatrix} + \sum_o \kappa_R \left(288 - \frac{109}{\delta_o} - 36\delta_o \right) \begin{pmatrix} x_0 - x_t \\ y_o - y_t \end{pmatrix} \right] \tag{36}$$

and the robot's instantaneous navigational total velocity $\mathbf{v}_T \in \mathbb{R}^2$, $\mathbf{v}_T = \left(x_T, \dot{y}_T\right)^\top$ is obtained by

$$\mathbf{v}_T = \int_{t_1}^{t_2} \mathbf{a}_t \mathrm{d}t. \tag{37}$$

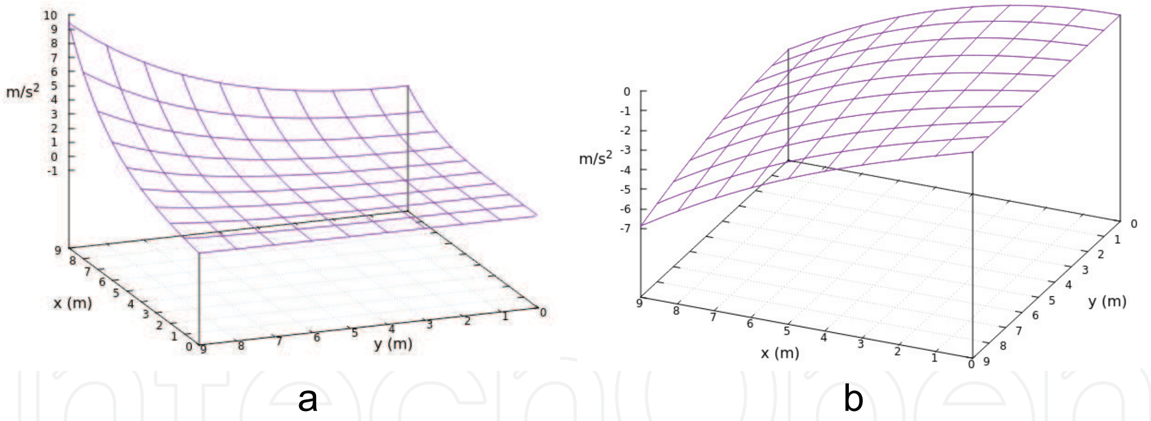

Figure 5.
Acceleration fields w.r.t. variations along the plane xy. *(a) Goals attractive motion generation. (b) Obstacles avoidance motion generation.*

Thus, to automatically limit the speed until the robot's maximal velocity $v_{\max}$, the real physical velocity v_{ph} is constrained by the final planning motion

$$v_{ph}\left(\frac{\|\mathbf{v}_T\|}{v_{max}}\right) = \begin{cases} \dfrac{\|\mathbf{v}_T\|}{v_{max}}, & \dfrac{\|\mathbf{v}_T\|}{v^{max}} < 1 \\ v_{max}, & \dfrac{\|\mathbf{v}_T\|}{v^{max}} \geq 1 \end{cases} \tag{38}$$

Figure 6 shows simulation results of the total navigation planner.

4. Path tracking

This section deduces an algebraic trajectory tracking linearized controller based on the kinematics geometry of the reference path: linear and angular velocities. Considering from previous Section 3, let v_{ph} be called the reference velocity that is going to be tracked. Likewise, let $(x_{1,2}, y_{1,2})$ be the next local planning coordinates to be reached (from location 1 to location 2). Hence, let us define $x_{1,2} \doteq x_2 - x_1$ and $y_{1,2} \doteq y_2 - y_1$. Such that $\dot{s}_{1,2}$ [m/s] is a segment of the planning speed. Hence, let us obtain the first-order derivative,

$$\dot{s}_{1,2} = \frac{d}{dt}\sqrt[2]{x_{1,2}^2 + y_{1,2}^2} = \frac{x_{1,2}\dot{x}_{1,2} + y_{1,2}\dot{y}_{1,2}}{\sqrt[2]{x_{1,2}^2 + y_{1,2}^2}}. \tag{39}$$

Similarly, by obtaining the desired robot's orientation $\theta_{1,2}$, its first-order derivative w.r.t. time is

$$\dot{\theta}_{1,2} = \frac{d}{dt}\left(\arctan\left(\frac{y_{1,2}}{x_{1,2}}\right)\right) = \frac{x_{1,2}\dot{y}_{1,2} - y_{1,2}\dot{x}_{1,2}}{x_{1,2}^2 + y_{1,2}^2}. \tag{40}$$

It follows that, the tracking control vector is $\dot{\mathbf{u}}_t \in \mathbb{R}^2$, such that $\dot{\mathbf{u}}_t = \left(\dot{s}_{1,2}, \dot{\theta}_{1,2}\right)^\top$. Therefore, by stating (39) and (40) as a system of linear equations, the first-order derivatives must simultaneously be solved:

$$\begin{aligned} \frac{x_{1,2}}{s_{1,2}}\dot{x}_{1,2} + \frac{y_{1,2}}{s_{1,2}}\dot{y}_{1,2} &= \dot{s} \\ -\frac{y_{1,2}}{s_{1,2}^2}\dot{x}_{1,2} + \frac{x_{1,2}}{s_{1,2}^2}\dot{y}_{1,2} &= \dot{\theta} \end{aligned}, \tag{41}$$

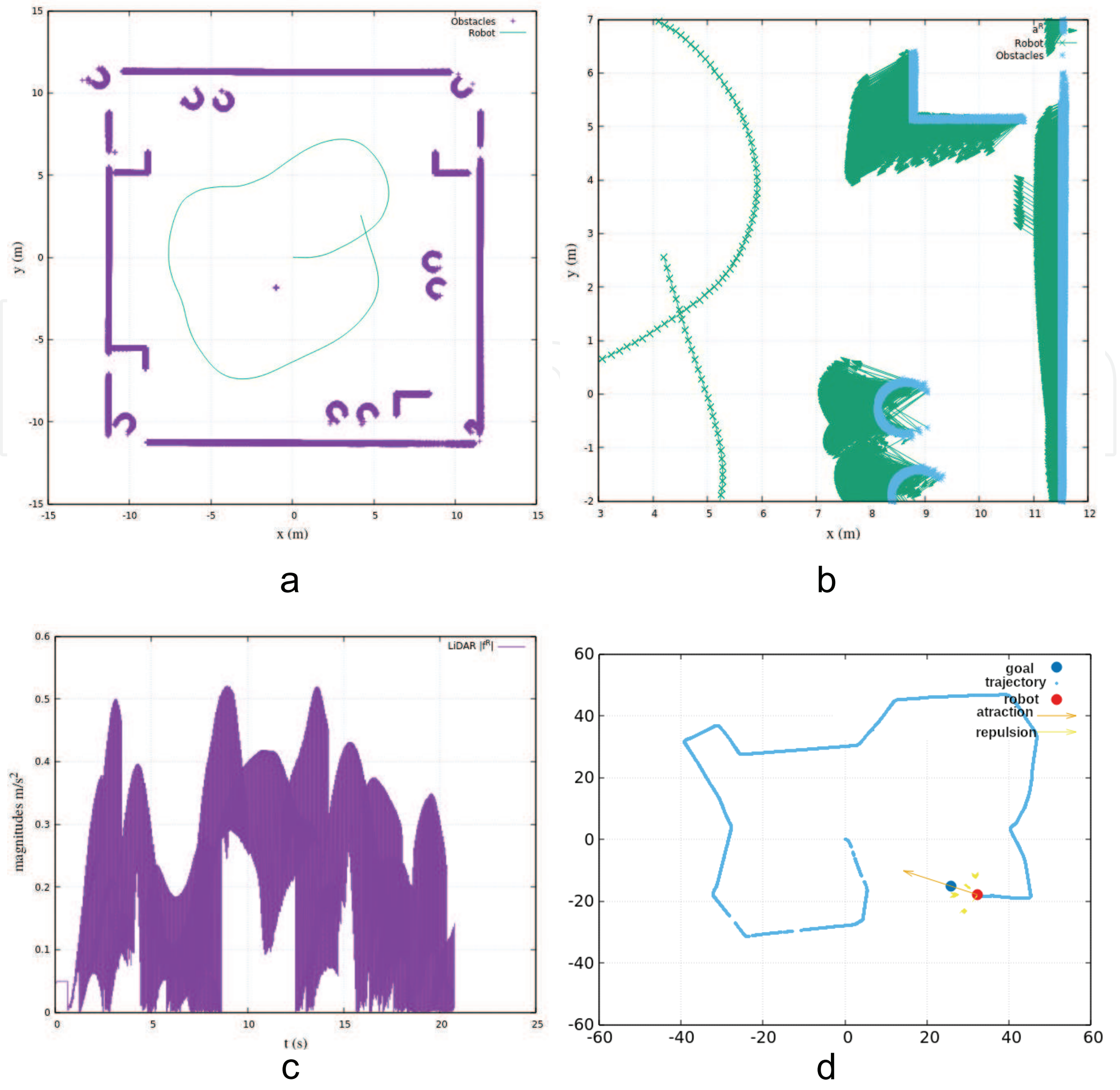

Figure 6.
Vector fields path generation. (a) High-density obstacles yielding robot's repulsion. (b) Obstacles repulsive field. (c) High density $\left\|\mathbf{f}^R\right\|$. *(d) Attractive goals along a route.*

thus, by factorizing both, the common term $1/s_{1,2}$ and the first-order derivatives, the matrix form of the forward tracking law is

$$\dot{\mathbf{u}} = \frac{1}{s_{1,2}} \begin{pmatrix} x_{1,2} & y_{1,2} \\ -\frac{y_{1,2}}{s_{1,2}} & \frac{x_{1,2}}{s_{1,2}} \end{pmatrix} \cdot \begin{pmatrix} \dot{x}_{1,2} \\ \dot{y}_{1,2} \end{pmatrix}. \tag{42}$$

Likewise, as the inverse tracking law is of our interest, it is inversely dropped off

$$\begin{pmatrix} \dot{x}_{1,2} \\ \dot{y}_{1,2} \end{pmatrix} = s_{1,2} \begin{pmatrix} x_{1,2} & y_{1,2} \\ -\frac{y_{1,2}}{s_{1,2}} & \frac{x_{1,2}}{s_{1,2}} \end{pmatrix}^{-1} \cdot \dot{\mathbf{u}}. \tag{43}$$

For notation simplicity, let us redefine $\dot{\zeta}_t \doteq \left(\dot{x}_{1,2}, \dot{y}_{1,2}\right)^\top$ and the time-variant control matrix $\mathbf{K}_t$ as

$$\mathbf{K}_t = \begin{pmatrix} x_{1,2} & y_{1,2} \\ -\frac{y_{1,2}}{s_{1,2}} & \frac{x_{1,2}}{s_{1,2}} \end{pmatrix}, \tag{44}$$

Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value
l [m]	0.03	a [m]	0.25	$\kappa_A [s^2]$	3/75	I_x [kg^2]	0.12
d [m]	0.15	b_l [m]	0.4	$\kappa_R [s^{-2}]$	1/440	I_y [kg^2]	0.12
e[a] [m]	0.08	$\ell_b \, [m]$	0.75	a^o [m/s^2]	0.5	I_z [kg^2]	0.13
r [m]	0.15	m [kg]	0.76	r_w [m]	0.1	$v_{max} \, [m/s]$	1.5

[a] *High stiffness non-rigid bar.*

Table 1.
Simulation parameters deployed with high-density range scanner Hokuyo URG 04LX.

thus from Section 3, let $\boldsymbol{u}^{ref}$ be the global reference planning model during time segment $t_{1,2} \doteq t_2 - t_1$ such that,

$$\boldsymbol{u}^{ref} = \begin{pmatrix} v_{ph} \cdot t_{1,2} \\ \arctan\left(\frac{\dot{y}_T}{\dot{x}_T}\right) \end{pmatrix}. \tag{45}$$

Therefore, the recursive path tracking control law is stated as

$$\begin{array}{lrcl} 1: & \zeta_{t+1} & = & \zeta_t + s_{1,2}\mathbf{K}_t^{-1} \cdot \left(\boldsymbol{u}^{ref} - \hat{\mathbf{u}}_t\right) \\ 2: & \mathbf{u}_{t+1} & = & \mathbf{u}_t + \frac{1}{s_{1,2}}\mathbf{K}_t \cdot \left(\zeta_{t+1} - \hat{\zeta}_t\right) \end{array} \tag{46}$$

where $\hat{\mathbf{u}}_t$ is the instantaneous sensors observation of both components, displacement $\hat{s}$ and yaw $\hat{\theta}$. The prediction control speed vector ζ_{t+1} is the next desired local reference in line 2, while $\hat{\zeta}_t$ is the robot's Cartesian speed observer vector. Finally, $\mathbf{u}_{t\ +\ 1}$ is the control vector global prediction (**Table 1**).

5. Conclusion

The mechanical design of the biped's lower-limb mechanical structure was configured as a double-crank four-bar linkage with passive-allowed motions. Motion planning began from determining the limbs' linkage positions causing the robot's height and pitch varying overtime producing unbalanced motions. Inferring balancing velocities to yield robot's vertical balance was possible and worked stable. The proposed balancing rolling condition was analyzed throughout its total energy model as a Lyapunov candidate function resulting stable in three criteria: $\mathbf{V}(0) = 0$, $v(x) = 0$, and $\dot{V}(x) = 0$. The proposed navigational approach allowed human-user to design short range-limited trajectories by cubic polynomials obtained from four empirical coordinates distance-acceleration by Lagrange interpolation. Both polynomial planning models have initial conditions $a_0(t_0) = 0[m/s^2]$. Thus, the robot did not require an infinite acceleration to reach a desired speed at $\mathbf{t}_0$. Physically, the model is applied to any motion in equilibrium. For goal attraction the maximal acceleration is reached at the 55% of the distance, subsequently decreases monotonically until the goal position.

The navigational general planning model is a set of partial derivatives model combined, allowing dynamic local planning among multiple obstacles, goals, and routes. The navigational general planning model worked as the reference model for the tracking control system. It consisted of a set of time-varying linear equations, with recursive feedback showing suitable performance. The work was implemented in simulation and coded in C++ under Linux. Future improvements will consider the inclusion of kinetic models and dynamic forces fitted to the proposed planer.

Appendix A

This appendix provides the algebraic deduction of the limb's Cartesian motion model described in Section 2.

Let $\mathbf{P}_1 = (x_1, y_1)^\top$ be the Cartesian point of passive joint between links b and ℓ, obtained by

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = r \begin{pmatrix} \cos(\phi) \\ \sin(\phi) \end{pmatrix}. \tag{47}$$

Likewise, let $\mathbf{P}_2 = (x_2, y_2)^\top$ be the coordinate of passive joint between links r and ℓ, expressed by:

$$\begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = r \begin{pmatrix} \cos(\phi) \\ \sin(\phi) \end{pmatrix} + \begin{pmatrix} \sqrt{l^2 - (y_1 - y_2)^2} \\ \sqrt{l^2 - (x_1 - x_2)^2} \end{pmatrix}. \tag{48}$$

Both expressions (47) and (48) are described in terms of ϕ, but can similarly be described in terms of the angle θ by

$$\begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = d \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix} + \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}, \tag{49}$$

where the distances Δx and Δy separate the joints *phi* and *theta* along the chassis, see **Figure 2(a)**. By algebraically treating the Cartesian components separately, let us equal the x components of (48) and (49) to form the equation

$$d\cos(\theta) + \Delta x = r\cos(\phi) + \sqrt{l^2 - (y_1 - y_2)^2}, \tag{50}$$

dropping off the squared root term of (50),

$$\sqrt{l^2 - (y_1 - y_2)^2} = d\cos(\theta) + \Delta x - r\cos(\phi) \tag{51}$$

and squaring both sides of the equality, the following is obtained,

$$l^2 - (y_1 - y_2)^2 = [d\cos(\theta) + \Delta x - r\cos(\phi)]^2 \tag{52}$$

and algebraically expanding the squared binomials and substituting y_1 and y_2 from expression (47) and (49) respectively into (52),

$$\begin{aligned} & d^2\cos^2(\theta) + 2\Delta x d\cos(\theta) - 2rd\cos(\theta)\cos(\phi) + (\Delta x)^2 \\ & \quad -2\Delta x r\cos(\phi) + r^2\cos^2(\phi) \\ & \quad = l^2 - r^2\sin^2(\phi) + 2rd\sin(\theta)\sin(\phi) + 2\Delta y r\sin(\phi) \\ & \qquad -d^2\sin^2(\theta) - 2\Delta y d\sin(\theta) - (\Delta y)^2 \end{aligned} \tag{53}$$

and organizing trigonometric terms by their degree, Eq. (53) becomes:

$$\begin{aligned}
&2rd\cos(\theta)\cos(\phi) + 2rd\sin(\theta)\sin(\phi)\\
&\quad = d^2\cos^2(\theta) + d^2\sin^2(\theta) + (\Delta x)^2 + (\Delta y)^2\\
&\qquad + r^2\cos^2(\phi) + r^2\sin^2(\phi) + 2\Delta x d\cos(\theta) - 2\Delta x r\cos(\phi)\\
&\qquad + 2\Delta y d\sin(\theta) - 2\Delta y r\sin(\phi).
\end{aligned} \tag{54}$$

By factorizing the common terms of (47) and exchanging terms with suitable identities such as $\cos^2(\alpha) + \sin^2(\alpha) = 1$ and $c^2 = (\Delta x)^2 + (\Delta y)^2$ in order to obtain Eq. (55)

$$\begin{aligned}
&\cos(\theta)\cos(\phi) + \sin(\theta)\sin(\phi) =\\
&\frac{r^2 + d^2 + c^2 - l^2 - 2r[\Delta x\cos\phi + \Delta y\sin(\phi)] + 2d\Delta x\cos(\theta) + 2d\Delta y\sin(\theta)}{2rd}
\end{aligned} \tag{55}$$

In order to simplify the main expression, some constant terms are rewritten as

$$K_1 = \frac{r^2 + d^2 + c^2 - l^2}{2rd}, \quad K_2 - \frac{1}{d}, \quad K_3 = \frac{\Delta x}{r}, \quad K_4 = \frac{\Delta y}{r}. \tag{56}$$

Therefore, the following simplified algebraic expression is deduced:

$$\begin{aligned}
&\cos(\theta)\cos(\phi) + \sin(\theta)\sin(\phi) = K_1 + K_2[\Delta x\cos(\phi) + \Delta y\sin(\phi)]\\
&\quad + K_3\cos(\theta) + K_4\sin(\theta).
\end{aligned} \tag{57}$$

The trigonometric identity forms $\sin(\theta) = \frac{2\tan\left(\frac{\theta}{2}\right)}{1+\tan^2\left(\frac{\theta}{2}\right)}$ and $\cos(\theta) = \frac{1-\tan^2\left(\frac{\theta}{2}\right)}{1+\tan^2\left(\frac{\theta}{2}\right)}$ are substituted into (57) to produce (58),

$$\begin{aligned}
&\cos(\phi)\left[\frac{1-\tan^2\left(\frac{\theta}{2}\right)}{1+\tan^2\left(\frac{\theta}{2}\right)}\right] + \sin(\phi)\left[\frac{2\tan\left(\frac{\theta}{2}\right)}{1+\tan^2\left(\frac{\theta}{2}\right)}\right] = K_1 + K_2[\Delta x\cos(\phi) + \Delta y\sin(\phi)]\\
&+K_3\left[\frac{1-\tan^2\left(\frac{\theta}{2}\right)}{1+\tan^2\left(\frac{\theta}{2}\right)}\right] + K_4\left[\frac{2\tan\left(\frac{\theta}{2}\right)}{1+\tan^2\left(\frac{\theta}{2}\right)}\right]
\end{aligned} \tag{58}$$

and by multiplying both sides of the equality (57) by common denominator $1 + \tan^2\left(\frac{\theta}{2}\right)$ and algebraically simplifying:

$$\begin{aligned}
&\cos(\phi) - \cos(\phi)\tan^2\left(\frac{\theta}{2}\right) + 2\sin(\phi)\tan\left(\frac{\theta}{2}\right) = K_1 + K_1\tan^2\left(\frac{\theta}{2}\right)\\
&\quad + K_2[\Delta x\cos(\phi) + \Delta y\sin(\phi)] + K_2[\Delta x\cos(\phi) + \Delta y\sin(\phi)]\tan^2\left(\frac{\theta}{2}\right)\\
&\quad + K_3 - K_3\tan^2\left(\frac{\theta}{2}\right) + 2K_4\tan\left(\frac{\theta}{2}\right).
\end{aligned} \tag{59}$$

Hereafter, to find a root for θ as independent variable, common terms are factorized and Eq. (59) is equated to zero to obtain

$$\begin{aligned} &\tan^2\left(\frac{\theta}{2}\right)[K_1 + K_2[\Delta x \cos(\phi) + \Delta y \sin(\phi)] - K_3 + \cos(\phi)] \\ &+\tan\left(\frac{\theta}{2}\right)[2K_4 - 2\sin(\phi)] + K_1 + K_2[\Delta x \cos(\phi) + \Delta y \sin(\phi)] \\ &+K_3 - \cos(\phi) = 0, \end{aligned} \tag{60}$$

in order to reduce expression complexity, the following Definitions are stated:

Definition 4.1 [term P]. Let P be defined as a Cartesian horizontal decreasing segment by expression:

$$P = K_1 + K_2[\Delta x \cos(\phi) + \Delta y \sin(\phi)] - K_3 + \cos(\phi). \tag{61}$$

Definition 4.2 [term Q]. Let Q be defined as a Cartesian vertical segment that is expressed by:

$$Q = 2K_4 - 2\sin(\phi) \tag{62}$$

and

Definition 4.3 [term P]. Let R be defined as a Cartesian horizontal increasing segment by expression:

$$R = K_1 + K_2[\Delta x \cos(\phi) + \Delta y \sin(\phi)] + K_3 - \cos(\phi) \tag{63}$$

Therefore, by substituting the terms of Definitions 4.1–4.3 into Eq. (60), the following quadratic form is deduced,

$$P\tan^2\left(\frac{\theta}{2}\right) + Q\tan\left(\frac{\theta}{2}\right) + R = 0. \tag{64}$$

By analytically solving (64), which is a second-degree equation using the general form, a real solution for θ is possible, thus

$$\theta = 2\arctan\left(\frac{-Q \pm \sqrt{Q^2 - 4PR}}{2P}\right) \tag{65}$$

Author details

Santiago de J. Favela Ortíz[†] and Edgar A. Martínez García[*†]
Laboratorio de Robótica, Instituto de Ingeniería y Tecnología, Universidad Autónoma de Ciudad Juárez, Chihuahua, Mexico

[*] Address all correspondence to: edmartin@uacj.mx

[†] These authors contributed equally.

References

[1] Azzabi A, Nouri K. Path planning for autonomous mobile robot using the potential field method. In: Proceedings of International Conference on Advanced Systems and Electric Technologies (IC_ASET 2017). 2017. pp. 389-394

[2] Reyes Muñoz JU, Martínez-García EA, Rodriguez Jorge R, Torres-Córdoba R. WMR kinematic control using underactuated mechanisms for goal-direction and evasion. In: Gorrostieta E, editor. Kinematics. IntechOpen; 2017. pp. 147-169. DOI: 10.5772/intechopen.70811

[3] Castro-Jiménez LE, Martínez-García EA. Thermal image sensing model for robotic planning and search. Sensors. 2016;**16**(8):1253

[4] Norouzi M, Valls Miro J, Dissanayake G. Probabilistic stable motion planning with stability uncertainty for articulated vehicles on challenging terrains. Autonomous Robots. 2016;**40**(2):361-381

[5] Kamalova A, Kim KD, Lee SG. Waypoint mobile robot exploration based on biologically inspired algorithms. IEEE Access. 2020;**8**: 190342-190355

[6] Ma L, Xue J, Kawabata K, Zhu J, Ma C, Zheng N. Efficient sampling-based motion planning for on-road autonomous driving. IEEE Transactions on Intelligent Transportation Systems. 2015;**16**(4):1961-1976

[7] Bai J, Lian S, Liu Z, Wang K, Liu D. Deep learning based robot for automatically picking up garbage on the grass. IEEE Transactions on Consumer Electronics. 2018;**64**(3):382-389. DOI: 10.1109/TCE.2018.2859629

[8] Fang B, Ding J, Wang Z. Autonomous robotic exploration based on frontier point optimization and multistep path planning. IEEE Access. 2019;**7**:46104-46113

[9] Lu C, Shan L, Jiang C, Dai Y. Reinforcement based mobile robot path planning with improved dynamic window approach in unknown environment. Autonomous Robots. 2021;**45**(1):51-76

[10] Zakiev A, Lavrenov R, Magid E, Svinin M, Matsuno F. Partially unknown environment exploration algorithm for a mobile robot. Journal of Advanced Research in Dynamical and Control Systems. 2019;**11**(8):1743-1753

[11] Yoon HS, Park TH. Motion planning of autonomous mobile robots by iterative dynamic programming. Intelligent Service Robotics. 2015;**8**(3): 165-174

[12] Qureshi AH, Ayaz Y. Potential functions based sampling heuristic for optimal path planning. Autonomous Robots. 2016;**40**(6):1079-1093

[13] Receveur JB, Victor S, Melchior P. Autonomous car decision making and trajectory tracking based on genetic algorithms and fractional potential fields. Intelligent Service Robotics. 2020;**13**(2):315-330

[14] Ferri G, Manzi A, Salvini P, Mazzolai B, Laschi C, Dario P. DustCart, an autonomous robot for door-to-door garbage collection: From DustBot project to the experimentation in the small town of Peccioli. In: 2011 IEEE International Conference on Robotics and Automation; 9–13 May 2011; Shanghai, China: IEEE; 2011. pp. 655-660. DOI:10.1109/ICRA.2011.5980254

[15] Han SI, Lee JM. Balancing and velocity control of a unicycle robot based on the dynamic model. IEEE Transactions on Industrial Electronics. 2015;**62**(1):405-413

[16] Klemm V, Morra A, Gulich L, Mannhart D, Rohr D, Kamel M, et al. LQR-assisted whole-body control of a wheeled bipedal robot with kinematic loops. IEEE Robotics and Automation Letters. 2020;**5**(2):3745-3752

[17] Li CG, Zhou L, Chao Y. Self-balancing two-wheeled robot featuring intelligent end-to-end deep visual-steering. IEEE/ASME Transactions on Mechatronics. 2021; **26**(5):2263-2273

[18] Jiang L, Qiu H, Wu Z, He J. Active disturbance rejection control based on adaptive differential evolution for two-wheeled self-balancing robot. Proceedings of the 28th Chinese Control Decision Conference (CCDC 2016). IEEE; 2016. pp. 6761-6766

[19] Chen X, Zhao H, Sun H, Zhen S. Adaptive robust control based on Moore-Penrose generalized inverse for underactuated mechanical systems. IEEE Access. 2021;**7**:157136-157144

[20] Iwendi C, Alquarni MA, Anajemba JH, Alfakeeh AS, Zhang Z, Bashir AK. Robust navigational control of a two-wheeled self-balancing robot in a sensed environment. IEEE Access. 2019;**7**:82337-82348

[21] Zhang C, Liu T, Song S, Meng MQH. System design and balance control of a bipedal leg-wheeled robot. International Conference on Robotics and Biomimetics (ROBIO 2019). IEEE; 2019. pp. 1869-1874

[22] Kim S, Kwon SJ. Dynamic modeling of a two-wheeled inverted pendulum balancing mobile robot. International Journal of Control, Automation and Systems. 2015;**13**(4):926-933

[23] Yue M, An C, Sun JZ. An efficient model predictive control for trajectory tracking of wheeled inverted pendulum vehicles with various physical constraints. International Journal of Control, Automation and Systems. 2018; **16**(1):265-274

[24] Tayefi M, Gen Z. Self-balancing controlled Lagrangian and geometric control of unmanned mobile robots. Journal of Intelligent and Robotic Systems: Theory and Applications. 2018; **90**(1–2):253-265

[25] Esmaeili N, Alfi A, Khosravi H. Balancing and trajectory tracking of two-wheeled mobile robot using backstepping sliding mode control: Design and experiments. Journal of Intelligent and Robotic Systems: Theory and Applications. 2017;**87**(3–4):601-613

[26] Sato R, Miyamoto I, Sato K, Ming A, Shimojo M. Development of Robot legs inspired by bi-articular muscle-tendon complex of cats. International Conference on Intelligent Robots and Systems (IROS). IEEE; 2015